
INVERSEVIS: REVEALING THE HIDDEN WITH CURVED SPHERE TRACING

Kai Lawonn

Friedrich Schiller University Jena
Jena, Germany
kai.lawonn@uni-jena.de

Monique Meuschke

Otto-von-Guericke University Magdeburg
Magdeburg, Germany
meuschke@isg.cs.uni-magdeburg.de

Tobias Günther

Friedrich-Alexander-Universität Erlangen-Nürnberg
Erlangen, Germany
tobias.guenther@fau.de

ABSTRACT

Exploratory analysis of scalar fields on surface meshes presents significant challenges in identifying and visualizing important regions, particularly on the surface’s backside. Previous visualization methods achieved only a limited visibility of significant features, i.e., regions with high or low scalar values, during interactive exploration. In response to this, we propose a novel technique, *InverseVis*, which leverages curved sphere tracing and uses the otherwise unused space to enhance visibility. Our approach combines direct and indirect rendering, allowing camera rays to wrap around the surface and reveal information from the backside. To achieve this, we formulate an energy term that guides the image synthesis in previously unused space, highlighting the most important regions of the backside. By quantifying the amount of visible important features, we optimize the camera position to maximize the visibility of the scalar field on both the front and backsides. *InverseVis* is benchmarked against state-of-the-art methods and a derived technique, showcasing its effectiveness in revealing essential features and outperforming existing approaches.

1 Introduction

The visualization of scalar fields on three-dimensional surfaces is a critical aspect of data analysis and interpretation in various scientific and engineering domains. In fields like fluid dynamics, material science, and geophysics, understanding the intricate details of scalar fields such as temperature, pressure, or stress distribution is crucial. For example, scalar fields such as Wall Shear Stress (WSS) and pressure, when encoded on a 3D surface, significantly aid in the decision-making process in material science or in treating conditions like aneurysms or vessel stenosis. A visual representation provides an intuitive understanding of critical vessel wall regions, thereby supporting more informed and precise treatment strategies.

Color encoding a scalar field on a 3D surface is a widely used technique in data visualization. However, this approach presents several challenges for users when exploring 3D data, especially in the context of complex surfaces characterized by, e.g., irregular shapes, a high level of detail, uneven curvatures, or overlapping structures. When the user rotates the object, they must remember distinct surface regions. Moreover, certain parts of the surface, such as highly concave regions, can be difficult to see. Consequently, important data color-coded on these obscured areas may be overlooked, even when toggling front and back face rendering, as shown in Fig. 1, or they require extensive model manipulation to become visible, which can also hardly be overcome by animations [PM20].

Previous work applied projections [NGB*09, GSK*12] or parametrizations [MGB*19, EEL*19] to generate occlusion-free 2D visualizations of the encoded scalar fields. However, the resulting 2D map-based depictions are affected by distortion and leading to a misrepresentation of spatial relationships. Moreover, the transition from 3D to 2D inherently results in the loss of depth cues, which complicates the accurate interpretation of 3D structure and spatial relationships.

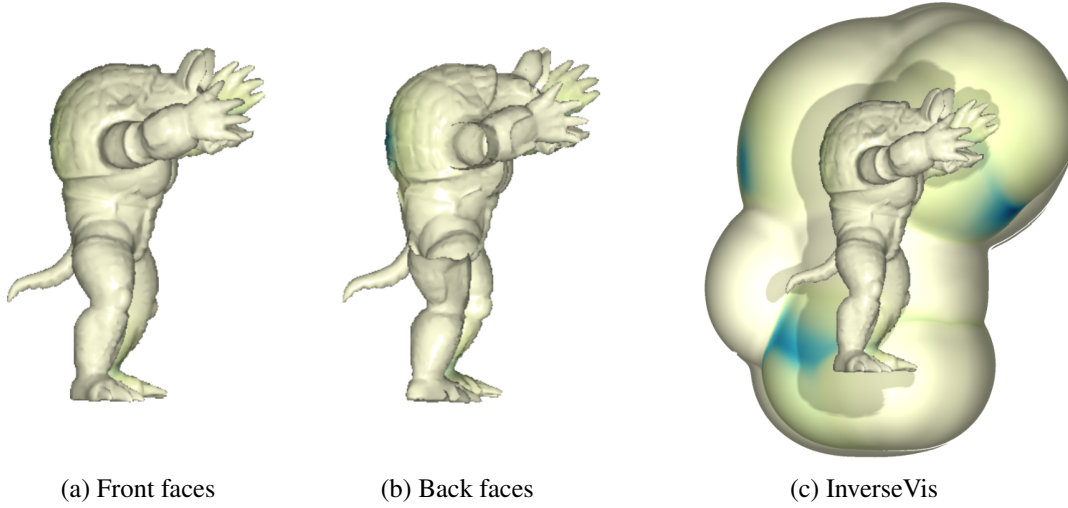


Figure 1: The first two images show the front and back faces. Hot spots cannot be detected while InverseVis detects them.

Further, surfaces with complex topologies, such as those with holes or intricate curves, can be particularly challenging to parametrize. Often, the user has to define appropriate cut lines, where the parametrization can result in overlapping regions in 2D space, which can obscure important details and can be confusing. Another problem arises from the high cognitive load needed to relate the 2D representations to the original 3D structure.

For this reason, we propose *InverseVis*, a novel optimization-based surface visualization method that bends rays around the object to reveal the scalar field on the backside. Pixels that directly see the surface remain unchanged to support normal 3D perception. To keep the visualization of the backside close to the geometry, we define an image space area that is located *behind* the object, from which non-linear rays are released that fall in a gravitational potential towards the surface, thereby landing on the backside. The scalar value found is then encoded at the location where the non-linear ray started. To judge how much information is conveyed, we determine both directly and indirectly visible surface parts and integrate the displayed scalar field. A higher score means that more information is shown. Based on this score, we perform a gradient-based optimization of our method’s parameter and identify an optimal viewpoint that conveys the most information. We incorporate shadowing and surface shading to improve depth perception and compare our approach with a previous projection method and the optimization of a quadratic mirror surface behind the object, see Fig 2. In summary,

- we propose a novel visual encoding technique that displays occluded scalar fields in otherwise empty pixels,
- we present a physically-inspired model that determines a mapping from the image space to the backside of a surface,
- we introduce a quality metric that measures the amount of directly and indirectly seen information,
- we perform an optimization of the viewpoint from which the most information is seen.

2 Related Work

The visualization of scalar fields on 3D surfaces is challenging due to occlusion. Existing methods that augment or alter the spatial arrangement can be sorted into two categories, as summarized next. Later, we compare our approach with a method from each category.

2.1 2D Map-based Visualizations

Map-based visualizations are adept at representing and analyzing spatial data distributions on complex surfaces, negating the need for 3D rotation. They are often used to examine vessel wall pathologies like aneurysms or stenosis, displaying scalar fields such as WSS, thickness, and plaque distributions. 2D maps are created through mesh projection or parameterization. Projection casts the 3D mesh onto shapes like cylinders or spheres, providing rapid computations

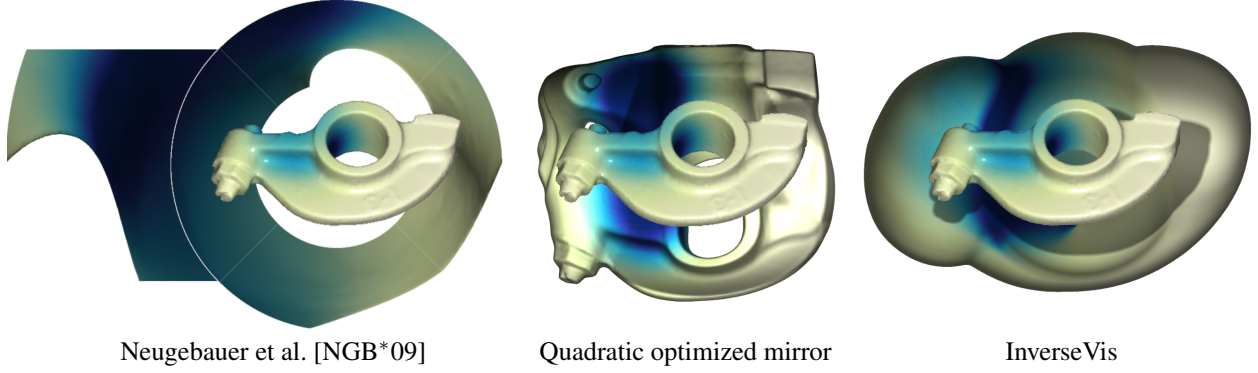


Figure 2: *The indirect visualization of scalar fields on the backsides of the rocker arm is challenging. We propose a novel visualization method that utilizes the empty space around the surfaces to display the hidden parts of the scalar field with less occlusion and distortion.*

but lacking bijective topology. Mesh parameterization allows for a direct bijective mapping between 3D surfaces and 2D maps. Both methods face challenges in handling branching surfaces, preserving features, standardizing layouts, and minimizing distortions.

Mesh Projections. Neugebauer et al. [NGB*09] describe the generation of an overview map for the analysis of scalar data on cerebral aneurysm surfaces. The map is generated using a cube-map-based approach, where an axis-aligned cube is positioned at the center of the aneurysm, and the aneurysm’s surface is projected onto five cube sides. This projection results in various map zones that encapsulate the 3D model, forming a ring. However, the projection introduces non-conformal spherical distortions and cut edges, which can cause orientation changes and confusion during interaction. Although a correlation tool aids in matching points on the aneurysm surface, the presence of projection artifacts and the complexity of the map’s interactivity can hinder the clarity of the data and pose a learning challenge for users. Moreover, it does not prioritize critical surface structures, such as regions with high scalar values. Goubergrits et al. [GSK*12] developed a method for mapping aneurysms to analyze WSS distributions, involving the repositioning of surface vertices towards the center of mass, forming a unit sphere, and employing azimuthal equidistant projection to preserve angular information. This technique is effective for convex structures but may lead to distortions in irregularly shaped aneurysms where the center of mass is external to the surface. In heart disease research, particularly for analyzing left ventricular function, the Cardiac Bull’s Eye Plot (BEP) has been popular [KHB*06, OKG*06, KMP*15]. This abstract 2D representation simplifies complex cardiac data by projecting the myocardium into 17 circularly arranged regions. Despite its usefulness, the BEP is optimized for myocardium grading. Due to this shape dependency, it can hardly be transferred to other 3D models.

Mesh Parametrizations. Various mesh parametrization techniques have been developed, aiding in scalar field analysis. Eulzer et al. [EMML22] presented an extensive overview of mapping vascular structures into 2D, and Kreiser et al. [KMM*18] explored mapping techniques in other medical areas like the brain and colon. In the following, we review vessel parametrization techniques, as they were also employed in the validation of our approach. Meuschke et al. employed Least Squares Conformal Maps (LSCM) [MVB*17a, MVPL18] and Spectral Conformal Parameterization (SCP) [MGB*19] for angle-preserving aneurysm maps, and later As-Rigid-As-Possible (ARAP) [MVG*21] for area-preserving maps. These methods facilitate the visualization of multiple scalar fields but require advanced interaction techniques to maintain a correspondence between the 3D surface and the 2D map. Antiga et al. [AS03, AS04] and Choi et al. [CCLC17, CCR20] used conformal techniques for mapping vessel bifurcations, crucial to analyze vascular calcium deposits on the inside of the vessel wall. Eulzer et al. [EEL*19] and Karim et al. [KMJ*14] focused on the mitral valve and left atrium, respectively, using methods like those described by Lichtenberg et al. [LER*20] for flattening complex structures. Ma et al. [MKH*12], Nuñez-Garcia et al. [NGBD*19], Paun et al. [PBI*17], and Roney et al. [RPM*19] presented various techniques for mapping the ventricles and atria. These include quasi-conformal mapping, and methods using Laplace’s equation. Nevertheless, in contrast to our method, these aforementioned techniques fall short in providing spatial context surrounding the focal object. Moreover, often different additional inputs such as manually selected cut lines or a centerline are needed to generate the map. These key distinctions are why we do not draw direct comparisons between our approach and these parametrization methods.

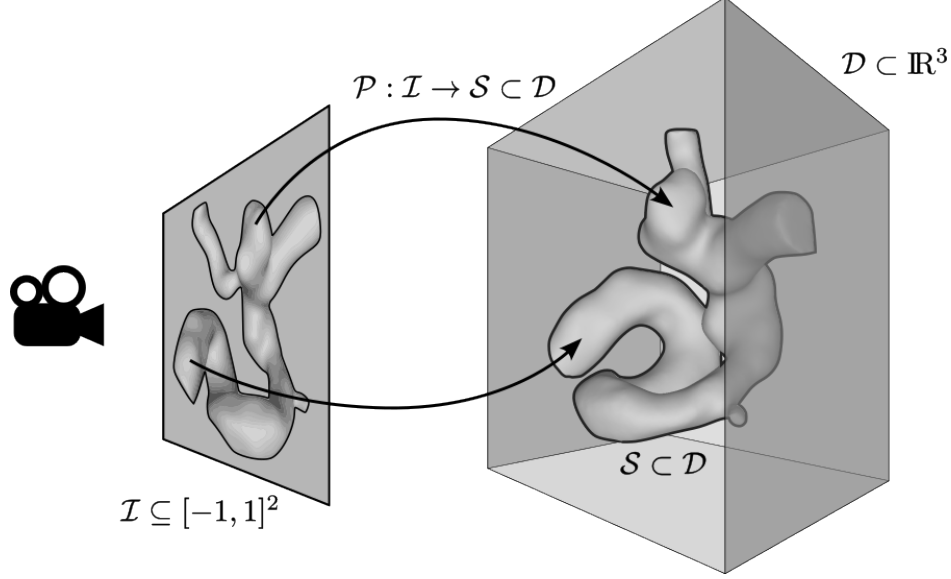


Figure 3: Our goal is to find a mapping $P : \mathcal{I} \rightarrow \mathcal{S}$ that maps from the image plane \mathcal{I} to the surface domain \mathcal{S} , for example, to show as much of the surface as possible or special areas with scalar fields.

2.2 Mirrors

Virtual mirrors simulate a mirror-like interface, enabling users to view parts of a three-dimensional object that are typically obscured or difficult to access. Navab et al. [NFB07] introduced the concept in augmented reality (AR), particularly for medical applications, such as navigated surgery. The virtual mirror is controllable within the AR environment and reflects the virtual part of the scene, adding a second perspective. Based on this work, Bichlmeier et al. [BSN06, BHRN07, BHFN09] evaluated the virtual mirror in advanced AR medical applications, such as navigated drilling in spine surgery and minimally invasive tumor resection in liver surgery. The virtual mirror is attached to tracked objects like surgical instruments, allowing it to reflect virtual entities of the AR scene. This integration simplifies its use, as there is no need for additional devices to control the mirror’s position. The mirror reflects only virtual objects, which are registered with and superimposed on the real objects in the scene, like the anatomy of interest or a drilling device. It aids in planning and controlling surgical procedures by providing a view of regions that are not directly visible. However, the virtual mirror also presents certain disadvantages. Its use is limited to convex structures where the center of mass lies inside the surface, and it may produce area distortions in maps of irregularly shaped objects. The complexity of its interactive nature can pose a learning challenge for users. Moreover, while the mirror enhances visualization, it does not prioritize critical surface structures or formulate optimization problems for mapping important regions, e.g., with high scalar values.

3 Problem Statement and Terminology

Occlusion of relevant information is among the most challenging problems in the visualization of three-dimensional spatial data. When the spatial domain contains a set of surface geometries, the interesting question arises if the two-dimensional image plane can be mapped to the two-dimensional surface space, such that:

1. object parts that are visible to the viewer remain visible,
2. object parts that were occluded are displayed in blank areas,
3. and emphasis is put onto relevant parts of the surface.

In the remainder of this paper, we refer to the image domain as $\mathcal{I} \subseteq [-1, 1]^2$, which identifies a location in screen space in normalized device coordinates (NDC). Further, a set of surfaces occupies the spatial domain $\mathcal{S} \subset \mathcal{D}$, which is a subset of the full spatial domain also known as world space $\mathcal{D} \subseteq \mathbb{R}^3$. Each location in the image domain $\mathbf{i} \in [-1, 1]^2$ corresponds to a world space location $\mathbf{i}_{\mathcal{D}} \in \mathcal{I}_{\mathcal{D}}$ on the near plane $\mathcal{I}_{\mathcal{D}} \subset \mathbb{R}^3$. A camera model provides for each point on the near plane $\mathbf{i}_{\mathcal{D}} \in \mathcal{I}_{\mathcal{D}}$ a view ray direction $\mathbf{r}_{\mathbf{i}_{\mathcal{D}}}$. When using an orthographic projection, the ray direction $\mathbf{r}_{\mathbf{i}_{\mathcal{D}}}$ is the same for every point $\mathbf{i}_{\mathcal{D}} \in \mathcal{I}_{\mathcal{D}}$. Under perspective projection, the view ray direction varies spatially. Our goal is to

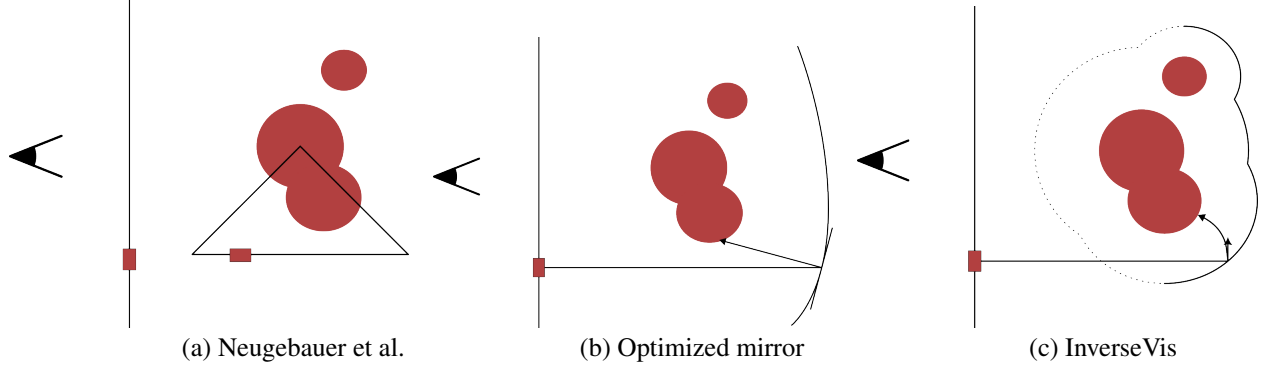


Figure 4: Schematic illustrations: (a) map-based visualizations by Neugebauer et al. [NGB*09], (b) optimized mirror, (c) InverseVis.

find a partial map $\mathcal{P} : \mathcal{I} \rightarrow \mathcal{S}$ that maps from a given image pixel to a point on the surface. A schematic illustration is provided in Fig. 3. Next, we introduce a common notation in which we describe methods that we later compare with. Afterwards, we introduce our novel optimization-based formulation.

Visibility. Under a projective mapping, some parts of the image will see a surface while others do not. We partition the image plane into three regions, depending on the visibility of the surfaces \mathcal{S} :

$$\mathcal{I} = \underbrace{\mathcal{I}^+}_{\text{directly visible}} \cup \underbrace{\mathcal{I}^-}_{\text{indirectly visible}} \cup \underbrace{\mathcal{I}^0}_{\text{not visible}} \quad (1)$$

Region \mathcal{I}^+ contains all pixels that directly see a surface,

$$\mathcal{I}^+ = \{\mathbf{i} \in \mathcal{I} \mid \min_{\lambda \geq 0, \mathbf{s} \in \mathcal{S}} \|\mathbf{i}_{\mathcal{D}} + \lambda \mathbf{r}_{\mathbf{i}_{\mathcal{D}}} - \mathbf{s}\| = 0\}. \quad (2)$$

Region \mathcal{I}^- denotes all the pixels that show parts of the surface in an *indirect* way, and region \mathcal{I}^0 denotes pixels that do not carry information about the surface, for example, they see the background clear color. A (straight) view ray can be defined as $p_{\mathbf{i}_{\mathcal{D}}}(\lambda) = \mathbf{i}_{\mathcal{D}} + \lambda \mathbf{r}_{\mathbf{i}_{\mathcal{D}}}$, $\lambda \in \mathbb{R}^+$. For all pixels $\mathbf{i} \in \mathcal{I}^+$ that hit a surface, the partial map \mathcal{P} is already defined:

$$\mathcal{P}(\mathbf{i}) = \arg \min_{\mathbf{s} \in \mathcal{S}} \min_{\lambda \geq 0} \|p_{\mathbf{i}_{\mathcal{D}}}(\lambda) - \mathbf{s}\|. \quad (3)$$

If only a projective mapping is used, then \mathcal{I}^- is empty. Our goal is to define the image region \mathcal{I}^- and to construct a visual mapping \mathcal{P} for \mathcal{I}^- that conveys information about occluded surface parts.

Map-based Visualization. Neugebauer et al. [NGB*09] constructed a projection from a center $\mathbf{c} \in \mathcal{D}$ in the scene. They defined \mathcal{I}^- as $\mathcal{I}^- = \{\mathbf{i} \in \mathcal{I} \mid r_1 \leq \|\mathbf{i} - \mathcal{P}^{-1}(\mathbf{c})\| \leq r_2\}$ where r_1, r_2 define the inner and outer circle radius and $\mathcal{P}^{-1}(\mathbf{c})$ is the back projection of the center to the screen plane. The 3D coordinates of $\mathcal{P}^{-1}(\mathbf{c})$ will be denoted as $\mathbf{c}_{\mathcal{D}}$. For every $\mathbf{i} \in \mathcal{I}^-$, an angle $\alpha = \arctan(\mathbf{i})$ and a radius $r_1 = \|\mathbf{i}\|$ can be determined. A (straight) ray is then defined as

$$p_{\mathbf{i}}(\lambda) = \mathbf{c}_{\mathcal{D}} + \lambda \cdot \left(\frac{\mathbf{i} - \mathbf{c}_{\mathcal{D}}}{\|\mathbf{i} - \mathbf{c}_{\mathcal{D}}\|} + \left(\frac{2r_1}{r_1 + r_2} - 1 \right) \cdot \mathbf{n} \right), \lambda \in \mathbb{R}^+ \quad (4)$$

where \mathbf{n} is the normal vector of the near plane $\mathcal{I}_{\mathcal{D}}$. Similarly, one can also determine the backsides of the surface and display them on the left-hand side of the screen. Again \mathcal{P} is defined as in Eq. (3). Their approach can be seen in Fig. 2 (left).

Mirror. An alternative is to place a mirror behind the scene. Let $\mathcal{M} \in \mathcal{D}$ be a (possibly curved) mirror surface with the normal field $\hat{\mathbf{n}}(\mathbf{x})$. For $\mathbf{i} \in \mathcal{I}^-$, let $\mathbf{i}_{\mathcal{M}}$ be the intersection point of $\mathbf{i}_{\mathcal{D}} + \lambda \cdot \mathbf{r}_{\mathbf{i}_{\mathcal{D}}}$, $\lambda \in \mathbb{R}^+$ with \mathcal{M} . Then, a (straight) ray is defined:

$$p_{\mathbf{i}}(\lambda) = \mathbf{i}_{\mathcal{M}} + \lambda \cdot \hat{\mathbf{n}}(\mathbf{i}_{\mathcal{M}}), \lambda \in \mathbb{R}^+, \quad (5)$$

and \mathcal{P} is defined as in Eq. (3), if the distance of the curve to the surface is equal to 0. This approach is shown in Fig. 2 (middle).

4 Method

Previous surface visualization methods had difficulties visualizing scalar fields on surface parts that are occluded or are facing away from the camera. In this paper, we propose a new visual encoding of hidden surface geometry on parts of the screen that would otherwise have been unused (Sec. 4.1). Further, we propose a viewpoint selection algorithm that determines the view that conveys the most information (Sec. 4.2). To judge the quality of the resulting visualization, we introduce an energy that measures how well the relevant parts of the surfaces are shown. For this, let $s(\mathbf{x}) : \mathcal{S} \rightarrow [0, 1]$ be a non-negative importance function that expresses how relevant each part of the surface is to see, where 0 means irrelevant and 1 means highly relevant. Then, the energy to maximize is:

$$E[\mathcal{P}] = \gamma \underbrace{\int_{\mathcal{I}^+} s(\mathcal{P}(\mathbf{i})) \, d\mathbf{i}}_{\text{visible part}} + \underbrace{\int_{\mathcal{I}^-} s(\mathcal{P}(\mathbf{i})) \, d\mathbf{i}}_{\text{occluded part}} \quad (6)$$

where the first term accounts for the information seen by regular surface rendering, and where the second term measures the information revealed by our proposed visual encoding. The optional weight γ allows favoring direct visibility, which we kept at $\gamma = 1$ in the paper.

4.1 Curved Ray Formulation

Information on the backside of a surface cannot be shown directly under regular perspective projection because it is hidden by other geometry. This opens a visual design degree of freedom: we may choose a spatial arrangement for the hidden information and place it elsewhere on the screen. To maintain a good spatial scene understanding, we aim to show the hidden information in an empty screen space that is *close* to where the surface would normally be. Since the hidden information is on the backside of the surface geometry, we place a *new* geometry behind the surface geometry, onto which we map the hidden scalar field. We refer to this new surface as the *hull surface*. To determine a mapping from the new geometry to the hidden surface, we propose a physically inspired model: for each point on the new geometry, we let a particle fall in a gravitational field onto the hidden surface. This results in non-linear rays that can reach important regions on the backside, while establishing a well-behaved mapping in the sense that nearby points remain close. The concept is illustrated in Fig. 4. In the following, we explain the necessary ingredients in more detail: the construction of the hull geometry, the seeding of the particles, the governing equation that describes their motion, and an optimization of the model parameters to maximize the visibility of the surface.

Hull Surface. The new hull surface onto which we aim to map the scalar field should be close to the original surface \mathcal{S} . For this purpose, we describe it as a level set in the signed distance field of the surface \mathcal{S} . Thus, we first convert the scene into an implicit representation. Let $\phi(\mathbf{x}) : \mathcal{D} \rightarrow \mathbb{R}$ be the signed distance field to \mathcal{S} , i.e., $\|\nabla\phi(\mathbf{x})\| = 1$ and $\phi(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{S}$. The distances are positive on the inside and negative on the outside. We then define the *hull surface* $\mathcal{H} \subset \mathcal{D}$ as isosurface of $\phi(\mathbf{x})$ with isovalue ϕ_0 :

$$\mathcal{H} = \{\mathbf{x} \mid \phi(\mathbf{x}) = \phi_0\}. \quad (7)$$

The hull surface \mathcal{H} encloses the original surface \mathcal{S} entirely. For each fragment, we use only the part of the hull surface that is farthest from the camera. Formally, for all $\mathbf{i} \in \mathcal{I}^-$, let $p_{\mathbf{i}}(\lambda) = \mathbf{i}_{\mathcal{D}} + \lambda \cdot \mathbf{r}_{\mathbf{i}_{\mathcal{D}}}$, $\lambda \in \mathbb{R}^+$, be a projective mapping, but let $\mathbf{i}_{\mathcal{H}}$ be the intersection point of $p_{\mathbf{i}}$ with \mathcal{H} , such that $\mathbf{i}_{\mathcal{H}} = p_{\mathbf{i}}(\lambda_{max})$, $\lambda_{max} = \arg \max_{\lambda \in \mathbb{R}^+} \{\lambda \in \mathbb{R}^+ \mid \|p_{\mathbf{i}}(\lambda) - h\| = 0, h \in \mathcal{H}\}$. Thus, it is the intersection point of the projective mapping with the hull surface, which is farthest away from the camera. The point $\mathbf{i}_{\mathcal{H}}$ is located behind the object and is treated as a seed point for a curved ray towards the surfaces.

Curved Particle Tracing. Intuitively speaking, we model the path from the hull surface onto the hidden surface by a particle that falls from the hull surface gravitationally onto the hidden surface of \mathcal{S} . The movement of a particle, which represents the ray, can be described by a velocity and an acceleration, which are formulated as an ordinary differential equation (ODE). To model the path of a particle, we initially utilize sphere tracing, followed by ensuring that it descends onto the surface denoted by \mathcal{S} . The concept of sphere tracing involves selecting the particle’s step size so that it advances in a direction equal to the shortest distance to the surface. This allows taking larger steps compared to ray marching with a small step size. Ultimately, this implies that the particle’s velocity is determined by multiplying its current normalized velocity by the nearest distance to the surface. The second principle dictates that the particle should be drawn towards the surface \mathcal{S} , signifying in physics that the acceleration vector is directed toward the surface. Both constraints can be described by a coupled first-order ODE:

$$\frac{d}{dt} \mathbf{p} = \phi(\mathbf{p}) \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad \frac{d}{dt} \mathbf{v} = -\nabla\phi(\mathbf{p}). \quad (8)$$

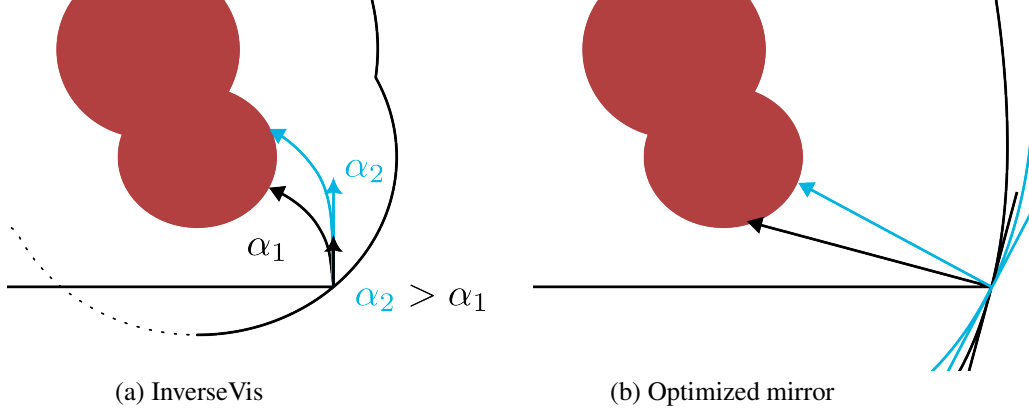


Figure 5: *InverseVis* (a) optimizes a scale factor α of the initial velocity, while the mirror (b) is optimized for its surface normal.

Given initial conditions $\mathbf{p}(0) = \mathbf{p}_0 = \mathbf{i}_{\mathcal{H}}$ and $\dot{\mathbf{p}}(0) = \mathbf{v}_0 = \alpha \cdot \nabla \phi(\mathbf{i}_{\mathcal{H}}) \times (\mathbf{r}_{\mathbf{i}_{\mathcal{D}}} \times \nabla \phi(\mathbf{i}_{\mathcal{H}}))$, we numerically solve this ODE using the symplectic Euler method. Again \mathcal{P} is defined as in Eq. (3), if the distance of the curve $\mathbf{p}(t)$ to the surface is equal to 0. Note, that the parameter α controls the initial velocity, as shown in Fig. 5 (a), which has an impact on where the curve will hit the surface.

Estimating the Impact of α . Our goal is to optimize α to maximize the amount of information visible on screen, according to Eq. (6). How much α impacts the trajectory can be studied through a continuum mechanical view onto a small variation of α [KPH*09]. For this, we first lift the coupled first-order ODE into a six-dimensional state $\bar{\mathbf{x}}$, which changes in direction $\mathbf{F}(\bar{\mathbf{x}})$:

$$\underbrace{\frac{d}{dt} \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix}}_{\bar{\mathbf{x}}} = \underbrace{\begin{pmatrix} \phi(\mathbf{p}) \frac{\mathbf{v}}{\|\mathbf{v}\|} \\ -\nabla \phi(\mathbf{p}) \end{pmatrix}}_{\mathbf{F}(\bar{\mathbf{x}})}. \quad (9)$$

To study how a change in the initial condition of the ODE $\bar{\mathbf{x}} = \mathbf{F}(\bar{\mathbf{x}})$ carries over time, we add a small perturbation δ :

$$\frac{d}{dt}(\bar{\mathbf{x}} + \delta) = \mathbf{F}(\bar{\mathbf{x}} + \delta) \quad (10)$$

$$\frac{d}{dt}\bar{\mathbf{x}} + \frac{d}{dt}\delta \approx \mathbf{F}(\bar{\mathbf{x}}) + \mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}}) \cdot \delta \quad (11)$$

$$\dot{\delta} \approx \mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}}) \cdot \delta \quad (12)$$

where we apply in Eq. (11) a first-order Taylor expansion around $\bar{\mathbf{x}}$ to describe the evolution from information at $\bar{\mathbf{x}}$. Note that the change δ is governed by the Jacobian $\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}})$ of the phase flow $\mathbf{F}(\bar{\mathbf{x}})$:

$$\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}}) = \frac{\partial \mathbf{F}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} = \begin{pmatrix} \frac{\mathbf{v}}{\|\mathbf{v}\|} \cdot \nabla \phi(\mathbf{p})^T & \phi(\mathbf{p}) \left(\frac{\mathbf{I}_{3 \times 3}}{\|\mathbf{v}\|} - \frac{\mathbf{v} \mathbf{v}^T}{\|\mathbf{v}\|^3} \right) \\ -\mathbf{H}_{\phi}(\mathbf{p}) & \mathbf{0}_{3 \times 3} \end{pmatrix}. \quad (13)$$

Over time, the perturbation in Eq. (12) is integrated by:

$$\delta(t) = \delta(0) + \int_0^t \underbrace{\mathbf{J}_{\mathbf{F}}(\bar{\mathbf{x}}(\tau))}_{\mathbf{J}(\tau)} \cdot \delta(\tau) d\tau \quad (14)$$

The integrand of Eq. (14) is a linear vector field in $\delta(\tau)$. Upon temporal discretization of this integral, the integration in the linear vector field can be solved analytically at each time step, integrating $\delta(t_i)$ in the linear vector field up until the next time step $\delta(t_{i+1})$.

$$\delta(t_{i+1}) = e^{\mathbf{J}(t_i)h} \delta(t_i) \quad (15)$$

where $h = t_{i+1} - t_i$ is the numerical integration step size. The matrix exponential $e^{\mathbf{J}(t)h}$ is calculated by diagonalization:

$$e^{\mathbf{J}(t)h} = \bar{\mathbf{E}}(t) \cdot \bar{\mathbf{D}}(t, h) \cdot \bar{\mathbf{E}}(t)^{-1} \quad (16)$$

where $\bar{\mathbf{E}}(t)$ is the eigenvector matrix of $\mathbf{J}(t)$, and where \mathbf{D} is the diagonal matrix of eigenvalues of $\mathbf{J}(t)$:

$$\bar{\mathbf{E}}(t) = (\bar{\mathbf{e}}_1, \dots, \bar{\mathbf{e}}_6), \quad \mathbf{D}(t) = \text{diag}(\bar{\lambda}_1, \dots, \bar{\lambda}_6) \quad (17)$$

and where $\bar{\mathbf{D}}(t, h) = e^{\mathbf{D}(t)h} = \text{diag}(e^{\bar{\lambda}_1(t)h}, \dots, e^{\bar{\lambda}_6(t)h})$ is the matrix exponential of the eigenvalue matrix computed at time t for step size h . Concatenating Eq. (15) over all time steps, this gives:

$$\boldsymbol{\delta}(t_n) = \underbrace{\left(\prod_{i=n-1}^0 e^{\mathbf{J}(t_i)h} \right)}_{\psi(\mathbf{t}_n)} \boldsymbol{\delta}(0) \approx \underbrace{\left(\prod_{i=n-1}^0 \mathbf{I} + h\mathbf{J}(t_i) \right)}_{\text{1st-order Taylor approx.}} \boldsymbol{\delta}(0) \quad (18)$$

which can be first-order Taylor approximated. The matrix $\psi_{6 \times 6}$ tells how an initial perturbation $\boldsymbol{\delta}(0)$ impacts the end of the path integration in Eq. (8). Since α is a linear scaling of the velocity subspace of the initial condition \mathbf{v}_0 , its impact is:

$$\frac{\partial \bar{\mathbf{x}}(t)}{\partial \alpha} = \psi(t) \cdot \left(\mathbf{0}_3, \frac{\partial \mathbf{v}_0}{\partial \alpha} \right)^T. \quad (19)$$

The first three components denote the change in the position $\frac{\partial \mathbf{p}(t)}{\partial \alpha}$. With t being chosen to be the time of the surface hit, this gives $\frac{\partial \mathcal{P}}{\partial \alpha}$.

Optimization. Knowing from Eq. (19) the impact that a variation of α has, we can perform a gradient-based optimization of α to find the setting that maximizes the energy E in Eq. (6) best:

$$\alpha^{(i+1)} = \alpha^{(i)} + h \cdot \frac{\partial E}{\partial \mathcal{P}} \frac{\partial \mathcal{P}}{\partial \alpha}, \quad (20)$$

where $\alpha^{(0)}$ is an initial guess, which we set to 0.5, and where the energy partial with respect to the endpoint is $\frac{\partial E}{\partial \mathcal{P}} = \nabla s(\mathcal{P})$. During user interaction, a gradient-based optimization is favorable over gradient-free methods, since the optimized α adjusts smoothly over time. To accelerate the convergence, an optimal step size h can be determined by a line search, for which we employ a golden-section search [Kie53] within the empirically chosen interval $h \in [0, 0.25]$. When given the interval $[X_1, X_4]$, it is divided in four parts: X_1, X_2, X_3, X_4 , with $X_2 = X_4 - (X_4 - X_1)/\Phi$ and $X_3 = X_1 + (X_4 - X_1)/\Phi$, where $\Phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. If $E_i + X_2 \nabla E_i > E_i + X_3 \nabla E_i$, then we repeat this procedure with the new interval $[X_1, X_3]$ otherwise, we use the interval $[X_2, X_4]$. This is repeated until the length $L([a, b]) = b - a$ of the interval falls below a threshold (in our case, 0.01), at which point h is set to the interval's mean value. After a few iterations, we identify the optimal step size h , and the gradient ascent is continued. The iteration halts if the energy decreases after the line search compared to its value before the search.

4.2 Viewpoint Optimization

The previous section introduced a visual encoding that conveys more information about hidden surface geometry. Since the approach is maximizing the energy in Eq. (6) for a given viewpoint, the interesting question arises, to whether an even better viewpoint could be found automatically. Previous research has investigated various methods for determining optimal camera positions, e.g. those that maximize high values in the scalar field [MEB*17] or enhance visibility [VFSH01]. In our work, we maximize the energy in Eq. (6) for every camera position to find the best view.

During our experiments, while searching for camera points that would maximize the energy, we encountered numerous local maxima. This necessitated the implementation of an optimization method capable of locating a global optimum within the camera space. To address this challenge, we used *simulated annealing*, which proved to be satisfactory both in terms of results and performance. In our configuration, the camera is permitted to move along a sphere that encircles the object in focus. Without loss of generality, the object's dimensions are confined within $\mathbf{p} \in [-1, 1]^3$, and the radius of the surrounding sphere is fixed at 2.5, which ensures that the whole object is visible and a margin exists that can be used for surrounding visualization. Consequently, the camera movement is defined by two degrees of freedom, specifically the polar and azimuthal angles, denoted as θ and φ , respectively.

Simulated annealing begins by determining the energy based on the camera's current position. Subsequently, a new camera position is randomly selected within the neighborhood of the existing position. The range for this neighborhood is set to $\pm 60^\circ$. We found that this range is optimal as smaller angles tend to confine the search to local areas, while larger angles might prevent the algorithm from refining the optimum on a local scale. When comparing the energy of the new camera position against the old one, a position is retained if it yields higher energy. However, if the new position

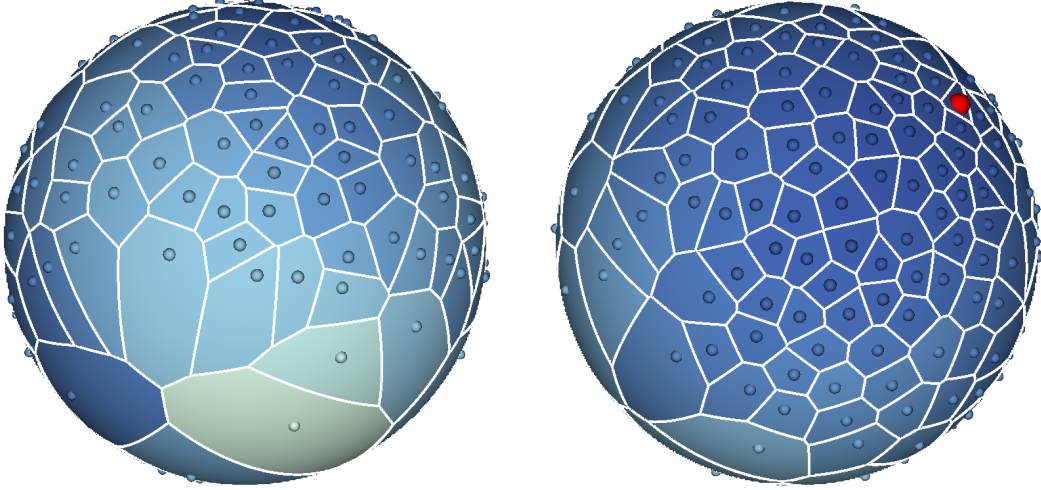


Figure 6: This sphere shows the tested camera positions during the simulated annealing (front and backside). Dark blue represents values of high energy, which have a higher sample rate compared to less optimal regions. The found optimum is shown in red.

has less energy, it is still accepted with a probability of $\exp\left(10 \cdot \frac{E_{new} - E_{old}}{T}\right)$, where T represents the temperature. The temperature starts at 1 and gradually decreases with each step. Illustrated in Fig. 6, we can observe the regions on a sphere visited by the camera during the simulated annealing process. In this instance, using the aneurysm dataset, we set s in Eq. (6) to represent visibility. This process showcases the effectiveness and adaptability of our simulated annealing approach in navigating and optimizing camera positions for complex surfaces.

5 Implementation

5.1 Signed Distance Field

At its core, our objective is to trace rays around a surface to visualize its backside. Since the rays are non-linear a ray-triangle intersection test after each numerical integration step is computationally expensive. To expedite this process, we integrate rasterization and ray marching. For rendering the front faces of the surface, we utilize rasterization as facilitated by the graphics pipeline. For linear rays that encompass the surface, such as for casting shadow rays, we employ ray marching. To accelerate this, we utilize sphere tracing, which requires a signed distance field (SDF).

We construct a voxelized space around this surface, which ranges in $[-2.5, 2.5]$. Inside this box, we construct 200 voxels in every dimension. The higher the number of voxels, the higher the resolution for the rendered backside of the surface. We found that 200 gives satisfactory results. For every voxel, we determine the closest point on the surface and store: (i) the distance, (ii) the corresponding triangle ID, and (iii) the barycentric coordinates of the closest point inside the triangle. During the sphere tracing process, we efficiently intersect the surface at a given distance. This not only facilitates a rapid computation but also provides us with the corresponding triangle and its barycentric coordinates. This crucial information enables us to effectively visualize the scalar field, as well. Additionally, we upload the triangle vertex indices to the GPU, where they are accessible alongside other data (i)-(iii) in a shader storage buffer. On the GPU, the available distances are accessible as a 3D texture, enabling us to perform sphere tracing on the surface. Through this process, we first identify the potential triangle ID that would be intersected, followed by obtaining the vertex IDs of the triangle. The barycentric coordinates then pinpoint the exact position, allowing us to accurately determine the scalar field at that specific location.

5.2 Mirror Optimization

We will compare our InverseVis method with an optimized curved mirror that is in the shape of a quadratic height field $H(\mathbf{x})$:

$$H(\mathbf{x}) = z - \omega_1 x^2 - \omega_2 y^2 - \omega_3 xy - \omega_4 x - \omega_5 y = 0 \quad (21)$$

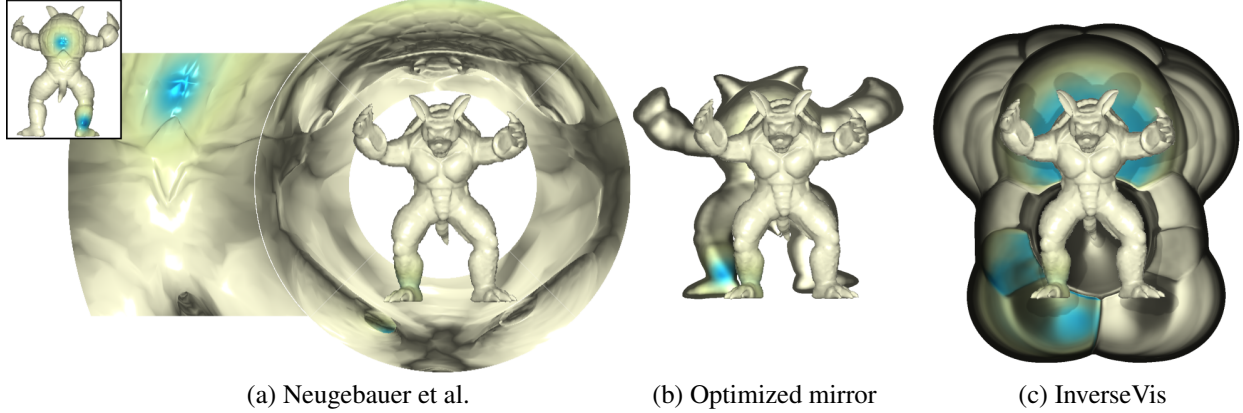


Figure 7: In this scene, two distinct hidden regions are marked as important (on the back and on the leg). Neugebauer et al. [NGB*09] shows the back, but the spot on the foot is rather unnoticeable. The mirror shows only the foot. InverseVis is able to reveal both hidden regions.

with $\mathbf{x} = (x, y, z)$. This height field is placed behind the surface at a distance of 1, which could also be optimized, resulting in six degrees of freedom. The normals $\mathbf{n}(x, y)$ of this height field are:

$$\mathbf{n}(x, y) = \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} -2\omega_1 x - \omega_3 y - \omega_4 \\ -2\omega_2 y - \omega_3 x - \omega_5 \\ 1 \end{pmatrix}. \quad (22)$$

We utilize this normal as the reflection vector to visualize the backsides. On the GPU, we intersect the mirror at the point $\mathbf{m}(x, y) = \mathbf{m}_0$ and determine its normal $\mathbf{n}(x, y) = \mathbf{n}_0$. Then, sphere tracing is employed to intersect the surface S applying the forward Euler method:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \phi(\mathbf{p}_i) \cdot \mathbf{n}_0, \quad \mathbf{p}_0 = \mathbf{m}_0. \quad (23)$$

The mirror has five degrees of freedoms $\omega_1, \dots, \omega_5$, which we optimize to maximize the energy Eq. (6), see Fig. 5 (b). Similar to the α optimization in Eq. (20), we perform a gradient-based optimization with a linear search. That is, we calculate the energy, then, for every ω_i , we slightly alter the parameter by an $\epsilon = 0.025$ to determine the resultant energy change, necessary for calculating $\frac{\partial E}{\partial \omega_i}$.

5.3 Symplectic Euler for InverseVis

Tracing the non-linear rays as governed by the ODE from Eq. (8) involves using a numerical procedure. For second-order ODEs, a symplectic Euler method shows better energy conservation compared to explicit integration methods:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + h \cdot \phi(\mathbf{p}_i) \frac{\mathbf{v}_{i+1}}{\|\mathbf{v}_{i+1}\|}, \quad \mathbf{v}_{i+1} = \mathbf{v}_i - h \cdot \nabla \phi(\mathbf{p}_i). \quad (24)$$

In our implementation, we use $h = 0.1$ as a step size.

5.4 OpenGL Implementation

Next, we outline the implementation steps of our visualization techniques. Initially, we determine the SDF for a given surface mesh, which includes distances, triangles, and barycentric coordinates (refer to Sec. 5.1). Following this, we either identify the intersection point with the mirror or with the hull surface at a radius of 0.4. Depending on whether we use the mirror or InverseVis, two main tasks are required: (i) optimizing the corresponding parameters and (ii) implementing the forward Euler method (see Sec. 5.2 and Sec. 5.3).

For each parameter under optimization, we increment it by a small value $\epsilon = 0.025$ (i) and then trace the path to the surface (ii). The process for the mirror involves sphere tracing, whereas, for InverseVis, we employ the forward Euler method with a step size determined through the golden-section search (Sec. 4.1). After adjusting every parameter with ϵ , we calculate the gradient using forward differences and again use the golden-section search to find the optimal parameter combinations.

The energy, cf. Eq. (6), can be based on either the scalar field or visibility. For the scalar field, we add the scalar values of each ray that hits the surface. This is achieved using the OpenGL extension `GL_NV_shader_atomic_float`, which

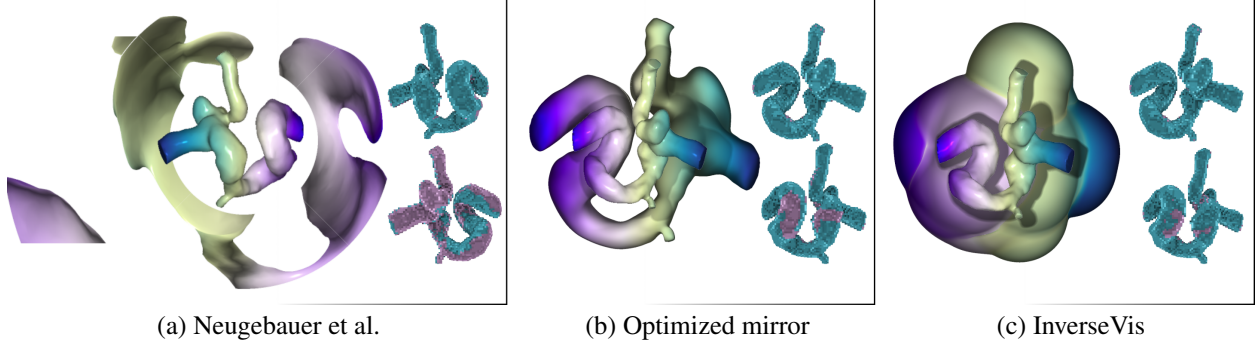


Figure 8: Optimal viewpoints on the Aneurysm 3 dataset. Shown are the techniques (vertices are colored according to their voxels) as well as the visible (teal) and non-visible (violet) voxels on the right with front and backside. From a total of 6,910 voxels, Neugebauer et al. [NGB*09] reaches 4,518, the optimized mirror 6,173, and InverseVis 6,654.

facilitates the addition of float values in the fragment shader. We also use an atomic counter to track the frequency of surface hits, which allows us to calculate the average scalar value in the image.

If optimizing for visibility, we create a 3D texture of size $200 \times 200 \times 200$ and mark all voxels within a distance smaller than the voxel size to the surface \mathcal{S} , identifying voxels near the surface. To assess visibility, we count voxels that are hit by rays. When a ray intersects a surface, we locate the current voxel and increment a count if it is a valid (marked) voxel that has not been visited by another ray. We then calculate the ratio of all marked and visited voxels. Optionally, users can determine the optimal camera position based on different energies (scalar field or visibility), cf Sec. 4.2.

6 Evaluation

To assess the effectiveness of our new methods, which are designed to reveal the backsides of surfaces, we carried out three evaluations. Firstly, it was crucial to quantify how much of the surface is visible when employing these different methods. Achieving values close to 100% indicates that the method is successful in achieving its goal of comprehensive surface visibility. The second aspect of our evaluation focused on the time required to identify optimal camera positions. A technique, despite offering excellent visibility, might not be user-friendly if it requires an excessively long time for optimization. Finally, the last part of our evaluation involved gathering qualitative feedback from various visualization experts, providing insights into the practicality and efficiency of the different methods.

6.1 Visibility Measure

We focused on measuring visibility, or more precisely, the extent of the surface \mathcal{S} that is visible when using different techniques. For this purpose, we utilized the voxelized space of the signed distance field, as detailed in Sec. 5.1. In this process, we marked every voxel within a distance smaller than the voxel size to the surface \mathcal{S} , effectively identifying all voxels in proximity to the surface.

When camera rays intersect the surface, the corresponding voxels are designated as visited. We count all visited voxels and divide this by the total number of marked voxels. This calculation yields a value in $[0, 1]$, indicating the portion of the surface that is visible. To achieve the best results, we employed the simulated annealing approach described in Sec. 4.2, aiming to maximize visibility for each technique, as shown in Tab. 1. Fig. 7 shows a scene, in which two distinct regions on the surface are marked as important (blue). Neugebauer et al. [NGB*09] and the optimized mirror, both fail to show both important structures, while our method clearly displays both regions. Fig. 8 and Fig. 9 illustrate the visibility achieved by all methods on different surfaces, with vertices colored according to their corresponding voxels used in the visibility measurement. The method by Neugebauer et al. [NGB*09] struggles to capture all backside information due to the model’s complexity, while the mirror reflects more of the backside but does not fully utilize the surrounding space. In contrast, InverseVis can capture the most information, showcasing its effectiveness in visualizing the surface. Fig. 10 gives an example of the mirror optimization focusing on only one small (yet important) part on the backside, which misses all other relevant structures on the backside, while the InverseVis approach can reveal all structures on the backside.

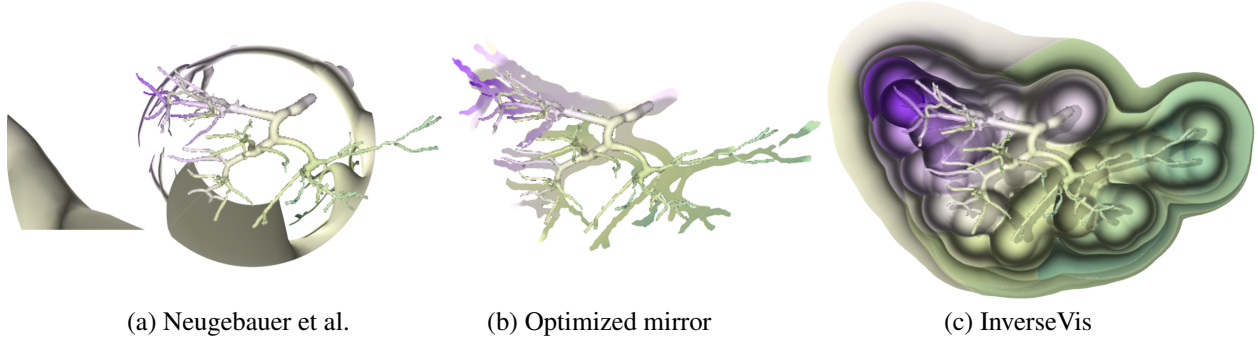


Figure 9: The vertices are colored according to their voxel, which is used for the visibility measure. Neugebauer et al. [NGB*09] (left) captures only a small part of the backside due to the complexity of the surface and the reference point for the views, the optimized mirror (middle) reflects more of the backside but the space is not fully used, and InverseVis (right) uses cavities for tracing the rays to the backsides.

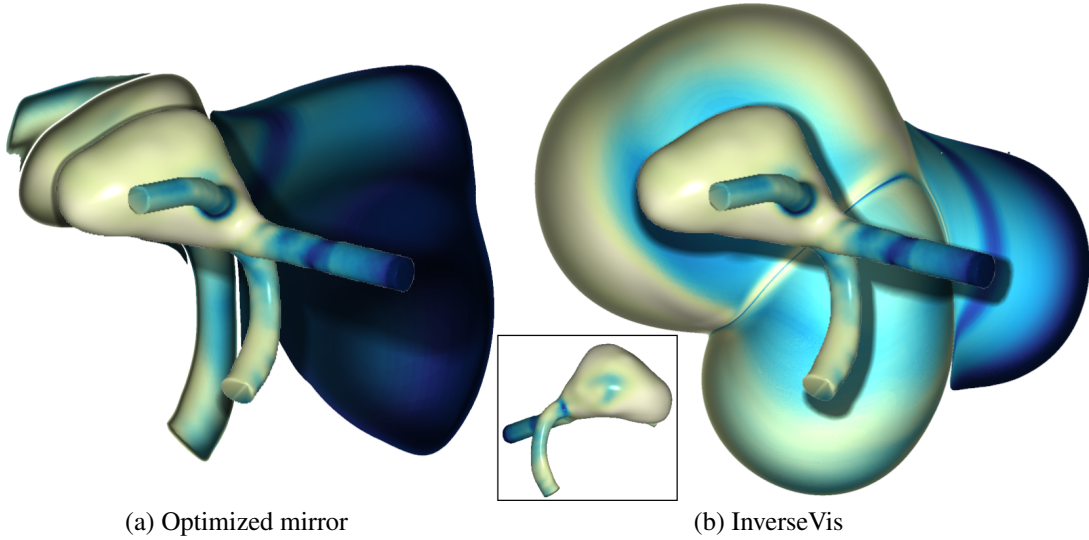


Figure 10: The mirror optimization (left) significantly enlarges the right vessel area, which increases the energy visibility, yet overlooks other relevant areas of the aneurysm. In contrast, InverseVis (right) also shows the aneurysm's backside, ensuring a more balanced visual representation. The boxed image below displays the backside.

6.2 Timings

In this section, we discuss the time required to identify an appropriate camera position that maximizes visibility. It is important to note that the time taken to optimize the scalar fields closely mirrors that for enhancing visibility. Therefore, we will focus on reporting the latter, as detailed in Tab. 2. Through empirical observation, we found that the camera consistently optimizes itself to the same position, regardless of whether the optimization resolution is set at 100 or 1000. This observation suggests that choosing a lower resolution is advantageous, as it leads to a significantly quicker optimization process without compromising the optimal camera positioning. The experiments were conducted on an Intel Core i9 @3.60GHz, 32 GB RAM, and an NVIDIA GeForce GTX 2080.

6.3 Qualitative Feedback

We conducted a qualitative user study with five visualization experts specialized in the visualization of biomedical data (V1-V5), two neuroradiologists (N1,N2)), and two CFD experts (C1,C2) to compare the method by Neugebauer et al. [NGB*09], the *mirror optimization*, and *InverseVis*. For this purpose, each expert explored six different datasets (a medical and a non-medical dataset for each method, see Tab. 1, models are marked with *). The study involves assessing the three methods regarding various aspects such as understandability, clarity in geometric correspondences, visibility of interesting regions, and performance with complex surfaces. All statements are structured to elicit responses

Table 1: The visibility in percent of different surfaces compared to the methods. Percentages in brackets present the visibility of the front faces, rendered with direct rendering (*are used for evaluation).

Surface	Neugebauer et al.	Mirror	InverseVis
Armadillo*	79 % (46 %)	88 % (47 %)	97 % (45 %)
Cow*	91 % (38 %)	95 % (38 %)	98 % (47 %)
Fertility*	71 % (41 %)	86 % (34 %)	96 % (43 %)
Gargoyle*	84 % (42 %)	88 % (39 %)	95 % (37 %)
Neptune*	72 % (43 %)	95 % (53 %)	96 % (53 %)
Rocker arm	84 % (55 %)	97 % (47 %)	98 % (46 %)
Sphere	100 % (43 %)	75 % (43 %)	100 % (43 %)
Aneurysma 1*	84 % (41 %)	92 % (43 %)	100 % (43 %)
Aneurysma 2*	81 % (41 %)	84 % (44 %)	96 % (44 %)
Aneurysma 3	65 % (37 %)	89 % (42 %)	96 % (42 %)
Carotis 5*	78 % (36 %)	93 % (46 %)	98 % (49 %)
Carotis 11*	68 % (44 %)	97 % (45 %)	100 % (49 %)
Vessel Tree*	73 % (47 %)	84 % (48 %)	98 % (45 %)

Table 2: Timings in seconds of the different methods on various data sets. The display was rendered with a resolution of 1000×1000 . The optimization resolution is set to either 100 or 1000.

Surface	Neugebauer et al.		Mirror		InverseVis	
Optim. Res.	100	1000	100	1000	100	1000
Armadillo	10	37	18	300	13	40
Neptune	13	36	16	245	10	35
Aneurysma 1	13	33	18	350	10	39
Carotis 5	23	42	26	298	23	120
Vesseltree	14	39	14	115	14	70

on a 5-point Likert scale, ranging from 'Strongly Disagree' to 'Strongly Agree', see Fig. 11. The statements (S1-S5) that each expert should evaluate for all three techniques are:

- **S1 Understandability:** This method is easy to understand.
- **S2 Geometric Correspondences:** It is easy to correlate scalar field values with their corresponding surface position.
- **S3 Visibility of Interesting Regions:** This method effectively highlights the most interesting surface regions.
- **S4 Handling of Complex Surfaces:** This approach performs well with surfaces of varying complexity.
- **S5 Clarity:** The visualization is both clear and well-organized.

Results of the 2D Map [NGB*09]. This approach faced some challenges in the user study, with eight experts rating it as less understandable (S1), indicating a general difficulty in grasping the method. Especially the medical experts found it difficult to understand that the back of the surface was projected onto a part to the left of the map ring. The experts quickly lost their orientation, especially when rotating more complex surfaces such as vascular trees. This difficulty further extended to correlating scalar fields with surface regions (S2), as reflected in the lower ratings from the experts. Due to the ring-shaped ordering of the projection results, the experts had problems correlating scalar field values with their corresponding surface position, especially in the case of complex surfaces, like vessel trees. A notable issue highlighted by most participants was the method's ineffectiveness in highlighting interesting regions (S3), suggesting a lack of clarity in visual emphasis. Depending on the surface complexity and camera position, hotspots on the surface may be partially or completely hidden in the projection. For example, the neuroradiologists were interested in surface regions with high stress values in the aneurysm data sets. However, such regions are difficult to detect if they occur in concave regions at the junction with the parent vessel. This also explains the ratings for handling complex surfaces (S4), where most experts leaned towards the lower end of the scale. For complex surfaces with fine structures, such as vascular branchings, only very small sections of the surface can be projected. Despite the challenges associated with this method, experts appreciated its well-structured design in the 2D map format (S5). The medical experts, in particular, are very familiar with 2D representations, which is why they liked the general 2D concept of the map display. Particularly for convex surfaces without holes, the ring-shaped representation was regarded as clear and uncluttered.

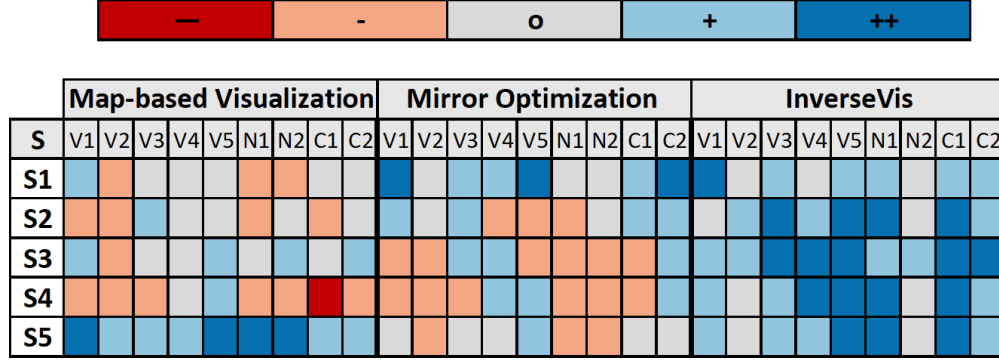


Figure 11: Likert scores of the five study participants regarding the five statements for each of the techniques, Neugebauer et al. [NGB*09] (left), optimized mirror (middle), and InverseVis (right).

Results of the Mirror. Six experts found the mirror method easy to understand (S1), noting that the concept of using a mirror is familiar and intuitive to most people. Three experts, in particular the neuroradiologists, struggled with its intuitiveness since, depending on the camera position and surface complexity, it might happen that surface parts are more deformed, leading to presentations that are difficult to understand. In terms of geometric correspondences (S2), participants were evenly divided, indicating a varied ability to correlate geometric features with the method. Again, possible strong distortions make it difficult to correlate scalar values and associated surface regions. The visibility of interesting regions (S3) was another area where the method received critique. Depending on the surface complexity and number of interesting regions, individual hotspots may be occasionally overshadowed. In line with previous assessments, handling complex surfaces (S4) proved challenging for some, with six experts finding it less effective in this regard, indicating difficulty in visualization. The deformation of the surface induced by the mirror can lead to confusing views, especially in the case of concave surfaces. This also explains the rather reserved assessment of its clarity (S5). This reasoning also sheds light on the somewhat cautious evaluation of its clarity (S5). The experts faced issues with orientation and navigation, particularly when adjusting the camera position and dealing with complex surfaces.

Results of InverseVis. InverseVis was received positively, with six experts rating it highly for its understandability (S1). Three experts indicated that it required some initial effort to learn, leading to neutral assessments. But they also mentioned that once they became accustomed to the method, it was easy to adapt to and use effectively. This positive response was echoed in its approach to geometric correspondences (S2), with the majority of experts (7 out of 9) assessing it as easy to correlate scalar field values with surface regions. Here, the experts also acknowledged the additional depth hints in the form of shadows that further support orientation. Only two experts encountered challenges with the correlation task, finding that the corresponding surface position was not always unambiguous, particularly when adjusting the camera position. In addition, InverseVis received very positive feedback regarding the visibility of interesting regions (S3), with all participants agreeing. Compared to the Mirror, InverseVis highlights intriguing regions on the surface, e.g., on the aneurysm compared to the less interesting inflow vessel, see Fig. 10. This shows the strength of the camera optimization to find interesting regions. Regarding the handling of complex surfaces (S4), the method received positive feedback, with a majority (7 out of 9) finding it effective, though two experts had some reservations. Specifically, for intricate structures like branched vessel trees, these experts experienced orientation difficulties with certain camera positions. This, in turn, also led to two cautious assessments regarding the structure of the presentation (S5), while all other experts perceived InverseVis as clear and well-structured.

Final Ranking. At the end of the user study, the participants were asked to rank the three methods regarding two major aspects:

1. Ease of Understanding
2. Revealing Interesting Regions

The InverseVis method emerged as the clear leader in both ease of understanding and effectiveness in revealing interesting regions. Participants found the camera optimization to be the most intuitive and straightforward, attributing high scores to its understandability and ability to correlate geometric features with surface regions. Additionally, InverseVis was praised for its efficient highlighting of important areas without overshadowing or overemphasizing, and its adaptability to complex surfaces. The Mirror ranked second in both categories, offering moderate ease of understanding. However, it faced challenges in consistently highlighting important regions and in its handling of complex surfaces. Its real-time deformation feature and adaptability were noted, but its balanced focus in visualization

was questioned. The Neugebauer et al. approach was found to be the least intuitive and understandable. Participants struggled with its learning curve, finding it challenging to correlate visualization aspects with surface regions and to highlight important regions effectively. It was also perceived as less efficient in handling complex surfaces, contributing to its lower ranking in these categories.

7 Discussion

The three methods aim to reveal hidden surfaces. Neugebauer et al. use the arrangement of different rendered pages and the Mirror and InverseVis try to increase visibility through optimizations. The mirror is slower than the other methods, which is because five parameters have to be optimized. For InverseVis and Neugebauer et al., the computation time and the comprehensibility both depend on the complexity of the mesh. InverseVis was slower for the vessel tree. For simple convex objects, such as an aneurysm, Neugebauer et al.’s method is easy to understand once users have gained some practice. However, this was more difficult for the vessel tree, as gaps occur that lead to loss of orientation. This is reduced with the mirror, but as soon as it bends strongly to emphasize important regions, strong distortions occur, which we want to avoid. These strong distortions cannot occur with Neugebauer et al. and InverseVis.

We found that the camera position was not sensitive to the resolution used for optimization. We suspect that this was due to the large spatial extent of important structures, which were recognizable even at low resolution. Certainly, the resolution will influence the data if there are small regions with high scalar values that the rasterizer does not cover. However, this generally leads to an interesting problem, which can also apply to normal direct rendering.

The qualitative user study reveals insightful feedback from experts with varying backgrounds on the use of the different visualization methods. Despite their different levels of expertise, the experts did not show any significant differences in their assessments of the individual methods. Medical experts preferred 2D representations in principle but found the 2D map-based method challenging to understand, especially compared to the InverseVis method. Both medical and CFD experts required more time to familiarize themselves with the InverseVis method compared to visualization experts, suggesting the potential benefit of small tutorials for new users in the future.

8 Conclusions and Future Work

Exploring surfaces with scalar fields can be a tedious process. For this reason, we have addressed the issue of displaying the backsides in addition to the standard front rendering. We took an existing method from Neugebauer et al., then improved an existing concept, the mirrors, and developed a new method, InverseVis. We were able to optimize the new methods concerning energy so that either the visibility or interesting scalar fields are highlighted. We tested the performance of the method by finding optimal camera positions and by comparing the visibility. Finally, we evaluated the methods in a qualitative study with participants. The optimal mirror and InverseVis were able to significantly improve the previous method, as we were able to demonstrate with several examples and prove in our studies. We are convinced that the combination of rasterization and ray marching, i.e., curved sphere tracing, offers a successful alternative and has great potential for the future.

Currently, we have only dealt with static scalar fields. In the future, we want to deal with the question of how to find not only the best camera position but also good parameter combinations to then visualize as much information as possible on average over a time-varying scalar field. Another interesting aspect would be to determine not only the (one) global optimum for the camera position but also the best viewpoints and to define a camera animation. Furthermore, additional information related to the detected best viewpoints should be presented, e.g., interesting flow patterns close to vessel wall regions with highest shear forces [EMKL21, NLB*13]. Besides, scalar fields, both vector and tensor fields mapped to surfaces play an important role, e.g., in analyzing medical flow simulation results [MVB*17b]. Therefore, we plan to extend our optimization concerning the visibility of vector and tensor information.

Acknowledgements

This work was supported by the Carl Zeiss Foundation, within the project Interactive Inference, and DFG grant no. GU 945/3-1.

References

- [AS03] ANTIGA L., STEINMAN D. A.: *Automated Parameterization and Patching of Bifurcating Vessels*. Tech. rep., EPFL Infoscience, 2003.
- [AS04] ANTIGA L., STEINMAN D. A.: Robust and objective decomposition and mapping of bifurcating vessels. *IEEE Transactions on Medical Imaging* 23, 6 (2004), 704–713. doi:10.1109/tmi.2004.826946.
- [BHFN09] BICHLMEIER C., HEINING S. M., FEUERSTEIN M., NAVAB N.: The virtual mirror: A new interaction paradigm for augmented reality environments. *IEEE Transactions on Medical Imaging* 28, 9 (2009), 1498–1510.
- [BHRN07] BICHLMEIER C., HEINING S. M., RUSTAE M., NAVAB N.: Laparoscopic virtual mirror for understanding vessel structure evaluation study by twelve surgeons. In *Proc. of 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007), IEEE, pp. 125–128.
- [BSN06] BICHLMEIER C., SIELHORST T., NAVAB N.: The tangible virtual mirror: New visualization paradigm for navigated surgery. In *Proc. of Augmented Reality Environments for Medical Imaging and Computer-Aided Surgery* (2006).
- [CCLC17] CHOI G. P. T., CHEN Y., LUI L. M., CHIU B.: Conformal mapping of carotid vessel wall and plaque thickness measured from 3D ultrasound images. *Medical & Biological Engineering & Computing* 55, 12 (2017), 2183–2195. doi:10.1007/s11517-017-1656-4.
- [CCR20] CHOI G. P. T., CHIU B., RYCROFT C. H.: Area-preserving mapping of 3D carotid ultrasound images using density-equalizing reference map. *IEEE Transactions on Biomedical Engineering* 67, 9 (2020), 2507–2517. doi:10.1109/tbme.2019.2963783.
- [EEL*19] EULZER P., ENGELHARDT S., LICHTENBERG N., DE SIMONE R., LAWONN K.: Temporal views of flattened mitral valve geometries. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 971–980. doi:10.1109/TVCG.2019.2934337.
- [EMKL21] EULZER P., MEUSCHKE M., KLINGNER C. M., LAWONN K.: Visualizing carotid blood flow simulations for stroke prevention. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 435–446.
- [EMML22] EULZER P., MEUSCHKE M., MISTELBAUER G., LAWONN K.: Vessel maps: A survey of map-like visualizations of the cardiovascular system. *Computer Graphics Forum* 41, 3 (2022), 645–673. doi:10.1111/cg.f.14576.
- [GSK*12] GOUBERGRITS L., SCHALLER J., KERTZSCHER U., VAN DEN BRUCK N., PÖTHKOW K., PETZ C., HEGE H.-C., SPULER A.: Statistical wall shear stress maps of ruptured and unruptured middle cerebral artery aneurysms. *Journal of the Royal Society Interface* 9, 69 (2012), 677–688. doi:10.1098/rsif.2011.0490.
- [KHB*06] KUEHNEL C., HENNEMUTH A., BOSKAMP T., OELTZE S., BOCK S., KRASS S., PREIM B., PEITGEN H.-O.: New software assistants for cardiovascular diagnosis. In *Proc. of INFORMATIK 2006 – Informatik für Menschen, Band 1* (2006), Gesellschaft für Informatik e.V., pp. 491–498.
- [Kie53] KIEFER J.: Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society* 4 (1953), 502–506.
- [KMJ*14] KARIM R., MA Y., JANG M., HOUSDEN R. J., WILLIAMS S. E., CHEN Z., ATAOLLAHI A., ALTHOEFER K., RINALDI C. A., RAZAVI R., O’NEILL M. D., SCHAEFTTER T., RHODE K. S.: Surface flattening of the human left atrium and proof-of-concept clinical applications. *Computerized Medical Imaging and Graphics* 38, 4 (2014), 251–266. doi:10.1016/j.compmedimag.2014.01.004.
- [KMM*18] KREISER J., MEUSCHKE M., MISTELBAUER G., PREIM B., ROPINSKI T.: A survey of flattening-based medical visualization techniques. *Computer Graphics Forum* 37, 3 (2018), 597–624.
- [KMP*15] KÖHLER B., MEUSCHKE M., PREIM U., FISCHBACH K., GUTBERLET M., PREIM B.: 2D plot visualization of aortic vortex flow in cardiac 4D PC-MRI data. In *Prof. of Bildverarbeitung für die Medizin 2015* (2015), Handels H., Deserno T. M., Meinzer H.-P., Tolxdorff T., (Eds.), Springer Verlag, pp. 257–262. doi:10.1007/978-3-662-46224-9_45.
- [KPH*09] KASTEN J., PETZ C., HOTZ I., NOACK B. R., HEGE H.-C.: Localized finite-time Lyapunov exponent for unsteady flow analysis. In *Proc. of International Symposium on Vision, Modeling, and Visualization* (2009), pp. 265–276.
- [LER*20] LICHTENBERG N., EULZER P., ROMANO G., BRCIC A., KARCK M., LAWONN K., DE SIMONE R., ENGELHARDT S.: Mitral valve flattening and parameter mapping for patient-specific valve diagnosis.

- International Journal of Computer Assisted Radiology and Surgery* (2020), 617–627. doi:10.1007/s11548-019-02114-w.
- [MEB*17] MEUSCHKE M., ENGELKE W., BEUING O., PREIM B., LAWONN K.: Automatic viewpoint selection for exploration of time-dependent cerebral aneurysm data. In *Proc. of Bildverarbeitung für die Medizin (BVM)* (2017), Springer Verlag, pp. 352–357. doi:10.1007/978-3-662-54345-0_79.
- [MGB*19] MEUSCHKE M., GUNTHER T., BERG P., WICKENHOFER R., PREIM B., LAWONN K.: Visual analysis of aneurysm data using statistical graphics. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 997–1007. doi:10.1109/tvcg.2018.2864509.
- [MKH*12] MA Y., KARIM R., HOUSDEN R. J., GIJSBERS G., BULLENS R., RINALDI C. A., RAZAVI R., SCHAEFFTER T., RHODE K. S.: Cardiac unfold: A novel technique for image-guided cardiac catheterization procedures. In *Information Processing in Computer-Assisted Interventions*. Springer Berlin Heidelberg, 2012, pp. 104–114. doi:10.1007/978-3-642-30618-1_11.
- [MVB*17a] MEUSCHKE M., VOSS S., BEUING O., PREIM B., LAWONN K.: Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 761–770. doi:10.1109/tvcg.2016.2598795.
- [MVB*17b] MEUSCHKE M., VOSS S., BEUING O., PREIM B., LAWONN K.: Glyph-based comparative stress tensor visualization in cerebral aneurysms. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 99–108. doi:https://doi.org/10.1111/cgf.13171.
- [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. doi:10.1016/j.cag.2021.05.005.
- [MVPL18] MEUSCHKE M., VOSS S., PREIM B., LAWONN K.: Exploration of blood flow patterns in cerebral aneurysms during the cardiac cycle. *Computers & Graphics* 72 (2018), 12–25. doi:10.1016/j.cag.2018.01.012.
- [NFB07] NAVAB N., FEUERSTEIN M., BICHLMEIER C.: Laparoscopic virtual mirror new interaction paradigm for monitor-based augmented reality. In *Proc. of IEEE Virtual Reality* (2007), pp. 43–50.
- [NGB*09] NEUGEBAUER M., GASTEIGER R., BEUING O., DIEHL V., SKALEJ M., PREIM B.: Map displays for the analysis of scalar data on cerebral aneurysm surfaces. *Computer Graphics Forum* 28, 3 (2009), 895–902. doi:10.1111/j.1467-8659.2009.01459.x.
- [NGBD*19] NUÑEZ-GARCIA M., BERNARDINO G., DOSTE R., ZHAO J., CAMARA O., BUTAKOFF C.: Standard quasi-conformal flattening of the right and left atria. In *Functional Imaging and Modeling of the Heart*, Coudière Y., Ozenne V., Vigmond E., Zenzemi N., (Eds.). Springer International Publishing, 2019, pp. 85–93. doi:10.1007/978-3-030-21949-9_10.
- [NLB*13] NEUGEBAUER M., LAWONN K., BEUING O., BERG P., JANIGA G., PREIM B.: Amnvis—a system for qualitative exploration of near-wall hemodynamics in cerebral aneurysms. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 251–260. doi:https://doi.org/10.1111/cgf.12112.
- [OKG*06] OELTZE S., KUSS A., GROTHUES F., HENNEMUTH A., PREIM B.: Integrated visualization of morphologic and perfusion data for the analysis of coronary artery disease. In *Proc. of the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization* (2006), Eurographics Association, pp. 131–138.
- [PBI*17] PAUN B., BIJNENS B., ILES T., IAIZZO P. A., BUTAKOFF C.: Patient independent representation of the detailed cardiac ventricular anatomy. *Medical Image Analysis* 35 (2017), 270–287. doi:10.1016/j.media.2016.07.006.
- [PM20] PREIM B., MEUSCHKE M.: A survey of medical animations. *Computers & Graphics* 90 (2020), 145–168. doi:https://doi.org/10.1016/j.cag.2020.06.003.
- [RPM*19] RONEY C. H., PASHAEI A., MEO M., DUBOIS R., BOYLE P. M., TRAYANOVA N. A., COCHET H., NIEDERER S. A., VIGMOND E. J.: Universal atrial coordinates applied to visualisation, registration and construction of patient specific meshes. *Medical Image Analysis* 55 (2019), 65–75. doi:10.1016/j.media.2019.04.004.
- [VFSH01] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In *Proc. of the Vision Modeling and Visualization Conference* (2001), p. 273–280.