Nov 29, 2024 Version 1

Accurate Digital Organ Models from DICOM V.1

DOI
https://dx.doi.org/10.17504/protocols.io.bp2l6d19dvqe/v1

Takashi Kimura[1]

[1]Fukushima Medical University

隆 Takashi Kimura
Fukushima Medical University

DOI: https://dx.doi.org/10.17504/protocols.io.bp2l6d19dvqe/v1

Protocol Citation: Takashi Kimura 2024. Accurate Digital Organ Models from DICOM. **protocols.io**
https://dx.doi.org/10.17504/protocols.io.bp2l6d19dvqe/v1

Protocol status: Working
We use this protocol and it's working

Created: November 27, 2024

Last Modified: November 29, 2024

Protocol Integer ID: 113024

Keywords: accurate digital organ models from dicom, format surface data of organ, surface data of organ, accurate digital organ model, organ data, dicom data, human anatomy, segmentation from scratch, editing dicom data, segmentation in dicom viewer, imaging data, organ, surface data, format surface data, segmentation, 3d, dicom, such as 3d, comprehensive digital dataset, necessary organ, dicom viewer, slicer, data, imaging

Funders Acknowledgements:
JSPS KAKENHI
Grant ID: JP19K09100

Disclaimer

Abstract

This protocol describes the procedure for editing DICOM data to create precise STL-format surface data of organs. Using this method, surface data of organs obtained through auto-segmentation in DICOM viewers such as 3D-Slicer, along with Multi-Planar Reconstruction (MPR) images generated from DICOM data, can be spatially aligned within Blender. By referencing the imaging data, it becomes possible to refine and process the data obtained via auto-segmentation and to create organ data that cannot be captured through auto-segmentation from scratch. As a result, this approach enables the construction of a comprehensive digital dataset of human anatomy that includes all necessary organs.

JPEG format MPR image creation from DICOM

1 Obtain Python script from https://github.com/tk1971-Jpn/DICOM-to-JPEG.

2 Run the script on **Jupyer Lab**.

3 Paste the path of the first file of the DICOM data you want to convert into `path_ct` (enclosed in quotation marks).

4 Create a folder to save the generated JPEG files, and paste the folder path into `storage_path`.

5 Retrieve information about the DICOM data each time a kernel is executed.

6 Execute the fourth kernel and adjust the values of `k` and `l` using the sliders to optimize the image.

7 Input the `l-k` and `l+k` values obtained from the fourth kernel into `vmin` and `vmax` in the fifth kernel.

8 When you execute the fifth kernel, JPEG files will be automatically generated and saved in the specified folder.

Auto-segmentation of organ surface data by 3D-Slicer

9 Launch **3D-Slicer**.

10 Use the Extension Manager to add **TotalSegmentator**.

11 Add the data you want to convert via **Add Data**.

12 Load the required data through **Add DICOM Data** by selecting and clicking **Load**.

13 Use **TotalSegmentator** to auto-segment organ surface data. After pressing the **Apply** button, you will see two options: "Full resolution (~5 to 50 minutes)" and "Fast (~2 minutes)." Choose the appropriate option based on your needs.

14 Save and extract the organ surface data as an STL file via **Segmentations**.
- Prepare a folder in advance to save the data.
- Specify the folder in **Destination Folder** under **Export to Files** and press the **Export** button. The data will automatically be saved in the specified folder.

Auto-segmentation of body surface data by Horos

15 In 3D-Slicer, body surface data is not extracted, so the body surface data will be extracted using Horos.

16 Launch **Horos**.

17 Import the data to be used via **Import**.

18 Open the imported data.

19 Select the **3D Surface Rendering** and set the pixel value to -200, then click "OK."

20 Skin, lungs, and other structures will be extracted. Select **Export as STL** from **Export 3D-SR**, specify the file name and save location, and save the file.

Import MPR images into Blender and display them using slider

21 Launch **Blender** and select the **Script mode**.

22 Click **New** to prepare for loading a new script.

23 Obtain the Python script from https://github.com/tk1971-Jpn/Slider-viewer-in-Blender and paste it into Blender's Script mode.
Choose the appropriate script based on the arrangement order of the DICOM data (head-to-foot or foot-to-head) and whether the data is longer in the anterior-posterior or cranial-caudal direction.

24 Assign the data obtained during the execution of **JPEG format MPR image creation from DICOM** to the corresponding variables in the Blender script

24.1 **pixel**(Blender) = pixel(Jupyter)

24.2 **pixel_pitch**(Blender) = pixel pitch(Jupyter)

24.3 **ax_slide_number**(Blender) = number of files(Jupyter)

24.4 **ax_slide_distance**(Blender) = slice distance(Jupyter)

25 Paste the path of the folder* where the JPEG format MPR image data is stored into **path_JPEG**
∗ A folder containing Folder **A**, Folder **C**, and Folder **S**

26 When the script is executed, a tag named **Image Slider** will appear in the sidebar of the Layout mode. You can use the sliders to display any Axial, Coronal, or Sagittal images.

27 *Since the size of the human body far exceeds Blender's workspace, it was scaled down to 1/50 for import.*

Import Organ Surface Data in STL Format into Blender

28 Select Script Mode in Blender

29 Click **New** to prepare for loading a new script.

30 Obtain the Python script from https://github.com/tk1971-Jpn/Import-organ-STL-data-into-Blender/blob/main/Import%20organ%20STL.ipynb and paste it into the Blender's

Script mode.

31 Past the path of the folder where the organ STL data is stored into **folder_path**.

32 When the script is executed, all organ STL data are imported into Blender in bulk, and the object names are appropriately converted during the process.

33 If there is a positional misalignment of the organs (which often occurs), use Blender's "Move" tool to adjust their positions based on the reference image.

34 (Option)Delete all the imported STL data, set the positions, and re-import them. The data used for position adjustments can be obtained from Blender's sidebar under Transform > Location. Based on this data, assign values to the script's x_move, y_move, and z_move (default is 0).
*#Delete all objects: press **A** and then press **X***

35 *Since the size of the human body far exceeds Blender's workspace, it was scaled down to 1/50 for import.*