


GERoMe—a Method for Evaluating Stability of Graph Extraction Algorithms Without Ground Truth

DOMINIK DREES^{1,2}, AARON SCHERZINGER^{1,2}, AND XIAOYI JIANG^{1,2} , (Senior Member, IEEE)

¹Faculty of Mathematics and Computer Science, University of Münster, D-48149 Münster, Germany

²Cells in Motion Cluster of Excellence, University of Münster, D-48149 Münster, Germany

Corresponding author: Xiaoyi Jiang (xjiang@uni-muenster.de)

ABSTRACT The extraction of graph structures in Euclidean vector space is a topic of interest with applications in many fields, such as the analysis of vascular networks in the biomedical domain. While a number of approaches have been proposed to tackle the problem of graph extraction, a quantitative evaluation of those algorithms remains a challenging task: In many cases, manual generation of ground truth for real-world data is time-consuming, error-prone, and thus not feasible. While tools for generating synthetic datasets with corresponding ground truth exist, the resulting data often does not reflect the complexity that real-world scenarios show in morphology and topology. As a complementary or even alternative approach, we propose GERoMe, the graph extraction robustness measure, which provides a means of quantifying the stability of algorithms that extract (multi-)graphs with associated node positions from non-graph structures. Our method takes edge-associated properties into consideration and does not necessarily require ground truth data, although available ground truth information can be incorporated to additionally evaluate the correctness of the graph extraction algorithm. We evaluate the behavior of the proposed graph similarity measure and demonstrate the usefulness and applicability of our method in an exemplary study on both synthetic and real-world data.

INDEX TERMS Evaluation, graph extraction, robustness, stability.

I. INTRODUCTION

Extracting graphs which are embedded in Euclidean vector space from different kinds of data (in particular, non graph-like structures) has been a topic of interest in various areas of research, especially with regard to biomedical applications. Here, researchers may be interested in the general structure and topology of the graph, the position of branching points, or specific (e.g., morphologic or geometric) properties of individual edges. Examples include the analysis of hepatic blood vasculature for surgical planning [1], measurement of airway trees [2], or the quantification of properties in lymphatic and blood vessel systems as a new approach to 3D histopathology for diagnosis in clinical applications [3]. In addition, the extracted graphs allow to make observation of structural or morphological changes in neural systems [4]. Registration of graph structures is an indispensable element of prognostic and diagnostic studies that require structural

analysis and comparison over time, among different samples, and to some gold standard [5], [6].

Besides the extraction of embedded graphs from 2D or 3D images, for which several approaches have been proposed [7]–[9], all of the aforementioned applications require the extraction of specific properties from individual vessel segments (i.e., sections without bifurcations in the vessel structure, represented by edges in the graph). Examples of such properties include geometrical features of vessel segments (e.g., *length* or *straightness*) as well as morphological features (e.g., *average radius* or *average roundness*) which are derived from the shape of the vessels' cross-section. These numerical edge-associated properties (usually a single scalar value for each individual property of an edge) can be extracted from the original dataset alongside the graph structure itself [10]–[12]. Such characteristics can also be determined after post-processing the extracted graph structures. For instance, the retinal vessel network can be further separated into arteries and veins, which is fundamental to computing the important artery–vein (A–V) caliber ratio [13].

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

While a number of existing algorithms produces plausible results with regard to a specific application domain, providing an objective evaluation of the quality of the extracted graph remains a challenging task. This is typically done by some form of comparison with manual ground truth [14]. Although manual ground truth generation is conceivable for the topological structure of the graph as well as the position of each node in some cases, this process needs to be carried out by domain experts and is both time-consuming and error-prone, especially for 3D structures such as complex vessel networks in biomedical imaging. Even more so, an accurate manual annotation of edge-associated properties that are implicitly provided in the original data (e.g., volume, average radius, or roundness of vessel segments) appears almost impossible in 3D, despite the fact that they can be derived computationally by a graph extraction algorithm. Although tools for producing synthetic datasets have been presented, they only provide a limited number of edge-associated properties [15]. Moreover, the complexity of the generated datasets does not compare to real-world data (see Section II).

A lot of works has been proposed for evaluating segmentation accuracy [14], [16], [17] for various image domains and segmentation tasks. Such works typically consider the issue of segmentation *accuracy* based on some form of comparison with manual ground truth. In addition to the accuracy there is, however, also the issue of segmentation *stability*. In the context of document image segmentation, for instance, a segmentation algorithm is considered stable [18] if it produces the same layout for all copies of the same document. One finding of that study was in fact that four state-of-the-art segmentation algorithms have a very poor stability. A recent survey of document segmentation algorithms [19] concludes that the stability of proposed algorithm is widely neglected in evaluation and is a problem in actual applications. This conclusion made on the task of document segmentation can be generalized to many other instances of image segmentation since there is a general lacking of stability studies. In this paper we consider this stability issue in the context of graph extraction algorithms.

Here we propose GERoMe, the **Graph Extraction Robustness Measure**, which provides a means of quantifying the stability of graph extraction algorithms on arbitrary (i.e., specifically including real-world) input data without requiring any ground truth information. Moreover, if ground truth information is available, it can either be used in combination with the introduced graph similarity measure to evaluate the correctness of an algorithm directly, or to extend the original procedure to evaluate the correctness in conjunction with the robustness of the algorithm. If the input to a robust algorithm is changed in a way that is semantically insignificant, but in terms of the execution relevant, the generated output should not change. In this context, we define *robustness* as the property of an algorithm to produce stable results under *a-posteriori reversible* transformations to the input dataset.

To evaluate a specific algorithm, our method generates a scalar robustness index for a given input dataset, a set of

transformations, and any edge-associated property. This is achieved by applying one of the transformations to the input data, and using the result to extract a graph, which is then retransformed to the original space. This graph is matched with a template graph directly extracted from the input. For each transformation, a similarity measure is computed based on the difference in features of matched edges and the quality of the matching itself. The similarities for all transformations are then combined to form the robustness index GERoMe. Our method does not require ground truth data for evaluating the robustness of an algorithm. However, if ground truth information is available, this data can be used as the template graph. In this case, the resulting GERoMe value does not only quantify the robustness of the examined algorithm, but also its accuracy.

The extracted graphs can be of arbitrary structure and may include multiple edges connecting two nodes (i.e., they may be multigraphs), and evaluation can be performed for arbitrary positive real-valued edge-associated properties. The input data of the considered graph extraction algorithm can be of arbitrary nature, as long as a geometric transformation can be applied to it. We demonstrate the applicability of our approach in an exemplary study using a preliminary version of the algorithm proposed in [11] on both artificial and real-world datasets. Additionally, we observe and evaluate the influence of various errors commonly produced by graph extraction algorithms on the proposed graph similarity measure.

Our contribution is twofold: We describe a general framework for evaluation of graph extraction algorithms that is especially useful in situations where no ground truth data is available (Section III-A). Additionally, we apply this framework to present a specific robustness measure with a graph similarity score tailored to a specific problem domain (Section III-B, Section III-C).

The remainder of this paper is structured as follows. In the following section we give an overview of related publications. Afterwards we provide an in-depth description of our proposed method. Finally, we evaluate the behavior of the introduced graph similarity measure and exemplarily apply the proposed graph robustness measure to an existing algorithm before discussing the results.

This paper is an extended version of the work previously published in [20]. We have extended our previous work by refinements to the method. In particular, we have removed an edge case by more precisely defining the requirements of properties, modified the edge distance used in the proposed similarity measure to properly match multi-edges with identical node positions, and replaced the suggested matching algorithm with a more efficient method. Furthermore, we have added an additional section evaluating the behavior of the introduced graph similarity measure in the presence of typical perturbations identified in other literature. Additionally, we have substantially expanded the discussion on prior research with respect to methods for evaluation of the specific case of vessel system analysis algorithms, graph matching,

as well as evaluation without ground truth information in general.

II. RELATED WORK

The most obvious validation strategy for graph extraction methods is the comparison of the extracted graphs to ground information for the corresponding datasets. However, due to the aforementioned difficulties, the generation of ground truth data is difficult for real-world data in many applications. One possibility to overcome this issue is the use of synthetic data for which ground truth information is automatically available. VasuSynth [15] is a tool for generating simulated 3D medical images of blood vasculature. In addition to the raw volume datasets the software provides ground truth data information, which includes a binary foreground segmentation, the generated graph (i.e., branching point positions as nodes and edges for connecting vessel segments), as well as information about the radius, length, and flow for each of the graph's edges. However, there are two major limitations when using synthetic data generated using VasuSynth. First, the resulting vessel networks always have a tree-like topology and thus do not include cycles or multiple edges connecting the same pair of nodes. Second, the approach only simulates images of blood vasculature where the generated vessels are of relatively simple morphology (e.g., vessels are relatively round). The generated data sets thus do not heavily challenge graph extraction algorithms in that regard. (This is also reflected in the results of our experimental study, see Section V-A.) Furthermore, VasuSynth has been integrated into a framework for validation of vessel segmentation algorithms [21] where it is used to simulate hepatic vasculature imaging data and to generate corresponding ground truth segmentations. Segmentation algorithms can be evaluated not only in terms of the differences in foreground segmentation when compared to the generated ground truth data, but also with regard to errors in graph-based metrics such as number of branches, branch length, branch volume, and branch diameter.

Drechsler and Laura [10] have developed a graph extraction method which is specifically designed for hepatic blood vasculature. In order to evaluate their algorithm quantitatively, they rotate and resample the original volume using various rotation angles. For each angle they extract a graph and make note of the number of generated nodes and edges. Afterwards, they visualize the results in a plot. This evaluation reveals that their algorithm is not rotation-invariant, although the authors note that an ideal algorithm should fulfill this requirement.

Mayerich *et al.* [22] have proposed two methods for quantitatively comparing a pair of input graphs. If one of the examined graphs is considered to be ground truth data, their tool can calculate false negative and false positive ratios both in terms of geometric differences and topological information, i.e., the connectivity of the graphs. However, at least the geometric method is based on the knowledge of paths connecting two bifurcation points and neither mode makes use of scalar

numeric properties of edges in the graph. Furthermore, for evaluating the performance of graph extraction algorithms using this method, a ground truth graph is required, which – as outlined earlier -- is often difficult to obtain.

Heumann and Wittum [23] propose using the tree-edit distance as a measure for quantifying morphological similarity of neurons. Their method incorporates additional information into the process by using numeric node labels, but is limited to tree-like structures. Moreover, their research is aimed at the comparison of neuronal structures, and would therefore require ground truth data for performance evaluation as well.

One important aspect of this paper is matching edges of two (multi-)graphs. Traditional graph matching, which aims to find a mapping between the nodes and edges of two graphs, is a current and popular research topic [24], [25]. In particular, for error-tolerant graph matching and similarity computation, the graph edit distance [26], which assigns costs to modification, insertion and deletion operations for edges and nodes and computes the modification sequence with the lowest cost, is a popular and general, but computationally expensive approach. The bipartite graph matching-based approximation by Riesen and Bunke [27] and Stauffer *et al.* [28] computes the distance (and the implied matching) by matching nodes with respect to their properties and the local surrounding structure. In contrast to [27] we instead employ a direct *edge* matching approach using both geometric and additional edge-associated information and avoid the need to explicitly formulate (application-specific) edge deletion or insertion costs (see Section III).

In many application domains unreliable, scarce or entirely unavailable ground truth information makes the quantitative evaluation and comparison of automatic data processing systems difficult or even impossible. In order to avoid relying on subjective evaluation and to improve the reliability of existing systems, there has been an interest in research towards ground truth-independent evaluation strategies in recent years. Lamiroy and Sun [29] reinterpret and compute precision and recall as a probabilistic measure of agreement with the consensus between binary classifiers in the pattern recognition context and thus do not require ground truth information. They observe that their modified measure can often reproduce classifier ranking, but (by its nature) is sensitive to collective bias. Lamiroy and Pierrot [30], present a statistical framework for assessing the risk of misranking an algorithm in the presence of error in the ground truth.

Spampinato *et al.* [31] define a number of application-specific features to describe tracks extracted from 2D or 3D data. Then they train a naive Bayes classifier using these features and known ground truth as well as artificially generated known bad tracks. The trained classifier is then used to estimate the performance of tracking algorithms on data without known ground truth. They observe that shape and appearance based features seem to be reasonably comparable between domains (tracking of fish and vehicles for example). On the other hand, Zhang *et al.* [32] evaluate tracking algorithms by, rather than comparing the result of the algorithm

with (unknown) ground truth, comparing the observation that was used to construct the result with a *simulated* observation using the result as a basis.

Reverse classification accuracy [33] is a way to estimate the quality of a segmentation operation on data without ground truth information. After training a model on data with available ground truth, it is applied to segment data in the actual application. This segmented data (without known ground truth) is then used to train a second model, for which the performance is measured on all datasets previously used to train the original method. The assumption is that if at least one dataset in the training set closely resembles the assessed dataset, the single dataset-model should perform as well as original model.

The measure introduced in our work does not try to approximate the accuracy of an algorithm, but instead evaluates the *stability* of an algorithm, which is a desirable property in many cases. As accurate algorithms are not necessarily robust, this method can even be used as an additional feature to test for in addition to accuracy if ground truth information is available. Indeed, in that case and if desired, the presented method can be used to compute a combined measure of accuracy and robustness.

III. METHOD

An embedded multigraph shall be defined as a tuple $G = (N, E)$ of a set of nodes $N \subset \mathbb{R}^n$ (where we assume nodes with the same spatial position to be identical) and a set of edges $E \subset (N \times N \times \mathbb{N})$. Edges (n_1, n_2, I) are defined by two nodes $n_1, n_2 \in N$ and a unique identifier $I \in \mathbb{N}$ (for the purpose of allowing multiple edges between the same pair of nodes). Additionally, all edges $e \in E$ have m associated positive real-valued properties $P_i(e) > 0$, $i \in \{1, \dots, m\}$.

A. THE GRAPH EXTRACTION ROBUSTNESS MEASURE

The graph extraction robustness measure (GERoMe), which will be denoted \mathcal{G} for the remainder of this paper, provides a stability measure for multigraph extraction algorithms. Conceptually, it describes a process which compares a template graph G_{tpl} to the result of the extraction algorithm \mathcal{A} applied to a transformed version of the input s . The kind of input data is arbitrary, but should allow for the extraction of a graph with associated spatial node positions. In practice this is often the case for 2D or 3D images, but also conceivable for other data, such as point cloud datasets. The template graph can either be given as ground truth G_{GT} , or – e.g., if ground truth information for the property of interest is not available -- extracted from the input dataset without applying any transformation, i.e., $G_{tpl} = \mathcal{A}(s)$. The input dataset is then transformed by a transformation T , and the result is used as input to the examined graph extraction algorithm, i.e. $\tilde{G} = (\mathcal{A} \circ T)(s)$. The extracted graph \tilde{G} is then transformed back into the original space using the inverse of T : $\tilde{G}' = T^{-1}(\tilde{G})$. The entire procedure can therefore be summarized as follows:

$$s \xrightarrow{\text{transform with } T} s' \xrightarrow{\text{extract with } \mathcal{A}} \tilde{G} \xrightarrow{\text{transform with } T^{-1}} \tilde{G}'$$

For a robust algorithm, the resulting graph \tilde{G}' should be similar to the template graph G_{tpl} for *any* T . Therefore, the measure \mathcal{G} is defined as the minimum similarity S_P (see Section III-B) over all elements T of a set of transformations \mathcal{T} for a given dataset. The entire process is illustrated in Figure 1. Hence, T must be an automorphism (and thus in particular be invertible) that can be applied to both the input dataset s and an extracted graph G' . Moreover, for a perfect extraction algorithm \mathcal{A}^* for the corresponding edges e and e' in $G = \mathcal{A}^*(s)$ and $G' = (T^{-1} \circ \mathcal{A}^* \circ T)(s)$ one should have $P(e) \approx P(e')$ for any property P . This can be illustrated by a simple example: If T includes a scaling operation (on s), and the information extracted via \mathcal{A}^* includes the distance between two nodes $P_{distance}$ for all edges, T^{-1} subsequently must scale $P_{distance}$ accordingly. For many properties in real-world applications, this is the case if the set of transformations \mathcal{T} is restricted to contain only rigid-body transformations.

More formally, given the parameters mentioned above, this procedure can be defined as follows:

$$\mathcal{G}_{s, \mathcal{T}, P}(\mathcal{A}) = \min_{T \in \mathcal{T}} S_P(G_{tpl}, (T^{-1} \circ \mathcal{A} \circ T)(s)) \quad (1)$$

It should be noted that $\mathcal{G}_{s, \mathcal{T}, P} \in [0, 1]$. A robust extraction algorithm \mathcal{A} will produce similar graphs regardless of any transformation $T \in \mathcal{T}$, yielding a GERoMe-value near the optimal value of 1. If ground truth information for P in form of a ground truth graph G_{GT} is available, \mathcal{G} also includes information about the accuracy of \mathcal{A} for $G_{tpl} = G_{GT}$. This follows directly from the common definition of accuracy and the robustness: An algorithm is accurate if it produces a graph that is similar to the ground truth graph of a dataset, i.e., if $S_P(G_{GT}, \mathcal{A}(s)) \approx 1$. If we add the identity \mathbb{I} to the set of transformations when quantifying the robustness of an algorithm using the ground truth graph, a high value for \mathcal{G} implies a high accuracy:

$$\mathcal{G}_{s, \mathcal{T} \cup \{\mathbb{I}\}, P}(\mathcal{A}) = \min_{T \in \mathcal{T} \cup \{\mathbb{I}\}} S_P(G_{GT}, (T^{-1} \circ \mathcal{A} \circ T)(s)) \quad (2)$$

$$= \min(\min_{T \in \mathcal{T}} S_P(G_{GT}, (T^{-1} \circ \mathcal{A} \circ T)(s)), S_P(G_{GT}, (\mathbb{I}^{-1} \circ \mathcal{A} \circ \mathbb{I})(s))) \quad (3)$$

$$= \min(\mathcal{G}_{s, \mathcal{T}, P}(\mathcal{A}), S_P(G_{GT}, \mathcal{A}(s))) \quad (4)$$

If no ground truth information G_{GT} is available, the accuracy of the algorithm cannot be computed. In this case, we set $G_{tpl} = \mathcal{A}(s)$ and only quantify the robustness of the algorithm.

B. GRAPH SIMILARITY

The first step in computing the similarity between two graphs in our method is to find correspondences in both graphs, i.e., to find a matching. Since we are interested in differences in edge-associated properties, and since nodes have an associated position and may be connected by multiple edges, it is essential (and also sufficient) to find a *matching* $M_{G_1, G_2} \subset E_1 \times E_2$ for two graphs $G_1 = (N_1, E_1)$, $G_2 = (N_2, E_2)$ which

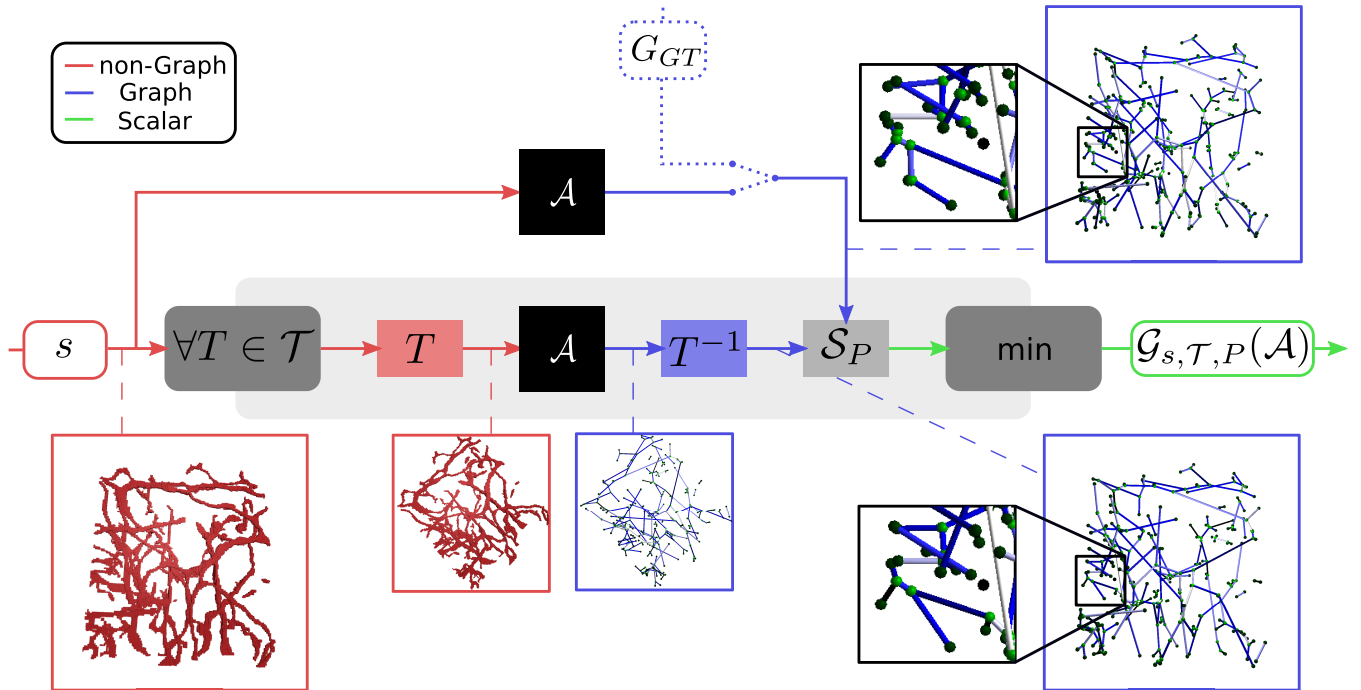


FIGURE 1. A schematic overview of the proposed method. \mathcal{T} is a set of transformations, P is an edge-associated property, \mathcal{A} is a graph extraction algorithm, s is a non-graph structure. Annotated images on the sides show intermediate results of the approach when applied to a preliminary version of the algorithm described by [11] and a lymphatic vessel foreground segmentation dataset [3].

matches edges in G_1 to edges in G_2 . The exact process used to find a matching is given in Section III-C. Note that not all of the edges in E_1 or E_2 necessarily have to be part of the matching, but that any edge in E_1 or E_2 can only be part of one pair in M_{G_1,G_2} .

$$M_{G_1,G_2} \subset E_1 \times E_2 \Rightarrow \forall e_1 \in E_1 : |\{(e_1, e) \in M_{G_1,G_2}\}| \leq 1 \wedge \forall e_2 \in E_2 : |\{(e, e_2) \in M_{G_1,G_2}\}| \leq 1 \quad (5)$$

Moreover, given a specific property $P > 0$ we define the relative error in terms of that property E_P of two edges e_1, e_2 as follows:

$$E_P(e_1, e_2) = \frac{|P(e_1) - P(e_2)|}{\max(P(e_1), P(e_2))} \in [0, 1] \quad (6)$$

Then, given a graph matching M_{G_1,G_2} and a property P , the relative error of a graph matching with respect to P can be defined using (6):

$$E_P(M_{G_1,G_2}) = \frac{1}{|M_{G_1,G_2}|} \sum_{(e_1,e_2) \in M} E_P(e_1, e_2) \quad (7)$$

However, $E_P(M_{G_1,G_2})$ ignores edges in the original graph that have not been matched. Therefore, in order to quantify the quality of a matching for two graphs G_1 and G_2 , we define the similarity (in terms of the property P) as follows:

$$S_P(G_1, G_2) = (1 - E_P(M_{G_1,G_2})) \cdot \frac{2|M_{G_1,G_2}|}{|E_1| + |E_2|} \quad (8)$$

The term $\frac{2|M_{G_1,G_2}|}{|E_1| + |E_2|}$, i.e., the edge match ratio, can be understood as the DICE index for $E_1 \cap E_2 := M_{G_1,G_2}$.

For $|E_1| = |E_2|$ the term simulates (arbitrarily) pairing all leftover (i.e., non-matched) edges while setting the relative error of all of these fake matches to 1.

C. MATCHING

A standard approach for error-tolerant graph matching is based on the graph edit distance [26]. As the exponential runtime complexity makes it infeasible to compute even for moderately sized graphs, an approximation based on bipartite graph matching has been developed [27]. While this method includes information about the edges in the form of local structure around nodes is included, the method focuses on the nodes, and implicitly only generates a node matching. At least the naive approach of constructing an edge matching from the node matching penalizes cases where an edge is disconnected near one of its nodes unnecessarily harshly (cf. Section IV-C). For these reasons we employ a direct edge matching approach that is similar to the method proposed in [27], which is only based on the distance d between two edges, and does use explicit insertion or deletion costs.

As a basis for d we first define d' which only relies on the spatial positions and Euclidean distances between the node positions of two edges. The distance d' is calculated by concatenating the nodes for both edges to form a $2n$ -dimensional vector, and computing the Euclidean distance. In comparison to a simple sum of node distances this punishes the matching of edges harder if they share one node but not the other. This of course is deliberate and inhibits matching of two distinct edges which share a node in favor of a (correct) matching of

slightly translated edges. Since the order of nodes within the definition of edges is arbitrary, the minimum distance of both unique node pairing permutations is denoted d' .

$$d'((n_1, n_2, I), (n'_1, n'_2, I')) = \min(\|(n_1 \circ n_2) - (n'_1 \circ n'_2)\|_2, \|(n_1 \circ n_2) - (n'_2 \circ n'_1)\|_2) \quad (9)$$

The distance d is then defined by increasing the spatial distance given by d' if the average of relative property errors (6) is large:

$$d(e_1, e_2) = \frac{d'(e_1, e_2) + \epsilon}{1 - \frac{1}{m} \sum_{i \in [1, m]} E_{P_i}(e_1, e_2)} \quad (10)$$

By incorporating the node information directly in the edge cost in this way, we avoid the need to formulate comparable costs for edges and nodes (in contrast to the graph edit distance-based approach). In order to omit false positive matches, we ignore all distances above a certain threshold t . The threshold is chosen to be equal to the $\frac{2 \cdot \min(|E_1|, |E_2|)}{|E_1| \cdot |E_2|}$ -quantile (i.e., the $2 \cdot \min(|E_1|, |E_2|)$ 'th smallest value) of the set of all possible $|E_1| \cdot |E_2|$ edge distances. In this way, obvious matches can still be found by the matching algorithm, while edges that do not have a correspondence in the other graph stay unmatched and do not skew the overall result by interfering with other matches in the search for a globally minimal sum of edge distances. In total, no more than $\min(|E_1|, |E_2|)$ matches can be found. However, in practice two edges connecting the same nodes will have similar distances to both corresponding edges in the other graph. Therefore, the $\min(|E_1|, |E_2|)$ 'th value cannot be a hard cutoff point. In order to include all likely match candidates the $2 \cdot \min(|E_1|, |E_2|)$ 'th smallest value is chosen as the threshold t . It should be noted that the threshold is designed for real-world applications such as the extraction of blood or lymphatic vasculature where only few pairs of nodes are connected by several edges. Extreme cases where a large percentage of nodes are connected by multiple edges may thus require a larger threshold.

The matching is computed by creating a bipartite matching graph for the sets of edges in both graphs (i.e., for E_1 and E_2). The *match-edges* (i.e., edges in the bipartite matching graph, not to be confused with edges *to* be matched from E_1 or E_2) are not weighted by the distance d , but by $d_{max} = t - d$. Additionally, all match-edges corresponding to a match with $d > t$ (and therefore match-edges with $d_{max} \leq 0$), are removed from the matching graph. The final matching is then computed by finding a maximum weight matching in the matching graph. It should be noted that (for $n = \min(|E_1|, |E_2|)$), as the number of edges in the matching graph is limited to $2 \cdot \min(|E_1|, |E_2|) \in \mathcal{O}(n)$, the total runtime of the matching procedure is within $\mathcal{O}(n^2 \log n)$ when using an efficient algorithm [34]. Additionally, by limiting the total number of possible matches, this approach (in contrast to graph edit distance-based algorithms) does not require the definition of insertion or deletion costs for nodes or edges.

The constant ϵ in (10) is a very small positive number which enables the differentiation of two pairs of edges connecting the same pair of nodes that spatially match up exactly in both graphs (i.e., $n_1 = n'_1 \wedge n_2 = n'_2$ or $n_1 = n'_2 \wedge n_2 = n'_1$), resulting in a spatial distance d' of 0. In a real-world scenario this is only required if the node positions are discrete, e.g., if nodes are positioned on a grid (and thus this scenario has a non-zero chance of occurring randomly). In that case a value for ϵ can be chosen so that either $d' = 0$ or $d' \gg \epsilon$. In the latter case, ϵ barely contributes to d and does not affect the result of the matching, while in the former case the value of d is defined solely by the average property error.

IV. EVALUATION

Since it is difficult to provide a quantitative evaluation of the entire proposed method, we restrict the evaluation to the introduced graph similarity measure \mathcal{S}_P . Instead of comparing the output of a graph extraction algorithm to a ground truth or template graph, in this section we take a graph G , and perturb it in a well-defined manner D_d by a degree $d \in [0, 1]$. The resulting graph $D_d(G)$ can then be compared to the original graph G by computing the similarity measure $\mathcal{S}_P(G, D_d(G))$. When performed for different perturbation degrees, this procedure illustrates the behavior of \mathcal{S}_P with regard to changes in the graph. The perturbation methods include geometrical, property-affecting, and topological procedures.

A. GEOMETRICAL PERTURBATION

To perform a geometrical perturbation of the input graph $G = (N, E)$, we shift the positions of each node $n \in N$ in a randomly selected direction. For each node, the amount of movement is sampled randomly from a normal distribution with a mean of 0 and a standard deviation of $2d$ multiplied by the average length of all edges in the graph.

B. PROPERTY PERTURBATION

In order to model error introduced to non-geometrical properties, we multiply the value of each property of each edge in the graph by a random value sampled from a log-normal distribution with $\sigma = d$ and $\mu = 0$. Thus, the transformed property values can be anywhere in the range of $(0, +\infty)$, but are still relatively close to the original value in most cases. Also, the modified property values are equally likely to be smaller or larger than the original value. A larger value of $d = \sigma$ increases the variance of the distribution and thus causes a larger property error on average.

C. TOPOLOGICAL PERTURBATION

Mayerich et al. [22] provide an overview of common topological errors in graph extraction algorithms and assign them to the following four categories. All four topological perturbation schemes in conjunction with the geometrical and property perturbations are also visualized schematically in Figure 2.

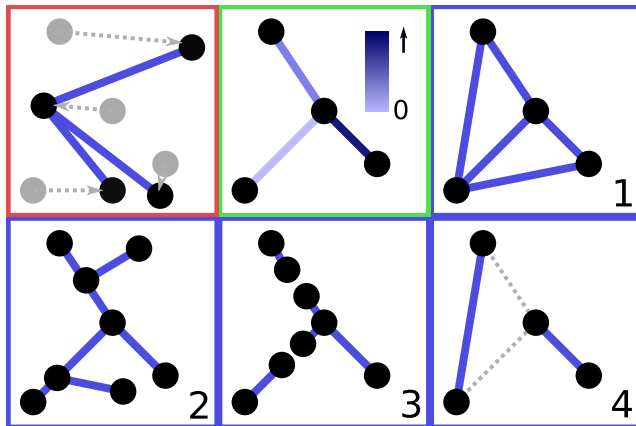


FIGURE 2. A schematic, exemplary overview of the used geometrical (red), property (green), and topological (blue) perturbation methods. The topological methods [22] include additional edges near branching points (1), edges subdivided by additional branches (2), split edges (3), and split nodes (4). The geometrical perturbation changes the positions of nodes while the property perturbation scales edge-associated property values up or down.

- 1) **Additional Edge:** An additional edge connects two nodes that are otherwise indirectly connected to each other via a bifurcation. Although this kind of error is most likely not introduced by graph extraction methods that operate on a binary volume, it may occur due to noisy branching points in the original volume either in a prior segmentation step or using graph extraction methods that operate on raw image data. The perturbation amount controls how many additional edges are added to the dataset so that for $d = 1$ the amount of edges within the graph is doubled and for $d = 0$ the graph is completely unchanged.
- 2) **Subdivided Edge:** An edge is subdivided by an additional node which is connected to an extra edge that is also added to the graph. Many existing skeletonization or graph extraction algorithms often introduce spurious edges. This is especially the case if the input data is noisy. In these cases the algorithm is likely to split an edge in half due to the added connection point (i.e., node in the graph). The perturbation amount controls the fraction of edges in the original graph that will be connected to artificial spurious edges and are thus subdivided. While this approach does not truly reflect reality where edges might be split multiple times, it makes the expected changes to the graph easier to reason about. Due to the nature of this perturbation, for the value $d = 1$ the number of edges in the graph is tripled as each original edge is replaced by the two components created by the subdivision plus the additional spurious edge.
- 3) **Split Edge:** An edge is split by removing a section along its run and thus inserting two additional nodes of degree 1 at the resulting stubs. This is an error that occurs most likely due to spurious signal along a vessel in the original volume, which produces a fragmentation of a segment, i.e., an edge of the graph.

The perturbation amount controls the fraction of edges in the original graph to be split. Similar to the previous perturbation, each edge is only split once, which does not reflect reality, but allows for simpler theoretical reasoning.

- 4) **Split Node:** Two edges are split off from a branching point (the node to be split) and replaced by a single edge. In a four-way branch this results in the removal of the entire node while in a three-way branch the original node remains, but its degree is changed to 1. If the node has a degree higher than four, the degree of the split node is reduced by two. Similar to the case of additional edges, nodes are most likely split due to bad initial imaging conditions around bifurcations. The perturbation amount controls the fraction of nodes with degree of two or more that will be perturbed. It should be noted that in actual applications on real-world data, there may occur even more complex scenarios where nodes are split into multiple components forming connected or unconnected subgraphs.

For artificial edges added in the context of one of the above perturbation schemes, properties that are not implicitly defined by the geometry itself are sampled randomly from the distribution of edge properties in the graph.

D. OBSERVATIONS

The perturbations listed above were applied to a graph G extracted from a real-world lymphatic vessel dataset. For comparison of similarities we chose the average roundness m (defined as the minimum radius divided by the maximum radius of the vessel, $m \in [0, 1]$) as the property of interest. By choosing a morphological property for comparison we ensure that only property perturbations affect the property value of edges, but geometrical or topological perturbations do not. For each perturbation type, 100 values of d were spread evenly in the range $[0, 1]$ and the resulting similarity was computed for each value of d . The simulations were performed 10 times for each perturbation type and value of d .

As illustrated in figure Figure 3a, small values of d (i.e., insignificant displacements) do not affect the similarity between the original and the perturbed graph. For a larger displacement, some edges cannot be matched to those on the original positions, reducing the relative size of the matching itself (i.e., the edge match ratio) and thus decreasing the similarity. Due to the choice of a geometry-independent property of interest, the similarity does in fact depend entirely on the edge match ratio and no artificial property error is introduced. In combination with the (overall) monotonically decreasing value of \mathcal{S}_P (for larger perturbation degrees d), this shows that the strategy of ignoring implausible matches in the matching phase is successful: There is no prominent, clearly defined increase in the similarity, which would originate from an incorrect, but globally optimal matching in terms of the distance function (cf. Section III-C). Instead, by and large the similarity monotonically decreases with increasing

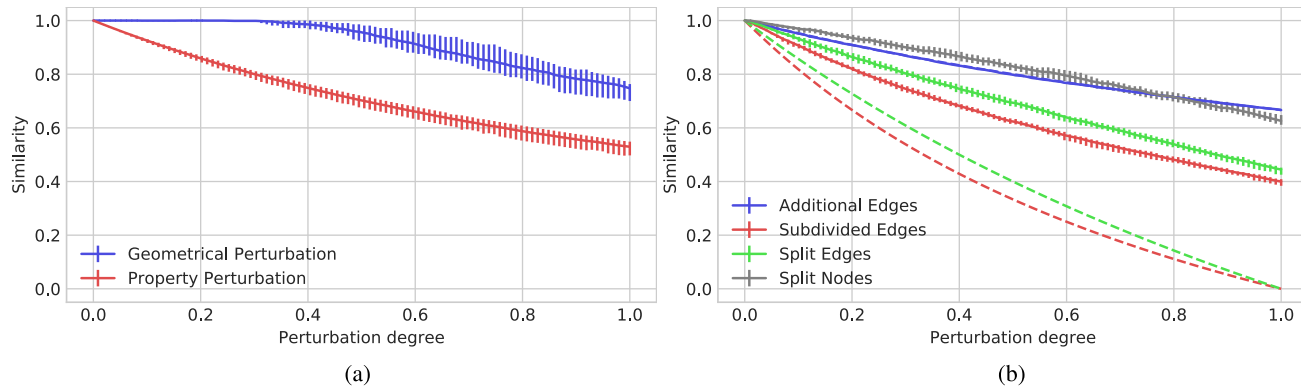


FIGURE 3. The similarity $S_P(G, D_d(G))$ between a graph by real-world lymphatic vessel dataset G and the perturbed versions $D_d(G)$ for a given degree d and $P =$ average roundness as the observed property. Both property and geometrical perturbations (a) and topological perturbations (b) were applied. In all cases an increased perturbation results in an approximately monotonic decrease in similarity. Each simulation was repeated 10 times. The average similarity is displayed as a solid line, while the minimum and maximum for each respective value of d are represented using the error bars. For topological properties (b) dashed lines indicate predicted lower bounds.

perturbation degree d . This is also true for all 10 individual simulations, although not directly visible in Figure 3a.

On the other hand, introducing errors to the property values itself while leaving spatial positions of edges intact does not impact the edge match ratio except in extreme unlikely (or artificial) cases of changes to the property values (i.e., so that for all properties $(1 - E_P) \approx \epsilon$). Consequently, the similarity depends entirely on the error introduced to the property of interest. The analysis of expected similarity values (assuming perturbation of properties by a log-normal distributed factor) is not discussed here in detail, but suggests a hyperbolic curve (as indicated in Figure 3a). The observed curve indeed appears to follow the prediction and (most importantly) decreases monotonically.

As can be seen in Figure 3b, for all topological perturbation types (by and large) the similarity decreases monotonically for an increasing degree d . In the case of additional edges, the generated curve almost exactly matches the following prediction and shows little to no variation between simulation runs.

$$\frac{2|M_{G,D_d(G)}|}{|E_G| + |E_{D_d(G)}|} = \frac{2}{1 + (1 + 2d)} = \frac{2}{2 + d}$$

This suggests that while computing the similarity measure, all original edges are correctly matched and the added edges are ignored. In the case of a perturbation by subdividing edges, a lower bound for the resulting similarity is given by:

$$\frac{2|M_{G,D_d(G)}|}{|E_G| + |E_{D_d(G)}|} = \frac{2 \cdot (1 - d)}{1 + (1 + 2d)} = \frac{1 - d}{1 + d}$$

(Assuming that subdivided edges are not matched to any edges in the original graph.) In the case of the example shown in Figure 3b, the general shape of the curve follows a hyperbolic path (and thus decreases monotonically), but is always larger than the lower bound, suggesting that at least some of the subdivided edges are matched with edges in the original graph. This should be expected (and is desirable in general) since at least the larger part of the original segment

can often still approximately match the original path. For split edges the lower bound changes to:

$$\frac{2|M_{G,D_d(G)}|}{|E_G| + |E_{D_d(G)}|} = \frac{2 \cdot (1 - d)}{1 + (1 + d)} = \frac{2 - 2d}{2 + d}$$

Again, the curve does not follow the lower bound curve exactly, but is similar in shape. Similar to the subdivision of edges, one should assume that larger parts of some split edges can be matched to the original edge, albeit with a lower similarity resulting in a relatively low (but not the lowest possible) total similarity. It should be noted that this is desired behavior. While errors that subdivide or split edges should be penalized by a similarity measure, the resulting drop in similarity should be lower than if the edges would have been deleted. This is not the case for the bipartite graph matching procedure, where splitting or subdividing an edge most likely results in the same cost as a deleted edge if both originally connected nodes are matched correctly, but are not directly connected by an edge anymore.

For split nodes a prediction of an expected curve shape is difficult due to the unknown relationship of branch nodes and edges in the graph. Still, observation shows that overall the curve decreases monotonically for an increased perturbation degree. Also, for fixed values of d there is little variation in similarity between simulations.

In total, for all considered perturbation types the similarity between the original and the modified graph for a given property (in this case the average roundness) seems to behave in a predictable and expected manner: The similarity of identical graphs is 1 and decreases for larger degrees of perturbation monotonically – in many cases more or less following a parabola suggesting the usefulness of the described measure even for larger differences in the input graphs.

V. EXEMPLARY STUDY

In order to demonstrate the applicability and usefulness of GERoMe, we have performed an exemplary study to evaluate the robustness of a preliminary version of the graph creation

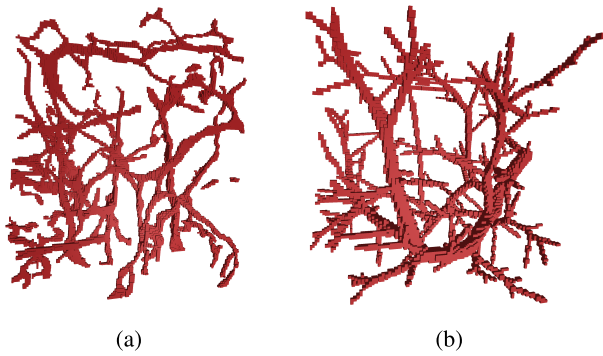


FIGURE 4. The datasets used in the exemplary study rendered as isosurfaces: The real-world dataset is a foreground segmentation of an ultramicroscopy image of human lymphatic vessel tissue [3] while the synthetic dataset is the foreground segmentation of a simulation of a blood vessel tree generated using VascuSynth [15]. The real-world lymphatic vessel dataset (a) shows higher complexity in both topology and morphology when compared to the synthetic dataset (b).

and feature extraction algorithm proposed in [11]. Given a binary volumetric input dataset, the algorithm first creates a binary voxel skeleton. From this skeleton, a graph embedded in 3D space is extracted. Afterwards, the algorithm calculates geometric and morphological edge-associated properties using the skeleton as well as the original binary volume. For the purpose of this study we restrict the set of examined edge-associated properties to *length* (the length of a branch when following the medial line), *distance* (the Euclidean distance of the connected nodes), *straightness* = $\frac{distance}{length}$, *avgRadius* (i.e., the average distance of the medial line to the surface of the branch) and *volume* (the total volume occupied by a branch in the original dataset).

For the set of applied transformations \mathcal{T} , 4 rotation axes (the diagonal axis (1, 1, 1) as well as the three main coordinate axes) are taken into account. For each axis, 36 rotations with respective rotation angles equally distributed in the range $[0, 2\pi)$ are considered, in total resulting in $4 \cdot 36 = 144$ transformations.

Using these parameters, $\mathcal{G}_{s, \mathcal{T}, p}$ is applied to the 3D ground truth foreground segmentation of an artificial blood vessel tree structure generated by VascuSynth [15] as well as a foreground segmentation of an ultramicroscopy image of human lymphatic vessel tissue [3] (both depicted in Figure 4). While the first dataset has isotropic resolution, i.e., constant voxel spacing in x-, y- and z-direction, the second is anisotropic (i.e., the voxel-spacing is larger in z-direction than in x-, and y-direction). For each rotation T the binary 3D image input data is transformed by resampling it from the original volume using nearest neighbor filtering. Since a rotation T (and thus also the inverse rotation T^{-1}) does not change the values of the aforementioned edge-associated properties (*length*, *distance*, *straightness*, *avgRadius*, and *volume*), an extracted graph can be transformed back into the original space by applying T^{-1} merely to the node positions. As both test datasets contain vessels of low *avgRadius*, the transformed volume (i.e., $T(s)$) is generated by doubling the resolution in

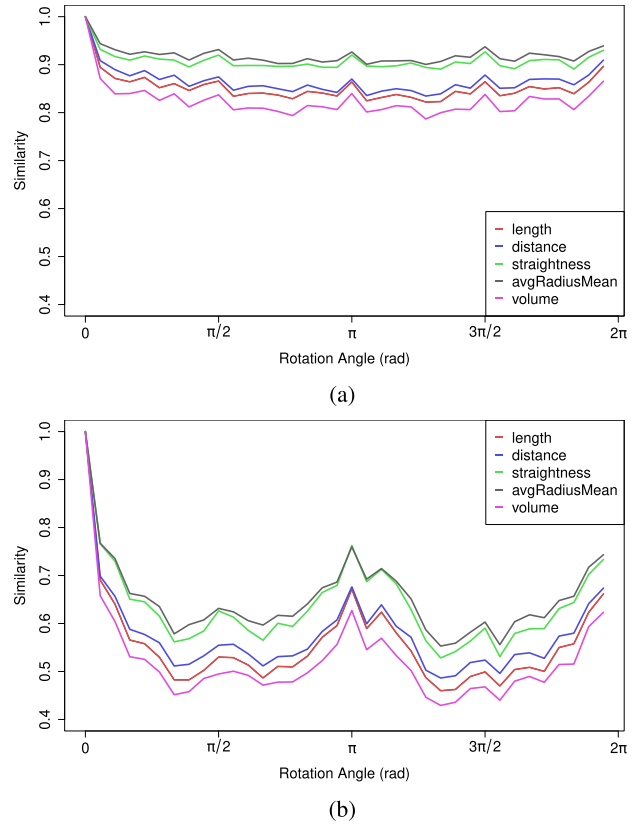


FIGURE 5. Intermediate results of the procedure depicted in Figure 1 for a synthetic (a) and a real-world dataset (b) without using ground truth information. A preliminary version of the algorithm described by [11] has been used to extract the examined graphs. For each dataset, the similarity \mathcal{S}_p of the graph extracted from the original dataset and the graphs extracted after applying 36 rotations around the x-axis are shown for 5 selected properties.

each dimension in order to reduce the error introduced in the resampling step itself.

Since there is no ground truth information available for the real-world dataset, and the ground truth for the synthetic dataset does not include all properties of interest, we use graphs extracted from the input dataset as template graphs and thus only consider the robustness of the algorithm as opposed to evaluating its accuracy which requires ground truth information. The resolution of each original dataset is also octupled prior to starting the graph extraction process in order to allow for an equal level of accuracy in the generation of the intermediate voxel skeleton and the extraction of edge-associated properties of the template graph.

A. SYNTHETIC DATA

The similarity of the original graph extracted from a synthetic blood vasculature dataset generated using VascuSynth and a transformed version of the input volume is illustrated in Figure 5a for the 5 selected properties. The set of applied transformations comprises 36 rotations around the x-axis of the coordinate system. As can be seen, the plot shows 4 peaks for all properties which correspond to the angles in which the transformed volume is aligned with the voxel grid of the

TABLE 1. The robustness measure GERoMe \mathcal{G} applied to a preliminary version of the algorithm proposed in [11] for 5 selected properties, using the synthetic dataset generated by VascoSynth and a real-world lymphatic vessel dataset. The sets of transformations comprise 36 rotations around each of the coordinate axes (\mathcal{T}_x , \mathcal{T}_y , \mathcal{T}_z) as well as the diagonal axis $(1, 1, 1)$ (\mathcal{T}_{xyz}).

$\mathcal{G}_{s,\mathcal{T},P}(\mathcal{A})$		<i>length</i>	<i>distance</i>	<i>straightness</i>	<i>avgRadius</i>	<i>volume</i>
Synthetic Dataset	\mathcal{T}_x	0.822	0.835	0.891	0.900	0.787
	\mathcal{T}_y	0.816	0.828	0.874	0.892	0.789
	\mathcal{T}_z	0.830	0.837	0.880	0.893	0.801
	\mathcal{T}_{xyz}	0.730	0.735	0.790	0.799	0.675
Real-World Dataset	\mathcal{T}_x	0.460	0.486	0.528	0.553	0.429
	\mathcal{T}_y	0.460	0.484	0.556	0.563	0.429
	\mathcal{T}_z	0.559	0.575	0.651	0.656	0.528
	\mathcal{T}_{xyz}	0.298	0.312	0.341	0.371	0.272

original volume (i.e., all angles that are multiples of $\frac{\pi}{2}$). This illustrates that the observed error can partially be attributed to the resampling process rather than the graph extraction algorithm itself. Moreover, for some properties the relative error seems to be affected more by the transformation process than for others: Both *avgRadius* and *straightness* are less affected than *distance*, *length*, and especially *volume*. The relative errors of *length* and *distance* are probably caused by small variations in node positions, while this does not have such a strong effect on *straightness* = $\frac{\text{distance}}{\text{length}}$. The per-edge *volume* can be expected to be more strongly affected by errors in the resampling process than other properties. The relative error of *avgRadius* is likely caused by errors in the resampling process as well, but to a lesser extent, since the property is averaged along the run of a branch. GERoMe values for sets of rotations for the 4 considered rotation axes are shown in Table 1.

B. REAL-WORLD LYMPHATIC VESSEL DATA

The similarity of the original graph extracted from a real-world lymphatic vasculature dataset and a transformed version of this data is shown in Figure 5b for the 5 selected properties. Again, the set of applied transformations comprises 36 rotations around the x-axis of the coordinate system. In comparison to the similarities extracted from the synthetic dataset, the property similarities S_P are much lower. These relatively low similarity values originate from both the relative property error (see Figure 6a) as well as the edge match ratio (see Figure 6b). As observed for the synthetic dataset, the similarity, the edge match ratio, and (to a lesser extent) the relative error seem to assume local extrema whenever the voxel grids of the original volume and the transformed volume are aligned, i.e., for rotation angles that are multiples of $\frac{\pi}{2}$ for coordinate axes as rotational axes, and for multiples of $\frac{2\pi}{3}$ for $(1, 1, 1)$. This behavior can at least partly be attributed to the resampling process involved in transforming the image data. This is illustrated in Figure 7, which shows the difference between the original dataset and a version transformed by \mathcal{T} and subsequently by \mathcal{T}^{-1} . The plot shows that the resampling process itself produces peaks at the same rotation angles and comparatively of similar magnitude. The overall lower error for rotations around the z-axis can be explained by the voxel spacing of the real world dataset which is equal

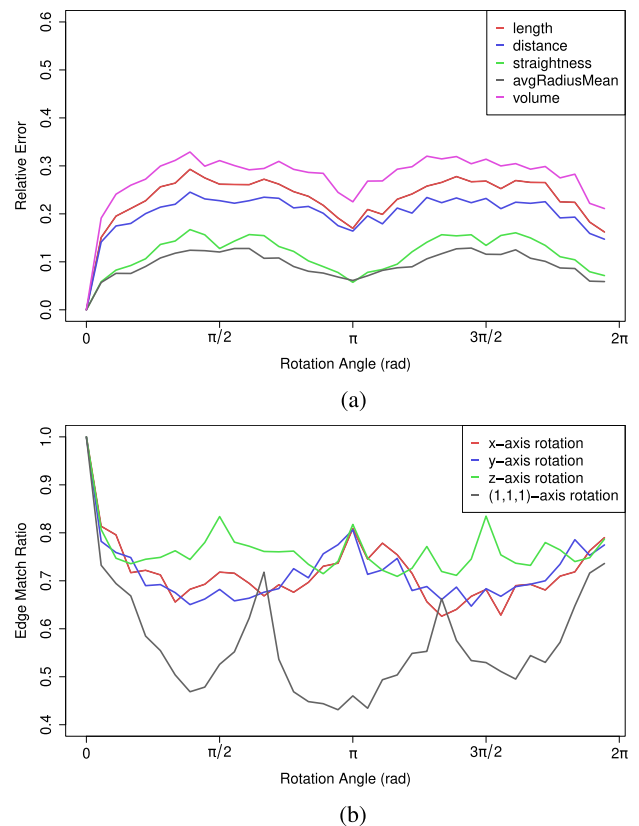


FIGURE 6. Intermediate measures generated from a real-world dataset for a preliminary version of the algorithm described by [11]. In (a), the relative errors of 5 selected properties are plotted for 36 rotations \mathcal{T} around the x-axis. In (b) the edge match ratios for 36 rotations around the x-, y-, and z-axis, as well as $(1,1,1)$ are shown.

in x- and y-direction, but larger in z-direction. It should be noted that the magnitude of the resampling errors in Figure 7 cannot be directly compared to the measured similarity to the template graph. Still, this suggests that at least part of the dissimilarity originates from resampling errors. However, this does not imply a weakness of the proposed method itself, as the parameter \mathcal{T} as well as optional upsampling can and should always be kept constant when comparing methods and specified along with the results. Indeed, at least in this example of binary volumetric input data, some amount of error *has* to be introduced to the initial dataset in order to force an examined (deterministic, but unspecified) algorithm to take a different path in its execution. An example of an

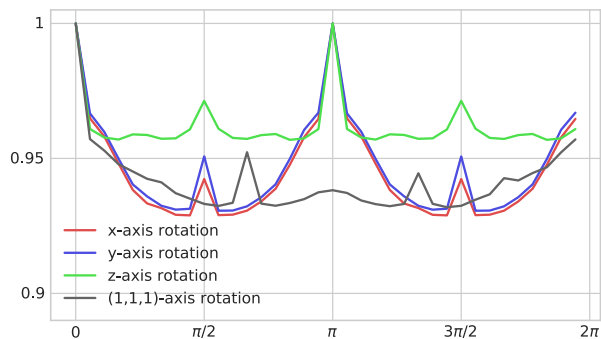


FIGURE 7. The voxel-wise DICE index between the original real-world dataset (s) and a version that has been transformed back and forth using \mathcal{T} (i.e., $\mathcal{T}^{-1}(\mathcal{T}(s))$) for 36 rotations \mathcal{T} around the x -, y -, and z -axis, as well as $(1,1,1)$.

(in terms of GERoMe) ineffective transformation T would be a translation in combination with the nearest neighbor resampling strategy: Except for a fixed voxel offset, the initial and transformed dataset would be equivalent and (even otherwise non-robust) algorithms would likely produce identical results regardless of the specific definition of T .

The fact that a rotational axis of $(1, 1, 1)$ produces larger resampling errors also becomes apparent in the final GERoMe values (see Table 1): For both datasets the minimum similarity for all properties was reached for a transformation around this (non-aligned) axis. Moreover, it can be observed that the amount of relative error introduced by the transformation and resampling process seems to be relatively independent of the dataset: Just like it is the case for the synthetic dataset, *avgRadius* and *straightness* seem to be less affected than *distance*, *length*, and *volume*.

Another aspect to note is that at least the examined algorithm does not produce outliers in terms of the similarity between two graphs for any transformation. This is an important property of robust graph extraction algorithms. Any potentially generated outliers (and thus flaws of the examined algorithm causing unstable results) would immediately become visible in \mathcal{G} , as it is defined as the minimum of all similarities.

These results also show that the examined extraction algorithm produces much more stable results for the synthetic dataset than for the real-world dataset. This indicates that evaluating graph extraction algorithms solely on the basis of synthetic datasets is a highly problematic strategy. In combination with difficulties in obtaining ground truth annotations for real-world data this underlines the usefulness of our method.

VI. CONCLUSION

We have proposed GERoMe, a novel robustness measure for graph extraction algorithms. Our approach does not necessarily require ground truth data and can be applied to evaluate the stability of any algorithm which extracts (multi-)graphs that are embedded in Euclidean space from non-graph structures for which an edge property-preserving

invertible transformation is defined. If ground truth data is available, the method and the introduced similarity measure can be used to quantify the accuracy of graph extraction algorithms in conjunction with the robustness. In order to be able to match true multigraphs, we use edge-associated properties to distinguish edges in addition to the node positions. The proposed graph similarity measure has been shown to behave predictably and expectedly on common graph perturbation patterns. The applicability and usefulness of our method has been demonstrated in an exemplary study on synthetic as well as real-world biomedical 3D image data. We are convinced that GERoMe will prove useful for evaluating graph extraction algorithms, especially in cases where ground truth data is not available.

In the future, we plan to study and compare the performance of state-of-the-art graph extraction algorithms using GERoMe. Moreover, we would like to augment and generalize the matching process and the similarity measure by incorporating information from node-associated properties. Similarly, it may be interesting to apply the presented framework for robustness evaluation to other problem domains, e.g., using the similarity measure presented in [22] or methods that focus on node similarity. Additionally, it may be worthwhile to consider and compare alternative matching approaches which utilize the expected spatial proximity of matched edges if the runtime of the proposed method (which is dominated by the edge matching computation) is problematic.

ACKNOWLEDGEMENT

The authors acknowledge support from the Open Access Publication Fund of the University of Münster.

REFERENCES

- [1] D. Selle, B. Preim, A. Schenk, and H. O. Peitgen, "Analysis of vasculature for liver surgical planning," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1344–1357, Nov. 2002.
- [2] S. A. Wood, E. A. Zerhouni, J. D. Hoford, E. A. Hoffman, and W. Mitzner, "Measurement of three-dimensional lung tree structures by using computed tomography," *J. Appl. Physiol.*, vol. 79, no. 5, pp. 1687–1697, 1995.
- [3] R. Hägerling et al., "VIPAR, a quantitative approach to 3D histopathology applied to lymphatic malformations," *JCI Insight*, vol. 2, no. 16, 2017, Art. no. e93424.
- [4] S. L. Wearne, A. Rodriguez, D. B. Ehlenberger, A. B. Rocher, S. C. Henderson, and P. R. Hof, "New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales," *Neuroscience*, vol. 136, no. 3, pp. 661–680, 2005.
- [5] J. H. Metzner, T. Kröger, A. Schenk, S. Zidowitz, H. Peitgen, and X. Jiang, "Matching of anatomical tree structures for registration of medical images," *Image Vis. Comput.*, vol. 27, no. 7, pp. 923–933, 2009.
- [6] S. Almasi, A. Lauric, A. M. Malek, and E. L. Miller, "Cerebrovascular network registration via an efficient attributed graph matching technique," *Med. Image Anal.*, vol. 46, pp. 118–129, May 2018.
- [7] Y. Chen, C. O. Laura, and K. Drechsler, "Generation of a graph representation from three-dimensional skeletons of the liver vasculature," in *Proc. Int. Conf. Biomed. Eng. Inf.*, Oct. 2009, pp. 1–5.
- [8] G. Gerig, T. Koller, G. Székely, C. Brechbühler, and O. Kübler, "Symbolic description of 3-D structures applied to cerebral vessel tree obtained from MR angiography volume data," in *Proc. Inf. Process. Med. Imag. (IPMI)*, 1993, pp. 94–111.
- [9] G. Klette, "Branch voxels and junctions in 3D skeletons," in *Proc. Workshop Combinat. Image Anal.*, 2006, pp. 34–44.

- [10] K. Drechsler and C. O. Laura, "Hierarchical decomposition of vessel skeletons for graph creation and feature extraction," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, Dec. 2010, pp. 456–461.
- [11] D. Drees, A. Scherzinger, R. Hägerling, F. Kiefer, and X. Jiang, "Scalable robust graph and feature extraction for arbitrary vessel networks in volumetric datasets," Tech. Rep.
- [12] A. Rodriguez, D. B. Ehlenberger, P. R. Hof, and S. L. Wearne, "Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images," *Nature Protocols*, vol. 1, no. 4, pp. 2152–2161, 2006.
- [13] K. Rothaus, X. Jiang, and P. Rhiem, "Separation of the retinal vascular graph in arteries and veins based upon structural knowledge," *Image Vis. Comput.*, vol. 27, no. 7, pp. 864–875, 2009.
- [14] X. Jiang, M. Lambers, and H. Bunke, "Structural performance evaluation of curvilinear structure detection algorithms with application to retinal vessel segmentation," *Pattern Recognit. Lett.*, vol. 33, no. 15, pp. 2048–2056, 2012.
- [15] G. Hamarneh and P. Jassi, "VascuSynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis," *Comput. Med. Imag. Graph.*, vol. 34, no. 8, pp. 605–616, 2010.
- [16] J. He, C. Kim, and C. J. Kuo, *Interactive Segmentation Techniques: Algorithms and Performance Evaluation* (Springer Briefs in Electrical and Computer Engineering). Singapore: Springer, 2014.
- [17] P. Theologou, I. Pratikakis, and T. Theoharis, "A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation," *Comput. Vis. Image Understand.*, vol. 135, pp. 49–82, Jun. 2015.
- [18] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "Evaluation of the stability of four document segmentation algorithms," in *Proc. 12th IAPR Workshop Document Anal. Syst.*, Apr. 2016, pp. 215–220.
- [19] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognit.*, vol. 64, pp. 1–14, Apr. 2017.
- [20] D. Drees, A. Scherzinger, and X. Jiang, "GERoMe—A novel graph extraction robustness measure," in *Proc. 11th IAPR Workshop Graph-Based Represent. Pattern Recognit. (GbrPR)*, 2017, pp. 73–82.
- [21] K. Drechsler, S. Meixner, C. O. Laura, and S. Wesarg, "A framework for validation of vessel segmentation algorithms," in *Proc. Int. Symp. Comput.-Based Med. Syst.*, Jun. 2013, pp. 518–519.
- [22] D. Mayerich, C. Björnsson, J. Taylor, and B. Roysam, "NetMets: Software for quantifying and visualizing errors in biological network segmentation," *BMC Bioinf.*, vol. 13, no. 8, p. S7, 2012.
- [23] H. Heumann and G. Wittum, "The tree-edit-distance, a measure for quantifying neuronal morphology," *Neuroinformatics*, vol. 7, no. 3, pp. 179–190, 2009.
- [24] P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 28, no. 1, 2014, Art. no. 1450001.
- [25] K. Riesen, X. Jiang, and H. Bunke, "Exact and inexact graph matching: Methodology and applications," in *Managing and Mining Graph Data*. Boston, MA, USA: Springer, 2010, pp. 215–246.
- [26] A. Sanfeliu and K.-S. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 3, pp. 353–362, May 1983.
- [27] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image Vis. Comput.*, vol. 27, no. 7, pp. 950–959, 2009.
- [28] M. Stauffer, T. Tschachtli, A. Fischer, and K. Riesen, "A survey on applications of bipartite graph edit distance," in *Graph-Based Representations in Pattern Recognition*. Cham, Switzerland: Springer, 2017, pp. 242–252.
- [29] B. Lamiroy and T. Sun, "Computing precision and recall with missing or uncertain ground truth," in *Proc. Int. Conf. Multimedia Retr.*, 2011, pp. 149–162.
- [30] B. Lamiroy and P. Pierrot, "Statistical performance metrics for use with imprecise ground-truth," in *Graphic Recognition. Current Trends and Challenges*. Cham, Switzerland: Springer, 2015, pp. 31–44.
- [31] C. Spampinato, S. Palazzo, and D. Giordano, "Evaluation of tracking algorithm performance without ground-truth data," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2012, pp. 1345–1348.
- [32] L. Zhang, J. Lan, and X. R. Li, "Performance evaluation of multi-target tracking without knowing ground truth," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 185–192.
- [33] V. V. Valindria et al., "Reverse classification accuracy: Predicting segmentation performance in the absence of ground truth," *IEEE Trans. Med. Imag.*, vol. 36, no. 8, pp. 1597–1606, Aug. 2017.
- [34] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.



DOMINIK DREES received the M.S. degree in computer science from the University of Münster, Germany, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and biomedical image processing with a focus on the analysis of vessel network structures.



AARON SCHERZINGER received the Ph.D. degree in computer science from the University of Münster, Germany, in 2018, under the supervision of X. Jiang and K. Hinrichs. He is currently a Post-Doctoral Research Associate at the Faculty of Mathematics and Computer Science, University of Münster. His research interests include biomedical image processing and analysis, pattern recognition, machine learning, scientific visualization, and computer graphics.



XIAOYI JIANG received the degree in computer science from Peking University and the Ph.D. and Venia Docendi (Habilitation) degrees in computer science from the University of Bern, Switzerland. He was an Associate Professor with the Technical University of Berlin, Germany. Since 2002, he has been a Full Professor with the University of Münster, Germany. He is a Fellow of IAPR. He is currently the Editor-in-Chief of the *International Journal of Pattern Recognition and Artificial Intelligence*. In addition, he also serves on the Advisory Board and Editorial Board of several journals, including the *IEEE TRANSACTIONS ON MEDICAL IMAGING*, the *International Journal of Neural Systems*, and *Pattern Recognition*.

• • •