VesselVAE: Recursive Variational Autoencoders for 3D Blood Vessel Synthesis

Paula Feldman[1,3(✉)], Miguel Fainstein[3], Viviana Siless[3], Claudio Delrieux[1,2], and Emmanuel Iarussi[1,3]

[1] Consejo Nacional de Investigaciones Científicas y Técnicas, Buenos Aires, Argentina
paulafeldman@conicet.gov.ar
[2] Universidad Nacional del Sur, Bahía Blanca, Argentina
[3] Universidad Torcuato Di Tella, Buenos Aires, Argentina

Abstract. We present a data-driven generative framework for synthesizing blood vessel 3D geometry. This is a challenging task due to the complexity of vascular systems, which are highly variating in shape, size, and structure. Existing model-based methods provide some degree of control and variation in the structures produced, but fail to capture the diversity of actual anatomical data. We developed VesselVAE, a recursive variational Neural Network that fully exploits the hierarchical organization of the vessel and learns a low-dimensional manifold encoding branch connectivity along with geometry features describing the target surface. After training, the VesselVAE latent space can be sampled to generate new vessel geometries. To the best of our knowledge, this work is the first to utilize this technique for synthesizing blood vessels. We achieve similarities of synthetic and real data for radius (.97), length (.95), and tortuosity (.96). By leveraging the power of deep neural networks, we generate 3D models of blood vessels that are both accurate and diverse, which is crucial for medical and surgical training, hemodynamic simulations, and many other purposes.

Keywords: Vascular 3D model · Generative modeling · Neural Networks

1 Introduction

Accurate 3D models of blood vessels are increasingly required for several purposes in Medicine and Science [25]. These meshes are typically generated using either image segmentation or synthetic methods. Despite significant advances in vessel segmentation [26], reconstructing thin features accurately from medical images remains challenging [2]. Manual editing of vessel geometry is a tedious and error prone task that requires expert medical knowledge, which explains the

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43907-0_7.

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14220, pp. 67–76, 2023.
https://doi.org/10.1007/978-3-031-43907-0_7

scarcity of curated datasets. As a result, several methods have been developed to adequately synthesize blood vessel geometry [29].

Within the existing literature on generating vascular 3D models, we identified two primary types of algorithms: fractal-based, and space-filling algorithms. Fractal-based algorithms use a set of fixed rules that include different branching parameters, such as the ratio of asymmetry in arterial bifurcations and the relationship between the diameter of the vessel and the flow [7,33]. On the other hand, space-filling algorithms allow the blood vessels to grow into a specific perfusion volume while aligning with hemodynamic laws and constraints on the formation of blood vessels [9,17,21,22,25]. Although these *model-based* methods provide some degree of control and variation in the structures produced, they often fail to capture the diversity of real anatomical data.

In recent years, deep neural networks led to the development of powerful generative models [30], such as Generative Adversarial Networks [8,12] and Diffusion Models [11], which produced groundbreaking performance in many applications, ranging from image and video synthesis to molecular design. These advances have inspired the creation of novel network architectures to model 3D shapes using voxel representations [28], point clouds [31], signed distance functions [19], and polygonal meshes [18]. In particular, and close to our aim, Wolterink et al. [27] propose a GAN model capable of generating coronary artery anatomies. However, this model is limited to generating single-channel blood vessels and thus does not support the generation of more complex, tree-like vessel topologies.

In this work we propose a novel *data-driven* framework named VesselVAE for synthesizing blood vessel geometry. Our generative framework is based on a Recursive variational Neural Network (RvNN), that has been applied in various contexts, including natural language [23,24], shape semantics modeling [14,15], and document layout generation [20]. In contrast to previous data-driven methods, our recursive network fully exploits the hierarchical organization of the vessel and learns a low-dimensional manifold encoding branch connectivity along with geometry features describing the target surface. Once trained, the VesselVAE latent space is sampled to generate new vessel geometries. To the best of our knowledge, this work is the first to synthesize multi-branch blood vessel trees by learning from real data. Experiments show that synth and real blood vessel geometries are highly similar measured with the cosine similarity: radius (.97), length (.95), and tortuosity (.96).

2 Methods

Input. The network input is a binary tree representation of the blood vessel 3D geometry. Formally, each tree is defined as a tuple $(T, \mathcal{E})$, where T is the set of nodes, and $\mathcal{E}$ is the set of directed edges connecting a pair of nodes (n, m), with $n, m \in T$. In order to encode a 3D model into this representation, vessel segments V are parameterized by a central axis consisting of ordered points in Euclidean space: $V = v_1, v_2, \ldots, v_N$ and a radius r, assuming a piece-wise tubular vessel for simplicity. We then construct the binary tree as a set of nodes $T = n_1, n_2, \ldots, n_N$,

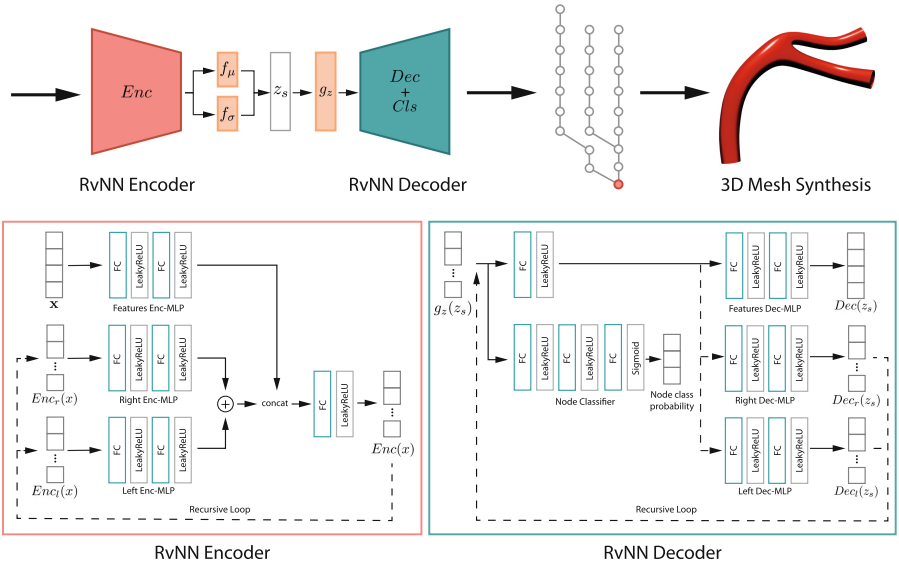

Fig. 1. Top: Overview of the Recursive variational Neural Network for synthesizing blood vessel structures. The architecture follows an Encoder-Decoder framework which can handle the hierarchical tree representation of the vessels. VesselVAE learns to generate the topology and attributes for each node in the tree, which is then used to synthesize 3D meshes. Bottom: Layers of the Encoder and Decoder networks comprising branches of fully-connected layers followed by leaky ReLU activations. Note that the right/left Enc-MLPs within the Encoder are triggered respectively when the incoming node in the tree is identified as a right or left child. Similarly, the Decoder only uses right/left Dec-MLPs when the Node Classifier predicts bifurcations.

where each node n_i represents a vessel segment v and contains an attribute vector $\mathbf{x}_i = [x_i, y_i, z_i, r_i] \in \mathbb{R}^4$ with the coordinates of the corresponding point and its radius r_i. See Sect. 3 for details.

Network Architecture. The proposed generative model is a Recursive variational Neural Network (RvNN) consisting of two main components: the Encoder (Enc) and the Decoder (Dec) networks. The Encoder transforms a tree structure into a hierarchical encoding on the learned manifold. The Decoder network is capable of sampling from this encoded space to decode tree structures, as depicted in Fig. 1. The encoding and decoding processes are achieved through a depth-first traversal of the tree, where each node is combined with its parent node recursively. The model outputs a hierarchy of vessel branches, where each internal node in the hierarchy is represented by a vector that encodes its own attributes and the information of all subsequent nodes in the tree.

Within the RvNN Decoder network there are two essential components: the Node Classifier (Cls) and the Features Decoder Multi-Layer Perceptron (Features Dec-MLP). The Node Classifier discerns the type of node to be decoded, whether it is a leaf node or an internal node with one or two bifurcations. This

is implemented as a multi-layer perceptron trained to predict a three-category bifurcation probability based on the encoded vector as input. Complementing the Node Classifier, the Features Dec-MLP is responsible for reconstructing the attributes of each node, specifically its coordinates and radius. Furthermore, two additional components, the Right and Left Dec-MLP, are in charge of recursively decoding the next encoded node in the tree hierarchy. These decoder's branches execute based on the classifier prediction for that encoded node. If the Node Classifier predicts a single child for a node, a right child is assumed by default.

In addition to the core architecture, our model is further augmented with three auxiliary, shallow, fully-connected neural networks: f_μ, f_σ, and g_z. Positioned before the RvNN bottleneck, the f_μ and f_σ networks shape the distribution of the latent space where encoded tree structures lie. Conversely, the g_z network, situated after the bottleneck, facilitates the decoding of latent variables, aiding the Decoder network in the reconstruction of tree structures. Collectively, these supplementary networks streamline the data transformation process through the model. All activation functions used in our networks are leaky ReLUs. See the Appendix for implementation details.

Objective. Our generative model is trained to learn a probability distribution over the latent space that can be used to generate new blood vessel segments. After encoding, the decoder takes samples from a multivariate Gaussian distribution: $z_s(x) \sim N(\mu, \sigma)$ with $\mu = f_\mu(Enc(x))$ and $\sigma = f_\sigma(Enc(x))$, where Enc is the recursive encoder and f_μ, f_σ are two fully-connected neural networks. In order to recover the feature vectors $\mathbf{x}$ for each node along with the tree topology, we simultaneously train the regression network (Features Dec-MLP in Fig. 1) on a reconstruction objective L_{recon}, and the Node Classifier using L_{topo}. Additionally, in line with the general framework proposed by β-VAE [10], we incorporated a Kullback-Leibler (KL) divergence term encouraging the distribution $p(z_s(x))$ over all training samples x to move closer to the prior of the standard normal distribution $p(z)$. We therefore minimize the following equation:

$$L = L_{recon} + \alpha L_{topo} + \gamma L_{KL}, \tag{1}$$

where the reconstruction loss is defined as $L_{recon} = \|Dec\,(z_s(x)) - x\|_2$, the Kullback-Leibler divergence loss is $L_{\mathrm{KL}} = D_{\mathrm{KL}}\,(p\,(z_s(x))\,\|p(z))$, and the topology objective is a three-class cross entropy loss $L_{topo} = \Sigma_{c=1}^{3} x_c \log(Cls(Dec(x))_c)$. Notice that x_c is a binary indicator (0 or 1) for the true class of the sample x. Specifically, $x_c = 1$ if the sample belongs to class c and 0 otherwise. $Cls(Dec(x))_c$ is the predicted probability of the sample x belonging to class c (zero, one, or two bifurcations), as output by the classifier. Here, $Dec(x)$ denotes the encoded-decoded node representation of the input sample x.

3D Mesh Synthesis. Several algorithms have been proposed in the literature to generate a surface 3D mesh from a tree-structured centerline [29]. For simplicity and efficiency, we chose the approach described in [6], which produces good quality meshes from centerlines with a low sample rate. The implemented method iterates through the points in the curve generating a coarse quadrilateral

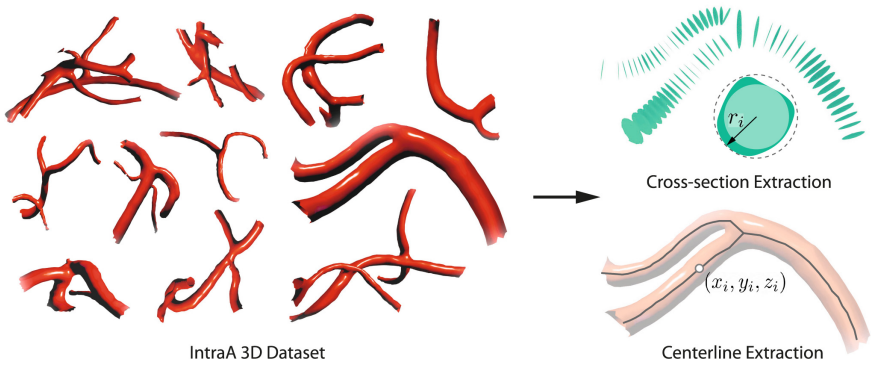

Fig. 2. Dataset and pre-processing overview: The raw meshes from the IntraA 3D collection undergo pre-processing using the VMTK toolkit. This step is crucial for extracting centerlines and cross-sections from the meshes, which are then used to construct their binary tree representations.

mesh along the segments and joints. The centerline sampling step is crucial for a successful reconstruction outcome. Thus, our re-sampling is not equispaced but rather changes with curvature and radius along the centerline, increasing the frequency of sampling near high-curvature regions. This results in a better quality and more accurate mesh. Finally, Catmull-Clark subdivision algorithm [5] is used to increase mesh resolution and smooth out the surface.

3 Experimental Setup

Materials. We trained our networks using a subset of the open-access IntrA dataset[1] published by Yang et al. in 2020 [32]. This subset consisted of 1694 healthy vessel segments reconstructed from 2D MRA images of patients. We converted 3D meshes into a binary tree representation and used the *network extraction* script from the VMTK toolkit[2] to extract the centerline coordinates of each vessel model. The centerline points were determined based on the ratio between the sphere step and the local maximum radius, which was computed using the advancement ratio specified by the user. The radius of the blood vessel conduit at each centerline sample was determined using the computed cross-sections assuming a maximal circular shape (See Fig. 2). To improve computational efficiency during recursive tree traversal, we implemented an algorithm that balances each tree by identifying a new root. We additionally trimmed trees to a depth of ten in our experiments. This decision reflects a balance between the computational demands of depth-first tree traversal in each training step and the complexity of the training meshes. We excluded from our study trees

[1] https://github.com/intra3d2019/IntrA.
[2] http://www.vmtk.org/vmtkscripts/vmtknetworkextraction.

that exhibited greater depth, nodes with more than two children, or with loops. However, non-binary trees can be converted into binary trees and it is possible to train with deeper trees at the expense of higher computational costs. Ultimately, we were able to obtain 700 binary trees from the original meshes using this approach.

Implementation Details. For the centerline extraction, we set the advancement ratio in the VMTK script to 1.05. The script can sometimes produce multiple cross-sections at centerline bifurcations. In those cases, we selected the sample with the lowest radius, which ensures proper alignment with the centerline principal direction. All attributes were normalized to a range of $[0, 1]$. For the mesh reconstruction we used 4 iterations of Catmull-Clark subdivision algorithm. The data pre-processing pipeline and network code were implemented in Python and PyTorch Framework.

Training. In all stages, we set the batch size to 10 and used the ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 1×10^{-4}. We set $\alpha = .3$ and $\gamma = .001$ for Eq. 1 in our experiments. To enhance computation speed, we implemented dynamic batching [16], which groups together operations involving input trees of dissimilar shapes and different nodes within a single input graph. It takes approximately 12 h to train our models on a workstation equipped with an NVIDIA A100 GPU, 80 GB VRAM, and 256 GB RAM. However, the memory footprint during training is very small ($\leq$1 GB) due to the use of a lightweight tree representation. This means that the amount of memory required to store and manipulate our training data structures is minimal. During training, we ensure that the reconstructed tree aligns with the original structure, rather than relying solely on the classifier's predictions. We train the classifier using a cross-entropy loss that compares its predictions to the actual values from the original tree. Since the number of nodes in each class is unbalanced, we scale the weight given to each class in the cross-entropy loss using the inverse of each class count. During preliminary experiments, we observed that accurately classifying nodes closer to the tree root is critical. This is because a miss-classification of top nodes has a cascading effect on all subsequent nodes in the tree (i.e. skip reconstructing a branch). To account for this, we introduce a weighting scheme that for each node, assigns a weight to the cross-entropy loss based on the number of total child nodes. The weight is normalized by the total number of nodes in the tree.

Metrics. We defined a set of metrics to evaluate our trained network's performance. By using these metrics, we can determine how well the generated 3D models of blood vessels match the original dataset distribution, as well as the diversity of the generated output. The chosen metrics have been widely used in the field of blood vessel 3D modeling, and have shown to provide reliable and accurate quantification of blood vessels main characteristics [3,13]. We analyzed tortuosity per branch, the vessel centerline total length, and the average radius of the tree. Tortuosity distance metric [4] is a widely used metric in the field of blood vessel analysis, mainly because of its clinical importance. It measures the amount of twistiness in each branch of the vessel. Vessel's total length and

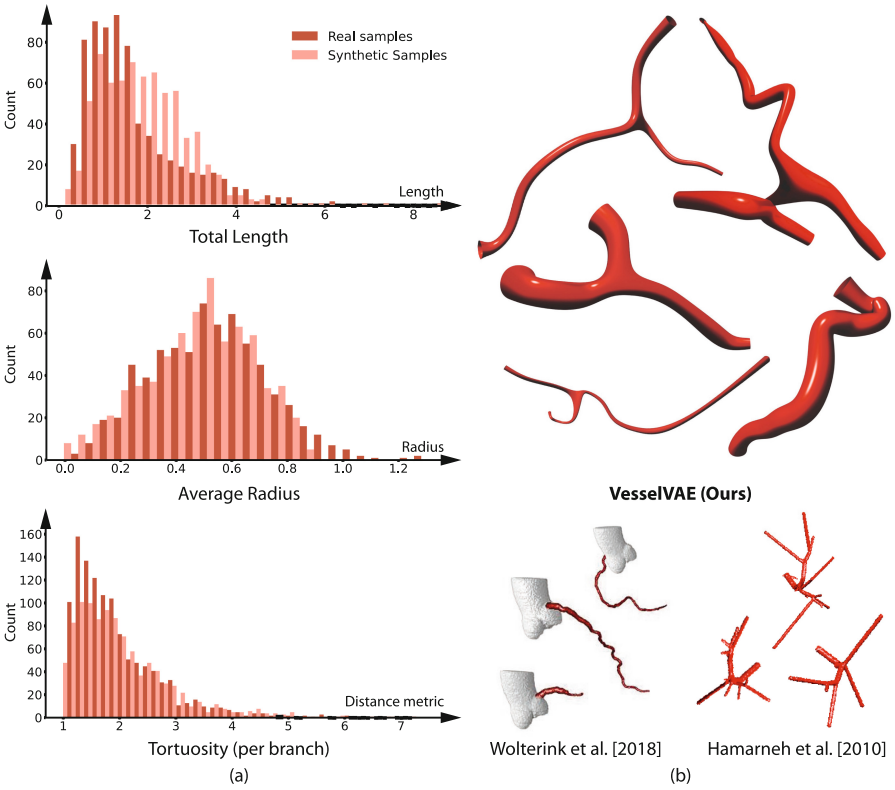

Fig. 3. (a) shows the histograms of total length, average radius and tortuosity per branch for both, real and synthetic samples. (b) shows a visual comparison among our method and two baselines [9,27].

average radius were used in previous work to distinguish healthy vasculature from cancerous malformations. Finally, in order to measure the distance across distributions for each metric, we compute the cosine similarity.

4 Results

We conducted both quantitative and qualitative analyses to evaluate the model's performance. For the quantitative analyses, we implemented a set of metrics commonly used for characterizing blood vessels. We computed histograms of the radius, total length, and tortuosity for the real blood vessel set and the generated set (700 samples) in Fig. 3 (a). The distributions are aligned and consistent. We measured the closeness of histograms with the cosine similarity by projecting the distribution into a vector of n-dimensional space (n is the number of bins in the histogram). Since our points are positive, the results range from 0 to 1. We obtain a radius cosine similarity of .97, a total length cosine similarity of .95,

and a tortuosity cosine similarity of .96. Results show high similarities between histograms demonstrating that generated blood vessels are realistic. Given the differences with the baselines generated topologies, for a fair comparison, we limited our evaluation to a visual inspection of the meshes.

The qualitative analyses consisted of a visual evaluation of the reconstructed outputs provided by the decoder network. We visually compared them to state-of-the-art methods in Fig. 3 (b). The method described by Wolterink and colleagues [27] is able to generate realistic blood vessels but without branches, and the method described by Hamarneh et al. [9] is capable of generating branches with straight shapes, missing on realistic modeling. In contrast, our method is capable of generating realistic blood vessels containing branches, with smooth varying radius, lengths, and tortuosity.

5 Conclusions

We have presented a novel approach for synthesizing blood vessel models using a variational recursive autoencoder. Our method enables efficient encoding and decoding of binary tree structures, and produces high-quality synthesized models. In the future, we aim to explore combinations of our approach with representing surfaces by the zero level set in a differentiable implicit neural representation (INR) [1]. This could lead to more accurate and efficient modeling of blood vessels and potentially other non-tree-like structures such as capillary networks. Since the presented framework would require significant adaptations to accommodate such complex topologies, exploring this problem would certainly be an interesting direction for future research. Additionally, the generated geometries might show self-intersections. In the future, we would like to incorporate restrictions into the generative model to avoid such artifacts. Overall, we believe that our proposed approach holds great promise for advancing 3D blood vessel geometry synthesis and contributing to the development of new clinical tools for healthcare professionals.

Acknowledgements. This project was supported by grants from Salesforce, USA (Einstein AI 2020), National Scientific and Technical Research Council (CONICET), Argentina (PIP 2021-2023 GI - 11220200102981CO), and Universidad Torcuato Di Tella, Argentina.

References

1. Alblas, D., Brune, C., Yeung, K.K., Wolterink, J.M.: Going off-grid: continuous implicit neural representations for 3D vascular modeling. In: Camara, O., et al. (eds.) STACOM 2022. LNCS, vol. 13593, pp. 79–90. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-23443-9_8
2. Alblas, D., Brune, C., Wolterink, J.M.: Deep learning-based carotid artery vessel wall segmentation in black-blood MRI using anatomical priors. arXiv preprint arXiv:2112.01137 (2021)

3. Bullitt, E., et al.: Vascular attributes and malignant brain tumors. In: Ellis, R.E., Peters, T.M. (eds.) MICCAI 2003. LNCS, vol. 2878, pp. 671–679. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39899-8_82
4. Bullitt, E., Gerig, G., Pizer, S.M., Lin, W., Aylward, S.R.: Measuring tortuosity of the intracerebral vasculature from MRA images. IEEE Trans. Med. Imaging **22**(9), 1163–1171 (2003)
5. Catmull, E., Clark, J.: Recursively generated b-spline surfaces on arbitrary topological meshes. Comput. Aided Des. **10**(6), 350–355 (1978)
6. Felkel, P., Wegenkittl, R., Buhler, K.: Surface models of tube trees. In: Proceedings Computer Graphics International, pp. 70–77. IEEE (2004)
7. Galarreta-Valverde, M.A., Macedo, M.M., Mekkaoui, C., Jackowski, M.P.: Three-dimensional synthetic blood vessel generation using stochastic l-systems. In: Medical Imaging 2013: Image Processing, vol. 8669, pp. 414–419. SPIE (2013)
8. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
9. Hamarneh, G., Jassi, P.: Vascusynth: simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis. Comput. Med. Imaging Graph. **34**(8), 605–616 (2010)
10. Higgins, I., et al.: beta-VAE: learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (2017)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
12. Kazeminia, S., et al.: GANs for medical image analysis. Artif. Intell. Med. **109**, 101938 (2020)
13. Lang, S., et al.: Three-dimensional quantification of capillary networks in healthy and cancerous tissues of two mice. Microvasc. Res. **84**(3), 314–322 (2012)
14. Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L.: Grass: generative recursive autoencoders for shape structures. ACM Trans. Graph. (TOG) **36**(4), 1–14 (2017)
15. Li, M., et al.: Grains: generative recursive autoencoders for indoor scenes. ACM Trans. Graph. (TOG) **38**(2), 1–16 (2019)
16. Looks, M., Herreshoff, M., Hutchins, D., Norvig, P.: Deep learning with dynamic computation graphs. arXiv preprint arXiv:1702.02181 (2017)
17. Merrem, A., Bartzsch, S., Laissue, J., Oelfke, U.: Computational modelling of the cerebral cortical microvasculature: effect of x-ray microbeams versus broad beam irradiation. Phys. Med. Biol. **62**(10), 3902 (2017)
18. Nash, C., Ganin, Y., Eslami, S.A., Battaglia, P.: Polygen: an autoregressive generative model of 3D meshes. In: International Conference on Machine Learning, pp. 7220–7229. PMLR (2020)
19. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 165–174 (2019)
20. Patil, A.G., Ben-Eliezer, O., Perel, O., Averbuch-Elor, H.: Read: recursive autoencoders for document layout generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 544–545 (2020)
21. Rauch, N., Harders, M.: Interactive synthesis of 3D geometries of blood vessels. In: Theisel, H., Wimmer, M. (eds.) Eurographics 2021 - Short Papers. The Eurographics Association (2021)

22. Schneider, M., Reichold, J., Weber, B., Székely, G., Hirsch, S.: Tissue metabolism driven arterial tree generation. Med. Image Anal. **16**(7), 1397–1414 (2012)
23. Socher, R.: Recursive deep learning for natural language processing and computer vision. Stanford University (2014)
24. Socher, R., Lin, C.C., Manning, C., Ng, A.Y.: Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 129–136 (2011)
25. Talou, G.D.M., Safaei, S., Hunter, P.J., Blanco, P.J.: Adaptive constrained constructive optimisation for complex vascularisation processes. Sci. Rep. **11**(1), 1–22 (2021)
26. Tetteh, G., et al.: Deepvesselnet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes. Front. Neurosci. 1285 (2020)
27. Wolterink, J.M., Leiner, T., Isgum, I.: Blood vessel geometry synthesis using generative adversarial networks. arXiv preprint arXiv:1804.04381 (2018)
28. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
29. Wu, J., Hu, Q., Ma, X.: Comparative study of surface modeling methods for vascular structures. Comput. Med. Imaging Graph. **37**(1), 4–14 (2013)
30. Xu, M., et al.: Generative AI-empowered simulation for autonomous driving in vehicular mixed reality metaverses. arXiv preprint arXiv:2302.08418 (2023)
31. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3D point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4541–4550 (2019)
32. Yang, X., Xia, D., Kin, T., Igarashi, T.: Intra: 3D intracranial aneurysm dataset for deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2656–2666 (2020)
33. Zamir, M.: Arterial branching within the confines of fractal l-system formalism. J. Gen. Physiol. **118**(3), 267–276 (2001)