Annika Niemann*, Gabor Janiga, Bernhard Preim, Daniel Behme, and Sylvia Saalfeld

Centerline and blockstructure for fast structured mesh generation

https://doi.org/10.1515/cdbme-2022-0004

Abstract: In contrast to unstructured meshes, structured meshes yield faster simulation results for bio-medical simulations, but are very time-consuming to create. A preprocessing step in the generation of structured meshes is the manual construction of a blockstructure approximating the vessel. Here, we present an automatic centerline calculation and blockstructure generation to reduce the user effort and time of structured mesh generation. The centerline is detected as points in between opposite faces. Based on the centerline, cross sections are determined and a blockstructure which approximates the vessel is automatically generated. The centerline detection does not require time-consuming user input and meshes with more than 195,000 vertices are processed in less than 160 seconds. The results of the presented automatic centerline detection are compared to a centerline with manual input generated by the widely used vmtk tool. The centerlines are similar, small differences occur at bifurcation and at the aneurysm.

1 Introduction

Research of deformation of vessels in the brain, like aneurysm or arteriovenous malformation, often requires several mesh processing steps of the 3D vessel model. For example the centerline and a blockstructure (Fig. 1) are used to generate structured meshes (Fig. 2) for hemodynamic simulations. An important part is the centerline detection. Based on the centerline, further analysis, for example morphological parameter calculation, or preparation of structured meshes for hemodynamic simulations is carried out [4, 12]. A common tool for centerline calculation is the vessel modelling toolkit (vmtk) [1, 5, 9]. However, the centerline calculation with vmtk often requires manual input in form of selecting points on the mesh and repetition if the first result is not sufficient, for example if the centerline of a vessel branch is missing. The presented approach does not require user input.

***Corresponding author: Annika Niemann, Gabor Janiga, Bernhard Preim, Sylvia Saalfeld,** Otto-von-Guericke University Magdeburg, STIMULATE, e-mail: annika.niemann@ovgu.de
Daniel Behme, Universitätsklinikum Magdeburg

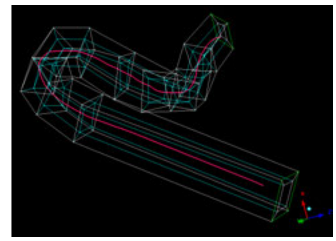

Fig. 1: Manual generated blockstructure with centerline (red)

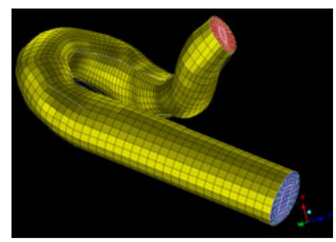

Fig. 2: Structured mesh for a segment of a blood vessel

2 Related Work

Guo et al. [6] presented a deep learning centerline detection for coronary arteries. They use a fully convolutional network with a minimal path extractor to generate single-pixel wide centerlines in binary segmentation masks. Yang et al. [14] also presented deep learning centerline detection, using a U-Net to predict the centerline of roads in aerial images. These deep learning solutions are tailored to specific use cases and can not be easily transferred to other applications. A none deep learning based framework calculating the centerline of 2D or voxelized 3D models was presented by Hassouna and Farag [7].

Antiga et al. [1] detect the centerline as shortest part between two extremal points based on a voronoi diagram. The detection requires start and end point of the centerline and is not suitable for circular structures, such as the Circle of Wilis. Wei et al. [13] presented a centerline calculation for vascular meshes. Their algorithm is based on the assumption that vascular structures consists of segments of cylindrical shape. In the first step, the vessel is segmented into several segments using k-means fuzzy clustering. After segmentation and possible interactive refinement of the segmentation, the centerline is calculated using cut planes for each vertex. Then, the centerline is smoothed and thinned.

Other approaches detect the centerline in images and are restricted to the image modality they were deveopled for [8, 10].

The presented approach works on various 3D models of vessels, for example aneurysms or circular structures like the circle of willis. It does not require user interaction.

3 Centerline

For each face f of the surface mesh, the center of the face f_c and the face normal f_n are calculated. Based on this and an estimated radius r of the vessel, search points P_s are calculated as:

$$P_s{=}f_c - r * f_n. \quad (1)$$

These search points are points, which are near the triangles opposite to the face center.The radius is approximated by using the radius of the nearest outlet. The nearest outlet is selected based on the distance between face and outlet along the mesh surface. The outlets are automatically detected by searching for edges which only belong to one face. This is the case for open meshes, where the outlets are not closed. If none of these are found, edges where the adjacent faces build an approximately 90 degree angle are searched for. In the next step, the vertices of these edges are analyzed. If they lie in one plane and form a roughly circular shape, they are classified as outlet.

For each search point the k closest face centers f_c respective faces are selected. It is then searched for an intersection of the line between f_c and corresponding search point P_s and the selected faces. The restriction of the faces which are tested for an intersection improves the run time of the algorithm. Empirically, k was set to 500 for reliable and timely determination of intersections. Next, the midpoint of the line between the face center and the intersection point is added to the initial centerline estimation.

Fig. 3: Four lines along the mesh surface

In the next step, outliers are removed. Especially with close branches, some faces from different vessel branches can be included in the faces selected with the search points. This leads to intersection points not on the opposite vessel but further away. As a result, some points from the initial centerline estimation might be outside of the aneurysm mesh. For each initial centerline point, the vector between the centerline point and the closest face center of the mesh is calculated and normalized. Under the assumption that the facenormales are oriented outwards, for points inside the mesh the vector in direction of the closest face center and the corresponding facenormal should have a similar orientation. Centerline points not fullfilling this, are removed as outliers. In the last step, close centerline points are merged together.

4 Block initialization

In order to fasten the structured mesh generation, blocks are created automatically. The first step is the calculation of the centerline as described in the previous section.

Cross sections of the mesh perpendicular to the centerline are determined. If a cross section is roughly circular, four points with largest distance to each other are determined. Circularity is determined based on the variance of the distance between points of the cross section contour and the center. The cross sections are shown in Figure 4. The points are ordered into four lines, as shown in Figure 3. In the first cross section, four points with maximal distance to each other are selected. In the following sections the points closest to the points in the previous section are selected. The points on these lines are initial corner points for the blocks. Each block consists of eight points from two cross sections, two points from each of the four lines. With exception of the points at the start and the end, each point belongs to exactly two blocks. These initial blocks are iterative merged. Two blocks are merged, if the resulting larger block is still inside the mesh.

Fig. 4: Points of cross-sections: intersections between mesh and plane perpendicular to centerline

5 Results

For various meshes from an own database and the aneurisk dataset [2] the centerline is calculated. The centerlines produced here are compared to centerlines produced by the vmtk toolkit, which is commonly used in research [1, 3]. In the vmtk tool the start and end points at the outlets are manually set. If

Fig. 5: Example of centerline detection, blue: own centerline, green: vmtk centerline

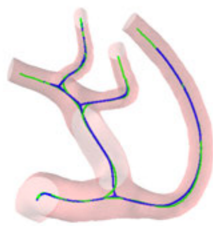

Fig. 6: Detail: different centerline at bifurcation, blue: own centerline, green: vmtk centerline

necessary, for example because not all branches are included, new points are set or the mesh is split up into several meshes (for example meshes containing cycles like the whole Circle of Willis) and the centerline calculation is repeated until the result is sufficient. From a qualitative point of view, both centerlines were similar (see Fig. 5). Vmtk is better in producing a smooth centerline in bifurcations and the aneurysm (see Fig. 7). The parameters of the presented algorithm were set to produce good results for most aneurysms (for example Fig. 5). However, this leads to sub-optimal results for some, which results in several points around the actual centerline as shown in Fig. 8. The centerline detection also provides a user interface where the user can adjust the number of merging iterations to address this problem. For Table 1 and Figures 5-9 the parameters were constant and not adjusted manual. The times for complete centerline detection without user input are shown in Table 1. For a complete Circle of Willis with 149,959 vertices the centerline calculation needed 124.81 seconds (see Fig. 9).

Tab. 1: Time for centerline calculation and difference between points of the presented centerline and the vmtk centerline

Dataset	number of vertices	time in sec	mean difference
BP	195278	156.56	0.041
C0079	38055	42.87	0.117
C0084	25567	27.90	0.108
C0055	32119	35.16	0.120
C0021	33215	36.30	0.131
C0047	29890	31.84	0.129
KM	45511	52.16	0.074
RNRN	23987	72.6	0.190
KE	88785	88.81	0.0623
SA	62997	132.20	0.100

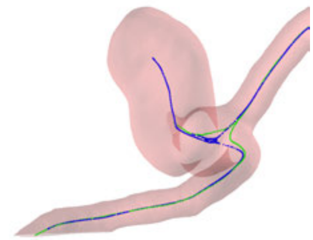

Fig. 7: Detail: centerline in/under aneurysm, blue: own centerline, green: vmtk centerline

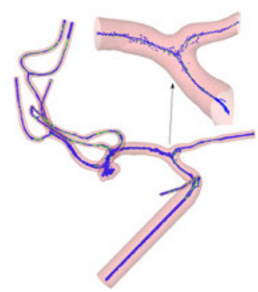

Fig. 8: Centerline with non optimal results. Further merging the points would improve the result. blue: own centerline, green: vmtk centerline

This could not be achieved with vmtk, as the many branches are major challenge and require several iterations of manual seed point selection to produce a complete centerline.

The proposed algorithm can approximate vessels with blocks (see Fig. 10).

6 Discussion

The presented centerline does not require user input. Therefore, the centerline calculation can be easily included in other algorithms, for example automatic morphological parameter calculation or structured mesh generation. It is suitable for a wide variety of structures and can also be used for large objects with cycle graphs, for example a whole Circle of Willis.

While for most datasets the quality of the centerline was good, for some meshes non optimal results are produced. Especially at bifurcations a sub-optimal centerline may occur.

The blockstructure could be used to reduce the necessary user input and simplify the generation of structured meshes.

Fig. 9: Result of centerline detection for circle of willis

Fig. 10: Blockstructure for vessel

7 Summary

The presented centerline calculation is automatic and produces results comparable to the often used semi-automatic vmtk tool. Based on the centerline detection an automatic blockstructure construction for structured mesh generation was described.

For future work, the centerline extraction can speed up the extraction of morphological parameters [11] or other hemodynamic preprocessing steps for subsequent blood flow simulation like flow splitting [12] and creation of structured meshes based on the blockstructure.

Author Statement Research funding: This study was funded by the Federal Ministry of Education and Research within the Forschungscampus *STIMULATE* (grant number 13GW0473A) and the German Research Foundation (grant number SA 3461/3-1). Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent has been obtained from all individuals included in this study. Ethical approval: The research related to human use complies with all the relevant national regulations, institutional policies and was performed in accordance with the tenets of the Helsinki Declaration, and has been approved by the authors' institutional review board or equivalent committee.

References

[1] The vascular modelling toolkit.

[2] Aneurisk-Team. AneuriskWeb project website, http://ecm2.mathcs.emory.edu/aneuriskweb. Web Site, 2012.

[3] Luca Antiga and David Steinman. Robust and objective decomposition and mapping of bifurcating vessels. *IEEE transactions on medical imaging*, 23:704–713, 07 2004.

[4] Philipp Berg et al. The Computational Fluid Dynamics Rupture Challenge 2013—Phase II: Variability of Hemodynamic Simulations in Two Intracranial Aneurysms. *Journal of Biomechanical Engineering*, 137(12), 11 2015. 121008.

[5] Mahsa Ghaffari, Kevin Tangen, Ali Alaraj, Xinjian Du, Fady T. Charbel, and Andreas A. Linninger. Large-scale subject-specific cerebral arterial tree modeling using automated parametric mesh generation for blood flow simulation. *Computers in Biology and Medicine*, 91:353–365, 2017.

[6] Zhihui Guo, Junjie Bai, Yi Lu, Xin Wang, Kunlin Cao, Qi Song, Milan Sonka, and Youbing Yin. Deepcenterline: A multi-task fully convolutional network for centerline extraction. In Albert C. S. Chung, James C. Gee, Paul A. Yushkevich, and Siqi Bao, editors, *Information Processing in Medical Imaging*, pages 441–453, 2019.

[7] M.S. Hassouna and A.A. Farag. Robust centerline extraction framework using level sets. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 458–465 vol. 1, 2005.

[8] Jiafa He, Chengwei Pan, Can Yang, Ming Zhang, Yang Wang, Xiaowei Zhou, and Yizhou Yu. Learning hybrid representations for automatic 3d vessel centerline extraction. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 24–34, Cham, 2020. Springer International Publishing.

[9] Anne Jorstad, Jérôme Blanc, and Graham Knott. Neuromorph: A software toolset for 3d analysis of neurite morphology and connectivity. *Frontiers in Neuroanatomy*, 12, 2018.

[10] Zhixun Li, Yingtao Zhang, Guangzhong Liu, Haoyang Shao, Weimin Li, and Xianglong Tang. A robust coronary artery identification and centerline extraction method in angiographies. *Biomedical Signal Processing and Control*, 16:1–8, 2015.

[11] S. Saalfeld, P. Berg, A. Niemann, M. Luz, B. Preim, and O. Beuing. Semiautomatic neck curve reconstruction for intracranial aneurysm rupture risk assessment based on morphological parameters. *International Journal of Computer Assisted Radiology and Surgery (IJCARS)*, 13(11):1781–1793, 2018.

[12] Sylvia Saalfeld, Samuel Voß, Oliver Beuing, Bernhard Preim, and Philipp Berg. Flow-splitting-based computation of outlet boundary conditions for improved cerebrovascular simulation in multiple intracranial aneurysms. *International Journal of Computer Assisted Radiology and Surgery*, 14(10):1805–1813, July 2019.

[13] Mingqiang Wei, Qiong Wang, Yichen Li, Wai-Man Pang, Luming Liang, Jun Wang, Kelvin Kian Loong Wong, Derek Abbott, Jing Qin, and Jianhuang Wu. Centerline extraction of vasculature mesh. *IEEE Access*, 6:10257–10268, 2018.

[14] Xiaofei Yang, Xutao Li, Yunming Ye, Raymond Y. K. Lau, Xiaofeng Zhang, and Xiaohui Huang. Road detection and centerline extraction via deep recurrent convolutional neural network u-net. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):7209–7220, 2019.