Development of a Three-Dimensional Computational Pipeline in Python for Personalized Heart Modeling

Filipe L Namorato[1], Daniel M P Leme[1], Thaís J Soares[1], Rafael S Oliveira[2], Thaiz R Schmal[1], Rodrigo W dos Santos[1], Joventino O Campos[1]

[1] Federal University of Juiz de Fora, Juiz de Fora, Brazil
[2] Federal University of São João del-Rei, São João del-Rei, Brazil

Abstract

This work presents the development of a three-dimensional Python-based pipeline for generating personalized cardiac models from segmented magnetic resonance images. The tool integrates anatomical alignment, reconstruction of cardiac and fibrotic surfaces, geometric smoothing, volumetric mesh generation, and myocardial fiber orientation assignment. It relies exclusively on open-source tools, employing techniques such as linear-regression-based alignment and direct reconstruction of fibrotic regions from annotated ROIs through extrusion and surface triangulation, ensuring anatomical fidelity, especially in the representation of fibrotic tissue. Then, a biventricular mesh is generated for use as input of finite-element and finite-volume solvers, enabling personalized electrophysiological and electromechanical simulations. Meshes produced by this open-source pipeline are compared with another publicly available tool, demonstrating similar results in terms of healthy tissue reconstruction and improvements in fibrotic representation.

1. Introduction

Advances in medical imaging, especially cardiac magnetic resonance imaging (MRI), combined with computational modeling techniques, have propelled the study of heart diseases [1]. Virtual models have proven effective for assessing and reducing the risk of sudden cardiac death (SCD) [2]. In particular, Camps et al. [3] demonstrated that personalized models can optimize clinical decisions and minimize invasive procedures. Additionally, myocardial fibrosis significantly alters electrical conduction [4], facilitating ventricular arrhythmias. Thus, including fibrotic tissue is essential for accurate electrophysiological simulations.

Nevertheless, constructing personalized models involves several complex steps: segmenting images, generating cardiac and fibrotic meshes, defining transmurality for cell heterogeneity assignment, determining fiber orientation, and converting data into simulator-compatible formats. Multiple tools partially automate this process: CemrgApp [5] provides a graphical interface with image processing, computer vision, and machine learning modules; Meshtool [6] enables efficient command-line mesh manipulation, including mesh generation and smoothing; Banerjee et al. [7] introduced an automated pipeline for 3D biventricular reconstruction from 2D images using deep learning and statistical modeling. Meanwhile, 3D-heart-models [8] [1], developed in MATLAB, integrates anatomical reconstruction and fiber generation for the openCARP simulator but depends on proprietary software.

Therefore, this work proposes a three-dimensional semi-automatic pipeline capable of converting previously segmented cardiac MRI images into detailed computational models, including fibrotic areas and fiber orientations, using only open-source tools. The developed pipeline, named MyoMesh[2], was adapted from 3D-heart-models and restructured in Python, offering greater accessibility, reproducibility, and direct compatibility with the MonoAlg3D simulator [9], FEniCS, and openCARP, broadening its applicability.

2. Methods

2.1. MRI segmentation

The three-dimensional geometry construction begins with segmented cardiac magnetic resonance images (MRI) using the Segment software [10]. These can be exported in MATLAB's format, including both cardiac structures and fibrotic regions, as shown in Figure 1(a).

Medical image segmentation was performed by cardiologists at the University Hospital of the Federal University of Juiz de Fora (HU-UFJF). These specialists were responsible for precisely delineating cardiac structures and iden-

[1] https://github.com/vildenst/3D-heart-models
[2] https://github.com/FISIOCOMP-UFJF/MyoMesh

tifying fibrotic regions in each patient, ensuring both the quality and accuracy of the personalized models produced.

2.2. Cardiac geometry and surfaces reconstruction

From the segmentations, a point cloud with coordinates (X, Y, Z) is obtained, where Z corresponds to the slice index. In order to correct respiratory artifacts that affect slice positioning during MRI scans [11], the pipeline calculates the centroid of the epicardium and endocardium in each slice, followed by a linear regression along the Z-axis to reposition the slices according to the estimated trajectory.

The segmented cardiac structures are then reconstructed as closed triangular surfaces and smoothed using methods from the VTK library [12], ensuring a precise and regular anatomical representation. The resulting surfaces are exported, as illustrated in Figure 1(b).

Fibrotic regions are reconstructed directly from annotated regions of interest (ROIs) and then integrated into the cardiac structures, forming the final three-dimensional geometry for mesh generation.

2.3. Fibrosis reconstruction and integration

Fibrotic tissue is segmented using structured ROIs in Segment [10], providing direct access to annotated coordinates.

Each ROI is extruded along the Z axis using the slice thickness and inter-slice gap. A closed surface is then generated by connecting contours between adjacent slices and applying 2D Delaunay triangulation to cap the ends. The resulting surface is smoothed and used to represent the fibrotic region.

This surface is used in a marking routine that identifies tetrahedra enclosed by fibrotic regions using `vtkSelectEnclosedPoints` and tags them accordingly in the mesh file.

2.4. Volumetric mesh and fiber generation

The cardiac and fibrotic surfaces are used in Gmsh [13] to create the volumetric finite element mesh. Each surface is labeled with specific tags: biventricular epicardium (40), left ventricle endocardium (30), right ventricle endocardium (20), and base plane (10). From these labels, two main volumes are defined: healthy myocardium (0) and fibrotic region (1).

The resulting mesh is smoothed with VTK [12] methods, eliminating degenerate elements. Next, the mesh is converted to a format suitable for FEniCS [14] and processed by a Python implementation of the Laplace-Dirichlet Rule-Based (LDRB[3]) algorithm [15], which is used to generate the fiber orientation and transmurality, as shown in 1(d). Then, the mesh and the microstrutucture properties are stored in a format compatible with electrophysiological simulations in MonoAlg3D [9], as shown in Figure 1(e). Additionally, the pipeline provides explicit routines for exporting meshes in openCARP format.

3. Results

We performed a qualitative comparison of the outputs from two pipelines: the one developed in this work (Python-based with meshes suitable for the MonoAlg3D simulator) and the original pipeline by Strøm et al. [8], implemented in MATLAB and designed to generate meshes compatible with the openCARP simulator. Notably, the original pipeline relies on proprietary tools for mesh generation, restricting accessibility and adaptability in open environments.

3.1. Anatomical fidelity and mesh quality

To qualitatively assess anatomical fidelity and mesh quality, we reconstructed 3D models from the same MRI segmentation and visualized them under identical conditions (Patient 1 in Fig. 2, Patient 2 in Fig. 3).

The proposed pipeline preserves key anatomical details (Fig. 2(a) and Fig. 3(a)), particularly fibrotic regions, maintaining spatial continuity and avoiding undesired intersections. In contrast, the original pipeline struggles to accurately represent smaller fibrotic regions (Fig. 2(b) and Fig. 3(b)), resulting in detail loss and occasional intersections, which may undermine the accuracy of electrophysiological simulations.

3.2. Pipeline accessibility and integration

Beyond technical aspects, we also considered the practical installation and execution of each pipeline. The proposed pipeline is implemented in a Python environment, integrating open-source tools such as Gmsh for mesh generation and incorporating C++ code originally developed by Strøm et al. [8]. This integration allows for greater flexibility and simplifies maintenance and adaptation across different operating systems, ensuring replicability. Moreover, the integration of routines capable of exporting meshes directly to MonoAlg3D, openCARP and FeNiCs significantly enhances the versatility of the proposed pipeline compared to the original MATLAB version.

On the other hand, the original pipeline is implemented with MATLAB code and depends on specific versions of libraries and external tools, some of which are no longer

[3] https://github.com/finsberg/ldrb

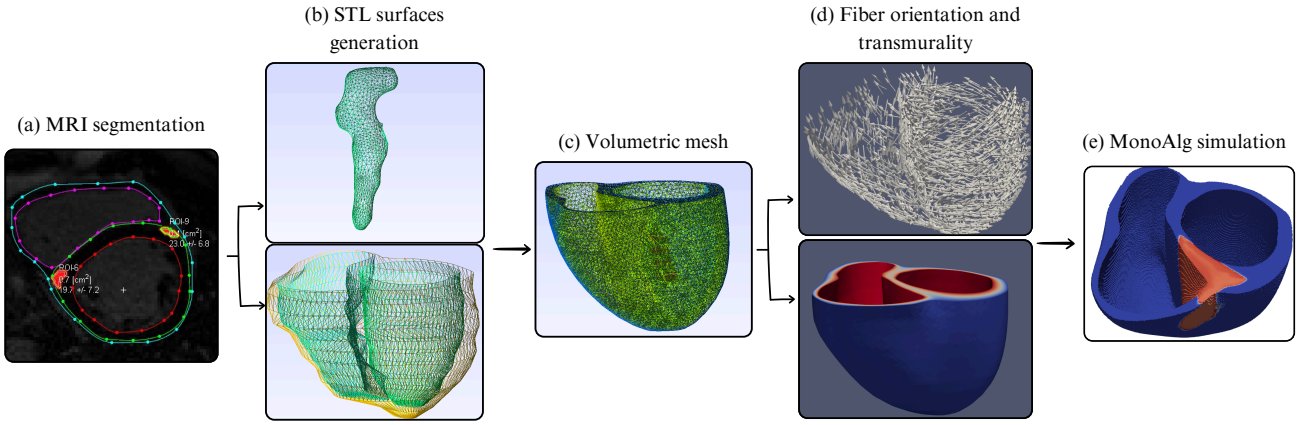

Figure 1. Proposed pipeline for generating three-dimensional personalized cardiac models: (a) Segmentation of cardiac MRI images; (b) Creation of surfaces in `.stl` format: endocardium for left and right ventricles, biventricular epicardium, base plane and fibrosis surfaces; (c) Generation of the volumetric finite element mesh; (d) Determination of myocardial fiber orientation and transmurality by using Laplace-Dirichlet rule based method; (e) Electrophysiological simulation using the MonoAlg3D simulator.

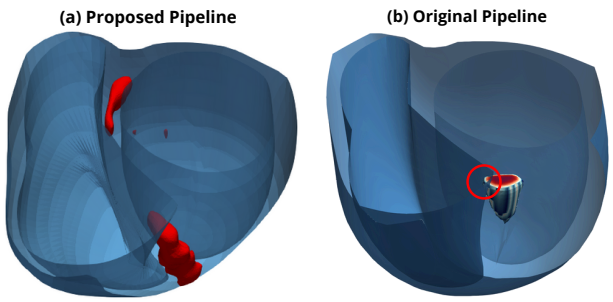

Figure 2. Patient 1. Biventricular geometry and fibrosis reconstruction in two pipelines: (a) Proposed pipeline, preserving anatomical consistency and clear separation of regions; (b) Original pipeline, presenting intersections and loss of detail.

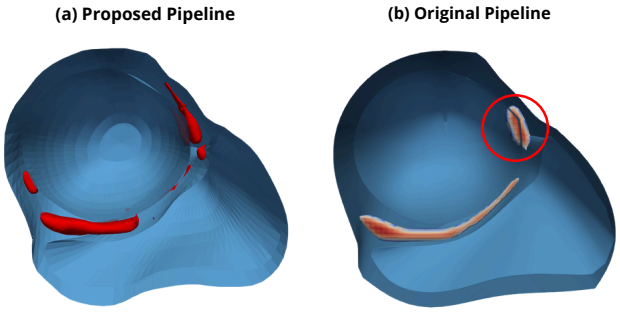

Figure 3. Patient 2. Biventricular geometry and fibrosis reconstruction in two pipelines: (a) Proposed pipeline, preserving anatomical consistency and clear separation of regions; (b) Original pipeline, presenting intersections and loss of detail.

actively maintained. The original version, not updated since 2017, poses compatibility issues on modern systems.

The proposed pipeline successfully generates complete three-dimensional models, including fibrotic tissue, enabling use in electrophysiological simulations. In contrast, the original pipeline produces models that require additional adaptation steps and lacks support for precise volumetric marking of fibrotic tissue, particularly for small or thin regions.

3.3. Volumetric comparison

We compared ventricles cavity volumes computed on the mesh with respect to Segment measurements in seven cases. Left ventricular volumes were very close to the ones reported in Segment, with a mean absolute percentage error (MAPE) of 4.55%, while for right ventricular volumes the computed MAPE was 12.3%. These results support practical agreement for the left ventricle and motivate simple post processing and quality checks for the right ventricle.

4. Conclusion

This work presents a three-dimensional open-source pipeline for generating personalized cardiac models compatible with finite-element and finite-volume simulators. The approach efficiently integrates multiple stages—from segmented image alignment to final mesh conversion in a Python-based environment designed to simplify installation and usability.

Compared with the original MATLAB-based pipeline,

the proposed pipeline offers superior anatomical preservation of fibrotic tissue, direct compatibility with simulators such as MonoAlg3D and openCARP, and requires no additional conversions or manual adjustments. The open-source implementation, along with provided Python environment configurations, significantly enhances reproducibility and facilitates broader community adoption in computational cardiology research.

Nevertheless, a current limitation is that the generated models lack full coverage of the basal heart, particularly affecting the right-ventricular roof and outflow tract. While this incompleteness may reduce accuracy in analyses involving RV outflow dynamics, future work will investigate complementary imaging strategies and hybrid modeling approaches to mitigate this limitation.

Acknowledgments

The authors would like to express their thanks to Minas Gerais State Research Support Foundation (FAPEMIG) - PCE-00048-25; APQ-02752-24, APQ-02445-24, FINEP, National Council for Scientific and Technological Development (CNPq); Empresa Brasileira de Serviços Hospitalares/ EBSERH; SINAPAD Santos-Dumont, and UFJF for funding this work.

References

[1] Corral-Acero J, Margara F, Marciniak M, Rodero C, Loncaric F, Feng Y, Gilbert A, Fernandes JF, Bukhari HA, Wajdan A, Martinez MV, Santos MS, Shamohammdi M, Luo H, Westphal P, Leeson P, DiAchille P, Gurev V, Mayr M, Geris L, Pathmanathan P, Morrison T, Cornelussen R, Prinzen F, Delhaas T, Doltra A, Sitges M, Vigmond EJ, Zacur E, Grau V, Rodriguez B, Remme EW, Niederer S, Mortier P, McLeod K, Potse M, Pueyo E, Bueno-Orovio A, Lamata P. The 'digital twin' to enable the vision of precision cardiology. European Heart Journal 03 2020;41(48):4556–4564. ISSN 0195-668X.

[2] Huynh K. Arrhythmia risk stratification using virtual heart models. Nature Reviews Cardiology 2016;13(7):381–381.

[3] Camps J, Berg LA, Wang ZJ, Sebastian R, Riebel LL, Doste R, Zhou X, Sachetto R, Coleman J, Lawson B, Grau V, Burrage K, Bueno-Orovio A, Weber dos Santos R, Rodriguez B. Digital twinning of the human ventricular activation sequence to clinical 12-lead ecgs and magnetic resonance imaging using realistic purkinje networks for in silico clinical trials. Medical Image Analysis 2024;94:103108. ISSN 1361-8415.

[4] Sachetto R, Alonso S, dos Santos RW. Killing many birds with two stones: Hypoxia and fibrosis can generate ectopic beats in a human ventricular model. Frontiers in Physiology 2018;9. ISSN 1664-042X.

[5] Razeghi O, Solís-Lemus JA, Lee AW, Karim R, Corrado C, Roney CH, de Vecchi A, Niederer SA. Cemrgapp: An interactive medical imaging application with image processing, computer vision, and machine learning toolkits for cardiovascular research. SoftwareX 2020;12:100570. ISSN 2352-7110.

[6] Neic A, Gsell MA, Karabelas E, Prassl AJ, Plank G. Automating image-based mesh generation and manipulation tasks in cardiac modeling workflows using meshtool. SoftwareX 2020;11:100454. ISSN 2352-7110.

[7] Banerjee A, Camps J, Zacur E, Andrews CM, Rudy Y, Choudhury RP, Rodriguez B, Grau V. A completely automated pipeline for 3d reconstruction of human heart from 2d cine magnetic resonance slices. Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences 2021;379(2212):20200257.

[8] Strøm V. Using Personalized Virtual Hearts to Assess Arrhythmia Risk in Acute Infarction Patients. Master's thesis, University of Oslo, 2018. URL https://www.duo.uio.no/handle/10852/63482.

[9] Sachetto Oliveira R, Martins Rocha B, Burgarelli D, Meira Jr W, Constantinides C, Weber dos Santos R. Performance evaluation of GPU parallelization, space-time adaptive algorithms, and their combination for simulating cardiac electrophysiology. International Journal for Numerical Methods in Biomedical Engineering 2018;34(2):e2913.

[10] Heiberg E, Sjögren J, Ugander M, Carlsson M, Engblom H, Arheden H. Design and validation of segment–freely available software for cardiovascular image analysis. BMC Medical Imaging 2010;10(1):1–13.

[11] Scott AD, Wylezinska M, Birch MJ. Motion artefacts in mri: a complex problem with many partial solutions. Journal of Cardiovascular Magnetic Resonance 2015;17(1):1–12.

[12] Schroeder W, Martin K, Lorensen B. The Visualization Toolkit (4th ed.). Kitware, 2006. ISBN 978-1-930934-19-1.

[13] Geuzaine C, Remacle JF. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 2009;79(11):1309–1331.

[14] Alnaes MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN. The fenics project version 1.5. Archive of Numerical Software 2015; 3.

[15] Bayer JD, Blake RC, Plank G, Trayanova NA. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. Annals of Biomedical Engineering 2012;40:2243–2254.

Address for correspondence:

Joventino de Oliveira Campos
Departamento de Ciência da Computação da UFJF, Juiz de Fora - MG, 36036-330
joventino.campos@ufjf.br