



Combined volume and surface rendering with global illumination caching

Uroš Šmajdek¹ · Žiga Lesar¹ · Matija Marolt¹ · Ciril Bohak^{1,2}

Accepted: 19 May 2023 / Published online: 26 June 2023
© The Author(s) 2023

Abstract

We present a combined volume and surface rendering technique with global illumination caching. Our approach uses volumetric path tracing to compute the global illumination volume and local shading models for rendering the isosurface. By joining both visualization approaches, we have enhanced the display and illumination of the surfaces while preserving physically realistic illumination of the participating media. To achieve real-time performance and avoid recomputing the image when the camera view changes, we compute the global illumination volume incrementally and defer the projection to a later step. We evaluated our technique by comparing different local shading models for isosurface rendering with the result of full volumetric path tracing and with the non-caching variant of our technique. Results show that the caching and non-caching variants perform comparably well, while the caching variant has the added benefit of being camera-view-independent. Additionally, we show that our approach emphasizes the surfaces within volumes better than volumetric path tracing.

Keywords Irradiance caching · Global illumination · Volumetric data

1 Introduction

In visualization, physically correct lighting contributes a lot to the accurate perception of details in the data, which is especially evident when rendering polygonal geometry. In this case, lighting contributes enormously to the correct perception of shapes, depth, and mutual arrangement of objects in the scene. When rendering volumetric data with different degrees of transparency, lighting is even more important, as the penetration of light through the substance allows for the recognition of details in the data. With simple and fast methods, which neglect physical correctness, such visualization may come at the cost of depth perception.

Volumetric data are used in a variety of scientific fields such as medicine, where such data are captured using various radiological techniques (computed tomography [22], magnetic resonance imaging [13], three-dimensional ultra-

sound [16], positron emission tomography [2]), meteorology, where such data can be captured by satellites and/or radars, astronomy, where volumetric data can also be captured in various ways [45] (by light or radio telescopes or with gravity wave detectors), all scientific fields using microscopy (transmission tomography [23], cryoelectron tomography [26]), physics, where such data are mostly the results of simulations, etc. All these areas have a common need for clear and accurate visualization of the captured data, which provides a good insight into their structure and details.

Often we want to visualize parts of volumetric data with the same properties (e.g., the same tissue densities), which can be represented by isosurfaces. In the past, researchers have presented many ways for direct and indirect isosurface rendering. Examples of indirect rendering involve calculating isosurfaces in the first step and presenting them as hardware-friendly geometry (e.g., a triangle list), while a common example of direct rendering is ray tracing [39]. While modern null-collision-based volume rendering techniques are more than fit for rendering participating media, they lack the capability of efficiently rendering isosurfaces since they correspond to large discontinuities in density. A suitable transfer function can simulate the solid appearance of an isosurface, but at a considerable cost of rendering efficiency. Until a general volume rendering method emerges

✉ Uroš Šmajdek
us6796@student.uni-lj.si

¹ Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia

² Visual Computing Center, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia

that can efficiently handle such discontinuities, combining volume and surface rendering methods is a necessity.

The goal of our work is to produce an interactive approach for displaying surface illumination while preserving the physically realistic illumination of the participating media. We obtain this by combining the two prevalent rendering approaches: isosurface rendering, which can indicate a specifically selected property in the data, and volumetric path tracing (VPT), which adds details that can only be attained with a global illumination technique. By ensuring that the individual techniques used in our method are physically based, we retain a high degree of realism without any computationally expensive work. Additionally, we include global illumination caching, which makes the method usable in interactive and exploratory scenarios. As a result, the method provides a good overview of the structures and details in the data while being fully interactive, thus enabling high-quality exploratory visualization in many scientific fields.

The main contributions of our work are:

- combining volumetric path tracing with locally illuminated isosurfaces in order to emphasize the desired surfaces in the volume, and achieve interactive but physically based visualization;
- adding global illumination caching to achieve real-time performance and interactivity; and
- comparison of the caching and non-caching variants of our technique with full volumetric path tracing and the use of different local lighting models for isosurface shading.

In Sect. 2, we present the related work and differentiate our contributions from it. In Sect. 3, we present our approach and compare it with selected existing ones. The results and evaluation of our method in comparison with the chosen techniques are presented in Sects. 4 and 5. In Sect. 6, we present the conclusions and give possible extensions and upgrades as part of further work.

2 Related work

The first approaches for interactive rendering of volumetric data were not physically based and were based on ad hoc techniques such as maximum intensity projection and emission-absorption model by Max [33]. Due to their simplicity and speed, they are still widely used in practice today, despite having undergone many upgrades. However, their simplicity is also the reason why they are mostly inadequate for more advanced visualizations, as they do not enable a good perception of the depths and shapes of individual structures in the data due to the lack of global illumination or its approximations such as volumetric ambient occlusion [41].

These shortcomings were the reason for the development of the first physically based approaches [10, 44], which were soon adapted for interactive use by Parker et al. [38]. An overview of the existing methods for imaging medical data is presented in the work of Tiede et al. [46]. Physically based approaches for calculating illumination in volumetric data include path tracing by Kajiya and Von Herzen [21] and radiosity calculation by Rushmeier and Torrance [42], which do not achieve better output quality or an increase in speed. An overview of the most commonly used methods for volumetric imaging in practice can be found in the work of Engel et al. [11].

When rendering volumetric data, however, we are most often really interested in what surfaces (isosurfaces) are present in the data. The most known approach for their calculation is the so-called marching cubes algorithm by Lorensen and Cline [32], using predefined templates to convert an isosurface into a triangular grid geometry that is most suitable for real-time rendering on graphics hardware. The approach has been upgraded and adapted several times, and the most well-known upgrades include marching tetrahedra [9, 48], implicit surface polygonization by Bloomenthal [4], and multi-level partition of unity implicits by Ohtake et al. [36]. A standard graphics pipeline can be used to represent the resulting geometry, but the disadvantage is the need for an additional step to calculate the geometry each time the isovalue is changed. Direct approaches do not have this disadvantage, as they take into account the isovalue during the rendering process itself. The most commonly used approach for rendering isosurfaces is ray tracing [5, 39]. While it is possible to perform such rendering in real time with modern graphics hardware, it is still not physically correct due to local illumination.

Ray marching isosurface rendering of surfaces within the volume data was first introduced by Levoy [28], where local shading was used during the accumulation of every non-transparent sample along the ray. This method was and still is a basis for many direct volume rendering techniques, as it offers great speed and a good approximation of a physically correct result. An adaptation of ray marching—sphere tracing—was introduced by Hart [15], assuring that one does not penetrate the implicit surface. Hart's method also approximates cone tracing [25] for antialiased rendering.

Voxels that represent the isosurface can also be considered as points in the 3D space—a point cloud—which can be used for surface reconstruction. A survey on surface reconstruction from point clouds is presented by Berger et al. [3].

The first physically based volume rendering approach [21], formalized with the rendering equation and its Monte Carlo solution by Kajiya [20], was later extended with support for visual mapping by Drebin et al. [10]. An unbiased approach to path tracing using delta tracking introduced by Yue et al. [54] was integrated into the implementation of

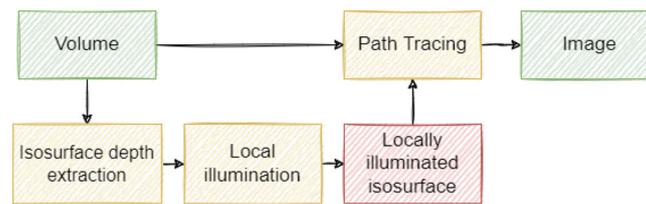


Fig. 1 An overview of the non-caching technique for combined volume and surface rendering

an interactive progressive volumetric rendering system by Kroes et al. [27]. A unified delta tracking framework was presented by Galtier et al. [14]. An overview of modern physically based Monte Carlo-based methods was presented by Novák et al. [35].

Combining multiple rendering techniques to achieve the desired results is not a new concept. Tietjen et al. [47] introduced combined surface and line rendering with ray marching for emphasizing the objects of interest within volumes. Their approach expects a segmented volume as well as volumetric and mesh representations of objects and allows users to render the final image in the desired style. Andersen et al. [1] present hybrid fur rendering where they join rendering of explicit hair strand geometry and rendering of volume texture of hair densities using ray marching of a prismatic shell volume with dynamic resolution. Their method creates a more detailed and soft fur appearance than either of the individual approaches. Isenberg et al. [18] present an observational study on how non-photorealistic rendering of 3D objects can be compared with traditional hand-drawn sketches. Bruckner and Gröller [6] present a style transfer approach for volumetric rendering for achieving sketch-like results. Often the desired results are achieved by carefully designing an appropriate transfer function. An overview of transfer function design is presented by Ljung et al. [30]. Most of the above methods aim to emphasize certain features in a volume using different approaches. Xu et al. [53] present a survey of feature-enhancing volume visualization techniques and propose their own approach. While they are not aiming to emphasize surfaces but rather certain voxel features, it is not directly comparable with our approach. They could be used together to emphasize features as well as surfaces in the volumes.

Even with the rapid development of GPU technology, real-time rendering of complex volumetric data with an unbiased physically based approach is not possible without additional steps such as denoising presented by Iglesias-Guietán et al. [17], super-resolution presented by Weiss et al. [51] or global illumination caching, such as radiance caching by Jarosz et al. [19], irradiance caching by Ribardieere et al. [40] and Khlebnikov et al. [24], and transmittance caching by Weber et al. [50]. In our approach, we implemented irradiance caching for computing a global illumination volume, allowing us to

use faster ray casting techniques to render the data while also displaying the isosurfaces.

There are several possible alternatives to our approach. The directional occlusion shading model [43] and its multidirectional extension [37] could be adapted for use with isosurfaces and a local shading model, although we would be severely limited by the illumination setup since these methods can only compute illumination in scenes with a single directional light source with a specific direction with respect to the camera. On the other hand, deep shadow maps [31] do not have such limitations, but again, only directed light sources can be simulated. Our approach has no such limitations.

3 Method

3.1 Overview

We first present a baseline *non-caching* technique for combined volume and surface rendering. An outline of the technique is shown in Fig. 1. For every pixel, we first extract the isosurface depth and calculate local illumination, then we pass this information to the path tracing module. In the path tracing module, the rays are cast from the camera into the volume, but only the light contributions gathered up to the isosurface are taken into account and blended with the local isosurface illumination contribution.

Next, we present the *caching* technique, which is an extension of the non-caching technique. An outline of the caching technique is presented in Fig. 2. We compute the global illumination of each voxel with path tracing and store it in a global illumination volume. Both the global illumination volume and local illumination contribution are taken into

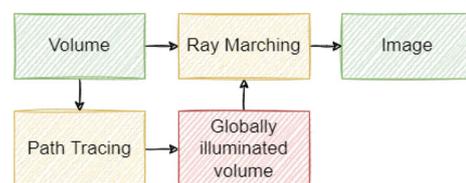


Fig. 2 An overview of the caching technique for combined volume and surface rendering

account during the ray marching step when accumulating the illumination along the ray.

individual steps from both techniques are presented in the following subsections.

3.2 Isosurface depth extraction

The isosurface separates the inside and outside regions where V is greater than or less than isovalue ρ , respectively. Let S be the isosurface of the volume V such that for every $s \in S$ $V(s) = \rho$. To display the isosurface with local illumination, only the visible part of the isosurface is needed, which we store in the depth buffer G . Each pixel G_{ij} in the depth buffer stores the distance to the isosurface along a viewing ray through that pixel, where the value 0 corresponds to the nearest intersection \mathbf{n} with the volume, and the value 1 corresponds to the farthest intersection \mathbf{f} with the volume. We are essentially solving the following equation to find the $G_{i,j}$:

$$V(\mathbf{n} + G_{ij} \cdot (\mathbf{f} - \mathbf{n})) - \rho = 0. \quad (1)$$

To compute the depths of individual pixels G_{ij} , we implement a simple stochastic process, in which we iteratively store the nearest point on the ray belonging to the inside region by randomly selecting new points along the ray, closer to the camera than the currently stored one, and checking whether it still belongs to the inner region or not. A single iteration of this process is described by Algorithm 1. Unlike classic techniques used for isosurface extraction, such as ray marching, the described technique only requires to sample the volume once per iteration. This allows it to be performed parallel to rendering, giving quicker results and improving interactivity.

Algorithm 1: Single iteration of the stochastic isosurface depth extraction

```

input : Volume  $V$ , Camera Position  $\mathbf{c}$ 
output: Isosurface depth buffer  $G$ 
begin
   $G \leftarrow 1$ ;
  foreach pixel  $p_{ij}$  in the final image do
     $\mathbf{r} \leftarrow$  ray cast from  $\mathbf{c}$  through pixel  $p_{ij}$ ;
     $\mathbf{n} \leftarrow$  near intersection between  $\mathbf{r}$  and  $V$ ;
     $\mathbf{f} \leftarrow$  far intersection between  $\mathbf{r}$  and  $V$ ;
     $\xi \leftarrow$  random value between 0 and  $G_{ij}$ ;
     $\mathbf{t} \leftarrow \mathbf{n} + \xi(\mathbf{f} - \mathbf{n})$ ;
    if  $V(\mathbf{t}) \geq \rho$  then
       $G_{ij} \leftarrow \xi$ ;
    end
  end
end

```

3.3 Local illumination

For the local illumination of the isosurface, we used three different models: Lambert's, Phong's and Disney's BRDF [7], with the gradient of the volume as the normal vector. We use one directional light in our demonstration. For the non-caching variant of the technique, we extract the depth for each pixel G_{ij} and use it to reconstruct the location of the isosurface inside the volume for the corresponding viewing ray. We sample the neighborhood of that location to compute the gradient and then, apply local illumination. For the caching variant of the technique, the isosurface location is computed during the ray marching, and the local illumination is computed in the same way as in the non-caching variant.

3.4 Global illumination

Light transport in a transparent medium is described by the radiative transfer equation [8]. It is composed of four terms, describing the change in radiance L due to emission, absorption, out-scattering, and in-scattering of the light traveling through an infinitesimal volume at the point \mathbf{x} in the direction ω :

$$(\omega \cdot \nabla)L(\mathbf{x}, \omega) = \sigma_a(\mathbf{x})L_e(\mathbf{x}_t, \omega) - \sigma_a(\mathbf{x})L(\mathbf{x}, \omega) - \sigma_s(\mathbf{x})L(\mathbf{x}, \omega) + \sigma_s(\mathbf{x})L_s(\mathbf{x}, \omega), \quad (2)$$

$$L_s(\mathbf{x}, \omega) = \int_{S^2} f_p(\mathbf{x}, \omega, \omega')L(\mathbf{x}, \omega') d\omega', \quad (3)$$

where σ_a and σ_s are the absorption and scattering coefficients, respectively, L_e is the emission, and f_p is the phase function, which describes the directional distribution of scattering. Integrating Eq. (2) along the direction ω up to the background at depth d gives the volume rendering equation, where the radiance contribution L_o at every point $\mathbf{x}_t = \mathbf{x} - t\omega$ along the ray is weighted by the transmittance T :

$$L(\mathbf{x}, \omega) = T(d)L(\mathbf{x}_d, \omega) + \int_{t=0}^d T(t)L_o(\mathbf{x}_t, \omega) dt, \quad (4)$$

$$L_o(\mathbf{x}_t, \omega) = \sigma_a(\mathbf{x}_t)L_e(\mathbf{x}_t, \omega) + \sigma_s(\mathbf{x}_t)L_s(\mathbf{x}_t, \omega), \quad (5)$$

$$T(t) = \exp\left(-\int_{s=0}^t (\sigma_a(\mathbf{x}_s) + \sigma_s(\mathbf{x}_s)) ds\right). \quad (6)$$

If we substitute the integral with a Monte Carlo estimate [12], we get the volumetric path tracing algorithm:

$$\langle L(\mathbf{x}, \omega) \rangle = T(d)L(\mathbf{x}_d, \omega) + \frac{1}{N} \sum_{i=1}^N \frac{T(t)}{p(t)} L_o(\mathbf{x}_t, \omega), \quad (7)$$

where $p(t)$ is an arbitrary probability density function along the ray. We generate the samples for the Monte Carlo simulation by first generating the free-flight paths of the photons

from the camera to an interaction with the medium, simulating the absorption or scattering event, and then repeating until we hit a light source. For the purposes of analytical and unbiased sampling of the free-flight paths, we homogenize the medium by adding a fictitious component with a density σ_n . The fictitious medium does affect light transport because it does not absorb light and exhibits perfect forward scattering:

$$\sigma_n(\mathbf{x})L(\mathbf{x}, \omega) = \sigma_n(\mathbf{x}) \int_{S^2} \delta(\omega - \omega')L(\mathbf{x}, \omega') d\omega'. \quad (8)$$

By adding the above equation to Eq. (2), we have to update the solution in Eq. (4) and the transmittance in Eq. (6) accordingly. By choosing σ_n such that $\bar{\sigma} = \sigma_a + \sigma_s + \sigma_n$ is a constant, the transmittance $T(t) = e^{-\bar{\sigma}d}$ is analytically invertible, the free-flight distance sampling with an exponential probability density function $p(t) = \bar{\sigma}T(t)$ can be unbiased, and as a bonus, we can avoid transmittance evaluation in Eq. (7). This process, known as Woodcock tracking or delta tracking [52], was later generalized to arbitrary values of σ_n by Galtier et al. [14]. We used it in our work to compute the global illumination volume by starting the tracking in the voxels.

3.5 Composition

There is a difference in how the composition is done for the *non-caching* and *caching* variants of the technique.

In the *non-caching* variant, the composition is combined with the path tracing procedure; when a ray passes the isosurface, we access the precomputed local illumination of the isosurface at the path’s origin pixel. The path is then redirected toward the light source.

Such an approach cannot be employed in the *caching* technique as the isosurface depth changes with the position of the camera, while the global illumination is computed independently of the camera position. To project the resulting global illumination volume on the screen and compute the isosurface, we employ ray marching with front-to-back alpha compositing [33], terminating the ray at the isosurface or once the accumulated opacity reaches a high value or the ray exits the bounding volume.

4 Evaluation and results

We evaluated the techniques on two volumes: a CT scan of the abdomen and pelvis¹ of size $512 \times 512 \times 174$, and a CT scan of a rainbow wrasse² of size $198 \times 470 \times 432$. The results were rendered in 1024×1024 pixel resolution. All

¹ <https://klacansky.com/open-scivis-datasets/>.

² http://digimorph.org/specimens/Halichoeres_pictus/.

tests were performed on a desktop computer with an AMD Ryzen 5 3600X 6-Core processor, 16 GB of RAM, and an Nvidia Titan Xp graphics card with 12 GB of RAM. The prototype application was running in the Google Chrome web browser version 99.

We evaluated the proposed techniques using different local illumination models (Lambert, Phong, and Disney). The volumes were illuminated with a white directional light originating from a lower frontal corner of the scene. To evaluate the convergence of the proposed methods over time, we used four different metrics: PSNR, SSIM [49], LPIPS [55], and VMAF [29], with respect to a converged reference image, which we acquired separately for each of the three shading models using the non-caching variant after 30 min of rendering time. The convergence rates for both volumes are plotted in Fig. 3. Both variants show a similar steady convergence in the first 5 s after which it slows down. LPIPS and SSIM even show an advantage in the first few seconds compared to the non-caching variant. The graphs show that the caching variant converges after around 15–20 s, after which point the metrics show a high similarity to the reference image. We make up for the difference with interactivity, which is only available when using the caching variant. The graphs in Fig. 3 show that different shading models have only a limited impact on the metrics, with Phong shading performing the worst.

We measured the similarity between the rendering results acquired with the caching and non-caching variants of the technique. We used the same image similarity metrics as for the convergence rate evaluation: PSNR, SSIM, LPIPS, and VMAF. Similarly, we used three different local illumination models: Lambert, Phong, and Disney. The resulting graphs in Fig. 4 show high similarity between the images rendered with the two variants, which supports the claim that the caching variant is capable of producing good quality images while also enabling interactive use.

Additionally, we measured the frame rate for both variants. The frame rate ranges from 10 to 30 frames per second for both variants, depending on the data, the view, and the rendering settings. However, most of the computation time in the caching variant is spent on the light transport simulation, which may be suspended after reaching sufficient convergence. According to Fig. 3, this happens after 15–20 s, after which point the light transport simulation can be suspended, leaving ray marching and local illumination as the only remaining computational efforts. Ray marching alone is capable of running at much higher frame rates, and we measured frame rates of up to 100 frames per second in our test cases.

To evaluate how the cache precision and size affect the final rendering, we compared results obtained using a full-resolution cache, and 32-bit precision floats with results obtained using a full-resolution cache and 16-bit precision floats, half-resolution cache and 32-bit precision floats and

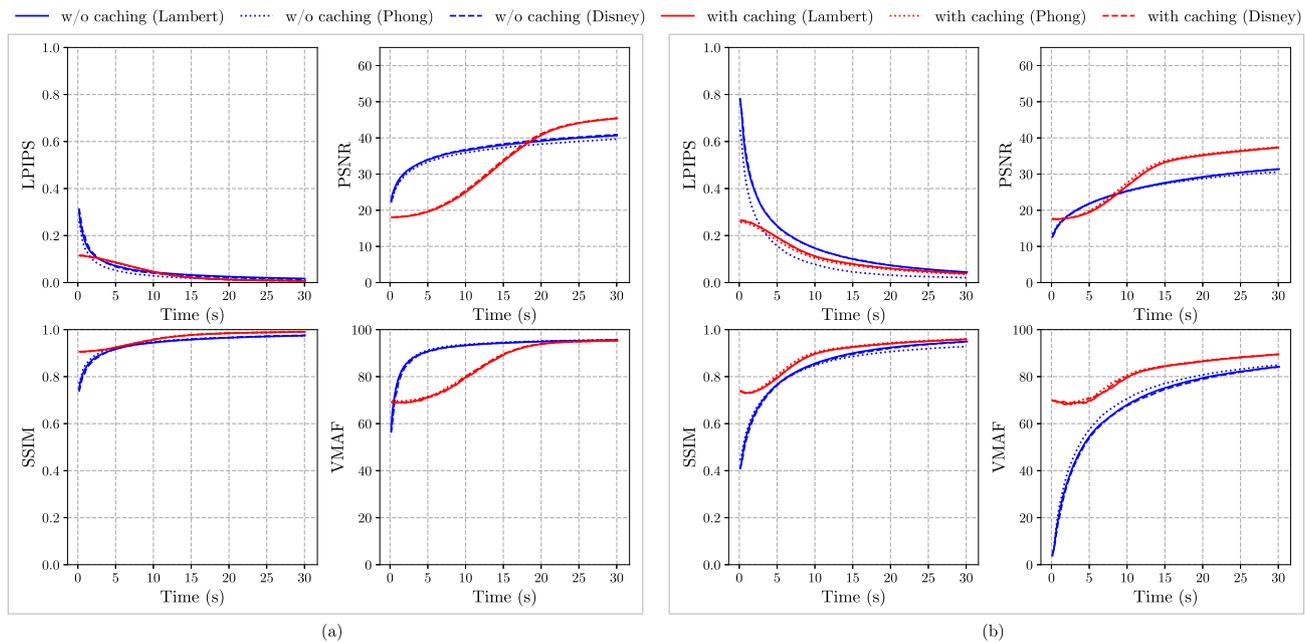


Fig. 3 Convergence rates of the two technique variants using different local illumination models in the first 30 s of rendering measured with different image similarity metrics and evaluated on two volumes: **a** a CT scan of a rainbow wrasse, and **b** a CT scan of the abdomen and

pelvis. For the reference image in each of the above graphs, we used the converged result obtained with the respective variant of the rendering method and a corresponding local illumination model

quarter-resolution cache and 32-bit precision floats on two volumes using PSNR and SSIM metrics as presented in Table 1. The results in Table 1 show that there is almost no difference between rendering outputs.

For qualitative evaluation, we present renderings of both technique variants using different local illumination models (Lambert, Phong, and Disney) after a 30-s run time. In Fig. 5, we show the images of the volumes we used for convergence evaluation and zoom-ins of the selected regions. In the top two rows, we show the resulting images of the abdomen and pelvis volume from left to right using: path tracing, non-caching variant using the Lambert, Phong, and Disney local illumination, followed by the caching variant of the presented technique for the same local illumination models. The bottom two rows show the same techniques applied to the volume of the rainbow wrasse. For the path-traced image, the transfer function was specifically set to emphasize the isosurface, where we used a high-opacity transfer function as a substitute for the isosurface. Note that this approach results only in an approximation of the isosurface and offers minimal control over its appearance. The user can still define the transfer function differently. We let the path tracing simulation run for 30 min for adequate convergence. We present and discuss the qualitative results in Sect. 5.

Next, we present the rendering results for a CT scan of a backpack¹ of size $512 \times 280 \times 374$ in Fig. 6. We show

the converged output of both technique variants and their difference image, which shows there are no major differences between the proposed technique variants. In Fig. 7, we show how the presented techniques converge in 30 s for the rainbow wrasse volume. In line with the results from Fig. 3, the images rendered with the cached variant catches up with the non-cached variant after the first few seconds.

For qualitative comparison of the presented methods with path tracing, we present the CT scan of a head with upper torso³ of size $512 \times 512 \times 460$ in Fig. 8. In the figure, we show a comparison between the path-traced images and our approach together with the difference images between variants of our method and the path tracing. The difference images show that the selected isosurfaces are additionally emphasized and that the use of the Disney shading model adds additional detail to the isosurface. This illustrates that the desired surfaces are distinguishable better with the presented method than with volumetric path tracing, while still preserving global illumination in the volume.

The memory demands of the presented techniques are as follows:

- *unmodified path tracing* requires 14 float values per pixel of the output image;

³ <https://www.osirix-viewer.com/resources/dicom-image-library/>.

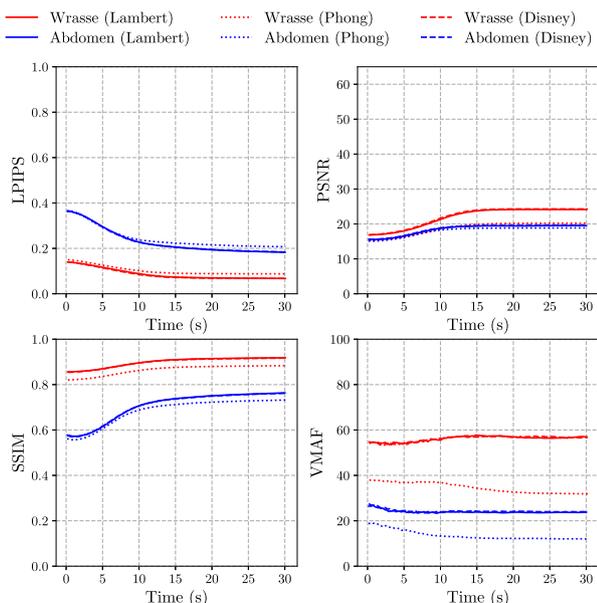


Fig. 4 Image similarity between the caching and non-caching variants with respect to rendering time, computed with different image similarity metrics

Table 1 PSNR and SSIM metrics of rendering results using 16-bit precision, half-resolution, and quarter-resolution on the backpack and rainbow wrasse volumes

Volume	Cache resolution	Float	PSNR	SSIM
Backpack	512 × 280 × 374	16-bit	49.663	0.997
	256 × 140 × 187	32-bit	46.925	0.995
	128 × 70 × 93	32-bit	43.761	0.994
Wrasse	198 × 470 × 432	16-bit	53.545	0.998
	99 × 235 × 216	32-bit	54.574	0.999
	49 × 117 × 108	32-bit	53.581	0.999

- the *non-caching variant* requires additional 4 float values per pixel of the output image;
- the *caching variant* requires 14 float values per voxel of the global illumination volume.

This makes the caching variant of the presented technique drastically more memory-demanding than the path tracing or non-caching variant of the presented technique. However, today’s high-performance GPUs typically come with a high amount of memory, making this issue less severe. The size of the global illumination volume increases proportionally

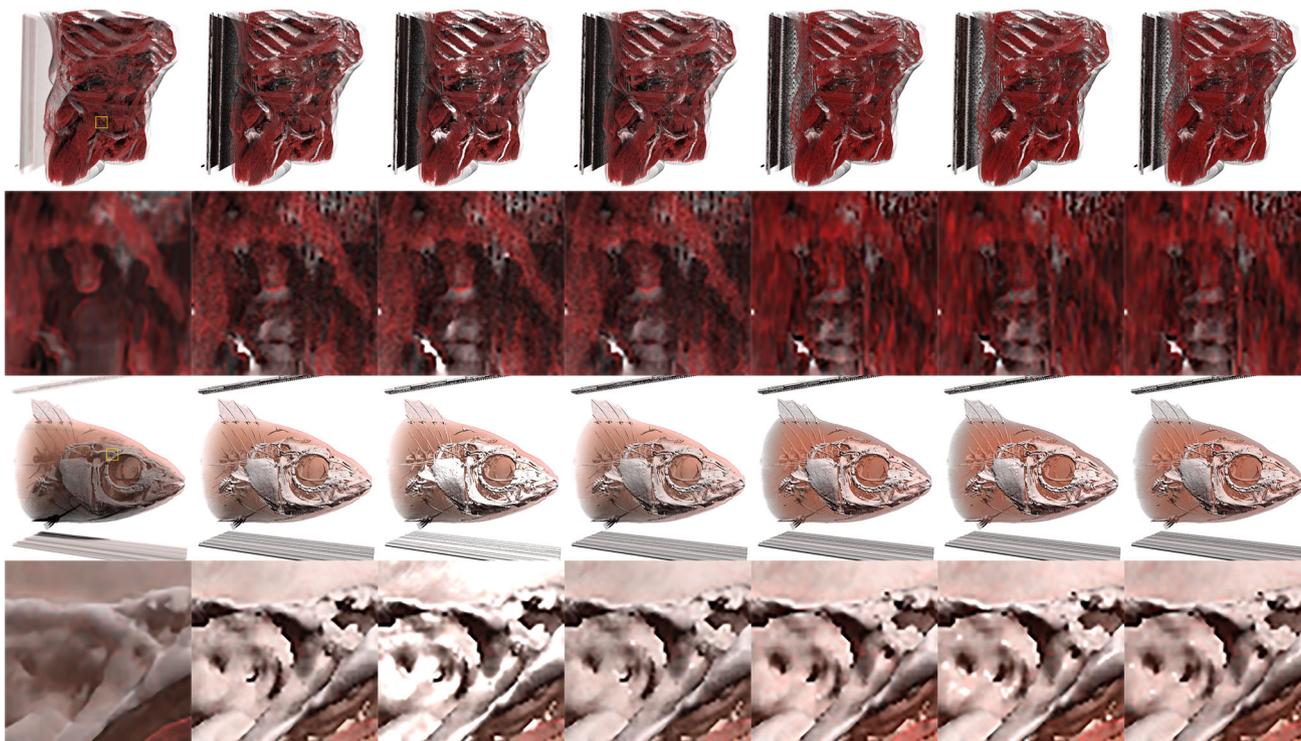


Fig. 5 Converged renders of the CT scan of the abdomen and pelvis (1st row) its crop (2nd row), and a rainbow wrasse volume (3rd row) and its crop (last row). From left to right the images are showing: path traced image (1st column), non-caching variant with Lambert (2nd col-

umn), Phong (3rd column), and Disney (4th column) local illumination, caching variant with Lambert (5th column), Phong (6th column), and Disney (last column) local illumination

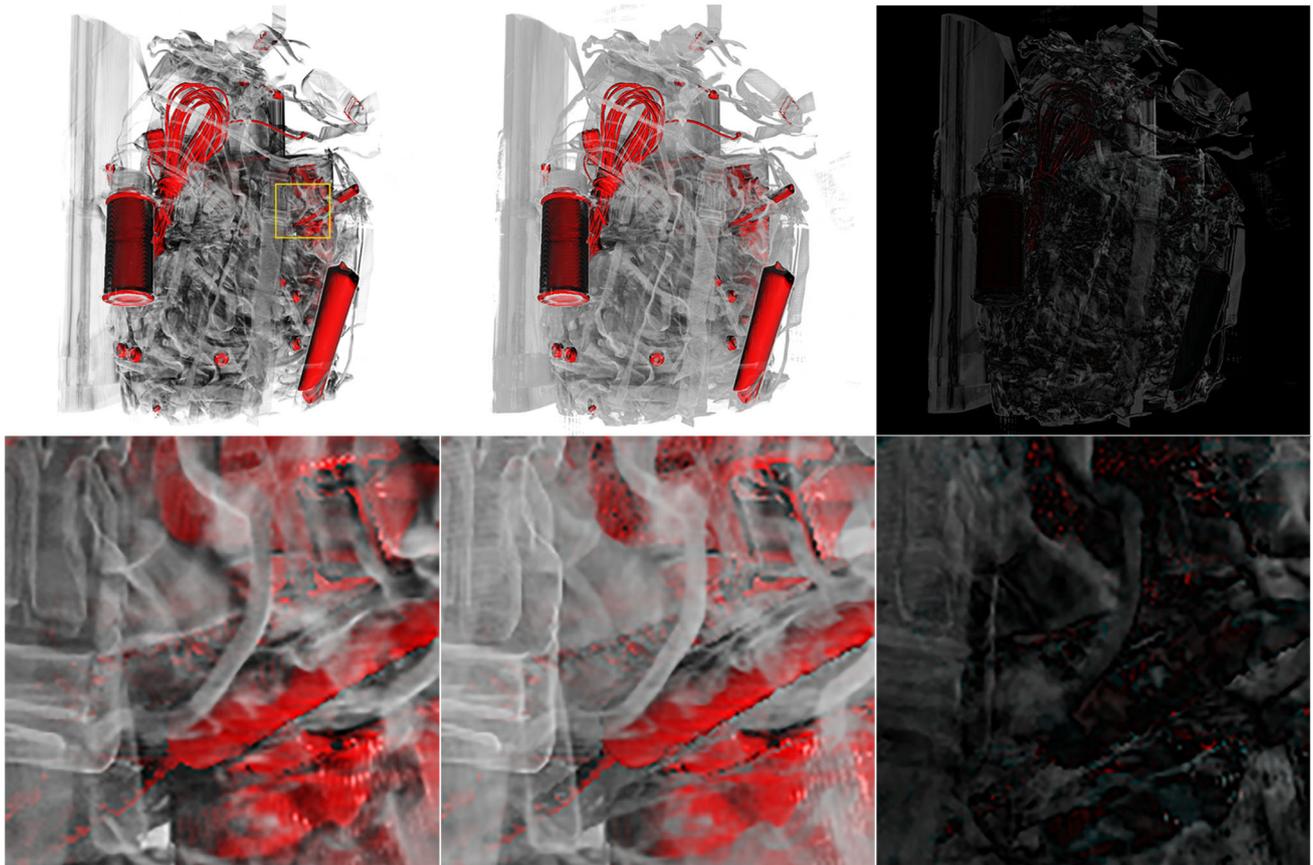


Fig. 6 Comparison of CT scan of a backpack rendered using non-caching variant (left) and caching variant (middle) of the presented technique, and a difference between them (right)

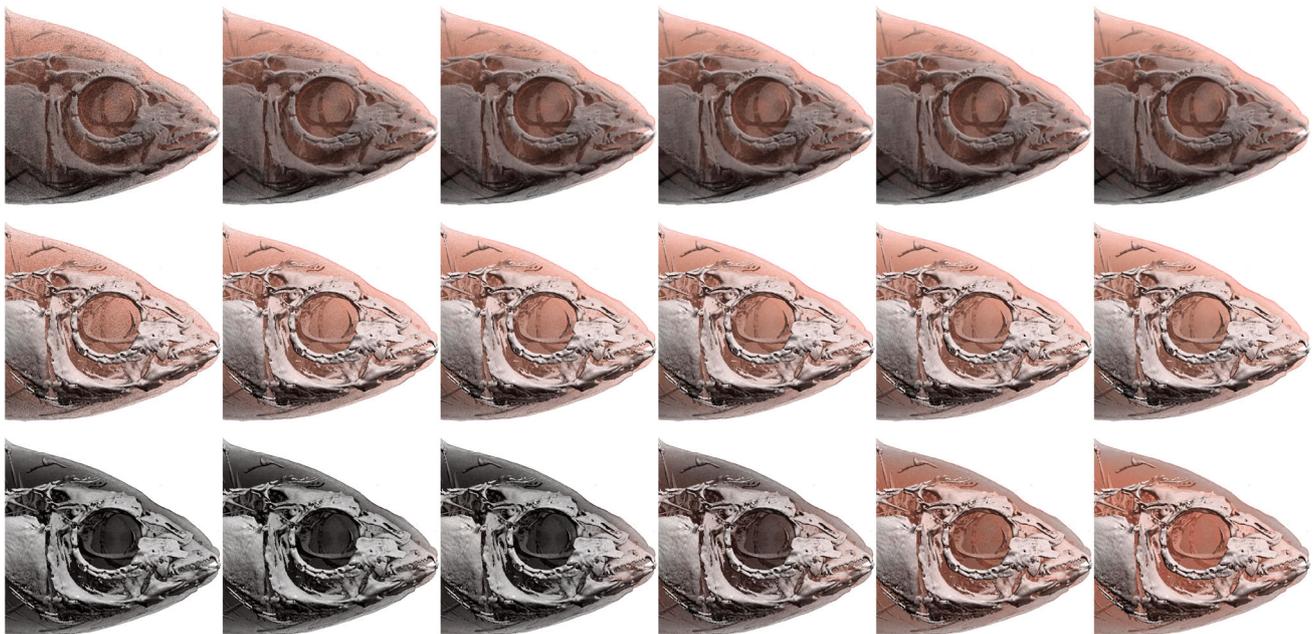


Fig. 7 Comparison of convergence of path tracing (top row), non-caching variant (middle row) and caching variant (bottom row) of our technique in different times: from left to right at 0.5, 2, 4, 8, 15, and 30 s

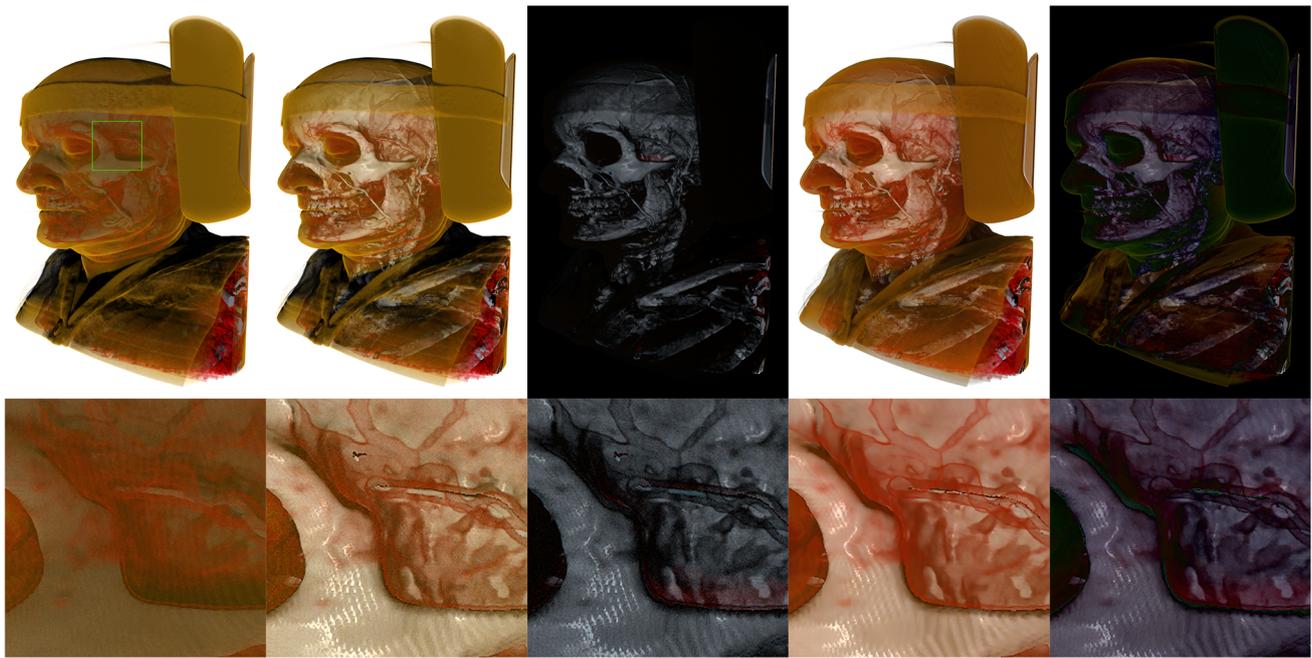


Fig. 8 Comparison of converged results (1st row) and their zoom-ins (2nd row) of path tracing (1st column), non-caching variant (2nd column), difference between path tracing and non-caching variant (3rd column), caching variant (4th column), and difference between path tracing and caching variant (5th column)

to the input volume size. This can be mitigated using lower precision numbers (e.g., 16-bit floats) and lower resolution global illumination volumes. Using modern high-end GPUs (e.g., with 80 GB RAM), it is possible to render volumes of sizes up to $1440 \times 1440 \times 1440$ with a full-resolution global illumination volume. A low-resolution global illumination volume with 16-bit floats significantly increases this limit.

We conducted further tests on data from various sources, as depicted in Fig. 9. The data include the Chameleon Dataset⁴ (top), acquired using Computed Tomography, the Hand dataset⁵ (middle), obtained with magnetic resonance imaging, and Virgo cluster simulation dataset⁶ (bottom). The first column shows volumetric path tracing results, the second shows the non-cached version of the presented method, and the third column shows the results of the cached version of the presented method. The results of each method are presented, highlighting both their benefits and limitations and are discussed in the following section.

5 Discussion

We must first point out that while all the presented methods in this paper are interactive (even path tracing), only the caching variant of our method retains the information even throughout the camera view changes, while the non-caching variant and path tracing methods start rendering from scratch.⁷ While path tracing and the non-caching variant of our method need to restart rendering after every parameter change—let that be the camera position or the isovalue—the cache is cleared only once the lighting conditions or the transfer function change, or when a new volume is loaded. The caching variant, however, takes a longer time to converge (see Figs. 3, 7). The illumination cache is updated while rendering; hence, no precomputation time is needed and no delay is experienced by the user. Consequently, the user can interact with the scene immediately after loading the volume or changing the transfer function, while illumination changes

⁴ University of Texas (UTCT Data Archive and Digimorph Library): <http://digimorph.org/index.phtml>.

⁵ Courtesy of one of the authors—Uroš Šmajdek.

⁶ MPA Numerical Cosmology Data Archive: https://wwwmpa.mpa-garching.mpg.de/Virgo/data_download.html.

⁷ This is demonstrated in the supplemental video.

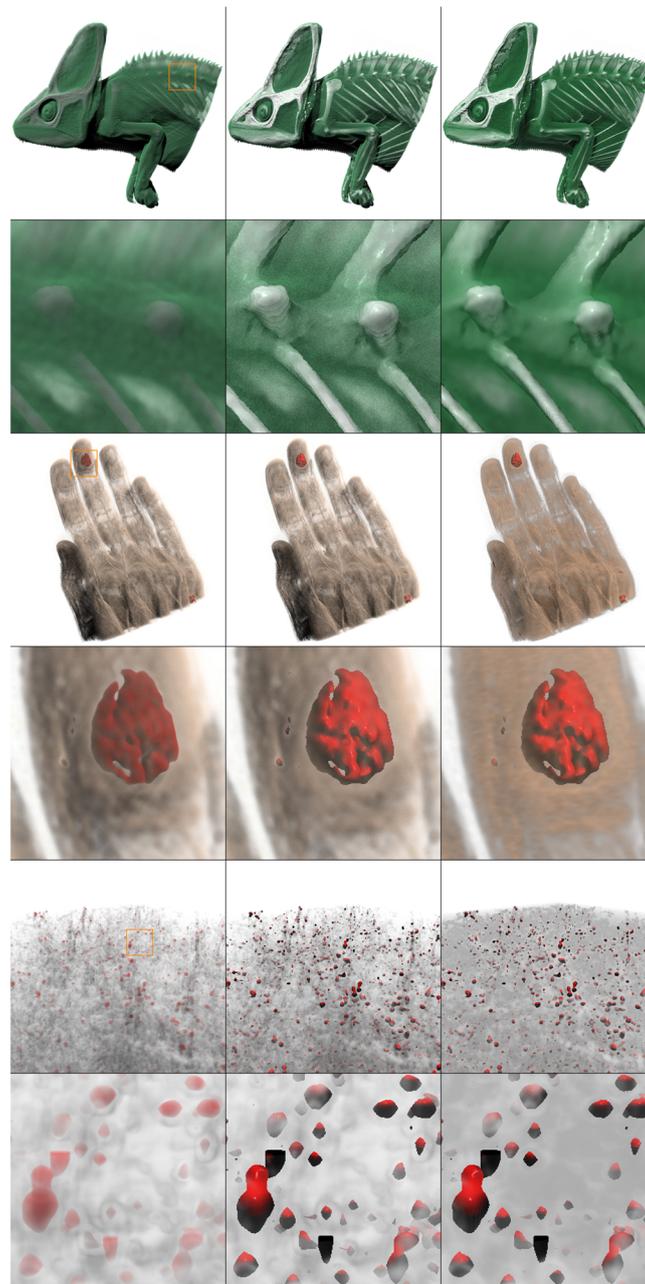


Fig. 9 Comparison of converged results (odd rows) and their zoom-ins (even rows) of path tracing (1st column), non-caching variant (2nd column), and caching variant (3rd column) for Chameleon (top), Hand (middle) and Virgo cluster (bottom) datasets

gradually take effect.

Our tests reveal comparable quality between the caching and non-caching variants of our method on the test volumes. While there are bigger differences at the start of the rendering (in the first 5 s), the results are later on par with each other (see Figs. 5, 7). Since memory consumption remains the single largest differentiating factor between the two variants, the decision about which variant to use can be resolved automatically by the application, given the available GPU memory and the size of the volume.

From Fig. 5, one can see that the shapes and their edges are much more pronounced when rendered with one of our techniques. More details can be seen on the isosurfaces in comparison with the path-traced image due to the local illumination. Both properties are beneficial for the perception of the structures within the volume, especially in cases where path tracing can substantially obscure them (see the structures inside the eye of the rainbow wrasse). The second row shows that the shadows are preserved with all techniques. Since shadows are essential for spatial per-

ception, this means that our method is not inferior in this regard. Moreover, since the volume and isosurface illumination are stored separately, the method can easily be extended by adding support for enhancing either of these two contributions, which may be valuable in certain application scenarios.

While there are smaller differences between the results in Fig. 5, we can see a bit more pronounced differences between the caching and non-caching variants of the presented technique in the example shown in Fig. 6, where the caching variant exhibits notably less contrast in the semi-transparent regions than the non-caching variant. We suspect that this is a consequence of the compositing step, where we use ray marching, which may significantly underestimate the transmittance due to Jensen's inequality. There are no noticeable differences between the results using different resolutions of the global illumination volume, which could be caused by light leaking.

The comparison of the presented methods to traditional volumetric path tracing for rendering data from various domains highlights both their advantages and disadvantages.

With regards to the Chameleon dataset Fig. 9, top, the presented methods demonstrate a clear improvement over regular volumetric path tracing. The bones exhibit more distinct features in our results compared to volumetric path tracing, and these results could not be achieved even with extensive modification of the transfer function.

A similar pattern is observed for the Hand dataset Fig. 9, middle, where our methods produce more pronounced tumor details compared to volumetric path tracing. The downside is that the cached variant of our method is not capable of preserving shadowing details, which are retained in both the non-cached version of our method and the traditional volumetric path tracing.

However, the Virgo cluster simulation dataset Fig. 9, bottom reveals a limitation of our method in rendering small objects with intricate shadow details. The non-cached version of our method provides the best depth perception, with distant objects occluded by the participating gas clouds. The structure of the gas clouds is best visible in both volumetric path tracing and the non-cached variant of our method, but is lost in the cached method due to the absence of shadows.

6 Conclusion

In this work, we presented two variants of a novel technique of combined volume and surface rendering, a non-caching and caching one, of which the latter enables real-time rendering whereas the former does not. We evaluated both techniques, showing their suitability for interactive use. The resulting images show that the quality of the approaches is comparable, with only minor differences in performance. Additionally, the caching variant allows for interaction with

the camera without having to recompute the image from scratch.

As a future extension of the presented technique, we will consider support for multiple isosurface rendering. For complete global illumination, one could also extend the method to take into account the parts of the volume within the regions enclosed by the isosurface. One could adapt the technique to exploit different global illumination techniques such as diffusion and incorporate a different composition of volume and surface rendering. Furthermore, the whole rendering pipeline could be replaced with by combining a NeRF-based approach [34] designed for surface rendering and Deep Direct Volume Rendering [51] for volume rendering. Further improvements could also target isosurface extraction.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00371-023-02932-9>.

Funding The authors did not receive any financial support for the submitted work.

Data availability statement Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors have no financial or proprietary interests in any material discussed in this article. Uroš Šmajdek declares that he/she has no conflict of interest. Author B declares that he/she has no conflict of interest. Žiga Lesar declares that he/she has no conflict of interest. Author B declares that he/she has no conflict of interest. Matija Marolt declares that he/she has no conflict of interest. Author B declares that he/she has no conflict of interest. Uroš Šmajdek declares that he/she has no conflict of interest. Ciril Bohak declares that he/she has no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Andersen, T.G., Falster, V., Frisvad, J.R., Christensen, N.J.: Hybrid fur rendering: combining volumetric fur with explicit hair strands. *Vis. Comput.* **32**(6), 739–749 (2016). <https://doi.org/10.1007/s00371-016-1252-x>

2. Bailey, D.L., Maisey, M.N., Townsend, D.W., Valk, P.E.: Positron Emission Tomography, vol. 2. Springer (2005)
3. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. *Comput. Graph. Forum* **36**(1), 301–329 (2016). <https://doi.org/10.1111/cgf.12802>
4. Bloomenthal, J.: IV.8.—an implicit surface polygonizer. In: Heckbert, P.S. (ed.) *Graphics gems*, pp. 324–349. Academic Press (1994). <https://doi.org/10.1016/B978-0-12-336156-1.50040-9>
5. Bosma, M.K., Smit, J., Lobregt, S.: Iso-surface volume rendering. In: Kim, Y., Mun, S.K. (eds.) *Medical Imaging 1998: Image Display*, vol. 3335, pp. 10–19. International Society for Optics and Photonics, SPIE (1998). <https://doi.org/10.1117/12.312490>
6. Bruckner, S., Gröller, M.E.: Style transfer functions for illustrative volume rendering. *Comput. Graph. Forum* **26**(3), 715–724 (2007). <https://doi.org/10.1111/j.1467-8659.2007.01095.x>
7. Burley, B., Studios, W.D.A.: Physically-based shading at Disney. In: *ACM SIGGRAPH*, vol. 2012, pp. 1–7 (2012)
8. Chandrasekhar, S.: *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications (1960)
9. Doi, A., Koide, A.: An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE Trans. Inf. Syst.* **E74-D**(1), 214–224 (1991)
10. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume rendering. *ACM Siggraph Comput. Graph.* **22**(4), 65–74 (1988). <https://doi.org/10.1145/378456.378484>
11. Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C., Weiskopf, D.: *Real-time volume graphics*. A K Peters/CRC Press (2006). <https://doi.org/10.1201/b10629>
12. Fong, J., Wrenninge, M., Kulla, C., Habel, R.: Production volume rendering. In: *ACM SIGGRAPH*, pp. 1–79. ACM Press, New York (2017). <https://doi.org/10.1145/3084873.3084907>
13. Foster, M.A.: Magnetic resonance in medicine and biology. *Prog. Nucl. Med.* **8**, 1–180 (1984)
14. Galtier, M., Blanco, S., Caliot, C., Coustet, C., Dauchet, J., El Hafi, M., Eymet, V., Fournier, R., Gautrais, J., Khuong, A., et al.: Integral formulation of null-collision Monte Carlo algorithms. *J. Quant. Spectrosc. Radiat. Transfer* **125**, 57–68 (2013)
15. Hart, J.C.: Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *Vis. Comput.* **12**(10), 527–545 (1996). <https://doi.org/10.1007/s003710050084>
16. Huang, Q., Zeng, Z.: A review on real-time 3D ultrasound imaging technology. *BioMed Res. Int.* (2017). <https://doi.org/10.1155/2017/6027029>
17. Iglesias-Guitian, J.A., Mane, P.S., Moon, B.: Real-time denoising of volumetric path tracing for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.* (2020). <https://doi.org/10.1109/TVCG.2020.3037680>
18. Isenberg, T., Neumann, P., Carpendale, S., Sousa, M.C., Jorge, J.A.: Non-photorealistic rendering in context: an observational study. In: *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 115–126 (2006). <https://doi.org/10.1145/1124728.1124747>
19. Jarosz, W., Donner, C., Zwicker, M., Jensen, H.W.: Radiance caching for participating media. *ACM Trans. Graph.* (2008). <https://doi.org/10.1145/1330511.1330518>
20. Kajiya, J.T.: The rendering equation. *SIGGRAPH Comput. Graph.* **20**(4), 143–150 (1986). <https://doi.org/10.1145/15886.15902>
21. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. *SIGGRAPH Comput. Graph.* **18**(3), 165–174 (1984). <https://doi.org/10.1145/964965.808594>
22. Kalender, W.A., Seissler, W., Klotz, E., Vock, P.: Spiral volumetric CT with single-breath-hold technique, continuous transport, and continuous scanner rotation. *Radiology* **176**(1), 181–183 (1990). <https://doi.org/10.1148/radiology.176.1.2353088>
23. Kawata, S., Minami, S.: The principle and applications of optical microscope tomography. *Acta Histochem. Cytochem.* **19**(1), 73–81 (1986). <https://doi.org/10.1267/ahc.19.73>
24. Khlebnikov, R., Voglreiter, P., Steinberger, M., Kainz, B., Schmalstieg, D.: Parallel irradiance caching for interactive Monte-Carlo direct volume rendering. *Comput. Graph. Forum* **33**(3), 61–70 (2014). <https://doi.org/10.1111/cgf.12362>
25. Kirk, D.B.: The simulation of natural features using cone tracing. *Vis. Comput.* **3**(2), 63–71 (1987). <https://doi.org/10.1007/BF02153662>
26. Koning, R.I., Koster, A.J.: Cryo-electron tomography in biology and medicine. *Ann. Anat.-Anat. Anz.* **191**(5), 427–445 (2009). <https://doi.org/10.1016/j.aanat.2009.04.003>
27. Kroes, T., Post, F.H., Botha, C.P.: Exposure render: an interactive photo-realistic volume rendering framework. *PLoS One* **7**(7), 1–10 (2012). <https://doi.org/10.1371/journal.pone.0038586>
28. Levoy, M.: Display of surfaces from volume data. *IEEE Comput. Graph. Appl.* **8**(3), 29–37 (1988). <https://doi.org/10.1109/38.511>
29. Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., Manohara, M.: Toward a practical perceptual video quality metric. *Netflix Tech Blog* **6**(2) (2016)
30. Ljung, P., Krüger, J., Groller, E., Hadwiger, M., Hansen, C.D., Ynnerman, A.: State of the art in transfer functions for direct volume rendering. *Comput. Graph. Forum* **35**(3), 669–691 (2016). <https://doi.org/10.1111/cgf.12934>
31. Lokovic, T., Veach, E.: Deep shadow maps. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pp. 385–392. ACM Press/Addison-Wesley Publishing Co., USA (2000). <https://doi.org/10.1145/344779.344958>
32. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comput. Graph.* **21**(4), 163–169 (1987). <https://doi.org/10.1145/37401.37422>
33. Max, N.: Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.* **1**(2), 99–108 (1995). <https://doi.org/10.1109/2945.468400>
34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: *European Conference on Computer Vision*, pp. 405–421. Springer (2020). https://doi.org/10.1007/978-3-030-58452-8_24
35. Novák, J., Georgiev, I., Hanika, J., Jarosz, W.: Monte Carlo methods for volumetric light transport simulation. *Comput. Graph. Forum* **37**(2), 551–576 (2018). <https://doi.org/10.1111/cgf.13383>
36. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. In: *ACM SIGGRAPH 2005 Courses, SIGGRAPH '05*, p. 173–180. Association for Computing Machinery, New York (2005). <https://doi.org/10.1145/1198555.1198649>
37. Šoltészová, V., Patel, D., Bruckner, S., Viola, I.: A multidirectional occlusion shading model for direct volume rendering. *Comput. Graph. Forum* **29**(3), 883–891 (2010). <https://doi.org/10.1111/j.1467-8659.2009.01695.x>
38. Parker, S., Parker, M., Livnat, Y., Sloan, P.P., Hansen, C., Shirley, P.: Interactive ray tracing for volume visualization. In: *ACM SIGGRAPH 2005 Courses*, pp. 15–es. ACM, New York (2005). <https://doi.org/10.1109/2945.795215>
39. Parker, S., Shirley, P., Livnat, Y., Hansen, C., Sloan, P.P.: Interactive ray tracing for isosurface rendering. In: *Proceedings Visualization '98 (Cat. No. 98CB36276)*, pp. 233–238. IEEE (1998). <https://doi.org/10.1109/VISUAL.1998.745713>
40. Ribardière, M., Carré, S., Bouatouch, K.: Adaptive records for volume irradiance caching. *Vis. Comput.* **27**(6), 655–664 (2011). <https://doi.org/10.1007/s00371-011-0573-z>
41. Ruiz, M., Szirmay-Kalos, L., Umenhoffer, T., Boada, I., Feixas, M., Sbert, M.: Volumetric ambient occlusion for volumetric mod-

- els. *Vis. Comput.* **26**(6), 687–695 (2010). <https://doi.org/10.1007/s00371-016-1252-x>
42. Rushmeier, H.E., Torrance, K.E.: The zonal method for calculating light intensities in the presence of a participating medium. *SIGGRAPH Comput. Graph.* **21**(4), 293–302 (1987). <https://doi.org/10.1145/37402.37436>
 43. Schott, M., Pegoraro, V., Hansen, C., Boulanger, K., Bouatouch, K.: A directional occlusion shading model for interactive direct volume rendering. *Comput. Graph. Forum* **28**(3), 855–862 (2009). <https://doi.org/10.1111/j.1467-8659.2009.01464.x>
 44. Sobierajski, L.M., Kaufman, A.E.: Volumetric ray tracing. In: *Proceedings of the 1994 Symposium on Volume Visualization*, pp. 11–18 (1994). <https://doi.org/10.1145/197938.197949>
 45. Taylor, R.: FRELLED: a realtime volumetric data viewer for astronomers. *Astron. Comput.* **13**, 67–79 (2015). <https://doi.org/10.1016/j.ascom.2015.10.002>
 46. Tiede, U., Hoehne, K., Bomans, M., Pommert, A., Riemer, M., Wiebecke, G.: Investigation of medical 3D-rendering algorithms. *IEEE Comput. Graph. Appl.* **10**(2), 41–53 (1990). <https://doi.org/10.1109/38.50672>
 47. Tietjen, C., Isenberg, T., Preim, B.: Combining silhouettes, surface, and volume rendering for surgery education and planning. In: *EuroVis*, pp. 303–310 (2005). <https://doi.org/10.2312/VisSym/EuroVis05/303-310>
 48. Treese, G.M., Prager, R.W., Gee, A.H.: Regularised marching tetrahedra: improved iso-surface extraction. *Comput. Graph.* **23**(4), 583–598 (1999). [https://doi.org/10.1016/S0097-8493\(99\)00076-X](https://doi.org/10.1016/S0097-8493(99)00076-X)
 49. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>
 50. Weber, C., Kaplanyan, A., Stamminger, M., Dachsbacher, C.: Interactive direct volume rendering with many-light methods and transmittance caching. In: *VMV*, pp. 195–202 (2013). <https://doi.org/10.2312/PE.VMV.VMV13.195-202>
 51. Weiss, S., Chu, M., Thuerey, N., Westermann, R.: Volumetric iso-surface rendering with deep learning-based super-resolution. *IEEE Trans. Vis. Comput. Graph.* **27**(6), 3064–3078 (2021). <https://doi.org/10.1109/TVCG.2019.2956697>
 52. Woodcock, E., Murphy, T., Hemmings, P., Longworth, S.: Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In: *Proc. Conf. Applications of Computing Methods to Reactor Problems*, pp. 557–579 (1965)
 53. Xu, C., Sun, G., Liang, R.: A survey of volume visualization techniques for feature enhancement. *Vis. Informat.* **5**(3), 70–81 (2021). <https://doi.org/10.1016/j.visinf.2021.08.001>
 54. Yue, Y., Iwasaki, K., Chen, B.Y., Dobashi, Y., Nishita, T.: Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Trans. Graph. (TOG)* **29**(6), 1–8 (2010). <https://doi.org/10.1145/1882261.1866199>
 55. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595. IEEE (2018). <https://doi.org/10.1109/CVPR.2018.00068>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Uroš Šmajdek is a M.Sc. student at the Faculty of Computer and Information Science, University of Ljubljana. His interests lie mainly in the fields of computer graphics and game technology.



Žiga Lesar is a Ph.D. student, a researcher and a teaching assistant at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. He received his B.Sc. in 2014 and M.Sc. in 2018 for his work on interactive volume rendering with web technologies. His research is focused primarily on interactive computer graphics, especially volume rendering and visualization.



Matija Marolt is an associate professor at the Faculty of Computer and Information Science, University of Ljubljana. His research is focused on segmentation and reconstruction of biological and medical data, obtaining information from music with emphasis on semantic descriptions and understanding of sound signals, acquisition and organization of music archives and human–computer interaction.



Ciril Bohak is a research scientist at King Abdullah University of Science and Technology (KAUST), Saudi Arabia, and an assistant professor in the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. He received his B.Sc., M.Sc., and Ph.D. degree from the University of Ljubljana. His research covers computer graphics, scientific visualization, and human–computer interaction.