Visibility Optimization for Direct and Indirect Volume Rendering using Level Set Propagation

Paul Himmler and Tobias Günther

***Abstract*—Volumetric data arises in many scientific disciplines, describing the tissue in our body, the composition of the Earth, or the distribution of matter in the universe, to name a few. Such volume data is commonly visualized with direct methods such as ray marching, and with indirect methods such as isocontours. The combination of these approaches is fruitful to provide context and to enhance depth perception. A common challenge, however, is that three-dimensional data inherently leads to occlusions, for which visibility optimization techniques have been employed in the past. The joint visibility optimization of volumetric data (direct volume rendering) and surface geometry (indirect volume rendering) is challenging. With this paper, we provide an optimization approach in which explicit and implicit geometry, such as isocontours or closed context geometry, as well as the volume itself give way to reveal structures that have been identified as important by a user. We model the geometry implicitly as level set of a signed distance field, which is evolved under a normal flow to reduce the occlusion. The non-linear optimization involves gradient descent solvers, level set propagation, and multi-grid optimization. We compare our approach to previous visibility optimizations.**

***Index Terms*—scientific visualization, volume rendering, visibility optimization, level set, signed distance field**

I. Introduction

DIRECT and indirect volume rendering are commonly-used visualization techniques for the display of three-dimensional scalar fields [1]–[4]. A long-standing challenge is the effective communication of relevant parts of a data set, which requires careful adjustment of the transparency to resolve occlusions [5]–[9]. In conventional direct volume rendering, the transparency is determined by a transfer function, which was optimized by early approaches [7]. Which parts of a volume cause occlusions, however, is not only a matter of the scalar value, but also a view-dependent problem. Hence, later approaches treated the extinction coefficient as a spatial field [8], [9], which can be interpreted as the optimization of a transfer function that takes the spatial location as input and not the scalar value. Recently, Himmler and Günther [9] introduced a differentiable visibility optimization that optimizes the extinction field and the camera position for direct volume rendering of continuous scalar fields. In this paper, we extend this approach to scenes containing closed opaque explicit or implicit geometry. Examples are isosurfaces as well as context geometry, such as the Earth's surface shown in Fig. 1. The surfaces provide context and depth cues, but unfortunately, they also cause occlusions. Rather than adjusting the transparency of the surfaces [10], [11], which can inhibit depth perception, or creating a cut away, we instead deform the surfaces to give way onto the relevant parts of the volume. Implicit geometry representations such as level sets are thereby useful, as they automatically handle changes in topology. We model the surface geometry as isocontours in a signed distance field and utilize a normal flow to deform the implicit geometry, such that the view is cleared onto relevant structures of the volume. Thereby modeling the surfaces with a finite thickness, as demonstrated in Fig. 2, gives rise to a shaded boundary that conveys shape information. We combine our novel optimization of the implicit geometry with the existing extinction optimization of the direct volume rendering [9], resulting in an optimization that clears the view onto relevant structures. Both individual optimizations are based on variational principles. However, the visibility

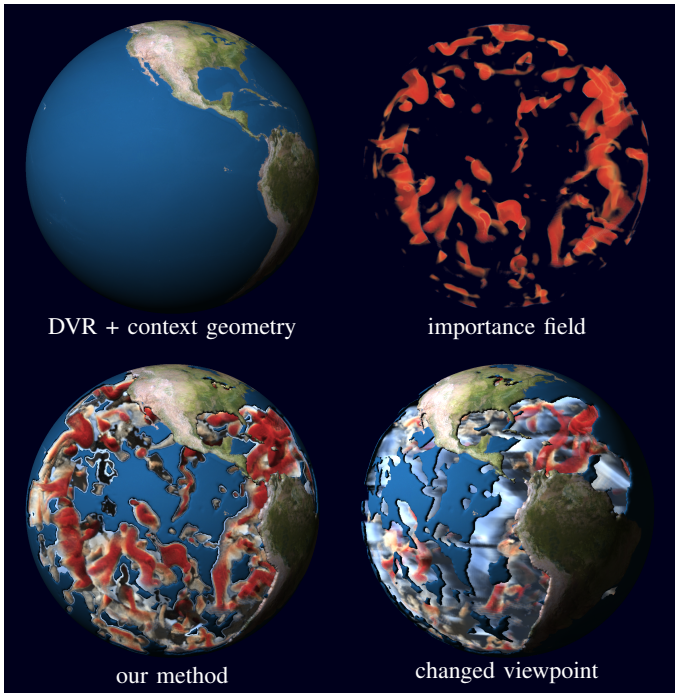


Fig. 1. Adding surface geometry to a direct volume rendering can provide meaningful context. In the top left image, however, the context geometry occludes the entire volume. The top right image shows a user-defined importance field that indicates which parts of the scene should be made visible. In the bottom left image, our visibility optimization deforms the context geometry via level set propagation in a signed distance field, and adjusts the extinction of the volume rendering to clear the view. Thereby, user-defined parts of the context geometry can be preserved, such as the continents. For explanatory purposes, the bottom right image views the same scene from a different viewpoint.

Paul Himmler and Tobias Günther are with the Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany.

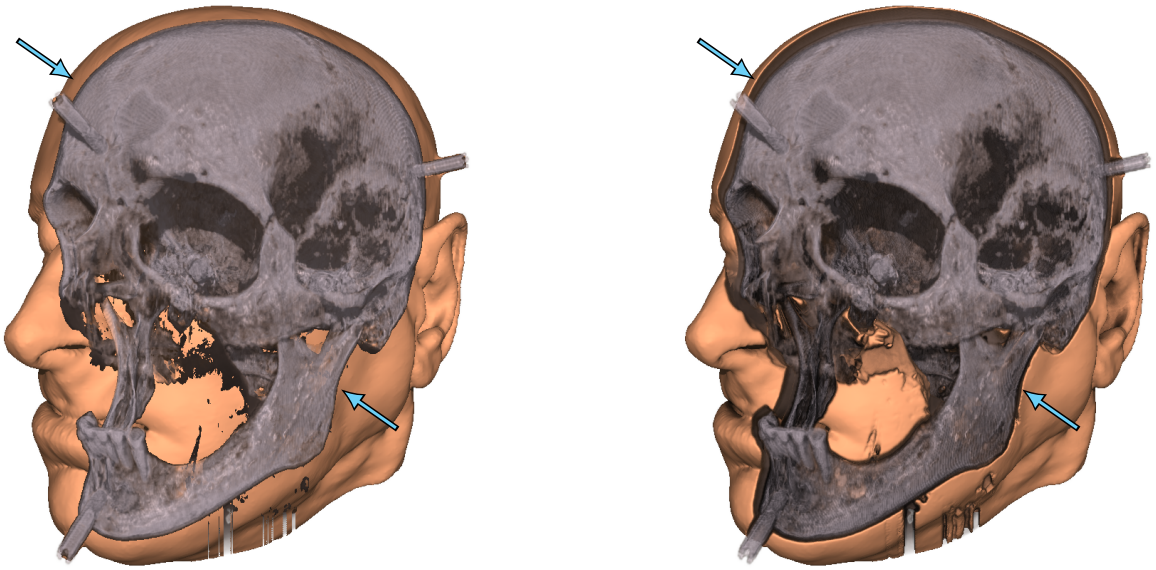


Fig. 2. Compared to cut aways of occluding geometry (left), our modeling of context geometries as level sets with a finite thickness (right) gives rise to a shaded border that helps to convey shape information. Note that on the left, it is difficult to tell whether the skull (gray) or the skin (orange) is in the foreground. On the right, the shaded border (see arrows) offers a depth cue that helps disambiguating the depth order.

optimization of Himmler and Günther [9] has a high memory consumption. Since we additionally require signed distance fields for the representation of context geometry, we evaluate the memory and performance implications for both 32-bit and 16-bit floating point precision. Further, we incorporate ambient occlusion in order to improve the depth perception. In summary, this paper makes the following contributions:

- We present a visibility optimization of explicit and implicit geometry by using level set propagation in a signed distance field, such that the isocontours give way to reveal relevant structures.
- We propose a visibility optimization that jointly optimizes the surface geometry (indirect volume rendering) and the extinction field of the volume (direct volume rendering) at an interactive frame rate to enable continuous exploration.
- We evaluate the memory, performance, and quality implications for both 32-bit and 16-bit floating point precision, in order to make room for the context geometry optimization and larger data sets.
- We give users interactive control over which parts of the context geometry to keep, which supports preservation of depth cues.

In Section II, we begin with a brief introduction of mathematical basics to set the notation, which is followed by a summary of related work on visibility optimization. Section III derives our level set-based visibility optimization for implicitly defined context geometry, which is afterwards combined with a visibility optimization of the extinction field for a combined direct and indirect volume rendering. The method is evaluated qualitatively and quantitatively in Section IV, including comparisons with previous work, a parameter study, performance measurements, a user study, and a discussion. Lastly, we conclude the paper in Section V with an outlook onto possible avenues for future work.

II. Background and Related Work

First, we summarize related work on variational optimization and on visibility optimization. Afterwards, we introduce the light transport model and the extinction optimization [9].

A. Variational Optimization

Variational calculus [12] is a tool that allows for the optimization of differentiable functions f_i that minimize a functional $\mathcal{F}$, which integrates a Lagrangian $\mathcal{L}$ over the domain $\mathbb{X}$:

$$\mathcal{F}[f_1, \ldots, f_m] = \int_{\mathbb{X}} \mathcal{L}\left(x_1, \ldots, x_n, f_1, \ldots, f_m, \frac{\partial f_1}{\partial x_1}, \ldots, \frac{\partial f_m}{\partial x_n}\right) \mathrm{d}\mathbf{x}, \tag{1}$$

A necessary condition is given by the Euler-Lagrange equations, which state that the functional derivatives $\frac{\delta \mathcal{F}}{\delta f_i}$ must vanish to zero:

$$\frac{\delta \mathcal{F}}{\delta f_i} = \frac{\partial \mathcal{L}}{\partial f_i} - \sum_{j=1}^{n} \frac{\partial}{\partial x_j}\left(\frac{\partial \mathcal{L}}{\partial f_{i,j}}\right) = 0, \tag{2}$$

with $f_{i,j} := \frac{\partial f_i}{\partial x_j}$ being the partial derivative of a function f_i with respect to a parameter x_j. Variational methods have found wide application in computer vision [13]–[16], for example for optical flow estimation [17], [18], image restoration [19], [20], and image segmentation [21], [22]. In geometric modeling, variational methods have been applied among others for surface modeling [23], mesh reconstruction [24], [25], and surface cutting [26]. In fluid simulation, the approach was used to minimize pressure [27], and to represent a flow using vortex filaments with a given filament strength [28]. In visualization, the variational approach was proposed for feature extraction [29], and it was used for visibility optimization in continuous volumetric data [9]. In this paper, we use a variational formulation to optimize for the scalar field that implicitly describes the deformed context geometry.

B. Related Work on Visibility Optimization

Visibility optimization provides a clearer view onto structures of interest, which has many facets, such as the perceptually-oriented optimization of transfer functions [6], viewpoints [30], clipping [31], ghosting [32], and the identification of what is important [33].

View-independent Transfer Functions: He et al. [34] evaluated the quality of transfer functions with metrics such as entropy, histogram variance, and edge energy, and let users choose interactively from stochastically mutated transfer functions. Kindlmann and Durkin [35] emphasized material boundaries by forming transfer functions semi-automatically from the scalar field and its first and second-order directional derivatives along the gradient. Wu and Qu [36] reconstructed the transfer function from user-selected features in multiple volume renderings that were generated with different transfer functions, allowing addition and removal. Ruiz et al. [37] minimized the informational divergence of the visibility distributions between data and view. Since occlusion is a view-dependent problem, we aim for a view-dependent adjustment of the transfer function, which depends not only on the scalar value but also on the position in space. In the following, we summarize work on view-dependent formulations.

View-dependent Transfer Functions: Viola et al. pioneered the view-dependent adjustment of opacity [5], showing the object with highest importance along the view ray [38], [39], and offering windowing functions to retain occlusion depth cues from the foreground. Krüger et al. [40] used an interactive lens to reveal important regions behind context geometry. Chan et al. [6] incorporated the human perception into the optimization of discrete opacity values along view rays, considering visibility, structural shape, and image variations. Marchesin et al. [41] adjusted the blending weights of color values along the view ray by maximizing the contribution of the farthest relevant sample point. To support transfer function editing Correa and Ma [7] displayed visibility histograms behind the transfer function, and optimized a Gaussian mixture model representation of the opacity. For line and surface geometry, Günther et al. [11], [42], [43] performed a constrained linear optimization that determines the opacity of line segments and surface patches such that occlusion of relevant information and background clutter are removed. Ament et al. [8] reformulated this in view-space (and light-space) for volumetric data, which led to a closed-form solution if smoothing was performed in a post-process. Recently, Lerzer and Dachsbacher [44] applied visibility optimization in Monte Carlo volume rendering. The idea of Ament et al. [8] has later been applied for geometric data [45], as well, which was further scaled up for larger scenes [46], [47]. Recently, opacity optimization has been combined with color optimization [48] to utilize the full range of the color map, which was inspired from earlier work of Waldin et al. [49] on the multi-scale adjustment of colormaps. Returning back to direct volume rendering, Himmler and Günther [9] used the transmittance to compute the contribution of a voxel to the image and optimized the extinction field and the viewpoint in a variational manner. To the best of our knowledge, no visibility optimization has optimized direct and indirect volume rendering concurrently. Thus, we present a novel level set-based optimization for geometric data, which we combine with extinction optimization [9] for volume data.

Importance Measures and Viewpoints: The foundation of visibility optimizations is the specification of what is important in a scene. A common choice is to let users specify the relevant structures by a scalar field [8], [42]. An automatic alternative is to employ entropy measures [33], [50], which had for example applications in volume visualization [51] and flow visualization [52]–[54]. Takahashi et al. [30] took a bottom-up approach, finding locally optimal viewpoints for parts of the domain, from which a globally optimal viewpoint was found as a compromise. Ji et al. [55] proposed a number of view-dependent metrics, including the even distributions of opacity, salient features, and curvature, which has been used in conjunction with differentiable volume rendering [56] for viewpoint optimization. Tao et al. [57] distinguished between local and global features, accounting for details on boundary structures, and the overall orientation. Leveraging the experience of experts, Tao et al. [58] used image similarity metrics to find views similar to those previously chosen by experts. Similarity metrics have also been estimated by deep learning techniques [59]. Himmler and Günther [9] evaluated the transmittance at user-defined regions of interest to measure how well relevant structures are seen, which guided a non-linear viewpoint optimization. In this work, we adopt their metric to geometric data.

Spatial Deformation: Similar to illustrations in assembly instructions, Agrawala et al. [60] constructed exploded views. Bruckner and Gröller [61] proposed a force and constraint-based system to generate exploded views for volume data. Correa and Ma [62] formulated a taxonomy of illustrative spatial deformation techniques, and applied focus and context methods to flow visualizations, tractography, and node-link diagrams. Using fish-eye deformations in a user-defined lens, Traoré et al. [63] created unobstructed views for interactive exploration in direct volume rendering. To reveal hidden scalar fields on the backside of surface geometries, Lawonn et al. [64] used curved sphere tracing to follow bent view rays. In this work, we deform signed distance field representations of the context to clear the view onto relevant structures.

C. Transmittance-based Visibility Optimization

We combine visibility optimization for direct and indirect volume rendering. For the direct volume rendering, i.e., the optimization of the extinction field, we employ the transmittance-based visibility optimization of Himmler and Günther [9]. In the following, we briefly summarize their approach, as well as the modifications needed to add surface geometry. First, we introduce quantities needed to express visibility along a ray.

Extinction: The extinction coefficient $\mu_t(\mathbf{x})$ models how much light is lost per unit distance, which will later be an unknown of the optimization [9]. The extinction coefficient is the sum of the scattering coefficient $\mu_s(\mathbf{x})$ and the absorption coefficient $\mu_a(\mathbf{x})$, i.e., $\mu_t(\mathbf{x}) = \mu_s(\mathbf{x}) + \mu_a(\mathbf{x})$. The scattering albedo is the ratio (μ_s/μ_t), which is in $[0, 1]$. In phenomenological volume rendering [1], the scattering albedo

arises from a transfer function and is commonly specified per color channel as $\mathbf{a}(\mathbf{x}) : \mathcal{D} \to [0,1]^3$:

$$\frac{\mu_s(\mathbf{x})}{\mu_t(\mathbf{x})} = \mathbf{a}(\mathbf{x}). \tag{3}$$

In physically-based rendering, $\mu_s(\mathbf{x})$ and $\mu_a(\mathbf{x})$ are specified independently as scalar fields and may depend on wavelength.

Transmittance: The fraction of light that arrives after traveling along a line segment is denoted as transmittance. To ease the integration in view space and light space, Himmler and Günther [9] described the view space transmittance $T_V(\mathbf{y}) : \mathcal{Y} \to [0,1]$ and the light space transmittance $T_L(\mathbf{z}) : \mathcal{Z} \to [0,1]$ in their respective coordinates using the perspective view space transformation $\mathbf{g}_V(\mathbf{x}) : \mathcal{D} \to \mathcal{Y}$ and the perspective light space transformation $\mathbf{g}_L(\mathbf{x}) : \mathcal{D} \to \mathcal{Z}$:

$$T_V(\mathbf{y}) = e^{-\int_{\mathbf{y}_0}^{\mathbf{y}} \mu_t(\mathbf{g}_V^{-1}(\mathbf{y}'))\,\mathrm{d}\mathbf{y}'} \tag{4}$$

$$T_L(\mathbf{z}) = e^{-\int_{\mathbf{z}_0}^{\mathbf{z}} \mu_t(\mathbf{g}_L^{-1}(\mathbf{z}'))\,\mathrm{d}\mathbf{z}'} \tag{5}$$

with the camera being placed at $\mathbf{y}_0$ and the light source being placed at $\mathbf{z}_0$. Due to the viewpoint optimization in [9], their camera position was restricted to be on a sphere that orbits the domain, which is not a requirement for us. Similar to the transmittance T_V, T_L, Himmler and Günther [9] further computed visibility-weighted importance integrals *behind* a given point in view space and light space, called G_V and G_L.

Radiative Transfer Equation: To describe the image synthesis process, Himmler and Günther [9] utilized the radiative transfer equation, which describes in its integral form the incident radiance $L(\mathbf{x}_0 \leftarrow \omega)$ that is seen at point $\mathbf{x}_0$ coming in from direction ω [65]:

$$L(\mathbf{x}_0 \leftarrow \omega) = \underbrace{L_s(\mathbf{x}_0 \leftarrow \omega)}_{\text{accumulated in-scattered}} + \underbrace{L_r(\mathbf{x}_0 \leftarrow \omega)}_{\text{reduced surface radiance}}. \tag{6}$$

The radiance $L(\mathbf{x}_0 \leftarrow \omega)$ consists of the accumulated in-scattered radiance $L_s(\mathbf{x}_0 \leftarrow \omega)$, which integrates the radiance that is reflected towards the viewer along the view ray from entry $\mathbf{x}_0$ to exit $\mathbf{x}_s$:

$$L_s(\mathbf{x}_0 \leftarrow \omega) = \int_{\mathbf{x}_0}^{\mathbf{x}_s} T_V(\mathbf{g}_V(\mathbf{x})) \cdot f_p(\mathbf{x}, \omega_L \leftrightarrow \omega) \cdot \mu_s(\mathbf{x}) \cdot L_i(\mathbf{x} \leftarrow \omega)\,\mathrm{d}\mathbf{x}. \tag{7}$$

The phase function $f_p(\mathbf{x}, \omega_L \leftrightarrow \omega)$ describes how much incoming light from direction ω_L is being scattered into the outgoing direction ω (and vice versa). Following Ament et al. [8], we combine normalized Blinn-Phong shading and Henyey-Greenstein forward-scattering [66]. In phenomenological volume rendering, the scattering coefficient $\mu_s(\mathbf{x})$ is determined component-wise by rearranging the ratio in Eq. (3) to $\mu_s(\mathbf{x}) = \mu_t(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x})$. Further, the reduced surface radiance $L_r(\mathbf{x}_0 \leftarrow \omega)$ in Eq. (6) accounts for the exitant surface radiance that is emitted from a surface point at $\mathbf{x}_s = \mathbf{x}_0 - s \cdot \omega$:

$$L_r(\mathbf{x}_0 \leftarrow \omega) = T_V(\mathbf{g}_V(\mathbf{x})) \cdot f_r(\mathbf{x}, \omega_L \leftrightarrow \omega) \cdot (\mathbf{n} \cdot \omega_L)_+ \cdot L_i(\mathbf{x} \leftarrow \omega). \tag{8}$$

We employ the normalized Blinn-Phong reflection model for the BRDF $f_r(\mathbf{x}, \omega_L \leftrightarrow \omega)$. In Eq. (8), $L_i(\mathbf{x} \leftarrow \omega)$ determines the amount of light incoming at point $\mathbf{x}$ and scattered in direction ω. For a point light with single-scattering that emits a radiance L_e, this is given by:

$$L_i(\mathbf{x} \leftarrow \omega) = T_L(\mathbf{g}_L(\mathbf{x})) \cdot L_e. \tag{9}$$

For visualization purposes, we omit the quadratic fall-off of the point light intensity. While Himmler and Günther [9] considered only the accumulated in-scattered radiance in Eq. (7), we also include the reflected surface radiance in Eq. (8).

Energy: Inspired from extinction optimization [8], a variational energy was decomposed into multiple terms [9]:

$$\mathcal{F}[\mu_t(\mathbf{x}), \mathbf{g}_V(\mathbf{x})] = \mathcal{F}_p + \mathcal{F}_q + \mathcal{F}_r \tag{10}$$

with each individual term serving a specific task:

$$\mathcal{F}_p = \frac{p}{2} \int_{\mathcal{D}} (\mu_t(\mathbf{x}) - \overline{\mu}_t(\mathbf{x}))^2\,\mathrm{d}\mathbf{x} \tag{11}$$

$$\mathcal{F}_q = -q \int_{\mathcal{D}} g(\mathbf{x}) \cdot \mu_t(\mathbf{x}) \cdot T_V(\mathbf{g}_V(\mathbf{x}))\,\mathrm{d}\mathbf{x} \tag{12}$$

$$\mathcal{F}_r = -r \int_{\mathcal{D}} g(\mathbf{x}) \cdot \mu_t(\mathbf{x}) \cdot T_L(\mathbf{g}_L(\mathbf{x}))\,\mathrm{d}\mathbf{x} \tag{13}$$

and where the extinction $\mu_t(\mathbf{x})$ is subject to the constraint:

$$0 \leq \mu_t(\mathbf{x}) \leq \widetilde{\mu}_t : \forall \mathbf{x} \in \mathcal{D}. \tag{14}$$

Eq. (11) lets the extinction field $\mu_t(\mathbf{x})$ strive to be similar to the base extinction $\overline{\mu}_t(\mathbf{x})$, which results in a standard volume rendering if the remaining terms do not act. Eq. (12) penalizes the occlusion of relevant structures by increasing the extinction and the transmittance. And lastly, Eq. (13) does the same from the perspective of the light source to ensure that relevant structures are well lit. The extinction field is bounded using Eq. (14) to prevent invalid minimizers. Inserting the energy from Eq. (10) into the Euler-Lagrange equations in Eq. (2) results in a gradient for an iterative optimization, to which we apply the AdEMAMix optimizer [67], which adds another exponential moving average to the Adam optimizer [68] to make better use of past gradients. In our experiments, we observed a slightly improved convergence speed compared to the Adam method [68] used in [9]. We refer to [9] for the derivation of necessary conditions for minimizing Eq. (10) and for the iterative optimization procedure.

III. Visibility Optimization for Direct and Indirect Volume Rendering

Adding explicit or implicit surface geometry to a direct volume rendering allows us to provide context, such as for the display of the planet's surface in Fig. 1. Usually, this would lead to unwanted occlusion, which could be addressed in different ways. One option would be to make the surfaces transparent [10], [11], which, however, would inhibit the depth perception since both the volume and the surfaces visually blend together. Instead, we made the design decision to keep the surfaces opaque, which usually requires careful hints about hidden geometry [69], [70] or surface clipping to look into the volume. Since manual clipping is time-consuming and view-dependent, we aim for an automatic method that adjusts which

parts of the surface to show. For our interactive system, we therefore set the following requirements:

- **R1.** optimize the extinction field according to [9], including the joint optimization from view and light direction,
- **R2.** optimize the visibility of context geometry by displacing it, supporting both thin shells and filled geometry,
- **R3.** reduce memory consumption to store implicit geometry and support volumes at higher resolution than [9],
- **R4.** add depth cues to support scene understanding, including shading, shadows, and ambient occlusion,
- **R5.** allow for interactive removal or preservation of context geometry through masks and lenses [39], [40].

In summary, we extend the visibility optimization for direct volume rendering [9] by including an optimization of both explicitly and implicitly defined context geometry, by improving the performance, by enhancing the rendering for improved depth perception, and by adding interactive tools for the removal or preservation of context geometry.

A. Problem Statement

The input to our system is a scalar field $s(\mathbf{x}) : \mathcal{D} \to \mathbb{R}$ in the spatial domain $\mathcal{D} \subset \mathbb{R}^3$ that the user wants to explore. In addition, the user is required to provide an importance scalar field $g(\mathbf{x}) : \mathcal{D} \to [0,1]$, which specifies for each voxel of the domain how important it is to see. Further, a transfer function is provided, which produces a scattering albedo field $\mathbf{a}(\mathbf{x}) : \mathcal{D} \to [0,1]^3$ and a base extinction field $\overline{\mu}_t(\mathbf{x}) : \mathcal{D} \to \mathbb{R}^+$. If the optimization is disabled, both $\mathbf{a}(\mathbf{x})$ and $\overline{\mu}_t(\mathbf{x})$ together result in a regular direct volume rendering. To resolve the occlusion of relevant structures in the domain, an extinction field $\mu_t(\mathbf{x})$ is optimized interactively using the volume visibility optimization of Himmler and Günther [9], which supports (**R1**). The novelty of our approach is that the user further specifies a set of closed surfaces, which may be extracted from the scalar field, or they may represent context geometry. Collectively, we denote those surfaces with a subset of the domain $\mathcal{C} \subset \mathcal{D}$. Rather than showing the surfaces $\mathcal{C}$ directly, which would lead to occlusion, we precompute a new scalar field ϕ, which is a signed distance field (SDF) and in which the surfaces $\mathcal{C}$ appear as zero level set. At runtime, the field ϕ is adjusted by an energy minimization that evolves the level set in a normal flow to remove the occlusions. The concept is illustrated in Fig. 3. Upon interactive navigation of the camera, both the volume and the context geometry give way to reveal relevant structures in the domain.

B. Visibility Optimization of Implicit Geometry

In the following, we derive a visibility optimization algorithm that displaces implicit geometry in order to clear the view onto relevant structures, which meets (**R2**). The algorithm is applicable with and without direct volume rendering.

Signed Distance Field: Given are the closed surfaces $\mathcal{C}$, which we first represent implicitly as zero level set in a signed distance field $\phi_0(\mathbf{x}) : \mathcal{D} \to \mathbb{R}$. Such an implicit representation is advantageous, since it allows deformations without explicitly tracking changes in topology. The signed distance field

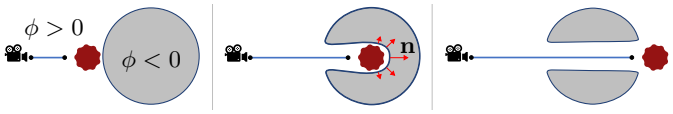


Fig. 3. Consider a relevant volumetric object (shown in red) and an implicitly defined closed context geometry, defined via a signed distance function $\phi = 0$. Left: if the important object is in front of the context, the isocontour does not have to change, as it is not occluding the view. Middle: if the important structure is nested inside the context geometry, the level set evolves in the inward pointing normal direction $\mathbf{n}$ to clear the view onto the relevant structure. Right: if the important structure is behind the context geometry, the occluding part of the level set shrinks from the front and the back until the isocontour splits into two parts, revealing the interesting structure behind. The level set adjusts interactively to the changing camera position and returns to its initial shape if the occlusion is no longer present.

$\phi_0(\mathbf{x})$ is precomputed, fulfills the Eikonal equation, and meets a Dirichlet condition at the surfaces $\mathcal{C}$:

$$\left\| \frac{\partial \phi_0(\mathbf{x})}{\partial \mathbf{x}} \right\| = 1, \qquad \text{s.t. } \phi_0(\mathbf{x}) = 0 \ : \ \forall \mathbf{x} \in \mathcal{C}. \tag{15}$$

By convention, we let the interior be negative ($\phi_0 < 0$) and the exterior be positive ($\phi_0 > 0$). Later on, we refer to this as the base SDF $\phi_0(\mathbf{x})$, which the deforming SDF strives to return to in the absence of occlusions in order to recover the original surfaces. During the optimization, the SDF evolves, which is modeled with a time variable t. For a temporally-evolving SDF $\phi(\mathbf{x}, t) : \mathcal{D} \times \mathbb{R} \to \mathbb{R}$, the differential properties of an isocontour at time t are easily calculated from the spatial gradient of $\phi(\mathbf{x}, t)$, i.e., from $\nabla\phi := \frac{\partial \phi(\mathbf{x},t)}{\partial \mathbf{x}}$ we get the inward pointing normal $\mathbf{n}$ and the curvature κ of the isocontour passing through at $(\mathbf{x}, t)$, cf. Caselles et al. [71]:

$$\mathbf{n}(\mathbf{x}, t) = -\frac{\nabla\phi}{\|\nabla\phi\|}, \qquad \kappa(\mathbf{x}, t) = \nabla \cdot \left(\frac{\nabla\phi}{\|\nabla\phi\|} \right). \tag{16}$$

with $\nabla\cdot$ being the divergence operator. To control the evolution of the level set, it is necessary to model how any given point of the level set moves. To identify a specific point explicitly, let $\mathbf{c}(x, t) : [0, 1] \to \mathcal{D}$ be an explicit parametric representation of a zero level set at time t. Osher and Sethian [72][1] have shown that when moving any given point $\mathbf{c}(x, t)$ in normal direction $\mathbf{n}$ with a given velocity magnitude $\nu(\mathbf{x}, t)$, then this motion can likewise be expressed in the level set formulation directly without ever explicitly computing an explicit representation:

$$\frac{\partial \mathbf{c}(x,t)}{\partial t} = \nu\, \mathbf{n}(\mathbf{c}(x,t), t) \quad \Leftrightarrow \quad \frac{\partial \phi(\mathbf{x},t)}{\partial t} = \nu \|\nabla\phi\|. \tag{17}$$

Thus, a level set $\mathbf{c}$ can be evolved directly through ϕ using the PDE in Eq. (17) with the initial condition $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$.

Redistancing: As can be seen from Eq. (17), the motion of the level set depends on $\|\nabla\phi\|$. To maintain a controllable velocity, it is common to perform a level set redistancing, such that the Eikonal equation remains fulfilled, i.e., $\|\nabla\phi\| = 1$, making $\phi(\mathbf{x}, t)$ again a signed distance field. Several solvers are available for computing a signed distance function from an

[1] Osher and Sethian [72] derived the relation between the explicit and implicit curve evolution starting from their Eq. (2.1), which is the left side of our Eq. (17), to (2.13), which is the right side of our Eq. (17), for ν depending on κ. Caselles et al. [71, Appendix C] have shown this for general ν. Moving a level set point in direction $(\nu\mathbf{n})$, the level set value does not change: $\nabla\phi \cdot (\nu\mathbf{n}) + \phi_t = 0$. Inserting normal $\mathbf{n}$ from Eq. (16) gives $\phi_t = \nu\|\nabla\phi\|$.

isocontour, such as fast marching [73, Sec. 7.4]. We use the redistancing formulation of Sussman et al. [74], which solves the following PDE for its steady state by evolving $\phi(\mathbf{x},t)$ according to:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} + S(\phi_0(\mathbf{x}))(\|\nabla\phi\| - 1) = 0, \tag{18}$$

with $S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2+(\Delta x)^2}}$, where the gradient norm $\|\nabla\phi\|$ is estimated using Godunov's method [73, Sec. 5.3.3] (which is an upwind scheme), where $\phi_0(\mathbf{x})$ is the initial value, and where Δx is the spatial grid spacing. The isocontour of a given scalar field $s(\mathbf{x})$ can be brought into an SDF representation by setting the initial condition $\phi(\mathbf{x},0) = s(\mathbf{x})$ or $\phi(\mathbf{x},0) = -s(\mathbf{x})$, respectively, depending on whether the interior has higher or lower values than the isocontour, and by then solving Eq. (18) for its steady state. There are a number of other options available for computing an SDF, such as sweeping methods [75], [76].

Minimization Problem: The zero level set of $\phi(\mathbf{x},t)$ at time t is a surface that occludes relevant regions in the scene, which are identified by the importance field $g(\mathbf{x})$. For a given level set, we can measure how important the structures are that are occluded by the level set. Since we want the most-relevant structures to appear, we determine the largest importance $\widehat{g}(\mathbf{x})$ along the view ray and the light ray behind a given point $\mathbf{x}$:

$$\begin{aligned}\widehat{g}(\mathbf{x}) =& \frac{1}{2}\underbrace{\max_{t_V\in[0,1]} g(\mathbf{x}+t_V\cdot\mathbf{d}_V(\mathbf{x}))}_{\text{view ray}} \\ &+\frac{1}{2}\underbrace{\max_{t_L\in[0,1]} g(\mathbf{x}+t_L\cdot\mathbf{d}_L(\mathbf{x}))}_{\text{light ray}}\end{aligned} \tag{19}$$

where $\mathbf{d}_V(\mathbf{x}) = \mathbf{g}_V^{-1}(\mathbf{y}_1) - \mathbf{x}$ is the view ray direction from $\mathbf{x}$ to the exit of the domain $\mathbf{y}_1$, cf. Eq. (4), and $\mathbf{d}_L(\mathbf{x}) = \mathbf{g}_L^{-1}(\mathbf{z}_1) - \mathbf{x}$ is analogous for the light ray. With this, we obtain an importance score per point $\mathbf{x}$ that is bounded by the value range of $g(\mathbf{x})$. We utilize the geodesic active contour energy of Caselles et al. [71, Eq. 8], with which we integrate the largest occluded importance with arclength parameterization around the level set, here expressed explicitly:

$$\mathcal{F}_i[\mathbf{c}] = \int_0^1 \widehat{g}(\mathbf{c})\cdot\|\dot{\mathbf{c}}\|\,\mathrm{d}x \tag{20}$$

Inserting Eq. (20) into the Euler-Lagrange equations in Eq. (2) leads to the following necessary condition, cf. [71, Eq. 13]:

$$\frac{\delta\mathcal{F}_i}{\delta\mathbf{c}} = \widehat{g}(\mathbf{c})\kappa\mathbf{n} - \left(\frac{\partial\widehat{g}(\mathbf{c})}{\partial\mathbf{x}}\cdot\mathbf{n}\right)\mathbf{n}. \tag{21}$$

This is analogously phrased in the implicit level set representation as was shown by Caselles et al. [71, Eq. 14], here adapted for our metric in Eq. (19):

$$\frac{\partial\phi'(\mathbf{x},t)}{\partial t} = \widehat{g}(\mathbf{x})\|\nabla\phi(\mathbf{x},t)\|\kappa + \frac{\partial\widehat{g}(\mathbf{x})}{\partial\mathbf{x}}\cdot\nabla\phi. \tag{22}$$

Solving Eq. (22) to a steady state evolves the level set in a normal flow to avoid occlusion. However, the level set is allowed to propagate outwards, which would inflate the context geometry. To ensure that occlusions may only drive the level set inwards, and to allow the level set to return to its base SDF ϕ_0 in the absence of occlusion, we employ the following PDE, consisting of two terms:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \max\left(0, \frac{\partial\phi'(\mathbf{x},t)}{\partial t}\right) + \lambda(\mathbf{x})\cdot(\phi_0(\mathbf{x}) - \phi(\mathbf{x},t)). \tag{23}$$

where $\lambda(\mathbf{x}) = \max(0, \lambda - \widehat{g}(\mathbf{x}))$ applies the return to the base SDF only if no relevant information is occluded, which is thresholded by a λ parameter that is studied later in Section IV-C. Eq. (23) constitutes our final PDE which can be solved to a steady state from the initial condition $\phi(\mathbf{x},0) = \phi_0(\mathbf{x})$. Although not explicitly used later, it is interesting to note that the level set evolves in a normal flow using Eq. (17) with the velocity magnitude $\nu(\mathbf{x},t)$:

$$\nu = \max\left(0, \operatorname{div}\left(\widehat{g}(\mathbf{x})\frac{\nabla\phi}{\|\nabla\phi\|}\right)\right) + \lambda(\mathbf{x})\cdot\frac{\phi_0(\mathbf{x}) - \phi(\mathbf{x},t)}{\|\nabla\phi\|}. \tag{24}$$

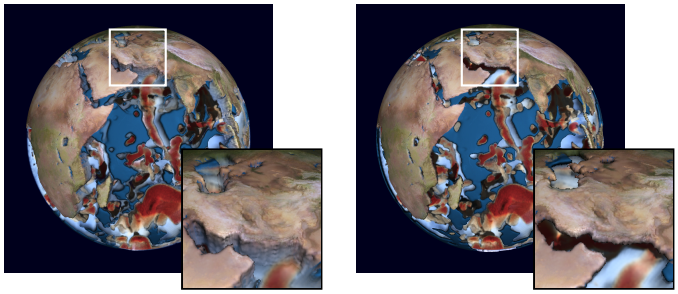
Time Stepping: The signed distance field $\phi(\mathbf{x},t)$ is discretized spatially onto a grid with values $\phi_{i,j,k}(t)$ at $\mathbf{x}_{i,j,k}$. We evolve the hyperbolic PDE in Eq. (23) with an explicit time stepping scheme:

$$\phi_{i,j,k}(t+\Delta t) = \phi_{i,j,k}(t) + \Delta t\cdot\frac{\partial\phi(\mathbf{x}_{i,j,k},t)}{\partial t}. \tag{25}$$

Filled vs. Thin Shells: The level set definition in Eq. (15) places the initial zero level set $\phi_0(\mathbf{x}) = 0$ directly on the given contours $\mathcal{C}$. This results in a setting where the entire interior of the isocontour receives a negative distance value, which we refer to as a *filled surface*. Depending on the scene, it may be useful to generate a *thin shell surface* with a finite thickness d around the given contours $\mathcal{C}$. This is achieved by transforming $\phi_0(\mathbf{x})$ as follows:

$$\overline{\phi}_0(\mathbf{x}) = |\phi_0(\mathbf{x})| - d. \tag{26}$$

While the thin shells reveal more volume, the filled surfaces may better preserve the geometric context. Both options are visualized in Fig. 4 for the EARTH MANTLE data set.



filled surface thin shell

Fig. 4. The difference between our optimization with context geometry that is modeled as a filled surface (left) in comparison to a thin shell (right).

C. Joint Visibility Optimization of Volumes and Surfaces

Next, we discuss the combination of (**R1**) and (**R2**). The iterative visibility optimization of the extinction volume in

Eq. (10) using [9, Eq. (30)] and the iterative propagation of the level set in Eq. (23) are alternated in order to solve for both $\mu_t(\mathbf{x})$ and $\phi(\mathbf{x})$ concurrently, i.e., to find the steady state solutions. After each step, the signed distance field is redistanced using N iterations of Eq. (18). Conservatively, we empirically set $N = 64$, although convergence is reached sooner in most scenes. Following [9], we employ a coarse-to-fine multi-grid solver that first performs the optimization on lower resolution, then upsamples it, and proceeds on higher resolution until the target resolution is reached. A numerical implementation of Eq. (23) is provided in Alg. 1.

Input: $\mathbf{x}$, ϕ, ϕ_0, $\widehat{g}$, λ, Δx, Δy, Δz, Δt, ϵ

$G \leftarrow \widehat{g}(\mathbf{x})$;

$\kappa \leftarrow \frac{(\phi_x^2+\phi_y^2)\phi_{zz}+(\phi_y^2+\phi_z^2)\phi_{xx}+(\phi_x^2+\phi_z^2)\phi_{yy}-2\phi_x\phi_{xy}\phi_y-2\phi_x\phi_{xz}\phi_z-2\phi_y\phi_{yz}\phi_z}{\|\phi_x^2+\phi_y^2+\phi_z^2\|^3+\epsilon}$;

if $G\kappa \leq 0$ **then**

$\|\nabla\phi\| \leftarrow \max(\phi_{x^-}^2 H(\phi_{x^-}), \phi_{x^+}^2 H(-\phi_{x^+}))$
$+ \max(\phi_{y^-}^2 H(\phi_{y^-}), \phi_{y^+}^2 H(-\phi_{y^+}))$
$+ \max(\phi_{z^-}^2 H(\phi_{z^-}), \phi_{z^+}^2 H(-\phi_{z^+}))$;

else

$\|\nabla\phi\| \leftarrow \max(\phi_{x^-}^2 H(-\phi_{x^-}), \phi_{x^+}^2 H(\phi_{x^+}))$
$+ \max(\phi_{y^-}^2 H(-\phi_{y^-}), \phi_{y^+}^2 H(\phi_{y^+}))$
$+ \max(\phi_{z^-}^2 H(-\phi_{z^-}), \phi_{z^+}^2 H(\phi_{z^+}))$;

end

$\Delta\phi \leftarrow \max(0, G\kappa\|\nabla\phi\| + G_x\phi_x + G_y\phi_y + G_z\phi_z) + \lambda(\mathbf{x}) \cdot (\phi_0 - \phi^{(i)})$;

$\phi^{(i+1)} \leftarrow \phi^{(i)} + \Delta t \cdot \Delta\phi$;

Algorithm 1: Upwind integration scheme for Eq. (23) using Godunov's method for calculating $\|\nabla\phi\|$. Here with forward finite differences (x^+, y^+, z^+), backward finite differences (x^-, y^-, z^-), and central differences (the remaining), respectively. κ is given by [72, p. 20]. The numerical epsilon $\epsilon = 10^{-6}$ prevents a division by zero.

D. GPU Implementation and Memory Consumption

To enable interactive exploration, we implemented the optimization of extinction and context geometry, as well as the subsequent rendering on the GPU using CUDA. While residing entirely in device memory ensures interactive frame rates, it also implies that all data is uploaded prior to optimization and thus has to fit completely into GPU memory.

The visibility optimization for the *direct* volume rendering [9] requires the allocation of transmittance fields $T_V(\mathbf{y})$, $T_L(\mathbf{z})$, and accumulated importance fields $G_V(\mathbf{y})$, $G_L(\mathbf{z})$, from both the view and light direction, cf. Eqs. (4)–(5). Our visibility optimization for the *indirect* volume rendering further requires a base SDF $\phi_0(\mathbf{x})$ in object space, the optimized SDF $\phi(\mathbf{x}, t)$, for which only the latest time step is kept in memory, as well as the field $\widehat{g}(\mathbf{x})$. Additional memory is required for further interactive features, such as screen-space and world-space masks, which are introduced in Section III-F. Hence, we require optimizations that save memory, which is in support of (**R3**). As evaluated later in Section IV-D, our implementation is able to switch between 32-bit floating point precision and 16-bit floating point precision (conforming to the IEEE-754 standard), when storing the transmittance, accumulated importance, $\widehat{g}(\mathbf{x})$, and the optimizer data required for AdEMAMix [67]. When switching to lower precision, we are able to load the data sets of [9] in higher resolution, and we enable the optimization of the implicit context geometry.

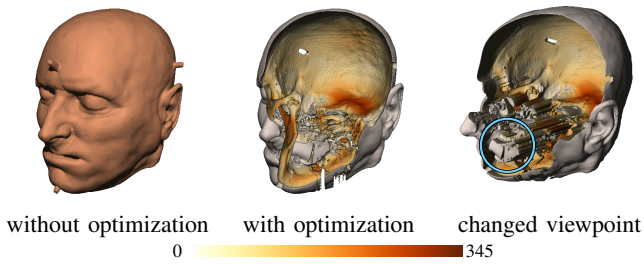


Fig. 5. Deforming the context geometry alters the given data, which needs to be communicated to the user. In order to visually convey how much isocontours have been changed by the optimization, we map the distance between ϕ and the base SDF ϕ_0 to color using a heatmap transfer function, here for the VISIBLE HUMAN. The distance measure is normalized to voxels, here in $[0, 345]$. Left: context geometry without optimization. Middle: only the context geometry after a joint optimization of context and extinction. Right: same scene from a different viewpoint. The blue circle marks parts above the lip that remained unchanged, since they did not occlude important structures.

E. Visualization

To visualize the final scene, we employ the light transport model in Eq. (6). In the following, we point out additional considerations that went into the visual encoding, as well as improvements made to increase the rendering quality, i.e., to support (**R4**).

Color Mapping: While the appearance of the volume is given by a user-defined transfer function, we adjust the color of the deformed surfaces to visually communicate how much the isocontours of ϕ deviate from the isocontours in the base SDF ϕ_0. Since, ϕ_0 is an SDF, this distance is simply obtained by sampling ϕ_0 at the level set of ϕ. The mapping is done by a user-defined colormap, which is demonstrated in Fig. 5. By changing the viewpoint, it can be seen that the level set made way for the skull. Thereby, parts above the lip remained unchanged, since they did not occlude important structures. In Fig. 1, the color was instead determined by a geophysical texture of the Earth, which we projected onto the SDF isocontour.

Ambient Occlusion: Ambient occlusion approximates the darkening in corners by assuming a uniform far-field illumination and by pre-computing the fraction of occluded rays in the upper hemisphere. The approach is commonly used in volume rendering, as it improves depth perception [77]. In our approach, the ambient occlusion is calculated from both the volume as well as the implicit surfaces in order to provide depth cues that convey their relative positioning. An example is shown in Fig. 6. Here, the ambient occlusion adds depth to the eye sockets, which receive less light due to the surrounding skull. Further, the neck is darkened, which is visible through the mouth. The ambient occlusion on the jaw bone is due to the surrounding context geometry, i.e., the cheeks, which visually embeds the jaw deeper in the volume.

Fig. 6. Depiction of ambient occlusion (AO). Left: scene without ambient occlusion. Middle: only AO is visualized. Right: both are combined.

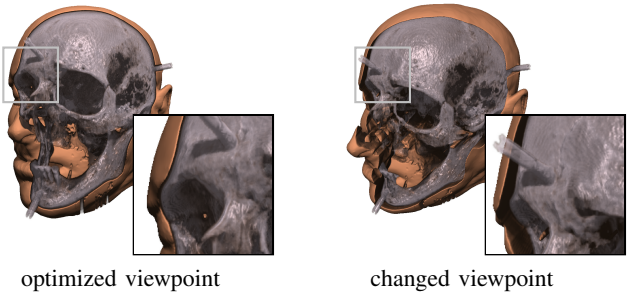


Fig. 7. For explanatory purposes, the scene can be viewed from a different view than it was optimized for, allowing to see how the important structures are placed relative to the preserved context geometry.

F. Interaction

Interactive editing and navigation are important instruments for exploration and scene understanding. In the following, we explain interaction concepts, supporting (**R5**).

Navigation: By default, the visibility optimization adjusts to new viewpoints during interactive camera navigation. If desired, the optimization can be disabled once it is completed for a given view. A subsequent minor modification of the viewpoint can be useful to improve the spatial understanding of the optimized context geometry due to motion parallax. For explanatory purposes, we show a rotated view in Fig. 7 for the VISIBLE HUMAN data set, where it can be seen which parts are preserved in the context geometry. Another example is shown in Fig. 1 for the EARTH MANTLE, where the rotated view conveys where the free-floating cold slabs are placed inside the mantle.

Screen-Space Masks: In context-preserving volume rendering [32], context is preserved around the structures of interest to help anchoring them in their surrounding. Viola al. [38], [39] proposed among others windowing functions, which have also been employed by Ament et al. [8] to preserve some of the geometry in front. Such windowing is easily added to our optimization by masking the importance along the view ray in Eq. (19) for selected pixels, which is demonstrated in Fig. 8. In some cases, it may also be fruitful to remove more of the context geometry, in order to show more of the region in which the important structures are spatially embedded. For this, we let the user interactively move a lens [40] via the mouse cursor across the screen, which likewise modulates the importance along the view ray.

World-Space Masks: In addition to the aforementioned screen-space masks, we also allow the user to specify world-space regions that remain unchanged by the level set optimization. Effectively, the first term in Eq. (23) is disabled, letting the optimized SDF $\phi(\mathbf{x}, t)$ strive to remain close to its base SDF $\phi_0(\mathbf{x})$. Preserving key structures of the context can be important to anchor the important volumetric structures in the scene. For example, Fig. 9 shows how the continental landmasses of the Earth are guaranteed to be preserved. Without the world-space mask, the continents are fragmented into small pieces, making them hardly recognizable. Of course, preserving too much geometry introduces occlusions. The user therefore has to choose carefully, which parts of the context geometry are essential for scene understanding, and which parts can optionally be removed to clear the view onto relevant structures inside the volume. Visualizations of the used world-space masks are provided in the supplemental material.

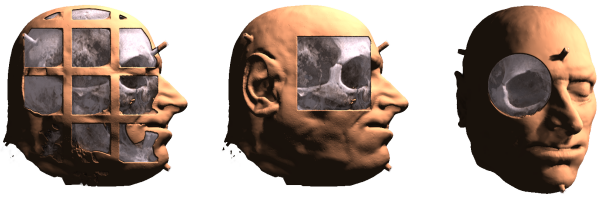


Fig. 8. To provide occlusion depth cues, the importance can be masked in view space to retain some of the context geometry in front of relevant parts of the scene, here shown for three different masks.

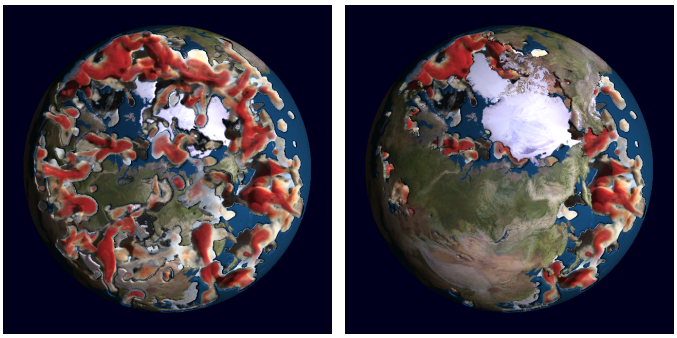


Fig. 9. Difference between the results obtained without and with a world-space mask. On the left, the optimization without world-space mask reveals all important structures in the volume. The location on Earth, however, is difficult to see. On the right, we add a world-space mask to the level set propagation to preserve the continental landmasses. While some of the important structures are hidden, the position on the planet is much better visible.

IV. RESULTS

To test the limits, stability, and utility of our visibility optimization, we apply the approach to a number of different scenes. We refer to the supplemental material for separate visualizations of the scalar fields, context geometries, masks, and importance fields. In the following, we compare with other approaches, conduct a user study, a parameter study, analyze the convergence, the performance, and discuss limitations.

A. Comparison

In Fig. 10, we compare our approach with other methods. The first column shows an extension of the maximum importance

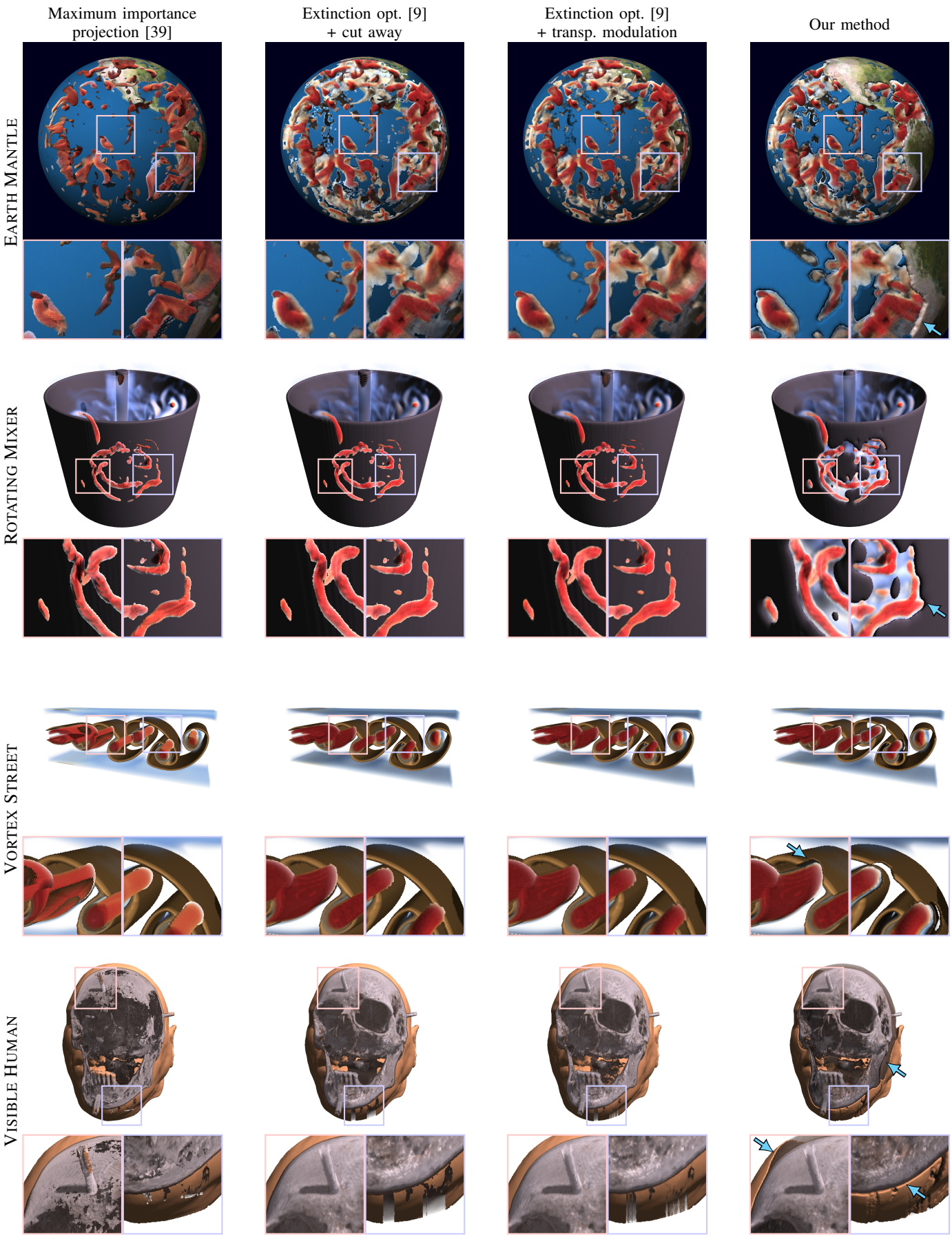


Fig. 10. Comparison with alternative visibility optimization techniques on four data sets. Unlike previous methods, our approach models the context geometry implicitly via level sets, which allows for the modeling of a thickness that conveys shape information from shading (see the arrow markers).

projection of Viola et al. [39], which retrieves the most important point along the view ray and shows its color. While originally developed for volume data, the extension to scenes containing both volumes and surfaces is straightforward. The technical modifications, for example to support shadows, are explained in detail in the supplemental material. The second and third column both utilize the extinction optimization of Himmler and Günther [9] for the volume, but incorporate the context geometry differently. The second column performs a thresholding on the $\widehat{g}(\mathbf{x})$ field to cut away all geometry that occludes important structures behind. The third column maps the $\widehat{g}(\mathbf{x})$ via a linear transfer function to transparency, which is similar to previous work on transparency modulation [40]. Finally, the fourth column shows our method.

The EARTH MANTLE simulation models the geothermal activity between Earth's crust and core. Over the course of geological timescales ($10^6 - 10^9$ years), the mantle behaves like a viscous fluid. Thereby, material of cold and dense nature descends from the crust, while hot plumes emanate from the Earth's core [78]. In Fig. 10 (first row), the difference between observed and radially averaged temperature is visualized, which is called temperature anomaly. Here, cold slabs are highlighted by mapping them to a high importance value. To provide context an implicitly given Earth surface is added onto which a geophysical texture is projected. Since the continental land masses are preserved in our approach, the location on Earth is easier to see. Note, however, that the Earth texture is decorative and not part of the actual simulation. In the right close-up, the thickness of the thin shell SDF can be seen, which suggests that the volume is inside the surface. In the other methods, it is ambiguous whether important volumetric structures are inside the Earth's surface or in front of it.

The ROTATING MIXER contains a turbulent fluid flow, in which a rotating paddle stirs a liquid into motion. The fluid is confined into a cylindrical container and the paddle is mounted on a rod that descends into the volume from the top. In Fig. 10 (second row), we highlight turbulent structures by calculating the vorticity magnitude from the velocity [79]. Furthermore, the rendering of the cylindrical container and the central rod provides additional context information. The implicit context geometry is displaced by our optimization to reveal structures with high vortical motion. The cut away and the transparency modulation give visually similar results, whereas the latter has a smoother transition between focus and context, which can be seen in both close-ups. By a suitable choice of λ, we can ensure that a little more cylinder geometry gets removed, which reveals more of the volume around the important structures, giving it more context. Further, the preservation of the central rod gives a strong depth cue since one of the vortices is shown to be behind the rod.

The VORTEX STREET contains a three-dimensional fluid flow around an obstacle, which induces a periodic shedding of vortices, resulting in a von-Kármán vortex street. The flow was simulated by Camarri et al. [80] and was resampled and made available by Tino Weinkauf [81]. In Fig. 10 (third row), we visualize the vorticity magnitude using direct volume rendering, which is likewise used as importance measure. To provide context, we place two explicit streak surface triangle meshes for which a thin shell SDF is precomputed using Eq. (26). Due to the finite thickness of the thin shell SDF, our method obtains a shaded border where the surface is displaced. This provides a clear hint that the volume is behind the streak surfaces. The other methods, on the other hand, remain ambiguous about the spatial placement of the important structures, as they could likewise be floating in front.

The VISIBLE HUMAN data set (version 2.0) [82] is a renowned collection of high-resolution CT and MRI scans of a male head and neck. For our visualization in Fig. 10 (fourth row), we used the CT scan. The focus of our visualization is the skull, while the isocontour of the skin serves as context geometry. Naturally, the skin surrounds the bone structures, impeding the view onto the cranial structure and other bone structures of importance. With the maximum importance projection, the eye sockets are hardly visible. In our method, the added ambient occlusion improves the depth perception, especially for the eye sockets and the neck. We also optimize the visibility from the light view, cf. Eq. (19), which leads to the removal of the skin on the top of the head. This is visible in the left close-up, which shows the transition between context geometry and volume. The light view optimization and the shading on the SDF boundary result in strong depth cues that anchor the volume inside the context surface. Unlike all other methods, our approach shows the space between the jaw bone and the cheek skin. The ambient occlusion causes a darkening of the jaw bone, which adds to a sense of depth. In the future, further considerations of human perception could help to improve the readability of the visualizations.

B. User Study

To assess the improvements in depth perception, we conducted a user study with 11 participants, all having corrected-to-normal color vision. All study partakers have 3 to 10 years of experience in visual computing and have an academic background as PhD student or PostDoc in visualization, computer vision, or computer graphics. Given 32 cropped views of our data sets, which are included in the supplemental material, we provided the statement that one marked point was closer to the camera than another. On a Likert scale, the participants were asked to rate how much the given picture conveyed this impression. To avoid memorization effects, the images were shown in a randomized order.

Summary: The aggregated results (mean and standard deviation) are marked on a range from 1 (strongly disagree) to 5 (strongly agree): Maximum importance projection [39] reached a score of $2.36\,(\pm 0.90)$. Extinction optimization [9] with cut away got a score of $2.33\,(\pm 0.88)$. Extinction optimization [9] with transparency modulation obtained $2.66\,(\pm 1.00)$. Our method achieved the highest score with $3.99\,(\pm 0.73)$. To assess the significance of the results, we performed a Friedman test and a Wilcoxon signed rank test in the supplemental material. The Friedman test ($\alpha = 5\%$) showed a significant overall difference between methods ($p \approx 2.6 \times 10^{-5}$). To identify which pairs differed, we performed Wilcoxon signed-rank tests for all method pairs (critical value = 13 for $n = 11$, two-tailed). The results indicate

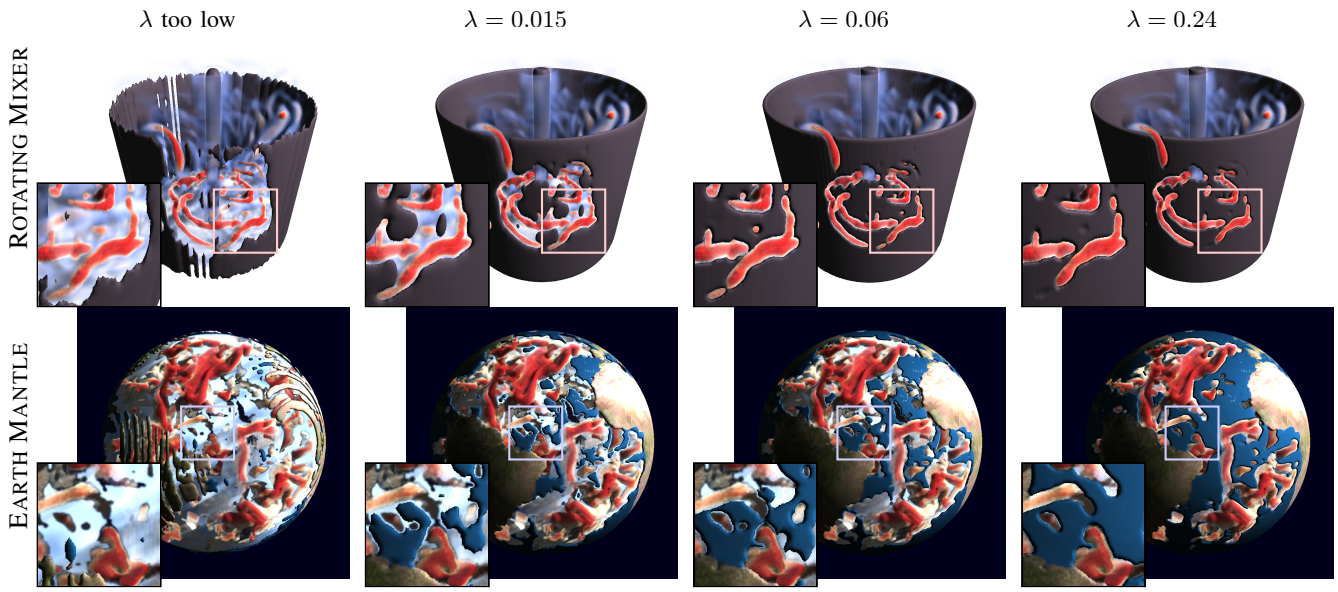


Fig. 11. Parameter study for different choices for the weight λ, which determines how much the level set aims to return into its initial state $\phi_0(\mathbf{x})$. If the value is low, the level set is able to clear the view even onto less relevant structures, which may, however, result in too much removal of context geometry. Increasing the weight λ leads to more context geometry being preserved. The first row shows the ROTATING MIXER and the second row shows the EARTH MANTLE, both with thin shell context geometry.

that our method differs significantly from the others. The pair of maximum importance projection and cut-away showed no significant difference due to their similar operating principles.

Feedback from Participants: In our extension of the maximum importance projection [39], participants noted that hard edges and strong shadows supported 3D perception. However, the direct and indirect volume rendering appeared disconnected from each other. This was pointed out in particular for the VORTEX STREET. The missing visual connection was likewise frequently pointed out for the extinction optimization [9] in combination with cut aways. The binary cutout in the surface was often perceived as a visible border between surface and volume. Thus, the volume was often described as hovering in front of the surface. Especially in the ROTATING MIXER, the surface was mistaken with a background color due to missing shading on the boundaries. Modulating transparency to smoothly fade between surface and important volume parts was generally considered visually more pleasant than the two methods described previously. Some participants tended to agree that blending supports the perception of depth order, as it seemed to introduce a shaded border. However, the perceived shading was considered implausible, as it was inconsistent with the shading in other parts of the image. Due to the depth ambiguities added by blending, the EARTH MANTLE felt overwhelming and confusing for some users. One participant described the visualization as 'a painting of some volume on a surface'. In contrast to all other methods, our method conveyed the impression of plasticity in the whole image, connecting both direct and indirect volume rendering. The shaded borders introduced by our approach were interpreted as a strong depth cue and were used as indicator for rating the depth order. Due to the small crop size, it was sometimes ambiguous whether parts curved to the front or to the back. Nevertheless, the plasticity helped disambiguate layers. Particularly in the VISIBLE HUMAN data set, our method was the only approach to visually convey that the skin surface encloses the skull.

C. Parameter Study

The time evolution of the level set is governed by the PDE in Eq. (23), which contains a weight λ that determines how much the level set strives to return into its initial state. In Fig. 11, we show optimization results for different choices of λ in two data sets. The parameter behaves intuitively: the lower the λ value, the more occluding geometry is removed. The higher the value, the less is the level set able to clear the view onto relevant structures. The minimal required value for λ is dependent on the thickness of the context geometry. We observed that for the thin shell representation, λ could be larger than for the filled surface representation, in which the level set needs to travel a farther distance to clear the view.

D. Performance Measurements

The following performance measurements were conducted on a workstation that is equipped with an AMD Ryzen 9 7950X CPU and an NVIDIA RTX 4090 GPU at an image resolution of 1024×1024. Table I lists the timings for all data sets, where we present the timings for the different stages in both 16-bit and 32-bit floating point precision, the time required for a single iteration of the redistancing process, the final render time, and the memory consumption. We thereby list the timings for the different multi-grid levels separately. The column 'Grid Updates' includes the calculation of the transmittance T_V, T_L, accumulated importance $G_V, G_L, \widehat{g}$, and the energy derivatives. For low resolutions 16-bit is faster, while the higher resolutions favor 32-bit precision. The extinction optimization ('Ext. Opt.') is generally faster for 32-bit, except for the VISIBLE HUMAN, which did not fit into memory. The

TABLE I
PERFORMANCE MEASUREMENTS FOR OUR METHOD. TIMINGS ARE PROVIDED FOR DIFFERENT RESOLUTIONS EMPLOYED BY OUR MULTI-GRID SOLVER. WE ADDITIONALLY SHOW THE TRADEOFF BETWEEN PERFORMANCE AND MEMORY CONSUMPTION WHEN USING 16/32-BIT FLOATING POINT PRECISION. THE 'REDIST.' COLUMN REPORTS THE TIME REQUIRED FOR ONE ITERATION OF THE REDISTANCING PROCESS, AS OUTLINED IN EQ. (18). ALL MEASUREMENTS WERE TAKEN WITH AN IMAGE RESOLUTION OF 1024×1024. THE TIMINGS ARE EXPRESSED IN MILLISECONDS. NOTE THAT THE VISIBLE HUMAN DATA SET DID NOT FIT INTO MEMORY AT 32-BIT FLOATING POINT PRECISION.

Data Set	Volume Resolution	Grid Updates		Ext. Opt.		SDF Opt.	Redist.	Rendering		VRAM Usage	
		16-bit	32-bit	16-bit	32-bit	16/32-bit	16/32-bit	16-bit	32-bit	16-bit	32-bit
EARTH MANTLE	$108 \times 108 \times 108$	**9.2**	9.9	0.55	**0.43**	0.04	0.38	7.8	**5.2**	**9.0 GB**	15.8 GB
	$216 \times 216 \times 216$	**33.3**	33.4	3.6	**2.8**	0.20	0.89	7.4	**5.3**		
	$432 \times 432 \times 432$	190.1	**162.8**	24.6	**19.5**	0.62	2.9	8.4	**6.4**		
ROTATING MIXER	$65 \times 33 \times 65$	6.9	**6.5**	0.16	**0.13**	0.02	0.09	4.7	**3.0**	**5.3 GB**	10.2 GB
	$130 \times 66 \times 130$	**18.1**	21.1	1.1	**0.86**	0.04	0.16	4.3	**2.9**		
	$260 \times 132 \times 260$	98.7	**93.7**	4.4	**4.3**	0.17	0.66	4.3	**3.3**		
VORTEX STREET	$96 \times 24 \times 36$	**7.4**	9.5	0.15	0.15	0.02	0.09	4.5	**2.9**	**7.7 GB**	14.8 GB
	$192 \times 96 \times 72$	**22.9**	28.8	1.1	**0.9**	0.04	0.19	4.3	**2.9**		
	$384 \times 192 \times 144$	143.9	**136.1**	4.8	**4.4**	0.18	0.75	4.5	**3.4**		
VISIBLE HUMAN	$128 \times 115 \times 128$	**12.0**	–	**1.3**	–	0.1	1.1	**6.4**	–	**17.7 GB**	>24 GB
	$256 \times 231 \times 256$	**57.8**	–	**8.0**	–	0.5	2.6	**6.5**	–		
	$512 \times 462 \times 512$	**400.2**	–	**75.2**	–	1.8	14.4	**7.0**	–		

SDF optimization ('SDF Opt.') and the redistancing ('Redist.') are fast and independent of the floating point precision. The runtime complexity of the redistancing is linear in the number of redistancing iterations N, which we empirically set to at most $N = 64$. We terminate early when the redistancing residual falls below a mean square error (MSE) of 10^{-4}. How many iterations are required depends on the step size of the level set evolution, as larger changes to ϕ require more work to redistance the SDF. The 'Rendering' step is faster when using 32-bit. Naturally, the 'VRAM Usage' is significantly lowered with 16-bit floating point precision, making it possible to optimize larger data sets. We refer to the supplemental material for a visual comparison between results using 32-bit and 16-bit floating point precision. There, the ꟻLIP [83] metric shows that the differences are negligible.

E. Discussion

Supported Surfaces: While implicit surfaces can always be turned into isocontours of an SDF, not all explicit surface geometries can be modeled implicitly with SDFs. The explicit surface is required to be a closed manifold, i.e., there are no self-intersections, no intersections with other surfaces, and there are no holes in the surface. For imperfect surfaces, which are not watertight, a heat-based SDF solver is available [84], which could be used. As with the streak surfaces in Fig. 10, it is, however, possible to compute a thin shell SDF for any explicit geometry, including points, lines, and surfaces.

Perceptual Metrics: The importance-driven quality metric we used for controlling the speed with which the level sets evolve is not yet considering the human perception, such as the dependence on the illumination and the contrast between foreground and background. Incorporating perceptually-motivated visual quality measures [6], [85], [86] and optimizing the placement of the light source [87] for a better scene understanding would be interesting avenues for future work.

Large-Scale Volumes: Representing the context geometry implicitly further increases the memory consumption of the extinction optimization. For example, the explicit surface representation in Fig. 10 is 6.2 MB in size, while the implicit representation requires 40.5 MB. For general volume rendering, a number of output-sensitive methods exist that guide the resource utilization in a ray-guided, visualization-driven, or display-aware manner [2]. The incorporation of such memory managements methods, as well as the use of narrow-band level sets [88], or a problem-specific bit compression would be interesting next steps to make the method more scalable.

V. CONCLUSIONS

Providing context in volume rendering is typically done in one of two ways. Either the focus is an isocontour, around which a semi-transparent direct volume rendering is added to provide context. Or a direct volume rendering is the focus, which allows for rich transfer functions, around which semi-transparent surfaces are shown for context. Having too much transparency in the scene inhibits the scene understanding, and hence we proposed a context visualization that shows opaque geometry. To avoid the occlusion of relevant information, we modeled the context geometry implicitly and deformed it in a normal flow to clear the view. Thus with this paper, we proposed the first level set formulation [73] for the displacement of implicit context geometry for visibility optimization, which we combined with a view-dependent extinction optimization of the volume [9]. With this, a visibility optimization of the direct and indirect part of the visualization is performed concurrently. To enable the optimization of the context geometry, we lowered the memory consumption of Himmler and Günther [9]. Further, we added ambient occlusion to offer more depth cues. Following earlier works on context-preserving volume rendering [32], [39], [40], we added interactive masks and lenses such that the user can manually explore the context.

For the future, it would be interesting to incorporate perceptual quality metrics [6] and to optimize the placement of the light source [87]. We currently require dense grids to store view space and light space quantities, which could be approximated instead [46]. To increase the frame rate, the expensive volume extinction optimization [9] could be approximated with existing smart visibility approaches [5]. Tuning the parameters to have an unobtrusive transition between the two would be an interesting problem.

ACKNOWLEDGMENTS

We thank Gábor Janiga for providing the ROTATING MIXER, Anna Gülcher for the EARTH MANTLE [78], Tino Weinkauf for resampling the VORTEX STREET [80], and the NLM Visible Human Project for providing the VISIBLE HUMAN [82]. This work was supported by DFG grant no. GU 945/3-1.

REFERENCES

[1] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski, "A survey of volumetric illumination techniques for interactive volume rendering," *Computer Graphics Forum*, vol. 33, no. 1, pp. 27–51, 2014.

[2] J. Beyer, M. Hadwiger, and H. Pfister, "State-of-the-art in gpu-based large-scale volume visualization," *Computer Graphics Forum*, vol. 34, no. 8, pp. 13–37, 2015.

[3] C. Xu, G. Sun, and R. Liang, "A survey of volume visualization techniques for feature enhancement," *Visual Informatics*, vol. 5, no. 3, pp. 70–81, 2021.

[4] J. Sarton, S. Zellmann, S. Demirci, U. Güdükbay, W. Alexandre-Barff, L. Lucas, J. M. Dischler, S. Wesner, and I. Wald, "State-of-the-art in large-scale volume visualization beyond structured data," *Computer Graphics Forum*, vol. 42, no. 3, pp. 491–515, 2023.

[5] I. Viola and M. E. Gröller, "Smart visibility in visualization," in *Proceedings of the First Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Goslar, DEU: Eurographics Association, 2005, p. 209–216.

[6] M.-Y. Chan, Y. Wu, W.-H. Mak, W. Chen, and H. Qu, "Perception-based transparency optimization for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1283–1290, 2009.

[7] C. D. Correa and K.-L. Ma, "Visibility histograms and visibility-driven transfer functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 192–204, 2011.

[8] M. Ament, T. Zirr, and C. Dachsbacher, "Extinction-optimized volume illumination," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1767–1781, 2017.

[9] P. Himmler and T. Günther, "Transmittance-based extinction and viewpoint optimization," *Computer Graphics Forum (EuroVis)*, vol. 43, no. 3, p. e15096, 2024.

[10] M. Hummel, C. Garth, B. Hamann, H. Hagen, and K. I. Joy, "IRIS: Illustrative rendering for integral surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1319–1328, 2010.

[11] T. Günther, M. Schulze, J. Martinez Esturo, C. Rössl, and H. Theisel, "Opacity optimization for surfaces," *Computer Graphics Forum (Eurographics Conference on Visualization)*, vol. 33, no. 3, pp. 11–20, 2014.

[12] I. M. Gelfand and S. V. Fomin, *Calculus of variations*. Prentice-Hall Inc., Englewood Cliffs, N. J., 1963.

[13] T. F. Chan and J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.

[14] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer, 2006, vol. 147.

[15] J.-M. Morel and S. Solimini, *Variational methods in image segmentation: with seven image processing experiments*. Springer Science & Business Media, 2012, vol. 14.

[16] L. A. Vese and C. Le Guyader, *Variational methods in image processing*. CRC Press Boca Raton, 2016.

[17] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[19] B. Goldluecke, E. Strekalovskiy, and D. Cremers, "The natural vectorial total variation which arises from geometric measure theory," *SIAM Journal on Imaging Sciences*, vol. 5, no. 2, pp. 537–563, 2012.

[20] T. Schoenemann and D. Cremers, "A coding-cost framework for super-resolution motion layer decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1097–1110, 2012.

[21] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[22] D. B. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[23] W. Welch and A. Witkin, "Variational surface modeling," in *Proc. Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '92. New York, NY, USA: ACM, 1992, p. 157–166.

[24] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," *ACM Trans. Graph.*, vol. 23, no. 3, p. 905–914, aug 2004.

[25] T. Zhao, L. Busé, D. Cohen-Steiner, T. Boubekeur, J.-M. Thiery, and P. Alliez, "Variational shape reconstruction via quadric error metrics," in *ACM SIGGRAPH 2023 Conference Proceedings*. New York, NY, USA: ACM, 2023.

[26] N. Sharp and K. Crane, "Variational surface cutting," *ACM Trans. Graph.*, vol. 37, no. 4, jul 2018.

[27] C. Batty, F. Bertails, and R. Bridson, "A fast variational framework for accurate solid-fluid coupling," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 100–es, jul 2007.

[28] S. Weißmann, U. Pinkall, and P. Schröder, "Smoke rings from smoke," *ACM Trans. Graph.*, vol. 33, no. 4, 2014.

[29] N. Daßler and T. Günther, "Variational feature extraction in scientific visualization," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 43, no. 4, pp. 109:1–109:16, 2024.

[30] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, "A feature-driven approach to locating optimal viewpoints for volume visualization," in *VIS 05. IEEE Visualization, 2005.*, 2005, pp. 495–502.

[31] D. Weiskopf, K. Engel, and T. Ertl, "Interactive clipping techniques for texture-based volume visualization and volume shading," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 298–312, 2003.

[32] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller, "Illustrative context-preserving volume rendering," in *EuroVis*, 2005, pp. 69–76.

[33] M. Chen and H. Jäenicke, "An information-theoretic framework for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1206–1215, 2010.

[34] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," in *Proceedings of Seventh Annual IEEE Visualization '96*, 1996, pp. 227–234.

[35] G. Kindlmann and J. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *IEEE Symposium on Volume Visualization (Cat. No.989EX300)*, 1998, pp. 79–86.

[36] Y. Wu and H. Qu, "Interactive transfer function design based on editing direct volume rendered images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1027–1040, 2007.

[37] M. Ruiz, A. Bardera, I. Boada, I. Viola, M. Feixas, and M. Sbert, "Automatic transfer functions based on informational divergence," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1932–1941, 2011.

[38] I. Viola, A. Kanitsar, and M. Gröller, "Importance-driven volume rendering," in *IEEE Visualization 2004*, 2004, pp. 139–145.

[39] I. Viola, A. Kanitsar, and M. E. Gröller, "Importance-driven feature enhancement in volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, p. 408–418, 7 2005.

[40] J. Krüger, J. Schneider, and R. Westermann, "Clearview: An interactive context preserving hotspot visualization technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 941–948, 2006.

[41] S. Marchesin, C. Mongenet, and J. Dischler, "Per-pixel opacity modulation for feature enhancement in volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 04, pp. 560–570, jul 2010.

[42] T. Günther, C. Rössl, and H. Theisel, "Opacity optimization for 3D line fields," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 32, no. 4, pp. 120:1–120:8, 2013.

[43] ——, "Hierarchical opacity optimization for sets of 3D line fields," *Computer Graphics Forum (Eurographics)*, vol. 33, no. 2, pp. 507–516, 2014.
[44] N. Lerzer and C. Dachsbacher, "View-dependent visibility optimization for Monte Carlo volume visualization," *Computer Graphics Forum (Proc. Eurographics)*, p. to appear, 2025.
[45] T. Günther, H. Theisel, and M. Gross, "Decoupled opacity optimization for points, lines and surfaces," *Computer Graphics Forum (Eurographics)*, vol. 36, no. 2, pp. 153–162, 2017.
[46] I. Baeza Rojo, M. Gross, and T. Günther, "Fourier opacity optimization for scalable exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3204–3216, 2020.
[47] M. Zeidan, T. Rapp, C. Peters, and C. Dachsbacher, "Moment-based opacity optimization," in *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2020.
[48] B. Tuncay and T. Günther, "Optimization of color and visibility for dense line sets," in *Vision, Modeling and Visualization (VMV)*. Munich, Germany: Eurographics, 2024.
[49] N. Waldin, M. Waldner, M. Le Muzic, E. Gröller, D. S. Goodsell, L. Autin, A. J. Olson, and I. Viola, "Cuttlefish: Color mapping for dynamic multi-scale visualizations," *Computer Graphics Forum*, vol. 38, no. 6, pp. 150–164, 2019.
[50] P.-P. Vázquez, E. Monclús, and I. Navazo, "Representative views and paths for volume models," in *Smart Graphics*, A. Butz, B. Fisher, A. Krüger, P. Olivier, and M. Christie, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 106–117.
[51] U. Bordoloi and H.-W. Shen, "View selection for volume rendering," in *VIS 05. IEEE Visualization, 2005.*, 2005, pp. 487–494.
[52] L. Xu, T.-Y. Lee, and H.-W. Shen, "An information-theoretic framework for flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1216–1224, 2010.
[53] T.-Y. Lee, O. Mishchenko, H.-W. Shen, and R. Crawfis, "View point evaluation and streamline filtering for flow visualization," in *2011 IEEE Pacific Visualization Symposium*, 2011, pp. 83–90.
[54] J. Tao, J. Ma, C. Wang, and C.-K. Shene, "A unified approach to streamline selection and viewpoint selection for 3D flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 393–406, 2013.
[55] H. Shen and G. Ji, "Dynamic view selection for time-varying volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 05, pp. 1109–1116, sep 2006.
[56] S. Weiss and R. Westermann, "Differentiable direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 01, pp. 562–572, jan 2022.
[57] Y. Tao, H. Lin, H. Bao, F. Dong, and G. Clapworthy, "Structure-aware viewpoint selection for volume visualization," in *IEEE Pacific Visualization Symposium*, 2009, pp. 193–200.
[58] Y. Tao, Q. Wang, W. Chen, Y. Wu, and H. Lin, "Similarity voting based viewpoint selection for volumes," *Computer Graphics Forum*, vol. 35, no. 3, pp. 391–400, 2016.
[59] C. Yang, Y. Li, C. Liu, and X. Yuan, "Deep learning-based viewpoint recommendation in volume visualization," *Journal of visualization*, vol. 22, pp. 991–1003, 2019.
[60] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky, "Designing effective step-by-step assembly instructions," *ACM Transactions on Graphics*, vol. 22, no. 3, p. 828–837, 2003.
[61] S. Bruckner and M. E. Gröller, "Exploded views for volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1077–1084, 2006.
[62] C. Correa, D. Silver, and M. Chen, "Illustrative deformation for data exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1320–1327, 2007.
[63] M. Traoré, C. Hurter, and A. Telea, "Interactive obstruction-free lensing for volumetric data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1029–1039, 2019.
[64] K. Lawonn, M. Meuschke, and T. Günther, "Inversevis: Revealing the hidden with curved sphere tracing," *Computer Graphics Forum*, vol. 43, no. 3, p. e15080, 2024.
[65] W. Jarosz, "Efficient monte carlo methods for light transport in scattering media," Ph.D. dissertation, University of California, San Diego, 2008.
[66] L. G. Henyey and J. L. Greenstein, "Diffuse radiation in the Galaxy." *The Astrophysical Journal*, vol. 93, pp. 70–83, Jan. 1941.
[67] M. Pagliardini, P. Ablin, and D. Grangier, "The AdEMAMix optimizer: Better, faster, older," 2024, https://arxiv.org/abs/2409.03137.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
[69] J. Fischer, D. Bartz, and W. Straßer, "Illustrative display of hidden iso-surface structures," in *VIS 05. IEEE Visualization, 2005.* IEEE, 2005, pp. 663–670.
[70] S. Born, A. Wiebel, J. Friedrich, G. Scheuermann, and D. Bartz, "Illustrative stream surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1329–1338, 2010.
[71] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, pp. 61–79, 1997.
[72] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
[73] S. Osher, R. Fedkiw, and K. Piechor, "Level set methods and dynamic implicit surfaces," *Appl. Mech. Rev.*, vol. 57, no. 3, pp. B15–B15, 2004.
[74] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, no. 1, pp. 146–159, 1994.
[75] H. Zhao, "A fast sweeping method for Eikonal equations," *Mathematics of computation*, vol. 74, no. 250, pp. 603–627, 2005.
[76] M. Detrixhe, F. Gibou, and C. Min, "A parallel fast sweeping method for the Eikonal equation," *Journal of Computational Physics*, vol. 237, pp. 46–55, 2013.
[77] F. Lindemann and T. Ropinski, "About the influence of illumination models on image comprehension in direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1922–1931, 2011.
[78] A. J. P. Gülcher, M. D. Ballmer, and P. J. Tackley, "Coupled dynamics and evolution of primordial and recycled heterogeneity in Earth's lower mantle," *Solid Earth*, vol. 12, no. 9, pp. 2087–2107, 2021.
[79] T. Günther and H. Theisel, "The state of the art in vortex extraction," *Computer Graphics Forum*, vol. 37, no. 6, pp. 149–173, 2018.
[80] S. Camarri, M.-V. Salvetti, M. Buffoni, and A. Iollo, "Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate Reynolds numbers," in *XVII Congresso di Meccanica Teorica ed Applicata*, 2005.
[81] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel, "Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1396–1403, 2008.
[82] P. Ratiu, B. Hillen, J. Glaser, and D. P. Jenkins, "Visible human 2.0–the next generation," in *Medicine Meets Virtual Reality 11*. IOS Press, 2003, pp. 275–281.
[83] P. Andersson, J. Nilsson, T. Akenine-Möller, M. Oskarsson, K. Åström, and M. D. Fairchild, "FLIP: A difference evaluator for alternating images," *Proc. ACM Computer Graphics and Interactive Techiques*, vol. 3, no. 2, Aug. 2020.
[84] N. Feng and K. Crane, "A heat method for generalized signed distance," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–19, 2024.
[85] Y. Wu, H. Qu, K. Chung, M. Chan, and H. Zhou, "Quantitative effectiveness measures for direct volume rendered images," in *IEEE Pacific Visualization Symposium (PacificVis 2010)*. Los Alamitos, CA, USA: IEEE Computer Society, mar 2010, pp. 1–8.
[86] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert, "An information-theoretic observation channel for volume visualization," *Computer Graphics Forum*, vol. 32, no. 3pt4, pp. 411–420, 2013.
[87] S. Stoppel, M. P. Erga, and S. Bruckner, "Firefly: Virtual illumination drones for interactive visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1204–1213, 2019.
[88] A. E. Lefohn, J. M. Kniss, C. D. Hansen, and R. T. Whitaker, "A streaming narrow-band algorithm: interactive computation and visualization of level sets," in *ACM SIGGRAPH 2005 Courses*. New York, NY, USA: ACM, 2005, p. 243–es.

Paul Himmler is a PhD student with the Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. His research interests include visualization and computer graphics. Contact him at paul.himmler@fau.de.

Tobias Günther is a professor of computer science at the Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. His research interests include visualization and computer graphics. He received his PhD from the University of Magdeburg in 2016. Contact him at tobias.guenther@fau.de.