Visualizing Vascular Structures in Virtual Environments

Thomas Wischgoll[a]

[a]Wright State University, 3640 Col. Glenn. Hwy., Dayton, OH 45431, USA

ABSTRACT

In order to learn more about the cause of coronary heart diseases and develop diagnostic tools, the extraction and visualization of vascular structures from volumetric scans for further analysis is an important step. By determining a geometric representation of the vasculature, the geometry can be inspected and additional quantitative data calculated and incorporated into the visualization of the vasculature. To provide a more user-friendly visualization tool, virtual environment paradigms can be utilized. This paper describes techniques for interactive rendering of large-scale vascular structures within virtual environments. This can be applied to almost any virtual environment configuration, such as CAVE-type displays. Specifically, the tools presented in this paper were tested on a Barco I-Space and a large 62x108 inch passive projection screen with a Kinect sensor for user tracking.

Keywords: Vascular structures, virtual environments, passive 3D stereo, Kinect, billboarding, GLSL, programmable graphics hardware

1. INTRODUCTION

Coronary heart diseases are the number one cause of death in the United States. A lot of research has been done to find out more about these diseases. Yet, there is still a lot to discover since current treatment options for patients with significant coronary heart diseases include stents, which address the symptoms but not its origins. Hence further studies are needed to find out more about these diseases. Similarly, improved detection methods can help identify the diseases at an early stage. For example, local blockages are relatively easy to identify in a standard angiogram. However, if the disease has not progressed very far yet or spreads over larger areas within the vasculature, this is much more difficult to detect.

One advantage of reconstructing the full geometry of a vascular tree over, for instance, direct volume visualization methods is that it allows for measurements of objects in the depicted scene. Distances can be measured accurately when the geometry is known. Volume visualization methods may not have the necessary precision due to their use of a discrete grid which may have an even coarser resolution than the actual model in order to maintain interactive frame rates. Similarly, additional quantitative data can be extracted more easily, such as vascular volume or cross-sectional areas of the vessels and incorporated into the visualization.

Depending on the size of the vascular structure, i.e. the number of vessels included in the model, it can be difficult to navigate the structure in a desktop environment with just a small screen and a mouse. Virtual environments can be of great help since they provide more screen real estate for the vascular structure to be displayed. Typically, virtual environment setups include some form of user tracking so that the view perspective is automatically adapted to the user's position. As a result, view changes can be performed by the user simply moving around resulting in very fine-grained control over the view settings. This also makes for very intuitive view manipulations as the virtual environment reacts similar to the real world which naturally is very familiar to the user. Overall, the utilization of virtual environments configurations for visualization of vascular structures provides a very user-friendly and intuitive way of investigating and analyzing vascular structures.

Since interactive rendering of the vascular structure is a strict requirement when using virtual environments, special care has to be taken to render even larger vascular models, such as the one depicted in figure 1 consisting of 220 million vessel segments. By using the programmable graphics hardware, rendering performance can be increased significantly as described in this paper.

Further author information: (Send correspondence to Thomas Wischgoll)
Thomas Wischgoll: E-mail: thomas.wischgoll@wright.edu, Telephone: 1 937 775 5057

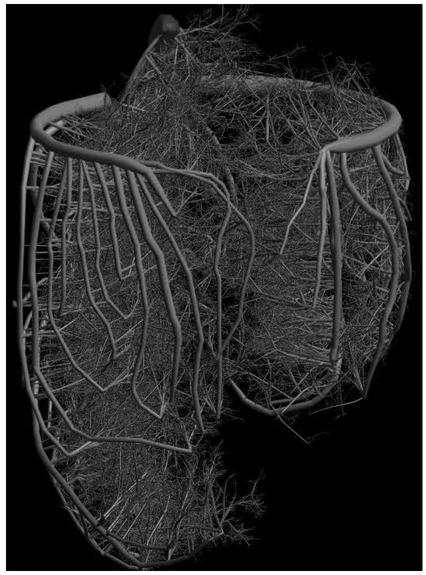

Figure 1. A very large vascular structure composed of 220 million individual vessel segments including the most proximal coronary vessels all the way down to the capillaries.

The structure of this paper is as follows. The next section lists work related to the visualization of vascular structures. This is followed by the presentation of the methodologies for visualizing vascular structures in fully immersive virtual environments. Finally, the results are presented and discussed.

2. RELATED WORKS

Several approaches exist for investigating tubular internal organs, such as vessels or colons. Many of these use the Marching Cubes algorithm as introduced by Lorensen and Cline[1] to generate geometry information based on iso-surfacing. Different methods have been proposed and implemented to avoid ambiguities in the original case table of the Marching Cubes algorithm[2].[3] Examples are given, for instance, in Hong et al.[4]

Different methods have been developed for generating 3D models from 2D CT or MRI data sets. Öltze et al. use convolution surfaces with a Gaussian filter in combination with preprocessing steps for external 3D tree visualizations from an external view point.[5]

Other techniques include the flattening of vascular structures, as shown by Kanitsar et al. with their enhancements of curved planar reformation (CPR), where an enclosing hierarchy of vessel hulls is used to approximate the projected vessel tree. Due to the flattened layout of the structure, the complete vessel wall is visible in a single view. This helps prevent occlusions by superimposed cells and allowing unobstructed views of stenoses and calcifications.[6] Puig et al. extract a symbolic model from the voxel model of cerebral blood vessels including detection of features such as bifurcations, aneurysms, and stenoses.[7] Wischgoll et al.[8] deploy a topological analysis of a vector field defined via the segmented vessel boundary to accurately identify the center lines of the vessels.

Using real data based on CT or MRI scans not only results in accurate geometric models. Derived models can also be used to perform a quantitative analysis of a specimen, for example computation of the volume or surface area of vessel segments and angles between different vessel segments to confirm.[9] Such data can also be used to confirm Murray's minimal work hypothesis.[10] Most of the examples in this paper are based on geometric models derived from CT scans of a pig's heart where detailed information about both the arterial[11] as well as the venous system[12] is available. In addition, a previously scanned specimen can be extended based on the statistical values derived in these studies according to Kaimovitz et al.[13] to obtain a model that includes even vessels on the capillary level. To create this highly detailed model, vessels are grown at the terminal vessels derived from the scan according to the statistical database. A simulated annealing process then prevents vessels from intersecting resulting in a large-scale model that comprises of vessels on multiple scales all the way down to the capillary level.

The center lines of the vessels in combination with the vessel radii computed at the center points allow for a geometric reconstruction of the vasculature. Various techniques for visualizing vascular structures can be found in the literature. Hahn et al.[14] employ geometrical primitives, such as truncated cones, to visualize vessels inside the human liver. Masutani et al.[15] used cylinders aligned to the vessel skeleton to visualize the vasculature. Different radii at branchings resulted in discontinuities when using this method. Felkel et al.[16] reconstructed liver vessels from center line and radius information to supply an augmented reality tool for surgery. Puig et al.[17] developed a system for exploring cerebral blood vessels using a symbolic model with a focus on geometric continuity and on realistic shading. Öltze et al.[18,19] use convolution surfaces to obtain a smoother representation of blood vessels extracted from CT or MR data. Ritter et al.[20] extended on illustrative rendering techniques to accentuate spatial depth by use of GPU-accelerated shadow-like depth indicators. The method was applied to vascular structures to better distinguish vessels in the front from vessels further in the back. Stoll et al.[21] introduced stylized primitives similar to billboarding[22] which utilize the GPU to render cylindrical objects using a single quadrilateral. The approach was used for fast rendering of streamlines in vector field data sets.

3. METHODOLOGY

Visualization of vascular structures can be a valuable tool for analyzing said structure, for example looking for narrowing vessels. Especially with larger models such analysis can be rather complex due to the sheer number of individual vessels. Navigational tools can help reduce that complexity. The task becomes easier once suitable virtual environment techniques are employed that allow the person analyzing the data to be fully immersed in the vascular structure. For this full immersion, large-scale display systems are required combined with a tracking system that reliably identifies the user's location to adjust the view settings based on the user's head position. Such a setup then allows the user to maneuver around within the vascular structure allowing for very fine-grained control of the view perspective thereby making it easier to go from an overview perspective down to specific vessel segment.

3.1 Rendering Vascular Segments

Typically, vascular structures are represented based on the center lines of the vessels and the radii of each vessel segment at both ends. This allows for a reconstruction based on tapered cylinders aligned with the center line using the appropriate radii at both ends.[23] The end caps are rotated in such a way that there is a seamless transition from one vessel segment to the next to achieve a smooth, high-quality visualization of the vascular structure. With increase in number of individual vessel segments the number of tapered cylinders increase accordingly. Since the discs at each end of the tapered cylinder should be discretized with at least 8 points to preserve a roundish appearance these cylinders are rendered using 16 triangles. As a result, the number of triangles can be challenging for the graphics hardware when several million vessel segments have to be rendered. Once the memory used by the geometry exceeds the amount of memory available, out-of-core methods[24] can be deployed to still be able to visualize the vascular structure.

Obviously, the sheer amount of geometry data puts significant stress on the bottlenecks of the computer, in most cases the hard drive, resulting in fairly low frame rates. For example, for a data set composed of 220 million vessel segments, the rendering can take over a minute per frame. Even though this out-of-core approach retains the full rendering quality, this rendering approach is not suitable for interactive rendering anymore once very large vascular structures have to be visualized. Since the bottleneck in the out-of-core approach is the hard drive, the amount of data that is transferred to the graphics hardware needs to be reduced in order to increase the rendering performance. Instead of rendering each vessel segment using a tapered cylinder, which requires 16 triangles, a billboarding approach can be utilized reducing this to just a single quad, thereby reducing the amount of geometry data to just a fraction. A fragment program can be used to colorize the quad in such a way that it appears like a cylindrical object. Of course, the billboard that is used to represent the vessel segment now needs to be recomputed for every frame such that it is always perpendicular to the view direction and the extent of the quad resembles the projection of the vessel segment. Using such an approach the rendering speed can be significantly improved so that a large-scale data set with 220 million vessel segment can be rendered in about a second.[25]

3.2 GPU-based Vascular Rendering

The advent of programmable graphics hardware, specifically the availability of geometry shaders, can help improve the rendering performance even further. Since the recalculations for each quad representing a vessel segment are independent of each other, this can be parallelized very well. Hence, it is ideal for utilizing the parallel hardware capabilities in the graphics hardware where the geometry shaders can all be utilized to their maximum computing all the quads required for the billboards.

In this implementation, the vessel segments are sent to the graphics card as just their center lines where shader programs turn this information into a suitable representation of the individual vessel segment. Since the geometry shader is allowed to generate additional geometry it will simply issue four vertex positions for every incoming line to form a quad created by using a triangle strip with two triangles. The radii at both ends of the vessel segments are encoded in the fourth coefficient of the vertices of the center line, i.e. the weight coefficient of the homogeneous coordinate. Of course, in this case this means that this coefficient is just used for encoding data and not for scaling the 3D location of the point. In order to create a seamless continuation from one vessel segment to the next, the geometry shader needs to know the direction of the preceding vessel segment so that it can angle off the beginning of the quad to perfectly match the end of the preceding vessel segment. Hence, the direction of the preceding vessel segment is encoded in the texture coordinate for the two end points. The geometry shader can then derive the direction easily be using the xy texture coordinates of the first point and combine it with the x texture coordinate of the second point to get all three coefficients of that direction.

The geometry shader then determines a vector that is orthogonal to the direction of the vessel segment and the view direction. This vector defines one end of the quad representing the vessel segment. Similarly, a vector perpendicular to the direction of the preceding vessel segment and the view direction is computed to represent the other end of the quad. Offsets are then computed by using these vectors and scaling them in positive and negative direction by the radii of each end of the vessel segment to represent the four vertices of the quad. Since the offset vectors are all orthogonal to the view direction and the center line of the current and preceding vessel segments respectively, the quad represents the projection of the vessel segment and is guaranteed to be parallel to the projection plane as required for billboarding.

The vertex program used in this implementation effectively disables the projection step normally performed in the vertex program. In this case, the projection is not required here since the geometry shader already takes care of that step. Even though this may be unusual, this strategy made it easier to port the code from the CPU to GLSL as it allowed for the reuse of most of the code and mathematical calculations requiring only minor changes due to the different syntax in GLSL.

In a last step, the fragment program makes the quad appear like a cylindrical object. For that, Phong shading is applied to each fragment using the light position and camera location as well as the normal vector calculated based on the cylindrical object the quad is supposed to represent. Following the Phong illumination model, the calculated intensity values create the appearance of a cylinder instead of just a quad.

3.3 Vascular Rendering in Virtual Environments

In order to create a fully immersive virtual environment, more is needed than just interactive rendering. First, a stereo capable display system is needed which can deliver different images for the left and right eye. Second, a tracking system is required to identify the user's current position. Third, a software setup needs to tie all this together to render images mimicking the user's view point in real time. There are several software packages that assist in creating virtual environments. Aside from commercial packages, such as VegaPrime or CAVElib, free software packages are available as well. Such free software packages are freeVR or VRjuggler. The visualization described in this paper is based on the Vrui toolkit[26] developed by Oliver Kreylos. Compared to the other freely available software packages, Vrui offers more support for different input devices as well as support for multithreaded and multipipe rendering resulting in better rendering performance on more complex cluster-based display configurations. Vrui offers a great deal of flexibility. It can be adapted to various different types of setups ranging from fishtank VR to full-scale CAVE-type displays. In fact, the same binary can be used and based on the hostname of the computer this binary based on Vrui identifies its settings from a configuration file to match the display system. Once the configuration is set up properly, the rendering algorithm needs to be integrated

Figure 2. Set of projectors in a passive stereo configuration.

into the Vrui framework. This is essentially done by adding the rendering routine to the display function of a basic Vrui sample program provided as part of the Vrui distribution.

In terms of hardware, there are a couple of display systems that the software described in this paper was tested on. The first one is a Barco I-space equipped with four Barco Galaxy projectors for three walls and the floor and an A.R.T optical tracking system. The second one is a passive single-screen projection system. The advantage of a passive system is that glasses are inexpensive and standard projectors can be utilized keeping the overall cost at a minimum. Even though there are 3D capable consumer-grade projectors at reasonable cost available, they typically support 3D at full HD resolution only at 24 Hz. This is mainly a limitation of the current HDMI 1.4 standard. This limitation does not apply to a passive setup since both projectors can run at the standard 60 Hz at full HD resolution. Combined with passive polarization filters that are available starting at $35 this results in a very smooth, flicker-free display. The setup that was used for this paper uses a rear-projection setup so that the user does not block any of the projected image. Since passive stereo is based on polarization, a special polarization-preserving projection screen is required, such as the Da-Lite 3D Virtual Black.

For the tracking, a very cost-effective setup was chosen based on Microsoft's Kinect. The Kinect sensor provides full 6DoF data, albeit the directional information is not overly accurate. However, the positional data provided by the Kinect is fairly precise. A dedicated computer running FAAST[27] captures the data from the Kinect sensor and transmits it to the rendering computer via the VRPN protocol.

4. RESULTS AND DISCUSSION

The software described in this paper is capable of visualizing vascular structures from small models to large-scale structures. Thanks to the flexibility inherited from Vrui, it can utilize various different display configurations ranging from desktop setups over 3D capable TV screens all the way to full-size CAVE-type displays. The described passive stereo rear-projection screen configuration provides a comparably low-cost environment with equipment cost of less than $6,000. In order to keep the cost low, a custom-built projection stand was designed consisting of wooden horizontal levels connected vertically by threaded rods. Figure 2 shows the setup of the projectors. The two center levels, where the projectors are sitting on, are very adjustable as they rest on wing nuts that can slide up and down the threaded rods by simply turning them. This provides very fine-grained control over the projected image as all four corners can be adjusted individually so that the overlap between the left and right image produced by each of the projectors can be dialed in very effectively. The linear polarization filters are mounted at a distance of a few centimeters in front of the lens to prevent overheating and melting of the filters.

Figure 3 depicts a user in front of the passive projection screen. The passive stereo glasses with linear polarized filters that match the ones right in front of the projector lenses separate the left and right images.

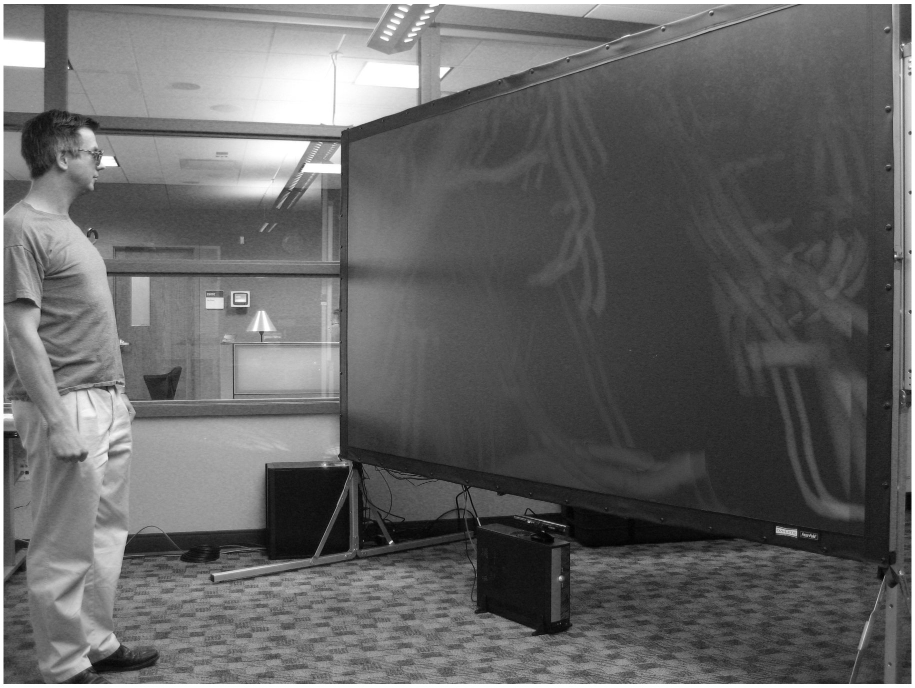

Figure 3. Tracked user in front of the passive projection screen with the tracking computer and the Kinect sensor on top underneath the screen.

Hence, only the left image is seen with the left eye and the right image is only seen by the right eye, unlike the double images that appear in the photograph since no polarization filter was used for taking the picture. The user is tracked by the Kinect sensor, which is located underneath the screen where it sits on top of the tracking computer. The tracked range is rather large ranging from around twelve feet down to just a couple feet in front of the screen, albeit the positional data gets somewhat distorted when getting too close to the screen.

Finally, figure 4 shows a visualization of a large-scale vascular structure consisting of 220 million individual vessel segments in a Barco I-Space. The four screens located on the floor and the three walls are driven by five computers. There is one computer dedicated to each projection surface plus a master node. Each of these computers is equipped with 4 GB of memory, a dual Xeon 2.5 GHz configuration with an Nvidia Quadro FX 5800. The visualization software is capable of rendering at 5.5 frames per second using the geometry shader-enabled rendering configuration. Note that these 5.5 frames per second are achieved in quadbuffered stereo mode. Hence single non-3D rendering could achieve 11 frames per second. So the performance is considerably improved over the out-of-core rendering as well as the CPU-based billboarding rendering approach.

Overall, the immersive visualization of vascular structures makes it easier to navigate the structure since the user's movements change the view perspective. If further navigation is required, for example beyond the current tracked area, joystick type devices are used, such as a Logitech gamepad or the A.R.T flystick 2. Different navigation modes are possible. First, the joystick input itself results in a motion forward or turning sideways. Second, the entire input device itself can be tracked so that – when pushing a button – the entire vascular structure can be moved around based on the motion of the input device. This allows the user to very intuitively maneuver around the vascular structure. As a result, areas of interests can be found more quickly speeding up the analysis. For example, areas with stenosis can be more easily spotted when looking at the vasculature from an overview perspective while the immersive virtual environment with its combined user tracking provides very fine-grained ways to dial in on the stenosis. Since the entire geometry of the vascular structure is available, almost any quantitative information about the vasculature, such as vessel volume or cross-sectional area, can be computed at any point in time and space as requested by the user and embedded in the visualization to provide further insight. Additional quantitative information can be overlaid with the visualization. For example, unusual

Figure 4. User experiencing the visualization of a large-scale vascular structure in a Barco I-Space.

bifurcation angles can be highlighted for further inspection by the user. Similarly, narrowing arteries indicating stenosis can be detected automatically based on the vascular geometry and color coded to make it easier for the user to find. This then presents a very powerful tool for the analysis of the vascular structure.

ACKNOWLEDGMENTS

The author would like to thank Ghassan Kassab for providing the CT data that was used to generate some of the vascular models and Benjamin Kaimovitz for providing the large-scale vascular data. Funding for this research was provided by the College of Engineering and the Department of Computer Science and Engineering at Wright State University.

REFERENCES

[1] Lorensen, W. E. and Cline, H. E., “Marching cubes: A high resolution 3D surface construction algorithm,” *Computer graphics* **21**, 163–168 (Jul. 1987).

[2] Nielson, G. M. and Hamann, B., “The asymptotic decider: Removing the ambiguity in marching cubes,” in [*IEEE Visualization '91*], Nielson, G. M. and Rosenblum, L. J., eds., 83–91, IEEE, IEEE Computer Society Press, San Diego, CA (1991).

[3] Montani, C., Scateni, R., and Scopigno, R., “A modified look-up table for implicit disambiguation of marching cubes,” *The Visual Computer* **10**, 353–355 (Dec. 1994).

[4] Hong, L., Muraki, S., Kaufman, A., Bartz, D., and He, T., “Virtual voyage: Interactive navigation in the human colon,” *Proceedings of SIGGRAPH'97* , 27–34 (1997).

[5] Oeltze, S. and Preim, B., “Visualization of vasculature with convolution surfaces: method, validation and evaluation,” *Medical Imaging, IEEE Transactions on* **24**, 540 –548 (april 2005).

[6] Kanitsar, A., Wegenkittl, R., Fleischmann, D., and Gröller, E., “Advanced Curved Planar Reformation: Flattening of Vascular Structures,” in [*IEEE Visualization 2003*], Turk, G., van Wijk, J. J., and Moorhead, R., eds., 43–50, IEEE (Oct. 2003).

[7] Puig, A., Tost, D., and Navazo, I., "An interactive cerebral blood vessel exploration system," in [*IEEE Visualization '97*], Yagel, R. and Hagen, H., eds., 443–446, IEEE, IEEE Computer Society Press, Los Alamitos, CA (1997).

[8] Wischgoll, T., Choy, J. S., Ritman, E., and Kassab, G. S., "Validation of image-based extraction method for morphometry of coronary arteries," *Annals of Biomedical Engineering* **36**(3), 356–368 (2008).

[9] Wan, S.-Y., Ritman, E. L., and Higgins, W. E., "Multi-generational analysis and visualization of the vascular tree in 3d micro-ct images," *Computers in Biology and Medicine* **32**, 55–71 (March 2002).

[10] Jiang, Y., Zhuang, Z. W., Sinusas, A. J., Staib, L. H., and Papademetris, X., "Vessel connectivity using Murray's hypothesis," *Med Image Comput Comput Assist Interv.* **14**(Pt 3), 528–536 (2011).

[11] Kassab, G. S., Rider, C. A., Tang, N. J., and Fung, Y.-C. B., "Morphometry of pig coronary arterial trees," *Am. J. Physiol. Heart Circ. Physiol.* **265**(1), H350–H365 (1993).

[12] Kassab, G. S., Lin, D. H., and Fung, Y.-C. B., "Morphometry of pig coronary venous system," *Am. J. Physiol. Heart Circ. Physiol.* **267**(6), H2100–H2113 (1994).

[13] Kaimovitz, B., Lanir, Y., and Kassab, G. S., "Large-scale 3-d geometric reconstruction of the porcine coronary arterial vasculature based on detailed anatomical data," *Ann. Biomed. Eng.* **33**(11), 1517–1535 (2005).

[14] Hahn, H. K., Preim, B., Selle, D., and Peitgen, H. O., "Visualization and interaction techniques for the exploration of vascular structures," in [*IEEE Visualization 2001*], 395–402 (2001).

[15] Masutani, Y., Masamune, K., and Dohi, T., "Region-growing-based feature extraction algorithm for tree-like objects," *Visualization in Biomedical Computing* , 161–171 (1996).

[16] Felkel, P., Fuhrmann, A. L., Kanitasar, A., and Wegenkittel, R., "Surface reconstruction of the branching vessels for augmented reality aided surgery," *BIOSIGNAL* **1**, 252–254 (2002).

[17] Puig, A., Tost, D., and Navazo, I., "An interactive celebral blood vessel exploration system," in [*IEEE Visualization 97*], 443–446 (1997).

[18] Oeltze, S. and Preim, B., "Visualization of Anatomic Tree Structures with Convolution Surfaces," in [*IEEE/Eurographics Symposium on Visualization*], 311–320 (2004).

[19] Oeltze, S. and Preim, B., "Visualization of Vasculature with Convolution Surfaces: Method, Validation and Evaluation," *IEEE Trans. Medical Imaging* **24**(4), 540–548 (2005).

[20] Ritter, F., Hansen, C., Dicken, V., Konrad, O., Preim, B., and Peitgen, H.-O., "Real-Time Illustration of Vascular Structures," in [*IEEE Visualization 2006*], 877–884 (2006).

[21] Stoll, C., Gumhold, S., and Seidel, H.-P., "Visualization with stylized primitives," in [*IEEE Visualization 2005*], Silver, C., Gröller, E., and Rushmeier, H., eds., 695–702 (2005).

[22] David Blythe, Silicon Graphics, T. M., "Advanced graphics programming techniques using opengl," in [*SIGGRAPH 2000 Course 32*], (2000).

[23] Wischgoll, T., Moritz, E., and Meyer, J., "A quantitative analysis tool for cardiovascular systems," in [*International Conference on Biomechanics (BioMech 2005)*], 121–126 (2005).

[24] Wischgoll, T., Meyer, J., Kaimovitz, B., Lanir, Y., and Kassab, G. S., "A Novel Method for Visualization of Entire Coronary Arterial Tree," *Annals of Biomedical Engineering* **35**(5), 694–710 (2007).

[25] Wischgoll, T., "Modeling and visualization of cardiovascular systems," *Dagstuhl Follow-Ups* **1** (2010).

[26] Kreylos, O., "Environment-Independent VR Development," *Lecture Notes in Computer Science* **5358**, 901–912 (2008).

[27] Suma, E., Lange, B., Rizzo, A., Krum, D., and Bolas, M., "Faast: The flexible action and articulated skeleton toolkit," in [*IEEE Virtual Reality*], 247–248 (2011).