

Distance Visualizations for Vascular Structures in Desktop and VR: Overview and Implementation

J. Hombeck¹, M. Meuschke², S. Lieb¹, N. Lichtenberg³, R. Datta⁴, M. Krone³, C. Hansen², B. Preim² and K. Lawonn¹

¹ Friedrich-Schiller-University Jena, ² Otto-von-Guericke-University Magdeburg,
³ Eberhard-Karls-University Tübingen, ⁴ University Hospital Cologne

Abstract

The role of expressive surface visualizations in rendering vascular structures has seen an increased impact over the last years. Surface visualizations provide an overview of complex anatomical structures and support treatment planning as well as medical education. To support decision-making, physicians need visualizations that depict anatomical structures and their spatial relations to each other, i.e., well perceivable visual encodings of egocentric and endocentric distances.

We give an overview of common techniques for encoding distance information of 3D vessel surfaces. We also provide an implementation of all the visualizations presented as a starting point for other researchers. Therefore, we provide a Unity environment for each visualization, as well as implementation instructions. Thirteen different visualizations are included in this work, which can be divided into fundamental, surface-based, auxiliary and illustrative visualizations.

CCS Concepts

• **Human-centered computing** → **Visualization; Visualization toolkits**; • **Computer systems organization** → **Real-time systems**;

1. Introduction

Expressive visualizations of vascular structures are essential for treatment planning and training in interventional radiology and surgery. An expressive visualization of a vascular tree conveys its branching pattern well and makes anatomical variants obvious. Vascular structures exhibit special properties in comparison with other anatomical structures, such as organs, that motivate special visualization techniques beyond the common surface and volume rendering techniques. As an example, the shrinking effect of surface smoothing may lead to disconnected vascular trees. Specialized vessel visualization techniques may incorporate model assumptions such as the connectedness of a vascular tree, circular cross sections, and a continuous change of the local vessel diameter. Implicit surface visualizations are superior to explicit surfaces in terms of smoothness [PO08].

Research on vessel visualization initially was focused on technical aspects. Visualization techniques were developed that aimed at an accurate representation of vascular surfaces, at high smoothness (measured as curvature) and at a high performance. Although these aspects are still important, the question arises how different vessel visualization techniques or different parameterizations of the same technique affect the *perception* of vascular structures. The whole motivation for 3D visualizations is to provide a good depth and shape perception. Starting with the seminal work of Ropinski [RSH06] there was a series of publications on psycho-

physical studies to investigate these perceptual aspects of vessel visualizations. While Ropinski's major idea was to employ color for depth encoding, other visualization techniques, such as textures and hatching, were employed as well.

The importance of perception-based vessel visualization further increased in recent years. Vessel visualizations were developed that convey not only the vascular surface, but also associated scalar fields, such as wall thickness or wall shear stress derived from a flow visualization. Finally, the growing use of VR in surgical and interventional training has the consequence that also advanced vessel visualizations are explored in immersive settings. This further increases the demands to carry out psycho-physical experiments to understand the perceptual and cognitive consequences of using certain vessel visualization techniques.

The conduction of psycho-physical experiments involves various challenges. The careful selection of a set of appropriate stimuli is among these challenges. Due to the importance of this process, research was carried out to support this process with automatically generated suggestions. The EVALVIZ [MSL⁺19] framework is the most notable example of this research.

As a contribution to this work, we provide an overview of the most commonly used vessel visualizations for egocentric and exocentric distance estimation and include an implementation setup for

each visualization (see here [†]). This should promote new research within this area, as these visualizations are now more easily accessible and bundled in one place. In addition, we provide an overview of the visual cues used in each of these visualizations. This way, a more informed selection of visualizations can be made. In general, this work can be seen as a starting point to promote further and more in-depth research in the field of vessel visualization.

2. Visual Perception

The human visual perception is based on light-sensitive cells (rods and cones) in the retina [Gib50]. Due to a high density of these cells in the fovea, we can perceive a high level of detail in a small region of our visual field. Visual perception occurs in two stages: the *pre-attentive* stage, where we are able to detect objects that are strongly different from others and the *attentive* stage, where we search for certain objects and their relations based on our current goals. This conscious process requires considerable more time. As an example, a vessel segment that is emphasized with a unique color would be perceived pre-attentively, whereas the search for a trifurcation in a vascular tree requires attentive processing. Visual perception has many aspects, such as color, contrast and motion perception. For our purpose of studying perceptual effects of vessel visualizations, shape and depth perception are particularly important and will be discussed in the following.

Although the light sensitive cells are arranged in a planar configuration, we perceive the world as three-dimensional. Various depth cues, including shadows, shading, perspective projection, and partial occlusion [PBC*16], enable us to infer the spatial configuration, the depth relation and the object shapes with its bumps and bulges. In addition to these *monoscopic* depth cues there is also *stereoscopic vision* that provides further depth cues based on the combination of the two slightly different images perceived by both eyes. Stereoscopic vision only works in a certain depth range. However, when viewing visualizations on a desktop we are located in this range. Thus, stereoscopy is essential for medical visualization.

Virtual reality, where users typically wear VR glasses increases *immersion*, i.e., users are completely surrounded by a virtual environment (VE) instead of just having a monitor as a window to a VE. This is beneficial for surgical training, where users carry out manipulations of the patient anatomy. Currently widespread VR glasses are still not optimal with respect to human perception. Their field of view (FOV) is smaller than the FOV that our visual system enables. Also the spatial resolution of the VR glasses is not as high as our eyes would be able to perceive. Despite these limitations, the immersion and the motivating effect of VR already enables strong learning experiences [CSH*19].

3. Previous and Related Work

In this paper, we focus on 3D surface visualizations, especially of vascular structures. Therefore, we do not discuss map-based visualizations (see the recent survey of Eulzer et al. [EMML22]) and

volume renderings of vascular structures (see the survey from Kubisch et al. [KGNP12]). This focus is motivated by the wide use of surface visualizations in VR. Furthermore, we focus on techniques developed for the perception of distances.

3.1. Surface Visualization of Vascular Structures

Surface visualizations are based on an explicit segmentation of vascular structures, often followed by the determination of the vessel centerline, the local vessel diameter, and the branching graph [SPSP02]. This information may be employed to fit graphics primitives, such as cylinders or truncated cones, to the vessel centerline and ensure a continuous transition at strongly curved regions and at branchings. Pioneering work in this direction was carried out by Ehricke et al. [EDKS94] and Puig et al. [PTN97]. As an alternative to explicit visualizations, implicit vessel visualizations were created. Care is necessary to avoid or reduce the typical “unwanted” effect of implicit surfaces, such as bulging and strong blending [OP04, SOB*07]. More recent work aims at watertight meshes [KGPS13] and at optimizing the resulting geometry with a curvature-based polygon construction [WWL*10].

While a lot of effort was spent on optimizing the geometry, the performance, and accuracy of vessel surface visualizations, the actual rendering was largely neglected. Only recently, Ostendorf et al. [OMB*21] and Hombeck et al. [HMZ*22] studied the effect of different shading styles.

3.2. Distance Perception

The correct estimation of distances plays an important role in treatment planning, e.g., to find a well suited access without violating risk structures. Here, two types of distances are distinguished: egocentric and exocentric distances. Egocentric distances describe distances between the viewer and their environment [RVH13, EJM19], which is also known as depth perception. In contrast, exocentric distances relate to distances between two objects other than the viewer. In the following, we summarize visual cues to support distance perception.

Perception of egocentric distances. Based on the visual information captured by the eyes, humans can judge distances to objects [ABK*15]. The visual system must interpret various monocular and binocular cues to enable spatial and 3D depth perception, e.g., to judge the distance, depth, and shape of objects. Reichelt et al. [RHFL10] distinguished two types of depth information of the visual system: oculomotor and visual depth information. Oculomotor depth information relies on the anatomical capabilities of the eye, such as changing the angle of rotation of the eyes, the shape of the lens, and the tension of the eye muscles to perceive depth information. In this work, however, we focus on visual depth information in the form of different visualization techniques applied to 3D surfaces.

There are *monoscopic* and *stereoscopic* depth cues. In the case of monoscopic cues, an open eye is sufficient to perceive the scene. They can be divided into *static* and *motion-based* cues. The most important static depth cues include shadows, shading, perspective

[†] Unity Project

projection, and partial occlusion [PBC*16]. Motion parallax is another essential depth cue: When we interact with a 3D model or observe how it is continuously rotated in an animation, we may infer depth relations. As an example, a static maximum intensity projection (MIP) does not provide many depth cues. However, in an animation, the vascular anatomy—highlighted by MIP—can be assessed well.

Stereoscopic cues are a natural way to perceive egocentric distances with both eyes. However, in case printouts are needed or dynamic visualizations would require a high degree of interaction (e.g., during an operation), static images are irreplaceable. In these cases, additional depth information is essential. Further subcategories of depth cues include motion-, surface- and illumination-based cues. Common techniques for these are color scales [Ste87, RSH06, KHR20], glyphs [LHL17] or illustrative line drawings [LP16, LVP118].

Perception of exocentric distances. While there are several works that address the perceptual-based encoding of depth information, there is little work that focuses on the visual representation of exocentric distances. There are works that research how to encode exocentric distances between a virtual needle and surroundings risk structures, such as vessels. Different visualization techniques, such as color-encoding [HMZ*22] or illustrative renderings [LLPH15], are used to explicitly encode the spatial distance between the needle and a second object. Similar techniques can also be used to encode exocentric distances between two 3D surface models extracted from clinical image data, such as a vessel branch and a tumor. Wartenberg and Wiborg [WW03] evaluated the perception of exocentric distances in a desktop and an immersive cave environment. Here, advantages of the cave setup were revealed. However, distances were rather overestimated. In contrast to explicitly encoding exocentric distances, shape cues can be employed to improve the shape perception of objects that in turn support the perception of exocentric distances. Commonly used shading techniques to improve shape perception are Phong and Toon shading [PBC*16].

4. Medical Background

Perception-based visualization plays a major role in the context of medical data. In clinical routine, 3D medical visualizations fulfill several tasks. They provide an overview of complex anatomical structures or fractures. Furthermore, they support treatment planning, e.g., the decision whether a tumor can be removed, the extent of the operation and the possible access paths. As an example, surgical removal of a liver tumor must be carefully planned in order to safeguard good surgical outcome [HZH*09, HZP*14]. For this purpose, it is essential to understand the morphology of the organ and internal vasculature to determine the resection volume while preserving as much of the vasculature as possible. Therefore, spatial relationships and the distances between structures should be made apparent.

Data Acquisition. To support decision-making when dealing with complex anatomy and pathology, high-quality 3D models of organs and vessels can be reconstructed from medical image data. In this work, we used oncological liver surgery as an example to provide

the implemented depth encodings. We used two liver data sets provided by our clinical partners. Each data set comprises a 3D surface mesh of the liver, the vessel tree, and a liver tumor. The 3D surface models of these three structures comprise approx. 16,000, 30,000, and 5,000 vertices, respectively. All data sets were segmented from a contrast-enhanced computed tomography scan by employing a threshold-based segmentation. From the binary segmentation masks, 3D models of the corresponding structures are reconstructed using Marching Cubes.

5. Implementation Setup

For many publications, it is difficult to replicate the presented work due to the lack of crucial implementation details. In this work, we provide a unified framework for selecting visualizations. This framework is intended to lower the entrance hurdle for performing experimental visualizations. This can increase their overall usability even for smaller projects. We consulted with our clinical partners and found that many visualization research projects, especially in VR, use 3D engines like Unity to prototype certain scenarios. With this and the cross-platform accessibility in mind, we chose Unity as our development platform. Unity is a game engine that provides essential computer graphics commands across platforms. This way, even users without in-depth computer graphics knowledge can work with the visualizations presented. Unity not only supports Mac OS, Linux, and Windows, but also provides a virtual and augmented reality interface that supports almost all devices available.

As we intend to expand this framework in the future and add newly developed visualizations, we have decided to focus on the long-term support version Unity 2021.3 LTS. Once the correct Unity version is installed, our framework needs to be downloaded (see here ‡). Each visualization has its own Unity scene that can be loaded separately. This scene contains only the code needed for the particular visualization. Simple visualization requires only a material and a shader file in Unity. The shader is assigned to the material, which again is assigned to the desired 3D mesh. If the visualization has additional properties, such as color, brightness, size, etc., these properties can be changed within the material. These parameters can also be changed at runtime, making it easier to customize the visualization. More complex visualizations may also include a C# script to execute additional functionality and/or pass uniform variables to the shader. Because some visualizations require information provided by scripts, they may appear incorrect in the editor but operate properly in the Unity game window. Each file for the visualization is open source and can be customized.

6. Shading Fundamentals

In this section we briefly explain basic shading algorithms. Shading comprises diffuse, specular, and ambient reflection. There are highly realistic global and local illumination models as well as non-photorealistic shading techniques. When creating a non-photorealistic rendering, the specular part describes the reflection

‡ Unity Project

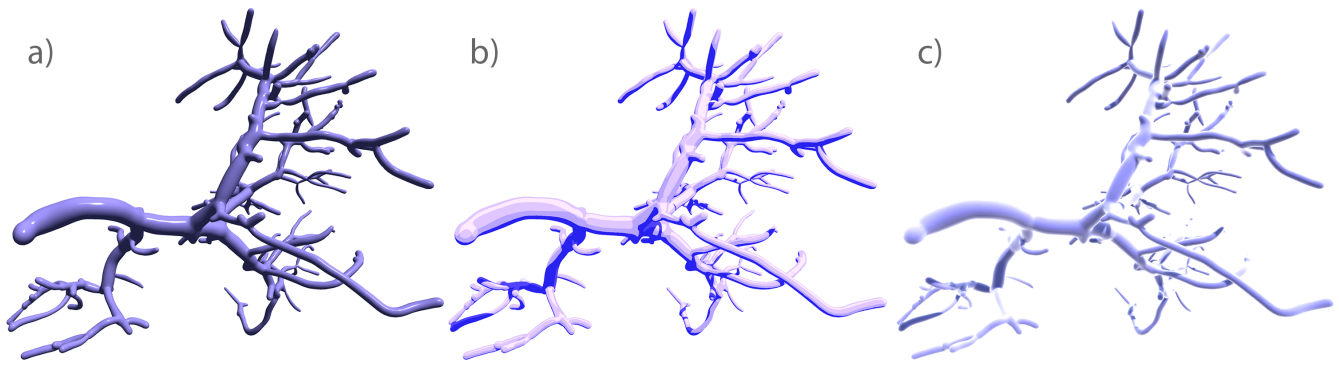


Figure 1: The fundamental shading techniques (a) Phong, (b) Toon, and (c) Fresnel effect applied on a vascular structure.

of light from smooth surfaces at certain angles, while the diffuse part is used to describe rough surfaces that reflect light in all directions. To create photorealistic renderings, many visualizations use the *Fresnel term* to calculate realistic-looking reflectance.

6.1. Phong

Phong shading is one of the most common surface visualization techniques [PBC*16] implemented as a base condition in numerous perception-based studies. The coloring of the surface is based on a combination of an opaque base color and shading effects resulting from a light source located somewhere in the scene, see Figure 1a.

Implementation. The *Phong* visualization requires three input parameters: The mesh color, the shininess, and the specular color. Each parameter is passed to the shader as a shader property. The calculation of the ambient, specular and diffuse components is performed within the fragment shader, taking into account the specified properties.

6.2. Toon

Toon shading (also called cel-shading) is a non-photorealistic surface rendering to mimic a cartoon representation [Dec96]. Although there are several ways to implement cell shading, we will focus on one in particular in this paper. Here, conventional smooth lighting values are calculated for each pixel and discretized into a small number of shades to create the characteristic "flat look". Thus, the shadows and highlights appear as blocks of color instead of being evenly blended in a gradient, see Figure 1b.

Implementation. The *Toon* visualization is based on a threshold calculation derived from the light calculation of the *Phong* model. It requires seven input parameters: the mesh color, the shininess, the specular color, the ambient color, the rim color, the rim amount, and the rim threshold. Each of the parameters is passed to the shader as a shader property and can be customized to make the *Toon* visualization unique. The important calculations of the ambient, specular, and diffuse components are performed in the fragment shader, similar to *Phong* shading.

6.3. Fresnel

The *Fresnel effect* describes the extent of reflection and refraction of light on a surface in relation to the viewing angle [Sch93]. The flatter the viewing angle on a surface, the more light is reflected, resulting in the surface appearing brighter when illuminated, see Figure 1c. A physically exact calculation of this effect is quite complicated, especially if one takes into account that the strength of the *Fresnel effect* also depends on the wavelengths of the light components due to chromatic dispersion. Instead, we use a simplified version of this effect.

Implementation. The *Fresnel* visualization requires three input parameters: The mesh color, the rim color (Fresnel color) and the Fresnel exponent. The Fresnel calculation is performed in a surface shader. By calculating the dot product between the surface normal and the viewing direction, the edge region can be determined. The edge region is then highlighted according to the Fresnel color and exponent.

7. Surface-based Visualizations

This section describes techniques that require the surface data to convey additional information. These techniques modify the visual representation of the surface either partially in regions of high interest or completely to convey information along the entire mesh. Both egocentric and exocentric distances can be encoded this way.

7.1. Heatmaps

Heatmaps are designed to convey information through color. They commonly indicate an area of varying interest. While *heatmaps* are typically used along with 2D data, an increasing number of studies are using *heatmaps* as a visual tool to focus the user's attention on specific parts of 3D meshes. In particular, *heatmaps* are increasingly used in treatment planning and training in interventional radiology and surgery. In 2D [KAB*20], *heatmaps* can be used to represent quantitative data such as gaze information and city population. In 3D, *heatmaps* can be used to visualize, for example, the distances between vascularity and tumors [HLL19,HMZ*22] or the degree to which the mitral valve is open [EEL*19, LER*20]. The

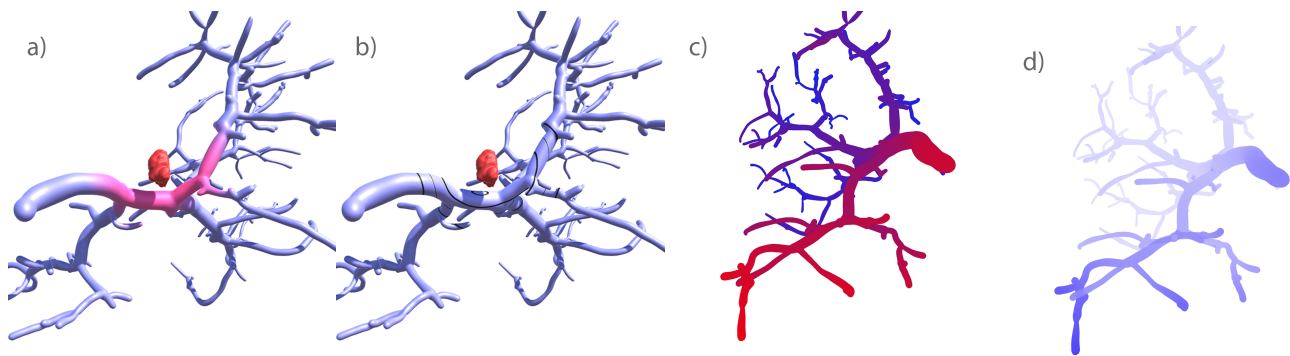


Figure 2: The surface-based visualizations (a) heatmap, (b) isolines, and (c) pseudo-chromadepth, and (d) fog applied on a vascular structure.

color of the *heatmap* then represents the quantitative data defined in the particular scenario, see Figure 2a. The color gradient must be defined by a colormap appropriate for the scenario. To ensure a smooth transition between the mesh and the *heatmap*, one side of the color gradient is usually defined by the color of the underlying mesh. If the mesh is colored according to the *heatmap* one has to use shading carefully and assure that different lighting conditions do not impair the readability of the colormap.

Implementation. For this implementation, the *heatmap* requires three parameters: the tumor position, the heatmap radius, and the color in the center of the *heatmap*. While the heatmap radius and color are passed to the shader as shader properties, the heatmap position is updated as uniform variable by a script. These parameters are then used to calculate the distance between each vertex and the tumor position. If the vertex is within the heatmap radius, the color of that vertex is adjusted based on the interpolated color between the surface color and the heatmap color. The shader also allows for multiple tumor positions to be entered if multiple tumors are present.

7.2. Isolines

Isolines, or contour lines, are lines that visualize data points from homogeneous regions within surface areas. *Isolines* help the user to understand the relationships between the surface and underlying acquired data, such as elevation or distance. Common applications for *isolines* are elevation maps for geographic data or atmospheric pressure maps. By dividing the surface into several homogeneous regions, the number of annotations can be reduced to a minimum and a more structured illustration can be achieved. This concept can also be applied to 3D mesh data. For example, *isolines* can convey the distance information between the vascular structure and a tumor location, similar to *heatmaps* [TPD06], see Figure 2b. Therefore, *isolines* are represented as equidistant lines on top of the vessel. In a 3D application, *isolines* may continue outside the visible region, e.g., on the backside of the vascular structure, and reappear in different regions. Matching these disappearing and reappearing *isolines* can be challenging. To facilitate this process, the thickness of every other isoline should be adjusted.

Implementation. The *isolines* require four input parameters: the

tumor position, the isoline radius, the number of isolines and the isoline thickness. While the isoline radius, number, and thickness are passed to the shader as a shader property, the tumor position is updated by a script. The isoline radius is divided into equidistant radii based on the number of isolines desired. The resulting radii are then thickened by the desired isoline thickness to create a tube at each radius. The thickness of every other isoline is calculated as thinner. Now the distance between each vertex and the tumor position is calculated. If the vertex is inside one of the tubes around the radii, the color of the vertex is set to black. The shader also allows you to enter multiple tumor positions.

7.3. Pseudo-Chromadepth

Steenblich [Ste87] introduced the *chromadepth* technique, where depths is encoded by the visible color spectrum. However, this results in depth encodings based on a variety of hues, which can be distracting from the shading information. Since the shading is necessary to perceive the 3D structure, Ropinski et al. [RSH06] introduced the *pseudo-chromadepth* (PCD) technique, which uses a reduced number of hues to encode depth information. Inspired by the scattering of light in the atmosphere, a red-to-blue color scale is employed. Surface areas closest to the observer are shown in red, while a blue area signifies the greatest distance, see Figure 2c. Intervening colors are linearly interpolated based on depth values in the range 0 to 1, resulting in a color spectrum consisting of red, magenta, violet, and blue hues.

Implementation. The *PCD* requires one input parameter: The bounding box of the mesh. Although the *PCD*'s color gradient is determined by the visualization itself, we provide the additional feature of specifying custom color maps by including the minimum and maximum distance color as a shader property. After calculating the minimum and maximum depth based on the bounding box of the mesh, the color of each fragment is determined by its depth within that volume. The color is interpolated between the maximum and minimum color values based on its depth value.

7.4. Fog

Gibson [Gib50] introduced a contrast-based depth encoding, called *air perspective*, depth perception depends on the scattering of light

in the atmosphere. The basic concept is that distant objects scatter incident light more before the light reaches the eye than nearby objects. Thus, distant objects are perceived with less contrast. Kersten-Oertel et al. [KOCC14] modelled the aerial perspective as a fog-like effect by reducing the contrast with increasing distance from the camera. The color spectrum consists of a single color, where the corresponding alpha value is changed from opaque to transparent based on the depth value. With this, nearer and farther parts appearing opaque and transparent, respectively, as shown in Figure 2d. Thus, this technique is similar to *PCD* shading in terms of encoding depth information based on egocentric distances. However, instead of mapping depth information to colors, it is encoded using transparency.

Implementation. The *fog visualization* requires one input parameter: The transparency falloff. For each fragment we calculate its depth value and determine the transparency based on this depth and the transparency falloff parameter. The higher the falloff value, the faster the object fades into the fog.

7.5. Stereoscopic Vision

Stereopsis is an important factor influencing depth perception. It describes the process of viewing the environment with both eyes. Instead of processing only one image, two slightly shifted images are processed by our brain. These binocular disparities are then used to create depth information. While in computer graphics we try to fake depth by using perspective renderings, the results are still displayed as a single image. Rendering stereoscopic images with 3D glasses or HMDs improves depth perception in almost all cases compared to their desktop counterpart [HMZ*22]. Thus, using stereoscopic renderings and devices can help to better perceive the scenario and further understand the depth information. Stereoscopic rendering can be combined with all presented visualizations.

8. Auxiliary Tools

Here we describe a category of perception supporting visualization techniques that has been identified by Preim et al. [PBC*16] as *Auxiliary Tools*. The following is mainly a recap of the publication by Lichtenberg and Lawonn [LL19], providing an overview of existing *Auxiliary Tools* for enhanced depth perception in the context of vascular visualization and define the term as: "*Auxiliary Tools in depth perception describe visual entities, (i.e., geometric objects) that augment a generated image of spatial data in order to encode depth information or to trigger and/or exaggerate depth cues.*"

Visual information channels on the surface of interest are therefore not directly affected, while additional depth information is packed into the final visualization. However, the additional geometry may overlap with the medical data, leading to a cluttered view and impaired perception of some parts of the visualization. While glyphs explicitly encode some given information in an arbitrary way they may become an *Auxiliary Tool* for depth perception. This is the case if they are designed and work as a trigger for depth cues and therefore encode depth information. The design of an *Auxiliary Tool* should always consider the pre-attentive and attentive phase of human perception.

8.1. Supporting Lines

The *Supporting Lines* (SL) by Lawonn et al. [LLPH15] utilize the shadow of an object as a natural depth cue to support depth perception. A vessel model is located above a plane with a shadow projection of the model, see Figure 3a. On the shadow plane, it is easy to estimate which point is further away from the viewer. The projection can be stylized to highlight certain parts of the model, such as tumor or vessel tissue. *SL* can be added by the user by selecting points on the vasculature. These points are then connected to their corresponding location on the shadow projection. Thus, a surface point and its shadow have an unambiguous connection which is useful in otherwise unclear cases. *SL* further invoke an overlapping depth cue with other parts of the vessel geometry. It enables the user to gain additional depth information for whole vessel branches with respect to a selected point.

Implementation. The shadow projection is achieved by a directional light source that is perpendicular to the ground plane. In this way, an orthographic shadow is projected onto the shadow plane. The supporting lines are generated by specifying a starting point on the vessel surface. From there, the supporting line geometry is created and scaled to the shadow plane by a script.

8.2. Supporting Anchors

A follow-up technique to the one above are the *Supporting Anchors* (SA) by Lawonn et al. [LLH17]. A drawback of the previous method is that the shadow plane is located at a fixed location outside the vascular model. The *SA*, however, are connected to an open cylinder which is view centered and oriented along the view direction, see Figure 3b. In this way, the selected points can be connected to their reference surface in any direction by projecting them to the closest point on the cylinder. This reduces visual clutter and allows to use the cylinder as an intuitive probe to select and highlight an area of interest within the vasculature. The connections then are augmented with an anchor-like extension along the cylinder within the same depth plane. This simplifies the comparison of reference points that are further away from each other.

The selection of points is furthermore done automatically, using an algorithm that picks end points of the vessel structure with respect to their spatial distribution inside the cylinder. Overlapping anchor geometry is therefore reduced.

Implementation. For the *Supporting Anchors*, a cylindrical object is generated and aligned with the camera position. This cylinder is rendered using a combination of fog and isoline visualization to create the desired transparency falloff with supporting lines. The *SA* are created in a similar manner to the supporting lines. After selecting a position on the vascular structure, the anchor geometry is created and scaled by a script.

8.3. Concentric Circle Glyphs

The *Concentric Circle Glyphs* (CCG) by Lichtenberg et al. [LHL17] are disc-shaped glyphs and also located at vessel end points and automatically selected by an algorithm to reduce overlap. Distance to the viewer is encoded by three concentric circles

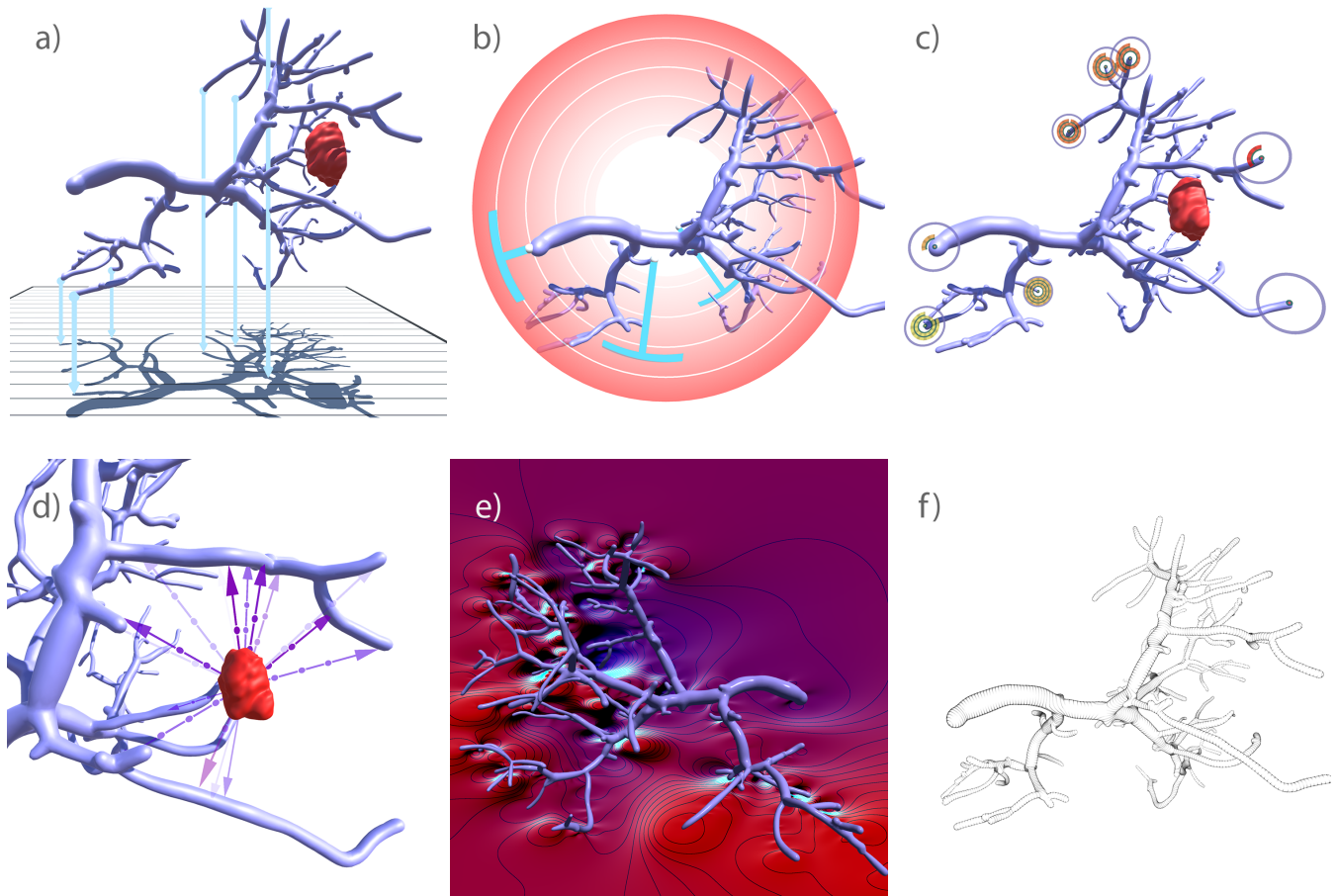


Figure 3: The auxiliary tools (a) supporting line, (b) supporting anchors, (c) concentric circle glyphs, (d) arrow glyphs, (e) void space surfaces, and (f) the illustrative visualization hatching applied on a vascular structure.

that get filled in sequence from inside out and in a clockwise manner, see Figure 3c. Three full circles are equal to the maximal distance and no circles at all are equal to the minimal distance. Additionally, the glyph size depends on the distance to the viewer. Glyphs further away are smaller in world space. The color of the glyphs can be used to indicate the proximity of the glyph relative to the tumor. In this way, the CCG can be interpreted during the pre-attentive phase and act as an *Auxiliary Tool*.

Implementation. For each *Concentric Circle Glyph*, the distance to the camera and the distance to the tumor are required and passed to the shader via a script. The fullness and color of each circle is calculated in the fragment shader based on these distance values. For each glyph, a view-aligned quad is created and rendered using the concentric circle glyph shader.

8.4. Arrow Glyphs

Arrow glyphs are a combination of color [Lev91] and shape [WPL96] that can be used to encode distances. Unlike surface visualizations, arrow glyphs do not utilize the surface as canvas, but in-

stead use the 3D space between the reference point and the surface. The visualization consists of multiple arrows pointing towards the surface, where the distribution of the arrows depends on the chosen endpoint, see Figure 3d. While an equidistant distribution is not naturally given, it can be computed by an algorithm of Lichtenberg et al. [LSHL18]. Given these sample points, any desired density of glyphs can be achieved. Hombeck et al. [HLL19] propose two different density levels for arrow glyphs: a denser distribution in critical areas, e.g., near vascularity, and a less dense distribution in non-critical areas. Adjusting the density can increase the information gain in critical areas, while maintaining a more structured view in non-critical areas. In addition to arrow shape, size, color, and transparency are used to encode information about the distance and shape of the mesh. The color of the arrow changes within hues of red, magenta, purple, and blue, similar to the gradient of the pseudo-chromadepth visualization. To provide information about the relative distances, each arrow contains a small dot every 2 cm. As the number of glyphs is increased, the size of the arrow is reduced to avoid cluttering the view. In addition to distance encoding, the arrow glyphs can also encode shape information through trans-

parency. The larger the angle between the arrow and the surface normal, the more transparently the glyph is rendered.

Implementation. *Arrow glyphs* require five input parameters: the tumor position, the maximum length, the switching distance, the thickness, and the glyph texture. While the maximum length, the switching distance, and the glyph texture are passed to the shader as shader properties, the tumor position and whether the mesh is considered a large or small glyph are updated by a script. While the mesh vertices can serve as anchor points for the glyphs, an additional sampling of points across the surface is required for an equidistant distribution and is already provided in the example. For each input vertex, the geometry shader is applied and generates a view-aligned quad between the input vertex and the tumor position. This square is filled with the selected glyph texture. The color and transparency are then calculated in the fragment shader. The shader also allows the input of multiple tumor positions.

8.5. Void Space Surfaces

The *Void Space Surfaces* by Kreiser et al. [KHR20] are a plain screen space method to convey depth information, where no actual additional geometry is used. Instead, the space in a rendered image which is not occupied by the projection of the displayed vessel structure is filled with a smooth surface which attaches to the contours of the vasculature, see Figure 3e. Shading, pseudo-chromadepth and iso-lines support the pre-attentive and attentive depth perception. The method does not require any selection of surface points to highlight and is conceptually free of any overlaps. On the downside, the depths of point pairs that are not in close proximity in screen space are not easy to compare.

Implementation. The normalized and linearized depth texture of the camera is required for the representation of *Void Space Surfaces*. After obtaining the depth texture, the contour lines of the vessel object are extracted and stored in a texture. Each point on the contour line is then passed to the shader and used as weights to fill the void space. At the end of this calculation, the illuminance, isolines, and colors are calculated and stored in another texture. The output texture is set as the background behind the vessel object.

9. Illustrative Visualization

Illustrative visualization is inspired by traditional illustration, e.g., in textbooks, and can be used to support the perception of depth and shape [LVPI18]. To convey shape or depth, *hatching methods* are commonly used in the area of illustrative visualization. Hatching is an illustration technique, where the object is covered with a dense set of lines. Originally, the density of lines were used to shade the object, i.e., regions which are brighter are illustrated with less lines than dark shaded regions, where a lot of lines are used. In an early approach, Ritter et al. [RHD*06] varied the hatching strokes to encode the distance to the viewer. Vessel segments that are close to the viewer are illustrated with thicker black lines and regions that are far away are illustrated with thinner lines. For illustrating purposes, Lawonn et al. [LLPH15, LLH17] employed hatching lines based on the *Confis* method [LMP13] and their extension [LKEP14] to convey the shape of the vessel structure, see Figure 3f. To encode the depth, they introduced glyph-based visualizations: *sup-*

porting lines and *supporting anchors*. One limitation of the first approach [LLPH15] was that the position of the glyph placement was fixed and needed to be set in advance. This limitation was then resolved by automatically placing the glyphs based on the view position [LLH17]. Later, Lichtenberg et al. [LHL17] extended this idea and placed the glyphs based on a cost function. Again, they employed hatching strokes to convey the shape of the vessel structures. Instead of pure illustration-based hatching strokes, Lichtenberg et al. [LSHL18] employed a real-time stripe pattern approach. This technique allows to vary the stroke size based on an arbitrary scalar field defined on the surface, e.g., curvature. It was also possible to alter the thickness of the strokes in real-time to encode the distance to the viewer or any arbitrary object. Hatching techniques allow an encoding on the surface, while the surface can still be colored to encode additional information [MVB*17, SLK*17]. Therefore, the use of illustrative visualization techniques has the potential for advanced use to encode several information on the surface. While we implement the supporting lines, supporting anchors, and hatching, more illustrative visualizations will follow as further work.

10. Discussion

We have presented multiple visualizations that can be used for egocentric and exocentric distance evaluation, and included an implementation for each visualization in Unity. While Unity is a widely used application for game development, it has seen increasing use in the visualization community due to its modular workflow. While Unity provides a lot of support for basic computer graphics commands, it still limits the user in some cases. When creating unique visualizations in other graphics frameworks, such as OpenGL, we enjoy more freedom, but at the cost of more programming effort.

10.1. Expandability and Limitations

By providing a unified and cross-platform application that contains multiple visualizations for egocentric and exocentric distance estimations, we want to increase accessibility to them and thus increase their potential use in different projects. Unity was chosen for this as it requires minimal setup effort. Other applications may not be cross-platform or may require significantly more time to get started. While the work is generalizable and theoretically executable in various languages, some effort must be made to convert Unity's built-in variables. Variables such as view direction, vertex position, or model, view, and projection matrices are provided automatically and do not need to be passed explicitly. Although these features are very convenient in most cases, they can be problematic as they cannot be manipulated prior to the shader phase. A major problem with our implementation was the limitation of point-based rendering. Unity does not support the ability to apply shaders directly to point values; a triangulated mesh is always required. We managed to avoid this problem by creating meshes from our point values and then interpreting them as points again in the geometry shader.

Most of the visualizations can be extended to different meshes and applications, since the associated shaders and scripts can be applied to different objects. However, some require a list of specific locations to create additional objects, such as the *Supporting Lines* or *Arrow Glyphs*. These candidates can be selected by hand

Visualization	Shading	Shadow	Color	Transparency	Surface	Void Space	Pre-attentive	Attentive	Egocentric	Exocentric	Realtime
Phong	✓	✓	×	×	×	×	✓	×	✓	×	✓
Toon	✓	✓	×	×	×	×	✓	×	✓	×	✓
Fresnel	✓	✓	×	×	×	×	✓	×	✓	×	✓
Supporting Lines	✓	✓	×	×	×	✓	×	✓	✓	✓	✓
Supporting Anchors	✓	×	×	⊙	×	✓	×	✓	✓	✓	✓
Concentric Circle Glyphs	✓	×	✓	×	×	⊙	×	✓	✓	✓	✓
Void Space Surfaces	✓	×	✓	×	×	✓	×	✓	✓	×	✓
Arrow Glyphs	×	×	✓	✓	×	✓	✓	✓	×	✓	✓
Heatmaps	×	×	✓	×	✓	×	✓	✓	×	✓	✓
Isolines	×	×	×	×	✓	×	✓	✓	×	✓	✓
Pseudo-Chromadepth	✓	×	✓	×	✓	×	✓	×	✓	×	✓
Fog	✓	×	×	✓	✓	×	✓	×	✓	×	✓
Hatching	✓	×	×	×	×	×	✓	×	✓	×	✓

Table 1: Overview of fundamental, surface based, auxiliary and illustrative visualizations. For each visualization we indicate the occupied visual cues, the attentive phase, the distance encoding and system information. Since perspective projection and partial occlusion are included in all visualizations, we have not explicitly mentioned them in the table.

or calculated in a pre-processing step. How this selection process should be handled is ultimately determined by the user. As a general guidance, we have provided a pre-calculated point selection for each visualization in our project. However, should the object be modified, these positions will need to be re-evaluated.

10.2. Visualization Selection

To support the selection of a visualization technique, we included Table 1. It provides the occupied information channels, the attentive phase, the distance category, and the system availability. We can see which information channels are still available and adjust the visualization or encode additional information. For example, the *heatmap* encodes the exocentric distance using color, while the *isolines* encode the exocentric distance using shape. Since these do not interfere with each other, both visualizations can be combined to increase the perception of exocentric distance.

In other cases, it is not possible to use the mesh surface to encode information, and we have to switch to glyph-based visualization. If this is also not possible due to spatial constraints, we can move to *illustrative visualizations*. Using this or a similar approach, we can simplify the selection of appropriate visualizations. Providing implementations for each visualization also lowers the threshold to actively use one of the presented techniques.

11. Conclusion and Future Work

We have presented an overview of thirteen different vascular visualizations for evaluating egocentric and exocentric distances. These visualizations can be categorized as fundamental, surface-based, auxiliary, and illustrative visualizations, each conveying distance in a unique way. Since it can be tedious to work with these visualizations, we provide an implementation and explanation for each. In addition to the implementation, we also provide an overview of the different visual cues used to perceive depth and distance. This can help to better understand which visualization can be used in which specific case, or to understand why one visualization may or may not work in combination with another. As we continue our work, we intend to incorporate upcoming visualizations as well as existing visualizations that prove useful in this domain to our project. In

this way, we can provide a framework that contains a wide variety of visualizations and support researchers in their work. In summary, this work provides the necessary overview and implementation to encourage further research in vascular visualization and promote psycho-physical studies within the visualization community.

References

- [ABK*15] ABHARI K., BAXTER J. S., KHAN A. R., PETERS T. M., RIBAUPIERRE S. D., EAGLESON R.: Visual enhancement of MR angiography images to facilitate planning of arteriovenous malformation interventions. *TAP 12*, 1 (2015), 1–15. 2
- [CSH*19] CHHEANG V., SAALFELD P., HUBER T., HUETTL F., KNEIST W., PREIM B., HANSEN C.: Collaborative Virtual Reality for Laparoscopic Liver Surgery Training. In *Proc. of IEEE Int. Conf. on Artificial Intelligence and Virtual Reality* (2019), pp. 1–8. 2
- [Dec96] DECAUDIN P.: Cartoon-looking rendering of 3d-scenes. *Syntim Project Inria 6* (1996). 4
- [EDKS94] EHRICKE H.-H., DONNER K., KOLLER W., STRASSER W.: Visualization of vasculature from volume data. *Computers & Graphics* 18, 3 (1994), 395–406. 2
- [EEL*19] EULZER P., ENGELHARDT S., LICHTENBERG N., DE SIMONE R., LAWONN K.: Temporal views of flattened mitral valve geometries. *IEEE Trans Vis Comput Graph* 26, 1 (2019), 971–980. 4
- [EJM19] EL JAMIY F., MARSH R.: Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality. *IET Image Process* 13, 5 (2019), 707–712. 2
- [EMML22] EULZER P., MEUSCHKE M., MISTELBAUER G., LAWONN K.: Vessel maps: A survey of map-like visualizations of the cardiovascular system. *Computer Graphics Forum* (2022). 2
- [Gib50] GIBSON J. J.: *The Perception Of The Visual World*. Boston: Houghton Mifflin, 1950. 2, 5
- [HLL19] HOMBECK J. N., LICHTENBERG N., LAWONN K.: Evaluation of spatial perception in virtual reality within a medical context. In *Bildverarbeitung für die Medizin*. 2019, pp. 283–288. 4, 7
- [HMZ*22] HOMBECK J., MEUSCHKE M., ZYLA L., HEUSER A.-J., TOADER J., POPP F., BRUNS C. J., HANSEN C., DATTA R. R., LAWONN K.: Evaluating Perceptual Tasks for Medicine: A Comparative User Study Between a Virtual Reality and a Desktop Application. In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2022), pp. 514–523. 2, 3, 4, 6
- [HZH*09] HANSEN C., ZIDOWITZ S., HINDENNACH M., SCHENK A., HAHN H., PEITGEN H.-O.: Interactive determination of robust safety

- margins for oncologic liver surgery. *J Comput Assist Radiol Surg* 4, 5 (2009), 469–474. 3
- [HZP*14] HANSEN C., ZIDOWITZ S., PREIM B., STAVROU G., OLD-HAFER K. J., HAHN H. K.: Impact of model-based risk analysis for liver surgery planning. *J Comput Assist Radiol Surg* 9, 3 (2014), 473–480. 3
- [KAB*20] KRAUS M., ANGERBAUER K., BUCHMÜLLER J., SCHWEITZER D., KEIM D. A., SEDLMAIR M., FUCHS J.: Assessing 2D and 3D Heatmaps for Comparative Analysis: An Empirical Study. In *Proc. of ACM SIGCHI Conference on Human Factors in Computing Systems* (2020), pp. 1–14. 4
- [KGNP12] KUBISCH C., GLASSER S., NEUGEBAUER M., PREIM B.: Vessel Visualization with Volume Rendering. In *Visualization in Medicine and Life Sciences II - Progress and New Challenges*, Linsen L., Hagen H., Hamann B., Hege H., (Eds.), Mathematics and visualization. Springer, 2012, pp. 109–134. 2
- [KGPS13] KRETSCHMER J., GODENSCHWAGER C., PREIM B., STAMMINGER M.: Interactive patient-specific vascular modeling with sweep surfaces. *IEEE Trans Vis Comput Graph* 19, 12 (2013), 2828–2837. 2
- [KHR20] KREISER J., HERMOSILLA P., ROPINSKI T.: Void space surfaces to convey depth in vessel visualizations. *IEEE Trans Vis Comput Graph* 27, 10 (2020), 3913–3925. 3, 8
- [KOCC14] KERSTEN-OERTEL M., CHEN S., COLLINS L.: An Evaluation of Depth Enhancing Perceptual Cues for Vascular Volume Visualization in Neurosurgery. *IEEE Trans Vis Comput Graph* 20 (2014), 391–403. 6
- [LER*20] LICHTENBERG N., EULZER P., ROMANO G., BRČIĆ A., KARCK M., LAWONN K., DE SIMONE R., ENGELHARDT S.: Mitral valve flattening and parameter mapping for patient-specific valve diagnosis. *J Comput Assist Radiol Surg* 15, 4 (2020), 617–627. 4
- [Lev91] LEVKOWITZ H.: Color icons-merging color and texture perception for integrated visualization of multiple parameters. In *IEEE Visualization* (1991), pp. 164–170. 7
- [LHL17] LICHTENBERG N., HANSEN C., LAWONN K.: Concentric Circle Glyphs for Enhanced Depth-Judgment in Vascular Models. In *Proc. of Eurographics Workshop on Visual Computing for Biology and Medicine* (2017), pp. 179–188. 3, 6, 8
- [LKEP14] LAWONN K., KRONE M., ERTL T., PREIM B.: Line integral convolution for real-time illustration of molecular surface shape and salient regions. *Computer Graphics Forum (Eurographics Conference on Visualization (EuroVis) 2014)* 33(3) (2014), 181–190. 8
- [LL19] LICHTENBERG N., LAWONN K.: Auxiliary Tools for Enhanced Depth Perception in Vascular Structures. In *Biomedical Visualisation*, vol. 1138. Springer International Publishing, 2019, pp. 103–113. 6
- [LLH17] LAWONN K., LUZ M., HANSEN C.: Improving Spatial Perception of Vascular Models Using Supporting Anchors and Illustrative Visualization. *Computers & Graphics* 63 (2017). 6, 8
- [LLPH15] LAWONN K., LUZ M., PREIM B., HANSEN C.: Illustrative visualization of vascular models for static 2d representations. In *Medical Image Computing and Computer Assisted Intervention* (2015), pp. 399–406. 3, 6, 8
- [LMP13] LAWONN K., MÖNCH T., PREIM B.: Streamlines for illustrative real-time rendering. *Computer Graphics Forum* 32, 3 (2013), 321–330. 8
- [LP16] LAWONN K., PREIM B.: *Feature Lines for Illustrating Medical Surface Models: Mathematical Background and Survey*. 2016, ch. Visualization in Medicine in Life Sciences III, pp. 93–132. 3
- [LSHL18] LICHTENBERG N., SMIT N., HANSEN C., LAWONN K.: Real-time field aligned stripe patterns. *Computers & Graphics* 74 (2018), 137–149. 7, 8
- [LVPI18] LAWONN K., VIOLA I., PREIM B., ISENBERG T.: A survey of surface-based illustrative rendering for visualization. *Computer Graphics Forum* 37, 6 (2018), 205–234. 3, 8
- [MSL*19] MEUSCHKE M., SMIT N. N., LICHTENBERG N., PREIM B., LAWONN K.: EvalViz - Surface visualization evaluation wizard for depth and shape perception tasks. *Computers & Graphics* 82 (2019), 250–263. 1
- [MVB*17] MEUSCHKE M., VOSS S., BEUING O., PREIM B., LAWONN K.: Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms. *IEEE Trans Vis Comput Graph* 23(1) (2017), 761–770. 8
- [OMB*21] OSTENDORF K., MASTRODICASA D., BÄUMLER K., CODARI M., TURNER V., WILLEMINK M. J., FLEISCHMANN D., PREIM B., MISTELBAUER G.: Shading Style Assessment for Vessel Wall and Lumen Visualization. In *Proc. of Eurographics Workshop on Visual Computing for Biology and Medicine* (2021), pp. 107–111. 2
- [OP04] OELTZE S., PREIM B.: Visualization of anatomic tree structures with convolution surfaces. In *EuroVis* (2004), pp. 311–320. 2
- [PBC*16] PREIM B., BAER A., CUNNINGHAM D., ISENBERG T., ROPINSKI T.: A survey of perceptually motivated 3D visualization of medical image data. *Computer Graphics Forum* 35, 3 (2016), 501–525. 2, 3, 4, 6
- [PO08] PREIM B., OELTZE S.: 3d visualization of vasculature: an overview. In *Visualization in Medicine and Life Sciences*. Springer, 2008, pp. 39–59. 1
- [PTN97] PUIG A., TOST D., NAVAZO I.: An interactive cerebral blood vessel exploration system. In *Proc. of IEEE Visualization* (1997), pp. 443–446. 2
- [RHD*06] RITTER F., HANSEN C., DICKEN V., KONRAD O., PREIM B., PEITGEN H.-O.: Real-Time Illustration of Vascular Structures. *IEEE Trans Vis Comput Graph* 12, 5 (2006), 877–884. 8
- [RHFL10] REICHEL T., HÄUSSLER R., FÜTTERER G., LEISTER N.: Depth cues in human visual perception and their realization in 3d displays. In *Proc. of SPIE* (2010), vol. 7690, pp. 92–103. 2
- [RSH06] ROPINSKI T., STEINICKE F., HINRICHS K. H.: Visually Supporting Depth Perception in Angiography Imaging. In *Proc. of Smart Graphics* (2006), pp. 93–104. 1, 3, 5
- [RVH13] RENNER R. S., VELICHKOVSKY B. M., HELMERT J. R.: The perception of egocentric distances in virtual environments—a review. *ACM Comput Surv* 46, 2 (2013), 1–40. 2
- [Sch93] SCHLICK C.: A customizable reflectance model for everyday rendering. In *Eurographics Works. on Rendering* (1993), pp. 73–83. 4
- [SLK*17] SMIT N., LAWONN K., KRAIMA A., DERUITER M., SOKOOTI H., BRUCKNER S., EISEMANN E., VILANOVA A.: PelVis: Atlas-based Surgical Planning for Oncological Pelvic Surgery. *IEEE Trans Vis Comput Graph* 23, 1 (2017), 741–750. 8
- [SOB*07] SCHUMANN C., OELTZE S., BADE R., PREIM B., PEITGEN H.-O.: Model-free surface visualization of vascular trees. In *EuroVis* (2007), pp. 283–290. 2
- [SPSP02] SELLE D., PREIM B., SCHENK A., PEITGEN H.-O.: Analysis of vasculature for liver surgical planning. *IEEE transactions on medical imaging* 21, 11 (2002), 1344–1357. 2
- [Ste87] STEENBLIK R. A.: The Chromostereoscopic Process: A Novel Single Image Stereoscopic Process. In *Proc. of SPIE* (1987), vol. 0761, pp. 27–34. 3, 5
- [TPD06] TAPPENBECK A., PREIM B., DICKEN V.: Distance-based transfer function design: Specification methods and applications. In *SimVis* (2006), pp. 259–274. 5
- [WPL96] WITTENBRINK C. M., PANG A. T., LODHA S. K.: Glyphs for visualizing uncertainty in vector fields. *IEEE Trans Vis Comput Graph* 2, 3 (1996), 266–279. 7
- [WW03] WARTENBERG C., WIBORG P.: Precision of exocentric distance judgments in desktop and cube presentation. *Presence* 12, 2 (2003), 196–206. 3
- [WWL*10] WU J., WEI M., LI Y., MA X., JIA F., HU Q.: Scale-adaptive surface modeling of vascular structures. *Biomedical Engineering Online* 9, 1 (2010), 1–16. 2