



Final Report

CT-Based Skeletal Reconstruction

A Robust and Reproducible Pipeline for Visualizing the Mass-Grave Multi-Session CT Scan Data

Monday 22nd December, 2025 12:19

Students:

Maarten Stork - 15761770

Benjamin van Altena - 13385224

Lecturer:

Dr. Rob Bellemans

Course:

Scientific Visualization and Virtual Reality

Course code:

5284SVVR6Y

Abstract

This work presents an end-to-end pipeline for reconstructing and visualizing skeletal remains from complex, multi-session CT scans of donor bodies for mass-grave research. Raw DICOM data is processed through automated slice sorting, overlap removal, alignment, and Hounsfield Unit-based reconstruction, with optional integration of precomputed anatomical segmentations. The final result is an interactive viewer using GPU-accelerated 3D volume rendering, supported by orthogonal slice views and maximum intensity projections for verification. Results show that the pipeline produces coherent, anatomically meaningful 3D reconstructions with minimal manual intervention, supporting forensic and archaeological documentation of heterogeneous CT datasets.

1 Introduction

Across the globe, numerous armed conflicts are ongoing, resulting in countless deaths and widespread human rights violations. Media outlets have reported accounts of mass fatalities: NBC News describes large groups of bodies piled together in bloodied masses in Sudan [7], soon to be disposed of in unmarked mass graves, while CNN reports that the Israeli military has repeatedly buried the bodies of Palestinians in shallow or unmarked mass graves across Gaza [11]. These mass graves are direct consequences of severe human rights violations, conflict, and social and political turmoil. When considered alongside similar incidents throughout history, they contribute to millions of individuals buried in unmarked graves worldwide [6].

This reality underscores an urgent humanitarian need for forensic investigation into mass graves: to identify the deceased, document what happened, and provide answers and a sense of closure to grieving families. However, the excavation of mass graves is inherently destructive and irreversible, requiring meticulous documentation by highly trained specialists to meet forensic and judicial standards [6]. As Dr. Hayley Mickleburgh notes, “You can only excavate

once” [13], highlighting the importance of getting the process right. At the same time, mass graves present major challenges for archaeological and forensic interpretation. The remains of multiple individuals are often commingled and disarticulated, with complex taphonomic processes contributing to internal grave collapse and subsequent mixing of skeletal elements [6]. These factors complicate individual identification and the reconstruction of events surrounding death. Research in this field is additionally constrained by limited resources, political pressure, and the involvement of inadequately trained first responders, which can result in irreparable damage to burial contexts and the permanent loss of critical information [6]. Consequently, further research into mass grave formation processes, human decomposition, and excavation methodologies is essential for improving the reliability, accuracy, and legal admissibility of forensic and humanitarian investigations.

Dr. Hayley Mickleburgh is an archaeologist specializing in forensic archaeology, mass-grave research, and 3D visualization. As Director of the 4D Research Lab at the University of Amsterdam, she applies 3D digital techniques to the study of complex human remains, with a focus on burial archaeology, forensic taphonomy, and ethical considerations [5]. Directly relevant to this project, Dr. Mickleburgh leads the *Mass Grave Project*, which investigates graves containing multiple donor bodies with the goal of improving the detection, documentation, and analysis of forensic mass graves. Her research combines human decomposition experiments with 3D visualization and imaging techniques to better understand taphonomic processes in degraded bodies in mass-grave contexts [5]. To support this work, digital methods such as CT imaging, photogrammetry, laser scanning, and 3D computer graphics are used to document and analyze skeletal remains.

Within this broader research framework, this work focuses on the development of an end-to-end pipeline for digital documentation and analysis of the pre-burial mass-grave CT dataset. While the overarching initiative aims to better understand decomposition and taphonomic processes through advanced imaging and visualization, our project addresses a specific computational challenge: automated segmentation and visualization of skeletal remains in degraded, heterogeneous full-body CT scans. This paper investigates automated segmentation strategies to separate bone tissue from surrounding soft tissue and imaging artifacts, and integrate these into a visualization workflow designed to support forensic and archaeological analysis. The central research question guiding this work is *whether a robust, largely automated pipeline can be developed to transform raw, multi-session DICOM data into coherent, anatomically meaningful 3D reconstructions for forensic exploration and documentation*. This task is expected to be challenging due to the characteristics of the mass-grave CT dataset, including overlapping scan sessions, variable acquisition parameters, incomplete or degraded anatomical structures, and limitations of existing pre-trained segmentation models. To address this question, an end-to-end workflow is developed and evaluated that reconstructs raw CT data, resolves inter-scan overlap, applies automated anatomical segmentation, and supports interactive 3D visualization. The complete repository (containing this pipeline) is available on GitHub¹.

2 Methods

The goal of this project was to build a practical pipeline that turns raw DICOM CT data into an interactive 3D reconstruction with clear, bone-focused visualization and hoverable anatomical labels. Beyond producing a single good-looking render, the emphasis was on robustness and repeatability: the workflow should run across the full donor dataset with minimal manual intervention and still produce correct, usable visuals at scale. The following sections describe how the data are loaded, reconstructed, optionally segmented, and rendered in an interactive viewer, next to some other visualization options. The order of subsections follows the dashed boxes in the flowchart in Figure 1.

¹See the full codebase here: https://github.com/MaartenStork/SVVR_Visual_MassGraves

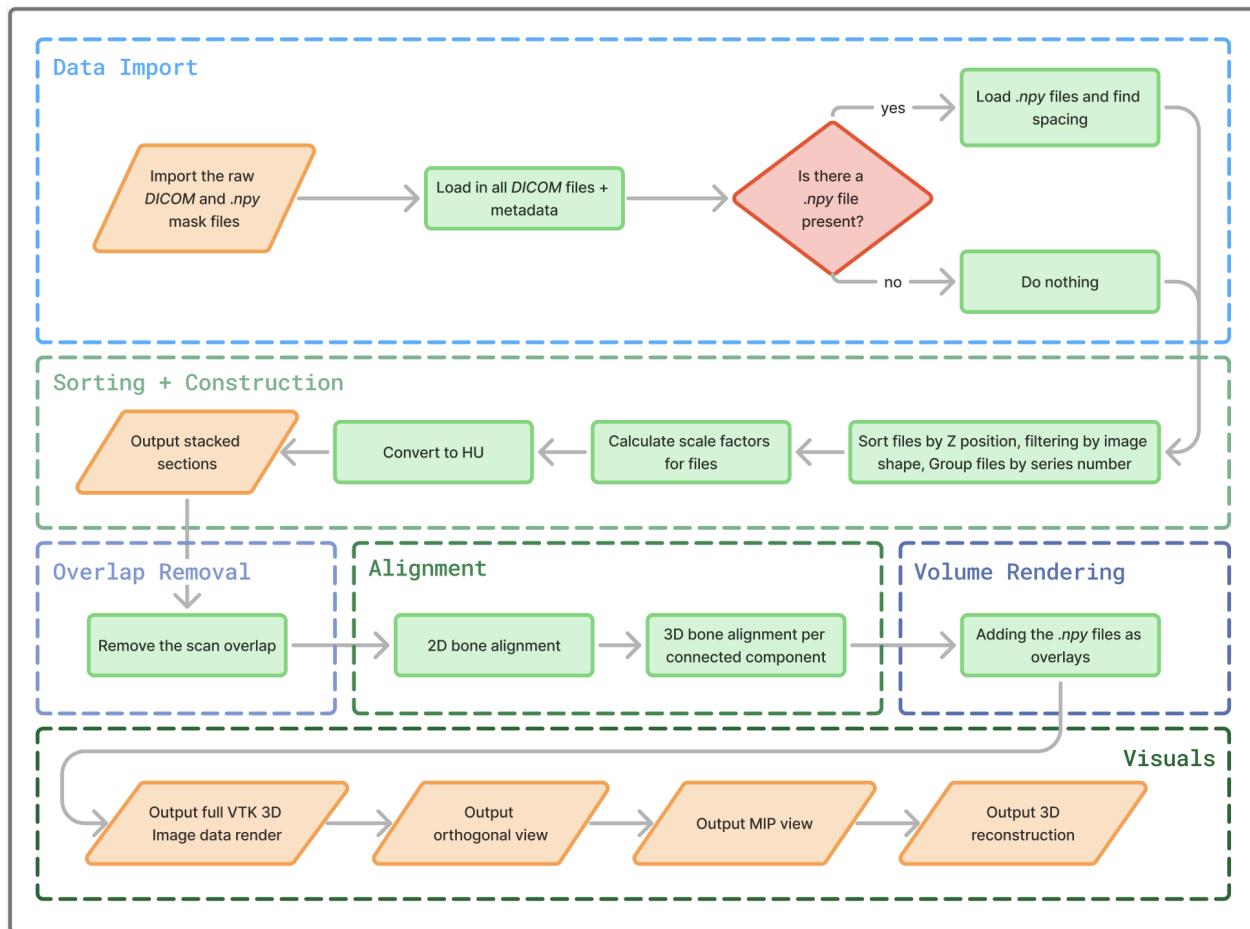


Figure 1: Flowchart showing the full pipeline triggered by running `run_viewer.py` in the project repository.

2.1 Data Import

Running `run_viewer.py` launches the interactive viewer. On startup, the user selects one of six donor datasets, after which the program loads the corresponding DICOM series. If precomputed segmentation files are present (`.npy` masks with JSON headers), these are loaded as well; otherwise, the viewer runs in CT-only mode.

DICOM data The mass grave dataset consists of six donor CT scans stored in DICOM (Digital Imaging and Communications in Medicine). In DICOM, each 2D slice is stored as a separate file containing both pixel data and metadata [1] (see Table 1 in the Appendix for an overview of the metadata in the donor dataset). For each donor, multiple CT series are available. Because full-body imaging required multiple scan sessions, the upper body (head to pelvis) and lower body (pelvis to feet) were typically acquired separately. This produces an overlap, which must be detected and removed during reconstruction to avoid duplicated anatomy.

Scan parameters vary between series and donors, which prevents straightforward concatenation. Differences include pixel spacing (approximately 0.74 to 0.93 mm in plane), slice thickness (typically around 5.0 mm), field of view (affecting the captured area and matrix size), and the presence of scout or localizer images with different dimensions that must be filtered out. Before visualization, the series are therefore resampled to a common resolution, spatially aligned, and merged. More on this in the next subchapter.

MONAI segmentation To actually make the interactive exploration of the anatomical structures possible, we need a way to identify which parts of the scan correspond to which bones or organs. We therefore generate segmentation masks as a preprocessing step. In practice, this converts the raw DICOM series into a separate labeled volume that the viewer can load alongside the CT data to enable anatomy-aware interaction; `run_viewer.py` will automatically use these masks when they are present.

Starting from the raw DICOM series, a pre-trained deep learning model needs to produce a labeled mask in which each voxel is assigned an integer corresponding to an anatomical structure (e.g., ribs, vertebrae, organs). The resulting label volumes are exported as NumPy arrays (`.npy`), while spatial metadata required for correct alignment is stored in a corresponding `*_header.json`. In addition, a labels file maps integer IDs to human-readable anatomical names and provides default visualization colors. When the viewer starts, it loads the `.npy` segmentation mask into memory and parses the accompanying header file. The voxel spacing is extracted from the diagonal of the `space directions` matrix as

$$\text{spacing} = [|d_{00}|, |d_{11}|, |d_{22}|] \quad (1)$$

and stored alongside the `space origin` coordinates. These spatial parameters are retained for use during the rendering stage, where the segmentation is transformed and aligned with the CT volume, further discussed in subchapter *Volume Rendering*.

For this research we implemented the segmentation step using MONAI (Medical Open Network for Artificial Intelligence), an open-source, PyTorch-based medical imaging toolkit that provides pre-trained models for tasks such as whole-body CT segmentation [9]. The same MONAI-based tooling is also accessible through visualization platforms such as 3D Slicer, which makes it easier to reproduce or adapt this step.

TotalSegmentator (nnU-Net based) was also considered due to its strong out-of-the-box performance and broad anatomical coverage [14], but we did not integrate it in this project because it does not natively support acceleration on Apple Silicon GPUs, and CPU-only inference for within this timeframe. Manual segmentation in 3D Slicer was similarly not feasible for six full-body scans due to those same time requirements. Important to know is that because the user can load these files in as a preprocessing step, our viewer is agnostic to how segmentations are produced: any user can generate segmentations using alternative tools and simply swap in their own `.npy` masks to use the same visualization workflow. We therefore also encourage readers to consider TotalSegmentator, which is widely often mentioned as one of the strongest free options for whole-body CT segmentation [14].

In addition to enabling anatomy-aware interaction, the segmentation step was essential for handling severe vascular calcifications present in some donors. Since calcified regions in the aorta and iliac arteries exhibit bone-level Hounsfield Unit (HU) values, they would otherwise be misclassified as skeletal tissue during HU-based visualization. By explicitly segmenting these vessels in preprocessing, they can be identified and highlighted separately, preventing confounding artifacts in the bone-focused reconstruction.

2.2 Sorting and Construction

After loading the individual DICOM slices, the code is set to reconstruct it into one coherent image. In practice this requires a lot more than simply stacking images: slices must be ordered in physical space, non-volumetric images (scouts/localizers) must be removed, multiple scan sessions may need to be combined, resolution differences must be normalized, and all data must be placed into a single aligned grid.

Slice ordering DICOM files are not guaranteed to be stored in any spatial order, so slices are sorted using the DICOM metadata. The preferred key is the Z-coordinate from `ImagePositionPatient`, which specifies the world-space

location of the first voxel of a slice in millimeters. If `ImagePositionPatient` is unavailable, the pipeline falls back to `SliceLocation`. As a last resort it uses `InstanceNumber`, which reflects acquisition sequence but is not always a reliable tell for spatial position (as some protocols could acquire slices in non-sequential order, and the scan direction (head-first versus feet-first) could vary between studies). Slices are sorted in descending Z so that the reconstructed volume follows a head-to-toe orientation.

Filtering images Some CT studies include scout or localizer images used for scan planning [4]. In our dataset these seem to be present, and they have different image dimensions than the main CT slice stack, so we filter them out before reconstruction. The code does this by looking at the image dimensions across all slices, identifying the most common shape as the primary series, and not using the slices with non-matching dimensions for the visuals here.

Handling multiple acquisitions As noted earlier, several donors required multiple scan sessions to achieve full-body coverage. In practice, the DICOM folders therefore contain multiple series corresponding to different scan runs, most commonly an upper-body scan (head to pelvis) and a lower-body scan (pelvis to feet). We first group slices by their `SeriesNumber` and estimate each series' coverage along the longitudinal axis by taking the minimum and maximum Z-coordinates (using `ImagePositionPatient` [2] when available). Very small series (fewer than five slices) are ignored, since these are typically scouts or incomplete fragments and do not form a usable volume for visualization.

After this, the pipeline compares the Z-ranges of the remaining series to decide whether they represent duplicates or different body regions. Sometimes multiple series cover the same region because the scanner outputs alternative reconstructions of the same scan (for example, with different reconstruction settings). These series overlap strongly in Z and show the same anatomy, so we keep only the series with the most slices. In other cases, series cover different regions with little overlap, such as an upper-body scan and a separate lower-body scan. Those series are candidates for concatenation into a single full-body volume. We only concatenate when it clearly increases anatomical coverage compared to using the largest series alone, using

$$\text{concatenate if } R_{\text{total}} > 1.2 \times R_{\text{largest}}, \quad (2)$$

where R_{total} is the combined Z-range of the selected series and R_{largest} is the Z-range of the single largest series. This 20% margin avoids merging series that mostly duplicate each other, while still allowing concatenation when it meaningfully extends coverage.

Resolution normalization and alignment Different series can have different in-plane resolutions (pixel spacings). If stacked directly, the resulting volume would be geometrically inconsistent as the same physical distance would correspond to different numbers of pixels in different regions. We therefore resample all selected series to a common resolution before concatenation.

The smallest pixel spacing (highest resolution) is chosen as the target. Any series with a coarser spacing is upsampled using the scale factor

$$f_{\text{scale}} = \frac{s_{\text{current}}}{s_{\text{target}}}, \quad (3)$$

where s_{current} and s_{target} are the pixel spacings in mm/pixel. Resampling is performed with `scipy.ndimage.zoom()` using bilinear interpolation, which estimates intermediate pixel values from their neighbors.

Importantly, this resampling does not create anatomical detail that was not present in the original scans. Interpolation only estimates intermediate values between existing pixels, so structures smaller than the native resolution cannot suddenly appear. In our dataset the spacing differences are modest (typically around 0.74 versus 0.93 mm/pixel), so the

upsampled series remains visually consistent while the main benefit is improved geometric alignment: bones and other structures line up correctly across series boundaries, allowing for accurate visualization and measurement.

Hounsfield Unit conversion Raw DICOM pixel values are stored in scanner-dependent units that can vary with the scanner, protocol, and reconstruction settings. As a result, the same tissue can correspond to different numbers across scans, so consistent thresholds and transfer functions cannot be applied directly to the raw values.

To resolve this, pixel values are converted to HU values, a standardized radiodensity scale used in CT imaging [3]. The conversion applies a simple linear rescaling using parameters stored in the DICOM metadata:

$$\text{HU} = V_{\text{pixel}} \times m + b, \quad (4)$$

where m is `RescaleSlope` and b is `RescaleIntercept`. After this transformation, HU values have a consistent interpretation across scans (Table 1), which allows us to choose thresholds and transfer functions that generalize across donors.

Data concatenation After sorting, filtering, resampling, and HU conversion, the pipeline assembles a single coherent image from all processed slices. This is done in two passes. First, each slice is prepared (HU conversion and optional resampling) while tracking the required in-plane offsets, which determines the canvas size needed to fit all series. Second, a canvas is created for each slice position, initialized to -1000 HU (air) (see Figure 5), and the slice is inserted at its computed offset. The resulting aligned slices are then stacked along the Z-axis using `numpy.stack()`, producing a final voxel array of shape (N_z, N_y, N_x) , where N_z is the number of slices and $N_y \times N_x$ are the in-plane canvas dimensions.

Finally, we store the voxel spacing of the reconstructed volume for correct rendering and measurement. The inter-slice spacing Δz is taken from `SliceThickness` when available, or estimated as the median distance between consecutive slice positions. The in-plane spacings Δy and Δx are taken from `PixelSpacing[0]` and `PixelSpacing[1]`.

2.3 Overlap Removal

As noted, when a body is scanned in multiple sessions, the resulting series usually overlap near their boundary. After simply concatenating the volumes (made possible through the steps described in previous subchapters), the shared region would appear twice, introducing duplicated anatomy and causing issues with possible downstream measurements (see Figure 2b). The pipeline therefore includes an automatic stage that detects the overlap and removes the redundant slices.

Finding the optimal cut point The correct cut location cannot be determined from Z-coordinates alone. Between scan sessions the body may shift on the table, so slices covering the same anatomy do not always share identical world coordinates. We therefore use image content to identify where the two series depict matching anatomy.

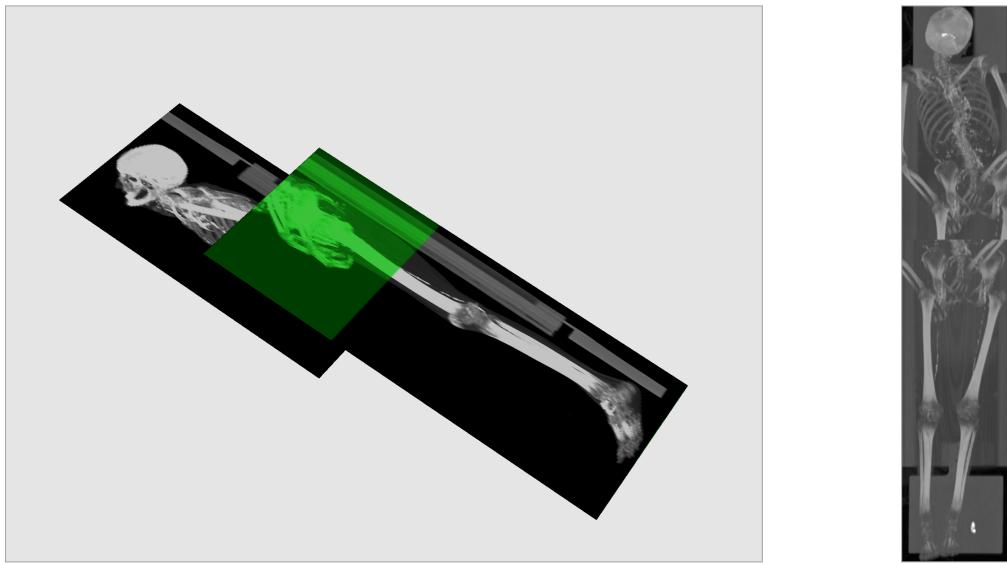
The algorithm searches along the Z-axis to find the slice in the lower-body series that best matches the anatomy at the end of the upper-body series. First, the boundary between acquisitions is located by finding where `SeriesNumber` changes in the concatenated slice list. From that point, the search extends through approximately 90% of the lower-body series to accommodate potentially substantial overlap. To keep computation manageable, the search proceeds in two stages: a coarse pass tests every fifth slice to find an approximate match, followed by a fine pass that checks every slice within a narrow window (± 6 slices) around the coarse result.

At each candidate location, a block of slices from the upper series is compared against a corresponding block from the lower series. The candidate with the strongest anatomical agreement is selected as the transition point.

Similarity metric Similarity is computed using bone structure as a stable landmark. As the reconstructed DICOM data does not contain per-voxel anatomical labels, bone matter cannot be identified directly from the image data. We therefore apply fixed HU-thresholds to robustly separate high-density bone from soft tissue, based on values explained in 2.5. We first create a simple bone mask by thresholding the CT volume at 300 HU, producing a binary volume that highlights high-density skeletal elements. For each candidate cut location, we compare two small slice blocks (one from the end of the upper series and one from the lower series) by computing the Intersection over Union (IoU) [10] of their bone masks:

$$\text{IoU} = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}, \quad (5)$$

where B_1 and B_2 are the binary bone masks of the two blocks. An IoU of 1 indicates perfect agreement, while 0 indicates no overlap. Importantly, at this stage we use IoU to detect whether two blocks correspond to the same anatomical level (e.g., pelvis versus mid-thigh), not to achieve perfect pixel-level alignment. Small lateral shifts between sessions can reduce exact pixel overlap, but the characteristic bone patterns in matching cross-sections still tend to produce a higher IoU than mismatched levels. Precise in-plane alignment is handled later; here we only need a reliable cut point. The cut location with the highest IoU is selected as the transition between series. Figure 2 shows visually how this cutoff point is selected by identifying the overlapping region between the two scans, and what happens if concatenate without doing this.



(a) *IoU-based overlap detection with duplicated slices removed.*

(b) *Scans concatenated without overlap removal.*

Figure 2: *Sagittal MIP projections for donor 003. Comparison between IoU-based overlap removal and unprocessed scan concatenation.*

Removing the overlap Once the optimal cut index has been found, we remove the overlapping part by discarding the first cut slices of the lower-body series (i.e., the slices that duplicate anatomy already present at the end of the upper-body series). We then append the remaining lower-body slices to the full upper-body volume. This yields a continuous full-body reconstruction while avoiding duplicated anatomy in the overlap region

2.4 Alignment

After overlap removal, the upper- and lower-body scans can be joined into a single volume, but a visible seam may still remain at the junction between the two. Considerable shifts on the scanner table between sessions cause the lower volume to appear offset relative to the upper volume. To reduce these artifacts, the pipeline applies two alignment steps: first a global in-plane shift that aligns the entire lower-body scan, and then an optional per-leg (per-connected-component) correction to handle cases where the left and right legs are offset differently. Technically this per-leg alignment is only done once it is available in 3D imaging, but for the sake of a more clear pipeline, we describe it here.

Global 2D alignment The first stage corrects the overall shift between sessions using a rigid 2D translation in the image plane. We compare the last slice of the upper-body volume with the first slice of the lower-body volume, which should depict similar anatomy around the femurs. Both slices are thresholded at 200 HU to create binary bone masks, since bone provides strong landmarks and is also the primary focus of our visualizations.

We then perform a brute-force search over integer pixel shifts (dx, dy) within a range of ± 80 pixels. For each candidate shift, the lower bone mask block is translated and compared with the upper bone mask block. The alignment score is simply the number of overlapping bone pixels:

$$\text{score}(dx, dy) = \sum_{y, x} (M_{\text{lower}}[y + dy, x + dx] \wedge M_{\text{upper}}[y, x]), \quad (6)$$

where M_{upper} and M_{lower} are the binary masks and \wedge denotes a logical AND. The shift that maximizes this score is selected and then applied to the entire lower-body volume using `numpy.roll()`, which reduces the visible seam so the pelvis and upper legs connect more smoothly across the boundary. This improved smoothness is clearly visible in Figure 3 in the "AFTER" image.

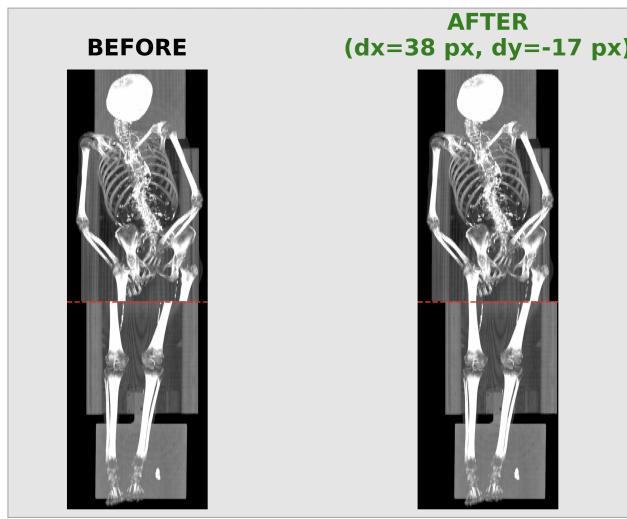


Figure 3: Visual representation of 2D alignment for donor 003. Left: upper and lower volumes before alignment, with a visible seam at the session boundary (red dashed line). Right: after the optimal shift ($dx = 38$ px, $dy = -17$ px) maximizing bone overlap, the transition is smoother.

3D per-leg alignment In our dataset, a single global shift was often not sufficient because the legs move differently between scan sessions. Since the legs are not rigidly fixed relative to the pelvis, one leg may rotate or shift more than

the other. A global translation effectively finds an average best fit, which can still leave one leg visibly misaligned at the junction.

To address this, we include an optional per-leg correction that is applied at render time. Near the junction, bone is detected using a 300 HU threshold and separated into connected components. The two dominant components (left and right femurs) are identified, and their centroids are computed in a slice just above and just below the junction. For each leg, the centroid offset is converted to a physical distance using the voxel spacing,

$$d_{\text{leg}} = \sqrt{(\Delta y \cdot s_y)^2 + (\Delta x \cdot s_x)^2}, \quad (7)$$

where Δy and Δx are centroid differences in pixels, and s_y and s_x are the in-plane pixel spacings in mm/pixel. If the misalignment exceeds a small threshold (default 3 mm), we construct a 3D mask for the affected leg in the lower volume (the femur component with a small dilation to include nearby soft tissue) and shift only that masked region by the required offset. Legs that already align well are left unchanged. Figure 4 visually illustrates how the program detects the two connected components.

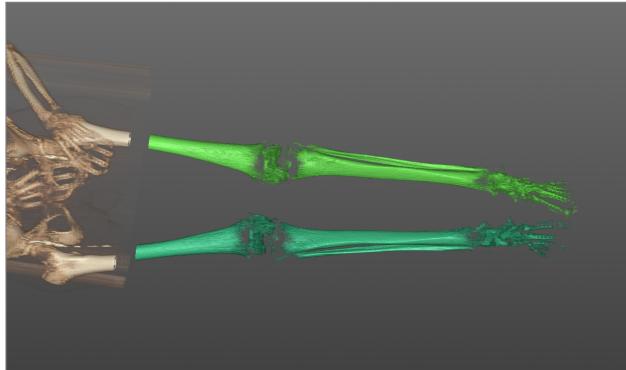


Figure 4: *Visual representation of per-leg bone segmentation for donor 003, showing the two dominant connected components (left and right femur regions). The components are displayed in two shades of green to illustrate the connected-component analysis.*

2.5 Volume Rendering

After reconstruction and the global 2D alignment step (and before the 3D per-leg correction), the CT volume is visualized using the Visualization Toolkit (VTK), an open-source library for 3D graphics and scientific visualization [12]. This is a good fit for our application because VTK provides GPU-accelerated volume rendering, allowing large medical imaging volumes to be explored interactively in real time.

Volume data preparation The reconstructed NumPy array is converted to VTK’s `vtkImageData` format. In practice, this means flattening the 3D array in C-order (row-major), wrapping it as a VTK array, and attaching it to a `vtkImageData` object with the correct dimensions. We also provide the physical voxel spacing computed during volume construction, so that the rendered volume has correct anatomical proportions.

Transfer functions and shading Volume rendering works by casting rays through the 3D data and accumulating color and opacity along each ray. The visual result is controlled by transfer functions that map HU values to color and transparency. We use a color transfer function that keeps air and low-density regions dark, renders soft tissue in muted

tones, and maps bone to bright near-white values so skeletal structures stand out clearly. Opacity is kept near zero for air and fat, very low for most soft tissue, and high for bone, which keeps the body largely transparent while preserving the skeleton as a solid, readable structure. See the visuals in Figure 5.

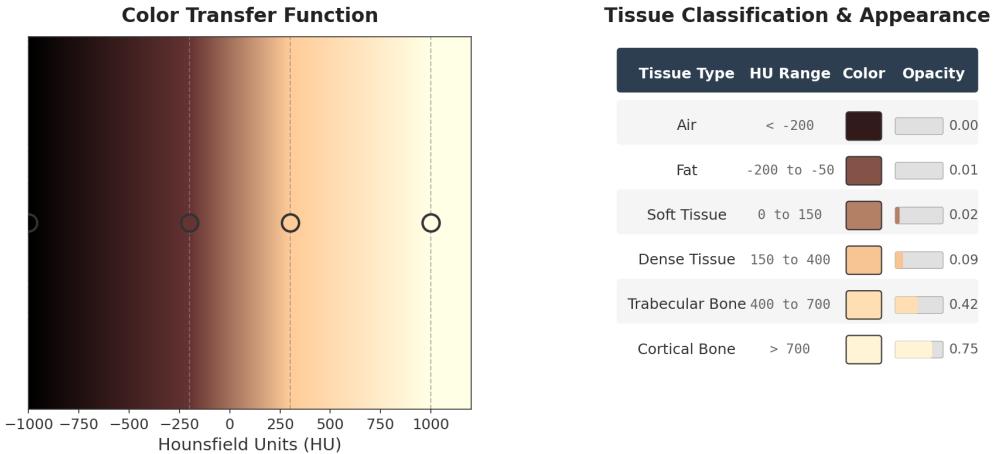


Figure 5: *Volume rendering transfer functions. Left: color mapping from HU values showing the gradient from air (black) to bone (ivory). Right: tissue classification with corresponding HU ranges, colors, and opacity values used in the visualization.*

To improve depth perception, we also apply gradient opacity to emphasize boundaries where intensity changes rapidly, creating an edge-enhancement effect. Finally, Phong shading [2] is enabled to provide consistent lighting cues (highlights and shadows), which helps surfaces read as three-dimensional rather than flat intensity clouds.

Segmentation overlay and interaction If precomputed segmentation masks are available (the .npy files mentioned earlier), they are rendered on top of the CT volume to highlight specific structures. Since these masks are produced by external AI models, they do not always share the exact same axis conventions as the reconstructed DICOM volume. Before rendering, we therefore align each mask to the CT volume using the spatial metadata in its header: this can include transposing axes, flipping the Z-direction to match head-to-toe ordering, and applying an origin offset based on the difference between the segmentation’s world origin and the CT volume origin.

Once aligned, the binary aorta mask is converted to a surface mesh using marching cubes [8] and rendered as an opaque red overlay, making the vessel immediately visible inside the volume. For labeled anatomical segmentations (e.g., bone structures), we generate a separate surface for each label (ribs, vertebrae, femurs, etc.) and render these transparent by default. The viewer supports interactive identification using VTK picking: when the cursor hovers over a bone structure from the mask, that surface from the VTK is highlighted and its anatomical name is shown in an on-screen tooltip, allowing bones to be identified directly in the 3D view without consulting external references.

2.6 Visuals

No single visualization method works best for every task, so the viewer provides a few different modes that can be used side by side during analysis.

3D reconstruction The default mode is the GPU-accelerated volume rendering that was discussed a lot in the previous subsections, which displays the full CT volume as a semi-transparent 3D object. It provides an immediate overview of body posture and major anatomy, and the user can rotate, zoom, and pan to inspect the skeleton from any angle.

The transfer function is tuned so that bone appears bright and relatively opaque, while most soft tissue remains faint, preserving anatomical context without obscuring the skeleton. When segmentation is available, hover interaction highlights individual structures and shows their labels, enabling quick identification during exploration.

Orthogonal view Slice mode shows three perpendicular cross-sections through the volume: axial, sagittal, and coronal. In the viewer, these are displayed as three synchronized 2D panels, and the user can scroll through each axis to move the cut plane through the body. This view provides undistorted 2D images, which are often better for spotting subtle details such as small fractures, loose fragments, or local density changes. It is also useful for confirming what is seen in the 3D rendering: when an unusual feature appears in the volume view, the examiner can navigate to the same location in the slice panels to check whether it reflects real anatomy or a rendering artifact.

Maximum intensity projection view Maximum Intensity Projection (MIP) creates 2D images by taking the maximum HU value along each viewing ray, which naturally emphasizes dense structures such as bone. In the viewer, this appears as radiograph-like images where the skeleton is clearly outlined and most soft tissue is largely suppressed. We provide MIPs along the axial, sagittal, and coronal directions, so the user can switch between front, side, and top-down perspectives derived from the same CT volume. This mode is useful for a rapid skeletal survey, but it can hide structures that lie behind denser anatomy because only the maximum value is displayed.

3 Results

To evaluate the visualization pipeline, results are presented on a single reference case (donor 003). The same processing and rendering steps can be applied across all donors, but this case is used because it includes both a bone segmentation and an aorta segmentation mask, and the aorta is visibly calcified. This makes it a particularly challenging example for the task and therefore a useful focus for evaluation.

3.1 3D Reconstruction View

Figure 6 shows the main 3D VTK visualization of donor 003, with calcified blood vessels explicitly colored. For 3D visualizations of the remaining five donors, see the Appendix, Section *3D Visuals*.

(a) *Frontal view.*(b) *Top down view.*(c) *Side view.*

Figure 6: *3D volume rendering of donor 003 with segmented anatomical structures, with the calcified aorta and iliac arteries colored.*

3.1.1 3D Segmentation View

When a labeled .npy segmentation (e.g., the MONAI output) is available beyond the aorta mask, the viewer enables the interactive behavior shown in Figure 7: hovering over a segmented structure highlights it and displays its anatomical label.

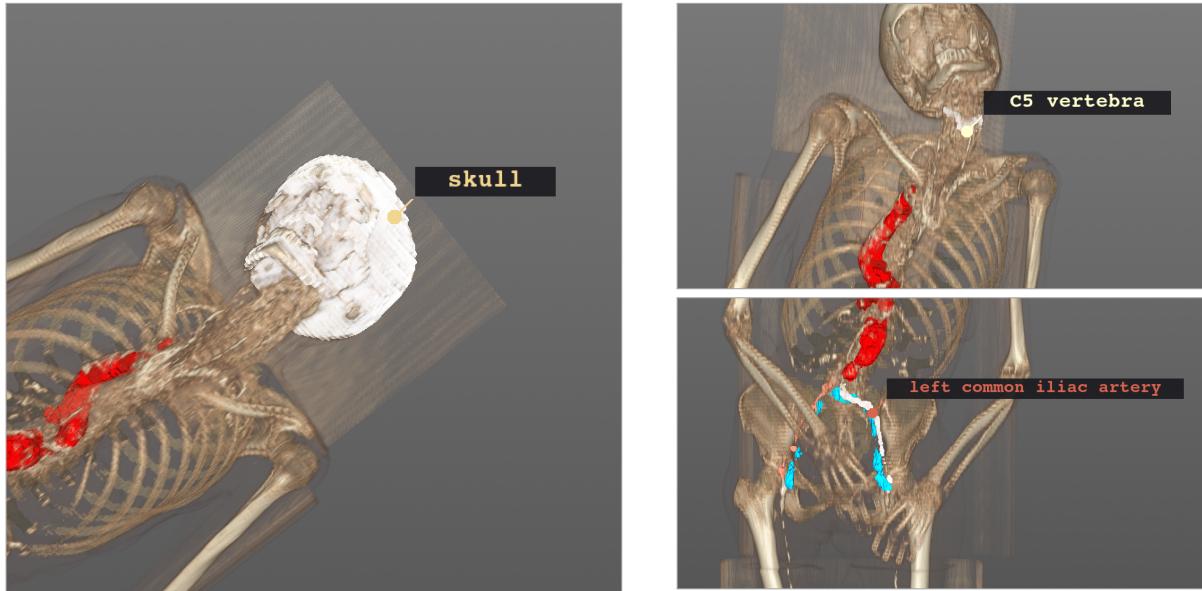


Figure 7: *Interactive 3D segmentation of donor 003 with hover labels. Left: skull. Right: C5 vertebra (top) and left common iliac artery (bottom).*

3.2 Orthogonal View

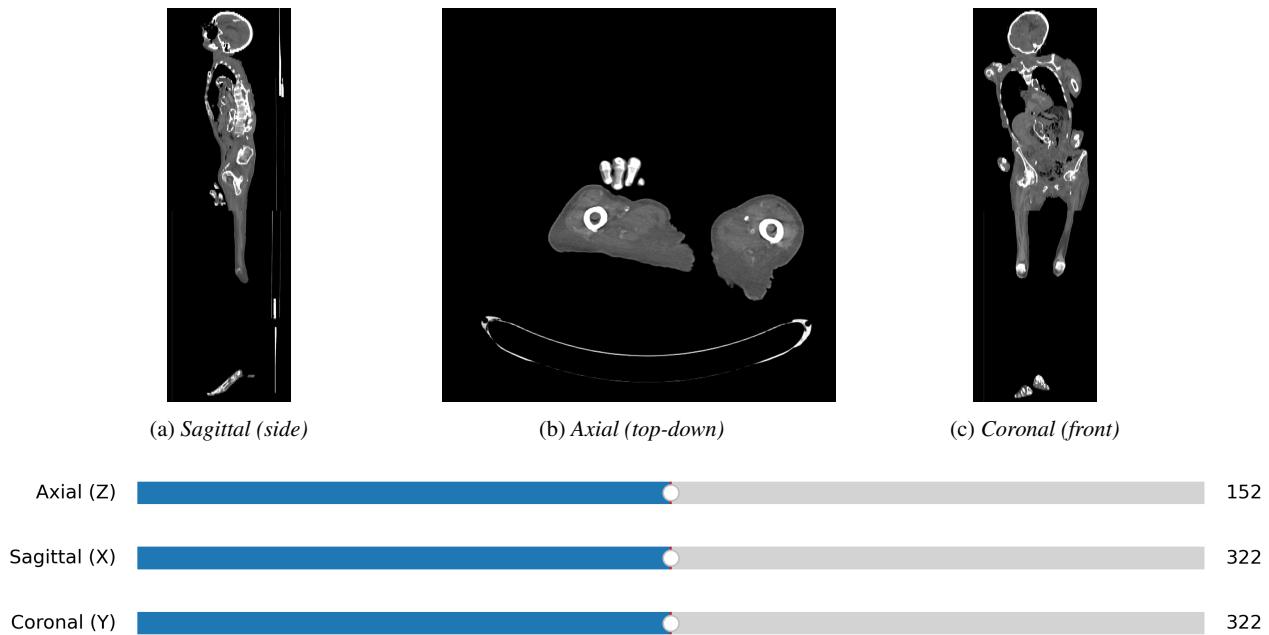


Figure 8: Orthogonal CT views of donor 003 with interactive slice navigation. Top: sagittal, axial, and coronal slices (axial centered). Bottom: axial, sagittal, and coronal slice sliders controlling the displayed planes.

Figure 8 shows the orthogonal CT slice views alongside the linked 3D view. The sliders below the images would allow the user to quickly scroll through the CT series automatically selected by the pipeline.

3.2.1 Orthogonal Segmentation View

Figure 9 shows the segmentation overlay on an orthogonal slice. The mid slice of the upper scan was selected for donor 003, as it contains a large region with visible segmented structures.

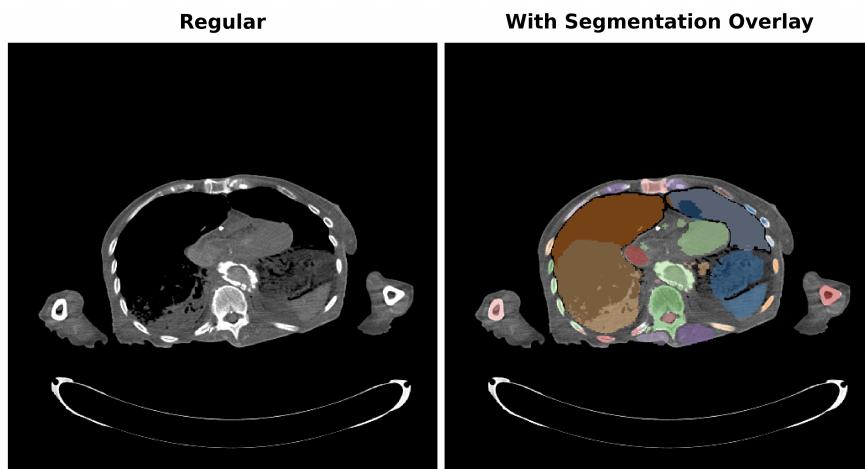


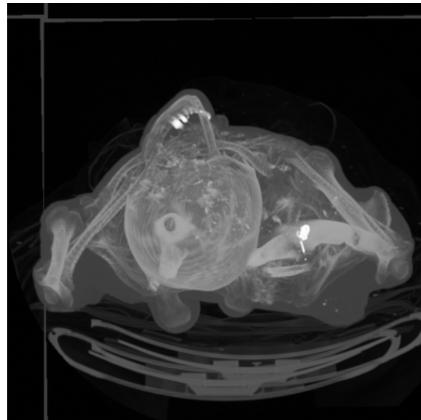
Figure 9: MONAI segmentation laid over slice 78/157 of donor 003 their torso, demonstrating segmentaion capabilities.

3.3 Maximum Intensity Projection View

Figure 10 shows MIP views from the same orientations as the other visualizations.



(a) Sagittal (side)



(b) Axial (top-down)



(c) Coronal (frontal)

Figure 10: Maximum Intensity Projection (MIP) visualizations of the reconstructed CT volume of donor 003. From left to right: sagittal, axial, and coronal projections. MIP highlights high-density structures such as bone and calcifications by projecting the maximum intensity along each viewing direction.

4 Discussion

4.1 3D view

Throughout the results presented in subsection *3D Reconstruction View* we obtain strong reconstruction, segmentation, and visualization results. Before discussing these in more detail, we highlight a large challenge on donor 003 once more: severe calcification of the aorta and iliac arteries. Since the primary objective was skeletal visualization and bone detection relied largely on HU thresholding, calcified soft tissue posed a complication because it can reach bone-like HU values and therefore appear in the same intensity range as the skeleton.

These results are shown in Figure 6, where the aorta is highlighted in red and the iliac arteries in both red and blue. The MONAI segmentation used prior to volume rendering clearly captures the aorta well, while the iliac arteries are segmented most reliably near the torso and become less complete further down the legs, which is mainly attributable to the model selection (see *Methods* for the model choice and tradeoffs).

As shown in Figure 7, labeled MONAI segmentations are also successfully loaded alongside the reconstructed bones in the CT volume and aligned using their stored spatial metadata. During rendering, these segmentations are exposed as interactive surfaces, allowing users to hover over bones or organs to retrieve anatomical labels. This provides direct visual feedback on segmentation quality and adds anatomical context during exploration, getting the intended results. Overall, any remaining artifacts in highlighting and vessel selection are attributable to the segmentation model used, which was constrained by available computational resources. More advanced models, such as TotalSegmentator, would likely yield even better results (again, we refer back to the *Methods*).

Next to the segmentation, Figure 6 demonstrates a high-quality 3D visualization of donor 003. Bone duplication is minimal, and visually, little anatomical information appears to be lost, indicating successful slicing, stitching, and thresholding. Furthermore, the two-stage alignment strategy achieved high-level alignment in the x, y, and z directions, despite significant limb displacement across the original scans. Additionally, volume rendering using VTK produced a clear and detailed skeletal visualization. We attribute this to techniques used, such as leveraging VTK's built-in

interpolation, ray casting with transfer functions, and shading. As a result, artifacts are limited and fine structural detail is preserved, despite the challenges and variability present in the original dataset.

However, some limitations are still observable in the current results. While the cut point used for slicing is generally accurate, in some scans (see the Appendix, Section *3D Visuals*) it is slightly offset. For example, donor 021, in the Appendix, shows two duplicated slices at the junction. Although this deviation is small, it can still lead to unintended inclusion or exclusion of anatomy near the junction. This observation motivates our inclusion of orthogonal views that allow users to scroll through the DICOM volume interactively (Figure 8). Such views support rapid verification and enable quick manual intervention when needed, making them a necessary component of the pipeline for robust processing.

4.2 2D View

The 2D orthogonal and MIP visualization modes play an important role in evaluating the effectiveness of reconstruction, and are also useful in their own right by providing different perspectives on the data. While the 3D volume rendering provides an intuitive global overview, artifacts and ambiguities are often easier to identify in a 2D, slice-by-slice view. In particular, small residual misalignments at scan boundaries and slight inaccuracies in the automatically detected cutoff point can be difficult to localize precisely from the 3D view alone.

The orthogonal slice view, shown in Figure 8, enables direct inspection of the underlying DICOM data without transfer functions or ray casting. By scrolling through axial, sagittal, and coronal planes, it becomes apparent when the transition between concatenated scan sessions is offset by a small number of slices. Although these offsets are often not obvious in the rendered volume, they can still lead to subtle inclusion or exclusion of anatomy near the junction. This slice-level inspection is therefore important for validating the automated overlap removal and alignment steps, and it enables rapid, targeted manual correction when required. We also know that in the orthogonal view, the 3D per-leg alignment has not been applied yet. This is expected, since that correction is applied after 3D rendering, and it makes any remaining inter-session differences more visible in the raw slice views.

In addition to visualizing raw CT slices, the orthogonal view also supports overlaying the precomputed MONAI segmentation masks. As shown in Figure 9, the labeled masks are aligned with the reconstructed DICOM volume and rendered directly on top of the axial slice. This allows segmented anatomical structures to be inspected in their original 2D context, providing a complementary perspective to the 3D segmentation view and enabling slice-level verification of segmentation quality and anatomical boundaries.

The final MIP results, as shown in Figure 10, further support the pipeline by emphasizing high-density structures such as bone and calcification. In practice, MIP views were useful for quickly assessing global skeletal continuity and spotting duplicated or truncated regions. The clarity of these projections makes them valuable for rapid screening and as an initial inspection tool in future extensions of the pipeline. At the same time, MIP collapses depth information, making it less suitable for precise localization. This reinforces the need to use it in conjunction with orthogonal slicing rather than as a standalone diagnostic view.

Overall, the 2D views act as a verification and quality-control layer within the pipeline. The main output is the 3D reconstruction, but the 2D views are important for confirming correctness, especially in edge cases where automated decisions may be slightly off. Combining 2D and 3D visualization therefore improves robustness and makes the system more practical when applied to heterogeneous, imperfect CT datasets.

5 Conclusion

This project developed an end-to-end visualization pipeline that transforms raw, multi-session DICOM CT data into an interactive skeletal 3D reconstruction, with optional anatomy-aware interaction via precomputed segmentation masks. The pipeline integrates practical reconstruction steps: slice sorting and filtering, HU conversion, resolution normalization, overlap detection and removal using IoU on thresholded bone masks, and a two-stage alignment strategy. In addition, the system renders a GPU-accelerated VTK volume with transfer functions and shading, and provides linked orthogonal slicing and MIP views for verification and analysis.

On the reference case (donor 003), the pipeline produced a clear skeletal rendering with minimal visible duplication and strong continuity across merged scan sessions. Calcified vessels, which would otherwise confound HU-based bone visualization, were handled by segmenting and overlaying the aorta and iliac arteries in a preprocessing step, allowing these structures to be highlighted rather than misclassified as bone. Remaining uncertainty is mainly limited to small deviations in the automatically selected cut point in some scans, for which the orthogonal view provides a fast and necessary quality-control mechanism.

Overall, the results show that a robust, largely automated workflow can convert heterogeneous, overlapping CT acquisitions into coherent 3D reconstructions suitable for interactive forensic exploration and documentation. The central research question is therefore answered positively: automated segmentation and interactive visualization can be combined into an integrated pipeline for this dataset, producing high-quality, usable visualizations with limited manual intervention.

5.1 Future work

When applying the pipeline across all donor bodies, we observe that some cases would benefit from refinements in series selection to maximize anatomical coverage. In a few donors, parts of an extremity fall outside the field of view of the automatically selected series, with the most extreme cases showing a partially missing arm (donors 020, 014, and 022). This behavior is not completely unexpected, as the current pipeline was evaluated primarily on the reference case discussed in the main body of the paper. These missing regions can be recovered by manually selecting and integrating additional CT series (e.g., dedicated limb scans acquired at a different slice thickness), but doing so requires a bit of case-specific manual effort. Since the focus of this work is automated reconstruction, bone segmentation, and visualization rather than solely exhaustive scan selection, we do not pursue this additional processing here, but it remains a straightforward extension.

Additionally, in a small number of cases (e.g., Body-021), the automatic overlap removal leaves a residual uncertainty of one to two slices, which can lead to minor duplication or truncation near the junction. Future versions could build on the current logic with improved heuristics for overlap handling and series selection, or introduce optional user guidance for edge cases. The modular design of the pipeline makes such extensions straightforward, and the current implementation already provides a flexible foundation for similar CT datasets.

References

- [1] W Dean Bidgood Jr, Steven C Horii, Fred W Prior, and Donald E Van Syckle. Understanding and using dicom, the data interchange standard for biomedical imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212, 1997.
- [2] Gary Bishop and David M Weimer. Fast phong shading. *ACM Siggraph Computer Graphics*, 20(4):103–106, 1986.

- [3] Tami D DenOtter and Johanna Schubert. Hounsfield unit. 2019.
- [4] Matthew H Lee, Meghan G Lubner, Vincent M Mellnick, Christine O Menias, Sanjeev Bhalla, and Perry J Pickhardt. The ct scout view: complementary value added to abdominal ct interpretation. *Abdominal Radiology*, 46(10):5021–5036, 2021.
- [5] Hayley Mickleburgh. Hayley mickleburgh — forensic archaeology, forensic taphonomy, 3d visualization, 2025. URL <https://www.hayleymickleburgh.com/>. Personal academic website. Accessed December 2025.
- [6] {Hayley L.} Mickleburgh, Noemi Procopio, Andrea Bonicelli, Nengi Ogbanga, Giulia Sguazzi, Sarah Gino, Rogier {van der Hulst}, Kennedy Doro, {Timothy P.} Gocha, {Daniel J.} Wescott, {Lisette M.} Kootker, and {Saskia T. M.} Ammer. *The Mass Grave Project*, pages 111–134. Springer Nature Switzerland AG, 2025. ISBN 9783031863073. doi: 10.1007/978-3-031-86308-0_8.
- [7] Sara Monetta. Visible from space, bloody sands expose the slaughter of tens of thousands in sudan, November 2025. URL <https://www.nbcnews.com/>. NBC News. Published November 1, 2025.
- [8] Gregory M. Nielson. On marching cubes. *IEEE Transactions on visualization and computer graphics*, 9(3):283–297, 2003.
- [9] Walter HL Pinaya, Mark S Graham, Eric Kerfoot, Petru-Daniel Tudosiu, Jessica Dafflon, Virginia Fernandez, Pedro Sanchez, Julia Wolleb, Pedro F Da Costa, Ashay Patel, et al. Generative ai for medical imaging: extending the monai framework. *arXiv preprint arXiv:2307.15208*, 2023.
- [10] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.
- [11] Abeer Salman, Yahya Abou-Ghazala, Thomas Bordeaux, Jeremy Diamond, Gianluca Mezzofiore, and Lou Robinson. Bulldozed corpses and unmarked graves, cnn investigates the fate of gaza’s missing aid seekers, December 2025. URL <https://www.cnn.com/>. CNN Investigates. Updated December 6, 2025.
- [12] William J Schroeder, Lisa Sobierajski Avila, and William Hoffman. Visualizing with vtk: a tutorial. *IEEE Computer graphics and applications*, 20(5):20–27, 2000.
- [13] Matthias van der Vlist. Budget cuts threaten research into mass graves: “you can only excavate once”, March 2025. URL <https://www.folia.nl/>. Folia (University of Amsterdam). Published March 18, 2025.
- [14] Jakob Wasserthal, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, Daniel Hinck, Alexander W Sauter, Tobias Heye, Daniel T Boll, Joshy Cyriac, Shan Yang, et al. Totalsegmentator: robust segmentation of 104 anatomic structures in ct images. *Radiology: Artificial Intelligence*, 5(5):e230024, 2023.

Appendix

A DICOM Metadata overview

1 shows all important metadata connected to our DICOM data in this project. This is referred to often, so having a quick overview to refer to could come in handy while reading this paper.

Table 1: Important DICOM metadata tags used for CT scan visualization.

Tag Name	DICOM Tag	Description
SeriesNumber	(0020,0011)	Numeric identifier for the series within a study. Used to separate different scan acquisitions (e.g., upper/lower body).
SeriesDescription	(0008,103E)	Human-readable description of the series, indicating the type of scan or reconstruction.
ImagePositionPatient	(0020,0032)	The x, y, z coordinates (in mm) of the upper-left corner of the image in the patient coordinate system. Critical for 3D volume alignment.
PixelSpacing	(0028,0030)	Physical spacing between pixel centers in mm (row spacing, column spacing). Essential for accurate spatial measurements.
SliceThickness	(0018,0050)	Nominal thickness of the image slice in mm. Used for z-axis spacing in 3D reconstruction.
SliceLocation	(0020,1041)	Relative position of the image plane in mm. Alternative to <code>ImagePositionPatient</code> for slice ordering.
InstanceNumber	(0020,0013)	Sequential number identifying the image within a series. Used for slice ordering when spatial info unavailable.
RescaleSlope	(0028,1053)	Multiplicative factor for converting stored pixel values to Hounsfield Units (HU).
RescaleIntercept	(0028,1052)	Additive offset for converting stored pixel values to Hounsfield Units (HU).
WindowCenter	(0028,1050)	Center value for display windowing. Defines the brightness level for optimal viewing.
WindowWidth	(0028,1051)	Width of the display window. Defines the contrast range for visualization.
PatientName	(0010,0010)	Name of the patient. Used for identification purposes.
StudyDescription	(0008,1030)	Description of the imaging study. Provides context about the examination.

B 3D Visuals

Figure 11 shows the results for all six donors produced with the same pipeline as described in the main paper. In some cases, parts of an arm or leg are missing because they fall outside the field of view of the automatically selected CT series. These parts can often be recovered by manually selecting an additional series, for example a separate limb scan

acquired at a different slice thickness (3 mm instead of 5 mm), but we do not pursue this here since the paper focuses mainly on automated visualization and segmentation rather than solely picking the complete scan selection. Despite the many dataset irregularities that must be handled automatically, these examples show the pipeline consistently produces stable and visually coherent results.



(a) Donor 003



(b) Donor 013



(c) Donor 014



(d) Donor 020



(e) Donor 021



(f) Donor 022

Figure 11: Overview of all six donors and their 3D visualizations.