

 Open access • Proceedings Article • DOI:10.1117/12.273941

Volume rendering for interactive 3D segmentation — [Source link](#)

Klaus D. Toennies, Claus Derz

Institutions: Technical University of Berlin

Published on: 07 May 1997

Topics: Computer graphics, Volume rendering, Stereo display, Visualization and 3D computer graphics

Related papers:

- [Method and apparatus for retarding pipeline processing](#)
- [Method and system for displaying a representation of a three-dimensional object with surface features that conform to the surface of the three-dimensional object](#)
- [Method and system for managing color specification using attachable palettes and palettes that refer to other palettes](#)
- [System for obtaining correct byte addresses by XOR-ING 2 LSB bits of byte address with binary 3 to facilitate compatibility between computer architecture having different memory orders](#)
- [Display device interface including support for generalized flipping of surfaces](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/volume-rendering-for-interactive-3d-segmentation-143gyrvdgu>

Volume rendering for interactive 3-d segmentation

Klaus D. Toennies^a, Claus Derz^b

^aDept. Neuroradiology, Inst. Diagn. Radiology, Inselspital Bern, CH-3010 Berne, Switzerland

^bFG Computer Graphics, Technische Universitaet Berlin, D-10587- Berlin, Germany

ABSTRACT

Combined emission/absorption and reflection/transmission volume rendering is able to display poorly segmented structures from 3-d medical image sequences. Visual cues such as shading and colour let the user distinguish structures in the 3-d display that are incompletely extracted by threshold segmentation. In order to be truly helpful, analysed information needs to be quantified and transferred back into the data. We extend our previously presented scheme for such display by establishing a communication between visual analysis and the display process. The main tool is a selective 3-d picking device. For being useful on a rather rough segmentation, the device itself and the display offer facilities for object selection. Selective intersection planes let the user discard information prior to choosing a tissue of interest. Subsequently, a picking is carried out on the 2-d display by casting a ray into the volume. The picking device is made pre-selective using already existing segmentation information. Thus, objects can be picked that are visible behind semi-transparent surfaces of other structures. Information generated by a later connected-component analysis can then be integrated into the data. Data examination is continued on an improved display letting the user actively participate in the analysis process. Results of this display-and-interaction scheme proved to be very effective. The viewer's ability to extract relevant information from a complex scene is combined with the computer's ability to quantify this information. The approach introduces 3-d computer graphics methods into user-guided image analysis creating an analysis-synthesis cycle for interactive 3-d segmentation.

Keywords: 3-d display, visualisation, volume rendering, segmentation, image analysis

1 INTRODUCTION

Medical information from radiological images represent three-dimensional patient-specific information. Radiological information, however, is not a unique mapping from anatomical or physiological properties into an image function. Images represent a measurement of parameters that are deemed clinically useful because of an - often non-unique - relationship to anatomical or physiological features. Although some imaging modalities, such as MR imaging, produce very realistic mappings of anatomy, it should always be clear that it still is not anatomy which is measured. Images need to be interpreted and, if interpretation is carried out by the computer, results have to be confirmed by the radiologist.

Three-dimensional visualisation may serve several purposes in this context. It may simply be a means by which information from images is transmitted to the viewer in a natural fashion. In this case, the image needs to be interpreted implicitly or explicitly prior to display. Otherwise, a differentiation between clinically useful information and other content of the image as well as further classification of anatomical or pathological information is not possible. Subsequent display of analysed information reduces the information such that shape information and spatial relationships are recognisable in a two-dimensional display. The usefulness of such a rendition is measured by the ability to suppress uninteresting features, to enhance the extracted information and to visualise the three-dimensional nature of the data.

Surface and volume rendering methods were introduced for producing realistic renderings from interpreted medical image data^{1,2,3,4,5,6}. Because of the development of fast rendering methods⁵, even a sequence of 3-d renderings may be produced of the scene moving in space in order to improve the recognisability of shape and spatial relationships from display.

Such rendering scheme is always preceded image interpretation. Image interpretation requires an analysis on two-dimensional images because information about the data does not suffice for creating a 3-d display. The display of 3-d spatial relationships, which may be helpful for analysing the data, is impossible. For extending the analytical capabilities of 3-d visualisation, we previously presented a method for displaying non-analysed data. It is intended to enable interactive interpretation of images from 3-d display⁷. Our method is able to display of 3-d depth cues such as surface curvature or object classification even though objects may only partially classified.

The rendering algorithm is based on back-to-front voxel projection. An operator decides for each voxel the kind of display action that is to be carried out. The display of voxels depends on the classification that is evaluated locally based on function value and derivative. Background, object or surface voxels are the three possible classifications. The combination of EA volume rendering and RT volume rendering enabled a display which made objects recognisable, even if they were incompletely classified.

Recognition of objects from partial classification puts much of the burden of image interpretation on the viewer. A quantitative analysis of such objects may still not be possible because many voxels are misclassified. Visualisation can only be a part of a comprehensive analysis scheme, where 3-d display simplifies differentiation of voxels into classes of objects. For being truly useful, the method needs to be extended as to be able to integrate the knowledge extracted from the display back into the data. Adding the ability for segmenting data based on the display and for transferring the segmentation information into the data is the purpose of the work which is presented in this paper.

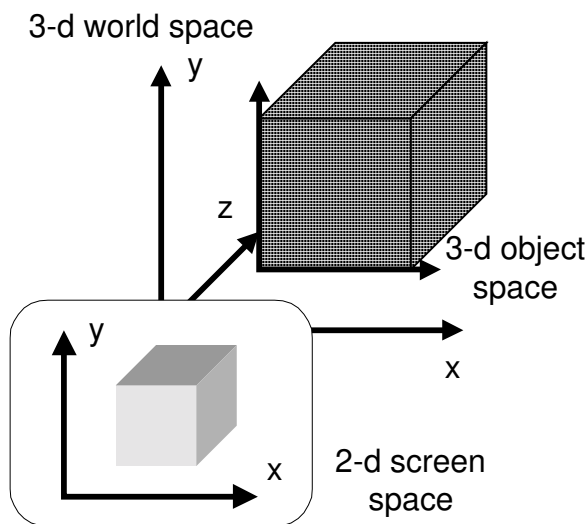


Fig.1: The voxel cube is projected from 3-d world space onto 2-d screen space. The cube has also its own 3-d object space associated to it.

2 DATA STRUCTURE AND BTF DISPLAY

Volume rendering is carried out on a three-dimensional data set consisting of cubic or non-cubic voxels. Each voxel consists of a group of elements containing

- its function value.
- the x-, y- and z-component of its surface normal.
- the length of the gradient.
- an optional class identifier.

The display operation is carried out as back-to-front projection from 3-d world space into 2-d screen space (see fig. 1). Prior to projection, a voxel mask is generated that circumscribes a voxel's projection into screen space. Projecting this mask rather than a single pixel for each voxel prevents the generation of holes during projection (see Toennies, Bock⁷ for details). After a voxel is read from the voxel cube, a visualisation operator is applied for deciding the kind of display action that shall be carried out. The operator is capable of initiating three different actions depending on the voxel's classification:

- A voxel may be part of the background. In this case no further action is taken.
- A voxel may be part of a surface. In this case, opacity is set to a surface-specific value and the reflected part of incoming light is modulated using an illumination model on the surface curvature.
- A voxel may be part of an object. In this case, object specific emission and absorption of possibly coloured light is computed and added to the output buffer.

Surfaces and object memberships depend on the voxel's function value and its gradient. A range for each object is specified by the user. Parts of surfaces may be rendered even though a complete surface definition is not given. Different objects or surfaces may be specified and displayed in the same data set.

Surfaces generally do not have a colour assigned to it because a surface, being at the transition between two different structures, cannot be attributed unambiguously to one of the two structures. Colouring surfaces without prior classification may lead to ambiguous results. Objects, however, may very well have a colour assigned to them making it easier to differentiate between various objects in the same display.

3 INTERSECTION PLANES AND MOVIE LOOPS

The rendition of a 3-d voxel set, using the method above, will give the experienced viewer some hints regarding the contents of the data. However, a classification based on the function value is rarely unique. Further improvement can be expected, if the viewer is allowed to point on structures in the image, names them, and then let the computer search for connected voxels in the 3-d data set. Such action may require removal of parts of the data in order to generate a display where 3-d relationships are distinguishable on the 2-d image.

specification of the first two points on the 2-display

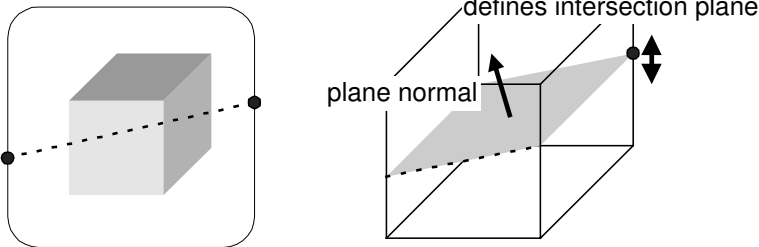


Fig. 2: The first two points of the intersection plane are specified on the original display. Subsequently, the third point and the direction of the plane normal a given on the display of a cube, where the plane is displayed with a 0° -tilt-angle.

on a display of the data set. This gives the orientation in the xy -plane of world space. For defining a tilt angle around the x -axis, a third point is specified on an extra window displaying a cube whose sides coincide with the xy - xz -, and yz -planes in world space and where the already specified line is displayed (see fig. 2). Subsequently, the direction of the plane normal has to be established. Voxels in front of the intersection plane - i.e., in the direction of the surface normal - will be removed while those behind the intersection plane will remain unchanged.

The parameters of the plane equation are passed to the visualisation operator. Additionally to its other decisions the operator decides in the presence of the intersection plane

- for voxels between intersection plane and the viewing position (i.e., in front of the intersection plane): To assign to them a background voxel classification requiring no further action.
- for voxels behind the intersection plane: To process them normally.
- for voxels on the intersection plane: To view them as intersected voxels.

The latter requires some further explanation. Intersection voxels are those, whose distance from the intersection plane is less than the diameter of a voxel. This is necessary because only few voxels will lay exactly with their centre on the intersection plane. The above mentioned definition of intersected voxels prevents the development of holes at the cost of several voxels possibly being projected on the same location of the intersection plane.

3.1 Intersection planes

The easiest approach for supporting the above-mentioned image analysis is to offer a dissection device. Intersection planes can be laid through the 3-d voxel set. An intersection plane is defined by the parameters a, b, c, d of its plane equation $a \cdot x + b \cdot y + c \cdot z + d = 0$. This equation is true for all points in the plane. The vector (a, b, c) is a normal to that plane and d is the distance from the plane to the origin. Points in front of the plane - i.e., in direction of the plane normal - give a result greater than zero, when their co-ordinates are put into the plane equation.

The intersection plane is defined by three points. The user specifies the first two of them

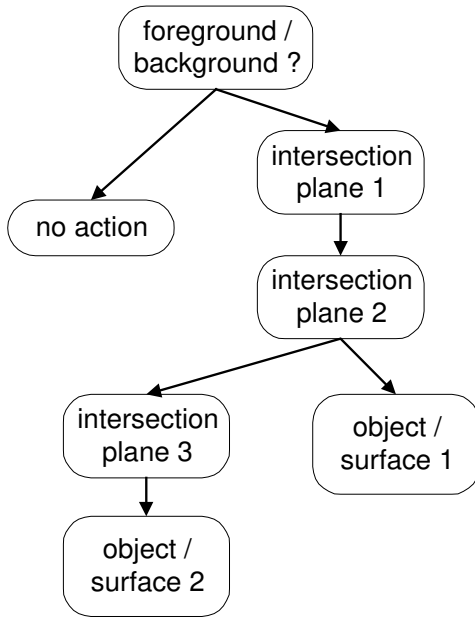


Fig. 3: Including the intersection operation into the visualisation operator: Intersection planes are defined there, where they shall be applied. Whereas in this example planes 1 and 2 cut both of the two objects, plane 3 selectively cuts only object 2. It is important to place any intersection plane behind the foreground / background decision in order to avoid the display of opaque background voxels.

Intersected voxels are treated like background voxels, if they are classified as such. Otherwise, an intersection plane will hide everything which lays behind it. All the other voxels forming the intersection plane will be displayed with their original grey value, since the dissection should give the impression of cutting into the volume. The grey values are shaded according to the orientation of the intersection plane with respect to viewing and illumination positions as to give a realistic impression when the structure is rotated.

Several intersection planes may be defined together in order to cut appropriate portions of the 3-d scene out of the whole data set. As there might be already a classification of structures by function value, we can also make the intersection plane selective. In this case, a set of explicitly defined objects are not intersected, therefore sticking out of the intersection plane. This further enhances the analysis capabilities of the visualisation tool.

Integrating intersection planes into the existing visualisation tool is easy, because of the way of using a freely definable operator for deciding on the kind of visualisation action. We only add the new operation „intersection“ before those structures that shall be intersected (see fig. 3). If the complete data set shall be intersected, then the intersection plane has to be inserted behind the foreground / background decision, in order to prevent complete opacity at background voxels.

3.2 Movie loops

Although not being suitable for interaction we included a tool for creating movie loops for enhancing the display of intersected data sets. The movie tool consists of a simple user interface, where rotation angles and translation vectors are specified for transforming the scene during

the loop. Subsequently, images are created and displayed for the movie loop.

If the visualisation operator contains an intersection plane, then two different modes are possible for applying the intersection. If the plane is defined in object space, then it moves with the object. If the object is transformed, the intersection planes are transformed accordingly. This enables the display of the intersected structure as seen from different points of view. If the plane is defined in device space, then the movie loop causes the voxel cube to be moved through the plane because transformations are not applied to the intersection planes. Intersection planes in device space enable the assessment of the shape of structures as they are moved through the planes. This is particularly useful for a combination of translations and a selective intersection plane. In this case, the viewer sees the varying intersection shape of a structure while it is moved through the plane.

4 3-D OBJECT SELECTION AND SEGMENTATION

The operations described in the previous section support the process of identifying structures of interest in the data set. They do not incorporate the knowledge into the data. This is the purpose of the 3-d segmentation module. A continuing interpretation loop is defined, consisting of 3-d display, analysis and subsequent incorporation of knowledge into the data set (see fig. 4). Improvement of the data display through interpretation is continued until the quality of segmentation suffices for quantitative analysis (e.g., for volumetric measurements).

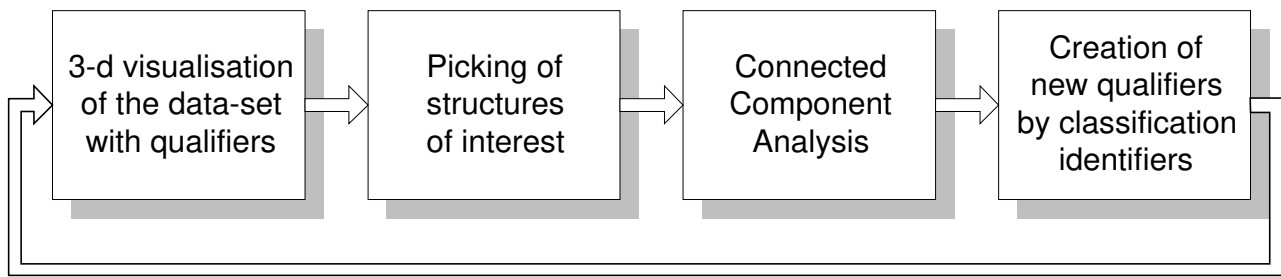


Fig. 4: Data analysis is a continuing process where the display is constantly improved.

For this purpose, we developed a 3-d picking device for selecting objects in the data set on the screen. Picking may be followed by a connected component analysis in order to find and segment the selected object in the data set. The knowledge on the segmented object can be integrated as a classification into the data set. The classification, in turn, serves as additional information for further renditions of the data.

4.1 Picking structures on the display

Picking takes place on a display of the data set. The range of function values, that the object of interest is expected to have, is specified by the user such that the object can be distinguished, e.g., by colour, from background. Intersection planes are defined if necessary in order to make the structure of interest visible. The user then selects the object by a mouse click at a location in screen space where he sees the object. This defines origin of a ray in device space. Its direction is orthogonal to the screen space co-ordinate system. For finding the voxel being hit by the ray, the ray is digitised in object space using a digital differential analyser algorithm⁸ (see fig. 5).

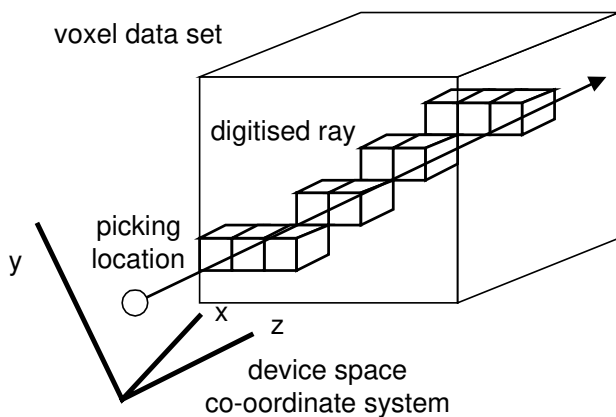


Fig. 5: Picking a structure of interest defines a ray into devices space that is digitised in object space

The first voxel, that is hit by the ray and that fulfils a given qualification condition, is utilised as a seed voxel for connected component analysis. The qualifying condition may be any requirement which can be evaluated locally on the 3-d data set. It may be, e.g.,

- the first non-background voxel.
- the first voxel in a specific range of function values.
- the first voxel that is member of a given object class.

Using this strategy, we are able to select structures that are not the first being hit by the ray. This is useful, if several structures are visualised in a semi-transparent mode and some inner object shall be selected that is visible through other objects.

In order to give a fast feedback as to whether a qualified structure is found on the path of the ray, an object buffer is created. For each pixel in screen space, the object buffer contains information on objects that are visible at this location. The object buffer is created during visualisation when voxel are projected onto locations in screen space. When a screen space location is picked, the object buffer is consulted and immediate feedback is given to the user as to whether the picked object can be found on a ray originating at this location.

4.2 Connected component analysis

After picking a seed voxel, segmentation is carried out by connected-component analysis based on a 27-voxel-neighbourhood. Voxels are included in the segment, if they fulfil the same qualification conditions than that for the seed voxel and if there is a path of neighboured voxels to the seed voxel that consists of voxels also fulfilling this condition.

The connected-component module assigns a unique class identifier to all voxels found to be connected to the seed voxel and returns the number of voxels that are part of the segment. After carrying out the connected component analysis, the number of voxels belonging to the segmented structure is displayed. This information can be used for deciding whether the selected voxel set has the expected size. If the analysis is found to be successful, the segmented structure can be given a name. The already defined class identifier is included into the data set. It can be used to further qualify the display.

5 RESULTS

The visualisation and analysis method was applied to 3-d CT data sets. Fig. 6 shows two visualisations of such data set before any further analysis was carried out. Three different structures, the bone, the cerebro-spinal fluid, and a large lesion, were qualified by their function value. The goal was to separate the ventricular system and the lesion. As the function value is a rather poor qualifier for all structures except for the bone, further analysis had to be carried out.

With intersection planes, as shown in figs. 7 and 8, both, the ventricular system and the lesion were easier visible although it was clear that both structures were not separated from other structures with the same function value. Fortunately, it appeared that they were geometrically separate from these other structures. Therefore, picking was expected to be successful. After applying this method followed by connected-component-analysis, we received the desired results as shown on the right sides of the two figures.

The computation time on a SUN 4/10 for a display of the data set (117 slices with 256*256 voxels per slice) was 32 seconds. If intersection planes were included, the computation time increased, depending on the number of intersection planes, to 37..55 seconds for one to three planes.

Picking and subsequent ray digitisation was not very time-consuming taking 0.7 to 1.0 seconds. However, the computation of the object buffer took up to 3 minutes because objects were assigned bit positions in the object byte. The bit operations

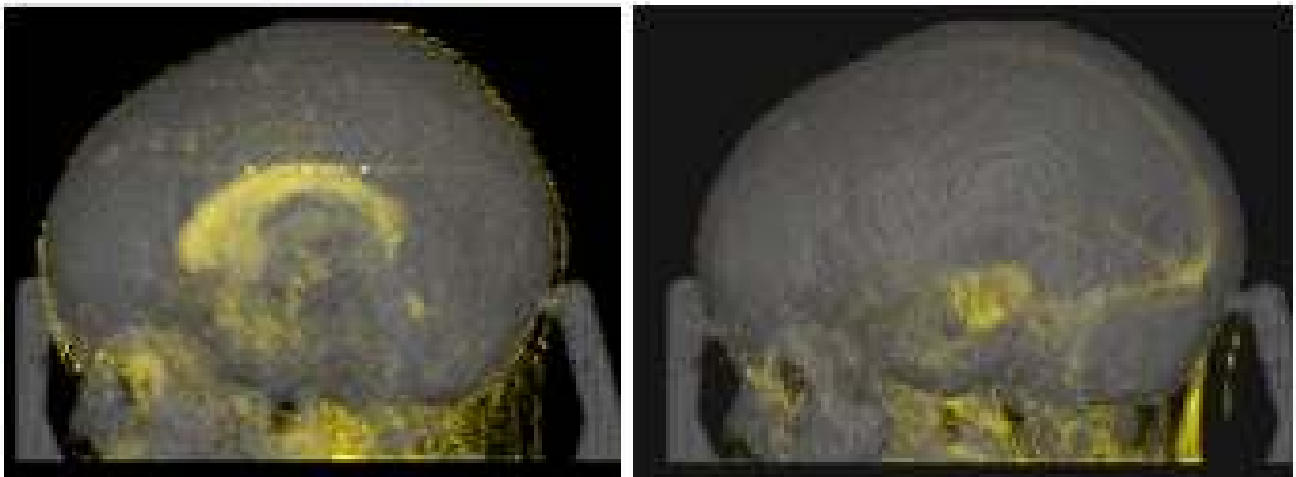


Fig. 6: Display of a 3-d data set from CT. Bone is made transparent in order to reveal the CSF (left) and a large lesion (right). The two structures are not easy to distinguish because of the poor qualification based on function value.

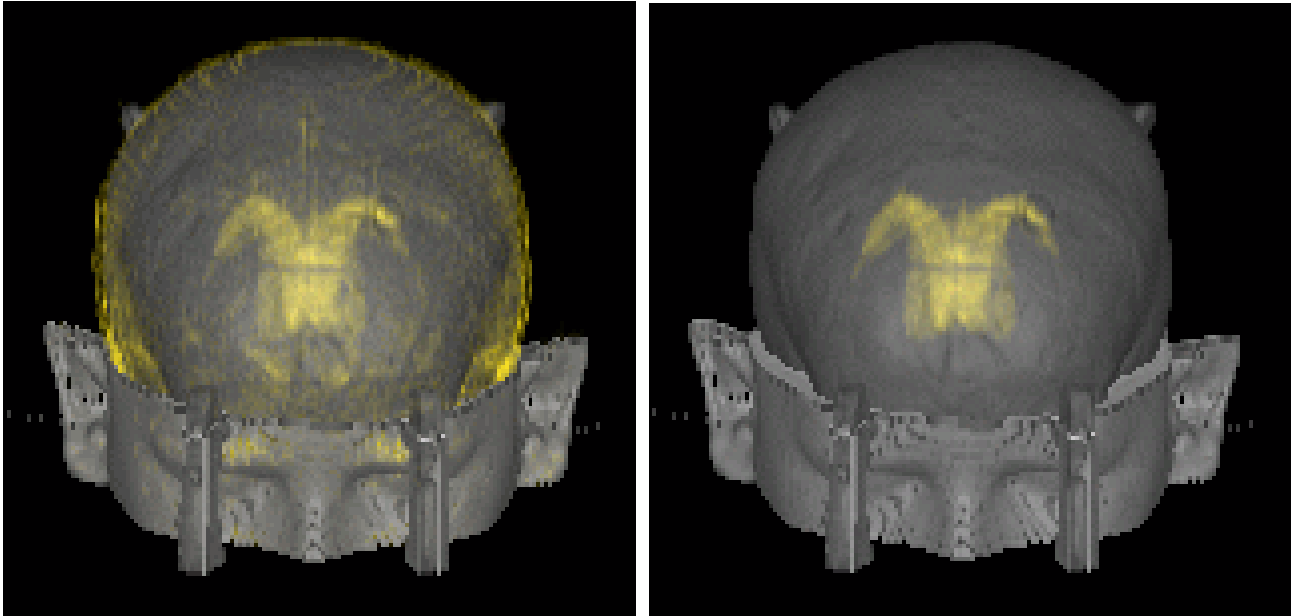


Fig.7: A selective intersection plane, not applying to bone and CSF, cause the two to stick out of the plane (left: without picking the ventricular system; right: after picking the ventricular system)

slowed down the computation time considerably. Because of the relatively fast digitisation, we will probably cease using the object buffer.

6 CONCLUSIONS

We presented a fast visualisation method for display of data at different stages of analysis. The data structure and visualisation concept is defined such, that interaction features were easy to integrate. We allow a user-specified preparation of the data by interactive analysis that makes full use of the radiologist's ability to perceive and interpret 3-d medical image data. This is a first steps towards integrating synthesis and analysis of radiological data in a repetitive cycle, where data may be displayed at any state of analysis and where the display supports further data interpretation.

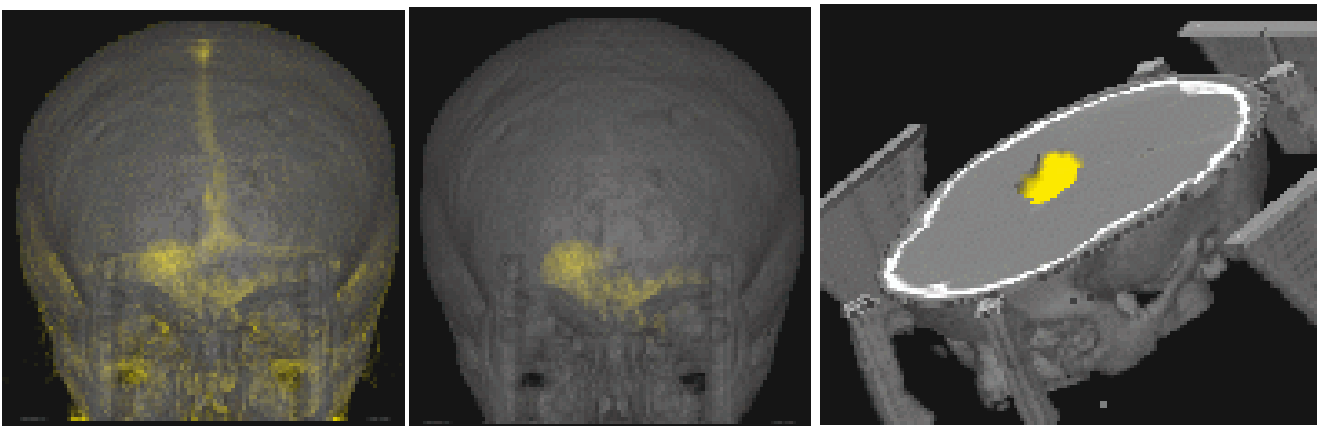


Fig. 8: Analysis sequence for the lesion. Left: display without further selection; middle: after picking the lesion; right: after applying a selective intersection plane and a rotation.

7 BIBLIOGRAPHY

1. H. Scharnweber, K.D. Toennies, 'Three-dimensional reconstruction and display of complex anatomical objects', *Proc. of the 1984 Intl. Joint Alpine Symposium*, IEEE Computer Society Press, Silver Spring, Md., 1984, pp.7-10.
2. R.A. Drebin, L. Carpenter, P. Hanrahan, 'Volume rendering', *Computer Graphics*, Vol. 22(4), 1988, pp.65-74.
3. M. Levoy, 'Display of surfaces from volume data', *IEEE Computer Graphics and Applications*, Vol. 8(3), 1988, pp. 29-37.
4. J.K.Udupa, D. Odhner, 'Fast visualisation, manipulation and analysis of binary volumetric objects', *IEEE Computer Graphics and Applications*, Vol. 11(3), 1991, pp.53-62.
5. J.K.Udupa, D. Odhner, 'Shell rendering', *IEEE Computer Graphics and Applications*, Vol. 13(6), 1993, pp.58-67.
6. K.D. Toennies, U. Tronnier, '3-d modeling using an extended cell enumeration representation', *Computer Graphics*, Vol. 24(5), 1990, pp.13-20.
7. K.D. Toennies, S. Bock, ' Exploration of 3-d medical data using combined emission and reflection volume rendering', *Proc. SPIE, Vol. 2707 (Medical Imaging '96: Image Display)*, 1996, pp.44-51.
8. W.D.Fellner, *Computer Grafik*, Wissenschaftsverlag, Reihe Informatik, Bd.58, 1988.