

Skriptbasierte Animationen für die Operationsplanung und Ausbildung

Konrad Mühler¹, Ragnar Bade¹ und Bernhard Preim¹

¹Institut für Simulation und Graphik, Fakultät für Informatik,
Otto-von-Guericke Universität Magdeburg, 39106 Magdeburg
Email: konrad@isg.cs.uni-magdeburg.de

Zusammenfassung. Ziel dieses Beitrages ist die Generierung von Animationen aus patientenindividuellen, segmentierten CT- und MRT-Datensätzen. Die Animationsgenerierung erfolgt auf der Basis von Skripten. Dafür wurde eine Skriptsprache entwickelt, die die Anwendung eines einzigen Skriptes auf mehrere unterschiedliche Datensätze ermöglicht. Zusätzlich erlaubt die Skriptsprache die Definition von Ersetzungsvorschriften. Dies gestattet die Abstraktion von Anweisungen hin zu allgemeinen Beschreibungen von Animationsabläufen, die auch weniger versierte Nutzer erstellen können. Die Animationen werden sowohl in der Operationsplanung als auch in der medizinischen Ausbildung genutzt.

1 Einleitung

Im klinischen Alltag etabliert sich zur Operationsplanung bei komplexen Fragestellungen, z.B. in der Tumorchirurgie, zunehmend die Nutzung von interaktiven 3d-Visualisierungen auf der Basis segmentierter CT- und MRT-Daten. Diese geben dem Betrachter einen besseren Eindruck von den räumlichen Zusammenhängen der Strukturen als dies bei zweidimensionalen Ansichten der Fall ist. Die interaktive Navigation (Zoom, Rotation) durch den Betrachter einer Darstellung ist jedoch sehr zeitaufwendig. Auch können dabei wichtige Strukturen oder Bereiche übersehen werden. Ebenfalls ist die individuelle Anpassung der Visualisierungsparameter der dargestellten Strukturen nur mit großem Interaktionsaufwand zu bewerkstelligen. Der zeitliche Aufwand macht sich im klinischen Alltag beispielsweise bei interdisziplinären Tumorboardbesprechungen negativ bemerkbar, weil relativ viele Therapieentscheidungen in kurzer Zeit getroffen werden müssen.

Vorbereitete Animationen, die von einem Autor sorgfältig erstellt werden, können dagegen effektiv die relevanten Strukturen darstellen und auf Besonderheiten aufmerksam machen. Allerdings ist die Erstellung solcher Animationen für den Autor sehr aufwändig. Daher war es unser Ziel, ein System zur Generierung von Animationen aus patientenindividuellen Daten zu entwickeln. Dabei soll die Beschreibung einer einzigen Animation auf verschiedene Datensätze anwendbar sein. So können mit einer Beschreibung für eine Fragestellung reproduzierbare Animationen aus ähnlichen Datensätzen generiert werden. Die Beschreibung der

Animationen soll sowohl für weniger versierte Nutzer einfach und schnell möglich sein, wie auch erfahrenen Benutzern genügend Freiheiten lassen, Animationen nach sehr speziellen Anforderungen zu generieren.

Grundlage der Visualisierung der Animationen sind segmentierte Strukturen aus CT- und MRT-Schichtdatensätze, die als Oberflächenvisualisierungen vorliegen. Diese Strukturen können mit unterschiedlichen Visualisierungsparametern wie Farbe und Transparenz dargestellt werden.

2 Stand der Forschung

Frühe Systeme der automatischen Animationsgenerierung wie BETTY [1] und ESPLANADE [2] basieren auf der hierarchischen Aufschlüsselung von abstrakten Kommunikationszielen hin zu elementaren Animationsanweisungen. Ein Nachteil ist hier die fehlende Flexibilität im Bezug auf kleine Änderungen an den Animationen. So müssen entweder komplette Kommunikationsziele neu definiert und beschrieben werden oder bestehende Kommunikationsziele global modifiziert werden.

Skriptsprachen bieten eine größere Flexibilität. Sie haben unter anderem in Multimedia-Entwicklungswerkzeugen eine weite Verbreitung erfahren. Dabei werden Skriptsprachen für ein weitaus größeres Aufgabenspektrum genutzt, als zur reinen Animationserzeugung. Der größere Funktionsumfang bringt allerdings einen wesentlich komplexeren Aufbau und eine umfangreichere Syntax hervor, als für die Erzeugung von Animationen benötigt wird.

In aktuellen, auf medizinische Visualisierung spezialisierten Softwaresystemen (z.B. MEVISLAB, AMIRA) sind einfachste Möglichkeiten der Animationserzeugung auf medizinischen Volumendaten vorhanden. Diese beschränken sich aber meist auf einfache Rotationen der Szene. In MEVISLAB können auch die Interaktionen eines Benutzers in einer interaktiven Darstellung als Video mitgeschnitten werden. Diese Methode ist aber sehr ungenau und nicht reproduzierbar.

Ein Ansatz zur Animationsbeschreibung im Bereich der Anatomieausbildung findet sich im ZOOMILLUSTRATOR [3]. Ziel ist es dort, in einer interaktiven Darstellung dem Nutzer einen optimalen Standpunkt zur Sicht auf die Objekte im Fokus zu gewährleisten und die Kamera nötigenfalls animiert an eine solche Position zu bewegen. Zusätzlich wird der Verlauf von Objekten durch Zeigergeräte veranschaulicht. Die dort verwendete Skriptsprache ist allerdings schwer zu handhaben. Die Grundlage des Systems bilden wenige ausgesuchte Polygonmodelle idealisierter anatomischer Strukturen, womit das Konzept für den massenhaften Einsatz mit patientenindividuellen Daten nicht geeignet ist.

3 Methoden

Bei unserem Verfahren zur Animationserzeugung nutzen wir einen skriptbasierten Ansatz. In einem Skript sind in Form von einzelnen Anweisungen die Verhaltensweisen der Objekte (anatomischer Strukturen) und der Szene einer Darstellung beschrieben. Die Anweisungen sind objektbezogen und damit daten-

satzunabhängig. Erst durch die Anwendung eines Skriptes auf einen Datensatz bzw. dessen dreidimensionale Darstellung wird die konkrete Animation generiert. Dies ermöglicht die Anwendung sowohl für die Therapieplanung wie für die vergleichende Anatomie in der Lehre.

Eine einzelne Skriptanweisung gliedert sich in vier Bereiche: eine Zeitangabe, eine Objektbezeichnung, den Anweisungsnamen und Parameter (Listing 1). Eine Veränderung, die die Anweisung in der Darstellung hervorruft, wird über den Zeitbereich interpoliert.

```
Time Object Command [Parameter]
[0,10] 'Tumor' emphasize
```

Listing 1. Aufbau einer Skriptanweisung sowie eine Beispielanweisung zur Hervorherung des Tumor-Objektes.

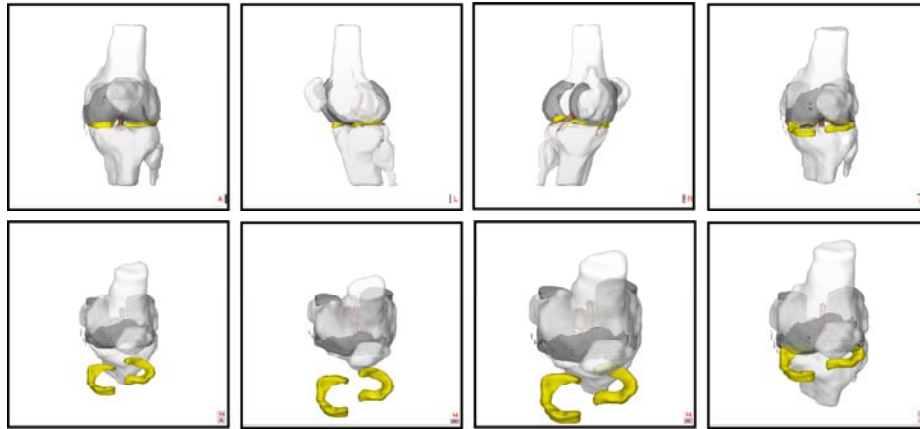
Damit die Skriptsprache auch von Nutzern ohne Programmierkenntnisse verwendet werden kann, haben wir aufbauend auf der Idee von KARP&FEINER [2] einen hierarchischen Ansatz mit unterschiedlichen Abstraktionsstufen entwickelt. Die unterste Stufe wird dabei von `LowLevel`-Anweisungen gebildet. Sie erlauben eine präzise Beschreibung der Animation und sind in ihrem Umfang und ihrem Aufbau komplex. Verschiedene `LowLevel`-Anweisungen können zu `HighLevel`-Anweisungen zusammengefasst werden. Dazu werden Ersetzungsvorschriften definiert. Eine `HighLevel`-Anweisung wird mit Hilfe dieser Ersetzungsvorschriften durch eine oder mehrere Anweisungen ersetzt. Hier ist, wie auch im Skript selbst, eine beliebige Mischung von `HighLevel`- und `LowLevel`-Anweisungen möglich. Auch können Parameter auf der `HighLevel`-Ebene vereinfacht werden (z.B. `red` anstelle des genauen RGB-Farbwertes). Eine weitere Ersetzung findet bei den Objektbezeichnungen statt. Hier können Objekte zu Objektgruppen beliebig kombiniert werden. Eine Anweisung, die sich auf das Objekt `Lung` bezieht, hat so Auswirkungen sowohl auf das Objekt `LungLeft` wie auch `LungRight`. Der Anweisungsumfang der Skriptsprache kann so einfach erweitert und an die jeweiligen Erfordernisse des Autors angepasst werden.

Eine Umwandlung der Objektbezeichnungen und Parameter in exakte Koordinaten erfolgt erst direkt bei der Ausführung des Skriptes. Als Bezugspunkt dient das Zentrum und die Ausdehnung der jeweiligen *bounding box* der Objekte. Dies ermöglicht die Anwendung eines einzigen Skriptes auf viele Datensätze.

```
[0] 'System' init
[0,10] 'All' overview
[10,14] 'Meniscus' explodeOn
[27,33] 'Meniscus' explodeOff
```

Listing 2. Beispielskript zur Animation eines Knies.

Abb. 1. Einzelbilder einer Animation, die mit dem Skript in Listing 2 erzeugt wurde.



4 Beispiel

Mit Hilfe des Listings 2 wurde die 3d-Darstellung eines segmentierten Knies animiert (siehe Abb. 1). Es wird zuerst ein Überblick über die Darstellung gegeben und anschließend der Meniscus näher betrachtet. Hierbei sollten mehr verschiedene Knie-Datensätze animiert werden um die pathologischen Unterschiede der Menisken zu veranschaulichen.

5 Ergebnisse

Durch den von uns entwickelten Ansatz einer skriptbasierten Generierung von Animationen ist es möglich geworden, effizient und reproduzierbar Animationen aus segmentierten Schichtbilddaten zu generieren. Der Schwerpunkt lag hierbei auf Animationen für die Operations- und Therapieplanung. Das entwickelte Animationssystem ist adaptiv, das heißt, es können eine einzige Beschreibung einer Animation auf mehrere patientenindividuelle Datensätze angewendet und so gleichartige Animationen ohne Mehraufwand generiert werden. Die Skriptsprache ist abstrahierbar, so dass auch weniger erfahrene Autoren mit geringem Lernaufwand Animationsbeschreibungen entwickeln können.

Die Skriptsprache wurde als Teil der Software zur Analyse und Visualisierung medizinischer Daten MEVISLAB implementiert. Dadurch konnten wir das Animationssystem erfolgreich zur Generierung von Animationen aus einer Vielzahl verschiedenster patientenindividuellen Datensätze (Abdominal-CT, MRTs von Knie- und Schultergelenken, HNO-Datensätze) nutzen. Diskussionen mit Ärzten zeigen, dass solche Animationen effizient die relevanten Informationen vermitteln können. Diese Animationen stellen eine gute Ergänzung zu statischen 3d-Darstellungen und eine Alternative zu interaktiven Darstellungen in

der Operationsplanung dar. Derzeit findet das System zusätzlich Anwendung in einem Lernsystem zur fallbasierten chirurgischen Ausbildung [4].

Die Reproduzierbarkeit der Animationen lässt sich zusätzlich dazu nutzen, statische Bilder mit genau definierten Parametern zu erzeugen. Dabei kann durch eine Reihe von Anweisungen eine visuelle Parametrisierung (z.B. bestimmte Farben) aller Objekte sowie eine immer gleiche Sicht auf die Darstellung für unterschiedliche Datensätze erreicht werden.

6 Diskussion

Der Anweisungsumfang der Skriptsprache ist erweiterbar: Zum Einen durch die Definition neuer `HighLevel`-Anweisung auf der Basis schon bestehender Anweisungen. Zum Anderen durch die Implementierung neuer `LowLevel`-Anweisungen zur Manipulation weiterer Parameter der Darstellungen. Dies bietet ein großes Potential sowohl für die Darstellung von speziellen Hervorhebungstechniken wie auch der Animation einer Vielzahl medizinischer Fragestellungen.

Derzeit wird ein Authoringsystem entwickelt, mit dem Autoren ohne Kenntnis der Skriptsprache Animationsbeschreibungen direkt manipulativ generieren und testen können. Mit der Fertigstellung eines solchen Authoringsystems ist dann eine klinische Evaluierung des gesamten Systems möglich.

Ein weiterer Ansatz für Erweiterungen des Konzeptes liegt in der Nutzung von Animationen in interaktiven Darstellungen. Solche interaktiven Animationen können genutzt werden, um beispielsweise bei der Selektion eines Objektes in einer 3d-Visualisierung die Kamera automatisch an eine günstige Position mit guter Sicht auf das Objekt zu bewegen. Auch könnten Wechsel von Hervorhebungen einzelner Objekte in einer interaktiven Darstellung mit Hilfe skriptbasierter Animationen visualisiert werden.

Danksagung: Wir danken Jeanette Cordes für die aufwändige Handarbeit der vielen Segmentierungen. Bei Dr. Stephan Sievers bedanken wir uns für die Bereitstellung der gezeigten MRT-Daten. Dieses Projekt wurde im Zusammenhang mit dem SOMIT-FUSION-Projekt durch das BMBF gefördert (FKZ 01-BE03B).

Literaturverzeichnis

1. Butz A. Betty: Planning and Generating Animations for the Visualization of Movements and Spatial Relations. In: *Advanced Visual Interfaces*. ACM Press; 1994.
2. Karp P, Feiner S. Automated presentation planning of animation using task decomposition with heuristic reasoning. In: *Graphics Interface*; 1993. p. 118–127.
3. Preim B, Ritter A, Steinicke G. Gestaltung von Animationen zur Erklärung komplexer 3D-Modelle. In: *Simulation und Animation für Planung, Bildung und Präsentation*. Magdeburg; 1996. p. 255–266.
4. Bade R, Mirschel S, Oldhafer KJ, Preim B. Ein fallbasiertes Lernsystem für die Behandlung von Lebertumoren. *Procs BVM*. Springer; 2004. p. 438–442.