A Sketch-Based Interface for 2D Illustration of Vascular Structures, Diseases, and Treatment Options with Real-Time Blood Flow

Patrick Saalfeld[1(✉)], Alexandra Baer[1],
Uta Preim[2], Bernhard Preim[1], and Kai Lawonn[1]

[1] Department of Simulation and Graphics, Otto-von-Guericke University, Magdeburg, Germany
saalfeld@isg.cs.uni-magdeburg.de
[2] Department of Diagnostic Radiology, Municipal Hospital Magdeburg, Magdeburg, Germany

Abstract. We present a sketching interface, which enables physicians to illustrate various vascular structures, diseases, and treatment options with integrated blood flow. This sketch-based interface provides medical doctors with an effective tool to illustrate different medical scenarios and support patient education. This work integrates methods from sketch-based interfaces and GPU-supported computational fluid dynamics. The usability of the prototype was assessed qualitatively and quantitatively. Additionally, we performed a structured interview with a physician to evaluate the benefits with respect to patient education. The results of the evaluation confirmed the usability of the prototype as well as the usefulness to support physicians during the process of patient education.

Keywords: Sketch-based interface · Vascular diseases · Treatment options · Patient education · Computational fluid dynamics

1 Introduction

The field of vascular diseases causes 31 % of deaths worldwide [1] and has a big impact on economics (€ 196 billion in Europe [2]). In treatment planning, including aspects as prevention, diagnosis, and therapy, physicians discuss treatment options not only with colleagues, but also with patients. In this process of patient education, patients place great value on an understandable presentation of their disease and therapy. Such an appropriate presentation results in various positive aspects [3]:

– the time of treatment may be reduced,
– patients need less medication,
– they are more active in dealing with their diseases and act more responsible, and
– they are more independent from their attending physician.

J. Braz et al. (Eds.): VISIGRAPP 2015, CCIS 598, pp. 19–40, 2016.
DOI: 10.1007/978-3-319-29971-6_2

As also stated by Keulers [3], 42 % of patients feel not adequately informed. Therefore, a method that supports physicians to illustrate and discuss vascular diseases is useful. Such a method are sketches [4,5]. We present a prototype that allows the sketching of different vessel structures, vascular diseases, and various treatment options. This work is an extension of our previous paper [6] and inspired by the work of Zhu et al. [7] about sketching tubular shapes and simulating liquids. Since Zhu et al.'s work refers to wider area of application domains, it has limitations with respect to specifically sketching vascular diseases and treatments. Furthermore, it is not designed for tablet devices, which could be integrated well in the process of patient education. Therefore, we make the following contributions:

- presentation of methods to sketch vessels, vascular diseases, and treatment options,
- plausible integration of real-time blood flow in the sketched vessel structures,
- demonstrate the applicability for physicians with qualitative and quantitative usability measures, and
- showing the usefulness of patient education with a structured interview.

2 Medical Background

This work focuses on the vascular diseases of arteries, i.e., vessels which transporting blood from the heart to the peripheral capillary of the body. A common reason of these diseases is arteriosclerosis, which leads to a hardening of vessels by deposition of blood fat, thrombi and lime [8]. This deposition affects vessels in two problematic ways:

1. A weakening of the vessel wall.
2. A narrowing of the vessel up to a complete occlusion.

The weakening can lead to a dilation of the vessel that may result in an aneurysm, see Fig. 1(a). These are dangerous out of two reasons: first, they can rupture and release blood into the space around the vessel. This is especially critical in the brain, because patients with a ruptured cerebral aneurysm have a mortality rate of 40–60 % [9,10]. Second, the blood can clot inside the aneurysm, which could be carried away and block other arteries. The narrowed vessel, called a *stenosis* (see Fig. 1(b)), can lead to an under supply of involved structures or also cause a clot formation.

For the medical treatment of such vascular diseases, the physician can use several methods. The choice of the treatment depends on parameters such as anatomical access or size and shape of the pathology. In particular, this work focuses on *clipping* as an example of an extravascular treatment method as well as *coiling* and *stenting* as examples of intravascular methods. The *clipping* procedure, e.g., for treating a cerebral aneurysm, starts with a craniotomy to disclose the aneurysm. Afterwards, a titanium clip is placed across the aneurysm neck to stop the blood from entering into the aneurysm [11]. *Coiling* is performed

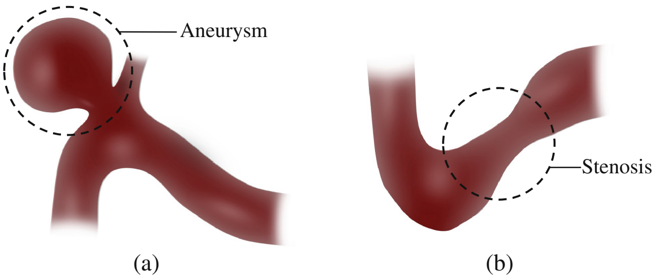


Fig. 1. Illustrations of a saccular aneurysm (a), which is the most common type, and a stenosis (b) caused by a narrowed vessel.

by entering an artery from the *inside*. The coil, a small titanium wire, is used to fill the aneurysm and to induce a thrombus formation [12]. The *stenting* method can be used to treat both, stenosis and aneurysms through an intravascular approach. Similar to the coiling procedure, a catheter is moved to the affected position from the inside. Afterwards, the stent is inflated with a balloon and forces the vessel to expand. For treating aneurysms, stents can help to support the involved vessel during, e.g., a coiling procedure.

Further descriptions of different forms of vascular diseases, their treatments, and possibilities for visualization and exploration are mentioned by Gasteiger [11]. A historical overview of different treatment options can be found in the work of Wong et al. [13].

3 Related Work

This work involves three main topics: computational hemodynamics, flow visualization, and Sketch-based Interfaces (SBIs), a form of user interface (UI) which deals with sketching.

3.1 Computational Hemodynamics

To simulate the behavior of blood, it is necessary to imitate a *non-Newtonian fluid*, i.e., a fluid with varying viscosity. Furthermore, in terms of fluid dynamics, blood is *compressible* and *inhomogeneous*. Such a simulation is complicated and expensive regarding calculation time [14]. Therefore, we consider blood as an incompressible, homogeneous *Newtonian fluid* to achieve a real-time simulation. Examples for methods to calculate non-Newtonian fluids can be found in [15, 16]. To describe the state of a fluid, there are two possibilities: the Lagrangian (particle-based) and the Eulerian (grid-based) description. We use the Eulerian description because the grid-based character is well suited to be calculated with fragment shaders on the GPU. Examples for the Lagrangian description can be found by Müller et al. [17] and Qin et al. [18]. Both deal with particle-based simulation of blood flow in vessels to support surgeons in virtual surgery scenarios.

3.2 Flow Visualization

The simulated flow can be visualized in a variety of ways. Since we use the Eulerian description, we illustrate parameters on each point of the discretized grid. These flow visualizations can be divided in three categories: direct, sparse, and dense visualizations [19,20]. The first two are relevant for our work since they can illustrate flow in a simple and thus, comprehensible way. In *direct flow visualizations*, glyphs such as arrows are used to indicate the flow direction on each grid-point. Additional parameters, such as the strength of the flow, can be visually encoded with the arrow length [21]. Since direct visualization enables the user to get a fast overview of the flow, we use it in our work. *Sparse flow visualization* techniques uses lines to illustrate the flow. The lines are obtained by seeding and tracking particles in the flow. Here, a challenge is to find appropriate seed positions for the particles [20]. We want to enable the physician to control where the particles are seeded. This gives the possibility to show important regions to the patient. Therefore, the seeding points can be placed interactively.

3.3 Sketch-Based Interfaces

The usefulness of SBIs to communicate ideas and concepts is described among others by Jorge and Samavati [22]. They state that the communication of complex issues is possible without the necessity to draw precisely and accurately. However, this is a challenge regarding the automatic recognition of the sketch. As a consequence, several works deal with the interpretation of sketches [22]. The foundations for this were laid by Ivan Sutherland [23] with the program *Sketchpad*. SBIs can be seen as a part of post-WIMP (windows, icons, menus, and pointers) UIs because the sketching is performed with direct input, e.g., a pen or touch [24]. Therefore, the pointer component of the WIMP paradigm is no longer necessary [25]. Xiaogang et al. [26] and Naya et al. [27] showed the advantage of reality-based interfaces (such as SBIs) by comparing WIMP-based interaction with reality-based interaction and presented two findings: first, the users preferred the sketch-based approach and second, they were more efficient with it. SBIs contain three processing steps: *resampling* of the input data, *beautification*, and *recognition*. This paper deals only with the resampling of the input data. Examples for the beautification step can be found, e.g., by Igarashi et al. [28]. There, line segments were analyzed according to geometric constraints such as perpendicularity and parallelism. After that, the program recommends different options of how to interpret the lines. The user chooses an option by clicking on it. The processing step *recognition* describes a procedure where the sketch is compared with an internal representation of symbols. The similarity is expressed with a parameter. If this parameter exceeds a value, the sketch is interpreted as the compared symbol [22]. A simple way to integrate recognition in an application is described with the 1¢ *Recognizer* [29].

An important use case of SBIs is geometric modeling, i.e., the creation of 3D structures. An example of an SBI for modeling (SBIM) medical structures is described by Pihuit et al. [5]. They describe methods to sketch and model branching vessels. To maintain the sketch-based look of 3D models, it is possible to visualize them with non-photorealistic rendering. A comparison of different line drawing in the medical domain can be found in the work of Lawonn et al. [30]. Contrary to SBIM, the following work addresses the creation of 2D vessels.

4 Methods

Our sketch-based interface consists of different concepts to create vessels with interior blood flow based on intuitive sketch-based interaction. In the following, we describe the methods and the implementation.

4.1 Blood Flow Simulation

We use the Eulerian grid-based method to describe the state of the fluid [14, 31, 32]. For the fluid simulation itself, we use the Navier-Stokes equations for *incompressible* and *homogenous* fluids, which are based on Newton's second law of fluid motion:

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{1}{\rho}\nabla p + v\nabla^2 u + F, \tag{1}$$

and

$$\nabla \cdot u = 0. \tag{2}$$

Equation 1 (*momentum equation*) describes the behavior of the velocity vector field u under influences such as advection, diffusion, pressure, and external forces, which will be described in more detail in the following. Equation 2 (*continuity equation*) ensures the incompressibility by defining u as a divergence-free vector field.

To achieve a real-time fluid simulation, the grid size is an important factor. While a small grid size accelerates the calculation time, details like whirls may be lost. Another problem are obstacles. To faithfully simulate the behavior of blood flow in the vicinity of obstacles, it is necessary to model the obstacles in the simulation grid. This is achieved by marking grid cells as occupied. Thus, the grid resolution also affects the possible level of detail of the obstacles. To allow a high spatial resolution of grid cells, the simulation is performed on the GPU. For the GPU-based calculation, we used the fragment shader similar to [31].

To solve the equations for the differential operators, the finite difference method is used. Furthermore, Eq. 1 needs to be split up in single terms, which are calculated separately.

An overview of the calculation steps is illustrated in Fig. 2. In the following, we describe the mathematical terms and their effects in the fluid simulation.

The first term $-(u \cdot \nabla)u$ describes the self-advection of the fluid, which is the process of moving the velocity itself through the fluid. Here, the self-advection is realized with semi-Lagrangian advection [32]. Mostly, fluid simulations use the Runge-Kutta method [33] for the integration, which is less error-prone than

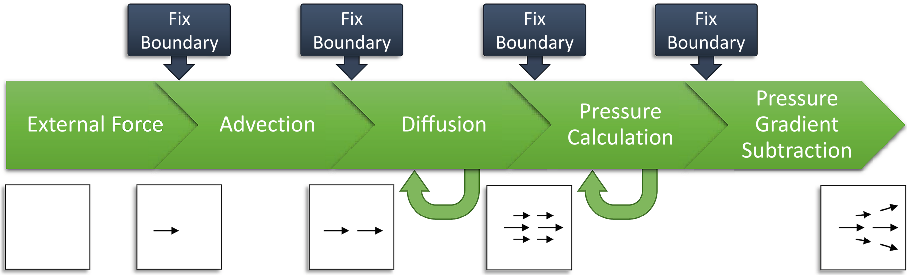


Fig. 2. Pipeline for the fluid simulation including the influence to the underlying vector field. Each step is realized with a separate fragment shader program. The steps provided with self-referencing arrows show the steps which are calculated with an adjustable amount of iterations. Here, a trade-off arises between accuracy (high amount of iterations) and calculation time (low amount of iterations).

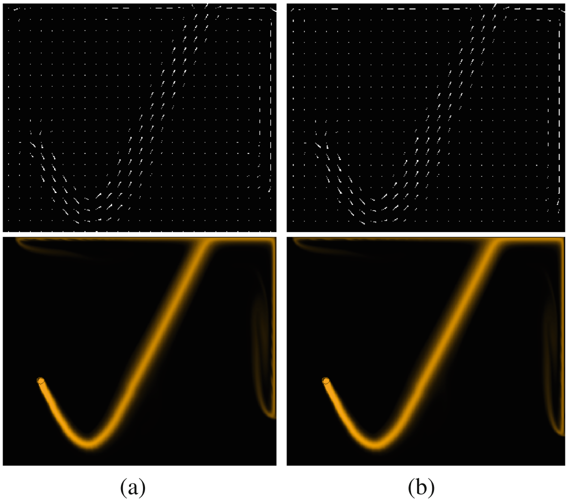


Fig. 3. Comparison of the Euler (a) and Runge-Kutta 4 (b) integration methods with a step size of 0.3. Both methods are depicted with a modified arrow plot visualization (top) and a scalar field visualization (bottom), which looks similar to a streakline visualization through the continuous placement of ink during the simulation. Since the differences between both methods are small, the faster Euler method is used.

the simpler and faster Euler method. Interestingly, as illustrated in Fig. 3, the differences between both methods are small. This is due to the small step size used in the application. Therefore, we used the Euler method to decrease the calculation time.

The pressure, a force that gradually spreads from regions with high to regions with low pressure, is described with the second term $\frac{1}{\rho}\nabla p$. The factor ρ is a constant to describe the density. Furthermore, this term also ensures the incompressibility of the velocity vector field, and thus, simultaneously ensures the Eq. 2. To achieve this, the term is calculated at the end. For a description of the derivation, see [14]. To calculate this term, it is necessary to solve a Poisson equation. This is accomplished by the Jacobi approach because it can be mapped directly to GPU facilities. For a discussion of different approaches, see [34].

The third term $v\nabla^2 u$ expresses the physical process of diffusion, i.e., the property of mixing materials without external forces. Here, v is a constant that describes the viscosity. The calculation of the diffusion also requires the solution of a Poisson equation. A disadvantage of diffusion is the resulting smoothing effect on the applied vector field, which causes a loss of details. Therefore, the choice whether diffusion is applied is left to the user.

The last term F describes external forces, which allows the user to influence the simulation dynamically. Normally, such forces are steady, so the influence of the force to the fluid is constant over time. This approach would not represent the pulsating character caused by the contractions of the heart. This pulsation can be imitated by applying a factor to the force, which changes over time. Therefore, we approximate the function measured by an electrocardiogram with the following formula:

$$f(x) = -\frac{3}{4}e^{-\frac{1}{2}(-\frac{1}{2}+x)^2} + 1,12e^{-\frac{x^2}{2}} - \frac{1}{4}e^{-\frac{1}{2}(1+x)^2} + 1. \tag{3}$$

This equation was determined by combining three Gaussian functions with varying heights and widths. In Fig. 4, the difference between a constant and a pulsating force is illustrated.

Finally, boundary conditions are used to simulate the behavior of the fluid at the vessel wall and the boundary of the sketching canvas. These conditions are necessary for the *velocity* vector field and for the scalar field, describing the *pressure*. For the velocity vector field, a Dirichlet boundary condition is used,

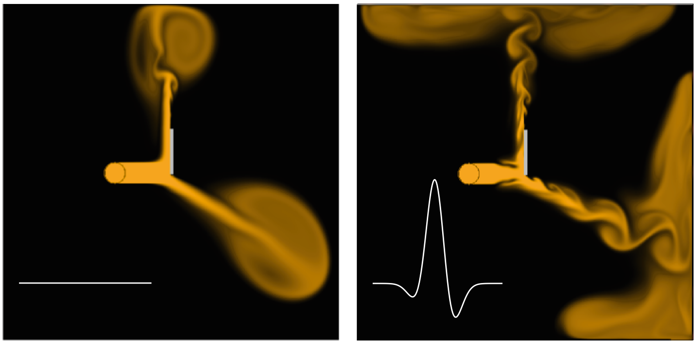


Fig. 4. In (a) a constant force is applied to the vector field. In (b) a heartbeat-like function is applied as a factor, which mimics a more realistic behavior. The corresponding functions are plotted at the left bottom.

which states that the velocity drops to zero at boundaries. For the scalar field, a Neumann condition is used, which states that the derivatives at boundaries are zero. To calculate the derivative, the normal of the boundary is necessary. The normals of the top, right, bottom, and left image boundaries are defined as $(0, -1), (-1, 0), (0, 1)$, and $(1, 0)$, respectively. For determining the normals of arbitrary boundaries, we use the neighborhood of the obstacle. That means that for each grid cell, which is marked as an obstacle, eight neighbor cells are analyzed. Depending on the state of these neighbors (marked as obstacle or no obstacle) the normal is approximated to one of eight possible directions. This approach is described in more detail by Wu et al. [35].

4.2 Blood Flow Visualization

To illustrate the unsteady vector field that represents the blood flow, we implemented two visualization concepts: (1) a direct and (2) a sparse flow visualization, see Fig. 3. The direct visualization is a modified line plot on which fans are drawn on an adjustable grid. Fans are used because they facilitate a fast realization of the flow direction for the user.

Especially for patients, a scientific visualization method for vector fields may be inappropriate. Thus, an additional method is used. It aims to be easily understandable to visualize the behavior of blood in areas such as aneurysms and stenoses in a descriptive way. The used scalar field visualization is inspired by the idea to place *colored ink* in the vector field (also known as *dye injection*). By diffusion and advection, this ink is transported through the vector field. The amount of ink is color-coded with a black-to-hue scale with different colors. This allows using multiple colors, e.g., to show how blood mixes in an aneurysm before and after a treatment, see Fig. 5.

The colors are taken from the *CIELAB* color space, which allows choosing colors that are roughly perceptually linearized regarding hue and brightness. To determine perceptually strongly different colors, the approach of Glaßer et al. [36] is used. If the ink is placed continuously over time, the visualization technique is similar to streaklines. In contrast to streaklines, the placed ink is not connected, but if the amount of placed ink is high enough, there is the impression of connectivity. A difficult task involved in sparse visualization techniques, namely to identify suitable seed point positions, is left to the user. This allows the physician to emphasize specific areas, which supports the patient's understanding.

4.3 Sketching

The obtained data from the input device inherits noise, which is caused by the conversion from the analog to a digital signal (quantization) as well as the imprecise input from the user. Especially the quantization, also depending on the sampling rate, reduces input information during fast input movements from the user. To remove the resulting noise and obtain equidistant input information, the received data of the input device is resampled and smoothed. These steps are commonly applied after the user finishes drawing, which causes abrupt

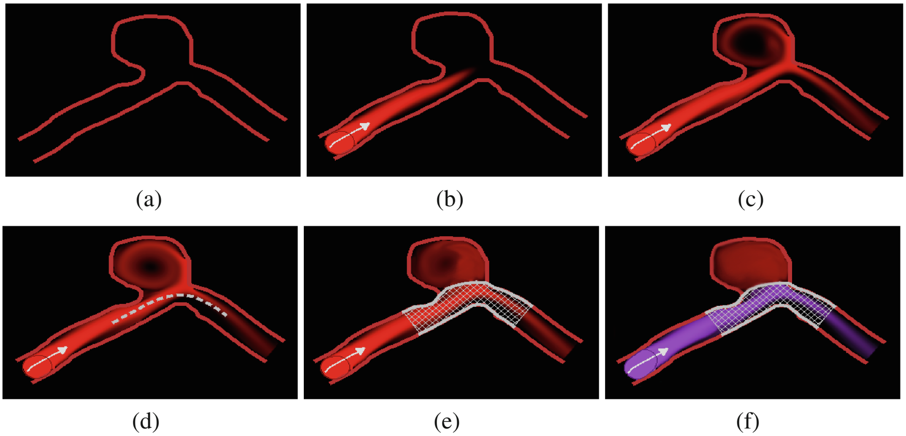


Fig. 5. Illustration of blood flow before and after a stenting treatment. First, the physician sketches a vessel with an aneurysm (a). Afterwards, he manipulates and visualizes the underlying fluid simulation to illustrate the blood flow (b-c). The consequences resulting from the stent treatment (d-e) are illustrated with another color (f).

changes in the sketch. To avoid these sudden changes, we use on-the-fly methods that are applied during sketching. This presents challenges regarding real-time capability. Ideally, the user did not even realize these steps. First, the resampling should reduce the obtained sample points. To achieve this, we use a simple strategy: we ignore all points that are too close to the last accepted point. We use the Euclidean distance to measure the distance of two points, see Fig. 6(a).

A disadvantage of this approach is that it leads to line smoothing and thus, complicates the sketching of zigzag lines. This can be neglected, because vessels, vascular diseases, stents, and coils usually do not have these shapes. To smooth the accepted sample points, a local Gaussian filter is used [37]. More precisely, the 1-neighborhood of the accepted points is used to adjust the points with a

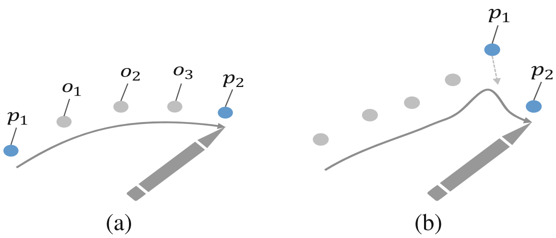


Fig. 6. (a) The new point p_2 has exceeded a specific distance to p_1 and thus, is sampled. The points o_1 - o_3 are too close and thereby omitted. (b) shows the smoothing approach. When p_2 is sampled, the pre-last point p_1 is translated. By transforming the pre-last point, a shrinking of the line is prevented.

Gaussian smoothing. The filter is only applied to the *pre-last* point, which was accepted to prevent the line segments from shrinking, see Fig. 6(b).

5 Application

The application should be as flexible as possible regarding the used direct input device. More precisely, it should be possible to control the application with touch and pen-based input. This implies the following limitations:

- Special functions from pen-based input devices like pressure sensitivity or additional buttons are not supported.
- Multitouch input is not supported.

Through these limitations, the used concept is theoretically usable for a mouse with a left button only. Furthermore, the application is designed according to major usability criteria, such as *suitability for the task* or *self-descriptiveness*. The evaluation of the application revealed possibilities for improvements, which were partially implemented. Improvements based on the quantitative part, the qualitative part, or the interview, are marked and discussed, respectively. The application can be downloaded under www.isg.cs.uni-magdeburg.de/~patrick/application/SketchingVessels.zip.

5.1 Sketch Vessels and Vascular Diseases

Mainly, drawing a vessel requires two lines representing the border of the vessel. Drawing each line separately would lead to strong variations in the resulting vessel structures. Therefore, we use a *create tool* that creates both lines simultaneously by using the sketched path as the *center line* and drawing the vessel wall around it. The general advantage of this process is that uniform vessels can be drawn easier and faster. We fix the width of the vessel to simplify the application. In contrast to Zhu et al. [7], where it is possible to draw vessels under and over already existing ones through a 2,5D sketching canvas, this work limits the sketching area to a *2D canvas*. This decision is motivated by the observation of what could happen if the user sketches over an already created vessel. Besides the possibility to draw the vessel over or under the existing ones, it is possible to *merge* the new vessel with the old ones. This offers the possibility to create more complex structures like branching vessels and aneurysms easily without changing the drawing mode. The merge behavior is illustrated in Fig. 7 and is realized with an already implemented polyline-based functionality in the used framework.

Additionally to the possibility to draw aneurysms, the application offers a possibility to draw stenoses. A *cut tool* is used to allow the user to create irregular non-symmetrical stenoses in a consistent sketch-based way. To prevent problems during cutting, e.g., ambiguity described by Heckel et al. [38], the user sees his sketched contour, which is used for the cutting process. Additionally, the start

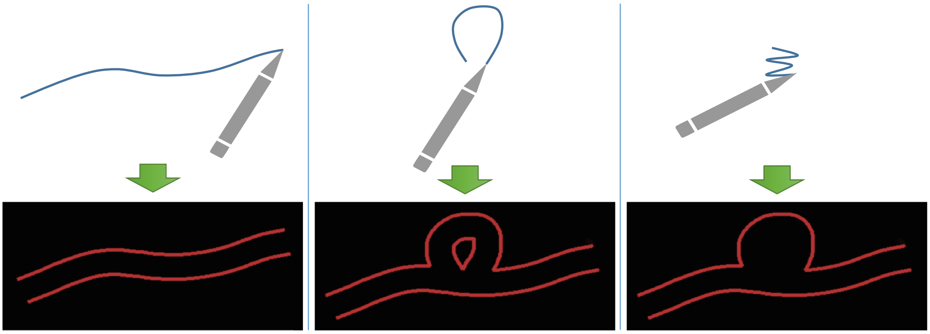


Fig. 7. The merging behavior can be used to draw a vessel containing an aneurysm by simply sketching on top of the previous drawing.

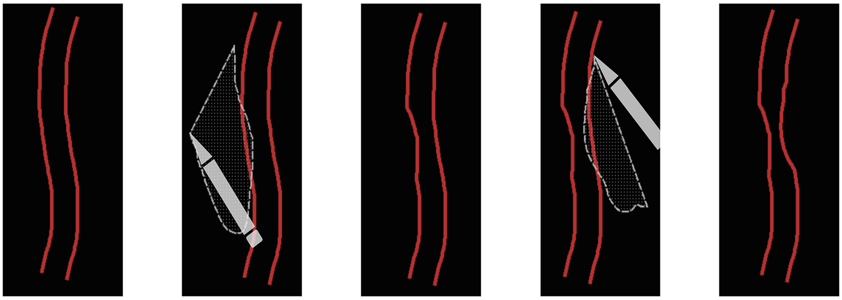


Fig. 8. The *cut tool* is used to remove structures from the already sketched vessel to create a stenosis.

and end point of the sketched contour are connected, and thus, span a cutting area. This area is subtracted from the existing vessels (see Fig. 8).

This tool does not only allow the creation of stenoses: in combination with the *create tool*, the user has a generic sketching tool, which allows the creation of any 2D structure under the usage of only two different modes. To support the medical expert during the sketching process, it is possible to load images in the background. The physician can use this function to load a slice of patient-specific MRI or CT data which contains, e.g., the vessel structures of the patient. In addition, frequently used vessel structures such as the Circle of Willis, can be loaded (see Fig. 9).

During the evaluation, some participants suggested a possibility to load vessel structures. This could help because it allows the physician to not only load standardized structures, but also patient-specific data. The functionality is implemented by loading a monochrome black-and-white image. Every black pixel is interpreted as a vessel and every white pixel as free space, where blood is able to flow. However, this approach has a disadvantage. It is only possible to show blood flow in the loaded vessel structure, but not to use the other tools, e.g.,

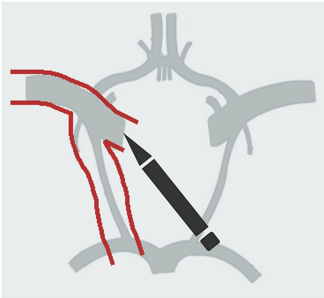


Fig. 9. A background image of the Circle of Willis is loaded to support the sketching process.

cutting or treating the vessel. This disadvantage can be avoided by implementing an object-based save and load mechanism instead of an image-based, which could be added in the future.

5.2 Manipulate and Visualize Blood Flow

The flow can be manipulated with a force term, which is represented with a 2D vector field (recall Sect. 4.1). To allow the user to influence this field in a flexible and easy way, a *direction tool* is provided, which is also implemented in a sketch-based way. After the *direction tool* is selected, the user can sketch lines that are represented by arrows pointing along the draw direction. This arrow represents the influenced force on the vector field.

To transfer the sketched arrow to the vector field, it would be obvious to manipulate only the vectors directly under the sketched arrow. This behavior is not desirable, because it only allows to manipulate the flow in small areas. Instead, the force is applied in a *region* around the sketched arrow. The size of this region is adjustable and is initially set to the width of a vessel. To achieve a natural effect, the force is slightly decreased at the border of the region by applying a Gaussian smoothing, resulting in a strong force at the center and a weak force at the border.

To visualize the flow, the user can use a *dye tool* to place ink (blood) in the fluid region. A circle of ink is placed by just tipping on the canvas or drawing over an area. The ink is interpreted as a source of infinite amount of ink, which is added in every render frame. The width of the ink area is adjustable, but set to the vessel width to allow the user to fill a vessel with ink in a fast way. To allow the physician to show the mixing behavior of fluid in an easy way, a new color is selected (recall Sect. 4.2) after each usage of the *dye tool*.

5.3 Treating Options

To show the patient how to treat aneurysms and stenoses, we implemented the following treatment methods:

- coiling,
- clipping, and
- stenting.

Initially, we implemented the *coiling tool* such that the user should fill out the aneurysm. This coil was illustrated with a line, which occurred during the drawing process. Our user study revealed that generating the coil with this approach is not applicable, since it takes too much effort and is too time-consuming. Therefore, we implemented another approach to improve the coil placement. First, an area is sketched, e.g., on the vessel wall of an aneurysm. After the user raises the pen, every sample point analyzes its neighborhood within a specific adjustable distance. This is performed by sending eight rays in a circular manner, starting at every sample point. Significantly less rays would result in an inaccurate vessel wall detection and more rays would not improve the result. We test every ray for a possible intersection with the vessel wall. If more than one ray collides with the wall, the ray with the shortest distance is chosen. After that, the corresponding sample point is placed on the intersection point of the ray. If no collision occurs, the point keeps the current position. The underlying grid cells are than treated as obstacles and thus, the blood flow changes dynamically corresponding to the drawn coil. Figure 10 illustrates this algorithm.

The *clipping method* was inspired by the line drawing method used in *Sketchpad* [23]. The point where the user starts drawing represents the first point of the clip. As long as the user draws with the pen, the current pen position represents the end point of the clip. These two points are connected with a dashed line to

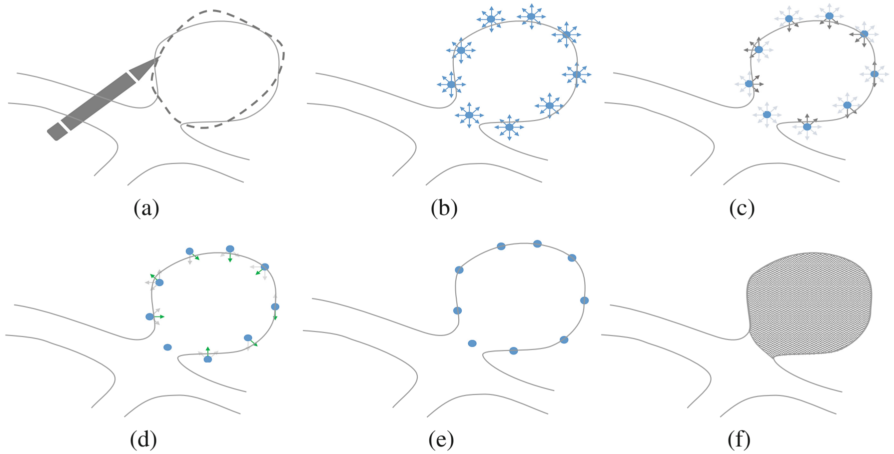


Fig. 10. (a) - (f) shows the algorithm to calculate the coil area. In (a) the user sketches imprecise over the vessel wall of the aneurysm. (b) shows the captured sample points as well as the rays, which are sent in a circular manner from each sample point. In (c) all rays that collide with the vessel wall are highlighted, from which those with the shortest distance are chosen (d). The sample points are moved to the intersection point or remain on their position if no intersection occurred (e). The adjusted sample points build the new area for the coil, which lies precisely on the vessel wall.

give the user a preview of the clipping result. After the user finishes drawing and raises the pen, the clip is placed and the blood flow simulation is affected by it.

The *stent placement* algorithm is inspired by the real treatment. Here, a balloon catheter is inflated to dilate the stent in the vessel. We use this inflating process for providing a stent placement algorithm. The user draws a line in the center of the vessel with the *stenting tool*, which represents the position of the balloon catheter. After the user finishes the sketching process, the application calculates the dilation of the stent. The stent should be dilated in the relevant vessel, but not enter the aneurysm. Since structures are not differentiated semantically (i.e., in vessels and aneurysms), the described behavior must be achieved in a different way. To accomplish this, the surrounding region of the sketched stent is analyzed. The algorithm is described in detail in Fig. 11. This method is more robust according to various input lines. The best results regarding visual aspects are achieved by drawing a line, which is as close as possible to the center of the vessel. Inaccurate lines may lead to an entering of the stent inside the aneurysm. A disadvantage of the described method is that it depends on four control points obtained through the start and the end point. If not all control points could be determined, e.g., if the vessel is too wide, not all normals are calculated and so the stent will not be placed. Similar to the coiling and clipping method, the grid cells under the stent are marked as occupied and thus, influence the fluid simulation.

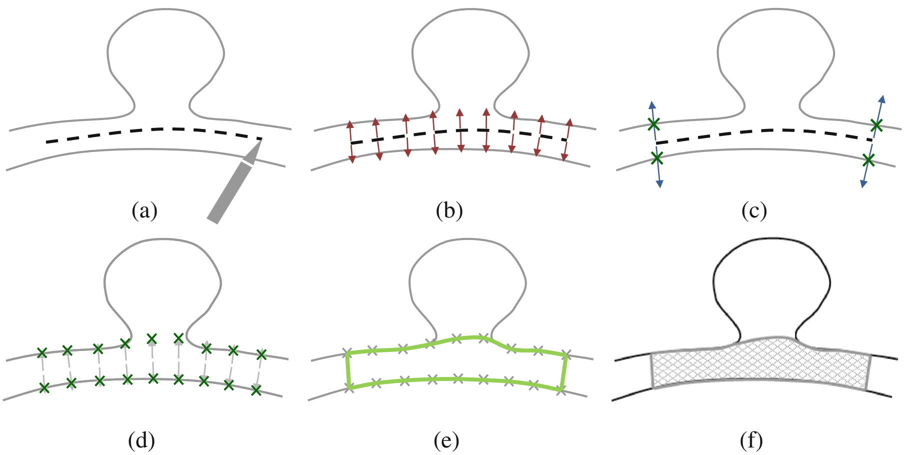


Fig. 11. (a) - (f) illustrate the algorithm to calculate the stent dilation. The user input is illustrated as a sketched line in (a). After the user finishes the stent, the normals of the start and end point are determined (b). The normals have the length of the vessel width. All four normals are tested if they intersect the vessel wall. If not, the stent is discarded. Otherwise, the distance l is determined, which is the longest length of the normal start point to its intersection point. Now, for every point of the sketched line, the normals with length l are determined (c). These normals are tested for intersections with the vessel wall. If they intersect, they are shortened to the intersection point, otherwise they keep the length l (d). The achieved end points of the normals are connected (e) and form the border of the stent, which is illustrated in (f).

5.4 Edit, Delete, and Copy Objects

The possibility to edit and delete objects such as stents and coils, is an important aspect regarding the controllability of the application. This is confirmed by the evaluation (see Sect. 6.1) in which the participants stated that a functionality to delete or edit objects would be useful. To realize this, it is necessary to implement a method to select already created objects. Since a requirement of the application is to control it with a pen without any further buttons, a suitable method is necessary. Besides the use of gestures or a double tap of the pen, the implemented approach is to *press and hold* the pen on the relevant object. Afterwards, the corresponding object is deleted. The possibility to edit objects, which is currently not implemented, can be realized with the same approach.

To provide the physician with an easy and fast possibility to illustrate more treatment options on the same vessel structures, a *copy tool* is added to the application. It allows to select an area, copy its content and paste it at another place. With this possibility, the patient can see different behaviors of the blood-flow, depending on the treatment option. This can help to understand why a specific treatment is chosen or why another is not possible.

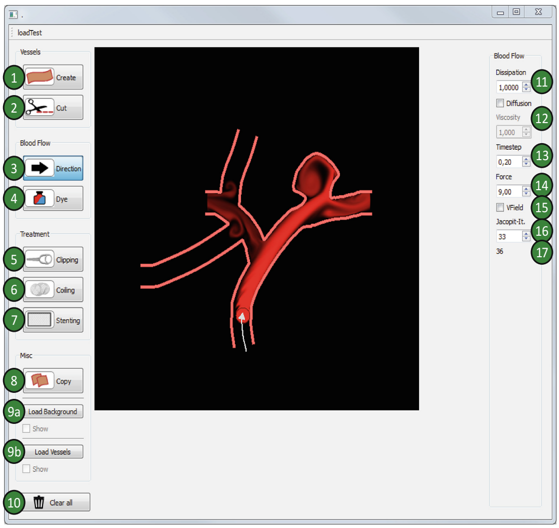


Fig. 12. Screenshot of the prototype. Buttons (1-2) are used to create and edit vessels, (3-4) are used to manipulate and visualize blood flow, (5-7) are used to sketch treatment options, (8) is used to select the *copy tool*, (9a-9b) are used to load a background image to assist the sketching or to load vessel structures out of a monochrome image and (10) is used to reset the whole canvas. The buttons on the right side (11-17) are used to influence the fluid simulation in different ways, e.g., to enable diffusion, change the viscosity, or show the arrow plot visualization.

5.5 User Interface

The structure of the user interface results from two considerations:

- The grid used for the fluid simulation is quadratic and
- a direct input device is used.

The first consideration implies that the canvas, in which the user sketches, is also quadratic. Due to the horizontal format of current displays this means that there is potential free space on the sides of the canvas. This space is used for the menus. In detail, the space is divided in a left and a right region and used for a semantic differentiation of the functionality. On the left side, menus and buttons are placed, which are used to create and manipulate vessels, to sketch treatment options, and to visualize and control blood flow. On the right side, there are menus to control simulation parameters such as the number of iterations for solving or to activate the diffusion process. The second consideration (using a direct input device) leads to the following requirements with respect to interaction. To achieve an easy interaction, all buttons have a bigger size and there are no sub-menus. Furthermore, classical WIMP input elements like spin boxes were omitted since exact inputs would be hard to enter with a pen or touch. In Fig. 12, a screenshot of the user interface is shown.

6 Evaluation

The evaluation is divided into two parts: in the first part, we used *qualitative* and *quantitative* methods for assessing the usability of the prototype. In the second part, we interviewed a physician to compare the procedure of typical patient education (with hand-drawn sketches) with the prototype and investigate advantages and disadvantages.

6.1 Usability

The *qualitative* part of the study was performed with the *think aloud protocol*, where the participants comment their activities while solving a problem. This is helpful to obtain insights into the misunderstandings of the participants as well as to understand how the participants predict the behavior of the prototype. The *quantitative* part of the evaluation was conducted with a questionnaire. The questions are modeled after the questionnaire for ergonomic principles from ISO 9241-110 (suitability for learning, suitability for the task, self-descriptiveness, conformity with user expectations, controllability, and error tolerance) [39]. The single questions were categorized in the different principles and rated with a 7-point Likert scale.

The evaluation started with a short introduction of the prototype on a SMARTBoard, a 70″ screen which allows pen interaction. All features were demonstrated and we asked to think-aloud and noted the spoken comments of the participants. Initially, the participants were asked to perform several tasks that were handed out in written form. For example, they should draw a vessel with a trifurcation. Then, they should use the *cut tool* to change the vessel to a bifurcation. Furthermore, they were asked to draw an aneurysm and treat it with the *clipping*, *coiling*, and *stenting tool*. Finally, they should create and visualize the blood flow in a specific direction. Afterwards, the participants were asked to fill out the questionnaire.

The evaluation was conducted with 14 researchers with medical visualization knowledge. We had three female and eleven male participants, aged from 25 to 44 with an average of 31 years. The participants are experienced computer users with an experience of 20 years on average (minimum: 14, maximum: 30). Ten participants were experienced with pen interaction. It took about 15 to 20 min to fill out the questionnaire. The statements of the participants are denoted with [P#].

Think-Aloud Method. Mostly, the participants were satisfied with the prototype, e.g., one stated that "it is possible to create vessels and aneurysms according to my own ideas" [P13]. Regarding the different tools, the majority of the participants had no problems using them. For example, it was stated that the *cut tool* "is more precise than conventional eraser tools" [P6]. Unfortunately, the *cut tool* leads to misunderstandings during the first use. This was caused by different expectations of its functionality, e.g., some participants thought that it can be used as a conventional eraser. After some practice, the functionality was understood and all participants rated this tool as positive. Another example of a positively rated tool is the *stenting tool*. Especially the automatic expanding behavior was noted as useful. Some participants highlight the alignment of the tools, as it supports the typical workflow (generate vessel, treatment, and visualize blood flow). In general, they stated that this prototype allows a fast and easy generation of vessels with simulated blood flow.

Questionnaire. The results of the questionnaire were determined by calculating the average of every answer. For this, we assigned the symbols of the 7-Point Likert scale ($---$ to $+++$) to the values from -3 to 3. Figure 13 depicts the average for every category. The category *error tolerance* has the lowest rates. Here, the users were asked if the effort to correct an inadvertently drawn error is significantly high. Mostly, the participants stated that it should be possible to delete drawn objects and the effort is high to manually correct them by redrawing. Thus, we added the possibility to delete drawn objects like stents and coils individually, see Sect. 5.4. However, with an average of 1.68, this category was rated well. In summary, the ratings of all categories were positive and the participants were satisfied with the functional range. For refinement, we used the results of the evaluation and improved the prototype according to the suggestions of the participants.

6.2 Structured Interview

We performed a structured interview of about 20 min with a physician, which has 12 years of experience in the field of vascular diseases and patient education.

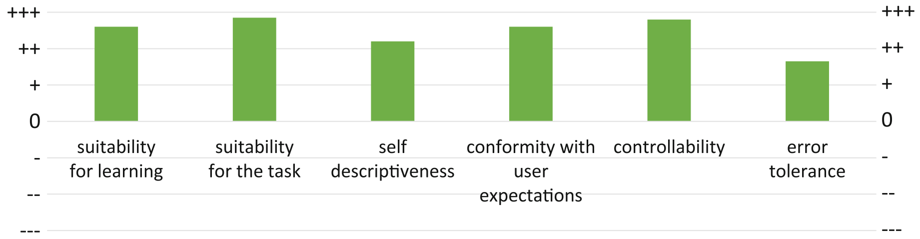
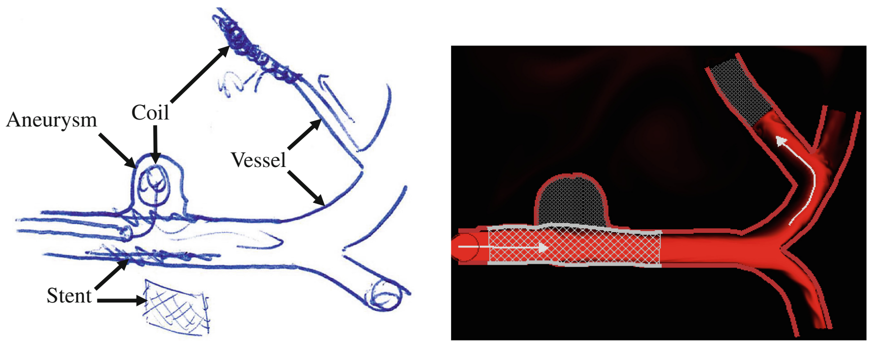


Fig. 13. The average results of the questionnaire for each category of the usability principles is shown.

The audio of the interview was recorded and analyzed after the interview. First, we asked the physician to explain typical patient education with an example and outline the typical education procedure. During typical patient education, the patient receives textual and image-based templates with respect to his disease. Additionally, the physician explains the intervention verbally and supports this with hand-drawn sketches (see Fig. 14). These can help to illustrate specific medical cases and answer individual questions from the patient. The description of the physician revealed the following disadvantages of the *hand-drawn* sketches:

- To correct errors, the physician draws over the existing sketch, which results in a cluttered image.
- The blood flow and the implications of a treatment are usually not drawn, because it is hard to illustrate these and could lead to a confusing image.
- The physician only uses one color to draw all structures, which makes it difficult to distinguish between different elements such as vessels and a clip.
- Some patients have problems to understand the sketch, because they find it difficult to imagine diseases and treatment options.



(a) (b)

Fig. 14. On the left side is the hand sketch of a physician with additional labels. On the right side are the same structures made with the application, which the physician assessed clearer and more understandable.

After that, we explained the prototype and asked to perform the patient education again using our application. Thereby, we let the physician compare his hand-drawn images with sketches from the prototype which showed the following:

- By using different colors, symmetrical, consistent, and uniform structures (e.g., for the vessels or stents), the result is more clear, descriptive and plastic than the hand sketching, and thereby, the perception is supported.
- Due to the illustration of blood flow, the vessel structures, the effects of different intervention methods and possible complications are more understandable and imaginable.

Furthermore, the physician stated additional advantages of the prototype, e.g., due to the clearer resulting images, other persons, who were not involved in the sketching process, are able to understand the sketch. This is difficult with hand sketches, because of their cluttering nature. The physician rated the tool as easy to learn and use, which matches the results of our usability evaluation.

7 Conclusions and Future Work

This work provides methods to sketch vessels, vascular diseases, and treatment options. It is possible to interactively create and manipulate blood flow, which adapts in real-time depending on the sketch.

The usability of the prototype was assessed with *qualitative* and *quantitative* methods. The positive feedback of the evaluation indicates that the proposed concept and prototype are suitable for sketching vascular structures and treatment options. The *structured interview* with the physician revealed further benefits compared to hand-drawn sketches and confirmed the idea to improve patient education and intelligibility by integrating animated blood flow.

A limitation of this work is the representation of the vessels and blood flow in 2D. The representation of vessels and fluids in 3D is a challenge with respect to visualization and interpretation [40]. However, advantages of the 2D representation are its easier intelligibility and lower computational effort, which allows the calculation of a more detailed fluid simulation in real-time.

An improvement of our system would be to automatically differentiate the sketched vascular structures semantically, i.e., distinguish between healthy vessels, stenoses, and aneurysms. With this differentiation, placing a stent in narrowed vessels could be implemented more easily. The sketched stent could inflate a stenosis only, without affecting the surrounding, healthy vessels. By distinguishing between vessels and aneuryms, an entering of the stent into the aneurysm could be prevented as well. Beneath this, we want to investigate how the described concept and prototype can be used for collaboration between physicians. This collaboration can happen at different places where each medical doctor interacts with an instance of the application. The different instances could be connected and mirror the input from one place to another. Additionally, the communication could be supported with voice chat and webcams. Another way of collaboration is possible with only one application, where several physicians are sketching at the same time in front of a big screen and discuss a medical case.

Further analysis could investigate the possibility to realize the described concept in 3D. This leads to challenges and questions regarding a real-time 3D fluid simulation and visualization as well as creating the vessels. For 3D interaction, there are input and output devices which are more suited than pen or touch interaction. By lifting the application in the third dimension, the behavior of blood in vessels could be simulated more accurately, and thus, make the application more relevant for scenarios like operation planning and training.

Acknowledgements. This work was partially funded by the German Federal Ministry of Education and Research (BMBF) within the research campus STIMULATE under grant number '13GW0095A'.

References

1. Mendis, S., Puska, P., Norrving, B.: Global Atlas on Cardiovascular Disease Prevention and Control. Nonserial Publications Series. World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization (2011)
2. European Heart Network: European cardiovascular disease statistics (2012). http://www.ehnheart.org/cvd-statistics.html (last visited: 19 August 2015)
3. Keulers, B.: Computer-based Patient Education: Its Potential in General and Plastic Surgery. Ph.D. thesis, University Nijmegen (2008)
4. Lidal, E.M.: Sketch-based Storytelling for Cognitive Problem Solving. Ph.D. thesis, Department of Informatics, University of Bergen, Norway (2013)
5. Pihuit, A., Cani, M.P., Palombi, O.: Sketch-based modeling of vascular systems: a first step towards interactive teaching of anatomy. In: Proceedings of the Sketch-Based Interfaces and Modeling Symposium, pp. 151–158 (2010)
6. Saalfeld, P., Baer, A., Preim, U., Preim, B., Lawonn, K.: Sketching 2D vessels and vascular diseases with integrated blood flow. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP), pp. 379–390 (2015)
7. Zhu, B., Iwata, M., Haraguchi, R., Ashihara, T., Umetani, N., Igarashi, T., Nakazawa, K.: Sketch-based dynamic illustration of fluid systems. In: Proceedings of the SIGGRAPH Asia Conference, pp. 134:1–134:8. ACM (2011)
8. Kutikhin, A., Brusina, E., Yuzhalin, A.E.: A hypothesis of virus-driven atherosclerosis. Viruses and Atherosclerosis. SpringerBriefs in Immunology, vol. 4, pp. 1–3. Springer, New York (2013)
9. Bederson, J.B., Connolly, E.S., Batjer, H.H., Dacey, R.G., Dion, J.E., Diringer, M.N., Duldner, J.E., Harbaugh, R.E., Patel, A.B., Rosenwasser, R.H.: Guidelines for the management of aneurysmal subarachnoid hemorrhage. Stroke **40**, 994–1025 (1994)
10. Neugebauer, M., Diehl, V., Skalej, M., Preim, B.: Geometric reconstruction of the ostium of cerebral aneurysms. In: Proceedings of VMV 2010-Vision, Modeling, Visualization. pp. 307–314 (2010)
11. Gasteiger, R.: Visual Exploration of Cardiovascular Hemodynamics. Ph.D. thesis, Otto-von-Guericke University Magdeburg (2014)

12. Teitelbaum, G.P., Higashida, R.T., Halbach, V.V., Larsen, D.W., McDougall, C.G., Dowd, C.F., Hieshima, G.B.: Flow-directed use of electrolytically detachable platinum embolization coils. J. Vasc. Interv. Radiol. **5**, 453–456 (1994)
13. Wong, G.K., Tan, H.B., Kwan, M.C., Ng, R.Y., Yu, S.C., Zhu, X.L., Poon, W.S.: Evolution of intracranial aneurysm treatment: from Hunterian ligation to the flow diverter. Surg. Pract. **15**, 16–20 (2011)
14. Bridson, R.: Fluid Simulation for Computer Graphics. A K Peters/CRC Press, Boca Raton (2008)
15. Ciarlet, P., Glowinski, R., Lions, J.: Numerical Methods for Non-Newtonian Fluids: Special Volume. Handbook of Numerical Analysis. Elsevier, New York (2011)
16. Ferziger, J.H., Perić, M.: Compressible flow. Computational Methods for Fluid Dynamics, pp. 309–328. Springer, Heidelberg (2002)
17. Müller, M., Schirm, S., Teschner, M.: Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. Technol. Health Care **12**, 25–31 (2004)
18. Qin, J., Pang, W.M., Nguyen, B.P., Ni, D., Chui, C.K.: Particle-based simulation of blood flow and vessel wall interactions in virtual surgery. In: Proceedings of the Symposium on Information and Communication Technology, pp. 128–133 (2010)
19. Hansen, C., Johnson, C.: The Visualization Handbook. Butterworth-Heinemann, Waltham (2005). Referex Engineering
20. Weiskopf, D.: Vector field visualization. GPU-Based Interactive Visualization Techniques, pp. 81–159. Springer, Heidelberg (2007)
21. Boring, E., Pang, A.: Directional flow visualization of vector fields. In: Proceedings of the Conference on Visualization, pp. 389–392 (1996)
22. Jorge, J., Samavati, F.: Sketch-based Interfaces and Modeling. Springer, London (2011)
23. Sutherland, I.E.: Sketchpad, A Man-Machine Graphical Communication System. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York (1963)
24. Preim, B., Dachselt, R.: Interaktive Systeme: User Interface Engineering, 3D-Interaktion, Natural User Interfaces, vol. 2. Springer, Heidelberg (2015)
25. van Dam, A.: Post-WIMP user interfaces. Commun. ACM **40**, 63–67 (1997)
26. Xu, X., Liu, W., Jin, X., A, Z.S.: Sketch-based user interface for creative tasks. In: Proceedings of Asia Pacific Conference on Computer Human Interaction, pp. 560–570 (2002)
27. Naya, F., Contero, M., Aleixos, N., Company, P.: ParSketch: a sketch-based interface for a 2D parametric geometry editor. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4551, pp. 115–124. Springer, Heidelberg (2007)
28. Igarashi, T., Matsuoka, S., Kawachiya, S., Tanaka, H.: Interactive beautification: a technique for rapid geometric design. In: Proceedings of ACM Symposium on User Interface Software and Technology, pp. 105–114 (1997)
29. Herold, J., Stahovich, T.F.: The $1^{¢}$; recognizer: a fast, accurate, and easy-to-implement handwritten gesture recognition technique. In: Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling. SBIM 2012, Eurographics Association, pp. 39–46 (2012)
30. Lawonn, K., Saalfeld, P., Preim, B.: Illustrative visualization of endoscopic views. In: Bildverarbeitung für die Medizin (BVM) pp. 276–281 (2014)
31. Harris, M.J.: GPU GEMS Chapter 38, Fast Fluid Dynamics Simulation on the GPU. Pearson Higher Education, New York (2004)
32. Stam, J.: Stable fluids. In: Proceedings of the Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), pp. 121–128 (1999)

33. Butcher, J.C.: A history of Runge-Kutta methods. Appl. Numer. Math. **20**, 247–260 (1996)
34. Krüger, J., Westermann, R.: Linear algebra operators for GPU implementation of numerical algorithms. ACM Trans. Graph. **22**, 908–916 (2003)
35. Wu, E., Liu, Y., Liu, X.: An improved study of real-time fluid simulation on GPU: research articles. Comput. Animat. Virtual Worlds **15**, 139–146 (2004)
36. Glaßer, S., Lawonn, K., Preim, B.: Visualization of 3D cluster results for medical tomographic image data. In: Proceedings of Conference on Computer Graphics Theory and Applications (VISIGRAPP/GRAPP), pp. 169–176 (2014)
37. Taubin, G.: Curve and surface smoothing without shrinkage. In: Proceedings of the International Conference on Computer Vision, pp. 852–857. IEEE Computer Society (1995)
38. Heckel, F., Moltz, J.H., Tietjen, C., Hahn, H.K.: Sketch-based editing tools for tumour segmentation in 3D medical images. Comput. Graph. Forum **32**, 144–157 (2013)
39. Prmper, J.: Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse zur Reliabilität und Validität. In: Software-Ergonomie: Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung, pp. 254–262 (1997)
40. Lawonn, K., Gasteiger, R., Preim, B.: Adaptive surface visualization of vessels with animated blood flow. Comput. Graph. Forum **33**(8), 16–27 (2014)

Springer

http://www.springer.com/978-3-319-29970-9

Computer Vision, Imaging and Computer Graphics Theory and Applications
10th International Joint Conference, VISIGRAPP 2015, Berlin, Germany, March 11-14, 2015, Revised Selected Papers

Braz, J.; Pettré, J.; Richard, P.; Kerren, A.; Linsen, L.; Battiato, S.; Imai, F. (Eds.)

2016, XXIII, 474 p. 255 illus. in color., Softcover
ISBN: 978-3-319-29970-9