5

Anwendungen

„In theory, there is no difference between theory and practice. But, in practice, there is. “

—Jan L. A. van de Snepscheut [1]

Auf der Grundlage, der im Rahmen dieser Arbeit entwickelten und in Kapitel 4 beschriebenen Simulationsbibliothek wurden vier Anwendungen entwickelt.

Abschnitt 5.1 beschreibt ein Trainingsprogramm, mit dem die Ablösung der innersten Schicht der Retina virtuell trainiert werden kann. Eine virtuelle Capsulorhexis-Prozedur wird in Abschnitt 5.2 vorgestellt. Abschnitt 5.3 präsentiert eine Anwendung, mit der die Entfernung eines Darmpolypen trainiert werden kann. Die Simulation in Abschnitt 5.4, dient dazu die Spannungen im Unterleib beim Stuhlgang zu untersuchen.

5.1 Modellierung der Ablösung der Innere Grenzmembran

In folgendem Abschnitt wird ein Trainingsprogramm beschrieben, das für den ophtalmochirurgischen Simulator EYESI-*Vitreoretinal* entwickelt wurde. Die Dissertation von Wagner [2003] beschreibt EYESI-*Vitreoretinal* umfassend. Verschiedene Aspekte von EYESI-*Vitreoretinal* wurden auch in Schill et. al. [1999b;a], Schill, Wagner, Bender und Männer [2000], Wagner, Schill, Hennen, Männer, Jendritza, Knorz und Bender [2001], Wagner, Schill und Männer [2002a;b] veröffentlicht.

[1] Niederländisch-Amerikanischer Computerwissenschaftler (1953–1994)

5.1.1 Medizinischer Hintergrund

Die *Innere Grenzmembran* oder *ILM* (von englisch *Internal Limiting Membrane*) ist die innerste Schicht der Retina. Bei bestimmten Pathologien muss die ILM im Bereich der Makula entfernt werden, um Zugang zu darunter liegenden Schichten der Retina zu ermöglichen.

Der Eingriff wird in mehreren Schritten durchgeführt. Häufig wird die ILM grün eingefärbt, um die transparente Membran besser sichtbar zu machen. Danach wird die ILM mit einem spitzen Instrument angerissen. Das angerissene Stück der Membran wird mit einer Pinzette gegriffen und abgezogen. Hierbei muss die Rissausbreitung in der Membran so kontrolliert werden, dass die ILM im gesamten Bereich der Makula möglichst in einem Stück abgelöst wird. Berührungen der Retina müssen dabei vermieden werden. Nach dem Ablösen wird die ILM mit einem Saugschneide-Instrument, dem so genannten *Vitrektor* abgesaugt.

5.1.2 Anforderungen an die Simulation

Bei der Auswertung von Video-Aufzeichnungen mehrerer chirurgischer ILM-Ablösungen wurde beobachtet, wie sich die Membran bei einem Eingriff verhält. Aus den Beobachtungen und durch Absprache mit Chirurgen ergeben sich folgenden Anforderungen an eine Simulation der ILM.

- Modellierung des steifen Gewebeverhaltens der ILM.
- Modellierung einer realistischen Rissausbreitung.
- Modellierung des Haftens und des Ablösens der ILM von den unteren Schichten der Retina.
- Berücksichtigung inhomogener Strukturen: Die ILM lässt sich in der Regel im Bereich der Blutgefäße leichter ablösen als in anderen Bereichen.
- Berücksichtigung individueller Unterschiede: je nach Pathologie und Alter des Patienten variiert die Tendenz der ILM, beim Ablösen zu fragmentieren.
- Die ILM befindet sich auf der Innenseite des Augapfels, die Simulation ist also auf das Innere einer Hohlkugel beschränkt.

5.1.3 Stand der Technik

Soweit dem Autor bekannt ist, existiert keine andere Simulation der Inneren Grenzmembran.

5.1.4 Das erstellte Trainingsprogramm

Gewebesimulation Für die Simulation der Inneren Grenzmembran wird ein Mass-Spring-Ansatz mit expliziter Integration gewählt, um topologische Änderungen in Echtzeit berücksichtigen zu können. Als Integrationsmethode wird ein Runge-Kutta Verfahren 4. Ordnung (siehe Anhang B.1.4) verwendet, da es für ein explizites Verfahren gute numerische Stabilität aufweist. Um die Ausbreitung lokaler Störungen zu beschleunigen, wird der Ansatz mit dem in Abschnitt 2.3 vorgestellte Dragnet-Algorithmus kombiniert. Das Reißen wird mit dem flächenerhaltenden Reißalgorithmus aus Abschnitt 3.3 modelliert.

Geometrie des Gitters Die virtuelle ILM besteht aus einem Gitter von Massepunkten, die durch Spring-String-Elemente verbunden sind. Bei dem verwendeten Reißalgorithmus pflanzen sich Risskanten immer entlang einer Feder fort. Die mögliche Rissausbreitung in einem Knoten wird also durch die Anzahl und Verteilung der Federn des Knotens bestimmt. Bei der Wahl der Gitterkonfiguration ist deshalb darauf zu achten, dass in jedem Knoten ein Reißen in möglichst viele Richtungen möglich ist. Aus diesem Grund wurde eine regelmäßige hexagonale Gitterstruktur gewählt (siehe Abbildung 5.1), in der jeder Knoten mit sechs Nachbarn verbunden ist. Die Winkel zwischen den Verbindungen sind alle identisch.

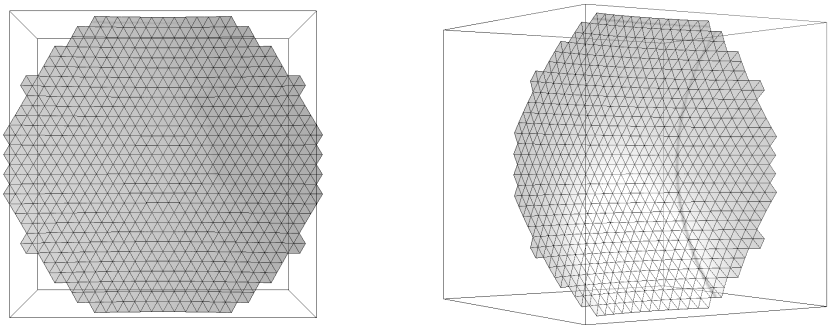


Abbildung 5.1: Das für die Simulation der ILM erzeugte Gitter aus Massenknoten, die mit Spring- und String-Elementen verbunden sind. Das Gitter hat eine hexagonale Grundstruktur um das Reißen in sechs mögliche Richtungen zu ermöglichen. Die konkave Wölbung passt die simulierte Membran an die Form der Retina an.

Instrument Interaktionen Während einer virtuellen ILM-Ablösung interagiert die Membran mit Nadel, Pinzette und Vitrektor. Wagner et. al. [2002a] beschreiben die im EYESI-Framework integrierten Verfahren zur Kollisionserkennung und -antworten und wie damit die Interaktionen zwischen Membranen und den verwendeten Instrumenten modelliert werden.

Für das erstellte Trainingsprogramm wird auf die im EYESI-Framework vorhandenen Interaktionen zurückgegriffen.

Interaktion mit der Augenumgebung Die Innere Grenzmembran haftet bis zur Ablösung mit der Pinzette an den darunter liegenden Schichten der Retina. Im Modell werden die Knoten der virtuellen ILM durch Kräfte an ihrer Ausgangsposition gehalten. Die Kräfte sind direkt proportional zur Entfernung des Knotens von seiner Ausgangslage. Überschreitet diese Entfernung einen vorgegebenen Betrag, wird der Knoten als abgelöst markiert und es wirken im weiteren Verlauf des Eingriffs keine Haltekräfte mehr auf den Knoten.

Eine weitere Randbedingung für die simulierten Knoten ist durch die Retina gegeben. Die ILM kann sich entlang der Retinaoberfläche bewegen, aber nicht in sie eindringen. Die Modellierung der Randbedingung gestaltet sich recht einfach. Die Retina befindet sich auf der Innenseite einer Kugelschale. In jedem Zeitpunkt muss der Abstand der Simulationspunkte zum Mittelpunkt des Auges bestimmt werden. Überschreitet der Abstand den inneren Radius des Auges wird er korrigiert.

Reißen Der in Abschnitt 3.3 vorgestellte Reißalgorithmus bildet die Grundlage für die Modellierung der Rissausbreitung bei der ILM. Bei Beobachtungen von realen Operationen fallen einige Punkte auf, die bei der Modellierung des Reißens im vorliegenden Fall berücksichtigt werden müssen.

- Die ILM reißt bevorzugt an der Stelle, an der sie sich von der Netzhaut ablöst.
- In den Bereichen, in denen die Netzhaut von Blutgefäßen durchzogen wird, reißt die ILM tendenziell schneller, d.h. bei geringerer Belastung.
- In der Regel kommt es bei der Ablösung der ILM nur zu einer Ausbreitung vorhandener Risse. Verzweigung von Rissen und die Bildung neuer Löcher sind hingegen selten zu beobachten.

Knoten der simulierten ILM, die an der Retina anhaften, sind in der Regel nur wenig aus ihrer Ausgangslage verschoben. Somit sind die wirkenden Federkräfte und die Knotenspannungen normalerweise gering im Vergleich zu Knoten, die sich bereits gelöst haben. Aus diesem Grund reißen gelöste Knoten schneller, was sich nicht mit obigen Beobachtungen deckt. Um das Verhalten der simulierten ILM an die realen Gegebenheiten anzupassen, werden bei der Berechnung der Knotenspannungen auch die Kräfte berücksichtigt, die den Knoten auf der Retinaoberfläche halten. Zusätzlich wird

der Rissfaktor t_i in Gleichung 3.1 beim Ablösen eines Knotens verringert, um das Reißen an der Retina-Oberfläche zu bevorzugen.

Das oben beschriebene inhomogene Reißverhalten der ILM wird ebenfalls mit dem Rissfaktor t_i modelliert. Knoten in der Nähe der Blutgefäße haben einen größeren Faktor. Dadurch reißt die ILM bevorzugt in diesen Bereichen.

Um die Rissausbreitung gegenüber den anderen Reißarten zu bevorzugen, wird den Schwellwert für das Reißen von Risskeimen deutlich niedriger gewählt als bei den anderen Knotentypen.

Variationen des Programms Abhängig von Alter, Pathologie und individuellen Besonderheiten eines Patienten variiert das Verhalten der ILM. Um den Bereich der möglichen Fälle abzudecken, wurden in dem erstellten Trainingsprogramm sechs Schwierigkeitsstufen verwirklicht, die sich in folgenden Punkten unterscheiden.

- Unterschiedlich starke Haftung der ILM an den darunter liegenden Schichten der Retina.
- Verschieden starke Tendenzen der abgelösten Membran zum Reißen.
- Einfärbung der ILM. Einige Chirurgen färben die ILM nicht ein, da eine toxische Wirkung des Farbstoffes nicht vollkommen ausgeschlossen werden kann.

Stärke Haftung der ILM an den unteren Schichten der Retina wurde durch eine Erhöhung der Haltekräfte und eine Vergrößerung der Entfernung erzielt, ab der ein Knoten als abgelöst betrachtet wird.

Die Tendenz der abgelösten Membran zu reißen wird hauptsächlich durch die oben beschriebene Erhöhung des Rissfaktors beim Ablösen bestimmt. Eine stärkere Erhöhung des Rissfaktors verringert die Tendenz der ILM, nach der Ablösung zu reißen.

Bei den niedrigeren Schwierigkeitsstufen ist die virtuelle Membran grün eingefärbt und weist eine mittlere Transparenz auf, damit sie gut sichtbar ist. Bei den höheren Schwierigkeitsstufen ist die ILM nicht eingefärbt und hat eine hohe Transparenz. Nur die Kanten der ILM sind bei Bewegungen schwach sichtbar.

5.1.5 Ergebnisse

Das Trainingsprogramm wurde wie oben beschrieben implementiert und in den EYESI-*Vitreoretinal* Simulator integriert. Ophtalmochirurgen, die

die virtuelle ILM-Ablösung evaluiert haben, beschrieben das Verhalten der Membran durchgängig als realistisch. Das Modul hat mittlerweile die Produktreife erreicht und wird zu Ausbildungszwecken eingesetzt. Die Firma VRmagic GmbH, Mannheim vertreibt das Trainingsprogramm kommerziell als optionales Modul für den EYESI-*Vitreoretinal* Simulator. Ein virtuell durchgeführter Eingriff ist in Abbildung 5.2 zu sehen.

5.2 Capsulorhexis

Nachfolgend wird ein Trainingsprogramm beschrieben, dass für den EYESI-*Cataract* Simulator entwickelt wurde. Bei dem *Capsulorhexis* genannten Eingriff handelt es sich um einen Arbeitsschritt bei einer *Katarakt*-Operation

5.2.1 Medizinischer Hintergrund

Bei dem so genannten *Grauen Star* trübt sich die Augenlinse und behindert das Eindringen des Lichts in das Auge. Bei einer Katarakt Operation wird die getrübte Linsenkapsel durch eine Kunstlinse ersetzt. Der betroffene Patient erhält sein Augenlicht zurück. Um an den Linsenkern zu gelangen, muss zunächst der umliegende Kapselsack geöffnet werden. Bei dem Capsulorhexis genannten Vorgang wird der Kapselsack mit einer Pinzette gegriffen und ein möglichst rundes Stück des oberen Kapselsacks entfernt. Die Schwierigkeit bei der Operation liegt in der Kontrolle der Rissausbreitung. Es muss verhindert werden, dass der Riss sich nach außen bis zur Rückseite des Kapselsacks fortpflanzt.

5.2.2 Anforderungen an die Simulation

Gespräche mit Ärzten und die Auswertung von Operationsvideos ergaben folgende Anforderungen an eine virtuelle Capsulorhexis.

- Der obere Teil des Kapselsacks muss simuliert werden. Das Gewebe verhält sich dort wie eine steife Membran.
- Die Dicke der Kapselsackmembran führt dazu, dass die Biegesteifigkeit des Gewebes nicht vernachlässigt werden kann.
- Der Kapselsack ist mit der Kapsel verwachsen. Die Ablösung des Kapselsacks muss modelliert werden.
- Der Kapselsack wird im Normalfall so abgelöst, dass sich der Riss kreisförmig um die Achse der Linse ausbreitet.

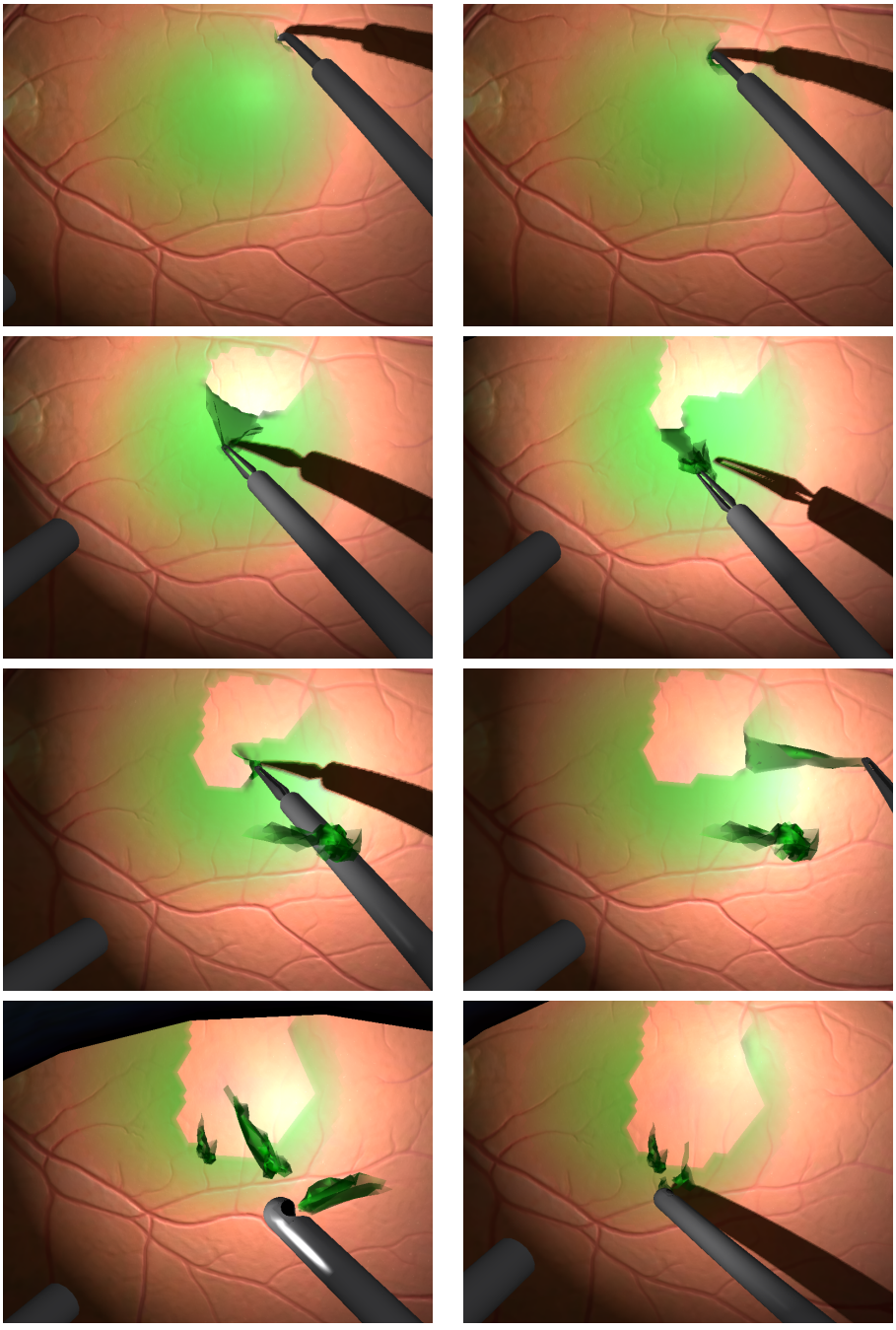


Abbildung 5.2: Die virtuelle Ablösung der Inneren Grenzmembran. Von links oben nach rechts untern: Zunächst wird die ILM mit einer gebogenen Kanüle angeritzt. Anschließen wird sie mit einer Pinzette gegriffen und von den unteren Schichten der Retina abgezogen. Zum Schluss saugt man mit einem Vitrektor die Membran ab.

- Eine häufige Komplikation bei falscher Zugrichtung ist eine radiale Rissausbreitung nach außen.

5.2.3 Stand der Technik

Webster, Sassani, Shenk und Zoppetti [2004b] verwirklichen die Simulation einer Capsulorhexis auf Basis des *PHANToM*s. Das PHANToM ist ein 3D Eingabegerät mit Force-Feedback von der Firma *SensAble Technologies*. Webster et. al. verwenden eine Mass-Spring Simulation mit implizier Integration. Sie beschränken die Simulation auf das Gebiet des Reißens. Wie das Reißen modelliert wird, beschreiben Webster et. al. nicht.

5.2.4 Das erstellte Trainingsprogramm

Gewebesimulation Bei der Simulation des Kapselsacks muss, wie bei der ILM in Abschnitt 5.1.4, eine steife Membran modelliert werden, die während der Simulation topologische Änderungen erfährt. Aus diesem Grund wird auch in diesem Fall ein Mass-Spring-Ansatz mit expliziter Runge-Kutta Integration 4. Ordnung verwendet. Da der Kapselsack relativ dick ist, wird neben den Feldern auch durch Biegeelemente modelliert. Die Bewegung der Pinzette ist durch Hornhaut und Linse auf einen kleinen Raum eingeschränkt. Im Vergleich zu der Ablösung der Inneren Grenzmembran sind die Verschiebungen einzelner Knoten bei der Capsulorhexis Simulation deshalb gering. Eine Verwendung des Dragnet-Algorithmus ist nicht notwendig. Der flächenerhaltende Reißmethode aus Abschnitt 3.3 bildet die Grundlage des verwendeten Reißmodels.

Geometrie des Gitters Bei der Wahl der Gitterkonfiguration ist darauf zu achten, dass sich die Risse während der Operation in der Regel entlang einer Kreisbahn um die optische Achse ausbreiten. Unter verschiedenen Voraussetzungen kommt es auch zu einem Riss, der sich schnell in radialer Richtung fortpflanzt. Um die Vorzugsrichtungen der Rissausbreitung zu unterstützen, wird ein Gitter mit einer konzentrischen Kreisstruktur erstellt. Zusätzlich existieren sechs radial verlaufende Verbindungslinien. Abbildung 5.3 zeigt das für die virtuelle Capsulorhexis verwendete Gitter.

Instrument-Interaktionen Wie bei der Instrument-Interaktion im ILM-Trainingsprogramm (Abschnitt 5.1.4) wird für die Kollisionserkennung und -antwort zwischen Pinzette und Kapselsack auf bestehende EYESI-Interaktionen zurückgegriffen.

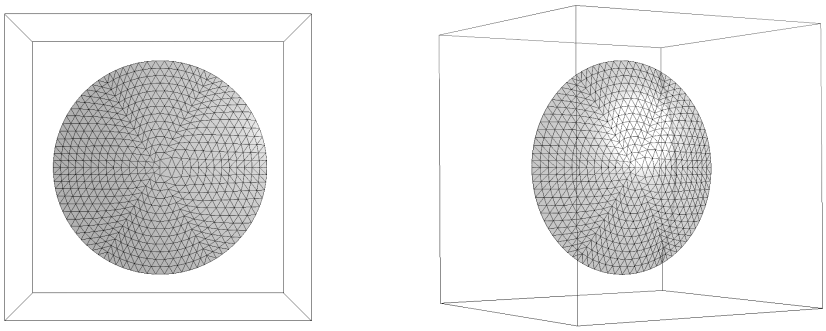


Abbildung 5.3: Beim Simulationsgitter des Kapselsacks liegen die Knoten auf konzentrischen Kreisen, um eine runde Rissausbreitung zu bevorzugen. Das Mesh ist konvex gewölbt.

Interaktion mit der Linsenumgebung Die Bewegungsfreiheit des Kapselsacks wird nach unten durch die Kapsel und nach oben durch die Hornhaut beschränkt. Der Bereich, in dem sich die Gitterknoten aufhalten können, wird durch zwei Kugelschalen mit unterschiedlichen Radien und Mittelpunkten beschrieben. Bei jedem Zeitschritt wird geprüft, ob sich ein Knoten innerhalb der Kugelschale, die die Hornhaut beschreibt und außerhalb der Kugelschale für den Linsenkern befindet. Ist dies nicht der Fall, wird seine Position radial korrigiert.

Das Anhaften des Kapselsacks an der Kapsel wird analog zum Anheften der ILM an der Retina modelliert (siehe Abschnitt 5.1.4).

Reißen Bei dem verwendeten Reißalgorithmus breitet sich der Riss entlang der Feder mit der geringsten relativen Länge fort (siehe Kapiteln 3.3.3). Die Wahrscheinlichkeit, dass ein Riss entlang einer Feder verläuft, wird somit entscheidend von der *Vorspannung* der Feder bestimmt. Unter Vorspannung der Federn versteht man, dass die Länge der Federn in der Ausgangsposition des Gitters größer als ihre Nulllänge ist. Bei der Simulation der Capsulorhexis ist es nicht möglich, dass Knoten die virtuelle Kapsel durchdringen. Das Gitter kann sich nicht zusammenzieht und so die Vorspannung abbaut. Eine Vorspannung der radial angeordneten Federn begünstigt eine radiale Rissausbreitung. Vorspannung der tangentialen Federn vereinfacht die Ausbildung eines runden Risses.

Variationen des Programms Wie stark der Kapselsack dazu tendiert kreisförmig oder radial nach außen zu reißen, ist individuell sehr unterschiedlich. Bei der Capsulorhexis ist das kreisförmige Reißen erwünscht, das radiale Reißen soll vermieden werden. Das Capsulorhexis-Trainingsprogramm beinhaltet fünf Stufen. Mit Erhöhung der Schwierigkeitsstufen wird

die Vorspannung des Gitters in radialer Richtung verstärkt und in tangentialer Richtung geschwächt. Somit wird es schwerer, den Kapselsack kreisförmig reißen zu lassen und radiale Risse nach außen zu vermeiden. Des Weiteren ist bei den einfachen Schwierigkeitsstufen der Kapselsack eingefärbt, um besser sichtbar zu sein.

5.2.5 Ergebnisse

Das Capsulorhexis-Trainingsprogramm wurde wie oben beschrieben implementiert und ist Teil des Operationssimulators EYESI-*Cataract*. Augenärzte bestätigen das plausible Verhalten des modellierten Kapselsacks. Das erstellte Capsulorhexis Modul befindet sich zusammen mit EYESI-*Cataract* in der Prototypenphase und soll Mitte 2005 die Produktreife erlangen. Geplant ist ein Vertrieb über die VRmagic GmbH, Mannheim. Eine simulierte Capsulorhexis im aktuellen Entwicklungsstand 5.4 ist in Abbildung dargestellt.

5.3 Simulation der Entfernung eines Darmpolypen

Der ungewohnte Blickwinkel und die schwierige Kontrolle der Instrumente bei endoskopischen Eingriffen verlangen nach großer Erfahrung und permanentem Training. Die *Virtuelle Endoskopie*[2] hat das Potenzial, diagnostische Eingriffe in vielen Fällen zu ersetzen. Bei nötigen therapeutischen Endoskopien fehlt dem Operateur dann die Erfahrung der diagnostischen Interventionen. Training an einem Endoskopiesimulator kann die fehlende Praxis ausgleichen. Im Folgenden wird ein Trainingsprogramm vorgestellt, das für den Endoskopiesimulator EndoSim entwickelt wurde. Die Dissertation von Körner [2003] beschreibt die Hardware- und Software-Architektur von EndoSim ausführlich. Vor allem der Aspekt des Force-Feedbacks wurde außerdem in Körner und Männer [2002; 2003] sowie in Körner, Rieger und Männer [2004] diskutiert.

5.3.1 Medizinischer Hintergrund

Die Entfernung eines Darmpolypen wird in folgenden Schritten durchgeführt.

- Der Operateur bewegt die Spitze des Endoskops durch den Darm bis zu der Stelle, an der sich der Polyp befindet.

[2] Bei der Virtuellen Endoskopie wird der diagnostische endoskopische Eingriff durch die Begutachtung von Datensätzen eines bildgebenden Verfahrens ersetzt.

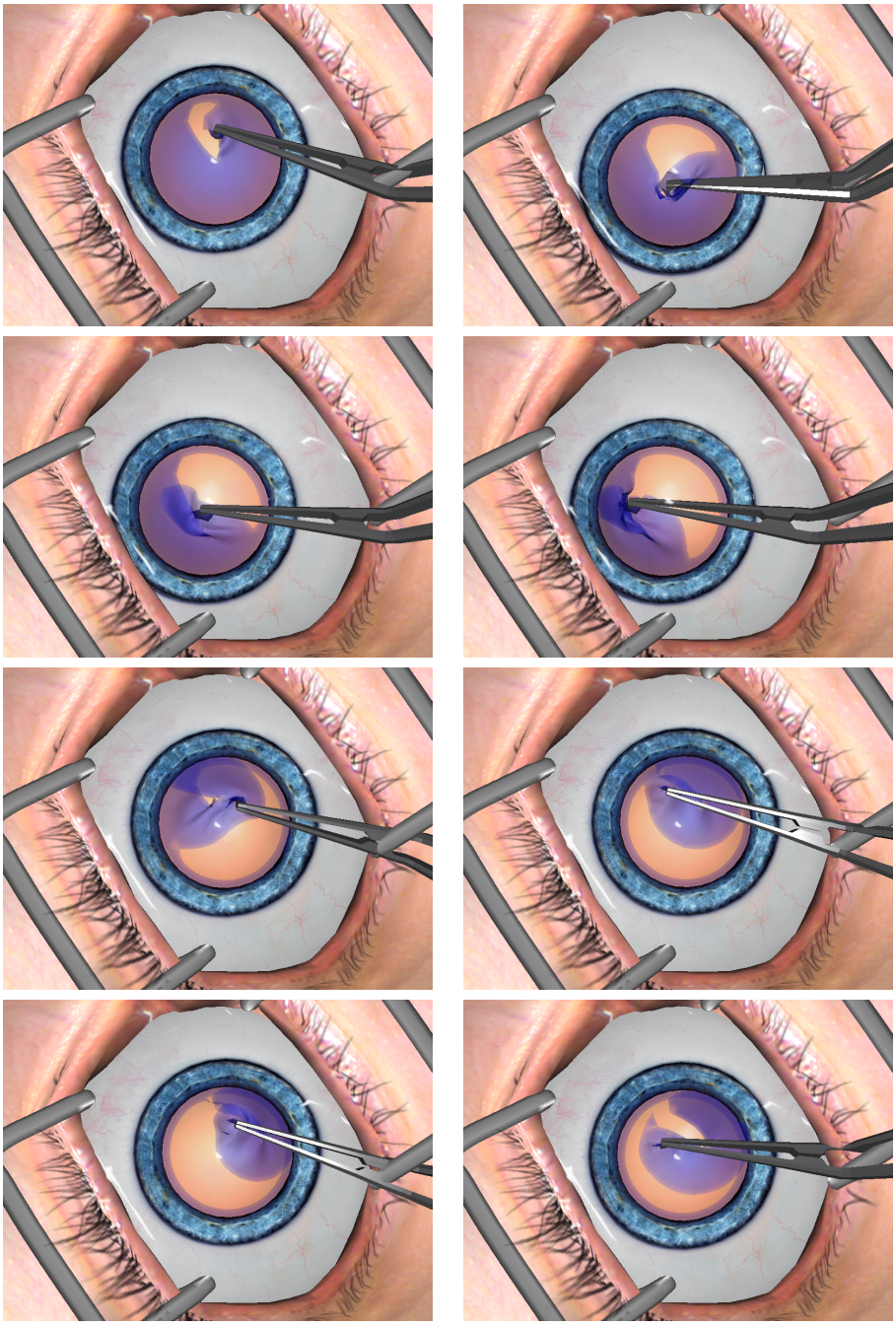


Abbildung 5.4: Eine virtuelle Capsulorhexis. Von links oben nach rechts unten ist dargestellt, wie der Kapselsack gegriffen und angerissen wird. Dann wird ein kreisförmiges Stück des Kapselsacks entfernt. Dazu ist mehrmaliges Nachgreifen nötig, um die Rissausbreitung kontrollierte zu können.

- Eine Drahtschlinge wird ausgefahren und um den Polypen gelegt.
- Ein elektrischer Strom durch die Schlinge verödet den Polypen, um Blutungen zu reduzieren.
- Der Polyp wird mit einer Zange gegriffen, abgerissen und aus dem Darm entfernt.

5.3.2 Anforderungen an die Simulation

Bei der Entwicklung des Trainingsprogramms waren folgende Probleme zu lösen.

- Simulation der elastischen Verformungen des Polypen
- Modellierung des Auseinaderreißens des Polyen
- Modellierung des Aus- und Einfahrens der Drahtschlinge
- Interaktion des Polypen mit den Instrumenten

5.3.3 Stand der Technik

Wildermuth et. al. [2001] haben den Prototypen eines Systems entwickelt, mit dem unter anderem die Entfernung eines Darmpolypen trainiert werden kann. Das System basiert auf dem Spring-Framework, das in Abschnitt 4.1 diskutiert wird. Das Simulationsmodell kann aus MRI und CT Aufnahmen erstellt werden. Die Simulation von Wildermuth et. al. basiert auf einem Feder-Masse-Modell mit einer expliziten Euler- oder Runge-Kutta-Integration. Für Interaktion zwischen Polyp und Drahtschlinge berechnen Wildermuth et. al. die Eindringtiefe der Drahtschlinge in den Polypen. Daraus berechnen sie die nötigen Deformationen des Polypen, um die Kollision aufzulösen. Überschreitet die Federkräfte einen Schwellwert, schneidet die Schlinge in den Polypen. Das Gitter wird neu trianguliert, um die von der Schlinge gegebene Schnittebene abzubilden.

5.3.4 Das erstellte Trainingsprogramm

Gewebesimulation des Polypen Die Deformationen des Polypengewebes werden durch einen Finite-Elemente-Ansatz (siehe Abschnitt 2.2.1) beschrieben. Verformungen werden physikalisch durch die lineare Elastizitätstheorie beschrieben (siehe Anhang A.3). Das Gitter wird aus 4-Knoten-Tetraederelementen aufgebaut wie sie in Anhang C.3 beschrieben sind. Zur Lösung der resultierenden Bewegungsgleichung 2.1 wird der Houbolt-Ansatz gewählt (siehe Abschnitt B.2.2). Die implizite Methode ist unbedingt stabil und bereitet somit keine numerischen Probleme.

Modellierung der Drahtschlinge Ein Catmull-Rom-Spline (siehe Catmull und Rom [1974], Faux und M.J. [1979]) wird verwendet, um das Aus- und Einfahren der Schlinge zu animieren. Anders als bei den häufig verwendeten Bezier-Splines liegen die Kontrollpunkte P_i beim Catmull-Rom-Ansatz auf der Kurve $F(t)$.

$$F(t_i) = P_i \tag{5.1}$$
$$F(t_{i+1}) = P_{i+1} \tag{5.2}$$

Die Steigung in den Kontrollpunkten wird durch die benachbarten Kontrollpunkte definiert.

$$F'(t_i) = \frac{1}{2}\left(P_{i+1} - P_{i-1}\right) \tag{5.3}$$
$$F'(t_{i+1}) = \frac{1}{2}\left(P_{i+2} - P_i\right) \tag{5.4}$$

Ein Punkt $F(t)$ auf dem Catmull-Rom-Spline zwischen zwei Kontrollpunkten P_i und P_{i+1} wird durch den Laufparameter $t \in [0;1]$ bestimmt.

$$F(t) = \frac{1}{2}\left(1, t, t^2, t^3\right) \begin{pmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} \tag{5.5}$$

Weil die Kontrollpunkte auf dem Spline liegen, ist es einfach, den Verlauf einer Kurve zu parametrisieren. Die Stetigkeit sorgt für einen Kurvenverlauf, der zu einer realistischen Drahtschlinge führt.

Der Verlauf der Schlinge wird durch eine Reihe von Kontrollpunkten festgelegt. Der Catmull-Rom-Spline wird zur Rekonstruktion der Kurve genutzt. Das Aus- und Einfahren der Schlinge wird durch Verschieben der Kontrollpunkte modelliert. Hierzu wird für jeden Kontrollpunkt ein weiterer Catmull-Rom-Spline definiert, der die Trajektorie eines Kontrollpunktes während dem Einfahren beschreibt.

Um den berechneten Kurvenverlauf wird ein Tetraedergitter gelegt. Das Durchbiegen der Schlinge wird mit der Finiten-Elemente-Methode simuliert. Wie der Polyp wird das Gitter der Drahtschlinge aus 4-Knoten-Tetraederelementen (siehe Anhang C.3) aufgebaut. Als Integrationsmethode wird wieder der implizite Houbolt-Ansatz verwendet.

Veröden und Auseinanderreißen des Polypen Die verwendete implizite Houbolt-Integration bei der Simulation des Polypen ist unbedingt stabil, hat aber den Nachteil, dass topologische Änderungen eine rechenintensive Matrixinvertierung zu Folge haben.

Die einzige topologische Änderung während einer Operation ist das Auseinanderreißen des Polypen unter Zuhilfenahme der Zange. Die genaue Lage des entstehenden Risses wird bereits zuvor bestimmt. Wenn der Strom durch die Drahtschlinge fließt, um den Polypen zu veröden, wird das Gewebe in diesem Bereich geschwächt. Beim anschließenden Ziehen mit der Zange reißt das geschwächte Gewebe durch.

Zwischen der Festlegung der Lage des Risses und dem Abreißen vergeht eine gewisse Zeit. Die Drahtschlinge muss eingefahren werden und die Zange zum Polypen bewegt werden. In der Zwischenzeit ist es möglich, die für die topologischen Änderungen nötigen Berechnungen als Hintergrundprozess parallel zu der laufenden Simulation durchzuführen.

Es werden zwei separate Gitter für Kopf und Stumpf des Polypen erstellt und die nötigen Vorverarbeitungsschritte durchgeführt. Um die laufende Simulation des Polypen nicht zu beeinflussen, werden die Berechnungen in einem separaten Thread mit niedriger Priorität im Hintergrund ausgeführt.

Sobald der Polyp mit der Zange gegriffen wird und das Reißen beginnt, schaltet die Simulation von dem Simulationsgitter des gesamten Polypen auf die beiden Gitter von Kopf und Stumpf um. Hierzu werden die aktuellen Positionen, Geschwindigkeiten und Beschleunigungen der Knotenpunkte vom alten in das neue Gitter übertragen. Dann läuft die Simulation auf den beiden Gittern der Polypenteile weiter. Um das Auseinaderreißen selbst zu modellieren, werden die Knoten an der Schnittstelle der beiden Gitter solange zusammengehalten, bis die Knotenkräfte einen bestimmten Schwellwert überschreiten.

5.3.5 Ergebnisse

Ein Prototyp einer virtuellen Polypenentfernung wurde wie oben beschrieben implementiert und läuft als eigenständige Anwendung. Die Instrumente werden mit Tastatur oder Maus kontrolliert. Zum Zeitpunkt der Erstellung dieser Arbeit wird an der Integration des Programms in den Simulator EndoSim gearbeitet. Alles in allem benötigen die Simulationen im schlechtesten Fall 11ms pro Zeitschritt. Eine Validierung der Simulation durch Experten steht momentan noch aus. Abbildung 5.5 zeigt die virtuelle Polypenentfernung.

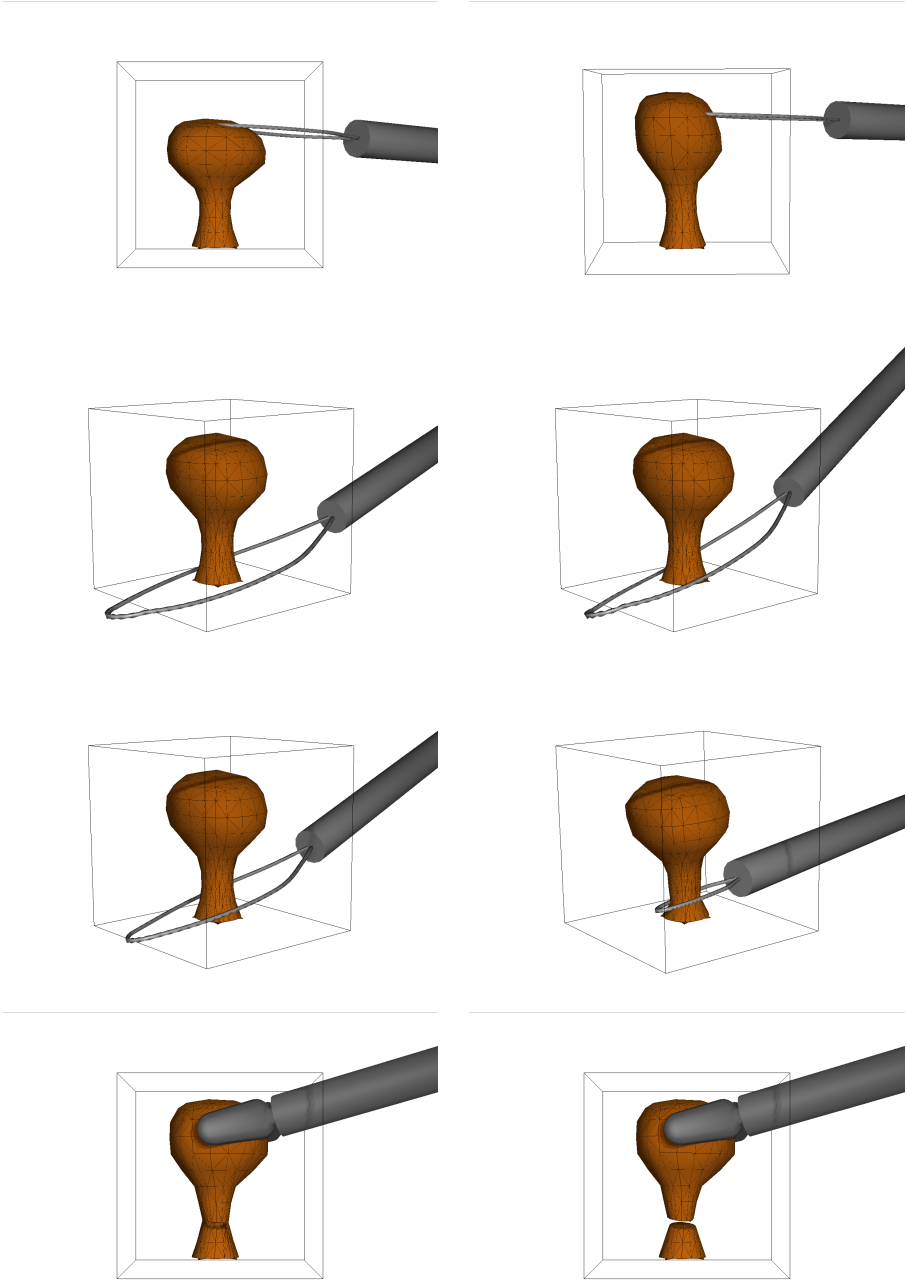


Abbildung 5.5: Eine virtuelle Polypenentfernung. Von links oben nach rechts unten sieht man wie die Schlinge um den Polypen gelegt und festgezogen wird. Dann wird er Polyp durch einen Strom verödet. Anschließend greift der Chirurg den Polyen mit einer Zange, zerreißt ihn und zieht ihn aus dem Darm.

5.4 Unterleibssimulation

Im Folgenden wird eine Anwendung beschrieben, die als Offline-Simulation konzipiert wurde. Mit der Simulation des Unterleibes sollen mögliche Risikofaktoren für das Auftreten einer Inkontinenz gefunden werden.

5.4.1 Medizinischer Hintergrund

Inkontinenz ist ein verbreitetes Problem unter Frauen. Häufig tritt Inkontinenz nach einer Geburt temporär oder permanent auf. Man geht davon aus, dass hohe Belastungen des Beckenbodens beim Pressen während des Stuhlgangs ebenfalls die Wahrscheinlichkeit einer Inkontinenz erhöhen. Mattox, Lucente, McIntyre, Miklos und Tomezsko vermuten, dass die Anordnung der Wirbelsäule und des Beckens die Kraftübertragung zum Beckenboden beeinflusst und somit eine Schlüsselrolle bei der Entwicklung von Inkontinenz spielt. Es wird angenommen, dass eine normale Konfiguration den Beckenboden vor großen Kräften schützt. Außerdem wird vermutet, dass die Festigkeit der Bauchdecke ebenso einen Einfluss auf die Kraftübertragung hat.

Abbildung 5.6: Midsagittale MR Aufnahmen eines gesunden Freiwilligen bevor und während des Pressens.

Deformationen und Bewegungen im intraabdominalen Raum können durch eine Zeitfolge von midsagittaler MR Bilder sichtbar gemacht werden (siehe Lienemann [1998]). Abbildung 5.6 zeigt zwei Bilder einer solchen Serie.

Wirkende Kräfte und die Spannungen innerhalb des Gewebes können mit dieser Methode nicht sichtbar gemacht werden. Der Vergleich verschiedener Knochenkonfigurationen gestaltet sich schwierig, da passende Probanden gefunden werden müssen.

5.4.2 Anforderungen an die Simulation

Ziel des Projekts war die Entwicklung einer prototypischen Simulation eines midsagittalen Querschnitts des Unterleibs. Mit der Anwendung sollen schnell und einfach verschiedene Konfigurationen qualitativ miteinander verglichen werden können. Die Anwendung dient dazu interessante Konfigurationen zu finden, für die sich eine nachfolgende aufwendige quantitative Untersuchung loht. Folgende Anforderungen wurden an die Simulation gestellt.

- Einfache Anpassbarkeit der Simulation an verschiedene Konfigurationen der Wirbelsäule und des Beckens.
- Einfache Festlegung der Gewebeeigenschaften.
- Veränderung der Stärke des Pressens zu Laufzeit.
- Ausgaben der Belastungen des Gewebes während des Pressens.

5.4.3 Stand der Technik

d'Aulignac, Martins, Mascarenhas, Jorge und Pires [2004b] entwickeln eine Finite-Elemente-Simulation des Beckenbodens. Ihre Simulation dient dazu, die Belastung des Beckenbodens während einer Geburt abzuschätzen. Das Model besteht aus zweidimensionalen Oberflächenelementen im dreidimensionalen Raum. d'Aulignac et. al. modellieren nicht den gesamten intraabdominalen Raum, sondern ausschließlich den Beckenboden. Ihr Ansatz ist aus diesem Grund nicht geeignet, um die Belastung des Beckenbodens beim Stuhlgang zu simulieren.

5.4.4 Die erstellte Anwendung

Die erstellte Anwendung ist eine zweidimensionale Finite-Elemente-Echtzeitsimulation des intraabdominalen Raums. Ausgehend von der linearen Elastizitätstheorie (siehe Anhang A.3) wurden 3-Knoten-Dreieckselemente und 2-Knoten-Linienelemente entwickelt, wie sie im Anhang C.1 und C.2 beschrieben sind. Zur Integration wurde der Houbolt Ansatz (Anhang B.2.2) wegen seiner numerischen Stabilität gewählt. Das Simulationsgitter wird aus einer Reihe von Kontrollpunkten bestimmt, die aus einer Konfigurationsdatei eingelesen werden. Damit ist es möglich, das Gitter an verschiedene Formen und Anordnungen von Wirbelsäule und Becken anzupassen. Der intraabdominalen Raum wird durch Dreieckselemente modelliert. Linienelemente beschrieben die umliegenden Muskeln: den Beckenboden, die Bauchdecke und das Zwerchfell. Abbildung 5.7 zeigt Beispiele der erstellten

Gitter. Die Bauchdecke kann zusätzlich fixiert werden, um ein Nachgeben komplett zu verhindern. Die Simulation benötigt vier Materialparameter. Die Poisson'sche Zahl und das Elastizitätsmodul beschreiben das elastische Verhalten, Massendichte und Dämpfungskonstante die kinematischen Eigenschaften des Gewebes. Die Parameter können für jeden Gewebetyp separat gesetzt werden. Auf das Zwerchfell wirkt eine sinusförmige Kraft, um das Pressen zu modellieren. Amplitude und Frequenz können bei laufender Simulation geändert werden.

Abbildung 5.7: Das FEM-Gitter mit ca. 6000 Dreieckselementen und ungefähr 200 Linienelementen. Der intraabdominale Raum wird von der Wirbelsäule (rechts), dem Zwerchfell (oben), den Bauchmuskeln (links), dem Beckenknochen (unten links) und dem Beckenboden (unten rechts) begrenzt.

5.4.5 Ergebnisse

Die Anwendung wurde wie oben beschrieben entwickelt. Die Anwendung benötigt auf einem Pentium III mit 800MHz ca. 4s für das Erzeugen des Gitters und die Initialisierung der Simulation. Jeder Zeitschritt benötigt ca. 300ms Rechenzeit.

Das Ergebnis der Simulation wird als Spannungsverteilung im intraabdominalen Raum und dem Beckenboden numerisch und in Falschfarben (siehe Abbildung 5.8) ausgegeben. Außerdem wird das korrespondierende Verschiebungsvektorfeld dargestellt (Abbildung 5.9).

Für eine Untersuchung wurden die vorkommenden physikalischen Größen durch Literaturwerte angenähert. Nach Levinson, Shinagawa und Sato [1995] wurden die Elastizitätsparameter für das Muskelgewebe abgeschätzt. Um die Größenordnung eines möglichen Drucks festzulegen, wurden Daten verwendet, die Sanchez, Tenofsky, Dort, Shen, Helmer und Smith [2001] beim Gewichtheben gemessen haben.

Der Vergleich der Simulation mit den dynamischen MR-Aufnahmen zeigt eine qualitative Übereinstimmung der Deformationen.

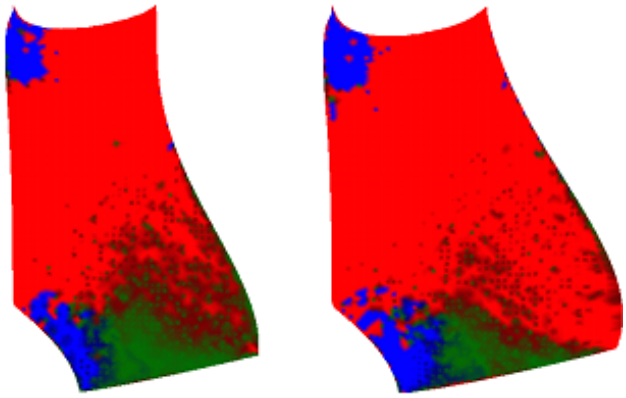


Abbildung 5.8: Die Spannungsverteilung im intraabdominalen Raum bei zwei verschiedenen Knochen-Anordnungen. Die schmale Konfiguration auf der linken Seite führt zu einer geringeren Spannung im Beckenboden.

Die Vermutungen, dass eine schmale Anordnung von Becken und Wirbelsäule und eine straffe Bauchmuskulatur die Spannungen im Beckenboden minimiert, ließ sich durch die Simulation bestätigen (siehe Abbildung 5.8). Sowohl die durchschnittliche Spannung als auch der Maximalwert nehmen ab, wenn die Konfiguration verschmälert wird oder das Elastizitätsmodul der Bauchmuskulatur erhöht wird. Abbildung 5.9 zeigt eine Ablenkung des Verschiebungsvektorfeldes in Abhängigkeit von der Nachgiebigkeit der Bauchdecke.

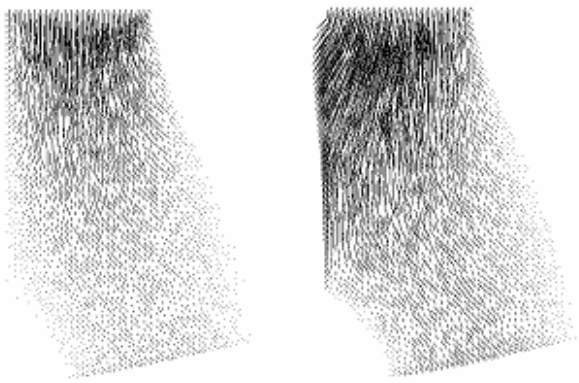


Abbildung 5.9: Das Verschiebungsvektorfeld einer Simulation mit fixierter Bauchdecke (links) und beweglicher Bauchdecke (rechts). Die bewegliche Bauchdecke führt zu einer horizontalen Ablenkung de Verschiebungsvektoren.

5.5 Zusammenfassung

In diesem Kapitel wurden die Anwendungen beschrieben, die im Rahmen dieser Arbeit entwickelt wurden.

Abschnitt 5.1 beschreibt eine virtuelle Ablösung der Inneren Grenzmembran. Eine Kombination aus dem vorgestellten Dragnet-Algorithmus und einer Mass-Spring-Simulation sowie dem vorgestellten Reißansatz ermöglicht die Modellierung eines steifen Gewebeverhaltens unter der Echtzeitrandbedingung. Die virtuelle Ablösung der ILM wurde von Chirurgen als realistisch eingeschätzt. Das erstellte Trainingsprogramm hat Produktreife erlangt und wird zur ophthalmologischen Ausbildung eingesetzt. Die Modellierung der Gewebedeformationen ist in Grimm et. al. [2004] beschrieben. Das verwendete Reißverfahren ist unter Grimm [2005b] erscheinen.

Ein Trainingsprogramm zum Üben einer Capsulorhexis-Prozedur wurde in Abschnitt 5.2 beschrieben. Die Simulation ähnelt der Simulation virtuellen ILM-Ablösung. Das Reißverhalten des Kapselsacks ist allerdings deutlich komplexer. Risse entstehen bevorzugt radial und tangential zu Linsenachse. Durch eine geeignete Gittergeometrie und verschiedene Vorspannung der radialen und tangentialen Federn kann das Reißverhalten an die Gegebenheiten angepasst werden. Dem Trainingsmodul wurde von Fachärzten ein realistisches Verhalten attestiert. Das Programm steht kurz vor der Produktreife und soll ab Mitte des Jahres als Modul von EYESI-*Cataract* zur Ausbildung eingesetzt werden.

In Abschnitt 5.3 wurde der Prototyp einer virtuellen Polypenentfernung vorgestellt, die in den Endoskopiesimulator EndoSim integriert werden soll. Das Modell kombiniert die Methode der Finiten-Elemente mit Animationstechniken, um unter der Echtzeitrandbedingung den Polypen und die verwendete Drahtschlinge zu modellieren. Der verwendete Ansatz ist dank impliziter Integration numerisch unbedingt stabil. Die Genauigkeit der Finiten-Elemente-Methode wird durch den verwendeten linearen Ansatz eingeschränkt. Bei kleinen Rotationen der Objekte ist die Genauigkeit für unsere Zwecke ausreichend. Die Geschwindigkeit der Simulation ist mit ca. 11ms ausreichend für eine flüssige Visualisierung. Für eine flüssige haptische Ausgabe müsste die Simulation weiter beschleunigt werden. Momentan bietet EndoSim allerdings kein Force-Feedback für den Instrumentenkanal. Eine Beschreibung des entwickelten Trainingsprogramms wird unter Grimm [2005a] veröffentlicht.

Die Unterleibssimulation aus Abschnitt 5.4 modelliert einen Querschnitt des intraabdominalen Raums mit der Methode der Finiten-Elemente. Die Anwendung ermöglicht einen einfachen und schnellen quantitativen Vergleich von Spannungsverteilung und Verschiebungen während des Stuhlgangs bei verschiedenen Knochenkonfigurationen. Die Validierung erfolgte hier durch

den visuellen Vergleich mit dynamischen MR-Aufnahmen. Die Ergebnisse der Simulation sind unter Grimm et. al. [2002] veröffentlicht.

6

Diskussion und Ausblick

„Predictions are hard, especially about the future.“

—wird Niels Bohr [1] zugeschrieben

Die Diskussion und der Ausblick nehmen Bezug auf die Ergebnisse der vorherigen Kapitel 2 bis 5 und gliedert sich entsprechend in die folgenden Unterabschnitte.

6.1 Modellierung von Deformationen

Im Rahmen dieser Arbeit wurde der Dragnet Algorithmus (siehe Kapitel 2) entwickelt um steifes Gewebeverhalten in Echtzeit zu modellieren, das während der Laufzeit topologische Änderungen wie Schneiden oder Reißen erfährt.

Gängige Simulationsalgorithmen wie Mass-Spring, FEM oder Tensor-Mass sind numerisch nur bedingt stabil, wenn sie ein explizites Integrationsverfahren einsetzen. Besonders bei der Modellierung eines steifen Gewebeverhaltens müssen kleine Zeitschritte gewählt werden und somit mehr Berechnungen pro Zeiteinheit durchgeführt werden, damit die Verfahren numerisch stabil bleiben. Beim Einsatz einer impliziten Integration wird das System durch eine Matrix beschrieben die invertiert werden muss, um die Gewebedeformationen zu berechnen. Ändert sich die Topologie des simulierten Gewebes muss die Matrix angepasst und neu invertiert werden. Durch den

[1] Dänischer Physiker (1885–1962)

hohen Rechenaufwand für die Invertierung der Matrix sind Simulationen mit impliziter Integration kaum geeignet um Gewebe zu simulieren, das während der Laufzeit topologisch verändert wird. Das verschiebungsorientierte Chain-Mail ist numerisch unbedingt stabil und kann topologische Änderungen in Echtzeit berücksichtigen. Lokale Rotationen des Gewebes können mit Chain-Mail allerdings nicht modelliert werden, da die Deformationsberechnungen für die Koordinatenachsen separat durchgeführt werden.

Die Idee des neu entwickelten Dragnet-Algorithmus ist das Ziehen an einem Netz aus zusammengeknoteten Schnüren. Der Ansatz ist wie Chain-Mail verschiebungsorientiert und numerisch stabil. Topologische Änderungen erfordern neben den Veränderungen des Simulationsgitters keine zusätzliche Rechenzeit. Dragnet kann im Gegensatz zu Chain-Mail lokale Rotationen modellieren. Kombiniert man eine Mass-Spring Simulation mit Dragnet, wird die Konvergenzgeschwindigkeit gegenüber dem Mass-Spring-Ansatz deutlich beschleunigt. In Messreihen zeigt sich, dass die Kombination Dragnet mit Mass-Spring bei lokalen Interaktionen schneller konvergiert wie existierende Ansätze zu Beschleunigung der Konvergenzgeschwindigkeit.

Nachteil von Dragnet ist der relativ hohe Rechenaufwand, der durch die interaktionspunkt-basierte Abarbeitung der Knoten bedingt ist. Außerdem wäre es wünschenswert mit Dragnet nicht nur plastische sondern auch elastische Deformationen modellieren zu können. Damit wäre es nicht mehr nötig Dragnet mit einem Mass-Spring-Ansatz zu kombinieren.

Neben der Erweiterung von Dragnet ist geplant, die Eignung der relativ neuen Gitterlosen Verfahren (siehe Abschnitt 2.2.3) für medizinische VR-Simulatoren zu untersuchen. Die Gitterlosen Verfahren ermöglichen es, topologische Veränderungen und sogar Schmelzen zu modellieren. Die Frage ist, ob das Verfahren schnell genug ist, um Gewebe in einer ausreichenden Auflösung zu modellieren.

6.2 Modellierung von Reißen

In der wissenschaftlichen Literatur ist wenig über die Echtzeitmodellierung von Reißen zu finden. Einfache Ansätze löschen Simulationselemente, wenn ihre Spannung einen Schwellwert überschreitet. In Kapitel 3 wurde ein Reißalgorithmus für flächige Objekte entwickelt. Der Ansatz greift Ideen auf, wie sie bei der offline Animation von Brüchen und dem Schneiden von Gewebe verwendet werden. Um zu bestimmen wann, wo und in welche Richtung das Gewebe reißt, wird die Spannungsverteilung im Simulationsgitter ausgewertet. Verschiedene Schwellwerte für Entstehung, Ausbreitung und Verzweigung eines Risses machen es möglich, das Reißverhalten verschiedenster Stoffe wie beispielsweise von spröden Objekten zu modellieren.

Verglichen mit dem Reißen durch Elementlöschen ist der entwickelte Ansatz nicht nur flexibler, sondern liefert auch realistischer Ergebnisse, da die Fläche beim Reißen nicht verringert wird. Der etwas höhere Rechenbedarf fällt in der Praxis meist nicht ins Gewicht, da in der Regel die Rechenzeit für die Modellierung des Reißens deutlich kleiner ist, als die Rechenzeit die für die Deformationsberechnung benötigt wird.

Hauptnachteil des vorliegenden Algorithmus ist, dass nur entlang von Dreieckskanten gerissen werden kann. Momentan wird an einem Schneidealgorithmus gearbeitet, der es ermöglicht, vorhandene Dreiecke zu unterteilen. Dabei ist zu beachte, dass sich Anzahl und Größe der Dreiecke nicht zu stark verändern, um keine negativen Auswirkungen auf numerische Stabilität und Geschwindigkeit der Simulation zu haben. Die Ergebnisse des Schneidealgorithmus sollen anschließend helfen den Reißalgorithmus so zu erweitern, dass in alle Richtungen gerissen werden kann.

Ein weiterer Nachteil die Beschränkung auf flächige Objekte. Die grundlegenden Ideen der knotenbasierten Spannungsauswertung und Aufteilung der Simulationselemente könnte auch für einen Reißalgorithmus auf Volumenelementen verwendet werden.

6.3 Software-Architektur

Im Rahmen dieser Arbeit wurde eine Simulationsbibliothek entwickelt, die sich in eine VR-Architektur eingliedert (siehe Kapitel 4).

Bestehende Architekturen haben den Nachteil, dass sie nur wenige Simulationsalgorithmen unterstützen (SPRING), nicht mehr gepflegt werden (KISMET) oder sich erst in der Entwicklung befinden (SOFA).

Die entwickelte Struktur basiert auf einer dualen Repräsentation der Simulationsgitter in Graph- und Array-Struktur. Während die Graphstruktur es erleichtert topologische Änderungen und gitterbasierte Deformationen zu modellieren, bietet die Arraystruktur einen deutlichen Geschwindigkeitsvorteil. Die Graphstruktur ist so implementiert, dass die simulierten Objekte, bei topologischen Veränderungen, automatisch in einem konsistenten Zustand bleiben. Im Vergleich zur Graphstruktur wird die Simulation durch die Array-Struktur um bis zu einem Faktor acht beschleunigt. Dadurch lassen bei gleicher Auflösung achtmal so viele Iterationsschritte pro Zeitschritt ausführen was z.B. bei einer Runge-Kutta Integration 4. Ordnung (siehe B.1.4) den numerischen Fehler um mehr als vier Größenordnungen reduziert.

Um die Geschwindigkeit der Algorithmen weiter zu beschleunigen wird darüber nachgedacht, die Bibliothek so zu erweitern, dass Simulation auf *GPU* (*Graphics Processing Unit*) oder *FPGA*s (*Field-Programmable Gate Array*)

ausgelagert werden können. Die massive Parallelverarbeitung auf diesen Prozessoren sollte zu deutlichen Verbesserungen des Laufzeitverhaltens führen. Außerdem sollen die Bibliothek um weitere Simulationsansätze wie die Gitterlosen Verfahren ergänzt werden.

6.4 Anwendungen

Die oben beschriebenen Algorithmen wurden entwickelt und implementiert um die in Kapitel 5 beschriebenen Anwendungen zu realisieren.

Eine Anwendung, die im Rahmen dieser Arbeit entwickelt wurde, ist die Simulation der Ablösung der Inneren Grenzmembran (ILM) im menschlichen Auge. Die Innere Grenzmembran wird mit einer Kombination aus Mass-Spring-Simulation und Dragnet-Ansatz modelliert. Die Kombination ermöglicht ein steifes Gewebeverhalten bei großen lokalen Interaktionen in Echtzeit, zu modellieren. Dadurch hat die virtuelle ILM ein ausreichend steifes Verhalten, selbst wenn ein einzelner Knoten mit einer virtuellen Pinzette gegriffen und um einen großen Betrag verschoben wird. Die Rissausbreitung wurde mit dem flächenerhaltenden Algorithmus modellierte. Fachärzte bestätigen das realistische Reiß- Ablöse- und Deformationsverhalten der virtuellen Membran. Das Trainingsprogramm hat die Produktreife erreicht und wird bereits zur ophthalmologischen Ausbildung eingesetzt. Ansatz für weitere Verbesserungen des Trainingsprogramms ist, die Auflösung des Simulationsgitters zu erhöhen. Dazu müsste die Geschwindigkeit der Algorithmen erhöht werden.

Ein weiteres Trainingsprogramm wurde für den EYESI-*Cataract* Simulator entwickelt. Das Programm ermöglicht das Training eines Capsulorhexis Eingriffs, bei dem aus der vorderen Linsenkapsel ein kreisförmiges Stück gerissen wird. Das Trainingsprogramm basiert auf einer Mass-Spring-Simulation. Die Rissausbreitung wird mit dem vorgestellten flächenerhaltenden Algorithmus modelliert.

Die virtuelle Capsulorhexis befindet sich momentan in der letzten Phase des Prototypen-Stadiums und soll ab Mitte nächsten Jahres als Modul von EYESI-*Cataract* zur Ausbildung eingesetzt werden. Fachärzte bewerten den virtuellen Eingriff bereits als realistisch.

Für das Problem der Capsulorhexis wird außerdem ein deskriptives Reißverfahren entwickelt, bei dem ein direkter Zusammenhang zwischen Riss- und Zugrichtung hergestellt wird. Dadurch ist es möglich, Fehler des Trainierenden festzustellen und zu korrigieren, um somit ein pädagogisch besseres Training zu ermöglichen.

Für EYESI werden momentan weitere neue Trainingssimulationen entwickelt. Zur Vervollständigung der virtuellen Kataraktoperation wird die Cap-

sulorhexis-Prozedur um eine Phakoemulsifikation und das Einsetzen einer Kunstlinse ergänzt. Die Korrektur einer Netzhautablösung mit einem Laser wird ebenfalls virtuell nachgebildet. Des Weiteren wird der Austausch des Glaskörpers durch eine Flüssigkeit oder ein Gas modelliert.

Für den Endoskopie-Simulator EndoSim wurde ein Modul entwickelt, mit dem die Entfernung eines Darmpolypen trainiert werden kann. Eine Finite-Elemente-Simulation modelliert die Deformationen des Polypen und der verwendeten Drahtschlinge. Aus Geschwindigkeits- und Stabilitätsgründen wurde ein linearer Ansatz gewählt. Der lineare Ansatz hat den Nachteil, dass große lokale Rotationen zu unrealistischen Volumenvergrößerungen führen. Für die, bei der virtuellen Polypenentfernung benötigte Genauigkeit ist der lineare Ansatz ausreichend.

Das Abreißen des Polypen wird in einem Hintergrund-Thread vorbereitet, nachdem durch das Veröden des virtuellen Polypen die Bruchstelle festgelegt ist. Durch die zeitliche Differenz zwischen Festlegung der Rissstelle und dem eigentlichen Abreißen ist es möglich, die zeitaufwendige Matrixmultiplikation parallel zur laufenden Simulation durchzuführen. Auf aktuellen HyperThreading Prozessoren wird die laufende Simulation damit nicht merklich verlangsamt. Es ist zu erwarten, dass das Verfahren außerdem von den angekündigten Multi-Core Prozessoren profitieren wird.

Bei der Unterleibssimulation ging es darum, einen Eindruck von der Spannungsverteilung im Unterleib und dem Beckenboden während des Stuhlgangs zu bekommen. Mit einem linearen Finiten-Elemente-Ansatz wurde ein Querschnitt des intraabdominalen Raums dynamisch modelliert. Die Ergebnisse der zweidimensionalen Querschnittssimulation stimmen qualitativ mit dynamischen MR-Aufnahmen überein. Infolge des linearen Ansatzes und der Beschränkung auf zwei Dimensionen lassen sich mit der Simulation keine quantitativen Aussagen machen. Lage und Form des Beckens, der Wirbelsäule und die Parameter des Gewebes lassen sich in der Anwendung frei definieren. Die Simulation läuft in Echtzeit und die externen Kräfte können zu Laufzeit variiert werden. Dies ermöglicht einen schnellen und einfachen Vergleich verschiedener Knochen-Konfigurationen. Dadurch kann eine Vorauswahl von Knochen-Konfigurationen getroffen werden für die sich einen zeitaufwendige nicht-lineare, dreidimensionale Analyse lohnt.

A

Dynamik deformierbarer Körper

Folgende Differentialgleichung beschreibt die Dynamik deformierbarer Körper in Abhängigkeit von der Zeit t und den Materialkoordinaten $\vec{r}$.

$$\frac{\partial}{\partial t}\left(\rho(\vec{r})\frac{\partial \vec{r}}{\partial t}\right) + \gamma(\vec{r})\frac{\partial \vec{r}}{\partial t} + \frac{\delta E}{\delta \vec{r}} = F_{ext}(\vec{r}, t) \tag{A.1}$$

Wobei $\rho(\vec{r})$ und $\gamma(\vec{r})$ die Massendichte beziehungsweise die Dämpfungsdichte des Körpers sind. $E(\vec{r})$ ist die potentielle Energie der Deformation und $F_{ext}(\vec{r}, t)$ sind die externen Kräfte, die auf den Körper einwirken.

Die Deformationsenergie ist dabei durch

$$E = \frac{1}{2}\int_{\Omega} \sigma^T \epsilon \quad d\Omega \tag{A.2}$$

gegeben, wobei σ und ϵ der Verzerrungstensor und der Spannungstensor sind. Beides sind symmetrische 3×3 Tensoren zweiter Ordnung.

A.1 Verzerrungstensor

Der Verzerrungstensor ϵ beschreibt die Verzerrung eines deformierbaren Körper in abhängig von der Veränderung des Abstandsvektors $\vec{u}$ zweier infinitesimal entfernter Punkte.

St. Venant-Kirchhoff Elastizität Bei der St. Venant-Kirchhoff Elastizität wird der Verzerrungstensor wie folgt definiert.

$$\epsilon = \frac{1}{2}\left(\nabla\vec{u} + \nabla\vec{u}^T + \nabla\vec{u}^T\nabla\vec{u}\right) \tag{A.3}$$

Oder in Komponenten Schreibweise:

$$\epsilon_{ik} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} + \sum_l \frac{\partial u_l}{\partial x_i}\frac{\partial u_l}{\partial x_k}\right) \tag{A.4}$$

ϵ wird als Green-Saint-Venant Verzerrungstensor bezeichnet.

A.2 Spannungstensor

Der Spannungstensor σ beschreibt die elastischen Spannungen, die aufgrund der Verzerrungen in einem Körper auftreten.

$$\sigma = \sigma(\epsilon) \tag{A.5}$$

$\sigma(\epsilon)$ hängt im Allgemeinen von den folgenden physikalischen Größen ab.

Elastizitätsmodul Das Elastizitätsmodul E oder auch Young's Modul ist definiert als Quotient aus angelegter einachsige Spannung und der resultierenden axialen Verzerrung.

$$E = \frac{\sigma_l}{\epsilon_l} \tag{A.6}$$

Poisson'sche Zahl Poisson'sche Zahl ν ist der negative Quotient aus transversaler Verzerrung zu axialer Verzerrung.

$$\nu = -\frac{\epsilon_t}{\epsilon_l} \tag{A.7}$$

Anschaulich beschreibt die Poisson'sche Zahl wie sich ein Material verdünnt, wenn es in die Länge gezogen wird. ν ist ein Indikator für die Kompressibilität des Materials. Ein Wert von $\nu = 0.5$ bedeutet völlige Inkompressibilität. Kleinere Werte stehen für leichter komprimierbares Material.

Schermodul Das Schermodul G wird als Quotient aus Scherspannung τ und der resultierenden Scherung θ definiert.

$$G = \frac{\tau}{\theta} \tag{A.8}$$

Im Allgemeinen sind E, G und ν richtungsabhängig und von der Verzerrung abhängig.

A.3 Lineare Elastizitätstheorie

Oft wird zur Vereinfachung eine lineare Beziehung zwischen Verschienung und Spannung angenommen. Die lineare Elastizitätstheorie basiert auf zwei Annahmen. Ist die Verzerrung linear zur Verschiebung, spricht man von geometrischer Linearität. Wird die Beziehung zwischen Verzerrungstensor und Spannungstensor linear angenommen, spricht man von einer linearen Elastizität des Materials oder von einem Hooke'schen Material.

A.3.1 Geometrische Linearität

Durch die geometrische Linearität vereinfachen sich A.3 und A.4 zu

$$\epsilon = \frac{1}{2}\left(\nabla\vec{u} + \nabla\vec{u}^T\right) \tag{A.9}$$

und

$$\epsilon_{ik} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right) \tag{A.10}$$

Die geometrisch lineare Vereinfachung führt dazu, dass der Verzerrungstensor nicht mehr rotationsinvariant ist. Rotationen führen zu von null verschiedenen Einträgen im Verzerrungstensor, obwohl physikalisch keine Verzerrung vorliegt. Seine Invarianz gegenüber Translationen behält der Verzerrungstensor bei der linearen Näherung bei.

A.3.2 Hooke'sches Material

Durch die Annahme eines Hooke'schen Materials lässt sich Gleichung A.5 wie folgt darstellen.

$$\sigma = C\epsilon \tag{A.11}$$

Dabei ist C ein Tensor vierter Stufe.

Berücksichtigt man, dass σ und ϵ symmetrisch sind kann man ihre Einträge in zwei Vektoren der Dimension 6 zusammenfassen.

$$\vec{\epsilon} = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{zx} \\ \epsilon_{xy} \end{pmatrix} \qquad \vec{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{pmatrix} \tag{A.12}$$

Gleichung A.11 lässt sich dann durch eine Matrixmultiplikation ausdrücken.

$$\vec{\sigma} = D\vec{\epsilon} \tag{A.13}$$

A.3.3 Isotrope Materialien

Für isotrope Materialien sind E, G und ν nicht richtungsabhängig. G lässt sich aus E und ν ableiten.

$$G = \frac{E}{2\,(1+\nu)} \tag{A.14}$$

Es gibt nur zwei unabhängige Parameter, die die elastischen Eigenschaften des Materials beschreiben.

Die Matrix D aus Gleichung A.13 wie nimmt für isotrope Materialien folgende Form an.

$$D_{iso} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \tag{A.15}$$

Eine äquivalente Darstellung die häufig in der Literatur zu finden ist basiert auf den Lamé-Koeffizienten λ und μ.

$$\mu := \frac{E}{2\,(1+\nu)} \tag{A.16}$$

$$\lambda := \frac{\nu E}{(1+\nu)\,(1-2\nu)} \tag{A.17}$$

Umgekehrt lassen sich auch das Elastizitätsmodul und die Poisson'sche Zahl durch die Lamé-Koeffizienten ausdrücken.

$$E = \frac{\mu\,(3\lambda+2\mu)}{\lambda+\mu} \tag{A.18}$$

$$\nu = \frac{\lambda}{2\,(\lambda+\mu)} \tag{A.19}$$

In dieser Darstellungsform wird die Matrix D_{iso} zu

$$D_{iso} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \tag{A.20}$$

A.3.4 Transversal isotrope Materialien

Eine besondere Form der der Anisotropie ist die transversale Isotropie wie sie zum Beispiel bei fasrigem Gewebe vorkommt.

Transversal isotrope Materialien haben eine ausgezeichnete Richtung. Das longitudinal Elastizitätsmodul für Verzerrungen entlang dieser Richtung E_l unterscheiden sich von dem dazu transversalen Parameter E_t. Da bei dem Schermodul und der Poisson-Zahl die Richtungen von ursächlicher Verzerrung und wirkender Spannung nicht übereinstimmen, müssen ν_{tl}, ν_{lt}, ν_{tt} und G_{tl}, G_{lt}, G_{tt} unterschieden werden.

Es lässt sich zeigen, dass

$$\nu_{lt} E_t = \nu_{tl} E_l \tag{A.21}$$

gilt.

Da das Material in der transversalen Ebene isotrop ist, lässt sich G_{tt} durch E_t und ν_{tt} ausdrücken.

$$G_{tt} = \frac{E_{tt}}{2\,(1+\nu_{tt})} \tag{A.22}$$

Es bleiben fünf unabhängige Parameter E_t, E_l, ν_{lt}, ν_{tt} und G_{lt}, die das Material beschreiben.

Im Folgenden wird die Matrix D aus Gleichung A.13 für transversal isotrope Materialien durch ihre Inverse definiert, da diese eine einfachere Form hat.

$$D_{trans}^{-1} = \begin{pmatrix} \frac{1}{E_t} & \frac{-\nu_{tt}}{E_t} & \frac{-\nu_{lt}}{E_l} & 0 & 0 & 0 \\ \frac{-\nu_{tt}}{E_t} & \frac{1}{E_t} & \frac{-\nu_{lt}}{E_l} & 0 & 0 & 0 \\ \frac{-\nu_{tl}}{E_t} & \frac{-\nu_{tl}}{E_l} & \frac{1}{E_l} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2G_{tl}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2G_{tl}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1+\nu_{tt}}{E_t} \end{pmatrix} \tag{A.23}$$

B

Integrationsverfahren

Anhang B listet einige Integrationsverfahren auf die bei der bei medizinischen Simulationen Anwendung finden. Die Integrationsverfahren dienen dazu die Lösungen der Gleichung A.1 zu finden.

B.1 Explizite Integrationsverfahren

B.1.1 Explizite Euler Integration

Beim punktweisen *expliziten Euler*-Verfahren werden die Kräfte $\vec{F}$ auf jeden Punkt i zu einem Zeitpunkt t berechnet. Es wird davon ausgegangen, dass die Kräfte über einen Zeitschritt Δt konstant beleiben. Mit dieser Annahme werden die Geschwindigkeit $\vec{v}_i$ für den Zeitpunkt $t + \Delta t$

$$\vec{v}_i(t+1) \quad = \quad \vec{v}_i(t) \quad + \quad \frac{\vec{F}_i\left(\vec{x}(t), \vec{v}(t), t\right)}{m_i} \cdot \Delta t \tag{B.1}$$

und anschließend die Positionen $\vec{x}_i$ abgeschätzt.

$$\vec{x}_i(t+1) \quad = \vec{x}_i(t) \quad + \quad \vec{v}_i(t+1) \cdot \Delta t \tag{B.2}$$

Beim expliziten Euler-Verfahren entsteht ein Fehler in der Größenordnung von $O(\Delta t^2)$

B.1.2 Mittelpunkt-Methode

Bei der expliziten Euler-Integration geht man vereinfachend davon aus, dass die Geschwindigkeit in einem Zeitschritt $t+1$ konstant $\vec{v}_i(t+1)$ ist. Bei der Mittelpunkt-Methode rechnet man deshalb mit der gemittelten Geschwindigkeit aus $\vec{v}_i(t)$ und $\vec{v}_i(t+1)$.

$$\vec{x}_i(t+1) \quad = \vec{x}_i(t) \quad + \quad \frac{\vec{v}_i(t) + \vec{v}_i(t+1)}{2} \cdot \Delta t \tag{B.3}$$

B.1.3 Quasi-statischer Euler

Bei der quasi-statischen Modifikation des Euler-Verfahrens geht man davon aus, dass alle Massepunkte nach einem Integrationsschritt in Ruhe sind ($\vec{v}(t) = 0$).

$$\vec{x}_i(t+1) \quad = \quad \vec{x}_i(t) \quad + \quad \frac{1}{2} \quad \frac{\vec{F}_i\left(\vec{x}(t), t\right)}{m_i} \cdot \Delta t^2 \tag{B.4}$$

B.1.4 Runge-Kutta

Bei der *Runge-Kutta* Integrationsmethode werden die Knotenverschiebungen $\Delta\vec{x}_i$ zu verschiedenen Zeiten des Intervalls Δt abgeschätzt. Anschließend werden die Abschätzungen gewichtet addiert.

Runge-Kutta zweiter Ordnung

Beim Runge-Kutta Verfahren 2. Ordnung werden die Verschiebungen in der Mitte des Intervalls Δt abgeschätzt.

$$k_{1i} := \Delta t \cdot \Delta\vec{x}_i(t, \vec{x}(t)) \tag{B.5}$$

$$k_{2i} := \Delta t \cdot \Delta\vec{x}_i(t + \frac{1}{2}\Delta t, \vec{x}(t) + \frac{1}{2}k_1) \tag{B.6}$$

$$\vec{x}_i(t + \Delta t) = \vec{x}_i(t) + k_{2i} + O(\Delta t^3) \tag{B.7}$$

Runge-Kutta vierter Ordnung

Beim Runge-Kutta Verfahren 4. Ordnung werden die Verschiebungen an vier Stellen des Intervalls Δt abgeschätzt.

$$k_{1i} := \Delta t \cdot \Delta\vec{x}_i(t, \vec{x}_i(t)) \tag{B.8}$$

$$k_{2i} := \Delta t \cdot \Delta\vec{x}_i(t + \frac{1}{2}\Delta t, \vec{x}_i(t) + \frac{1}{2}k_{1i}) \tag{B.9}$$

$$k_{3i} := \Delta t \cdot \Delta\vec{x}_i(t + \frac{1}{2}\Delta t, \vec{x}_i(t) + \frac{1}{2}k_{2i}) \tag{B.10}$$

$$k_{4i} := \Delta t \cdot \Delta\vec{x}_i(t + \Delta t, \vec{x}_i(t) + k_{3i}) \tag{B.11}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \frac{k_{1i}}{6} + \frac{k_{2i}}{3} + \frac{k_{3i}}{3} + \frac{k_{4i}}{6} + O(\Delta t^5) \tag{B.12}$$

B.2 Implizite Integrationsverfahren

Bei den impliziten Verfahren kann nicht mehr punktweise integriert werden. Im Folgenden wird der Vektor U_t eingeführt, der sich aus den Koordinatenänderungen aller Knoten $U_t = (\Delta\vec{x}_i(t))$ zusammensetzt.

Die Beschreibung folgender Verfahren orientiert sich an Bathe [1990].

B.2.1 Newmark Methode

Die *Newmark Methode* geht von folgenden Annahmen aus.

$$\dot{U}_{t+\Delta t} = \dot{U}_t + \left[(1-\delta)\ddot{U}_t + \delta\ddot{U}_{t+\Delta t}\right] \cdot \Delta t \tag{B.13}$$

$$U_{t+\Delta t} = U_t + \dot{U}_t\Delta t + \left[(\frac{1}{2} - \alpha)\ddot{U}_t + \alpha\ddot{U}_{t+\Delta t}\right] \cdot \Delta t^2 \tag{B.14}$$

Gelöst wird das System indem Gleichung B.14 nach $\ddot{U}_{t+\Delta t}$ aufgelöst und in Gleichung B.13 eingesetzt wird. Dadurch lassen sich G $\dot{U}_{t+\Delta t}$ und $\ddot{U}_{t+\Delta t}$ in Abhängigkeit von $U_{t+\Delta t}$ ausdrücken. $\dot{U}_{t+\Delta t}$ und $\ddot{U}_{t+\Delta t}$ werden dann in die

Bewegungsgleichung 2.1 zum Zeitpunkt $t + \Delta t$ eingesetzt. So kann $U_{t+\Delta t}$ bestimmt werden und durch Einsetzen in die aufgelösten Gleichungen anschließend $\dot{U}_{t+\Delta t}$ und $\ddot{U}_{t+\Delta t}$.

Die Parameter α und δ des Ansatzes sind frei wählbar und bestimmen die Integrationsgenauigkeit und Stabilität. Eine Wahl von $\alpha = 1/6$ und $\delta = 1/2$ führt zu so genannten *linearen Beschleunigungsmethode*, bei der von einer linearen Veränderung der Beschleunigung ausgegangen wird. $\alpha = 1/4$ und $\delta = 1/2$ führt zu ursprünglichen unbedingt stabilen Newmark Methode. Zhuang und Canny [1999] benuten die Newmark Methode als explizites Integrationsschema mit $\alpha = 0$ und $\delta = 1/2$.

B.2.2 Houbolt

Die Houbolt Methode beruht auf folgender Finite-Differenzen-Entwicklung:

$$\ddot{U}_{t+\Delta t} = \frac{2}{\Delta t^2} \cdot U_{t+\Delta t} - \frac{5}{\Delta t^2} \cdot U_t + \frac{4}{\Delta t^2} \cdot U_{t-\Delta t} - \frac{1}{\Delta t^2} \cdot U_{t-2\Delta t} \quad \text{(B.15)}$$

$$\dot{U}_{t+\Delta t} = \frac{11}{6\Delta t} \cdot U_{t+\Delta t} - \frac{3}{\Delta t} \cdot U_t + \frac{3}{2\Delta t} \cdot U_{t-\Delta t} - \frac{1}{3\Delta t} \cdot U_{t-2\Delta t} \quad \text{(B.16)}$$

Einsetzen von B.16 und B.16 in die Bewegungsgleichung 2.1 zum Zeitpunkt $t + \Delta t$ liefert $U_{t+\Delta t}$ in Abhängigkeit von U_t, $U_{t-\Delta t}$ und $U_{t-2\Delta t}$. Die Startwerte für U_t, $U_{t-\Delta t}$ und $U_{t-2\Delta t}$ werden durch ein spezielles Verfahren abgeschätzt.

C

Benutzte Finite-Elemente

C.1 Linienelement

Nachfolgend die Herleitung des Linienelements wie sie bei Schill [1995] zu finden ist.

Das Linienelement wird in der xy-Ebene betrachtet. Die zwei Eckpunkte als Knotenpunkte haben die Knotenvariablen $\vec{u}_1 = (u_{x,1})^T$ und $\vec{u}_2 = (u_{x,2})^T$

Der Verzerrungsvektor für den linearen Fall vereinfacht sich zu

$$\vec{\epsilon} = (\epsilon_{xx})^T = \left(\frac{\partial u_x}{\partial x} \right), \tag{C.1}$$

da Verzerrungen in y- und z-Richtung nicht möglich sind.

Die restlichen Komponenten sind null und werden zur Vereinfachung weggelassen.

Ausgehend von linearer Elastizität und einem isotropen Material hat D dann folgende Form.

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1 \end{pmatrix}. \tag{C.2}$$

Elementansatz Als Elementansatz wird eine lineare Funktion gewählt.

$$\vec{u}(x) = \begin{pmatrix} u_x \end{pmatrix} = \begin{pmatrix} c_{10} + c_{11} x \end{pmatrix} \tag{C.3}$$

Für den Verzerrungsvektor ϵ ergibt sich hiermit.

$$\vec{\epsilon} = \frac{1}{2} \begin{pmatrix} \frac{\partial u_x}{\partial x} \end{pmatrix} = \begin{pmatrix} c_{11} \end{pmatrix} \tag{C.4}$$

D und ϵ werden in das Energiefunktional A.2 eingesetzt.

Durch Koeffizientenvergleich bekommt man so die Elementsteifigkeitsmatrix im Koeffizientenraum S

$$S = LA \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{C.5}$$

mit L und A der Länge und Querschnittsfläche des Linien-Elements.

Mit S und

$$\vec{c} = (c_{10}, c_{11})^T \tag{C.6}$$

lässt sich das Energiefunktional in Abhängigkeit der Parameter der Ansatzfunktion als

$$E = \vec{c}^T S \vec{c} \tag{C.7}$$

schreiben.

Verschieben der Ansatzfunktion Gesucht wird die Matrix A, die den Vektor der Knotenvariablen

$$\vec{u} = (u_{x,1}, u_{x,2})^T \tag{C.8}$$

in den Vektor der Koeffizienten der Ansatzfunktion $\vec{c}$, wie er in (C.6) definiert ist, transformiert.

$$\vec{c} = A\vec{u}, \tag{C.9}$$

Setzt man die zwei Endpunkte des Linien-Elements mit den jeweiligen Koordinaten (x_i, $i = 1, 2$ in die Definition der Ansatzfunktion (C.3) ein, so ergeben sich die Verschiebungen $\vec{u}_i$ mit ihren Komponenten $u_{x,i}$und $u_{y,i}$ zu

$$u_{x,i}(x_i) = c_{10} + c_{11} x_i \tag{C.10}$$

In Matrixform stellt sich das Gleichungssystem dann wie folgt dar

$$\vec{u} = A^{-1}\vec{c}. \tag{C.11}$$

Schreibt man das Ganze aus, so ergibt sich.

$$\begin{pmatrix} u_{x,1} \\ u_{x,2} \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} \begin{pmatrix} c_{10} \\ c_{11} \end{pmatrix} \tag{C.12}$$

Durch Invertieren bekommt man dann die Transformationsmatrix A vom Koeffizientenraum in den Verschiebungsraum.

$$\tilde{S} = A^T S A \tag{C.13}$$

C.2 Dreieckselement

Nachfolgend die Herleitung des Dreieckselements wie sie bei Meißner [1998] zu finden ist.

Das Dreieckselement wird in der xy-Ebene betrachtet. Die drei Eckpunkte als Knotenpunkte haben die Knotenvariablen

$\vec{u}_1 = (u_{x,1}, u_{y,1})^T$, $\vec{u}_2 = (u_{x,2}, u_{y,2})^T$ und $\vec{u}_3 = (u_{x,3}, u_{y,3})^T$

Der Verzerrungsvektor für den linearen Fall vereinfacht sich zu

$$\vec{\epsilon} = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{xy})^T = \left(\frac{\partial u_x}{\partial x}, \frac{\partial u_y}{\partial y}, \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right), \tag{C.14}$$

da eine Verzerrung in z-Richtung nicht möglich ist.

Die restlichen Komponenten sind null und werden zur Vereinfachung weggelassen.

Ausgehend von linearer Elastizität und einem isotropen Material hat D dann folgende Form.

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}. \tag{C.15}$$

Elementansatz Als Elementansatz wird eine lineare Funktion gewählt.

$$\vec{u}(x,y) = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} c_{10} + c_{11}x + c_{12}y \\ c_{20} + c_{21}x + c_{22}y \end{pmatrix} \tag{C.16}$$

Für den Verzerrungsvektor ϵ ergibt sich hiermit.

$$\vec{\epsilon} = \frac{1}{2} \begin{pmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \end{pmatrix} = \begin{pmatrix} c_{11} \\ c_{22} \\ c_{12} + c_{21} \end{pmatrix} \tag{C.17}$$

D und ϵ werden in das Energiefunktional A.2 eingesetzt.

Durch Koeffizientenvergleich bekommt man so die Elementsteifigkeitsmatrix im Koeffizientenraum S

$$S = hF \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \nu \\ 0 & 0 & \frac{1-\nu}{2} & 0 & \frac{1-\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1-\nu}{2} & 0 & \frac{1-\nu}{2} & 0 \\ 0 & \nu & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{C.18}$$

mit h und F der Hohe der Fläche des Dreieckselements.

Mit S und

$$\vec{c} = (c_{10}, c_{11}, c_{12}, c_{20}, c_{21}, c_{22})^T \tag{C.19}$$

lässt sich das Energiefunktional in Abhängigkeit der Parameter der Ansatzfunktion als

$$E = \vec{c}^T S \vec{c} \tag{C.20}$$

schreiben.

Verschieben der Ansatzfunktion Gesucht wird die Matrix A, die den Vektor der Knotenvariablen

$$\vec{u} = (u_{x,1}, u_{y,1}, u_{x,2}, u_{y,2}, u_{x,3}, u_{y,3})^T \tag{C.21}$$

in den Vektor der Koeffizienten der Ansatzfunktion $\vec{c}$, wie er in (C.19) definiert ist, transformiert.

$$\vec{c} = A\vec{u}, \tag{C.22}$$

Setzt man die drei Eckpunkte des Dreiecks mit den jeweiligen Koordinaten (x_i, y_i), $i = 1, 2, 3$ in die Definition der Ansatzfunktion (C.16) ein, so ergeben sich die Verschiebungen $\vec{u}_i$ mit ihren Komponenten $u_{x,i}$und $u_{y,i}$ zu

$$u_{x,i}(x_i, y_i) = c_{10} + c_{11}x_i + c_{12}y_i \tag{C.23}$$

$$u_{y,i}(x_i, y_i) = c_{20} + c_{21}x_i + c_{22}y_i \tag{C.24}$$

In Matrixform stellt sich das Gleichungssystem dann wie folgt dar

$$\vec{u} = A^{-1}\vec{c}. \tag{C.25}$$

Schreibt man das Ganze aus, so ergibt sich.

$$\begin{pmatrix} u_{x,1} \\ u_{y,1} \\ u_{x,2} \\ u_{y,2} \\ u_{x,3} \\ u_{y,3} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} c_{10} \\ c_{11} \\ c_{12} \\ c_{20} \\ c_{21} \\ c_{22} \end{pmatrix} \tag{C.26}$$

Durch Invertieren bekommt man dann die Transformationsmatrix A vom Koeffizientenraum in den Verschiebungsraum.

$$\tilde{S} = A^T S A \tag{C.27}$$

C.3 Tetraederelement

Nachfolgend die Herleitung des Tetraederelements wie sie bei Grimm [2000] zu finden ist.

Bei einem Volumenelement sind Verzerrungen in alle Raumrichtungen möglich. Der Verzerrungsvektor ϵ sieht im linearen Fall wie folgt aus.

$$\vec{\epsilon} = \frac{1}{2} \begin{pmatrix} 2\frac{\partial u_x}{\partial x} \\ 2\frac{\partial u_y}{\partial y} \\ 2\frac{\partial u_z}{\partial z} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{pmatrix} \tag{C.28}$$

Ausgehend von linearer Elastizität und einem isotropen Material hat D die Form von Gleichung A.15.

$$D = \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \tag{C.29}$$

Elementansatz Als Elementansatz wird eine lineare Funktion gewählt.

$$\vec{u}(\vec{x}) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} c_{11} + c_{12}x + c_{13}y + c_{14}z \\ c_{21} + c_{22}x + c_{23}y + c_{24}z \\ c_{31} + c_{32}x + c_{33}y + c_{34}z \end{pmatrix} \tag{C.30}$$

Für den Verzerrungsvektor ϵ ergibt sich hiermit.

$$\vec{\epsilon} = \frac{1}{2} \begin{pmatrix} 2\frac{\partial u_x}{\partial x} \\ 2\frac{\partial u_y}{\partial y} \\ 2\frac{\partial u_z}{\partial z} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{pmatrix} = \begin{pmatrix} c_{11} \\ c_{22} \\ c_{33} \\ \frac{1}{2}(c_{12} + c_{21}) \\ \frac{1}{2}(c_{23} + c_{32}) \\ \frac{1}{2}(c_{13} + c_{31}) \end{pmatrix} \tag{C.31}$$

D und ϵ werden in des Energiefunktional A.2 eingesetzt.

Durch Koeffizientenvergleich bekommt man so die Elementsteifigkeitsmatrix im Koeffizientenraum S:

$$S = \frac{E}{(1+v)(1-2\nu)}$$

$$\begin{pmatrix}
1-\nu & 0 & 0 & 0 & 0 & \nu & 0 & 0 & 0 & 0 & \nu & 0 \\
0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & 0 & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\nu & 0 & 0 & 0 & 0 & 1-\nu & 0 & 0 & 0 & 0 & \nu & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 & \frac{1}{8}-\frac{1}{4}\nu & 0 & 0 \\
\nu & 0 & 0 & 0 & 0 & \nu & 0 & 0 & 0 & 0 & 1-\nu & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

mit der sich das Energiefunktional in Abhängigkeit der Parameter der Ansatzfunktion als

$$E = \vec{c}^T S \vec{c} \tag{C.32}$$

mit

$$\vec{c} = (c_{10}, c_{11}, c_{12}, c_{13}, c_{20}, c_{21}, c_{22}, c_{23}, c_{30}, c_{31}, c_{32}, c_{33})^T. \tag{C.33}$$

schreiben lässt.

Verschieben der Ansatzfunktion Gesucht wird die Matrix A, die den Vektor der Knotenvariablen

$$\vec{u} = (u_{x,1}, u_{y,1}, u_{z,1}, u_{x,2}, u_{y,2}, u_{z,2}, u_{x,3}, u_{y,3}, u_{z,3})^T \tag{C.34}$$

in den Vektor der Koeffizienten der Ansatzfunktion $\vec{c}$, wie er in (C.33) definiert ist, transformiert.

$$\vec{c} = A\vec{u} \tag{C.35}$$

Setzt man die vier Eckpunkte des Tetraeders mit den jeweiligen Koordinaten (x_i, y_i),$i = 1, 2, 3, 4$ in die Definition der Ansatzfunktion (C.30) ein, so ergeben sich die Verschiebungen $\vec{u}_i$ mit ihren Komponenten $u_{x,i}$, $u_{y,i}$und $u_{z,i}$ zu

$$u_{x,i}(x_i, y_i, z_i) = c_{10} + c_{11}x_i + c_{12}y_i + c_{13}z_i \tag{C.36}$$

$$u_{y,i}(x_i, y_i, z_i) = c_{20} + c_{21}x_i + c_{22}y_i + c_{23}z_i \tag{C.37}$$

$$u_{z,i}(x_i, y_i, z_i) = c_{30} + c_{31}x_i + c_{32}y_i + c_{33}z_i \tag{C.38}$$

In Matrixform hat das Gleichungssystem dann folgende Form.

$$\vec{u} = A^{-1}\vec{c}. \tag{C.39}$$

mit der Matrix

$$A^{-1} = \begin{pmatrix}
1 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & z_1 \\
1 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & z_3 \\
1 & x_3 & y_3 & z_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_3 & y_3 & z_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_3 & y_3 & z_3 \\
1 & x_4 & y_4 & z_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & x_4 & y_4 & z_4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_4 & y_4 & z_4
\end{pmatrix} \tag{C.40}$$

Durch Invertieren bekommt man die Matrix A die S vom Koeffizientenraum in den Verschiebungsraum transformiert.

$$\tilde{S} = A^T S A \tag{C.41}$$

D

Messwerte

Korrekturfaktor	*Längenkorrektur*	*Dragnet*
0, 1	12291	1855
0, 2	7392	298
0, 3	5398	191
0, 4	4243	129
0, 5	3505	85
0, 6	2981	59
0, 7	2585	34
0, 8	2269	25
0, 9	2005	17
1, 0	1783	1
1, 1	1589	—
1, 2	1419	—
1, 3	1262	—
1, 4	1122	—
1, 5	989	—
1, 6	851	—
1, 7	707	—
1, 8	566	—
1, 9	401	—
2, 0	273	—
2, 1	—	—

Tabelle D.1: Die benötigten Zeitschritte, bis die Simulation, nach der Verschiebung eines Knoten, einen stabilen Zustand erreicht hat, in Abhängigkeit vom Korrekturfaktor.

Anzahl der Strings	Laufzeit *Best Case*	Laufzeit *Worst Case*
758	0,0027 ms	1,4 ms
1560	0,0028 ms	3,3 ms
2070	0,0025 ms	4,6 ms
2652	0,0029 ms	5,9 ms
3306	0,0025 ms	7,5 ms
4032	0,0026 ms	9,5 ms
4830	0,0029 ms	12,1 ms
5700	0,0031 ms	14,9 ms
6642	0,0027 ms	17,3 ms
7656	0,0027 ms	20,4 ms
8742	0,0027 ms	23,6 ms
9900	0,0025 ms	27,2 ms
11130	0,0031 ms	31,4 ms
12432	0,0028 ms	35,3 ms
13806	0,0027 ms	39,1 ms
15252	0,0027 ms	44,3 ms
16770	0,0029 ms	48,6 ms
18360	0,0030 ms	53,2 ms
20022	0,0027 ms	59,9 ms
21756	0,0028 ms	65,7 ms
23562	0,0025 ms	72,0 ms
25440	0,0029 ms	77,3 ms

Tabelle D.2: Die Laufzeit des Dragnet-Algorithmus pro Zeitschritt im worst und im Best Case.

Anzahl der Knoten	Anzahl der Federn	Anzahl der Dreiecke	Laufzeit Reißen durch Löschen	Laufzeit flächener-haltendes Reißen
272	758	486	0, 31 ms	0, 08 ms
397	1122	726	0, 23 ms	0, 45 ms
547	1560	1014	0, 42 ms	0, 64 ms
721	2070	1350	0, 63 ms	0, 81 ms
919	2652	1734	0, 82 ms	1, 03 ms
1141	3306	2166	1, 05 ms	1, 25 ms
1387	4032	2646	1, 26 ms	1, 44 ms
1657	4830	3174	1, 50 ms	1, 63 ms
1951	5700	3750	1, 76 ms	1, 86 ms
2269	6642	4374	2, 04 ms	2, 06 ms
2611	7656	5046	2, 39 ms	2, 38 ms
2977	8742	5766	2, 67 ms	2, 61 ms
3367	9900	6534	3, 02 ms	2, 69 ms
3781	11130	7350	3, 42 ms	2, 91 ms
4219	12432	8214	3, 85 ms	3, 14 ms
4681	13806	9126	4, 24 ms	3, 34 ms
5167	15252	10086	4, 69 ms	3, 58 ms
5677	16770	11094	5, 15 ms	3, 83 ms
6211	18360	12150	5, 64 ms	3, 99 ms
6769	20022	13254	6, 14 ms	4, 22 ms
7351	21756	14406	6, 71 ms	4, 48 ms
7957	23562	15606	7, 27 ms	4, 66 ms
8587	25440	16854	7, 83 ms	5, 00 ms
9241	27390	18150	8, 43 ms	5, 08 ms
9919	29412	19494	9, 00 ms	5, 45 ms
10621	31506	20886	9, 65 ms	5, 53 ms
11347	33672	22326	10, 31 ms	5, 82 ms
12097	35910	23814	10, 90 ms	6, 10 ms

Tabelle D.3: Die Laufzeiten des Flächenerhaltenden Reißalgorithmus und dem Ansatz, der mit Löschen von Federn arbeitet.

Abbildungsverzeichnis

2.1 Explizite Integration 13
2.2 Implizite quasi-statische Integration 14
2.3 Mehrgitter-Integration 15
2.4 Dragnet Beispiel 28
2.5 Konvergenz von Dragnet 29
2.6 Laufzeit des Dragnet-Algorithmus 30
2.7 Vergleich verschiedener Simulationen 32

3.1 Beispiel für Löschen überdehnter Federn 45
3.2 Knotenarten 48
3.3 Modellierung eines Lochs im Gitter 51
3.4 Schritte beim flächenerhaltenden Reißen 53
3.5 Trennung der Gitterteile 54
3.6 Aufteilung der Knoten 55
3.7 Beispiel für flächenerhaltendes Reißen 57
3.8 Zerplatzender Ballon 59
3.9 Laufzeiten der Reißalgorithmen 60
3.10 Vergleich beiden Reißansätze auf einem groben Gitter . . . 61
3.11 Vergleich beiden Reißansätze auf einem Gitter mit mittlerer Auflösung 62
3.12 Vergleich beiden Reißansätze auf einem feinen Gitter 63

4.1 Die VRM-Architektur 68
4.2 Aufbau des *VfemgfxNode* 70

4.3 Aufbau der femLAB Anwendung 71
4.4 Vergleich der Integration auf Array- und Graphstruktur . . 78

5.1 Das ILM-Gitter 85
5.2 Virtuelle Ablösung der Inneren Grenzmembran 89
5.3 Das Capsulorhexis Gitter 91
5.4 Eine virtuelle Capsulorhexis 93
5.5 Eine virtuelle Polypenentfernung 97
5.6 Midsagittale MR Aufnahmen während des Pressens 98
5.7 Das Gitter der Unterleibssimulation 100
5.8 Die Spannungsverteilung im intraabdominalen Raum 101
5.9 Verschiebungsvektorfeld im intraabdominalen Raum 101

Tabellenverzeichnis

2.1 Vergleich der Konvergenz verschiedener Simulationsverfahren 33

2.2 Überblick über die verschiedenen kraftbasierte Simulationsansätze. 37

2.3 Überblick über die verschiedenen verschiebungsorientiert Simulationsansätze. 38

4.1 Überblick über die wichtigsten implementierten Simulationsklassen. 80

4.2 Vergleich der Integration auf Array- und Graphstruktur . . 81

D.1 Konvergenz von Dragnet und der Längenkorrektur 129

D.2 Laufzeit des Dragnet-Algorithmus 130

D.3 Laufzeiten der Reißalgorithmen 131

Literaturverzeichnis

1 Baraff und Witkin 1998 Baraff, David ; Witkin, Andrew: Large Steps in Cloth Simulation. In: *Computer Graphics* 32 (1998), Nr. Annual Conference Series, S. 43–54. – URL `citeseer.nj.nec.com/baraff98large.html`

2 Bathe 1990 Bathe, K.-J.: *Finite-Elemente-Method*. Springer, 1990

3 Bianchi et. al. 2004 Bianchi, Gerald ; Solenthaler, Barbara ; Szekely, Gabor ; Harders, Matthias: Simultaneous Topology and Stiffness Identification for Mass-Spring Models Based on FEM Reference Deformations. In: Barillot, C. (Hrsg.) ; Haynor, D.R. (Hrsg.) ; Hellier, P. (Hrsg.): *miccai2004* Bd. 3217, Springer, September 2004, S. 393–301

4 Bielser und Gross 2000 Bielser, Daniel ; Gross, Markus H.: Interactive Simulation of Surgical Cuts. In: *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, 2000, S. 116. – ISBN 0-7695-0868-5

5 Bielser et. al. 1999 Bielser, Daniel ; Maiwald, Volker A. ; Gross, Markus H.: Interactive Cuts through 3-Dimensional Soft Tissue. In: Brunet, P. (Hrsg.) ; Scopigno, R. (Hrsg.): *Computer Graphics Forum (Eurographics '99)* Bd. 18(3), The Eurographics Association and Blackwell Publishers, 1999, S. 31–38. – URL `citeseer.ist.psu.edu/article/bielser98interactive.html`

6 Bierbaum und Just 1998 Bierbaum, Allen ; Just, Christopher: Software Tools for Virtual Reality Application Development. In: *SIGGRAPH '98 Course 14: Applied Virtual Reality* (1998)

7 Bro-Nielsen und Cotin 1996 Bro-Nielsen, Morten ; Cotin, Stephane: Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. In: *Computer Graphics Forum* 15 (1996), Nr. 3, S. 57–66. – URL `citeseer.nj.nec.com/bro-nielsen96realtime.html`

8 Brown et. al. 2001a Brown, J. ; Sorkin, S. ; Bruyns, C. ; Latombe, K. ; Stephanides, M.: *Real-Time Simulation of Deformable*

Objects: Tools and Application. 2001. – URL citeseer.ist.psu.edu/brown01realtime.html

9 Brown et. al. 2001b Brown, Joel ; Montgomery, Kevin ; Latombe, Jean-Claude ; Stephanides, Michael: A Microsurgery Simulation System. In: *Lecture Notes in Computer Science* 2208 (2001), S. 137ff. – URL citeseer.ist.psu.edu/brown01microsurgery.html

10 Brown et. al. 2001c Brown, Joel ; Sorkin, Stephen ; Latombe, Jean-Claude ; Montgomery, Kevin ; Stephanides, Michael: Algorithmic Tools for Real-Time Microsurgery Simulation. In: *Lecture Notes in Computer Science* 2208 (2001)

11 Bruyns et. al. 2001 Bruyns, C. ; Montgomery, K. ; Wildermuth, S.: *A Virtual Environment for Simulated Rat Dissection*. 2001. – URL citeseer.ist.psu.edu/montgomery01virtual.html

12 Catmull und Rom 1974 Catmull, E. ; Rom, R.: A Class of Local Interpolationg Splines. In: *Barnhill R.E. and R.F. Riesen ed (eds.), Computer Aided Geometric Design* (1974)

13 Çakmak und Kühnapfel 2000 Çakmak, H. ; Kühnapfel, U.: *Animation and simulation techniques for vrtraining systems in endoscopic surgery*. 2000. – URL citeseer.ist.psu.edu/cakmak00animation.html

14 Çakmak et. al. 2002 Çakmak, Hüseyin ; Maasz, H. ; Strauss, G. ; Trantakis, C. ; Nowatius, E. ; Kühnapfel, U.: Modellierung chirurgischer Simulationsszenarien für das Virtuelle Endoskopie Trainingssystem (VEST). In: *Proceedings of CURAC, 1. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie e.V.*, 2002

15 Cotin 2005 Cotin, Stephane: First Steps Towards the Development of an Open Framework for Medical Simulation. In: *Medicine Meets Virtual Reality (MMVR05)*, 2005

16 Cotin et. al. 2000 Cotin, Stéphane ; Delingette, Hervé ; Ayache, Nicholas: A Hybrid Elastic Model allowing Real-Time Cutting Deformations and Force-Feedback for Surgery Training and Simulation. 16 No 8 (2000), S. 437–452

17 d'Aulignac et. al. 2004a d'Aulignac, D. ; Martins, J.A.C. ; Mascarenhas, T. ; Jorge, R.M. N. ; Pires, E.B.: Physical Modeling of the Pelvic Floor Muscles using Shell Elements. In: Neittaanmaki, P. (Hrsg.) ; Rossi, T. (Hrsg.) ; Korotov, S. (Hrsg.) ; Onate, E. (Hrsg.) ; Periaux, J. (Hrsg.) ; Knorzer, D. (Hrsg.): *European Congress on Computational Methods in Applied Sciences and Engineering*. Jyvaskyla, July 2004

18 d'Aulignac et. al. 2004b d'Aulignac, D. ; Martins, J.A.C. ; Mascarenhas, T. ; Jorge, R.M. N. ; Pires, E.B.: A Shell Finite Element Model of the Pelvic Floor Muscles Elements. In: *6th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering*. Madrid, 2004

19 Debunne et. al. 1999 Debunne, Gilles ; Desbrun, Mathieu ; Barr, Alan H. ; Cani, Marie-Paule: Interactive multiresolution animation of deformable models. In: *Eurographics Workshop on Computer Animation and Simulation*, URL `http://www-imagis.imag.fr/Publications/1999/DDBC99`, Sep 1999

20 Debunne et. al. 2000 Debunne, Gilles ; Desbrun, Mathieu ; Cani, Marie-Paule ; Barr, Alan H.: Adaptive Simulation of Soft Bodies in Real-Time. In: *Computer Animation 2000, Philadelphia, USA*, URL `http://www-imagis.imag.fr/Publications/2000/DDCB00`, May 2000, S. 133–144

21 Delingette 1998 Delingette, H.: Towards Realistic Soft Tissue Modeling in Medical Simulation. In: *Proceedings of the IEEE : Special Issue on Surgery Simulation* (1998), April, S. 512–523

22 Delingette und Ayache 2004 Delingette, H. ; Ayache, N.: *Computational Models for the Human Body*. Kap. Soft Tissue Modeling for Surgery Simulation, Elsevier, 2004 (Handbook of Numerical Computing (Ed : Ph. Ciarlet))

23 Delingette et. al. 1999 Delingette, H. ; Cotin, S. ; Ayache, N.: A Hybrid Elastic Model allowing Real Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation. In: *Computer Animation*. Geneva, Switzerland, Mai 26–28 1999

24 Faux und M.J. 1979 Faux, I.D. ; M.J., Pratt: Computational Geometry for Design and Manufacture. In: *Barnhill R.E. and R.F. Riesen ed (eds.), Computer Aided Geometric Design* (1979)

25 Forest et. al. 2002 Forest, C. ; Delingette, Herve ; Ayache, Nicholas: Cutting Simulation of Manifold Volumetric Meshes. In: *Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part II*, Springer-Verlag, 2002, S. 235–244. – ISBN 3-540-44225-1

26 Forest et. al. 2004 Forest, Clément ; Delingette, Hervé ; Ayache, Nicholas: Surface Contact and Reaction Force Models for Laparoscopic Simulation. In: *International Symposium on Medical Simulation*, Juni 2004

27 Ganovelli et. al. 2000 Ganovelli, Fabio ; Cignoni, Paolo ; Montani, Claudio ; Scopigno, Roberto: A Multiresolution Model for Soft Objects supporting interactive cuts and lacerations. In: *Proceedings of Eurographics* Eurographics Association (Veranst.), 2000

28 Gelder 1998 Gelder, Allen V.: Approximate simulation of elastic membranes by triangulated spring meshes. In: *J. Graph. Tools* 3 (1998), Nr. 2, S. 21–42. – ISSN 1086-7651

29 Gibson und Mirtich 1997 Gibson, S.F. ; Mirtich, B.: A Survey of Deformable Modeling in Computer Graphics / MERL – A Mitsubishi Electric Research Laboratory. 1997 (TR-97-19). – Forschungsbericht

30 Grimm et. al. 2002 Grimm, J. ; Schill, M. ; Männer, R. ; Lienemann, A. ; Janssen, U.: Finite Element Simulation of the Intra-Abdominal Cavity. In: *International Congress on Biological and Medical Engineering.* Singapur, December 2002

31 Grimm et. al. 2004 Grimm, J. ; Wagner, C. ; Männer, R.: Interactive Real-time Simulation of the Internal Limiting Membrane. In: *International Symposium on Medical Simulation.* Cambridge, MA, June 2004, S. 153–160

32 Grimm 2000 Grimm, Johannes: *Simulation eines diffus schwellenden Gehirns mit der Methode der finiten Elemente*, Universität Mannheim, Diplomarbeit, 2000

33 Grimm 2005a Grimm, Johannes: Interactive real-time simulation of an endoscopic polyp removal. In: *Medicine Meets Virtual Reality (MMVR05)*, 2005

34 Grimm 2005b Grimm, Johannes: Tearing of membranes for interactive realtime surgical training. In: *Medicine Meets Virtual Reality (MMVR05)*, 2005

35 Huang et. al. 2003 Huang, R. ; Sukumar, N. ; Prévost, J.-H.: Modeling Quasi-Static Crack Growth with the ExtendedFinite Element Method. Part II: Numerical applications. In: *International Journal of Solids and Structures* 40 (2003), S. 7539–7552

36 James und Pai 1999 James, Doug L. ; Pai, Dinesh K.: ArtDefo - Accurate Real Time Deformable Objects. In: Rockwood, Alyn (Hrsg.): *Siggraph 1999, Computer Graphics Proceedings.* Los Angeles : Addison Wesley Longman, 1999, S. 65–72. – URL `citeseer.ist.psu.edu/james99artdefo.html`

37 Körner et. al. 2004 Körner, O. ; Rieger, K. ; Männer, R.: Haptic Display for all Degrees of Freedom of a Simulator for Flexible Endoscopy.

In: *International Symposium on Medical Simulation*. Cambridge, MA, June 2004, S. 161–167

38 Körner und Männer 2002 Körner, Olaf ; Männer, Reinhard: Haptic Display for a Virtual Reality Simulator for Flexible Endoscopy. In: Müller, S. (Hrsg.) ; Stürzlinger, W. (Hrsg.): *8th Eurographics Workshop on Virtual Environments*. Barcelona, May 2002, S. 13–18

39 Körner und Männer 2003 Körner, Olaf ; Männer, Reinhard: Implementation of a Haptic Interface for a Virtual Reality Simulator for Flexible Endoscopy. In: Hannaford, Blake (Hrsg.) ; Tan, Hong (Hrsg.): *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE-VR2003*. Los Angeles, USA, March 2003, S. 278–284

40 Körner 2003 Körner, Olaf: *Entwicklung eines computergesteuerten Trainingssimulators für die Koloskopie mit aktivem Force-Feedback*, Institute for Computational Medicine, University of Mannheim, Dissertation, 2003

41 Kühnapfel et. al. 2000 Kühnapfel, U. ; Cakmak, H. ; Maass, H.: Endoscopic surgery training using virtual reality and deformable tissue simulation. In: *Computers & Graphics* 24 (2000), S. 671–682. – URL `citeseer.nj.nec.com/kuhnapfel00endoscopic.html`

42 Kühnapfel et. al. 1999 Kühnapfel, U. ; Cakmak, H.K. ; Maass, H.: 3D Modeling for Endoscopic Surgery. In: *Proc. IEEE Symposium on Simulation*. Delft University, Delft, NL, Oct 13 1999, S. 22–32. – ISBN: 90-804551-7-2

43 Kühnapfel et. al. 1997 Kühnapfel, U. ; Kuhn, Ch. ; Hübner, M. ; Krumm, H.-G. ; Maass, H. ; Neisius b.: The Karlsruhe Endoscopic Surgery Trainer as an Example for Virtual Realita in Medicla Education. In: *Minimal Invasive Therapy and Allied Technologies (MITAT)* (1997), Nr. 6, S. 122–125

44 Kühnapfel et. al. 1995 Kühnapfel, Uwe ; Krumm, H. ; Kuhn, C. ; Hübner, M. ; Neisius, B.: Endosurgery Simulations with KISMET: A flexible tool for Surgical Instrument Design, Operation Room Planning and VR Technology based Abdominal Surgery Training. In: *Proceedings VR'95 World Conference*, 1995

45 Levinson et. al. 1995 Levinson, S.F. ; Shinagawa, M. ; Sato, T.: Sonoelastic determination of human skeletal muscle elasticity. In: *J. Biomechanics* 28 (1995), Nr. 10, S. 1145–1154

46 Lienemann 1998 Lienemann, A.: An easy approach to functional magnetic resonance imaging of pelvic floor disorder. In: *Techniques in Colorproctology* 2 (1998), Nr. 3

47 Lienemann et. al. 1997 Lienemann, A. ; Anthuber, C. ; Baron, A. ; Kohz, P. ; Reiser, M.: Dynamic MR colpocystorectography assessing pelvic-floor descents. In: *European Radiology 7* (1997), S. 1309–1317

48 Liu et. al. 2003 Liu, Alan ; Tendick, Frank ; Cleary, Kevin ; Kaufmann, Christoph: A survey of surgical simulation: applications, technology, and education. In: *Presence: Teleoper. Virtual Environ.* 12 (2003), Nr. 6, S. 599–614. – ISSN 1054-7460

49 Logan et. al. 1994 Logan, I.P. ; Wills, D.P.M. ; Avis, N.J.: Deformable Objects in Virtual Environments. In: *In Second UK VR-SIG Conference*, 1994, S. 88–103

50 Louchet et. al. 1995 Louchet, Jean ; Provot, Xavier ; Crochemore, David: Evolutionary identification of cloth animation models. In: Terzopoulos, Dimitri (Hrsg.) ; Thalmann, Daniel (Hrsg.): *Computer Animation and Simulation '95*, Springer-Verlag, 1995, S. 44–54. – URL `citeseer.ist.psu.edu/louchet95evolutionary.html`

51 Maciel et. al. 2003 Maciel, A. ; Boulic, R. ; Thalmann, D.: *Deformable Tissue Parameterized by Properties of Real Biological Tissue.* 2003. – URL `citeseer.ist.psu.edu/maciel03deformable.html`

52 Mattox et. al. Mattox, F.T. ; Lucente, V. ; McIntyre, P. ; Miklos, J.R. ; Tomezsko, J.: Abnormal spinal curvature and its relationship to pelvic organ prolapse. In: *Am J Obstet Gynecol* 183, Nr. 6, S. 1381–1383

53 Maurel et. al. 1998 Maurel, W. ; Wu, Y. ; Magnenat-Thalmann, N. ; Thalmann, D.: *Biomechanical Models for Soft Tissue Simulation.* Berlin, Heidelberg, New York : Springer, 1998 (Basic Research Series)

54 Mazarak et. al. 1999 Mazarak, Oleg ; Martins, Claude ; Amanatides, John: Animating Exploding Objects. In: *Graphics Interface*, URL `citeseer.ist.psu.edu/mazarak99animating.html`, 1999, S. 211–218

55 Meißner 1998 Meissner, Joachim: *Biomechanische Simulation mit finiten Elementen*, Universität Mannheim, Diplomarbeit, 1998

56 Mendoza und Laugier 2003 Mendoza, C. ; Laugier, C.: Simulating Cutting in Surgery Applications using Haptics and Finite Element Models. In: *Proceedings of the IEEE Virtual Reality 2003*, IEEE Computer Society, 2003, S. 295. – ISBN 0-7695-1882-6

57 Moes et. al. 1999 Moes, N. ; Dolbow, J. ; Belytschko, T.: A finite element method for crack growth without remeshing. In: *Int. J. Numer. Methods Eng.* 46 (1999), S. 131–150

58 Mollemans et. al. 2004 Mollemans, Wouter ; Schutyser, Filip ; Cleynenbreugel, Johan V. ; Suetens, Paul: Fast Soft Tissue Deformation with Tetrahedral Mass Spring Model for Maxillofacial Surgery Planning Systems. In: Barillot, C. (Hrsg.) ; Haynor, D.R. (Hrsg.) ; Hellier, P. (Hrsg.): *miccai2004* Bd. 3217, Springer, September 2004, S. 371–379

59 Montgomery et. al. 2002 Montgomery, Kevin ; Bruyns, Cynthia ; Brown, Joel ; Sorkin, Stephen ; Mazzella, Frederic ; Thonier, Guillaume ; Tellier, Arnaud ; Lerman, Benjamin ; Menon, Anil: Spring: A General Framework for Collaborative, Real-time Surgical Simulation. In: *Medicine Meets Virtual Reality (MMVR02)*, 2002

60 Montgomery et. al. 2001 Montgomery, Kevin ; Heinrichs, LeRoy ; Bruyns, Cynthia ; Wildermuth, Simon ; Hasser, Christopher ; Ozenne, Stephanie ; Bailey, David: Surgical simulator for hysteroscopy: a case study of visualization in surgical training. In: *Proceedings of the conference on Visualization '01*, IEEE Computer Society, 2001, S. 449–452. – ISBN 0-7803-7200-X

61 Montgomery et. al. 1998 Montgomery, Kevin ; Stephanides, Michael ; Schendel, Stephen ; Ross, Muriel: A case study using the virtual environment for reconstructive surgery. In: *Proceedings of the conference on Visualization '98*, IEEE Computer Society Press, 1998, S. 431–434. – ISBN 1-58113-106-2

62 Mor und Kanade 2000 Mor, Andrew B. ; Kanade, Takeo: Modifying Soft Tissue Models: Progressive Cutting with Minimal New Element Creation. In: *MICCAI*, URL `citeseer.ist.psu.edu/mor00modifying.html`, 2000, S. 598–607

63 Mosegaard 2004 Mosegaard, Jesper: LR-Spring Mass model for Cardiac Surgical Simulation. In: *Medicine Meets Virtual Reality 12*, 2004, S. 256–258

64 Mosegaard 2005 Mosegaard, Jesper: A GPU Accelerated Spring Mass System for Surgical Simulation. In: *Medicine Meets Virtual Reality (MMVR05)*, 2005

65 Müller et. al. 2002 Müller, Matthias ; Dorsey, Julie ; McMillan, Leonard ; Jagnow, Robert ; Cutler, Barbara: Stable real-time deformations. In: *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 2002, S. 49–54. – ISBN 1-58113-573-4

66 Müller et. al. 2001 Müller, Matthias ; McMillan, Leonard ; Dorsey, Julie ; Jagnow, Robert: Real-time simulation of deformation and

fracture of stiff materials. In: *Proceedings of the Eurographic workshop on Computer animation and simulation*, Springer-Verlag New York, Inc., 2001, S. 113–124. – ISBN 3-211-83711-6

67 Müller et. al. 2004 Müller, Matthias ; Teschner, Matthias ; Gross, Markus: Physically-Based Simulation of Objects Represented by Surface Meshes. In: *Computer Graphics International*, IEEE Computer Society, 2004

68 Neff und Fiume 1999 Neff, Michael ; Fiume, Eugene: A Visual Model For Blast Waves and Fracture. In: *Graphics Interface*, URL `citeseer.ist.psu.edu/neff99visual.html`, 1999, S. 193–202

69 Nienhuys und van der Stappen 2001 Nienhuys, Han-Wen ; Stappen, A. F. van der: A Surgery Simulation Supporting Cuts and Finite Element Deformation. In: Niessen, Wiro J. (Hrsg.) ; Viergever, Max A. (Hrsg.): *Medical Image Computing and Computer-Assisted Intervention* Bd. 2208. Utrecht, The Netherlands : Springer-Verlag, October 2001, S. 153–160. – URL `citeseer.ist.psu.edu/505578.html`. – ISBN 3-540-42697-3

70 Nienhuys und van der Stappen 2002 Nienhuys, Han-Wen ; Stappen, A. F. van der: *A Delaunay approach to interactive cutting in Triangulated Surfaces*. 2002. – URL `citeseer.ist.psu.edu/nienhuys02delaunay.html`

71 Nürnberger et. al. 1998 Nürnberger, Andreas ; Radetzky, Arne ; Kruse, Rudolf: A Problem Specific Recurrent Neural Network for the Description and Simulation of Dynamic Spring Models. In: *Proceedings of the IEEE International Joint Conference on Neural Networks, (IJCNN '98)*, URL `citeseer.ist.psu.edu/156338.html`, 1998, S. 468–73

72 O'Brien und Hodgins 1999 O'Brien, James F. ; Hodgins, Jessica K.: Graphical modeling and animation of brittle fracture. In: *Proceedings of ACM SIGGRAPH 1999*, ACM Press/Addison-Wesley Publishing Co., August 1999, S. 137–146. – ISBN 0-201-48560-5

73 Picinbono et. al. 2001 Picinbono, G. ; Delingette, H. ; Ayache, N.: Non-linear and anisotropic elastic soft tissue models for medical simulation. In: *ICRA2001: IEEE International Conference Robotics and Automation*. Seoul Korea, May 2001. – Best conference paper award

74 Picinbono et. al. 2000 Picinbono, G. ; Lombardo, J. ; Delingette, H. ; Ayache, N.: *Anisotropic Elasticity and Force Extrapolation to Improve Realism of Surgery Simulation*. 2000. – URL `citeseer.ist.psu.edu/picinbono00anisotropic.html`

75 Provot 1995 Provot, Xavier: Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In: Davis, Wayne A. (Hrsg.) ; Prusinkiewicz, Przemyslaw (Hrsg.): *Graphics Interface '95*, Canadian Human-Computer Communications Society, 1995, S. 147–154. – URL `citeseer.nj.nec.com/provot96deformation.html`

76 Provot 1997 Provot, Xavier: Collision and self-collision handling in cloth model dedicated to design garments. In: *Graphics Interface '97* (1997), S. 177–189

77 Sanchez et. al. 2001 Sanchez, N.C. ; Tenofsky, P.L. ; Dort, J.M. ; Shen, L.Y. ; Helmer, S.D. ; Smith, R.S.: What is Normal Intra-Abdominal Pressure? In: *The American Surgeon* 6 (2001), S. 243–248

78 Schill et. al. 2000 Schill, M. ; Wagner, C ; Bender, H.-J. ; Männer, R.: Simulator for Intra Ocular Surgery Virtual Reality for Micro-Surgical Education and Training. In: *Proc. 10th Internal Conference On Biomedical Engineering*. Singapur, 2000, S. 554

79 Schill et. al. 1999a Schill, M. ; Wagner, C. ; Hennen, M. ; Jendritza, B. ; Knorz, M.C. ; Bender, H.-J. ; Männer, R.: Virtuelle Realität für die ophthalmochirurgische Ausbildung. In: *12. Kongr. der Deutschen Ophthalmochirurgen*, 1999. – (Abstraktband:65)

80 Schill 1995 Schill, Markus: *Simulation der Flax cerebri unter Verwendung der Methode der finiten Elemente*, Universität Mannheim, Diplomarbeit, 1995

81 Schill 2001 Schill, Markus A.: *Biomechanical Soft Tissue Modeling – Techniques, Implementation and Applications*, University of Mannheim, Dissertation, 2001

82 Schill et. al. 1998 Schill, Markus A. ; Gibson, Sarah F. F. ; Bender, H.-J. ; Männer, R.: Biomechanical simulation of the vitreous humor in the eye using an enhanced chainmail algorithm. In: Colchester, Alan (Hrsg.) ; Wells, William M. (Hrsg.) ; Delp, Scott (Hrsg.): *Medical Image Computing and Computer- Assisted Intervention – MICCAI '98*. Cambribge, MA, USA : Springer, October 1998, S. 679–687

83 Schill et. al. 1999b Schill, Markus A. ; Wagner, Clemens ; Hennen, Marc ; Bender, Hans-Joachim ; Männer, Reinhard: EyeSi – A Simulator for Intra-ocular Surgery. In: Taylor, Chris (Hrsg.) ; Colchester, Alan (Hrsg.): *Medical Image Computing and Computer- Assisted Intervention – MICCAI '99* Bd. 1679. Cambridge; UK : Springer, September 1999, S. 1166–1174

84 Seymour et. al. 2002 SEYMOUR, Neal E. ; GALLAGHER, Anthony G. ; ROMAN, Sanziana A. ; O'BRIEN, Michael K. ; BANSAL, Vipin K. ; ANDERSEN, Dana K. ; SATAVA, Richard M.: Virtual Reality Training Improves Operating Room Performance: Results of a Randomized, Double-Blinded Study. In: *Annals of Surgery* 236 (2002), Nr. 4, S. 458–464

85 Shahram SHAHRAM, Hui Z.: *Simulation of Progressive Cutting on Surface Mesh Model.* – URL `citeseer.ist.psu.edu/621146.html`

86 Smith et. al. 2000 SMITH, Jeffrey ; WITKIN, Andrew ; BARAFF, David: Fast and Controllable Simulation of the Shattering of Brittle Objects. In: *Graphics Interface*, May 2000

87 Sukumar und Prévost 2003 SUKUMAR, N. ; PRÉVOST, J.-H.: Modeling Quasi-Static Crack Growth with the ExtendedFinite Element Method. Part I: Computer Implementation. In: *International Journal of Solids and Structures* 40 (2003), S. 7513–7537

88 Teran et. al. 2003 TERAN, J. ; BLEMKER, S. ; HING, V. Ng T. ; FEDKIW, R.: Finite volume methods for the simulation of skeletal muscle. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, Eurographics Association, 2003, S. 68–74. – ISBN 1-58113-659-5

89 Terzopoulos und Waters 1990 TERZOPOULOS, D. ; WATERS, K.: Physically based facial modeling, analysis and animation. In: *J. Visual. Comp. Anim.* (1990), Nr. 1, S. 73–80

90 Terzopoulos und Fleischer 1988 TERZOPOULOS, Demetri ; FLEISCHER, Kurt: Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In: *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, ACM Press, 1988, S. 269–278. – ISBN 0-89791-275-6

91 Terzopoulos et. al. 1987 TERZOPOULOS, Demetri ; PLATT, John ; BARR, Alan ; FLEISCHER, Kurt: Elastically deformable models. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM Press, 1987, S. 205–214. – ISBN 0-89791-227-6

92 Thalmann und Thalmann 1998 THALMANN, Nadia M. ; THALMANN, Daniel: Virtual Reality Software and Technology. In: *Encyclopedia of Computer Science and Technology, Marcel Dekker* 41 (1998)

93 Vigneron et. al. 2004 VIGNERON, Lara M. ; VERLY, Jacques G. ; WARFIELD, Simon K.: Modelling Surgical Cuts, Retractions, and Resections via Extended Finite Element Method. In: BARILLOT, C. (Hrsg.) ; HAYNOR, D.R. (Hrsg.) ; HELLIER, P. (Hrsg.): *miccai2004* Bd. 3217, Springer, September 2004, S. 311–318

94 Wagner 2003 WAGNER, Clemens: *Virtuelle Realitäten für die chirurgische Ausbildung: Strukturen, Algorithmen und ihre Anwendung*, Institute for Computational Medicine, University of Mannheim, Dissertation, 2003

95 Wagner et. al. 2001 WAGNER, Clemens ; SCHILL, Markus A. ; HENNEN, Marc ; MÄNNER, Reinhard ; JENDRITZA, Bettina ; KNORZ, Michael C. ; BENDER, Hans-Joachim: Virtuelle Realitäten für die augenchirurgische Ausbildung. In: *Der Ophthalmologe* 98 (2001), Nr. 4, S. 409–413

96 Wagner et. al. 2002a WAGNER, Clemens ; SCHILL, Markus A. ; MÄNNER, Reinhard: Collision Detection and Tissue Modeling in a VR-Simulator for Eye Surgery. In: MÜLLER, S. (Hrsg.) ; STÜRZLINGER, W. (Hrsg.): *8th Eurographics Workshop on Virtual Environments*. Barcelona, May 2002, S. 27–36

97 Wagner et. al. 2002b WAGNER, Clemens ; SCHILL, Markus A. ; MÄNNER, Reinhard: Intraocular Surgery on a Virtual Eye. In: *Communications of the ACM* 45 (2002), July, Nr. 7, S. 45–49

98 Webster et. al. 2004a WEBSTER, Roger ; SASSANI, J. ; SHENK, R. ; GOOD, N.: A Haptic Surgical Simulator for the Continuous Curvilinear Capsulorhexis Procedure During Cataract Surgery. In: *Proceedings of the Annual Medicine Meets Virtual Reality Conference*. Newport Beach, California, January 2004, S. 404–406

99 Webster et. al. 2004b WEBSTER, Roger ; SASSANI, J. ; SHENK, R. ; ZOPPETTI, G.: Simulating the Continuous Curvilinear Capsulorhexis Procedure During Cataract Surgery. In: *Proceedings of the Annual IASTED International Conference on Modeling and Simulation*. Marina Del Rey, California, March 2004, S. 262–265

100 Wildermuth et. al. 2001 WILDERMUTH, Simon ; BRUYNS, Cynthia ; MONTGOMERY, Kevin ; BEEDU, Bharath ; MARINCEK, Borut: Patient Specific Surgical Simulation System for Procedures in Colonoscopy. In: *Vision Modeling and Visualization*, 2001, S. 241–248

101 Witkin und Welch 1990 WITKIN, Andrew ; WELCH, William: Fast Animation and Control of Nonrigid Structures, URL `citeseer.ist.psu.edu/witkin90fast.html`, 1990, S. 243–252

102 Wu et. al. 2001 WU, X. ; DOWNES, M.s. ; GOTEKTEKIN, F: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In: *Eurographics Volume 20* Bd. 20, 2001, S. 349–358

103 Wu und Tendick 2004 WU, Xunlei ; TENDICK, Frank: Multigrid Integration for Interactive Deformable Body Simulation. In: *International Symposium on Medical Simulation*. Cambridge, MA, June 2004, S. 92–104

104 Zhuang und Canny 2000 ZHUANG, Y. ; CANNY, J.: *Haptic interaction with global deformations.* 2000. – URL citeseer.ist.psu.edu/zhuang00haptic.html

105 Zhuang und Canny 1999 ZHUANG, Yan ; CANNY, John: Real-time and physically realistic simulation of global deformation, URL citeseer.ist.psu.edu/article/zhuang99realtime.html, 1999, S. 270–270

Sachregister

Adams, Douglas, 1
Ansatzfunktion, 16, 19, 20, 122, 124, 126, 127
Augustine, Norman R., 65

BEM, 20
Bohr, Niels, 105
Boundary-Element-Method, 20

Chain-Mail, 23–24, 31, 32, 35, 38, 74, 76, 80
  Enhanced, 24, 33, 35

Dämpfungsmatrix, 17, 19
Delaunay-Triangulierung, 43
Displacement, 76
Dragnet, 1, 2, 5, 7, 25–36, 38, 71, 74, 76, 80, 85, 90, 102, 105, 106, 108
  Algorithmus, 26
  Konvergenzgeschwindigkeit, 28
  Konzept, 25
  Laufzeit, 29
  Modifikationen, 27
  Vergleich, 31

Endknoten des Risses, 47
Extended Finite Element Method, 19

FDM, 20
Finite-Elemente-Methode, ii, 3, 16–19, 41, 66, 69, 71, 77, 80, 94, 95, 99, 102, 105, 109, 121–127
Finite-Volumen-Methode, 20
Finiten-Differenzen-Methode, 20

genetische Programmierung, 21
Gitterlosen Verfahren, 20

Hooke'sches Gesetz, 17, 43, 113
Hysterese-Effekt, 8, 9

innerer Knoten, 47
Integration
  Euler, 31, 32, 34, 66, 76, 77, 94, 117
    quasi-statisch, 118
  explizite, i, 1, 5, 13–16, 18, 19, 21–23, 31, 32, 34, 37, 42
  implizite, 1, 14, 17, 18
  Mittelpunkt-Methode, 117–118
  Newmark Methode, 19
  Runge-Kutta
    2. Ordnung, 118
    4. Ordnung, 85, 90, 118–119
Integrationen
  explizite, 66, 85, 90, 94, 117, 120
  Houbolt, 76, 94–96, 99, 120
  implizite, 76, 95, 96, 102, 119–120
  Newmark Methode, 119–120

Kantenfeder, 47
Kantenknoten, 47
Kantennachbar, 47
Knotenvariable, 16–19, 121–124, 127
Kopernikus, Nikolaus, 39

Längenkorrektur, 21–22, 27, 29, 31–34, 38, 76
  sortierte, 22, 32–34
Lamé-Koeffizienten, 8, 114, 115

Level-of-Detail, 43
LR-Mass-Spring, 22, 32, 33

Mass-Lumping, 18, 19
Mass-Spring, 1–3, 5, 6, 20–23, 25, 31–34, 36–39, 44, 56, 77, 80, 85, 90, 102, 105, 106, 108
Multiresolution, 43

Neuronale Netze, 21
Nyquist, 10

Partikelsimulationen, 21
Phyxel, 20
Picasso, Pablo, i
Plato, 1
Poisson-Zahl, 100, 112, 115
progressive cut, 42

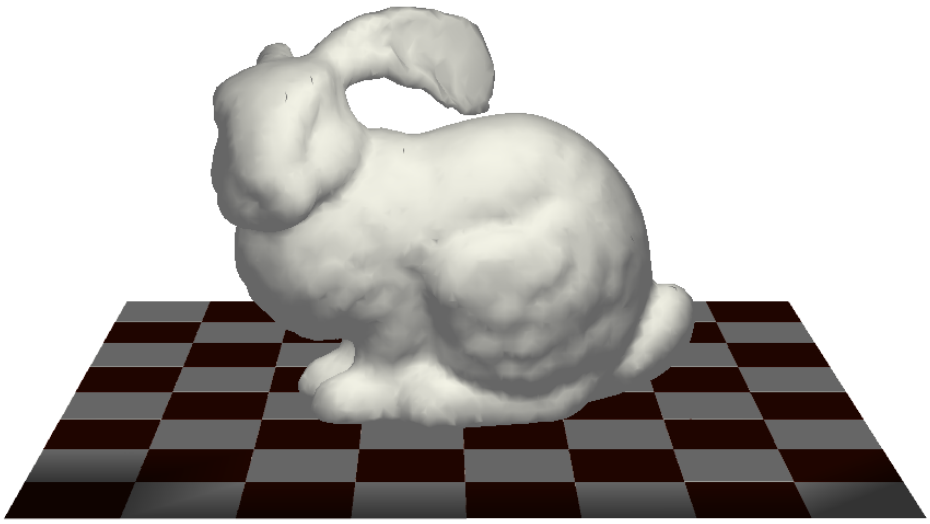
Rayleigh-Dämpfung, 18, 19
Rigid-Body, 18, 19, 40–42, 66, 77, 80
Rissfeder, 47
Risskeim, 47
Rissknoten, 47
Runge-Kutta, 66, 69, 76, 94, 118–119
Rutherford, Ernest, 7

Schermodul, 8, 113, 115
St. Venant-Kirchhoff Elastizität, 23, 112
Steifigkeitsmatrix, 17, 18, 23
Stiffness Warping, 19
String, 1, 25–31, 34, 35, 38, 76, 85
Superelastizität, 21

Tensor-Mass, 18, 23, 37, 42, 105

van de Snepscheut, Jan L. A., 83
Voxel, 40

XFEM, 19



Das virtuelle Versuchskaninchen nach getaner Arbeit.[1]

[1] Ein mit der Methode der Finiten-Elemente simulierter Hase lässt unter dem Einfluss einer Schwerkraft seine Ohren hängen. Damit die Simulation in Echtzeit ablaufen kann, wird der Hase auf einem groben Gitter simuliert. Für die Umsetzung der Simulationsergebnisse auf das feine Visualisierungsgitter sorgt ein Animationsschritt.