Proceedings M2D2022 - 9th International Conference on Mechanics and Materials in Design
Funchal/Portugal 26-30 June 2022, Editors J.F. Silva Gomes and S.A. Meguid, pp.1109-1114
https://paginas.fe.up.pt/~m2d/proceedings_m2d2022/

PAPER REF: 19196

DEALING WITH CT CARDIAC IMAGING USING PYTHON: AN APPROACH FOR FUTURE HEMODYNAMIC SIMULATIONS

L. Matias[1,2(*)], C.F. Castro[1,2], C.C. António[1,2], L.C. Sousa[1,2], S.I.S. Pinto[1,2], S. Silva[3]

[1]Engineering Faculty, University of Porto, Porto, Portugal
[2]Institute of Science and Innovation in Mechanical and Industrial Engineering (LAETA-INEGI), Porto, Portugal
[3]Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, Aveiro, Portugal
[(*)]*Email:* up201502891@edu.fe.up.pt

ABSTRACT

Multiple detector computed tomography (MDCT) has become an indispensable complementary diagnostic tool in cardiac disease providing a dynamic assessment of cardiac function supporting various diagnostic exams, such as cardiac angiography and perfusion. Overall, these exams can consist of multiple cardiac volumes, over time, a huge amount of data that is often not explored to its full extent. In this regard, advancing methods that can help clinicians take the most out of this data to support their diagnosis is very important. Given the size and complexity of the data, proposing novel processing and analysis methods poses several challenges, one of them being the establishment of a base framework supporting reading the imaging data, along with all the associated metadata (e.g., acquisition parameters), visualizing the volume data (e.g., allowing multiple cutting planes and volume rendering), and applying processing and analysis methods (e.g., noise removal). In this regard, the work presented here is a preliminary effort in the context of project and aimed to establish the pipeline for a method prototyping infrastructure capable of dealing with these images to identify cardiac structures for further hemodynamic simulations. These preliminary results show that the devised solution adopting the python scripting language is adequate to handle the task.

Keywords: computed tomography, cardiac imaging, medical images, python.

INTRODUCTION

Cardiac DICOM files obtained with CT scanning enables storing not only a digital image but also associated metadata such as, acquisition parameters and patient information. Digital images, relevant for clinical diagnosis, are composed of pixels, organized in a 2D matrix, with finite and discrete quantities of numeric representation for its intensity or grayscale. For CT scanning, it is usual to use an imaging matrix of 512x512 grayscale pixels (Pianych, 2012). Each pixel, stored as 16-bit binary number, provides 2^{16} = 65536 possible shades of gray. The open-source library Pydicom allows an easy access to DICOM data dictionary that contain all standard data items (attributes) used in digital medicine. Each item has his own *(Group, Element)* tag. Each tagged item is also known as DICOM attribute or DICOM element. Scanner settings, patient information, image parameters, image pixel data, etc. are examples of attributes stored in DICOM files.

To differentiate the body structures in a CT DICOM image, tissue density values are represented using Hounsfield scale. Tissues denser than water have positive values of Hounsfield units (HU) while tissues less dense have negative HU values, for example, bones (~1000 HU) and air (-1000 HU), respectively (Kamalian et al. 2016). Scikit-image library

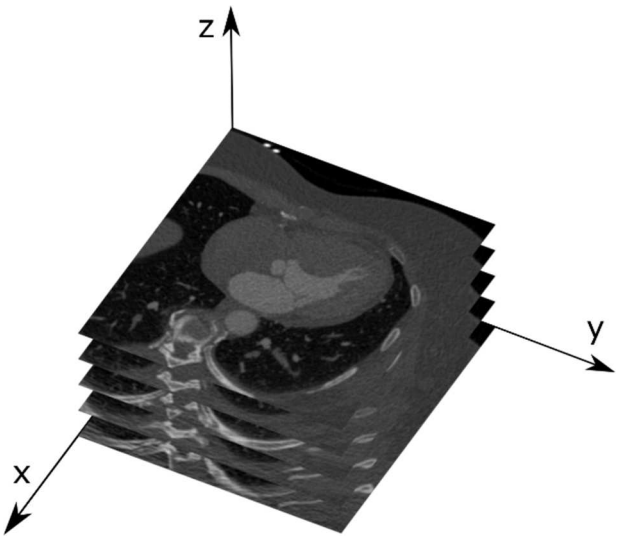
provide an infrastructure for computer vision applications with optimized algorithms for computer vision and machine learning. Such algorithms allow image enhancement and the application of spatial filters that multiply a kernel (matrix used to select the pixel neighborhood) by each pixel – convolution. Image smoothing is an example of spatial filters applications by applying a linear averaging filter (e.g., Gaussian Blur) or a non-linear median filter (the best solution when in the presence of salt and pepper noise). Smoothing allows noise reduction although it removes small details from the image (Chityala et al. 2021, Sandipan 2018). Thus, the goal of the present work is to deal with DICOM images using python scripting language to create an infrastructure capable of leading with these files for the purpose of segmenting the different cardiac structures for future hemodynamic simulations.

METHODOLOGY

In first place, using Python, the CT DICOM file set is examined using Pydicom library to obtain the attributes (Figure 1a) including the stored pixel data values (SV). Each DICOM file represents an axial slice (x-y plane) of human body. Real dimensions of the CT volume are obtained with Pixel Spacing (0028,0030) and Slice Thickness (0018,0050) attributes. Pixel spacing attribute contains the dimensions of each pixel in x and y directions.

```
(0028, 0011) Columns                              US: 512
(0028, 0030) Pixel Spacing                        DS: [0.4296875, 0.4296875]
(0028, 0100) Bits Allocated                       US: 16
(0028, 0101) Bits Stored                          US: 12
(0028, 0102) High Bit                             US: 11
(0028, 0103) Pixel Representation                 US: 0
(0028, 0106) Smallest Image Pixel Value           US: 0
(0028, 0107) Largest Image Pixel Value            US: 4089
(0028, 1050) Window Center                        DS: [50, -600]
(0028, 1051) Window Width                         DS: [350, 1200]
(0028, 1052) Rescale Intercept                    DS: '-1024.0'
(0028, 1053) Rescale Slope                        DS: '1.0'
(0028, 1054) Rescale Type                         LO: 'HU'
(0028, 1055) Window Center & Width Explanation    LO: ['WINDOW1', 'WINDOW2']
```

(a)



(b)

Fig. 1 – (a) DICOM dictionary extract with (Group, Element) tag and corresponded values. (b) Stack of different slices obtained with the stored pixel data values (SV). Axial view represented in x-y plane.

To obtain the image pixel data in HU scale, the *Rescale Type (0028,1054)* must be *"HU"* and the stored pixel data values are converted using Eq. 1:

$$HU = m \times SV + b \tag{1}$$

Where m and b are *Rescale Slope (0028,1054)* and *Rescale Intercept (0028,1052)*, respectively.

From this point, all image transformations will be applied to an HU scaled image, that allow us to separate the different structures present in the CT image applying a window of HU levels. All slices are stacked (Fig. 1b) to obtain a three-dimensional array and create a volumetric model of the DICOM files. Often, CT images have an unwanted change in pixel values – noise. Salt and pepper noise is one of the most ordinary noises present in medical images.

To denoise/smooth the image, a median filter, implemented in *scikit-image* library, is applied with different spatial kernel dimensions for comparison. A 2D kernel – 2D smooth – that uses only the neighbor pixels of the slice, and a 3D kernel – 3D smooth – that considers the neighbor pixel values in adjacent slices.

An efficient marching cubes algorithm (Lewiner et al. 2003), already implemented in *scikit-image* library, is used to create an iso-surface mesh, and obtain the vertices and faces of the volume.

RESULTS AND CONCLUSIONS

DICOM file set provided by the Cardiology Department of Gaia/Espinho Hospital Centre, has 176 files/slices. Attributes in Table 1 were extracted.

Table 1 – DICOM attributes extracted from files.

Attribute Name	Attribute Tag	Value
Rows	(0028,0010)	512
Columns	(0028,0011)	512
Pixel Spacing	(0028,0030)	[0.4296875, 0.4296875]
Slice Thickness	(0018,0050)	0.6
Rescale Type	(0028,1054)	HU
Rescale Slope	(0028,1054)	1
Rescale Intercept	(0028,1052)	-1024

Regarding the number of columns, rows and number of slices, the CT volume in study has the real dimensions $220 \times 220 \times 106.5\ mm$. Rescale type attribute is "HU" that allow us to transform stored pixel data values to an HU scale using Equation 1 and obtain the image in

Figure 2 (a). A median filter was applied to denoise the image. It was used a 2D and 3D kernel with different sizes – Figure 2b to Figure 2e.

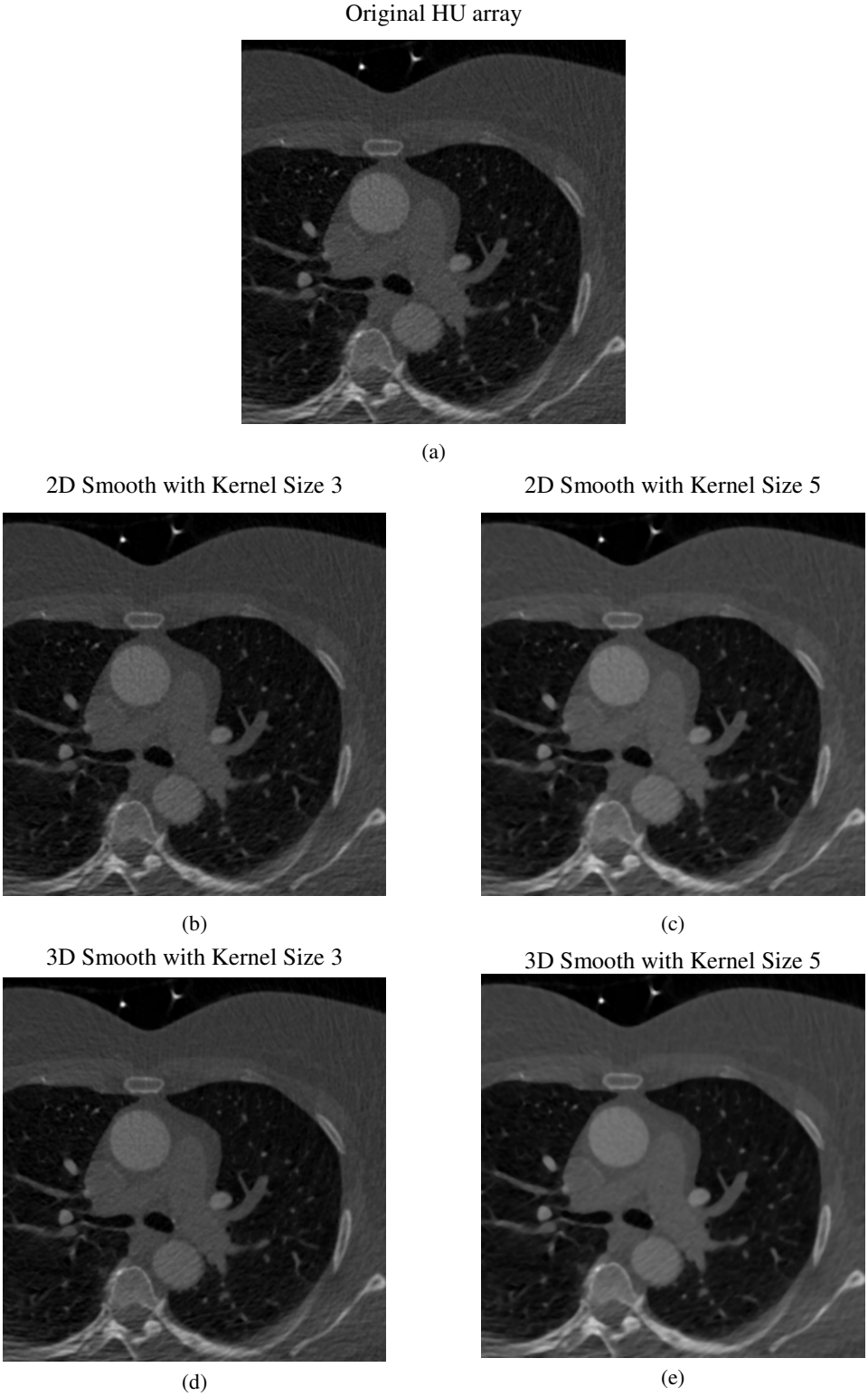


Fig. 2 – First slice of DICOM file set: (a) original HU array; (b) smoothed array with 2D kernel with size 3; (c) smoothed array with 2D kernel with size 5; (d) smoothed array with 3D kernel with size 3; (e) smoothed array with 3D kernel with size 5.

For further volume rendering, it was applied a window of 300 – 301 to remove all soft and fat tissues, $\sim[-120\ HU : +300\ HU]$, present in image. Windowed images are shown in Figure 3.

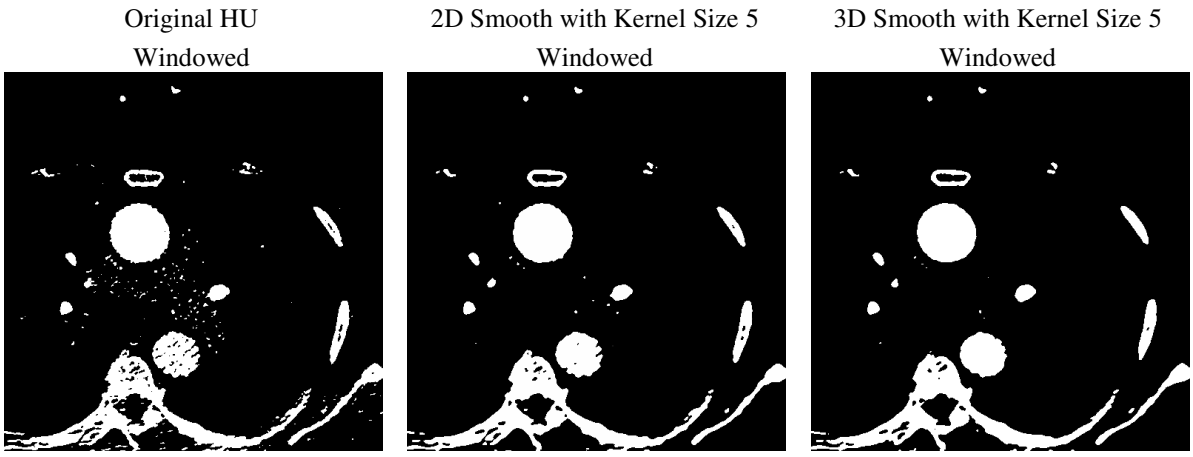


Fig. 3 – (a) First slice of original HU with a 300-301 window; (b) First slice of a 2D smoothed array with kernel size 5; (c) First slice of a 3D smoothed array with kernel size 5;

The marching cube algorithm was applied for a volume reconstruction of the CT images and results are in Figure 4.

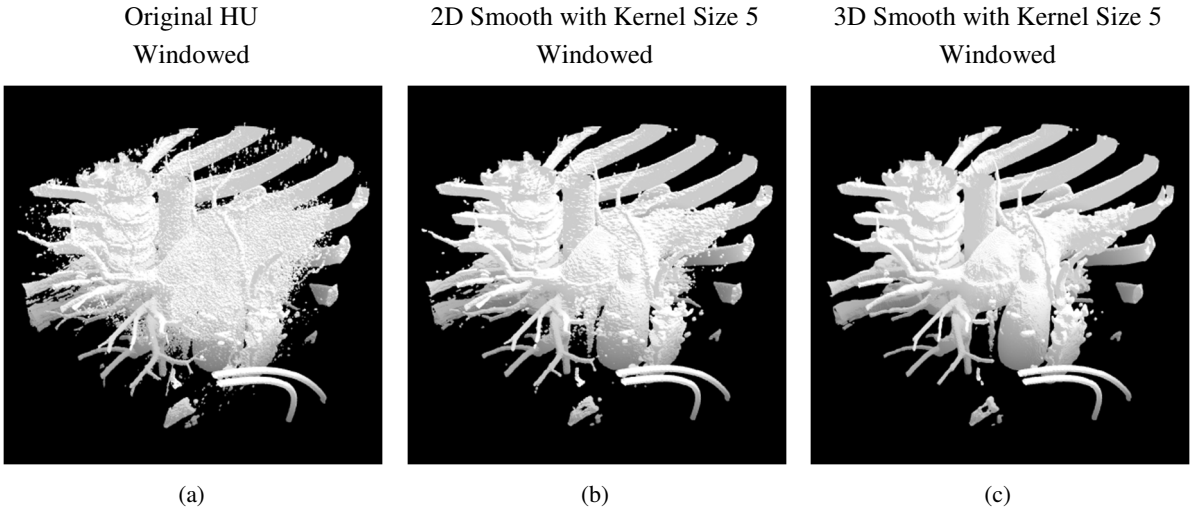


Fig. 4 – Volumetric representation of (a) original HU array; (b) smoothed array with 2D kernel size 5; (c) smoothed array with 3D kernel size 5.

These results evidence that python scripting language is adequate to deal with medical imaging data. Comparing the smooth median filter with different kernel size we concluded that images will be smoother but will lose detail. The results for a 2D or 3D kernel show that with the same kernel size, the 3D kernel is more effective on denoising.

Hearth and coronary arteries segmentation with colored masks, dynamic volume rendering and interactive visualization interface (e.g., region growing) and hemodynamics simulations are examples of the following steps.

ACKNOWLEDGMENTS

Authors gratefully acknowledge the financial support of FCT (Portugal) regarding the R&D Project "CADS-FACT – PTDC/EMD-EMD/0980/2020", the Engineering Faculty of University of Porto, the Institute of Science and Innovation in Mechanical and Industrial Engineering, the Institute of Electronics and Informatics Engineering of Aveiro, the University of Aveiro, the Cardiovascular R&D Unit of the Medicine Faculty of University of Porto and the Cardiology Department of Gaia/Espinho Hospital Centre.

REFERENCES

[1] Chityala R, Pudipeddi S. Image Processing and Acquisition Using Python. Taylor &Francis Group, 2021, pp.61-121.

[2] Kramme R, Hoffman K, Pozos R. Springer Handbook of Medical Tehcnology. Springer, 2011, pp.311-338.

[3] Kamalian S, Lev MH, Gupta, R. Computed tomography imaging and angiography – principles. Handbook of Clinical Neurology, Elsevier, 2016, 135, pp.3-20.

[4] Lewiner T., Lopes H., Vieira A. W., Tavares G., Efficient Implementation of Marching Cubes' Cases with Topological Guarantees. Journal of Graphics Tools. Taylor & Francis, 2003, pp.1-15.

[5] Pianykh O. Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide. Springer, 2012, pp.27-111.

[6] Sandipan D. Hands-on Image Processing with Python. Packt Publishing, 2018, pp.132-171.