Adaptive script based animations for intervention planning

Konrad Muehler, Ragnar Bade, and Bernhard Preim

Department of Simulation and Graphics, Department of Computer Science,
Otto-von-Guericke University of Magdeburg, Germany
muehler|bade|preim@isg.cs.uni-magdeburg.de

Abstract. We describe scripting facilities to create medical animations for intervention planning based on medical volume data and derived segmentation information. A data independent scripting language has been developed to separate animation scripts from imaging data. The scripting facilities are adaptive and allow to reuse one script to create animations for many different patients. With expressive animations, we support the individual planning process, the preoperative documentation as well as discussions between medical doctors, for example in a tumor board. We also discuss the enhancement of interactive explorations with animations generated on the fly.

1 Introduction

Advances in imaging technology are continuously increasing the amount and the complexity of data and information available for intervention planning. To take advantage of these large amounts of data, the effective presentation and interaction with these data has to be considered. Currently, slice-based visualizations of medical volume data, such as CT or MRI, and derived segmentation information, are used. In very specific applications, standardized *static* three-dimensional visualizations are provided as a service [1]. Two major drawbacks of those visualizations are: (i) The difficulty to recognize exact spatial relations between the objects and (ii) the occlusion of relevant objects or parts thereof. These disadvantages can be compensated by interactive 3d-visualizations. However, navigating in 3d-scenes is time consuming and the viewer can lose sight of important objects or lose orientation [2]. Therefore, interactive exploration is rarely used in discussions between medical doctors, for example when a radiologist demonstrates a case to a surgical team.

Some of the problems of both static graphics and interactive scenes can be avoided by using carefully designed animations which effectively convey information in a limited and pre-defined period of time. The creation of expressive animations for intervention planning involves the specification of camera movements, movements of clipping planes, and the interpolation of transfer function parameters for volume rendering or parameters for surface rendering. This process is time-consuming, error-prone and requires deep and specialist skills. In intervention planning, appropriate visualizations and animations are required for many different patients. The therapeutic question, however, is often the same: for example the evaluation of resectability of a certain kind of tumor. Therefore, it is essential that animations designed for a particular case can be easily adapted or reused for another patient where the same therapeutic question is addressed. The reuse of animations is not only an issue of reducing effort but also essential to make the preoperative planning process more reproducible.

Besides the use of completely rendered animations the execution of animations during interactive exploration is conceivable. Depending on user interactions, such as selection of an object, animations can be executed which provide a scene overview or an object accentuation.

For an effective use of animations in intervention planning, we defined scripting facilities independent of a particular case, which enable an author to describe the behavior of objects in a simple and independent manner. To separate the description of animations from the individual patient data, our animation system is adaptive.

2 Related Work

The pioneering work of Zeltzer [3] introduced intent-based animation using a high-level scripting language and decomposition rules, to map higher level commands to lower level commands. This approach was used to decompose communication goals to animation instructions [4]. According to [5] animations must be slow and clear enough for viewers to perceive and understand changes. Therefore, emphasis techniques are a crucial aspect of animations [6].

For clinical purposes standardization is an important issue in the design of animations. Iserhardt-Bauer et al. [7] generated simple clinical videos efficiently using standard views and visualizations parameters. Tomandl et al. [8] evaluated the benefit of standardized high quality videos in clinical routine and state that those videos are a viable approach for standardized and reproducible visualizations. Facilities to improve interactive dataset traversal by animation of transfer functions for volume rendering were introduced by Correa et al. [9]. Krueger et al. [10], in cooperation with medical doctors, defined a standardized process for visualization of patient data in Ear-Nose-Throat-surgery (ENT-surgery). They identified default colors, texturing and viewpoints for important structures in the neck region. Preim et al. [11] introduced a scripting language for anatomy education. However, volume rendering and slice-based rendering was not considered.

Modern software libraries for scientific and medical visualizations provide facilities to create animations. In Amira [12] and Paraview [13], animations can be created for single data sets by defining events to be processed or by writing scripts in a TCL extending language. The scripts can not be applied to other data sets. Furthermore, the script instructions are very complex and difficult to handle. MeVisLab [14] provides the possibility to execute simple actions like rotations automatically. Also, a low-level scripting facility is available.

3 Script Concept

In this section, we discuss requirements for animations for clinical purposes and present a script-based concept. High-level instructions are mapped to executable low-level instructions according to decomposition rules. The selection of rules is based on an object classification (e.g. muscles, vessels, organs). Furthermore, we describe the parsing process and semantic relations to enhance animation. Finally, we present the integration of animations in the interactive exploration of a 3d scene.

3.1 Requirements

We target at *reusable* and *reproducible* animations for cases with the same therapeutic question. Thus our design is guided by four requirements:

1. **Adaptiveness:** The approximate location as well as the number of many anatomic structures, such as organs, is the same for virtually all patients. However, they differ in their specific shape and size. Thus the process of mapping high-level to low-level commands has to consider to these differences automatically. It also has to adapt to differences in pathologic structures which are unknown in size, location and shape.
2. **Convenience and specialisation:** Originating from different fields of application, the process of creating animations must strike a balance between convenience, speed and specialisation of animation.
3. **Animation techniques for intervention planning:** In addition to rotation, zoom or translation it must be possible to animate particular medical visualization techniques, such as volume rendering, clipping planes and slice-based visualizations.
4. **Standardization:** Animations for intervention planning should be standardized to afford comparisons and to facilitate the recognition of structures in different animations [7], [8], [10].

3.2 Basic Concept

We introduce a scripting language to describe animations of medical volume data and derived segmentation information. The design of our scripting language is dedicated to the requirements described above.

A script instruction consists of four parts: *time*, *object name*, *instruction name* and *parameters*. A script is executed on a visualization i.e. all instructions are applied to their associated objects over time. This can be a structure like an organ or a vessel but also the scene or the camera. The order of instructions in a script need not be the order of execution. Also an overlapping of time ranges is possible. This sample instruction moves a clipping plane through a scene from the left clipping all structures except the vessels (see also Fig. 1):

```
[0,8] 'All' clipPlane on left 0.8 except 'Vessels'
```

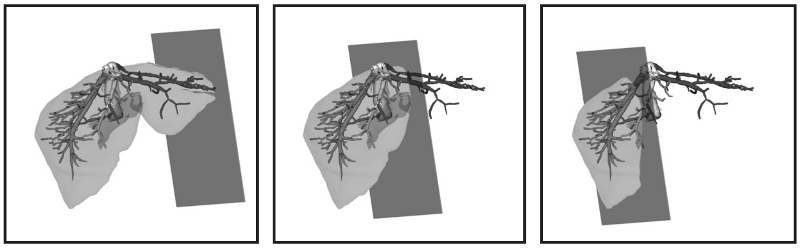

Fig. 1. Sample frames for clipping plane instruction above.

Adaptiveness. Instructions for camera placement are relative to the corresponding object and are geared to the size and location of its bounding box. Thus the animations adapt themselves to size and location of objects. This adaption is dedicated to compact structures, such as tumors, lung nodules and lymph nodes as well as to organs. For elongated or branching structures, the bounding box is not a good approximation.

Decomposition. To afford an enhancement of the scripting language for particular therapeutical questions, we use a hierarchical approach similar to [4]. In contrast to [4] we represent (therapeutic) questions by instructions and do not abstract parts of animations like scenes or shots. By using decomposition rules, script instructions can be abstracted. We distinguish between low-level (elementary) and high-level (abstract) instructions. Low-level instructions are e.g. `move`, `rotate` and `setColor`. High-level instructions are e.g. `emphasize`, `groupOverview` and `setAsFocusObject`. By abstracting instructions, we achieve a convenient way of describing whole animations with few instructions. To support fine-grained modifications and specialized specifications high-level and low-level instructions can be mixed.

High-level commands for medical animations. In addition to instructions for object property changes and camera changes, instructions with special regard to medical visualization techniques are provided. We created an instruction to move clipping planes (see section 3.2) as well as instructions for slice-based animations and simple animations of volume rendering.

For animations of slice-based 2d visualizations, the slices containing the corresponding object are identified automatically. The `showSlices` instruction has a parameter for margin slices above and below objects' slices. The following instruction shows an animation of slices containing a tumor with 5 slices margin:

```
[0,10] 'Tumor' showSlices 5
```

According to the requirements, we also provide facilities to create animations of volume rendered scenes. We do not incorporate strongly specialized facilities to control volume rendering parameters since we do not consider to use volume rendering as the only visualization mode. Instead, we assume that volume rendering is combined with surface rendering (of segmented objects). With the `volumeRendering` instruction, the transfer functions for gray values and transparency can be changed over time. For each transfer function, there are two parameters for *center* and *width*, to specify simple linear ramp functions. By replacing parameters with standard values they can be abstracted to parameters like *bones* or *soft tissue*:

```
[0] 'All' volumeRendering Bones
[0,5] 'All' volumeRendering softTissue
```

3.3 Script Parsing

Before its execution, a script is parsed, i.e. high-level instructions, composed objects and high-level parameters will be decomposed (Fig. 2). High-level instructions can be replaced by high-level as well as by low-level instructions. The substitution of instructions can be declared in general or depending on the anatomic type of the object the instruction refers to. Objects can be combined to composed objects so that instructions referring to a composed object are applied to all elementary objects. Parameters can be replaced by values, e.g. `red` by its rgb-value `255,0,0`. The decomposition rules are declared in XML and can be easily edited.

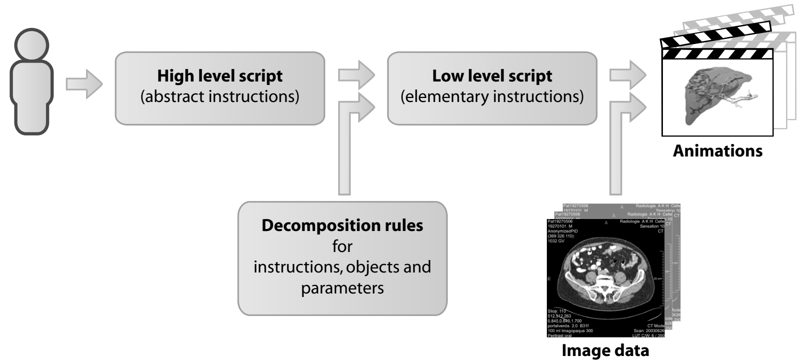

Fig. 2. Concept of transfering a high-level script to a low-level script by decomposing instructions and composed objects, and by replacing parameters. The low-level script is executed on visualization of data sets to create an animation.

3.4 Interactive animations

Besides the use of animations as videos, animations can be combined with interactive manipulation. For the individual process of intervention planning, it is essential to provide the freedom of exploring the scene. Animations may support the exploration, e.g. by presenting an overview, an introduction of the scene or an automatic camera areal view. We consider the following scenarios for such interactive animations:

1. **Emphasis:** By selecting an object, e.g. by picking it or selecting it from a list, an animation emphasizes this object by moving the camera to a "good" viewpoint. The style of emphasis depends on the (anatomic) type of selected object.
2. **Object overview:** An overview of a selected object can animated instead of its emphasis. This can be a rotation around the object or a serial emphasis of some near focus objects.
3. **Scene overview:** To provide a re-orientation of the current position in the scene, a short animation provides a camera pan to a global viewpoint and return back to the starting position.
4. **Scene introduction:** To introduce an interactive exploration, an animation presents the scene with a rotation and an emphasis of the most important objects.
5. **Fade in and out:** If the viewer explores a selected object from different viewpoints occluding objects can be smoothly faded out when they appear in the field of view and faded in when they no longer hide the selected object.

We integrated the scripting facilities in an interactive planning system to enrich it with interactive animations.

4 Examples

Our concepts for generating medical animations are based on the analysis of numerous medical animations and fruitful discussions with medical professionals. Our scripting facilities are included in NeckVision [15], a software to segment image data and create and present visualizations for ENT-intervention-planning, and are used in the intervention planning for liver surgery. They are also a part of the LiverSurgeryTrainer [16], an educational system for liver surgeons. In the following we present two examples.

4.1 Planing a neck dissection

In ENT-surgery, the spatial relations in the vicinity of enlarged lymph nodes are crucial. Navigating in a scene, the surgeon can interactively explore the lymph nodes. Our animations provide amongst others a support on finding and analyzing large lymph nodes. In the example the scene is introduced and all lymph nodes are emphasized according to their size while other structures like bones and muscles are toned down to give a little context information (Fig. 3):

```
[1,7] 'Scene' introduce         // rotating the scene 360 degrees
[8,10] 'Lymphnodes' showBySize  // show lymph nodes colored by size
```

If the surgeon picks a lymph node to explore its anatomic context, this high-level script instruction will be executed:

```
[0,6] 'SELECTED OBJECT' emphasizeInContext
```

As a consequence, the structures in the near surrounding are set to nearly opaque while all other structures are set transparent. So the surgeon can analyze possible infiltrations in the muscles or vessels.

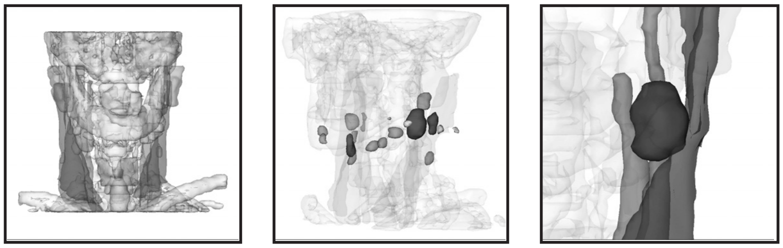

Fig. 3. Sample frames from an interactive animation of a scene introduction, an accentuation of all lymph nodes and the emphasis of a selected lymph node in its anatomic context.

4.2 Liver tumor resection

Before resecting a liver tumor, a surgeon has to plan where to dissect. The distance of the tumor to its surrounding vessels is an essential information in this process. With the following script, the camera is rotated around the tumor to give an overview and the camera is moved to show the minimal distance between tumor and vessels afterward (Fig. 4):

```
[0,5] 'Scene' sceneIntroduction
[6,9] 'Tumor' viewDistance 'Vessels'
```

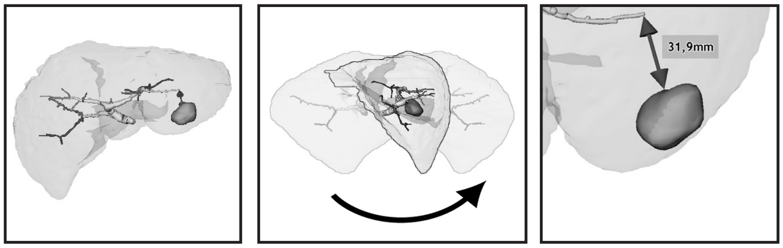

Fig. 4. Sample frames from an animation to explore a tumor and its distance to surrounding vessels. The arrow is included to clarify the rotation.

Because the scripts are reusable and adaptive the script above is used for many datasets with the medical question of minimal distance around a tumor. The animation can be played as a pre-rendered video in a tumor board discussion as well as an introduction in an individual interactive exploration process by the surgeon.

5 Conclusion and future work

We introduced scripting facilities which enable authors to create animations for therapeutic questions. We developed a data independent scripting language to separate the description of animations from data sets. Because our scripting facilities are adaptive, a script can be reused for many different but similar cases. The script-based approach also allows to enhance interactive animations. The scripted animations are used to enhance important interactions with animations directly in the scene. The scripting concept can be easily extended to other therapy questions. We intend to incorporate functions to identify ”good” viewpoints for objects automatically and by providing camera path planning facilities. This work was supported by the BMBF (SOMIT-FUSION project FKZ 01—BE 03B).

References

1. MeVis-Distant-Services-AG: http://www.mevis-distant-services.com/ (2005)
2. Bade, R., Ritter, F., Preim, B.: Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3d Rotation. In: Smart Graphics. (2005) 138 – 150
3. Zeltzer, D.: Task-Level Graphical Simulation: Abstraction, Representation, and Control. In: Making Them Move: Mechanics, Control, and Animation of Articulated Figures. Morgan Kaufmann, San Mateo, CA (1990) 3–33
4. Karp, P., Feiner, S.: Automated presentation planning of animation using task decomposition with heuristic reasoning. In: Graphics Interface. (1993) 118–127
5. Tversky, B., Morrison, J., Betrancourt, M.: Animation: can it facilitate? International Journal of Human Computer Studies **57** (2002) 247–262
6. Preim, B., Tietjen, C., Doerge, C.: NPR, Focussing and Emphasis in Medical Visualizations. In: Simulation und Visualisierung. (2005) 139–152
7. Iserhardt-Bauer, S., Rezk-Salama, C., et al.: Automated 3d Video Documentation for the Analysis of Medical Data. In: BVM2001. (2001) 409–413
8. Tomandl, B.F., Hastreiter, P., Iserhardt-Bauer, S., et al.: Standardized Evaluation of CT Angiography with Remote Generation of 3d Video Sequences for the Detection of Intracranial Aneurysms. RadioGraphics **23** (2003) 12e
9. Correa, C.D., Silver, D.: Dataset Traversal with Motion-Controlled Transfer Functions. In: Proceedings of IEEE Visualization 2005, Minneapolis (2005) 359–366
10. Krueger, A., Tietjen, C., et al.: Interactive Visualization for Neck Dissection Planning. In: IEEE/Eurographics Symposium on Visualization. (2005) 295–302
11. Preim, B., Ritter, A., Strothotte, T.: Illustrating Anatomic Models: A Semi-Interactive Approach. In: 4th International Conference on Visualisation in Biomedical Computing. (1996) 23–32
12. Stalling, D., Westerhoff, M., Hege, H.C.: Amira: A Highly Interactive System for Visual Data Analysis. In: The Visualization Handbook. Elsevier (2005) 749–767
13. Law, C.C., et al.: An application architecture for large data visualization. In: IEEE symposium on parallel and large-data visualization and graphics. (2001)
14. Hahn, H., Link, F., Peitgen, H.: Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization. In: SimVis2003. (2003) 283–298
15. Cordes, J., Dornheim, J., Preim, B., et al.: Preoperative Segmentation of Neck CT Datasets for the Planning of Neck Dissections. In: SPIE Medical Imaging. (2006)
16. Bade, R., Riedel, I., Schmidt, L., et al.: Combining Training and Computer-assisted Planning of Oncologic Liver Surgery. In: Bildverarbeitung fuer die Medizin. (2006)