Tool for peripheral artery segmentation and reconstruction from angiography images

Nikola J. Petrovic[1], Vassiliki T. Potsika[1], Mohammed AboArab[1] Maria Zacharopoulou[2], Linnea Tscheuschner[2], Achilleas Chatziioannou[2], Fragsika Sigala[2], Dimitris I. Fotiadis[1, 3]

[1]Unit of Medical Technology & Intelligent Information Systems Department of Materials Science & Engineering, University of Ioannina, Ioannina, Greece (e-mail: n.petrovic@uoi.gr, vpotsika@gmail.com, m.aboarab@uoi.gr, fotiadis@uoi.gr)

[2]School of Medicine, National Kapodistrian University of Athens, Athens, Greece (e-mail: marizazach@gmail.com , tscheuschnerlinnea@gmail.com , achatzi@med.uoa.gr , drfsigala@yahoo.gr)

[3]Biomedical Research Institute – FORTH, Ioannina, Greece

Abstract The aim of this work is the development of the system that performs importing, archiving, processing and display of angiographic image data as well as conversion into 3D models. The developed system has been divided into eight stages. The implementation of the system has been performed involving the data from an animal study and 9 patients with angiography images of abdomen, upper and lower part of the legs. The development of the system was performed in the Python programming language and was planned to be integrated with other programming libraries that provide other functionalities within the DECODE project. The used artery detection algorithms are based on semi-automatic and automatic algorithms based on conventional image filters as well as supervised/unsupervised machine learning approaches. The aim of this research is the development of a system that can perform three-dimensional reconstruction of peripheral arteries based on two-dimensional X-ray angiography images. The determination of the spatial points of the peripheral arteries is performed based on the pixel distance from the edge of the blood vessel, assuming that the blood vessel is cylindrical in shape. This method provides fast and simple results in the form of a mesh of a three-dimensional object, while providing the possibility for use on smaller data sets that lack images from other angles or from other forms of medical imaging such as Optical Coherence Tomography and Intravascular Ultrasound. The results depict significant potential for 3D artery reconstruction with limited data.

Keywords Peripheral artery disease; Angiography; Image segmentation; Image reconstruction.

1 Introduction

The peripheral artery disease is characterizing partial or complete blockage of noncardiac arteries which is most caused by atherosclerotic plaque. In addition to atherosclerosis, some of the less frequent causes are inflammatory disorders of the arterial wall and non-inflammatory arteriopathies such as fibromuscular dysplasia. Recent studies show that the patients with atherosclerosis have high risk of other diseases such as cerebrovascular disease (30%), ischemic heart disease (60%) and those patients with intermittent claudication have a 15% probability of fatal outcome in a period of five years. This is the main reason why the medical imaging techniques play such important role in the treatment and prevention of the peripheral artery diseases [1, 2].

X-ray angiography is a method which has been used for more than 50 years and usually consists of array of two-dimensional (2D) images with determined time step. Using the contrast which is visible in X-ray images it is possible to mark arteries and distinguish them from the rest of the image. Even though 2D imaging techniques omit significant amounts of information regarding the third dimension, it is possible to compensate it with synchronized images from two different angles or using the devices such as Computed Tomography, Magnetic Resonance Imaging, Optical Coherence Tomography (OCT) or Intravascular Ultrasound [3, 4].

2 Methodology

The peripheral Artery Reconstruction and Segmentation tool (PARSEC) has been developed in order to process the obtained medical imaging data. This tool can import and process angiography and OCT data in Digital Imaging and Communications in Medicine (DICOM) and video format. The output of this system is surface and volumetric mesh which can be used by other tools. Fig. 1 shows the modules and information flow of the developed PARSEC tool.

2.1 Import module

The role of this module is to import from a specified directory to process and organize the data. Imported dataset is organized at: patient ID, study ID, series ID, image dataset ID.

By selecting the specific folder and data type (DICOM or video), the Import module is starting the loading of all available data. In the case of video data, this created a list of all files. For single video, the algorithm creates a 4D array whose dimensions are width, height, channels, and frames. The metadata of the video are stored into a dictionary. In the case of DICOM data, the algorithm creates a tree of

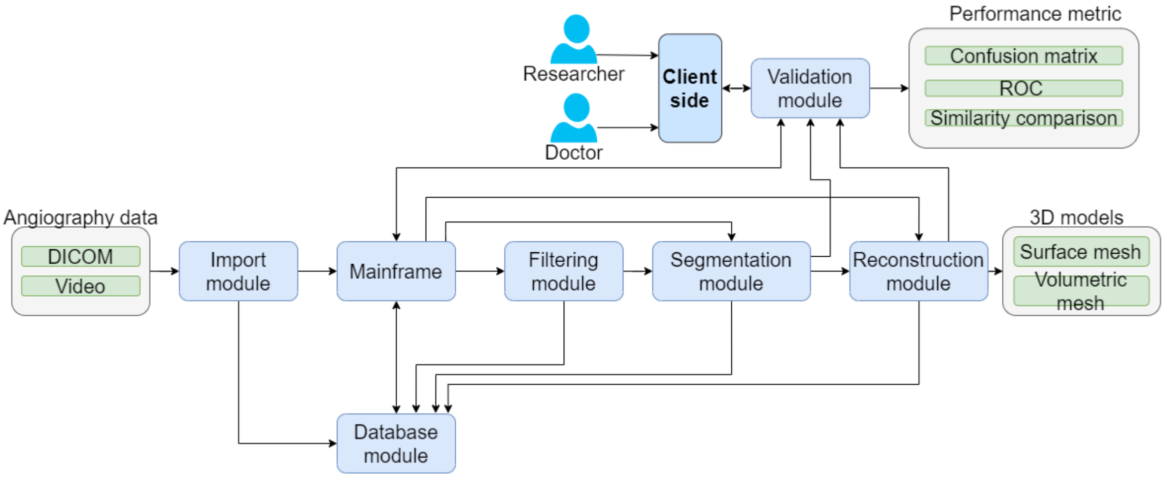

Fig. 1 Modules and information flow of the PARSEC tool.

data for each patient, patient's study, study's series, and series' image. This module is using OpenCV [5, 6] and PyDicom [7] for the loading of video and DICOM data, NumPy [8] for conversion into datasets, Matplotlib [9] for the visual representation and OS [10, 11] for file scanning and operations. All the loaded data can be displayed before being forwarded to the Database module.

2.2 Database module

The database module organizes, archives, and updates the records which has been provided from the Import, Filtering, Segmentation and Reconstruction modules. The stored data are stored, while in the future, a cloud-based json format will be used.

The locally stored database is based on allocated files and folders of angiography images. As such each angiography file has a database file with the same name, which is easily readable.

The global database is organized to archive all acquired imaging data coupled with a propriate metadata. Each time the user reads new file, there is an update of the corresponding series, study, and patient.

This module uses the python libraries pathlib [10, 11], pickle [12], json [13], os [10, 11] and PyDicom [7].

2.3 Filtering module

Before performing high quality segmentation and reconstruction, it is necessary to perform filtering of the data to supress the imaging data of bones and tissues.

Filtering of angiography images has frequency filtering and intensity enhancement techniques. The intensity enhancement techniques are used to adjust the contrast, value range, as well as to perform linear and nonlinear operations.

The result of the filtering module is an image which is clearly representing the peripheral arteries with contrast.

Since this module is being used for multiple operations, it consists of five functions to perform the required operations. High – pass and Low – pass filtering functions have been developed to perform the frequency filtering with the specified frequency filter size. The processed image is first transformed using the Fourier transformation, and then we apply the filter and transform the resulting image using the reverse Fourier transformation. The normalization function is most frequently used since it is necessary for almost any other operation, it is rescaling the pixel intensities to the desired range of values. The main filter function provides a choice of filters and processes of the input image. In addition to the frequency filtering, Laplacian and Gaussian image filtering can be used. We assume that the only significant movement in the imaging dataset is the contrast flow and by using the filters such as median, summation and differential between the images we can extract the necessary imaging data, which is the peripheral artery image.

This module is using python libraries NumPy [8], SciPy [14-16], Matplotlib [9] and PIL [17, 18].

2.4 Segmentation module

The main task of the Segmentation module is to provide the binary image of peripheral arteries. As the output the binary image with positive values for pixels of the artery and false values for pixels of the background is provided.

Two approaches have been implemented for the segmentation module:

- Adaptive threshold segmentation – this method is based on setting the threshold which will distinguish the arteries from the rest of the image since the injected contrast is making the arteries to appear darker than the rest of the image. The threshold is calculated from minimal, maximal, and mean value of the input image.
- K-means clustering – We have used unsupervised K-means clustering which is closely related to the KNN algorithm. Since the data we segment refer to the peripheral arteries of legs, we have defined three clusters: left leg, right leg, and background [15].

This module uses python libraries such as OpenCV [5, 6], NumPy [8, 14], scikit-learn [16] and Matplotlib [9].

2.5 Reconstruction module

Since the provided data was captured just from one angle, it was necessary to develop another algorithm to perform the reconstruction based on single angiography image. In order to complete this task, the artery was approximated as symmetric geometry which resembles the mathematical model of a tube. As the first necessary step, the Reconstruction module is applying the developed function called Contour Extractor, which performs the recursive edge detector and, in this manner, it is computing the between each pixel and its closest edge. Based on the calculated distance, the function is calculating the maximal pixel distance from the edge, which is being used as the radius of the peripheral artery. Using Eq. 1, this function generates two points with same x and y coordinates but the opposite z value which creates the points for the peripheral artery wall symmetrical in x-y plane, where r is the maximal pixel distance from the closest edge in the image and a is current pixel distance from the closest edge.

$$z = \pm\sqrt{(r^2 - (r - a)^2)} \tag{1}$$

The calculated points are given as the input to the main Reconstruction Function which is performing the conversion of the given coordinates into a point cloud. Using the Delaunay 3D library [19], the mentioned point cloud is converted into volume, and with extract_geometry function we are obtaining the surface mesh which is saved as stl.

This module is using the following python libraries: NumPy [8, 14], Pyvista [20], PIL [17, 18], Matplotlib [9, 21].

2.6 Validation module

The validation module has been developed to compare the obtained results with annotations provided by the medical experts. It is based on the presentation of angiography images overlapping the segmented image. The task for the medical expert is to put the x annotations on the places where the algorithm has not detected the artery and o annotations on the places where the algorithm has falsely detected the artery.

2.7 Client side

The client side is defined as user interface which allows to technicians, doctors, and researchers to use the developed platform to import, export, edit and visualize the imaging data of the peripheral arteries. The client side plays a significant role to collect the annotation data from doctors to perform the semi-automatic segmentation and validation.

2.8 Mainframe

The mainframe module controls every other module, and it can be located on a local machine or server. All processes are automatic, and they can be updated via the client side by the users.

3 Results

The results obtained are divided into three categories, filtering, segmentation and reconstruction. The greatest focus was placed on the processing of angiographic images of the Femoral and Tibial arteries, where the mentioned images were used as inputs for all three systems, where the mentioned results can be seen in the Figures 3, 4 and 5. The Filtration module contains multiple functions for the filtration of angiography images and Fig. 3 provides the comparison of the original image, the frequency filtration and the median filtration with their histograms. After analysing the entire set of results from available data, it can be concluded that the best results are obtained with median filtration which preserves all the information and suppress the stochastic noise from individual images.

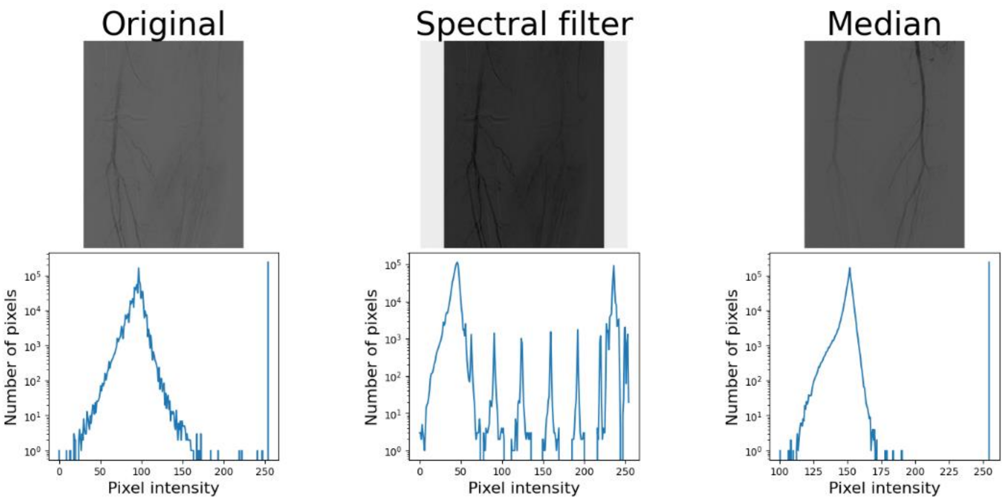

Fig. 2 Original image, frequency filter and median filter outputs.

Fig. 3 depicts the comparison of K-means segmentation and Adaptive threshold segmentation. The segmented artery in the white colour and the False values of pixels as segmented as background black colour. As it can be seen, the K-means segmentation results have more noise, but Adaptive threshold segmentation have more False Negative pixels which removes significant information.

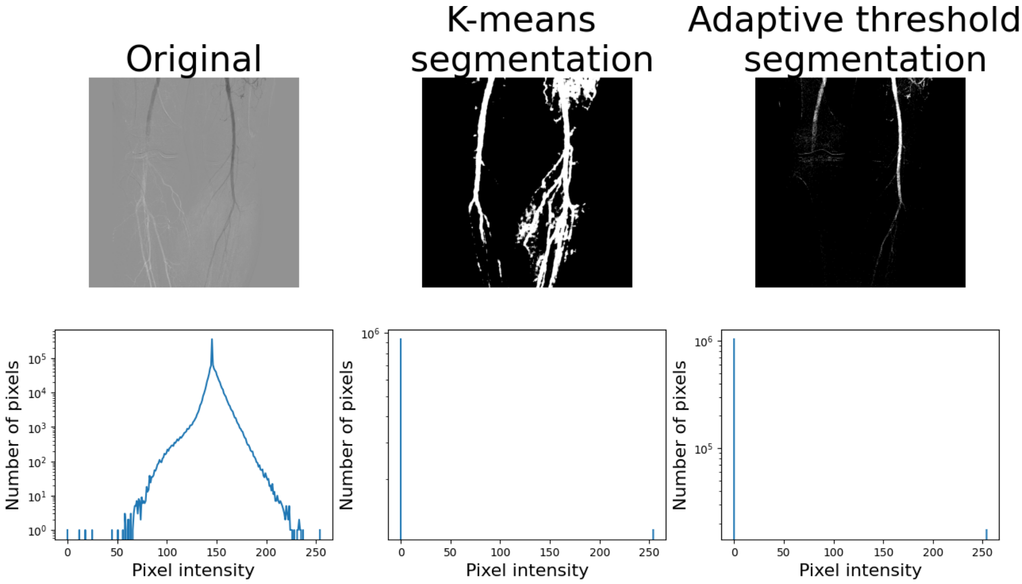

Fig. 3 Original image, K-means segmentation results and adaptive threshold segmentation filter results

The reconstruction results significantly depend on the filtration and segmentation results. Fig. 5 depicts the reconstruction based on the median filtration of the original image and Adaptive threshold segmentation of the filtered image.

4 Conclusion

The obtained results demonstrate the potential for the improvement of the processing, segmentation, and reconstruction of peripheral arteries from angiography images. The developed tools, algorithms and procedures will be used for further work which will be coupled with other tools for Finite Element Method and Computational Fluid Dynamics.

As one of the next steps, it is planned to extend the tool to be able to handle the filtration, segmentation and reconstruction from different types of medical imaging data such as OCT, CT and others.

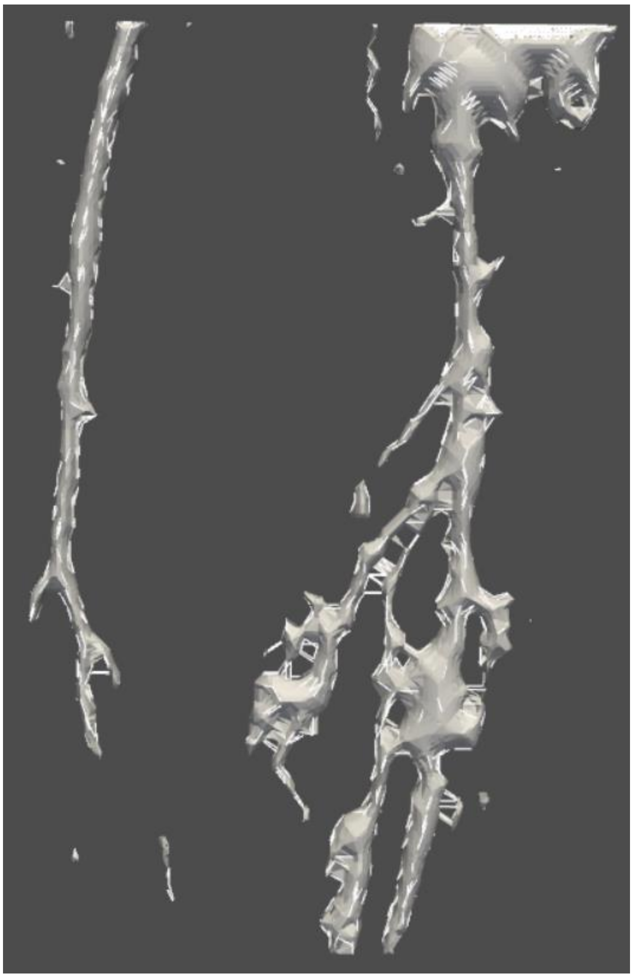

Fig. 5 Reconstruction results of the peripheral arteries

5 Discussion

The latest scientific works in this field show us that filtering, segmentation and reconstruction of peripheral arteries based on X-ray angiographic images provides a wide field of application.

The main disadvantages of this system come down to the robustness of the filter, the examination of the library for unsupervised learning, the sensitivity to noise in the input signal, the quality of converting the cloud of points into a network. By comparing the results of filtering and segmentation of Iliac, Femoral, Popliteal, Infrapopliteal and Tibial arteries, it has been noticed that the quality differs significantly, which is why it is necessary to further improve and optimize the system so that there are no quality deviations for different arteries. K-means segmentation does not provide sufficiently high-quality results and requires additional investigation regarding the dimensions, resolution and shape of the input images. After calculating the spatial points of the reconstructed blood vessel, errors occur during connecting the points and openings are observed due to problems with surface and volumetric meshing. The idea is to overcome this obstacles with additional filtering and algorithms that specify the environment of each point in more detail.

The developed system and the obtained results indicate a further need for expansion and improvement of the current methodology. As one of the first next steps, the input data will be augmented with additional angiographic images from a different angle. After that, the addition of other input images such as IVUS, OCT and CT will be considered.

6 Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No GA 956470, as part of the DECODE project.

7 References

1. Solomon, C., I. Kullo, and T.J.N.E.J.M. Rooke, *Peripheral artery disease.* 2016. **374**(9): p. 861-87.
2. Morley, R.L., et al., *Peripheral artery disease.* 2018. **360**.
3. Andrikos, I.O., et al. *A new method for the 3D reconstruction of coronary bifurcations pre and post the angioplasty procedure using the QCA*. in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019. IEEE.
4. Çimen, S., et al., *Reconstruction of coronary arteries from X-ray angiography: A review.* 2016. **32**: p. 46-68.
5. Bradski, G. and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. 2008: " O'Reilly Media, Inc.".
6. Gollapudi, S., *Learn computer vision using OpenCV*. 2019: Springer.
7. Mason, D.J.M.P., *SU-E-T-33: pydicom: an open source DICOM library.* 2011. **38**(6Part10): p. 3493-3493.
8. Oliphant, T.E., *A guide to NumPy*. Vol. 1. 2006: Trelgol Publishing USA.
9. Ari, N. and M. Ustazhanov. *Matplotlib in python*. in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*. 2014. IEEE.
10. Python, W.J.P.R.f.W., *Python.* 2021. **24**.

11. vanRossum, G.J.D.o.C.S., *Python reference manual.* 1995(R 9525).
12. Fasnacht, L.J.J.o.O.S.S., *mmappickle: Python 3 module to store memory-mapped numpy array in pickle format.* 2018. **3**(26): p. 651.
13. Dhalla, H.K. *A Performance Analysis of Native JSON Parsers in Java, Python, MS. NET Core, JavaScript, and PHP*. in *2020 16th International Conference on Network and Service Management (CNSM)*. 2020. IEEE.
14. Bressert, E., *SciPy and NumPy: an overview for developers.* 2012.
15. Bisong, E., *Introduction to Scikit-learn*, in *Building machine learning and deep learning models on Google cloud platform*. 2019, Springer. p. 215-229.
16. Kramer, O., *Scikit-learn*, in *Machine learning for evolution strategies*. 2016, Springer. p. 45-53.
17. Mágr, M., *Python imaging library: Python Imaging Library*. 2019, Brno University of Technology.
18. Umesh, P.J.C.C., *Image processing in python.* 2012. **23**(2).
19. Liu, Y., G.J.C.S. Yin, and D. Analysis, *The Delaunay triangulation learner and its ensembles.* 2020. **152**: p. 107030.
20. Sullivan, C. and A.J.J.o.O.S.S. Kaszynski, *PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK).* 2019. **4**(37): p. 1450.
21. Tosi, S., *Matplotlib for Python developers*. 2009: Packt Publishing Ltd.