Direct Volume Interaction for Visual Data Analysis

Part 2: Direct Manipulation

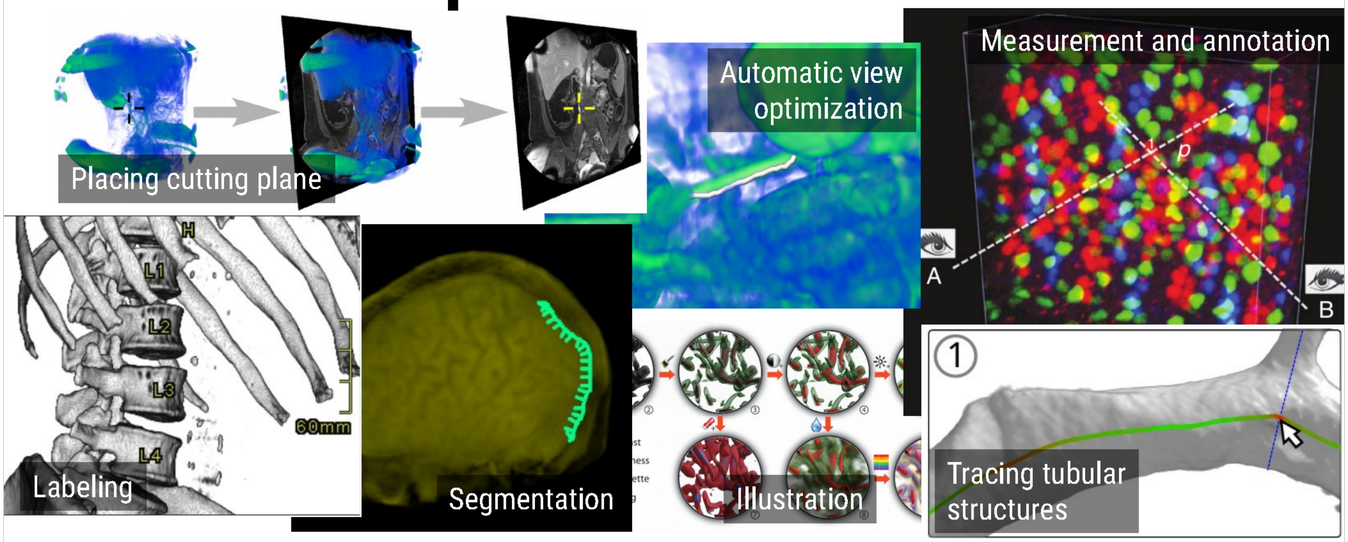
Alexander Wiebel, *Worms University of Applied Sciences*

IEEE VIS Tutorial 2015

Hochschule Worms
University of Applied Sciences

For some of the slides additional information is presented in the notes below the slides. This can be of interest where the idea of the slide is not obvious due to the image-oriented design of the slides, or for additional information on the presented techniques.

Examples of Analysis Tasks Requiring Direct Manipulation at 3D Locations



The examples given here will appear as labels to specific techniques presented later. The labels are provided in order to give examples of which tasks are supported by each for the low-level techniques.

Overview

Direct manipulation is enabled by

“Translating given 2D input into application-specific 3D locations, actions and manipulations regarding the volume“

- Input types (2D → 3D)
  - Points → picking
  - Lines → sketching
  - Regions → marking (often regions are defined by closed lines)
- Semantics (3D actions)
  - Selecting points, lines and surfaces in 3D
  - Segmentation, labeling, 3D painting
  - Manipulating transfer function or underlying data

Abstract:

Interaction with and manipulation of volumetric renderings and volumetric data are naturally non-trivial tasks when carried out on common 2D displays with mouse and keyboard input. This part will concentrate on techniques which take the provided 2D input and translate it into application-specific direct 3D actions and manipulations regarding the volume. Input types we will discuss include points (picking), lines (sketching) or regions (marking). Depending on the user task at hand, semantics of the resulting 3D actions range from simple selection of positions, lines and surfaces in 3D, to segmentation, labeling, 3D painting, and adaption of the transfer function or the data themselves. We will give a detailed description of the possibly needed additional or meta data for each technique. An overview of common applications of the mentioned techniques for imaging and simulation data from medicine and natural sciences will demonstrate the utility of the techniques.

Overview

Direct manipulation is enabled by

"Translating given 2D input into application-specific 3D locations, actions and manipulations regarding the volume"

- **Input types (2D → 3D)**
  - Points → picking
  - Lines → sketching
  - Regions → marking (often regions are defined by closed lines)
- Semantics (3D actions)
  - Selecting points, lines and surfaces in 3D
  - Segmentation, labeling, 3D painting
  - Manipulating transfer function or underlying data

Abstract:

Interaction with and manipulation of volumetric renderings and volumetric data are naturally non-trivial tasks when carried out on common 2D displays with mouse and keyboard input. This part will concentrate on techniques which take the provided 2D input and translate it into application-specific direct 3D actions and manipulations regarding the volume. Input types we will discuss include points (picking), lines (sketching) or regions (marking). Depending on the user task at hand, semantics of the resulting 3D actions range from simple selection of positions, lines and surfaces in 3D, to segmentation, labeling, 3D painting, and adaption of the transfer function or the data themselves. We will give a detailed description of the possibly needed additional or meta data for each technique. An overview of common applications of the mentioned techniques for imaging and simulation data from medicine and natural sciences will demonstrate the utility of the techniques.

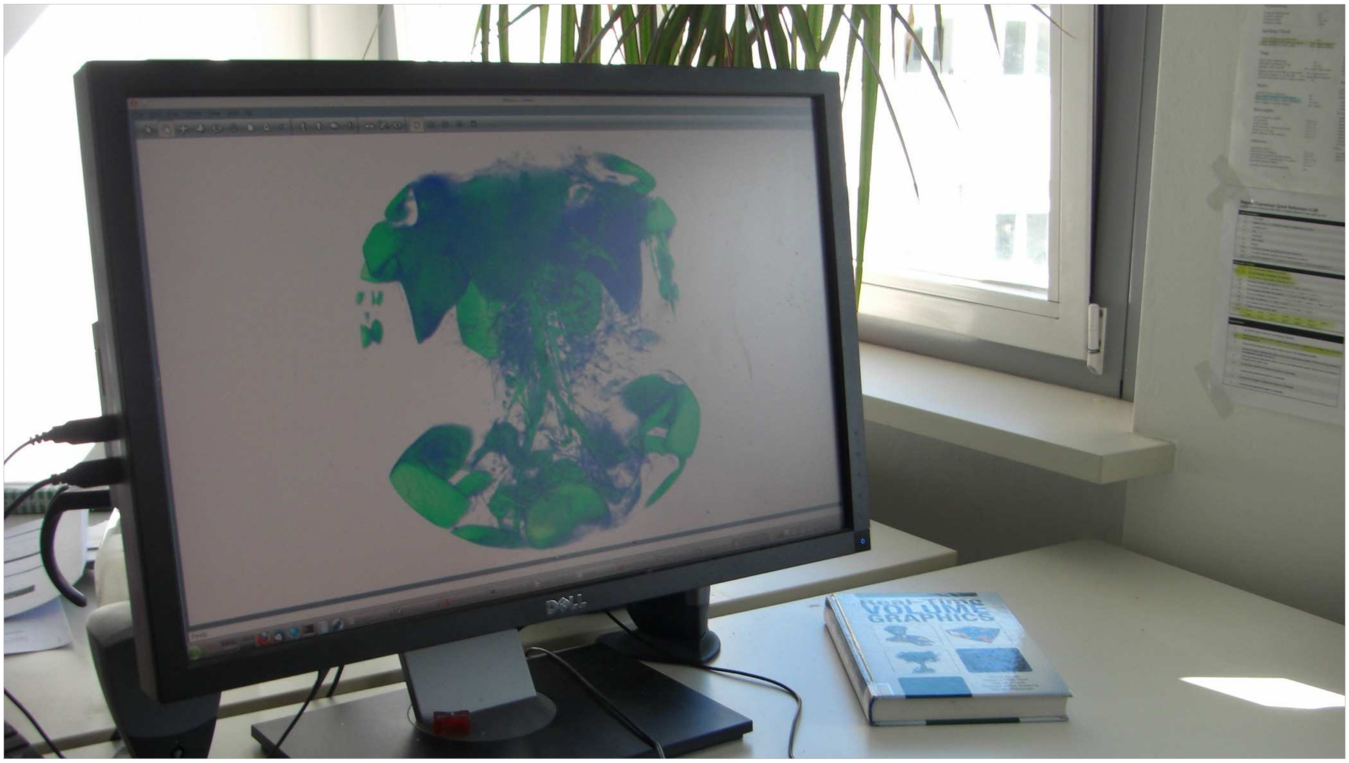
Picking Points: Definition

- In general
  - Selection of a position by pointing at it

No additional comments.

When a person is pointing at some object a human observer will usually look in the direction of the person's arm. One can imagine a straight line (ray, arrow) extending from the arm. The observer will usually interpret the object belonging to the first surface hit by the straight line as being the object the person intended to point at. In the above example this object is the tree. Depending on what the person says while pointing, the result can also be a part of the tree, e.g. a branch, or even a point on the branch. This point is exactly located where the ray intersects with the branch.

In the following we will explain how the natural process of pointing at objects can be translated to actions using a mouse on a 2D computer screen.



In the following we will explain how the natural process of pointing at objects can be translated to actions using a mouse on a 2D computer screen.

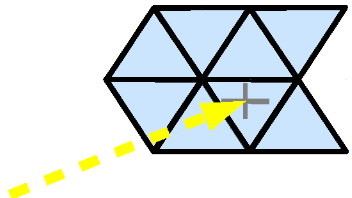
Picking Points: Definition

- In general
  - Selection of a position by pointing at it
- Related to a computer-generated image
  - In a virtual environment
    - Selection of a 3D position using a 3D input device
  - **In front of a computer monitor**
    - Selection of a 3D position by **clicking a 2D position** on the screen
  - **In front of a mobile device**
    - Selection of a 3D position by **touching a 2D position** on the screen

No additional comments.

Picking Points

- Easy for surfaces:



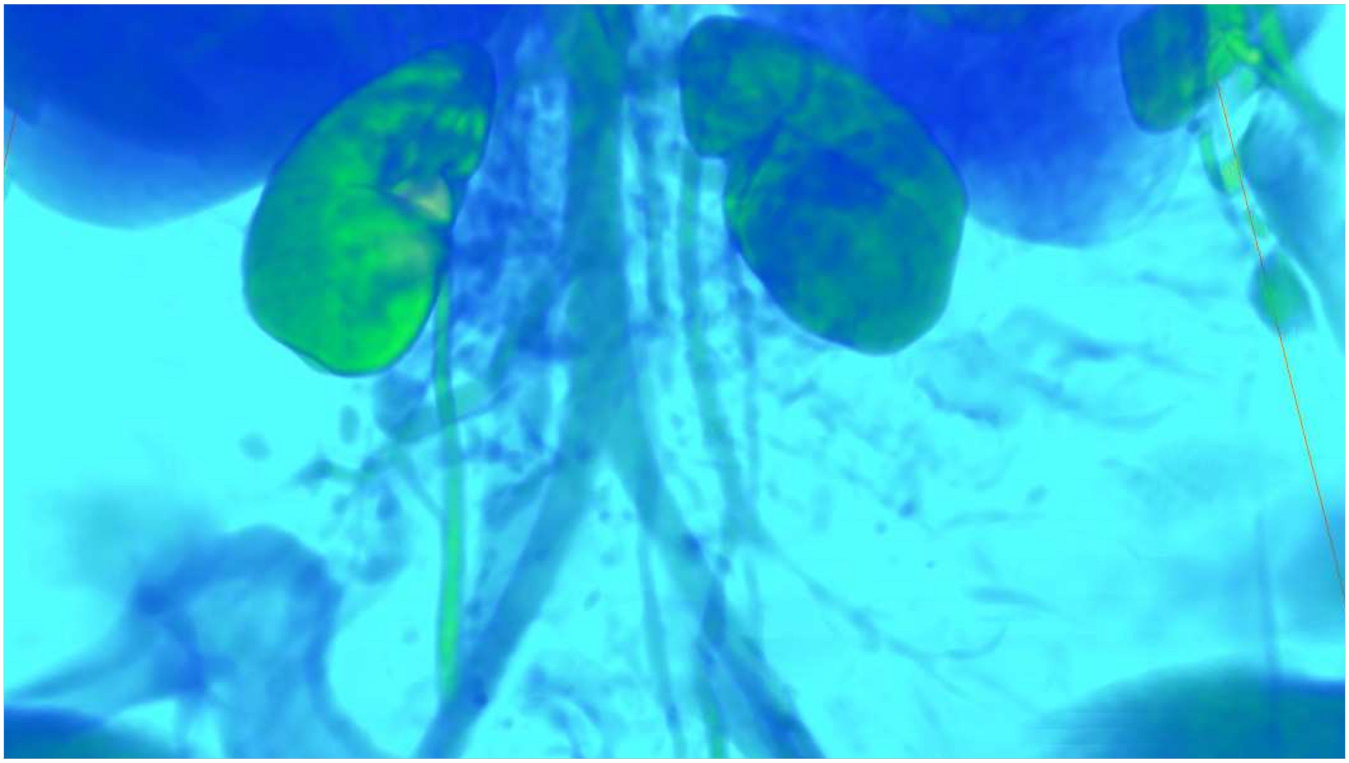
A **single** ray-triangle intersection

“All earthly environments consist of objects with well-defined surfaces, ...“
“Our visual systems are built to perceive the shapes of 3D surfaces.“

[Colin Ware, Information Visualization, 2012. With references to Gibson 1979]

Picking points on (opaque) surfaces represented as triangle meshes is a straight forward task. We cast a ray along the current viewing direction through the position of the mouse on the 2D screen. Then we check for every triangle if the ray intersects the triangle. The intersection point closest to the viewer (the only one visible when dealing with opaque surface) is chosen as result.

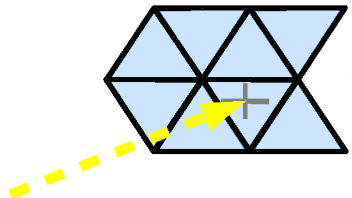
A ray-triangle intersection is easy to compute but obviously testing all triangles can be very costly. In an efficient implementation, acceleration data structures like, e.g., spatial trees can be used.



If the rendering on the screen does not consist only of surfaces but a direct volume rendering is used, picking points becomes more complicated. First, there are no triangles to compute the intersection with. Second, most direct volume renderings use transparency, which usually results in multiple objects being visible at a single 2D screen location. Some objects can be seen „through“ others.

Picking Points

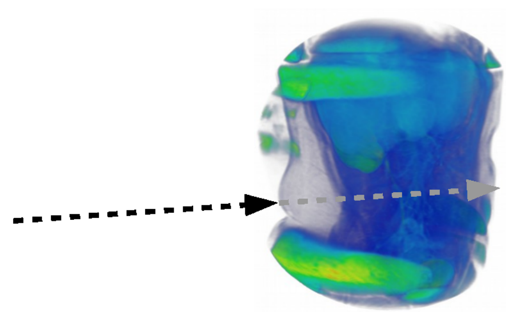
- Easy for surfaces:



A **single** ray-triangle intersection

- How for direct volume rendering techniques?

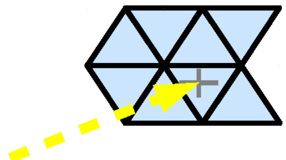
  Problem: **No single** intersection point with volume along pointing ray!



No additional comments.

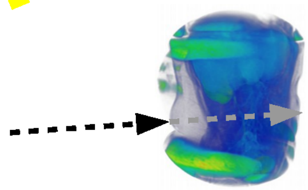
Picking Points

- Easy for surfaces:

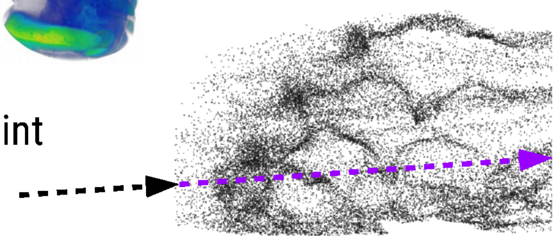


A **single** ray-triangle intersection

- How for direct volume rendering techniques?

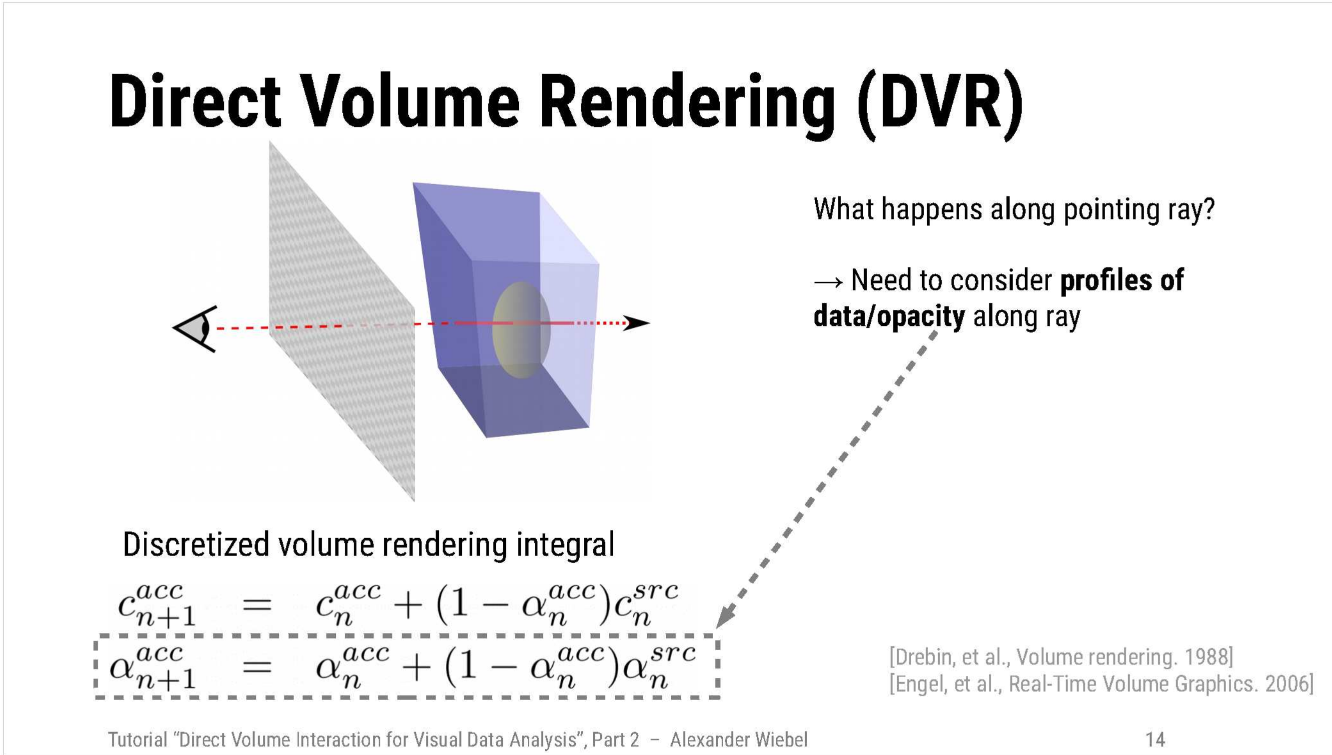


- Or for point clouds?



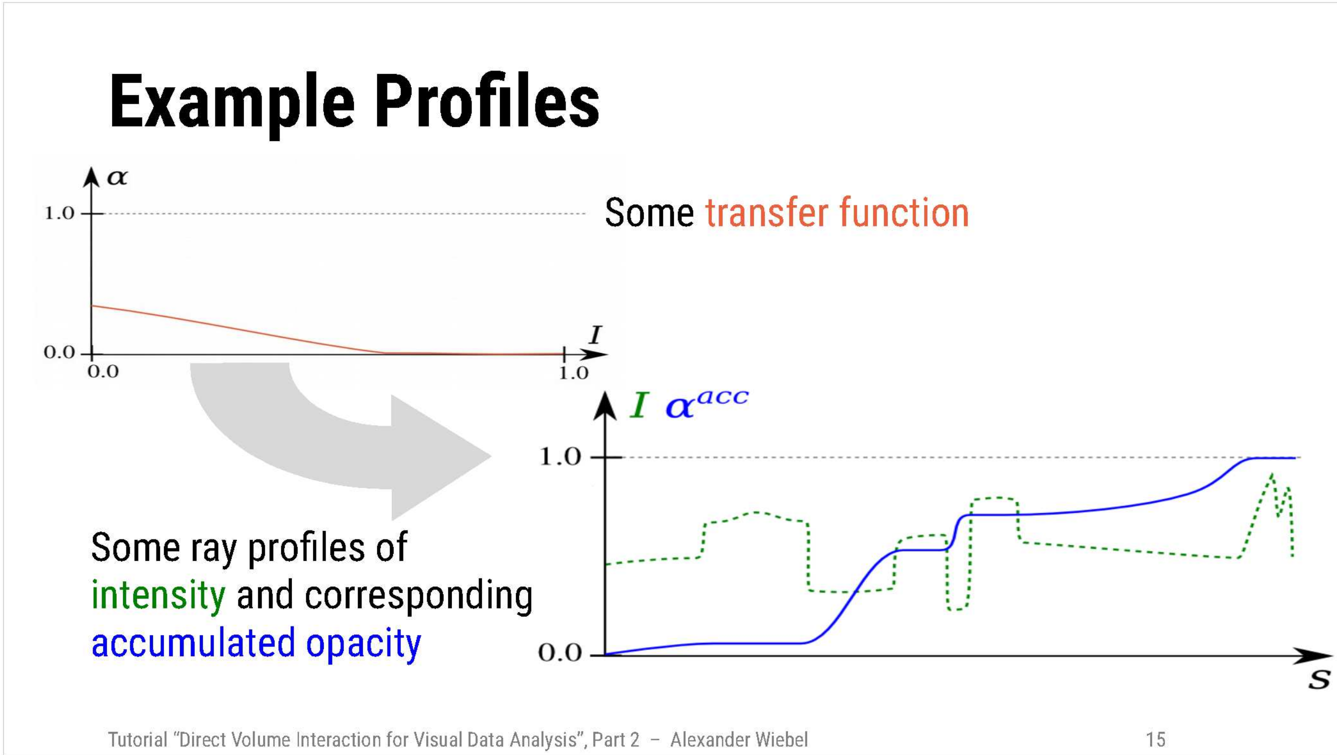
Problem: **No single** intersection point with volume along pointing ray!

No additional comments.



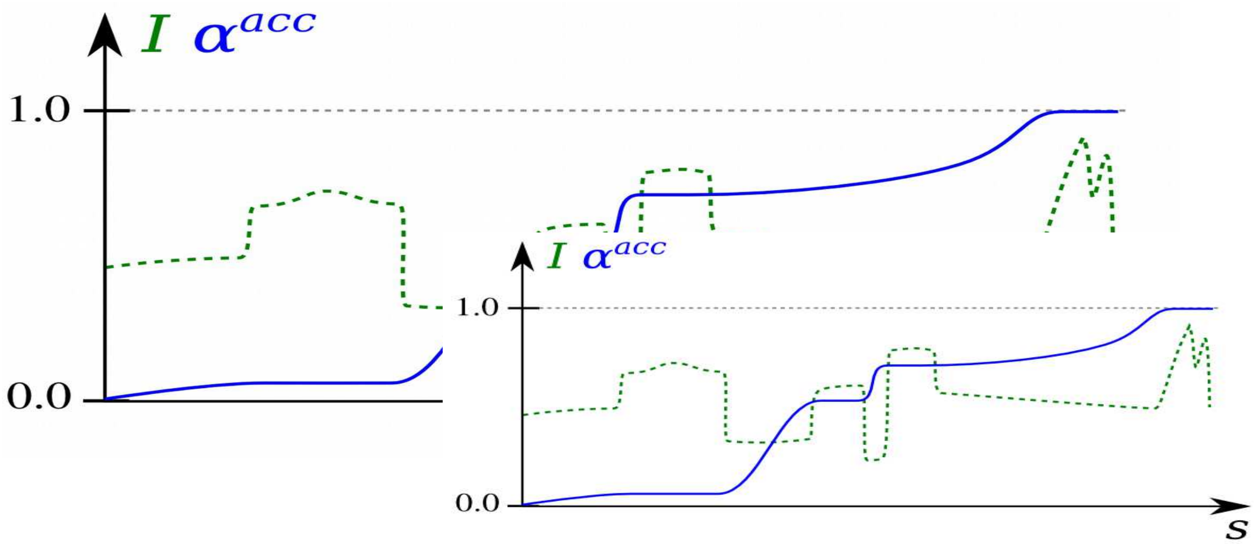
As mentioned before, one of the challenges of picking in direct volume rendered images lies in the use of transparency and thus its converse quantity, the opacity. The second equation in the slide describes how opacity is accumulated along the ray for many common volume rendering approaches. It uses opacity values for discrete locations (sample points) along the ray.

The different opacity values along the ray are used by many picking techniques to determine the picked 3D position in the volume rendered images.



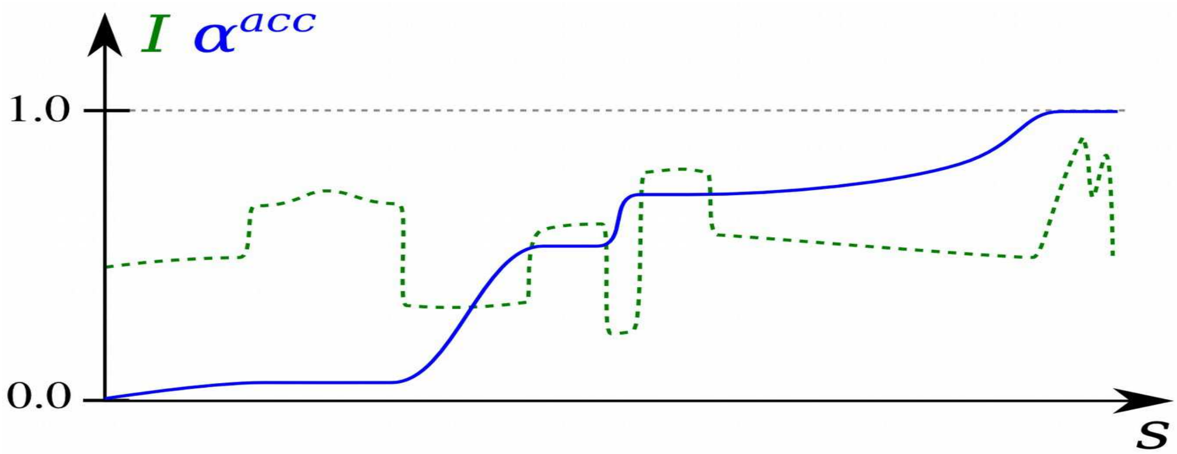
The curve of the values (e.g. intensity, accumulated opacity) changing along the ray is often called „ray profile". Using the previously mentioned equation for the opacity accumulation together with a certain transfer function which maps intensity to opacity, the ray profile for the accumulated opacity can be computed from the ray profile for the intensity. Here the „intensities" are simply the values in the original dataset at the locations for the sample points.

Example Profiles



The shown ray profiles for accumulated opacity and intensity will be used as an example in the following slides.

Example Profiles



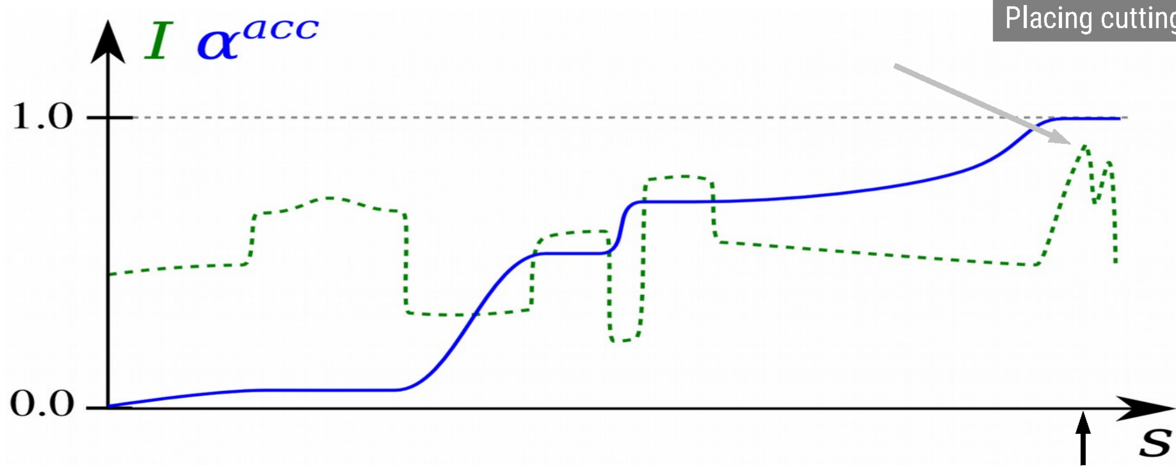
The shown ray profiles for accumulated opacity and intensity will be used as an example in the following slides.

Picking: Maximum Intensity

Measurement and annotation

Labeling

Placing cutting plane



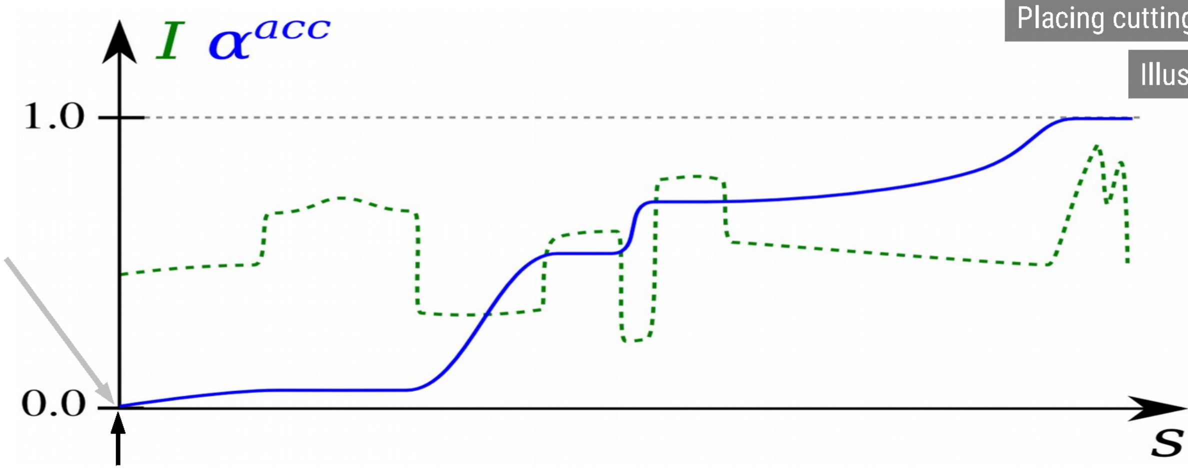
Limitation: Does not consider visibility. Invisible location might be picked.

[Wallis, et al., 1989]

One of the simplest types of volume rendering is the so called maximum intensity projection (MIP). An MIP image is the result of mapping the maximum value (maximum intensity) along the viewing ray to a color (usually gray scale) for every pixel of the image. For MIP images picking is as simple as choosing the position with the maximum intensity along the ray through the picked position. Although the picked position is perfect for MIP, in a general direct volume rendering setting it might be invisible due to occluding parts or simply because the transfer function maps its opacity to a value close to zero. Briefly summarized, picking maximum intensity positions considers only the underlying data while ignoring the visibility of the position in the rendering.

Measurement and annotation
Labeling
Placing cutting plane
Illustration

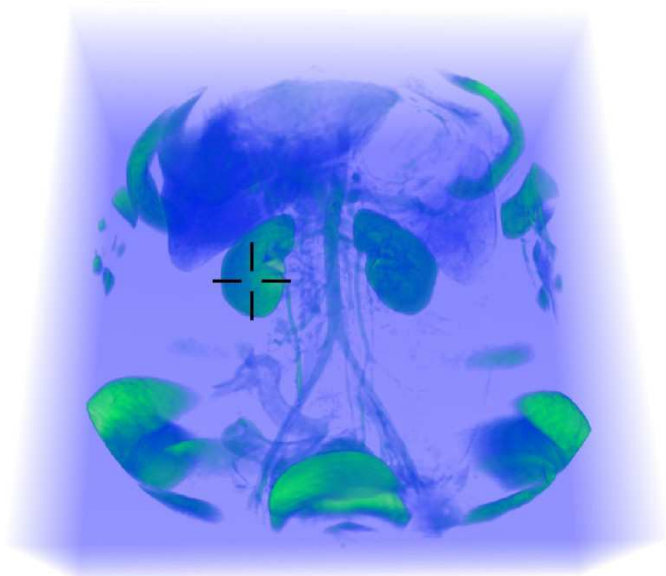
Picking: First Hit (Opacity)



Limitation: „foggy“ rendering → picked position will always lie on dataset boundary (see also next slide)

[Mayer-Spradow, et al.,Voreen, 2009]
[Doux, et al., Amira, 2015]

The so-called „first-hit“ approach, considers visibility. Using this approach the first position along the ray having non-zero opacity is chosen. This will always produce locations that are visible because no other region can occlude the position and the position is not completely transparent However, although the opacity is non-zero it might be very low. In this case the position might be „barely“ visible because other samples along the ray have a much bigger influence on the final color at the clicked 2D pixel position.



Problematic case for first hit approach.

First hit approach works only if space in front of features is completely transparent.

Limitation: „foggy“ rendering → picked position will always lie on dataset boundary

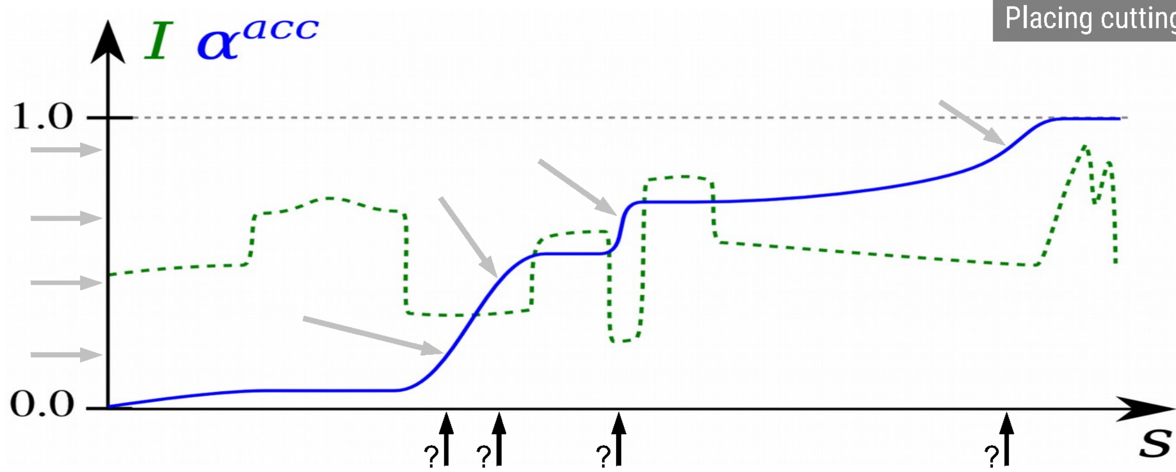
The usefulness of the „first hit“ approach in cases of transfer functions yielding „foggy“ renderings (see slide) is also limited. In these cases the picked position will always lie on the boundary of the dataset instead of on top of a relevant feature in the rendering. This results from the fact that nearly all positions in such a rendering have non-zero opacity. A solution to this can be to use a threshold chosen by the user instead of the fixed zero threshold. This will be discussed on the next slide.

Measurement and annotation

Labeling

Placing cutting plane

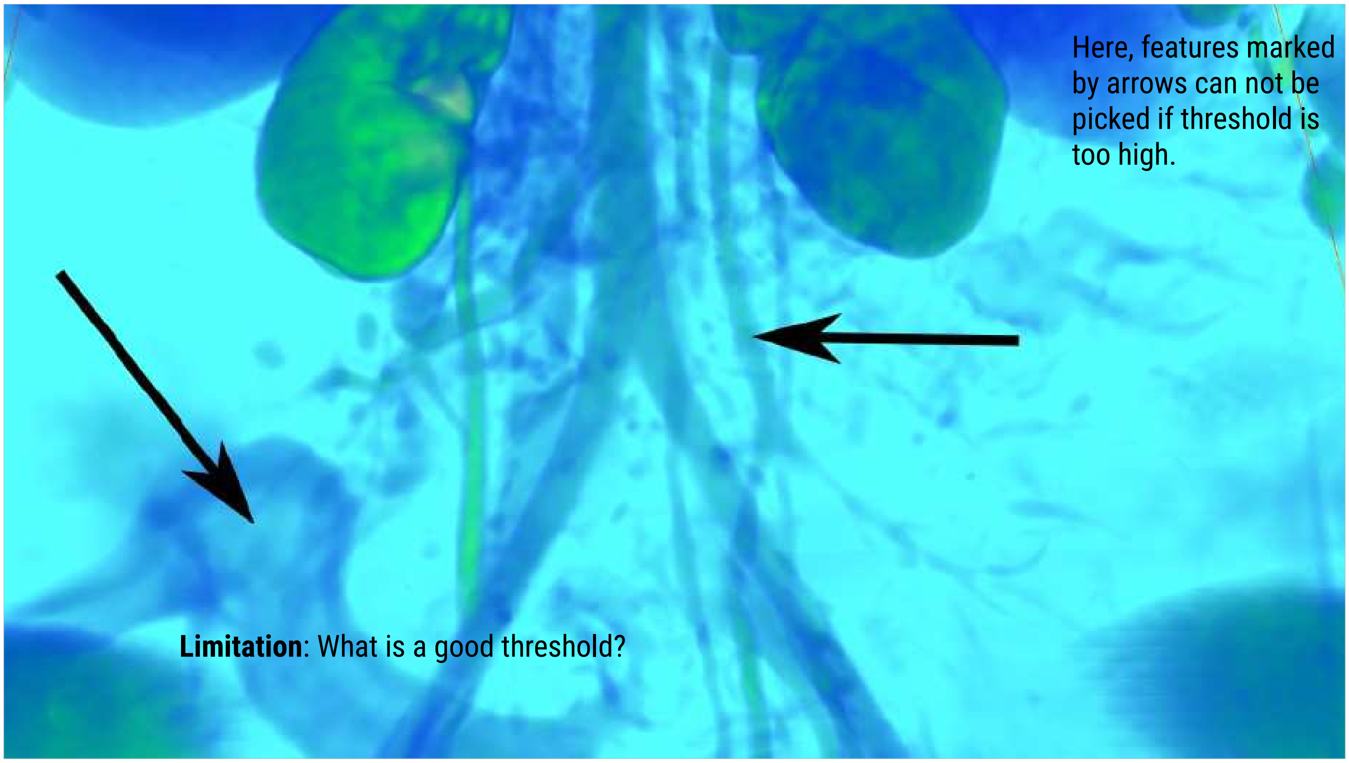
Picking: Opacity Threshold



Limitation: What is a good threshold?

[Parker, et al., Interactive ray tracing for isosurface rendering. 1998]
[Gobetti et al., Interactive Virtual Angioscopy. 1998]

One of the first known variants of picking 3D positions in direct volume renderings uses opacity threshold values to determine the position of interest along the ray. The resulting location is determined by the first sample that exceeds the threshold. This approach is more flexible than the "first hit" approach. The threshold is usually provided by the users. Thus it can be adapted to the data and the transfer function at hand. The possibly problematic part of this methods is the choice of the threshold. Different threshold yield different results and, as will be shown on the next slide, a single threshold might not be sufficient even for a single image.



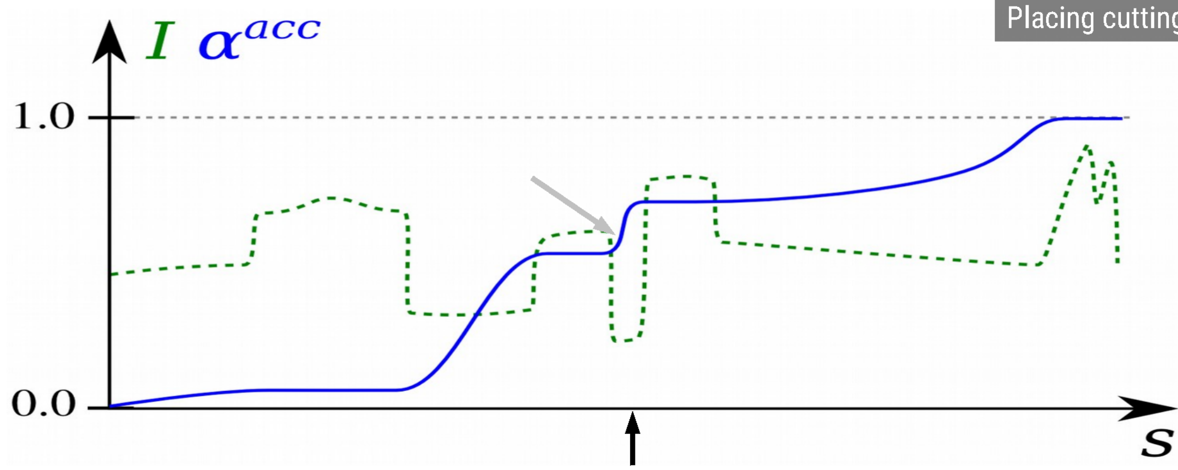
If the opacity threshold is chosen to be zero, the problematic cases are the same as for the “first hit” approach. If the opacity threshold is chosen higher, depending on the exact threshold, some very transparent features can not be picked because their opacity does not reach the threshold. In many cases undesired occluding regions have the same opacity as some of the features of interest. In such cases a single threshold can not be the appropriate picking criterion.

Measurement and annotation

Labeling

Placing cutting plane

Picking: Most Contributing Sample



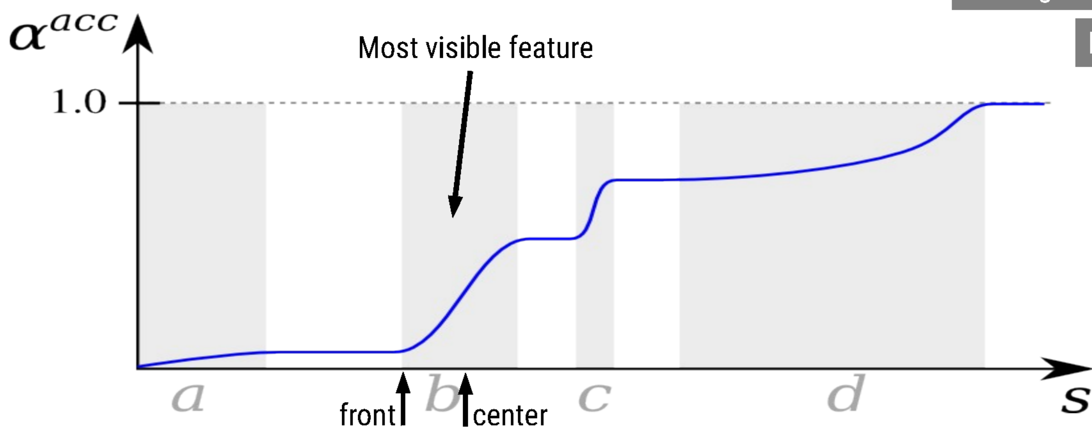
Limitation: Most contributing sample might not belong to most visible feature.

[Bruckner et al. 2009]

A way to become independent of a single opacity threshold is to determine the most contributing sample along the current ray. This is similar to choosing the sample with maximum intensity but considers the visibility of the position. The most contributing sample is the one where the accumulated opacity exhibits the strongest increase, i.e. where the profile has the steepest slope. This is the single position that is most visible along the current ray. However, it might or might not belong to the feature (a group of consecutive samples) that is most visible. Thus it might not be the position of interest for picking. Furthermore, even if the most contributing sample is part of the most contributing and thus most visible feature, it still can appear at any arbitrary position in the feature. The user trying to pick the feature, however, is often interested in picking the front (first sample) or center position of the feature. The most contributing sample is not guaranteed to be one of these two.

Measurement and annotation
Labeling
Placing cutting plane
Illustration

Picking: Highest Opacity Jump



Limitation: Choice in case of similarly visible features.

[Wiebel et al., WYSIWYP, 2012]

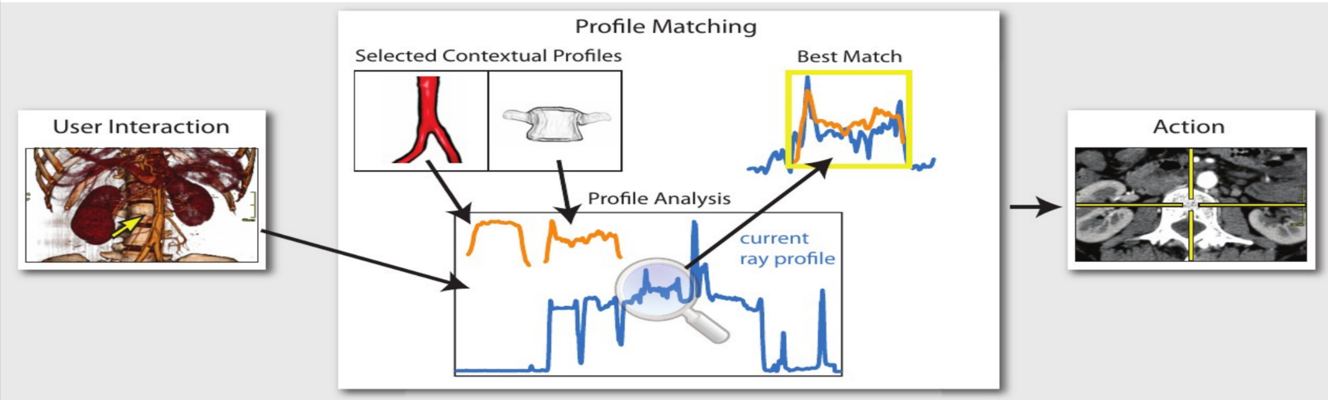
One possibility to determine the most visible feature along the picking ray is to partition the opacity profile into intervals of similarly continuously increasing accumulated opacity and intervals where no or only few opacity is accumulated. The intervals with accumulating opacity represent the visible features. The interval with the highest increase of accumulated opacity (not necessarily the steepest increase) is the one contributing most to the final pixel color and thus roughly corresponds to the most visible feature. Problematic cases for this approach are features with very similar visibility. As the approach determines intervals, i.e. the whole extent of the feature, it is possible to provide the user with a position at the front (left end of interval) or the center of the feature (center of the interval).

Measurement and annotation

Labeling

Placing cutting plane

Picking: Contextual Picking



[Kohlmann, et al., Contextual picking of volumetric structures. 2009]

Limitation: Needs additional meta data.

- Data profiles for different anatomical structures
- Type of current examination

All previously presented methods are focused on the visibility or the data of the samples along the ray. They completely ignore the type of analysis or protocol being currently conducted by the user. This makes them independent of the type of dataset being inspected but ignores possibly helpful meta information. The "contextual picking" approach stores typical intensity profiles for different anatomical structures and selects the appropriate ones for the current user task. Information on the users current task is obtained from meta data stored with the original volume data. For medical applications an example of formats storing such meta data is the widely used DICOM standard.

Picking the desired position with this approach is achieved by fitting the selected profiles to the intensity profile of the current ray. The interval along the ray that best matches one of the selected profiles is chosen as feature of interest. As this method determines an interval of interest, the picking result can be a position at the front or in the center of the feature.

Different Picking for Diff. 3D Rendering

Table compiled according to descriptions in [Wiebel, et al., Perception-Oriented Picking of Structures in Direct Volume Rendering. ZIB-Report 11-45, 2011.]	First Hit	Max./min. Intensity	Threshold (intensity or opacity)	Most Contributing Sample	Perceptual Picking (WYSIWYP)	Contextual Picking
(Iso-)surface	X					
Iso ray casting	X					
MIP (max. intensity projection)	X	X				
mIP (min. intensity projection)	X	X				
AIP (avg. intensity projection)						
CVP (closest vessel projection)			X			
DVR (direct volume rendering)	X	X	X	X	X	X
Slices	X					
Transparent surfaces	X			X	X	

The table shows that most picking approaches are especially appropriate for certain types of representations of volumetric data.

There are two exceptions:

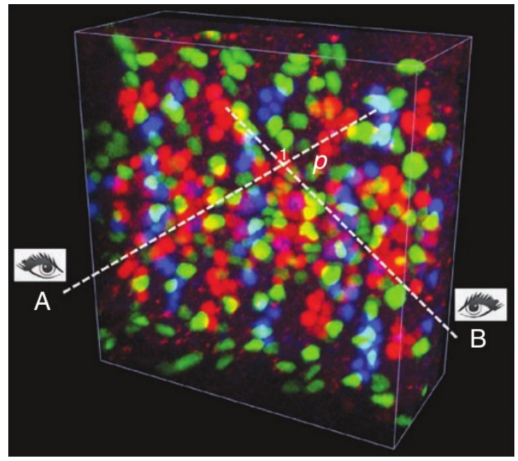
1) All described approaches can be applied to direct volume rendering images to some extent.

2) The "first hit" approach (within its limitations) makes some sense for most rendering types.

Picking: With Two Clicks

- Looking at feature from two view points
- Clicking feature for both view points an casting rays
- Pick position at approximate intersection of the rays

Limitation: Feature must be visible from multiple view points.



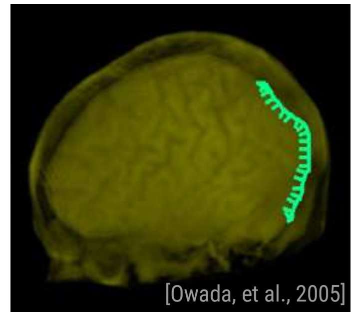
[Peng, et al., 2010]

Whereas all techniques mentioned until now use only one click to select the 3D position of interest, a method presented by Peng et al. uses two clicks. The method is independent of the data and only needs the final rendering together with two clicks to determine the 3D position. No meta data is used. The 3D position is determined by intersecting the two rays cast from the two clicked 2D positions. The method only works if the user clicks at the **same** feature from the two viewing angles, otherwise the rays will not intersect, sometimes even not approximately intersect. This leads to one of the limitations of this method. In order to be able to click at the same feature from the two different viewing angles, the feature of interest needs to be visible from both viewing angles.

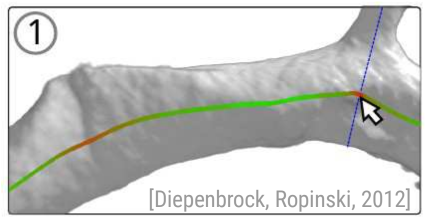
Sketching Lines and Marking Regions

Can be considered as **extensions to picking** points because

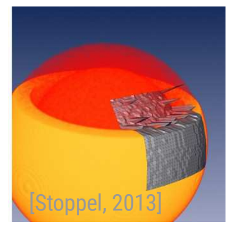
- Lines consist of 1D array of points
- Regions can be stored as multi-dimensional arrays of points
- Interaction also in 2D
- Result also in 3D



[Owada, et al., 2005]



[Diepenbrock, Ropinski, 2012]



[Stoppel, 2013]

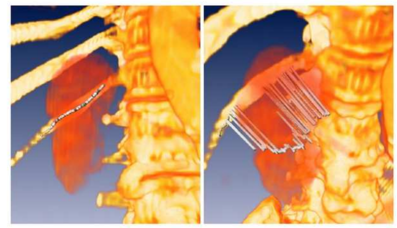
No additional comments.

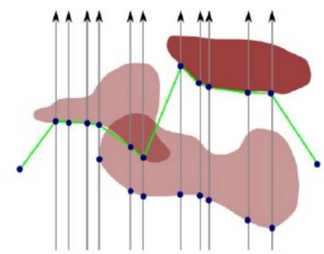
Automatic view optimization

Tracing tubular structures

Sketching Lines

- Creating a 3D line by sketching it in 2D (stroke) on the screen
- Simple solution
  - Independently pick 3D points for all points on 2D line and connect them
  - Problem: Can result in jaggy lines jumping to front an back
- Advanced solution
  1. Compute 3D candidate points on rays for all 2D points on stroke
  2. Select one candidate on each ray depending on candidates on other rays
  3. Connect selected candidates to form 3D line



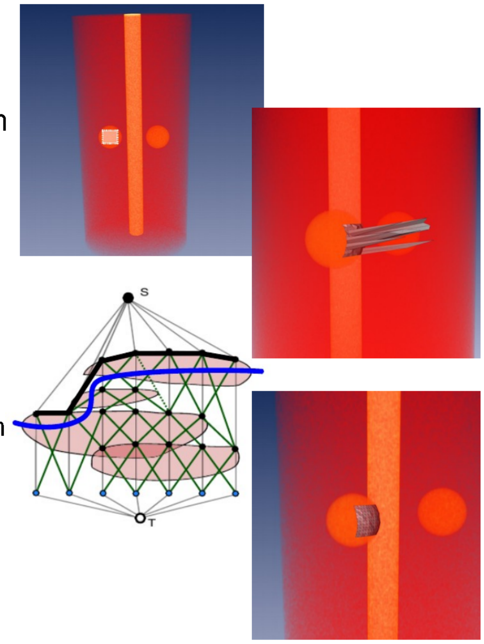


An alternative method to obtain a smoother line is to compute a line that fits the 3D points well while adhering to certain smoothness criteria. An example would be a B-spline curve. Such a line only approximates the 3D points instead of exactly interpolating them. This might or might not be acceptable for a certain application scenario.

Semi-automatic segmentation

Marking Regions

- Creating a 3D surface by marking a 2D region on the screen
  - Region is **usually** obtained from sketched 2D **closed line**
- Simple solution
  - Independently pick 3D points for all points in 2D region and connect them
  - Problem: Can result in jaggy surfaces jumping to front an back
- Advanced solution [Stoppel, 2013]
  1. Compute 3D candidate points on rays for all 2D points in region
  2. Select one candidate on each ray depending on candidates on other rays
  3. Connect selected candidates to form 3D surface



Similarly, to what has been described for the lines, a spline surface can be used to achieve the desired degree of smoothness. Again, this surface will only approximate the 3D points, i.e. it will not run exactly through the points. This might or might not be acceptable for a certain application scenario.

Overview

Direct manipulation is enabled by

> "Translating given 2D input into application-specific 3D locations, actions and manipulations regarding the volume"

- **Input types (2D → 3D)**
  - Points → picking
  - Lines → sketching
  - Regions → marking (often regions are defined by closed lines)
- Semantics (3D actions)
  - Selecting points, lines and surfaces in 3D
  - Segmentation, labeling, 3D painting
  - Manipulating transfer function or underlying data

Abstract:

Interaction with and manipulation of volumetric renderings and volumetric data are naturally non-trivial tasks when carried out on common 2D displays with mouse and keyboard input. This part will concentrate on techniques which take the provided 2D input and translate it into application-specific direct 3D actions and manipulations regarding the volume. Input types we will discuss include points (picking), lines (sketching) or regions (marking). Depending on the user task at hand, semantics of the resulting 3D actions range from simple selection of positions, lines and surfaces in 3D, to segmentation, labeling, 3D painting, and adaption of the transfer function or the data themselves. We will give a detailed description of the possibly needed additional or meta data for each technique. An overview of common applications of the mentioned techniques for imaging and simulation data from medicine and natural sciences will demonstrate the utility of the techniques.

Overview

Direct manipulation is enabled by

"Translating given 2D input into application-specific 3D locations, actions and manipulations regarding the volume"

- Input types (2D → 3D)
  - Points → picking
  - Lines → sketching
  - Regions → marking (often regions are defined by closed lines)
- **Semantics (3D actions)**
  - Selecting points, lines and surfaces in 3D
  - Segmentation, labeling, 3D painting
  - Manipulating transfer function or underlying data

Abstract:

Interaction with and manipulation of volumetric renderings and volumetric data are naturally non-trivial tasks when carried out on common 2D displays with mouse and keyboard input. This part will concentrate on techniques which take the provided 2D input and translate it into application-specific direct 3D actions and manipulations regarding the volume. Input types we will discuss include points (picking), lines (sketching) or regions (marking). Depending on the user task at hand, semantics of the resulting 3D actions range from simple selection of positions, lines and surfaces in 3D, to segmentation, labeling, 3D painting, and adaption of the transfer function or the data themselves. We will give a detailed description of the possibly needed additional or meta data for each technique. An overview of common applications of the mentioned techniques for imaging and simulation data from medicine and natural sciences will demonstrate the utility of the techniques.

Semantics

So far

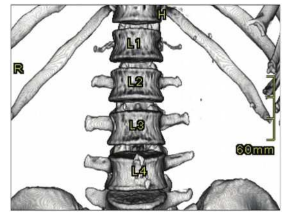
- **How to obtain** 3D points, lines and surfaces

Now

- **Use** of the 3D lines, points and surfaces

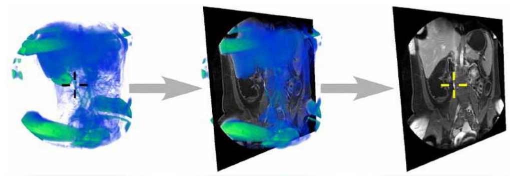
No additional comments.

Semantics: Picked Positions

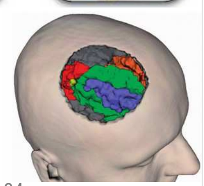


Use positions

- For placing
  - Labels e.g. [Kohlmann, et al., 2009]
  - Slices e.g. [Wiebel, et al., 2012]



- For specifying positions to
  - Manipulate the rendering [Guo, et al., 2011]
  - Manipulate the data [Bürger, et al., 2008]
  - Extract and display local data values



The 3D positions, 3D lines and 3D surfaces selected by the techniques described so far can be used for different basic tasks giving them a certain special meaning.

Positions can be used to place labels or markers for the feature they are lying in. The position together with the current viewing direction can be used used to place a cutting plane or slice in the data. Such a cutting plane allows a more detailed, yet 2D examination of the feature of interest.

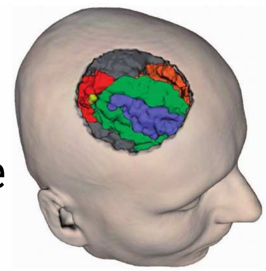
Futhermore, the positions can be used steer algorithms manipulating the rendering the data. These algorithms will extract the local data or locally applied rendering parameter and can adapt the data or the parameters corresponding to the specified location. The data or parameter extraction performed can also be used to display the local data or parameters to the users. This can be helpful for the analysis they are conducting.

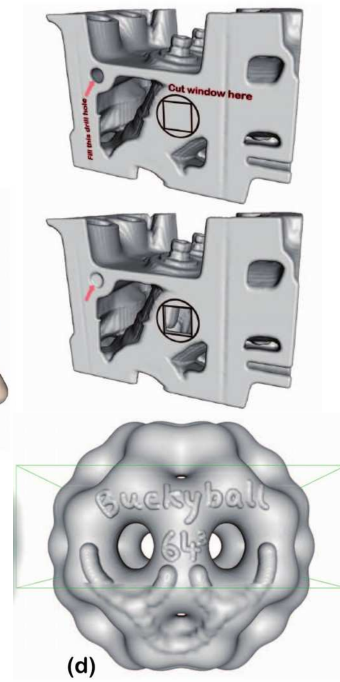
Manipulating the Data

- Data values close to or on the isosurface are slightly **altered** by an offset [Bürger, et al., 2008]

  → results in different isosurface

  - Filled holes
  - Removed parts
  - Altered shape



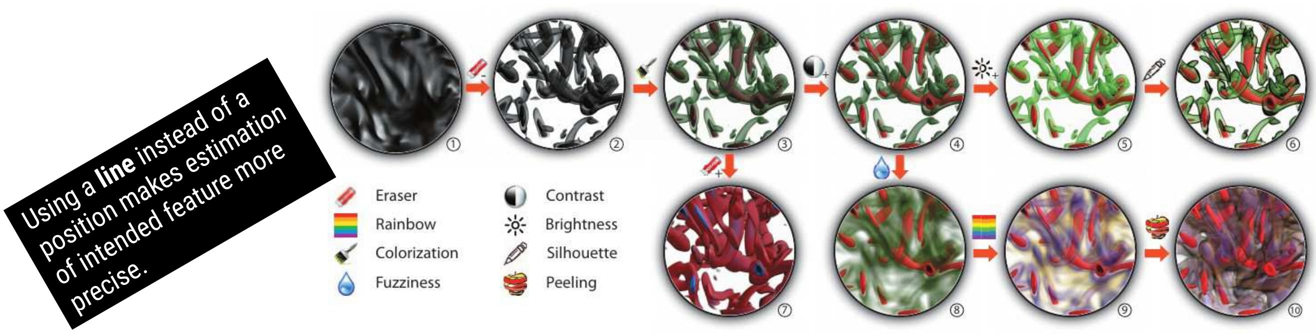


No additional comments.

Manipulating the rendering

- Transfer function is adapted for features found at picked position as prescribed by used tool [Guo, et al., 2011]

  → Rendering immediately changes



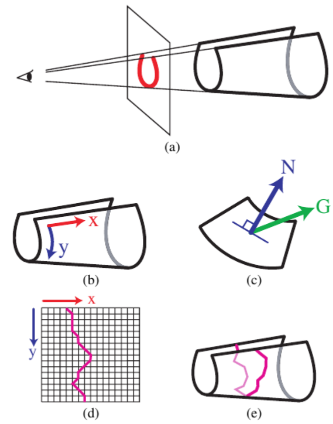
As mentioned the picked position can be used to determine the rendering parameters used for this particular location. In the context of volume rendering this boils down to finding the part of the transfer function applied to the data at this position. The method by Guo et al. allows to manipulate the relevant parts of the transfer function in various ways, and to achieve immediately displayed (WYSIWYG) changes of the rendering:

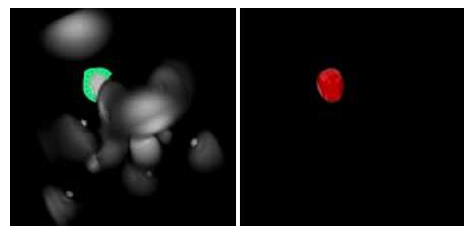
- Eraser (decreasing or increasing opacity)
- Rainbow: apply rainbow color map
- Colorization
- Fuzziness (make transfer function more smooth at boundary)
- Contrast
- Brightness
- Silhouette
- Peeling (different opacity changes to reveal hidden features)

Semantics: 3D Lines

Use lines

- For marking edges of features for 3D segmentation [Owada et al., 2005]
- Depth according to strength of gradient normal to cone constructed from the 2D stroke
- Generation of *segmentation constraints* along 3D line



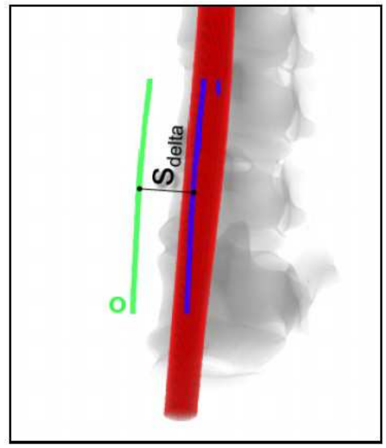


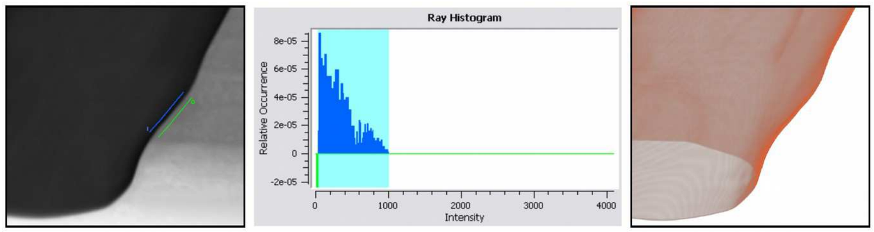
A line along the silhouette of a feature is drawn in 2D. From this line a surface similar to a cone is a cast in the viewing direction. The 3D line is determined to lie in this surface where there is a strong gradient normal to the surface in the data. The 3D line thus lies on the border of the 3D feature. The 3D line is then used to produce "constraints" for a segmentation algorithm.

Semantics: Strokes

Use lines

- For marking edges of features in order to select their histogram in a transfer function
  [Ropinski, et al., 2008]
- No actual 3D line, instead histogram of intensities along projected line



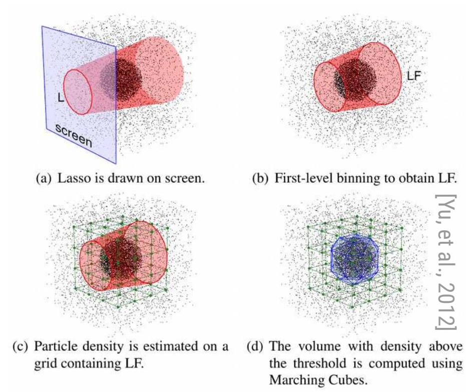


This technique allows to mark the contour of a feature of interest with a 2D line. This line is then offset two both of its sides, one offset line denoting the inside and one the background (outside) of the feature. 3D extensions of the lines in the viewing volume are then used to compute histograms of the values behind the lines. A difference of the inside and background histograms allows to determine the feature of interest and manipulate its rendering parameters.

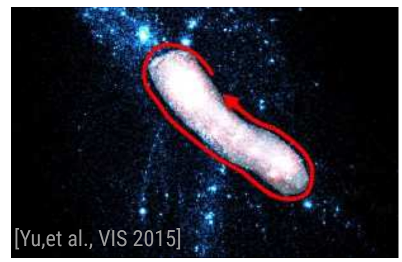
Semantics: 3D Lines

Use lines

- For encircling 2D projection of feature for 3D segmentation [Yu, et al., 2012], [Yu,et al., VIS 2015]
  - Especially for point clouds
  - Needed derived data: point density volume



(a) Lasso is drawn on screen. (b) First-level binning to obtain LF. (c) Particle density is estimated on a grid containing LF. (d) The volume with density above the threshold is computed using Marching Cubes.

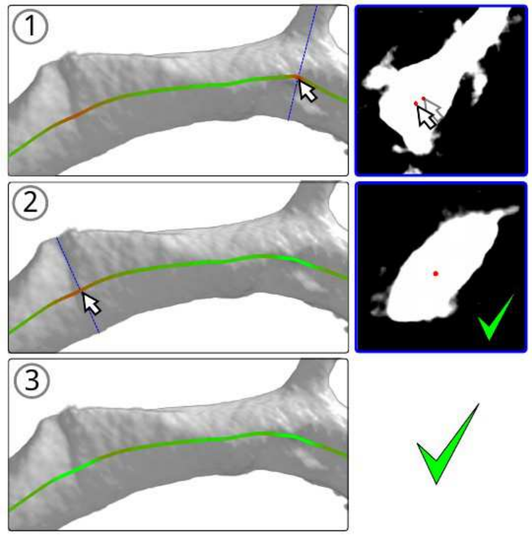


Similar to the method for segmenting regions in continuous volume data, 2D strokes can be used to segment regions in point clouds. Again the 2D line is projected into the volume along the viewing direction. The resulting cone-like surface is then used select parts of the point cloud depending on the density of the points.

Semantics: 3D Lines

Use lines

- For tracing the center of tubular structures [Diepenbrock, Ropinski, 2012]
  - Might be imprecise
  - Pick and drag resulting line to correct inaccuracies
- For guiding orthogonal slices through the volume [Diepenbrock, Ropinski, 2012]



No additional notes.

References

- F. Argelaguet, C. Andujar. A survey of 3D object selection techniques for virtual environments. Computers & Graphics 37.3 (2013): 121-136.
- K. Bürger, J. Krüger, R. Westermann. Direct Volume Editing. IEEE TVCG, 14(6):1388-1395, 2008.
- S. Bruckner, V. Šoltészová, E. Gröller, J. Hladuvka, K. Bühler, J. Yu, B. Dickson. Braingazer - visual queries for neurobiology research. IEEE Transactions on Visualization and Computer Graphics, 15(6):1497–1504, Nov. 2009.
- S. Diepenbrock, T. Ropinski. From Imprecise User Input to Precise Vessel Segmentations. EG Visual Computing for Biology and Medicine – 2012.
- P. Doux, S. Baillet, D. Lichau. Release notes – Amira for FEI Systems 6.0. FEI, 2015.
- R. A. Drebin, L. Carpenter, P. Hanrahan. Volume rendering. SIGGRAPH Comput. Graph., 22(4):65–74, June 1988.
- K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, D. Weiskopf. Real-Time Volume Graphics. AK Peters, 2006.
- E. Gobbetti, P. Pili, A. Zorcolo, 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.
- H. Guo, N. Mao, X. Yuan. WYSIWYG (What You See is What You Get) Volume Visualization. IEEE TVCG, 17(12):2106-2114, 2011.
- P. Kohlmann, S. Bruckner, A. Kanitsar, M. E. Gröller. Contextual Picking of Volumetric Structures. In Proceedings of IEEE Pacific Visualization, p. 185–192, 2009.
- J. Meyer-Spradow, T. Ropinski, J. Mensmann, 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.
- S. Parker, P. Shirley, Y. Livnat, C. Hansen, P.-P. Sloan: Interactive ray tracing for isosurface rendering. In Proceedings of the conference on Visualiza-tion '98 (Los Alamitos, CA, USA, 1998), VIS '98, IEEE Computer Society Press, pp. 233–238, 1998.
- H. Peng, Z. Ruan, F. Long, J. H. Myers, 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.
- H. Peng, et al.. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. Nature Communications 5, Article number: 4342, 2014.
- T. Ropinski, J. Praßni, F. Steinicke, K. Hinrichs. Stroke-Based Transfer Function Design. IEEE/EG International Symposium on Volume and Point-Based Graphics, page 41-48 – 2008.
- S. Stoppel: Visibility-Driven Depth Determination of Surface Patches in Direct Volume Rendering. Master's thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, 2013.
- S. Stoppel, H.-C. Hege, A. Wiebel. Visibility-Driven Depth Determination of Surface Patches in Direct Volume Rendering. EuroVis Short Papers 2014, p. 97-101, Eurographics Association, 2014.
- J. W. Wallis, T. R. Miller, C. A. Lerner, E. .C. Kleerup. Three-dimensional display in nuclear medicine. IEEE Trans. Med. Imaging 8 (4): 297–303, 1989.
- A. Wiebel, F. M. Vos, H.-C. Hege. Perception-Oriented Picking of Structures in Direct Volume Rendering. ZIB-Report 11-45, 2011.
- A. Wiebel, F. M. Vos, D. Foerster, H.-C. Hege. WYSIWYP: What You See Is What You Pick. IEEE TVCG, 18(12):2236-2244, 2012.
- L. Yu, K. Efstathiou, P. Isenberg, T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. IEEE TVCG, 18(12):2245-2254, Dec. 2012.
- L. Yu, K. Efstathiou, P. Isenberg, T. Isenberg. CAST: Effective and Efficient User Interaction for Context-Aware Selection in 3D Particle Clouds. IEEE TVCG, 22(1), Jan. 2016

Links to most papers can be found on the tutorial website:

http://tutorials.awmw.org/DirectVolumeInteraction

No additional comments.

More Information

- Picking and selection in virtual environments
  - [Argelaguet, Andujar, 2013]
- Tutorial website
  - http://tutorials.awmw.org/DirectVolumeInteraction

Questions?

Methods for picking 3D positions have also been developed in the context of virtual environments.

Course Outline

14:00 Introduction (*Timo Ropinski*)
14:05 **Visualization of Volume Data and the Need for Interaction** (*Timo Ropinski*)
14:45 Questions
14:50 **Direct Manipulation** (*Alexander Wiebel*)
15:30 Questions and Discussion
15:40 Break
16:15 **Interaction with Non-Standard Input and Output Devices** (*Tobias Isenberg*)
16:55 Questions
17:00 **Guided Navigation and Exploration** (*Stefan Bruckner*)
17:40 Questions and Discussion
17:50 **Closing Remarks** (*Alexander Wiebel*)
17:15 The End

No additional comments.