

# INVANER: Interactive Vascular Network Editing and Repair

**Valentin Z. Nigolian**  
 École Polytechnique Fédérale  
 de Lausanne  
 Lausanne, Switzerland  
 vnigolian@acm.org

**Takeo Igarashi**  
 The University of Tokyo  
 Tokyo, Japan  
 takeo@acm.org

**Hirofumi Seo, MD**  
 The University of Tokyo  
 Tokyo, Japan  
 seo.hirofumi@mail.u-  
 tokyo.ac.jp

## ABSTRACT

Vascular network reconstruction is an essential aspect of the daily practice of medical doctors working with vascular systems. Accurately representing vascular networks, not only graphically but also in a way that encompasses their structure, can be used to run simulations, plan medical procedures or identify real-life diseases, for example. A vascular network is thus reconstructed from a 3D medical image sequence via segmentation and skeletonization. Many automatic algorithms exist to do so but tend to fail for specific corner cases. On the other hand, manual methods exist as well but are tedious to use and require a lot of time. In this paper, we introduce an interactive vascular network reconstruction system called INVANER that relies on a graph-like representation of the network's structure. A general skeleton is obtained with an automatic method and medical practitioners are allowed to manually repair the local defects where this method fails. Our system uses graph-related tools with local effects and introduces two novel tools, dedicated to solving two common problems arising when automatically extracting the centerlines of vascular structures: so-called "Kissing Vessels" and a type of phenomenon we call "Dotted Vessels."

## CCS Concepts

•**Human-centered computing** → **Information visualization**; **Visualization toolkits**; **Graphical user interfaces**;

## Author Keywords

medical system; surgical planning; graph-based interactions; anatomical visualization;

## INTRODUCTION

Computer systems are now widely used in the medical field and are generally limited to visualization or exploration. Medical practitioners use those in many different ways such as planning procedures, discussing treatments or diagnosing patients [13, 7]. A particular subset of those systems is used to consider vascular networks. By their cylindrical tree-like nature, they are fit to be represented as *skeletons*. Skeletons

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

UIST '19, October 20-23, 2019, New Orleans, LA, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-6816-2/19/10...\$15.00

DOI: <https://doi.org/10.1145/3332165.3347900>

are representations that abstract structures as a set of points linked by a set of curves. They offer a way to easily understand the general topology of a vascular structure. A wide range of algorithms exists to automatically extract a vascular skeleton from medical images, as shown in a recent survey [29], but those methods all fail in some corner cases. On the other hand, there exist interactive methods such as the one presented in [1, 31]. Practitioners might even use fully manual methods such as placing all points of the vessels' centerline manually in a 3D environment like Maya or Blender. Although these methods can fix any corner cases, their usage is often very tedious.

The main application considered for our system was to allow medical doctors, particularly neurosurgeons, to plan, discuss or teach medical procedures. A widespread disease of the human neurovascular system was in mind during our work: brain aneurysms. In short, aneurysms are deformations occurring in blood vessels where their wall is locally weaker, resulting in this thin wall inflating, similarly to a balloon. Treatments for this disease [10] require an accurate representation of the vascular network surrounding the aneurysm. Throughout the development of our system, we interacted with two neurosurgeons to ensure that our work was relevant to their daily practice "on the field." They regularly exchanged with one of the authors of this work who is himself a medical doctor to ensure the validity of our medical-wise claims.

In this paper, we introduce a novel, hybrid method that relies on an automatic skeleton extraction, or *skeletonization*, algorithm and lets the user "fix" the corner-cases, or defects, of this automatic method. We aim at providing a system used by medical doctors to decrease the time and effort required to obtain a faithful reconstruction of a vascular network. Our system, called "INVANER" (INteractive VAscular Network Editing and Repair), relies on a graph-like representation of vascular networks and graph-related tools to modify said networks.

Moreover, we introduce two such tools dedicated to solving two common problems that occur when using skeletonization algorithms. The first one is called the "Kissing Vessel" problem [38] and is depicted in Figure 1. This happens typically when multiple vessels occupy the same space on multiple layers of the volumetric data, thus appearing to be a single vessel but actually being multiple vessels very close to each other, hence the term "kissing." The second one comes from the original volumetric data containing inconsistent density values with repeated variations where the density should be close to constant. That is, a situation where a vessel, a physical object

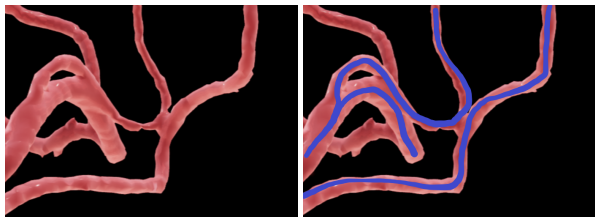


Figure 1: An example of Kissing Vessels. A 3D representation of vessels that are so close that they appear to be merged (left) and the actual vessels suggested with blue strokes (right).

with regular density over its volume, is detected as a set of aligned, shorter and unconnected pieces, as shown by Figure 2. Although some information is missing from the data, medical experts can identify this set of pieces as being part of an actual blood vessel. We call those defects "Dotted Vessels" as they look like the equivalent of "dotted lines" to blood vessels' 3D representation.

These two problems were unequivocally chosen as being important in daily medical practice by the experts involved. As our senior MD collaborator quotes:

"Dotted Vessels are almost everywhere in medical imaging because even modern medical devices cannot capture small vessels. It is not always necessary to trace all small vessels but damaging important ones could cause an important brain dysfunction. Visualizing them is very important for precision surgical planning. There are not many Kissing Vessels in typical medical imaging but at least 5 or 10 instances to fix in a single brain scan. Again, it is not always necessary to fix them all but it is definitely necessary when it is related to surgical planning. In this context, the proposed system is very beneficial."

We claim that our contribution is twofold:

1. We present a hybrid method for constructing vascular networks from medical images combining automatic skeletonization and graph-based interactive editing. This hybrid approach saves time and effort for medical doctors by focusing their interactions on defects occurring when using automatic skeletonization algorithms.
2. We also introduce two interactive tools specialized in solving the Kissing Vessel and Dotted Vessel problems respectively. Their effectiveness is demonstrated with the results of a comparison study and with feedback from neurosurgeons.

## BACKGROUND

In this section, we briefly introduce past research related to our work, namely medical data, skeletonization, editing tools for medical data and graph-based interactions and simulations.

### Volumetric Medical Data

Medical imagery data can be obtained via a wide range of methods (MRI, CT, angiography, X-ray, etc.) and one of the most common format to store this data is called DICOM [17].

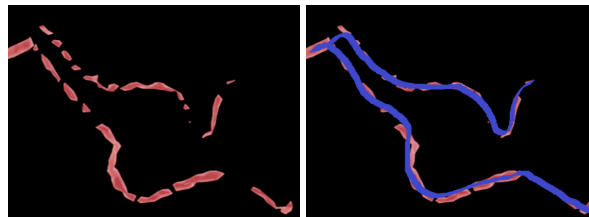


Figure 2: An example of Dotted Vessels. The various pieces (left) suggest the actual shape of the blood vessels, represented by blue strokes (right).

In essence, it represents a volume as a discrete 3D grid, in which each cell or *voxel* contains a single value representing the density of the volume at this point in space. Medical practitioners can then visualize this data by considering isolated individual layers and map a certain range of density to a black-to-white intensity range to obtain a grey-scale image of this layer. The main problem with this kind of data is to find the right threshold to distinguish the desired organ from its surrounding tissues. Indeed, it often happens that organs in close contact have a very similar density and their separation can hence be hard to locate precisely. Moreover, the appropriate threshold might have to vary spatially to obtain optimal results. A key consideration for volumetric medical data is to label or *segment* the volume between the organ (or part of) in focus (1) and the rest (0) in a binary fashion [21, 23, 12]. In our case, we assume that our data has been processed by medical experts beforehand. That is, that our input data contains little to no noise and that the vascular network has been correctly segmented. The sole limitation of the quality of our input is theoretically the limitations of the data acquisition method. We do not concern ourselves with how medical experts obtain data satisfying those assumptions. Although obtaining a correctly segmented volume is definitely not a trivial problem, it is common practice for medical doctors and not a focus of our work. Furthermore, the medical doctors involved in our work all agreed that in the presence of artifacts, the "perfect segmentation" sometimes cannot be ascertained, as experts might disagree on the correct topology. We thus considered those to be reasonable assumptions. Our work should be seen as a kind of generic plugin rather than a system to perform a full segmentation. It aims at reducing the manual effort to reconstruct the final vascular network, the "final touch", to a minimum.

### Skeletonization

As summarized in [29] or [33], there are a large number of existing extraction algorithms and methods. A wide variety of techniques such as graph-based [39, 26], physics-based [34] or voxel-based techniques [19] are employed. Moreover, those techniques rely on different types of input data, e.g., point-cloud, 3D-surfaces or volumetric data, as explored in length in this thesis [18]. However, as its author notes:

"It should be mentioned that most previous works only considered the vascular network as directed binary graphs without loops [...]. This assumption presents many limitations for clinical purposes."

Indeed, Kissing Vessels typically create loops (or *cycles*), and there is an actual cycle in the human brain called the "Circle of Willis," therefore cycles should not be disregarded. Unfortunately, most of the aforementioned works mainly focus on increasing the speed or accuracy of the skeletonization and usually fail when encountering corner-cases violating the quoted assumption. We did find an automatic method that tried to solve the Kissing Vessel problem [11] but as the authors discuss, it fails for vessels kissing on a distance too long or if more than two vessels kiss in the same area. We thus lack a general skeletonization solution that works well even if the data contains some defects. For our system, we use an "asymmetric thinning scheme" [9] relying on volumetric data represented as a set of voxels. It reduces the set of voxels representing the vascular network to a set of one-voxel-wide curves aligned with the vascular network's centerlines. We chose this method because we also use voxels as input to our system and graph extraction algorithm and because it can handle cycles. Cycles are essential for our system, as we use them to spot Kissing Vessels in vascular structures as explained below.

### Editing Tools for Medical Data

As mentioned before, interactive systems are used everywhere in the medical field, but there are not that many that are *truly* interactive in the sense that they are usually meant for visualization rather than editing. However, efforts have been made recently in that direction with, for instance, interactive volume segmentation systems [14, 35, 16] or surgical planning systems [25, 24, 28]. In these works, the authors present simple-to-use interfaces which provide natural means for experts to discuss complex concepts. A notable related work in this direction is Foldit [8], the well-known system that allows random users to contribute to high-level research. Their task is to find geometric configuration of protein that minimizes an energy, while our task is to reconstruct plausible vascular network from medical images. Other existing mesh-editing interfaces such as [2] or [41] allow user to improve the quality of a surface mesh. In contrast, degenerate triangles or varying resolution are irrelevant to our data context (e.g., blood vessels do not have sharp edges). Moreover, the notion of meshes and its potential artifacts is most often foreign to medical doctors.

We also found a system meant for vascular extraction [40] but it does not make any mention of the potential corner-cases. The author of [1] considers the Kissing Vessel problem and provides a system to define where a vessel begins and ends. It then infers the complete centerline of this vessel and lets the user give supplementary information if this method fails, typically when encountering a Kissing Vessel. If the vascular network to skeletonize contains no defects, a user of [1] would hence still have to extract each vessel of the network manually, which can be very time-consuming for large networks. Moreover, this method is bound to the vascular network's defining volumetric data and does not allow for any vessel outside of it. It is thus not fit for any situation where a user would need to freely sculpt one or more vessels. Such a situation can arise when medical practitioners might want to explore how some surgery would need to change if the vascular network was slightly different. Similarly, it could help in visualizing how simulations change in different situations. This also relates

to [28] in which sketches are translated into fluid dynamics simulations. This work, however, is limited in the sense that it relies on user sketches only, rather than importing real-world data. The work presented in [15] highlights the interest of human intervention in modifying the result of a skeletonization but fails to offer efficient and convenient ways to do so. We draw inspiration from those works and attempt to provide a system that reduces the required human intervention to only where and when medical knowledge is necessary and lets the users modify the existing network in a free-form manner.

### Graphs: Extraction, Interactions, and Simulations

By representing the centerlines of a network as a set of nodes or *vertices* connected by curves or *edges*, we obtain a graph. Indeed, these are mathematical abstractions that are very well-suited to represent networks and are used extensively for scientific visualization [36]. Graphs are also very useful to summarize and visualize complex phenomena that can be represented by networks [37] or to run simulations. This is especially true for biomedical simulations, which often involve networks, in particular, vascular networks [42, 27]. In those latter cases, the shape of the vessels are simplified into straight cylinders in an effort for simplicity, but it may be that the actual shape of the vessels is still necessary. By coupling skeletons and surfaces or *rigging* [3, 4], we can make high-level deformations of the surface by performing operations on the skeleton only. A good example of this concept is SkeletonLab [5], which provides a set of elementary graph operations such as adding vertices on edges, displacing vertices, adding edges, etc. Although we share our graph-based approach with SkeletonLab, we do not show this abstraction to our users. From their point-of-view, they are manipulating a network without requiring any knowledge of the nature of graphs. Furthermore, our work, does not involve directly rigging the input surface as we consider cylindrical curves to be close enough to the surface to faithfully represent its topology. We do, however, use a set of somewhat similar elementary, low-level graph *operations* as a basis to build a set of higher-level *tools* used directly by the users.

### PRELIMINARIES

In our graph representation, each vertex is assigned a 3D position, and each edge is assigned a 3D polyline going between both vertices the edge connects. Note that in Graph Theory terms, we always consider our graph to be an *undirected multigraph*. That is, edges are not considered to have a direction (i.e., there is no source or target vertex, both have the same status), and there can be multiple edges connecting the same two vertices. We say that two vertices are *adjacent* if an edge connects them. Similarly, we say that two edges are *adjacent* if they both connect the same vertex to other vertices. There is a *path* between two vertices if we can reach one vertex to the other by following a series of adjacent vertices between them. The number of edges connected to a vertex is called the *degree* of the vertex. We let the reader refer to Graph Theory literature for more formal definitions of those concepts. We now establish a set of elementary graph operations, as summarized in Figure 3.

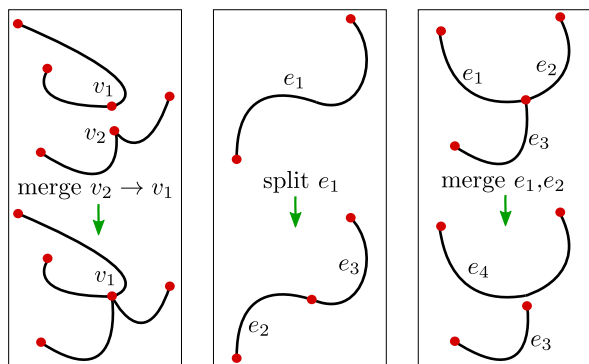


Figure 3: Graph operations. The upper row shows the graphs before and the bottom row the graphs after. Left:  $v_1$  and  $v_2$  are merged and  $v_2$  is deleted. Center:  $e_1$  is split into two edges  $e_2$  and  $e_3$ . Right:  $e_1$  and  $e_2$  are merged into  $e_4$ . Since the middle vertex is not of degree 0 after the merge, it is not deleted.

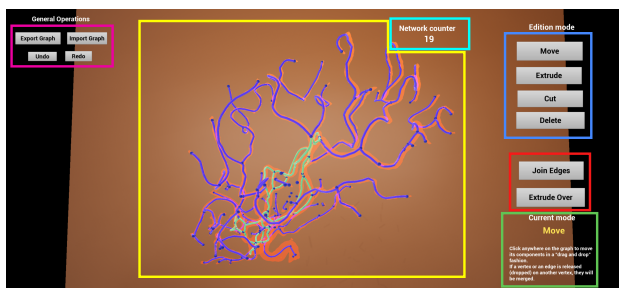


Figure 4: Screenshot of the complete interface. In the upper-left corner, the Import/Export and Undo/Redo buttons (magenta). The editing tools are on the right: the upper group consists of the basic tools (blue), and the bottom group consists of the new tools (red). The current tool is indicated in the bottom-right corner above the instructions for this tool (green). The current network being edited is in the center of the screen (yellow), and the Network Counter is placed above it (cyan).

- *Merging* vertices means to take two vertices into a single one with the same edges as those of the two previous vertices.
- *Splitting* an edge means to separate it into two edges with a degree-2 vertex connecting the two new edges.
- *Merging* edges is the inverse operation of splitting: if we have a vertex with at least two edges connecting two other vertices to it, we connect those two vertices and disconnect them from the first one. If the first vertex is now of degree 0, we remove it.

### USER INTERFACE

This section describes the interface of our system as seen by the user (cf. Figure 4). The user performs editing operations using standard 2D input devices such as a mouse or stylus. Assume that segmentation, skeletonization, and graph extraction have already been completed and the user is provided with a 3D reconstruction of a vascular network.

### Basic Tools

We first designed four tools: "Move," "Cut," "Extrude" and "Delete." Those were considered to be "basic" as we felt they

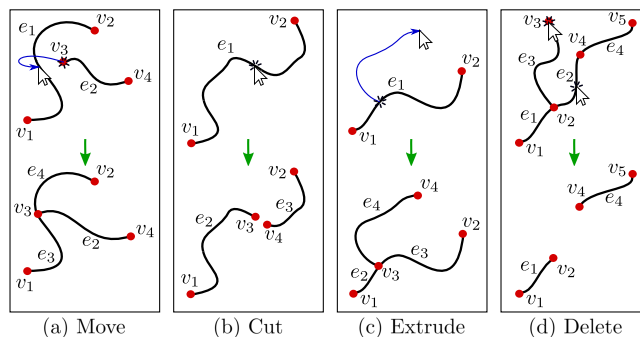


Figure 5: This figure depicts the effect of each basic tool on a simple graph. The cursor shows the mouse movement or the clicks required to use the tools. a) : Moving  $v_3$  on  $e_1$  splits it into  $e_3$  and  $e_4$  and merges  $v_3$  with the new vertex (not shown since it is already merged). b) : Cutting  $e_1$  creates two new edges and two new vertices. c) : Extruding from  $e_1$  first splits it into  $e_2$  and  $e_3$  and creates a new vertex  $v_3$ . When the mouse button is released, a new edge  $e_4$  and a new vertex  $v_4$  are created. d) : Deleting  $e_2$  simply removes the edge while Deleting  $v_3$  also removes its only connected edge ( $e_3$ ).

were natural when dealing with graphs. Figure 5 shows the effect of each of these tools.

"Move" simply displaces a vertex or an edge, thus deforming the edges' curves in an *as-rigid-as-possible* manner [32] (Figure 5.a). The subtlety here is that the network is "magnetic" in the sense that bringing a moved vertex close to another part of the network on the screen will snap this vertex exactly on the network in the 3D space. This saves the user from manually adjusting depth. When released on the network in such a way, the vertex is merged wherever it was released. If it is on another vertex, they are simply merged, and if it is on an edge, the edge is split, and the resulting vertex is merged with the moved vertex. Note that this snapping behavior can be disabled if desired by the user. Additionally, the depth of the vertex being moved can be manually changed with the mouse wheel. "Move" is also the default tool as the system switches back to it after each use of another tool.

"Cut" cuts an edge in two, thus creating two smaller edges and two new vertices at their respective tip (Figure 5.b). In terms of elementary operations, it consists of splitting an edge twice, thus resulting in three consecutive edges and then removing the middle one. The two splits are made close to each other around where the user has clicked on the curve, the deleted middle edge thus being very short.

"Extrude" creates a new edge either from an existing vertex or from an edge (thus splitting this edge first, as in Figure 5.c). This new edge's curve follows exactly the mouse, creating vessels of arbitrary shape. The tip vertex of this new edge, located at the mouse's position, snaps to nearby vertices or edges as with "Move."

Finally, "Delete" is used to remove either vertices or edges (Figure 5.d). When deleting an edge, the vertices it connects are removed as well if they become of degree 0 after removing it. If we delete a vertex, all connected edges are also deleted as described in the previous sentence.



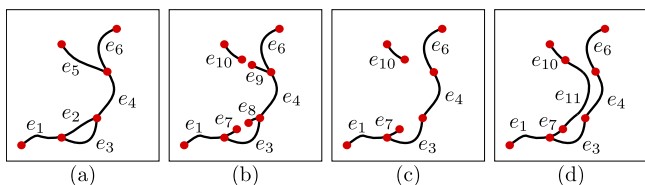


Figure 6: The successive steps to fix a Kissing Vessel with our *basic tools*. From the initial situation (a), it requires to "Cut"  $e_2$  and  $e_5$  (2 clicks) to obtain (b), then "Delete" the extra edges  $e_8$  and  $e_9$  (2 more clicks) to obtain (c) and finally "Extrude" from  $e_7$  to  $e_{10}$  to obtain (d) (1 more click-and-drag) for a total of 5 clicks (10 if counting the clicks to select the tools). Note that one could also delete  $e_2$  and  $e_5$  and then manually extrude the new vessel but this can prove very difficult for long and/or tortuous vessel.

### New Tools

We explored how these could be used to fix Kissing Vessels and Dotted Vessels, and it appeared that they required too many clicks and precision to be comfortable. As shown in Figure 6, it requires five operations to fix a single Kissing Vessel. Fixing Dotted Vessels with this toolset would require as many extrusions as there are "dots" composing the vessel. Considering this to be a significant drawback of our hybrid approach, we looked for novel ways to tackle those problems. Our collaborators concurred on the need to provide ad hoc tools to solve our two key problems.

"Join Edges" *joins* edges as it creates a new path between two edges by doubling the shortest path between their respective vertices. An intuitive roadwork analogy would be splitting an already-existing highway into different lanes. The user decides where the lane begins and ends and the system infers where to put the hedges separating them. It is typically used with a single-edge (i.e., two-vertices) path, where the Kissing Vessel is represented by a single edge, as shown by Figure 7. The algorithm can be summarized as 1) find the shortest path between two edges  $e_1$  and  $e_2$ , 2) double the edges connecting the vertices of this path as a single edge  $e_3$  and 3) merge  $e_1$ ,  $e_2$  and  $e_3$  to obtain a single edge  $e_4$ . The doubled path can technically be of any length, shape or number of vertices. Note that if there are multiple shortest paths, then the choice between the two is random. For this reason, "Join Edges" should only be used in the appropriate situation (i.e., to fix Kissing Vessels). With this new tool, we reduce the user's interaction to a single click-and-drag operation per Kissing Vessel. Moreover, the geometric information is enough to create the new path, without having to re-use the volumetric data again.

"Extrude Over" works in a manner similar to "Extrude": we start from any point on the network and start extruding freely. Network pieces passed over by the mouse are erased and replaced by a single curved edge connecting them all as shown by Figure 8. Again, this reduces the required number of clicks to a single click-and-drag per Dotted Vessel, regardless of how many pieces compose the entire vessel. This solution finds its inspiration in the name of the problem. Indeed, since "Dotted Vessels" evoke "dotted lines," it appeared natural to fix them in the same way we "connect the dots" with a pen. Note that



Figure 7: Join Edges. The edge representing the Kissing Vessel (circled with a dashed white curve) is adjacent to two pairs of edges (left), but each pair should be a single vessel (right). The user clicks on the first edge and drags the mouse to the second edge to join both edges together. While dragging, a straight yellow cylinder appears from the starting point on the edge to the mouse's current location to indicate that the join is happening. Moreover, the user receives direct feedback of the path that will be created as the corresponding edges are colored in yellow. The cyan trace shows the mouse movement to obtain the result on the right. This operation is much simpler than the method shown in Figure 6. Note that the result is independent of where on the edge the user clicked exactly. What matters is *which* edges have been clicked and released on.

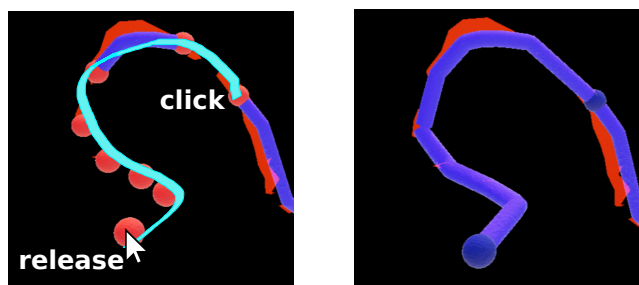


Figure 8: Extrude Over. Whenever an isolated (degree-0) vertex or both the end-vertices of an edge are passed over, they are colored red and marked to be removed. Once the extrusion is over (the user releases the mouse button) those pieces are linearly interpolated to create a single new edge. The cyan trace indicates the single mouse movement required to fix the Dotted Vessel. The difference with "Extrude" is that "Extrude Over" only connects existing pieces of a network while "Extrude" allows to freely sculpt new edges in the air. An advantage of this principle is that the mouse stroke does not need to match the pieces (short edges) exactly when using "Extrude Over".

contrary to "Extrude," the final shape does not depend on the user's exact stroke but solely on the pieces themselves. Indeed, these pieces are connected by straight curve segments, regardless of how the user connects them. This allows users to focus on the pieces only, without having to care about their gesture precision.

### Defects Identification

Even though we could expect medical experts to identify the location of the potential defects by themselves, we also wanted to provide them with practical ways to automatically indicate them. For Kissing Vessels, we considered implementing the detection method described in [11] but as we mentioned before, it still fails in some cases (e.g. vessels kissing on a long distance). Additionally, we wanted to explore other ways to highlight Kissing Vessels, if possible using our graph representation. We are thus using *cycles* to indirectly identify Kissing Vessels. If we consider a vascular network to be *connected* in

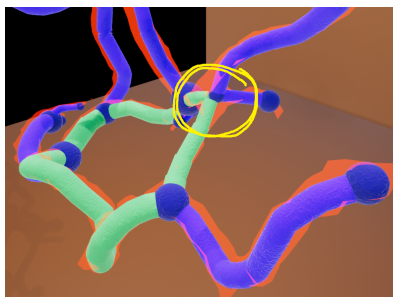


Figure 9: A small network with a Kissing Vessel. The presence of the Kissing Vessel (circled in yellow) creates a cycle (shown in green instead of blue), but it is still up to the user to find it exactly (or to decide that this cycle is normal).

Graph Theory terms (i.e., for any two vertices of the graph, there exists a path between them), there cannot be a Kissing Vessel without there being a cycle. Of course, the converse is not true since, as mentioned before, there can be actual cycles in the human brain (e.g. the "Circle of Willis"). That is why we chose to merely visualize cycles by changing the color of the corresponding edges. It is then up to the user to use their medical knowledge to decide whether this cycle is a problem or not. Moreover, we said we "indirectly" identify Kissing Vessels because it remains up to the user to locate where they are on the cycle exactly. Figure 9 shows how Kissing Vessels and cycles relate to each other.

Regarding Dotted Vessels, we again chose to give indirect indications of their presence. Instead of pointing to where they are exactly, we merely indicate how many connected components are present in the network. Since most users of our system would not be familiar with Graph Theory, we called this the "Network Counter" (c.f. Figure 4). By looking at it, users can instantly see if there are more than one sub-network, indicating that they should look for disconnected pieces in this network. We chose this indirect approach because we assumed that 1) Dotted Vessels were fairly easy to spot and 2) that adding more information to the interface would be too much.

**IMPLEMENTATION**

Let us now describe our system in more detail. We start by establishing some general concepts we adopted for our work and follow by detailing how our system works and the functionalities it offers.

**Work Environment**

Our system was implemented using C++ and relies on Unreal Engine 4 (UE4) for its interface. The main advantage of using UE4 is that it allows for high-quality graphics with little effort and provides convenient Graphical User Interface (GUI) design tools. All the code handling the skeletonization and the graph extraction is independent of UE4.

**Graph Extraction Pipeline**

Our system takes DICOM data as input and applies the "Marching Cubes (MC)" algorithm [22] against a user-specified iso-value to obtain both its isosurface mesh representation and

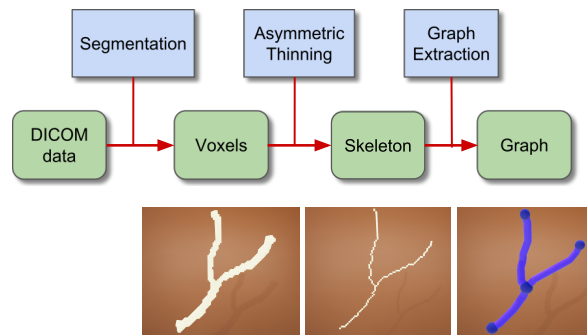


Figure 10: The complete graph extraction pipeline. We can see the successive shapes that our data take (middle) and how we go from one to the next (top). The bottom row gives an example of each representation.

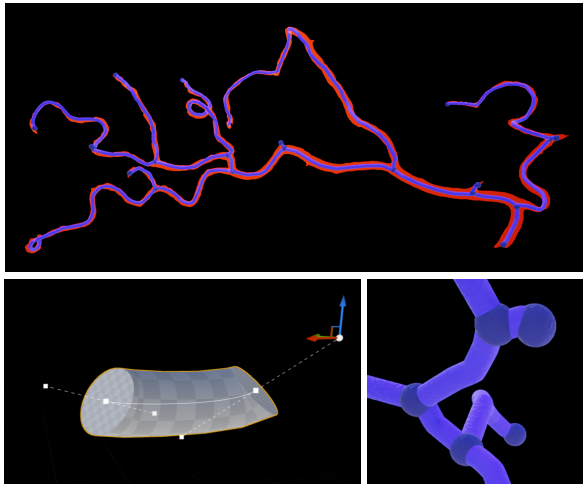
voxel-space occupancy data. By "occupancy data," we mean which voxels are occupied by the isosurface and its interior. We then apply the "Asymmetric Thinning Scheme" [9] to this voxel set to obtain a 1-voxel-wide skeleton. Finally, we apply an upgraded version of the graph extraction algorithm described in [18]. We improved this algorithm to handle multiple connected components, which are present when we consider Dotted Vessels. Our algorithm creates a graph comprising of a set of vertices and the set of edges connecting them. We also compute the approximate radius of each junction (vertex). The curves are made from the successive voxels' center positions and are then re-sampled to require a minimum of control points without losing much in shape precision. We thus obtain a piece-wise spline curve defined by a series of control points and tangents. Note that these are only *internal* control points handling the looks of the edges and are not directly manipulated by the users. The complete pipeline is depicted in Figure 10. It is important to note that although this is how we chose to extract the graph of the vascular network, other methods can be used. Our interaction model is solely based on a graph representation and would still work with other pipelines resulting in a similar graph.

**Network Visualization**

For the graphic representation of our graph, we use simple spheres for the vertices and UE4 objects called *SplineMeshes* for the edges. These allow us to easily obtain smooth cylindrical vessels with lightweight representation (i.e., the graph's actual surface mesh is generated at runtime). The spheres' size match the vertices' radii, and the curves' width is interpolated between the corresponding edge's vertices' radii. However, we let the user fix a constant width and radius over the whole network since we observed that it was easier to manipulate and understand the topology this way. Finally, we superpose the surface obtained with the MC algorithm by rendering it in transparent red to give the user some reference of the original data. Figure 11 shows a ready-to-use vascular network as represented by our system.

**COMPARISON STUDY**

To support our claim that the two new tools are more natural and efficient to solve the Dotted Vessel and Kissing Vessel



**Figure 11:** A vascular network as shown in our system (top). The blue cylindrical curves are the user-editable network, and the transparent red surface represents the surface extracted from the DICOM data with the Marching Cubes algorithm. Vertices are larger than the connected edges and of darker color, so they are easier to distinguish (bottom right). The edges are made from sequenced *SplineMeshes* whose shapes are defined by control points and tangents (bottom left). These points and tangents are not directly manipulated by the user and are shown here only for illustrative purpose.

problems, we designed a study comparing them with our basic tools ("Cut," "Delete" and "Extrude"). Note that in this section, "Move" is part of both toolsets, as we see it as an essential tool in any case. The study consisted of a practice phase and two comparison phases. In the practice phase, participants learned how to use the whole toolset. In the first comparison phase, *phase one*, we compared the two tool subsets in a within-subject manner. Each participant solved a series of simple problems, each of which was addressed by using one of the toolsets exclusively and then by the other toolset. In the second phase, *phase two*, we assessed which tools were preferred by the participants by allowing them to use them all freely. Additionally, phase two assessed whether our system allowed users to fix actual vascular networks with reasonable effort.

### Participants

Nine people participated in the study. They were all computer science students from the graphics or HCI field. Their daily work usually involves performing point-and-click or drag-and-drop operations, or manipulate 3D objects on a computer screen for hours. Similarly, we expect our target users (medical doctors) to be familiar with 3D-based volume-editing or similar software, as it is now standard practice in the medical field. As such, we believe that the two groups' skills in this matter are close enough so that the CS students' performance is relevant to our target group. Our participants' age ranged from 23 to 33 years old and were mainly male, with only one female participant. They were given a 1'000 JPY (~10 USD) book gift card as compensation for their participation. It took about 40 minutes on average for each participant to complete the study.

### Study Procedure

Participants were first introduced to the system in general by listening to a descriptive introduction and watching an introductory video describing each tool individually. Furthermore, they were provided a summary, "cheat sheet" containing an imaged description of each tool for quick reference. Both the cheat sheet and the videos were available for consultation at any time during the study and were emphasizing visual over textual explanation.

They were then presented with the system and told they were going to practice using it. Indeed, they were asked to solve a single problem by following the instructions of the examiner. Solving this problem required to use each tool at least once so they could understand how they worked. The examiner gave the successive instructions to participants, and they were encouraged to ask questions at any time. After completion of this training process, they could freely use the system for a moment if they wished. Once they considered they had understood how the system worked, we moved to the two comparison phases.

Participants were then told of the two-phase process without explaining what the second phase would be. In the first phase, they were presented with a series of minor *problems*, vascular networks (or subsets of) displaying one or more defects. Problems can be seen as a kind of "puzzle" that participants had to solve for a given *solution*, which is a target, repaired network that participants had to reproduce. Half of the participants first worked with the basic tools and the other half first worked with the new tools, and each problem had to be solved both ways before moving to the next. Moreover, they were told that they could see the target solution on another screen. The system showing the goal network was the same as the one described in this paper except that all editing tools were disabled and only the camera controls remained. This allowed participants to consider both the solution and their current work in the same way, i.e., with the same camera controls. The complete experimental set-up can be seen in Figure 12 and two examples of problems can be seen in Figure 13. We measured both the time and number of clicks (excluding camera movements) required to perform the tasks in the two different ways. Furthermore, participants were asked the "Single Ease Question (SEQ)" [30] after solving each problem each way (8 times in total for phase one). The SEQ is "Overall, how difficult or easy was the task to complete?" and the answer is on a scale from 1 (very difficult) to 7 (very easy).

In the second phase, participants were told that they would have to solve another series of 6 problems of increasing difficulty. This time, participants had complete control over which tools they could use. The goal of this second phase was to assess whether "Extrude Over" and "Join Edges" were preferred over the other tools or not when users were given the choice on how to solve a particular problem. The difficulty was measured in the number of Kissing Vessels ( $N_{KV}$ ), the number of Dotted Vessels ( $N_{DV}$ ) and the total number of defects ( $N_D$ ) that the vascular networks presented. Figure 14 shows an example of a phase two problem. Participants were asked to confirm when their work was done, i.e., when they thought their net-



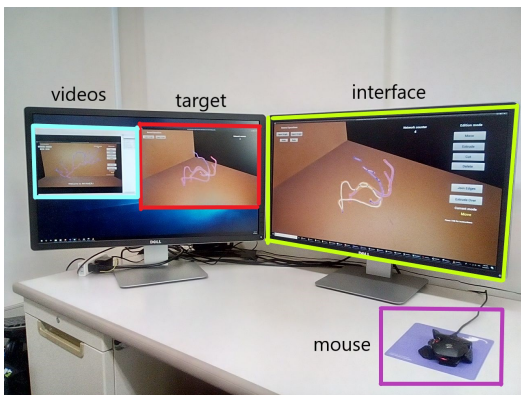


Figure 12: The experimental set-up. On the right, the main screen with the system used to solve the problems (green). On the left, the helper screen with the system showing the target vascular network (the solution, red) and the video player to play the videos (cyan). Participants used the mouse to use the system (purple).

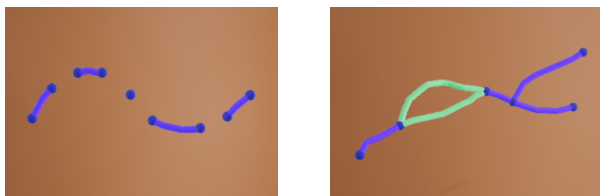


Figure 13: Two examples of the kind of problems participants had to solve during the first phase.

work was the same as the solution. All participants faced the same problems in the same order in a within-subject manner. Again, they were asked the SEQ after solving each problem.

Finally, participants were asked to answer a questionnaire consisting of 27 qualitative questions regarding how they felt using the system, if the instructions were clear, if they found the new tools useful, etc. All questions were answered using a five-level Likert scale [20]. The ten last questions of the questionnaire were borrowed from the "System Usability Scale (SUS)" [6]. This scale is based on a five-level scale then translated and scaled to a general score ranging from 0 to 100, the average score being 68. The goal of the questionnaire was to qualitatively assess both the usability of the system and the quality of the evaluation process.

### Results

The results of phase one are summarized by Figure 15. This figure shows that there is a distinct difference between the two methods, in terms of the number of clicks and time to solve the problems of phase one, as well as the difficulty felt by participants. It shows that on average, participants were ~2.5 times faster in performing their task with the new tools. It is even more flagrant for the number of clicks with the new tools requiring ~3.8 times fewer clicks on average. While there are inconsistencies in the participants' evaluation of difficulty when employing the basic tools, there is near

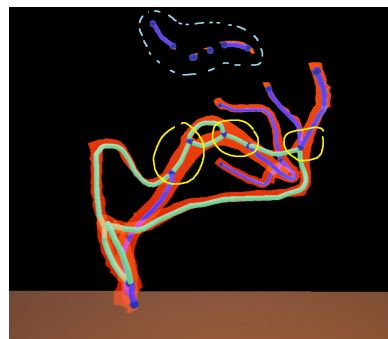


Figure 14: An example of a bigger problem to solve during phase two. It corresponds to problem (e) in Table 1, with  $N_{KV} = 3$ ,  $N_{DV} = 1$ ,  $N_D = 4$ . The three Kissing Vessels are circled with continuous yellow curves and the Dotted Vessel is circled with a dashed cyan curve. Note that  $N_{KV}$  counts the number of times where vessels are actually kissing rather than the number of vessels that are kissing. That is, if two vessels kiss only once, we count  $N_{KV} = 1$  and if two vessels kiss three times we count  $N_{KV} = 3$ . However,  $N_{DV}$  counts the number of vessels that are dotted and not the number of "dots" that compose the vessels.

unanimous consensus on the fact that tasks performed with the new tools were "very easy." These statements remain true when fixing both Kissing Vessels or Dotted Vessels. For phase two, we can confirm that each problem was indeed solved with reasonable effort by looking at Table 1. Moreover, Figure 16 clearly shows that the novel tools were used more than the basic ones.

The questionnaires gave very positive outputs as the general satisfaction for the system was rated as 4.71 ( $\sigma = 0.62$ ) over a maximum of 5 and a SUS score of 81.67 over 100 which is well above the average SUS score (68). Moreover, participants seemed to be satisfied with their experience as shown both by the evaluation-related questions (score of 4.41,  $\sigma = 0.89$ , over 5) and by their positive comments at the end of the experiment. Although the feedback was generally very positive, some participants found it rather difficult to compare their progress to the solution. Indeed it was possible that participants would think they were done but had actually missed one or more defects and we also recorded the number of such mistakes. Since this aspect is directly related to the study and not the purpose of our system, we do not consider this to be a flaw in the system and expected to have a minimal number of them. We recorded an exact number of 1 mistake per participant, thus confirming that the side screen was an adequate method to understand a problem's solution, even if it made the participant's experience a little uncomfortable. Another remark that often came up was that it was somewhat tedious to always select the tool before using it. Almost all participants would have preferred to simply switch between tools whenever they wanted.

**Extrude Over.** Almost all participants tended to directly start by fixing the Dotted Vessels rather than the Kissing Vessels. Our interpretation is that the Dotted Vessel problem and how to fix it is straightforward to understand for anyone by its "dotted" nature and also easy to spot, even in big networks. However, users tended to use "Extrude Over" multiple times per Dotted



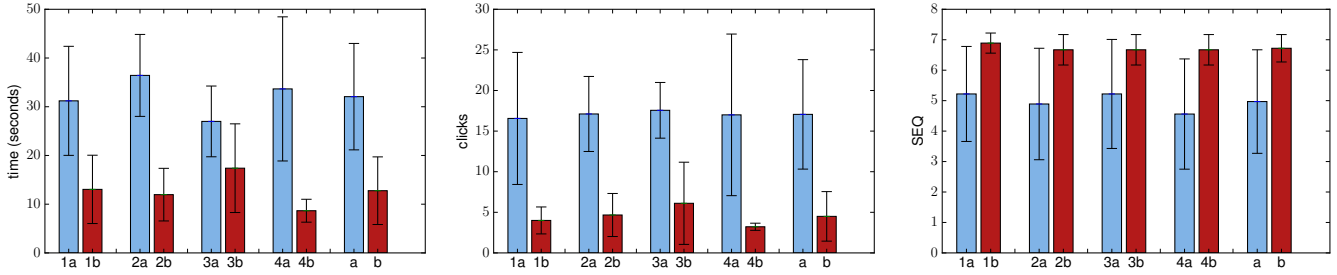


Figure 15: The results of phase one in terms of time (left) and number of clicks (center) required to solve the problems and their perceived difficulty (right). In blue (left bars), the values obtained with method a) (basic tools) and in red (right bars), the values obtained with method b) (novel tools). The black error bars are the standard deviation. The rightmost pairs of bars in each figure is the overall average for both methods.

Problem	$N_{KV}$	$N_{DV}$	$N_D$	time(s)	#clicks	SEQ
(a)	0	1	1	21.48	7.78	6.67
(b)	1	0	1	21.34	4.78	6.44
(c)	1	3	4	126.03	46.56	4.44
(d)	1	4	5	90.46	29.67	4.56
(e)	3	1	4	128.52	37.44	3.28
(f)	4	4	8	260.85	86.77	2.72

Table 1: The average time and number of clicks required to solve each problem of phase two compared to their characteristics. These results show that our system is suited to repair vascular networks with reasonable time-wise and click-wise effort, even for new users. We can see that  $N_D$  is a good indicator of the difficulty of the task, although Kissing Vessels were perceived as being harder to fix than Dotted Vessels by participants. We observed that the time mainly increased because it became more and more difficult to compare the problem to the solution. In comparison, fixing a complete left brain hemisphere with 13 Kissing Vessels and 19 Dotted Vessels took an expert user around 8 minutes and 103 clicks.

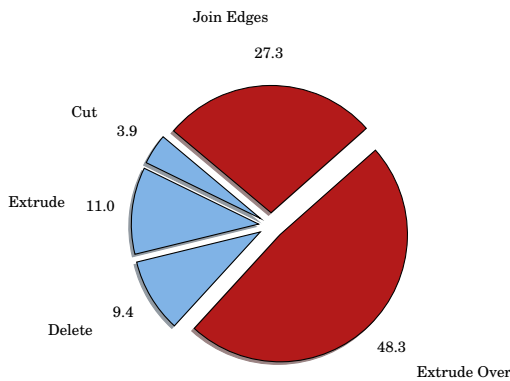


Figure 16: Usage count for each tool during phase two. The blue slices correspond to the basic tools (Cut, Extrude, Delete) and the red slices to the novel tools (Extrude Over, Join Edges). All users preferred using the novel tools.

Vessel instead of the minimum requirements of only once per Dotted Vessel. Some users needed to use it indeed only once but some required two or more uses. Moreover, we observed that they tended to stop at longer Dotted Vessel pieces and start

again after them thus fixing a single Dotted Vessel in a two-step process. We believe that they felt they had to follow the longer pieces precisely for them to be replaced even though it is not the case. Indeed, if both ends of any piece, no matter the length, were passed over, this piece would be replaced, as it was shown in the introductory video. We believe that this detail of the video was missed by most participants. Those observations are corroborated by the average number of "Extrude Over" uses required to fix all Dotted Vessels. If we consider the average number of uses of "Extrude Over" and  $N_{DV}$  for each problem we see that participants required an average of 1.48 "Extrude Over" operations per Dotted Vessel, thus 48% more than the minimum of 1 "Extrude Over" operation per Dotted Vessel. Interestingly, we did not mention the red coloring of the vertices to be replaced by the new curved blue edge (as described in Figure 8) at any point but still observed that it was a natural indicator for participants of the expected result. If a vertex was not colored in red, they immediately stopped, undid this last "Extrude Over" operation and started over.

**Join Edges.** The most important observation we made was that this tool was the hardest to understand. Most "undo" operations were performed right after a "Join Edges" operation with unwanted results. Even though it was generally clear for participants how to use it during phase one when the single defects were focused on, they were more confused when facing bigger situations. Moreover, they tended first to try fixing Kissing Vessels with "Join Edges" and then switch to other tools when the results were not as expected. This typically happened in situations such as the one shown in Figure 17, where they were joining an edge to the kissing point of the vessels rather than the other part of the actual vessel. Even if they switched to the basic tools to fix a problem, they still tried again with "Join Edges" for the next problem. The result is that by the time they reached the final problem, all participants either used "Join Edges" perfectly or failed for exactly one Kissing Vessel. By "perfectly" we mean a single time per Kissing Vessel and without any mistake in the resulting topology. Unfortunately, the yellow coloring of the edges did not seem to help users understand the expected result. Finally, some participants tried to use "Join Edges" in a way similar to "Extrude" or "Extrude Over." It hence occurred to us that the name "Join Edges" was confusing and should eventually



Figure 17: A bad usage of Join Edges. Almost all participants tried at least once to join an edge with the edge representing the Kissing Vessel. The correct usage was already shown in Figure 7.

be replaced by a more faithful description of its effect such as "Split Path." We will use this latter denomination from now on.

### FEEDBACK FROM PROFESSIONALS

We evaluated our system further by presenting it to three neurosurgeons to get general feedback from the point-of-view of our target users. Those doctors were not involved in the development of the system to ensure the objectivity of their feedback. We used the conclusions drawn from our first study and the comments of its participants to make minor improvements (e.g. changing the name "Join Edges" to "Split Path" or disabling "Move" as the default tool). This time, we decided to show all the potential of our system to our participants by focusing on the new tools. Instead of showing a video, the examiner manually demonstrated how to use the system. The examiner emphasized the details of how to use "Extrude Over" and "Split Path", highlighting common mistakes and best practices. The other tools were also demonstrated but with only little emphasis. Participants then also solved a practice problem with multiple Kissing Vessels and Dotted Vessels. They then solved the two hardest problems of phase two. They were told that the two new tools were entirely sufficient to perform their task but were allowed to use the others if desired. The first difference with the first study is that whenever they mistakenly used a tool, the examiner notified them of their mistake and guided them on how to use it properly. Another major difference is that they had no "solution" provided in any way. This was meant to evaluate how well our defects indicators (green cycles and Network Counter) could convey the information required by actual medical doctors to perform their task. After completing their tasks, they were asked questions directly related to the medical field and gave qualitative answers. These questions involved for example whether our system would be useful for their work or if they had seen similar systems somewhere else. Similarly to the first study, participants were asked to tell the examiner when they were done with their tasks, thus again leaving the possibility of making mistakes. No defect was missed by the participants, and we thus believe that our indicators are a valid way to identify defects in the extracted vascular network.

Overall, our medical experts were more than satisfied by our system and considered that it was a fit and accurate means of solving problems they often encounter during their daily

practice. They were generally very interested in the center-line/graph representation of vascular networks and seemed to find this to be a truly novel approach. "Split Path" particularly caught their attention and was mentioned as being the most exciting feature. When asked how they currently solved this kind of issue, they mentioned tedious processes involving lots of manual interaction. They all agreed that our system could greatly reduce their effort in that matter. All of them noted the potential of free-form sculpting (the "Extrude" tool) for bypass and simulation. As they noted, bypass surgery simulation indeed requires doctors to add new vessels that are not existent in the scanned image and is therefore generally not supported by typical segmentation systems. Although they already have means of discussing the addition of vessels in vascular networks, these are not 3D systems.

Additionally, two of our participants suggested to enrich the visualization with meaningful information. For instance, there is currently no way to distinguish two vessels aside from their shape or location. Coloring them in a way that suggests this topology would certainly improve the understanding of the general topology of the network. One of them suggested changing the vessels' coloration depending on their difficulty to operate or their nature (vein or artery). Another very interesting suggestion was to integrate physical constraints such as elasticity into our system. Vessels might, for instance, change color depending on the elastic stress resulting from the deformation done by the users. They could also limit the deformation to keep users from performing operations that would break blood vessels. This would allow neurosurgeons to get feedback on the feasibility of a particular intervention.

### LIMITATIONS AND FUTURE WORK

The main limitation of our system is the fact that although we rely on actual volumetric data to reconstruct the initial network, all subsequent operations are detached from it. If we consider free-form extrusion or sculpting for the practices described above, it is a desirable property, but when focusing on reconstructing a network as faithful as possible, it becomes necessary to be precise in the location of the fixed vessels' centerlines. Our method to fix Kissing Vessels after user intervention relies purely on the surrounding vessels' geometry and thus provides no guarantee of its exactitude. This comment also applies to the vessels' radii since their re-evaluation after applying "Split Path" is not trivial. Experience showed that using "Split Path" produced results generally close enough to the original volume, but more work will be required in the quest for absolute accuracy.

As suggested by the second study's participants, some additional efforts towards the visual quality of our representation would undoubtedly improve the users' comfort. By that, we do not mean the quality of the "graphics" but rather how they can be used to translate more information visually. We envision the ability to switch between various coloring to highlight different attributes of the network such as hierarchical relationships, thickness, length, etc. as this information might prove quite useful to medical practitioners. This aspect could be further extended with various fluid-dynamics simulations visualization. Moreover, a full network is quite intricate and

complex, and it would help to visually isolate a part of it to focus on where the defects are.

## CONCLUSION

In this paper, we presented INVANER, an interactive system for medical practitioners, in particular, neurosurgeons, to reconstruct faithful vascular networks from defective data and conveniently visualize them. Our system represents vascular networks as graphs enhanced with cylindrical curves and modern graphics empowered by Unreal Engine 4. Furthermore, it provides a set of graph-based tools to edit and more importantly repair networks from recurring defects called "Kissing Vessels" and "Dotted Vessels." Two of those tools, "Split Path" and "Extrude Over," were shown by a study to be means fit to solve these respective defects, even more so when compared with other, more basic tools. Our study further assessed the general usability of our system and showed unanimous satisfaction from participants. Its examiner also made qualitative observations that offer more nuanced and complex insights into the participants' behavior. They remain subjective in nature, and the arguably limited number of participants involved in the study, unfortunately, prevents us from drawing absolute conclusions. We still, however, believe that the results of the study are encouraging enough to pursue research in that direction. This is further supported by our second study, which showed that our system was considered to be very promising by professional medical practitioners. Future work involves a continued exchange with and feedback from medical doctors in order to test our system on the field with real-life situations. Furthermore, we see this work as an intermediate step towards providing a full stack of tools for medical doctors to centralize all vascular networks-related practices.

## ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 17H00752. The authors would like to thank Dr. Kin from The University of Tokyo and his team for their collaboration. The first author would also like to fondly thank the people from the Tōdai cafeteria for their amazing Akamon Ramen. Surely, one works best when the food is the greatest.

## REFERENCES

- [1] Sasakthi S. Abeysinghe and Tao Ju. 2009. Interactive skeletonization of intensity volumes. *The Visual Computer* 25, 5 (01 May 2009), 627–635. DOI: <http://dx.doi.org/10.1007/s00371-009-0325-5>
- [2] Marco Attene and Bianca Falcidieno. 2006. ReMESH: An interactive environment to edit and repair triangle meshes. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. IEEE, 41–41.
- [3] J. Andreas Bærentzen, Rinat Abdrashitov, and Karan Singh. 2014. Interactive Shape Modeling Using a Skeleton-mesh Co-representation. *ACM Trans. Graph.* 33, 4, Article 132 (July 2014), 10 pages. DOI: <http://dx.doi.org/10.1145/2601097.2601226>
- [4] Ilya Baran and Jovan Popović. 2007. Automatic Rigging and Animation of 3D Characters. *ACM Trans. Graph.* 26, 3 (July 2007). DOI: <http://dx.doi.org/10.1145/1276377.1276467>
- [5] Simone Barbieri, Pietro Meloni, Francesco Usai, L. Davide Spano, and Riccardo Scateni. 2016. An interactive editor for curve-skeletons: SkeletonLab. *Computers & Graphics* 60 (2016), 23 – 33. DOI: <http://dx.doi.org/10.1016/j.cag.2016.08.002>
- [6] John Brooke. 1986. System usability scale (SUS): a quick-and-dirty method of system evaluation user information. *Reading, UK: Digital Equipment Co Ltd* (1986), 43.
- [7] Jacques Marescaux Christophe Koehl, Luc Soler. 2002. PACS-based interface for 3D anatomical structure visualization and surgical planning. *Proc.SPIE* 4681 (2002), 4681 – 4681 – 8. DOI: <http://dx.doi.org/10.1117/12.466924>
- [8] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and others. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756.
- [9] Michel Couprie and Gilles Bertrand. 2016. Asymmetric parallel 3D thinning scheme and algorithms based on isthmuses. *Pattern Recognition Letters* 76 (2016), 22 – 31. DOI: <http://dx.doi.org/10.1016/j.patrec.2015.03.014> Special Issue on Skeletonization and its Application.
- [10] Tim E Darsaut, J Max Findlay, Elsa Magro, Marc Kotowski, Daniel Roy, Alain Weill, Michel W Bojanowski, Chiraz Chaalala, Daniela Iancu, Howard Lesiuk, John Sinclair, Felix Scholtes, Didier Martin, Michael M Chow, Cian J O'Kelly, John H Wong, Ken Butcher, Allan J Fox, Adam S Arthur, Francois Guilbert, Lu Tian, Miguel Chagnon, Suzanne Nolet, Guylaine Gevry, and Jean Raymond. 2017. Surgical clipping or endovascular coiling for unruptured intracranial aneurysms: a pragmatic randomised trial. *Journal of Neurology, Neurosurgery & Psychiatry* 88, 8 (2017), 663–668. DOI: <http://dx.doi.org/10.1136/jnnp-2016-315433>
- [11] Anneleen De Clercq, Wilfried 1966-promotor (viaf)33975937 Philips, and Aleksandra 1969-copromotor (viaf)305674182 Pizurica. 2011. Accurate segmentation of vascular structures. (2011). <http://lib.ugent.be/catalog/rug01:001805428>
- [12] J. Dolz, L. Massoptier, and M. Vermandel. 2015. Segmentation algorithms of subcortical brain structures on MRI for radiotherapy and radiosurgery: A survey. *IRBM* 36, 4 (2015), 200 – 212. DOI: <http://dx.doi.org/10.1016/j.irbm.2015.06.001>
- [13] Antoine Iannessi, Pierre-Yves Marcy, Olivier Clatz, Anne-Sophie Bertrand, and Maki Sugimoto. 2018. A review of existing and potential computer user interfaces for modern radiology. *Insights into Imaging* 9, 4 (01 Aug 2018), 599–609. DOI: <http://dx.doi.org/10.1007/s13244-018-0620-7>

- [14] Takeo Igarashi, Naoyuki Shono, Taichi Kin, and Toki Saito. 2016. Interactive Volume Segmentation with Threshold Field Painting. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 403–413. DOI : <http://dx.doi.org/10.1145/2984511.2984537>
- [15] H. Jiang and Noam Alperin. 2004. A new automatic skeletonization algorithm for 3D vascular volumes. In *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, Vol. 26 II. 1565–1568. DOI : <http://dx.doi.org/10.1109/IEMBS.2004.1403477>
- [16] Jan Kretschmer, Christian Godenschwager, Bernhard Preim, and Marc Stamminger. 2013. Interactive Patient-Specific Vascular Modeling with Sweep Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 19 (12 2013), 2828–37. DOI : <http://dx.doi.org/10.1109/TVCG.2013.169>
- [17] Michele Larobina and Loredana Murino. 2014. Medical image file formats. *Journal of digital imaging* 27, 2 (2014), 200–206.
- [18] Fan Li. 2016. *Segmentation and Symbolic Representation of Brain Vascular Network : Application to ArterioVenous Malformations*. Theses. Université Paris-Est. <https://tel.archives-ouvertes.fr/tel-01526456>
- [19] Kristína Lidayová, Hans Frimmel, Chunliang Wang, Ewert Bengtsson, and Örjan Smedby. 2016. Fast vascular skeleton extraction algorithm. *Pattern Recognition Letters* 76 (2016), 67 – 75. DOI : <http://dx.doi.org/https://doi.org/10.1016/j.patrec.2015.06.024> Special Issue on Skeletonization and its Application.
- [20] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).
- [21] Jin Liu, Min Li, J. Wang, Fangxiang Wu, T. Liu, and Yi Pan. 2014. A survey of MRI-based brain tumor segmentation methods. *Tsinghua Science and Technology* 19, 6 (Dec 2014), 578–595. DOI : <http://dx.doi.org/10.1109/TST.2014.6961028>
- [22] William E. Lorensen and Harvey E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (Aug. 1987), 163–169. DOI : <http://dx.doi.org/10.1145/37402.37422>
- [23] Ahmed M. Mharib, Abdul Rahman Ramli, Syamsiah Mashohor, and Rozi Binti Mahmood. 2012. Survey on liver CT image segmentation methods. *Artificial Intelligence Review* 37, 2 (01 Feb 2012), 83–95. DOI : <http://dx.doi.org/10.1007/s10462-011-9220-3>
- [24] Y Mori, Takeo Igarashi, Ryo Haraguchi, and K Nakazawa. 2007. A Pen-based Interface for Generating Graphical Reports of Findings in Cardiac Catheterization. *Methods of information in medicine* 46 (02 2007), 694–9. DOI : <http://dx.doi.org/10.1055/s-0038-1625430>
- [25] Kazutaka Nakashima, Yuki Koyama, Takeo Igarashi, Takashi Ijiri, Shin Inada, and Kazuo Nakazawa. 2016. Interactive Deformation of Structurally Complex Heart Models Constructed from Medical Images. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers (EG '16)*. Eurographics Association, Goslar Germany, Germany, 49–52. DOI : <http://dx.doi.org/10.2312/egsh.20161012>
- [26] Rosalie Plantefève, Samuel Kadoury, An Tang, and Igor Peterlik. 2017. Robust Automatic Graph-based Skeletonization of Hepatic Vascular Trees. In *CIIV STENT 2017 (Proceedings CIIV STENT 2017)*. Québec, Canada. <https://hal.inria.fr/hal-01576771>
- [27] Johannes Reichold, Marco Stampanoni, Anna Lena Keller, Alfred Buck, Patrick Jenny, and Bruno Weber. 2009. Vascular graph model to simulate the cerebral blood flow in realistic vascular networks. *Journal of cerebral blood flow and metabolism : official journal of the International Society of Cerebral Blood Flow and Metabolism* 29 8 (2009), 1429–43. DOI : <http://dx.doi.org/10.1038/jcbfm.2009.58>
- [28] Patrick Saalfeld, Alexandra Baer, Uta Preim, Bernhard Preim, and Kai Lawonn. 2015. Sketching 2D Vessels and Vascular Diseases with Integrated Blood Flow. In *GRAPP*.
- [29] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. 2016. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters* 76 (2016), 3 – 12. DOI : <http://dx.doi.org/10.1016/j.patrec.2015.04.006> Special Issue on Skeletonization and its Application.
- [30] Jeff Sauro and Joseph S. Dumas. 2009. Comparison of Three One-question, Post-task Usability Questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1599–1608. DOI : <http://dx.doi.org/10.1145/1518701.1518946>
- [31] David Selosse, Jérémie Dequidt, and Laurent Grisoni. 2012. *A Sketch-Based Interface for Annotation of 3D Brain Vascular Reconstructions*. Research Report RR-7954. INRIA. 19 pages. <https://hal.inria.fr/hal-00698832>
- [32] Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Geometry Processing*, Alexander Belyaev and Michael Garland (Eds.). The Eurographics Association. DOI : <http://dx.doi.org/10.2312/SGP/SGP07/109-116>
- [33] Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, and Alexandru Telea. 2016. 3D Skeletons: A State-of-the-Art Report. *Computer Graphics Forum* 35, 2 (2016), 573–597. DOI : <http://dx.doi.org/10.1111/cgfm.12865>



- [34] Julien Touati, Marco Bologna, Adeline Schwein, Francesco Migliavacca, and Marc Garbey. 2017. A robust construction algorithm of the centerline skeleton for complex aortic vascular structure using computational fluid dynamics. *Computers in Biology and Medicine* 86 (2017), 6 – 17. DOI: <http://dx.doi.org/10.1016/j.combiomed.2017.04.017>
- [35] Y. Wan, H. Otsuna, C. Chien, and C. Hansen. 2012. Interactive extraction of neural structures with user-guided morphological diffusion. In *2012 IEEE Symposium on Biological Data Visualization (BioVis)*. 1–8. DOI: <http://dx.doi.org/10.1109/BioVis.2012.6378577>
- [36] Chaoli Wang. 2015. A Survey of Graph-Based Representations and Techniques for Scientific Visualization. In *Eurographics Conference on Visualization (EuroVis) - STARs*, R. Borgo, F. Ganovelli, and I. Viola (Eds.). The Eurographics Association. DOI: <http://dx.doi.org/10.2312/eurovisstar.20151111>
- [37] Lijie Xu and Han-Wei Shen. 2010. Flow Web: A Graph Based User Interface for 3D Flow Field Exploration. *Proceedings of SPIE - The International Society for Optical Engineering* 7530 (01 2010), 75300. DOI: <http://dx.doi.org/10.1117/12.838659>
- [38] Ahmed Yureidini. 2014. *Robust blood vessel surface reconstruction for interactive simulations from patient data*. Theses. Université des Sciences et Technologie de Lille - Lille I. <https://tel.archives-ouvertes.fr/tel-01010973> Rédigée au 21 janvier 2014.
- [39] Fan Zhang, Pei Lu, Xiaoyun Liu, and Shoujun Zhou. 2017. Vascular centerline extraction of CTA images based on minimal path and Bayesian tracking. 1–5. DOI: <http://dx.doi.org/10.1109/CISP-BMEI.2017.8302208>
- [40] Huang Zhanpeng, Jiang Shizhong, Liu Yaohui, and Bao Susu. 2014. Interactive 3D liver vessels centerline extraction based on moving sphere model. *Proceedings - 2014 International Conference on Information Science, Electronics and Electrical Engineering, ISEEE 2014 3* (11 2014), 2100–2103. DOI: <http://dx.doi.org/10.1109/InfoSEEE.2014.6946295>
- [41] Denis Zorin, Peter Schröder, and Wim Sweldens. 1997. Interactive multiresolution mesh editing. In *SIGGraph*, Vol. 97. Citeseer, 259–268.
- [42] Radosław Łazarz. 2016. Graph-based Framework for 3-D Vascular Dynamics Simulation. *Procedia Computer Science* 101 (12 2016), 416–424. DOI: <http://dx.doi.org/10.1016/j.procs.2016.11.048>