

Ein Simulator für Schädelbohrungen bei stereotaktischen Eingriffen

Eine Java 3D-Anwendung

Kai Annacker^{*#}, Hans-Gerd Lipinski^{*}, Dietrich H. W. Grönemeyer[#]

^{*}Fachhochschule Dortmund
Fachbereich Informatik
Abteilung für Medizinische Informatik
Sonnenstraße 96, D-44139 Dortmund

[#]Universität Witten/Herdecke
Lehrstuhl für Radiologie und Mikrotherapie
Institut für Mikrotherapie
Universitätsstraße 142, D-44799 Bochum

1 Einleitung

Zur symptomatischen Behandlung einer zentralmotorischen Störungen wie Tremor und Dystonie dient häufig ein stereotaktischer Eingriff. Um aber Zugang zum therapeutisch relevanten Hirnareal erlangen zu können, muß zuvor ein Teil der Schädelkalotte entfernt werden.

Hierzu wird ein kreisförmiger Bereich des Schädelknochens ausgesägt, der nach Abschluß des diagnostischen oder therapeutischen Eingriffs wieder replantiert werden kann (osteoplastische Trepanation). Durch das entstandene Loch in der Kalotte kann dann entlang eines - anhand prä-operativ erzeugter Computertomogramme definierten - Trajekts z.B. eine Mikroelektrode zur exakten Zielpunktbestimmung oder eine Thermokoagulationssonde zur therapeutischen Zerstörung eines kleinen, für die Symptomatik verantwortlichen Gehirnareals geführt werden.

Zur präzisen Führung des Operationswerkzeugs wird ein Rahmen am Kopf des Patienten befestigt. Dieser Führungsrahmen ermöglicht dann die Einstellung des gewünschten Trajekts und die Bewegung der Sonde bzw. Elektrode exakt entlang dieses Trajekts.

In der Entwicklung befindet sich ein Neuronavigations-Simulator, der die notwendigen Schritte von der Definition des Trajekts bis zum eigentlichen therapeutischen Eingriff abdeckt. In einem ersten Projekt wurde die interaktive Trajektbestimmung und die daraus resultierende virtuelle Schädeltrepanation implementiert, die sich ebenfalls interaktiv beeinflussen läßt. Aber auch die Navigation von Operationswerkzeugen innerhalb des trepanierten Schädels, sowie zusätzliche Orientierungshilfen in Form von Hirnatlanten und CT- bzw. MRT-Bildern wurden bereits umgesetzt.

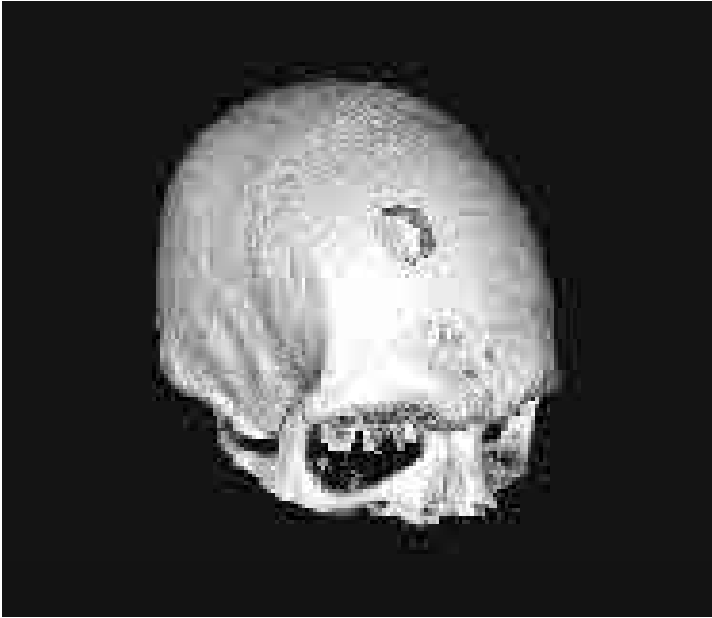


Abb. 1: Schädel während der Trepanations-Simulation.

2 Methoden und Ergebnisse

2.1 Vorbereitungen

Entscheidend für den therapeutischen Erfolg ist eine exakte Trajektdefinition, also in erster Linie eine optimale Bestimmung des Zielpunktes der Sonde bzw. Elektrode innerhalb des Diencephalons und daraus resultierend der Weg, der das Operationswerkzeug zu diesem Zielpunkt führt [LS01]. Nachdem das Trajekt mit Hilfe prä-operativ angefertigter Computertomogramme festgelegt wurde, ist gleichzeitig auch das notwendige Bohrloch definiert.

Zuerst wird aus den CT-Bildern der Patienten-Schädel segmentiert und u. a. mit Hilfe des Marching-Cube-Algorithmus [Bo97a, Bo97b] dreidimensional rekonstruiert. Interaktiv wird dann das Trajekt für die therapeutische Maßnahme festgelegt, aus dem sich aber auch der Bohrweg für die anstehende Trepanation ergibt.

Da nun der Bohrweg (und auch der Bohrdurchmesser) feststehen, läßt sich der virtuelle Schädel in zwei Teile zerlegen:

In einen Teil, der innerhalb des Bohrvolumens liegt und daher im Zuge der Trepanations-Simulation entfernt wird, und in den Restschädel, der außerhalb des Bohrvolumens liegt und deshalb während der virtuellen Operation unverändert bleibt. Der Vorteil dieser Unterscheidung liegt darin, daß der größte Teil des Schädels, nämlich der statische Teil, nur einmal zu Beginn der Simulation dreidimensional rekonstruiert, dann aber nicht mehr nachberechnet werden muß. Nur der kleine dynamische Anteil des Schädels, der virtuell entfernt wird, muß synchron zur Bohreravigation rekalkuliert werden. Dies bedeutet eine erhebliche Ersparnis an Rechenleistung, wodurch sich eine performantere Navigations-Simulation realisieren läßt.

2.2 Trepanations-Simulation

Ist der statische Teil des Patienten-Schädels dreidimensional rekonstruiert, beginnt die eigentliche Simulation der Bohrung.

Der gewünschte Vortrieb des Bohrwerkzeuges wird interaktiv mit Hilfe eines Eingabegeräts, beispielsweise eines Force-Feedback-Joysticks, gesteuert.

Dieser ermöglicht es zudem, daß dem virtuellen Operateur eine direkte haptische Rückkopplung über den Bohrerstatus (Drehzahl, Widerstand) geliefert wird. Dazu sind in einem Force-Feedback-Joystick Motoren eingebaut, die z. B. Vibrationen erzeugen oder Kraft gegen beabsichtigte Lenkbewegungen ausüben können. Geeignet angesteuert entsteht so eine realitätsnahe Simulation des Bohrvorgangs.

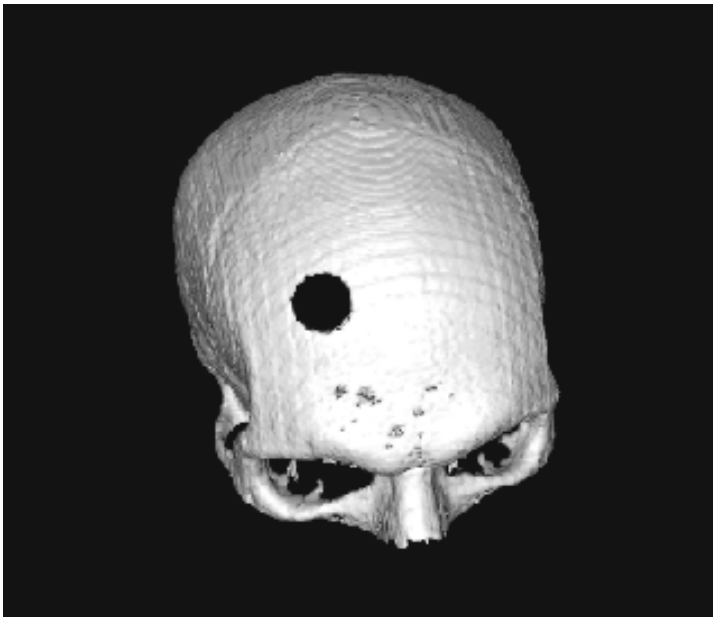


Abb. 2: Schädel nach der Trepanationssimulation

Da der während der Simulation zu entfernende Teil des Schädelknochens bereits bekannt ist, kann nun, abgestimmt auf die Joystick-Steuerung, die Trepanation visualisiert werden (Abbildungen 1 und 2). Gleichzeitig wird der Bohrwiderstand anhand der noch nicht entfernten Strukturen ermittelt und an die Steuereinheit zurückgemeldet.

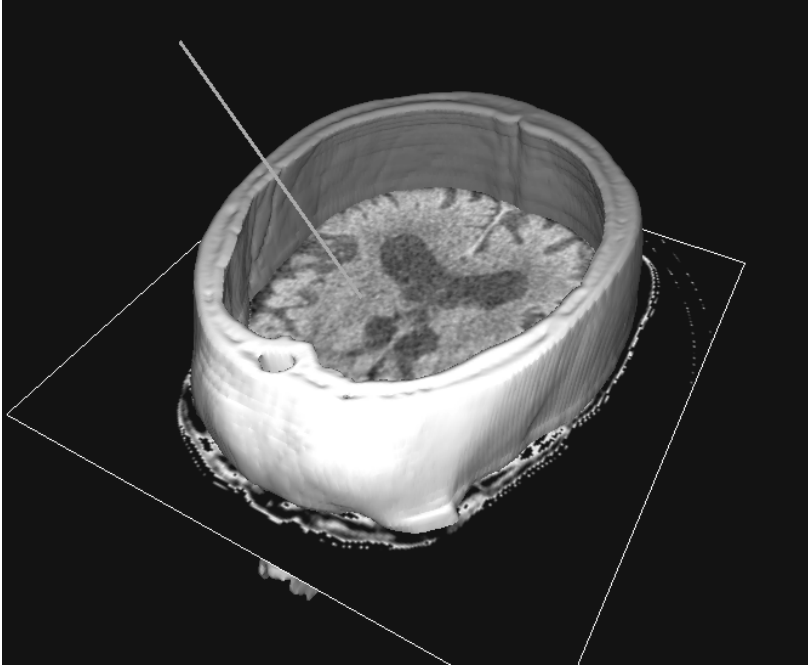


Abb. 3: Simulation eines stereotaktischen Eingriffs.

2.3 Stereotaktischer Eingriff

Nachdem die Schädelkalotte trepaniert wurde, kann entlang des Trajekts durch das Bohrloch eine Sonde geführt werden. Zur präzisen Bestimmung des therapeutisch relevanten Zielgebiets dienen dabei neben den CT-Bildern (Abbildung 3) auch MRT-Bilder und Hirnatlanten (Abbildung 4). Darüberhinaus werden Stimulationselektroden eingesetzt um die Korrektheit des Zielgebiets zu überprüfen.

Ist das für die Symptomatik verantwortliche Zielgebiet bestimmt, wird eine Thermokoagulations-Sonde eingebracht, die exakt im Zielareal neuronales Gewebe durch Hitze zerstört und damit den Tremor bzw. die Dystonie therapiert. Auch dieser Vorgang kann simuliert und abschließend mit Original-Tomogrammen verglichen werden.

2.4 Soft- und Hardware

Zur Implementation des Simulators wurde die system- und plattformunabhängige Programmiersprache JAVA 2 (Standard Edition Version 1.3) der Firma Sun Microsystems eingesetzt [Ja01a, Ja01b]. Neben der Möglichkeit, den Simulator auf beliebigen Computersystem einsetzen zu können (nur eine *Java Virtual Machine*, die Java-Laufzeitumgebung, muß installiert sein), hat Java als streng objektorientierte Sprache den Vorteil, daß der Code gut verständlich und leicht für ähnliche Problemstellungen wiederverwendbar ist.

Zusätzlich wurde JAVA 3D benutzt, um Zugriff auf die Fähigkeiten der eingesetzten 3D-Hardware zu erlangen [Ja01c]. Die Java 3D-Technologie kapselt auf objektorientierte Weise Funktionalität, die in anderen Programmiersprachen nur mit Hindernissen via OpenGL oder Direct3D erreichbar ist, und ermöglicht so die komplexe dreidimensionale Visualisierung [Op01, Di01].

Trotz geringfügiger Performance-Nachteile der Programmiersprache Java gegenüber nativ kompilierenden Dialekten wie C oder Pascal, konnte gerade auch im Bereich der zeitkritischen 3D-Darstellung, auch dank der eingesetzten Hardware (800MHz-Dual-Prozessor-System inklusive Hochleistungs-3D-Graphikkarte auf Intel-/Windows-Basis), ein schnell reagierender Simulator umgesetzt werden.

Da Java als systemunabhängige Sprache keine direkte Zugriffsmöglichkeit auf spezielle Eingabegeräte (z. B. Joysticks) bietet, mußte die Joystick-Steuerung in ein C-Programm verlagert werden. Unter Nutzung des DirectInput-API unter Microsoft Windows können so der Status des Eingabegeräts registriert und die Force-Feedback-Motoren angesteuert werden [Di01].



Abb. 4 Darstellung einer Sonde im Hirnatlas von Schaltenbrand und Wahren

Das Problem, zwischen Steuer-Programm und Simulator Informationen austauschen zu müssen, wurde derart gelöst, daß beide Programme in die Lage versetzt wurden, über ein

lokales Netzwerk (LAN) miteinander zu kommunizieren. Per *User Datagram Protocol* (UDP) [Ud01] sendet das Steuer-Programm auf der einen Seite den Status des Eingabegeräts und empfängt gleichzeitig Anforderungen des Simulations-Programms, Vibrationen und Widerstände an den virtuellen Operateur zurückzumelden. Auf der anderen Seite empfängt so der Simulator die Joystick-Steuerung, errechnet daraus eine aktualisierte Visualisierung und sendet zeitgleich die erforderliche haptische Rückmeldung an das Steuerungs-Programm (Abbildung 5).

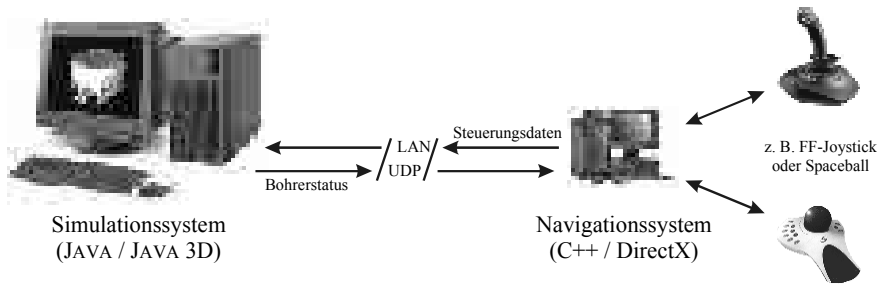


Abb. 5: Schema der Funktionsweise des Operations-Simulators.

Durch die lose Kopplung der beiden Programme via UDP ergibt sich zusätzlich die Möglichkeit, beide Programmteile auf verschiedenen Computersystemen betreiben zu können und damit das Simulations-System von den Steuerungs-Aufgaben zu entlasten.

3 Diskussion

Die virtuelle Schädelbohrung (Trepanation) als Teil einer stereotaktischen Operationssimulation wurde implementiert.

Anhand von Patienten-CTs wird ein Trajekt für Sonden definiert, das gleichzeitig den Bohrweg festlegt. Interaktiv kann die Bohrung vom virtuellen Operateur navigiert und visuell beurteilt werden. Ist der Schädel eröffnet, können Sonden eingebracht und diagnostische bzw. therapeutische Maßnahmen durchgeführt werden, die sich ebenfalls innerhalb der Simulationsumgebung darstellen lassen. Unterstützt wird die Navigation durch die Darstellung von Patienten-CTs bzw. -MRTs und Hirnatlanten.

Die Entkopplung von Steuerungs- und Simulations-Einheit ermöglicht zudem den Einsatz spezieller Eingabegeräte, auch solcher mit haptischer Rückkopplung, ohne dabei den Vorteil des in Java implementierten Simulators, auf beliebigen Computersystemen einsetzbar zu sein, einzubüßen.

Beim Einsatz des Simulations-Systems auf weniger performanten Systemen als dem von uns eingesetzten sind beim derzeitigen Entwicklungsstand allerdings noch Probleme mit der Visualisierung des Bohrvorganges in Echtzeit zu beobachten. Gerade aber beim derzeitigen Fortschritt im Bereich der 3D-Hardware-Beschleunigung ist zu erwarten, daß

diese Probleme schon in einer der nächsten Rechnergenerationen nicht mehr zu vorhanden sein werden.

4 Danksagung

Diese Arbeit wurde mit Mitteln des Ministeriums für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen (Assistentenprogramm NRW) und der Fachhochschule Dortmund (Rektorat II / Forschung und Entwicklung) gefördert.

Literaturverzeichnis

- [Bo97a] Bourke,~Paul: Polygonizing a scalar field, 1997, <http://www.swin.edu.au/astronomy/pbourke/modelling/polygonise/>
- [Bo97b] Bourke, Paul: Surface (polygonal) simplification, 1997, <http://www.swin.edu.au/astronomy/pbourke/modelling/surfsimp/>
- [Di01] Microsoft Corporation: DirectX 8.0, <http://www.microsoft.com/directx/>
- [Ja01a] Sun Microsystems Inc.: The JAVA Technology, <http://java.sun.com/>
- [Ja01b] Sun Microsystems Inc.: The JAVA 2 Standard Edition 1.3, <http://java.sun.com/j2se/1.3/>
- [Ja01c] Sun Microsystems Inc.: The JAVA 3D API, <http://java.sun.com/products/java-media/3D/>
- [LS01] Lipinski, H.-G.; Struppler, A.: Wissensbasierte funktionelle Neurochirurgie, in: Informatik, Biometrie und Epidemiologie in Medizin und Biologie, **24** (2), 50-56.
- [Op01] Silicon Graphics Inc.: OpenGL - The Industry's Foundation for High Performance Graphics, <http://www.opengl.org/>
- [Ud01] Internet RFC/STD/FYI/BCP Archives: Request For Comments 768, The User Datagram Protocol (UDP), <http://www.faqs.org/rfcs/rfc768.html>