



gVirtualXray (gVXR): Simulating X-ray radiographs and CT volumes of anthropomorphic phantoms

Pointon, Jamie; Wen, Tianci; Tugwell-Allsup, Jenna; Létang, Jean Michel; Vidal, Franck

Software Impacts

DOI:

[10.1016/j.simpa.2023.100513](https://doi.org/10.1016/j.simpa.2023.100513)

Published: 22/05/2023

Peer reviewed version

[Cyswllt i'r cyhoeddiad / Link to publication](#)

Dyfyniad o'r fersiwn a gyhoeddwyd / Citation for published version (APA):

Pointon, J., Wen, T., Tugwell-Allsup, J., Létang, J. M., & Vidal, F. (2023). gVirtualXray (gVXR): Simulating X-ray radiographs and CT volumes of anthropomorphic phantoms. *Software Impacts*, 16, Article 100513. <https://doi.org/10.1016/j.simpa.2023.100513>

Hawliau Cyffredinol / General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

gVirtualXray (gVXR): Simulating X-ray radiographs and CT volumes of anthropomorphic phantoms

Jamie Lea Pointon¹, Tianci Wen¹, Jenna Tugwell-Allsup², Jean Michel Létang³, and Franck Patrick Vidal^{1,}*

¹ *School of Computer Science & Electronic Engineering, Bangor University, UK*

² *Radiology Department, Betsi Cadwaladr University Health Board (BCUHB), North Wales, Ysbyty Gwynedd, UK*

³ *Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1294, F-69373, LYON, France*

** Corresponding author, f.vidal@bangor.ac.uk*

Abstract

gVirtualXray (gVXR) is an open-source framework that relies on the Beer-Lambert law to simulate X-ray images in realtime on a graphics processor unit (GPU) using triangular meshes. We produced four Jupyter Notebooks to compare images simulated with gVXR and ground truth image of an anthropomorphic phantom: (i) an X-ray projection generated using a Monte Carlo simulation code, (ii) real digitally reconstructed radiographs (DRRs), (iii) computed tomography (CT) slices, and (iv) a real radiograph acquired with a clinical X-ray imaging system. Image registration was deployed in two Notebooks to align the simulated image on the corresponding ground truth image. We demonstrated that accurate images can be generated in milliseconds with gVirtualXray when scattering can be ignored.

Keywords

X-rays; Computed tomography (CT); Simulation; Graphics processor unit (GPU) programming; Image registration; Digitally reconstructed radiograph (DRR)

Code metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	1.0
C2	Permanent link to code/repository used for this code version	https://github.com/effepivi/gvvr-CMPB/
C3	Permanent link to Reproducible Capsule	
C4	Legal Code License	BSD 3-Clause "New" or "Revised" License.
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python for the front-end, C++ and GLSL for the back-end, OpenGL
C7	Compilation requirements, operating environments & dependencies	python \geq 3.10, numpy, pandas, VTK, scikit-learn, scikit-image, SimpleITK, cil=22.2.0, tigre=2.2, spekpy, pymoo=0.5.0, gVXR, opencv-python, K3D-jupyter
C8	If available Link to developer documentation/manual	https://gvirtualxray.fpvidal.net/documentation
C9	Support email for questions	gvirtualxray-discuss@lists.sourceforge.net

1. Introduction

We present a series of Jupyter notebook to demonstrate how to generate X-ray radiographs of anthropological phantoms using gVirtualXRay (gVXR), the Virtual X-Ray Imaging Library on graphics processor unit (GPU). X-ray simulation has been used in many applications, ranging from medical physics in radiology and oncology, to non-destructive testing using ionising radiation, medical virtual reality (VR) and machine learning (ML). Monte Carlo (MC) simulation is known as the state-of-the-art technique to simulate X-ray images in terms of accuracy. However, prohibitive computing time may be required to simulate an X-ray projection at high spatial resolution with an acceptable level of noise, which limits the use of MC simulations in applications where real-time performance is a requirement [1]. Badal and Badano [2] adapted PENELOPE’s MC code to GPU computing in 2009. They reported runtimes in minutes incompatible with real-time performance. Whilst GPU-based MC simulations are extremely fast compared to their central processor unit (CPU) counterparts (about $1,000\times$ faster), there are still far from real-time performance. About 13 seconds are necessary to compute the voxel dose distribution in a $122 \times 62 \times 372$ volume, which is relatively small compared to a typical computed tomography (CT) volume [3].

gVXR implements a deterministic simulation model based on the Beer-Lambert law to generate noise-free images. Depending on the application, they can provide a good compromise between speed and accuracy [4] and can be implemented on GPUs for a further increase of speed [5, 6]. Unlike Monte Carlo methods, deterministic simulations tend to ignore scattering and noise, although the latter can easily be added as a post-process.

2. Description

2.1 gVirtualXRay in a nutshell

gVXR is an open-source application programming interface (API) written in C++ to compute the Beer-Lambert law (also known as attenuation law) from polygon meshes, the main type of models to represent 3D objects or characters in video games, VR and computer-aided design (CAD) software. 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 modelled with the Beer-Lambert attenuation law:

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

$\mathbf{I}(x, y)$ is the integrated energy in electron volt (eV), a unit of energy commonly used in atomic and nuclear physics, received by pixel (x, y) . The beam spectrum emitted by the X-ray source is discretised in several energy channels in the polychromatic case. E_i corresponds to the energy in eV 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.

It is straightforward to implement the Beer-Lambert attenuation law using ray-tracing. However, Freud et al. [7] proposed a method based on a much faster computer graphics (CG) technique, rasterisation, to efficiently compute \mathbf{d}_j in Eq. 1 from polygon meshes. We first ported their method for monochromatic beams in the early days of GPU programming [5] to simulate fluoroscopy in a medical VR simulator [8]. Polychromatism and the focal spot of the detector were then introduced to improve realism [9]. In 2013, gVXR made its debut as an open-source project on SourceForge¹ and simplistic Monte Carlo simulations were used for unit testing of the Beer-Lambert law implementations [6]. The simulation can be enhanced with the impulse response of the detector, Poisson noise, and phase contrast. When used in the objective function of an optimisation algorithm, it is possible to closely reproduce real experiments performed with synchrotron radiation (see Figure 10) [10]. More recently, anthropomorphic phantoms were used to benchmark gVXR against MC simulations and data acquired with clinically utilised X-ray devices, including a radiograph and a CT scan [11]. In a nutshell, gVXR can be summarised as follows:

- Cross-platform:
 - Windows, GNU/Linux, and MacOS;
 - Laptop, desktop PC, supercomputer, cloud infrastructure.
- Wide range of programming languages (C/C++, Python3, R, Ruby, Tcl, C#, Java, and GNU Octave).
- Its Python package “gVXR” is listed on the Python Package Index ².
- Various source shapes:
 - Cone beam (point source);
 - Cone beam (Focal spot);
 - Parallel beam.
- Incident beams:
 - Monochromatic;
 - Polychromatic, including X-ray tubes with kV and filtration.
- CT scan acquisition (parallel and cone beams);
- Poisson noise;
- Impulse response of the detector;
- Energy response of the detector (e.g. due to the use of a scintillator);
- Geometries:
 - Surface meshes (triangles) in most popular file formats (eg. STL, PLY, 3DS, OBJ, DXF, X3D, DAE);

¹<https://sourceforge.net/projects/gvirtualxray/> (accessed: 2 May 2023).

²<https://pypi.org/project/gVXR/> (accessed: 2 May 2023).

- Volume meshes (tetrahedrons) in the Abacus format (experimental).
- Multi-material objects;
- Material properties:
 - Chemical elements (e.g. the symbol W or the atomic number 74 for tungsten);
 - Compounds, e.g. H₂O for water;
 - Mixture, e.g. Titanium-aluminum-vanadium alloy, Ti90Al6V4;
 - Hounsfield units (for medical applications).
- xraylib³ for the Photon Cross Sections to compute μ in Eq. 1 [12];
- Containerisation using Docker.

2.2 Comparison with MC simulation

In the first notebook, we demonstrate that gVirtualXRay is able to generate analytic simulations on GPU comparable to images generated with a state-of-the-art Monte Carlo simulation package (see Figure 1. An anthropomorphic phantom is used. It corresponds to a 5-year old boy. We take into account i) a realistic beam spectrum (tube voltage and filtration) and ii) the energy response of the detector.

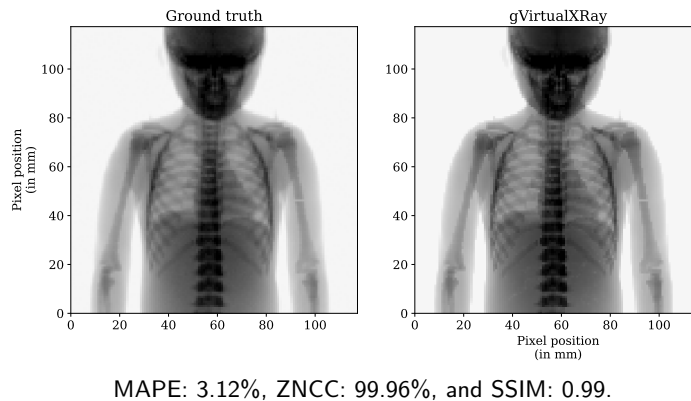


Figure 1: Comparison between X-ray projections simulated with GATE and gVirtualXRay. For a fair comparison, each image is displayed using the same look-up table.

2.3 Comparison with DRR images

In the second notebook, we demonstrate that gVirtualXRay is able to generate analytic simulations on GPU comparable to digitally reconstructed radiographs (DRRs) of a real CTscan computed with Plastimatch (see Figure 2). First, a CT scan of the Lungman phantom was acquired at one of our local hospitals (Ysbyty Gwynedd) using a 128-slice Somatom Definition Edge scanner, Siemens Healthcare, Erlangen, Germany. The CT volume was then segmented into individual structures. Surface meshes were eventually extracted from the segmented CT scan. Simulated corresponding pairs of X-ray images were generated i) with gVirtualXRay from the surface meshes, and ii) with Plastimatch from the original CT volume (see Figure 3) [13].

2.4 Comparison with original CT slices

In the third notebook, we show how gVirtualXRay can be used to generate CT data from polygon meshes. For this experiment, we recreate a CT volume from X-ray projections of the Lungman chest phantom simulated with gVirtualXRay (see Figure 4). The CT volume is reconstructed with the Core Imaging Library (CIL) [14] using the simulated X-ray projections (see Figure 5).

³<https://github.com/tschoonj/xraylib> (accessed: 2 May 2023).

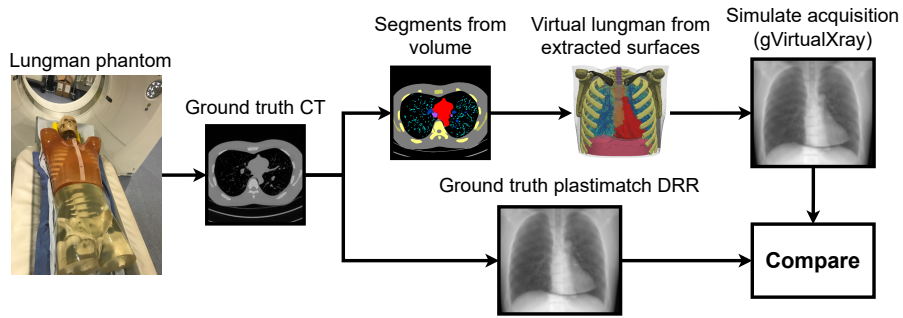
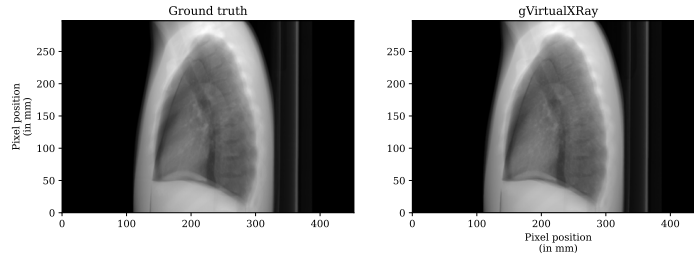


Figure 2: Flow chart demonstrating the validation of projections of the Lungman simulated with gVirtualXRay process.



RL chest view: MAPE: 1.76%, ZNCC: 99.66%, SSIM: 0.98.

Figure 3: Comparison between X-ray projections simulated with gVirtualXRay and DRR images generated with Plastimatch. For a fair comparison, each image of a row is displayed using the same look-up table.

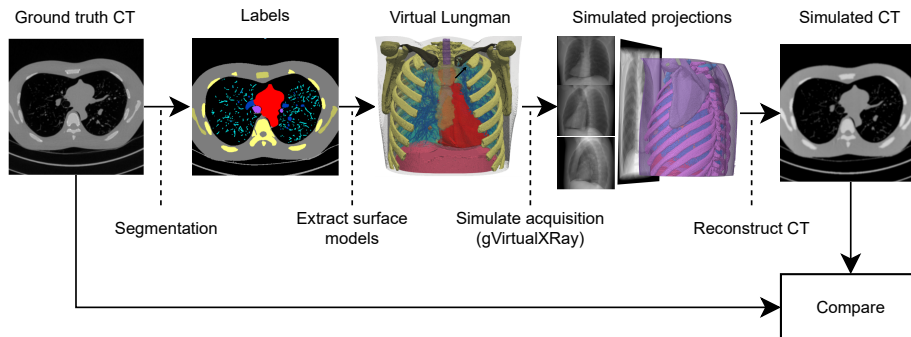
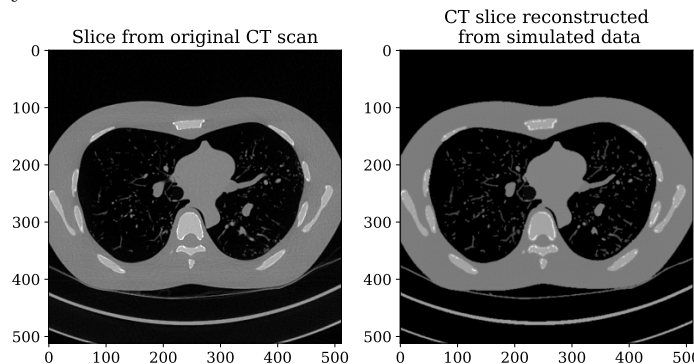


Figure 4: Flow chart demonstrating the validation of CT slices reconstructed from projections of the Lungman simulated with gVirtualXRay.



First slice: MAPE: 5.01%, ZNCC: 98.44%, SSIM: 0.78.

Figure 5: Comparison between CT slices reconstructed from simulated projections with slices from the original CT scan. For a fair comparison, all the images are displayed using the same look-up table.

2.5 Comparison with a digital radiograph

In the last notebook, we reproduce a real digital radiograph taken with a clinically utilised X-ray equipment. Image registration is used to make sure the same geometrical set up is used in both the real data acquisition and the virtual one (see Figure 6). Figure 7 shows the two images side-by-side.

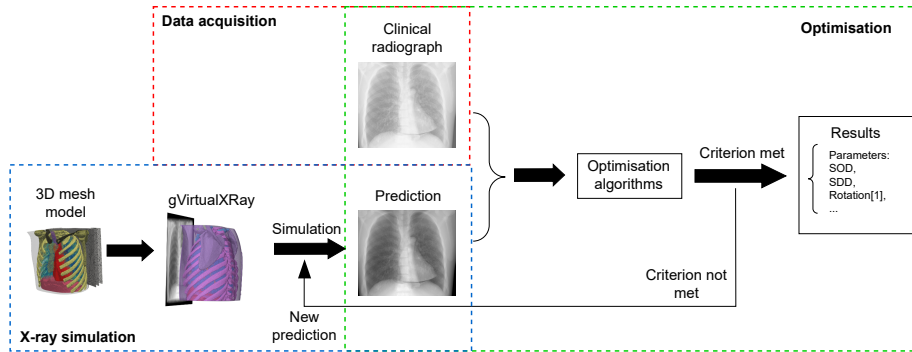
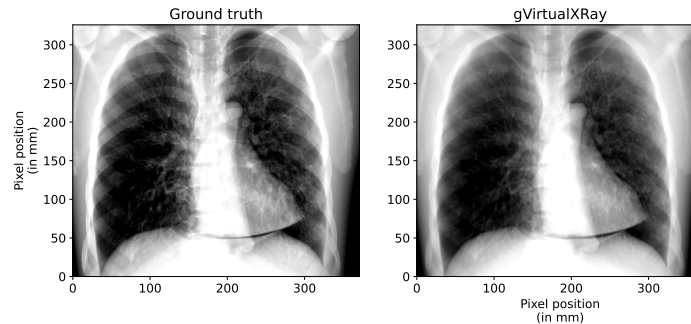


Figure 6: Illustration of the registration framework.



Posterior-anterior (PA) chest view: MAPE: 1.56%, ZNCC: 98.91%, and SSIM: 0.94.

Figure 7: Comparison between a registered X-ray projection simulated with gVirtualXray and a digital radiograph taken using a clinically utilised X-ray equipment. For a fair comparison, each image is displayed using the same look-up table.

3. Impact overview

The assumption is that X-ray images simulated in milliseconds with a deterministic algorithm on GPU are realistic for clinical applications where speed is a requirement, whether this is VR simulators where X-ray images must be simulated in realtime from dynamic patient models, or high-throughput data processing where a massive amount of data must be generated very quickly.

- Superior computational performance for VR and high-throughput data applications:

Accurate images that would take days, hours or minutes with Monte Carlo can be generated in milliseconds with gVirtualXray for high-throughput data (e.g. real-time VR, image registration, and training/testing data generation for ML) when scattering can be ignored.

- Quick creation of CT acquisition data:

Simulated projections can be fed to a reconstruction algorithm to produce CT volumes.

- Dynamic models with animation and physiology:

The use of surface models enables the combination of X-ray simulation with real-time soft-tissue deformation [8, 6] and character animation [15], which is hardly possible with DRRs computed directly from CT volumes.

- Interoperability with CT reconstruction software:

During the “CIL Training and Bring Your Own Data User Hackathon” held in Cambridge, UK, in March 2023, a data reader was implemented and tested to integrate gVXR and CIL. It makes it straightforward to simulate CT scans with gVXR and reconstruct the corresponding tomography data with CIL. Improving the interoperability between the two is important. People can use realistic images simulated in known conditions and test their reconstruction algorithms, for example to compensate motion artefacts that degrade the image quality in medicine.

4. Ongoing research projects

We are developing digital twins of actual devices. It will make it possible i) to predict images acquired with real devices, and ii) to optimise scanning parameters before conducting timely and costly experiments. We are developing this proof-of-concept for computed tomography in nondestructive testing and for synchrotron computed tomography in particular.

5. Limitations and future work

Simulations, deterministic or MC, are approximations of real physics phenomena. Users of such simulations must always consider the context in which they want to deploy the simulation and understand the trade-offs that must be made. For speed in real-time VR for training purposes, gVirtualXray may be used; Monte Carlo would be required for accuracy for treatment planning in radiotherapy.

We are continually improving the simulation. For example, we are modelling exposure time and photon scattering from real images to predict images acquired with real devices.

6. Publications enabled by gVirtualXray

It is possible to use gVirtualXray in real-time applications such as medical training simulators (see Figures 8 and 9) [16, 17, 18]. GPUs also make it possible to embed gVirtualXray within an optimisation algorithm to

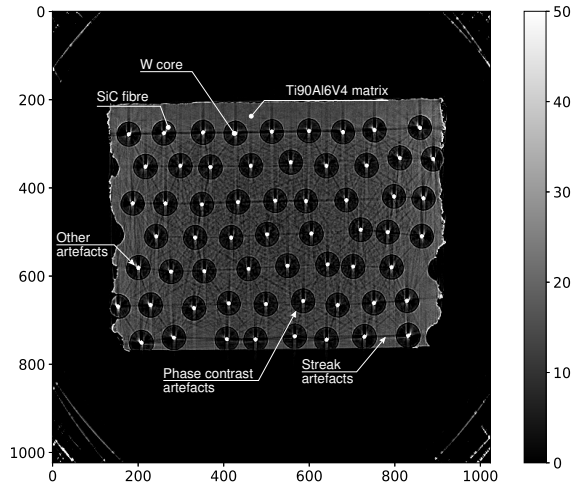


Figure 8: Orthopaedic surgical simulator providing user-controlled image intensifier functionality. Reproduced with permission from Racy et al. [17].

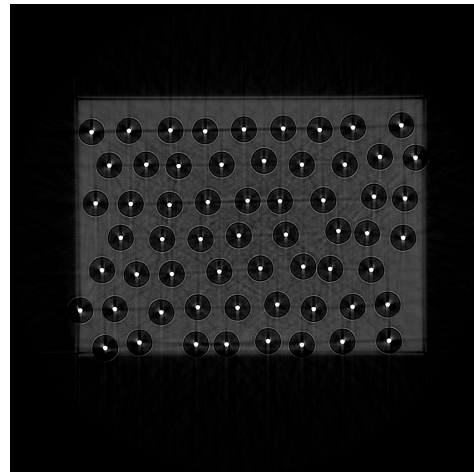


Figure 9: Radiography teaching tool with real-time character animation and soft-tissue deformations to interactively change the patient’s pose in simulated X-ray radiographs.

register 3D hand models on 2D radiographs [19] or produce accurate digital twins in material science [10]. Figure 10a corresponds to a CT reconstruction of a fibre composite where strong CT artefacts are present. Dark streak artefacts are due to beam hardening and the response of the detector, black and white fringes are due to phase contrast. Figure 10b shows a similar CT slice reconstructed from simulated projections after registration. Source of artefacts are also modelled within the simulation. gVirtualXray can also be used to produce a high number of realistic simulated images to train machine learning algorithms [20, 21], which would

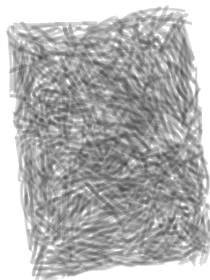


(a) CT slice reconstructed from experimental data acquired at ESRF.

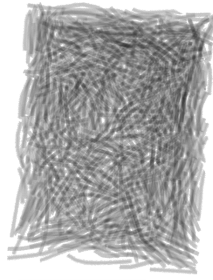


(b) CT slice reconstructed from data simulated using gVirtualXray.

Figure 10: CT reconstructions of a fibre composite where strong CT artefacts are present. Dark streak artefacts are due to beam hardening and the response of the detector, black and white fringes are due to phase contrast.



(a) Real image.



(b) Simulated image.



(c) 3D surface model of the batonnet cut potatoes in Figure 11b.

Figure 11: X-ray images of fried batonnet cut potatoes. Left: Real image; Right: Image simulated with gVirtualXray. Reproduced with permission from [20].

not have been possible a few years ago. Albiol et al. [22] proposed a new type of densitometric radiographic images (a technique that combines two radiographs that were produced with two different tube voltages). They replaced one of the radiographs with a simulated one, thanks to the use of a contour sensor (e.g. Microsoft Kinect) and gVirtualXray (see Figure 12).

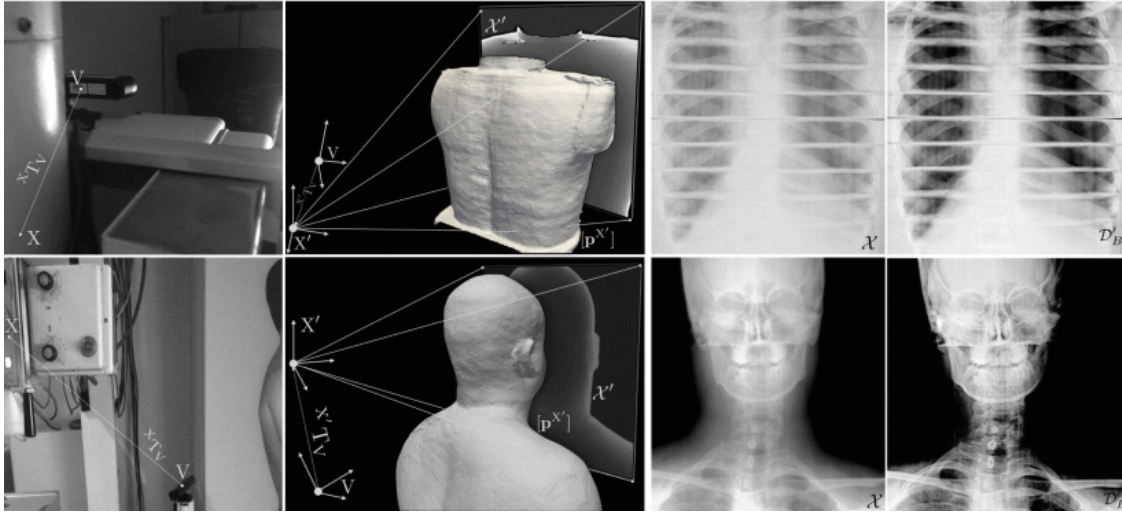


Figure 12: Examples of virtual densitometric image (\mathcal{D}'_B , 4th column). 1st column: experimental setup. 2nd column: iso-surfaces from the depth sensor. Virtual (high-energy) water-equivalent radiographs (\prime) from water-filled volumes [$\mathbf{p}X'$]. 3rd column: original radiographs (\mathcal{X}). Top row: chest section of a RANDO anthropomorphic phantom (105 kVp and 6 mAs). Bottom row: real patient (55 kVp and 9 mAs). Reproduced with permission from Albiol et al. [22].

It is also possible to integrate further simulation models to take into account the patient's physiology such as the respiration during the CT scan acquisition. In this case, reconstructed slices will exhibit motion artefacts [23]. Andreozzi et al. [24] used gVirtualXray to generate noise free X-ray radiographs. Noise is added as a post-process to study real-time edge-aware denoising in fluoroscopic devices. Corbi integrated gVirtualXray with other relevant software packages, such as Geant4, into a Docker container that is used online to teach particle physics and X-ray imaging to undergraduate students in engineering [25].

References

10.5281/zenodo.7757285

Illustrative examples

A file has been uploaded.

Acronyms

- API** application programming interface.
- CAD** computer-aided design.
- CG** computer graphics.
- CPU** central processor unit.
- CT** computed tomography.
- DRR** digitally reconstructed radiograph.
- eV** electron volt.
- GPU** graphics processor unit.
- gVXR** gVirtualXray.
- MC** Monte Carlo.
- ML** machine learning.

VR virtual reality.

References

- [1] N. Freud, J.-M. Létang, D. Babot, A hybrid approach to simulate x-ray imaging techniques, combining Monte Carlo and deterministic algorithms, *IEEE Transactions on Nuclear Science* 52 (5) (2005) 1329–1334. [doi:10.1109/TNS.2005.858223](https://doi.org/10.1109/TNS.2005.858223).
- [2] A. Badal, A. Badano, Accelerating Monte Carlo simulations of photon transport in a voxelized geometry using a massively parallel graphics processing unit, *Medical Physics* 36 (11) (2009) 4878–4880. [doi:10.1118/1.3231824](https://doi.org/10.1118/1.3231824).
- [3] D. F. Bosman, V. G. Balcaza, C. Delgado, S. Principi, M. A. Duch, M. Ginjaume, Validation of the MC-GPU Monte Carlo code against the PENELoPE/penEasy code system and benchmarking against experimental conditions for typical radiation qualities and setups in interventional radiology and cardiology., *Physica Medica* 82 (2021) 64–71. [doi:10.1016/j.ejmp.2021.01.075](https://doi.org/10.1016/j.ejmp.2021.01.075).
- [4] P. Duvauchelle, N. Freud, V. Kaftandjian, D. Babot, 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) (2000) 245–258. [doi:10.1016/S0168-583X\(00\)00185-3](https://doi.org/10.1016/S0168-583X(00)00185-3).
- [5] F. P. Vidal, M. Garnier, N. Freud, J. M. Létang, N. W. John, Simulation of x-ray attenuation on the GPU, in: *Proceedings of Theory and Practice of Computer Graphics 2009*, Eurographics Association, Cardiff, UK, 2009, pp. 25–32. [doi:10.2312/LocalChapterEvents/TPCG/TPCG09/025-032](https://doi.org/10.2312/LocalChapterEvents/TPCG/TPCG09/025-032).
- [6] F. P. Vidal, P.-F. Villard, Development and validation of real-time simulation of x-ray imaging with respiratory motion, *Computerized Medical Imaging and Graphics* 49 (2016) 1–15. [doi:10.1016/j.compmedimag.2015.12.002](https://doi.org/10.1016/j.compmedimag.2015.12.002).
- [7] N. Freud, P. Duvauchelle, J. M. Létang, D. Babot, 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) (2006) 175–180. [doi:10.1016/j.nimb.2006.03.009](https://doi.org/10.1016/j.nimb.2006.03.009).
- [8] P.-F. Villard, F. P. Vidal, C. Hunt, F. Bello, N. W. John, S. Johnson, D. A. Gould, A prototype percutaneous transhepatic cholangiography training simulator with real-time breathing motion, *International Journal of Computer Assisted Radiology and Surgery* 4 (6) (2009) 571–578. [doi:10.1007/s11548-009-0367-1](https://doi.org/10.1007/s11548-009-0367-1).
- [9] F. P. Vidal, M. Garnier, N. Freud, J. M. Létang, N. W. John, Accelerated deterministic simulation of x-ray attenuation using graphics hardware, in: *Eurographics 2010 - Poster*, Eurographics Association, Norrköping, Sweden, 2010, p. Poster 5011.
- [10] F. P. Vidal, I. T. Mitchell, J. M. Létang, Use of fast realistic simulations on GPU to extract CAD models from microtomographic data in the presence of strong CT artefacts, *Precision Engineering* 74 (2022) 110–125. [doi:10.1016/j.precisioneng.2021.10.014](https://doi.org/10.1016/j.precisioneng.2021.10.014).
- [11] J. L. Pointon, T. Wen, J. Tugwell-Allsup, A. Sújár, J. M. Létang, F. P. Vidal, Simulation of x-ray projections on gpu: Benchmarking gvirtualxray with clinically realistic phantoms, *Computer Methods and Programs in Biomedicine* 234 (2023) 107500. [doi:10.1016/j.cmpb.2023.107500](https://doi.org/10.1016/j.cmpb.2023.107500).
- [12] T. Schoonjans, A. Brunetti, B. Golosio, M. Sanchez del Rio, V. A. Solé, C. Ferrero, L. Vincze, The xraylib library for X-ray–matter interactions. recent developments, *Spectrochimica Acta Part B: Atomic Spectroscopy* 66 (11) (2011) 776–784. [doi:10.1016/j.sab.2011.09.011](https://doi.org/10.1016/j.sab.2011.09.011).
- [13] G. Sharp, R. LI, J. Wolfgang, G. Chen, M. Peroni, M. Spadea, S. Mori, J. Zhang, J. Shackleford, N. Kandasamy, Plastimatch – an open source software suite for radiotherapy image processing, in: *Proceedings of the XVI'th International Conference on the use of Computers in Radiotherapy (ICCR)*, Amsterdam, Netherlands, 2010.

- [14] J. S. Jørgensen, E. Ametova, G. Burca, G. Fardell, E. Papoutsellis, E. Pasca, K. Thielemans, M. Turner, R. Warr, W. R. B. Lionheart, P. J. Withers, 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) (2021) 20200192. [arXiv:https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2020.0192](https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2020.0192), [doi:10.1098/rsta.2020.0192](https://doi.org/10.1098/rsta.2020.0192).
- [15] A. Sujar, G. Kelly, M. García, F. P. Vidal, 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) (2022) 85–95. [doi:10.1007/s11548-021-02499-7](https://doi.org/10.1007/s11548-021-02499-7).
- [16] Z. Zuo, W. Y. Qian, X. Liao, P.-A. Heng, Position based catheterization and angiography simulation, in: 2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH), 2018, pp. 1–7. [doi:10.1109/SeGAH.2018.8401369](https://doi.org/10.1109/SeGAH.2018.8401369).
- [17] M. Racy, A. Barrow, J. Tomlinson, F. Bello, Development and validation of a virtual reality haptic femoral nailing simulator, *Journal of Surgical Education* 78 (3) (2021) 1013–1023. [doi:10.1016/j.jsurg.2020.10.004](https://doi.org/10.1016/j.jsurg.2020.10.004).
- [18] A. Sújar, G. Kelly, M. García, F. Vidal, Projectional Radiography Simulator: an Interactive Teaching Tool, in: F. P. Vidal, G. K. L. Tam, J. C. Roberts (Eds.), *Computer Graphics and Visual Computing (CGVC)*, The Eurographics Association, 2019, pp. 125–128. [doi:10.2312/cgvc.20191267](https://doi.org/10.2312/cgvc.20191267).
- [19] T. Wen, R. P. Mihail, F. P. Vidal, 3D-2D registration using x-ray simulation and CMA-ES, in: P. A. Castillo, J. L. Jiménez Laredo (Eds.), *Applications of Evolutionary Computation*, Springer International Publishing, Cham, 2021, pp. 453–468.
- [20] J. Lovitt, [Estimating length statistics of aggregate fried potato product via electromagnetic radiation attenuation](#), Master’s thesis, Boise State University, Idaho, USA (2016). URL <https://scholarworks.boisestate.edu/td/1211>
- [21] M. Haiderbhai, S. Ledesma, S. C. Lee, M. Seibold, P. Fűrnhstahl, N. Navab, P. Fallavollita, pix2xray: converting RGB images into x-rays using generative adversarial networks, *International Journal of Computer Assisted Radiology and Surgery* 15 (6) (2020) 973–980. [doi:10.1007/s11548-020-02159-2](https://doi.org/10.1007/s11548-020-02159-2).
- [22] F. Albiol, A. Corbi, A. Albiol, Densitometric radiographic imaging with contour sensors, *IEEE Access* 7 (2019) 18902–18914. [doi:10.1109/ACCESS.2019.2895925](https://doi.org/10.1109/ACCESS.2019.2895925).
- [23] F. P. Vidal, P.-F. Villard, Simulated motion artefact in computed tomography, in: K. Bühler, L. Linsen, N. W. John (Eds.), *Eurographics Workshop on Visual Computing for Biology and Medicine*, The Eurographics Association, 2015, pp. 213–214. [doi:10.2312/vcbm.20151228](https://doi.org/10.2312/vcbm.20151228).
- [24] E. Andreozzi, A. Fratini, D. Esposito, M. Cesarelli, P. Bifulco, Toward a priori noise characterization for real-time edge-aware denoising in fluoroscopic devices, *BioMedical Engineering OnLine* 20 (1) (2021) 36. [doi:10.1186/s12938-021-00874-8](https://doi.org/10.1186/s12938-021-00874-8).
- [25] A. Corbi, D. Burgos, F. Vidal, F. Albiol, A. Albiol, X-ray imaging virtual online laboratory for engineering undergraduates, *European Journal of Physics* 41 (1) (2019) 014001. [doi:10.1088/1361-6404/ab5011](https://doi.org/10.1088/1361-6404/ab5011).