V-Objects: Flexible Direct Multi-Volume Rendering in Interactive Scenes

Sören Grimm* University of Technology Vienna, Austria

Stefan Bruckner† University of Technology Vienna, Austria

Armin Kanitsar‡ Tiani Medgraph AG Vienna, Austria

Eduard Gröller§ University of Technology Vienna, Austria

ABSTRACT

In this paper we describe methods to concurrently visualize and interact with multiple volumetric objects. We introduce the concept of V-Objects. V-Objects represent abstract properties of an object connected to a volumetric data source. The key to flexibility is the explicit separation of V-Objects and data sources. We present a method to perform direct volume rendering of a scene comprised of an arbitrary number of V-Objects. Using this concept, we present a novel approach to combine visualization, interaction, and exploration techniques. We focus on medical applications, such as surgical planning, diagnosis, and education. An interactive and intuitive 3D framework for visualization of multiple volumetric objects has been developed. We have integrated our methods into a commercial medical visualization system.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing;

Keywords: medical visualization, multi volume rendering, volume raycasting

1 INTRODUCTION

Direct volume rendering is an important and flexible technique for visualizing 3D data. It allows the generation of high quality images without a need of an intermediate interpretation. Traditionally medical volume visualization systems feature only simply scenes consisting of a single volumetric data set. It has been proposed to extend these scenes to a more complex description [10]. This, however, is rarely used, due to lack of benefit. In this paper we introduce a flexible data structure for representing scenes containing multiple volumetric objects. We demonstrate that medical application can take advantage of our data structure by employing direct volume rendering of multi volume scenes. The main contribution of this paper is to show that standard volume visualization systems can greatly extend their flexibility by supporting concurrent display of multiple volumetric objects. We give practical examples for possible applications.

2 RELATED WORK

The past decade has seen significant progress in volume visualization, driven by applications such as medical imaging. A number of different volume rendering algorithms have been developed, improved, and extended [7, 5, 9]. Today, it is possible to perform interactive high-quality volume rendering on commodity hardware [2, 12]. Hybrid algorithms have been designed, which allow concurrent display of intersecting volumetric and polygonal objects [8, 4]. Direct rendering of scenes consisting of multiple volumetric objects, however, has gotten less attention. With the increasing performance of modern hardware, we feel that this topic will become increasingly important in the future. Our work was inspired by Leu and Chen, who introduced a two-level hierarchy for complex scenes of non-intersecting volumes [6]. Additionally, we base our approach on the techniques for combining intersecting volumetric objects by Cai and Sakas [1].

3 V-OBJECTS

A V-Object is an element of a scene description which is connected to a volumetric data source. The V-Object comprises the following visual properties:

- *Illumination:* The selected Illumination model and its parameters. For example, for the Phong-Blinn Illumination model the ambient, diffuse, specular, and emissive material coefficients.
- *Transfer Functions:* For each defined region in the attached volumetric data source a mapping function between scalar values and colors as well as opacities is stored.
- *Region of Interest:* An arbitrary number of planes defining a convex region of interest.
- *Transformation:* An affine transformation defining position, orientation, and scaling of the V-Object.

The separation of visual properties and volumetric data sources allows for an arbitrary number of varying representations of the same data source within one scene as illustrated in Figure 1. This is achieved by assigned several V-Objects to the same volumetric data source. To take full advantage of the V-Objects we apply direct volume rendering to simultaneously visualize a scene consisting of several V-Objects. V-Objects allow feature centric visualization of volumetric data sets by simple modification of their properties.

3.1 Direct Volume Rendering

We choose volume raycasting for visualization of multiple V-Objects to allow direct simultaneous display of surfaces and interior structures. This is especially important in medical applications. The optical model used in direct volume rendering allows meaningful combination of multiple amorphous as well as opaque objects.

Consider a ray that is cast into a scene, as illustrated in Figure 2, passing through a pixel in the image plane and intersecting with several volumetric objects. A ray can either intersect a single object or multiple objects along it's path through the scene. In our system a volume is decomposed in equally sized blocks (32^3). We can quickly determine whether a block contributes to the final image. We project for each visible block onto the image plane. Thereby, we can determine how many volumes a ray intersects. We differentiate between no hit, one hit, and more than two hits. If a ray does not hit any block, it is not processed at all. If rays hit only blocks of one volume they are all processed in parallel in a block-wise manner. If a ray hits more blocks from more than one volume it has to process the volumes simultaneously.

*e-mail: grimm@cg.tuwien.ac.at

†e-mail: bruckner@cg.tuwien.ac.at

‡e-mail: kanitsar@tiani.com

§e-mail: groeller@cg.tuwien.ac.at

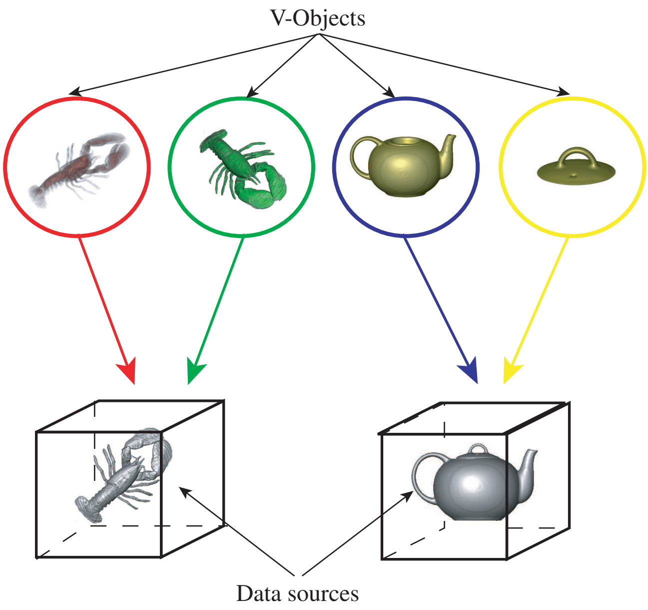

Figure 1: Different representations of the same data source using V-Objects.

In general, for a single object the final color and opacity of the image pixel is determined by the following integral:

$$I(x,r) = \int_0^L C_\lambda(s)\mu(s)e^{(-\int_0^s \mu(t)dt)} ds \quad (1)$$

L is the length of ray r. Considering the volume as being composed of particles with certain densities μ, then these particles receive light from all surrounding light sources and reflect this light towards the eye point according to their specular and diffuse material properties. Additionally, the particles may also emit light on their own. Thus, in Equation 1, C_λ is the light of wavelength λ reflected and/or emitted at location s in the direction of r. To account for the higher reflectance of particles with larger densities, the reflected color is weighted by the particle density. The light scattered at s is then attenuated by the densities of the particles between s and the eye according to the exponential attenuation function.

However, in practise it is impossible to evaluate the integral analytically. Therefore, the integral is approximated by repeated application of the over-operator [11] in front-to-back order. That is, at each re-sample location, the current color and alpha values for a ray are computed in the following way:

$$\begin{aligned} c_{out} &= c_{in} + c(x)\alpha(x)(1-\alpha_{in}) \\ \alpha_{out} &= \alpha_{in} + \alpha(x)(1-\alpha_{in}) \end{aligned} \quad (2)$$

c_{in} and α_{in} are the color and opacity the ray has accumulated so far. x is the reconstructed function value and $c(x)$ and $\alpha(x)$ are the classified and shaded color and opacity for this value.

If it comes to simultaneous processing of multiple volumes one has to decide how they should be combined. We consider volumes as clouds of particles and take into account all volume simultaneously at the corresponding sample position. The simultaneously compositing of multiple volume is achieved by sequentially applying Equation 2 for each individual volume. Hereby we obtain an approximation for compositing multiple volumes at the same location.

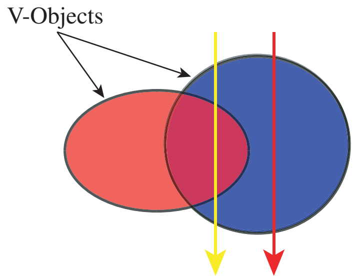

Figure 2: Red ray passes through a single V-Object and the blue ray passes through two V-Objects.

4 Application of V-Objects

In general, in a volume rendering system there exist several means to enhance visual perception. Such means, are for example transfer function, segmentation and clipping. Since their introduction to volume visualization by Levoy [7], piecewise linear transfer functions mapping scalar values to colors and opacities are featured in virtually every volume visualization system. Many researchers have proposed to extend transfer function to higher dimensions to increase their usability, as their specification is a non trivial task. Some approaches for semi automatic transfer function definition have been presented, however fully automatic transfer function selection remains a widely unsolved problem. Still, much research needs to be done to produce results automatically as shown in Figure 3.

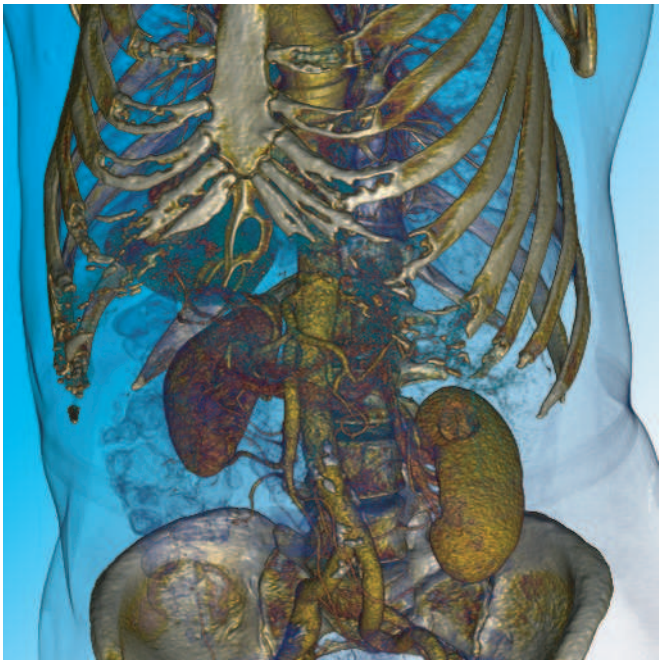

Figure 3: Transfer function example.

One way to simplify the transfer function specification is segmentation. Many methods exists to identify certain structures within in the data. Most of these approaches produce a labelling of the data. Different transfer functions can be assigned to the identify regions or objects. Figure 4 shows an example of segmentation based on region growing. The main vascular structures and the kidneys have been segmented.

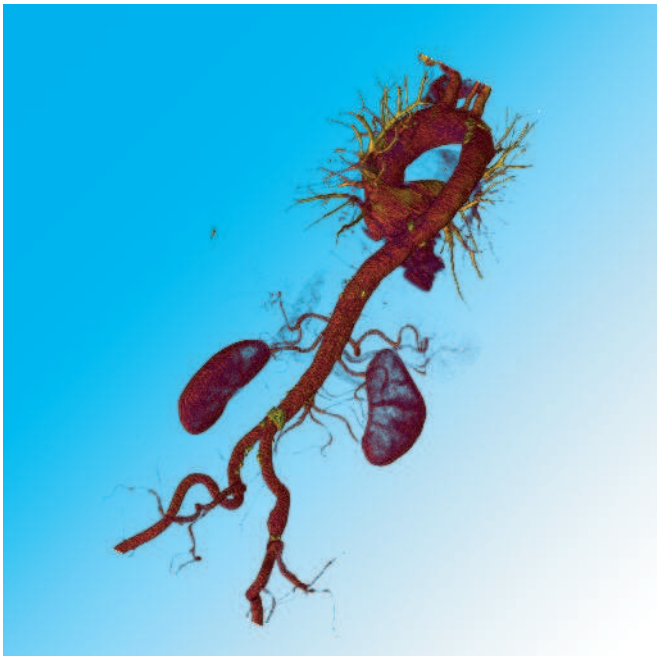

Figure 4: Segmentation example: The main vascular structures and the kidneys have been segmented.

One problem in the visualization of volumetric data is occlusion. While transparency can be useful to simultaneously display different structures of interest, it can lead to cluttering when used extensively. It is therefore common to cut away opaque objects to reveal occluded features. Cutting can be performed using axis aligned or arbitrary oriented planes, convex regions or arbitrarily shaped objects. However complex cutting shapes are often difficult to interpret by the user. Figure 5 shows the use of cutting planes to reveal an aneurisms.

In the following we show how such basic tools can be greatly enhanced by the use of V-Objects.

4.1 Exploring V-Objects capabilities

Though many systems allow modification of transfer functions and illumination properties can be specified on a per object basis certain limitations apply: objects typically can not intersect, are unique and static. By assigning V-Objects to components of a data set we can overcome these limitations in a natural way. For example, with V-Objects it is possible to move an object while keeping a virtual copy in place. These two objects can have a totally different appearance. This capability can be used in applications such as surgical planning, education, illustration, and for investigation or exploration of data. In the following we give several examples for the use of V-Objects in combination with the previous described basic tools. We show examples for moving objects, simultaneously changing the appearance of objects, time varying, and multi-modal data. As some of these concepts are very hard to illustrate in still images, we have produced several animation sequences. Our application features an interactive tool for the generation of animation sequences using key-framing. V-Objects can be interactively positioned in the scene and their properties can be modified. Between the key-frames, V-Object states are interpolated. This enables specification of animation paths, transfer function fades, light movement, control

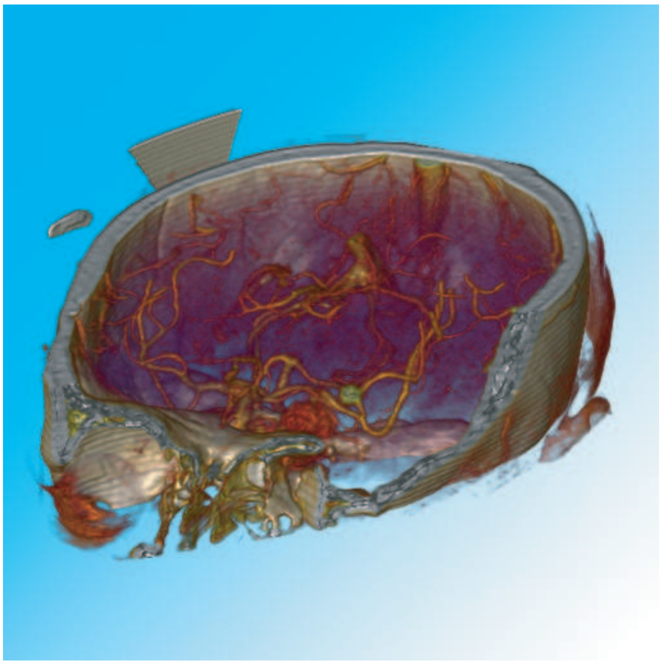

Figure 5: Clipping example.

of clipping planes, change of data sources, enabling and disabling of objects, etc. For each of these properties, different interpolation schemes can be applied. In the following we present our application examples.

4.1.1 Advanced Browsing Techniques

Traditional techniques for inspecting volumetric data like cutting involve removing portions of the data. This has the disadvantage of potentially hiding important contextual information. McGuffin et al. [3] have presented novel tools for browsing volume data which employ transformations and deformations of semantic layers contained in the data set. In the future, we feel that multi-volume rendering will be an important tool to improve the visual quality of such interaction techniques. In Figure 8, we demonstrate that V-Objects can be used to realize this kind of visualization. Although we support only affine transformations at present, we are confident that we can extent our concept to more complex deformations in the future. One the other hand V-Objects can be used to indicated the positions of displaced structures. Multiple V-Objects can be assigned to the same structure in the data, only one of these V-Objects is deformed, the other objects remains in place to indicate the original position in space. Transparency is especially useful to indicate the position while not hiding surrounding important information. In Figure 7, we show an example of such an application of V-Objects. The images represents stills of a animation we made. In the animation you can see for example, that a kidney is relocated to the left to reveal a tumor. Also other interesting features, such as transfer function fading, moving, and object specific cutting planes are shown.

4.1.2 Time Varying Data

Visualization of time varying volume data is a very complex task. Animations due to their dynamic nature often do not allow an in depth analysis of certain data characteristics. Static images on the other hand, often suffer from cluttering when many time-steps are visualized. Therefore, it has been proposed to apply more advanced projection and mapping techniques to aid the understanding of such data. For example, Woodring at al. [13] apply hyper-slicing to a 4D dataset. The flexibility of V-Objects allows to use a variety of different mappings using the combination of transfer functions and spatial arrangement of objects. We show an example of a time varying data set, see Figure 9. It is a electrocardiogram triggered CT scan of beating heart. Time is mapped to color and opacity. This type of visualization allows the three dimensional examination of several time steps simultaneously. The fanning in time allows to convey similarities and differences in the progress of time. Furthermore, a topological relationship between different time steps is visualized. In general, V-objects support the explorations of useful layouts and mappings. In the future, we therefore seek to further investigate the application of multi volume rendering to 4D data visualization.

4.1.3 Multi Modal Imaging

Multi-modal imaging is a method for combining disparate sets of 3D imaging data that contain complementary information on overlapping length scales. In medicine, for instance, morphological modalities are combined with functional modalities to increase the information content of the resulting image. The concept of V-Objects inherently includes the ability to perform multi-modal visualization of registered data sets. In Figure 6 we show an example of a combination of computed tomography (CT) and positron emission tomography (PET). While the CT method supplies high precision, it is difficult to distinguish between tumors and healthy tissue. PET, on the other hand, is a functionally oriented method that allows to identify tumors, but provides lower resolutions.

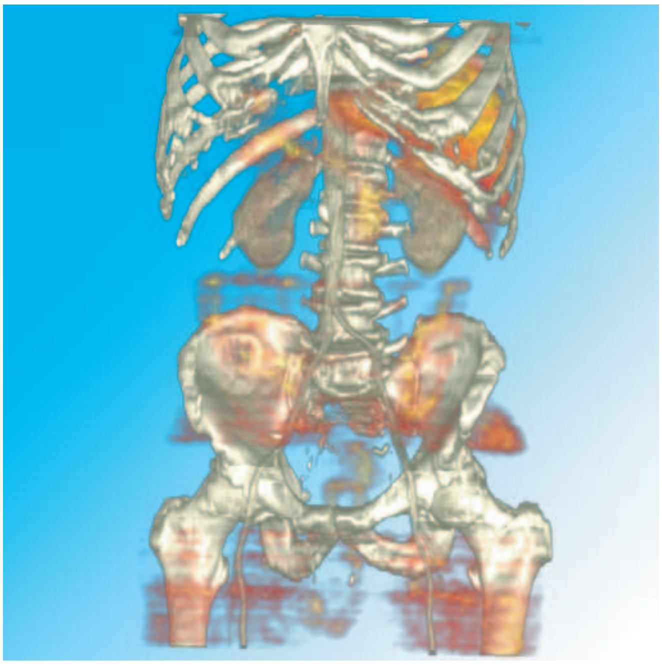

Figure 6: Fusion of CT and PET scan based on V-Objects.

5 Conclusion and Future Work

We presented V-Objects, a concept of modelling scenes consisting of multiple volumetric objects. We have demonstrated that this concept, in combination with direct volume rendering, is a promising technique for visualizing medical data. We showed three examples for its application, browsing, time varying data, and multi-modal imaging. Each of them showed that the concept of V-Objects can provide advanced means to explore and investigate data. In the future we will further investigate the topic of multi-volume visualization as we believe it has great potential to improve medical applications.

The animation sequences described in this paper and additional material is available at:
http://www.cg.tuwien.ac.at/research/vis/adapt/2004_vobjects/index.html

6 Acknowledgements

The work presented in this publication has been funded by the ADAPT project (FFF-804544). ADAPT is supported by *Tiani Medgraph*, Vienna (http://www.tiani.com), and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria. See http://www.cg.tuwien.ac.at/research/vis/adapt for further information on this project. The used data sets are courtesy of Univ.-Klinik Innsbruck, AKH Vienna, Univ.-Klinikum Freiburg, and NLM.

References

[1] Wenli Cai and Georgios Sakas. Data intermixing and multi-volume rendering. In P. Brunet and R. Scopigno, editors, *Computer Graphics Forum (Eurographics '99)*, volume 18(3), pages 359–368. The Eurographics Association and Blackwell Publishers, 1999.

[2] Frank Dachille, Kevin Kreeger, Baoquan Chen, Ingmar Bitter, and Arie Kaufman. High-quality volume rendering using texture mapping hardware. In Stephen N. Spencer, editor, *Proceedings of the 1998 EUROGRAPHICS/SIGGRAPH Workshop on Graphics Hardware (EGGH-98)*, pages 69–76, New York, August 31– September 1 1998. ACM Press.

[3] Michael Mc Guffin, Liviu Tancau, and Ravin Balakrischnan. Using deformations for browsing volumetric data. In *Visualization*, pages 401–408, 2003.

[4] Kevin A. Kreeger and Arie E. Kaufman. Mixing translucent polygons with volumes. In David Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 191–198, San Francisco, 1999. IEEE.

[5] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics*, 28(Annual Conference Series):451–458, July 1994.

[6] Adrian Leu and Min Chen. Modelling and rendering graphics scenes composed of multiple volumetric datasets. In David Duke, Sabine Coquillart, and Toby Howard, editors, *Computer Graphics Forum*, volume 18(2), pages 159–171. Eurographics Association, 1999.

[7] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[8] Marc Levoy. A hybrid ray tracer for rendering polygon and volume data. *IEEE Computer Graphics and Applications*, 10(2):33–40, March 1990.

[9] K. Mueller, N. Shareef, J. Huang, and R. Crawfis. High-quality splatting on rectilinear grids with efficient culling of occluded voxels. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):116–134, April/June 1999.

[10] David R. Nadeau. Volume scene graphs. In *Proceedings of the 2000 IEEE symposium on Volume visualization*, pages 49–56, 2000.

[11] T. Porter and T. Duff. Compositing digital images. *Computer Graphics*, 18(3):253–259, July 1984.

[12] Stefan Roettger, Stefan Guthe, Daniel Weiskopf, Thomas Ertl, and Wolfgang Strasser. Smart hardware-accelerated volume rendering. In C. D. Hansen G.-P. Bonneau, S. Hahmann, editor, *Proceedings of the symposium on Data visualisation 2003*, pages 231–238. Eurographics Association, 2003.

[13] Jonathan Woodring, Chaolin Wang, and Han-Wei Shen. High dimensional direct rendering of time-varying volumetric data. In *Visualization*, pages 417–414, 2003.

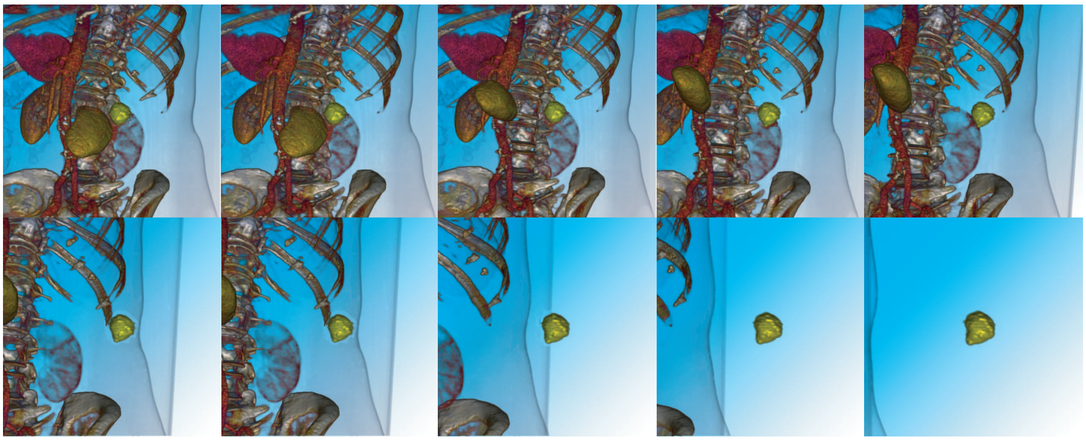

Figure 7: Stills of animation to illustrate advanced browsing of volumetric data based on V-Objects. Enhancing anatomical features by spatial displacement.

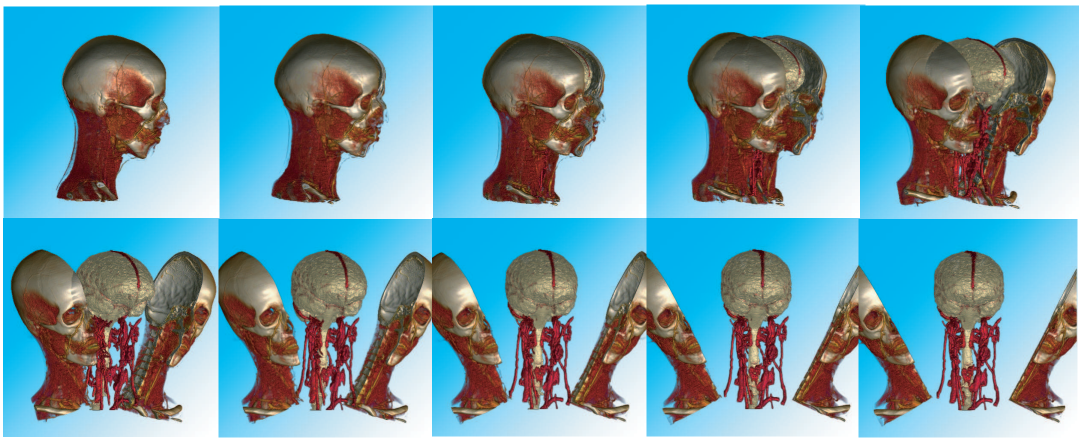

Figure 8: Stills of animation to illustrate advanced browsing of volumetric data based on V-Objects. Virtual dissection of the human skull, uncovering the nervous and vascular system.

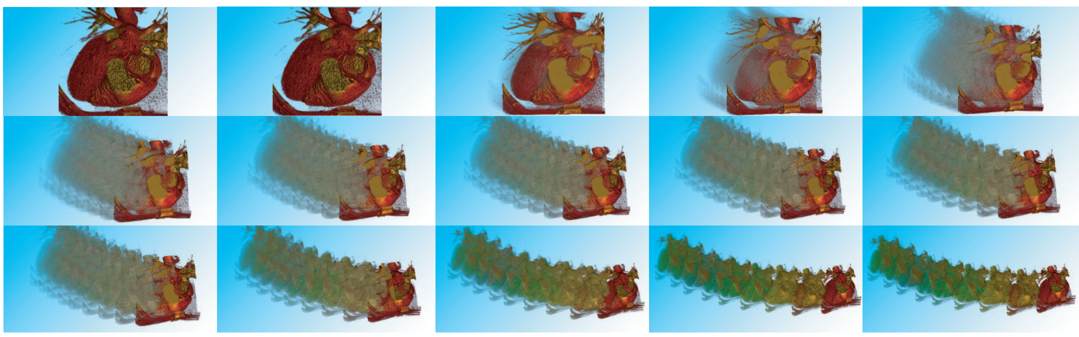

Figure 9: Stills of animation to illustrate 4D visualization based on V-Objects. Fanning in time allows to convey similarities and differences in the progress of time.