X-ray simulations with gVXR as a useful tool for education, data analysis, set-up of CT scans, and scanner development

Franck P. Vidal[a,b], Shaghayegh Afshari[c], Sharif Ahmed[d], Carolyn Atkins[e], Éric Béchet[f], Alberto Corbí Bellot[g], Stefan Bosse[h,i], Younes Chahid[e], Cheng-Ying Chou[c], Robert Culver[j], Lewis Dixon[b], Martí Puig Fantauzzi[k,l], Johan Friemann[m], Amin Garbout[l], Clémentine Hatton[n], Audrey Henry[n], Christophe Leblanc[f], Alberto Leonardi[d], Jean Michel Létang[o], Harry Lipscomb[l], Tristan Manchester[d], Bas Meere[p], Simon Middleburgh[b], Iwan Mitchell[b], Liam Perera[d], and Jenna Tugwell-Allsup[q]

[a]Scientific Computing, Ada Lovelace Centre, Science Technology Facilities Council, UK
[b]School of Computer Science & Engineering, Bangor University, UK
[c]Department of Biomechatronics Engineering, National Taiwan University, Taiwan
[d]DIAD beamline, Diamond Light Source, UK
[e]UK Astronomy Technology Centre, Royal Observatory, Edinburgh, UK
[f]Département d'Aérospatiale et Mécanique, Université de Liège, Belgium
[g]Escuela Superior de Ingeniería y Tecnología - Universidad Internacional de La Rioja, Spain
[h]Department of Computer Science, University of Koblenz, Koblenz, Germany
[i]Department of Mechanical Engineering, University of Siegen, Siegen, Germany
[j]The Manufacturing Technology Centre, UK
[k]Department of Engineering Science, University of Oxford, UK
[l]Henry Royce Institute, Henry Moseley X-ray Imaging Facility, Department of Materials, The University of Manchester, UK
[m]Department of Industrial and Materials Science, Chalmers University of Technology, Sweden
[n]Scalian DS, Rennes, France
[o]INSA-Lyon, Université Claude Bernard Lyon 1, CNRS, Inserm, CREATIS UMR 5220, U1294, Lyon, France
[p]Department of Mechanical Engineering, Eindhoven University of Technology, Netherlands
[q]Radiology Department, Betsi Cadwaladr University Health Board (BCUHB), Ysbyty Gwynedd, UK

ABSTRACT

gVirtualXray (gVXR) is an open-source framework that relies on the Beer-Lambert law to simulate X-ray images in real time on a graphics processor unit (GPU) using triangular meshes. A wide range of programming languages is supported (C/C++, Python, R, Ruby, Tcl, C#, Java, and GNU Octave). Simulations generated with gVXR have been benchmarked with clinically realistic phantoms (i.e. complex structures and materials) using Monte Carlo (MC) simulations, real radiographs and real digitally reconstructed radiographs (DRRs), and X-ray computed tomography (CT). It has been used in a wide range of applications, including real-time medical simulators, proposing a new densitometric radiographic modality in clinical imaging, studying noise removal techniques in fluoroscopy, teaching particle physics and X-ray imaging to undergraduate students in engineering, and XCT to masters students, predicting image quality and artifacts in material science, etc. gVXR has also been used to produce a high number of realistic simulated images in optimization problems and to train machine learning algorithms. This paper presents applications of gVXR related to XCT.

Keywords: X-ray imaging, Computed tomography, Simulation, GPU programming, Digital twinning

Further author information: (Send correspondence to F.P.V.)
F.P.V.: E-mail: franck.vidal@stfc.ac.uk

1. INTRODUCTION

The simulation of accurate and fast X-ray images remains a challenge. State-of-the-art Monte Carlo (MC) methods can mimic the physics, by tracking photons as they travel from the source, through matter, to the detector. The computational cost makes it prohibitive in many applications where speed is a requirement, e.g. interactive virtual reality (VR) or high data throughput support. However, it is possible to trade off some of the physical effects such as scattering to speed-up computations, whilst retaining a high level of accuracy.

In this paper, we describe an open-source framework called gVirtualXray (gVXR) and show how it has been deployed in various scientific contexts. No proprietary technology is used, making it portable and deployable on a wide range of hardware and software platforms. gVXR implements a deterministic simulation model based on the Beer-Lambert law to generate noise-free images. They can provide a good compromise between speed and accuracy[1] and can be implemented on graphics processor units (GPUs) for a further increase of speed.[2,3] Unlike Monte Carlo methods, deterministic simulations tend to ignore scattering and noise. In gVXR, the latter is added as a post-process.

2. DESCRIPTION

gVXR is an open-source application programming interface (API) written in C++ to compute the Beer-Lambert law, also known as the attenuation law. If scattering is neglected and an ideal (i.e. Dirac) point-spread function is assumed, X-ray projections $\mathbf{I}(x, y)$ can simply be modeled with the Beer-Lambert attenuation law:

$$\mathbf{I}(x, y) = \sum_i \mathbf{R}(E_i)\, \mathbf{D}(E_i)\ \exp\left(-\sum_j \mu_j(E_i)\ \mathbf{d}_j(x, y)\right) \tag{1}$$

$\mathbf{I}(x, y)$ is the integrated energy in electronvolt (eV), keV or MeV, units of energy commonly used in atomic and nuclear physics, received by pixel (x, y). The beam spectrum emitted by the X-ray source is discretized in several energy channels in the polychromatic case. E_i corresponds to the energy of the i-th energy channel. $\mathbf{D}(E_i)$ is the number of photons emitted by the source at that energy E_i. When the source is monochromatic, e.g. in the case of synchrotron radiation, a single energy channel is used. The detector response $\mathbf{R}(E_i)$ mimics the use of a scintillator by replacing the incident energy E_i with a smaller value, i.e. $\mathbf{R}(E_i) < E_i$. The detector response is assumed space-invariant in Equation 1. j indicates the j-th material being scanned when a multi-material "object" is considered. $\mu_j(E_i)$ is the linear attenuation coefficient in cm^{-1} of the j-th material at energy E_i. $\mathbf{d}_j(x, y)$ is the path length in cm of the ray from the X-ray source to pixel (x, y) crossing the j-th material.

Polygon meshes, e.g. triangles, are used in gVXR to represent 3D objects. This method is commonly used in computer graphics (CG), including real-time video games and VR, animations, and computer-aided design (CAD). It is intuitive to compute the Beer-Lambert law with ray-tracing when polygon meshes are used. However, this technique is relatively computationally intensive: i) a ray must be fired between the source and each detector pixel, and ii) intersection tests for each ray for each triangle of each 3D object must be performed. Freud et al. adapted the Z-buffer technique to efficiently compute $\mathbf{d}_j$ in Eq. 1 from polygon meshes.[4] It relies on rasterization and does not require to sort intersections. In this case, each polygon is processed a single time, projecting it on the detector plane, and using an accumulator buffer. The computational complexity is considerably reduced.

gVXR implements Freud's algorithm on GPU using a graphics API.[2] Since its inception, functionalities have been added to gVXR to improve the level of realism of the simulations. A monochromatic source was initially used to mimic fluoroscopy in a real-time medical VR simulator.[5] Polychromatism and the focal spot of the detector were then introduced to improve realism.[6] In 2013, the code was redeveloped to become, gVXR, and was made available to the community as an open-source project on SourceForge (`https://sourceforge.net/projects/gvirtualxray/`, accessed: 18 Jul 2024).[3] The impulse response of the detector and Poisson noise are also supported.[7] The scintillator material of the detector and the tube voltage and beam filtration can now be specified.[8]

gVXR is cross-platform: it runs on Windows, GNU/Linux, and MacOS computers (Intel architecture only, although ARM support is planned). It supports GPUs from any manufacturer. gVXR is scalable: it runs on laptops, desktop PCs, supercomputers, and cloud infrastructures. Containerization using Docker is even possible.[9] A wide range of programming languages (C/C++, Python, R, Ruby, Tcl, C#, Java, and GNU Octave) can be used. Its Python package "gVXR" is available on the Python Package Index (https://pypi.org/project/gVXR/, accessed: 18 Jul 2024).

Scanned objects are defined using polygon meshes. Surface meshes (triangles) in most popular file formats (eg. STL, PLY, 3DS, OBJ, DXF, X3D, DAE) can be used. Volume meshes (tetrahedrons) in the Abacus format may also be used but their support is experimental. The material property must be specified for each scanned object. Chemical elements (e.g. the symbol 'W' or the atomic number 74 for tungsten); compounds, e.g. H_2O for water; mixtures, e.g. Titanium-aluminum-vanadium alloy, Ti90Al6V4; and Hounsfield units (for medical applications) are supported. The photon cross-sections provided by Xraylib (https://github.com/tschoonj/xraylib, accessed: 18 Jul 2024) are used to compute μ values in Eq. 1.[10]

Cone beam geometries (both point sources and focal spots) are supported to mimic X-ray tubes. A parallel beam can be used to mimic synchrotrons. The beam spectrum can be either monochromatic or polychromatic. Both SpekPy[11] and Xpecgen[12] are supported as backends to specify the tube voltage and the beam filtration used. To increase realism, photonic noise can be turned on. In this case, the photon flux must be specified.

It is possible to model ideal detectors as well as realistic detectors. In this case, the user can specify a point spread function (PSF), i.e. the level of blur inherent to the detector, and the thickness and material composition of the scintillator. It is also possible to simulate spectral imaging.

Orbital, helical and arbitrary trajectories are supported to simulate a CT acquisition. Appendix A provides a full example of CT simulation with gVXR and CT reconstruction with CIL.[13] It is written in pure Python. It is also possible to describe the simulation and CT acquisition in a user-friendly JSON file. Appendix B shows JSON file corresponding to the previous example. When a JSON file is used, the Python code can be significantly simplified (see Appendix C). Working copies of these programs are available as Jupyter notebooks on GitHub (https://github.com/TomographicImaging/gVXR-SPIE2024).

3. VALIDATION

Unit tests are implemented using CMake, CTest (https://cmake.org/cmake/help/book/mastering-cmake/, accessed: 18 Jul 2024) and GoogleTest (https://google.github.io/googletest/, accessed: 18 Jul 2024). Continuous integration is deployed using Jenkins (https://www.jenkins.io/, accessed: 18 Jul 2024) to ensure the robustness and replicability of the code. Coverage, the amount of code that has been executed during unit tests, is 52% and is improving on a regular basis. Results of the nightly builds can be consulted on CDash (https://my.cdash.org/index.php?project=gVirtualXray, accessed: 18 Jul 2024).

To validate the accuracy of gVXR, successive validation tests of increasing complexity were performed. Each milestone was validated individually with an appropriate methodology. For the Beer-Lambert implementation, we initially compared simple images simulated with gVXR with corresponding images simulated with a state-of-the-art Monte Carlo package (Geant4/Gate).[3]

More advanced functionalities, such as voltage, beam filtration and scintillation, were validated using two anthropomorphic phantoms. The first one is a digital phantom: pEdiatRic dosimetRy personalized platfORm (ERROR).[14] It corresponds to the anatomy of a 5-year-old boy. It is provided as a labeled $512 \times 511 \times 190$ volume, which includes 24 different structures, such as air, muscle, bone, stomach-interior, cartilage, etc. As it is a digital phantom, it can be used to compare gVXR and Gate's simulations. The number of photons impinging the detector was 10^9. About 10 days of computations were required on the test computer to produce a simulated image of 128×128 pixels with Gate; only a few microseconds gVXR. Both simulations are visually close (see Figure 1). All the image comparison metrics indicate that the images are extremely similar when scattering is ignored: Zero-mean normalised cross-correlation (ZNCC) is 99.99%; mean absolute percentage error (MAPE) is 2.23%, and structural similarity index (SSIM) is 0.99.

The second phantom is the Lungman anthropomorphic chest phantom (Kyoto Kagaku, Tokyo, Japan).[15] It represents a 70 kg male. The phantom is made of materials with X-ray absorption properties close to those of human tissue. Tumors of various densities are embedded. A computed tomography (CT) scan of the phantom

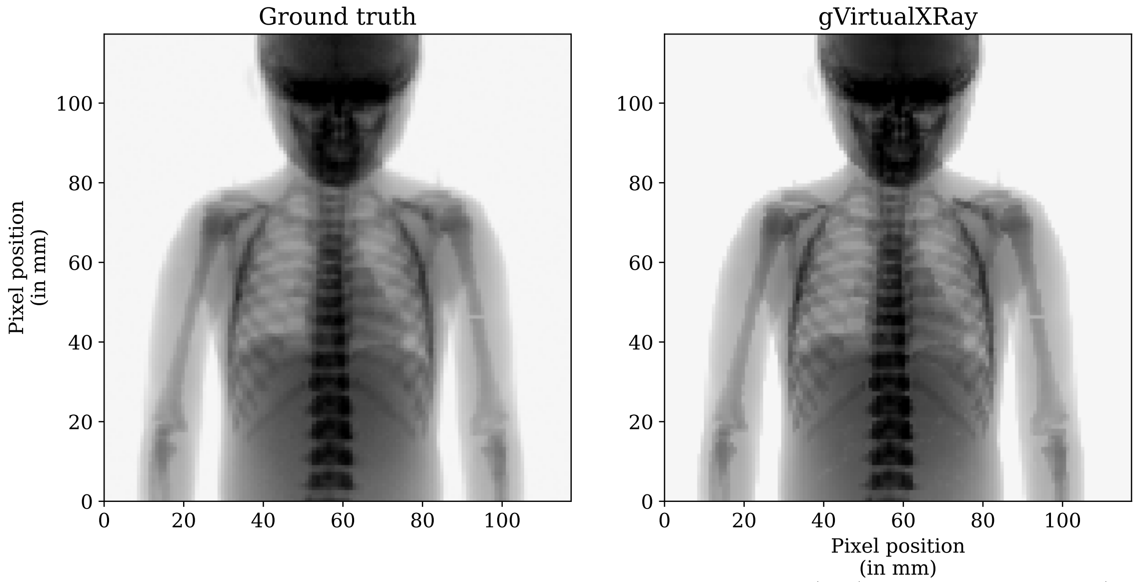


Figure 1: Comparison between X-ray projections simulated with GATE (left) and gVirtualXray (right). For a fair comparison, each image is displayed using the same look-up table. MAPE: 2.23%, ZNCC: 99.99%, and SSIM: 0.99.

was acquired with a device clinically utilized at Ysbyty Gwynedd Hospital (UK), a 128-slice Somatom Definition Edge scanner by Siemens Healthcare (Erlangen, Germany). A digital phantom was first created by image segmentation using open-source toolkits, the Insight Toolkit (ITK)[16] and Visualization Toolkit (VTK).[17] The digital phantom is freely available on Zenodo.[18] The material composition of each segmented structure is derived from the average Hounsfiled unit of the structure in the original CT scan. Schneider et al.[19]'s method is built in gVXR to convert the Hounsfield values into material compositions and densities. A CT scan acquisition is then simulated using gVXR and reconstructed with CIL.[13] The original CT scan taken with the Somatom

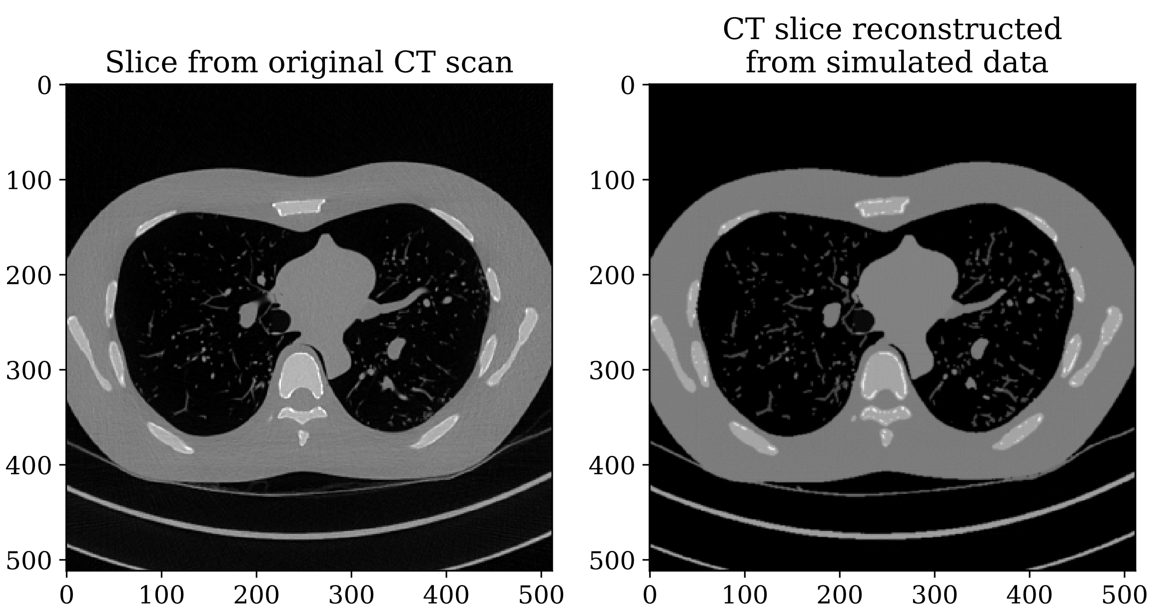


Figure 2: Comparison between a CT slice reconstructed from simulated projections with a slice from the original CT scan. For a fair comparison, all the images are displayed using the same look-up table. MAPE: 5.01%, ZNCC: 98.44%, SSIM: 0.78.

Definition Edge scanner can be compared with CT volume reconstructed from simulated data. Figure 2 shows the corresponding slices are close to each other. Hounsfield values are comparable. MAPE is about 5% and the ZNCC is above 98%, indicating a high level of correlation between the two volumes.

gVXR is so fast that it is possible to embed the X-ray simulation into objective functions and register a simulated radiograph on experimental data (see Figure 36). A real digital radiograph was taken with a clinical X-ray machine by GE Healthcare (Chicago, Illinois, USA) at Ysbyty Gwynedd Hospital. The digital Lungman phantom was registered to reproduce the same position and orientation as in the digital radiography taken with the clinical device (see Figure 3). ZNCC is 98.91%, which is close to 100%; SSIM is 0.94, which is relatively close

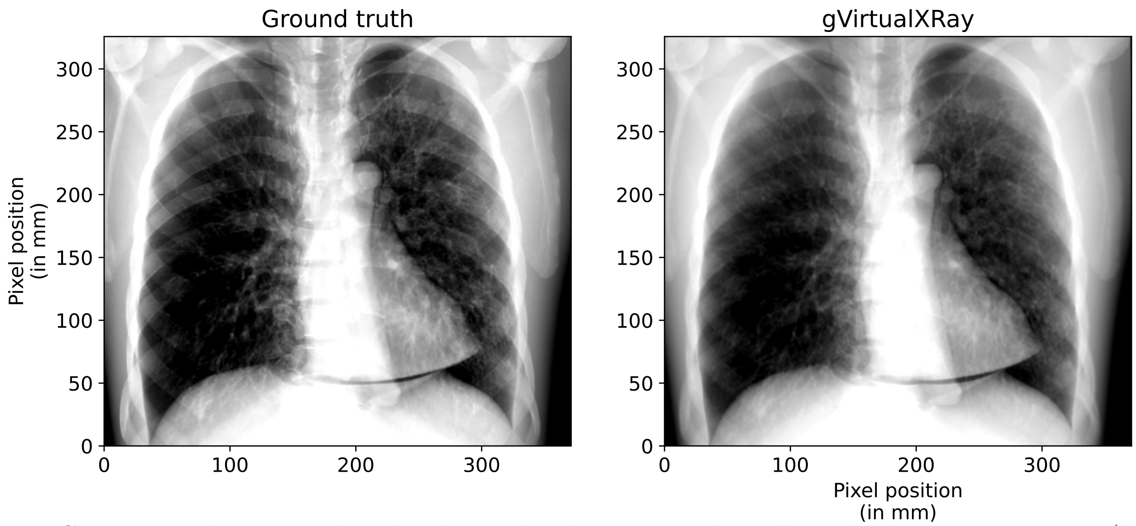


Figure 3: Comparison between a digital radiograph taken using a clinically utilized X-ray equipment (left) and a registered X-ray projection simulated with gVirtualXray (right). For a fair comparison, each image is displayed using the same look-up table. MAPE: 1.56%, ZNCC: 98.91%, and SSIM: 0.94.

to 1; and MAPE is 1.56%, which is close to 0%. It demonstrates the ability of gVXR to reproduce radiographs taken with clinically utilized devices.

4. APPLICATIONS

4.1 Digital twinning

Digital Twinning is the creation of virtual models of real-life components. In this case, gVirtualXRay allows true representative X-ray simulations calibrated to real-life machines. To create a Digital Twin, all factors of an X-ray system must be taken into account, ranging from the mechanics of the system (can the detector or source move? What clearance is available for the sample?, etc) to X-ray source and detector properties (maximum kV, focal spot size, pixel resolution, scintillator properties, PSF, and so forth).

A core part of creating a Digital Twin is calibrating the noise of a system based on the target amperage, this involves an experimental method to measure the noise characteristics at differing mAs values;- which then can be exposed in the model as a parameter to users of the Virtual Twin. gVirtualXRay's flexible API has allowed the development of WebCT (`https://webct.io/`, accessed: 18 Jul 2024), an interactive real-time web-based app for X-ray simulation, allowing anyone of any skill level to quickly simulate an X-ray scanner (see Figure 4). This is excellent for scan planning, answering feasibility questions, and teaching/training on X-ray systems without requiring access to expensive equipment.

We are developing digital twins of specific beamlines, including specific laboratory computed tomography (labCT) devices and a synchrotron. One of them is a new dual-beam XRCT laboratory equipment of the MateIS laboratory (Lyon, France).[20] The original concept of this design is that two beamlines are perfect twins, high energy (300 kV), oriented at $\pi/2$ to each other, and share the same rotation stage (see Figure 5). This dual-beam

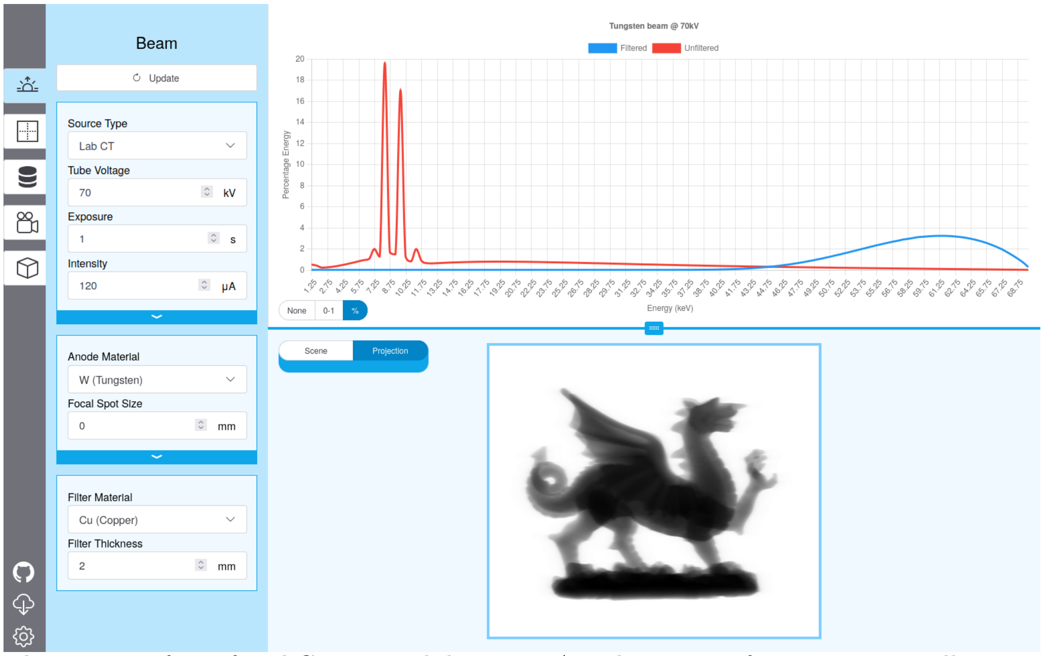


Figure 4: Video 1 – Interface of WebCT in a web browser. A wide variety of X-ray settings allow quick, iterative scan planning and training. http://dx.doi.org/doi.number.goes.here

setup makes it possible to analyze a sample simultaneously from two different viewing angles. The noise model is under validation and a specific dual-beam calibration protocol has been proposed.[21] Several test samples have been used to prove the feasibility of this dual-twin. We report here the acquisition and simulation of a part of a lab tensile machine for in situ stress in scanning electron microscopes (SEM) because a CAD model is available. Table 1 lists the most significant data acquisition parameters used during the experiment with the dual-beam XRCT device. Simulated X-ray projections of the CAD model is registered onto the experimental (see Figure 36). All the parameters available in Table 1 are used as parameters of the simulation. An optimization algorithm moves the CAD model in the 3D space until the simulated and experimental images match. Figure 6 shows a great level of similarity between images acquired with the actual device and its digital twin. In an ideal scenario, CT slices of a sample made of a single and homogeneous material correspond to binary images, i.e. air vs. sample. However, beamhardening, focal spot, impulse response of the detector and noise corrupt the experimental data. All the artifacts visible in slices reconstructed from experimental data are also visible in the simulated ones (see Figure 7).

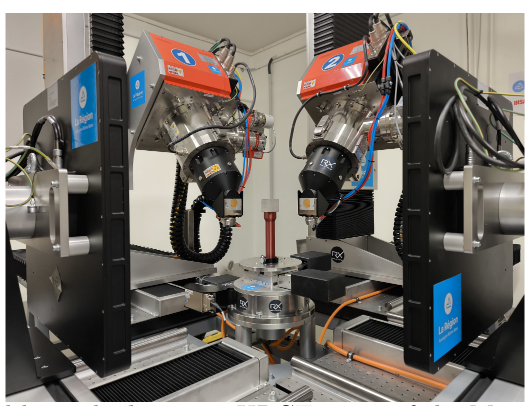


Figure 5: Dual-beam high-energy XRCT setup of the MateIS laboratory.

Table 1: CT scan parameters used during both the experimental scan and the digital twin.

Tube voltage [in kV]	160
Exposure	0.167s
Current	200uA
Beam filtration	0.4 mm of copper
Number of projections over 360°	1120
Detector pixel pitch [in mm]	0.150×0.150
Image resolution [in pixels]	1432×872
SOD [in mm]	306.414
SDD [in mm]	807.248
Reconstruction algorithm	FDK
Voxel size [in mm]	$0.0569369 \times 0.0569369 \times 0.0569369$

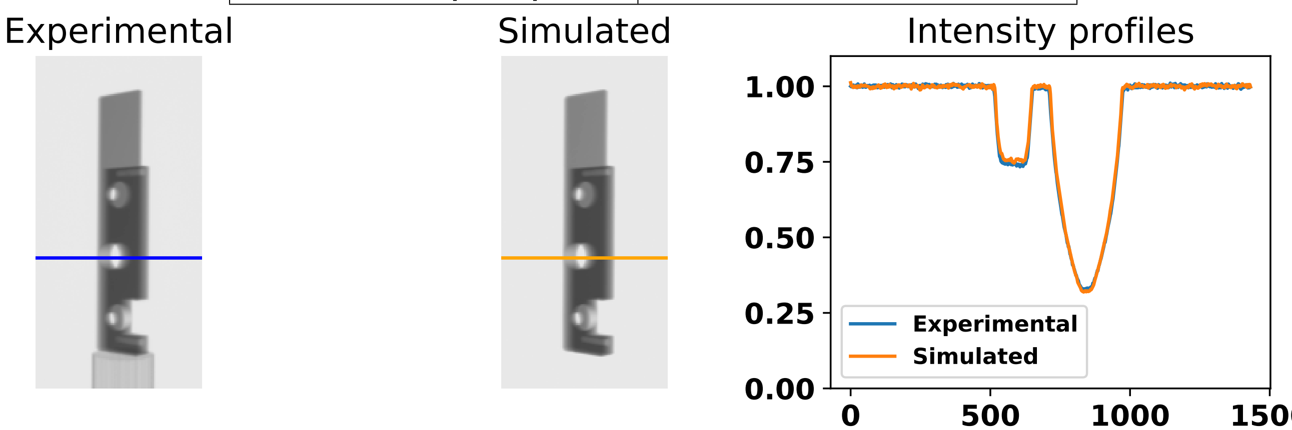


Figure 6: Comparison between X-ray projections taken with a real dual-beam XRCT laboratory device and its digital twin. For fair comparison, both projections are displayed using the same lookup table.

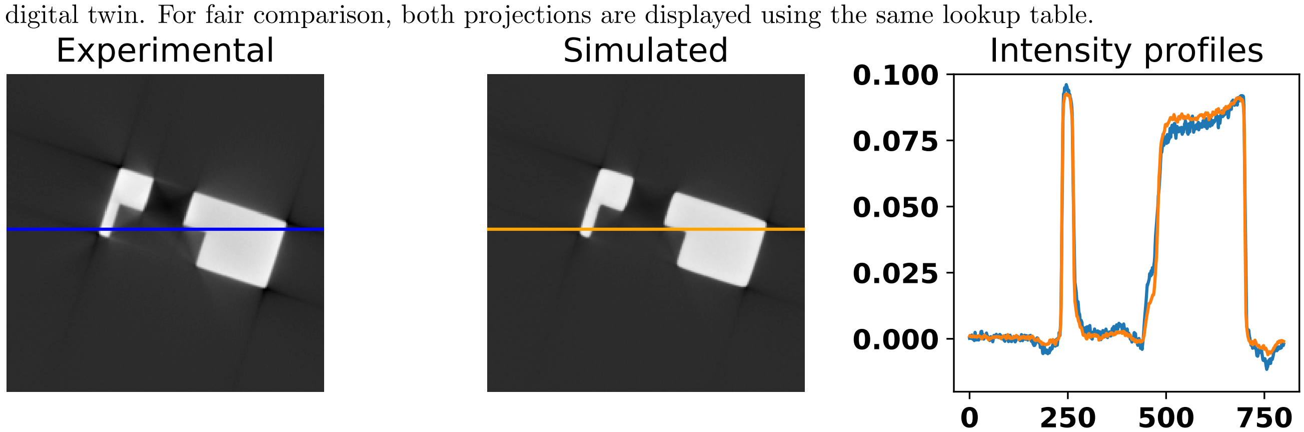
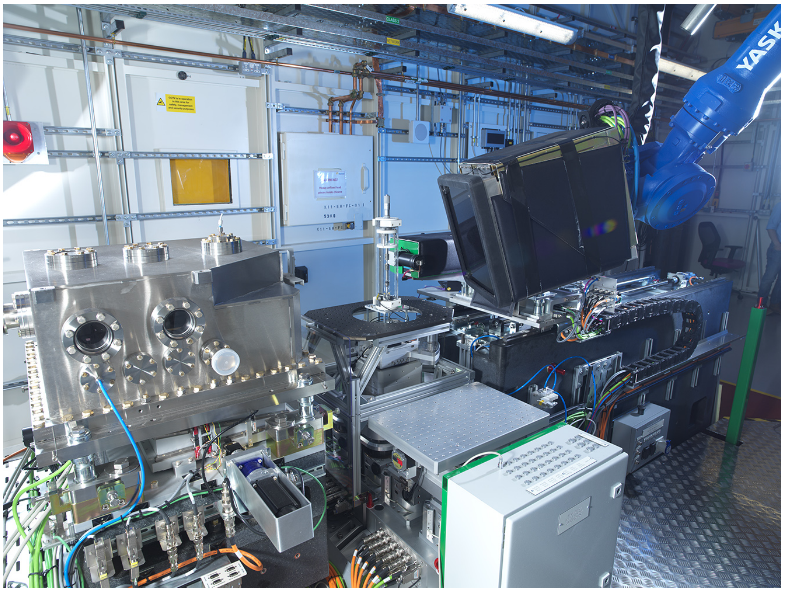


Figure 7: Comparison between CT slices reconstructed from data taken with a real dual-beam XRCT laboratory device and its digital twin. For fair comparison, both slices are displayed using the same lookup table.

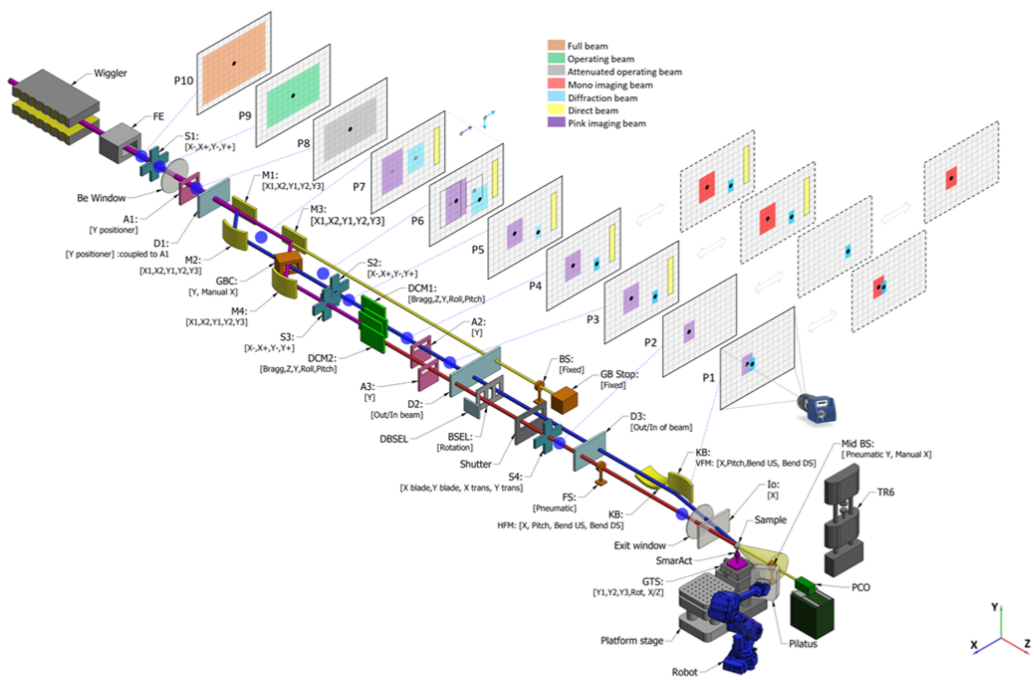
We are also building a predictive tool for synchrotron µ-CT at the Dual Imaging And Diffraction (DIAD) beamline of the Diamond Light Source[22] and integrating it in gVXR. DIAD is a dual-beam X-ray instrument for quasi-simultaneous imaging and diffraction, which operates two independent beams at energies of 7-38 keV (Figure 8). The simulation model will be integrated into our web user interface for gVXR, WebCT.

The development of such digital twins opens up new perspectives, it is now possible:

- To train users on specific devices;
- To predict what experimental data will look like from CAD models;



(a) A photograph of the DIAD beamline endstation, showing the KB optics on the left, and the imaging and diffraction detectors on the right.



(b) Schematic of source, X-ray optics, and endstation on the DIAD beamline. The configuration of each beam is displayed at each optical component leading to the registration of both beams at the sample location.

Figure 8: DIAD beamline of the Diamond Light Source.

- To assess the feasibility of scanning specific samples on specific devices before submitting beamtime proposals to facilities;
- To optimize scanning parameters offline, i.e. before beamtime;
- To generate a large amount of automatically annotated data for training machine learning algorithms;
- To design new systems.

4.2 Education

The use of simulation in the curriculum has numerous benefits in both clinical[23] and industrial radiology. Simulation can avoid several risks and provide a realistic experience in areas where there are fewer opportunities for direct access to devices.[24] It permits trainees to familiarize to CT system without the risk of damaging very expensive instruments. For example, detectors can be damaged by trainees crashing samples into them or saturating them, with detectors costing over £50,000 each.

However, care must be given to provide high-fidelity simulations as inaccurate ones can lead to the transferability of inaccurate learning into practice, causing risks involving ionizing radiation. This is particularly true in clinical radiology due to the harmful effect of ionizing radiation, and as per IR(ME)R regulations 2000, ionizing radiation should be as low as reasonably practical, students have to be directly supervised with no room for errors and flexibility in practice. Due to limited placement opportunities and workforce capacity issues to train, the use of simulation is also increasing and evolving in education to create capacity and flexibility. In addition, some imaging examinations are rarely performed or are only performed in specialised centres, simulation allows such examinations to be replicated to develop some experience in their acquisition. Demonstrating how image quality and radiation dose if influenced by modifications in acquisition parameters (e.g. kVp, mAs) on scanners/equipment is extremely valuable in gaining a richer understanding of radiographic practices. Again, this is particularly important in clinical routine, due to the harmful effects of ionizing radiation, we are limited to how we can test our radiographic practices as it is unethical to use patients. For the same reason, caution may also be required with samples. A high radiation dose can damage samples, e.g. by creating cracks in their structures. Placement experience can only provide certain scenarios as patients walk through the door, whereas simulation can provide complex, different scenarios to enhance the learning experience. The same benefits are also applicable to industrial radiography as samples will vary a lot from one user to another. Allowing modification of parameters and multiple exposures is obviously not unethical in this case; however, it enables operators to select suitable parameters by trial and error much more efficiently.

We have deployed gVXR in material science lab-sessions of the INSA-Lyon engineer school. It has been embedded in a Jupyter notebook together with the open-source reconstruction toolkit RTK.[25] Several interactive exercises have been proposed to enable students to learn and gain hands-on experience with X-ray tomographic setups. They first study both digitally with the twin and experimentally with the bench the critical sensibility to crack orientation in a cylindrical sample (additive manufacturing) in which a through crack has been added. Then the students gradually familiarize themselves with the 3D reconstruction technique: (i) first with mono-energy and no noise (i.e. infinite stat), then (ii) with a given exposure (i.e. number of X-rays per pixel) to highlight photon starvation, and (iii) finally with a realistic energy distribution typical of an X-ray generator to understand beam hardening. An example of the 3D visualization of the reconstructed simulated volume is shown in Figure 9. It is worth noting that this progression in the complexity of the imaging setup cannot be done experimentally. The digital twinning of the X-ray setup is crucial for those lab sessions of the material science department.

gVirtualXray is used to simulate X-ray images in real-time VR applications such as medical training simulators.[26–28] we took this approach and embedded gVXR in the Unreal Engine 5, a very popular game engine developed by Epic Games, as there is relatively little education or training material available for introducing members of the public, academics or operators to lab CT.[29] The user evolves in a 3D virtual environment with a lab CT scanner (see Figure 10). The user can interact with this in a very similar way to how they would in real life, including procedures such as correctly positioning samples, choosing beam energy and performing simple image analysis. The application also provides an interface for gVXR library without the need for the

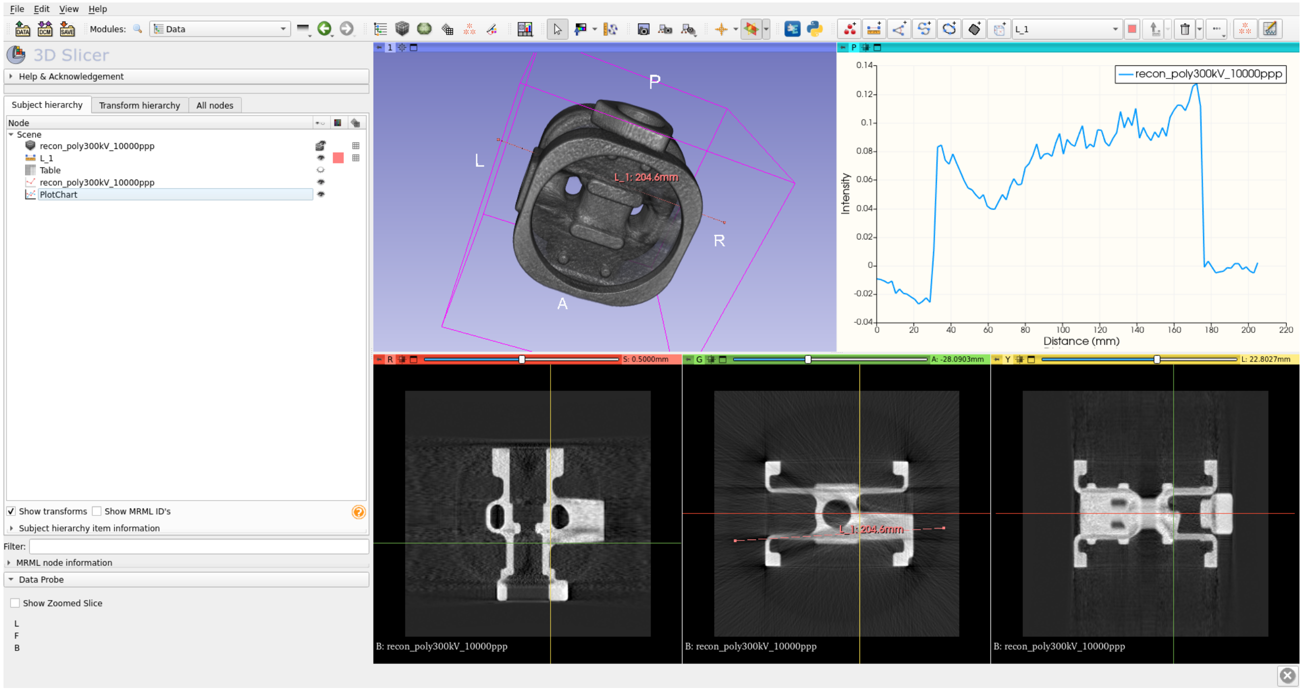


Figure 9: Example of visualization (3D-Slicer) of a reconstructed gVXR-simulated volume of an industrial part: volume rendering, sampled profile in the volume and orthogonal sections.

Figure 10: Video 2 – Virtual environment featuring a labCT device powered by gVXR and the Unreal Engine. http://dx.doi.org/doi.number.goes.here

user to understand how to use the command line. Following its development, the application was submitted to several experts in the field of lab CT, some of whom had previous experience with using game engines in their work. The general consensus was that the application was relatively usable, scoring above 60 points in a system usability survey, and feedback was generally positive about the ease of use, features provided and similarities with working with real hardware.

4.3 Set-up of experiments

4.3.1 Case study in the energy sector: CT scan of mock nuclear fuel

In non-destructive testing (NDT), X-ray computed tomography (CT) is commonly used to find defects in materials. Simulations were performed to ascertain the feasibility of CT scans of ceramic kernels held within a dissimilar ceramic matrix. Ceramic-ceramic matrix composites are garnering a great deal of interest in many applications, including as nuclear fuels for high-temperature gas reactors. The aim is to conduct experiments i) to detect the interface between two very similar materials (in terms of composition and density), and ii) to assess the defects in the structure that exist as a result of manufacturing methods in spherical ZrB_2 kernels held within a cylindrical zirconium dioxide (ZrO_2) matrix material.

A loss of density compared to theoretical values is expected due to the manufacturing process. Prior to the simulations, samples were produced. The diameter and height of the cylindrical matrix and the diameter of a typical spherical kernel were measured using a caliper. Their masses were assessed using a digital weighing scale. It makes it possible to compute the volume and material densities of the ZrB_2 kernels and the ZrO_2 matrix. This way, we can ensure the simulations are based on realistic values in terms of sizes, densities and material compositions. Table 2 provides a summary of the sample composition.

Table 2: Description of the sample composition.

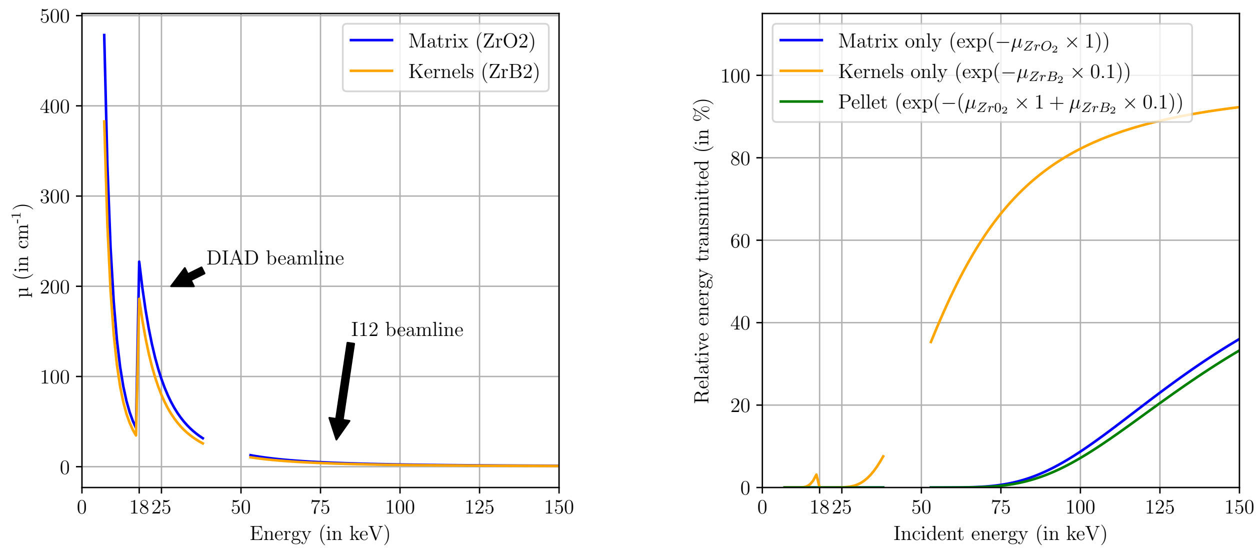
Material / Properties	Matrix	Kernels
Composition	ZrO_2	ZrB_2
Shape	Cylinder	Spheres
Diameter Height	8 to 10 mm 10 mm	0.8 to 1 mm N/A
Theoretical density Measured density Measured reduction of density	5.68 g/cm^3 3.23 g/cm^3 43%	6.08 g/cm^3 2.43 g/cm^3 60%

As the materials are close to each other and as the samples are relatively dense, i.e. opaque to X-rays, we will favor synchrotron radiation over the use of conventional X-ray tubes used in labCT. This is because synchrotron radiation can provide almost monochromatic spectra with high flux.

We use the Diamond Light Source, UK's national synchrotron radiation facilities, as an example. Two CT beamlines are available: the low-energy DIAD beamline, and higher-energy I12 beamline. A suitable energy must be selected i) to maximize the contrast between the two materials, and ii) to allow a sufficient level of radiation transmission through the sample. As CT images correspond to maps of linear attenuation coefficients, μ in Eq. 1, we aim at maximizing the difference between the coefficients of zirconium diboride (ZrB_2) and ZrO_2. Figure 11a demonstrates the linear attenuation coefficients of the matrix and kernels along the energies supported by both beamlines. The difference is the largest for 7 keV. However, when we apply the Beer-Lambert law in Eq. 1 using μ and $\mathbf{d}$ values corresponding to the sample, the transmission through the sample is 0%. The transmission remains low (below 5%) until roughly 95 keV (see Figure 11b). At first sight, the issue is that we achieve the best absolute differences at low energies ($\mu_{ZrO_2} - \mu_{ZrB_2}$ in Table 3), but only high energies seem to be suitable to image the sample. Indeed, Table 3 also shows that the transmission remains below 5% until 110 keV. We must therefore ascertain that a difference in attenuation coefficient of 0.39, 0.28, or 0.22 cm^{-1} is significant enough to be

Table 3: Theoretical linear attenuation coefficients and photon transmission through the sample at energies supported by the CT beamlines at the Diamond Light Source.

Energy (in keV)	μ_{ZrO_2} (matrix) (in cm^{-1})	μ_{ZrB_2} (kernels) (in cm^{-1})	$\mu_{ZrO_2} - \mu_{ZrB_2}$	$\frac{\mu_{ZrO_2} - \mu_{ZrB_2}}{\mu_{ZrO_2}}$	Transmission
7	479.61	383.02	96.59	20.14%	0.00%
38	31.63	25.84	5.79	18.31%	0.00%
53	12.78	10.42	2.36	18.47%	0.00%
60	9.13	7.43	1.70	18.62%	0.01%
70	6.05	4.91	1.14	18.84%	0.28%
90	3.16	2.55	0.61	19.30%	4.61%
110	1.95	1.56	0.39	20.00%	14.96%
130	1.36	1.07	0.28	20.59%	26.79%
150	1.02	0.80	0.22	21.57%	37.05%



(a) (b)

Figure 11: (a) Linear attenuation coefficients and (b) relative transmission through the sample for the energy ranges at the low-energy DIAD beamline and higher-energy I12 beamline of the Diamond Light Source.

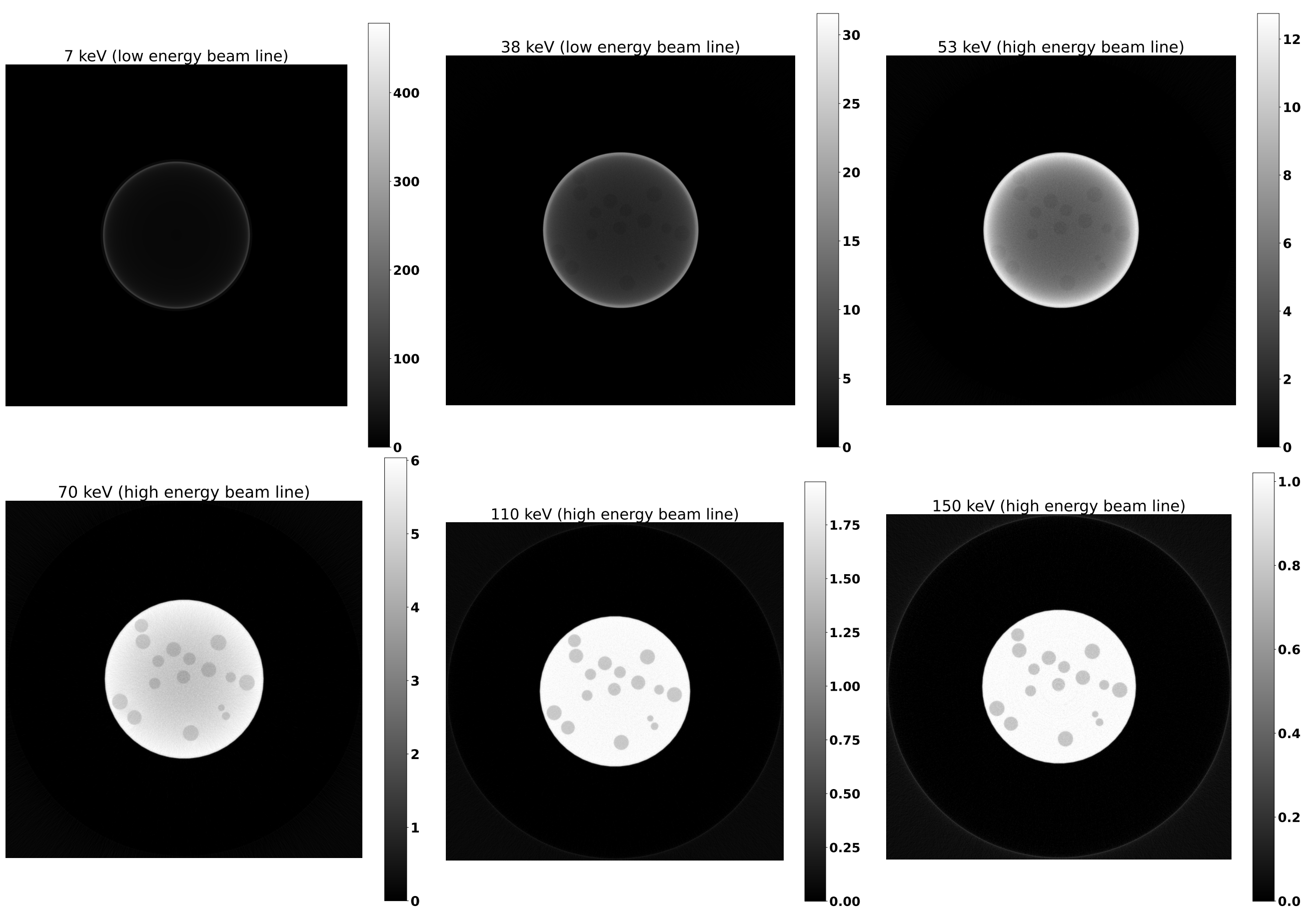


Figure 12: CT slices of mock nuclear fuel reconstructed from simulated data.

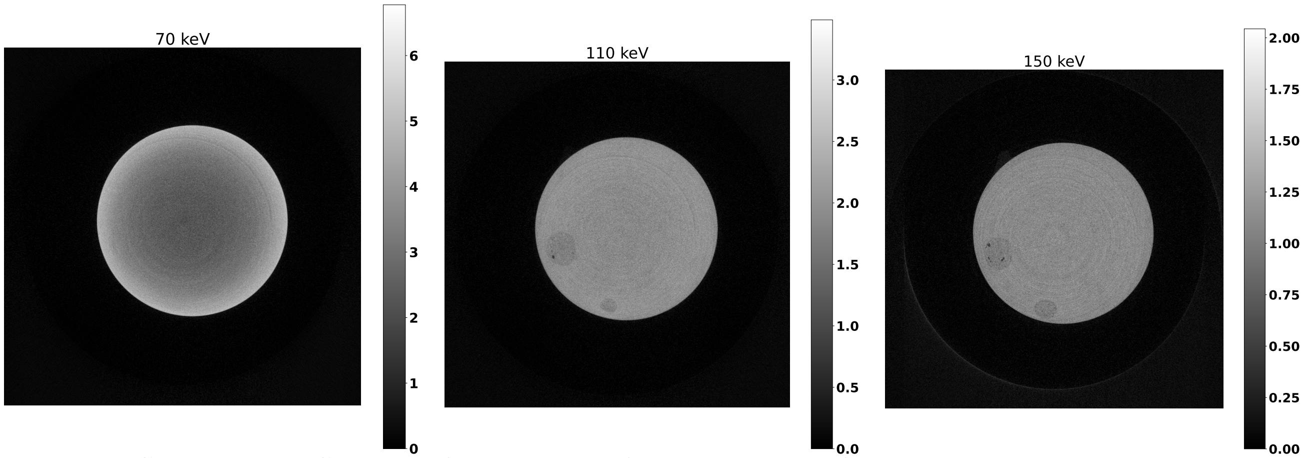


Figure 13: Corresponding CT slices of mock nuclear fuel reconstructed from experimental data acquired at the high energy beamline. The same geometrical set up is used. The only change is the incident energy.

visualised in reconstructed CT scans. The relative difference ($\frac{\mu_{ZrO_2}-\mu_{ZrB_2}}{\mu_{ZrO_2}}$) remains constant ($19.54\% \pm 1.11\%$) across all the energies, despite both material being close to each other. To select a suitable energy and make sure this relative difference in μ is sufficient enough, we performed simulated CT acquisitions at energies with the ranges [7, 38] and [53, 150] keV supported by the DIAD and I12 beamlines. Photonic noise and a few percent of harmonics were empirically added to the beam spectrum for added realism. Figure 12 shows that according to our initial assumption energies below 85 keV were inappropriate. At 7, 38 and 53 keV, we were not able to scan the sample due to photon starvation. At the lowest energy, 70 keV, the reconstructed attenuation coefficients in the matrix are not homogeneous (see standard deviations in Table 4). The intensity is much darker in the centre of the matrix. For this reason, energies from 90 to 150 keV seem to be appropriate. Figure 13 shows CT slices reconstructed from experimental data acquired at the high energy beamline. As observed with the simulated data, attenuation coefficients in the matrix are not homogeneous when 70 keV is used. No kernel is visible. As expected, 110 keV and 150 keV are suitable energies. However, 150 keV was selected as it can lead to faster scans due to a higher transmission rate.

Table 4: Linear attenuation coefficients measured on both experimental and simulated data.

Energy (in keV)	Mean μ_{ZrO_2} (matrix) (in cm^{-1})		Mean μ_{ZrB_2} (kernels) (in cm^{-1})	
	Simulated	Experiment	Simulated	Experiment
70	5.236 ± 0.42	3.037 ± 0.854	4.429 ± 0.334	3.138 ± 0.556
90	3.011 ± 0.106	2.506 ± 0.477	2.474 ± 0.092	2.426 ± 0.359
110	1.885 ± 0.065	1.801 ± 0.307	1.533 ± 0.058	1.657 ± 0.248
130	1.313 ± 0.048	1.273 ± 0.217	1.059 ± 0.045	1.158 ± 0.216
150	0.994 ± 0.04	1.031 ± 0.205	0.794 ± 0.036	0.936 ± 0.197

4.3.2 Case study in medical physics: Sensor fusion and computer-generated densitometric images

Densitometric radiographic images is a technique that combines two radiographs that were produced with two different tube voltages. It is now possible to use ubiquitous motion sensing input devices originally developed for the video game industry, e.g. the Microsoft Kinect, to capture the 3D envelope of a real patient and saved it as a 3D polygon mesh that can be used by gVXR. This 3D object can then be registered with radiographic images from that patient (see Figure 14). The integration of this information (3D object + real radiographic images) has allowed for the acquisition of densitometric images, where one of the two images is acquired experimentally. gVXR proved to be very useful for generating the second image with another tube voltage.[30] This simulated image includes a simulated beam with new custom-defined parameters (kVp, mAs, hardening, filtration, etc.) but with the very same exact geometry, including that of the patient at the moment the radiograph was acquired.

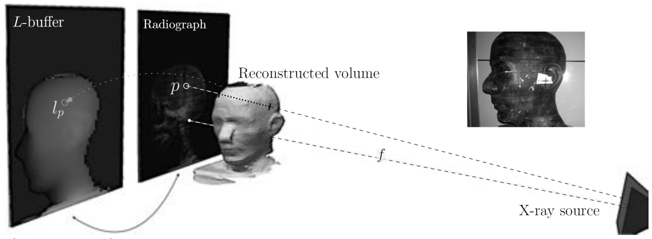


Figure 14: Real path length (L-buffer) computed with gVXR of the contour data of a head-like anthropomorphic phantom.

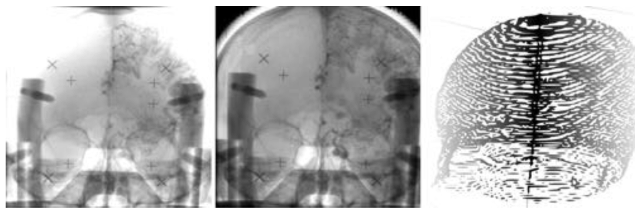


Figure 15: Example of a densitometric image (middle image). The left radiograph represents the original X-ray instance and the rightmost image represents the traversed volume.

Substituting one of the two images needed to generate a densitometric radiographic image leads to i) a lower radiation dose for the patient than in traditional densitometric imaging, and ii) a better image contrast than in typical single exposure radiographic imaging (see Figure 15).

Also, to prepare for real sessions in hospital settings, gVXR was used to simulate the scenarios to be studied beforehand, prior to the application of any radiation.[31]

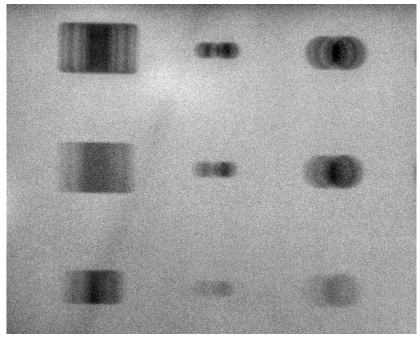
4.3.3 Case study in condition monitoring: Drone-based radiography for wind turbine blades

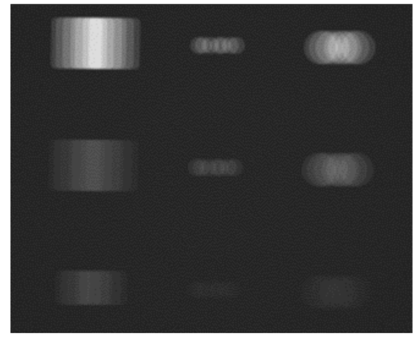
Approximately 2-3% of wind turbine blades require annual replacement, often due to defects in their internal composite layers.[32] These defects are invisible from the outside, prompting wind farm operators to use NDT techniques for internal inspections. A common method involves technicians rappelling down the blade while manually scanning each section with handheld ultrasonic equipment. However, this approach is costly, time-consuming, and potentially hazardous. Aerial radiography offers an autonomous alternative by using two drones equipped with the source and detector of a digital radiograph (DR) system, as illustrated in Figure 16a.[33] Despite its advantages, this approach faces challenges from drone- and environment-induced disturbances, leading to motion blur in the images.

Mitigating motion blur requires both effective controllers to stabilize the drones and deblurring algorithms to enhance image quality. By integrating gVXR with a drone simulation environment, we can generate X-ray images corresponding to various trajectories, models, and disturbances, effectively closing the control system's feedback loop on image quality. The post-processing methods are integrated directly into this simulation pipeline to develop a comprehensive solution. The ability of gVXR to rapidly generate large datasets enables the application of learning methods to both drone control as well as image post-processing. Our preliminary research has focused on validating the accuracy of gVXR for motion-blurred images using an experimental dataset obtained with a commercially available portable DR system. An example of these results is shown in Figures 16b and 16c.

4.4 Focal spot assessment

In the context of non-destructive testing and metrology of mechanical parts, CT scans represent a valuable tool.[34] For instance, when developing a new product or material, there is a need in the industry to ensure quality (i.e. low presence of cracks, pores and other defects), and meet safety standards and dimensional requirements of that product. However, high precision tomographic images are required for internal defect detection. This high precision requires extensive scanning times and/or high illumination beams. Despite vast improvements in the conception of CT devices[34] and 3D reconstruction algorithms,[35] image precision depends on several factors. Among these factors, one can cite the number of considered projections, the input X-ray intensity, the allowed





(a) (b) (c)

Figure 16: (a) Dual drone NDT inspection system. (b) Image of a simple aluminum sample obtained with both the source and detector mounted on a motion platform. (c) Corresponding simulated projection from gVXR, aimed at replicating the blur structure. Note: The experimental image is contrast-enhanced for visualization purposes.

time for acquiring one projection and the maximum thickness and constitutive materials of scanned mechanical parts.

In the industry of non-destructive testing, common X-ray generating devices consist of a wire-cathode emitting an electron beam which is accelerated towards an anode (Figure 17). The scattering of electrons on the anode causes the emission of X-rays via the bremsstrahlung, Auger and X-ray fluorescence effects.[35] This type of device can be relatively compact and is therefore well-suited for industrial applications. However, this type of device suffers from some image blurring as the electron beam impacts the anode on a non-punctual area; creating a focal spot. Controlling the size and shape of the focal spot is an important challenge as nowadays non-destructive testing and dimensional measurements require spatial resolution in the micrometer range.[36–38] Achieving spatial resolution of the order of the micrometer requires small focal spot sizes and large magnification factors, resulting in low X-ray flux; thus impeding the ability to scan thick mechanical pieces in a reasonable time frame. The X-ray flux may be increased by increasing the applied current, but usually at the expense of a larger focal spot size, and a consecutive loss in spatial resolution.[35]

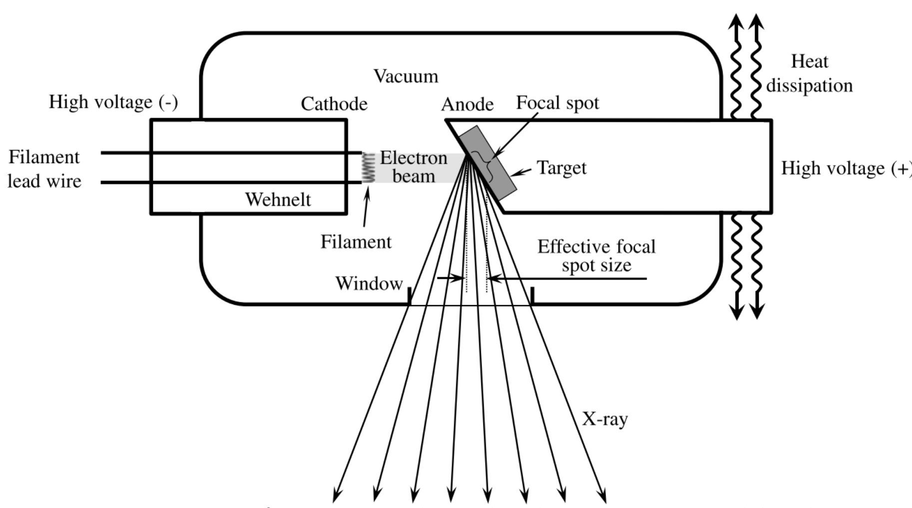


Figure 17: Schematic representation of an X-ray tube. The electrons generated by thermionic emission at the cathode are accelerated by the applied voltage between the cathode and the anode. When the electron reach the anode, X-rays are generated. (Scheme from Figure 4.4 in Reference[35]).

Therefore, obtaining high-brightness X-ray sources for non-destructive testing of thick/dense material parts while achieving a spatial resolution of the order of the micrometer and within a reasonable time-frame is still a challenge today. Of course, on the one hand, high-brightness X-ray sources can be generated by synchrotron facilities. However, access to such facilities is not readily available and require extensive effort in sample preparations. On the other hand, efforts are conducted for developing compact electron accelerators[35] and high-brightness micro-focus sources.[35, 39] Nonetheless, these facilities are non-standard and their performances and characteristics still under study; thus, at least for the moment, not suitable in an industrial environment subject to heavy regulations.

In this context, deconvolution techniques are a promising and full software solution for retrieving sharp X-ray images from blurred ones. Indeed, new deconvolution techniques have been rapidly developing last decade[40, 41] and offer the possibility to use devices with large focal spots, while generating images with sufficient resolution. However, deconvolution techniques require the knowledge of the shape of the focal spot, or PSF. Numerous techniques exist for determining the PSF of a CT device. The most direct technique consists in using a pin-hole camera from which a focal spot can be directly extracted.[42] However, manufacturing a pin-hole camera is technically challenging.[43] Some other techniques aim to estimate only line profiles of the PSF (or line spread-function (LSF)) using thin wires, narrow slits or sharp edges.[44–46] Yet, tilting angles should be selected with care for taking into account the finite resolution of digital imaging system.[45, 47] A more general method considers a small circular aperture from which the focal spot shape is deduced from edge profiles in all directions using filtered back projection.[48, 49] Additional techniques rely on more complex shapes, such as coded masks,[50] converging line group patterns (Siemens star),[51] spheres,[52–56] and other phantoms.[57] Here, we consider sphere phantoms that present two main advantages: first, they are relatively easy to manufacture with a given precise radius, second, due to their symmetry, they do not require a precise mechanical alignment with respect to the measurement direction.

We propose to estimate 2D PSFs of a CT device from 2D images of tungsten sphere phantoms. Given a 2D experimental image of a tungsten sphere, a sharp theoretical 2D X-ray image with a punctual focal spot is generated using gVXR. The corresponding PSF h is then estimated from the two theoretical x and experimental y images using a Richardson-Lucy[58, 59] algorithm along with a total variation (TV) regularisation,[60] see Eq. 2.

$$h^{(k+1)} = \frac{h^{(k)}}{1 - \lambda\, div\left(\frac{\vec{\nabla} h^{(k)}}{\|\vec{\nabla} h^{(k)}\|}\right)} \cdot \left(x\left(-(u,v)\right) * \frac{y(u,v)}{x(u,v) * h^{(k)}} \right) \tag{2}$$

(u, v) designates pixel position, $x\left(-(u,v)\right)$ is the 180° rotated image, λ is a regularization parameter, div stands for the divergence operator, $\vec{\nabla}$ is the gradient operator, $*$ is the convolution operator, and multiplication and division are performed component-wise. As proposed by Engelhardt and Baumann,[61] the idea is to retrieve the PSF by inversion of a convolution process, where the experimental image is regarded as the blurred version of the theoretical image. A robust optimization procedure is performed using the simplex method[62] for taking into account the uncertainties on the position of the sphere with respect to the X-ray source. The optimization procedure is depicted in Figure 18. gVXR plays a central role as it generates a new theoretical image $x(P)$ for each new guessed sphere's position P during this optimization procedure. Especially, the fact that gVXR is fast in generating an X-ray image allows it to run the optimization within a reasonable time frame.

4.4.1 Results

Four experimental images of a $1mm$ tungsten sphere were acquired at different voltages and currents (see Figure 19). The X-ray generator is a *Comet MXR 320 HP 11 FB 90* with two focal spots of $0.4\,mm$ and $1.0\,mm$. The detector is a *Varex XRD 1620 xN CS* of $41 \times 41\,cm$ with 2048×2048 pixels and a pixel pitch of $200\,\mu m$. The source-to-detector distance (SDD) was $1150\,mm$, while the source to object (a.k.a. the sphere) distance was approximately measured at $90\,mm$. Figure 20 schematizes the experimental set-up.

Figure 21 shows the corresponding estimated PSFs to the acquired X-ray sphere images of Figure 19 using the algorithm described in Figure 18. Figure 22 displays the corresponding deblurred spheres of Figure 19 using their related estimated PSFs of Figure 21 through the "fast total variation deconvolution" algorithm.[63] Table 5 provides sharpness measurement[64] improvements before and after deblurring using the estimated PSFs. Finally, Figure 23 shows profile comparisons between four acquired X-ray images displayed in Figure 19, the theoretical images $x(P)$ generated with gVXR, and the theoretical images blurred with the corresponding estimated PSFs

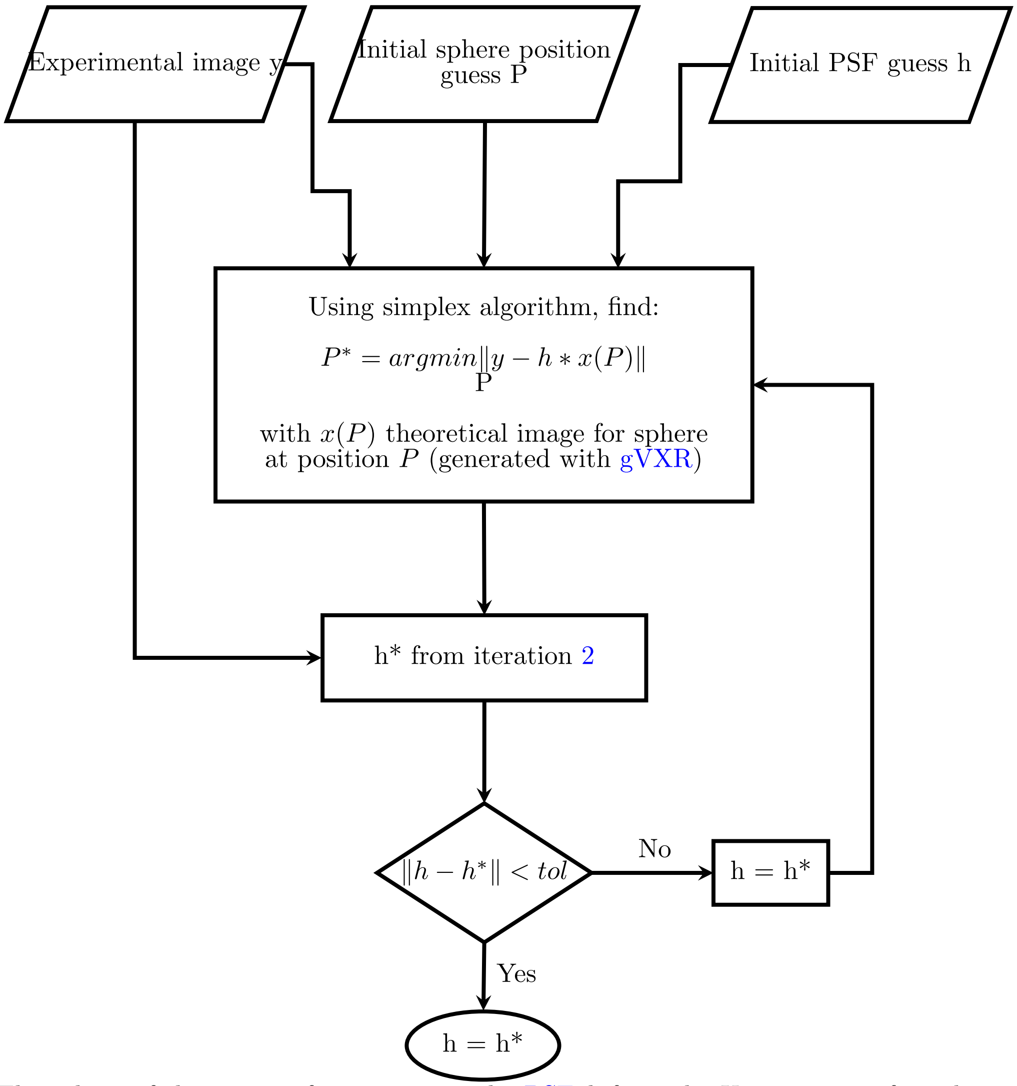
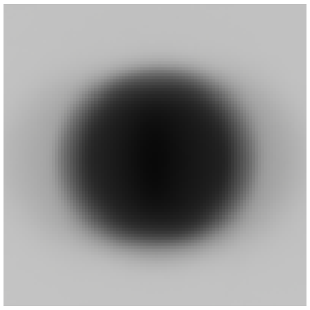
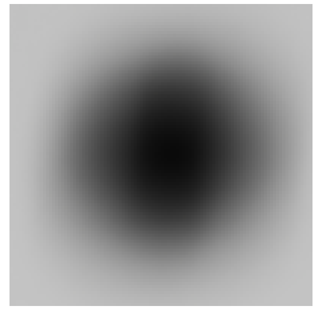


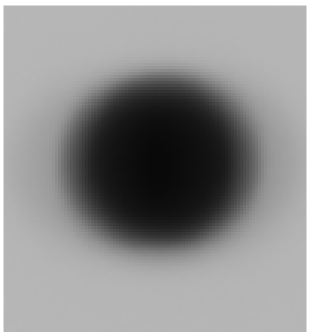
Figure 18: Flow chart of the process for estimating the PSF h from the X-ray image of a sphere. See text for details.



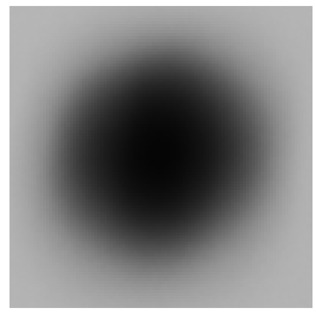
(a) $100\,kV$ at $650\,\mu A$, $0.4\,mm$ focal spot.



(b) $100\,kV$ at $650\,\mu A$, $1.0\,mm$ focal spot.



(c) $240\,kV$ at $3333\,\mu A$, $0.4\,mm$ focal spot.



(d) $240\,kV$ at $7500\,\mu A$, $1.0\,mm$ focal spot.

Figure 19: Acquired X-ray images of a $1\,mm$ tungsten sphere at different voltages, currents and focal spots sizes using a *Comet MXR 320 HP 11 FB 90* tube. Note: for visualisation purposes, images were cropped and contrast enhanced.

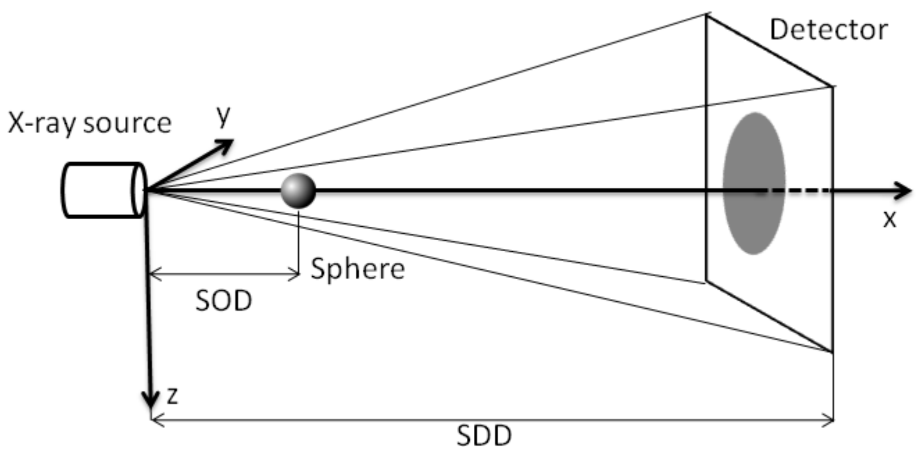


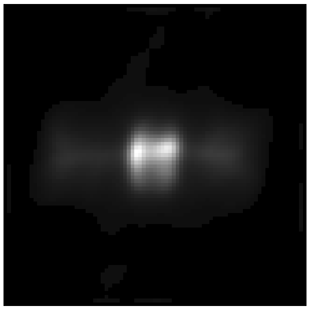
Figure 20: Schematic of the experimental set-up for acquiring the X-ray image of a tungsten sphere.

displayed in Figure 21. From the profiles shown in Figure 23 it can be seen that there is a good agreement between the experimental image of spheres and the theoretical images generated by gVXR blurred with their corresponding estimated PSFs.

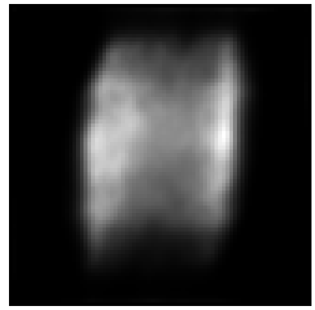
4.5 Machine Learning

4.5.1 Data augmentation

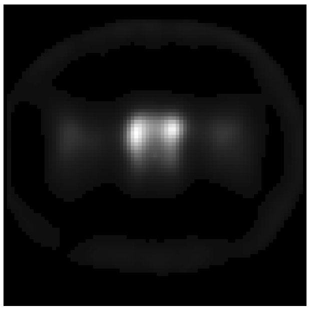
The use of deep learning models has become increasingly common in medical image analysis, particularly for detection, classification and segmentation tasks. To effectively train these diagnostic tools, a large database of labelled images is required to prevent overfitting and promote model generalisation. Collecting these samples is



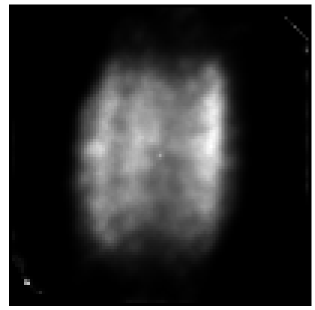
(a) $100\,kV$ at $650\,\mu A$, $0.4\,mm$ focal spot.



(b) $100\,kV$ at $650\,\mu A$, $1.0\,mm$ focal spot.



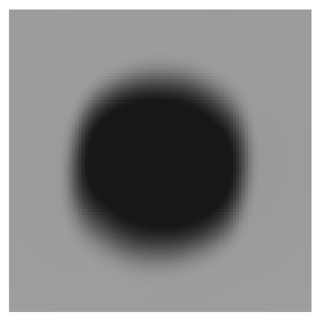
(c) $240\,kV$ at $3333\,\mu A$, $0.4\,mm$ focal spot.



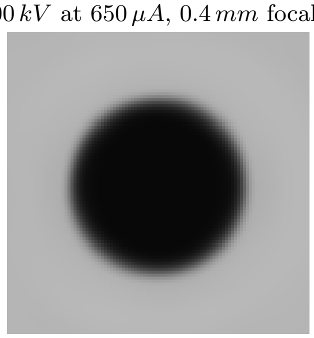
(d) $240\,kV$ at $7500\,\mu A$, $1.0\,mm$ focal spot.

Figure 21: Corresponding 81×81 pixels PSFs estimated to the X-ray sphere images shown in Figure 19.

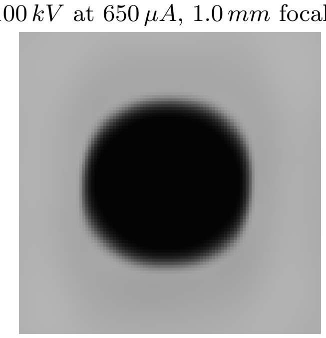
(a) $100\,kV$ at $650\,\mu A$, $0.4\,mm$ focal spot.



(b) $100\,kV$ at $650\,\mu A$, $1.0\,mm$ focal spot.



(c) $240\,kV$ at $3333\,\mu A$, $0.4\,mm$ focal spot.

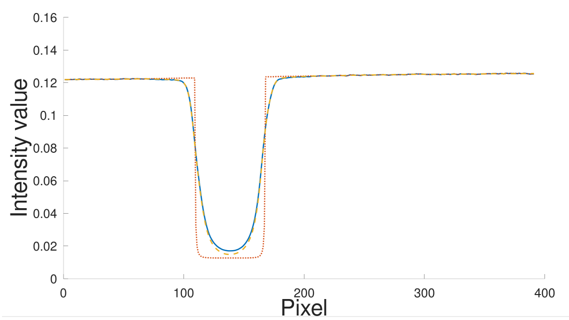


(d) $240\,kV$ at $7500\,\mu A$, $1.0\,mm$ focal spot.

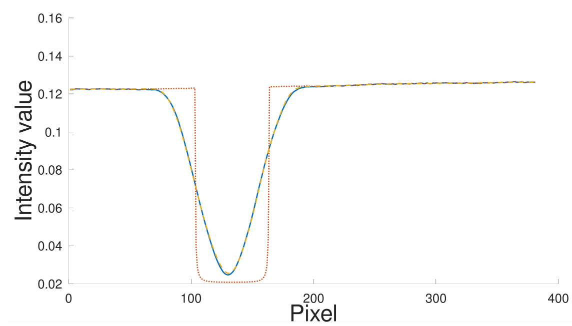
Figure 22: Deblurred spheres of Figure 19, using corresponding estimated PSFs of Figure 21. Note: for visualisation purposes, images were cropped and contrast enhanced.

Table 5: Sharpness indices[64] of the X-ray images of a tungsten sphere before (Figure 19) and after (Figure 22) deblurring, using the estimated PSFs shown in Figure 21.

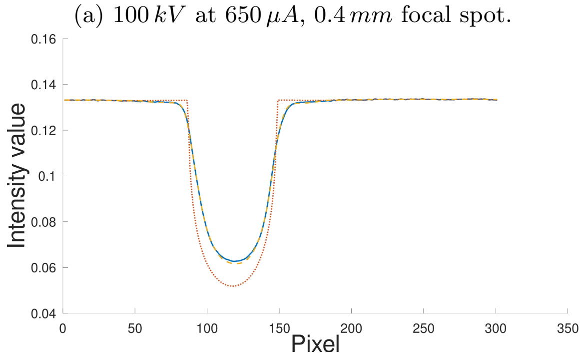
Sphere image	**Before**	**After**	**Improvement**
$100\,kV$, $650\,\mu A$, $0.4\,mm$	0.8916	5.5856	+4.6920
$100\,kV$, $650\,\mu A$, $1.0\,mm$	0.2854	3.2678	+2.9824
$240\,kV$, $3333\,\mu A$, $0.4\,mm$	0.8768	4.2807	+3.4040
$240\,kV$, $7500\,\mu A$, $1.0\,mm$	0.2697	7.8034	+7.5337



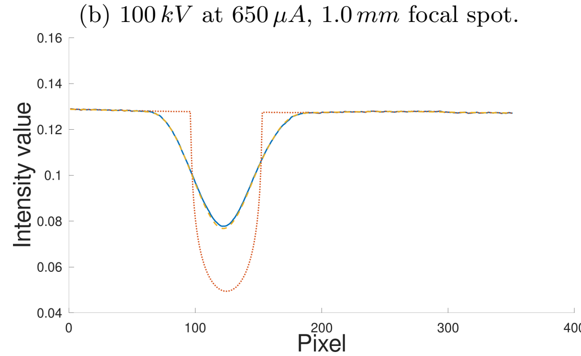
(a) $100\,kV$ at $650\,\mu A$, $0.4\,mm$ focal spot.



(b) $100\,kV$ at $650\,\mu A$, $1.0\,mm$ focal spot.



(c) $240\,kV$ at $3333\,\mu A$, $0.4\,mm$ focal spot.



(d) $240\,kV$ at $7500\,\mu A$, $1.0\,mm$ focal spot.

Figure 23: Profile comparisons between for four acquired X-ray images displayed in Figure 19 (continuous blue line), the theoretical images $x(P)$ generated with gVXR (dotted red line), and the theoretical images blurred with the corresponding estimated PSFs displayed in Figure 21 (dashed yellow line).

a major obstacle due to the time, money, and human resources required to acquire labelled images, as well as data anonymisation requirements.[65]

To improve the performance of these diagnostic tools, various data augmentation strategies have been developed to generate synthetic images along with their corresponding labels. The most common method is the use of generative networks.[66] However, generative networks have limitations, including the need for a large database for training and often limited generalisation capabilities.[67]

The combined use of a virtual anthropomorphic model with gVirtualXRay would allow the rapid generation of a large dataset of labelled synthetic images. This method allows the injection of artificial lesions into the virtual model, as well as the deformation of organ shapes and the modification of scanner settings to replicate the variability found in real datasets. Our preliminary project aims to improve deep learning algorithms for lung nodule detection using a virtual model derived from "Z-Anatomy" - CC-BY-SA 4.0", as shown in Figure 24.

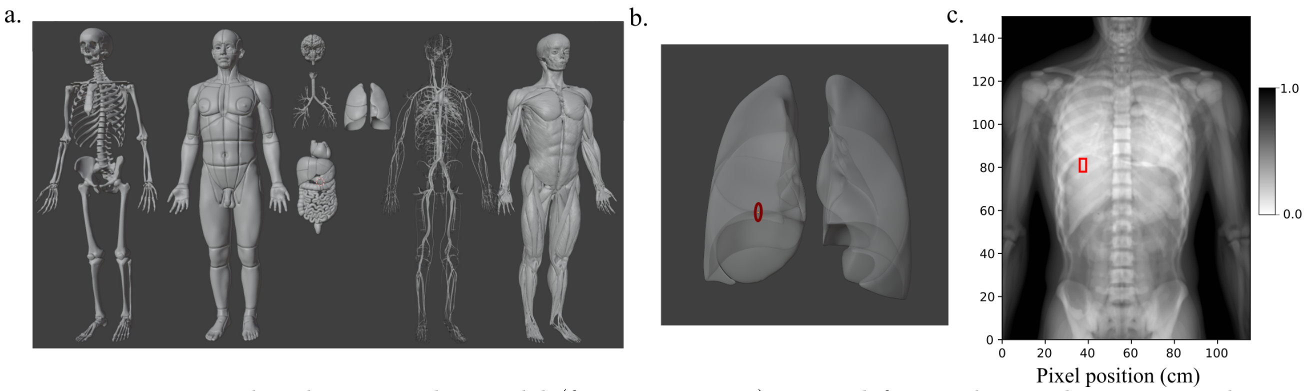


Figure 24: a. Virtual anthropomorphic model (from Z-anatomy). From left to right: meshes represent bones, skin, visceral organs and brain, blood vessels, muscles. b. Example of a synthetic lung model with an artificially inserted nodule (red). c. Example of a generated synthetic chest radiography with a pulmonary nodule. Bounding box in red

4.5.2 Simulation to real transfer of segmentation algorithm

X-ray tomography can be used for the generation of realistic representative volume elements for composite materials modeling. To translate the tomographic reconstruction into a geometry suitable for finite element simulations, the different phases need to be segmented and meshed. The segmentation problem can prove very challenging for low contrast scans where the phases have similar chemical compositions, such as carbon fiber-reinforced polymers. This is especially true in mesoscale scans of woven composites, where the yarns are impregnated with polymer matrix material reducing contrast further. Classical segmentation algorithms like thresholding and watershed often fail in separating the diffuse phase boundaries.

As an alternative to classic segmentation, machine learning based segmentation algorithms are highly investigated and show promise in a wide variety of segmentation tasks. The perhaps largest drawback to machine learning based segmentation is the requirement of very large amounts of training data. As a tomographic scan can take several hours it becomes unfeasible to generate a large enough dataset to train a fairly general segmentation model. Furthermore, labelling the tomographic slices' ground truth value can take several hours per slice for scans with particularly low contrast. In order to circumvent the need for time consuming tomographic scans, and painstaking labelling, synthetic automatically labelled data is of interest. By simulating the tomographic scan, the training data will contain the same type of noise and artifacts one expects to find in a similar experimentally derived image. This helps a machine learning based algorithm learn to ignore these phenomena when being utilized in simulation to real transfer.

The applicability of gVirtualXRay for synthetic training data generation is demonstrated by training DeepLabV3 with a Resnet50 backbone using simulated tomograms.[68] The geometries to be scanned are generated with the open source software TexGen.[69] The segmented ground truths are received by voxelizing the input surface meshes in the same frame of reference as the simulated scan is performed. This is shown in Figure 25. The model

(a) A simulated tomographic slice.

(b) The slice in (a) segmented in to air, weft yarns, warp yarns, and pure matrix.

Figure 25: A tomographic slice of a virtual recreation of a woven carbon fiber reinforced woven composite (a) is shown next to its corresponding voxelization, i.e segmentation ground truth (b).

(a) A tomographic slice of a woven carbon fiber reinforced polymer sample.

(b) A machine learning derived segmentation. Air is shown in red, weft yarns in blue, warp yarns in green, and pure matrix is shown in yellow.

Figure 26: A tomographic slice of a carbon fiber reinforced polymer woven composite (a) is shown together with a machine learning derived segmentation (b).

is trained on 30 synthetic tomograms, where the scan settings and geometry have been domain randomized. Inference on an experimentally derived (not-simulated) tomographic slice is shown in Figure 26. This initial test shows promise, and it is likely a larger dataset and better tuned training procedure will yield better performance.

4.5.3 Spectral CT Data Simulation and Material Decomposition

The capacity of gVirtualXRay to simulate spectral CT data plays a critical role in advancing material decomposition techniques within medical imaging. By generating synthetic data sets across high, low and conventional CT energy spectra, this technology supports the data required for the training process to build deep learning models. These models utilize pairs of reconstructed CT images and material density maps, created through the voxelization of 3D meshes from segmented phantom STL files. This method reliably simulates the varied material properties and their distinct attenuation characteristics at different energies, which is crucial for precise material identification in spectral CT. The synthetic datasets closely replicate the variability and complexity of real clinical scenarios, enabling comprehensive training of algorithms without the significant costs and logistical challenges of gathering extensive clinical data. This approach not only enhances the precision and utility of AI models, but also demonstrates the flexibility of gVirtualXRay in advancing the creation of sophisticated diagnostic tools through synthetic data generation.

4.6 Non-destructive Testing and Materials Science

In this use case automated X-ray-based damage detection and characterization in composite and laminate materials by data-driven predictor models is considered. The overall goal is to train data-driven models with synthetic data only, finally applying the models to real measuring data. NDT of materials and structures can exploit different imaging methods, mainly:

- X-ray radiography (single projection) and Computed Tomography (CT, multi-projection);
- Guided Ultrasonic Waves (GUW) and Ultrasonic Sonography (Pulse-echo method).

The accurate detection of hidden damages, defects, and impurities (e.g., pores) is still a challenge! X-ray radiography is an in-depth accumulation of material properties along the ray axis, making the identification of defects and damages difficult. The modification of GUW signals due to interaction with defects can be even smaller and harder to be identify.

In this work, we address different specimens under test, structure geometries, materials, and defects. They pose different coincidences between material (defect) and image features:

1. Homogeneous aluminum die casting plates with pore defects
2. Composite Fibre-Metal laminate plates (FML, aluminum and PREG layers with impact damages posing layer delaminations, deformation, cracks, and kissing bond defects (loss of adhesive contact between layers).

Automated feature detection and marking in measuring images can occur on different levels:

- Region-of-Interest search;
- Feature marking and maps;
- Damage and defect classification;
- Damage and defect localization;
- Global statistical aggregates (e.g., pore density, defect distribution).

An overview of the NDT case study is given in Figure 27.

One of the major issues in data-driven modeling in materials science is the low variance of data with respect to the parameter space. The number of features in measuring data is often limited. For example, impact damages, breakage defects, or tensile tests can only be applied once to a specimen. To overcome the limitation of the sparse experimental parameter space, simulation of measuring data (e.g., X-ray images) using parameterizable mechanical models should be used.

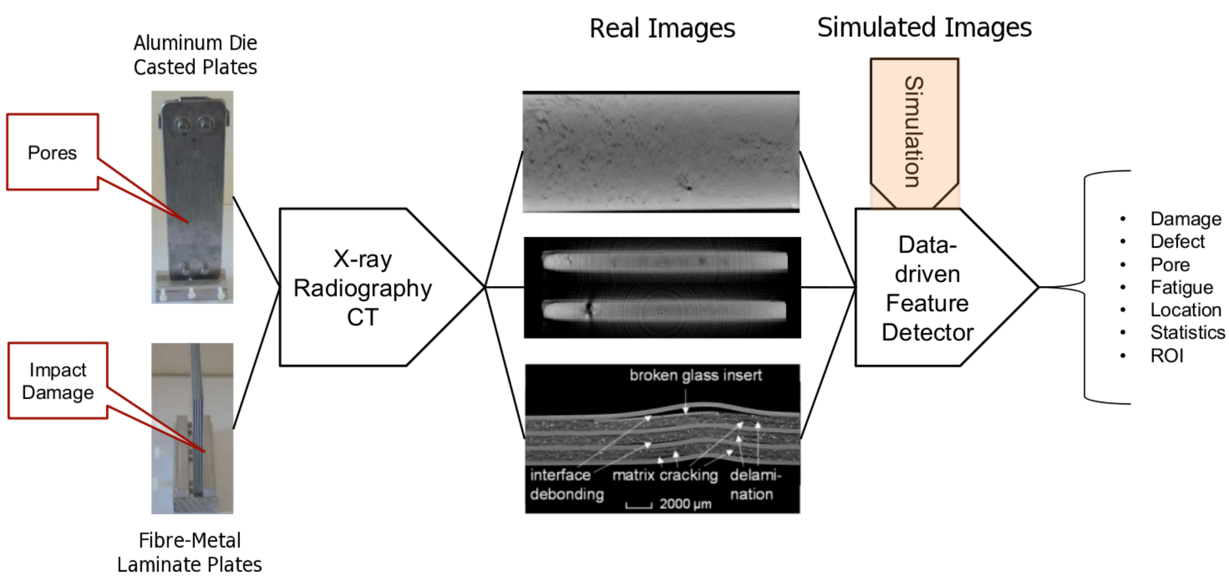


Figure 27: Overview of the NDT case study providing a fusion of real and synthetic X-ray images for data-driven feature detection

4.6.1 Workflow

The entire workflow consists of different model and simulation levels:

1. Constructive Solid Geometry (CSG) modeling of material structures, defects, and damages ⇒ Parameterizable CAD model with ground-truth geometric feature tables (e.g., containing defect positions and sizes);
2. Transformation of the CSG model to a triangular mesh-grid model (STL format) using OpenSCAD;
3. X-ray image simulation (single- and multi-projection) using the gVXR library integrated in our XraySim tool[70] with a post-processing applying Gaussian and Binomial distributed noise to simulate noise from real X-ray image sensing systems;
4. In the case of radial multi-projection images, a filtered-back projection (FBP) algorithm is applied to reconstruct two-dimensional material slices;
5. Training of machine learning (ML) algorithms (e.g., convolutional neural network (CNN)-based pixel classifiers, SAM[71] or Detectron2 models[72]);
6. Test and validation;
7. Application of ML predictor models to real measuring data with experimental validation.

The CAD model is basically a digital twin of real materials and mechanical parts based on X-ray CT and optical micro-slice analysis, as shown in Figure 28.

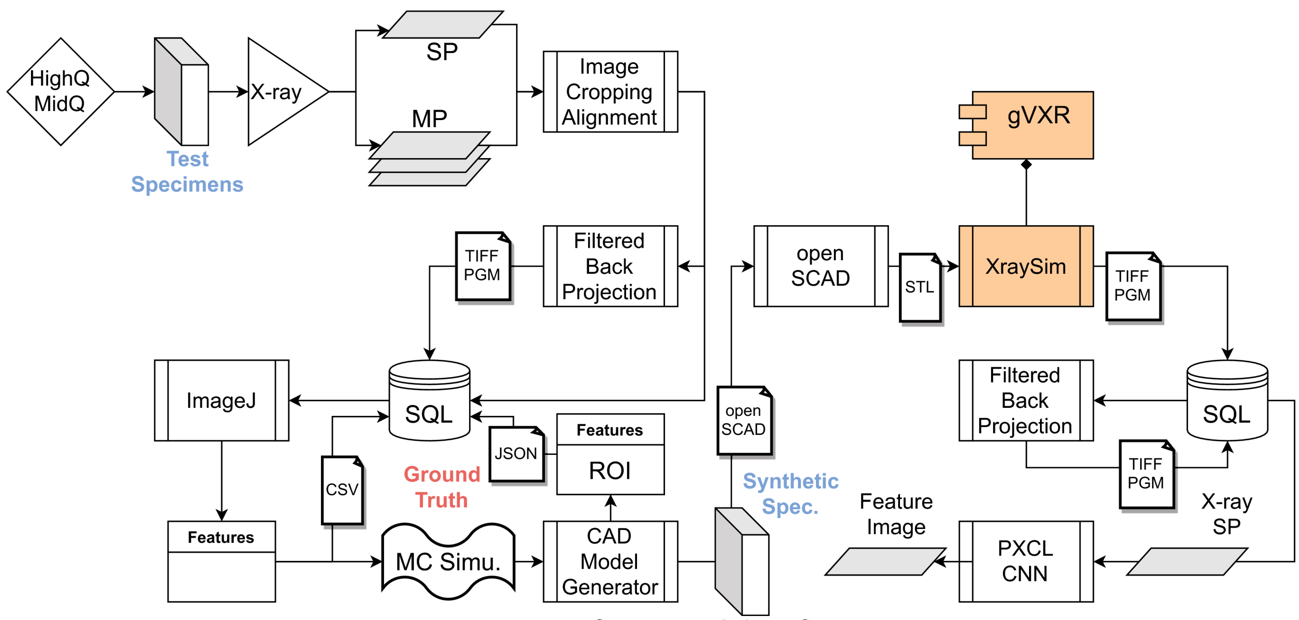


Figure 28: Operational data flow

4.6.2 CAD Modeling

The mechanical model is used for the X-ray image simulation. The model consists of geometric objects with different material densities. CSG modeling is used to create complex 3-dim bodies. CSG consists of additive and subtractive Boolean operations combining shapes. A subtractive operation assumes a host material and a tool shape, e.g., using cylinders for creating holes. We are using the OpenSCAD tool[73] to create triangular mesh-grid models (STL format) that are processed by gVXR.

CSG provides a set of basic operations that can be combined (Boolean chain), as shown in Def. 1.

```
// Basic shapes
cube([sx,sy,sz])
sphere([sx,sy,sz])
cylinder([sx,sdy,sz])
// Geometrical Operators
rotate ([ax,ax,az])
translate([dx,dy,dz])
scale([sx,sy,sz])
// Boolean combinators
difference()
union()
// Extrusion
hull()
rotate_extrude()
```

4.6.3 XraySim

Our XraySim tool[70] integrates the gVXR library and provides a command line tool to create X-ray radiography and radial multi-projection image sets. The input data format is STL, and the output data format is either TIFF or PGM (16 Bits). Both the computational CPU and GPU backends are supported.

Multi-material parts are imported with multiple STL files, each file is associated with a material density. All parts are merged internally with the respective material density.

A typical simulator call for creating a radial (CT) projection set is:

```
xraysim1 -soft \
  -u 2 impactGauss1-1.stl \
  -u 1 impactGauss1-2.stl
  -p 0.07 -e 0.07 -w 1024 -h 1024 -ct 0.45 \
  -S 0 80 65535 \
  -o xray-ct-%04d.tif
```

The `-u density stlfile` pairs load the multi-material parts. This set-up creates 800 radial projection images with a delta angle of 0.45°. The output is saved here in a 16 Bits TIFF format with a full range (65535) scaling corresponding to an absolute X-ray intensity range [0,80], given by the `-S min max` range argument. In radiography, the maximal intensity within the specimen area should be around 70-80% of the full scale of a detector. Recording radial projections of plates is a challenge since the directional material thickness varies significantly with respect to the rotation angle, e.g., a plate of $50 \times 50 \times 2.5$ mm has an absorption ratio of 1:20. Therefore, choosing the right intensity range (if scaled to integer values) is important.

4.6.4 ML-driven Pore Analysis

The pore analysis use-case is described in detail in.[74] The primary goal is automated pore annotation, detection, and characterization in die-casted materials using X-ray radiography images. Due to the missing ground truth data (data annotation), the secondary goal is the training of the data-driven pore ML predictor with synthetic (simulated) data only. For this purpose, a parameterizable pore distribution model was designed based on CT analysis data. The basic CSG-CAD model for a plate with pores is shown in Figure 29 and in App. D. Pores are modeled as ellipsoids. Monte Carlo simulation is used to provide random distributions of locations, sizes, and orientations of pores. The CSG model is a pure substractive model, i.e., the pores remove material and reduce the entire material mass-volume.

Pores will reduce the total material density along an X-ray path, resulting in brighter areas. The gVXR-based xraysim tool was used to create X-ray radiography images (1024×1024 pixels), finally overlayed with Gaussian distributed random noise (average SNR=2 with respect to the pore intensity variation). Examples of noisy X-ray images and the output of the feature marking ML model are shown in Figure 30.

The feature marking predictor model (a classical low-complexity CNN model) could be trained with the synthetic images and applied successfully to real measuring images, producing pore feature maps with realistic

```
rotate ([90,90,90])
difference () {
  rotate ([90,0,0]) cube([100,4,40],true);
  union () {
    translate([3.17,6.14,0.67])
      rotate ([0,0,-1.43])
      scale([1.15,1.12,0.31])
      sphere(r=0.5,$fn=20);
    translate([-16.66,-4.05,0.39])
       rotate ([0,0,40.14])
       scale([0.89,2.21,1.46])
       sphere(r=0.5,$fn=20);
…
  }
}
```

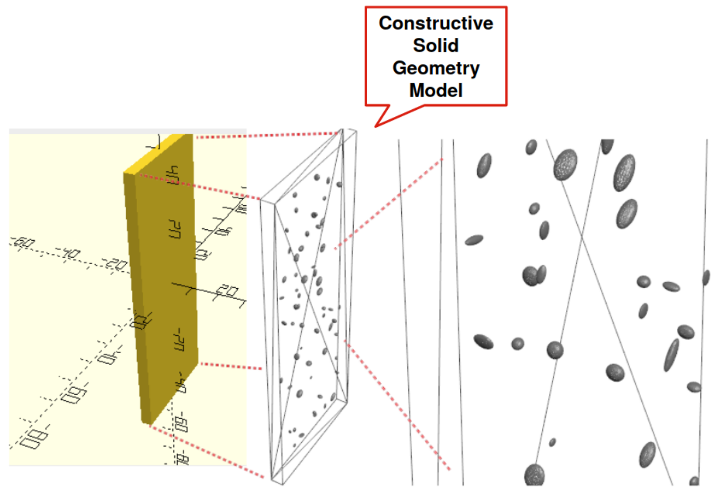


Figure 29: CSG-CAD model of a plate with pores

accuracy. The false-negative and false-positive pixel marking error rates in synthetic X-ray images were below 1%.

Not relevant for the outcome in the work as described in[74] (because in this work only statistical aggregates were evaluated), we found some artifacts in the feature marking of synthetic X-ray images produced by gVXR. Different synthetic models with different random pore distributions and Monte Carlo simulation of the pore parameters showed some false-positive pore markings computed by the pixel classifier CNN model at the same positions, as shown in Figure 30. The measuring noise was created for each image independently. Note that the intensity variation of pores is very low, especially in the case of small pores with sizes below 100 μm. The pixel

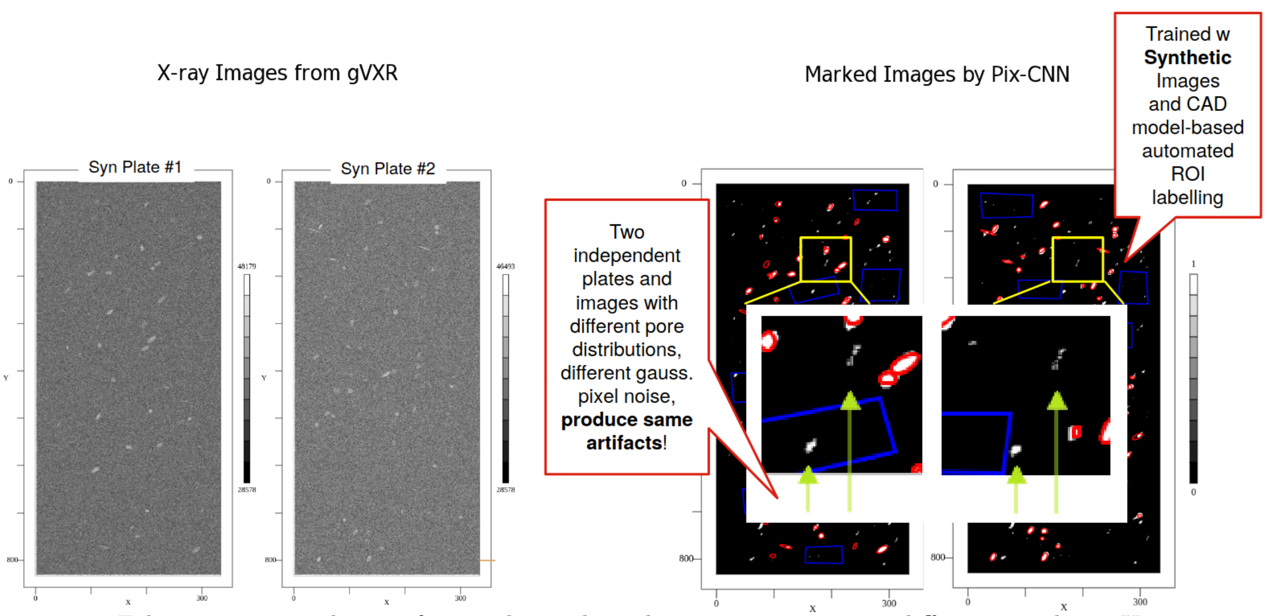


Figure 30: False-positive markings of pores located at the same positions in different synthetic X-ray images with different specimens.

classifier is a black-box model with non-explainable behavior on noise patterns, including numerical noise due to rounding, discretization, and approximation errors. There are no clearly visible artifacts in the original gVXR images, only the predictor model produces markings based on unknown patterns in the source images. This is not a flaw of gVXR, but should be considered if black-box ML models are trained using synthetic X-ray images, produced, e.g., by gVXR.

4.6.5 ML-driven Damage Analysis

In contrast to the pore analysis use-case, impact damage analysis is still a challenge[75] and a work in progress. Impact damages in laminate structures create deformation and delamination, and at high impact energies, cracks occur, too. With respect to CSG modeling, the addition of defects is a subtractive operation, too, but must satisfy constraints, i.e., preserve the total mass-volume. A defect does not remove material, it shifts material, only, and considering non-compressible materials, the entire mass-volume does not change. An impact damage can be approximated with a Gaussian-like deformation, with its profile computed by a Gaussian-like function with a damage radius r and height h:

$$\sigma = 0.35 + \left(\sqrt{r}^{0.45} - 1\right), y(x) = h\left(\frac{e}{\sqrt{2 \cdot \pi \cdot \sigma^2}}\right)^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}}$$

Figure 31 shows the CAD model of a laminate plate consisting of three aluminum and two fiberglass layers (each layer has a thickness of 0.5 mm), and an impact damage in the centre of diameter of 4 mm and a height of 0.8 mm. A set of 800 radial projections was created using gVXR and finally reconstructed to cross-section

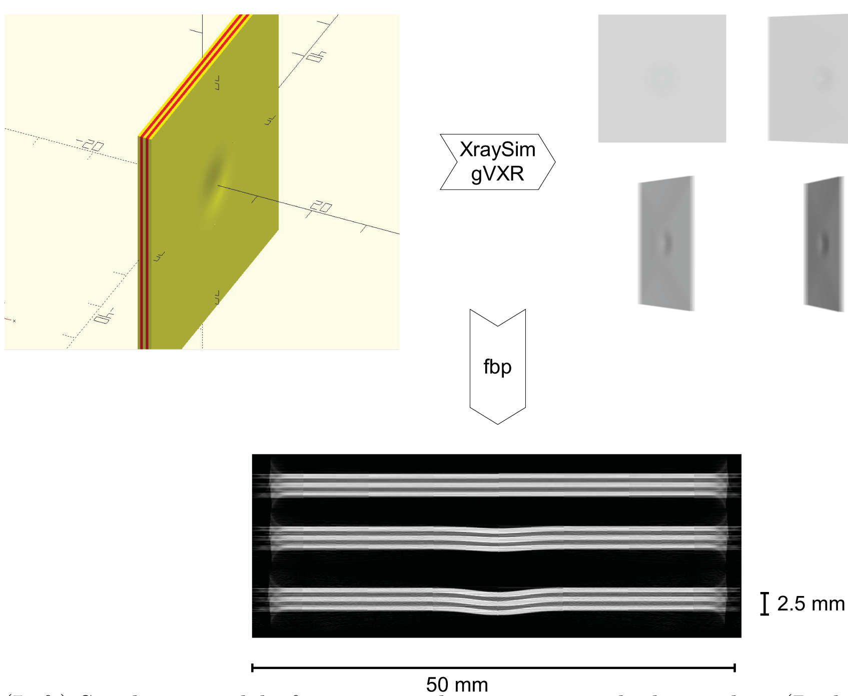


Figure 31: (Left) Synthetic model of an impact damage in a multi-layer plate (Right) Examples of radial X-ray multi-projections (Middle) Reconstructed material density slices at three different cross-section positions

slice images using a classical sine-wave filtered back-projection algorithm (using our *fbp* tool). The CSG model is shown in App. D. The impact damage could be reconstructed with a high resolution, comparable to damages visible in real specimens and CT analysis, but showing small material inhomogeneities as a result of a material distribution violation in the CAD model. The synthetic data can be used to refine and validate the damage models as well as damage feature marking, classification, and characterization algorithms.

4.6.6 Results

The simulated X-ray images created by gVXR show high accuracy compared to real measured radiography and reconstructed CT images. The xraysim tool provides easy access to the underlying gVXR library, supporting CPU and GPU back-ends. Radial CT scans are performed automatically, reusing the imported geometric material model. We showed a complete modeling workflow, starting with a CSG model that can be easily generated and parametrized, then using OpenSCAD to create the material model with a triangular plane model processed by gVXR. Multi-material specimens must be provided separately by parts of the same material density. The different parts are merged in xraysim to one material model. Complex components and material combinations can be easily created. Even very complex models like fiberglass layers consisting of tens of thousands of fibers (approximated by cylinders) can be processed efficiently and accurately by gVXR. The computed X-ray image patterns correspond to real image observations. The synthetic data can be used to refine and validate the damage models as well as damage feature marking, classification, and characterization algorithms.

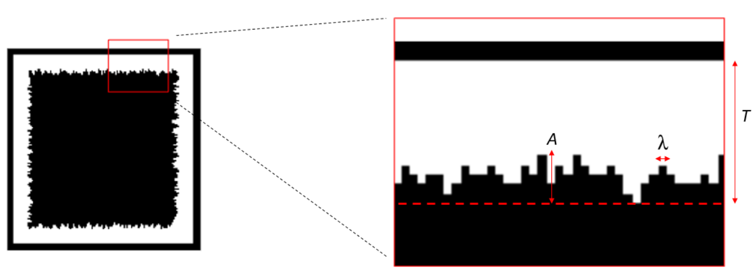
4.7 Characterization of surface roughness for additive manufacturing using Deep Learning and X-ray CT

In this project, the use of gVXR is coupled with Machine Learning to improve the resolution of CT scans. Additive manufacturing (AM), also known as 3D printing, is currently seen as one of the most promising methods to fabricate components for a wide variety of industries, including the aerospace, automotive and medical fields. As additive manufacturing techniques are quickly growing and becoming industry benchmarks, it is important to have rapid and easily applicable quality control and inspection techniques.

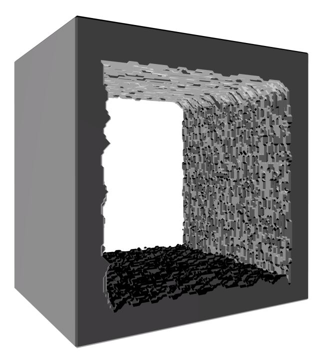
One of the main issues posed by surfaces in 3D-printed objects is the presence of defects and roughness. XCT is currently the only valid method that can determine complete internal and external geometry without constraints of traditional tactile and optical techniques. However, XCT is a more expensive metrology tool and usually takes a longer time for part scanning and data processing, e.g., surface determination (identification of the material boundary of the scanned part), which limits its usefulness for industrial applications.

Additionally, XCT scanners do not always return data that clearly shows the edge of a surface due to issues related to the scanner's resolution, artifacts and to the fact it is required to segment the data to identify an object's surface.

This project investigates the use of deep learning (DL) as a tool to quickly determine where the "real" surface is in a reconstructed XCT image with a good level of precision. To do so, synthetic virtual rough surfaces were created to obtain the "ground truth" data (Figure 32a) using a newly developed plugin written in ImageJ macro language. The macro allows the generation of a cuboid hollow shape presenting various surface roughness (Figure 32b). These cuboids were successively virtually scanned using X-ray simulation code gVirtualXRay. Using simulations to generate virtual scans and data provided several advantages over real experimental scans: a full



(a) On the left, a "slice" of the cuboid can be seen. In the middle, a zoomed-in section of the surface is shown, with the A being the amplitude, I the wavelength and T the thickness. The dashed red line is the baseline from which the rough surface was constructed.



(b) Three-dimensional representation of one of the cuboids.

Figure 32: Synthetic data created by the software.

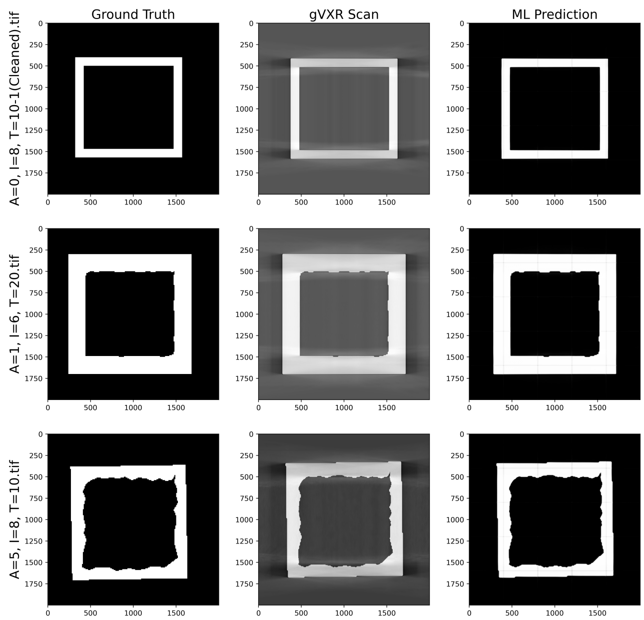


Figure 33: Comparison of the ground truth, gVXR scan and neural network prediction for 3 different cube slices: the cube parameters are shown at the left of each ground truth image. The x and y axes are labelled with the pixel numbers.

CT scan can take several hours to complete, whereas simulations are a time-efficient tool that can rapidly generate data for analysis. The trained neural network demonstrated significant success in improving the resolution of the rough surface images, as evidenced by a relative mean average error on the test data of 1.92% (Figure 33). Furthermore, the deep learning output exhibited a 17-fold improvement in similarity to the true data compared to the input when measured by the mean square error. The main sources of error were found to be related to the lack of background noise inclusion in the scan simulations, and to the relatively small size and limited variety of the training data.

Finally, a comparison between a real XCT scan of a 3D printed cuboid with its respective DL predictions is shown in Figure 34. It is evident in this case, that the streaking artifacts at the top and bottom of the cuboids are carried through by the neural network. This suggests that the virtual scans do not capture all of the features and artifacts present in a real scan.

4.8 Comparison of a manufactured object with its original CAD design

When a scanned object is manufactured from CAD, it is possible to register simulated projections of the CAD model onto the X-ray projections taken during a CT scan acquisition. If the simulated projections are acquired using the same geometrical properties as the experimental scan, then we have a perfect geometrical alignment of the simulated CAD model and the object in the CT device. If we can segment the experimental CT scan, then we can extract the 3D surface of the scanned object, and we can compare the two surface meshes to quantitatively analyze the geometric manufacturing discrepancies.

We report here on the evaluation of an optical component (mirror petal) for a nanosatellite ($<$ 10 kg) produced with AM (see Figure 35).[76] Figure 36 illustrates the whole process. An internal lattice (triply periodic minimum surface diamond lattice) was used in the design (CAD) to reduce the mass by 44%. Finite element analysis and prototyping experiments were conducted to ascertain the robustness of the design to ensure optimal performance and manufacturability. The selected model was 3D printed using laser powder bed fusion in an aluminum alloy (AlSi10Mg) and postprocessed using conventional machining, followed by single point diamond turning to generate the reflective surface. XCT acquisitions were performed to assess the presence of porosity or fractures and the accuracy of the print versus the CAD. The same scan was replicated virtually after 3D registration, i.e. same geometrical properties, material compositions, kV and filtration. Once the simulated projections of

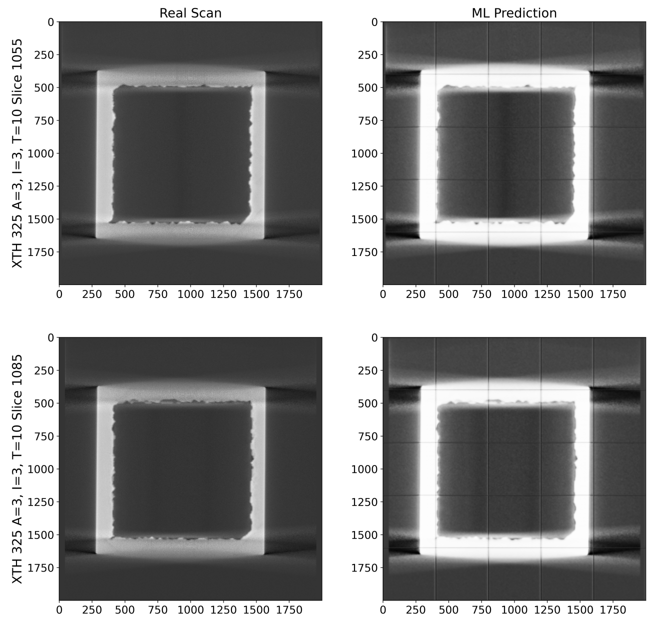


Figure 34: Comparison of two XCT reconstruction slices from a real XCT scan done with the Nikon Xtek XTH 320 scanner to the prediction made by the developed neural network. The values at the x and y axis display the pixel number in the image.

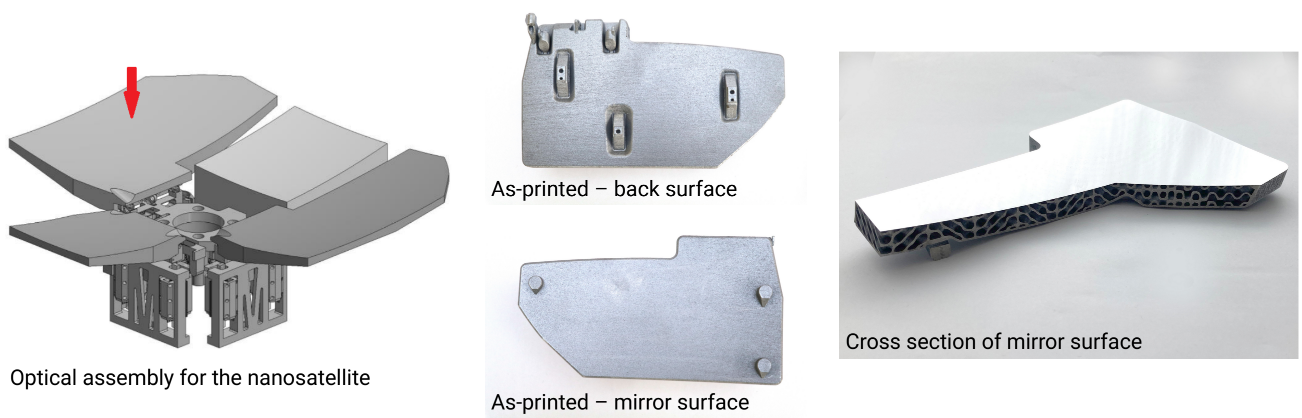


Figure 35: (left) the nanosatellite optical assembly highlighting the mirror petal; (middle) the as-printed design that underwent XCT acquisition; and (right) a cross section through the reflective surface showing the internal lattice.

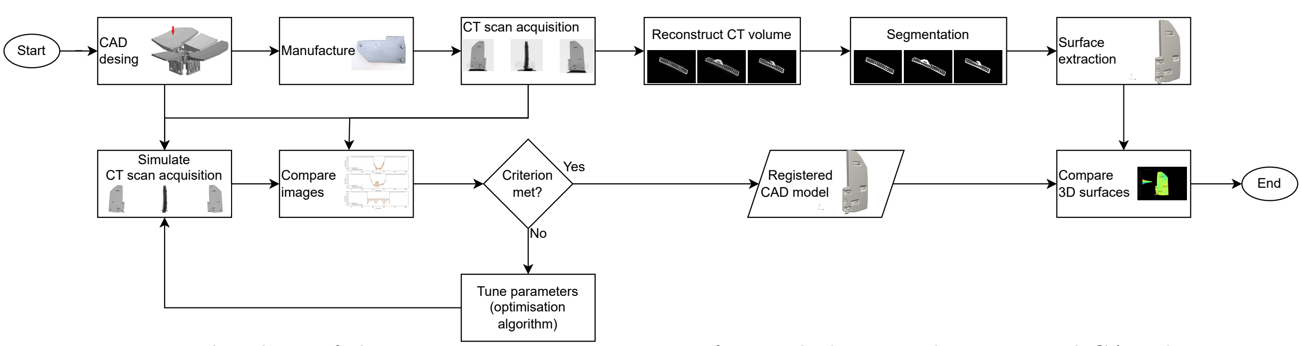


Figure 36: Flowchart of the process to compare a manufactured object with its original CAD design.

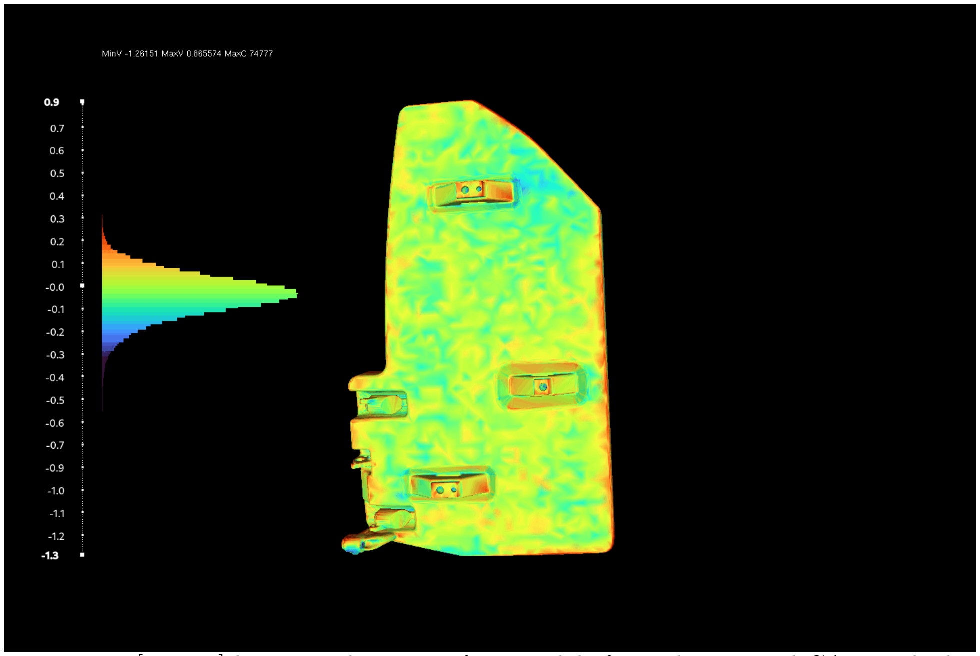


Figure 37: Discrepancies [in mm] between the 3D surface models from the original CAD with the one extracted from the segmentation of the experimental scan.

the CAD model are perfectly aligned with the experimental ones, we compare the 3D surface models from the original CAD with the one extracted from the segmentation of the experimental scan (see Figure 37). Table 6 shows that the 3D printing errors are very small, 0.02 mm on average, which is well below the size of a voxel (0.22 mm).

Table 6: Displacement (error between manufactured and CAD).

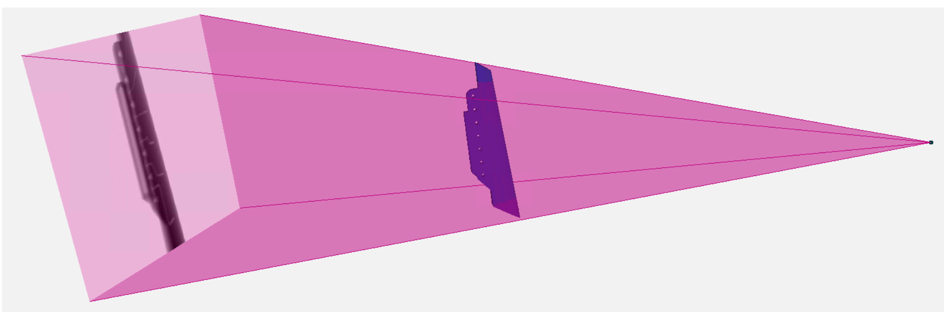
Absolute value	mm	pixels
Min	0.00	0.00
Max	1.27	5.76
Mean $\pm$ stddev	0.02 ± 0.06	0.11 ± 0.26
RMS	0.06	0.28

4.9 Assessment and Optimization of Industrial XCT Performance

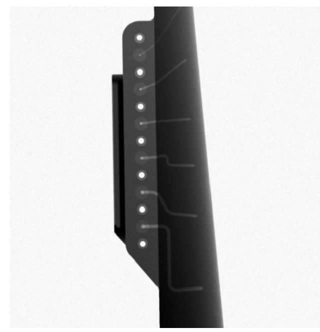
It is well known that the performance of an XCT scan depends on many factors. Furthermore, the performance of an XCT scan is likely to vary across the reconstructed volume of a given component. For example, in NDT for industrial components using polychromatic sources, the contrast generated by a defect in the part is expected to vary depending on the amount of material being penetrated across each projection in the scan. The amount of scatter will also vary across the images, and the prevalence of reconstruction artifacts will vary across the volume.

While the performance will clearly be affected by modifying the X-ray source parameters and modifying the pose of the part, it is difficult to gain a good understanding of the spatial variability of the inspection performance for a given setup without running a significant experimental campaign utilizing a large number of samples with defects deliberately seeded at specific locations. However, experimental testing can be unattractive, not only due to associated high cost and material waste, but it requires the development to be mature enough for component manufacturing. This is often not the case for newly developing products, and often the design may need to be re-iterated, and sometimes re-designed to alleviate any identified inspection challenges.

There is, therefore, a tremendous value in X-ray simulation tools, not only to estimate the expected inspection performance but also to maximize the inspection performance for a given component. Towards this end, this section describes how gVXR has been used to assess and map the inspection performance for an industrially-relevant component, and then maximize this performance by geometrically optimizing the setup. The part geometry used was the Digital Reconfigurable Additive Manufacturing facilities for Aerospace (DRAMA) additively manufactured 250 × 15 × 70 mm aerofoil in Ti-6Al-4V with laser powder bed fusion. The system being modelled was based on a diondo d2 system, with a 225 kV source and a 400 mm by 400 mm detector, shown in Figure 38. For computational speed, the detector resolution was set to 750 × 750 pixels, a quarter of the genuine system. The computational hardware was a workstation using an AMD Ryzen Threadripper 1950X, NVIDIA Titan XP, 128 GB RAM (2800 MHz) and NVMe SSDs.



(a) The arrangement of the part with respect to the source and detector. Note that the 3D visualisation is a built-in feature of gVXR. It can be used interactively.



(b) The initial projection.

Figure 38: The initial simulation setup for the DRAMA aerofoil blade component, prior to optimization.

4.9.1 Framework Generating Synthetic Defect Indications

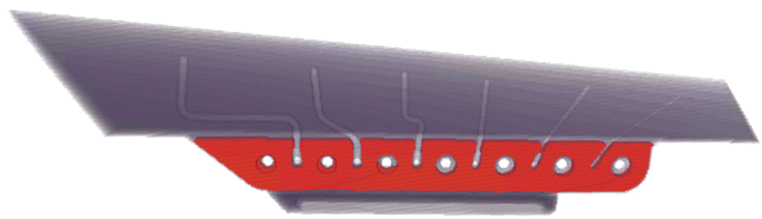
The MTC has developed a Python framework to automatically configure gVXR setups, seed defect geometries into a part, and track the positions and orientations of these defects in 3D space as the part is reoriented. From this, defect positions in each projection and the reconstructed volume are also computed, which will become important in the following sections.

Depending on the application, the user can seed defects in a stochastic manner, weighting defect locations to specific regions of the part, or with specific behaviors such as clustering or proximity to the component surface. Furthermore, the defect geometries used can be a specific defect definition (such as a pore of a specified diameter), or can be taken from a library of pre-saved defect geometry STLs extracted from genuine XCT datasets. The size and material of defects can be redefined as desired, and multiple defect types can easily be seeded into the same part. The intention of this framework covered two goals, firstly, to provide a method for quickly generating a large number of randomized defect indication images for training automated defect recognition algorithms (using randomized locations and defect definitions), and secondly, to provide a method for assessing the variability of defect detectability across the part to locate inspection blind spots (using predefined locations and a rigid definition of a defect). The work presented in this section focuses on the second application.

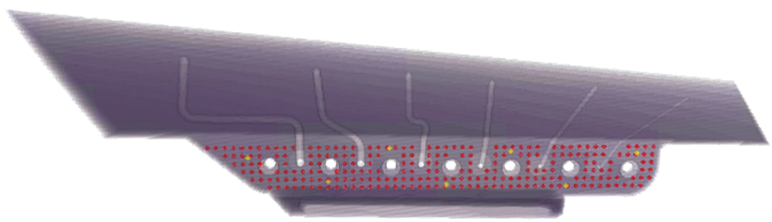
While the detector resolution has been defined to be 750 by 750 pixels, a super-sampling factor of 3 was used (resulting in a simulated resolution of 2250 by 2250 pixels), and then re-binned to the original resolution. Across several X-ray simulations we have used, we have found this super-sampling approach to be essential for eliminating aliasing artifacts around features that span only a few pixels at the original resolution.

4.9.2 Calculating an Inspection Performance Map

To assess the inspection performance before and after optimization, this work sought to create an inspection performance map similar to that shown in previous work.[77] For generating predefined locations, a Boolean voxelized representation of the part geometry was derived using a voxelization algorithm on the part STL with a voxel spacing of 0.4 mm. A region of interest voxel map was then defined by taking voxels which overlapped with the bracket attachment section, shown in Figure 39a. The region of interest was then used as a weighting array for



(a) The bracket region of interest is highlighted in red with respect to the aerofoil geometry.



(b) Defect seed locations (not defects themselves) are highlighted. The locations highlighted in red were used in the inspection performance assessment, while the locations highlighted in yellow were used to compute the objective function during the optimization.

Figure 39: A voxelized representation of the geometry showing regions of interest used for the analysis and optimization.

the defect generation algorithm (weight = 1 inside this region, and 0 outside). In this case, the weighting volume was slightly offset from the surface of the part to avoid creating surface-breaking defects, as the performance metrics used here do not work well around the sharp edges of the part. From here, a grid sampling algorithm was used to take every 7th voxel and use these as hypothetical defect locations, corresponding to a sampling spacing of 2.8 mm. The result was a total of 234 hypothetical defect locations, shown in Figure 39b. The defect definition used was a spherical void of diameter 0.5 mm.

Each defect in the grid was assessed using the approach shown in Figure 40. Each defect needs to be assessed individually, and rather than calculating a full projection set for each defect, it was found to be more computationally efficient to compute a "clean" (non-defective, no noise), projection stack for a given setup, and then assess each defect by simulating small 11×11 pixel patches of the detector, with each patch centred on the defect location as the part rotates around the scan. Defective image patches were then stitched back into the clean projection set to produce a full projection image set containing the defect. An additional benefit of computing patches is that the computational impact of super-sampling is practically negligible when sampling such small patches of the detector. Once each defective projection stack was generated, the stacks undergo Poissonian noise post-processing and reconstruction operations using the ASTRA toolbox. The seed used in the

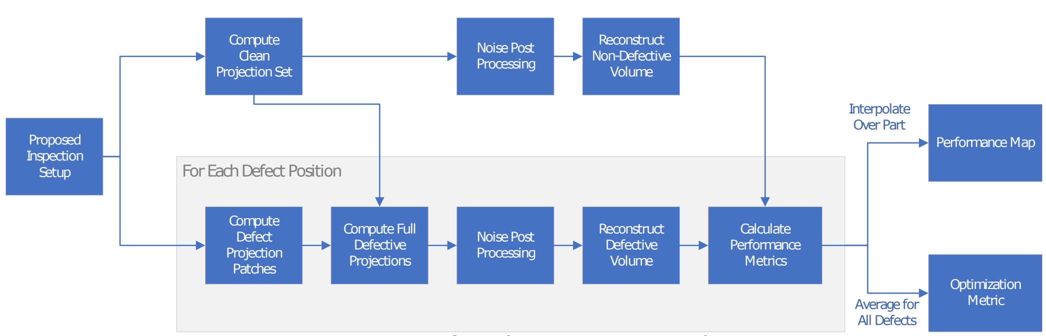


Figure 40: Diagram showing the simulation workflow for generating defective data and calculating performance metrics.

noise generation was identical for each projection, such that the underlying noise did not change, but was scaled based on the intensity changes incurred by adding each defect. Finally, the results were interpolated across the region of interest to produce a final performance map that will be shown in the next section.

While there are several approaches for assessing image quality,[78,79] there are usually additional bespoke image processing steps required to implement these approaches. The metric used here was based on the Contrast-to-Noise Ratio (CNR).

To calculate the contrast metric for each defect, reconstructed $30 \times 30 \times 30$ voxel volumetric patches were compared with and without the defect. A difference volume was computed, and a 3D mask of the defect was extracted using a slight Gaussian blur (with a sigma value of 0.5 pixels) followed by Otsu binarization thresholding, and a dilation operation. The difference values that overlap with the defect mask were extracted and averaged, thus giving a representation of the defect contrast.

To calculate the noise metric in the vicinity of each defect, the non-defective volumetric patch underwent edge-preserving bilateral filtering and was subtracted from the original to produce a volume that should only contain noise. This patch was then overlaid with the defect mask calculated earlier to extract voxel noise in the defect vicinity. Finally, the standard deviation of these values was taken to give a representation of the noise. By dividing the contrast metric by the noise metric, the final CNR metric can be deduced. As will be seen in the visualizations in the following section, the noise metric is quite susceptible to local variations, which we expect is related to the resolution of the sampling grid used in Figure 39b. Many different approaches were tested to reduce this variability, but the metric described was the best solution found.

To process all defects and complete the inspection performance map, the computation took 74 minutes in total, with 33 minutes spent calculating projection images, 30 minutes performing reconstructions, and the remaining time spent with additional processing of the data. This could be reduced by further optimization of the implementation, but for the purposes of making a performance map, this computation time was acceptable.

4.9.3 Optimizing the Setup

The SciPy Nelder-Mead optimizer was used to optimize the geometrical parameters of the setup using the setup shown in Figure 38 as a starting point. The six parameters addressed by the optimization were: two parameters to set the orientation of the part (the rotation around the CT axis was ignored), three translational offset parameters to position the part with respect to the CT axis, and finally, the source-to-object distance (the source-to-detector distance was set to the maximum of the system).

While the time taken to compute a full performance map is acceptable for establishing a comprehensive understanding of the spatial variability of the inspection performance, a simplified computation is required to quickly assess new proposed setups within the optimization loop. In this case, 7 positions were picked to evenly cover the bracket zone (highlighted in yellow in Figure 39b). For each proposed setup, the same workflow outlined in Figure 40 was used to compute performance metrics across all 7 positions, which were then averaged to create a single objective metric for the optimizer. In this case, each evaluation required approximately 100 seconds. Depending on the setups being proposed, some defect indications may lie outside the field of view of the scan for some projections, which can be easily detected using the known spatial mapping between the setup space, the detector space, and the reconstructed volume space. In these cases, these evaluations were skipped over without simulation.

The timeline of the optimization is shown in Figure 41. Due to the anticipated long duration of the optimization, the optimizer was set to terminate when it reached a maximum evaluation count of 850 (the number of iterations able to be completed in 24 hours). It can be seen that the optimizer was able to improve the calculated inspection performance considerably, but it was unclear how many more iterations would be required to achieve true convergence. The final simplex used by the optimizer spanned less than 0.5 degrees across the rotational parameters, less than 0.5 mm in the translational offsets, and approximately 1 mm for the source-to-object distance parameter, suggesting the optimizer had mostly converged. The best setup found is shown in Figure 42.

The inspection performance metrics before and after the optimization are presented in Figure 43, where it can be seen that the calculated performance has improved considerably. Figure 44 shows defect indications evaluated at each position, where it can be seen that the defect indications appear significantly improved, both

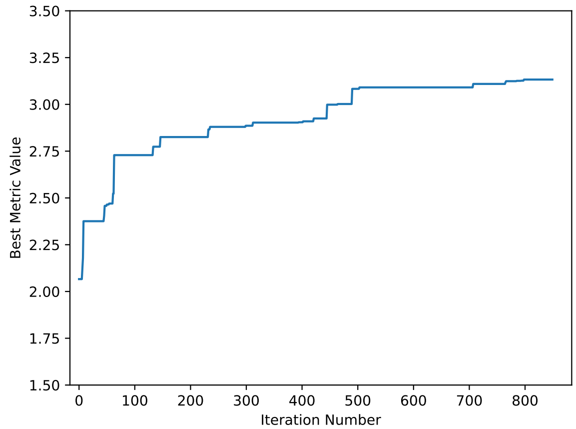
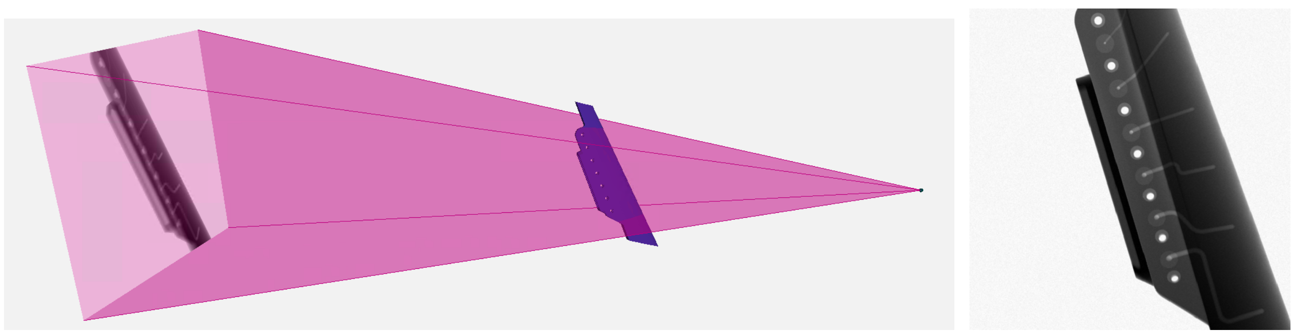


Figure 41: The timeline of the optimization, spanning roughly 24 hours. It can be seen that the model likely did not reach full convergence, however, good progress had been made.



(a) The arrangement of the part with respect to the source and detector. (b) The initial projection.

Figure 42: The initial simulation setup for the DRAMA aerofoil blade component, after optimization.

in magnification and contrast. The most notable change between the initial and optimized setup is that the magnification has been increased to focus the scan on the bracket section. Additionally, the aerofoil has been tilted slightly, which explores the trade-off of incurring slightly longer material path lengths across the part (reducing performance), at the benefit of reducing the vertical extent of the bracket to increase magnification (increasing performance). It can be seen that the tip section of the bracket region lies outside of the scan. This is because the optimizer did not check that all parts of the bracket were in the field of view (only the 7 positions were checked) but in future, more points along the outer extent of the region of interest should be factored into this field of view check.

4.9.4 Future Work for Assessing and Optimizing Inspection Performance

This section has described an approach to leverage the GPU-accelerated X-ray simulation tools offered by gVXR for industrial NDT inspection. The approaches described are expected to be useful across the product development lifecycle; from design validation and inspection optimization, through to inspection qualification, where the outcome can be used to target the scope of experimental qualification trials.

For optimization, the trade-off between simulation fidelity and metric evaluation time is paramount for making progress in a reasonable time frame, and the performance of gVXR is sufficient for this goal. In future, we plan to extend the model to include scatter estimation, through Monte-Carlo techniques,[80] or through deep-learning techniques.[81] However, the feasibility of implementing these within an optimization loop remains to be determined.

Future work will also seek to extend the optimization parameters to include the source and filtering parameters. To do this, a digital twinning exercise to match the simulation behavior to the genuine system using

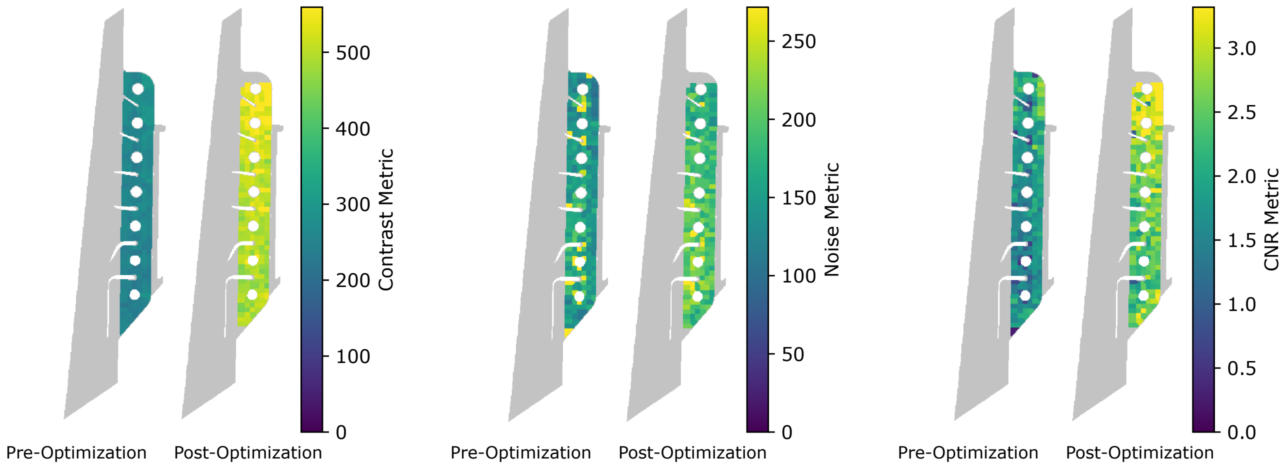


Figure 43: The performance metrics for the part bracket pre- and post-optimization. A drastic improvement of the contrast metric was seen, with only minor changes seen in the noise metric. The result for the CNR metric is significantly improved across the bracket after the optimization.

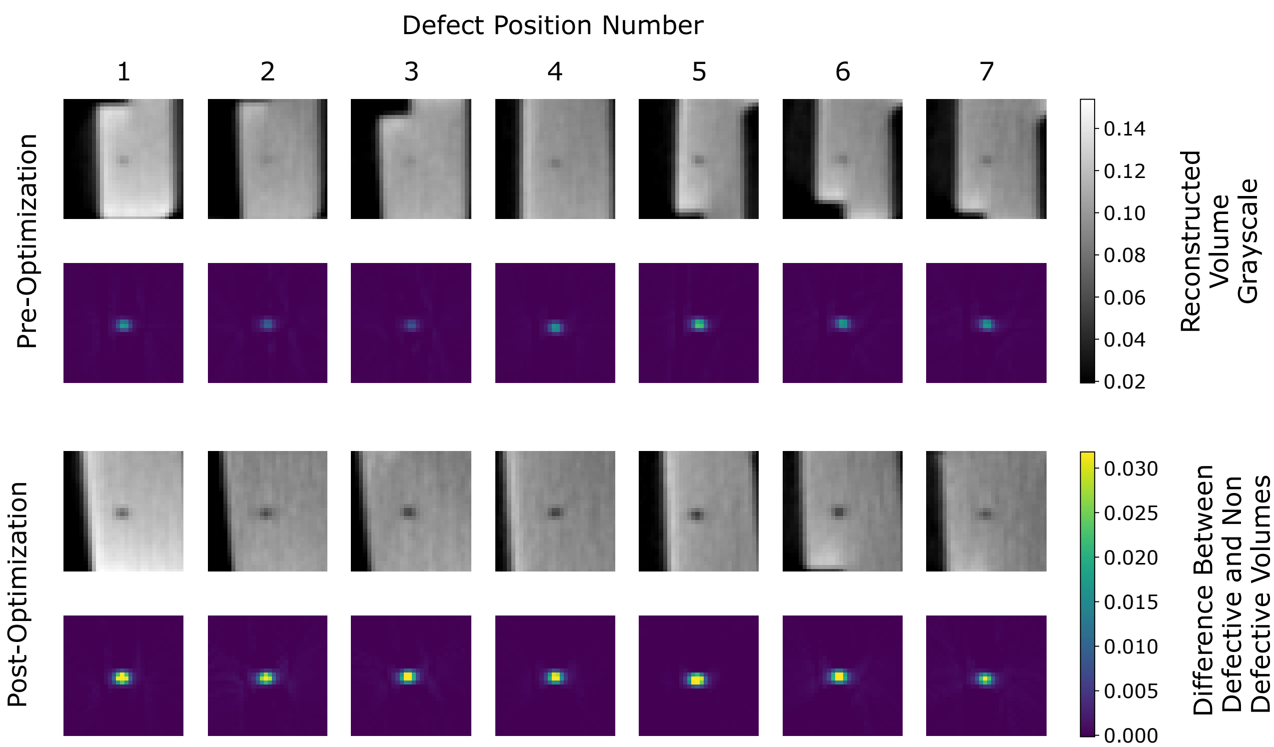


Figure 44: Defect indications for the seven positions before and after optimization. The grayscale range used for plotting is identical across all of the reconstructed volume patches. The grayscale range used for plotting the difference volume patches is also identical across all different images.

experimental data (as described in Section 4.1) will enable the simulation engine and optimizer to reliably explore a range of source parameters. Future work will also look at optimizing the inspection performance beyond defect CNR to consider performance metrics for performing dimensional metrology with XCT.

5. CONCLUSION

This paper presents gVXR, an open-source framework for simulating X-ray images in real time using GPUs. We have demonstrated gVXR's versatility and applicability across various domains, including education, experimental setup optimization, digital twinning, machine learning, non-destructive testing, and materials science. Extensive validation efforts, including comparisons with Monte Carlo simulations and real experimental data, have confirmed the accuracy of gVXR's simulations. The framework's ability to rapidly generate verifiably accurate X-ray simulations enables researchers and practitioners to explore complex scenarios, optimize experimental parameters, and develop novel approaches in X-ray imaging and analysis. As gVXR continues to evolve, potential areas for future development include improved scatter estimation and enhanced support for spectral imaging. The open-source nature of gVXR enables collaboration and innovation, encouraging future advancements in X-ray simulation and analysis across multiple disciplines.

ACKNOWLEDGMENTS

The development of gVirtualXRay is currently supported by the Ada Lovelace Centre. It was initially co-funded by the European Union through the FP7-PEOPLE-2012-CIG - Marie-Curie Action 'Fly Algorithm in PET Reconstruction for Radiotherapy Treatment Planning' under grant # 321968). F. Vidal thanks NVIDIA Corporation for the donation of the NVIDIA TITAN Xp GPU used in the development and validation of gVirtualXRay. F. Vidal, I. Mitchell and S. Middleburgh acknowledge the Diamond Light Source for beamtime # MG29820 on 'Real experiments, hi-fidelity simulations, machine learning and GPUs: Toward the virtual design of nuclear fuel rods'. I. Mitchell and F. Vidal would like to thank Taith for a travel grant awarded to Vidal and Mitchell that made it possible to develop the digital twin of the new dual-beam XRCT laboratory equipment of the MateIS laboratory (Lyon, France). Y. Chahid and C. Atkins acknowledge the UKRI Future Leaders Fellowship 'Printing the future of space telescopes' under grant # MR/T042230/1. J. Friemann acknowledges funding from Horizon Europe through the MSCA Doctoral Network RELIANCE, grant # 101073040. A. Garbout, M. Puig, and H. Lipscomb acknowledges support from the National Research Facility for Lab X-ray CT (NXCT), funded through EPSRC grants # EP/T02593X/1 and # EP/V035932/1 R. Culver acknowledges the 'DRAMA (Digital Reconfigurable Additive Manufacturing facilities for Aerospace)' project funded by UKRI under grant # 113157.

Acronyms

AM additive manufacturing.
API application programming interface.
CAD computer-aided design.
CG computer graphics.
CNN convolutional neural network.
CSG Constructive Solid Geometry.
CT computed tomography.
DIAD Dual Imaging And Diffraction.
DL deep learning.
DR digital radiograph.
DRR digitally reconstructed radiograph.
ERROR pEdiatRic dosimetRy personalized platfORm.
eV electronvolt.
FBP filtered-back projection.
GPU graphics processor unit.
gVXR gVirtualXray.
ITK Insight Toolkit.
labCT laboratory computed tomography.

LSF line spread-function.
MAPE mean absolute percentage error.
MC Monte Carlo.
ML machine learning.
NDT non-destructive testing.
PSF point spread function.
SDD source-to-detector distance.
SSIM structural similarity index.
TV total variation.
VR virtual reality.
VTK Visualization Toolkit.
XCT X-ray computed tomography.
ZNCC zero-mean normalised cross-correlation.
ZrB_2 zirconium diboride.
ZrO_2 zirconium dioxide.

REFERENCES

[1] Duvauchelle, P., Freud, N., Kaftandjian, V., and Babot, D., "A computer code to simulate x-ray imaging techniques," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* **170**(1), 245–258 (2000).

[2] Vidal, F. P., Garnier, M., Freud, N., Létang, J. M., and John, N. W., "Simulation of x-ray attenuation on the GPU," in [*Proceedings of Theory and Practice of Computer Graphics 2009*], 25–32, Eurographics Association, Cardiff, UK (June 2009).

[3] Vidal, F. P. and Villard, P.-F., "Development and validation of real-time simulation of x-ray imaging with respiratory motion," *Computerized Medical Imaging and Graphics* **49**, 1–15 (2016).

[4] Freud, N., Duvauchelle, P., Létang, J. M., and Babot, D., "Fast and robust ray casting algorithms for virtual x-ray imaging," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* **248**(1), 175–180 (2006).

[5] Villard, P.-F., Vidal, F. P., Hunt, C., Bello, F., John, N. W., Johnson, S., and Gould, D. A., "A prototype percutaneous transhepatic cholangiography training simulator with real-time breathing motion," *International Journal of Computer Assisted Radiology and Surgery* **4**(6), 571–578 (2009).

[6] Vidal, F. P., Garnier, M., Freud, N., Létang, J. M., and John, N. W., "Accelerated deterministic simulation of x-ray attenuation using graphics hardware," in [*Eurographics 2010 - Poster*], Poster 5011, Eurographics Association, Norrköping, Sweden (May 2010).

[7] Vidal, F. P., Mitchell, I. T., and Létang, J. M., "Use of fast realistic simulations on GPU to extract CAD models from microtomographic data in the presence of strong CT artefacts," *Precision Engineering* **74**, 110–125 (2022).

[8] Pointon, J. L., Wen, T., Tugwell-Allsup, J., Sújar, A., Létang, J. M., and Vidal, F. P., "Simulation of x-ray projections on gpu: Benchmarking gvirtualxray with clinically realistic phantoms," *Computer Methods and Programs in Biomedicine* **234**, 107500 (2023).

[9] Corbi, A., Burgos, D., Vidal, F., Albiol, F., and Albiol, A., "X-ray imaging virtual online laboratory for engineering undergraduates," *European Journal of Physics* **41**, 014001 (Dec. 2019).

[10] Schoonjans, T., Brunetti, A., Golosio, B., Sanchez del Rio, M., Solé, V. A., Ferrero, C., and Vincze, L., "The xraylib library for x-ray–matter interactions. recent developments," *Spectrochimica Acta Part B: Atomic Spectroscopy* **66**(11), 776–784 (2011).

[11] Poludniowski, G., Omar, A., Bujila, R., and Andreo, P., "Technical note: SpekPy v2.0—a software toolkit for modeling x-ray tube spectra," *Medical Physics* **48**(7), 3630–3637 (2021).

[12] Hernández, G. and Fernández, F., "A model of tungsten anode x-ray spectra," *Medical Physics* **43**(8Part1), 4655–4664 (2016).

[13] Jørgensen, J. S., Ametova, E., Burca, G., Fardell, G., Papoutsellis, E., Pasca, E., Thielemans, K., Turner, M., Warr, R., Lionheart, W. R. B., and Withers, P. J., "Core Imaging Library - Part I: a versatile Python framework for tomographic imaging," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **379**(2204), 20200192 (2021).

[14] Papadimitroulas, P., Erwin, W. D., Iliadou, V., Kostou, T., Loudos, G., and Kagadis, G. C., "A personalized, monte carlo-based method for internal dosimetric evaluation of radiopharmaceuticals in children," *Medical Physics* **45**(8), 3939–3949 (2018).

[15] Rodriguez Perez, S., Marshall, N., Struelens, L., and Bosmans, H., "Characterization and validation of the thorax phantom lungman for dose assessment in chest radiography optimization studies," *Journal of Medical Imaging* **5**, 1 (02 2018).

[16] McCormick, M., Liu, X., Ibanez, L., Jomier, J., and Marion, C., "ITK: enabling reproducible research and open science," *Frontiers in Neuroinformatics* **8**, 13 (2014).

[17] Schroeder, W., Martin, K., and Lorensen, B., [*The Visualization Toolkit – An Object-Oriented Approach To 3D Graphics*], Kitware, Inc., fourth ed. (2006).

[18] Vidal, F. P. and Tugwell-Allsup, J., "CT scans, 3D segmentations, digital radiograph and 3D surfaces of the Lungman phantom." https://doi.org/10.5281/zenodo.10782644 (Mar. 2024). [Accessed 19-07-2024].

[19] Schneider, W., Bortfeld, T., and Schlegel, W., "Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions," *Physics in Medicine & Biology* **45**, 459–478 (Jan. 2000).

[20] Maire, E., Bonnard, G., Adrien, J., Boulnat, X., Létang, J. M., and Lachambre, J., "Dual beam microfocus high-energy tomography: Towards multimodal and faster laboratory experiments," *Tomography of Materials and Structures* **5**, 100030 (June 2024).

[21] Létang, J. M., Lachambre, J., and Maire, E., "Cross-detector scatter issues in dual synchronous tomography: an affine projection correction protocol," *Tomography of Materials and Structures* **6**, 100039 (Sept. 2024).

[22] Reinhard, C., Drakopoulos, M., Ahmed, S. I., Deyhle, H., James, A., Charlesworth, C. M., Burt, M., Sutter, J., Alexander, S., Garland, P., Yates, T., Marshall, R., Kemp, B., Warrick, E., Pueyos, A., Bradnick, B., Nagni, M., Winter, A. D., Filik, J., Basham, M., Wadeson, N., King, O. N. F., Aslani, N., and Dent, A. J., "Beamline K11 DIAD: a new instrument for dual imaging and diffraction at Diamond Light Source," *Journal of Synchrotron Radiation* **28**, 1985–1995 (Nov 2021).

[23] Vestbøstad, M., Karlgren, K., and Olsen, N. R., "Research on simulation in radiography education: a scoping review protocol," *Systematic Reviews* **9**(263) (2020).

[24] Partner, A., Shiner, N., Hyde, E., and Errett, S., "First year student radiographers' perceptions of a one-week simulation-based education package designed to increase clinical placement capacity," *Radiography* **28**, 577–585 (Apr. 2022).

[25] Rit, S., Oliva, M. V., Brousmiche, S., Labarbe, R., Sarrut, D., and Sharp, G. C., "The reconstruction toolkit (rtk), an open-source cone-beam ct reconstruction toolkit based on the insight toolkit (itk)," *Journal of Physics: Conference Series* **489**, 012079 (mar 2014).

[26] Zuo, Z., Qian, W. Y., Liao, X., and Heng, P.-A., "Position based catheterization and angiography simulation," in [*2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*], 1–7 (2018).

[27] Racy, M., Barrow, A., Tomlinson, J., and Bello, F., "Development and validation of a virtual reality haptic femoral nailing simulator," *Journal of Surgical Education* **78**(3), 1013–1023 (2021).

[28] Sujar, A., Kelly, G., García, M., and Vidal, F. P., "Interactive teaching environment for diagnostic radiography with real-time x-ray simulation and patient positioning," *International Journal of Computer Assisted Radiology and Surgery* **17**(1), 85–95 (2022).

[29] Dixon, L., *XRayViewer - A C++ Unreal Engine 5 Wrapper for gVirtualXRay*, Master's thesis, Bangor University (2022).

[30] Albiol, F., Corbi, A., and Albiol, A., "Densitometric radiographic imaging with contour sensors," *IEEE Access* **7**, 18902–18914 (2019).

[31] Corbi, A., *Environment recognition applied to particle detectors*, PhD thesis, University of Valencia, Valencia, Spain (2017).

[32] Mishnaevsky Jr, L., "Sustainable end-of-life management of wind turbine blades: Overview of current and coming solutions," *Materials* **14**(5), 1124 (2021).

[33] Lee, H., Hwang, Y. M., Lee, J., Kim, N.-W., and Ko, S.-K., "A drone-driven x-ray image-based diagnosis of wind turbine blades for reliable operation of wind turbine," *IEEE Access* **12**, 56141–56158 (2024).

[34] Carmignato, S., Dewulf, W., and Leach, R., [*Industrial X-Ray Computed Tomography*], Springer International Publishing, Cham (2018).

[35] Toda, H., [*X-Ray CT - Hardware and Software Techniques*], Springer, Singapore (2021).

[36] Uhlman, N., Salamon, M., and Burtzlaff, S., "Components and Methods for Highest Resolution Computed Tomography," *International Symposium on NDT in Aerospace (Fürth, Germany)* **1**, 3–9 (2008).

[37] Kerckhofs, G., Pyka, G., Moesen, M., Van Bael, S., Schrooten, J., and Wevers, M., "High-Resolution Microfocus X-Ray Computed Tomography for 3D Surface Roughness Measurements of Additive Manufactured Porous Materials," *Advanced Engineering Materials* **15**, 153–158 (mar 2013).

[38] Nikishkov, Y., Seon, G., and Makeev, A., "Structural analysis of composites with porosity defects based on X-ray computed tomography," *Journal of Composite Materials* **48**(17), 2131–2144 (2014).

[39] Coleman, S., Yun, W., Lewis, S. J. Y., Burnett, J. C., and Litzer, E., "Developments in an ultra-high-brightness quasi-monochromatic laboratory x-ray source with tender/soft x-ray production," in [*Advances in X-Ray/EUV Optics and Components XVIII*], Mimura, H., Morawe, C., and Khounsary, A. M., eds., 14, SPIE (oct 2023).

[40] Makarkin, M. and Bratashov, D., "State-of-the-art approaches for image deconvolution problems, including modern deep learning architectures," *Micromachines* **12**(12) (2021).

[41] Zhu, J., Li, K., and Hao, B., "Hybrid variational model based on alternating direction method for image restoration," *Advances in Difference Equations* **2019**(1) (2019).

[42] Arnold, B. A., Bjarngard, B. E., and Klopping, J. C., "A modified pinhole camera method for investigation of X-ray tube focal spots," *Physics in Medicine and Biology* **18**(4), 540–549 (1973).

[43] Wu, P., Boone, J. M., Hernandez, A. M., Mahesh, M., and Siewerdsen, J. H., "Theory, method, and test tools for determination of 3D MTF characteristics in cone-beam CT," *Medical Physics* **48**(6), 2772–2789 (2021).

[44] Cunningham, I. A. and Reid, B. K., "Signal and noise in modulation transfer function determinations using the slit, wire, and edge techniques," *Medical Physics* **19**(4), 1037–1044 (1992).

[45] Létang, J. M. and Peix, G., "On-line X-ray focal spot assessment based on deconvolution using standard imaging devices," *NDT and E International* **36**(5), 303–317 (2003).

[46] Shi, L., Bennett, N. R., and Wang, A. S., "Characterization of x-ray focal spots using a rotating edge," *Journal of Medical Imaging* **8**(2), 023502 (2021).

[47] Boone, J. M., "Determination of the presampled MTF in computed tomography," *Medical Physics* **28**(3), 356–360 (2001).

[48] Di Domenico, G., Cardarelli, P., Contillo, A., Taibi, A., and Gambaccini, M., "X-ray focal spot reconstruction by circular penumbra analysis - Application to digital radiography systems," *Medical Physics* **43**(1), 294–302 (2016).

[49] Bircher, B. A., Meli, F., Küng, A., and Sofiienko, A., "Traceable x-ray focal spot reconstruction by circular edge analysis: from sub-microfocus to mesofocus," *Measurement Science and Technology* **33**, 074005 (jul 2022).

[50] Russo, P. and Mettivier, G., "Method for measuring the focal spot size of an x-ray tube using a coded aperture mask and a digital detector," *Medical Physics* **38**(4), 2099–2115 (2011).

[51] Probst, G. M., Hou, Q., Boeckmans, B., Xiao, Y. S., and Dewulf, W., "Characterization and stability monitoring of X-ray focal spots," *CIRP Annals* **69**(1), 453–456 (2020).

[52] Chen, Z. and Ning, R., "Three-dimensional point spread function measurement of cone-beam computed tomography system by interactive edge-blurring algorithm," *Physics in Medicine and Biology* **49**(10), 1865–1880 (2004).

[53] Thornton, M. M. and Flynn, M. J., "Measurement of the spatial resolution of a clinical volumetric computed tomography scanner using a sphere phantom," in [*Medical Imaging 2006: Physics of Medical Imaging*], Flynn, M. J. and Hsieh, J., eds., **6142**, 61421Z, International Society for Optics and Photonics, SPIE (2006).

[54] Lee, C. and Baek, J., "A new method to measure directional modulation transfer function using sphere phantoms in a cone beam computed tomography system," *IEEE Transactions on Medical Imaging* **34**(4), 902–910 (2015).

[55] Robert, N., Mainprize, J. G., and Whyne, C., "Determination of 3D PSFs from computed tomography reconstructed x-ray images of spherical objects and the effects of sphere radii," *Medical Physics* **46**, 4792–4802 (nov 2019).

[56] Lee, C., dong Song, H., and Baek, J., "3D MTF estimation using sphere phantoms for cone-beam computed tomography systems," *Medical Physics* **47**(7), 2838–2851 (2020).

[57] Friedman, S. N., Fung, G. S., Siewerdsen, J. H., and Tsui, B. M., "A simple approach to measure computed tomography (CT) modulation transfer function (MTF) and noise-power spectrum (NPS) using the American College of Radiology (ACR) accreditation phantom," *Medical Physics* **40**(5), 1–9 (2013).

[58] Richardson, W. H., "Bayesian-based iterative method of image restoration∗," *J. Opt. Soc. Am.* **62**, 55–59 (Jan 1972).

[59] Lucy, L. B., "An iterative technique for the rectification of observed distributions," *The Astronomical Journal* **79**, 745 (jun 1974).

[60] Shaked, E., Dolui, S., and Michailovich, O. V., "Regularized richardson-lucy algorithm for reconstruction of poissonian medical images," in [*2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*], 1754–1757, IEEE (2011).

[61] Engelhardt, M. and Baumann, J., "Determination of Size and Intensity Distribution of the Focal Spot of a Microfocus X-ray Tube Using Image Processing," *Ecndt* **Th.2.5.4**, 1–13 (2006).

[62] Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization," *The Computer Journal* **7**, 308–313 (01 1965).

[63] Wang, Y., Yang, J., Yin, W., and Zhang, Y., "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences* **1**(3), 248–272 (2008).

[64] Leclaire, A. and Moisan, L., "No-Reference Image Quality Assessment and Blind Deblurring with Sharpness Metrics Exploiting Fourier Phase Information," *Journal of Mathematical Imaging and Vision* **52**, 145–172 (may 2015).

[65] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I., "A survey on deep learning in medical image analysis," *Medical Image Analysis* **42**, 60–88 (Dec. 2017).

[66] Garcea, F., Serra, A., Lamberti, F., and Morra, L., "Data augmentation for medical imaging: A systematic literature review," *Computers in Biology and Medicine* **152**, 106391 (Jan. 2023).

[67] Osuala, R., Kushibar, K., Garrucho, L., Linardos, A., Szafranowska, Z., Klein, S., Glocker, B., Diaz, O., and Lekadir, K., "Data synthesis and adversarial networks: A review and meta-analysis in cancer imaging," *Medical Image Analysis* **84**, 102704 (Feb. 2023). arXiv:2107.09543 [cs, eess].

[68] Friemann, J., Mikkelsen, L. P., Oddy, C., and Fagerström, M., "Automated generation of labeled synthetic training data for machine learning based segmentation of 3D-woven composites," in [*Proceedings of the 21st European Conference on Composite Materials: Special sessions*], **8**, 333–338 (2024).

[69] Brown, L. and Long, A., "8 - modeling the geometry of textile reinforcements for composites: Texgen," in [*Composite Reinforcements for Optimum Performance (Second Edition)*], Boisse, P., ed., *Woodhead Publishing Series in Composites Science and Engineering*, 237–265, Woodhead Publishing, second edition ed. (2021).

[70] Bosse, S., "GitHub - bsLab/XraySim: Radiography and CT Xray Simulator and Parallel Filtered Back Projection — github.com." https://github.com/bslab/xraysim. [Accessed 15-07-2024].

[71] Zhang, Y., Shen, Z., and Jiao, R., "Segment anything model for medical image segmentation: Current applications and future directions," *Computers in Biology and Medicine* , 108238 (2024).

[72] "GitHub - facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks. — github.com." https://github.com/facebookresearch/detectron2. [Accessed 15-07-2024].

[73] "OpenSCAD — openscad.org." https://openscad.org/. [Accessed 15-07-2024].

[74] Bosse, S., Lehmhus, D., and Kumar, S., "Automated porosity characterization for aluminum die casting materials using x-ray radiography, synthetic x-ray data augmentation by simulation, and machine learning," *Sensors* **24**(9), 2933 (2024).

[75] Shah, C., Bosse, S., and von Hehl, A., "Taxonomy of damage patterns in composite materials, measuring signals, and methods for automated damage diagnostics," *Materials* **15**(13), 4645 (2022).

[76] Westsik, M., Wells, J. T., Chahid, Y., Morris, K., Milanova, M., Beardsley, M., Harris, M., Ward, L., Alcock, S. G., Nistea, I.-T., Cottarelli, S., Tammas-Williams, S., and Atkins, C., "From design to evaluation of an additively manufactured, lightweight, deployable mirror for Earth observation," in [*Astronomical Optics: Design, Manufacture, and Test of Space and Ground Systems IV*], Hull, T. B., Kim, D., and Hallibert, P., eds., **12677**, 1267704, International Society for Optics and Photonics, SPIE (2023).

[77] Brierley, N., Nye, B., and McGuinness, J., "Mapping the spatial performance variability of an x-ray computed tomography inspection," *NDT & E International* **107**, 102127 (2019).

[78] Reiter, M., Weiß, D., Gusenbauer, C., Erler, M., Kuhn, C., Kasper, S., and Kastner, J., "Evaluation of a histogram-based image quality measure for x-ray computed tomography," in [*5th Conference on Industrial Computed Tomography (iCT)*], (02 2014).

[79] Obaton, A., Gaillard, Y., Bouvet, P., Guiraud, O., Genot, S., and Gay, L., "Evaluation of xct image quality," in [*12th Conference on Industrial Computed Tomography (iCT)*], (02 2023).

[80] Alsaffar, A., Kieß, S., Sun, K., and Simon, S., "Computational scatter correction in near real-time with a fast monte carlo photon transport model for high-resolution flat-panel ct," *Journal of Real-Time Image Processing* **19**, 1063 (2022).

[81] Maier, J., Sawall, S., Knaup, M., and KachelrieSS, M., "Deep scatter estimation (dse): Accurate real-time scatter estimation for x-ray ct using a deep convolutional neural network," *Journal of Nondestructive Evaluation* **37**, 57 (2018).

APPENDIX A. ACQUISITION SIMULATION AND RECONSTRUCTION WITHOUT THE JSON FILE

```python
#!/usr/bin/env python3
import os
import numpy as np

from gvxrPython3 import gvxr # Simulate X-ray images
from gvxrPython3.utils import loadSpectrumSpekpy
from gvxrPython3 import gvxr2json # Simulate X-ray images

# CT reconstruction using CIL
from cil.io import TIFFStackReader, TIFFWriter
from cil.processors import TransmissionAbsorptionConverter
from cil.framework import AcquisitionGeometry, AcquisitionData
from cil.recon import FDK
from cil.optimisation.algorithms import SIRT
from cil.optimisation.functions import IndicatorBox
from cil.plugins.astra.operators import ProjectionOperator

# Create an OpenGL context
print("Create an OpenGL context")
gvxr.createOpenGLContext();


# Set up the detector
```

```python
print("Set up the detector");
gvxr.setDetectorPosition(0.0, 40.0, 0.0, "cm");
gvxr.setDetectorUpVector(0, 0, -1);
gvxr.setDetectorNumberOfPixels(512, 512);
gvxr.setDetectorPixelSize(500, 500, "um");

# Set the impulse response of the detector, a convolution kernel
gvxr.setLSF([0.00110698, 0.00122599, 0.00136522, 0.00152954, 0.00172533, 0.00196116,
    0.0022487, 0.00260419, 0.00305074, 0.00362216, 0.00436939, 0.00537209, 0.00676012,
     0.0087564, 0.01176824, 0.01659933, 0.02499446, 0.04120158, 0.0767488, 0.15911699,
     0.24774516, 0.15911699, 0.0767488, 0.04120158, 0.02499446, 0.01659933,
    0.01176824, 0.0087564, 0.00676012, 0.00537209, 0.00436939, 0.00362216, 0.00305074,
     0.00260419, 0.0022487, 0.00196116, 0.00172533, 0.00152954, 0.00136522,
    0.00122599, 0.00110698]);

# Set the scintillator
gvxr.setScintillator("CsI", 500, "um");

# Create a source
print("Set up the beam");
gvxr.setSourcePosition(0.0,  -150.0, 0.0, "cm");
gvxr.usePointSource();
#  For a parallel source, use gvxr.useParallelBeam();

# Set its spectrum, here a monochromatic beam
# 1000 photons of 80 keV (i.e. 0.08 MeV) per ray
# gvxr.setMonoChromatic(80, "keV", 1000);

# Or use a polychromatic beam
# The tube voltage is 160 keV
# The filtration is 1mm of tin (Sn)
# The anode angle is 12 degrees
# mAs is 0,5
# The source to detector distance in 50 cm
loadSpectrumSpekpy(160, filters=[["Sn", 1.0]], th_in_deg=12, mAs=0.5, z=150 - -40);

# Poisson noise will be enable
gvxr.enablePoissonNoise(); # Not needed as mAs was used in the function call above

# Locate the sample STL file from the package directory
path = os.path.dirname(gvxr.__file__);
fname = os.path.join(path, "welsh-dragon-small.stl");

gvxr.loadMeshFile("Dragon", fname, "mm");
gvxr.moveToCentre("Dragon");

# Material properties

# Iron (Z number: 26, symbol: Fe)
# gvxr.setElement("Dragon", 26);
# gvxr.setElement("Dragon", "Fe");

# Liquid water
# gvxr.setCompound("Dragon", "H2O");
# gvxr.setDensity("Dragon", 1.0, "g/cm3");
# gvxr.setDensity("Dragon", 1.0, "g.cm-3");
```

```python
# Titanium Aluminum Vanadium Alloy
# gvxr.setMixture("Dragon", "Ti90Al6V4");
gvxr.setMixture("Dragon", [22, 13, 23], [0.9, 0.06, 0.04]);
# gvxr.setMixture("Dragon", ["Ti", "Al", "V"], [0.9, 0.06, 0.04]); # Not yet
    implemented
# gvxr.setDensity("Dragon", 4.43, "g/cm3");
gvxr.setDensity("Dragon", 4.43, "g.cm-3");

# Compute an X-ray image
x_ray_image = np.array(gvxr.computeXRayImage()).astype(np.single) / gvxr.getWhiteImage
    ();

# Interactive visualisation
# The user can rotate the 3D scene and zoom-in and -out in the visualisation window.
# It can be useful to rotate the visualisation of the 3D environment and zoom in/out
# to take the best posible screenshots

# - Keys are:
#     - Q/Escape: to quit the event loop (does not close the window)
#     - B: display/hide the X-ray beam
#     - W: display the polygon meshes in solid or wireframe
#     - N: display the X-ray image in negative or positive
#     - H: display/hide the X-ray detector
# - Mouse interactions:
#     - Zoom in/out: mouse wheel
#     - Rotation: Right mouse button down + move cursor```
# gvxr.renderLoop()

# Take and display a screenshot
gvxr.setZoom(2500);
gvxr.displayScene();
screenshot = gvxr.takeScreenshot();

# Simulate the CT acquisition and save the projections
gvxr.computeCTAcquisition("../results/dragon-projs", # Where to save the projections
    "screenshots", # Where to save the screenshots
    200, # Total number of projections
    0, # First angle
    True, # Include the last angle
    360, # Last angle
    60, # Number of flat images
    0, 0, 0, "mm", # Centre of rotation
    *gvxr.getDetectorUpVector()); # Rotation axis

# Save a JSON file
gvxr2json.saveJSON("../results/dragon.json");

# Set the CT reconstruction parameters
# Create the TIFF reader by passing the directory containing the files
reader = TIFFStackReader(file_name="../results/dragon-projs", dtype=np.float32);

# Read in file, and return a numpy array containing the data
data_original = reader.read();

# Normalisation
# Not strictly needed as the data was already corrected
data_normalised = data_original / data_original.max();
```

```python
# Prevent log of a negative value
data_normalised[data_normalised<1e-9] = 1e-9;

# Linearisation
data_absorption = -np.log(data_normalised);

# The data is stored as a stack of detector images, we use the CILlabels for the axes
axis_labels = ['angle','vertical','horizontal'];

# Create the CIL geoemtry
geometry = AcquisitionGeometry.create_Cone3D(source_position=gvxr.getSourcePosition("
    cm"),
    detector_position=gvxr.getDetectorPosition("cm"),
    detector_direction_x=gvxr.getDetectorRightVector(),
    detector_direction_y=gvxr.getDetectorUpVector(),
    rotation_axis_position=gvxr.getCentreOfRotationPositionCT("cm"),
    rotation_axis_direction=gvxr.getRotationAxisCT());

# Set the angles, remembering to specify the units
geometry.set_angles(np.array(gvxr.getAngleSetCT()), angle_unit='degree');

# Set the detector shape and size
geometry.set_panel(gvxr.getDetectorNumberOfPixels(), gvxr.getDetectorPixelSpacing("cm"
    ));

# Set the order of the data
geometry.set_labels(axis_labels);

# Set the angles, remembering to specify the units
geometry.set_angles(np.array(gvxr.getAngleSetCT()), angle_unit='degree');

# Set the detector shape and size
geometry.set_panel(gvxr.getDetectorNumberOfPixels(), gvxr.getDetectorPixelSpacing("cm"
    ));

# Shutdown the simulation engine
gvxr.terminate();

# Prepare the data for the reconstruction
acquisition_data = AcquisitionData(data_absorption, deep_copy=False, geometry=geometry
    );
acquisition_data.reorder(order='tigre');

# Perform the FDK reconstruction
ig = acquisition_data.geometry.get_ImageGeometry();
fdk =  FDK(acquisition_data, ig);
recon = fdk.run();

# Save the CT volume as a TIFF stack
TIFFWriter(data=recon, file_name=os.path.join("../results/dragon-recons-FDK", "out")).
    write();

# Perform the CT reconstruction using the SIRT algorithm and save the reconstructed
    volume
# Create projection operator using Astra-Toolbox.
acquisition_data.reorder(order='astra');
```

```
A = ProjectionOperator(ig, geometry, "gpu");

# Create the initial guess
x0 = ig.allocate();

# non-zero constraint
constraint = IndicatorBox(lower=0);

# Instantiate the reconstruction algorithm
sirt = SIRT(initial=x0, operator=A, data=acquisition_data, constraint=constraint,
    max_iteration=500);

# Perform 500 iterations
sirt.update_objective_interval = 50;
sirt.run(500);

recon_sirt_noisy = sirt.solution;

# Save the CT volume as a TIFF stack
TIFFWriter(data=recon_sirt_noisy, file_name=os.path.join("../results/dragon-recons-
    SIRT", "out")).write();
```

APPENDIX B. CORRESPONDING JSON FILE


```
{
    "File format version": [1, 0, 0],
    "Window size": [500, 500],
    "Source": {
        "Position": [0.0, -1500.0, 0.0, "mm"],
        "Shape": "POINT",
        "Beam": {
            "Peak kilo voltage": 160.0,
            "Tube angle": 12.0,
            "mAs": 0.5,
            "filter": [
                ["Sn", 1.0, "mm"]
            ]
        }
    },
    "Detector": {
        "Position": [0.0, 400.0, 0.0, "mm"],
        "UpVector": [0.0, 0.0, -1.0],
        "RightVector": [-1.0, 0.0, 0.0],
        "NumberOfPixels": [512, 512],
        "Size": [256.0, 256.0, "mm"],
        "LSF": [0.0011069800239056349, 0.0012259900104254484,
   0.0013652200577780604, 0.0015295400517061353, 0.0017253300175070763,
   0.001961159985512495, 0.0022487000096589327, 0.0026041900273412466,
   0.003050740109756589, 0.0036221600603312254, 0.004369390197098255,
   0.005372089799493551, 0.00676011992618416, 0.00875640008598566,
   0.011768239550292492, 0.016599329188466072, 0.024994460865855217,
   0.04120158031582832, 0.0767488032579422, 0.1591169834136963,
   0.24774515628814697, 0.1591169834136963, 0.0767488032579422,
   0.04120158031582832, 0.024994460865855217, 0.016599329188466072,
   0.011768239550292492, 0.00875640008598566, 0.00676011992618416,
```

```
    0.005372089799493551, 0.004369390197098255, 0.0036221600603312254,
    0.003050740109756589, 0.0026041900273412466, 0.0022487000096589327,
    0.001961159985512495, 0.0017253300175070763, 0.0015295400517061353,
    0.0013652200577780604, 0.0012259900104254484, 0.0011069800239056349],
        "Scintillator": {
            "Material": "CsI",
            "Thickness": 500.0,
            "Unit": "um"
        }
    },
    "Scan": {
        "OutFolder": "../results/dragon-projs",
        "GifPath": "screenshots",
        "NumberOfProjections": 200,
        "AngleStep": 1.7999999523162842,
        "StartAngle": 0.0,
        "FinalAngle": 360.0,
        "IncludeLastAngle": true,
        "Flat-Field Correction": true,
        "NumberOfWhiteImages": 60,
        "CentreOfRotation": [0.0, 0.0, 0.0, "mm"],
        "RotationAxis": [0.0, 0.0, -1.0]
    },
    "Samples": [
        {
            "Label": "Dragon",
            "Path": "path_on_disk/welsh-dragon-small.stl",
            "Unit": "mm",
            "Material": ["Mixture", "Al6Ti90V4"],
            "Density": 4.43,
            "Transform": [
                [
                    "Matrix",
                    [
                        1.0,  0.0, 0.0, 0.0,
                        0.0, 1.0, 0.0, 0.0,
                        0.0, 0.0, 1.0, 0.0,
                        10.479137420654297, 671.8119506835938,
    -298.12640380859375, 1.0
                    ]
                ]
            ],
            "Type": "inner",
            "AmbientColour": [1.0, 1.0, 1.0, 1.0],
            "DiffuseColour": [1.0, 1.0, 1.0, 1.0],
            "SpecularColour": [1.0, 1.0, 1.0, 1.0],
            "Shininess": 50.0
        }
    ]
}
```

APPENDIX C. ACQUISITION SIMULATION AND RECONSTRUCTION WITH THE JSON FILE

```python
#!/usr/bin/env python3
import os
import numpy as np

from gvxrPython3 import gvxr # Simulate X-ray images
from gvxrPython3 import json2gvxr # Simulate X-ray images

# CT reconstruction using CIL
from gvxrPython3.JSON2gVXRDataReader import *

from cil.io import TIFFWriter
from cil.processors import TransmissionAbsorptionConverter
from cil.framework import AcquisitionGeometry, AcquisitionData
from cil.recon import FDK
from cil.optimisation.algorithms import SIRT
from cil.optimisation.functions import IndicatorBox
from cil.plugins.astra.operators import ProjectionOperator

# Initialise gVXR using our JSON file
json_fname = "../results/dragon.json"

# MS Windows
if os.name == "nt":
    json2gvxr.initGVXR(json_fname, renderer="EGL")
# MacOS
elif str(os.uname()).find("Darwin") >= 0:
    json2gvxr.initGVXR(json_fname, renderer="OPENGL")
# GNU/Linux
else:
    json2gvxr.initGVXR(json_fname, renderer="EGL")

# Set up the detector
json2gvxr.initDetector(json_fname, verbose=0)

# Create a source
json2gvxr.initSourceGeometry(verbose=0)
json2gvxr.initSpectrum(verbose=0);

# Load the sample
json2gvxr.initSamples(verbose=0)

# Compute an X-ray image
x_ray_image = np.array(gvxr.computeXRayImage()).astype(np.single) / gvxr.getWhiteImage
    ()

# Interactive visualisation
# The user can rotate the 3D scene and zoom-in and -out in the visualisation window.
# It can be useful to rotate the visualisation of the 3D environment and zoom in/out
# to take the best posible screenshots

# - Keys are:
#     - Q/Escape: to quit the event loop (does not close the window)
#     - B: display/hide the X-ray beam
#     - W: display the polygon meshes in solid or wireframe
```

```
#     - N: display the X-ray image in negative or positive
#     - H: display/hide the X-ray detector
# - Mouse interactions:
#     - Zoom in/out: mouse wheel
#     - Rotation: Right mouse button down + move cursor```
# gvxr.renderLoop()

# Take and display a screenshot
gvxr.setZoom(2500);
gvxr.displayScene();
screenshot = gvxr.takeScreenshot();

# Simulate a CT scan acquisition
json2gvxr.initScan();
angles = json2gvxr.doCTScan()

# Create the JSON2gVXR reader by passing the filename
data_original = JSON2gVXRDataReader(file_name=json_fname).read()

# Normalisation and linearisation
data_absorption = TransmissionAbsorptionConverter(white_level=data_original.max(),
    min_intensity=1e-9)(data_original)

# Prepare the data for the reconstruction
acquisition_data = AcquisitionData(data_absorption, deep_copy=False, geometry=
    data_absorption.geometry);
acquisition_data.reorder(order='tigre');
ig = acquisition_data.geometry.get_ImageGeometry();

# Perform the FDK reconstruction
fdk =  FDK(acquisition_data, ig);
recon = fdk.run();

# Save the CT volume as a TIFF stack
TIFFWriter(data=recon, file_name=os.path.join("../results/dragon-recons-FDK", "out")).
    write();

# Perform the CT reconstruction using the SIRT algorithm and save the reconstructed
    volume
# Create projection operator using Astra-Toolbox.
acquisition_data.reorder(order='astra');
A = ProjectionOperator(ig, data_absorption.geometry, "gpu");

# Create the initial guess
x0 = ig.allocate();

# non-zero constraint
constraint = IndicatorBox(lower=0);

# Instantiate the reconstruction algorithm
sirt = SIRT(initial=x0, operator=A, data=acquisition_data, constraint=constraint,
    max_iteration=500);

# Perform 500 iterations
sirt.update_objective_interval = 50;
sirt.run(500);
```

```
recon_sirt_noisy = sirt.solution;

# Save the CT volume as a TIFF stack
TIFFWriter(data=recon_sirt_noisy, file_name=os.path.join("../results/dragon-recons-
    SIRT", "out")).write();
```

APPENDIX D. CSG-CAD MODELS

D.1 CSG Plate with Pores Model

```
rotate ([90,90,90])
difference () {
  rotate ([90,0,0]) cube([height,thickness,width],true);
  union () {
    translate([px,py,pz])
      rotate ([0,0,pangle])
        scale([pw,ph,pd])
         sphere(r=0.5,$fn=20);
    ...
  }
}
```

D.2 CSG FML Plate with Impact Damage Model

```
//Gauss-Extrusion

pi = 3.14159265358979323846;
e  = 2.71828182845904523536;

//rendering parameter
rp = 100;

//sample parameters
length = 50;
width = 50;
thickness = 0.5;
mu=0;

//damage parameters
dmg_depth = 0.8;
dmg_reach = 2;

step = (1/rp);
function gauss(x, mu, sig) = ((1/sqrt(2*pi*(sig^2))*e)^
         (-((x-mu)^2)/(2*(sig^2))));

material=0;
//impacted layer 1/5
module layer(z,thickness,dmg_depth,dmg_reach,material) {
  sigma = 0.35+(sqrt(dmg_reach)^0.45-1);
  color(material==1?&quot;yellow&quot;:&quot;red&quot;)
  translate([0,0,z]) union(){
    if (1) difference(){
      cube([length, width, thickness], center = true);
      color(&quot;red&quot;)
      cylinder(h = 1.1*thickness, r =
        dmg_reach, center = true, $fn = rp);
    }
```

```
    fixheight=gauss(dmg_reach, mu, sigma);

    //color(&quot;white&quot;)
    translate([0, 0, -0.5*thickness])
    rotate_extrude($fn = rp){
      for (i = [0:step:dmg_reach-step]){
        j = i + step;
        hull(){
          translate([i, gauss(i, mu, sigma)*dmg_depth-fixheight, 0])
            square(size = thickness, center = false);
          translate([j, gauss(j, mu, sigma)*dmg_depth-fixheight, 0])
            square(size = thickness, center = false);
        }
      }
    };
  };
}
rotate([90,0,90]) union () {
if (material==1 || material==0)
  layer(0,thickness=0.5,dmg_depth=0.5,dmg_reach=8,material=1);
if (material==2 || material==0)
  layer(0.5,thickness=0.5,dmg_depth=0.5,dmg_reach=8,material=2);
if (material==1 || material==0)
  layer(1,thickness=0.5,dmg_depth=0.5,dmg_reach=8,material=1);
if (material==2 || material==0)
  layer(1.5,thickness=0.5,dmg_depth=0.5,dmg_reach=8,material=2);
if (material==1 || material==0)
  layer(2,thickness=0.5,dmg_depth=0.5,dmg_reach=8,material=1);
};
```

D.3 XraySim and FBP settings

```
xraysim1 -soft \
  -u 2 impactGauss1-1.stl \
  -u 1 impactGauss1-2.stl
  -p 0.07 -e 0.07 -w 1024 -h 1024 -ct 0.45 \
  -S 0 80 65535 \
  -o xray-ct-%04d.tif

fbp64 -f -a -bv 65535 -m 10 -t 24 \
      -o xray-ct-slices.tif xray-ct-proj.tif

===== xraysim ======

[-s(ourcePos) x y z] [-d(etPos) x y z] [-D(detUpVec) x y z]
[-e(nergy) MeV] [-S(caleIntensity) min max range] [-n addgaussnoiseperc]
[-r(otate-z) degree] [-R(otate) x y z] [-ct(-rotate-z) delta degree]
[-w pixels]  [-h pixels] [-p pixelsize] [-g openglwinsize]
[-show(_scene)] [-soft(ware_cpu_renderer)]
[-u density g/cm3] file.stl [-u density file2.stl]
[-o out[%04d].tif/pgm]

======= fbp ========

[-e(xtra_frames) #] [-f(ull_turn)] [-g(amma) #] [-G(ain) #]
[-a(uto_size_and_range)] [-n(ormalizeByframe)] [-ba(ckground_auto)]
```

```
[-bv(ackground:value) #] [-bb(ackground) x y w h]
[-c(rop) x y w h] [-r firstframe last]
[-p(reprocess)] [-t(hreads) #]
[-m(ask_margin_pixel) #]
[-R(otate_additional) x y z] [-C(rop_before_rotate) x y w h]
[-o(utput) dir|file.tif(f)]
scan.[avi mp4 tif tiff]
```