WYSIWYP: What You See Is What You Pick

Alexander Wiebel, *Member, IEEE*, Frans M. Vos, David Foerster, and Hans-Christian Hege, *Member, IEEE*

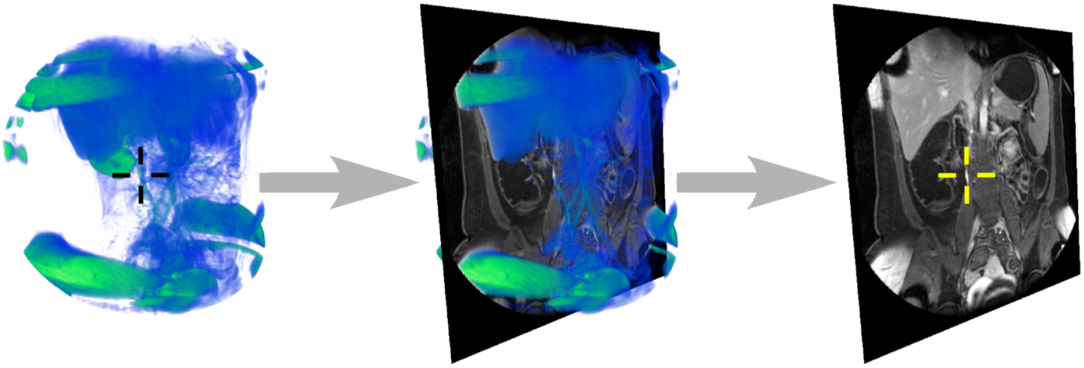

Fig. 1. Visibility-based picking in volume rendering: The user clicks on a visible structure in the volume rendering (left image), a slice crossing this structure and oriented according to the viewing direction is automatically selected and displayed (middle). With an optional second click the volume rendering is switched off to provide an unobstructed view of the slice (right). The slice can then be moved using standard navigation mechanisms. The data set used in this example is from abdominal magnetic resonance imaging.

Abstract—Scientists, engineers and physicians are used to analyze 3D data with slice-based visualizations. Radiologists for example are trained to read slices of medical imaging data. Despite the numerous examples of sophisticated 3D rendering techniques, domain experts, who still prefer slice-based visualization do not consider these to be very useful. Since 3D renderings have the advantage of providing an overview at a glance, while 2D depictions better serve detailed analyses, it is of general interest to better combine these methods. Recently there have been attempts to bridge this gap between 2D and 3D renderings. These attempts include specialized techniques for volume picking in medical imaging data that result in repositioning slices. In this paper, we present a new volume picking technique called WYSIWYP ("what you see is what you pick") that, in contrast to previous work, does not require pre-segmented data or metadata and thus is more generally applicable. The positions picked by our method are solely based on the data itself, the transfer function, and the way the volumetric rendering is perceived by the user. To demonstrate the utility of the proposed method, we apply it to automated positioning of slices in volumetric scalar fields from various application areas. Finally, we present results of a user study in which 3D locations selected by users are compared to those resulting from WYSIWYP. The user study confirms our claim that the resulting positions correlate well with those perceived by the user.

Index Terms—Picking, volume rendering, WYSIWYG.

1 Introduction

Direct volume rendering (DVR) [30] is the state-of-the-art for the display of volumetric data from medicine, engineering and natural sciences. As a flexible and versatile tool, it is adaptable to virtually all application problems dealing with 3D scalar fields. The latest hardware developments allow DVR to be used interactively even on consumer type systems. Although this makes it available for the analysis and inspection of volumetric data, physicians, scientists and engineers still rely mainly on the examination of slice-like depictions (including multi-planar reformatting, MPR). Motivated by this fact previous work has already addressed the combination of DVR and MPR representations [13], [14], [15], [37]. By providing interaction techniques (commonly called volume picking, point picking or volume pinpointing) that allow users to pick in the volumetric rendering to adjust a slice, and vice versa to pick on the slice to reorient the DVR, it is possible to integrate DVR in the daily routine of physicians, engineers and scientists dealing with three-dimensional scalar data. DVR can serve as an overview, while the slices are still used for the detailed examinations.

With this background in mind, the motivation for the technique presented in this paper is to overcome the three limitations of current methods, which either require metadata, are designed for medical data only, or provide only very basic picking techniques like *first-hit* or opacity-threshold. The basic assumption of this paper is that the user wants to examine structures that can be made visible with DVR and suitable transfer functions. We aim to introduce a picking technique which takes a visibility-based view and overcomes the mentioned limitations. Picking is probably the most intuitive interaction technique, as it is the technical equivalent of one of the most natural actions in the real world: *pointing* at something that we *see*.

We present a method, called WYSIWYP ("**w**hat **y**ou **s**ee **i**s **w**hat **y**ou **p**ick"), which enables users to intuitively select spatial positions in volumetric renderings. Target users of WYSIWYP are all users of DVR from all backgrounds. In particular, the main contributions of this paper are

- a new technique allowing users to pick 3D structures visible in their direct volume rendering images,
- the technique's independence of any information apart from the volume data and the transfer function of the direct volume rendering,

- *Alexander Wiebel, David Foerster, and Hans-Christian Hege are with Zuse Institute Berlin (ZIB), e-mail: {wiebel, foerster, hege}@zib.de.*
- *Frans M. Vos is with TU Delft and AMC Amsterdam, Netherlands, e-mail: F.M.Vos@tudelft.nl.*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

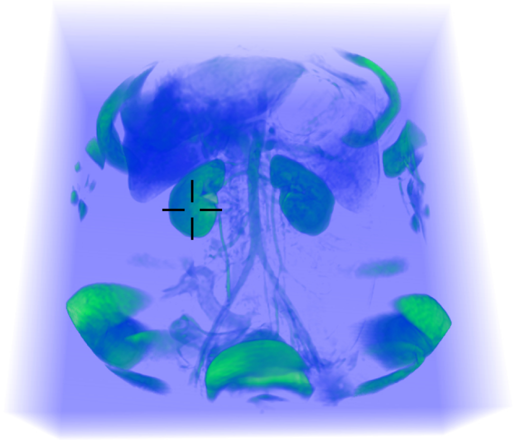

Fig. 2. Problem of *first-hit* method with zero threshold and "foggy" rendered image. The resulting position of picking the location marked with the crosshairs will be on the bounding box of the dataset instead of on the kidneys because every position in the volume has non-zero opacity.

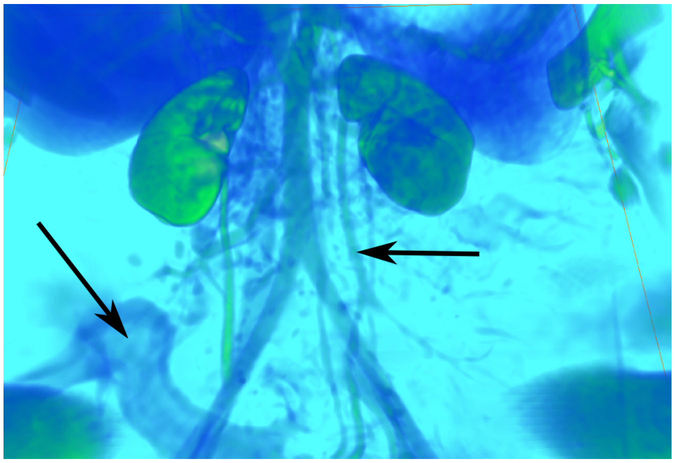

Fig. 3. Problem of threshold picking method with relatively transparent regions. Example regions (vessel, terminal ileum) are marked with arrows. Due to a high threshold such regions might be missed although they are clearly visible. The cast ray will reach the end of the dataset without the accumulated opacity exceeding the threshold. Thus the ray's exit position will be picked instead of the clearly visible features.

- application of the technique to renderings for any volumetric scalar dataset and all types of transfer functions (e.g. "foggy" looking images),
- its usefulness for navigating (e.g. selecting slices) in the resulting visualizations,
- a comparison of WYSIWYP to other techniques,
- and a user study that supports our claims by comparing user selected 3D locations to locations provided by WYSIWYP.

We do not aim to replace well established picking techniques, but rather see WYSIWYP as a complementary technique in cases where users would like to pick what they actually see in the *rendering* instead of what is known to be in the *data*. We therefore intend to pave the way for further application of DVR in application areas that are still reluctant to adopt this fundamental visualization technique.

2 Related Work

In this section we review the previous work on picking in volumetric renderings and the combination of direct volume rendering with slices. There has been a lot of work on perception and visibility in visualization research [12], [35], however, this work is mainly concerned with designing interaction for visualization according to perception principles, rather than analyzing how volumetric visualizations are perceived.

2.1 Picking

Direct volume rendering has been around for over twenty years now [30] and over time has developed into an interactively usable rendering technique [1], [16]), which has resulted in research that aims at facilitating the interaction with volumetric depictions. Volume picking, the interaction technique that is the focus of this article, has been adapted from its well-known predecessor which is used for picking real geometry like surfaces. Accordingly the first volume picking techniques mimicked the surface picking by searching for the first surface-like structure along the viewing ray passing through the picked screen position. Gobbetti et al. [11] introduced the most widely used technique. It searches along the ray using the usual compositing scheme (described in Section 3) and stops as soon as the accumulated opacity exceeds a user defined threshold. This means a surface is assumed to be at locations where the opacity threshold is exceeded. The endpoint of the search is returned as a 3D position resulting from the picking. A simplified version of this approach sets the threshold to zero. This results in selecting the first position in the volume which is not completely transparent (*first-hit*). Both variants may produce undesirable results. Using a zero threshold will return positions in regions surrounding the features in "foggy" looking renderings (see Figure 2). If on the other hand the threshold is non-zero some relatively transparent, but still visible, regions may be missed (see Figure 3).

Another widely used method selects the largest data value along the ray. While yielding perfect results in conjunction with maximum intensity projection renderings, this technique is not suitable for DVR in general. For common DVR it can result in selecting positions that are completely transparent, i.e. deliberately not shown, due to the selected transfer function. Toennies and Derz [34] present a technique that searches for user-defined data values or user-defined properties of metadata along the ray. In our setting, it suffers from the same problem as the previously described method. Bruckner et al. [7] select the position along the ray that contributes most to the final pixel. They report that it works well with the special volume rendering technique they used in their *BrainGazer* system. As the sample contributing most to the final pixel does not necessarily belong to the most visible *object*, i.e. the group of samples contributing most to the final pixel, their method can yield undesirable results. Furthermore, as it considers only one sample, the method can result in positions that lie at some arbitrary location in the perceivable structures instead of at the front or center of the perceived structure.

The following list of visualization tools and their volume picking techniques gives an impression of the use of volumetric picking techniques: MeVisLab [24] provides a technique selecting the maximum data value as well as opacity threshold-based picking. Voreen [22] and Avizo [3] use the first-hit approach. VTK [31] and thus ParaView [32] employ the opacity threshold method [25].

Kohlmann et al. [15] employ a more sophisticated picking method called *contextual picking* that is especially tailored to medical data in DICOM [4] format. It uses the meta information given in the DICOM files to deduce which anatomical parts of the volumetric image the user intends to pick (e.g. angiography→vessels). Very few, initially user specified, ray profile samples are matched against the data curve along the viewing ray to find the intended structures. As the matching identifies the approximate extent of the picked structure, Kohlmann et al. are able to provide picking positions either on the front of the structure or in its center. Malik et al. [18] use ray-profiles similarly in a different context, the division of the data into different *peelable* layers. Like our method, they use the derivatives of a ray-profile to find "features" along the ray. However, they search for features in the data whereas our method searches for features in the visible rendering (profile of data vs. profile of accumulated opacity along the ray). Additionally, in contrast to our method their *transition points* are extrema and are thus easily detected as zeros of the first derivative. Another peeling technique somewhat related to the present work is the so-called opacity peeling by Rezk-Salama and Kolb [29]. Opacity peeling uses several rounds of opacity accumulation (each up to some threshold) to render layers originally occluded behind other rendered layers of the data. Users can interactively select the layer they would like to see.

In contrast to most techniques discussed above, which only select one position in 3D, Owada et al. [26] present a method that manages to locate the intended location of a 2D stroke in 3D space. Their technique uses the determined 3D line as the basis for segmenting a 3D object marked by the stroke and thus the 3D line. For this task the stroke has to follow the border or contour of the object visible in the rendering. Determining the location of the 3D line translates to the task of finding the 3D line that is a projection of the 2D stroke and fits contours in the data best (dot product of normal vector of extruded stroke and gradient in data). We consider their aim and method to be different from ours for two reasons: Firstly it only works at contours of objects, whereas our method is applicable to any position in the rendering. Secondly, their method segments objects whose contour best fits the stroke. Such objects are not necessarily those that we aim at, i.e. the most visible ones. In the same spirit as Owada's technique for DVR, Yu et al. [40] describe a lasso-based technique to spatially select 3D subsets in massive particle cloud datasets.

One of the most recent papers on picking in volumetric rendering we are aware of is by Peng et al. [27]. They use two different techniques: (1) a one-click method restricted to their data with a blob-like structure where it is easy to guess the desired position as the center of the blob hit by the viewing ray; (2) a two-click method where the user clicks on the desired location from two different viewing directions. The picked position is then the (fuzzy) intersection of the two viewing rays. This method works for arbitrary data, but the desired location has to be visible from both viewing directions. The method we present is superior to Peng's method in so far as it provides picking in renderings of arbitrary data with only one mouse click (or a similar pointing action) from one viewing direction.

Picking is also a very important interaction technique in virtual environments. Argelaguet and Andujar [2] describe one method and give a good overview of other techniques with references in their paper. The main difference between all the techniques mentioned by Argelaguet and Andujar and our work is that the picking is targeted to real geometric objects and not volume renderings.

An extensive list of different three-dimensional rendering techniques and how picking can be implemented in their context can be found in a recent technical report by Wiebel et al. [39, Sec.3]. The list also contains comments on how exact the respective picking methods are.

2.2 Combining Slices and DVR

Many techniques dealing with isosurfaces in volumetric data provide picking on slices for reorienting the isosurface to a view point offering good visibility of the selected position. Picking on the surface is often used to change the position of a slice in the data. Recently this has been combined with picking in volume renderings by Kohlmann et al. [13], [14] and others [6], [37]. In their work picking on a slice results in a reorientation of the volume rendering and a local adaption of the opacity (transfer function) such that the view on the selected position is improved. For the reverse direction, i.e. for picking in the volume and adapting the position of the slice, they use either the first-hit or their contextual picking approach [15] that we described above. In their framework the selected position can also be used for placing labels.

In addition to placing slices according to a certain user interaction, there are also methods which allow the automatic placement of cross sections at interesting locations in the 3D data. See for example the method by Mori et al. [23].

2.3 Metadata

Medical datasets often come with metadata that can help to infer the position a user intends to pick. If metadata are available, it makes sense to exploit the information provided by them, and, in fact, some of the above described methods do so [15], [34]. We nevertheless consider WYSIWYP's independence of metadata to be one of its important features because many datasets, particularly from outside the medical domain, come without metadata. In such cases the metadata-based approaches can not be applied. Our method, in contrast, is applicable to any data. Examples of fields where the data often lacks meta information are flow simulations as well as any types of *derived* density fields e.g. from a 3D scatter plot.

For medical data with suitable meta information we suggest allowing the users to interactively choose whether they would like to pick only positions with an a-priori known meaning (metadata) or if they would like to pick the features they see in the actual rendering (WYSIWYP). Their choice will vary with the task at hand.

3 Background

To understand the proposed picking approach, a basic understanding of the volume rendering procedure is necessary. We therefore give a summary of the most relevant aspects and in this way introduce the notation.

3.1 Volume Rendering Integral

As DVR tries to make volumetric data directly visible to the user, its most natural implementation is casting rays along the viewing direction through the volume and accumulating color information for the values of the volumetric data along these rays. The density of the rays and the samples along the rays are chosen to cover the volume sufficiently. The color information for the data values is determined by the *transfer function*. The mentioned accumulation can be formalized mathematically by the *volume rendering integral* [10], [21], [28], [30]:

$$I(r_{max}) = I_0 e^{\int_{r_0}^{r_{max}} \tau(t)dt} + \int_{r_0}^{r_{max}} Q(s) e^{\int_{s}^{r_{max}} \tau(t)dt} ds$$

In this equation, I is the intensity in a color channel resulting from accumulating the color for a certain distance along the ray. $[r_0, r_{max}]$ is an interval along the ray, with r_{max} being at the eye point and r_0 at the back end of the volume; s is a parameter in this interval; τ is the attenuation coefficient and Q the source term describing emission for a certain sample.

For a numerical approximation the volume rendering integral has to be discretized: compositing (accumulation) is performed for a finite number of samples along the ray. The iterative computation of the discretized version in a *front-to-back* fashion can be denoted as follows [9], [10]:

$$c_{n+1}^{acc} = c_n^{acc} + (1 - \alpha_n^{acc}) c_n^{src} \quad (1)$$

$$\alpha_{n+1}^{acc} = \alpha_n^{acc} + (1 - \alpha_n^{acc}) \alpha_n^{src} \quad (2)$$

Here, c denotes color, α denotes opacity, n denotes the step number, *acc* indicates the accumulated values and *src* indicates values of the transfer function for the data found at the current sample position.

3.2 Compositing

Equation 2 describes the steps that have to be performed to compute the opacity at a certain sample on the ray. This opacity is accumulated along the ray up to that position. It determines how much the final pixel value is influenced by the values of the samples on the ray that lie behind the current sample. Later in this paper we will discuss how α^{acc} varies along the ray. Therefore it is worth noting three important properties of α^{acc} that can be easily deduced from Equation 2 by mathematical induction: first, the accumulated opacity will never be larger than one. Second, α^{acc} is monotonically increasing along the ray, and, third, $\alpha_n^{acc} \in [0,1]$.

Another important fact about the compositing in Equation 2 is that with a change of the sampling density along the ray the series of accumulated opacity changes. More samples result in a faster increasing

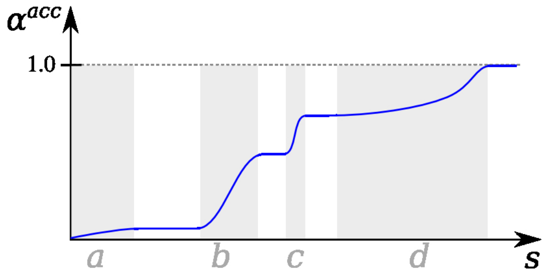

Fig. 4. Jumps in accumulated opacity α^{acc} along the ray (parameter s). The gray areas are the intervals used in equation 3. The parameters i_0 and i_{max} in the same equation correspond to the left and right borders of the gray regions respectively. Note that the jump denoted c is steeper than jump b, but that b is higher than c. The increase in interval d represents the feature with largest extent while contributing only a small amount to the overall opacity.

opacity compared to the location of the samples along the ray. This can be compensated by scaling α^{src} with respect to the sample density (*opacity correction*) [10], [36]. Opacity correction is also necessary if non-equidistant samples are used. For sake of simplicity we restrict all explanations to equidistant samples throughout this paper. All presented methods are easily extendable to the general case.

4 WYSIWYP

In this section we give a detailed description of the new *visibility-oriented picking* technique WYSIWYP. A comparison with previous techniques which emphasizes its advantages is provided in Section 6.

The overall procedure of all picking techniques is similar. First, the user clicks on a position in the screen. This position and the user's viewing direction are transformed from screen coordinates into world coordinates. The result is then used to cast a ray through the scene (see Figure 6). Along this ray a number of samples are used to gather information about the volume data. Finally certain criteria are applied to the gathered data to determine the position resulting from the pick. This last step is the one that the new method focuses on.

4.1 Visibility-Oriented Picking Criterion

At the heart of the new technique are the characteristics of the values of α^{acc} along the viewing ray, i.e. the discretized version of the opacity accumulation described by the volume rendering integral. Previous work on volume rendering already noted that opacity (resp. accumulated opacity) along the ray is strongly correlated with the *visibility* of positions (resp. regions or features) in the volume [5, Sec.3.2], [8, Sec.3], [38, Sec.3]. Consequently, we hypothesize (confirmed by our experiments, Section 5) that the user usually *perceives* those features at a screen position which contribute the highest amount of opacity or in other words, the highest *jump* of α^{acc} along the ray (Figure 4). The amount of opacity contribution of a spatial feature determines its influence on the final color of the pixel and thus defines which feature is perceived. This means that an object's visibility does not only depend on the optical properties of a single location but on the properties of a number of consecutive locations. Furthermore as all our examples confirm, it does not seem to depend on the steepness of the increasing opacity, but on how much the opacity increases in an interval of consecutive samples, i.e. on how large the contribution of the interval to the final is. An example of this effect can be seen in Figure 12. Finally, it is sensible to consider the changes of opacity for selecting the picked position because high opacity is usually assigned to important features during transfer function design, in short: opacity correlates to importance.

Figure 4 illustrates the criterion. Here, the largest jump can be found in region b, while the the steepest jump, appears in region c. Consequently, the region used to determine the picking position is b.

To determine the highest jump, the first task is to define the regions of Figure 4 as *intervals* $I = [i_0, i_{max}] \subset [r_0, r_{max}]$ along the ray.

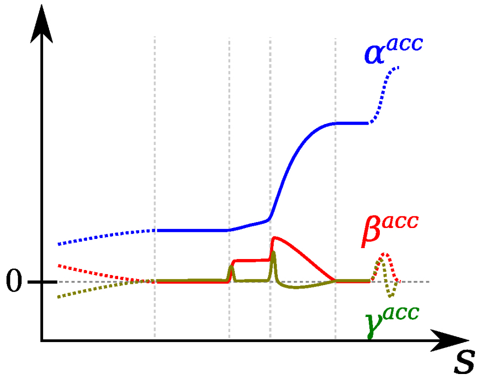

Fig. 5. Detection of borders of jumps via first and second derivative of α^{acc} along the ray (parameter s). The blue curve represents the accumulated opacity, the red curve its first derivative β^{acc} and the green curve its second derivative γ^{acc}. The dashed gray lines mark the detected borders. The curves are only sketched for illustration purposes and thus are only qualitatively correct.

Thereafter, the difference between α^{acc} at the start and the end of the interval, i.e. the jump j as

$$j = \alpha^{acc}(i_0) - \alpha^{acc}(i_{max}), \tag{3}$$

has to be computed. Extracting the boundaries i_0 and i_{max} of the jumps is similar to the task of edge detection [19] in one dimension. Consequently, our method for detecting the boundaries is inspired by computer vision methods [20] and incorporates the second derivative of α^{acc}. We denote the first derivative of α^{acc} as β^{acc} and the second derivative as γ^{acc}. Figure 5 illustrates the idea behind our method for extracting the interval boundaries. In principle, the boundaries are the positions where the second derivative γ^{acc} crosses zero from below, i.e. from negative to positive values. This criterion, however, is only reliable if α^{acc} is strictly increasing. As α^{acc} has plateau like regions and thus γ^{acc} has extended regions where it is constantly zero, the criterion is adapted as follows. The lower bounds i_0 of such intervals are the positions where accumulated opacity starts to grow stronger, that is where γ^{acc} becomes positive after being negative or zero. The criterion for the upper bounds i_{max} is that α^{acc} stops decreasing again. For γ^{acc} this means that it becomes zero or positive after being negative.

After having determined the interval boundaries and having computed all jumps j one simply selects the interval with the largest jump j. This is the interval dominantly perceived at the picked screen position.

4.2 Front vs. Center of Perceived Feature

The criterion described above does not directly yield a position. It only yields the interval seen most prominently along the viewing ray through the picked screen position. This is not a problem but rather an advantage of the criterion because it allows to choose the final position according to the task at hand by application of further criteria. For labeling features in the volume rendering the front most position of the feature is of interest, whereas for repositioning slices to display most of the picked feature, the center of the feature is of interest. This has also been noted by Kohlmann et al. [15] and has been implemented for their contextual picking.

For WYSIWYP determining the center and the front position is straightforward because the front and back positions are implicitly computed as the start and end of the jump interval. A feature's front is simply the first position i_0 of the interval corresponding to the largest jump. A feature's approximate center is the center i_c of the interval, i.e. $i_c = \frac{1}{2}(i_0 + i_{max})$. Of course,other task and data specific criteria are conceivable. However, in our applications the described methods proved to be sufficient.

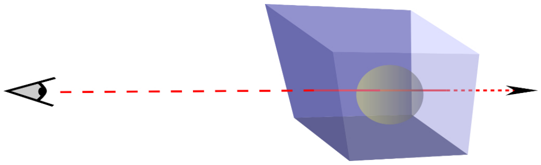

Fig. 6. Ray casting for WYSIWYP in DVR. The dashed line is the part of the ray that is determined by common object picking of the proxy geometry. Samples on the solid part of the ray are used for WYSIWYP. The dotted line is outside the bounding cube of the dataset. Stepping along the ray will be stopped before reaching it.

4.3 Implementation Details

In our implementation, casting the ray through the volume is realized by a combination of usual surface picking and straightforward ray casting on the CPU. We draw a transparent bounding cube (*proxy geometry*) around the volume rendered data in the scene. The standard geometry picking mechanism of the scene graph is then used to determine the position where the viewing ray intersects the proxy geometry and enters the data volume (see Figure 6). The direction of the ray is computed as the difference between the intersection point and the camera position or eye point. With this information we can step through the volume and gather the desired information. As soon as a step gets outside the dataset's bounding box we stop gathering information. The information obtained for each step is the data value $d(\mathbf{x})$ at the position and the result of applying the transfer function to this data value, i.e. color C_n^{src} and opacity α_n^{src}. Using Equation 2, these values are accumulated to provide the values of α^{acc} along the ray.

At this point it is worth noting that the parameters of the DVR implementation and of the procedures described above need to be coordinated. This is because usual compositing (Equation 2 and, e.g. reference [10]) does not consider the distance of the samples. Thus, if the sample distances of DVR and picking are not equal, the accumulated opacity may vary differently along the viewing ray. The following example demonstrates the possible issues. Consider the DVR using half the step size, i.e. twice as many samples, as the ray for the picking. The easiest way to achieve consistency between DVR and picking is to use the same number of steps and the same step size. If this cannot be achieved, then the previously mentioned *opacity correction* needs to be applied during compositing.

4.4 Transparent Surfaces

Figure 12, an example that will be described in detail in Section 6, suggests that WYSIWYP for transparent polygonal surfaces can be implemented analogous to the method for DVR. One simply traces a ray through the volume, accumulates or composits the opacities of the different polygons that are hit by the ray, and finally selects the intersection of the ray with the polygon that has the largest opacity contribution after compositing. Computing derivatives of α^{acc} is not necessary in this case because the intervals are the intersections of the ray with the surfaces, and thus infinitesimally small.

5 Evaluation

We conducted a user study to show the suitability and appropriateness of the presented picking method and to further support the claim that the parts contributing the most opacity are observed by the user. In this section, we first describe the experimental setup of the study that led to the results we present afterwards.

5.1 Experimental Setup

We set up the user study to determine the 3D position that the participants perceive at a certain 2D screen position. The positions recorded throughout the study can then be compared to the position the presented picking method selects automatically.

The general set-up of the test cases consists of a volume rendered image, small crosshairs indicating the currently considered position and a method to let the participants specify the actual 3D position they perceive. The latter is a slice that can be moved through the rendering by manipulating a slider until the slice intersects the 3D position perceived at the 2D location marked by the crosshairs. The setup is exemplified in Figure 7. The left image shows the initial state where only the rendering and the crosshairs are visible. The second image also contains the slice that has been placed by a participant. The slice appears as soon as the participant moves the slider the first time for a certain test case. Usually participants do not place the slice by a single slider movement, but move it back and forth in order to select the visible position in an exploratory process. In this process the part of the volume lying behind the slice is not visible anymore. This simplifies the task of position selection, as the participant can move the slice to the position where the observed feature just disappears. Please note that this does not influence the position which is actually marked as being perceived. Furthermore, simplifying the specification of positions does not influence the study because it is not our objective to judge the depth perception of the participants. Instead the intention is to get positions that can be compared with positions that have been automatically selected by picking.

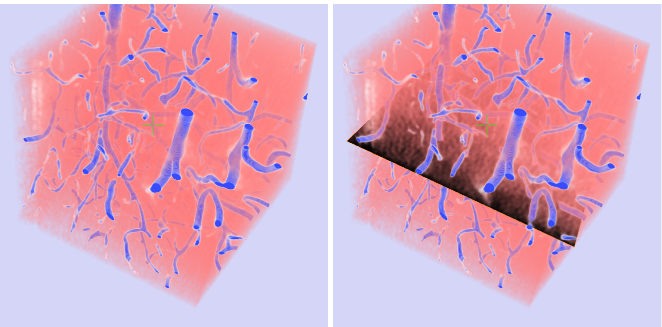

Fig. 7. Two images (taken directly from the user study) showing the steps of the evaluation. In the first image a scene showing capillary blood vessels in the brain with a position marked by crosshairs is given. The dataset is courtesy of MPI für Biologische Kybernetik Tübingen, AG Logothetis (Bruno Weber). The second image shows the situation after a participant has moved a slice to a position he/she perceives to intersect with the position marked with the crosshairs. In order to provide the best view of the marked features for the participant, the crosshairs are intentionally very thin. *If the crosshairs are not visible to you please zoom in using an electronic version of the paper.*

Excluding the pretests, 20 people participated in the study. Their ages ranged from 18 to 55 years (mean=30.95, median=32). Most participants were familiar with volumetric rendering at least superficially (14 professional, 3 student, 3 non-professional) and they had no known vision deficiencies or wore appropriate glasses or contact lenses. 7 females and 13 males participated in the study.

We chose nine datasets from different domains for the user study. Two datasets stem from abdominal MRI measurements (Figures 1, 2 and 3) conducted in the VIGOR++ project. They were used for four resp. three test cases. Three datasets from the volvis repository[1], i.e. Bonsai (see video), Hydrogen Atom and Nucleon (Figure 8), were used for ten test cases. From the downloadable example datasets of the osirix viewer[2] we used FELIX (cerebral aneurysm) for five test cases. Six more test cases use the *mouseCTdata* that comes with free trial versions of amira[3]. The dataset shown in Figure 7 was used in six test cases. The last dataset, used for two test cases, comes from a CT capturing bone structure. In total, we used 36 test cases. For each test case we defined a transfer function, a viewing direction and a position of interest. We mostly varied the viewing direction and the position of interest. For some datasets the transfer functions varied between test cases. All participants performed the same 36 tests. It took them an average of 18.01 minutes (σ =6.49 minutes, min=9.20

[1] http://www.volvis.org

[2] http://www.osirix-viewer.com/datasets/

[3] http://www.amira.com/downloads/trial.html

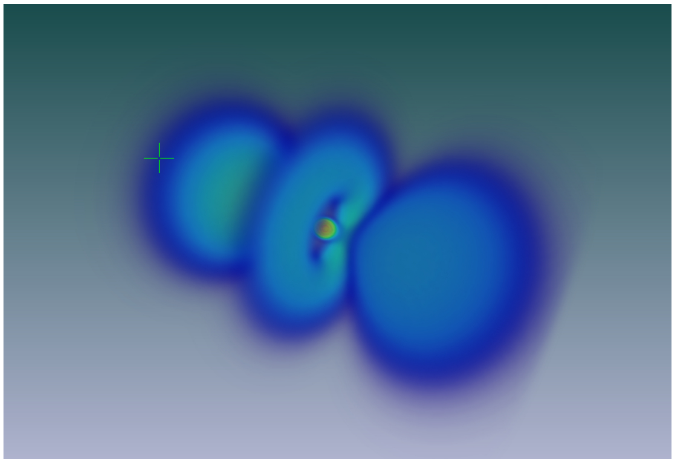

Fig. 8. The location of the position marked by the crosshairs is vague. Participants of the study marked very different locations along the viewing ray.

minutes, max=30.12 minutes) to complete all test cases. We however did not impose any time limit and were not specifically interested in the duration of the experiments.

5.2 Comparison to Picking Method

As mentioned before, the aim of the user study was to find out how close the position chosen by WYSIWYP is to the position of structures that humans perceive as essential. To quantify the answers to this question we performed some basic statistical analysis for each test case. As the selected positions of the participants naturally vary to some degree we computed the average μ of the positions chosen by the participants. This is simple and appropriate as all positions lie on the viewing ray. The average position μ is then compared to the position selected by WYSIWYP by computing their distance. The quality of the WYSIWYP position is then established by comparing its distance to the standard deviation σ of the distances of the positions picked by the participants. We consider WYSIWYP to perform well if the picked position lies in an interval of $[\mu-\sigma,\mu+\sigma]$ around the average position μ.

The actual analysis of the recorded experimental data and the positions picked by WYSIWYP yielded a correspondence of WYSIWYP and the perceived positions: For 89% (32 of 36) of the test cases the picked position lay inside the desired interval I of one standard deviation (see Figure 9 for an example). In four cases where the participants' choices varied strongly compared to the dataset size, the picking also did not perform well. See Figure 8 for an example where the choice of the participants varied strongly. The exact numbers for the experiments can be found in a table in the supplemental material.

6 Results, Comparison and Discussion

This paper introduces a new picking technique that does not need any metadata, can be applied to volumetric scalar fields from all application domains and nevertheless picks the visible 3D location corresponding to a selected 2D position. To demonstrate these characteristics we applied the method to a selection of very different volumetric scalar fields. As the most sophisticated previous picking techniques come from the area of medical visualization, an abdominal MRI scan with intravenous contrast is our first example. Figure 1 shows how a position in a DVR image (DVRI) is picked, how a slice with the appropriate orientation is positioned so that it cuts the picked vessel, and how the slice can be subsequently used to examine the vessel in detail. The DVR is hidden in the final image to provide a completely free view of the slice. Figures 2 and 3 show DVRIs of the same dataset with different transfer functions. As adumbrated before, threshold-based picking fails for the DVRI in Figure 3 if the threshold is chosen too high and for the DVRI in Figure 2 if the threshold is chosen too low (e.g. zero, *first-hit*). In the first case the cast ray would go through the volume without identifying any position as picked. In the second case the ray tracing would stop as soon as it reaches the bounding box of the dataset because opacity can be found everywhere. WYSIWYP can handle the DVRs of all three transfer functions correctly. Please see the accompanying video for another comparison of the techniques using the fuel dataset which is described later in this section. The video also compares WYSIWYP to picking the position with the highest contribution to the final pixel. The comparison shows that the latter method (known from Bruckner et al. [7]) has problems in situations where WYSIWYP performs well.

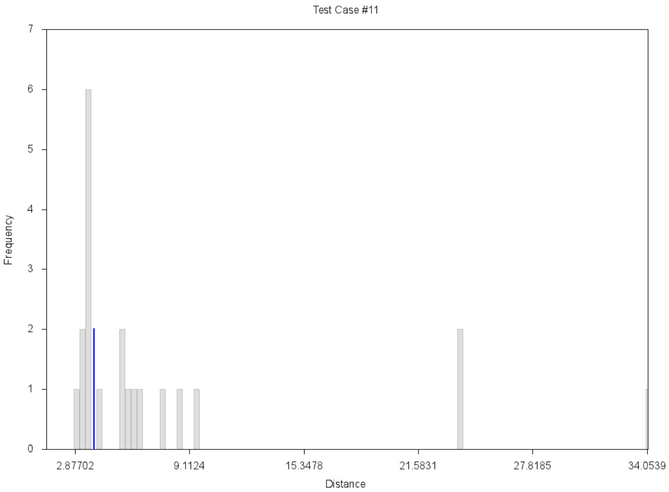

Fig. 9. Histogram of the distances from the positions selected by the participants to the average selected position in test case number 11. The blue bar shows the distance of the position chosen by WYSIWYP.

The second example dataset comes from a numerical simulation of flow around an ellipsoidal body. The images in Figure 11 show DVR of the vorticity $\|\nabla\times\mathbf{v}\|$ of the velocity vector field $\mathbf{v}$. The images have been rotated for illustration purposes so that the flow comes from below. Like for the MRI dataset, the steps of WYSIWYP are shown. Additionally, the curves of the accumulated opacity α^{acc} illustrate the interval selection. While approaches employing metadata, like *contextual picking* [15], are possibly applicable to the MRI dataset, they are not applicable for the flow field as there are no clearly defined structures that can be named and matched for detection. Vortices might be considered as such structures but there is still no vortex definition commonly agreed upon (see e.g. Lugt [17]).

A synthesized scalar function increasing from two locations provides data for the third example. The transfer function used for rendering produces two balls that are visible in the DVRIs of Figure 12. This synthetic example has been included because the shapes of the α^{acc}, β^{acc} and γ^{acc} curves are clearly discernable. The principle of choosing the highest jump is nicely visible in the lower right image of Figure 12. The first, last and central location of the selected interval are marked by gray bars. As our picking criterion suggests, the marks coincide with zero crossings of γ^{acc}. Although the two jumps corresponding to the first two peaks of β^{acc} are steeper, the criterion selects the marked interval because it exhibits the highest jump. The result is that a position in the shell of the ball in the background is picked through two transparent shell areas of the ball in front of it. This example also shows that material boundaries parallel to the viewer, which are usually well perceived, are easily picked because they are represented by a long and strong increase in opacity and thus a high jump of accumulated opacity. Finally, this rendering is another example where choosing the position with the highest contribution to the final pixel fails. It will select the front-most shell. The steepness of the first jump in Figure 12 confirms that single samples in this area have a high contribution to the final pixel.

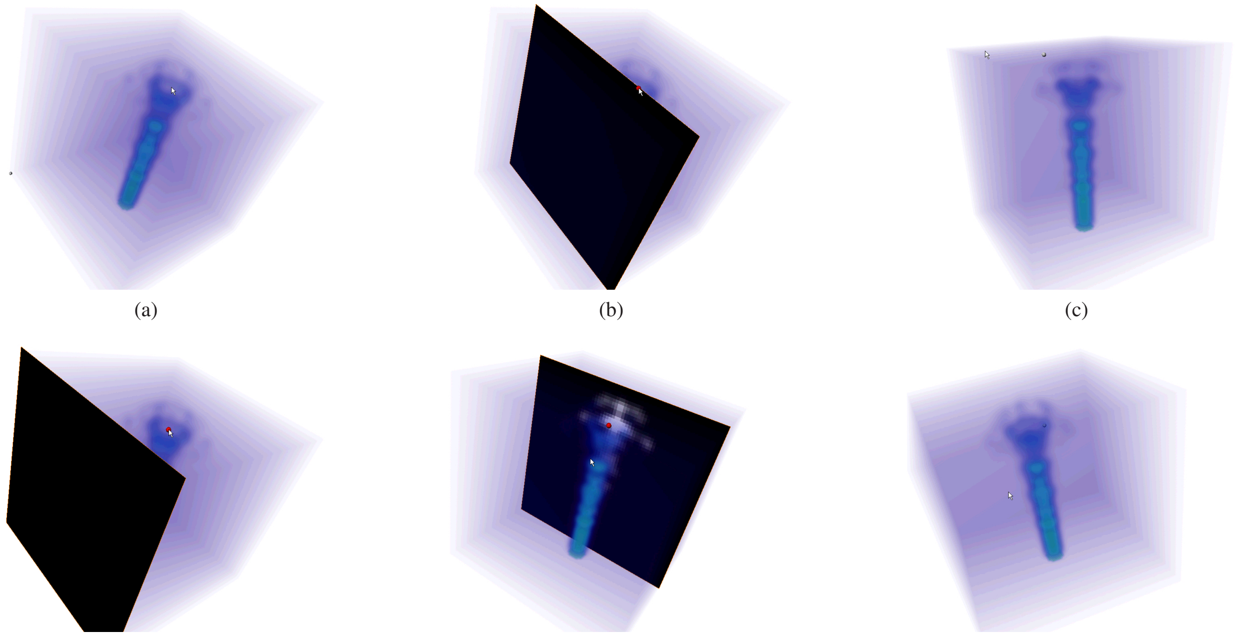

Fig. 10. Snapshots from the video accompanying this paper show a comparison of picking techniques in the fuel dataset. For a detailed discussion see Section 6.

The last dataset we present stems from a simulation of fuel injection into a combustion chamber. It contains density values of gas: the higher the density value, the less air. We use this dataset for comparing different picking techniques in Figure 10. The figure shows snapshots from the video accompanying this paper. Figure 10(a) shows the initial state of the rendering and the mouse pointer over the position that will be used for picking with the different techniques in the following. Figure 10(b) shows the situation directly after picking with the first-hit strategy. The slice is situated where nothing interesting is visible because the picked position is on the border of the dataset (see Figure 10(b) for another perspective). This is the case because there is opacity in a large area of the DVR. In Figure 10(d) the slice is on the front of the dataset because no position could be found by the threshold-based picking. The last two images (10(e), 10(f)) show the picking with WYSIWYP. The slice in Figure 10(e) cuts exactly through the intended position. Finally, Figure 10(f) shows the rendering from a different position in order to demonstrate that the small gray sphere indicating the picked position is located at the desired position. It becomes clear that only WYSIWYP can reliably yield the desired position in this dataset for which no metadata are available.

The presented technique is intrinsically interactive and three-dimensional, and thus hard to demonstrate in static 2D images. Therefore, a video with a live demonstration using some of the described and several additional datasets accompanies this paper.

6.1 Limitations

As may be deduced from the images throughout this paper, the proposed method deals with volume rendering using the standard emission-absorption model. This does not impose any constraint on the type of transfer function (e.g. one-dimensional vs. multi-dimensional). However, we did not investigate how the method deals with images in which local illumination has been applied after evaluating the transfer function. Perception theory [19] tells us that lighting, color and context influence the perception of transparency. Therefore we expect that the method will have to be extended to correctly handle volume rendering using local illumination. It is probable that complex computer vision methods are not required because more information than only the resulting image is available. The data and the transfer function are highly valuable information for the picking task.

Our current implementation does not ensure that close positions in screen space also result in close 3D locations. In noisy data sets this might be desirable. A solution one could imagine in conjunction with the handling of local illumination is the following: The 3D locations corresponding to positions lying next to (probably on pixel base) the picked position can be taken into account. In other words, one could cast additional rays for pixels around the picked position. An outlier filtering for the resulting 3D locations could then avoid rapid changes in the selected depth. Overall, this problem will only become relevant if we allow the user to *drag* the mouse while continuously updating the picked position and thus the slice. However, this is a quite unusual scenario for *picking*. Furthermore, it somehow contradicts the idea of picking what is visible. Nevertheless, we plan to incorporate the presented technique into a methods that allows to *trace* visible structures by dragging the mouse.

7 Conclusion and Future Work

We have presented a method to allow users to pick positions in volumetric renderings of three-dimensional data in a WYSIWYG type of interaction. Users can select, in an intuitive manner, the 3D position of structures that they really perceive in the rendering. In contrast to previous methods, the described approach is rendering-centered and is thus applicable for any type of volume rendered data. It only uses the transfer function of the volume rendering together with the data itself to determine the opacity and thus the visible structures along the viewing direction. Observable structures are characterized by large jumps in the accumulated opacity; the picked structure corresponds to the largest jump of the accumulated opacity. We emphasized the fact that no metadata is needed by demonstrating the method with data from a flow simulation where no metadata is available. The usefulness of the proposed technique for medical data has been shown by its application to an abdominal MRI scan, and the claims are supported by a user study. The application to flow and other data shows that the method is useful far beyond the medical domain.

As mentioned before, WYSIWYP has been developed for volume rendering without local illumination; research into picking in illuminated direct volume rendering is one of the next steps. Furthermore, we are already working on incorporating information from rays in the vicinity of the ray through the picked position.

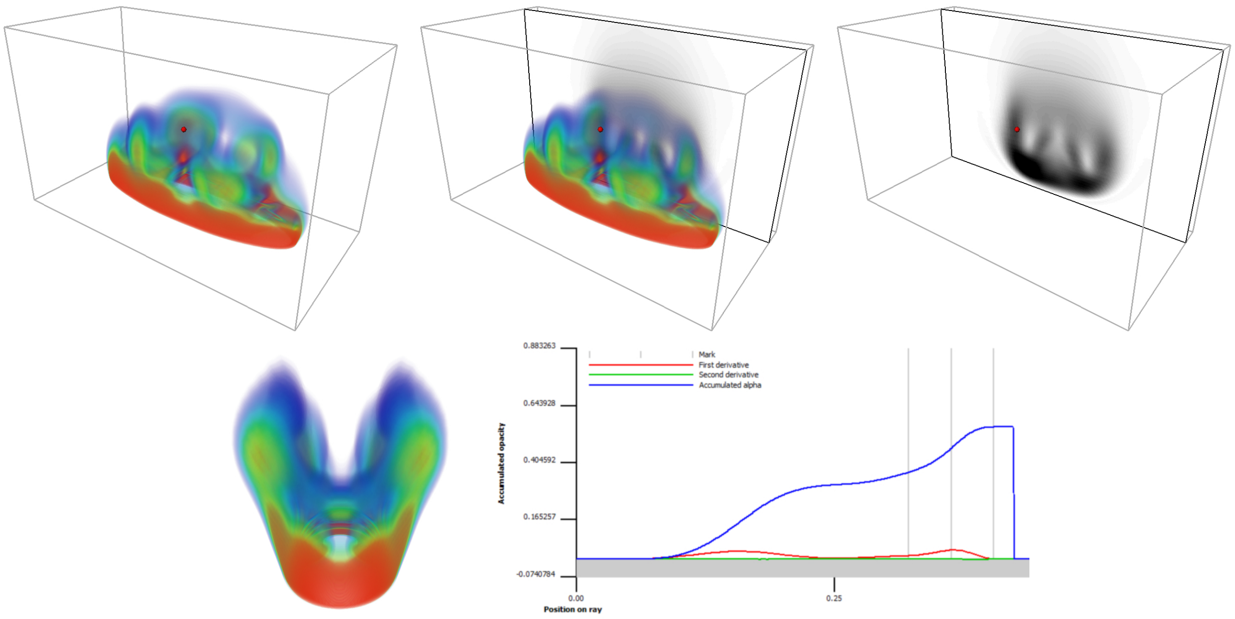

Fig. 11. Picking in DVR of vorticity field of flow around an ellipsoid. The images in the upper row show the picking process (picked position as red dot). The lower left image shows the volume rendered data from a different perspective. The lower right shows the accumulated opacity and its derivatives along the ray. Curves of derivatives are scaled by a factor of ten, but changes in second derivative are still hardly visible. The data is courtesy of Markus Rütten, DLR Göttingen.

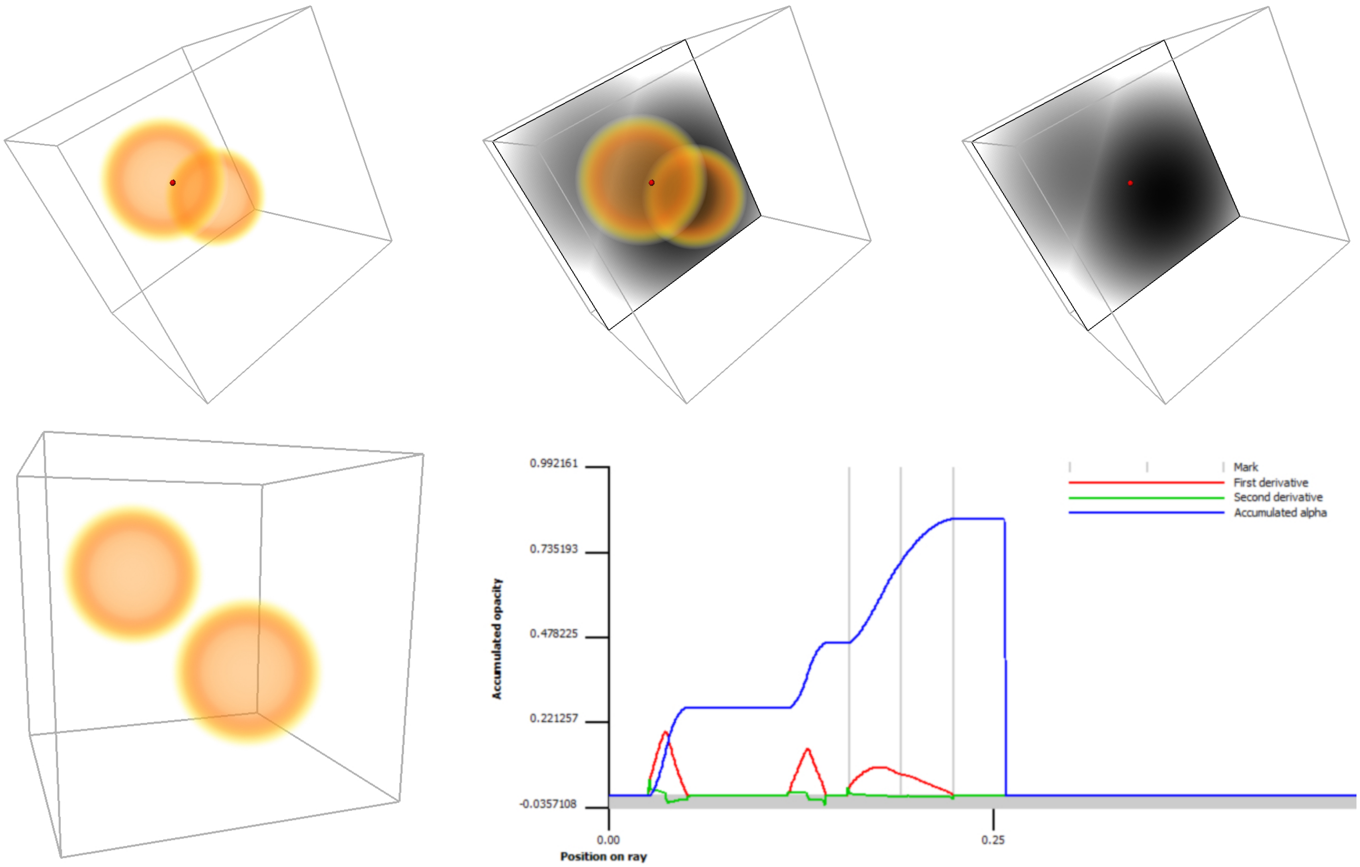

Fig. 12. The images in the upper row show the picking process (red dot=picked position) in a synthetic dataset consisting of two spheres. The picked 3D position can be inferred from the position and the gray-scale map of the slice. It shows that a position on the border of the sphere in the back is picked. This fits with what can be seen in the upper left image where the border of the sphere in the back shines through the second sphere in front of it. The lower left image shows the DVR from a different perspective to show the spatial relation of the spheres. The lower right shows the accumulated opacity and its derivatives along the ray. Curves of derivatives are scaled by a factor of ten. It is clear that the highest jump corresponding to the border of the sphere in the back is chosen from the intervals determined by the second derivative.

We believe that the proposed technique can help to pave the way for further application of DVR in application areas that are still reluctant to adopt this fundamental visualization technique.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) of the VIGOR++ Project under grant agreement nr. 270379. The information presented is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The opinions expressed in the document are of the authors only and in no way reflect the European Commission's opinions. All visualizations were produced with Amira [33]. The authors are grateful to the reviewers for their valuable comments that helped to improve the paper. Many thanks go to Jesica Makanyanga for proofreading.

References

[1] M. Agus, E. Gobbetti, J. Iglesias Guitián, F. Marton, and G. Pintore. GPU accelerated direct volume rendering on an interactive light field display. *Computer Graphics Forum (EUROGRAPHICS 2008)*, 27(3):231–240, 2008.

[2] F. Argelaguet and C. Andujar. Efficient 3d pointing selection in cluttered virtual environments. *IEEE Computer Graphics & Applications*, 29(6):34–43, Nov. 2009.

[3] Avizo - the 3D analysis software for scientific and industrial data. http://www.vsg3d.com/avizo/overview, last visited 2011-09-21.

[4] W. D. Bidgood and S. C. Horii. Introduction to the ACR-NEMA DICOM standard. *Radiographics a review publication of the Radiological Society of North America Inc*, 12(2):345–355, 1992.

[5] U. D. Bordoloi and H.-W. Shen. View selection for volume rendering. In *IEEE Visualization 2005*, pages 487–494. IEEE Computer Society, 2005.

[6] S. Bruckner, P. Kohlmann, A. Kanitsar, and M. E. Gröller. Integrating volume visualization techniques into medical applications. In *Proceedings of 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 820–823, May 2008.

[7] S. Bruckner, V. Šoltészová, E. Gröller, J. Hladůvka, K. Bühler, J. Yu, and B. Dickson. Braingazer - visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1497–1504, Nov. 2009.

[8] C. D. Correa and K.-L. Ma. Visibility-driven transfer functions. In *Proceedings of the 2009 IEEE Pacific Visualization Symposium*, PACIFICVIS '09, pages 177–184, Washington, DC, USA, 2009. IEEE Computer Society.

[9] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *SIGGRAPH Comput. Graph.*, 22(4):65–74, June 1988.

[10] K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. AK Peters, 2006.

[11] E. Gobbetti, P. Pili, A. Zorcolo, and M. Tuveri. Interactive virtual angioscopy. In *Proc. of the conference on Visualization '98*, VIS '98, pages 435–438, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[12] C. G. Healey and J. T. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2011.

[13] P. Kohlmann, S. Bruckner, A. Kanitsar, and M. E. Gröller. LiveSync: Deformed viewing spheres for knowledge-based navigation. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1544–1551, Oct. 2007.

[14] P. Kohlmann, S. Bruckner, A. Kanitsar, and M. E. Gröller. LiveSync++: Enhancements of an interaction metaphor. In *Proceedings of Graphics Interface 2008*, pages 81–88. Canadian Information Processing Society, May 2008.

[15] P. Kohlmann, S. Bruckner, A. Kanitsar, and M. E. Gröller. Contextual picking of volumetric structures. In H.-W. S. Peter Eades, Thomas Ertl, editor, *Proceedings of the IEEE Pacific Visualization Symposium 2009*, pages 185–192. IEEE Computer Society, May 2009.

[16] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 287–292, Washington, DC, USA, 2003. IEEE Computer Society.

[17] H. J. Lugt. *Introduction to Vortex Theory*. Vortex Flow Press, Inc., Potomac, Maryland, 1996.

[18] M. M. Malik, T. Möller, and E. Gröller. Feature peeling. In *Proceedings of Graphics Interface 2007*, pages 273–280. A K Peters Ltd, May 2007.

[19] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[20] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[21] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1:99–108, 1995.

[22] J. Meyer-Spradow, T. Ropinski, J. Mensmann, and K. H. Hinrichs. Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, 29(6):6–13, Nov./Dec. 2009.

[23] Y. Mori, S. Takahashi, T. Igarashi, Y. Takeshima, and I. Fujishiro. Automatic cross-sectioning based on topological volume skeletonization. In A. Butz, B. Fisher, A. Krüger, and P. Olivier, editors, *Smart Graphics*, volume 3638 of *Lecture Notes in Computer Science*, pages 924–924. Springer Berlin / Heidelberg, 2005.

[24] MeVisLab - development environment for medical image processing and visualization. http://www.mevislab.de, last visited 2011-09-21.

[25] VTK/New CellPicker - for vtkVolume objects. http://www.vtk.org/Wiki/VTK/New_CellPicker, last visited 2011-09-21.

[26] S. Owada, F. Nielsen, and T. Igarashi. Volume catcher. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, I3D '05, pages 111–116, New York, NY, USA, 2005. ACM.

[27] H. Peng, Z. Ruan, F. Long, J. H. Myers, and E. W. Myers. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat Biotech*, 28(4):348–353, 2010.

[28] B. Preim and D. Bartz. *Visualization in Medicine*. Morgan Kaufman, 2007.

[29] C. Rezk-Salama and A. Kolb. Opacity peeling for direct volume rendering. *Computer Graphics Forum*, 25(3):597–606, 2006.

[30] P. Sabella. A rendering algorithm for visualizing 3D scalar fields. *SIGGRAPH Comput. Graph.*, 22:51–58, June 1988.

[31] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit*. Prentice Hall, 2nd edition, 1998.

[32] A. Squillacote. *The ParaView Guide*. Kitware, Inc., 3rd edition, 2008.

[33] D. Stalling, M. Westerhoff, and H.-C. Hege. Amira: A highly interactive system for visual data analysis. In C. D. Hansen and C. R. Johnson, editors, *The Visualization Handbook*, pages 749–767. Elsevier, 2005.

[34] K. D. Toennies and C. Derz. Volume rendering for interactive 3-d segmentation. In *Proceedings of the SPIE (Medical Imaging 1997)*, volume 3031, pages 602–609, 1997.

[35] M. Tory and T. Möller. Human factors in visualization research. *IEEE Transctions on Visualization and Computer Graphics*, 10(1):72–84, January 2004.

[36] J. van Scheltinga, M. Bosma, J. Smit, and S. Lobregt. Image quality improvements in volume rendering. In *Visualization in Biomedical Computing*, pages 87–92. Springer, 1996.

[37] J. Vlietinck. Picking on fused 3D volume rendered images and updating corresponding views according to a picking action, 2008. Patent No EP1923840A2.

[38] Y. Wang, J. Zhang, W. Chen, H. Zhang, and X. Chi. Efficient opacity specification based on feature visibilities in direct volume rendering. *Computer Graphics Forum (Pacific Graphics 2011)*, 30(7):2117–2126, 2011.

[39] A. Wiebel, F. M. Vos, and H.-C. Hege. Perception-oriented picking of structures in direct volumetric renderings. Technical Report 11-45, ZIB, Takustr. 7, 14195 Berlin, 2011.

[40] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Scientific Visualization / Information Visualization 2012)*, 18(12), Dec. 2012. In this issue.