Vesselformer: Towards Large Scale Complete 3D Vessel Graph Generation from Images

No Author Given

No Institute Given

Abstract. Image-to-graph appears in many problems, especially vessel graph extraction from medical and biological images. Traditionally, this problem is tackled by segmentation followed by skeletonization. However, the ambiguity in skeletonization makes it hard to get a compact yet faithful graph representation out of it. Recently, Relationforme proposed an end-to-end solution to extract graphs directly from images. However, they did not consider edge features, particularly radius information, which is crucial in many applications such as flow simulation. Further, their method does only patch-based graph prediction. In this work, we predict the radius with the help of a radius token. Second, we propose an efficient algorithm to infer a large 3D graph from patch inference. Finally, we show experimental results in 3 datasets for the first 3D complete graph prediction.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

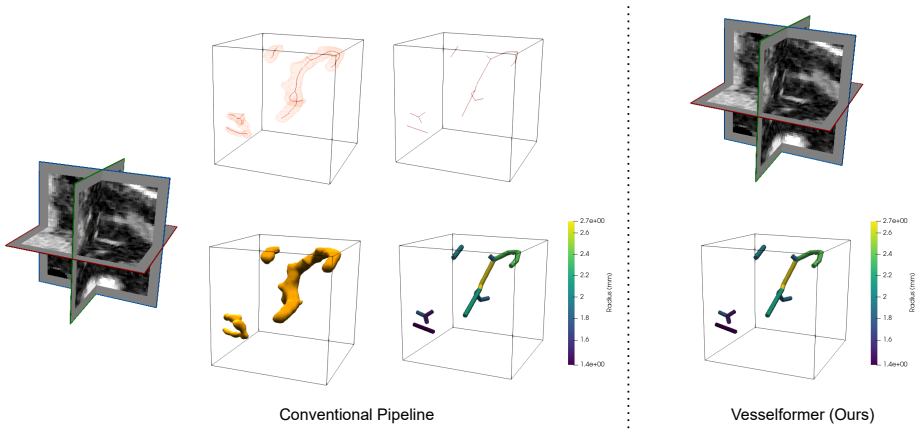

Fig. 1. Vesselformer pipeline

Extracting graphs of tubular structures is an important task in medical imaging and pre-clinical results. This involves identifying underlying graph structures, such as vessel graphs and neuronal graphs from various image modalities. This image-to-graph

extraction is a crucial task For a variety of downstream applications. For example, simulating blood flow in vessels requires complete graph information. In case of blood vessels, the graph usually consists of nodes, which denotes bifurcation points or vessel location with significant curvature and edges corresponds to the vessel segments. Traditionally the image-to-graph problem is broken down into a handful of sub-problems. These problems are typically vessel segmentation, center line extraction, graph pruning, and feature extraction. Not only does this whole pipeline involve many stages it also relies on multiple hand-engineered heuristics. Father, these approaches break the end-to-end differentiability of downstream tasks relying on the inferred graph. Hence a learnable graph extractor is needed to bridge the gap.

In computer vision, image-to-graph appears in different flavors. the graph structures instead of being structural, as in medical imaging semantics. A recent method, namely Relationformer [7], unified these two categories and proposed a general image-to-graph model. Relationformer is a transformer-based architecture that operates end-to-end from image to graph. This one-stage model comes with simplicity and explainability. However, Relationformer only predicts the structural graph without any features associated with the graph. Note that edge features, radius in particular, are of paramount importance for vessel applications. The radius information is absolutely Indispensable for blood flow simulation [8], and vessel graph labeling [9]. Additionally, Relationformer generates graphs from small image patches. In application scenarios, a complete graph, however, is needed in order to facilitate the downstream tasks.

To mitigate these challenges, we propose Vesselformer, a transformer-based model that can predict vessel graphs with edge properties such as radius. We achieve this by learning dedicated representation of edge features in addition to node-to-node similarity measures. Second, we propose an efficient algorithm to combine graphs extracted from image patches into a complete graph. We validated our method in three data sets to demonstrate the efficacy of our proposed method.

2 Related Literature

Despite significant progress in image acquisition and segmentation of structure of interest thereafter, only a limited number of available methods exist to extract graphs and processes. Note that all of them are based on hand-engineered rules, and no fully automated solution exists so far. We will discuss a few prominent ones in this section. TubeTK [1] is an open-access tool, but this only produces vessel segments as tubes and not a complete graph out f it. Vessegraph [6] uses Voreen [5], which extracts local properties in each point of centerlines and produces a graph. This, however, reduces to a metric graph, which loses important structural information, such as the twist and curvature of vessels. On a similar note, VesselVio [2] proposes a processing pipeline to extract graphs from centerlines. The common problem all skeleton-based method inherits is dense centerline graphs, and there is no good way to prune the graph into a compact yet structurally faithful representation that can be seamlessly transferred to the downstream tasks. In this spirit, our objective is to learn where to place nodes in image location and how to connect two nodes in data driven fashion.

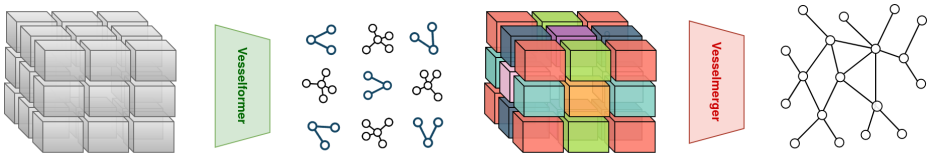

Fig. 2. Vesselformer pipeline

Graph extraction from images is a long-standing problem in computer vision that ranges from road network extraction [10, 3] or scene-graph extraction [4]. Most of these modern computer vision image to graph extractors are based on transformer, which leverages the set-prediction formulation to swiftly transit from image representation to discrete graph representation. Note that all these methods deal with 2D images and are not equipped to tackle increased computational complexity in 3D.

3 Methodology

In this section we will first briefly describe problem statement below. Subsequently, we discuss recent set-based prediction based on transformer, specifically the recent variant namely Relationformer. Next, we will elaborate on our proposed Vesselformer framework. Finally, we will explain our proposed graph merging algorithm namely Vesselmerger.

Problem Statement Given an image $I \in \mathbb{R}^{H \times W \times D \times C}$, an image-to-graph task is to predict $\mathcal{G}$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a graph with vertices (or objects) $\mathcal{V}$ and edges (or relations) $\mathcal{E}$. Specifically, the ith vertex $v^i \in \mathcal{V}$ has a node or object location specified by a bounding box $\boldsymbol{v}^i_{\text{box}} \in \mathbb{R}^6$ and each edge $e^{ij} \in \mathcal{E}$ has an edge features $e^{ij}_{\text{feat}} \in \mathbb{R}$.

3.1 Background

Relationformer is a set-based predictor that is built upon Deformable DETR [11]. Relationformer relies upon a backbone feature extractor to obtain image tokens. The image tokens are then processed with a series of multi-head self-attention layers. The resultant contextualized image tokens transfers image information to learnable obj-tokens and rln-token through cross attention. The refined obj-tokens produces the node coordinates and the tuple of two obj-token and rln-token produce the final edges through two separate head consist of 3 layers MLP.

3.2 Vesselformer

Vesselformer originates with the aim of mitigating the shortcomings of Relationformer. A key purpose is to integrate edge attributes such as radius prediction at the same time. Note that, unlike edge prediction, edge attributes require looking into the image features more thoroughly because radius information is implicit in terms of vessel thickness.

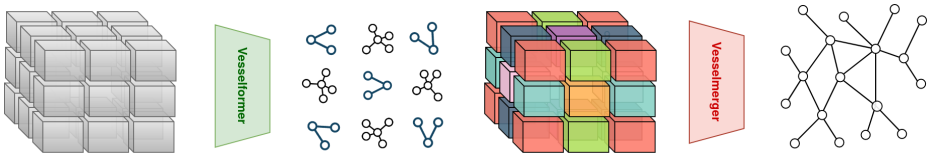

Fig. 3. Vesselformer architecture

Further, edge classification requires solving an affinity measure between two nodes, while radius information is an actual physical quantity. Hence we argue that the edge representation and edge attribute representation lie in different embedding spaces. With this intuition, we propose the following Vesselformer architecture.

Backbone: We use a 3D fully-convolutional network to extract features from the input image. The final features are flattened to obtain the input sequence for the transformer. A positional embedding is added before feeding the sequence to the transformer.

Transformer: We use a transformer encoder-decoder architecture with deformable attention [11], which considerably speeds up the training convergence of DETR by exploiting spatial sparsity of the image features. Our encoder remains unchanged and uses multi-scale deformable self-attention. In the decoder, we introduce a new learnable token in addition to [obj]-tokens and [rln]-token, namely [rad]-token.

Object Head: Object detection head consists of two components; one to predict the node location and the other to classify whether the node is valid or background. Additionally, we consider a hypothetical object bounding box around the node coordinate. The classification head is trained with cross-entropy loss, while the location head is trained with l1 regression and generalized IoU losses.

Relation Head: The relation head is similar to Relationformer. This takes the tuple of two object tokens concatenated with the relation token as input and classifies whether an edge exists between two nodes. This head is trained with cross-entropy loss with stochastically sampled edges.

Radius Head: The radius head is a three-layer MLP, which takes concatenated features of two obj-token and the rad-token and predicts the radius. This head is trained with and l1 regression loss.

3.3 Vesselmerger

Here we address the second shortcoming of Relationformer and propose Vesselmerger in the hope of scaling up the image-to-graph models from patch to image size. In image segmentation, this task is trivial because the prediction lies in the same regular grid. In graph space, this does not hold because the node coordinates lie in real numbers instead of integers. Further, the overlapping patch is easy to combine in the case of images. For

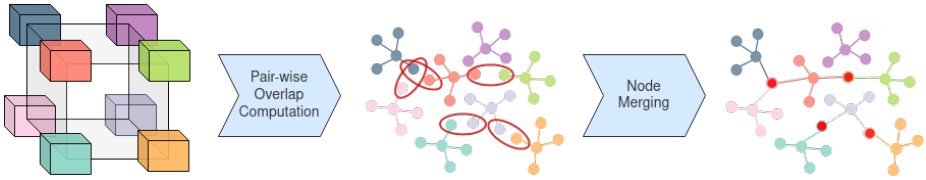

Fig. 4. Vesselmerger

graphs which has an overlap, is hard to combine because it requires a subgraph matching problem. To avoid this extra computation, we stick to non-overlapping patches.

Algorithm 1: Vesselmerger

Input Image: $I | I \in \mathbb{R}^{H \times W \times D}$
Input Parameters: w_p, τ, m
Output Graph: $\mathcal{G}$
$\{I_1, I_2, \cdots, I_n\} \leftarrow$ Patchify(I, w_p);
$\{l_1, l_2, \cdots, l_n\} \leftarrow$ Patchlabel$(\{I_1, I_2, \cdots, I_n\})$;
$\{\mathcal{G}_1, \mathcal{G}_2, \cdots \mathcal{G}_n\} \leftarrow$ Vesselformer$(\{I_1, I_2, \cdots, I_n\})$;
/* Initialize variables */
$M_{\text{occ}} \leftarrow \mathbf{0}_{8 \times H \times W \times D}$;
$c \leftarrow 0$;
$\mathcal{G} \leftarrow \phi$;
/* Concatenate predicted graphs */
for $i = 0$ **to** n **do**
  $\mathcal{V}_i, \mathcal{E}_i \leftarrow \mathcal{G}_i$;
  $L \leftarrow$ Nodelabel$(\mathcal{V}_i) + c$;
  $M_{\text{occ}}[l_i, \mathcal{V}_i] \leftarrow$ Greydilation(L, m) ;
  $\mathcal{G} \leftarrow$ Gather$(\mathcal{G}, \mathcal{V}_i, \mathcal{E}_i, c)$;
  $c \leftarrow c + |\mathcal{V}_i|$;
end
/* Merge nodes from occupancy */
for $(i, j) \in {}^8C_2$ **do**
  $M_1, M_2 \leftarrow M_{\text{occ}}[l_i], M_{\text{occ}}[l_j]$;
  $S, l_{1,2} \leftarrow$ IoU(M_1, M_2) ;
  $\mathcal{E} \leftarrow$ Threshold$(S, l_{1,2}, \tau)$;
  $\mathcal{G} \leftarrow$ Gather$(\mathcal{G}, \mathcal{E})$;
end

We observe that a non-overlapping patch creates a checkerboard pattern in 3D. Importantly, this can be coloured using eight different colours. This means adjacent patches will have different colours. This implies that glue our patch graph into a big graph, we need to compare adjacent patches instead of matching all nodes to all other nodes. Specifically, this means we can efficiently parallelize the pairwise patch merging of different colours. Further, we can merge patches at a logarithmic scale on the

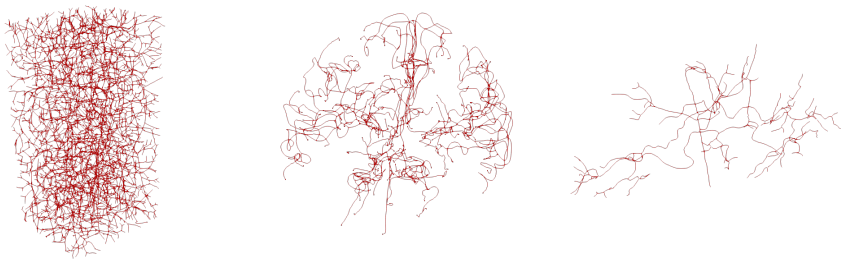

Fig. 5. Different datasets used

Table 1. Main results

Dataset	Model	Graph	Node Det.		Edge Det.		Radius
		SMD ↓	mAP ↑	mAR ↑	mAP ↑	mAR ↑	MSE ↓
Synthetic Vessel	U-net[**?**]+heuristics	0.01982	18.94	29.81	17.88	27.63	-
	Relationformer	**0.01107**	**78.51**	**84.34**	**78.10**	**82.15**	N.A.
	Vesselformer	0.01260	76.56	82.83	76.04	80.65	0.514
TubeTK	U-net[**?**]+heuristics	-	-	-	-	-	
	Relationformer	-	-	-	-	-	N.A.
	Vesselformer	0.01816	12.21	23.65	7.73	18.57	0.997
Peripancreatic	U-net[**?**]+heuristics	-	-	-	-	-	
	Relationformer	-	-	-	-	-	N.A.
	Vesselformer	0.01923	14.12	25.51	9.26	20.95	0.957

image size by divide and conquer policy which further accelerates the node merging. To compute the node proximity for a pair of nodes from two different colours, we leverage efficient voxelwise region growth. Subsequently, we compute the IoU of the node regions and glued the nodes together, which has an overlap.

This algorithm is shown and explained in Fig and the algorithm. The complexity id $\mathcal{O}(pk^2 \log n)$ where k is the number of nodes in a patch, p is patch size, and n is the image size.

4 Experiments

4.1 Datasets

4.2 Evaluation metrics

4.3 Main Results

Patch Level

Table 2. Ablation for Vesselformer

Model	# rad-token	Graph	Node Det.		Edge Det.		Radius
		SMD ↓	mAP ↑	mAR ↑	mAP ↑	mAR ↑	MSE ↓
Vesselformer	0	0.01283	64.42	73.86	65.11	73.22	0.580
	1	0.01260	76.56	82.83	76.04	80.65	0.514
	2	0.01261	76.12	82.29	75.48	80.25	0.510

Table 3. Sensitivity analysis of Vesselmerger

Model	Noise Level	Graph	Node Det.		Edge Det.	
		SMD ↓	mAP ↑	mAR ↑	mAP ↑	mAR ↑
Vesselmerger	$\sigma = 1$	-	-	-	-	-
	$\sigma = 2$	-	-	-	-	-
	$\sigma = 5$	-	-	-	-	-

Image Level

4.4 Ablation & Sensitivity Analysis

4.5 Explainability

5 Conclusion

References

1. Aylward, S.R., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE transactions on medical imaging **21**(2), 61–75 (2002)
2. Bumgarner, J.R., Nelson, R.J.: Open-source analysis and visualization of segmented vasculature datasets with vesselvio. Cell reports methods **2**(4), 100189 (2022)
3. Can, Y.B., Liniger, A., Paudel, D.P., Van Gool, L.: Topology preserving local road network estimation from single onboard camera image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17263–17272 (2022)
4. Koner, R., Shit, S., Tresp, V.: Relation transformer network. arXiv preprint arXiv:2004.06193 (2020)
5. Meyer-Spradow, J., Ropinski, T., Mensmann, J., Hinrichs, K.: Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. IEEE Computer Graphics and Applications **29**(6), 6–13 (2009)
6. Paetzold, J.C., McGinnis, J., Shit, S., Ezhov, I., Büschl, P., et al.: Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021)
7. Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., et al.: Relationformer: A unified framework for image-to-graph generation. In: European Conference on Computer Vision (ECCV). pp. 422–439. Springer Nature (2022)
8. Shit, S., Prabhakar, C., Paetzold, J.C., Menten, M.J., Wittmann, B., Ezhov, I., et al.: Graflow: Neural blood flow solver for vascular graph. In: Geometric Deep Learning in Medical Image Analysis (Extended abstracts) (2022)
9. Sobisch, J., Bizjak, Z., Chien, A., Spiclin, Z.: Automated intracranial vessel labeling with learning boosted by vessel connectivity, radii and spatial context. In: Geometric Deep Learning in Medical Image Analysis (2022)
10. Xu, Z., Liu, Y., Sun, Y., Liu, M., Wang, L.: Rngdet++: Road network graph detection by transformer with instance segmentation and multi-scale features enhancement. arXiv preprint arXiv:2209.10150 (2022)
11. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021)