EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2010)
M. Alexa and E. Do (Editors)

Sketch-Based Modeling of Vascular Systems: a First Step Towards Interactive Teaching of Anatomy

Adeline Pihuit[1], Marie-Paule Cani[1] and Olivier Palombi[1,2]

[1]Grenoble Universities & CNRS, Jean Kuntzmann Lab, INRIA Grenoble Rhône-Alpes, France
[2]Laboratory of Anatomy of Grenoble Universities, France

Abstract
We present a sketch-based modeling system, inspired from anatomical drawing, which constructs plausible 3D models of branching vessels from a single sketch. The input drawing typically includes non-flat silhouettes and occluded parts. We exploit the sketching conventions used in anatomical drawings to infer depth and curvature from contour and skeleton curves extracted from the sketch. We then model the set of branching vessels as a convolution surface generated by a graph of skeleton curves: while these curves are set to fit the sketch in the front plane, non-uniform B-spline interpolation is used to give them smoothly varying depth values that meet the set of constraints. The final model is displayed using an expressive rendering method that imitates the aspect of chalk drawing. We discuss the future use of this system as a step towards the interactive teaching of anatomy.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—

1. Introduction

Neither learning nor teaching anatomy is an easy task. On one hand, students not only have to memorize hundreds scientific words but also to understand and remember approximate shapes and spatial relationships of tens of human anatomical structures, which are presented to them through 2D sketches. On the other hand, professors try to make these complex structures easy to understand and remind through an incremental presentation, by progressively sketching simplified shapes on a blackboard with color chalks. In addition to the skills required to draw 2D views of complex interlinked structures such as vessels, several sketches at different resolution or from different viewpoints can be needed to fully represent an organ and show its placement relative to neighboring structures. Due to the use of blackboard and chalks, each of these sketches needs to be done from scratch, and insuring an acceptable degree of coherence between them requires a lot of training (Fig. 1).

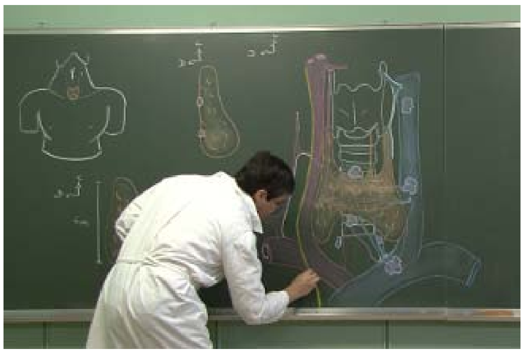


Figure 1: *A professor sketching the vascular system of the thyroid, from different viewpoints. Three colors are used for the arterial blood vessels: dark-red, light-red and white.*

Anatomy professors are attached to the pedagogy of this interactive, step by step approach: while sketching, they explain the function of the organs, enhance relationships with others, and annotate them with technical terms. Meanwhile, students are trained to reproduce the sketches on their own sheet of paper, which helps them memorize. Therefore, visualizing 3D models of the organs instead of sketching them would not be a solution.

Sketch-based modeling would be a promising tool for teaching anatomy: it would provide an *interactive virtual blackboard* where the professor would sketch as usual, but

DOI: 10.2312/SBM/SBM10/151-158

which would infer some plausible 3D geometry from the sketches, using sketching conventions taught to students. This virtual blackboard would enable the professor not only to display the resulting anatomical structures from different viewpoints and zoom factors, but also to sketch neighboring organs from other viewpoints; this would greatly ease the understanding of 3D geometry.

This paper describes a first step towards this goal: to demonstrate the feasibility of the approach, we focus on *inferring the 3D geometry of vascular structures from a single sketch*. This is a particularly challenging problem since vessels form networks of smoothly branching, self-occluding tubular shapes. This leads to complex sketches with non-planar silhouettes and multiple T-junctions on contours, a situation scarcely covered by previous sketch-based modeling systems. How can we retrieve the spatial relationship between vessels? How can we infer a plausible solution for their smoothly curved geometry? Our solution relies on simple a priori knowledge such as the tubular nature of vessels and on some specific sketching conventions used for teaching anatomy.

1.1. Previous Work

Modeling 3D shapes from 2D sketches has attracted a lot of attention for the last ten years. Two classes of problems were studied: sketch-based methods for modeling general free-form shapes on one hand, and specialized sketching systems which rely on some a priori knowledge on the object being sketched to the other hand.

Since they have little knowledge on the model being sketched, the first class of approaches [IMT99, TZF04, SWSJ05, BPCB08] tend to infer the simplest possible shape that fits a user-sketched 2D contour. The latter is usually interpreted as a flat silhouette, which restricts the set of resulting shapes. To cover complex 3D models, these interactive systems enable users to progressively fill out them, by adding simple components sketched from arbitrary viewing angle and zoom factors, or deforming control curves that define the surface [NISA07]. Sketching vascular systems using these systems would be very tedious, since lots of different viewpoints would be needed to model interlinked and self-occluding non-planar structures. Keeping in mind the application to the teaching of anatomy, the sketching strategy would also be too different from the standard approach with blackboard and chalks to result in sketches that could be reproduced and learned by students.

Other sketching systems have tackled very specific applications such as sketching clothing [TWB*07], flowers [APS08, IYKI08] or trees [IOI06, WBCG09]. To handle such complex cases, some a priori knowledge on the structure of the object being sketched is taken into account to infer the third dimension, which generally makes objects modeling easier: the user can do most of the work with a single sketch rather than needing to sketch small parts of the model from lots of different viewpoints. Our sketching system inspires from these previous object-specific methods: it also relies on some a priory knowledge of the structures -here, vascular systems- being sketched. Meanwhile, our method is the first one, to our knowledge, to tackle the case of a network of smooth, branching tubes. Thanks to our use of sketching conventions in anatomy, we are able to reconstruct these complex structures from a single sketch.

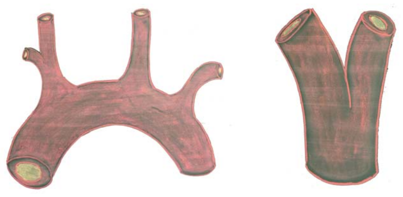


Figure 2: *Real sketches on blackboard with chalk, illustrating the anatomical sketching conventions. Left: the aortic arch. Right: a branching situation where the right vessel is slightly behind the other one.*

The only previous work that tackles the case of sketches with non-planar silhouettes and self-occlusions is the excellent work from Karpenko and Hughes [KH06]. Cups and T-junctions on contours are classified and grouped to reconstruct the occluded contours. Then, a simple physically-based system is used to infer some plausible depth. Similarly to this method, we identify and classify T-junctions, but we come out with a quite different solution for inferring depth: indeed, the specific nature of vessels enables us to directly work on a graph of smooth, skeleton curves; moreover, we exploit the specific sketching conventions used to express 3D in anatomical drawings, such as the use of hatchings to indicate highly curved regions.

1.2. Contributions

We present a sketch-based modeling system, inspired from anatomical drawing, which constructs plausible 3D models of branching vessels from a single sketch.

The first challenge is to extract depth-clues from the sketch. This is done by studying contour-strokes, hatching, T-junctions and colors. We interpret these clues using anatomical sketching conventions to build a semantic for the sketch, which is then converted into a set of *depth constraints*.

The second contribution is a method for constructing a plausible 3D geometry for the sketched vessels. We rely on skeletal implicit surfaces (namely, *convolution surfaces*) to model each vessel. However, in contrast with previous convolution-based sketching-systems [TZF04, BPCB08], computing 2D skeletons is not sufficient to infer appropriate depth to each vessel. We therefore use *depth constraints* to create a graph of smooth, 3D curves for the vessel skeletons. To do so, we exploit the smoothness of Bsplines for inferring depth while keeping an exact fit with the sketch in the other dimensions.

Finally, the resulting graph of 3D skeletons is used to generate an implicit representation of the vascular system. We display the latter in a way that imitates chalk drawing.

The remainder of this paper is organized as follows. Section 2 reviews the sketching conventions used by anatomists, with a focus on those that may bring depth clues in the case of vessels; it also derives some extra rules expressing the fact that the sketch represents a graph of branching tubes. We present our method for processing the contours and extracting depth constraints in section 3. Our method for computing 3D skeletons and reconstructing the vessels geometry is detailed in section 4. Section 5 presents our results. We then discuss directions for future work.

2. Anatomical sketching conventions for blood vessels

Anatomical sketches comply with several conventions aimed at ensuring that every one imagines the same structure in 3D. Since this project is the result of a collaborative approach with an anatomist, we started by analyzing real cases of anatomical lectures performed on a blackboard. We observed that arterial blood vessels are usually sketched with three colors (Fig 1 and 2): dark-red for contour strokes, light-red for filling the surface, and white chalk for annotating the cross-section (ie. representing the hole in the vessel).

Since blood flows through vessels, anatomists try to conform with the Kirchhoff rule, stating that the sum of area of the sections of outgoing vessels at a branching point is equal to the area of the incoming vessel's section at this point. Even if only approximated in practice, this clue is very useful to understand the relationship between vessels at branching points, as will be detailed in section 3.3.

In addition, the knowledge that vessels are tube-shaped structures can be used to infer their orientation at endpoints. While the cross-section of a tube is a circle when seen in front, it becomes an ellipse due to perspective from any other viewpoint. Thus, we can infer the orientation of a vessel by detecting the form of the sketched section: the more circled the section, the more orthogonal to the drawing plane.

We can also observe that the orientation of a vessel at a branching point can be approximately inferred from the shape of the junction (see Fig. 2, right, and Sec. 3.3).

Lastly, we add an extra rule, not always used by anatomists, to express the distribution of curvature along the vessel. Let us consider the case when the first extremity of a tube lie in a plane and the other one is orthogonal to the plane: then, one can imagine many solutions for the tube to be curved. For instance, the tube may be mostly straight and bend just before the second extremity, but it also may be uniformly curved from one extremity to the other one. Thus, we use an additional convention that consists in drawing some inner hatching strokes along the tube in regions of high curvature (Fig. 3 and Sec. 3.2). Note that the curvature of the vessel is set uniform if there is no hatching strokes.

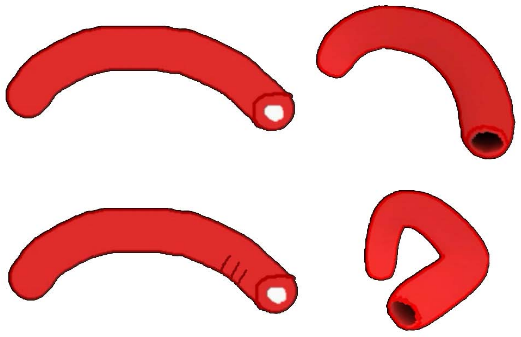


Figure 3: *Input sketches (left) and inferred 3D shapes (right) illustrating uniform curvature, versus the use of hatching for indicating the highly curved region.*

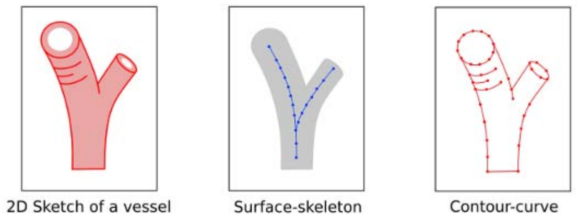


Figure 4: *Surface-skeleton and contour-curve extracted from a 2D sketch.*

3. Extracting depth clues from a 2D sketch

Having a look at any 2D sketch, one would naturally think of a 3D shape by observing contours and strokes-silhouette. We similarly extract depth clues from contours analysis of the 2D sketch.

To do so, we compute two curves from the sketch, namely the *surface-skeleton* and the *contour-curve*. We first extract a pixel-based skeleton by iterative erosion of the global 2D sketch, using [Hal89], and then convert it into a graph of segments, using [BPCB08]: this is the *surface-skeleton* (Fig. 4). We build the *contour-curve* the same way, but only taking into account the pixels of the contour-color (dark-red).

Computing the intersection of *surface-skeleton* and *contour-curve* enables us to extract many depth clues, as explained below.

3.1. Orientation of a vessel

As introduced in section 2, the orientation of a tube from a specific viewpoint can be deduced from the form of its cross-section. We identify the cross-sections of vessels by extract-

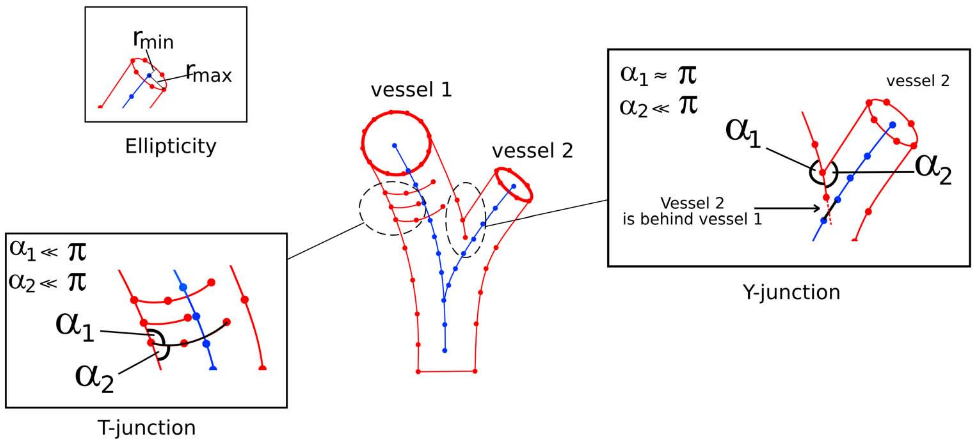


Figure 5: *In our system, the ellipticity of cross-sections, hatchings and Y-junction in branching regions are used as clues to infer orientation and relative depth of vessels.*

ing the cycles of the *contour-curve* that include an extremity of the *surface-skeleton* (see bold red cycles in Fig. 5).

For each of these cycles, we define the *ellipticity* $E(s)$ of the section s as:

$$E(s) = 1 - \frac{r_{min}(s)}{r_{max}(s)} \tag{1}$$

with $r_{min}(s)$ the minor axis of the section and $r_{max}(s)$ the major axis: if $E(s)$ tends to 0, the section approaches a circle and the vessel turns towards the viewpoint, else the vessel tends to be directed in the drawing plane. $E(s)$ is then used for computing the depth of the control point corresponding to this extemity (Sec. 4.2.3).

3.2. Curvature of a vessel

Since the orientation of a vessel does not give precise information about its global curvature, as detailed in section 2, we need the user to sketch inner hatching strokes in areas of high curvature. However, extracting hatching strokes from the the *contour-curve* is not trivial. As depicted on figure 5, a *T-junction* does not convey same information as a *Y-junction*: the first one indicates a strong curvature of the vessel, whereas the second one indicates a branching point with another vessel. Then, for each inner segment, we compute angles α_1 and α_2 at junction: if one angle tends to be a flat angle ($\alpha_i \geq \frac{3*\Pi}{4}$), then the junction is defined as a *Y-junction*; else this is a *T-junction* and the segment is defined as a hatching stroke.

3.3. Branching point

Branching points are detected in the *surface-skeleton* with segments having more than one neighbor. These points provide two different types of clues, as described below.

3.3.1. Relationship between vessels

At each branching point, the vessel with highest diameter is set to be *the parent* of the other one(s). For instance, vessel 1 on figure 5 is the parent of vessel 2. This relationship is then used for ordering the reconstruction of the vessels, as detailed in section 4.

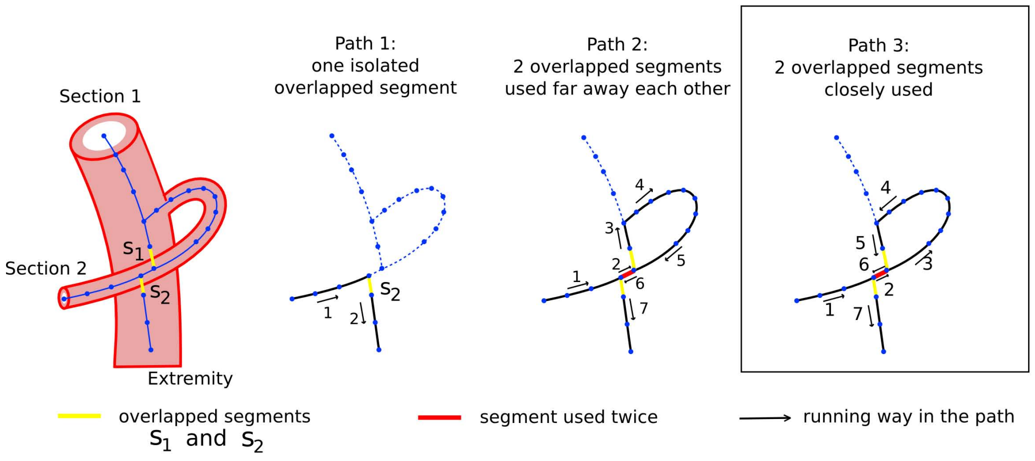
3.3.2. Relative positions between vessels

We use *Y-junction* as defined above to extract the relative position between two vessels at a branching point, which is then used to compute the tangent of chid vessel at this point (see 4.2.5). Note that each *Y-junction* of the *contour-curve* corresponds to a branching point of the *surface-skeleton*: this correspondance enables us to know the vessels involved in the junction. Then, computing the intersection of the *contour-curve* with the *surface-skeleton* help us to determinate which vessel is behind the other one. As depicted on figure 5, the vessel having a segment intersected by the *Y-junction* is behind the other vessel. Note that if there is no *Y-junction*, the two branching vessels are locally in the same plane.

3.4. Self-occluding situation

In case of interlinked vessels, their relative position is deduced from the analysis of the intersection of the *contour-curve* with the *surface-skeleton* (similarly to 3.3.2). In order to infer the appropriate depth along each vessel, we first need to assign to each vessel its own set of skeleton-segments, including occluded components. As depicted on figure 6, identifying to which vessel a segment belongs is not trivial; in some cases, a segment would even need to be duplicated since it belongs to two vessels (see the red segment in Fig. 6).

In the following, we call an *overlapped segment* a segment



Section 1
Section 2
Extremity
s_1
s_2
Path 1: one isolated overlapped segment
Path 2: 2 overlapped segments used far away each other
Path 3: 2 overlapped segments closely used
1
2
3
4
5
6
7
overlapped segments s_1 and s_2
segment used twice
running way in the path

Figure 6: *Example of possible paths between section 2 and the extremity. The best path is path 3.*

of the *surface-skeleton* that is intersected by the *contour-curve* (see the yellow segments in Fig. 6).

To solve the problem of *segments assignment*, we first compute all the paths between each pair of endpoints in the *surface-skeleton*. We then evaluate each path according to some criteria that define a *more plausible path*, and keep the first combination of best paths. Note that if this combination does not allow to go through all segments of the *skeleton-surface*, we keep the second combination, and so on, up to ensure to use all segments.

3.4.1. Defining a more plausible path

We consider as *more plausible paths* the paths with a pair of *overlapped segments* located close to each other -since they are more likely to correspond to the endpoints of an occluded part (path 3 on Fig. 6)-, against paths with a single *overlapped segment* (path 1) or with *overlapped segments* located far from each other (path 2). In addition, we include a segment twice along a path if and only if it is connected to a cycle (red segment on Fig. 6); in that case, the segment is duplicated with distinct depths to model distinct parts of vessels.

3.4.2. Extracting depth relationship

Once each vessel has been assigned its own skeleton, finding out the depth relationship between the vessels is then straightforward: the vessel with *overlapped segments* is behind the other one. The local thickness of the vessels gives us the relative depth constraint insuring that the vessels will not intersect. Note that, for each overlapping situation, we store the pair of involved segments; they will be used later for inferring the respective depth to each vessel (Sec. 4.2).

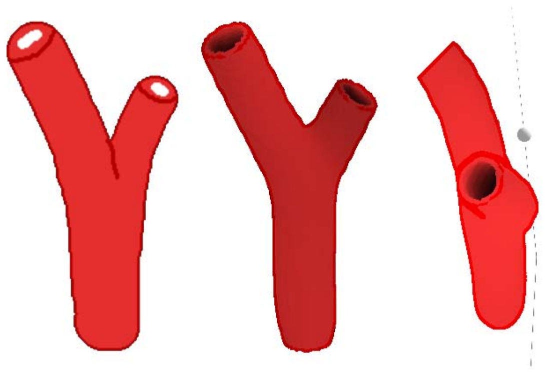


Figure 7: *Input sketch (left) and 3D model for two branching vessels: the child vessel springs back from the parent vessel due to Y-junction drawn in the sketch (see profile view on the right)*

4. Inferring the depth of vessels

4.1. Model for the skeleton-curves in 3D space

Splines are good candidates for modeling the skeleton-curves of vessels in 3D space, since we are looking for smooth curves that interpolate between points with depth or orientation constraints. One could think of using interpolation splines, such as Hermite or Cardinal splines, to ensure that the vessels exactly fulfill constraints at each control point. However, preventing a vessel from intersecting with another one would be difficult, since such spline curves do not remain in the convex hull of their control points. In addition, this spline model requires a regular sampling of control points, which is quite restrictive, else the tension of the spline will be uneven. In our case, the constraints from the sketch are not necessarily well distributed along the curve.

In contrast, approximation splines such as cubic Bsplines retain the curve in the convex hull of control points, which will ease preventing intersection between vessels. In addition, using *Non-Uniform Bsplines* allows to deal with non-equidistant control points while still generating a natural-looking shape. Moreover, the use of approximation is not a problem in our case since most of the constraints from the sketch are depth range constraints (with an infinity of valid positions) rather than exact position constraints (and the latter case, aligning 4 control points solves the problem).

In practice, we use *Non-Uniform Bsplines* to only infer a plausible depth to each point of the *surface-skeleton*, while keeping the other coordinates of the *surface-skeleton* equal to those read on the sketch.

4.2. Computing depth constraints

While setting up depth constraints onto each vessel is not trivial since constraints often depend on each other - especially in case of overlapping vessels -, using the relationship between vessels, as detailed in 3.3, is very helpful: while inferring depth of a parent vessel, we assign some depths values for the children on the fly.

Regarding one vessel, we now detail how control points are computed, in the following order.

4.2.1. First control point

The depth of the first control point of a vessel depends on the depth of its parent at the branching point. If the vessel has no parent, then its initial depth is equal to the depth of the drawing plane. This point is duplicated three times to be interpolated.

4.2.2. Control points with absolute depth constraints

For each control point arising from crossing vessels above or below this one, we compute an *absolute depth constraint* for this point p, depending on the depth of the corresponding point p' onto the other vessel involved V' (3.4). To ensure this vessel V will not intersect V', the depth $Z(p)$ of this point is set equal to:

$$Z(p) = Z(p') + pos * th * (r_{p'} + r_p) \qquad (2)$$

with $Z(p')$ the depth of p' and r_p (resp. $r_{p'}$) the radius of the section of p (resp. p'). pos expresses the position of V relative to V' ($pos = 1$ (resp. $pos = -1$) if V is above V' (resp. below)), and th is a threshold ($th > 1$) that allows to adjust the distance between vessels. In practice, we set $th = 1.2$ to ensure the vessels to be close but without blending (since human structures tend to be as close as possible for optimizing the space). To make the vessel interpolating this specific depth, we create three additional control points, around this one, with the same depth: doing so, we ensure to keep the C^2 continuity of the vessel (note that this would not be the case if we duplicated the point three times).

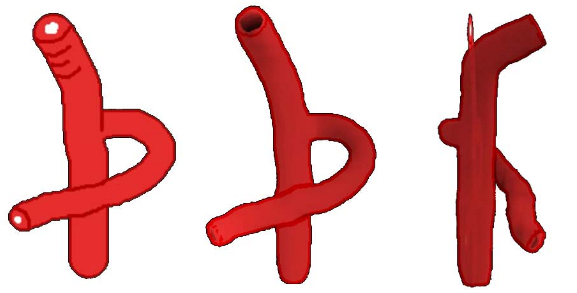


Figure 8: *Self-occluding vascular system: input sketch (left) and 3D model (right). Note the fact that the two vessels do not intersect but are very close.*

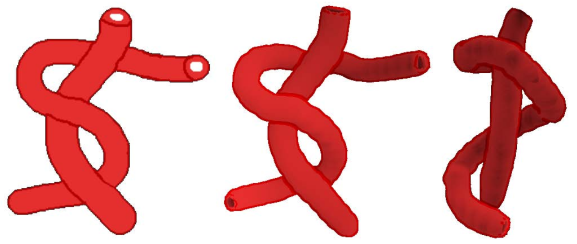


Figure 9: *A more complex situation of two interlinked vessels: note that these vessels do not intersect.*

4.2.3. Last control point

The depth of this point p depends on the orientation of the vessel: the more the vessel towards the viewpoint, the higher the variation of depth with previous control point p_{prev}. To compute its depth, we assume that the vessel remains in a bounding box defined by its length, and whose depth d_{max} is set equal to a third of its length (this experimental value relies on the fact that vessels tend to optimize the space and can not be curved so much). Then, the depth at this point is equal to:

$$Z(p) = max(d_{max}, Z(p_{prev}) + (1 - E(s)) * d_{max}) \qquad (3)$$

with $E(s)$ the ellipticity of the section as defined in section 3.1. This control point is duplicated three times to be interpolated.

4.2.4. Control points with relative depth constraints

We now deal with intermediate control points due to hatching (3.2). To enforce the curvature to be sharpened at a specific point, we set the depth of this point equal to the depth of the previous one. For instance, regarding a vessel with only hatching constraints and its section towards the viewpoint (see Fig 3), clamping the depth of hatching point equal to the depth of the first point produces the same effect as stretching out the tangent.

4.2.5. Tangent at the first point

The direction of the tangent of a vessel at a branching point is deduced from the *Y-junction*, as detailed in 3.3.2. To infer the right direction, we insert a control point with a depth smaller or higher than the first point, according to the required direction. In the case where there is no *Y-junction*, the two vessels are branched in the same plane: we assign to this control point the same depth as the branching point.

4.2.6. Tangent at the last point

Similarly to the tangent at first point, this tangent may have several directions. In case of elliptic section, we insert a control point with a depth equal to the depth of the last point. On the contrary, to bend the vessel towards the viewpoint, we insert a control point close to the last point with much smaller depth, to make the last segment of the 3D skeleton orthogonal to the drawing plane.

Finally, once all 3D control points are set onto the 2D skeleton, we compute the depth of each segment of the skeleton, and then use it to produce an implicit surface. We use convolution so that a skeleton curve can be used with no bulges at joint between segments [TZF04]. Thanks to this model, a smooth branching of the tubular shapes is seamlessly generated where skeletons joint.

This implicit surface is then displayed through expressive rendering, as detailed in next section.

5. Visualization and results

5.1. Expressive rendering

While some previous work focused on making the visualization of vascular structures easy to understand [OP05, JQD*08], none of them, to our knowledge, intended to imitate the aspect of chalk sketches done on a blackboard. In this light, we apply on final models an expressive rendering, computed on GPU. We first store the grid of implicit field values generated by 3D skeletons as a 3D texture on the GPU. Within this representation, the surface of the shape is represented by a given isovalue. This grid is then rendered using a slice-based volume rendering technique [EHK*04]. In addition to the rendered color buffer, a depth map is also generated by this process.

While light-red is used for displaying the surface, we compute silhouettes according high variations of depth between neighbor pixels, and render them with dark-red, as imposed by the conventions of anatomy. Then, we render cross-sections using cut-planes at vessels extremities. However, instead of displaying the inside surface in white as in the initial sketch, we use transparency to imitate empty tubes. Finally, we compute shading using normals to the surface (estimated by the gradient of field values), to ease the understanding of the depth and volumes in the scene.

5.2. Performances

Some performances have been measured on a 2.13Ghz dual core computer with a GeForce 9800 GTX, for the aortic arch model (Fig. 10). While computing 3D skeletons takes less than 400ms, the evaluation of the field function requires more time (6s), which is too much for setting up a fully interactive system. However, while these steps are done on CPU, we have experimented the use of the OpenMP API for the field function evaluation: times have been divided by 2. Then, since this latter step can be parallelized on GPU (the field at each voxel of the grid does not depend on the other voxels), we can expect to get a real-time system doing so. Regarding the rendering of the models, the display is fully interactive (34fps).

5.3. Results

Our system enables to reconstruct the geometry of some simple vessels configurations. Figures 7 to 10, as well as the attached video, depict our results regarding basic examples we mentioned in section 3. This is a first step towards sketching vascular systems with higher complexity. Even though no formal user study took place yet, each step has been evaluated by the professional of anatomical teaching we work with; and the results, displayed in a way that imitates the aspect of chalk drawing, looked impressive to him.

6. Discussion and future work

An interactive sketch-based modeling system would be of a great help for teaching anatomy, since it would enable to display different views of the organs while keeping a step by step, simplified presentation of their geometry through 2D sketching. We have studied the feasibility of this approach for one of the most intricate anatomical structures: a vascular system. Our method computes 3D models of these self-occluding, branching structures from a single sketch. Convolution surfaces and non-uniform Bsplines are used to model the 3D geometry and to infer a plausible depth variation while taking constraints into account.

Even though this system could be generalized to the sketching of any tubes network, we used some a priory knowledge from the sketching conventions used in anatomy to build the semantic of the sketch. Since our system uses these conventions as constraints on the 3D geometry, a professor is able to interactively draw 3D vessels in a 2D view, exactly as he would do on a blackboard with chalks.

However, dealing with complex vascular systems with several close, independant structures is still a challenge; inferring the right depth for each vessel is not trivial since the problem is under-constrained: on figure 9, one can imagine many other plausible solutions for the curvature of each vessel. Thus, for elaborating a relevant anatomical lecture and planning a practical test, we have to extend this system to

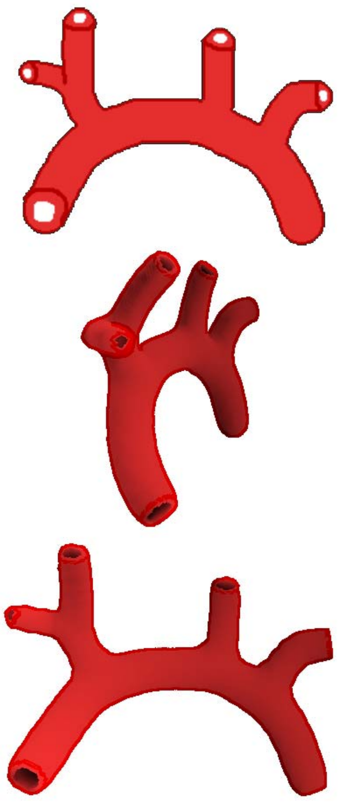


Figure 10: *Sketch-based modeling of the human aortic arch.*

underlying non-vascular structures that would ease inferring the right depth for vessels. As one can see on figure 1, vessels connected to the thyroid are constrained by the surface of this organ. Exploring sketching onto support surfaces would thus be a natural extension of our work. To do so, we plane to combine our system with more general systems based on implicit surfaces modeling, such as [BPCB08], to be able to design more complex shapes including both vascular and non-vascular structures.

Finally, this project has been carried for medical students: while cut-outs and transparency would be helpful to explain anatomical relations, this would not be easy for students to reproduce them on a sheet of paper. However, we plan to add some tools of this type to be able to interact with the final 3D models. As an example, it would also be useful to get handles at vessel extremities for adjusting their depth or moving them to show the hidden structures.

Acknowledgements

We would like to thank Victor Ostromoukhov for the inspiring early discussions at the beginning of this project. Many thanks to Cyril Crassin and Maxime Tournier for their practical help on GPU rendering.

References

[APS08] ANASTACIO F., PRUSINKIEWICZ P., SOUZA M. C.: Sketch-based parameterization of l-systems using illustration-inspired construction lines. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM)* (Annecy, France, june 2008). 2

[BPCB08] BERNHARDT A., PIHUIT A., CANI M.-P., BARTHE L.: Matisse: Painting 2d regions for modeling free-form shapes. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM)* (Annecy, France, june 2008), pp. 57–64. 2, 3, 8

[EHK*04] ENGEL K., HADWIGER M., KNISS J. M., LEFOHN A. E., SALAMA C. R., WEISKOPF D.: Real-time volume graphics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes* (New York, NY, USA, 2004), ACM, p. 29. 7

[Hal89] HALL R. W.: Fast parallel thinning algorithms: parallel speed and connectivity preservation. *Commun. ACM 32*, 1 (1989), 124–131. 3

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 409–416. 2

[IOI06] IJIRI T., OWADA S., IGARASHI T.: The sketch l-system: Global control of tree modeling using free-form strokes. In *Smart Graphics* (2006), pp. 138–146. 2

[IYKI08] IJIRI T., YOKOO M., KAWABATA S., IGARASHI T.: Surface-based growth simulation for opening flowers. In *GI '08: Proceedings of graphics interface 2008* (Toronto, Ont., Canada, Canada, 2008), Canadian Information Processing Society, pp. 227–234. 2

[JQD*08] JOSHI A., QIAN X., DIONE D., BULSARA K., BREUER C., SINUSAS A., PAPADEMETRIS X.: Effective visualization of complex vascular structures using a non-parametric vessel detection method. *IEEE Transactions on Visualization and Computer Graphics 14* (2008), 1603–1610. 7

[KH06] KARPENKO O. A., HUGHES J. F.: Smoothsketch: 3d free-form shapes from complex sketches. In *SIGGRAPH '06* (New York, NY, USA, 2006), ACM, pp. 589–598. 2

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. 26*, 3 (2007), 41. 2

[OP05] OELTZE S., PREIM B.: Visualization of vasculature with convolution surfaces:method, validation and evaluation. *MedImg 24*, 4 (April 2005), 540–548. 7

[SWSJ05] SCHMIDT R., WYVILL B., SOUSA M., JORGE J.: Shapeshop: Sketch-based solid modeling with blobtrees. In *SBIM* (2005). 2

[TWB*07] TURQUIN E., WITHER J., BOISSIEUX L., CANI M.-P., HUGHES J. F.: A sketch-based interface for clothing virtual characters. *IEEE Comput. Graph. Appl. 27*, 1 (Jan. 2007), 72–81. 2

[TZF04] TAI C., ZHANG H., FONG J.: Prototype modeling from sketched silhouettes based on convolution surfaces. *Computer Graphics Forum 23* (2004), 71–83. 2, 7

[WBCG09] WITHER J., BOUDON F., CANI M.-P., GODIN C.: Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *Comput. Graph. Forum 28*, 2 (2009), 541–550. Special issue: Eurographics 2009. 2