Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Open-Source Tool for Converting 3D Mesh volumes into Synthetic DICOM CT Images for Medical Physics Research

Michael Douglass

michael.douglass@adelaide.edu.au

University of Adelaide and Royal Adelaide Hospital https://orcid.org/0000-0002-3862-2644

Research Article

Keywords: Synthetic CT, Medical Imaging, Radiation Oncology, 3D Mesh, DICOM, Treatment Planning

Posted Date: May 7th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6356394/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Version of Record: A version of this preprint was published at Physical and Engineering Sciences in Medicine on July 24th, 2025. See the published version at https://doi.org/10.1007/s13246-025-01599-x.

Abstract

Background

Access to medical imaging data is crucial for research, training, and treatment planning in medical imaging and radiation therapy. However, ethical constraints and time-consuming approval processes often limit the availability of such data. This study introduces DICOMator, an open-source Blender add-on designed to address this challenge by enabling the creation of synthetic CT datasets from 3D mesh objects.

Purpose

DICOMator aims to provide researchers and medical professionals with a flexible tool for generating customisable and qualitatively realistic synthetic CT data, including 4D CT capabilities. The add-on leverages Blender's powerful 3D modelling environment, utilising its mesh manipulation, and rendering capabilities to create synthetic data ranging from simple phantoms to accurate anatomical models. DICOMator incorporates various features to simulate common CT imaging artefacts, bridging the gap between 3D modelling and medical imaging.

Methods

The methodology involves voxelising 3D mesh objects, assigning appropriate Hounsfield Unit values, and applying artefact simulations. These simulations include detector noise, metal artefacts and partial volume effects. By incorporating these artefacts, DICOMator produces synthetic CT data that more closely resembles real CT scans. The resulting data is then exported in DICOM format, ensuring compatibility with existing medical imaging workflows and treatment planning systems.

Results

To demonstrate DICOMator's capabilities, three synthetic CT datasets were created: a simple lung phantom to illustrate basic functionality, a realistic cranial CT scan to demonstrate complex anatomical modelling, and a thoracic 4DCT scan featuring multiple breathing phases to demonstrate dynamic imaging capabilities. These examples were chosen to highlight DICOMator's versatility in generating diverse and complex synthetic CT data suitable for various research and educational purposes, from basic quality assurance to advanced motion management studies.

Conclusions

DICOMator offers a promising solution to the limitations of patient CT data availability in medical physics research. By providing a user-friendly interface for creating customisable synthetic datasets, it has the potential to accelerate research, improve treatment planning algorithms, and enhance educational resources in the field of radiation oncology. Future developments may include integration with other imaging modalities, such as MRI or PET, further expanding its utility in multi-modal imaging research.

Introduction

Advancements in medical imaging and radiation therapy have significantly improved the diagnosis and treatment of various diseases, particularly cancer. Central to these advancements in radiation oncology is the ability to accurately image and simulate patient anatomy and the interaction of radiation with biological tissues. This capability has enabled modern radiation therapy approaches such as volumetric modulated arc therapy (VMAT) and proton therapy.

A significant challenge in medical physics research is the limited access to patient CT data. Such data is valuable for retrospective studies, training machine learning models, and radiomic analyses. Obtaining and using CT data for research typically requires formal approval from hospital ethics committees, a process that can be time-consuming and complex. While open-access datasets are available [1], they may not always provide a suitable patient cohort for specific research projects.

Synthetic data offers an alternative solution in some scenarios, bypassing ethical issues [2] and allowing the creation of more customised datasets tailored to specific research needs. Custom synthetic datasets often provide more controlled environments with fewer uncontrollable variables, allowing for more precise and targeted research.

Synthetic data in radiation oncology has already been extensively utilised for various purposes, including training machine learning models [3–6], validating computational algorithms [7], education [2] and inter-modality medical image conversion [5,8,9]. While these applications have significantly contributed to advancing the field, there has been a limitation on creating customizable, complex, and realistic synthetic datasets that can be seamlessly imported into treatment planning systems.

This limitation has hindered the development of more sophisticated research methodologies and the evaluation of treatment planning algorithms under diverse and challenging scenarios.

To address this gap, we have developed DICOMator [10], a Blender [11] add-on that bridges the robust modelling capabilities of Blender with medical imaging formats such as DICOM (Digital Imaging and Communications in Medicine) [12]. DICOM is the international standard for medical images, ensuring compatibility across different medical imaging devices and software. DICOMator facilitates the voxelisation of 3D mesh objects and their export as DICOM-compliant computed tomography (CT) series, enabling the generation of highly detailed and adaptable synthetic CT datasets that closely mimic real patient data, complete with simulated artifacts and anatomical variations. Ultimately, the goal of this

tool is to quickly and easily generate synthetic CT images that are similar enough to real CT data for specific use cases.

DICOMator leverages Blender's [11] native capabilities to process 3D mesh objects, converting them into volumetric data. Blender, an open-source 3D creation suite, offers powerful modelling tools, a flexible Python [13] API, and a large community of users and developers, making it an ideal platform for this application. DICOMator optimises the voxelisation process by focusing on relevant areas, enhancing overall performance. The tool offers flexible density assignment methods, allowing users to precisely control material representation in complex geometries. These features enable researchers to explore a wide range of clinical scenarios without the ethical and logistical constraints associated with real patient data.

Although mesh voxelisation algorithms have been extensively studied [14–16], current solutions predominantly lack intuitive, user-friendly tools tailored specifically for medical physics researchers. Existing workflows typically rely on external three-dimensional models (e.g., Standard Triangle Language [STL] format), followed by manual voxelisation and conversion processes to obtain DICOM data suitable for import into clinical treatment planning systems [17,18]. This approach is both cumbersome and limits flexibility, particularly when developing complex virtual phantoms or dynamic simulations. DICOMator addresses these critical limitations by integrating voxelisation directly within the Blender environment, enabling interactive, real-time creation and refinement of synthetic CT datasets. Its intuitive interface streamlines the entire process, allowing researchers to seamlessly generate and export phantoms, including those with complex geometries and spatially varying Hounsfield Units (HU). Additionally, the novel integration of Blender’s built-in animation features facilitates the generation of advanced 4DCT datasets, significantly expanding the scope of phantom-based studies and enhancing practical utility for medical physics research and quality assurance applications.

A key feature of DICOMator is its ability to generate four-dimensional (4D) CT data, capturing motion or changes over time. This functionality is particularly valuable for simulating anatomical movements in radiotherapy treatment planning, where organ motion can significantly impact treatment accuracy and robustness. In radiation therapy, accounting for organ motion (such as breathing-induced movement of lung tumours) is crucial for accurate dose delivery and minimising damage to healthy tissue. By providing 4D CT capabilities, DICOMator facilitates the development and evaluation of advanced motion management strategies in a controlled, customisable environment.

The tool's output in DICOM CT format ensures compatibility with existing medical imaging workflows and treatment planning systems. This seamless integration allows for easy importation of synthetic CT images, including 4D datasets, into radiotherapy planning software. Such capability proves beneficial for educational purposes, research and commissioning of new treatment techniques, and the design and evaluation of novel phantoms.

DICOMator's user-friendly interface within Blender [11] streamlines the workflow, offering intuitive controls for various parameters and settings. This ease of use enhances productivity when working with

multiple mesh objects and animations, making it an efficient tool for creating complex, customised synthetic CT datasets.

While synthetic data offers many advantages, it is important to acknowledge potential limitations. Synthetic datasets may not fully capture the complexity and variability of real patient data, and care must be taken to ensure that algorithms trained on synthetic data perform well on real-world cases. Additionally, the creation of realistic synthetic data requires expertise in both medical imaging and 3D modelling.

This paper presents the development and implementation of DICOMator, detailing its features—including the 4D CT capabilities—underlying algorithms, and potential applications in medical physics. We discuss optimisation strategies employed to improve performance, such as restricting voxelisation to relevant regions and efficient density assignments. The utility of DICOMator is demonstrated through three case studies: a simple lung phantom to illustrate basic functionality, a realistic cranial CT scan demonstrating complex anatomical modelling, and a thoracic 4DCT scan featuring multiple breathing phases to demonstrate 4D imaging capabilities. These examples highlight DICOMator's versatility in generating diverse and complex synthetic CT data suitable for various research and educational purposes, from basic quality assurance to advanced motion management studies.

Method

DICOMator, written in Python [13] using the Blender scripting API [19], facilitates the conversion of complex 3D meshes into medical DICOM CT images. The extension not only voxelises the selected meshes but also simulates various imaging artefacts commonly encountered in clinical CT scans. The following sections provide a detailed description of the steps taken by the code and examples of how users can use its features.

A mesh object is a collection of vertices, edges, and faces that define the shape of a 3D object. These objects are typically created using computer-aided design (CAD) software or 3D modelling tools like Blender [11]. Mesh objects can represent complex anatomical structures, medical devices, or any other three-dimensional structures relevant to medical imaging and radiation therapy planning. The flexibility of 3D mesh objects allows for precise representation of both simple geometries and intricate anatomical details, making them ideal starting points for generating synthetic CT datasets.

The conversion of these mesh objects into a format compatible with medical imaging systems involves a process called voxelisation. Voxelisation transforms the continuous geometry of a mesh into a discrete 3D grid of volumetric pixels, or voxels, each representing a specific density value. In the case of CT medical imaging, these values are typically "Hounsfield units," which measure the amount a photon beam is attenuated by the material in each voxel. This attenuation is related to physical or electron density of the material using a CT density conversion curve in a radiation therapy treatment planning system.

The voxelisation process converts between the vector-based representation of 3D models and the raster-based format of CT images, enabling the creation of synthetic datasets that visually mimic real CT scans.

Voxelisation Process

The voxelization method implemented in this study employs an adaptive sampling technique designed to convert 3D mesh objects into a detailed voxel grid representation suitable for generating synthetic DICOM image series. The voxelization procedure comprises the following steps:

Bounding Box Calculation

The spatial extent of the voxel grid is determined by calculating the combined bounding box of all selected mesh objects. Minimum (bbox_min) and maximum (bbox_max) coordinates of the bounding box are computed and stored for subsequent processing.

Grid Dimensions Calculation

The dimensions of the voxel grid are established based on the computed bounding box size. The number of voxels along each dimension is determined by dividing the bounding box dimensions by the voxel size and rounding up to ensure full coverage.

Voxel Grid Initialization

A three-dimensional numpy [20] array is initialized to represent the voxel grid, initially populated with air-equivalent HU values (-1000 by default). Concurrently, a mask array of identical dimensions is initialized to track boundary voxels.

Mesh Data Preparation

Each mesh object undergoes data preparation involving retrieval of density and priority attributes, creation and triangulation of a BMesh representation, and construction of a Bounding Volume Hierarchy (BVHTree) to facilitate efficient spatial queries. BMesh is Blender’s dynamic mesh editing system. It provides a flexible data structure for modifying vertices, edges, and faces—including complex n-gon shapes. BVHTree is a hierarchical spatial data structure that speeds up geometric queries like ray-casting and nearest-neighbour searches, analogous to efficient collision detection in physics simulations. Prepared mesh data are stored in a structured list for subsequent voxelization steps.

First Pass: Low-Resolution Sampling

An initial, low-resolution sampling pass is performed using a ray casting [21] approach to determine if sample points are inside mesh structures (Fig. 1). Sample points are generated at the geometric center of each voxel, and neighbouring voxels are evaluated for differences in density or internal/external status to detect boundary voxels.

Boundary Detection

The voxel grid is segmented into manageable chunks to optimize memory usage during processing. For each voxel, the spatial coordinates of its minimum corner are calculated, and ray casting is applied to ascertain if the voxel center is located within a mesh volume. Density and priority data are recorded accordingly. Differences between neighbouring voxels' density or inclusion status are analysed to accurately identify boundary voxels, which are marked within the boundary mask for further processing.

Second Pass: High-Resolution Sampling

Boundary voxels identified during the initial sampling pass undergo a secondary, higher-resolution sampling procedure. Within each boundary voxel, multiple high-resolution sample points are generated, and a ray casting approach is again employed to determine if each sample point lies within a structure. The density values obtained from these sample points are averaged to refine voxel density assignments, enhancing the representation accuracy near surfaces.

Cleanup and Output

Post-processing cleanup involves freeing BMesh objects to reclaim memory resources. The finalized voxel grid, with density values precisely assigned through adaptive sampling, is then outputted. This adaptive voxelization approach effectively balances computational efficiency with high spatial accuracy, providing detailed structure representation particularly suited to medical imaging applications.

Artefact Simulation Tools

After iterating over all voxels within the bounding box and processing all selected 3D mesh objects, the code produces an idealised CT dataset without artefacts. To enhance the realism of this synthetic dataset, the software offers several options to simulate typical image artefacts commonly found in CT scans.

Noise

Users can simulate Gaussian noise to replicate the random noise inherent in CT imaging systems. The add_noise function simulates random noise in the synthetic CT image by adding Gaussian noise to the voxel grid representing the image data. Specifically, if the noise_std_dev (standard deviation of the noise)

parameter is greater than zero, the function generates a noise array of the same shape as the voxel grid, with values drawn from a normal distribution centred at zero. This noise array is then added to the original voxel grid, introducing random fluctuations in voxel intensities throughout the entire 3D image. This process mimics the random noise commonly present in real CT images due to factors like electronic sensor noise and photon counting statistics, thereby enhancing the realism of the simulated data for testing and validation purposes.

Metal Artifacts

Metal artefact simulation replicates the artefacts caused by metal implants, appearing as streaks emanating from high-density regions. The function simulates bright radial streak artefacts in a computed tomography (CT) image by altering voxel values around high-density regions within each axial slice. It operates by iterating over each slice along the axial direction of the 3D voxel grid representing the CT image. For every slice, it identifies voxels that exceed a specified metal density threshold, classifying them as metal voxels.

Around each identified metal voxel, the function projects multiple rays outward in various directions determined by the num_angles parameter, which specifies the number of angles to cover a full 360-degree circle. Along these rays, it increases the intensity of the voxel values to create bright streaks. An attenuation factor based on the radial distance from the metal voxel is applied, causing the intensity effect to diminish as it moves away from the source, mimicking the physical attenuation observed in real CT images.

After modifying the voxel intensities along all rays for all metal voxels in the slice, the function ensures that the voxel values remain within the valid Hounsfield Unit (HU) range for CT images by clipping them appropriately. It then updates the original voxel grid by replacing the original slice with the modified slice containing the simulated artefacts.

By repeating this process for each slice containing metal voxels, the function effectively simulates bright radial streak artefacts commonly observed around metal objects in CT images.

Partial Volume Effects

Partial volume effects occur in medical imaging when a single voxel contains multiple tissue types, resulting in an averaged HU intensity for that voxel. This phenomenon leads to blurring of boundaries between different tissues and can affect the accuracy of structure delineation, especially for small structures or regions with rapid density changes.

The apply_partial_volume_effect function simulates the partial volume effect in computed tomography (CT) images by applying a Gaussian blur to the voxel grid representing the image data. It uses the gaussian_filter function from the scipy. ndimage[22] module, where the sigma parameter determines the standard deviation of the Gaussian kernel used for blurring. By convolving the voxel grid with this

Gaussian kernel, the function smooths the image, reducing high-frequency details and mimicking the averaging of HU that occurs when a single voxel encompasses multiple tissue types in real CT imaging.

Ring Artefacts

The ring artefact simulation mimics simple ring artefacts that can occur due to detector imperfections or calibration errors in CT scanners. The simulate_ring_artefacts function simulates ring artefacts in the synthetic computed tomography (CT) image by introducing concentric sinusoidal ring patterns to each axial slice of the voxel grid. For each slice, it calculates the radial distance from the centre of the slice to every voxel using the Euclidean distance. It then generates a ring pattern by applying a sine function over these radial distances, with the frequency parameter controlling the number of rings. The resulting ring pattern is scaled by the specified intensity and added to the original slice data, modifying the voxel intensities to mimic ring artefacts. This process is repeated for all slices in the voxel grid, introducing ring artefacts throughout the entire CT volume.

DICOM CT Series Export

Once all post-processing effects have been applied, the voxel grid is exported as a series of DICOM CT image slices, compatible with medical imaging software. The code uses the "pydicom" [23] library to create and populate DICOM file datasets, generating unique identifiers (UIDs) for the study, series, and instances using the "pydicom.uid.generate_uid" function. Essential metadata such as patient information, study and series dates, modality-specific attributes, and image type are assigned to each DICOM file. The code sets modality-specific attributes such as Modality (CT), Manufacturer and Software Versions (1.0), and assigns patient information, which can be customised.

Spatial information of each slice is represented using the Image Position (Patient) and Image Orientation (Patient) tags, calculated based on the user defined voxel size and bounding box coordinates.

For 4D datasets, temporal position identifiers are added to the DICOM files to represent different temporal phases. The code includes temporal identifiers such as the temporal_position_index and NumberOfTemporalPositions tags, allowing the DICOM series to represent time-varying data.

A flowchart representing this workflow is shown in Fig. 2 and a list of all DICOM tags exported in the synthetic DICOM series can be found in the code repository [10].

Testing and Verification

To demonstrate the capabilities of the add-on, three synthetic CT datasets of varying complexity were created. Each example was designed to illustrate a potential use case for medical physics research.

Simple lung phantom using basic 3D shapes

For the first example, a simple lung phantom was created, consisting of three cylindrical meshes, stretched in the x-y (lateral, ant-post) plane to simulate the appearance of a basic lung anatomy. One

cylinder represented the external contour of a human body, while the other two cylinders represented the left and right lung volumes (Fig. 3).

The external cylinder was assigned a Hounsfield Unit (HU) value of 0, to approximate that of water. The two lung cylinders were assigned a value of -200 HU, approximating the density of a typical lung material.

Two different synthetic DICOM CT image series were created from this geometry; one with the lung volumes located 1 cm superior to the centre of the external structure and a second with the lung volumes positioned 1 cm inferior. This was done to demonstrate the potential of this software for validating rigid and deformable image registration (DIR) algorithms. The two CT datasets were exported as DICOM series and then imported into Raystation (Raysearch Laboratories, Sweden) Version 12A where the external and lung volumes were automatically segmented using threshold-based segmentation.

The two CT volumes were then deformably registered to each other using the Hybrid intensity and structure-based algorithm. The default deformation strategy was selected using the correlation coefficient as the similarity measure. The lung volumes were selected as the controlling ROIs.

Once registered, twelve points were placed on the superior and inferior edges of the lung volumes to determine the magnitude of the displacement field between the two datasets to confirm that the value matched the 2 cm value originally simulated in Blender.

In addition, as a further demonstration of the metal artefact simulator tool, a third CT dataset, using the simple lung phantom, was exported from Blender, where a small spherical volume was placed in the region between the lung volumes and assigned a HU value of 2000, representative of a high-density implant. The purpose of this was to demonstrate the simulated streaking artefacts from the metal volume.

Synthetic cranial CT scan with realistic human anatomy

For the second example, a pseudo-realistic human cranial CT was created using a multi-step process. First, a CT image of a human head was obtained from the CERMEP-IDB-MRXFDG dataset [24]. The CT image series was imported into Slicer 3D [25] where the Total Segmentator [26] deep learning extension was used to automatically contour the patient's external, brain, and skull volume.

After segmentation, these volumes were converted to 3D models and exported as .obj files, a common text-based 3D format. The resulting models, representing the external, skull, and brain structures, were then imported into Blender (Fig. 4). DICOMator was used to assign an appropriate Hounsfield Unit (HU) value to each structure: 0 HU for the external structure, 1000 HU for the skull, and 200 HU for the brain. The noise artefact tool was enabled with a Gaussian noise sigma of 5 HU. The geometry was voxelised with a grid spacing of 2 mm and exported as a DICOM CT series to the Raystation treatment planning system. Three spherical PTV volumes were manually contoured within the brain volume using the brush

tool in the Patient Modelling section of Raystation to serve as targets for the optimiser. A simple VMAT plan was produced to deliver a uniform dose of 10 Gy to the PTV targets while minimising dose to the brain volume. This was achieved using the dose falloff objective in the optimiser.

The purpose of this example was to demonstrate the ability of the add-on to produce synthetic CT volumes for dose calculations in treatment planning systems.

Thoracic 4DCT scan with realistic anatomy

Finally, the 4DCT feature of DICOMator was demonstrated by creating a 4D thoracic DICOM CT series with ten temporal phases. To achieve this, a CT scan of a thoracic region was obtained from the Visible Human Project dataset [27]. The dataset was subsequently imported into Slicer3D [25], where the Total Segmentator [26] extension was used to automatically segment the external contour of the patient, the lung volumes, and the heart.

These segmentations were then exported in OBJ format from Slicer3D and imported into Blender. Within Blender, the external structure, lungs, and heart regions were assigned Hounsfield Unit (HU) values of 0, -200, and 100, respectively, to simulate typical tissue densities for each region (Fig. 5). The noise artefact option was enabled with a Gaussian noise sigma of 5 HU.

To simulate the respiratory motion of the lungs, the lung volumes were animated in Blender by scaling the lung 3D meshes along the superior-inferior axis relative to the most superior point of the lung volumes and key-framed at multiple steps in the breathing cycle. The animation was configured to run for ten frames, representing ten breathing phases of a 4DCT.

Finally, frames 0–9 were voxelised with a resolution of 2.5 mm and exported as a 4D DICOM series to the Raystation treatment planning system.

Results

Simple lung phantom using basic 3D shapes

The two synthetic lung phantom CTs were deformably registered to each other in Raystation, with the displacement field visualised and quantified. Twelve points were placed at the superior and inferior edges of the lung volumes and the magnitude of the displacement field between the deformably registered CT images was quantified. The magnitude of the displacements in the R-L, I-S and P-A directions for the twelve points were 0.001 ± 0.007 cm, 1.998 ± 0.055 cm, and 0.001 ± 0.035 cm, respectively (Fig. 6). This result matches the expected displacement of 0 cm, 2 cm and 0 cm simulated in Blender.

In addition, a visual demonstration of the metal artefact simulation tool was performed. The result of this is illustrated in Fig. 7 where a series of streaks emanating from the simulated metal object can be

seen. The radial attenuation, number of streaks and intensity of the streaks are all customisable via the DICOMator user interface.

Synthetic cranial CT scan with realistic human anatomy

The synthetic cranial CT generated in Blender was exported as a DICOM CT series and successfully imported into the Raystation treatment planning system. The synthetic CT image series with simulated HU values was converted to physical density using the clinical HU-to-Density conversion curve in the Raystation. The average physical densities for the normal tissue (external – Brain – skull), brain and skull regions reported by Raystation were 1.003 g/cm^3, 1.070 g/cm^3 and 1.66 g/cm^3, respectively.

To demonstrate a potential application of this synthetic dataset, a sample Volumetric Modulated Arc Therapy (VMAT) radiotherapy treatment plan was created in Raystation. This plan consisted of two full VMAT arcs designed to uniformly irradiate three spherical targets, while minimising dose to the surrounding brain tissue using the dose fall-off objective (Fig. 8). This example highlights the potential of the add-on for generating realistic datasets for treatment planning and dosimetry studies.

Thoracic 4DCT scan with realistic anatomy

Finally, the synthetic 4DCT lung volume was exported from Blender and imported into Raystation. Raystation was able to successfully import and interpret the synthetic lung dataset as a 4D CT volume and display the animated lung volume (Fig. 9). Raystation was also able to successfully recognise and use this dataset in the robust optimisation and robust evaluation tools.

This process demonstrated the capability of the add-on to generate realistic 4D-CT datasets, which can be valuable for research and clinical applications in respiratory motion management and adaptive radiotherapy.

Discussion

The work presented herein demonstrates a wide range of potential applications for the proposed DICOMator add-on in the field of medical physics research and development. For image processing, it excels as a tool for validating various image registration algorithms, including rigid and deformable registration techniques. Researchers can create synthetic datasets with pre-defined deformations, providing a controlled environment with known ground truth for assessing the accuracy and robustness of new algorithms.

Furthermore, the effectiveness of the add-on in creating simple or realistic phantoms for various medical physics applications has been demonstrated. Notably, the software simulates various CT artefacts, including partial volume effects, metal artefacts, and noise. These artefacts, which are common in clinical CT scans, add a layer of realism to the synthetic dataset, making it particularly valuable for research and educational purposes in medical physics.

In this study, we employed two open-source patient datasets as a template for the generation of the initial synthetic phantom. While this approach may appear counterintuitive—given that the primary objective of the DICOMator tool is to produce fully synthetic datasets—it serves a critical role in demonstrating the tool's flexibility. Following segmentation, the anatomical structures derived from the patient CT data were converted into individual 3D meshes, which could then be programmatically manipulated. This facilitated the generation of a wide range of synthetic phantoms through systematic variation of parameters such as tissue density, organ size, and respiratory motion. As a result, an extensive array of synthetic datasets can be derived from a single anatomical template. Thus, the initial use of real patient data does not compromise the synthetic nature or purpose of the DICOMator framework; rather, it underscores its utility and adaptability for diverse applications in medical physics research.

For machine learning applications, the add-on may prove invaluable for generating synthetic data for model training. This is particularly beneficial when real patient data is scarce or when simulating specific rare conditions. Initially, machine learning models can be trained using synthetic data generated by DICOMator and subsequently fine-tuned with a smaller sample of actual patient data. Moreover, using synthetic data addresses ethical concerns related to patient privacy and data protection.

In education, DICOMator enables the creation of customised datasets for teaching and training. For instance, educators can design cases that demonstrate the effects of metal artefacts on dose calculations or simulate various stages of tumour growth, providing students with hands-on experience across a diverse range of scenarios without using real patient data.

For treatment planning and quality assurance, DICOMator offers several key applications. It enables the verification of dose calculation accuracy in the presence of metal artefacts, a common challenge in radiotherapy planning. By creating controlled scenarios with known artefact patterns, medical physicists can assess the impact of these artefacts on dose calculations and develop strategies to mitigate their effects. Furthermore, researchers can use the tool to study the effect of image noise on dose distribution, helping to optimise imaging protocols and understand the relationship between image quality and treatment planning accuracy.

In motion management, DICOMator's capability for simulated organ motion allows for the development and testing of advanced motion management strategies in radiation therapy. This feature is particularly relevant for improving the precision of treatments in areas affected by respiratory or cardiac motion. By creating realistic 4D datasets, researchers can evaluate different motion compensation techniques and develop more effective strategies for delivering accurate treatments to moving targets.

DICOMator also plays a significant role in quality assurance by facilitating the creation of "ideal" physics phantom datasets. This ensures consistent quality across different institutions and helps maintain high standards in medical physics practices. DICOMator offers flexibility in creating custom phantoms and integrates seamlessly with the Blender environment, making it uniquely positioned for research and educational purposes.

Looking ahead, DICOMator opens new research directions, such as developing advanced artefact reduction techniques for 4D imaging and optimising protocols for adaptive radiotherapy. Its versatility also extends beyond medical physics, with potential applications in biomedical engineering and computer vision research, further broadening its impact on interdisciplinary studies.

Limitations

Despite DICOMator's significant capabilities, it is important to acknowledge several limitations. The current version does not accurately replicate all types of CT artefacts, particularly more complex ones such as scatter or beam hardening from highly attenuating materials. Nor does it generate artefacts based on realistic physics models. In addition, the current version of the software assigns uniform HU values for each structure, which, while improving over existing tools [17,18], may not fully capture the heterogeneity of real tissues. While noise can be added, it may not perfectly mimic the complex noise patterns found in real CT scanners. The primary aim of this work was to develop synthetic datasets which were visually and qualitatively similar real CT data for specific use cases. Further work is needed to enhance the quantitative and visual accuracy of the synthetic images.

Another limitation relates to voxelisation time. Complex models with intricate topology require a larger number of sample rays to correctly voxelise the geometry, with the voxelisation time increasing with the selected number of rays. All examples in this study were generated on a PC with a 13th Generation Intel (Santa Clara, California, USA) Core® i9-13900HX CPU with 32 Gb of RAM. Simple phantoms, such as the lung phantom in example one, took only a few seconds to create. However, for more complex geometries, like the synthetic 4D lung in example three, the voxelisation process took approximately 15 minutes. The original 3D models for the external body, heart, and lungs contained between 10,000 and 15,000 vertices each. While the current code uses only the CPU, future improvements could potentially leverage GPU processing power for faster computation.

Future Work

Building upon the current capabilities of DICOMator, several avenues for future development and enhancement have been identified. The realism of existing artefact simulations, particularly for complex artefacts such as beam hardening and scatter, could be improved. This would involve implementing more sophisticated models to accurately mimic realistic CT artefacts, providing a more comprehensive tool for researchers and clinicians studying image quality and artefact reduction techniques.

Extending DICOMator's capabilities to simulate other imaging modalities, such as MRI, PET, and SPECT, would expand the tool's utility in multi-modal imaging research. This would involve developing new algorithms to mimic the unique characteristics and artifacts of each modality. Furthermore, implementing more sophisticated tissue models that better represent the heterogeneity and complexity of human anatomy could enhance the tool's realism. This could include incorporating texture maps for organs and simulating variations in tissue density within structures.

To improve performance, optimising the voxelisation process by leveraging GPU computing could significantly reduce processing times for complex geometries and enable the creation of higher-resolution synthetic datasets. Integration with popular deep learning frameworks through plugins or APIs would facilitate the generation of large-scale synthetic datasets for training and validation of AI models in medical imaging.

Enhancing the 4D simulation features to include more complex motion patterns and physiological processes, such as cardiac motion, blood flow, and contrast agent dynamics, would further expand DICOMator's capabilities in dynamic imaging simulation.

These future developments aim to further establish DICOMator as a comprehensive, versatile tool for medical imaging research, education, and quality assurance, addressing the evolving needs of the medical physics community.

Conclusion

In summary, DICOMator has been demonstrated to be useful in medical physics, offering a flexible and versatile solution for generating synthetic CT datasets. This open-source Blender add-on bridges a crucial gap in research, education, and quality assurance by enabling the creation of customizable, realistic CT images without relying on actual patient data or physical phantoms. Its open-source nature fosters collaborative improvement and widespread adoption within the medical physics community.

Demonstrated through examples such as simple lung phantoms, pseudo-realistic human head CTs, and 4DCT lung scans, DICOMator enhances various aspects of medical physics research, including validating image registration algorithms, training machine learning models, and improving treatment planning and motion management strategies.

While acknowledging current limitations, such as the simplification of certain tissue properties and artifacts, DICOMator provides a solid foundation for future developments.

As medical physics continues to evolve, tools like DICOMator will play an increasingly crucial role in driving innovation, enhancing educational practices, and improving patient care. By offering researchers, educators, and clinicians a flexible platform for synthetic CT generation, DICOMator paves the way for more robust, efficient, and ethical advancements in medical imaging and radiation therapy, ultimately enhancing treatment planning accuracy and patient outcomes in radiation oncology.

Declarations

The author has no conflicts of interest to disclose. OpenAI (San Francisco, California, USA) GPT o1-preview and 4o large language models were used to assist in parts of the code development and debugging. Microsoft (Redmond, Washington, USA) Copilot for Word was used to assist in manuscript editing to improve clarity and to correct grammar and spelling.

References

1. Clark K, Vendt B, Smith K, et al. The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. *J Digit Imaging*. 2013;26(6):1045-1057. doi:10.1007/s10278-013-9622-7
2. Koetzier LR, Wu J, Mastrodicasa D, et al. Generating Synthetic Data for Medical Imaging. *Radiology*. 2024;312(3). doi:10.1148/radiol.232471
3. Keal J, Santos A, Penfold S, Douglass M. Radiation dose calculation in 3D heterogeneous media using artificial neural networks. *Med Phys*. 2021;48(5):2637-2645. doi:10.1002/mp.14780
4. Douglass MJJ, Keal JA. DeepWL: Robust EPID based Winston-Lutz analysis using deep learning, synthetic image generation and optical path-tracing. *Physica Medica*. 2021;89:306-316. doi:10.1016/j.ejmp.2021.08.012
5. Douglass M, Gorayski P, Patel S, Santos A. Synthetic cranial MRI from 3D optical surface scans using deep learning for radiation therapy treatment planning. *Phys Eng Sci Med*. 2023;46(1):367-375. doi:10.1007/s13246-023-01229-4
6. Brouwer CL, Dinkla AM, Vandewinckele L, et al. Machine learning applications in radiation oncology: Current use and needs to support clinical implementation. *Phys Imaging Radiat Oncol*. 2020;16:144-148. doi:10.1016/j.phro.2020.11.002
7. Kerns JR. Pylinac: Image analysis for routine quality assurance in radiotherapy. *J Open Source Softw*. 2023;8(92):6001. doi:10.21105/joss.06001
8. Jarrett D, Stride E, Vallis K, Gooding MJ. Applications and limitations of machine learning in radiation oncology. *Br J Radiol*. 2019;92(1100). doi:10.1259/bjr.20190001
9. Liu X, Emami H, Nejad-Davarani SP, et al. Performance of deep learning synthetic CTs for MR-only brain radiation therapy. *J Appl Clin Med Phys*. 2021;22(1):308-317. doi:10.1002/acm2.13139
10. Douglass M. DICOMator. https://github.com/drmichaeldouglass/DICOMator.
11. Blender Online Community. Blender - a 3D modelling and rendering package. Accessed August 5, 2022. http://www.blender.org
12. National Electrical Manufacturers Association RVU. Digital Imaging and Communications in Medicine (DICOM) Standard. NEMA PS3 / ISO 12052. *http://www.dicomstandard.org/*.
13. Python Software Foundation. Python Language Reference. https://www.python.org/about/.
14. Aleksandrov M, Zlatanova S, Heslop DJ. Voxelisation Algorithms and Data Structures: A Review. *Sensors*. 2021;21(24):8241. doi:10.3390/s21248241
15. Nita C, Stroia I, Itu L, et al. GPU accelerated, robust method for voxelization of solid objects. In: *2016 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE; 2016:1-5. doi:10.1109/HPEC.2016.7761582
16. Schwarz M, Seidel HP. Fast parallel surface and solid voxelization on GPUs. *ACM Trans Graph*. 2010;29(6):1-10. doi:10.1145/1882261.1866201
17. khkurzeja. Voxelizer. https://github.com/khkurzeja/Voxelizer.

18. Qian Chen. voxelizer. https://github.com/topskychen/voxelizer.
19. Blender Foundation. Blender API. https://docs.blender.org/api/current/index.html.
20. Harris CR, Millman KJ, van der Walt SJ, et al. Array programming with NumPy. *Nature*. 2020;585(7825):357-362. doi:10.1038/s41586-020-2649-2
21. Roth SD. Ray casting for modeling solids. *Computer Graphics and Image Processing*. 1982;18(2):109-144. doi:10.1016/0146-664X(82)90169-1
22. Virtanen P, Gommers R, Oliphant TE, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*. 2020;17(3):261-272. doi:10.1038/s41592-019-0686-2
23. Mason DL, et al. pydicom: An open source DICOM library. https://github.com/pydicom/pydicom.
24. Mérida I, Jung J, Bouvard S, et al. CERMEP-IDB-MRXFDG: a database of 37 normal adult human brain [18F]FDG PET, T1 and FLAIR MRI, and CT images available for research. *EJNMMI Res*. 2021;11(1):91. doi:10.1186/s13550-021-00830-6
25. Fedorov A, Beichel R, Kalpathy-Cramer J, et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn Reson Imaging*. 2012;30(9):1323-1341. doi:10.1016/j.mri.2012.05.001
26. Wasserthal J, Breit HC, Meyer MT, et al. TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images. *Radiol Artif Intell*. 2023;5(5). doi:10.1148/ryai.230024
27. U.S. National Library of Medicine. Visible Human Project. https://www.nlm.nih.gov/research/visible/visible_human.html.

Figures

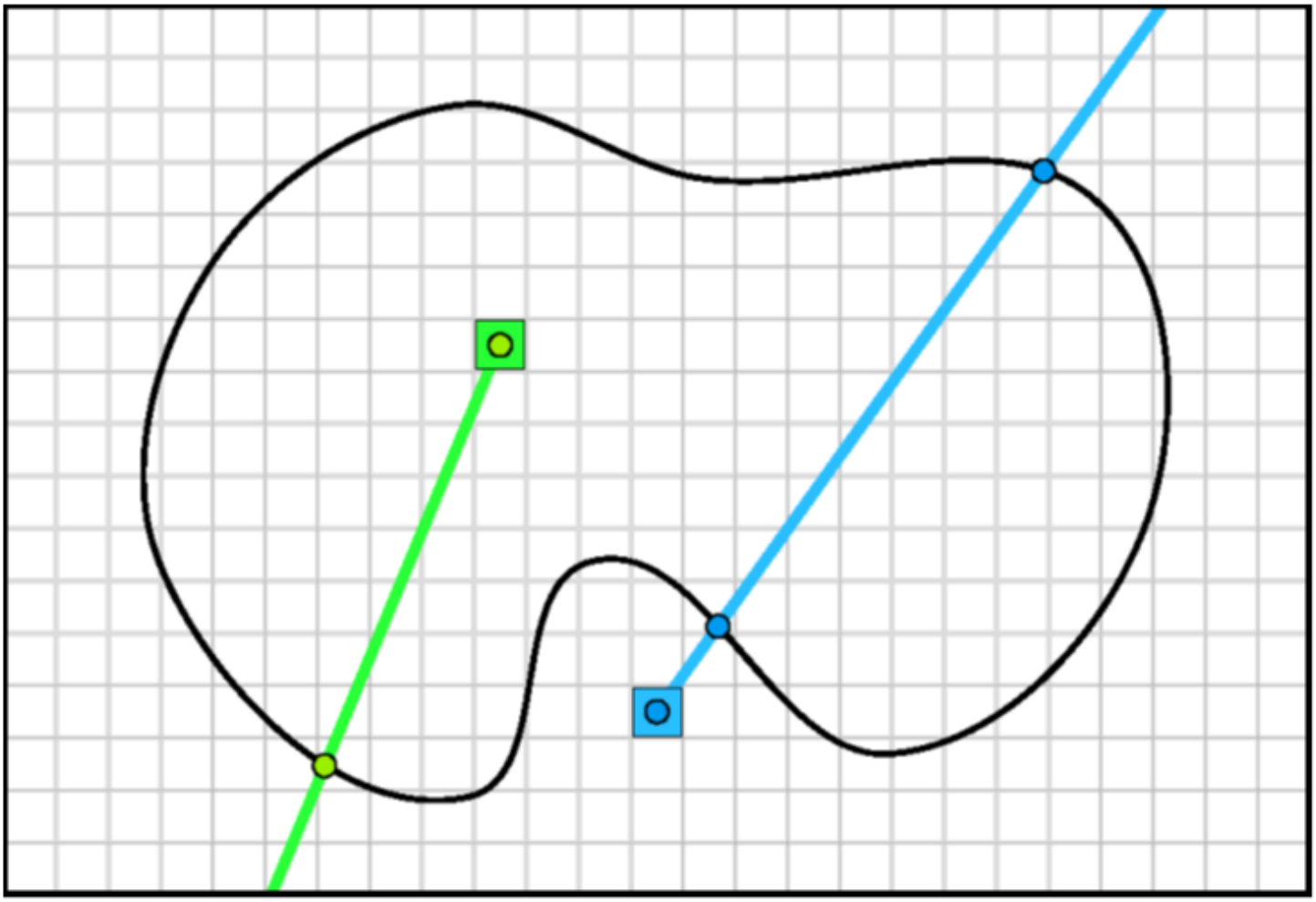


Figure 1

Illustration of the Ray Tracing approach used in the current work to voxelise a 3D mesh into a raster representation of the object. The green ray intersects with the exterior surface of the mesh once (and odd number of intersections) and therefore is classified as inside the mesh. The blue ray intersects twice (even) and therefore is classified as outside the mesh.

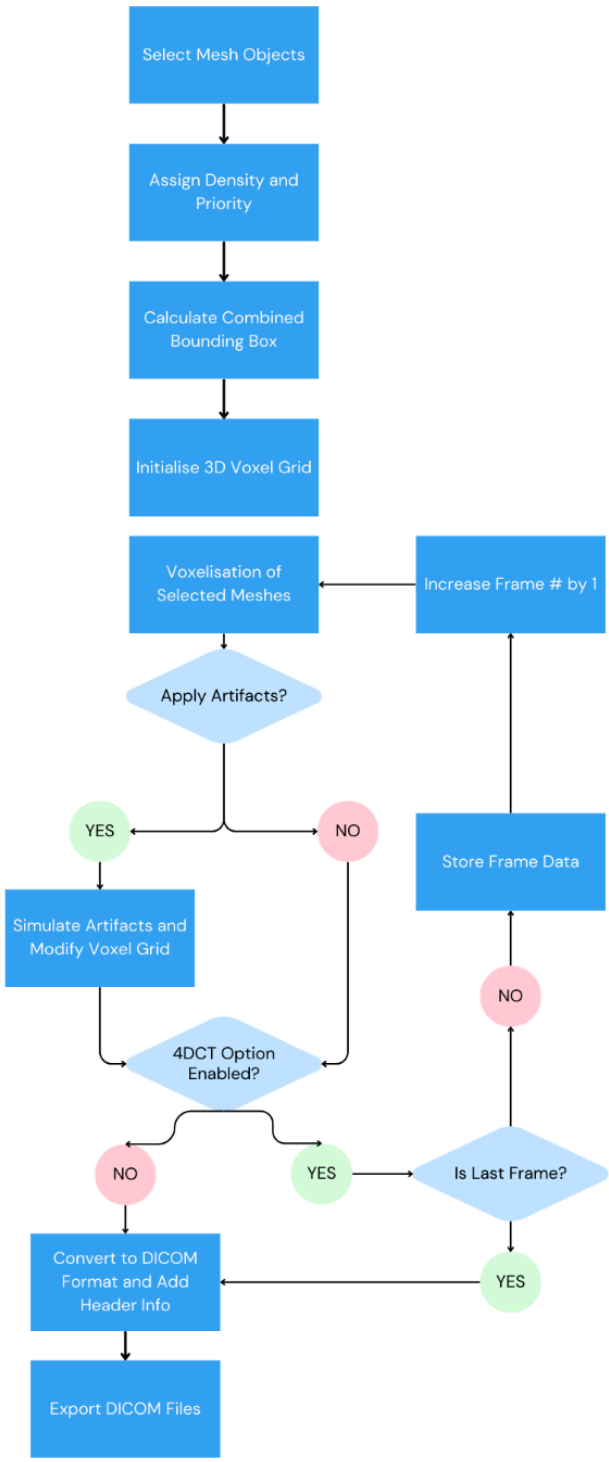


Figure 2

The workflow of the code to convert a collection of 3D mesh objects in Blender into a voxelised DICOM CT series.

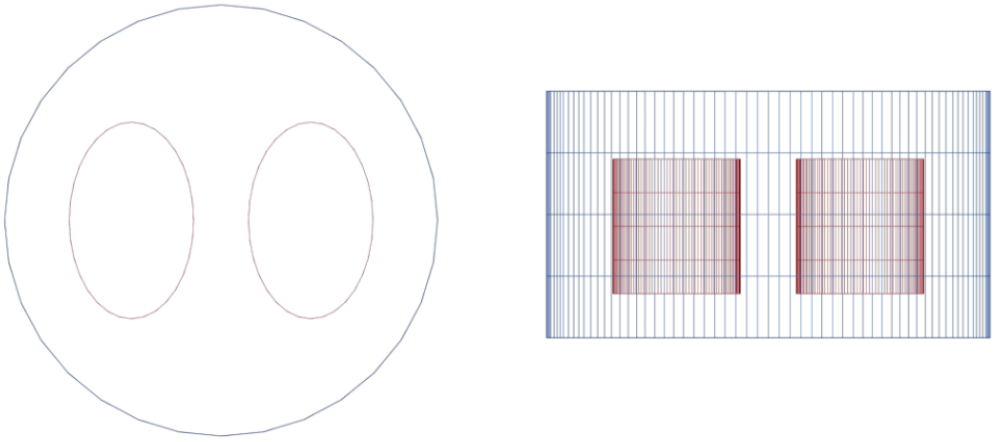


Figure 3

Visualisation of the simple lung phantom geometry. Left: axial view showing the external body contour in blue and lung volumes in red and Right: Coronal view.

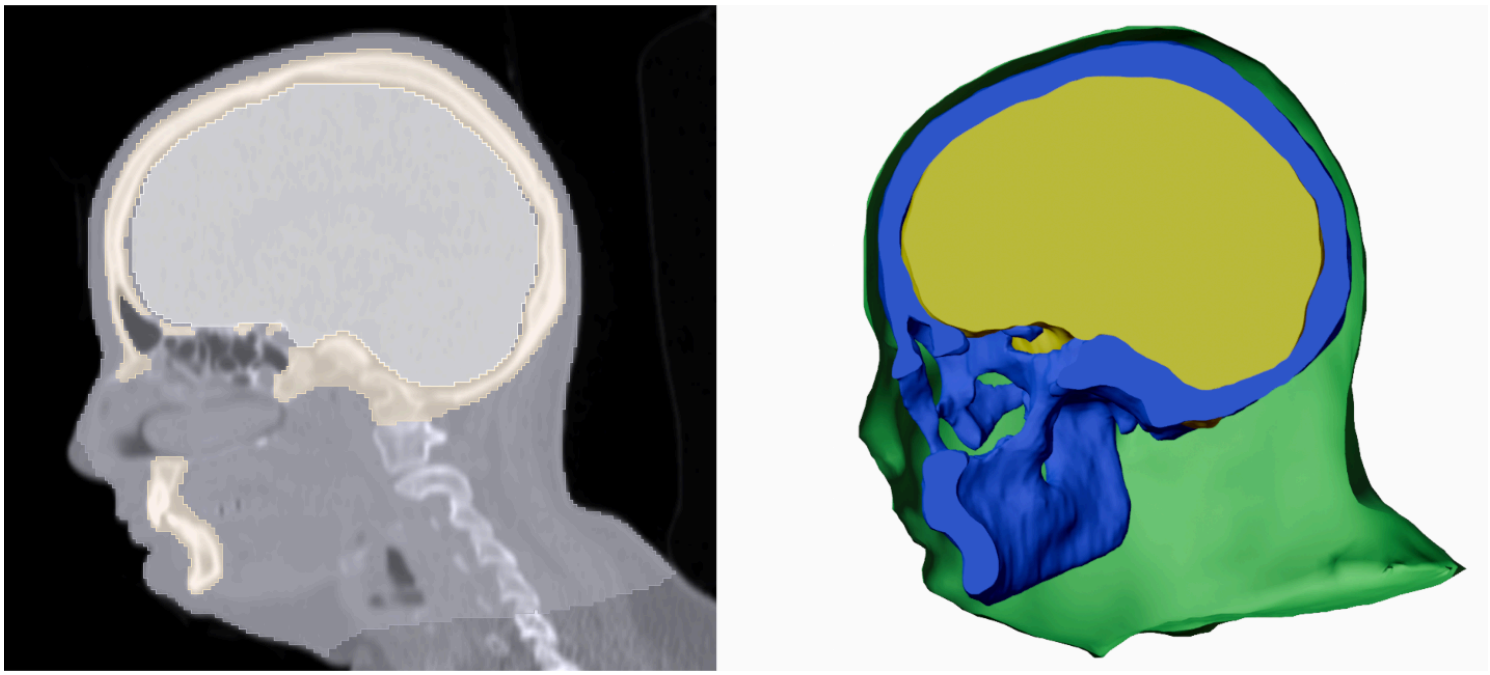


Figure 4

Left: The external, skull and brain segmentations produced by Total Segmentator overlaid with the original CT image. Right: The 3D mesh volumes of the external (green), skull (blue) and brain (yellow) meshes in Blender.

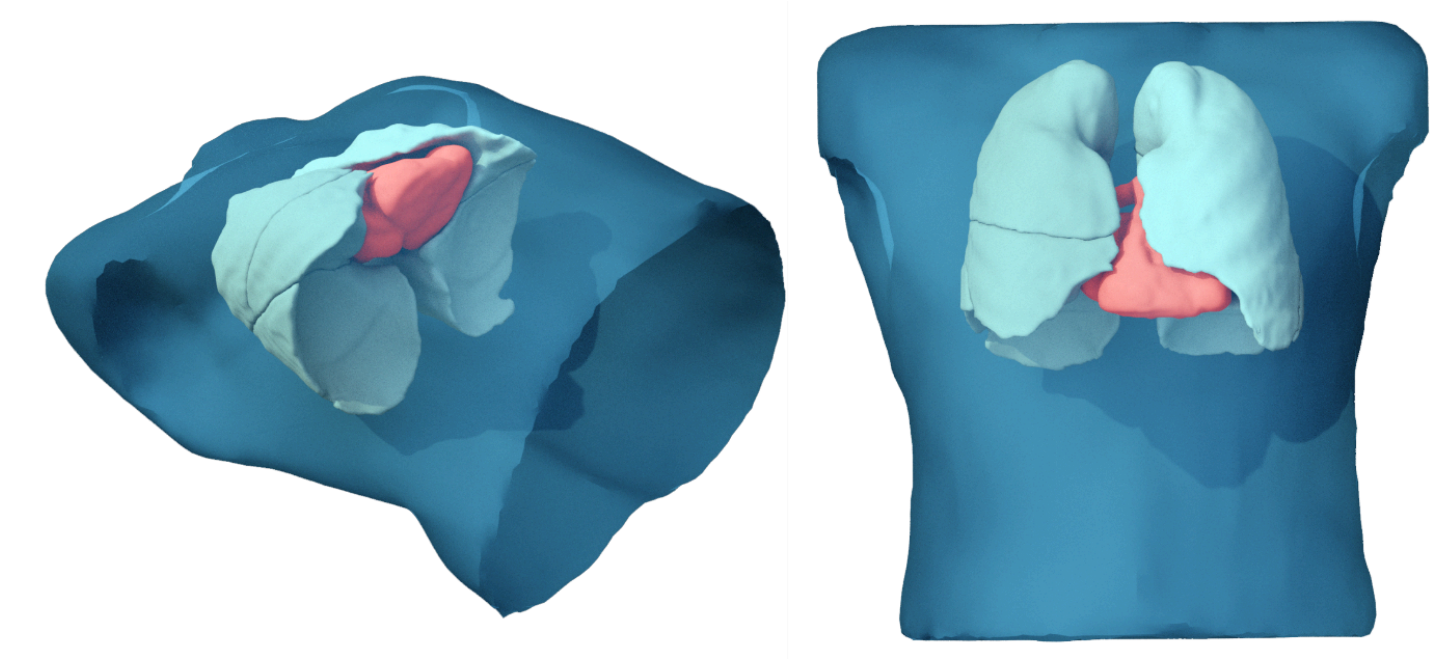


Figure 5

A visualisation of the 3D geometry of the 4D lung geometry in Blender. Dark Blue: External, Light Blue: Lungs, Red: Heart.

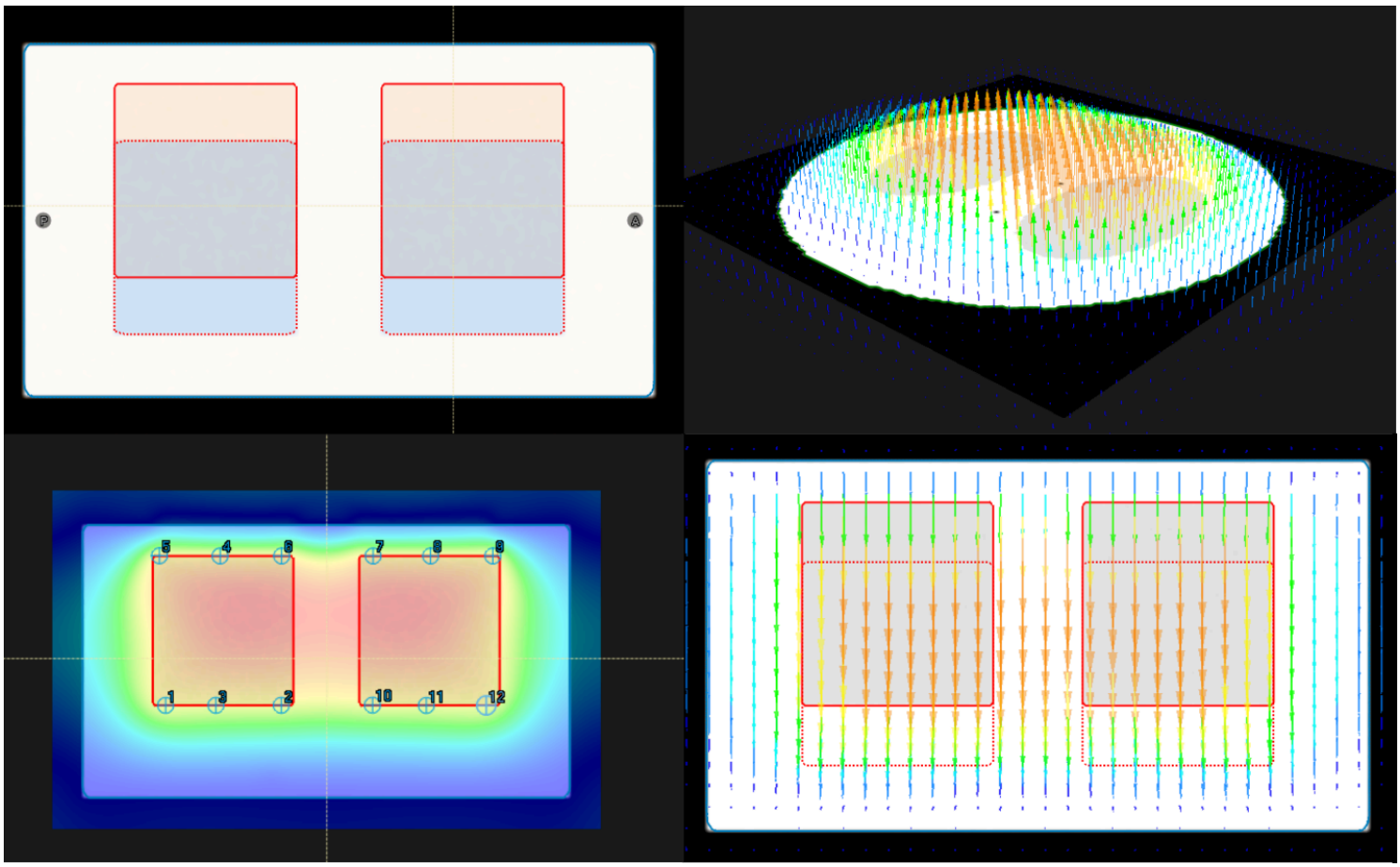


Figure 6

Demonstration of the proposed workflow for validating deformable image registration algorithms. Top Left: The difference map between the first and second lung phantom CT illustrating the simulated lung motion in the superior-inferior direction. Top Right: An axial slice of the lung phantom with a 3D vector displacement map corresponding to that slice. Bottom Left: The displacement intensity map of the lung CT. 12 points are shown which were used to calculate the magnitude of the displacement. Bottom Right: The original (solid red) and deformed (dotted red) contours of the lung volumes with vector displacement field overlaid.

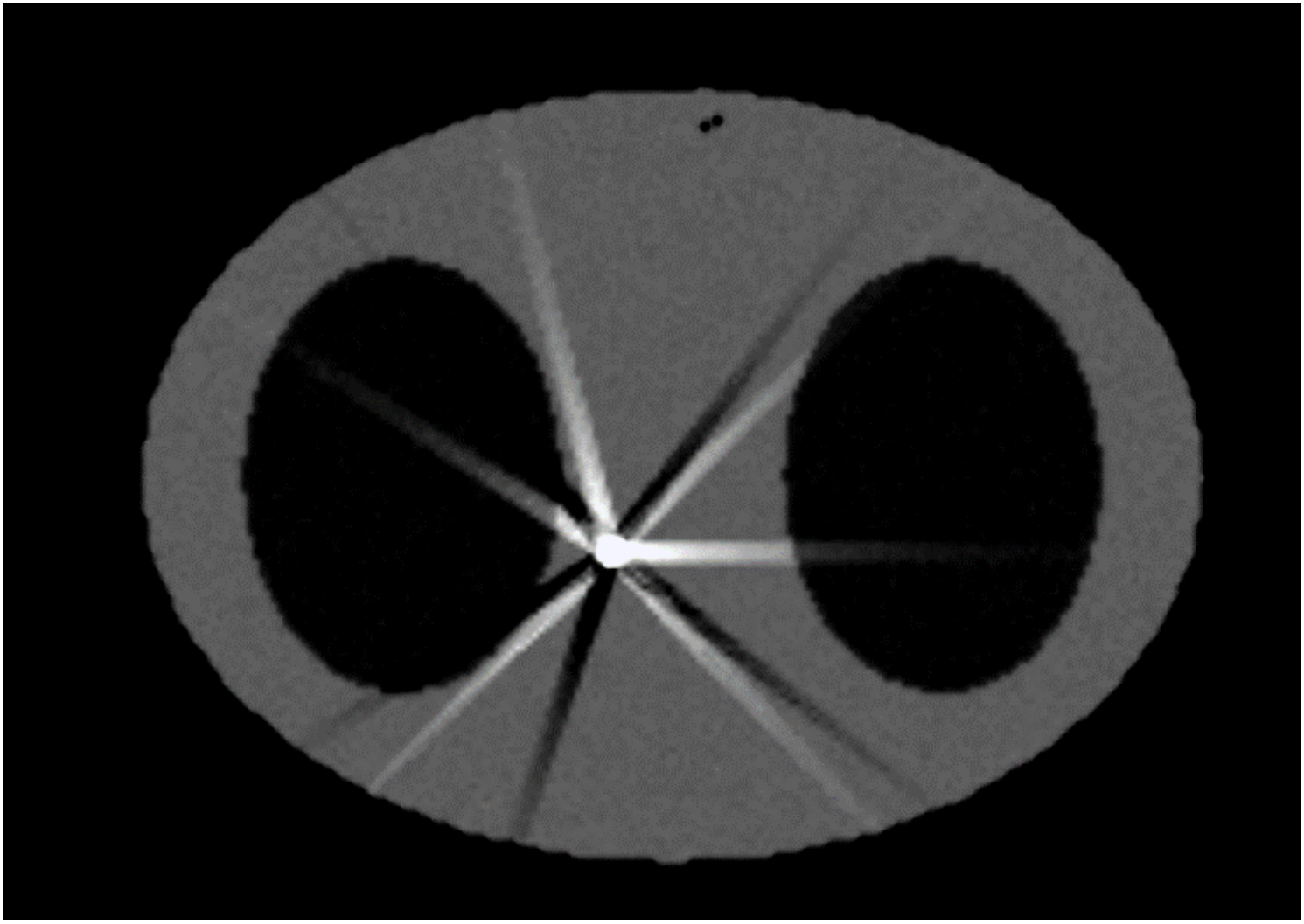


Figure 7

A visual demonstration of the metal artefact simulator implemented in the current work. The 2000 HU spherical volume between the lung volumes is creating a simulated streaking artefact.

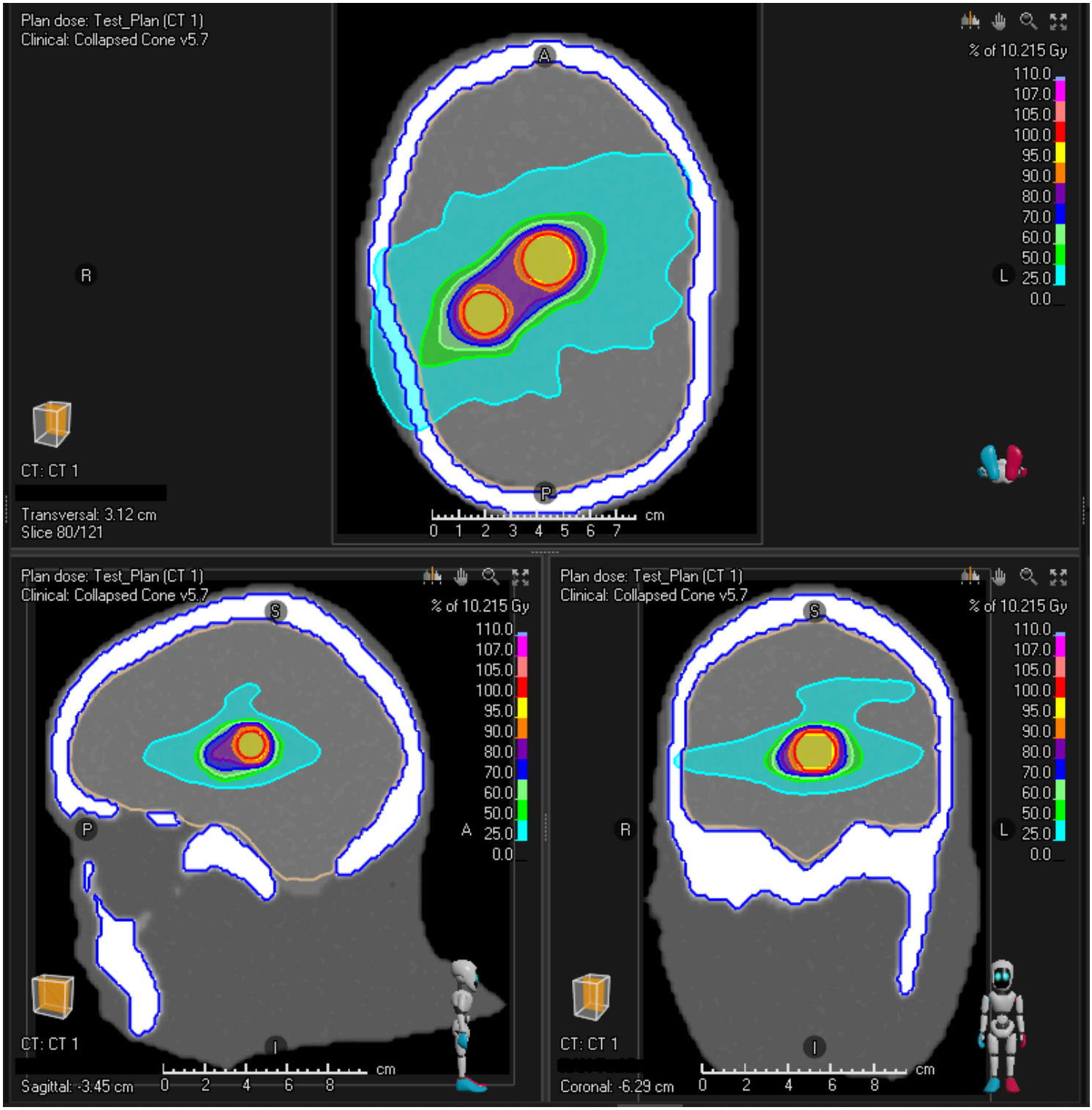


Figure 8

Visualisation of the calculated dose from Raystation TPS on the synthetic cranial CT image for a simulated single isocentre, three metastasis VMAT treatment.

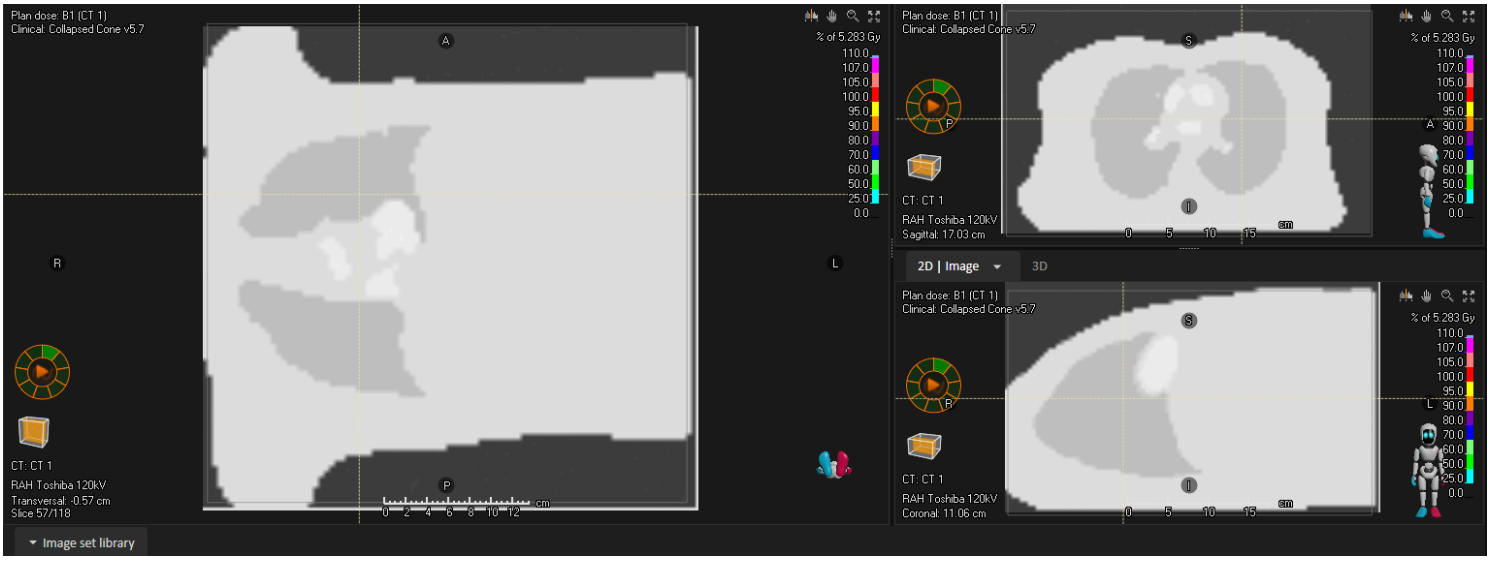


Figure 9

The 4D-CT lung dataset in the Raystation treatment planning system.