Eurographics Workshop on Visual Computing for Biology and Medicine (2021)
K. Nieselt and T. Schultz (Editors)

Short Paper

Automatic Animations to Analyze Blood Flow Data

V. Apilla[1], B. Behrendt[1], K. Lawonn[2], B. Preim[1], M. Meuschke[1,2]

[1]Department of Simulation and Graphics, University of Magdeburg, Germany
[2]Institute of Computer Science, University of Jena, Germany

Abstract

We present an approach for computing camera animations composed of optimal views to support the visual exploration of blood flow data using cerebral aneurysms as major example. Medical researchers are interested in hemodynamic parameters and vessel wall characteristics. The time-dependent character of blood flow data complicates the visual analysis.
Our approach is modeled as an optimization problem to automatically determine camera paths during the cardiac cycle. These consist of optimal viewpoints showing regions with suspicious characteristics of wall- and flow-related parameters. This provides medical researchers with an efficient method of obtaining a fast overview of patient-specific blood flow data.

CCS Concepts
• ***Applied computing*** → ***Life and medical sciences;*** • ***Human-centered computing*** → ***Visualization;***

1. Introduction

Blood flow data is used to study variants of normal and pathological flow in abdominal, cerebral, and coronary arteries. The exploration of time-dependent 3D flow data is challenging. Observing the temporal changes from one viewpoint, even if carefully chosen, can only provide a small portion of the complex spatial domain. Changing the viewpoint to better understand different parts of the relevant vascular structure, however, leads to missing temporal changes. In this paper, we provide an optimization-based computation of animations related to blood flow and vessel wall characteristics by employing cerebral aneurysms as major example.

Hemodynamic attributes can be obtained by Computational Fluid Dynamic (CFD) simulations that model the patient-specific blood flow behavior or by blood-flow measurements with special ultrasound or MRI protocols. Besides these attributes the interrelation between wall-related parameters of a pathology and its parent vessel and the underlying blood flow is essential for researchers and physicians. A well-selected camera perspective facilitates the exploration by giving an overview about the data. There are methods to compute optimal views on aneurysms represented by a 3D model [MEB*17]. However, these methods only consider hemodynamic attributes defined on the vessel wall, ignoring the blood flow.

Therefore, we propose an automatic camera path with optimal viewpoints on 3D vessel models showing interesting regions during the cardiac cycle. We evaluated our technique with three neuroradiologists that are familiar with aneurysm data. Moreover, we compared manually selected viewpoints by the neuroradiologists to our automatically computed viewpoints. Our method produces similar results to manual selections, demonstrating its suitability to support the analysis of medical blood flow data.

2. Related Work

The task of finding appropriate viewpoints on a scene within a 3D space and connecting them to form a meaningful camera path is common in many applications. Preim and Meuschke [PM20] provide an overview about animation generation and evaluation for medical data. Since our approach considers surface-based representations and time-dependent flow data, we discuss related work w.r.t. animation approaches optimized for these data types.

Vázquez et al. [VFSH01] determine view-space entropy for a set of potential viewpoints around *polygonal objects* to select optimal viewing angles. Mühler et al. [MNTP07] select optimal viewpoints by calculating the size of visible surface area, stability of the view, preferred viewing angle as well as the importance and number of both, occluding and occluded objects. Feixas et al. [FSG09] model the problem of optimal viewpoint selection as an information channel between a set of viewpoints and polygons. Additionally, they explore strategies for ordering such viewpoints on a camera path, creating either *guided* (minimal difference between adjacent viewpoints) or *exploratory* (maximal difference) tours. Ma et al. [MTW*19] use the entropy of a *flow field* to identify critical regions in each time step that are used for the camera animation.

Lawonn et al. [LMPH19] compute viewpoints on 3D surface representation of automatically detected brain aneurysms. The generated views show each aneurysm from four sides, which are integrated into a patient report. Meuschke et al. [MVB*17, MEB*17] introduce an interactive planning of camera animations summarizing simulated blood flow information for cerebral aneurysms w.r.t. hemodynamic parameters defined on the vessel wall. Based on user-selected thresholds of these parameters, views on the aneurysm are selected over time that fulfill the threshold criterion.

DOI: 10.2312/vcbm.20211349

3. Data Acquisition and Pre-processing

First, clinical image data, e.g., computed tomography angiography (CTA) of the patient-specific vasculature are acquired. From these images, a 3D surface model of the vasculature is reconstructed following the pipeline by Mönch et al. [MNP11]. Then, the 3D vessel surface is transformed to an unstructured volumetric grid as input for the simulation. Patient-specific hemodynamics are calculated numerically using CFD, solving the Navier-Stokes equation (see [MVB*17] for details). As a result, we receive data with a high temporal resolution comprising 93 time steps ($\Delta t = 0.01s$). The cell-based variables are mapped to the vertices of the triangular vessel surface. This results in different scalar fields defined on the vessel surface. Regarding the patient-specific hemodynamics, we enable the exploration of pressure and wall shear stress (WSS), which are important quantities to describe the flow field. Moreover, path lines have been seeded on the centers of the ostium triangles (the separating surface between aneurysm and parent vessel) and were traced in forward and backward direction [OLK*14]. Every 0.01 s 100 path lines are emitted using an adaptive fifth order Runge-Kutta method. For each vertex of a path line, three scalar fields are defined: pressure, flow velocity and time of occurrence in the cardiac cycle.

4. Automatic Animations for Blood Flow Data

The automatic camera animation is based on optimal viewpoints that are used to generate an animation path.

4.1. Initialization and Pre-processing

Based on a selected data set, scalar fields of interest for the surface $(s_1, s_2, \dots, s_l)$ and path lines $(p_1, p_2, \dots, p_k)$ are manually chosen. Each scalar field is normalized to the range $[0, 1]$. The selected fields are used in the target function to find optimal views. Similar to Meuschke et al. [MGB*18], we reduce the number of time steps and consider every 4th time step for upcoming computations.

Path lines. Each path line is made up of densely sampled points. We used the Ramer-Douglas-Peucker algorithm [DP73] to create a simplified version of the original path lines. The algorithm creates a curve with a lower number of points for each line based on a user-defined parameter ε by preserving curve topology. We set $\varepsilon = 0.03$. Lower values do not significantly simplify the original path line and larger values result in massive topological changes.

Ellipsoid Fitting. A surface around the vessel model is required to sample viewpoints that form the camera path. For this purpose, we use an ellipsoid and employed the method by Meuschke at al. [MEB*17]. By moving the camera on the ellipsoid surface, the distance to the vessel remains approximately the same.

4.2. Optimal Viewpoint Selection

The goal is to form a camera path with a low number of unique viewpoints to provide a compact overview about the data. Similar to Meuschke et al. [MEB*17], these optimal viewpoints are determined using a target function. However, unlike in their method, our target function takes into account the behavior of internal blood flow. Thus, regions of interest are determined w.r.t. both structures, the vessel wall and the blood flow.

4.2.1. Target Function

Each point sampled on the ellipsoid surface contains a position $p = (x, y, z, time)$ used as a viewpoint position and the view direction as the inverted vertex normal. To assess a viewpoint, a target function is used that takes both the vessel wall (comprising parent vessel and aneurysm part) and the path lines into account:

$$f(p) = \sum_i t_s(p_{is}) + \sum_i t_p(p_{ip}). \tag{1}$$

A viewpoint scores higher if it centrally displays i) suspicious surface regions, i.e., high values of t_s regarding the user-selected scalar fields defined on the vessel wall and ii) path lines with suspicious scalar values, i.e., high values of t_p. For a viewpoint p, two images are rendered. One for the vessel surface and one for the path lines. The vessel surface image pixels (p_{is}) and the path lines image pixels (p_{ip}) are assigned a pixel value which is computed by their respective sub-functions $t_s(p_{is})$ and $t_p(p_{ip})$. The pixel value determined by the sub-functions depends on how interesting the surface area is and how centered the fragment is within the camera view:

$$t_s(p_{is}) = \sum_{j=1}^{l} {s_j}^2 \cdot B_{is}. \tag{2}$$

The normalized scalar fields $(s_1, s_2, \dots, s_l)$ belonging to each fragment of the vessel surface image (p_{is}) are squared. This results in strong values for the scalar fields which exhibits higher/lower values. Next, the squared scalar fields $({s_1}^2, {s_2}^2, \dots, {s_l}^2)$ are each multiplied by a binomial filter value of the fragment (B_{is}) to strongly boost the pixel values in the center of the view. The binomial filter is a constant factor $B_{ij} = \binom{n-1}{i}\binom{n-1}{j}$ assigned to a fragment of an image that is divided into $n \times n$ sub images. We used $n = 5$ resulting in 25 sub images as suggested in [MEB*17]. Finally, the pixel values from the vessel surface image are summed in the target function. Note that we assume that regions with high scalar values are of interest. If the user indicates that low values are interesting, we invert the original scalar field values with $(1 - s_1)^2$.

Implementation. We render each of the two images at a fixed resolution of 576 $\times$ 324 pixels (16 : 9) to improve the performance. The vertex data containing the scalar fields of the vessel surface and path lines are uploaded together with their mesh attributes to the GPU. Based on the current temporal value of the animation, the vertex data is changed. OpenGL shaders are used to write the scalar field values into the (r, g, b, a) channels of the rendered images. First, the normalized scalar field values are written into the r and g channel, whereas the normalized binomial filter value is assigned to the b channel. The a channel encodes the binary values that determine if a fragment contains a front face. Later, a *compute shader* reads the normalized data from the *rgba* textures to calculate the target function values.

4.2.2. Viewpoint Clustering

Since considering all points on the ellipsoid would lead to unnecessarily long camera paths, we reduced the number of potential best views. Similar to Meuschke et al. [MEB*17], we first compute the target function value for each point on the ellipsoid. From this scalar field, we calculate the viewpoints from the upper 90 % quantile and use them as input for a clustering algorithm. We used the

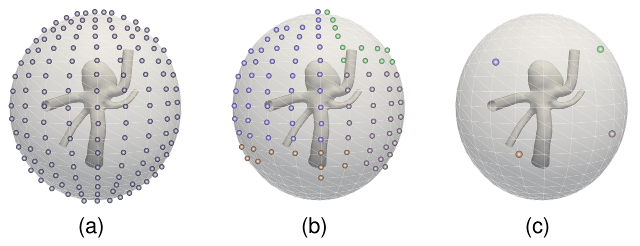

Figure 1: *Viewpoint selection: sampled viewpoints on the ellipsoid (a), clusters from the upper 90 % quantile viewpoints (b), and center viewpoint of each cluster (c).*

agglomerative hierarchical clustering (AHC) with complete linkage where the similarity between two clusters is determined by the Euclidean distance between the 3D viewpoint positions. Once the clusters are formed (Fig. 1(b)), the center of the clusters are further optimized, while the other points are discarded (Fig. 1(c)).

4.2.3. Viewpoint Optimization

Next, we optimize the cluster midpoints. A viewpoint x_{old} is iteratively changed for its direction $(pitch, yaw)$ and position (x, y, z) by applying the gradient ascent approach according to the target function (Eq. 1) . The viewpoint x_{old} is moved in the direction with a step size s. This results in an updated viewpoint x_{new}. Next, the new viewpoint x_{new} is evaluated for the target function value $f(x)_{new}$. Moreover, $f(x)$ is evaluated for different values of s, where the values range from 0.01 to 0.05 with an offset of 0.01 [MEB*17]. The algorithm continues until $f(x)_{new} - f(x)_{old} > t$ with $t = 0.001$. In case $f(x)_{new} - f(x)_{old} \leq t$, x_{old} is saved and the process terminates.

4.3. Camera Path Generation

After determining the optimal viewpoints for each time step, the camera path is generated in two steps. First, the viewpoints are connected to produce a local camera path for each time step. Second, all the local camera paths from subsequent time steps are chained together to create a global camera path. In the following, these steps are described in greater detail.

4.3.1. Local Camera Path Generation

With the viewpoints of a time step, a local camera path is produced and considered as a Traveling Salesman Problem (TSP). The goal is to find a route from the 1^{st} viewpoint to the n^{th} viewpoint with the shortest tour length in terms of Euclidean distance, see Fig. 2.

We used an evolutionary algorithm to solve the TSP problem. It starts with a set of random elements within a search space Ω and iteratively refines this set to find elements optimizing a *fitness function* $f : \Omega \rightarrow \mathbb{R}$. Each iteration ("*generation*") consists of a list of n candidates ("*individuals*"), sorted by their assigned *fitness* function value. In our method, an individual whose fitness is determined by the route distance is represented by an array of viewpoints. The ordered crossover and reverse sequence mutation are used as the crossover-mutation operators. The roulette wheel method [LL12] is used as the selection operator. The maximum number of iterations = 100 was used as the stopping criteria for the algorithm.

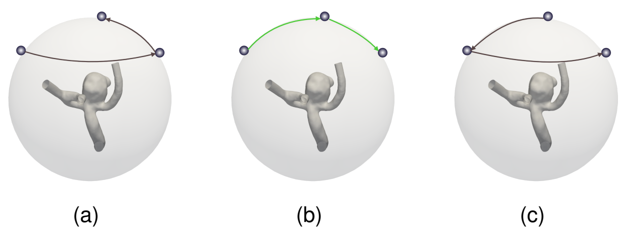

Figure 2: *Three possibles local paths between three local viewpoints. The path in (b) is chosen due to the smaller path distance.*

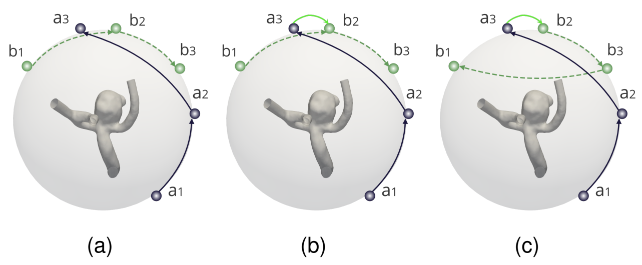

Figure 3: *Global path generation by connecting two subsequent local paths a and b. The last viewpoint of path a (a_3) is connected to the closest viewpoint in path b (b_2).*

4.3.2. Global Path Generation

The global camera path used for the final animation is created by concatenating all local camera paths. The process connects the closest viewpoints between the local paths of the current time step and the subsequent time step. The linking process of the local paths is demonstrated in Fig. 3. In this example, the current path a is connected to the subsequent path b (Fig. 3(a)). The connection is made by linking the endpoint a_3 of the current path a to the closest point b_2 from the subsequent path b. After connecting (Fig. 3(b)), the subsequent path b is rearranged if necessary. In this case, a rearrangement is required because b_2 is now the starting point of b instead of b_1. Therefore, the path is rearranged to accommodate b_1 by linking it using b_3 and making it the endpoint of b (Fig. 3(c)).

4.4. Animation Generation

The camera travels on the global camera path to produce the animations. For our animations, the parameters – camera flight and camera speed are crucial.

Camera Flight. The trajectory taken by the camera to move its position from the current viewpoint V_{source} to the next subsequent viewpoint V_{target} is defined as the *camera flight*. The simplest of all camera flights would be traveling between V_{source} and V_{target} in a straight line. Although computationally inexpensive, an undesirable effect is produced by this approach. When the straight line produced between V_{source} and V_{target} pierces the aneurysm, the camera taking this flight penetrates through the aneurysm.

To avoid this problem, the camera must maintain sufficient distance from the aneurysm surface. This can be achieved by making the camera take a flight on the surface of the ellipsoid which surrounds the vessel at a fixed distance—orbiting around the aneurysm. Such a smooth collision-free flight between the nodes (V_{source}, V_{target}) is given by the *Geodesic Path* which is computed according to Sharp and Crane [SC20].

Camera Speed. The speed at which the camera moves on the path is defined as the *camera speed*. The camera must move at a consistent speed to generate suitable animations. Therefore, we assigned the camera a maximum speed at which it moves at a steady pace between V_{source} and V_{target}. Furthermore, to smoothly deviate from one position and smoothly land at another position, the *Smoothstep* function is used:

$$t \leftarrow t^2 \cdot (3 - 2t). \tag{3}$$

This function accelerates and decelerates the camera at the beginning and the end of the camera flight while maintaining a consistent speed during the rest of the flight [MEB*17].

5. Evaluation and Results

To assess our camera animations, we conducted a two-part evaluation with three neuroradiologists. First, we conducted an informal evaluation based on a questionnaire to assess specific aspects of our approach. Second, the experts were asked to manually select optimal viewpoints over the cardiac cycle. We compared these viewpoints with our automatically calculated viewpoints regarding deviations in camera position.

5.1. Informal Expert Interviews

The informal expert interviews are conducted as follows:

1. Introducing the animation tool and explaining its features.
2. Exploration of different data sets, where we indicated that we used the think-aloud technique and note the spoken comments.
3. A questionnaire that inquires the quality of our animations as well as the usefulness of the exploration methods.

The experts confirmed that the animations provide an overview about wall- and flow-related aspects. They appreciated that the camera follows the flow direction and that the flow is paused in case additional views of prominent wall regions are shown. In combination with the temporal progress bar, physicians have always been able to keep track of which time period of the cardiac cycle the animation is currently in. However, they noted that the camera animations could be further optimized w.r.t. flow patterns that occur. Areas with laminar flow could be passed more quickly.

Moreover, they emphasized the potential of our animations to support the exploration and navigation within the time-dependent 3D data. While manual explorations are very time-consuming, the experts can focus on the displayed quantities during the animation. However, they noted that the animations could be supported by more guidance to point the user more directly to regions of interest.

In general, the animations help to identify rupture-prone surface regions. However, animation alone is not sufficient to assess the aneurysm. This always requires a manual, detailed analysis as well. In this context, the provided interaction window was highly appreciated. Here, the experts may change the displayed scalar fields as well as to adjust the opacity of the vessel in order to focus more on wall-related aspects.

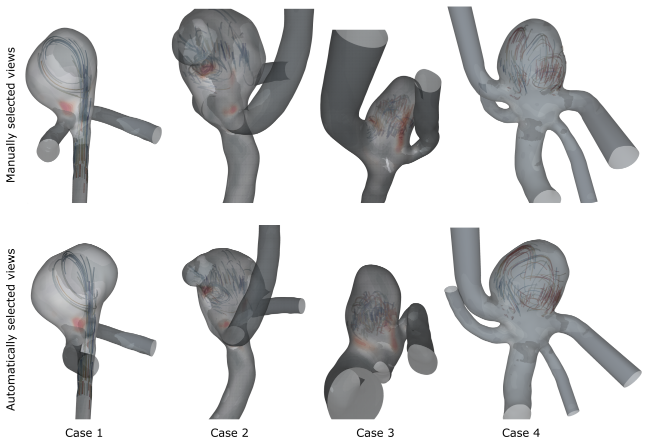

Figure 4: *Exemplary viewpoints for four data sets (Case 1-4). The upper row shows manually selected views, while the lower row shows the most similar views from the automatic animation. In all examples, WSS is color-coded and pressure is depicted by hatching.*

5.2. Comparing Manual and Automatic Results

To verify that the automatically calculated views show interesting regions of wall- and flow-related scalar fields, we compared our results with manually selected views. The neuroradiologists were already familiar with the data sets. This reduces the risk of overlooking interesting views during manual selection.

The experts had to manually search for suspicious surface regions depending on the selected scalar fields s_1 and s_2. For each manually selected viewpoint, we store the corresponding time step and camera position. Moreover, we determine the closest automatically calculated viewpoint w.r.t. the most similar camera position and time step and store the deviations in the two camera angles, the distance of the camera and the corresponding time steps.

Figure 4 shows example comparisons between manually selected views and the most similar automatically calculated views for four data sets (Case 1 to case 4). The top row shows the manual views, while the automatic results are shown below. On the mesh, the WSS is color-coded, and the pressure is depicted by hatching. Reddish, dense hatched areas indicate suspicious surface regions. On the path lines, the flow velocity is encoded, where flow with a high velocity is of particular interest.

Basically, the manual views and our determined views are very similar. The deviations of the camera angles are 15 to 57 degrees for θ and 10 to 45 degrees for φ. In addition, only very small temporal deviations of maximum 1.5 time steps occur. Regarding the zoom level, the experts chose views where the camera distance to the mesh is 10 to 25 % smaller than in the automatic views.

In principle, neuroradiologists prioritized views in which the aneurysm is seen from the front. This resulted in smaller angular deviations (Case 1 and Case 4). The larger angular deviations resulted for Case 3 in which vessel segments led to occlusions of the aneurysm in many views. The automatic calculation chooses views where the high WSS values at the aneurysm junction are well visible and vessel areas are not obscured. In contrast, in the manual selection, the outgoing vessels partially obscure the aneurysm.

More zoomed views were manually selected mainly for Case 2. This aneurysm has an additional bulge, which the neuroradiologists were particularly interested in and therefore selected closer views.

6. Discussion

Our computation of optimal viewpoints leads to animations over the cardiac cycle where both interesting wall and flow regions are shown. The computation time of a whole animation takes between 2.32 and 6.67 min on a standard desktop computer.

The computation of target function values for the path lines depends on the used seeding technique. We use a standard seeding technique that was employed in [OLK*14]. To reduce the computation time of the animations, we also reduce the spatial resolution of the path lines by using subsampling. To reduce the risk that path line points with high scalar values are deleted, we set a very small epsilon value to resample the path lines.

We observed that neuroradiologists sometimes had difficulties focusing on wall and flow properties simultaneously during animation. Therefore, they watched the animations several times. To improve this, the user should be guided by giving hints on which structures to pay attention to.

Limitations. In principle, our target function can consider up to l scalar fields defined on the vessel mesh and k scalar fields defined on the path lines. However, we used a common approach where a maximum of two scalar fields can be visualized on the vessel wall, simultaneously. Furthermore, one scalar field on the path lines is color-coded. More advanced methods, such as the *skyline visualization* of Meuschke et al. [MVG*21], could be used to display more scalar fields on the surface. Moreover, in case the same view points are connected over several time steps, this can lead to fluctuating camera animations. To avoid this, a more complex space-time dependent approach for view selection would need to be used.

We use the *L-method* to determine the number of clusters, where the center of each cluster is used as input for the gradient ascent approach. However, the viewpoints found are local optima for each time step. So, globally, it may happen that some viewpoints are not very interesting. However, we decide to keep these views so that the animations start at the beginning of the heart cycle.

Other applications. Our method can easily be applied to other scenarios. Prerequisites are a 3D surface mesh and path lines with scalar values defined for their vertices. In addition to cerebral aneurysms, blood flow data are often used to diagnose diseases of cardiac vessels, e.g., to evaluate aortic aneurysms or stenoses. Our animations could also support data exploration in such scenarios.

7. Conclusion and Future Work

Since the manual analysis of blood flow data is very time-consuming, methods are needed to support and guide the analysis process. Animations have the potential to draw the user's attention to interesting regions in the data, which can then be manually examined in more detail. In this work, we present an automated approach to create animations for the analysis of blood flow data.

We plan to conduct a larger user study to further investigate the usefulness of our animations compared to non-animation-based systems. We also want to perform a parameter study to evaluate more deeply the robustness and suitability of the chosen parameter settings. In addition, we want to integrate vector and tensor fields that represent bio-mechanical stresses of blood flow on the vessel wall into the calculation of the viewpoints. Another point for future work is to use narrative visualization techniques such as labels and annotated diagrams to guide the user's attention and to show additional details about the data.

References

[DP73] Douglas D. H., Peucker T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica 10*, 2 (1973), 112–22. 2

[FSG09] Feixas M., Sbert M., González F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans Appl Percept 6*, 1 (2009), 1–23. 1

[LL12] Lipowski A., Lipowska D.: Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications 391*, 6 (2012), 2193–96. 3

[LMPH19] Lawonn K., Meuschke M., Preim B., Hildebrandt K.: A Geometric Optimization Approach for the Detection and Segmentation of Multiple Aneurysms. *Comput Graph Forum 38(3)* (2019), 413–25. 1

[MEB*17] Meuschke M., Engelke W., Beuing O., Preim B., Lawonn K.: Automatic Viewpoint Selection for Exploration of Time-dependent Cerebral Aneurysm Data. In *BVM* (2017), Springer Verlag, pp. 352–57. 1, 2, 3, 4

[MGB*18] Meuschke M., Günther T., Berg P., Wickenhöfer R., Preim B., Lawonn K.: Visual Analysis of Aneurysm Data using Statistical Graphics. *IEEE Trans Vis Comput Graph 25*, 1 (2018), 997–1007. 2

[MNP11] Mönch T., Neugebauer M., Preim B.: Optimization of Vascular Surface Models for Computational Fluid Dynamics and Rapid Prototyping. In *Int. Workshop on Digital Engineering* (2011), pp. 16–23. 2

[MNTP07] Mühler K., Neugebauer M., Tietjen C., Preim B.: Viewpoint Selection for Intervention Planning. In *Proc. of EuroVis* (2007), pp. 267–74. 1

[MTW*19] Ma J., Tao J., Wang C., Li C., Shene C.-K., Kim S. H.: Moving with the flow: an automatic tour of unsteady flow fields. *Journal of Visualization 22*, 6 (2019), 1125–44. 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), 761–70. 1, 2

[MVG*21] Meuschke M., Voss S., Gaidzik F., Preim B., Lawonn K.: Skyscraper visualization of multiple time-dependent scalar fields on surfaces. *Computers & Graphics 99* (2021), 22–42. 5

[OLK*14] Oeltze S., Lehmann D. J., Kuhn A., Janiga G., Theisel H., Preim B.: Blood Flow Clustering and Applications in Virtual Stenting of Intracranial Aneurysms. *IEEE Trans Vis Comput Graph 20(5)* (2014), 686–701. 2, 5

[PM20] Preim B., Meuschke M.: A Survey of Medical Animations. *Computer & Graphics 90* (2020), 145–68. 1

[SC20] Sharp N., Crane K.: You can find geodesic paths in triangle meshes by just flipping edges. *ACM Transactions on Graphics 39*, 6 (2020), 1–15. 4

[VFSH01] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Viewpoint Selection Using Viewpoint Entropy. In *Proc. of VMV* (2001), pp. 273–80. 1