

Technische Universität München
Lehrstuhl für Angewandte Mechanik

**Robotergestützte Modellidentifikation und
Simulation von deformierbaren Körpern
für haptische Anwendungen**

Dipl.-Tech. Math. Univ. Gerhard Schillhuber

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. Michael Zäh

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Dr.-Ing. habil. Heinz Ulbrich
2. Univ.-Prof. Dr.-Ing. Horst Baier

Die Dissertation wurde am 17. Dezember 2008 bei der Technischen Universität
München eingereicht und durch die Fakultät für Maschinenwesen am 11. Mai 2009
angenommen.

Meinen Eltern

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Angewandte Mechanik des Instituts für Mechatronik an der Technischen Universität München. Sie wurde im Rahmen des Sonderforschungsbereichs 453 “Wirklichkeitsnahe Telepräsenz und Teleaktion” aus den Mitteln der Deutschen Forschungsgemeinschaft (DFG) gefördert.

Mein herzlicher Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. habil. Heinz Ulbrich für die Möglichkeit an seinem Lehrstuhl promovieren zu dürfen. Sein entgegengebrachtes Vertrauen und seine stete Förderung und Unterstützung haben erheblich zum Gelingen dieser Arbeit beigetragen. Durch die von ihm gewährten Freiräume und übertragenen Aufgaben konnte ich viele wertvolle Erfahrungen sammeln und machten meine Lehrstuhljahre zu einer schönen und lehrreichen Zeit.

Bei Herrn Prof. Dr.-Ing. Horst Baier bedanke ich mich für sein Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens. Außerdem danke ich Herrn Prof. Dr.-Ing. Michael F. Zäh für die Übernahme des Prüfungsvorsitzes.

Allen Mitarbeitern des Lehrstuhls gilt mein Dank für die große Hilfsbereitschaft und das freundliche Arbeitsklima. Die zahlreichen Diskussionen auf unterschiedlichen Fachgebieten und die daraus gewonnenen Erkenntnisse haben einen wesentlichen Beitrag zur Entstehung dieser Arbeit geleistet. Zugleich bedanke ich mich bei der tatkräftigen Unterstützung der Werkstattmitarbeiter bei der Realisierung des Versuchstands.

Außerdem gilt mein Dank den Kollegen des SFB 453 für die gute Zusammenarbeit, insbesondere Jordi Artigas, Stella Clarke, Philip Kremer, Hermann Mayer und Helena Pongrac. Zusätzlich mochte ich mich bei Joachim Georgii für die Diskussionen zur korotierten Finite-Elemente-Methode bedanken und bei Max Kriegleder für die Unterstützung bei der Objektrekonstruktion.

Für die kritische Durchsicht des Manuskripts und die vielen konstruktiven Kommentare danke ich den Kollegen Thomas Buschmann, Bastian Esefeld, Keita Ono und Roland Zander. Die Anregungen haben zu einer verbesserten Darstellung des Themas beigetragen.

Abschließend gilt mein besonderer Dank meiner Familie und meinen Lieben für ihre Geduld und Vertrauen. Herzlich danke ich meinen Eltern für die Unterstützung während der Zeit meiner Ausbildung und Promotion.

München, im Mai 2009

Gerhard Schillhuber

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einführung und Problemstellung	1
1.2	Stand der Forschung	2
1.3	Ziel und Aufbau der Arbeit	6
2	Modellierung deformierbarer Körper	8
2.1	Kontinuumsmechanik	8
2.2	Diskretisierung mit der Finite-Elemente-Methode	21
2.2.1	Nichtlineare Finite-Elemente	26
2.2.2	Lineare Finite-Elemente	28
2.2.3	Korotierte Finite-Elemente	29
2.2.4	Tetraederelemente	31
2.2.5	Vergleich der Ansätze	32
3	Experimentelle Modellidentifikation	34
3.1	Geometrische Modellbildung	34
3.1.1	Computer Vision	35
3.1.2	3D Rekonstruktion	43
3.1.3	Experimente	44
3.2	Identifikation der Materialparameter	51
3.2.1	Optimierung der Materialparameter	51
3.2.2	Experimente	52
4	Simulation von weichen Kontakten	56
4.1	Simulationsarten	57
4.1.1	Statische Simulation	58
4.1.2	Dynamische Simulation	60
4.2	Interaktionsarten	68
4.3	Numerische Lösungsverfahren	69
4.4	Rechenzeitvergleich	72
5	Haptisches Rendering	77
5.1	Menschliche Wahrnehmung	78
5.2	Glättung von diskretisierten Kraftsignalen	79
5.3	Experimente und Ergebnisse	84
6	Kraftprädiktion in der Telepräsenz	89
6.1	Kraftprädiktion zur Kompensation von Latenzzeiten	91
6.2	Experimente und Ergebnisse	92

7 Zusammenfassung und Ausblick	96
7.1 Zusammenfassung	96
7.2 Ausblick	97
A Nomenklatur	99
Literaturverzeichnis	102

1 Einleitung

1.1 Einführung und Problemstellung

Die Simulation von Systemen mit deformierbaren Körpern wird in zahlreichen Fachgebieten eingesetzt. So lassen reale Systeme oft erst eine Aussage über Dynamik, Schwingungsverhalten und Festigkeit zu, wenn man Komponenten als flexibel und verformbar annimmt. Insbesondere in der Mechatronik, wo der Trend zu immer leichteren und schnelleren Maschinen geht, muss bei der Entwicklung, Auslegung und Regelung auf elastische Komponenten geachtet werden. In den letzten Jahrzehnten wurden viele Werkzeuge zur Simulation solcher Systeme entworfen und erfolgreich eingesetzt. Die Genauigkeit der Modelle und der experimentelle Abgleich stehen dabei bei industriellen Fragestellungen im Vordergrund, während die Rechenzeit oftmals nur ein beiläufiger Aspekt ist. Im Gegensatz dazu wurden in der Computergrafik Modelle entwickelt, die sehr schnell eine Deformation eines Körpers visualisieren können. Diese basieren aber oftmals nicht auf physikalischen Grundlagen. Die Verbindung dieser entgegengesetzten Ansätze, physikalische und genaue Modelle auf der einen Seite und schnelle, sogar interaktive Simulationen auf der anderen Seite, ist ein zur Zeit besonders aktives Forschungsgebiet. Der Bedarf solcher Simulation zeigt sich in zahlreichen Anwendungen.

Ein Teil dieser Arbeit entstand aus dem Teilprojekt M7 „Kraft- und Bewegungsprädiktion bei Teleoperation mit weichen Kontakten“ des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Sonderforschungsbereichs 453 „Wirklichkeitsnahe Telepräsenz und Teleaktion“. Die Ideen und Ansätze sind daher durch Telepräsenzsysteme motiviert, aber nicht darauf beschränkt. Die Telerobotik bietet die Möglichkeit, Arbeiten in für den Menschen nicht erreichbaren Umgebungen durchzuführen. Die möglichen Gründe hierfür sind vielfältig. So kann es zu teuer oder zu aufwendig sein, den Menschen in eine bestimmte Umgebung zu bringen, wie zum Beispiel für Arbeiten im Weltraum. Oder falls die Umgebung nicht weit entfernt ist, kann sie dennoch ein hohes Risiko für den Menschen darstellen, wie beispielsweise die Entschärfung einer Bombe oder die Arbeiten in radioaktiven Umgebungen. Manche Szenarien machen durch eine Größen- oder Kräfteskalierung den Einsatz eines Roboters notwendig, wie zum Beispiel die Mikromontage. Gemeinsam haben die genannten Anwendungen, dass der Mensch mit seinem Eingabegerät vom Roboter räumlich getrennt ist. Die anfängliche Idee war, eine Simulation der entfernten Teleoperatorumgebung in das Telepräsenzsystem zu integrieren. Mit dieser Simulation wird die Möglichkeit eröffnet, auf beliebige Signale und Größen des Systems zuzugreifen. So kann eine parallel ablaufende Simulation für eine Kraftprädiktion genutzt werden, um störende Zeitverzögerungen durch die Übertragungsstrecke zu

kompensieren. Da der Teleoperator zumeist in einer unbekanntem Umgebung agiert und daher ein Modell vorab nicht bekannt ist, muss in einer Identifikationsphase das Simulationsmodell erzeugt und parametrisiert werden. Greift oder manipuliert der Teleoperator weiche Objekte in der entfernten Umgebung, so ist eine Modellierung von deformierbaren Körpern erforderlich. Diese Simulationen haben die oben genannten Anforderungen der Echtzeitfähigkeit und der Genauigkeit. Über Schnittstellen muss es möglich sein, mit der Simulation zu interagieren und Daten auszutauschen. Eine entscheidende Rolle spielt die Übereinstimmung der Simulation mit der Realität um eine Kraftprädiktion zu ermöglichen. Das Simulationsmodell muss daher mit experimentellen Daten erzeugt werden und ein Abgleich mit den Messungen ist notwendig.

Eine weitere Anwendung der echtzeitfähigen, deformierbaren Simulationsmodelle findet sich im haptischen Rendering einer virtuellen Szene. Dabei wird über ein haptisches Gerät eine Krafrückkopplung zum Benutzer generiert. So ist es zum Beispiel möglich bei einer Eingabebewegung eine virtuelle Trägheit zu spüren oder gegen virtuelle Wände zu stoßen. Haptisches Rendering wird unter anderem in chirurgischen Trainingssimulatoren eingesetzt, in denen reale Objekte wie Organe und Gewebe schnell und genau simuliert werden müssen. Die Komplexität steigt, wenn Verformungen im Zusammenspiel mit Kontakten berücksichtigt werden müssen.

Außerhalb der wissenschaftlichen Anwendungen gibt es gleichzeitig einen steigenden Bedarf in der Unterhaltungsindustrie an realistischen Modellen, wie zum Beispiel in Animationsfilmen oder Computerspielen. Hier steht meist die Visualisierung im Mittelpunkt und es werden bewusst physikalische Gesetze verletzt, um beeindruckende Effekte zu erzeugen.

1.2 Stand der Forschung

In dieser Arbeit werden Erkenntnisse aus mehreren Forschungsgebieten aufgegriffen. Dies sind unter anderem aus dem ingenieurwissenschaftlichen Bereich des Maschinenbaus die Kontinuumsmechanik zur Modellierung von deformierbaren Körpern. Die Ansätze zur Echtzeitberechnung, zum haptischen Rendering und zur 3D Rekonstruktion stammen aus der Computergrafik und der Computer Vision aus dem Bereich der Informatik. Existierende Verfahren aus dem mathematisch-naturwissenschaftlichen Bereich finden sich in der numerischen Aufbereitung der Algorithmen zur Lösung der Gleichungssysteme. Ein aktueller Stand der Forschung zu den einzelnen Fragestellungen wird im Folgenden zusammengefasst.

Deformierbare Körper In vielen industriellen Fragestellungen werden Modelle für elastische Komponenten zur Beschreibung von Phänomenen in Maschinen eingesetzt. Der Forschungsbereich, der sich mit linearen und nichtlinearen Schwingungssystemen und mit deren experimenteller Identifikation beschäftigt, ist umfassend und kann hier nur beispielhaft behandelt werden. Ein umfangreiches Standardwerk zum

Thema der Schwingungsprobleme und Modalanalyse ist von HARRIS [57]. Einige Beispiele von Schwingungen und deren Regelung im Bereich der Maschinendynamik und Rotoren sind in den Büchern von ULBRICH [119], BREMER [19] und PFEIFFER [98] aufgeführt. Die Berücksichtigung von elastischen Bauteilen in Robotern findet in den Arbeiten von BREMER [20] und THÜMMEL [117] statt.

Eine sehr gute Zusammenfassung über die Simulationsansätze von deformierbaren Körpern, insbesondere im Zusammenhang mit Mehrkörpersystemen, bietet der Übersichtsartikel von WASFY [122] mit 877 Referenzen. Neben zahlreichen Konferenz- und Zeitschriftenbeiträgen existieren fast ebenso viele Lehr- und Fachbücher, wie von BATHE [14], WRIGGERS [123], ZIENKIEWICZ [125], BELYTSCHKO [16], SCHWERTASSEK [108], SHABANA [109]. Die Bücher von PFEIFFER [99] und WRIGGERS [124] behandeln ausführlich die Kontaktmechanik bei starren und deformierbaren Körpern.

Neben den wissenschaftlich-industriellen Fragestellungen werden deformierbare Körper auch in der Computergrafik beschrieben und dargestellt. Hierbei haben die Schnelligkeit und die Visualisierung der Deformation eine höhere Priorität als die physikalische Exaktheit. So gibt es eine Reihe von Modellierungsansätzen, die nicht auf physikalischen Grundlagen basieren. In den Übersichtsartikeln von NEALEN [88] und GIBSON [54] werden die Ansätze zur Beschreibung von deformierbaren Körpern in der Computergrafik zusammengefasst dargestellt. Dabei fasst GIBSON die Forschungen von Beginn der Computergrafik bis zum Jahr 1997 zusammen und NEALEN die Entwicklungen der letzten 10 Jahre. Alle diese Arbeiten unterscheiden sich in den Ansätzen der Modellierung, Zeitintegration oder Berechnungsvorschriften und Algorithmen. Ein anfänglich beliebtes Verfahren war die Beschreibung des deformierbaren Körpers mit einem Feder-Masse-System. Dies ist einer der einfachsten Ansätze, der aber immer noch gerne eingesetzt wird, insbesondere da sich solche Systeme leicht parallelisieren und somit auf aktuellen Grafikprozessoren sehr effizient berechnen lassen, wie GEORGII [51] zeigt. GIBSON [53] stellt das sogenannte *Chain-Mail*-Verfahren vor, das die Verformung eines Objekts ähnlich einer Verbindung von einzelnen Kettengliedern berechnet. Eine der ersten Veröffentlichungen in der Computergrafik, die einen deformierbaren Körper mit kontinuumsmechanischen Gesetzen beschreibt, ist von TERZOPOULOS [116]. Seit dem entwickelt sich dieses Fachgebiet unter Berücksichtigung neuer Modellierungsansätze, Lösungsstrategien, numerischer Verfahren und Rechnerarchitekturen beständig weiter. BRO-NIELSEN [21] wählt einen linearen Finite-Elemente-Ansatz und die auftretende Systemmatrix wird in einer Vorberechnung invertiert, wodurch eine echtzeitfähige Simulation ermöglicht wird. In der Arbeit von JAMES [66] wird die Boundary-Element-Method zur Beschreibung der Deformationen verwendet. GRINSPUN [56] verfeinert den deformierbaren Körper lokal adaptiv, um Bereiche mit großen Verformungen genauer auflösen zu können als Bereiche mit kleinen Verformungen. GEORGII [50] löst die diskretisierten Gleichungen mit einem Mehrgitteralgorithmus. TAYLOR [115] implementiert eine nichtlineare Finite-Elemente-Methode mit einem expliziten Zeitintegrationsalgorithmus auf dem Grafikprozessor.

Die Modellierung mit dem korotierten Finite-Elemente-Ansatz erfolgt in der Com-

putergrafik erst seit kurzem, beispielsweise durch MÜLLER [86, 87], ETZMUSS [45], HAUTH [59, 60], NESME [89]. Die korotierte Finite-Elemente-Methode wird auch von GEORGII [49, 52] diskutiert und in Kombination mit einem Mehrgitterverfahren verwendet, um statische und dynamische Modelle mit vielen Elementen zu visualisieren.

Haptisches Rendering Die obigen vorgestellten Arbeiten aus der Computergrafik befassen sich hauptsächlich mit der schnellen Darstellung des deformierbaren Körpers, dem visuellen Rendering. Eine Erweiterung davon ist das haptische Rendering, bei dem die Simulation auch eine Krafrückkopplung besitzt. Von SALISBURY [102] wird als haptisches Gerät das PHANToM vorgestellt. Da der Mensch Kräfte zeitlich genauer auflösen kann als Bilder, sind die Zeitanforderungen beim haptischen Rendering entsprechend größer als beim visuellen Rendering. So wird versucht beim haptischen Rendering eine Rate von 1000 Hz zu erreichen, während bei der Visualisierung 30 Hz ausreichend sind. Die existierenden Techniken beim haptischen Rendering werden in den Artikeln von LAYCOCK [74] und BASDOGAN [13] ausführlich vorgestellt. Aufgrund der größeren Zeitanforderungen beim haptischen Rendering, werden Simulatoren oft als Multirate-Systeme aufgebaut, das heißt die Simulatoren besitzen intern unterschiedliche Taktraten für Grafik und Haptik. Der regelungstechnische Entwurf solcher Systeme wird in den Arbeiten von BARBAGLI [10, 11], CAVUSOGLU [22], DURIEZ [39], LEE [75] vorgestellt.

Viele Ansätze für deformierbare Körper aus der Computergrafik werden um die Haptik erweitert. So wird in der Arbeit von HIROTA [64] eine haptische Simulation vorgestellt, die auf der linearen Finite-Elemente-Methode basiert und die Systemmatrix in einer Vorabrechnung invertiert. JAMES [67] kombiniert die Boundary-Element-Method mit der Haptik und von DEBUNNE [34] wird eine lokale adaptive Verfeinerung von Bereichen mit großen Verformungen ebenfalls mit einer Krafrückkopplung erweitert. MAHVASH [82] präsentiert ein Verfahren zur Berechnung des haptischen Kontakts mit nichtlinearen Modellen deformierbarer Körper. Eine Koordinatenreduktion von nichtlinearen Modellen führt BARBIC [12] durch. Kontakte mit Reibung werden von DURIEZ [40] berücksichtigt und SAUPIN [103] löst unilaterale haptische Kontakte über ein lineares Komplementaritätsproblem. Ein Anwendungsbeispiel für das haptische Rendering mit deformierbaren Körpern sind Trainingssimulatoren für die Chirurgie, wie sie von COTIN [28], KÜHNAPFEL [70], ALLARD [3] vorgestellt werden. Für die chirurgischen Simulatoren werden von COTIN [29], NIENHUYS [93, 94], MAHVASH [81], JERABKOVA [68] auch Modelle entwickelt, die Manipulationen wie Schneiden ermöglichen. Das Stechen mit einer Nadel wird von DIMAIO [37, 38] berücksichtigt. Ein experimenteller Abgleich von Stech- und Schneidevorgängen wird in den Arbeiten von HING [62] und CHANTHASOPEEPHAN [23] gegeben.

Modellidentifikation Um einen deformierbaren Körper simulieren zu können, wird ein Modell benötigt. Dabei sind die relevanten Modellparameter die Geometrie und die Materialparameter. Ist die Geometrie nicht vorab bekannt, wie zum Beispiel aus

CAD-Daten, muss sie experimentell bestimmt werden. Es gibt unterschiedliche Methoden um die Oberfläche eines Objekts zu erfassen. Diese können in kontaktlose und kontaktbehaftete Messverfahren eingeteilt werden. Die kontaktlosen klassifizieren sich in reflektive und transmissive Verfahren, wie beispielsweise die Computertomographie. Die reflektiven Verfahren können in nicht-optische und optische Verfahren aufgeteilt werden. Die Klasse der nicht-optischen Verfahren beinhaltet unter anderem das Sonar oder Mikrowellenradar. Alle kamerabasierten Verfahren, wie zum Beispiel eine Stereokamera, zählen zu den optischen Verfahren.

Das Fachgebiet des Computersehens (Computer Vision) beschäftigt sich mit der Frage der Erzeugung von dreidimensionalen Informationen aus zweidimensionalen Bildern. In den Büchern von SCHREER [106], FAUGERAS [47], FORSYTH [48] und HARTLEY [58] werden die Grundlagen und Lösungsansätze ausführlich aufbereitet. Eine zentrale Frage bei der 3D Rekonstruktion ist das Korrespondenzproblem, bei dem korrespondierende Punkte in verschiedenen Ansichten des gleichen Objekts gefunden werden müssen. Es gibt Algorithmen, wie zum Beispiel von LOWE [77], die einzelne Merkmale in den Bildern extrahieren und zuordnen können. Diese Algorithmen werden auch in Programmen zur Panoramaerzeugung eingesetzt, wo mehrere Bilder mit überlappenden Randbereichen aneinandergesetzt werden. Ein Vergleich der Algorithmen, die nicht mit Merkmalen, sondern eine Disparitätskarte des gesamten Bildes erzeugen, führt SCHARSTEIN [104] durch. In dem quelloffenen Programmpaket OPENCV ist der Algorithmus von BIRCHFIELD [17] implementiert. Von LUCAS [79] und BOUGUET [18] wird eine weitere Möglichkeit zur Berechnung der Verschiebung der Punkte zwischen den Bildern mit Hilfe des optischen Flusses präsentiert. Eine Übersicht über die Verfahren zum Verfolgen von 3D Objekten ist in dem Artikel von LEPETIT [76] gegeben. Nach dem Lösen des Korrespondenzproblems muss aus der entstandenen Punktwolke eine Oberfläche rekonstruiert werden. Zahlreiche Methoden werden im Übersichtsartikel von FABIO [46] zusammengefasst. Ausgewählte Ansätze werden in den Arbeiten von EDELSBRUNNER [42], HOPPE [65], TURK [118], CURLESS [30], AMENTA [4] diskutiert. Die Erzeugung eines Volumennetzes aus einer Oberflächenbeschreibung wird in der Arbeit von SI [111] beschrieben und in dem Programm [110] zur Verfügung gestellt.

Die Erzeugung von Kraftmodellen und deren Materialparameteridentifikation anhand von Messungen wird in zahlreichen Arbeiten behandelt. Mit den Experimenten von MAASS [80], DAULIGNAC [31] und KAUER [69] werden Modelle von Oberschenkeln und menschlichen Gewebe für Chirurgesimulatoren erstellt. Als Messverfahren kommen dabei Ultraschall, Saugrohre und auch Roboter zum Einsatz. Weitere Beiträge, die ebenfalls einen Roboter mit Kraftsensorik zum Abtasten eines Objekts verwenden, sind von PAI [95], LANG [71–73], SCHONER [105], ERICKSON [44] und MAHVASH [83].

Telerobotik Ein zentrales Problem der Telerobotik ist die Zeitverzögerung durch die Übertragungsstrecke. Sind die Zeitverzögerungen zwischen dem menschlichen Operator und dem Teleroboter in der entfernten Umgebung zu groß, kann dies

zu Instabilitäten des gesamten Systems führen. Es gibt regelungstechnische Ansätze von ANDERSON [6], NIEMEYER [90,91], ARTIGAS [7], [9], TANNER [114] und HIRCHE [63] zur Stabilisierung des Systems mittels passivitätsbasierter Regler und Wave-Variablen. Das Konzept der Wave-Variablen, die eine Linearkombination aus Kraft und Geschwindigkeit sind, wird von NIEMEYER [92] zusammengefasst. Die Anwendung in Testsystemen auf der Erde und im Weltall wird in den Arbeiten von ARTIGAS [8] und PREUSCHE [100,101] gezeigt. Weitere Ansätze, die insbesondere Kontaktsituationen stabilisieren, wurden von CORTESAO [27], PARK [97] und MITRA [85] vorgestellt.

Numerik Die Verfahren der numerischen Mathematik zur Lösung von linearen und nichtlinearen Gleichungssystemen, Zeitintegration, dünnbesetzte Matrizen und Datenstrukturen sind vielfältig. Etablierte Verfahren werden beispielsweise in den Numerikbüchern von STOER [112,113], DEUFLHARD [35,36], SCHWARZ [107] und DAVIS [32] behandelt. Ebenfalls bieten die FEM-Fachbücher von BATHE [14] und WRIGGERS [123] bereits für die Verformungsberechnung angepasste Einführungen und Zusammenfassungen.

1.3 Ziel und Aufbau der Arbeit

Das Ziel dieser Arbeit ist die interaktive Simulation deformierbarer Körper für haptische Anwendungen. Um ein geschlossenes Vorgehen zu erhalten, das insbesondere, aber nicht ausschließlich, für Telepräsenzsysteme geeignet ist, wird diese Idee systematisch ausgebaut, von der experimentellen Identifikation bis hin zur Anwendung.

An die Simulation der deformierbaren Körper werden dabei die folgenden Anforderungen gestellt. Es soll möglich sein, kleine und große Verformungen realistisch abzubilden, was bereits in der Modellierungsphase durch entsprechende nichtlineare Terme berücksichtigt werden muss. Außerdem sollen sowohl statische als auch dynamische Simulationen erfolgen können. Um die Simulation als Modul innerhalb eines größeren Systems integrieren zu können, müssen Schnittstellen zum Datenaustausch definiert werden.

Es wird davon ausgegangen, dass eine Modellbeschreibung zu Beginn nicht vorhanden ist und dass ein Objekt zuerst in einer Lernphase experimentell identifiziert werden muss. Daher wird ein Verfahren vorgestellt, das in einem ersten Schritt die Geometrie eines Objekts erfasst und in einem zweiten Schritt die Materialparameter bestimmt. Die Messungen erfolgen dabei mit einem Robotersystem, das mit einer Kamera und einem Kraft-Momentensensor ausgestattet ist.

Ist die Modellbeschreibung vollständig, soll eine möglichst schnelle Auswertung der Systemgleichungen erfolgen. Je nach Simulationsanforderung werden hierfür aufbereitete effiziente Algorithmen bereitgestellt. Diese Algorithmen kommen in zwei Anwendungen zum Einsatz: dem haptischen Rendering und der Kraftprädiktion in

einem Telepräsenzsystem. Für die Krafrückkopplung wird zudem ein Ansatz entwickelt, dem Menschen diskretisierte Kräfte als glattes Kraftsignal darstellen zu können.

Der Aufbau der Arbeit ist im Bild 1.1 skizziert. Zunächst werden in Kapitel 2 die theoretischen Grundlagen aus der Kontinuumsmechanik vorgestellt und die Diskretisierung mittels der Finite-Elemente-Methode. In Kapitel 3 wird ein experimentelles Vorgehen zur Modellidentifikation vorgestellt, das speziell für Roboter und Telepräsenzsysteme geeignet ist. Anschließend werden in Kapitel 4 die numerischen Algorithmen für eine schnelle Berechnung der deformierbaren Körper präsentiert. Die erste Anwendung dieser Algorithmen erfolgt in Kapitel 5 im haptischen Rendering. Die zweite Anwendung ist die Kraftprädiktion in Telepräsenzsystemen, die in Kapitel 6 vorgestellt wird. Das Kapitel 7 schließt die Arbeit mit Vorschlägen zur Fortführung und Erweiterung der Arbeit ab.

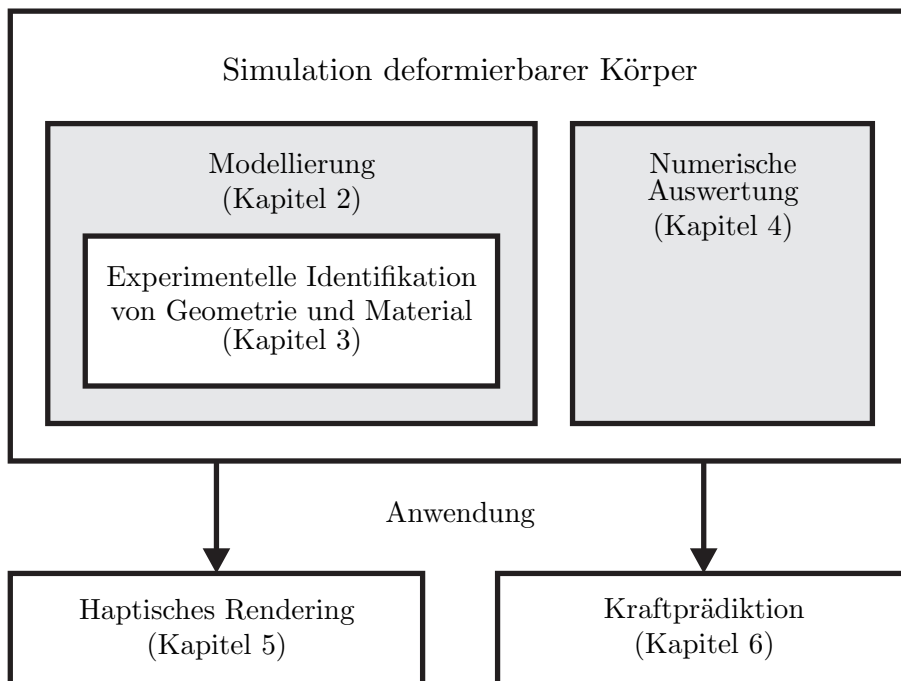


Bild 1.1: Aufbau der Arbeit

2 Modellierung deformierbarer Körper

Es gibt eine Vielzahl von Ansätzen um deformierbare Körper zu modellieren und anschließend zu simulieren. Dabei ist die Modellierung die Beschreibung des Körpers mit dem Ziel, mit Gesetzmäßigkeiten gewünschte Eigenschaften zu erfassen und darzustellen. Die Simulation ist die Auswertung der erzeugten Modelle. Die populärsten Ansätze zur Modellierung von elastischen Bauteilen und deformierbaren Körpern sind Feder-Masse-Systeme, modale Systeme und Finite-Elemente. Jeder Ansatz hat Vor- und Nachteile bei der Beschreibung von einzelnen Effekten und dem damit verbundenen Rechenaufwand. Modale Systeme beinhalten ausgewählte Eigenformen des deformierbaren Körpers, mit denen die Gesamtverformung beschrieben wird. Dadurch besitzt das Modell eine relativ geringe Anzahl von Freiheitsgraden und ist daher auch schnell berechenbar. Der Nachteil ist, dass sich nichtlineare Effekte, wie große Verformungen oder nichtlineares Materialverhalten, nur ungenau darstellen lassen. Die Beschreibung dieser Effekte erfolgt meist mit der Finite-Elemente-Methode. Diese erfordert aufgrund der hohen Zahl der Freiheitsgrade jedoch einen großen Rechenaufwand. In der vorliegenden Arbeit wird aufgrund der höheren Flexibilität des Ansatzes die Finite-Elemente-Methode verwendet. So können geometrische Nichtlinearitäten beschrieben werden, aber auch durch eine lineare Systembetrachtung eine echtzeitfähige Auswertung erzielt werden. Diese Methodik ermöglicht eine Skalierung bezüglich Rechenzeit und Genauigkeit der Simulation.

Im Folgenden werden die funktionellen Zusammenhänge mit Hilfe der Kontinuumsmechanik hergeleitet. Die anschließende Diskretisierung des Körpers erfolgt mit der Finite-Elemente-Methode. Für eine ausführlichere Darstellung der Theorie wird auf die zahlreichen Lehr- und Fachbücher für die Finite-Elemente-Methode und Kontinuumsmechanik verwiesen [14], [125], [123], [96], [16], [120].

2.1 Kontinuumsmechanik

In der Kontinuumsmechanik wird der Zusammenhang zwischen dem Bewegungszustand und den Spannungen eines Körpers formuliert. Im Folgenden werden die grundlegenden Begriffe und Größen definiert, die später für die numerische Auswertung und Simulation erforderlich sind.

In der Herleitung erfolgt die Notation, wie in der Kontinuumsmechanik üblich, mit Tensoren und in Indexschreibweise. Für eine rechentechnische Umsetzung wird später anstelle der Tensornotation eine Matrixnotation verwendet. Die verwendete Nomenklatur folgt dabei dem Buch von PARISCH [96] und dem Skript von WALL [121] und ist im Anhang A zusammengefasst.

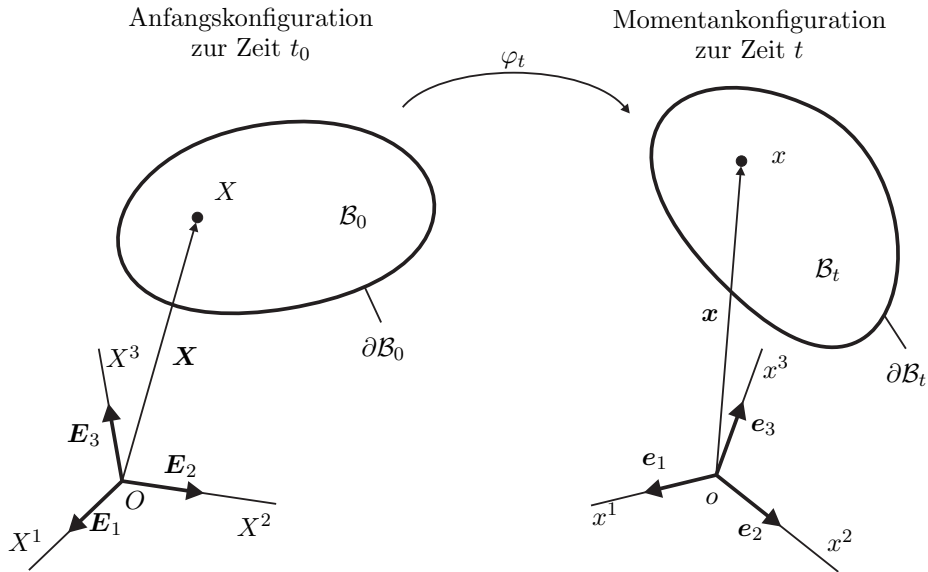


Bild 2.1: Körper in Ausgangskonfiguration und aktueller Konfiguration

Konfigurationen Zunächst wird die zeitliche Abfolge von Konfigurationen eines Körpers im Raum betrachtet, wie im Bild 2.1 skizziert. Eine Konfiguration \mathcal{B}_t des Körpers zum Zeitpunkt t ist das Gebiet, welches er im Raum zu dieser Zeit ausfüllt. Diese Konfiguration \mathcal{B}_t wird auch *momentane* oder *aktuelle Konfiguration* genannt. Die Konfiguration \mathcal{B}_0 zum Zeitpunkt t_0 wird *initiale*, *Anfangs-*, *Referenz-* oder *Ausgangskonfiguration* genannt. Der Rand des Körpers zum Zeitpunkt t_0 ist $\partial\mathcal{B}_0$ und zur Zeit t lautet er $\partial\mathcal{B}_t$. Das Koordinatensystem zum Zeitpunkt t_0 ist $\{O, \mathbf{E}_A\}$ und zum Zeitpunkt t lautete es $\{o, \mathbf{e}_a\}$. Die beiden sind kovariante Basen und die dazugehörigen kontravarianten Basen sind $\{\mathbf{E}^B\}$ und $\{\mathbf{e}^b\}$. Für die Indizes gilt $A, a, B, b = 1, 2, 3$.

Für eine erleichterte Lesbarkeit der mathematischen Formulierungen werden Größen in den verschiedenen Konfigurationen auch unterschiedlich notiert. So sind Größen mit Großbuchstaben in der Ausgangslage definiert und Größen mit Kleinbuchstaben in der Momentanlage. Die Repräsentation von Größen in der Ausgangskonfiguration wird *materielle* oder *Lagrange'sche Betrachtungsweise* genannt und die Darstellung von Größen in der Momentankonfiguration *räumliche* oder *Euler'sche Betrachtungsweise*.

Anstelle unabhängiger Basen ist es im Weiteren ausreichend, ein einheitliches, festes, kartesisches Koordinatensystem zu wählen. Es gilt daher $o = O$, $\mathbf{e}_a = \mathbf{E}_A$ und $\mathbf{e}^b = \mathbf{E}^B$. Bei der Notation der weiteren Größen wird die Unterscheidung der Basen mit Groß- und Kleinbuchstaben und Stellung der Indizes fortgeführt.

Ein Materialpunkt $X \in \mathcal{B}_0$ wird beschrieben über den Ortsvektor \mathbf{X} und hat die Darstellung

$$\mathbf{X} = \overrightarrow{OX} = X^1 \mathbf{E}_1 + X^2 \mathbf{E}_2 + X^3 \mathbf{E}_3 = X^A \mathbf{E}_A \quad (2.1)$$

In der Momentankonfiguration wird ein Materialpunkt $x \in \mathcal{B}_t$ mit dem Ortsvektor

\mathbf{x} beschrieben und besitzt die Darstellung

$$\mathbf{x} = \overrightarrow{ox} = x^1 \mathbf{e}_1 + x^2 \mathbf{e}_2 + x^3 \mathbf{e}_3 = x^a \mathbf{e}_a \quad (2.2)$$

Bewegung Als nächstes erfolgt eine mathematische Beschreibung des Zusammenhanges zwischen den Konfigurationen. Ein Materialpunkt, der sich zum Zeitpunkt t_0 an der Stelle $X \in \mathcal{B}_0$ befindet, befindet sich zum Zeitpunkt t an der Stelle $x \in \mathcal{B}_t$. Die *Bewegung* φ_t des Körpers wird definiert als die Abbildung der Anfangskonfiguration \mathcal{B}_0 auf die Momentankonfiguration \mathcal{B}_t :

$$\varphi_t : \begin{cases} \mathcal{B}_0 \rightarrow \mathcal{B}_t \\ \mathbf{X} \mapsto \mathbf{x} = \varphi(\mathbf{X}, t) = \varphi_t(\mathbf{X}) \end{cases} \quad (2.3)$$

Ferner existiert eine eindeutige Umkehrabbildung der Bewegung

$$\varphi_t^{-1} : \begin{cases} \mathcal{B}_t \rightarrow \mathcal{B}_0 \\ \mathbf{x} \mapsto \mathbf{X} = \varphi^{-1}(\mathbf{x}, t) = \varphi_t^{-1}(\mathbf{x}) \end{cases} \quad (2.4)$$

Mit der Definition der Bewegung kann das Verschiebungsfeld $\mathbf{U}(\mathbf{X}, t)$ definiert werden. Es ist die Differenz zwischen den Ortsvektoren eines Punktes X in der momentanen Lage und der Ausgangslage

$$\mathbf{U}(\mathbf{X}, t) = \varphi(\mathbf{X}, t) - \mathbf{X} = \mathbf{x}(\mathbf{X}, t) - \mathbf{X} \quad \text{oder} \quad U^A = \delta_a^A x^a - X^A \quad (2.5)$$

Das Verschiebungsfeld $\mathbf{U}(\mathbf{X}, t)$ ist ein materielles Vektorfeld, das heißt es ist mit Größen der Ausgangskonfiguration dargestellt. Das entsprechende räumliche Verschiebungsfeld $\mathbf{u}(\mathbf{x}, t)$ bezüglich der momentanen Konfiguration lautet

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{x} - \varphi^{-1}(\mathbf{x}, t) = \mathbf{x} - \mathbf{X}(\mathbf{x}, t) \quad \text{oder} \quad u^a = x^a - \delta^a_A X^A \quad (2.6)$$

In einem einheitlichen Koordinatensystem sind die Verschiebungsfelder für einen Punkt $\mathbf{X} \mapsto \mathbf{x}$ identisch

$$\mathbf{U}(\mathbf{X}, t) = \mathbf{u}(\mathbf{x}, t) \quad (2.7)$$

In gleicher Weise wie das Verschiebungsfeld werden das Geschwindigkeits- und das Beschleunigungsfeld definiert. Das Geschwindigkeitsfeld ist die zeitliche Änderung der Lage der Materialpunkte. Für das materielle Geschwindigkeitsfeld $\mathbf{V}(\mathbf{X}, t)$ ergibt sich

$$\begin{aligned} \mathbf{V}(\mathbf{X}, t) &= \frac{d\varphi(\mathbf{X}, t)}{dt} = \dot{\varphi}(\mathbf{X}, t) = \dot{\mathbf{x}}(\mathbf{X}, t) \\ &= \frac{d(\mathbf{X} + \mathbf{U}(\mathbf{X}, t))}{dt} = \frac{d\mathbf{U}(\mathbf{X}, t)}{dt} = \dot{\mathbf{U}}(\mathbf{X}, t) \end{aligned} \quad (2.8)$$

und für das räumliche Geschwindigkeitsfeld $\mathbf{v}(\mathbf{x},t)$

$$\mathbf{v}(\mathbf{x},t) = \dot{\mathbf{u}}(\mathbf{x},t) = \mathbf{v}(\boldsymbol{\varphi}(\mathbf{X},t),t) = \mathbf{V}(\mathbf{X},t) \quad (2.9)$$

Die zeitliche Ableitung des Geschwindigkeitsfeldes ergibt das materielle Beschleunigungsfeld $\mathbf{A}(\mathbf{X},t)$

$$\begin{aligned} \mathbf{A}(\mathbf{X},t) &= \frac{d\mathbf{V}(\mathbf{X},t)}{dt} = \frac{d\dot{\mathbf{U}}(\mathbf{X},t)}{dt} \\ &= \dot{\mathbf{V}}(\mathbf{X},t) = \ddot{\mathbf{U}}(\mathbf{X},t) \end{aligned} \quad (2.10)$$

und das räumliche Beschleunigungsfeld $\mathbf{a}(\mathbf{x},t)$

$$\begin{aligned} \mathbf{a}(\mathbf{x},t) &= \frac{d\mathbf{v}(\mathbf{x},t)}{dt} = \frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(\mathbf{X},t)}{\partial t} \\ &= \frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x},t) \end{aligned} \quad (2.11)$$

mit der Summe aus der räumlichen Zeitableitung $\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial t}$ und der *konvektiven Ableitung* $\frac{\partial \mathbf{v}(\mathbf{x},t)}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x},t)$.

Deformationsgradient Um funktionelle Zusammenhänge zwischen den Größen in der Ausgangs- und Momentankonfiguration formulieren zu können, wird der *Deformationsgradient* definiert, der eine Beziehung zwischen den beiden Konfigurationen herstellt und daher eine zentrale Größe in der Kontinuumsmechanik ist. Der Deformationsgradient \mathbf{F} ist definiert mit Hilfe von infinitesimalen Elementen und er bildet das materielle Linienelement $d\mathbf{X}$ der Ausgangskonfiguration auf das räumliche Linienelement $d\mathbf{x}$ der Momentankonfiguration ab

$$d\mathbf{x} = \mathbf{F}(\mathbf{X},t)d\mathbf{X} \quad \text{oder} \quad dx^a = F^a_A dX^A \quad (2.12)$$

Somit ist der Deformationsgradient \mathbf{F} definiert als

$$\mathbf{F}(\mathbf{X},t) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \text{Grad} \mathbf{x} \quad \text{oder} \quad F^a_A = \frac{\partial x^a}{\partial X^A} = x^a_{,A} \quad (2.13)$$

mit dem materiellen Gradient Grad, der ebenso wie der räumliche Gradient grad im Anhang A erläutert ist.

Mit dem Verschiebungsfeld aus Gleichung (2.6) $\mathbf{x} = \mathbf{X} + \mathbf{u}$ folgt für den Deformationsgradienten

$$\mathbf{F}(\mathbf{X},t) = \mathbf{1} + \text{Grad} \mathbf{u} \quad \text{oder} \quad F^a_A = \delta^a_A + u^a_{,A} \quad (2.14)$$

Es sei noch angemerkt, dass der Deformationsgradient ein Zwei-Punkt-Tensor ist, der Basen in zwei Räumen besitzt und somit eine Verbindung zwischen den beiden

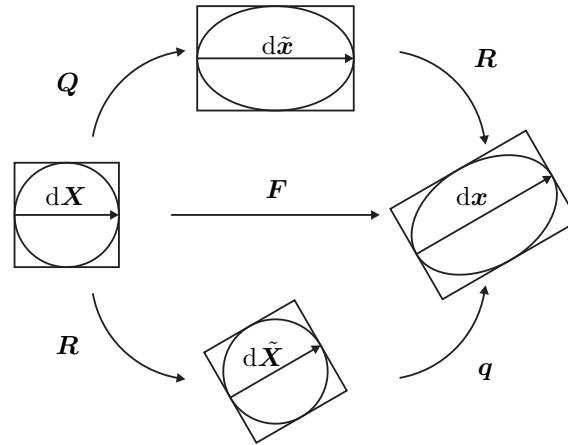


Bild 2.2: Polare Zerlegung des Deformationsgradienten

Räumen darstellt. Die erste Basis liegt im Tangentialraum der Momentankonfiguration und die zweite Basis liegt im Kotangentialraum der Ausgangskonfiguration. Der Deformationsgradient ist der *pushforward* φ_* von der Referenzkonfiguration in die Momentankonfiguration.

Da die Abbildung φ_t invertierbar ist, existiert der inverse Deformationsgradient

$$\mathbf{F}^{-1} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \text{grad} \mathbf{X} \quad \text{oder} \quad F^A_a = \frac{\partial X^A}{\partial x^a} = X^A_{,a} \quad (2.15)$$

Die Rücktransformation von der Momentankonfiguration in die Ausgangskonfiguration ist der *pullback* φ^* .

Polare Zerlegung Der Deformationsgradient kann zerlegt werden in eine Starrkörperbewegung und in eine Streckung. Wie im Bild 2.2 dargestellt, gibt es zwei Reihenfolgen für diese Transformationen der polaren Zerlegung. Die Zerlegung des Deformationsgradienten in ein Rotationsfeld \mathbf{R} und in ein materielles Streckungsfeld \mathbf{Q} beziehungsweise in ein räumliches Streckungsfeld \mathbf{q} lautet

$$\mathbf{F} = \mathbf{R}\mathbf{Q} = \mathbf{q}\mathbf{R} \quad \text{oder} \quad F^a_A = R^a_B \delta^{BC} Q_{CA} = q^{ca} \delta_{cb} R^b_A \quad (2.16)$$

In der Literatur werden die Streckungstensoren \mathbf{Q} und \mathbf{q} , auch rechter und linker Strecktensor genannt, mit \mathbf{U} und \mathbf{v} bezeichnet. Um Verwechslungen mit dem Verschiebungs- und Geschwindigkeitsfeld zu vermeiden, wurde hier eine andere Bezeichnung gewählt.

Änderung von infinitesimalen Elementen Neben der Transformation eines infinitesimalen Linienelements aus der Definition des Deformationsgradienten in Gleichung (2.12) $d\mathbf{x} = \mathbf{F}d\mathbf{X}$ werden weitere Änderungen von infinitesimalen Elementen

benötigt. Ein infinitesimales Volumenelement transformiert sich wie folgt

$$dv = \det(\mathbf{F}) dV = JdV \quad (2.17)$$

mit der Determinante des Deformationsgradienten

$$J = \det(\mathbf{F}) \quad (2.18)$$

Die Änderung eines infinitesimalen Flächenelements dA mit dem zugehörigen Normalenvektor \mathbf{N} aus der Ausgangskonfiguration zu dem Flächenelement da mit Normalenvektor \mathbf{n} der Momentankonfiguration lautet

$$\mathbf{n}da = \det(\mathbf{F}) \mathbf{F}^{-T} \mathbf{N}dA = J\mathbf{F}^{-T} \mathbf{N}dA \quad (2.19)$$

Verzerrungsmaße Nach der Bereitstellung der elementaren Größen der Kontinuumsmechanik werden nun Verzerrungen und Spannungen definiert. Zur Beschreibung von Verzerrungen können verschiedene Verzerrungsmaße eingeführt werden. Der *Green-Lagrange-Verzerrungstensor* beschreibt die Verzerrungen bezüglich der Ausgangskonfiguration. Er ist definiert als

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{1}) = \frac{1}{2} \left((\text{Grad} \mathbf{u})^T + \text{Grad} \mathbf{u} + (\text{Grad} \mathbf{u})^T \text{Grad} \mathbf{u} \right) \quad (2.20)$$

oder

$$E_{AB} = \frac{1}{2} (F_A^a \delta_{ab} F_B^b - \delta_{AB}) = \frac{1}{2} (\delta_{Ab} u^b_{,B} + \delta_{aB} u^a_{,A} + u^a_{,A} \delta_{ab} u^b_{,B}) \quad (2.21)$$

Der Green-Lagrange-Verzerrungstensor ist ein symmetrisches, materielles 2-Tensorfeld. Ein weiteres Verzerrungsmaß ist der *Euler-Almansi-Verzerrungstensor*. Er beschreibt die Verzerrungen in der Momentankonfiguration und ist definiert als

$$\mathbf{e} = \frac{1}{2} (\mathbf{1} - \mathbf{F}^{-T} \mathbf{F}^{-1}) = \frac{1}{2} \left((\text{grad} \mathbf{u})^T + \text{grad} \mathbf{u} - (\text{grad} \mathbf{u})^T \text{grad} \mathbf{u} \right) \quad (2.22)$$

oder

$$e_{ab} = \frac{1}{2} (\delta_{ab} - F_a^A \delta_{AB} F_b^B) = \frac{1}{2} (\delta_{ac} u^c_{,b} - \delta_{cb} u^c_{,a} + u^c_{,a} \delta_{cd} u^d_{,b}) \quad (2.23)$$

Der Euler-Almansi-Verzerrungstensor ist ebenso ein symmetrisches 2-Tensorfeld, allerdings in räumlicher Betrachtungsweise. Die beiden eingeführten Verzerrungstensenoren sind über pullback beziehungsweise pushforward ineinander überführbar. Der Green-Lagrange-Verzerrungstensor ist das Ergebnis des pullback des Euler-Almansi-Verzerrungstensors

$$\mathbf{E} = \varphi^* (\mathbf{e}) = \mathbf{F}^T \mathbf{e} \mathbf{F} \quad \text{oder} \quad E_{AB} = F_A^a e_{ab} F_B^b \quad (2.24)$$

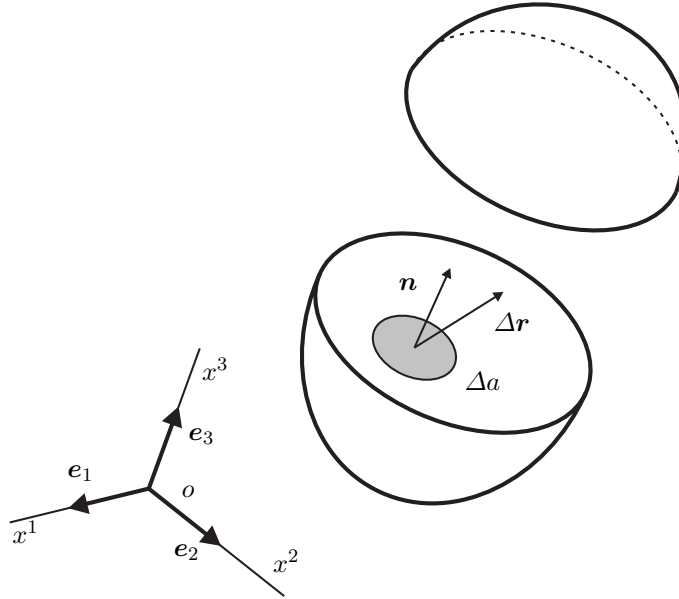


Bild 2.3: Deformierter Körper mit Schnittfläche

beziehungsweise der Euler-Almansi-Verzerrungstensor ist das Resultat des pushforward des Green-Lagrange-Verzerrungstensors

$$\mathbf{e} = \varphi_*(\mathbf{E}) = \mathbf{F}^{-T} \mathbf{E} \mathbf{F}^{-1} \quad \text{oder} \quad e_{ab} = F_a^A E_{AB} F_b^B \quad (2.25)$$

Die beiden Verzerrungstensoren (2.20) und (2.22) sind nichtlinear in den Verschiebungsableitungen. Als ein weiteres Verzerrungsmaß ist der *lineare Verzerrungstensor* \mathbf{E}_{lin} definiert

$$\mathbf{E}_{\text{lin}} = \frac{1}{2} \left((\text{Grad} \mathbf{u})^T + \text{Grad} \mathbf{u} \right) \quad (2.26)$$

oder

$$E_{\text{lin} AB} = \frac{1}{2} \left(\delta_{Ab} u^b_{,B} + \delta_{aB} u^a_{,A} \right) \quad (2.27)$$

Dieser wird in der linearen Elastizitätstheorie verwendet und ist hier der Vollständigkeit halber angegeben. Die Herleitungen erfolgen weiterhin mit den nichtlinearen Größen.

Cauchy-Spannungen Zur Darstellung von Spannungen wird der Körper zunächst in seiner deformierten Konfiguration \mathcal{B}_t betrachtet, wie es im Bild 2.3 dargestellt ist. Der Körper wird in zwei Teilkörper geschnitten und auf der Schnittfläche wird das Flächenelement Δa herausgegriffen. Der dazugehörige Normalenvektor \mathbf{n} ist dabei nach außen gerichtet. Die auf das Flächenelement wirkenden Schnittkräfte werden zusammengefasst im resultierenden Kraftvektor $\Delta \mathbf{r}$. Der Grenzwert für verschwin-

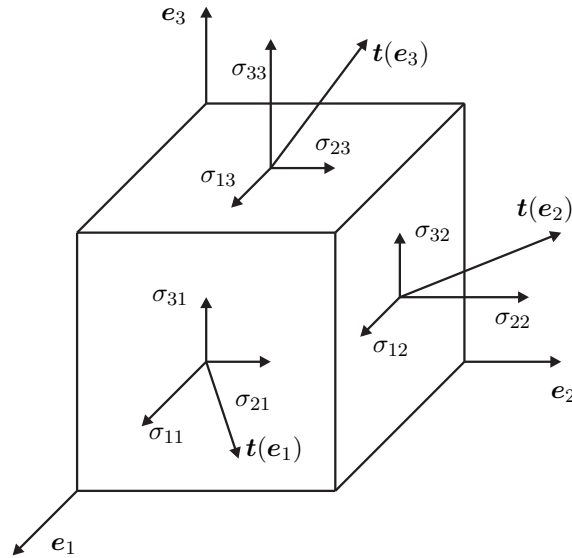


Bild 2.4: Würfel mit Komponenten des Cauchy-Spannungstensors

dende Flächenelemente

$$\mathbf{t}(\mathbf{n}) = \lim_{\Delta a \rightarrow 0} \frac{\Delta \mathbf{r}}{\Delta a} \quad (2.28)$$

ist der *Cauchy-Randspannungsvektor* \mathbf{t} für eine Schnittfläche. Nimmt man nun drei senkrecht aufeinander stehende Schnittflächen, so beschreiben sie den Spannungszustand in einem Materialpunkt vollständig. Die neun Koordinaten der Randspannungsvektoren der drei Schnittflächen ergeben die Koordinaten des *Cauchy-Spannungstensors* $\boldsymbol{\sigma}$. Mit dem Zusammenhang

$$\mathbf{t}(\mathbf{n}) = \boldsymbol{\sigma} \cdot \mathbf{n} \quad \text{oder} \quad t^a = \sigma^{ab} \delta_{bc} n^c = \sigma^{ab} n_b \quad (2.29)$$

und den Basisvektoren \mathbf{e}_1 , \mathbf{e}_2 , \mathbf{e}_3 als Schnittflächennormalen ergibt sich die Darstellung

$$\mathbf{t}(\mathbf{e}_1) = \boldsymbol{\sigma} \cdot \mathbf{e}_1 = \sigma^{11} \mathbf{e}_1 + \sigma^{21} \mathbf{e}_2 + \sigma^{31} \mathbf{e}_3 \quad (2.30)$$

$$\mathbf{t}(\mathbf{e}_2) = \boldsymbol{\sigma} \cdot \mathbf{e}_2 = \sigma^{12} \mathbf{e}_1 + \sigma^{22} \mathbf{e}_2 + \sigma^{32} \mathbf{e}_3 \quad (2.31)$$

$$\mathbf{t}(\mathbf{e}_3) = \boldsymbol{\sigma} \cdot \mathbf{e}_3 = \sigma^{13} \mathbf{e}_1 + \sigma^{23} \mathbf{e}_2 + \sigma^{33} \mathbf{e}_3 \quad (2.32)$$

Eine graphische Interpretation des Cauchy-Spannungstensors ist im Bild 2.4 dargestellt. Die drei senkrecht stehenden Schnittflächen werden zu einem Würfel erweitert und die zugehörigen Cauchy-Randspannungsvektoren sind an die jeweilige Seitenfläche angehängt.

Der Cauchy-Spannungstensor $\boldsymbol{\sigma}$ und die dazugehörigen Randspannungsvektoren \mathbf{t} beziehen sich auf die Momentankonfiguration. Der infinitesimale resultierende Kraftvektor ergibt sich daher zu

$$d\mathbf{r} = \mathbf{t} da \quad (2.33)$$

Neben dieser materiellen Beschreibung ist eine Darstellung der Spannungen in der Ausgangskonfiguration erforderlich. Wird in Gleichung (2.33) anstelle der räumlichen Maßfläche da die materielle Maßfläche dA verwendet und wird die Transformation (2.19) eines infinitesimalen Flächenelements eingesetzt, so resultieren die *Piola-Kirchhoff-Randspannungen 1. Art* \mathbf{t}_0

$$d\mathbf{r} = \mathbf{t} da = \underbrace{\boldsymbol{\sigma} \cdot \mathbf{n}}_{\mathbf{t}_0} da = J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{N} dA \quad (2.34)$$

$$\mathbf{t}_0 = J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{N} \quad (2.35)$$

Aus dieser Beziehung leitet sich der *Piola-Kirchhoff-Spannungstensor 1. Art* \mathbf{P} ab

$$\mathbf{t}_0 = \mathbf{P} \cdot \mathbf{N} \quad \text{oder} \quad t_0^a = P^{aB} \delta_{BC} N^C = P^{aB} N_B \quad (2.36)$$

$$\mathbf{P} = J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \quad \text{oder} \quad P^{aB} = J \sigma^{ab} F_b^B \quad (2.37)$$

Bezieht man in Gleichung (2.34) schließlich die infinitesimale Resultierende $d\mathbf{r}$ von der Momentankonfiguration auf die Ausgangskonfiguration $d\mathbf{R}$ mit der Gleichung (2.15) $d\mathbf{R} = \mathbf{F}^{-1} d\mathbf{r}$, so ergeben sich die *Piola-Kirchhoff-Randspannungen 2. Art* \mathbf{T}

$$d\mathbf{R} = \mathbf{F}^{-1} \cdot \mathbf{t}_0 dA = \mathbf{T} dA \quad (2.38)$$

$$\mathbf{T} = \mathbf{F}^{-1} \cdot \mathbf{t}_0 = \underbrace{J \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}}_{\mathbf{S}} \cdot \mathbf{N} \quad (2.39)$$

und es ergibt sich der *Piola-Kirchhoff-Spannungstensor 2. Art* \mathbf{S}

$$\mathbf{T} = \mathbf{S} \cdot \mathbf{N} \quad \text{oder} \quad T^A = S^{AB} \delta_{BC} N^C = S^{AB} N_B \quad (2.40)$$

$$\mathbf{S} = \mathbf{F}^{-1} \cdot \mathbf{P} = J \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \quad \text{oder} \quad S^{AB} = F_a^A P^{aB} = J F_a^A \sigma^{ab} F_b^B \quad (2.41)$$

Bilanzgleichungen – starke Form Der mechanische Gleichgewichtszustand des deformierbaren Körpers ist durch drei Bilanzgleichungen beschrieben: Massenbilanz, Impulsbilanz und Drehimpulsbilanz. Hier wird eine Formulierung der Bilanzgleichungen bezüglich der Ausgangskonfiguration gewählt, da sich diese Lagrange'sche Betrachtungsweise für strukturmechanische Probleme als geeigneter erweist.

Die Massenerhaltung bedeutet, dass die Masse unabhängig von der Konfiguration konstant bleiben muss. Die Masse m des Körpers ist

$$m(t) = \int_{B_t} \rho(\mathbf{x}, t) dv = \int_{B_0} \rho(\mathbf{X}) J dV = \int_{B_0} \rho_0(\mathbf{X}) dV \quad (2.42)$$

wobei ρ die momentane Massendichte und ρ_0 die Ausgangsdichte ist. Außerdem wurde Gleichung (2.17) zur Transformation des Volumenelements eingesetzt.

Für die Massenerhaltung muss die materielle Zeitableitung der Masse verschwinden

$$0 = \frac{d}{dt}m(t) = \frac{d}{dt} \int_{\mathcal{B}_0} \rho_0 dV = \int_{\mathcal{B}_0} \dot{\rho}_0 dV \quad (2.43)$$

Die Ableitung kann in das Integral gezogen werden, weil das Gebiet \mathcal{B}_0 zeitlich konstant ist. Dadurch erhält man als erste Bilanzgleichung die materielle Form der Massenerhaltung

$$\dot{\rho}_0 = 0 \quad (2.44)$$

Die Impulsbilanz besagt, dass die zeitliche Ableitung des Gesamtimpulses gleich der Summe der äußeren Kräfte ist. Für einen Körper lautet sie zunächst in der Momentankonfiguration

$$\int_{\partial\mathcal{B}_t} \mathbf{t} da + \int_{\mathcal{B}_t} \mathbf{b} dv = \frac{d}{dt} \int_{\mathcal{B}_t} \rho \mathbf{v} dv \quad (2.45)$$

Hier sind \mathbf{t} die Cauchy-Randspannungen auf dem momentanen Rand $\partial\mathcal{B}_t$ des Körpers, \mathbf{b} die äußeren Volumenkräfte pro momentanen Einheitsvolumen und \mathbf{v} das Geschwindigkeitsfeld.

Bezieht man (2.45) auf die Ausgangskonfiguration lautet der Impulssatz

$$\int_{\partial\mathcal{B}_0} \mathbf{t}_0 dA + \int_{\mathcal{B}_0} \mathbf{b}_0 dV = \frac{d}{dt} \int_{\mathcal{B}_0} \rho_0 \mathbf{V} dV \quad (2.46)$$

oder

$$\int_{\partial\mathcal{B}_0} t_0^a dA + \int_{\mathcal{B}_0} b_0^a dV = \frac{d}{dt} \int_{\mathcal{B}_0} \rho_0 V^a dV \quad (2.47)$$

dabei sind \mathbf{t}_0 die Piola-Randspannungen auf dem Rand $\partial\mathcal{B}_0$, \mathbf{b}_0 die äußeren Volumenkräfte bezogen auf die Referenzkonfiguration und \mathbf{V} das materielle Geschwindigkeitsfeld. Durch Einsetzen von Gleichung (2.36) und Anwendung des Gauß'schen Integralsatzes erhält man die lokale Form der Impulsbilanz

$$\text{Div}(\mathbf{P}) + \mathbf{b}_0 - \rho_0 \dot{\mathbf{V}} = \mathbf{0} \quad \text{oder} \quad P^{aB}{}_{,B} + b_0^a - \rho_0 \dot{V}^a = 0 \quad (2.48)$$

beziehungsweise mit Gleichung (2.41) $\mathbf{P} = \mathbf{F} \cdot \mathbf{S}$

$$\text{Div}(\mathbf{F} \cdot \mathbf{S}) + \mathbf{b}_0 - \rho_0 \dot{\mathbf{V}} = \mathbf{0} \quad \text{oder} \quad (F^a{}_C S^{CB})_{,B} + b_0^a - \rho_0 \dot{V}^a = 0 \quad (2.49)$$

Gemäß dem Drehimpulssatz ist die zeitliche Änderung des Dralls um einen ortsfesten Punkt y mit dem Ortsvektor \mathbf{y} gleich der Summe der äußeren Momente. Für jeden

Punkt $x \in \mathcal{B}_t$ sei der Differenzvektor $\mathbf{d} = \overrightarrow{ox} - \overrightarrow{oy} = \mathbf{x} - \mathbf{y}$. Der Drehimpulssatz für einen Körper in der Momentankonfiguration lautet

$$\int_{\partial\mathcal{B}_t} \mathbf{d} \times \mathbf{t} da + \int_{\mathcal{B}_t} \mathbf{d} \times \mathbf{b} dv = \frac{d}{dt} \int_{\mathcal{B}_t} \mathbf{d} \times \rho \mathbf{v} dv \quad (2.50)$$

Mit dem Differenzvektor $\mathbf{D} = \overrightarrow{OX} - \overrightarrow{OY} = \mathbf{X} - \mathbf{Y}$ bezüglich dem Punkt Y lautet der Drehimpulssatz in der Referenzkonfiguration

$$\int_{\partial\mathcal{B}_0} \mathbf{D} \times \mathbf{t}_0 dA + \int_{\mathcal{B}_0} \mathbf{D} \times \mathbf{b}_0 dV = \frac{d}{dt} \int_{\mathcal{B}_0} \mathbf{D} \times \rho \mathbf{V} dV \quad (2.51)$$

oder mit der Permutation ϵ_{bc}^a für das Kreuzprodukt

$$\int_{\partial\mathcal{B}_0} \epsilon_{bc}^a D^b t_0^c dA + \int_{\mathcal{B}_0} \epsilon_{bc}^a D^b b_0^c dV = \frac{d}{dt} \int_{\mathcal{B}_0} \epsilon_{bc}^a D^b \rho V^c dV \quad (2.52)$$

Durch mehrere Umformungen und Anwendung des Gauß'schen Integralsatzes erhält man die lokale Form des Drehimpulssatzes in der Ausgangskonfiguration

$$\mathbf{P} \cdot \mathbf{F}^T = \mathbf{F} \cdot \mathbf{P}^T \quad \text{oder} \quad P^{aB} F_B^b = F_A^a P^{Ab} \quad (2.53)$$

oder

$$\mathbf{S}^T = \mathbf{S} \quad \text{oder} \quad S^{BA} = S^{AB} \quad (2.54)$$

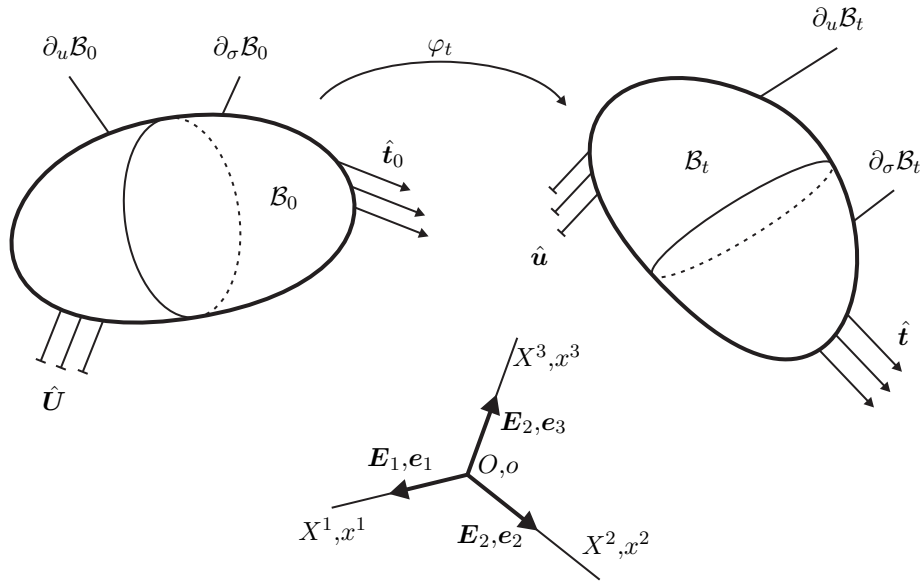
Aus der Drehimpulsbilanz folgt somit die Symmetrie des 2. Piola-Kirchhoff-Spannungstensors \mathbf{S} .

Randbedingungen Um das Problem vollständig zu formulieren, muss der Rand des Körpers noch Bedingungen bezüglich Beweglichkeit und Belastung genügen, wie im Bild 2.5 skizziert. Hierbei gibt es zwei unterschiedliche Randbedingungen.

Mit der *Dirichlet-Randbedingung* werden auf Teilbereichen $\partial_u \mathcal{B}_t$ die Verschiebungen $\hat{\mathbf{u}}$ und somit auch die Lage vorgegeben. Dies beinhaltet unter anderem die Lagerung des Körpers. In der materiellen Konfiguration lautet der Dirichlet-Rand $\partial_u \mathcal{B}_0$ und die vorgegebenen Randverschiebungen sind $\hat{\mathbf{U}}$.

Eine *Neumann-Randbedingung* liegt vor, wenn an Teilbereichen $\partial_\sigma \mathcal{B}_t$ des Randes $\partial \mathcal{B}_t$ die Cauchy-Randspannungen $\hat{\mathbf{t}}$ wirken. Dies beinhaltet auch freie Ränder, das heißt die Randspannungen sind Null. In der materiellen Konfiguration ist der Neumann-Rand $\partial_\sigma \mathcal{B}_0$ definiert als der Teilbereich des Randes $\partial \mathcal{B}_0$ an dem die Piola-Randspannungen $\hat{\mathbf{t}}_0$ wirken.

Die Neumann- und Dirichlet-Ränder müssen disjunkt sein. Es muss in der Momen-

**Bild 2.5:** Randbedingungen

tankonfiguration gelten

$$\partial_\sigma \mathcal{B}_t \cap \partial_u \mathcal{B}_t = \emptyset \quad (2.55)$$

$$\partial_\sigma \mathcal{B}_t \cup \partial_u \mathcal{B}_t = \partial \mathcal{B}_t \quad (2.56)$$

und entsprechend in der Ausgangskonfiguration

$$\partial_\sigma \mathcal{B}_0 \cap \partial_u \mathcal{B}_0 = \emptyset \quad (2.57)$$

$$\partial_\sigma \mathcal{B}_0 \cup \partial_u \mathcal{B}_0 = \partial \mathcal{B}_0 \quad (2.58)$$

Das bedeutet, dass auf jedem Randbereich genau eine Randbedingung vorliegen muss.

Anfangsrandwertproblem Fasst man die Bilanzgleichungen (2.44), (2.49), (2.54) und die Randbedingungen zusammen und fügt noch verträgliche Anfangswerte hinzu, erhält man ein Anfangsrandwertproblem. Bezogen auf die Ausgangskonfiguration lautet es

$$\text{Div}(\mathbf{F} \cdot \mathbf{S}) + \mathbf{b}_0 - \rho_0 \dot{\mathbf{V}} = \mathbf{0} \quad \text{Impulsbilanz} \quad (2.59)$$

$$\mathbf{S} = \mathbf{S}^T \quad \text{Drallbilanz} \quad (2.60)$$

$$\dot{\rho}_0 = 0 \quad \text{Massenerhaltung} \quad (2.61)$$

$$\mathbf{P} \cdot \mathbf{N} = \hat{\mathbf{t}}_0 \quad \text{auf } \partial_\sigma \mathcal{B}_0 \quad \text{Neumann-Randbedingung} \quad (2.62)$$

$$\mathbf{U} = \hat{\mathbf{U}} \quad \text{auf } \partial_u \mathcal{B}_0 \quad \text{Dirichlet-Randbedingung} \quad (2.63)$$

$$\mathbf{U}(\mathbf{X}, t_0) = \hat{\mathbf{U}}_0 \quad \text{Anfangsposition} \quad (2.64)$$

$$\mathbf{V}(\mathbf{X}, t_0) = \hat{\mathbf{V}}_0 \quad \text{Anfangsgeschwindigkeit} \quad (2.65)$$

Schwache Form Das Anfangsrandwertproblem ist eine nichtlineare partielle Differentialgleichung. Um von dieser lokalen Form zu einer schwachen Form zu kommen, multipliziert man die Impulsbilanz mit dem virtuellen Verschiebungsfeld als zulässige Testfunktion und integriert über das Gebiet. Durch das Einsetzen der weiteren Bilanzgleichungen und der Randbedingungen ergibt sich das Prinzip der virtuellen Arbeit. Es lautet bezogen auf die Ausgangskonfiguration

$$\underbrace{\int_{\mathcal{B}_0} \mathbf{S} : (\mathbf{F}^T \cdot \text{Grad} \delta \mathbf{u}) dV}_{\delta W_{\text{int}}} - \underbrace{\int_{\mathcal{B}_0} \mathbf{b}_0 \delta \mathbf{u} dV - \int_{\partial_\sigma \mathcal{B}_0} \hat{\mathbf{t}}_0 \delta \mathbf{u} dA}_{\delta W_{\text{ext}}} + \underbrace{\int_{\mathcal{B}_0} \rho_0 \dot{\mathbf{V}} \delta \mathbf{u} dV}_{\delta W_{\text{kin}}} = 0 \quad (2.66)$$

oder

$$\int_{\mathcal{B}_0} S^{AB} F_B^a \delta u_{a,A} dV - \int_{\mathcal{B}_0} b_0^a \delta u_a dV - \int_{\partial_\sigma \mathcal{B}_0} \hat{t}_0^a \delta u_a dA + \int_{\mathcal{B}_0} \rho_0 \dot{V}^a \delta u_a dV = 0 \quad (2.67)$$

mit den Anteilen der virtuellen inneren Arbeit δW_{int}

$$\delta W_{\text{int}} = \int_{\mathcal{B}_0} \mathbf{S} : (\mathbf{F}^T \cdot \text{Grad} \delta \mathbf{u}) dV \quad (2.68)$$

der virtuellen externen Arbeit δW_{ext}

$$\delta W_{\text{ext}} = \int_{\mathcal{B}_0} \mathbf{b}_0 \delta \mathbf{u} dV + \int_{\partial_\sigma \mathcal{B}_0} \hat{\mathbf{t}}_0 \delta \mathbf{u} dA \quad (2.69)$$

und der virtuellen kinetischen Arbeit δW_{kin}

$$\delta W_{\text{kin}} = \int_{\mathcal{B}_0} \rho_0 \dot{\mathbf{V}} \delta \mathbf{u} dV \quad (2.70)$$

Materialgesetz Für eine vollständige Problembeschreibung wird noch eine konstitutive Beziehung zwischen den Spannungen und den Verzerrungen benötigt. Für das verallgemeinerte Hooke'sche Gesetz $\mathbf{S} = \mathbf{C}^4 : \mathbf{E}$ mit dem Tensor 4. Ordnung \mathbf{C}^4 ergeben sich insgesamt $3^4 = 81$ freie Parameter. Ein einfaches Materialgesetz ist das *St. Venant-Kirchhoff-Gesetz*. Es beschreibt einen linearen Zusammenhang zwischen dem Green-Lagrange-Verzerrungstensor und dem 2. Piola-Kirchhoff-Spannungstensor mit 2 freien Parametern λ und μ

$$\mathbf{S} = \lambda \text{spur}(\mathbf{E}) \mathbf{1} + 2\mu \mathbf{E} \quad (2.71)$$

Die Lamé-Konstanten λ und μ können in den Elastizitätsmodul E und die Querkontraktionszahl ν umgerechnet werden mit

$$E = \frac{(3\lambda + 2\mu)\mu}{\lambda + \mu} \quad (2.72)$$

$$\nu = \frac{\lambda}{2(\lambda + \mu)} \quad (2.73)$$

beziehungsweise umgekehrt mit

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad (2.74)$$

$$\mu = \frac{E}{2(1 + \nu)} \quad (2.75)$$

2.2 Diskretisierung mit der Finite-Elemente-Methode

Mit der Finite-Elemente-Methode wird das betrachtete Kontinuum durch eine endliche Anzahl von Elementen diskretisiert. Auf jedes einzelne Element wird das Prinzip der virtuellen Arbeit angewendet und anschließend zu einem Gesamtsystem zusammengesetzt.

Der gesamte deformierbare Körper sei mit n_{knot} Knoten und mit n_{ele} Elementen diskretisiert, n Knoten bilden dabei ein Element. Die Lage der Knoten in der Ausgangslage sei $\overset{\circ}{X}_i^A$, $i = 1..n_{\text{knot}}$ und die Knotenverschiebung sei $\overset{\circ}{u}_i^a$, $i = 1..n_{\text{knot}}$. Da n Knoten der n_{knot} globalen Knoten ein Element bestimmen, ist ein notwendiger Schritt die Zuordnung von globalen Knotenindizes auf lokale Elementknotenindizes. Dieselbe Zuordnung wird später auch bei der Einordnung der elementweisen Zusammenhänge in eine globale Struktur benötigt. Es wird hier nicht näher darauf eingegangen und angenommen, dass diese Zuordnungen durch geeignete Softwarestrukturen korrekt aufgelöst werden. Die Lage der Elementknoten wird mit $\overset{\circ}{X}_i^{(e)A}$, $i = 1..n$ und die Verschiebung der Elementknoten mit $\overset{\circ}{u}_i^{(e)a}$, $i = 1..n$ bezeichnet. Der Index (e) soll andeuten, dass es sich jeweils um Elementgrößen handelt.

Die Diskretisierung für ein Element erfolgt mit dem isoparametrischen Ansatz für das Verschiebungsfeld und den Ort. Durch die Wahl von n Ansatzfunktionen $\Phi^i = \Phi^i(\xi_1, \xi_2, \xi_3)$, $i = 1..n$ folgt für den Ort

$$X^{(e)A} = \Phi^i \overset{\circ}{X}_i^{(e)A}, \quad \text{für } i = 1..n \quad (2.76)$$

und entsprechend für das Verschiebungsfeld

$$u^{(e)a} = \Phi^i \overset{\circ}{u}_i^{(e)a} \quad (2.77)$$

und für das Geschwindigkeitsfeld

$$v^{(e)a} = \dot{u}^{(e)a} = \Phi^i \dot{\overset{\circ}{u}}_i^{(e)a} \quad (2.78)$$

Um den Ansatz in das Prinzip der virtuellen Arbeit (2.67) einsetzen zu können, benötigt man noch die Diskretisierung der Variation $\delta \mathbf{u}$

$$\delta u_a^{(e)} = \delta \left(\Phi^i \overset{\circ}{u}_i^{(e)a} \right) = \Phi^i \delta \overset{\circ}{u}_{ia}^{(e)} \quad (2.79)$$

und die partielle Ableitung $\delta u_{a,A}$ der Variation

$$\delta u_{a,A}^{(e)} = \Phi^i{}_{,A} \delta \dot{u}_{ia}^{(e)} \quad (2.80)$$

Dabei berechnet sich die Ableitung der Ansatzfunktionen Φ^i nach den Variablen X^A wie folgt

$$\begin{aligned} \Phi^i{}_{,A} &= \frac{\partial \Phi^i}{\partial X^A} = \frac{\partial \Phi^i}{\partial \xi^j} \frac{\partial \xi^j}{\partial X^A} = \frac{\partial \Phi^i}{\partial \xi^j} \left(\frac{\partial X^A}{\partial \xi^j} \right)^{-1} \\ &= \frac{\partial \Phi^i}{\partial \xi^j} \left(\frac{\partial (\Phi^i X_i^{(e)A})}{\partial \xi^j} \right)^{-1} = \frac{\partial \Phi^i}{\partial \xi^j} \left(\frac{\partial \Phi^i}{\partial \xi^j} \dot{X}_i^{(e)A} \right)^{-1} \end{aligned} \quad (2.81)$$

Setzt man die Diskretisierung in das Prinzip der virtuellen Arbeit (2.67) ein, folgt für ein Element (e) in Indexschreibweise

$$\begin{aligned} \int_{\Omega^{(e)}} S^{AB} F_B{}^a \Phi^i{}_{,A} \delta \dot{u}_{ia}^{(e)} dV - \int_{\Omega^{(e)}} b_0^a \Phi^i \delta \dot{u}_{ia}^{(e)} dV - \int_{\partial_\sigma \Omega^{(e)}} \hat{t}_0^a \Phi^i \delta \dot{u}_{ia}^{(e)} dA \\ + \int_{\Omega^{(e)}} \rho \Phi^j \ddot{u}_j^{(e)a} \Phi^i \delta \dot{u}_{ia}^{(e)} dV = 0 \end{aligned} \quad (2.82)$$

Da die Gleichung (2.82) für alle Variationen gelten muss, erhält man

$$\left[\underbrace{\int_{\Omega^{(e)}} S^{AB} F_B{}^a \Phi^i{}_{,A} dV}_{f_{\text{int}}^{(e)ai}} - \underbrace{\int_{\Omega^{(e)}} b_0^a \Phi^i dV}_{f_{\text{ext}}^{(e)ai}} - \underbrace{\int_{\partial_\sigma \Omega^{(e)}} \hat{t}_0^a \Phi^i dA}_{f_{\text{kin}}^{(e)ai}} + \int_{\Omega^{(e)}} \rho \Phi^j \ddot{u}_j^{(e)a} \Phi^i dV \right] \delta \dot{u}_{ia}^{(e)} = 0 \quad (2.83)$$

mit den diskretisierten inneren Elementkräften $f_{\text{int}}^{(e)ai}$

$$f_{\text{int}}^{(e)ai} = \int_{\Omega^{(e)}} S^{AB} F_B{}^a \Phi^i{}_{,A} dV \quad (2.84)$$

den diskretisierten externen Elementkräften $f_{\text{ext}}^{(e)ai}$

$$f_{\text{ext}}^{(e)ai} = \int_{\Omega^{(e)}} b_0^a \Phi^i dV + \int_{\partial_\sigma \Omega^{(e)}} \hat{t}_0^a \Phi^i dA \quad (2.85)$$

und den diskretisierten kinetischen Elementkräften $f_{\text{kin}}^{(e)ai}$

$$f_{\text{kin}}^{(e)ai} = \int_{\Omega^{(e)}} \rho \Phi^j \ddot{u}_j^{(e)a} \Phi^i dV = \int_{\Omega^{(e)}} \rho \Phi^j \Phi^i dV \ddot{u}_j^{(e)a} \quad (2.86)$$

Da die Gleichung (2.83) für alle $\delta \dot{u}_{ia}^{(e)}$ gelten muss, muss das Kräftegleichgewicht für jedes Element erfüllt sein

$$f_{\text{int}}^{(e)ai} - f_{\text{ext}}^{(e)ai} + f_{\text{kin}}^{(e)ai} = 0 \quad (2.87)$$

Das Kräftegleichgewicht für ein Element (2.87) besteht aus $3 \cdot n$ Gleichungen, da $a = 1,2,3$ und $i = 1..n$. Für eine spätere Implementierung wird hier auf die Matrixnotation übergegangen. Dazu müssen Hilfsvektoren und -matrizen eingeführt werden. Die Koordinaten der n Elementknoten $\dot{X}_i^{(e)A}$ werden zusammengefasst in dem Knotenvektor $\dot{\mathbf{X}}^{(e)} \in \mathbb{R}^{3n}$

$$\dot{\mathbf{X}}^{(e)} = \left[\dot{X}_1^{(e)1} \quad \dot{X}_1^{(e)2} \quad \dot{X}_1^{(e)3} \quad \dot{X}_2^{(e)1} \quad \dots \quad \dot{X}_n^{(e)3} \right]^T \quad (2.88)$$

und die Knotenverschiebungen $\dot{u}_i^{(e)A}$ in dem Knotenverschiebungsvektor $\dot{\mathbf{u}}^{(e)} \in \mathbb{R}^{3n}$

$$\dot{\mathbf{u}}^{(e)} = \left[\dot{u}_1^{(e)1} \quad \dot{u}_1^{(e)2} \quad \dot{u}_1^{(e)3} \quad \dot{u}_2^{(e)1} \quad \dots \quad \dot{u}_n^{(e)3} \right]^T \quad (2.89)$$

Die globalen Knotenkoordinaten und -verschiebungen des gesamten Körpers werden ebenfalls in Vektoren $\dot{\mathbf{X}} \in \mathbb{R}^{3n_{\text{knot}}}$ und $\dot{\mathbf{u}} \in \mathbb{R}^{3n_{\text{knot}}}$ zusammengefasst

$$\dot{\mathbf{X}} = \left[\dot{X}_1^1 \quad \dot{X}_1^2 \quad \dot{X}_1^3 \quad \dot{X}_2^1 \quad \dots \quad \dot{X}_{n_{\text{knot}}}^3 \right]^T \quad (2.90)$$

$$\dot{\mathbf{u}} = \left[\dot{u}_1^1 \quad \dot{u}_1^2 \quad \dot{u}_1^3 \quad \dot{u}_2^1 \quad \dots \quad \dot{u}_{n_{\text{knot}}}^3 \right]^T \quad (2.91)$$

Mit den Ansatzfunktionen Φ^i wird die Matrix der Ansatzfunktionen $\Phi \in \mathbb{R}^{3 \times 3n}$ gebildet

$$\Phi = \begin{bmatrix} \Phi^1 & 0 & 0 & \Phi^2 & 0 & 0 & \dots & \Phi^n & 0 & 0 \\ 0 & \Phi^1 & 0 & 0 & \Phi^2 & 0 & \dots & 0 & \Phi^n & 0 \\ 0 & 0 & \Phi^1 & 0 & 0 & \Phi^2 & \dots & 0 & 0 & \Phi^n \end{bmatrix} \quad (2.92)$$

Mit diesen Definitionen folgt für die Verschiebungen $\mathbf{u}^{(e)} \in \mathbb{R}^3$ der Punkte eines Elements aus der Indexschreibweise (2.77)

$$\mathbf{u}^{(e)} = \Phi \dot{\mathbf{u}}^{(e)} \quad (2.93)$$

Die Ortsvektoren $\dot{\mathbf{X}}$ können nach (2.76) geschrieben werden als

$$\mathbf{X}^{(e)} = \Phi \dot{\mathbf{X}}^{(e)} \quad (2.94)$$

Da der Green-Lagrange-Verzerrungstensor \mathbf{E} und der 2. Piola-Kirchhoff-Spannungstensor \mathbf{S} symmetrisch sind, bestehen sie aus sechs unabhängigen Komponenten. Der Vektor der Green-Lagrange-Verzerrungen $\hat{\mathbf{E}} \in \mathbb{R}^6$ ist definiert als

$$\hat{\mathbf{E}} = [E_{11} \quad E_{22} \quad E_{33} \quad 2E_{12} \quad 2E_{23} \quad 2E_{13}]^T \quad (2.95)$$

und der Vektor der 2. Piola-Kirchhoff-Spannungen $\hat{\mathbf{S}} \in \mathbb{R}^6$ als

$$\hat{\mathbf{S}} = [S_{11} \ S_{22} \ S_{33} \ S_{12} \ S_{23} \ S_{13}]^T \quad (2.96)$$

Das St. Venant-Kirchhoff-Materialgesetz kann angegeben werden mit der Matrix $\mathbf{C} \in \mathbb{R}^{6 \times 6}$

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (2.97)$$

mit dem Elastizitätsmodul E und der Querkontraktionszahl ν . Die konstitutiven Gleichungen lauten somit

$$\hat{\mathbf{S}} = \mathbf{C} \hat{\mathbf{E}} \quad (2.98)$$

Des Weiteren werden die Matrizen $\mathbf{B}_{L0}^i \in \mathbb{R}^{6 \times 3}$ und $\mathbf{B}_{L1}^i \in \mathbb{R}^{6 \times 3}$ benötigt

$$\mathbf{B}_{L0}^i = \begin{bmatrix} \Phi^i_{,1} & 0 & 0 \\ 0 & \Phi^i_{,2} & 0 \\ 0 & 0 & \Phi^i_{,3} \\ \Phi^i_{,2} & \Phi^i_{,1} & 0 \\ 0 & \Phi^i_{,3} & \Phi^i_{,2} \\ \Phi^i_{,3} & 0 & \Phi^i_{,1} \end{bmatrix}, \quad i = 1..n \quad (2.99)$$

$$\mathbf{B}_{L1}^i = \begin{bmatrix} \Phi^i_{,1} u_{1,1} & \Phi^i_{,1} u_{2,1} & \Phi^i_{,1} u_{3,1} \\ \Phi^i_{,2} u_{1,2} & \Phi^i_{,2} u_{2,2} & \Phi^i_{,2} u_{3,2} \\ \Phi^i_{,3} u_{1,3} & \Phi^i_{,3} u_{2,3} & \Phi^i_{,3} u_{3,3} \\ \Phi^i_{,2} u_{1,1} + \Phi^i_{,1} u_{1,2} & \Phi^i_{,2} u_{2,1} + \Phi^i_{,1} u_{2,2} & \Phi^i_{,2} u_{3,1} + \Phi^i_{,1} u_{3,2} \\ \Phi^i_{,3} u_{1,2} + \Phi^i_{,2} u_{1,3} & \Phi^i_{,3} u_{2,2} + \Phi^i_{,2} u_{2,3} & \Phi^i_{,3} u_{3,2} + \Phi^i_{,2} u_{3,3} \\ \Phi^i_{,3} u_{1,1} + \Phi^i_{,1} u_{1,3} & \Phi^i_{,3} u_{2,1} + \Phi^i_{,1} u_{2,3} & \Phi^i_{,3} u_{3,1} + \Phi^i_{,1} u_{3,3} \end{bmatrix}, \quad i = 1..n \quad (2.100)$$

sowie deren Zusammensetzung zu $\mathbf{B}_{L0} \in \mathbb{R}^{6 \times 3n}$ und $\mathbf{B}_{L1} \in \mathbb{R}^{6 \times 3n}$

$$\mathbf{B}_{L0} = [\mathbf{B}_{L0}^1 \ \mathbf{B}_{L0}^2 \ \dots \ \mathbf{B}_{L0}^n] \quad (2.101)$$

$$\mathbf{B}_{L1} = [\mathbf{B}_{L1}^1 \ \mathbf{B}_{L1}^2 \ \dots \ \mathbf{B}_{L1}^n] \quad (2.102)$$

und deren Summe \mathbf{B}_L^i beziehungsweise \mathbf{B}_L

$$\mathbf{B}_L^i = \mathbf{B}_{L0}^i + \mathbf{B}_{L1}^i \quad \mathbf{B}_L = \mathbf{B}_{L0} + \mathbf{B}_{L1} \quad (2.103)$$

Der Green-Lagrange-Verzerrungsvektor $\hat{\mathbf{E}}$ kann mit den obigen Matrizen geschrieben werden als

$$\hat{\mathbf{E}} = \left(\mathbf{B}_{L0} + \frac{1}{2} \mathbf{B}_{L1} \right) \hat{\mathbf{u}}^{(e)} \quad (2.104)$$

Die Elementkräfte $f_{\text{int}}^{(e)ai}$, $f_{\text{ext}}^{(e)ai}$ und $f_{\text{kin}}^{(e)ai}$ werden analog zu Kraftvektoren $\mathbf{f}_{\text{int}}^{(e)}$, $\mathbf{f}_{\text{ext}}^{(e)}$, $\mathbf{f}_{\text{kin}}^{(e)} \in \mathbb{R}^{3n}$ zusammengefasst

$$\mathbf{f}_{\text{int}}^{(e)} = \left[f_{\text{int}}^{(e)11} \quad f_{\text{int}}^{(e)21} \quad f_{\text{int}}^{(e)31} \quad f_{\text{int}}^{(e)12} \quad \dots \quad f_{\text{int}}^{(e)3n} \right]^T \quad (2.105)$$

$$\mathbf{f}_{\text{ext}}^{(e)} = \left[f_{\text{ext}}^{(e)11} \quad f_{\text{ext}}^{(e)21} \quad f_{\text{ext}}^{(e)31} \quad f_{\text{ext}}^{(e)12} \quad \dots \quad f_{\text{ext}}^{(e)3n} \right]^T \quad (2.106)$$

$$\mathbf{f}_{\text{kin}}^{(e)} = \left[f_{\text{kin}}^{(e)11} \quad f_{\text{kin}}^{(e)21} \quad f_{\text{kin}}^{(e)31} \quad f_{\text{kin}}^{(e)12} \quad \dots \quad f_{\text{kin}}^{(e)3n} \right]^T \quad (2.107)$$

In Matrixnotation lassen sich die Kräfte für ein Element mit obigen Hilfsgrößen berechnen

$$\mathbf{f}_{\text{int}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_L^T \hat{\mathbf{S}} dV \quad (2.108)$$

$$\mathbf{f}_{\text{ext}}^{(e)} = \int_{\Omega^{(e)}} \boldsymbol{\Phi}^T \mathbf{b}_0 dV + \int_{\partial_\sigma \Omega^{(e)}} \boldsymbol{\Phi}_\sigma^T \mathbf{t}_0 dA \quad (2.109)$$

$$\mathbf{f}_{\text{kin}}^{(e)} = \int_{\Omega^{(e)}} \underbrace{\rho \boldsymbol{\Phi}^T \boldsymbol{\Phi}}_{\mathbf{M}^{(e)}} dV \ddot{\mathbf{u}}^{(e)} \quad (2.110)$$

mit der Elementmassenmatrix $\mathbf{M}^{(e)}$

$$\mathbf{M}^{(e)} = \int_{\Omega^{(e)}} \rho \boldsymbol{\Phi}^T \boldsymbol{\Phi} dV \quad (2.111)$$

Die Matrix $\boldsymbol{\Phi}_\sigma \in \mathbb{R}^{3 \times 3n}$ ist dabei die Matrix mit den Ansatzfunktionen Φ_σ^i für die Oberflächenkräfte und wird in gleicher Weise wie (2.92) gebildet.

Das Kräftegleichgewicht (2.87) für ein Element lautet in Matrixnotation

$$\mathbf{f}_{\text{int}}^{(e)} - \mathbf{f}_{\text{ext}}^{(e)} + \mathbf{f}_{\text{kin}}^{(e)} = \mathbf{f}_{\text{int}}^{(e)} - \mathbf{f}_{\text{ext}}^{(e)} + \mathbf{M}^{(e)} \ddot{\mathbf{u}}^{(e)} = \mathbf{0} \quad (2.112)$$

Der Übergang auf den gesamten diskretisierten Körper erfolgt durch Einordnung der Elementgleichungen in ein gemeinsames Gleichungssystem. Das Zusammensetzen der Elemente wird hier symbolisch mit \cup notiert. Es soll damit der Zusammenhang zwischen lokalen und globalen Knotenindizes veranschaulicht werden und auch

ein Verfahren die Komponenten eines Elements in die Gesamtstruktur einzuordnen. In der Literatur wird anstelle der symbolischen Darstellung auch eine mathematische Darstellung mit Einordnungs- beziehungsweise Zuordnungsmatrizen verwendet. Diese bestehen aus Null- und Einseinträgen an den entsprechenden Stellen. In einer programmtechnischen Umsetzung sind diese Einordnungsmatrizen nicht empfehlenswert und finden keine Verwendung, da sie einen unnötig hohen Speicherbedarf beanspruchen. Die Zusammensetzung von (2.112) resultiert in dem Kräftegleichgewicht für den gesamten diskretisierten Körper

$$\bigcup_{e=1}^{n_{\text{ele}}} \mathbf{f}_{\text{int}}^{(e)} - \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{f}_{\text{ext}}^{(e)} + \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{f}_{\text{kin}}^{(e)} = \mathbf{0} \quad (2.113)$$

$$\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{kin}} = \mathbf{0} \quad (2.114)$$

Für die globalen Kräfte gilt $\mathbf{f}_{\text{int}}, \mathbf{f}_{\text{ext}}, \mathbf{f}_{\text{kin}} \in \mathbb{R}^{3n_{\text{knot}}}$. Die globale Massenmatrix $\mathbf{M} \in \mathbb{R}^{3n_{\text{knot}} \times 3n_{\text{knot}}}$ setzt sich analog aus den Elementmassenmatrizen $\mathbf{M}^{(e)}$ zusammen

$$\mathbf{M} = \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{M}^{(e)} \quad (2.115)$$

und die globale kinetische Kraft \mathbf{f}_{kin} lautet

$$\mathbf{f}_{\text{kin}} = \mathbf{M} \ddot{\mathbf{u}} \quad (2.116)$$

2.2.1 Nichtlineare Finite-Elemente

Zur Lösung der nichtlinearen Gleichung des globalen Kräftegleichgewichts (2.114) kann das Newton-Verfahren verwendet werden. Dazu wird die Linearisierung der Residuumsform der Gleichung $\mathbf{G} = \mathbf{f}_{\text{int}} - \mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{kin}} = \mathbf{0}$ benötigt.

Die Linearisierung von \mathbf{G} erfolgt um den Punkt $\bar{\mathbf{u}}$.

$$\mathbf{G} = \mathbf{G}(\bar{\mathbf{u}}) + \underbrace{\frac{d\mathbf{G}}{d\mathbf{u}} \Big|_{\mathbf{u}=\bar{\mathbf{u}}}}_{\mathbf{K}_{\text{tang}}} (\mathbf{u} - \bar{\mathbf{u}}) + \mathcal{O}((\mathbf{u} - \bar{\mathbf{u}})^2) = \mathbf{0} \quad (2.117)$$

Die Ableitung des Kraftresiduums \mathbf{G} nach den Verschiebungen \mathbf{u} , ausgewertet an der Stelle $\bar{\mathbf{u}}$, wird *tangentiale Steifigkeitsmatrix* \mathbf{K}_{tang} genannt. Die Berechnung von $\mathbf{K}_{\text{tang}} = \mathbf{K}_{\text{tang}}(\bar{\mathbf{u}})$ wird auf verformungsunabhängige äußere Kräfte beschränkt, das heißt

$$\frac{d\mathbf{f}_{\text{ext}}}{d\mathbf{u}} = \mathbf{0} \quad (2.118)$$

Da hier die Trägheitskräfte unabhängig von \mathbf{u} sind, ergibt sich

$$\frac{d\mathbf{f}_{\text{kin}}}{d\mathbf{u}} = \mathbf{0} \quad (2.119)$$

Somit lautet die tangentielle Steifigkeitsmatrix

$$\mathbf{K}_{\text{tang}} = \frac{d\mathbf{G}}{d\mathbf{u}} = \frac{d\mathbf{f}_{\text{int}}}{d\mathbf{u}} \quad (2.120)$$

Die globale tangentielle Steifigkeitsmatrix \mathbf{K}_{tang} setzt sich aus den tangentialen Elementsteifigkeitsmatrizen $\mathbf{k}_{\text{tang}}^{(e)}$ zusammen

$$\mathbf{K}_{\text{tang}} = \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{k}_{\text{tang}}^{(e)} \quad (2.121)$$

Für ein Element berechnet sich die tangentielle Steifigkeitsmatrix wie folgt, wobei die Auswertung der Größen an dem Linearisierungspunkt $\bar{\mathbf{u}}$ erfolgt,

$$\begin{aligned} \mathbf{k}_{\text{tang}}^{(e)} &= \frac{d\mathbf{G}^{(e)}}{d\mathbf{u}} = \frac{d\mathbf{f}_{\text{int}}^{(e)}}{d\mathbf{u}} = \int_{\Omega^{(e)}} \frac{d}{d\mathbf{u}} \mathbf{B}_{\text{L}}^T \hat{\mathbf{S}} dV = \int_{\Omega^{(e)}} (\mathbf{B}_{\text{L},u}^T \hat{\mathbf{S}} + \mathbf{B}_{\text{L}}^T \hat{\mathbf{S}}_{,u}) dV \\ &= \underbrace{\int_{\Omega^{(e)}} \mathbf{B}_{\text{L},u}^T \hat{\mathbf{S}} dV}_{\mathbf{k}_{\text{geo}}^{(e)}} + \underbrace{\int_{\Omega^{(e)}} \mathbf{B}_{\text{L}}^T \hat{\mathbf{S}}_{,u} dV}_{\mathbf{k}_{\text{mat}}^{(e)}} \end{aligned} \quad (2.122)$$

$$\mathbf{k}_{\text{tang}}^{(e)} = \mathbf{k}_{\text{geo}}^{(e)} + \mathbf{k}_{\text{mat}}^{(e)} \quad (2.123)$$

Für die Elementsteifigkeitsmatrizen gilt $\mathbf{k}_{\text{tang}}^{(e)}, \mathbf{k}_{\text{geo}}^{(e)}, \mathbf{k}_{\text{mat}}^{(e)} \in \mathbb{R}^{3n \times 3n}$. Der erste Summand wird *geometrische Steifigkeitsmatrix* genannt und beinhaltet die Ableitungen der Matrizen mit den Ansatzfunktionen nach der Verschiebung

$$\mathbf{k}_{\text{geo}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_{\text{L},u}^T \hat{\mathbf{S}} dV \quad (2.124)$$

Mit den Hilfsmatrizen $\tilde{\mathbf{B}}_{\text{L}0} = \left[\tilde{\mathbf{B}}_{\text{L}0}^1 \quad \tilde{\mathbf{B}}_{\text{L}0}^2 \quad \dots \quad \tilde{\mathbf{B}}_{\text{L}0}^n \right] \in \mathbb{R}^{9 \times 3n}$

$$\tilde{\mathbf{B}}_{\text{L}0}^i = \begin{bmatrix} \Phi^i_{,1} & 0 & 0 \\ \Phi^i_{,2} & 0 & 0 \\ \Phi^i_{,3} & 0 & 0 \\ 0 & \Phi^i_{,1} & 0 \\ 0 & \Phi^i_{,2} & 0 \\ 0 & \Phi^i_{,3} & 0 \\ 0 & 0 & \Phi^i_{,1} \\ 0 & 0 & \Phi^i_{,2} \\ 0 & 0 & \Phi^i_{,3} \end{bmatrix}, \quad i = 1..n \quad (2.125)$$

und $\tilde{\mathbf{S}} \in \mathbb{R}^{9 \times 9}$

$$\tilde{\mathbf{S}} = \begin{bmatrix} \mathbf{S} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \end{bmatrix} \quad (2.126)$$

kann man die geometrische Steifigkeitsmatrix für ein Element auch schreiben als

$$\mathbf{k}_{\text{geo}}^{(e)} = \int_{\Omega^{(e)}} \tilde{\mathbf{B}}_{L0}^T \tilde{\mathbf{S}} \tilde{\mathbf{B}}_{L0} dV \quad (2.127)$$

Der zweite Summand in (2.122) ist die *materielle Steifigkeitsmatrix* und beinhaltet die Ableitung der Spannungen nach den Verschiebungen

$$\mathbf{k}_{\text{mat}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_L^T \hat{\mathbf{S}}_{,u} dV \quad (2.128)$$

Setzt man ein lineares St. Venant-Kirchhoff-Materialgesetz voraus, lautet die materielle Steifigkeitsmatrix für ein Element

$$\mathbf{k}_{\text{mat}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_L^T \mathbf{C} \mathbf{B}_L dV \quad (2.129)$$

Setzt man noch die Beziehung $\mathbf{B}_L = \mathbf{B}_{L0} + \mathbf{B}_{L1}$ ein, ergibt sich

$$\mathbf{k}_{\text{mat}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_{L0}^T \mathbf{C} \mathbf{B}_{L0} dV + \int_{\Omega^{(e)}} (\mathbf{B}_{L0}^T \mathbf{C} \mathbf{B}_{L1} + \mathbf{B}_{L1}^T \mathbf{C} \mathbf{B}_{L0} + \mathbf{B}_{L1}^T \mathbf{C} \mathbf{B}_{L1}) dV \quad (2.130)$$

Zusammengefasst ergibt sich die tangentielle Elementsteifigkeitsmatrix

$$\mathbf{k}_{\text{tang}}^{(e)} = \int_{\Omega^{(e)}} \tilde{\mathbf{B}}_{L0}^T \tilde{\mathbf{S}} \tilde{\mathbf{B}}_{L0} dV + \int_{\Omega^{(e)}} \mathbf{B}_L^T \mathbf{C} \mathbf{B}_L dV \quad (2.131)$$

2.2.2 Lineare Finite-Elemente

Erfolgt die Linearisierung um die Nulllage $\bar{\mathbf{u}} = \mathbf{0}$, so entspricht das Ergebnis dem Vorgehen nach der linearen Finite-Elemente-Methode. Die Herleitung in der Literatur erfolgt in der Regel nicht über die nichtlinearen Gleichungen, sondern über einen linearen Verzerrungstensor (2.26).

Bei einer Linearisierung um $\bar{\mathbf{u}} = \mathbf{0}$ ergeben sich folgende Vereinfachungen

$$\mathbf{B}_{L1} = \mathbf{0} \quad (2.132)$$

$$\mathbf{S} = \mathbf{0} \quad (2.133)$$

Die geometrische Steifigkeitsmatrix $\mathbf{k}_{\text{geo}}^{(e)}$ ist somit Null und die tangentielle Elementsteifigkeitsmatrix vereinfacht sich zu

$$\mathbf{k}_{\text{tang}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_{L0}^T \mathbf{C} \mathbf{B}_{L0} dV = \mathbf{k}_{\text{lin}}^{(e)} \quad (2.134)$$

Die *lineare Elementsteifigkeitsmatrix* $\mathbf{k}_{\text{lin}}^{(e)}$ ist nicht von \mathbf{u} abhängig. Die globale lineare Steifigkeitsmatrix $\mathbf{K}_{\text{lin}} \in \mathbb{R}^{3n_{\text{knot}} \times 3n_{\text{knot}}}$ erhält man durch entsprechendes Ein-sortieren

$$\mathbf{K}_{\text{lin}} = \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{k}_{\text{lin}}^{(e)} \quad (2.135)$$

Das linearisierte Kräftegleichgewicht (2.117) lautet somit für lineare Finite-Elemente

$$\mathbf{K}_{\text{lin}} \dot{\mathbf{u}} - \mathbf{f}_{\text{ext}} + \mathbf{M} \ddot{\mathbf{u}} = \mathbf{0} \quad (2.136)$$

2.2.3 Korotierte Finite-Elemente

Zu dem nichtlinearen und dem linearen Ansatz wird noch ein weiterer Ansatz vorgestellt. Bei der korotierten Finite-Elemente-Methode wird in jedes Element des Körpers ein eigenes Koordinatensystem gelegt. In der Literatur werden auch Ansätze beschrieben, die in jeden Elementknoten ein Koordinatensystem legen, was aber zu unerwünschten Zusatzkräften führen kann [87].

Die Rotationsmatrix zwischen dem Ausgangskordinatensystem und dem Koordinatensystem der deformierten Lage des Elements (e) sei $\mathbf{R}^{(e)}$ und hängt von der Verformung ab

$$\mathbf{R}^{(e)} = \mathbf{R}^{(e)}(\bar{\mathbf{u}}^{(e)}) \quad (2.137)$$

Setzt man diese n -mal auf der Hauptdiagonalen einer Matrix $\mathbf{R} \in \mathbb{R}^{3n \times 3n}$ ein, so erhält man

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}^{(e)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}^{(e)} \end{bmatrix} \quad (2.138)$$

In dem rotierten System wird nun der folgende Ansatz für die interne Kraft des Elements gemacht [87]

$$\mathbf{f}_{\text{int}}^{(e)} = \mathbf{R} \mathbf{k}_{\text{lin}}^{(e)} \left(\mathbf{R}^T \hat{\mathbf{x}}^{(e)} - \hat{\mathbf{X}}^{(e)} \right) \quad (2.139)$$

Mit der diskretisierten Form der Gleichung (2.6) $\dot{\mathbf{x}}^{(e)} = \dot{\mathbf{X}}^{(e)} + \dot{\mathbf{u}}^{(e)}$ ergibt sich

$$\mathbf{f}_{\text{int}}^{(e)} = \mathbf{R}\mathbf{k}_{\text{lin}}^{(e)} \left(\mathbf{R}^T \left(\dot{\mathbf{X}}^{(e)} + \dot{\mathbf{u}}^{(e)} \right) - \dot{\mathbf{X}}^{(e)} \right) \quad (2.140)$$

$$= \mathbf{k}_{\text{korot}}^{(e)} \dot{\mathbf{u}}^{(e)} + \mathbf{f}_{\text{korot}}^{(e)} \quad (2.141)$$

mit der korotierten Steifigkeitsmatrix $\mathbf{k}_{\text{korot}}^{(e)}$

$$\mathbf{k}_{\text{korot}}^{(e)} = \mathbf{R}\mathbf{k}_{\text{lin}}^{(e)}\mathbf{R}^T \quad (2.142)$$

und dem korotierten Kraftvektor $\mathbf{f}_{\text{korot}}^{(e)}$

$$\mathbf{f}_{\text{korot}}^{(e)} = \mathbf{R}\mathbf{k}_{\text{lin}}^{(e)} \left(\mathbf{R}^T \dot{\mathbf{X}}^{(e)} - \dot{\mathbf{X}}^{(e)} \right) \quad (2.143)$$

Die globale korotierte Steifigkeitsmatrix erhält man durch Einsortieren der Elementmatrizen

$$\mathbf{K}_{\text{korot}} = \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{k}_{\text{korot}}^{(e)} \quad (2.144)$$

und analog erhält man den globalen korotierten Kraftvektor

$$\mathbf{f}_{\text{korot}} = \bigcup_{e=1}^{n_{\text{ele}}} \mathbf{f}_{\text{korot}}^{(e)} \quad (2.145)$$

Das Kräftegleichgewicht (2.114) lautet nun mit dem korotierten Ansatz

$$\mathbf{K}_{\text{korot}}\dot{\mathbf{u}} + \mathbf{f}_{\text{korot}} - \mathbf{f}_{\text{ext}} + \mathbf{M}\ddot{\mathbf{u}} = \mathbf{0} \quad (2.146)$$

Die globale korotierte Steifigkeitsmatrix $\mathbf{K}_{\text{korot}} = \mathbf{K}_{\text{korot}}(\bar{\mathbf{u}})$ hängt aufgrund der Definition von der vorgegebenen Verformung $\bar{\mathbf{u}}$ ab. Ist die vorgegebene Verformung $\bar{\mathbf{u}} = \mathbf{0}$, so ergibt sich für die Matrix $\mathbf{K}_{\text{korot}} = \mathbf{K}_{\text{lin}}$, da $\mathbf{R}^{(e)} = \mathbf{I}$. Da die Steifigkeitsmatrix von der Verformung abhängt, muss die Berechnung der korotierten Lösung iterativ erfolgen. Die Erfahrung zeigt, dass man bereits mit einer Iteration eine ausreichend gute Lösung erzielt. Deshalb wird in der Literatur bei dem korotierten Lösungsverfahren oft genau ein Iterationsschritt angenommen.

Der Ansatz (2.139) für die interne Kraft kann wie folgt interpretiert werden. Mit $\mathbf{R}^T \dot{\mathbf{x}}^{(e)}$ wird zunächst die deformierte Lage in die Ausgangslage rotiert. Anschließend wird die Kraft mit $\mathbf{k}_{\text{lin}}^{(e)} \left(\mathbf{R}^T \dot{\mathbf{x}}^{(e)} - \dot{\mathbf{X}}^{(e)} \right)$ berechnet. Diese Kraft wird schließlich wieder mit \mathbf{R} in die deformierte Lage rotiert.

Die bis jetzt noch nicht festgelegte Bestimmung der Rotationsmatrizen kann im Falle der linearen Tetraeder mit Hilfe der polaren Zerlegung des Deformationsgradienten erfolgen. Es wird später gezeigt, dass der Deformationsgradient für lineare Tetraeder in einem Element konstant ist und sich daher genau eine Rotation für das Element

aus der polaren Zerlegung ergibt. In Gleichung (2.16) wird der Deformationsgradient in einen Rotationstensor \mathbf{R} und einen Strecktensor \mathbf{Q} aufgeteilt

$$\mathbf{F} = \mathbf{R}\mathbf{Q} \quad (2.147)$$

Bei dem korotierten Ansatz muss nun bei einer gegebenen Verformung und damit gegebenen Deformationsgradienten der Rotationsanteil extrahiert werden. Dazu eignet sich der Algorithmus von HIGHAM [61], der iterativ die Rotationsmatrix bestimmt.

Algorithmus 2.1 : Algorithmus zur Berechnung der Rotationsmatrix

```

 $\mathbf{R}_0 = \mathbf{F}$ 
for  $i = 1$  to  $k$  do
  |  $\mathbf{R}_i = \frac{1}{2} (\mathbf{R}_{i-1} + \mathbf{R}_{i-1}^{-T})$ 
end

```

Dieser Algorithmus konvergiert schnell gegen die gesuchte Rotationsmatrix $\mathbf{R}^{(e)}$, in der Regel sind drei Iterationen ausreichend.

2.2.4 Tetraederelemente

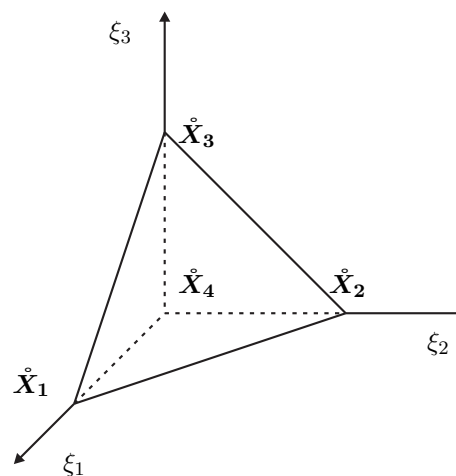


Bild 2.6: Tetraeder

Für die Diskretisierung können verschiedene Elemente verwendet werden. Einen Überblick über die Menge der Elemente ist in [14], [125], [41] gegeben. In dieser Arbeit werden Tetraeder mit vier linearen Ansatzfunktionen verwendet. Die Freiheitsgrade des Elements sind die vier Eckknoten des Tetraeders \mathbf{X}_1 , \mathbf{X}_2 , \mathbf{X}_3 und \mathbf{X}_4 . Die zugehörigen Ansatzfunktionen $\Phi^i, i = 1..4$ lauten

$$\Phi^1 = \xi_1, \quad \Phi^2 = \xi_2, \quad \Phi^3 = \xi_3, \quad \Phi^4 = 1 - \xi_1 - \xi_2 - \xi_3 \quad (2.148)$$

Durch diese Wahl ergeben sich einige Vereinfachungen in der Berechnung der Systemmatrizen. Die Ableitungen der Ansatzfunktionen Φ^i nach den lokalen Variablen ξ_j

sind jeweils konstante Funktionen

$$\frac{\partial \Phi^i}{\partial \xi_j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad (2.149)$$

Daraus folgt unter anderem, dass innerhalb eines Elements der Deformationsgradient konstant ist. Die Berechnung der Steifigkeitsmatrizen vereinfacht sich, da auf Grund der konstanten Funktionen auf eine Gauß-Quadratur verzichtet werden kann. So ergibt sich zum Beispiel für die lineare Elementsteifigkeitsmatrix

$$\mathbf{k}_{\text{lin}}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}_{\text{L}0}^T \mathbf{C} \mathbf{B}_{\text{L}0} dV = V^{(e)} \mathbf{B}_{\text{L}0}^T \mathbf{C} \mathbf{B}_{\text{L}0} \quad (2.150)$$

wobei $V^{(e)}$ das Volumen des Tetraeders ist.

Ein Nachteil des linearen Tetraederelements ist der sogenannte *Locking*-Effekt, der auftritt, wenn sich der Materialparameter ν dem Wert 0.5 nähert. Dies ist bei inkompressiblen Materialien der Fall. Eine Möglichkeit das Locking zu vermeiden ist Ansatzfunktionen mit höherer Ordnung zu verwenden, zum Beispiel quadratische Ansatzfunktionen, und eine reduzierte Integration durchzuführen [125]. Dies führt zu Systemen mit einer größeren Anzahl von Freiheitsgraden und damit auch zu einem höheren Rechenaufwand. In dieser Arbeit wird auf Grund von Rechenzeitaspekten auf Techniken zur Vermeidung von Locking verzichtet. Eine sorgfältige Beobachtung der Systemparameter ist daher stets notwendig.

2.2.5 Vergleich der Ansätze

Die Ansätze der nichtlinearen, linearen und korotierten Finite-Elemente führen zu unterschiedlichen Ergebnissen. Für einen Vergleich wird als Beispiel ein statisches Problem betrachtet. Ein Balken ist auf einer Seite eingespannt und er deformiert sich auf Grund der Schwerkraft. Es wird jeweils die Verformung des Balkens mit den vorgestellten Ansätzen berechnet. Die Diskretisierung des Balkens und die Materialparameter sind identisch. Für die korotierte Lösung wird die Verformung des linearen Ansatzes als Vorgabeverformung verwendet und es wird nur ein Iterationsschritt durchgeführt.

In den Bildern 2.7 und 2.8 kann man die Verformungen des Balkens sehen. Bei einer geringeren Schwerkraft (Bild 2.7) ergeben sich kleinere Verformungen und die drei vorgestellten Ansätze stimmen gut überein. Bei einer großen Verformung (Bild 2.8) kann man eine deutliche Abweichung der linearen Lösung von der nichtlinearen Lösung erkennen. Die korotierte Lösung stimmt mit der nichtlinearen Lösung noch gut überein und stellt somit einen guten Kompromiss zwischen Genauigkeit und Rechenaufwand dar.

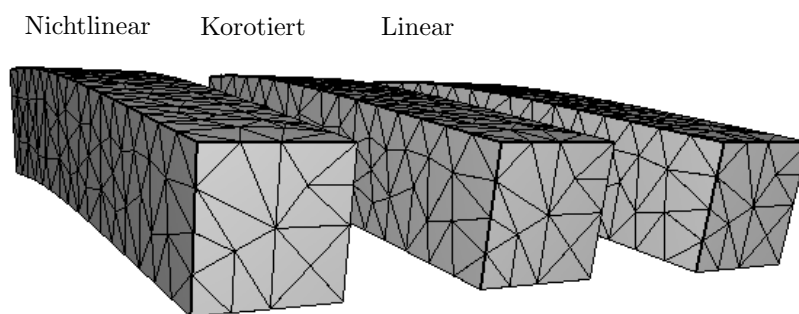


Bild 2.7: Vergleich der Ansätze bei kleiner Verformung

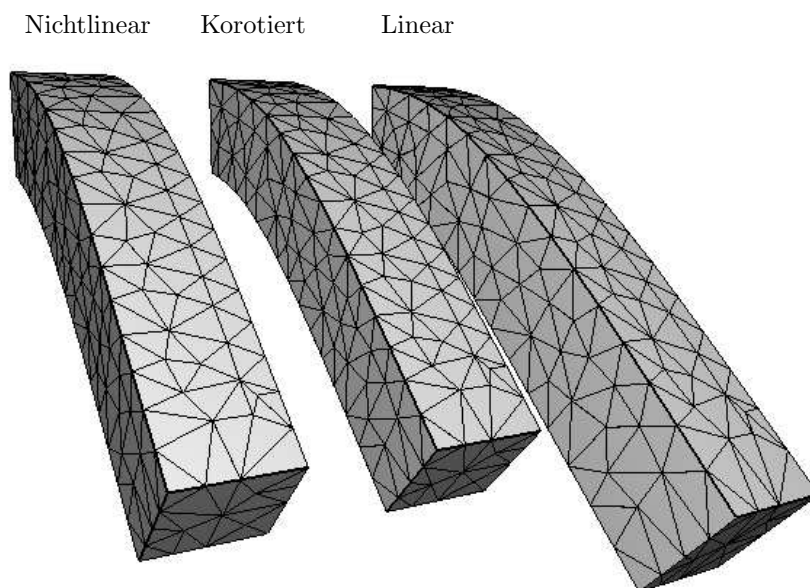


Bild 2.8: Vergleich der Ansätze bei großer Verformung

3 Experimentelle Modellidentifikation

Die Modelle aus dem vorherigen Kapitel haben jeweils zum Ziel, das Deformationsverhalten eines realen Objekts zu beschreiben. Dabei wird von bekannten Modellparametern ausgegangen und basierend auf physikalischen Gesetzen ein System von Gleichungen für das Kräftegleichgewicht hergeleitet. Die Erstellung eines Modells für ein reales Objekt setzt die Bestimmung der notwendigen Modellparameter voraus. Bei den meisten realen Objekten ist eine rechen-technische Objektbeschreibung, wie zum Beispiel CAD-Daten, nicht vorhanden und es müssen daher experimentelle Verfahren zur Identifikation eingesetzt werden. Dadurch wird auch eine gute Übereinstimmung des Modells mit der Realität gewährleistet.

Ein Ziel der vorliegenden Arbeit ist es, mit einem einzigen System eine abgeschlossene Lösung, von der Identifikation der deformierbaren Körper bis zur Anwendung der Simulation, zu bieten. Deshalb werden die Messungen für die Modellidentifikation mit einem Robotersystem durchgeführt. Des Weiteren wird der Roboter als Teil eines Telepräsenzsystems verwendet, in dem auch die Simulation des deformierbaren Körpers integriert ist.

In diesem Kapitel wird ein Vorgehen zur Identifikation der Modellparameter eines deformierbaren Körpers präsentiert. Erfolgt die Modellierung mit dem Finite-Elemente-Ansatz, so sind die notwendigen Modellparameter eine Beschreibung der Geometrie und des Materials. Die experimentelle Bestimmung der dreidimensionalen Geometrie erfolgt durch die Auswertung von Kamerabildern. Zur Identifikation der Materialparameter werden Verformungen und Kontaktkräfte des Körpers gemessen.

3.1 Geometrische Modellbildung

Die geometrische Modellbildung verfolgt das Ziel, eine Darstellung der Objektgeometrie zu erzeugen, die in dem FEM-Ansatz verwendet werden kann. Da in dieser Arbeit die Diskretisierung mit linearen Tetraederelementen erfolgt, wird ein Volumennetz des Körpers benötigt, das aus Tetraedern besteht. Um ein Volumennetz zu erstellen, muss zunächst die Objektoberfläche erfasst werden. Hierfür gibt es viele experimentelle Verfahren, die nach verschiedenen Kriterien klassifiziert werden können, wie es im Kapitel 1 geschehen ist. Die eingesetzten Sensoren sind ebenso zahlreich, bei den optischen Verfahren werden zum Beispiel Interferometer, Lasertriangulation, einfache Kameras oder Stereokameras verwendet.

In dieser Arbeit kommt eine einfache Kamera zum Einsatz, mit der Bilder aus mehreren Ansichten aufgenommen werden. Damit lässt sich aus zwei Ansichten eine

Pseudo-Stereokamera erzeugen, wodurch dreidimensionale Informationen der Szene verfügbar sind und die Oberfläche des Objekts rekonstruiert werden kann. Der Vorteil dieser Lösung sind die geringen Kosten, da bereits eine einfache digitale Kamera oder Webcam ausreichend ist. Der Nachteil ist, dass die Lage und Orientierung der Kamera bekannt sein müssen. Außerdem kann dieses Verfahren nur bei unbewegten Objekten angewendet werden, da eine gewisse Zeit für die Positionierung der Kamera notwendig ist und sich dabei die Szene nicht ändern darf.

3.1.1 Computer Vision

Computer Vision ist das Fachgebiet, das sich mit der Analyse von natürlichen Bildern beschäftigt und daraus Informationen über die reale Welt ableitet. Anwendungsbeispiele sind Objekterkennung und -verfolgung, Lageerkennung und Bauteilprüfung, die in vielen industriellen und natürlichen Umgebungen eingesetzt werden. Einen guten Überblick über die Möglichkeiten, Algorithmen und Prinzipien bieten die Bücher [106], [58], [47], [48]. Im Folgenden wird ein Verfahren zur Rekonstruktion eines 3D Punktes aus einem Stereobildpaar entwickelt. Dieser Schritt hat eine zentrale Bedeutung um später die gesamte Oberfläche des Objekts zu rekonstruieren. Die für das 3D Rekonstruktionsverfahren benötigten mathematischen Grundlagen und Herleitungen werden in den folgenden Abschnitten zusammengefasst.

Grundlagen Die Grundlage für alle weiteren Überlegungen ist ein mathematisches Modell für eine Kamera. Eine Kamera ist eine Abbildung von einer dreidimensionalen Welt auf ein zweidimensionales Bild. Das grundlegende Prinzip dieser Abbildung ist die Zentralprojektion und kann mit dem Lochkameramodell beschrieben werden, wie im Bild 3.1 skizziert. Bei einer klassischen Lochkamera befindet sich die Bildebene hinter dem Projektionszentrum. Im Weiteren wird die Bildebene vor das Projektionszentrum gelegt.

Das Projektionszentrum ist der Punkt C , in dem auch das Kamerakoordinatensystem liegt. Die optische Achse der Kamera zeigt in Z -Richtung und schneidet die Bildebene im Kamerahauptpunkt c . Ein Punkt M mit dem Ortsvektor $\mathbf{M} = [X \ Y \ Z]^T \in \mathbb{R}^3$ des euklidischen Raums wird abgebildet auf den Punkt m mit

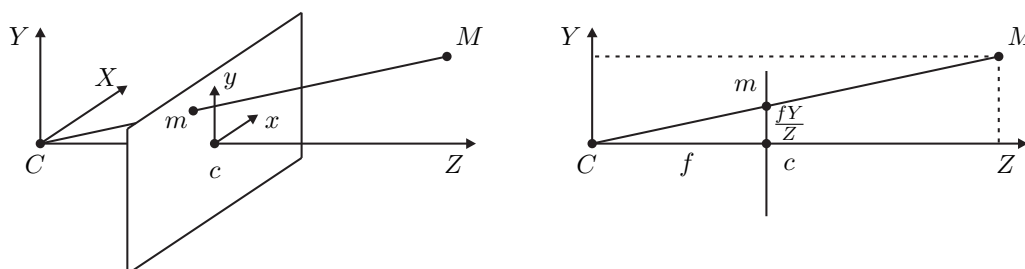


Bild 3.1: Lochkamera

dem Ortsvektor $\mathbf{m} = \left[\frac{fX}{Z} \quad \frac{fY}{Z} \right]^T \in \mathbb{R}^2$ der Bildebene. Mit Hilfe der projektiven Geometrie lautet die Projektionsabbildung in homogenen Koordinaten

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.1)$$

Dabei werden die homogenen Koordinaten von M im projektiven Raum zusammengefasst im Vektor

$$\tilde{\mathbf{M}} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.2)$$

und vom Punkt m in der projektiven Ebene im Vektor

$$\tilde{\mathbf{m}} = \begin{bmatrix} x \\ y \\ s \end{bmatrix} \quad (3.3)$$

Die Bildkoordinaten von m sind somit

$$\mathbf{m} = \begin{bmatrix} \frac{x}{s} \\ \frac{y}{s} \\ s \end{bmatrix} \quad (3.4)$$

Es werden Vektoren aus dem projektiven Raum beziehungsweise der projektiven Ebene mit der Tilde $\tilde{}$ gekennzeichnet.

Die Gleichung für die Gerade l mit den Punkten p in der projektiven Ebene lautet

$$\tilde{\mathbf{l}}^T \tilde{\mathbf{p}} = 0 \quad (3.5)$$

Die Dualität in der projektiven Ebene besagt, dass der Schnitt von zwei Geraden einen Punkt ergibt und dass die Verbindung von zwei Punkten eine Gerade ergibt

$$\tilde{\mathbf{p}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2 \quad (3.6)$$

$$\tilde{\mathbf{l}} = \tilde{\mathbf{p}}_1 \times \tilde{\mathbf{p}}_2 \quad (3.7)$$

Nach der bisherigen Festlegung befindet sich das Bildkoordinatensystem im Kamerahauptpunkt c . Legt man den Ursprung des Bildkoordinatensystems in die linke untere Ecke des Bildes, wie im Bild 3.2 skizziert, muss bei der Projektionsabbildung zusätzlich eine Verschiebung p_x und p_y berücksichtigt werden. Beim Einsatz einer CCD-Kamera will man die Koordinaten in Pixeleinheiten dargestellt haben, was mit einer entsprechenden Skalierung erfolgt. Nichtquadratische Pixel werden durch die

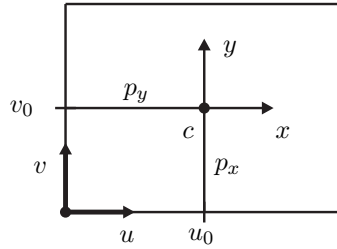


Bild 3.2: Bild- und Kamerakoordinatensystem

Skalierungsfaktoren m_x, m_y berücksichtigt. Die Abbildung (3.1) erweitert sich zu

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fm_x & 0 & p_x m_x & 0 \\ 0 & fm_y & p_y m_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.8)$$

mit den Pixelkoordinaten $\mathbf{m} = [u \ v]^T$ des Punktes m . Die Matrix

$$\mathbf{K} = \begin{bmatrix} fm_x & 0 & p_x m_x \\ 0 & fm_y & p_y m_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_u & 0 & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

wird *intrinsische Matrix* genannt und beinhaltet ausschließlich interne Parameter der Kamera. Bisher sind die 3D Koordinaten bezüglich des Kamerasystems mit dem Projektionszentrum als Ursprung dargestellt. Sind die Punkte M nicht im Kamerakoordinatensystem gegeben, sondern in einem Weltsystem, muss diese Transformation in der Projektionsgleichung berücksichtigt werden. Die Drehmatrix zwischen Kamera- und Weltsystem ist \mathbf{R} und der Translationsvektor ist \mathbf{t} . Die homogene Transformationsmatrix

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.10)$$

ist die *extrinsische Matrix* der Kamera, die die Lage der Kamera im Raum beschreibt. Demnach ergibt sich die allgemeine Abbildungsvorschrift für eine Kamera

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_u & 0 & u_0 & 0 \\ 0 & a_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.11)$$

mit der Projektionsmatrix \mathbf{P} der Kamera

$$\mathbf{P} = \mathbf{K} [\mathbf{R} \ \mathbf{t}] \quad (3.12)$$

Die Lage des Projektionszentrums C im Weltsystem ist

$$C = -R^T t \quad (3.13)$$

Die Gerade durch das Projektionszentrum C und den Bildpunkt m wird optischer Strahl oder Sehstrahl genannt. Der Punkt D auf dem optischen Strahl im Unendlichen hat die homogenen Koordinaten

$$\tilde{D} = \begin{bmatrix} D \\ 0 \end{bmatrix} \quad (3.14)$$

mit

$$D = R^T K^{-1} \tilde{m} \quad (3.15)$$

Stereosystem Bei einem Stereosystem betrachten zwei Kameras dieselbe Szene, wie im Bild 3.3 dargestellt. Der Punkt M wird in der Bildebene \mathcal{I}_1 der ersten Kamera auf den Bildpunkt m_1 abgebildet und auf den Bildpunkt m_2 in der Bildebene \mathcal{I}_2 der zweiten Kamera. Die Projektionsmatrizen für die Kameras sind P_1 und P_2 mit den Projektionszentren C_1 und C_2 . Die Projektionsgleichungen für die beiden Bilder lauten nach (3.11)

$$\tilde{m}_1 = P_1 \tilde{M} \quad (3.16)$$

$$\tilde{m}_2 = P_2 \tilde{M} \quad (3.17)$$

Zur Veranschaulichung sind in Bild 3.4 zwei Ansichten einer virtuellen Szene dargestellt.

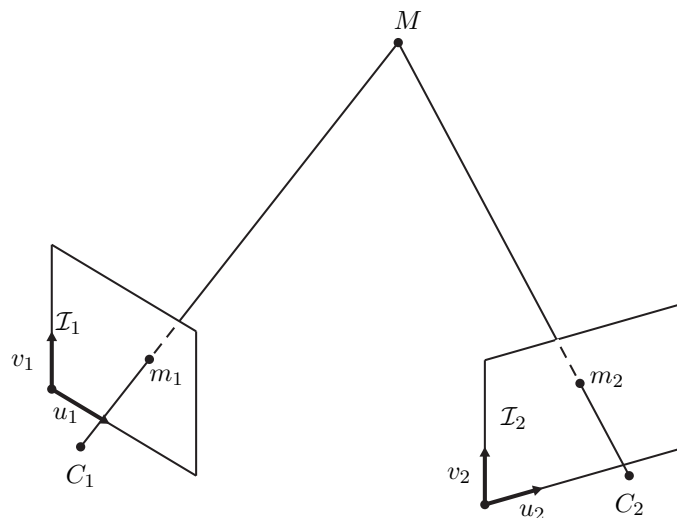


Bild 3.3: Stereosystem

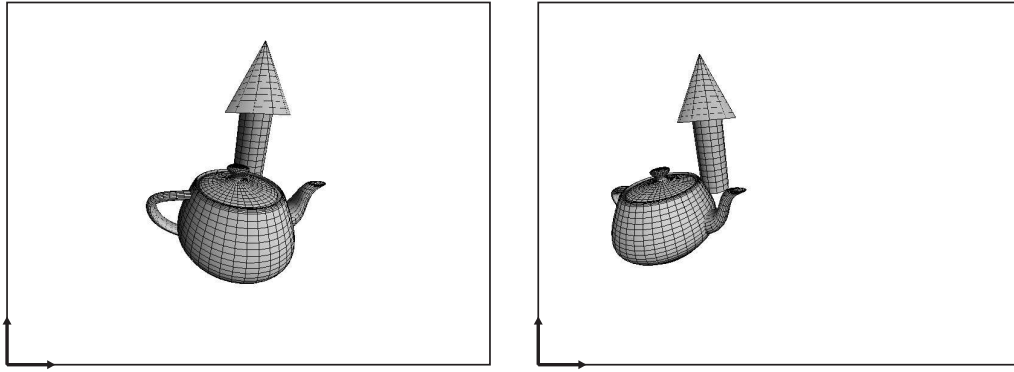


Bild 3.4: Zwei Ansichten einer virtuellen Szene

Epipolargeometrie In dem Stereosystem können über geometrische Bedingungen zusätzliche Größen definiert und neue Zusammenhänge gefunden werden, die später eine 3D Rekonstruktion erleichtern. Diese geometrischen Beziehungen bilden die *Epipolargeometrie* und sind im Bild 3.5 skizziert.

Der Raumpunkt M und die beiden Projektionszentren C_1 und C_2 bilden eine Ebene, die *Epipolarebene* \mathcal{E} . Die Schnittpunkte der Verbindungsgeraden der Projektionszentren mit den beiden Bildebenen sind die *Epipole* e_1 und e_2 . Die Schnittgeraden der Epipolarebene mit den Bildebenen sind die *Epipolarlinien* l_1 und l_2 .

Die Epipole sind die Projektionen der Zentren auf die jeweilig andere Bildebenen und berechnen sich wie folgt

$$\tilde{e}_1 = P_1 \tilde{C}_2 \quad (3.18)$$

$$\tilde{e}_2 = P_2 \tilde{C}_1 \quad (3.19)$$

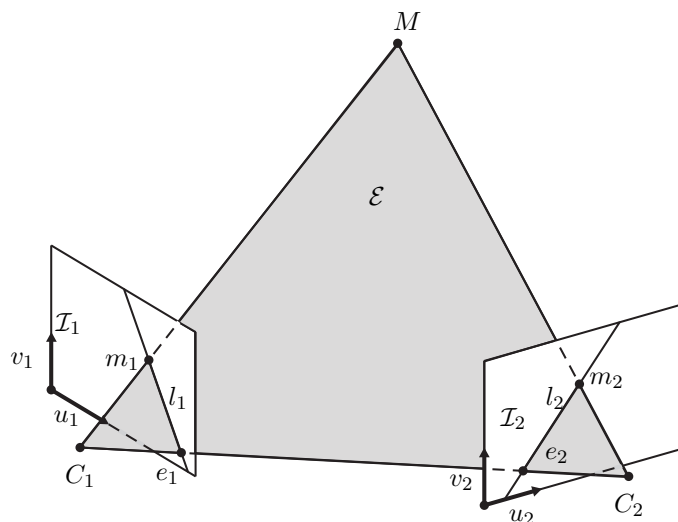


Bild 3.5: Epipolargeometrie

Epipolarlinien Zur Berechnung der Epipolarlinien werden gemäß (3.7) zwei Punkte auf der Geraden benötigt. Die folgenden Ausführungen werden für die zweite Bildebene gemacht, gelten aber in gleicher Weise für die erste Bildebene. Auf der Epipolarlinie l_2 im zweiten Bild liegt als erster benötigter Punkt der Epipol e_2 . Als zweiter Punkt wird der Punkt im Unendlichen des optischen Strahls durch m_1 verwendet. Die homogenen Koordinaten lauten nach (3.15)

$$\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{R}_1^T \mathbf{K}_1^{-1} \tilde{\mathbf{m}}_1 \\ 0 \end{bmatrix} \quad (3.20)$$

und dessen Projektion auf die zweite Bildebene

$$\tilde{\mathbf{m}}_2 = \mathbf{P}_2 \tilde{\mathbf{D}} = \mathbf{K}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{K}_1^{-1} \tilde{\mathbf{m}}_1 \quad (3.21)$$

Die Epipolarlinie l_2 lautet somit nach (3.7)

$$\begin{aligned} \tilde{\mathbf{l}}_2 &= \tilde{\mathbf{e}}_2 \times \tilde{\mathbf{m}}_2 \\ &= \tilde{\mathbf{e}}_2 \times \mathbf{K}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{K}_1^{-1} \tilde{\mathbf{m}}_1 \\ &= \mathbf{F} \tilde{\mathbf{m}}_1 \end{aligned} \quad (3.22)$$

mit der Matrix

$$\mathbf{F} = [\tilde{\mathbf{e}}_2]_{\times} \mathbf{K}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{K}_1^{-1} \quad (3.23)$$

Der schiefsymmetrische Tensor $[\tilde{\mathbf{e}}_2]_{\times}$ ist definiert als

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \quad (3.24)$$

mit dem sich das Kreuzprodukt als Matrix-Vektorprodukt schreiben lässt.

Die Matrix \mathbf{F} heißt *Fundamentalmatrix* und beinhaltet die intrinsischen und extrinsischen Matrizen der beiden Kameras. Mit der Fundamentalmatrix ist die Stereoanordnung eindeutig beschrieben. Für alle Punkte m_2 auf der Epipolarlinie l_2 gilt die Geradengleichung

$$\tilde{\mathbf{m}}_2^T \mathbf{F} \tilde{\mathbf{m}}_1 = 0 \quad (3.25)$$

Die Gleichung wird auch *Epipolargleichung* genannt. Mit demselben Vorgehen ergeben sich die Epipolarlinien l_1 in der ersten Bildebene

$$\tilde{\mathbf{l}}_1 = \mathbf{F}^T \tilde{\mathbf{m}}_2 \quad (3.26)$$

und es resultiert die Geradengleichung

$$\tilde{\mathbf{m}}_1^T \mathbf{F}^T \tilde{\mathbf{m}}_2 = 0 \quad (3.27)$$

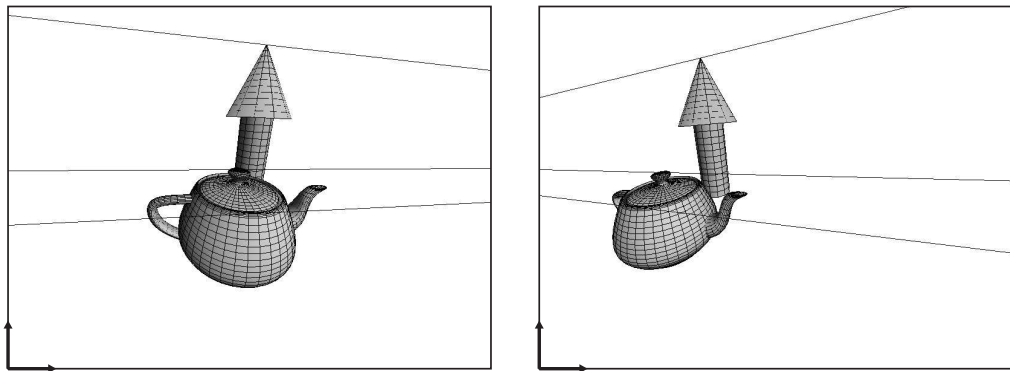


Bild 3.6: Epipolarlinien

Mit den Gleichungen (3.25) (3.27) ist es demnach möglich zu einem Punkt in dem einen Bild und gegebener Fundamentalmatrix die Epipolarlinie in dem anderen Bild zu berechnen. Im Bild 3.6 sind drei Epipolarlinien für die beiden Ansichten der virtuellen Szene eingezeichnet.

Korrespondenzproblem Für die Rekonstruktion eines 3D Punktes benötigt man ein korrespondierendes Punktepaar in den beiden Bildern. Es existieren zahlreiche Ansätze und Algorithmen [77], [78], [15], [84] zur Bestimmung dieser Punkte. Bei der Korrespondenzsuche müssen zueinander gehörende Punkte mit gewissen Umgebungen in den beiden Ansichten erkannt werden. Anstatt korrespondierende Punkte im gesamten Bild zu suchen, kann die Dimension des Suchraums reduziert werden, indem man die Erkenntnisse aus der Epipolargeometrie nutzt. Es wurde zuvor gezeigt, dass alle Punkte auf dem optischen Strahl denselben Bildpunkt ergeben. In dem anderen Bild ergeben diese Punkte die Epipolarlinie. Sucht man nun zu einem gegebenen Bildpunkt einen korrespondierenden Punkt in dem anderen Bild, so reicht es, auf der Epipolarlinie zu suchen. Das Korrespondenzproblem reduziert sich somit von einem zweidimensionalen auf ein eindimensionales. Die Voraussetzung ist, wie bereits oben erwähnt, eine gegebene Fundamentalmatrix. Dazu müssen entweder die intrinsischen und extrinsischen Kameramatrizen bekannt sein, das heißt das System muss kalibriert sein, oder es sind Punktkorrespondenzen bekannt, mit denen man die Fundamentalmatrix bestimmen kann. Mit mindestens 8 gegebenen Punktkorrespondenzen kann mit dem *8-Punkt-Algorithmus* die Matrix berechnet werden, indem man die Epipolargleichung (3.25) ausmultipliziert und das lineare Gleichungssystem löst [106], [58].

Um die eindimensionale Suche noch weiter zu vereinfachen, werden die beiden Bildebenen in eine weitere Ebene abgebildet, so dass die Epipole im Unendlichen liegen. Diese Ebene ist parallel zu der Verbindung der beiden optischen Zentren. Dadurch werden die Epipolarlinien parallel, wie im Bild 3.7 beispielhaft für die virtuelle Szene vorgeführt ist. Dieses Vorgehen wird *Rektifizierung* genannt. Die Suche vereinfacht sich dadurch nochmals, da nur noch entlang horizontaler Linien gesucht werden muss.

Ein einfacher Suchalgorithmus ist das *Block-Matching*. Dabei wird mit einem Maß

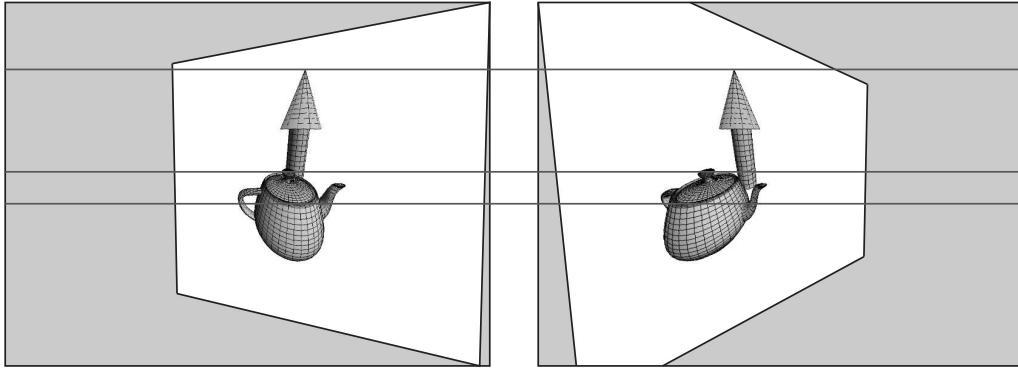


Bild 3.7: Rektifizierte Bilder

ein Bildblock des einen Bildes mit einem Bildblock des anderen Bildes entlang der Suchlinie verglichen. Ein mögliches Maß ist der mittlere quadratische Fehler (engl. sum of squared differences (SSD)). Das Minimum liefert die Verschiebung, die sogenannte *Disparität*, der korrespondierenden Bildpunkte. Es gibt eine Reihe von Erweiterungen dieses Verfahrens, wie zum Beispiel in [17] und auch Alternativen [106], [104]. Das Ergebnis ist eine Disparitätskarte, die angibt, um wie viele Pixel ein Punkt in dem anderen Bild verschoben ist. Eine weitere Möglichkeit zur Korrespondenzsuche ist der optische Fluss, der die Verschiebungsgeschwindigkeit der Bildpunkte beschreibt [79], [18]. Den optischen Fluss kann man auch für unrektifizierte Bilder berechnen, jedoch erhält man eine sehr robuste Lösung, wenn man den Algorithmus auf rektifizierte Bilder anwendet. Wurden korrespondierende Punkte in den rektifizierten Ansichten gefunden, können diese durch die inverse Abbildung der Rektifizierung auf das korrespondierende Punktepaar in den ursprünglichen Ansichten abgebildet werden.

Rekonstruktion eines Punktes Liegt ein korrespondierendes Punktepaar m_1, m_2 vor, können daraus die dreidimensionalen Koordinaten des Punktes M über eine *Triangulation* gewonnen werden. Dazu werden die beiden optischen Strahlen miteinander geschnitten und man erhält den gesuchten Raumpunkt, wie im Bild 3.3 skizziert.

Die Grundlage für die Rechnung bilden die Projektionsgleichungen (3.16), (3.17), die für die Bildpunkte zunächst ausführlich geschrieben werden. Für den Punkt m_1 im ersten Bild lautet die Gleichung

$$\tilde{m}_1 = s \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \tilde{\mathbf{M}} = \begin{bmatrix} p_1^{11} & p_1^{12} & p_1^{13} & p_1^{14} \\ p_1^{21} & p_1^{22} & p_1^{23} & p_1^{24} \\ p_1^{31} & p_1^{32} & p_1^{33} & p_1^{34} \end{bmatrix} \tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{p}_1^1 & p_1^{14} \\ \mathbf{p}_1^2 & p_1^{24} \\ \mathbf{p}_1^3 & p_1^{34} \end{bmatrix} \tilde{\mathbf{M}} \quad (3.28)$$

mit den Zeilenvektoren $\mathbf{p}_1^i = [p_1^{i1} \ p_1^{i2} \ p_1^{i3}]$. Der Skalierungsfaktor s kann durch die Bildung des Kreuzprodukts $\tilde{m}_1 \times \mathbf{P}_1 \tilde{\mathbf{M}} = \mathbf{0}$ eliminiert werden und es folgt das

Gleichungssystem

$$\begin{aligned} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \times \mathbf{P}_1 \tilde{\mathbf{M}} &= \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{p}_1^1 \mathbf{M} + p_1^{14} \\ \mathbf{p}_1^2 \mathbf{M} + p_1^{24} \\ \mathbf{p}_1^3 \mathbf{M} + p_1^{34} \end{bmatrix} \\ &= \begin{bmatrix} v_1(\mathbf{p}_1^3 \mathbf{M} + p_1^{34}) - \mathbf{p}_1^2 \mathbf{M} + p_1^{24} \\ -u_1(\mathbf{p}_1^3 \mathbf{M} + p_1^{34}) + \mathbf{p}_1^1 \mathbf{M} + p_1^{14} \\ u_1(\mathbf{p}_1^2 \mathbf{M} + p_1^{24}) - v_1(\mathbf{p}_1^1 \mathbf{M} + p_1^{14}) \end{bmatrix} = \mathbf{0} \end{aligned} \quad (3.29)$$

Von diesen Gleichungen sind zwei linear unabhängig, beispielsweise die ersten beiden Gleichungen. Für den Punkt m_2 im zweiten Bild ergeben sich ebenfalls zwei linear unabhängige Gleichungen, die man nun zu einem überbestimmten Gleichungssystem für den gesuchten Punkt M zusammensetzen kann

$$\underbrace{\begin{bmatrix} \mathbf{p}_1^1 - u_1 \mathbf{p}_1^3 \\ \mathbf{p}_1^2 - v_1 \mathbf{p}_1^3 \\ \mathbf{p}_2^1 - u_2 \mathbf{p}_2^3 \\ \mathbf{p}_2^2 - v_2 \mathbf{p}_2^3 \end{bmatrix}}_{\mathbf{A}} \mathbf{M} = \underbrace{\begin{bmatrix} u_1 p_1^{34} - p_1^{14} \\ v_1 p_1^{34} - p_1^{24} \\ u_2 p_2^{34} - p_2^{14} \\ v_2 p_2^{34} - p_2^{24} \end{bmatrix}}_{\mathbf{b}} \quad (3.30)$$

Das überbestimmte Gleichungssystem $\mathbf{A}\mathbf{M} = \mathbf{b}$ kann als ein lineares Ausgleichsproblem behandelt werden. Es ergibt sich die folgende Gleichung zur Lösung der unbekanntenen Raumkoordinaten \mathbf{M}

$$\mathbf{A}^T \mathbf{A} \mathbf{M} = \mathbf{A}^T \mathbf{b} \quad (3.31)$$

Für die Bestimmung des Raumpunktes M würden bereits drei Gleichungen reichen. Da sich aber aufgrund von Ungenauigkeiten in Bildverarbeitung die beiden optischen Strahlen nicht exakt schneiden, werden die vier Gleichungen verwendet und das Ausgleichsproblem gelöst. Dadurch wird der Raumpunkt berechnet, der minimalen Abstand zu den Strahlen hat.

3.1.2 3D Rekonstruktion

Nachdem im vorherigen Abschnitt die Formeln zur Rekonstruktion eines Punktes aus einem korrespondierenden Punktepaar aus zwei definierten Ansichten hergeleitet wurden, werden die einzelnen Schritte zur 3D Rekonstruktion des gesamten Objekts zusammengefasst. Das Ziel ist, wie eingangs erwähnt, die Erzeugung eines Volumenmodells des Objekts unter Verwendung von Bildern aus verschiedenen Ansichten.

Die ersten drei Schritte ergeben sich aus den vorgestellten Herleitungen zur Rekonstruktion eines Punktes. Die verbleibenden Schritte zeigen eine Lösung zur Generierung eines Volumennetzes aus einer Punktwolke.

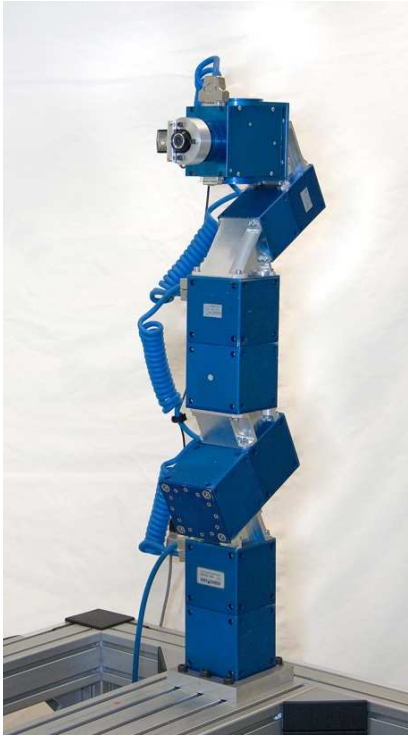
1. **Schritt:** *Aufnahme des Objekts und Speicherung der intrinsischen und extrinsischen Kameraparameter.* Für jede Ansicht muss die intrinsische und extrinsische Kameramatrix verfügbar sein. Wenn sich die Kamerageometrie nicht ändert, bleiben die intrinsischen Parameter für alle Ansichten gleich.
2. **Schritt:** *Rektifizierung der Bildpaare und Korrespondenzsuche.* Über die Epipolargeometrie wird jedes Bildpaar rektifiziert und ein anschließender Suchalgorithmus erzeugt eine Liste von korrespondierenden Punkten. Hier kann man zusätzliche Filteralgorithmen einbinden um die Anzahl der Punkte zu begrenzen und die Qualität zu erhöhen. So kann man die Verarbeitung auf bestimmte Bereiche der Bilder einschränken oder die Korrespondenzsuche nur auf charakteristische Punkte, sogenannte *Features*, anwenden. Durch die Umkehrabbildung der Rektifizierung erhält man die korrespondierenden Punkte in den ursprünglichen Ansichten.
3. **Schritt:** *3D Rekonstruktion der Punkte durch Triangulation.* Für jedes Punktepaar wird eine Triangulation durchgeführt und es ergibt sich eine Punktwolke im 3D Raum. Hier können Ausreißer, die aus einer schlechten Punktekorrespondenz entstehen, aussortiert werden.
4. **Schritt:** *Erzeugung eines Oberflächennetzes.* Die Punktwolken aus den Bildpaaren werden zusammengeführt und mit Dreiecken vernetzt. Falls das Oberflächennetz verrauscht ist, kann noch eine Glättung des Netzes erfolgen.
5. **Schritt:** *Erzeugung eines Volumennetzes.* Aus dem Oberflächennetz wird ein Volumennetz aus Tetraedern erstellt. Da das Netz die Kondition der Systemmatrizen der FEM beeinflusst, sollte das Netz möglichst keine Tetraeder mit spitzen Winkeln beinhalten.

3.1.3 Experimente

Die einzelnen Schritte des Vorgehens werden im Folgenden anhand eines Beispiels demonstriert. Als Experimentierplattform wird ein Roboter mit montierter Kamera verwendet, der im Bild 3.8 dargestellt ist. Der Roboter besteht aus Amtec Powercube Modulen und besitzt sechs Freiheitsgrade. Eine genaue Beschreibung des Roboters ist in der Arbeit [43] gegeben. Die Kommunikation der Module erfolgt über einen CAN-Bus. Die Kamera ist an der Spitze der Roboters befestigt und kann im Arbeitsraum beliebig positioniert werden. Mit diesem Aufbau aus Kamera und Roboter ist es möglich die extrinsischen Parameter der Kamera zu bestimmen.

Die Kamera ist eine Logitech Quickcam Pro for Notebooks und ist im Bild 3.9 abgebildet. Die 2 Megapixel Kamera kann Bilder bis zu einer Auflösung von 1600×1200 Pixel aufnehmen.

Die Kalibrierung, das heißt die Identifikation der intrinsischen Matrix, kann einmalig zu Beginn durchgeführt werden, da sie sich im Weiteren für eine gewählte Bildauflösung nicht mehr ändert. Dazu müssen mehrere Bilder eines vermessenen Referenzobjekts aufgenommen werden. Mit den bekannten Weltkoordinaten und zugehörigen

**Bild 3.8:** Roboter mit Kamera**Bild 3.9:** Logitech Kamera

Bildkoordinaten des Objekts können die unbekannt Parameter der intrinsischen Matrix bestimmt werden. Dafür geeignete Algorithmen sind in [106], [58], [47], [48] beschrieben. Hier wird ein Kalibrierungsalgorithmus aus dem Paket OPENCV [2] genutzt, der als bekanntes Objekt ein Schachbrettmuster verwendet. Im Bild 3.10 sind mehrere Aufnahmen für die Kalibrierung der Kamera abgebildet. Für die Logitech Kamera ergibt sich so die intrinsische Matrix

$$\mathbf{K} = \begin{bmatrix} 1.347 \cdot 10^3 & 0 & 829 \\ 0 & 1.361 \cdot 10^3 & 566 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

Werden die Kalibrierungsaufnahmen mit der am Roboter montierten Kamera gemacht, so kann ebenfalls eine Kalibrierung des Roboters bezüglich des Referenzobjekts durchgeführt werden. Dadurch können Ungenauigkeiten in der Roboterkinematik kompensiert werden und das System besitzt eine höhere Absolutgenauigkeit.

Das zu rekonstruierende Objekt in diesem Experiment ist ein Kunststoffkrokodil. Im Bild 3.11 ist der Roboter mit dem Objekt zu sehen. Der erste Schritt ist die Aufnahme des Objekts aus mehreren Ansichten. Dazu wird zu jedem Bild die intrinsische und extrinsische Kameramatrix gespeichert, während sich der Roboter um das Objekt herum bewegt. Bild 3.12 zeigt mehrere Ansichten des Objekts. Da die Objektoberfläche glatt ist und eine gleichmäßige Farbe besitzt, wurde zusätzlich ein Rauschbild auf das Objekt projiziert um die Korrespondenzanalyse zu erleichtern. Dadurch ist das Bild kontrastreicher und die Algorithmen funktionieren robuster. Aus zwei gewählten Ansichten kann die Fundamentalmatrix (3.23) aus der gemeinsa-

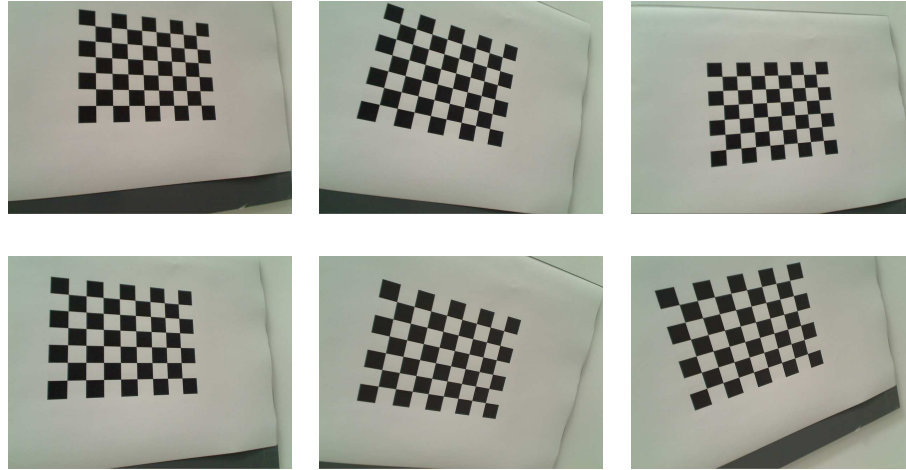


Bild 3.10: Kalibrierung der Kamera mit einem Schachbrettmuster

men intrinsischen und den zwei extrinsischen Matrizen berechnet werden und damit auch die Epipolarlinien. Diese sind im Bild 3.13 für zwei Ansichten eingezeichnet. Die explizite Berechnung der Epipolarlinien ist in diesem Verfahren nicht notwendig, wird aber für eine verbesserte Anschauung hier mit aufgeführt. Die beiden Ansichten werden nun rektifiziert, wie im Bild 3.14 zu sehen ist, um die Epipolarlinien horizontal auszurichten. Weiter erfolgt im zweiten Schritt des Vorgehens die Korrespondenzanalyse. Sie liefert zunächst Punktepaare in den rektifizierten Ansichten, die im Bild 3.15 eingezeichnet sind. Nach der Umkehrabbildung der Rektifizierung erhält man die korrespondierenden Punkte in den beiden ursprünglichen Ansichten, wie im Bild 3.16 dargestellt. Für jedes Punktepaar wird mittels der Triangulation der entsprechende Raumpunkt berechnet. Die dabei entstehende Punktwolke ist im Bild 3.17 zu sehen, wobei die Kontur eines Teiles des Objekts erkennbar ist. Die Punktwolken aus den Ansichtspaaren werden zu einer gemeinsamen zusammengefügt und man erkennt im Bild 3.18 die Form des ganzen Objekts. Aus der Gesamtpunktwolke wird in dem vierten Schritt ein Oberflächennetz mit dem Algorithmus POWERCRUST [4] und dem Programm MESHLAB [25] erstellt. Die Objektoberfläche aus Dreiecken ist im Bild 3.19 zu sehen. Das Programm bietet zudem die Möglichkeit das Netz zu glätten und zu vereinfachen. Der letzte Schritt ist die Erzeugung eines Volumennetzes mit Tetraedern aus dem Oberflächennetz, wie es im Bild 3.20 dargestellt ist. Dafür wurde das Programm TETGEN [110] verwendet, welches als Option die Generierung von Tetraedern mit spitzen Winkeln vermeidet. Das resultierende Tetraedernetz kann bei Modellierung mit der Finite-Elemente-Methode verwendet werden, womit die geometrischen Modellparameter bekannt sind. Für die vollständige Modellbeschreibung verbleiben die Materialparameter, deren Identifikation im nachfolgenden Abschnitt behandelt wird.

Die Genauigkeit des Rekonstruktionsverfahrens wird von mehreren Faktoren beeinflusst, wie beispielsweise der Kameraauflösung oder dem Abstand der Kamera zum Objekt. Für eine Genauigkeitsaussage kann ein Größenvergleich des Testobjekts verwendet werden. Aus der Rekonstruktion resultiert eine Länge von 245 mm, eine Höhe von 95 mm und eine Breite von 101 mm. Mit einer separat durchgeführten Längen-

messung ergibt sich eine Länge, Höhe und Breite von 241 mm, 96 mm und 102 mm. Somit weist die Objektdimension in diesem Experiment einen relativen Fehler von weniger als 2% auf.

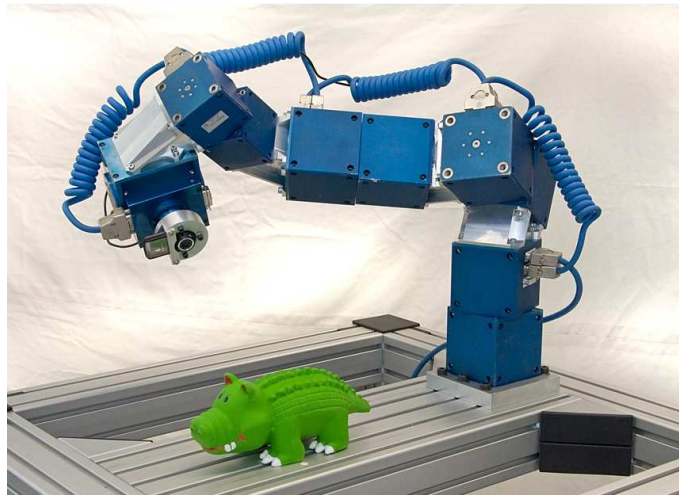


Bild 3.11: Roboter mit Objekt

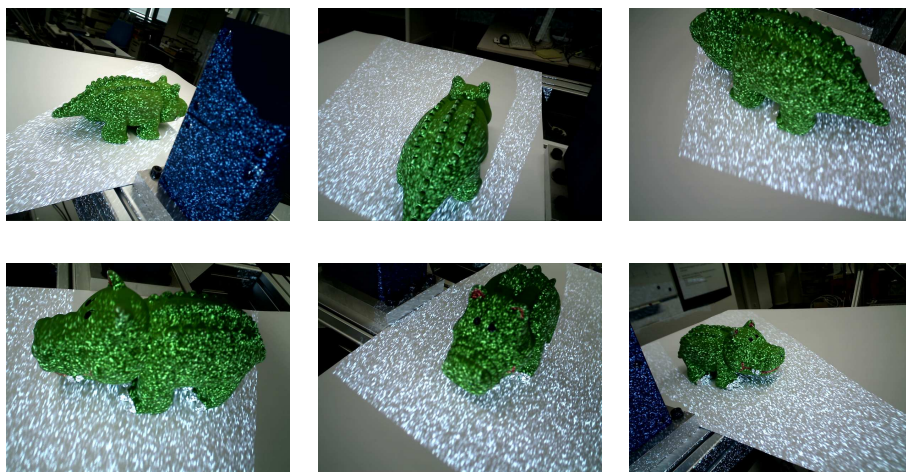


Bild 3.12: Objekt in vielen Ansichten

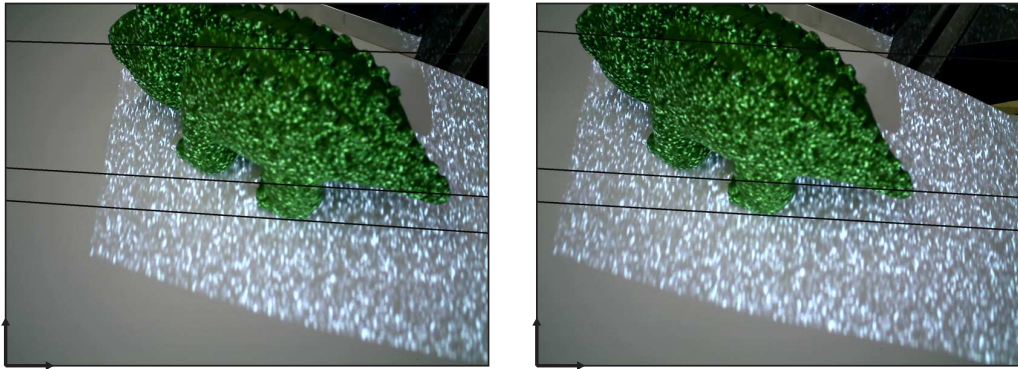


Bild 3.13: Objekt in zwei Ansichten mit Epipolarlinien

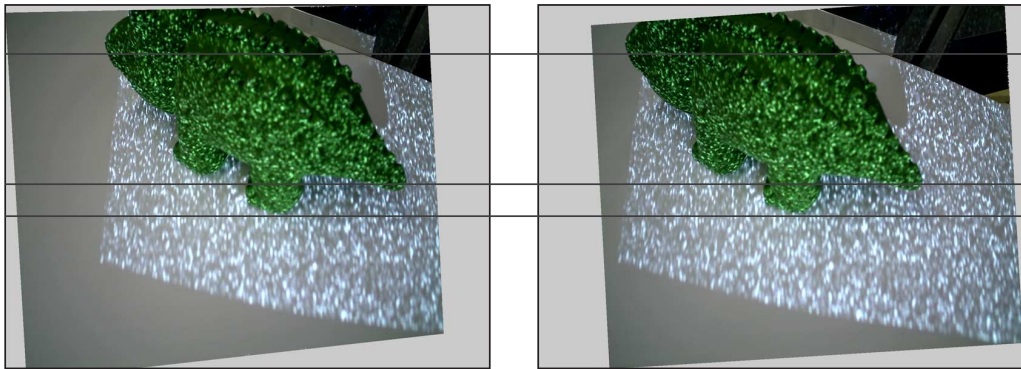


Bild 3.14: Rektifizierte Ansichten

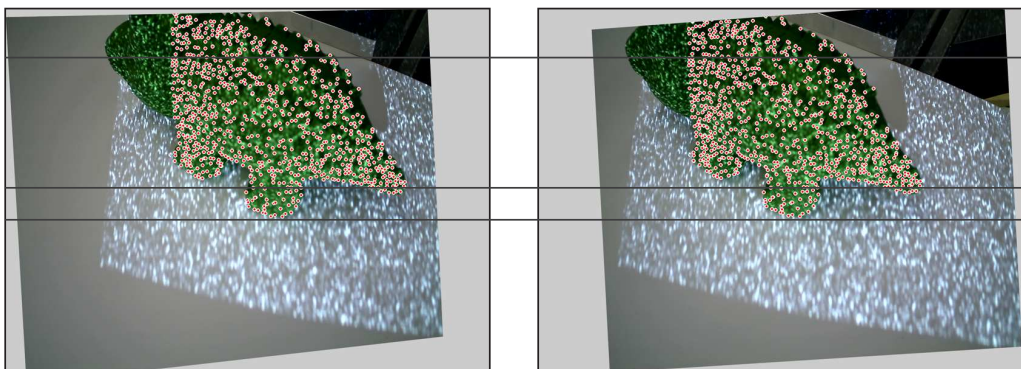


Bild 3.15: Korrespondierende Punkte in den rektifizierten Ansichten

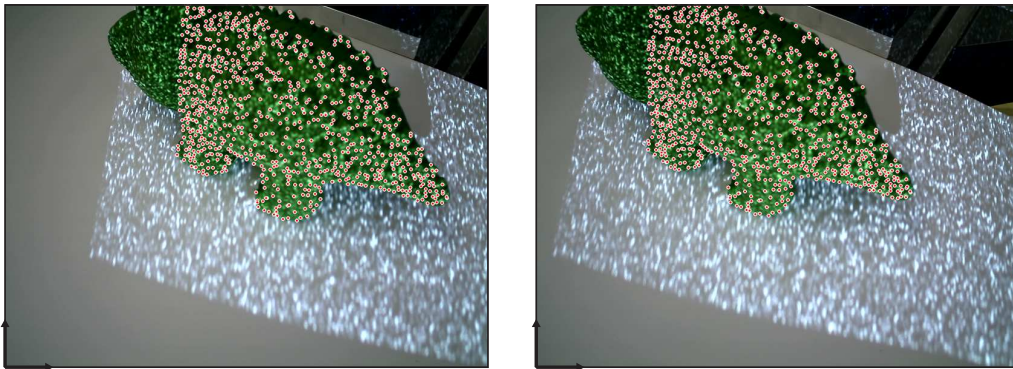


Bild 3.16: Korrespondierende Punkte in den ursprünglichen Ansichten

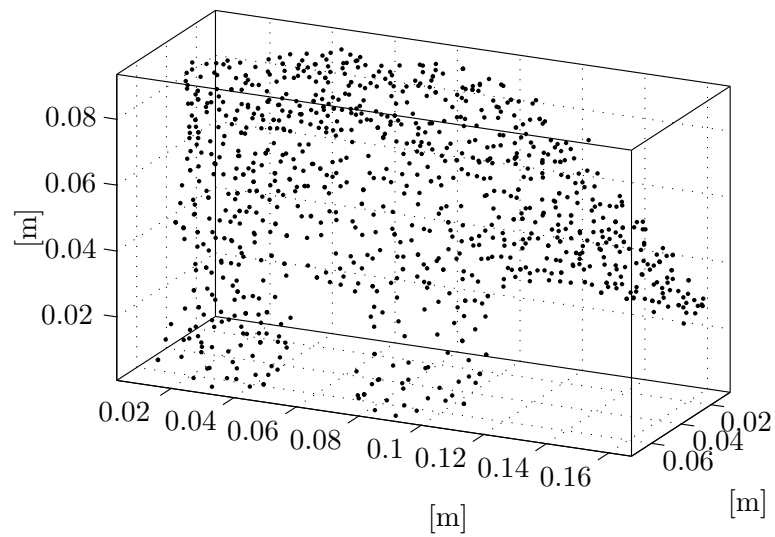


Bild 3.17: Punktwolke aus einem Ansichtspaar

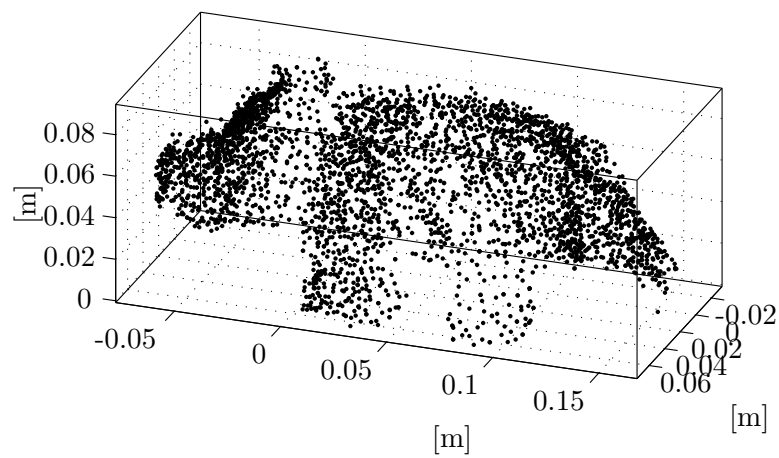


Bild 3.18: Zusammengesetzte Punktwolke

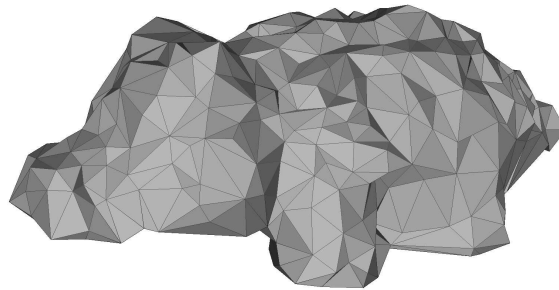


Bild 3.19: Oberflächennetz

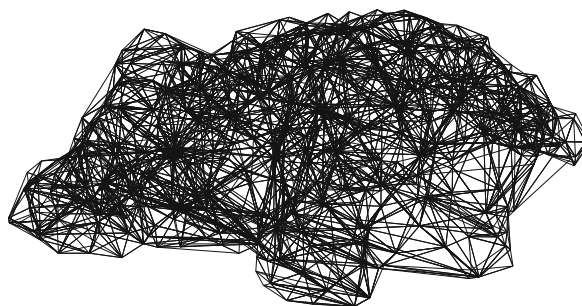


Bild 3.20: Volumennetz aus Tetraedern

3.2 Identifikation der Materialparameter

Neben einer Beschreibung der Geometrie werden bei einer FEM-Modellierung Parameter zur Beschreibung des Materials benötigt. Bei einem St. Venant-Kirchhoff-Material sind die zwei Parameter der Elastizitätsmodul und die Querkontraktionszahl (2.71). Ein Vorgehen zur experimentellen Identifikation der Materialparameter wird in diesem Abschnitt vorgestellt.

Die Identifikation erfolgt mit einer Optimierung, in der Simulationsergebnisse mit Messdaten abgeglichen werden. Das Simulationsmodell wird mittels der Finite-Elemente-Methode erstellt, wie es in Kapitel 2 erläutert wurde. Dies setzt voraus, dass eine Geometriebeschreibung des Objekts bereits vorliegt. Als Messdaten für die Materialparameteridentifikation werden die Verformung des Objekts und die Kontaktkraft verwendet, die während eines Deformationsvorgangs aufgezeichnet werden.

3.2.1 Optimierung der Materialparameter

Der Ablauf der Optimierung ist im Bild 3.21 skizziert. Die Simulation für deformierbare Körper wird mit einem Optimierungsalgorithmus gekoppelt. Die Optimierungsparameter sind die Materialparameter \mathbf{p} der Körper. Das Simulationsmodell wird an den gemessenen Verformungen \mathbf{U}_{mess} ausgewertet. Als Gütefunktional wird die Norm der Differenz zwischen gemessener Kraft \mathbf{F}_{mess} und simulierter Kraft \mathbf{F}_{sim} verwendet. Das Funktional soll minimiert werden und man erhält die Optimierungsaufgabe

$$\min_{\mathbf{p}} \|\mathbf{F}_{\text{mess}} - \mathbf{F}_{\text{sim}}(\mathbf{p})\| \quad (3.33)$$

Die simulierte Kraft \mathbf{F}_{sim} resultiert aus der Auswertung des Kräftegleichgewichts aus dem vorherigen Kapitel. Da sich die Gleichgewichtsgleichungen für die vorgestellten Modellierungsansätze, nichtlinear, linear oder korotiert, unterscheiden, kann die Materialparameteroptimierung jeweils nur für einen gewählten Ansatz erfolgen. Für die Optimierung wird der statische Fall des Kräftegleichgewichts verwendet. Die kinetischen Kräfte verschwinden und es wird die Dynamik des Objekts nicht identifiziert. Eine Berücksichtigung der Dynamik kann mit demselben prinzipiellen Optimierungsaufbau geschehen. Der Vektor der Optimierungsparameter muss dafür

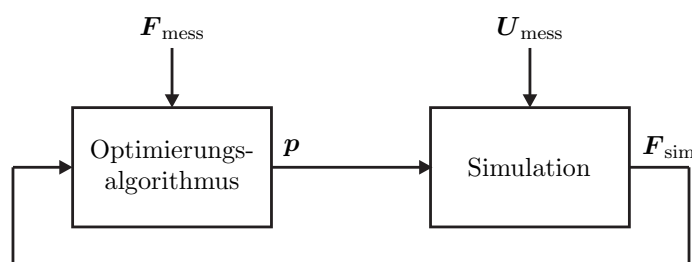


Bild 3.21: Optimierungsschleife

um die Dynamikparameter des Objekts, wie zum Beispiel Dichte oder Dämpfung, entsprechend erweitert werden.

3.2.2 Experimente

Zur Demonstration der experimentellen Materialidentifikation kommt ebenfalls das Kunststoffkrokodil als Testobjekt zum Einsatz. Die Objektgeometrie ist bereits in dem vorherigen Experiment bestimmt worden und wird nun als bekannt weiter verwendet. Die Modellierung des deformierbaren Körpers erfolgt mit dem statischen, linearen FEM-Ansatz.

Für die Experimente und Messungen wird wie im vorherigen Versuch der Roboter eingesetzt, wobei anstelle der Kamera ein Kraft-Momentensensor mit einer Werkzeugschuppe montiert ist. Der Roboter mit Sensor ist im Bild 3.22 abgebildet. Als Sensor wird das Modell Mini40 SI-80-4 der Firma ATI verwendet. In dem Experiment fährt der Roboter eine vorprogrammierte Bahn ab, wobei die Roboterspitze auf das Testobjekt drückt, wie im Bild 3.23 zu sehen ist. Die Eindringtiefe wird schrittweise um einen Millimeter erhöht und die Sensorkräfte werden zusammen mit der Roboterposition aufgezeichnet. Die vertikale Position der Roboterspitze und die entsprechende Kontaktkraft sind im Bild 3.24 zu sehen. Da sich zu Beginn des Experiments der Roboter noch nicht in Kontakt mit dem Testobjekt befindet, ist die Kraft Null und steigt erst bei Kontakt in etwa linear an. Die lokalen Spitzen im Kraftverlauf sind die Folge des Regelfehlers beim Positioniervorgang.

Für die Optimierung wird nicht der gesamte Kraft- und Verformungsverlauf verwendet, sondern nur Messungen während der Roboter sich in Ruhe befindet. In diesem Fall ergeben sich 25 Messpaare für die Optimierung. Der Optimierungsalgorithmus ist in Matlab implementiert und ist ein direktes Simplex-Suchverfahren (Matlab-Funktion *fminsearch*), das für eine nichtlineare Optimierung mit mehreren Variablen geeignet ist. Die gemessene und die mit optimierten Materialparametern simulierte Position-Kraft-Beziehung ist im Bild 3.25 dargestellt. Da näherungsweise ein lineares Verhalten in der Messung zu erkennen ist, ist die Wahl eines linearen Simulationsmodells gerechtfertigt. Der Optimierungsalgorithmus konvergiert und liefert den Elastizitätsmodul $E = 25529 \frac{\text{N}}{\text{m}^2}$ und die Querkontraktionszahl $\nu = 0.01$. Ein Vergleich des gemessenen Kraftverlaufs mit dem simulierten Kraftverlauf ist im Bild 3.26 zu sehen.

Die Modellierung erfolgt mit einem Volumenmodell aus Tetraedern, obwohl das untersuchte Objekt hohl ist. Dieses Experiment bestätigt durch die sehr gute Übereinstimmung von Simulation und Messung die Erkenntnis aus anderen durchgeführten Untersuchungen, dass dieses Ersatzmodell mit optimierten Materialparametern in der Lage ist hohle Objekte realistisch abzubilden.

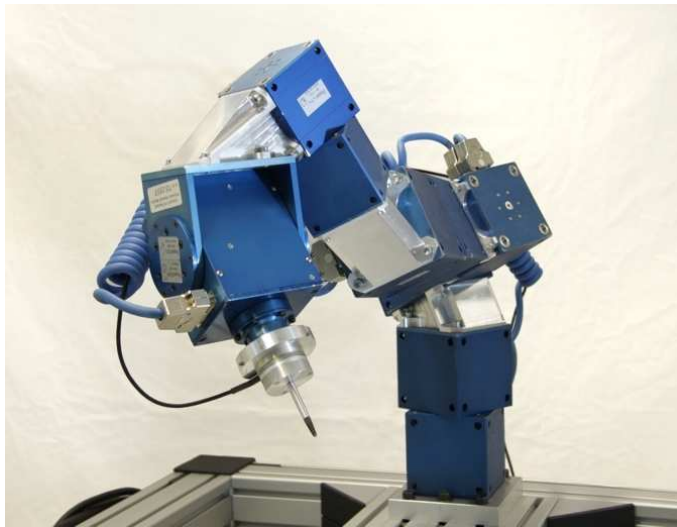


Bild 3.22: Roboter mit Kraft-Momentensensor

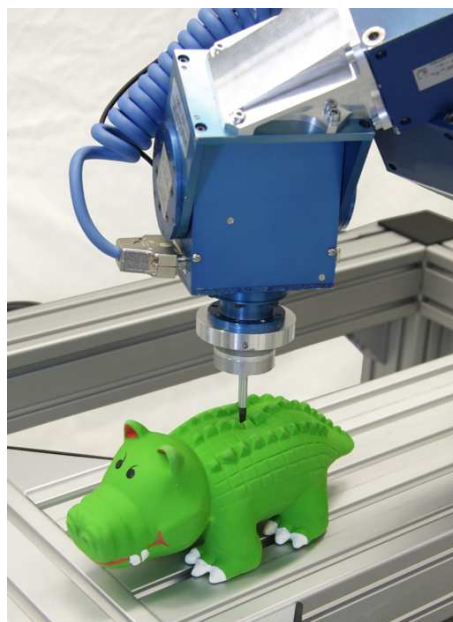


Bild 3.23: Roboter drückt auf Testobjekt

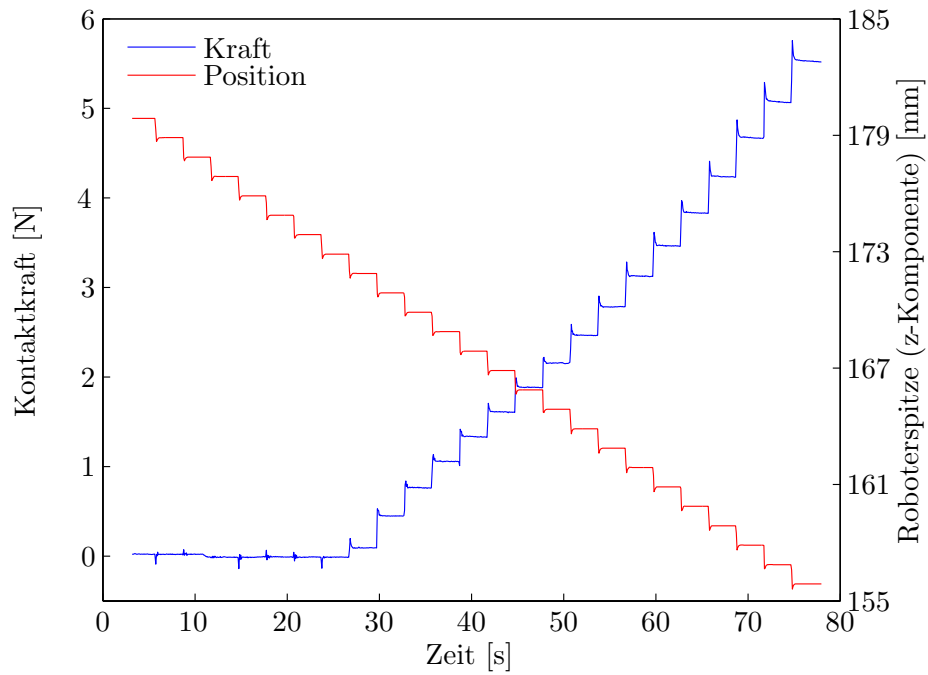


Bild 3.24: Kraft- und Positionsverlauf während des Drückvorgangs

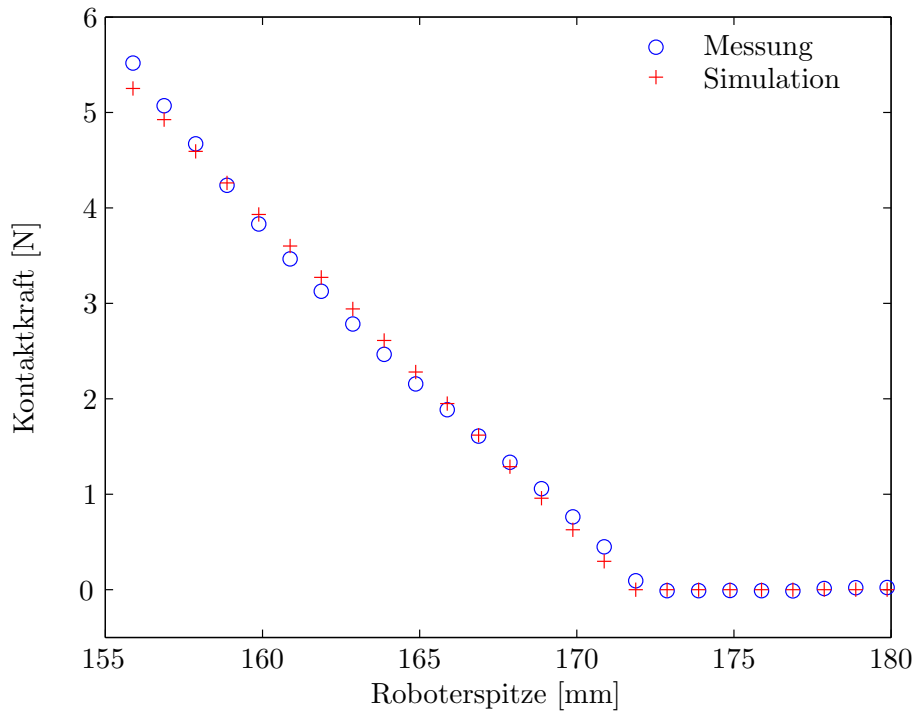


Bild 3.25: Position-Kraft-Beziehung nach der Optimierung

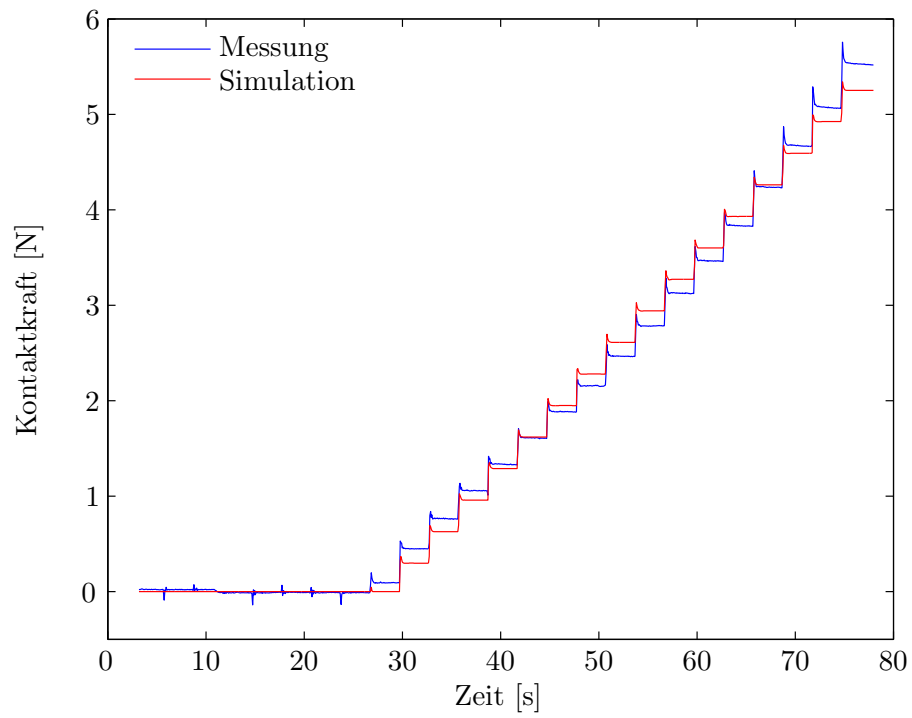


Bild 3.26: Vergleich Kraftverlauf Messung Simulation

4 Simulation von weichen Kontakten

Die Simulation von weichen Kontakten beinhaltet die Berechnung der Verformung eines elastischen Körpers auf Grund von äußeren Einwirkungen. Zu Grunde liegt dabei das Kräftegleichgewicht, das mit einer FEM-Modellierung hergeleitet wurde. Die Simulation entspricht somit der numerischen Auswertung der Gleichgewichtsbedingungen. Je nach Anforderung und Modellierung ergibt sich eine Simulation mit unterschiedlichen Rechenabläufen. In diesem Kapitel werden diese Abläufe analysiert und in Algorithmen zusammengefasst. Außerdem wird untersucht, bei welchen Schritten des Simulationsablaufs sich Größen ändern oder während eines Simulationslaufs konstant bleiben. Diese Analyse ermöglicht später eine effiziente und rechenzeitoptimierte Implementierung. Abschließend werden die Rechenzeiten der einzelnen Algorithmen gegenübergestellt.

In der vorliegenden Arbeit wird die Simulation von deformierbaren Körpern als eigenständiges Modul betrachtet, das selbst wieder ein Teil eines größeren Systemaufbaus sein kann. Das bedeutet, dass andere Module über eine Schnittstelle mit der Simulation kommunizieren können und somit über Eingänge Daten an die Simulation liefern und über Ausgänge Daten erhalten können. Beispielsweise können an festgelegten Netzknoten die Verschiebungen oder externe Kräfte vorgegeben werden, oder, wenn der Körper mit einem weiteren Körper in Kontakt kommt, die Kontaktknotenindizes mit einer Kollisionserkennung ermittelt werden. Die Simulation des deformierbaren Körpers liefert dann Verformungen oder Reaktionskräfte zurück. Innerhalb der Simulation findet somit keine geschlossene Lösung eines Kontaktproblems statt, wie sie beispielsweise in [124] mittels Lagrange-Multiplikatoren für einen Kontakt zwischen zwei deformierbaren Körpern ermittelt wird.

Die Aufstellung der Systemgleichungen und des Simulationsablaufs erfordert zunächst eine Klassifizierung der an die Simulation gestellten Anforderungen. Dies kann anhand von drei Kriterien geschehen, die im Bild 4.1 dargestellt sind. Das erste Kriterium ist der Modellierungsansatz. So kann zwischen linearer, nichtlinearer und korotierter FEM unterschieden werden. Als Zweites kann die Simulation in statische und dynamische Problemstellungen eingeteilt werden. Der dritte Klassifizierungspunkt ist die Schnittstellendefinition der Simulation. Hier wird festgelegt, auf welche Art mit der Simulation interagiert wird. Als Simulationseingang kann die Randkraft oder die Randverschiebung gewählt werden. Aufgrund dieser Klassifizierungskriterien ergeben sich unterschiedliche Gleichungssysteme und Varianten von Simulationsabläufen. Die Analyse der Simulation hinsichtlich der einzelnen Klassifizierungspunkte erfolgt in den folgenden Abschnitten, wobei die Unterscheidung bezüglich der Modellierung bereits ausführlich in Kapitel 2 erfolgte.

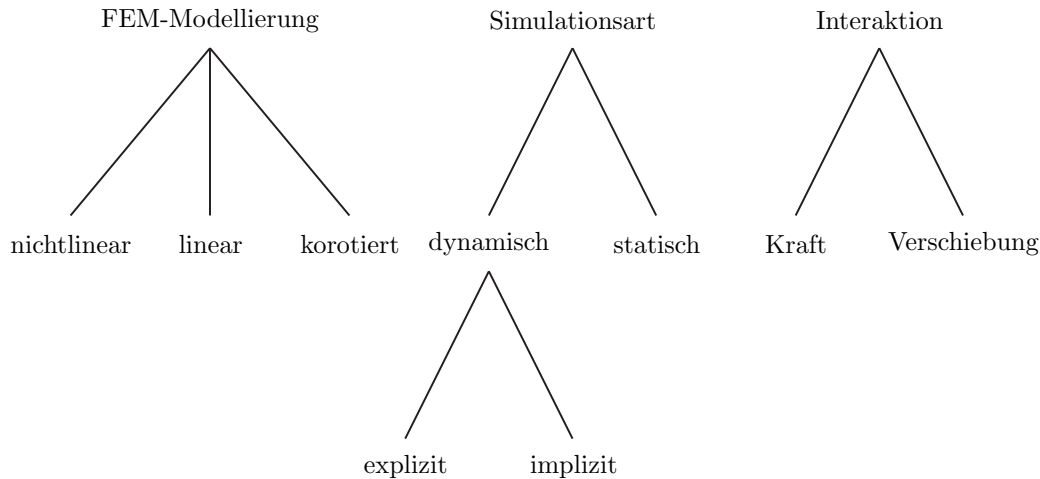


Bild 4.1: Klassifizierung der Simulationen

Der algorithmische Ablauf der Simulation erfolgt in mehreren Schritten. Für die weiteren Erklärungen ist es hilfreich, zwischen den verschiedenen Schritten und Zeitpunkten innerhalb eines Simulationslaufs zu unterscheiden. Die Zeitpunkte, an denen die Simulation über die Ein- und Ausgänge der Schnittstellen Daten austauscht, sind die Simulationszeitpunkte t_I mit entsprechenden Simulationsschritten I . Die Zeitpunkte der Zeitintegration sind die t_i mit den Integrationsschritten i . Innerhalb eines Simulationsschritts können auch mehrere Zeitintegrationsschritte erfolgen. Der Laufindex in einem iterativen Lösungsverfahren, wie zum Beispiel dem Newton-Verfahren für nichtlineare Gleichungssysteme, ist k .

In diesem Kapitel wird eine kleine Änderung der Notation vorgenommen. Im Kapitel 2 wurden die kontinuierlichen Größen des Körpers, wie beispielsweise der Ortsvektor \mathbf{X} (2.1) oder die Verschiebung \mathbf{u} (2.6), ohne überstehenden Zeichen notiert. Um die diskretisierten Größen davon zu unterscheiden, wurden diese mit einem Kreisakzent $^\circ$ gekennzeichnet, wie zum Beispiel $\mathring{\mathbf{X}}$ (2.90) und $\mathring{\mathbf{u}}$ (2.91). Da im Folgenden stets mit diskretisierten Größen gerechnet wird, wird der Kreisakzent für eine erleichterte Lesbarkeit weggelassen.

4.1 Simulationsarten

In jedem Simulationsschritt ist die Lösung des Kräftegleichgewichts (2.114) des deformierbaren Körpers gesucht

$$\mathbf{f}_{\text{int}} - \mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{kin}} = \mathbf{0} \quad (4.1)$$

Dies ist die zentrale Gleichung der gesamten Simulation, die im Folgenden analysiert und algorithmisch aufbereitet wird. Das erste Klassifizierungskriterium ist die Simulationsart, bei der man zwischen einer statischen und einer dynamischen Problemstellung unterscheiden kann.

4.1.1 Statische Simulation

Bei einer statischen Simulation ist die Geschwindigkeit und Beschleunigung gleich Null und es folgt, dass die kinetischen Kräfte verschwinden

$$\mathbf{f}_{\text{kin}} = \mathbf{0} \quad (4.2)$$

Das Gleichungssystem (4.1) reduziert sich im statischen Fall zu

$$\mathbf{f}_{\text{int}} = \mathbf{f}_{\text{ext}} \quad (4.3)$$

Dieses Gleichungssystem muss in jedem Simulationsschritt I gelöst werden, Zeit-integrationsschritte i existieren in diesem Fall nicht.

Bei der Modellierung des deformierbaren Körpers mit der nichtlinearen FEM sind die internen Kräfte \mathbf{f}_{int} eine nichtlineare Funktion der Verschiebungen \mathbf{u} . Das statische Kräftegleichgewicht (4.3) ist ein nichtlineares Gleichungssystem, das iterativ mit dem Newton-Verfahren gelöst werden kann. Für den Algorithmus des Newton-Verfahrens wird die Linearisierung (2.117) verwendet, die im statischen Fall an der Stelle \mathbf{u}^k lautet

$$\mathbf{f}_{\text{int}}^k - \mathbf{f}_{\text{ext}} + \mathbf{K}_{\text{tang}}^k (\mathbf{u}^{k+1} - \mathbf{u}^k) = \mathbf{0} \quad (4.4)$$

Mit $\Delta\mathbf{u}^k = \mathbf{u}^{k+1} - \mathbf{u}^k$ lautet somit der Iterationsalgorithmus für das Newton-Verfahren

Algorithmus 4.1 : Newton-Verfahren für statische, nichtlineare FEM

```

foreach Simulationsschritt  $I$  do
  | for Newton-Iteration  $k = 1$  to  $k_{\text{max}}$  do
  | | Berechne  $\mathbf{K}_{\text{tang}}^k = \mathbf{K}_{\text{tang}}(\mathbf{u}^k)$ 
  | | Löse  $\mathbf{K}_{\text{tang}}^k \Delta\mathbf{u}^k = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}^k$ 
  | |  $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}^k$ 
  | end
end

```

In jedem Iterationsschritt k müssen die Kräfte $\mathbf{f}_{\text{int}}^k$ und die tangentiale Steifigkeitsmatrix $\mathbf{K}_{\text{tang}}^k$ neu berechnet werden, da diese von der Verformung abhängen.

Da die Berechnung der tangentialen Steifigkeitsmatrix rechenaufwendig sein kann, kann sie auch nur für jeden neuen Simulationsschritt einmal berechnet und während der Iteration beibehalten werden. Man erhält dadurch das sogenannte *modifizierte*

Newton-Verfahren, welches eine geringere Konvergenzordnung besitzt. Der Algorithmus für das modifizierte Newton-Verfahren lautet

Algorithmus 4.2 : Modifiziertes Newton-Verfahren für statische, nichtlineare FEM

```

foreach Simulationsschritt  $I$  do
  Berechne  $\mathbf{K}_{\text{tang}} = \mathbf{K}_{\text{tang}}(\mathbf{u}^0)$ 
  for Newton-Iteration  $k = 1$  to  $k_{\text{max}}$  do
    Löse  $\mathbf{K}_{\text{tang}}\Delta\mathbf{u}^k = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}^k$ 
     $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}^k$ 
  end
end

```

Die Problematik der Eindeutigkeit der Lösung und Singularitäten der tangentialen Steifigkeitsmatrix werden hier nicht näher behandelt. Eine ausführliche Darstellung dieser Thematik, von pfadabhängigen Lösungen und Pfadverfolgungsalgorithmen findet in den Büchern [123] und [96] statt.

Das Kräftegleichgewicht der linearen FEM (2.136) ist im statischen Fall ein lineares Gleichungssystem und lautet

$$\mathbf{K}_{\text{lin}}\mathbf{u} = \mathbf{f}_{\text{ext}} \quad (4.5)$$

Die lineare Steifigkeitsmatrix \mathbf{K}_{lin} ist unabhängig von der Verformung und muss bei einer Simulation daher nur einmal berechnet werden. Jeder Simulationsschritt erfordert somit die Lösung des linearen Gleichungssystems (4.5). Der Algorithmus lautet

Algorithmus 4.3 : Lösung für statische, lineare FEM

```

Berechne  $\mathbf{K}_{\text{lin}}$ 
foreach Simulationsschritt  $I$  do
  Löse  $\mathbf{K}_{\text{lin}}\mathbf{u} = \mathbf{f}_{\text{ext}}$ 
end

```

Das Kräftegleichgewicht für die Modellierung mit der korotierten FEM lautet gemäß Gleichung (2.146)

$$\mathbf{K}_{\text{korot}}\mathbf{u} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{korot}} \quad (4.6)$$

Bei der korotierten FEM muss in jedem Simulationsschritt ein lineares Gleichungssystem gelöst werden. Die korotierte Steifigkeitsmatrix $\mathbf{K}_{\text{korot}}$ und der korotierte Kraftvektor $\mathbf{f}_{\text{korot}}$ hängen dabei von einer Vorgabeverformung $\bar{\mathbf{u}}$ ab und müssen folglich in jedem Simulationsschritt ausgewertet werden. Der Aufwand dieser Berechnung verringert sich, wenn eine Berechnung der linearen Steifigkeitsmatrix bereits erfolgt ist, da sich die korotierte Steifigkeitsmatrix gemäß (2.142) aus der Multiplikation der linearen Steifigkeitsmatrix mit einer Rotationsmatrix ergibt. In der Regel wird bei der korotierten FEM als Vorgabeverformung $\bar{\mathbf{u}}$ die Verformung des letzten

Simulationsschritts genommen. Der Algorithmus für die statische, korotierte FEM lautet also

Algorithmus 4.4 : Lösung für statische, korotierte FEM

```

foreach Simulationsschritt  $I$  do
  | Berechne  $\mathbf{K}_{\text{korot}} = \mathbf{K}_{\text{korot}}(\bar{\mathbf{u}})$ 
  | Löse  $\mathbf{K}_{\text{korot}} \mathbf{u} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{korot}}$ 
end

```

4.1.2 Dynamische Simulation

Bei einer dynamischen Simulation wird das vollständige Kräftegleichgewicht (4.1) gelöst. Mit der kinetischen Kraft aus (2.116) lautet die Differentialgleichung 2. Ordnung

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}_{\text{int}}(\mathbf{u}) = \mathbf{f}_{\text{ext}} \quad (4.7)$$

Die Lösung der Gleichung erfolgt in der Regel durch eine Zeitintegration mit einem gewählten Diskretisierungsverfahren. Hierzu existieren in der Literatur zahlreiche Methoden [35], [113]. In dieser Arbeit werden zwei Ansätze vorgestellt, die insbesondere bei der FEM-Simulation weit verbreitet sind. Das erste ist das *zentrale Differenzenverfahren* und das zweite ist das *Newmark-Verfahren*. Das zentrale Differenzenverfahren ist ein explizites Integrationsverfahren, während dagegen das Newmark-Verfahren ein implizites Verfahren ist. Beide Verfahren verwenden in der hier vorgestellten Form eine konstante Zeitschrittweite Δt .

Die Bewegungsgleichung wird oftmals um einen Dämpfungsterm erweitert, um die Strukturdämpfung des Körpers zu berücksichtigen. Dies kann durch Einführung einer konstanten Dämpfungsmatrix \mathbf{D} geschehen, die man als eine Kombination aus Massen- und linearer Steifigkeitsmatrix ansetzen kann. Mit den Faktoren a und b resultiert die Dämpfungsmatrix

$$\mathbf{D} = a\mathbf{M} + b\mathbf{K}_{\text{lin}} \quad (4.8)$$

Mit der Dämpfungsmatrix erweitert sich die Bewegungsgleichung zu

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{f}_{\text{int}}(\mathbf{u}) = \mathbf{f}_{\text{ext}} \quad (4.9)$$

Explizites Verfahren Ein explizites Verfahren zur Integration der Bewegungsgleichungen ist die zentrale Differenzenmethode. Hierzu wird eine Diskretisierung der Beschleunigung und der Geschwindigkeit in der folgenden Form durchgeführt

$$\ddot{\mathbf{u}}_i = \frac{1}{\Delta t^2} (\mathbf{u}_{i+1} - 2\mathbf{u}_i + \mathbf{u}_{i-1}) \quad (4.10)$$

$$\dot{\mathbf{u}}_i = \frac{1}{2\Delta t} (\mathbf{u}_{i+1} - \mathbf{u}_{i-1}) \quad (4.11)$$

Setzt man diese Diskretisierung in die Bewegungsgleichung (4.9) zum Zeitpunkt t_i ein, so ergibt sich

$$\mathbf{M} \frac{1}{\Delta t^2} (\mathbf{u}_{i+1} - 2\mathbf{u}_i + \mathbf{u}_{i-1}) + \mathbf{D} \frac{1}{2\Delta t} (\mathbf{u}_{i+1} - \mathbf{u}_{i-1}) + \mathbf{f}_{\text{int}}(\mathbf{u}_i) = \mathbf{f}_{\text{ext}} \quad (4.12)$$

Aufgelöst nach den unbekanntem Verschiebungen \mathbf{u}_{i+1} zum Zeitpunkt t_{i+1} folgt

$$\begin{aligned} \left(\mathbf{M} \frac{1}{\Delta t^2} + \mathbf{D} \frac{1}{2\Delta t} \right) \mathbf{u}_{i+1} &= \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}(\mathbf{u}_i) \\ &+ \mathbf{M} \frac{1}{\Delta t^2} (2\mathbf{u}_i - \mathbf{u}_{i-1}) + \mathbf{D} \frac{1}{2\Delta t} \mathbf{u}_{i-1} \end{aligned} \quad (4.13)$$

Diese Gleichung ist noch für alle drei Modellierungsansätze gültig und unterscheidet sich erst durch die Formulierung der internen Kraft \mathbf{f}_{int} gemäß einem Ansatz aus Kapitel 2. Für die nichtlineare FEM muss $\mathbf{f}_{\text{int}}(\mathbf{u}_i)$ nach (2.108) berechnet werden. Für die lineare FEM gilt nach (2.136)

$$\mathbf{f}_{\text{int}}(\mathbf{u}_i) = \mathbf{K}_{\text{lin}} \mathbf{u}_i \quad (4.14)$$

und für die korotierte FEM gemäß (2.146)

$$\mathbf{f}_{\text{int}}(\mathbf{u}_i) = \mathbf{K}_{\text{korot}}^i \mathbf{u}_i + \mathbf{f}_{\text{korot}} \quad (4.15)$$

Die Matrix auf der linken Seite von (4.13) wird als Systemmatrix \mathbf{A} geschrieben

$$\mathbf{A} = \mathbf{M} \frac{1}{\Delta t^2} + \mathbf{D} \frac{1}{2\Delta t} \quad (4.16)$$

Da sich die Massen- und Dämpfungsmatrix während der Simulation nicht ändern, ist auch die Matrix \mathbf{A} konstant und eine Berechnung einmalig zu Beginn der Simulation ausreichend.

Zusammengefasst ergibt sich der folgende Algorithmus für die Zeitintegration mit dem zentralen Differenzenverfahren

Algorithmus 4.5 : Zeitintegration mit zentralem Differenzenverfahren

Berechne $\mathbf{A} = \mathbf{M} \frac{1}{\Delta t^2} + \mathbf{D} \frac{1}{2\Delta t}$

foreach Simulationsschritt I **do**

for Zeitintegrationsschritt $i = 1$ **to** i_{end} **do**

$t_i = i\Delta t$

 Berechne $\mathbf{f}_{\text{int}} = \mathbf{f}_{\text{int}}(\mathbf{u}_i)$ gemäß gewähltem FEM-Ansatz

 Berechne $\mathbf{b} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}} + \mathbf{M} \frac{1}{\Delta t^2} (2\mathbf{u}_i - \mathbf{u}_{i-1}) + \mathbf{D} \frac{1}{2\Delta t} \mathbf{u}_{i-1}$

 Löse $\mathbf{A}\mathbf{u}_{i+1} = \mathbf{b}$

end

end

Implizites Verfahren Ein implizites Zeitintegrationsverfahren ist das Newmark-Verfahren. Es basiert auf der folgenden Diskretisierung der Beschleunigung und der Geschwindigkeit

$$\ddot{\mathbf{u}}_{i+1} = \frac{1}{\alpha\Delta t^2} (\mathbf{u}_{i+1} - \mathbf{u}_i) - \frac{1}{\alpha\Delta t} \dot{\mathbf{u}}_i - \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \quad (4.17)$$

$$\dot{\mathbf{u}}_{i+1} = \frac{\delta}{\alpha\Delta t} (\mathbf{u}_{i+1} - \mathbf{u}_i) + \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i + \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \quad (4.18)$$

mit den beiden Parametern $\alpha \in [0; \frac{1}{2}]$ und $\delta \in [0; 1]$. Für $\alpha = \frac{1}{4}$ und $\delta = \frac{1}{2}$ erhält man die Trapezregel.

Setzt man die Diskretisierungen in die Bewegungsgleichung (4.9) zum Zeitpunkt t_{i+1} ein, so ergibt sich

$$\begin{aligned} & \mathbf{M} \left(\frac{1}{\alpha\Delta t^2} (\mathbf{u}_{i+1} - \mathbf{u}_i) - \frac{1}{\alpha\Delta t} \dot{\mathbf{u}}_i - \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \right) \\ & + \mathbf{D} \left(\frac{\delta}{\alpha\Delta t} (\mathbf{u}_{i+1} - \mathbf{u}_i) + \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i + \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \right) \\ & + \mathbf{f}_{\text{int}}(\mathbf{u}_{i+1}) - \mathbf{f}_{\text{ext}} =: \mathbf{G}(\mathbf{u}_{i+1}) = \mathbf{0} \end{aligned} \quad (4.19)$$

Die diskretisierte Gleichung wird mit \mathbf{G} abgekürzt und es muss gelten $\mathbf{G} = \mathbf{0}$. Die Berechnung der gesuchten Verformungen \mathbf{u}_{i+1} zum Zeitpunkt t_{i+1} unterscheidet sich für die vorgestellten Modellierungsansätze.

Bei der nichtlinearen FEM wird zur Lösung der nichtlinearen Gleichung (4.19) für \mathbf{u}_{i+1} das Newton-Verfahren eingesetzt und es ergibt sich die Iterationsvorschrift mit dem Iterationsindex k

$$\left(\mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{tang}}^k \right) \Delta \mathbf{u}^k = -\mathbf{G}(\mathbf{u}_{i+1}^k) \quad (4.20)$$

$$\mathbf{u}_{i+1}^{k+1} = \mathbf{u}_{i+1}^k + \Delta \mathbf{u}^k \quad (4.21)$$

Die Summe der Matrizen auf der linken Seite der Gleichung (4.20) ist hier die Systemmatrix \mathbf{A}

$$\mathbf{A} = \mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{tang}}^k \quad (4.22)$$

Mit diesen gewählten Ansätzen besteht der Simulationsablauf aus drei ineinander geschachtelten Schleifen. Zusammengefasst lautet der Rechenalgorithmus für die Zei-

tintegration des nichtlinearen Ansatzes mit dem Newmark-Verfahren

Algorithmus 4.6 : Zeitintegration mit Newmark-Verfahren des nichtlinearen Ansatzes

```

foreach Simulationsschritt  $I$  do
  for Zeitintegrationsschritt  $i = 1$  to  $i_{\text{end}}$  do
     $t_i = i\Delta t$ 
    for Newton-Iteration  $k = 1$  to  $k_{\text{max}}$  do
      Berechne  $\mathbf{A} = \mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{tang}}^k(\mathbf{u}_{i+1}^k)$ 
      Berechne  $\mathbf{G}(\mathbf{u}_{i+1}^k)$ 
      Löse  $\mathbf{A}\Delta\mathbf{u}^k = -\mathbf{G}(\mathbf{u}_{i+1}^k)$ 
       $\mathbf{u}_{i+1}^{k+1} = \mathbf{u}_{i+1}^k + \Delta\mathbf{u}^k$ 
    end
    Berechne  $\dot{\mathbf{u}}_{i+1}$ 
    Berechne  $\ddot{\mathbf{u}}_{i+1}$ 
  end
end

```

Beim Ansatz mit der linearen FEM kann (4.19) nach \mathbf{u}_{i+1} aufgelöst werden und man erhält das lineare Gleichungssystem

$$\begin{aligned}
& \left(\mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{lin}} \right) \mathbf{u}_{i+1} = \mathbf{f}_{\text{ext}} \\
& + \mathbf{M} \left(\frac{1}{\alpha\Delta t^2} \mathbf{u}_i + \frac{1}{\alpha\Delta t} \dot{\mathbf{u}}_i + \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \right) \\
& + \mathbf{D} \left(\frac{\delta}{\alpha\Delta t} \mathbf{u}_i - \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i - \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \right)
\end{aligned} \tag{4.23}$$

Die Summe der Matrizen auf der linken Seite \mathbf{A} ist während der Simulation konstant, da die einzelnen Summanden konstant sind. Die Matrix muss daher nur einmal zu Beginn der Simulation berechnet werden.

$$\mathbf{A} = \mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{lin}} \tag{4.24}$$

Die Berechnungsvorschrift für die Zeitintegration des linearen Ansatzes mit dem Newmark-Verfahren ist im Algorithmus 4.7 dargestellt.

Beim Ansatz mit der korotierten FEM kann (4.19) nach \mathbf{u}_{i+1} aufgelöst werden. Es ergibt sich das lineare Gleichungssystem

$$\begin{aligned}
& \left(\mathbf{M} \frac{1}{\alpha\Delta t^2} + \mathbf{D} \frac{\delta}{\alpha\Delta t} + \mathbf{K}_{\text{korot}}(\mathbf{u}_i) \right) \mathbf{u}_{i+1} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{korot}} \\
& + \mathbf{M} \left(\frac{1}{\alpha\Delta t^2} \mathbf{u}_i + \frac{1}{\alpha\Delta t} \dot{\mathbf{u}}_i + \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \right) \\
& + \mathbf{D} \left(\frac{\delta}{\alpha\Delta t} \mathbf{u}_i - \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i - \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \right)
\end{aligned} \tag{4.25}$$

Algorithmus 4.7 : Zeitintegration mit Newmark-Verfahren des linearen Ansatzes

 Berechne $\mathbf{A} = \mathbf{M} \frac{1}{\alpha \Delta t^2} + \mathbf{D} \frac{\delta}{\alpha \Delta t} + \mathbf{K}_{\text{lin}}$
foreach Simulationsschritt I **do**
for Zeitintegrationsschritt $i = 1$ **to** i_{end} **do**
 $t_i = i \Delta t$

 Berechne $\mathbf{b} = \mathbf{f}_{\text{ext}} + \mathbf{M} \left(\frac{1}{\alpha \Delta t^2} \mathbf{u}_i + \frac{1}{\alpha \Delta t} \dot{\mathbf{u}}_i + \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \right) +$
 $\mathbf{D} \left(\frac{\delta}{\alpha \Delta t} \mathbf{u}_i - \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i - \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \right)$

 Löse $\mathbf{A} \mathbf{u}_{i+1} = \mathbf{b}$

 Berechne $\dot{\mathbf{u}}_{i+1}$

 Berechne $\ddot{\mathbf{u}}_{i+1}$
end
end

Die Systemmatrix \mathbf{A} als Summe der linken Seite von Gleichung (4.25) lautet

$$\mathbf{A} = \mathbf{M} \frac{1}{\alpha \Delta t^2} + \mathbf{D} \frac{\delta}{\alpha \Delta t} + \mathbf{K}_{\text{korot}}(\mathbf{u}_i) \quad (4.26)$$

Da die korotierte Steifigkeitsmatrix $\mathbf{K}_{\text{korot}}$ von der Verformung abhängt, ändert sich die Summe der Matrizen auf der linken Seite in jedem Zeitschritt. Der Algorithmus lautet zusammengefasst für die Zeitintegration des korotierten Ansatzes mit dem Newmark-Verfahren

Algorithmus 4.8 : Zeitintegration mit Newmark-Verfahren des korotierten Ansatzes

foreach Simulationsschritt I **do**
for Zeitintegrationsschritt $i = 1$ **to** i_{end} **do**
 $t_i = i \Delta t$

 Berechne $\mathbf{A} = \mathbf{M} \frac{1}{\alpha \Delta t^2} + \mathbf{D} \frac{\delta}{\alpha \Delta t} + \mathbf{K}_{\text{korot}}(\mathbf{u}_i)$

 Berechne $\mathbf{b} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{korot}} + \mathbf{M} \left(\frac{1}{\alpha \Delta t^2} \mathbf{u}_i + \frac{1}{\alpha \Delta t} \dot{\mathbf{u}}_i + \left(\frac{1}{2\alpha} - 1 \right) \ddot{\mathbf{u}}_i \right) +$
 $\mathbf{D} \left(\frac{\delta}{\alpha \Delta t} \mathbf{u}_i - \left(1 - \frac{\delta}{\alpha} \right) \dot{\mathbf{u}}_i - \left(1 - \frac{\delta}{2\alpha} \right) \Delta t \ddot{\mathbf{u}}_i \right)$

 Löse $\mathbf{A} \mathbf{u}_{i+1} = \mathbf{b}$

 Berechne $\dot{\mathbf{u}}_{i+1}$

 Berechne $\ddot{\mathbf{u}}_{i+1}$
end
end

Vergleich der Verfahren Für den Vergleich der beiden Integrationsverfahren wird das Beispiel des eingespannten Balkens verwendet. Der Balken ist zur Zeit $t = 0$ unverformt und in Ruhe. Er wird losgelassen und die Verformung konvergiert gegen die statische Verformung. Für die Darstellung des Zeitverlaufs wird die vertikale Komponente des ausgewiesenen Knotens P verwendet, der im Bild 4.2 eingezeichnet ist.

Das Newmark-Verfahren ist für alle Zeitschrittweiten stabil, während dagegen die

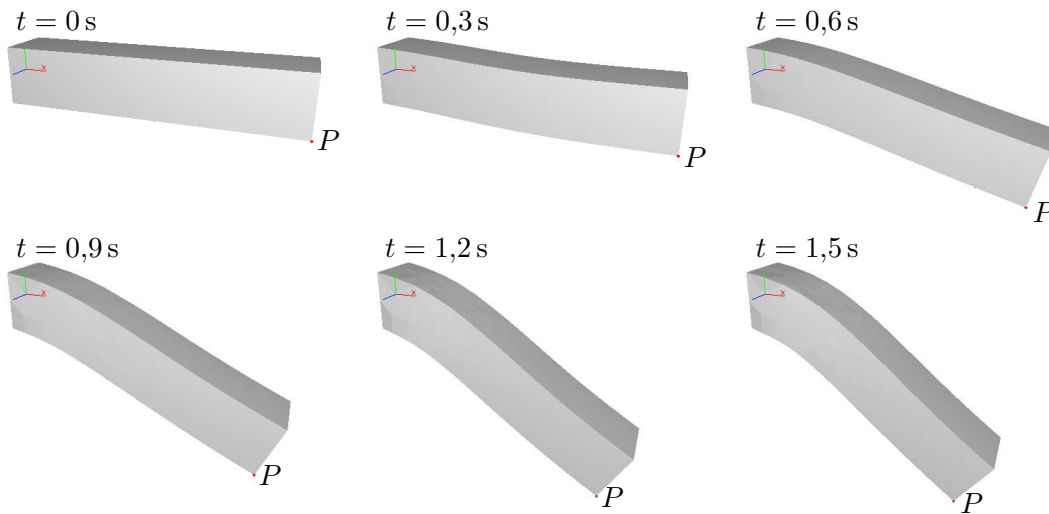


Bild 4.2: Dynamische Verformung des Balkens (Momentaufnahmen)

Stabilität des zentralen Differenzenverfahrens von mehreren Größen abhängt, wie beispielsweise von der Diskretisierung des Volumens, von den Materialparametern oder von den Parametern der Zeitintegration. Die Zeitschrittweite Δt muss dabei stets kleiner oder gleich einer kritischen Zeitschrittweite Δt_{kr} sein

$$\Delta t \leq \Delta t_{\text{kr}} \quad (4.27)$$

Die kritische Zeitschrittweite ist die kleinste Elementlänge L_e , in diesem Fall der kleinste Abstand zwischen zwei Knoten des Netzes, geteilt durch die Wellengeschwindigkeit c

$$\Delta t_{\text{kr}} = \frac{L_e}{c} \quad (4.28)$$

Die Wellengeschwindigkeit in einem St. Venant-Kirchhoff-Material lautet [115]

$$c = \sqrt{\frac{E(1 - \nu)}{\rho(1 + \nu)(1 - 2\nu)}} \quad (4.29)$$

Für das gewählte Beispielmodell ist jeweils eine Zeitschrittweite kleiner 10^{-3} notwendig um das System stabil zu integrieren, unabhängig von der Modellierungsart. In den Bildern 4.3, 4.4 und 4.5 sind die Zeitverläufe mit den beiden Integrationsverfahren dargestellt. In den vergrößerten Bereichen kann man keinen großen Unterschied im zeitlichen Verlauf erkennen. Der Zeitverlauf im Bild 4.4 bei der Modellierung mit der linearen FEM schwingt stark um die statische Lösung. Wie schon im statischen Fall weicht auch hier die lineare Lösung stark von der nichtlinearen ab, wie ein Vergleich mit Bild 4.3 zeigt. Im Bild 4.6 sind die Ergebnisse mit dem Newmark-Verfahren der nichtlinearen und der korotierten FEM zusammen aufgetragen. Es ist eine qualitativ gute Übereinstimmung der beiden Zeitverläufe erkennbar, wie sie bereits in Kapitel 2 Bild 2.8 bei der statischen Rechnung festgestellt wurde.

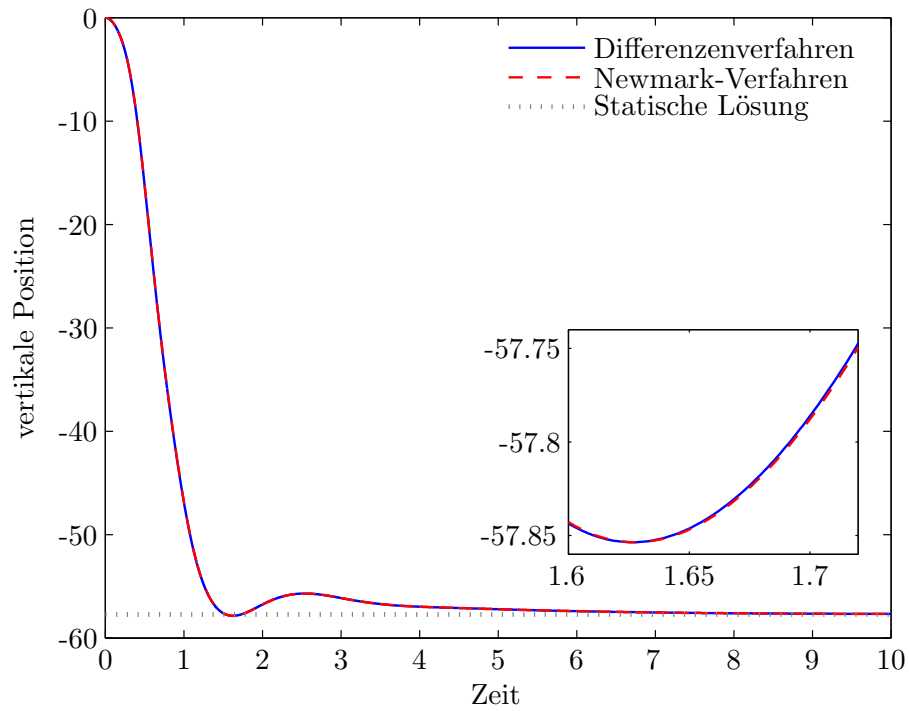


Bild 4.3: Vergleich Dynamik nichtlineare FEM (explizit/implizit)

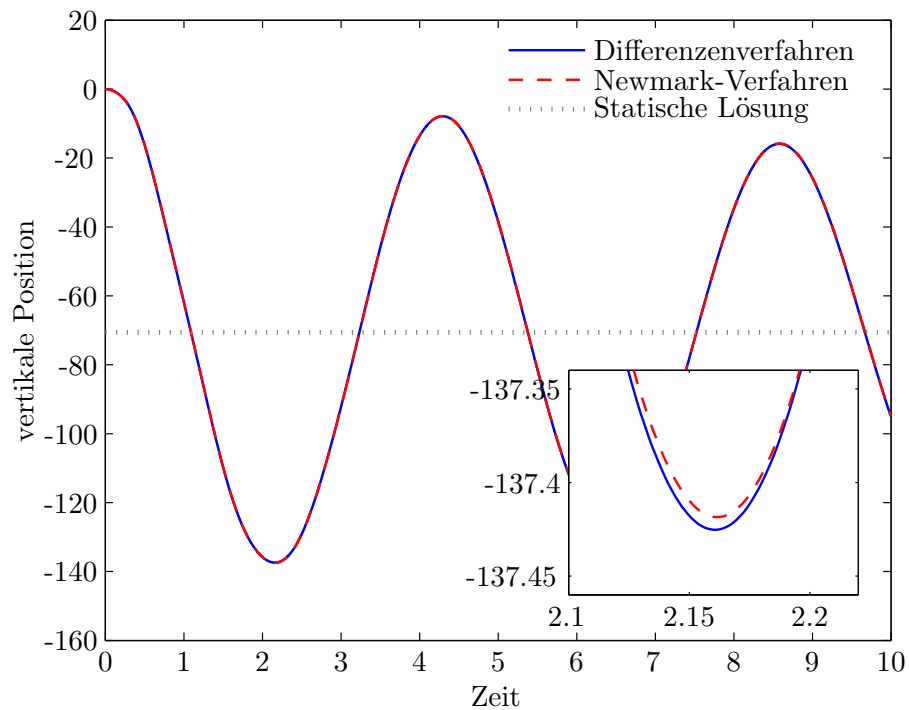


Bild 4.4: Vergleich Dynamik lineare FEM (explizit/implizit)

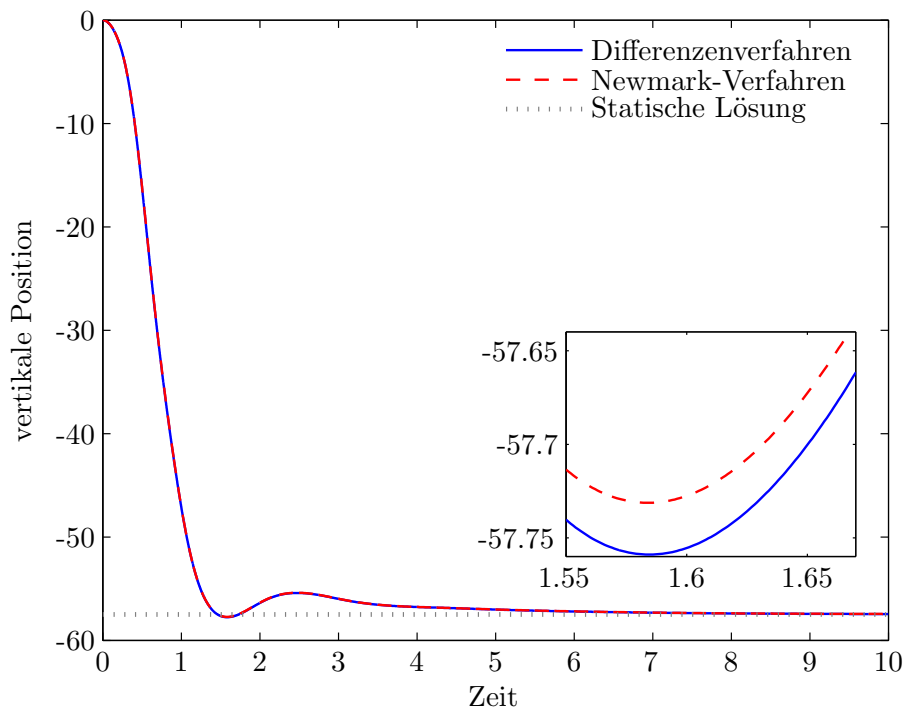


Bild 4.5: Vergleich Dynamik korotiert (explizit/implizit)

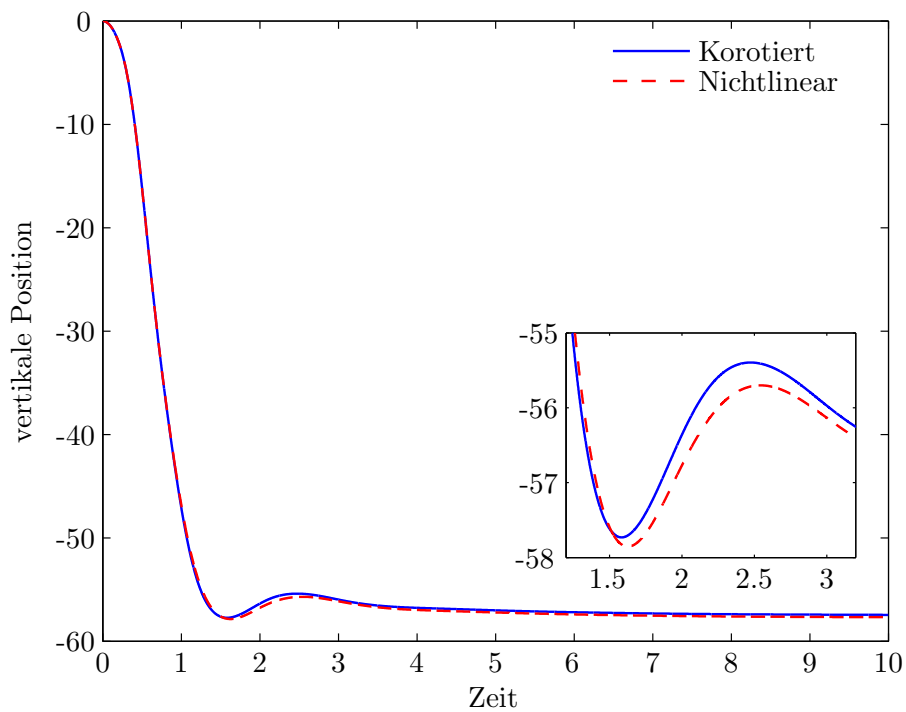


Bild 4.6: Vergleich des Zeitverlaufs bei nichtlinearer und korotierter FEM

4.2 Interaktionsarten

Das dritte Klassifizierungskriterium für die Simulation ist die Interaktionsart. Sie definiert die Schnittstelle zum Datenaustausch mit der Simulation. Es gibt zwei Arten um mit der Simulation zu kommunizieren, die im Bild 4.7 skizziert sind. Zum einen kann die externe Kraft an gegebenen Knoten als Simulationseingang genutzt werden. Der Ausgang ist in diesem Fall die Verformung des Körpers. Zum anderen kann die Verformung an gegebenen Knoten als Eingang dienen. Die Reaktionskräfte auf Grund der Verformung stellen dann den Ausgang der Simulation dar.

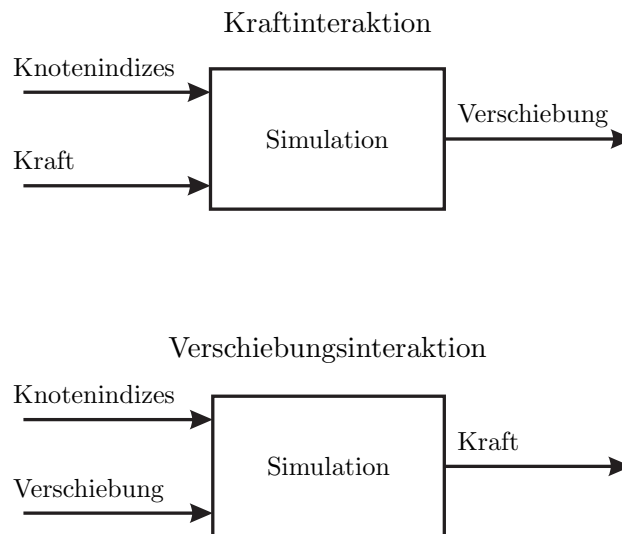


Bild 4.7: Interaktionsarten

Um die beiden Interaktionarten in die Simulation zu integrieren, werden zunächst die Systemgleichungen genauer betrachtet. Alle bisher untersuchten Modellierungsansätze und Simulationsarten resultieren in derselben Gleichungsstruktur. Es muss jeweils ein lineares Gleichungssystem der Form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (4.30)$$

gelöst werden. Der Lösungsvektor \mathbf{x} ist entweder der Verschiebungsvektor \mathbf{u} oder dessen Änderung $\Delta\mathbf{u}$. Die Systemmatrix \mathbf{A} und der Vektor \mathbf{b} werden nach Wahl des FEM-Ansatzes und der Simulationsart, wie in den vorherigen Abschnitten dargestellt, berechnet.

Die Interaktion erfolgt über die Vorgabe von Randbedingungen. Im Folgenden wird vorausgesetzt, dass eine endliche Zahl der Knoten eine Dirichlet-Randbedingung erfüllen, zum Beispiel durch eine feste Einspannung, und damit das Gesamtproblem immer statisch bestimmt ist.

Zunächst werden die Auswirkungen durch eine Vorgabe von Randbedingungen auf das Gleichungssystem (4.30) untersucht. Durch die Vorgabe einer Dirichlet-Randbedingung in Form einer Knotenverschiebung sind Komponenten des Vektors \mathbf{x} gegeben und die entsprechenden Komponenten auf der rechten Seite \mathbf{b} sind gesucht.

Zur Lösung dieses Problems wird das Gleichungssystem partitioniert

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \quad (4.31)$$

Alle gegebenen Komponenten von \mathbf{x} werden in dem Vektor \mathbf{x}_2 zusammengefasst. Die entsprechenden Komponenten \mathbf{b}_2 auf der rechten Seite sind somit unbekannt. Die unbekanntenen Komponenten der freien Knoten sind im Vektor \mathbf{x}_1 zusammengefasst, mit der entsprechenden rechten Seite \mathbf{b}_1 , deren Komponenten wieder alle gegeben sind. Die Lösung des Gleichungssystems erfolgt in zwei Schritten

$$\mathbf{A}_{11}\mathbf{x}_1 = \mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2 \quad (4.32)$$

$$\mathbf{b}_2 = \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 \quad (4.33)$$

Die Partitionierung des Gleichungssystems wird somit durch die Interaktionsart und natürlich durch die betroffenen Knotenindizes bestimmt. Für den Fall, dass sich die Interaktionsart oder die Knotenindizes ändern, muss die Partitionierung neu durchgeführt werden.

Kraftinteraktion Bei der Kraftinteraktion (Bild 4.7 oben) werden in jedem Simulationsschritt Kräfte an Knoten vorgegeben. Diese Kraftvorgaben können nur an freien Knoten erfolgen und gehen daher in Komponenten des Vektors \mathbf{b}_1 ein. Es ändert sich dadurch die rechte Seite des Gleichungssystems. Die Partitionierung des Gleichungssystems bleibt gleich, auch wenn die Kraftvorgaben in jedem Simulationsschritt an unterschiedlichen freien Knoten erfolgen.

Verschiebungsinteraktion Bei der Verschiebungsinteraktion (Bild 4.7 unten) werden in jedem Simulationsschritt Verschiebungen an Knoten vorgegeben. Diese Komponenten sind ein Teil des Vektors \mathbf{x}_2 . Neben dem Betrag dieser Verschiebungsvorgabe kann sich auch die Menge der Knoten mit Verschiebungsvorgabe ändern, wenn beispielsweise an einem zunächst freien Knoten eine Verschiebung vorgegeben wird. Das hat zur Folge, dass die Partitionierung des Gleichungssystems erneut durchgeführt werden muss.

4.3 Numerische Lösungsverfahren

Die Simulation von deformierbaren Körpern mit den vorgestellten Algorithmen erfordert die Lösung eines großen linearen Gleichungssystems (4.30). In den vorherigen Abschnitten wurde gezeigt, dass sich die Systemmatrix \mathbf{A} während der Simulation ändern kann. So kann es sein, dass sich die Matrix nie, in jedem Simulationsschritt, in jedem Zeitschritt oder in jedem Iterationsschritt ändert. Das mögliche Änderungsverhalten aufgrund des gewählten Modellierungsansatzes, der Simulationsart und des Integrationsverfahrens ist in der Tabelle 4.1 zusammengefasst.

Tabelle 4.1: Änderung der Systemmatrix

Systemmatrix \mathbf{A} ändert sich	FEM-Ansatz	Simulationsart	Zeitintegration	Algorithmus
nie	linear	statisch	-	4.3
	nichtlinear	dynamisch	explizit	4.5
	linear	dynamisch	explizit	4.5
	korotiert	dynamisch	explizit	4.5
	linear	dynamisch	implizit	4.7
in jedem Simulationsschritt	korotiert	statisch	-	4.4
in jedem Zeitschritt	korotiert	dynamisch	implizit	4.8
in jedem Iterationsschritt	nichtlinear	statisch	-	4.1
	nichtlinear	dynamisch	implizit	4.6

Zur Lösung des linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ können direkte und indirekte Verfahren eingesetzt werden. Die Wahl des Verfahrens hängt von der Größe, der Struktur und auch vom Änderungsverhalten der Matrix ab.

Die Matrix \mathbf{A} ist symmetrisch, positiv definit und dünnbesetzt. Die Belegung der linearen Steifigkeitsmatrix des Balkenbeispiels ist im Bild 4.8 zu sehen und hat eine Belegungsichte von 4%. Zur Speicherung von dünnbesetzten Matrizen kann man verschiedene Strukturen verwenden. Ein einfaches Format ist das *coordinate*-Format. Es speichert die Zeile, die Spalte und den Wert des entsprechenden Matrixeintrages ab. In den Implementierungen für diese Arbeit wurde das *compressed sparse row*-Format verwendet. Es besitzt einen geringeren Speicherbedarf als das *coordinate*-Format, da für jede Matrixzeile nur ein Vektor mit Spaltenindizes gespeichert wird.

Direkte Verfahren Die direkten Verfahren basieren auf einer Cholesky-Zerlegung der symmetrisch, positiv definiten Matrix \mathbf{A}

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T \quad (4.34)$$

mit einer unteren Dreiecksmatrix \mathbf{L} . Liegt die Cholesky-Zerlegung vor, kann durch ein Vorwärts- und Rückwärtseinsetzen der Lösungsvektor \mathbf{x} berechnet werden.

$$\mathbf{L}\mathbf{z} = \mathbf{b} \quad (4.35)$$

$$\mathbf{L}^T \mathbf{x} = \mathbf{z} \quad (4.36)$$

Der größte Rechenaufwand bei diesem Verfahren liegt in der Zerlegung der Matrix. Insbesondere bei großen, dünnbesetzten Matrizen müssen spezielle Algorithmen eingesetzt werden. Es existieren hierfür eine Reihe von Softwarebibliotheken, wie LAPACK [1], [5] für vollbesetzte Matrizen oder CHOLMOD [33], [24] für dünnbesetzte Matrizen.

Die direkten Verfahren bieten sich insbesondere an, wenn sich die Systemmatrix \mathbf{A}

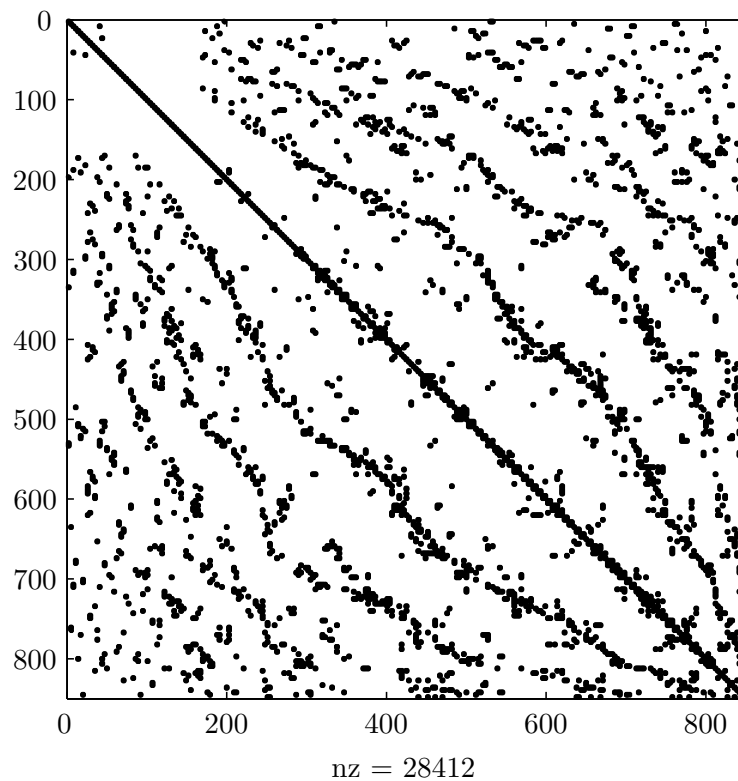


Bild 4.8: Belegung der Steifigkeitsmatrix des Balkenbeispiels

während der Simulation nicht ändert. Die Cholesky-Zerlegung kann in diesem Fall zu Beginn der Simulation berechnet werden. Zur Lösung des Gleichungssystems muss dann lediglich ein Vorwärts- und Rückwärtseinsetzen durchgeführt werden.

Wenn sich die Partitionierung der Matrix \mathbf{A} auf Grund einer Änderung der Indexmenge bei der Verschiebungsinteraktion ändert, kann dies durch entsprechendes Setzen der Spalten und Zeilen in der Dreiecksmatrix \mathbf{L} der Cholesky-Zerlegung berücksichtigt werden. Das bedeutet, dass man anstelle von (4.32)

$$\mathbf{A}_{11}\mathbf{x}_1 = \mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2 \quad (4.37)$$

das erweiterte Gleichungssystem

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 - \mathbf{A}_{12}\mathbf{x}_2 \\ \mathbf{x}_2 \end{bmatrix} \quad (4.38)$$

löst. Da die Cholesky-Zerlegung der Matrix \mathbf{A} bereits berechnet wurde, bestimmt sich die Cholesky-Zerlegung des erweiterten Systems zu

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T \quad (4.39)$$

Somit muss bei einer Änderung der Partitionierung der Matrix durch veränderte Randbedingungen keine erneute Cholesky-Zerlegung durchgeführt werden.

Iterative Verfahren Es gibt eine Reihe von iterativen Verfahren zur Lösung von linearen Gleichungssystemen [55]. Das Ziel ist durch Rechenoperationen eine Folge von Lösungsvektoren zu erzeugen, die gegen die exakte Lösung konvergiert. Für symmetrisch, positiv definite Matrizen eignet sich das *konjugierte Gradientenverfahren* (CG-Verfahren). Dabei müssen in jedem Iterationsschritt eine Matrix-Vektor-Multiplikation und mehrere Skalarprodukte berechnet werden. Die Iterationsschleife wird abgebrochen, wenn das Residuum unterhalb einer gesetzten Toleranz ist oder wenn eine maximale Anzahl von Iterationsschritten erreicht ist.

Das CG-Verfahren kann beschleunigt werden, indem eine Vorkonditionierung verwendet wird. Als einfache und auch schnelle Vorkonditionierer bieten sich eine inverse Diagonalmatrix und eine unvollständige Cholesky-Zerlegung an. In dieser Arbeit wird eine unvollständige Cholesky-Zerlegung verwendet, die die Dünnbesetztheit der Matrix erhält und keine sogenannten Fill-in erzeugt.

4.4 Rechenzeitvergleich

Ein Vergleich von Rechenzeiten unterschiedlicher Algorithmen ist im Allgemeinen schwierig, da er von vielen Faktoren abhängig ist. Neben vergleichbaren Genauigkeitsvorgaben, spielt die Implementierung und die Optimierung durch den Compiler eine große Rolle. Im Folgenden werden zwei verschiedene Rechenzeitvergleiche präsentiert. Der erste ist die benötigte Zeit zur Lösung des dünnbesetzten Gleichungssystems. Dabei wird das direkte mit dem indirekten Verfahren für unterschiedliche Matrixgrößen verglichen. Beim zweiten Vergleich wird die Rechenzeit für einen Zeitschritt des Newmark-Verfahrens untersucht. Dieser Vergleich dient als Ausgangspunkt für eine spätere Echtzeitsimulation. Wenn die Zeitschrittweite des Integrationsverfahrens gleich der benötigten Rechenzeit ist, dann stimmt die Simulationszeit mit der realen Zeit überein. Als Beispiel dient wieder der einseitig eingespannte Balken, der mit unterschiedlich feinen Tetraedernetzen diskretisiert wird. Mit der Diskretisierung ändert sich die Systemgröße und die Matrixdimension und es können die Auswirkungen auf die Rechenzeit analysiert werden.

Lösung des linearen Gleichungssystems Zum Vergleich des direkten mit dem indirekten Verfahren zur Lösung des großen dünnbesetzten Gleichungssystems wird ein Cholesky-Verfahren und ein CG-Verfahren verwendet. Das Cholesky-Verfahren ist implementiert in dem Paket CHOLMOD. Das CG-Verfahren ist selbstimplementiert mit einer eigenen Bibliothek zur schnellen Berechnung der Matrix-Vektor-Operationen. Sämtliche Rechnungen wurden mit einem PC mit Intel P4 Prozessor mit 3.2 GHz Taktrate und 1 GB Hauptspeicher durchgeführt.

In Tabelle 4.2 und 4.3 sind die Rechenzeiten für eine statische Lösung des Balkens mit linearer FEM eingetragen. Das Gleichungssystem wird gemäß Algorithmus 4.3 genau einmal gelöst. Für das CG-Verfahren wurden unterschiedliche Abbruchtoleranzen getestet und als Startvektor wurde der Nullvektor eingesetzt. Beim direkten

Verfahren umfasst die Rechenzeit die Cholesky-Zerlegung und das Vorwärts- und Rückwärtseinsetzen. Die Zeit zum Aufstellen der Systemmatrix wird in beiden Verfahren nicht berücksichtigt, da dies identisch erfolgt.

Im Bild 4.9 wird der Einfluss eines Vorkonditionierers auf die Gesamtrechenzeit des CG-Verfahrens für die Abbruchtoleranz von 10^{-8} dargestellt und mit der Rechenzeit des direkten Verfahrens verglichen.

Tabelle 4.2: Rechenzeit des CG-Verfahrens in ms

Matrixdimension	CG (ohne Vorkond.)			CG (Diag.)			CG (Unvollst. Chol.)		
	10^{-4}	10^{-6}	10^{-8}	10^{-4}	10^{-6}	10^{-8}	10^{-4}	10^{-6}	10^{-8}
261	5	5	6	3	3	4	4	4	4
489	18	18	21	10	11	11	11	12	12
849	37	38	42	22	23	24	24	25	26
1317	69	71	82	44	46	47	47	49	51
1920	131	136	148	87	89	93	92	96	99
2835	255	266	298	167	173	186	164	170	177
5097	615	639	677	432	446	460	392	409	427
9597	1484	1543	1648	1068	1118	1150	970	1014	1049

Tabelle 4.3: Rechenzeit des Cholesky-Verfahrens in ms

Matrixdimension	Einträge	Cholesky-Zerlegung	Gesamtrechenzeit
261	7533	2	2
489	15021	3	4
849	28467	9	10
1317	46629	18	20
1920	71190	34	36
2835	106515	59	62
5097	201123	162	170
9597	392391	511	535

Die Rechenzeit für die Lösung mit dem Cholesky-Verfahren kann noch genauer analysiert werden (siehe Tabelle 4.3). Es zeigt sich, dass den Großteil der Rechenzeit die Cholesky-Zerlegung benötigt und das Vorwärts- und Rückwärtseinsetzen demgegenüber nur kurze Zeit beansprucht.

Zusammenfassend kann man erkennen, dass bei den untersuchten Systemdimensionen das direkte Verfahren stets schneller ist als das indirekte. Ein weiterer Vorteil des Cholesky-Verfahrens ist die Vorabberechnung der Zerlegung. Wenn sich die Systemmatrix während der Simulation nicht ändert, kann die Cholesky-Zerlegung zu Beginn durchgeführt werden und zur Lösung ist lediglich ein schnelles Vorwärts- und Rückwärtseinsetzen erforderlich. Zusätzlich ist die benötigte Rechenzeit für ein System immer identisch und somit planbar, was für eine echtzeitfähige Simulation wichtig ist. Die Vorteile des CG-Verfahrens sind der Zugriff auf Näherungslösungen zwischen den Iterationen, die Nutzung von Startlösungen aus vorherigen Simulationsschritten und die Einstellbarkeit der Abbruchtoleranz. Da für die späteren Anwendungen die Rechenzeit ausschlaggebend ist, erweist sich das direkte als das besser geeignete Verfahren.

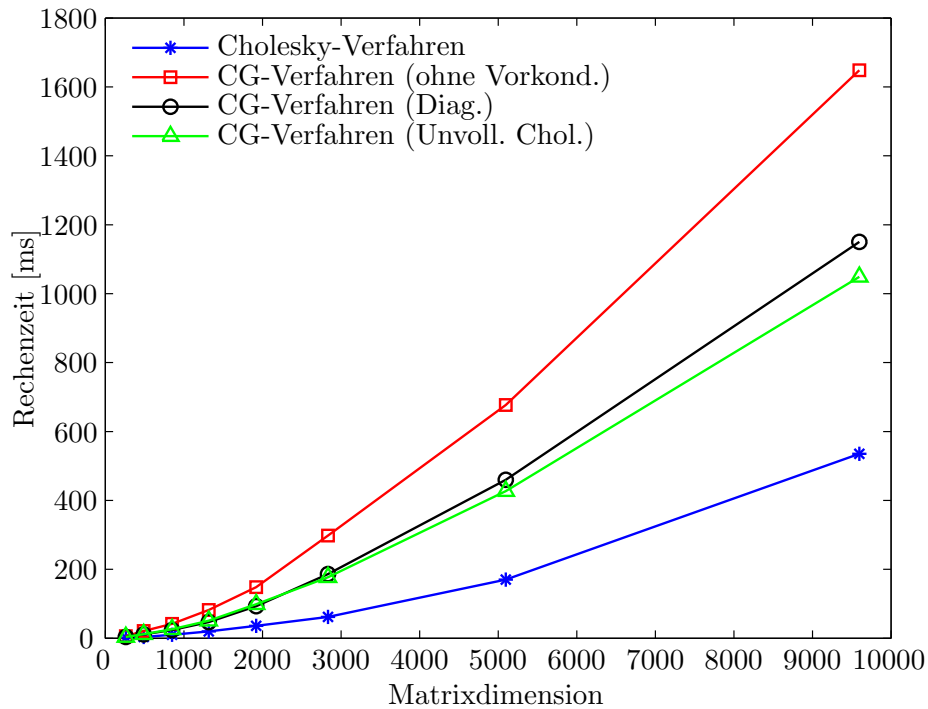


Bild 4.9: Vergleich des Rechenzeitbedarfs

Zeitschritt mit Newmark-Verfahren Der zweite Vergleich untersucht die Rechenzeit für einen Zeitschritt mit dem Newmark-Verfahren für die vorgestellten FEM-Modellierungsansätze (Algorithmus 4.6, 4.7 und 4.8). Innerhalb der Zeitintegration wird das Cholesky-Verfahren zur Lösung des Gleichungssystems eingesetzt. Beim nichtlinearen Ansatz wird die Anzahl der Newton-Iterationen variiert und deren Auswirkung auf die Rechenzeit und Genauigkeit untersucht.

In Tabelle 4.4 sind die ermittelten Rechenzeiten in Millisekunden eingetragen. Der lineare Ansatz ist erwartungsgemäß hier am schnellsten. Der Grund liegt in der Vorberechnung der Cholesky-Zerlegung (Algorithmus 4.7), womit in jedem Zeitschritt nur ein schnelles Vorwärts- und Rückwärtseinsetzen durchgeführt werden muss. Mit dem korotierten Ansatz (Algorithmus 4.8) muss in jedem Zeitschritt das Gleichungssystem neu aufgebaut und anschließend gelöst werden. Die Rechenzeit für einen Zeitschritt ist daher größer als bei der alleinigen Lösung des linearen Gleichungssystems (Tabelle 4.3). Die Rechenzeit für den nichtlinearen Ansatz (Algorithmus 4.6) skaliert mit der Anzahl der durchgeführten Newton-Iterationen. Bei einer Newton-Iteration muss ein Gleichungssystem gelöst werden und der Aufwand entspricht daher in etwa dem des korotierten Ansatzes. Bei mehr Newton-Iterationen steigt die Rechenzeit entsprechend. So wird bei fünf Iterationen ungefähr die fünffache Rechenzeit benötigt.

Des Weiteren stellt sich die Frage, wie viele Newton-Iterationen durchzuführen sind. Im Bild 4.10 wird der Einfluss der durchgeführten Newton-Iterationen auf die Genauigkeit der Lösung dargestellt. Dazu wurde der zeitliche Verlauf der Knotenverschiebungen $\mathbf{u}(t)$ mit fünf Newton-Iterationen $\mathbf{u}^{k=5}(t)$ als Referenzlösung angenommen.

Fünf Iterationen sind für das betrachtete Beispiel ausreichend, da eine größere Anzahl keine Verbesserung der Genauigkeit mit sich bringt. In jedem Zeitschritt wird die Differenz der Verschiebungsverläufe $\mathbf{u}^{k=1}(t)$, $\mathbf{u}^{k=2}(t)$, $\mathbf{u}^{k=3}(t)$ und $\mathbf{u}^{k=4}(t)$ von der Referenzlösung gebildet. Mit der Norm dieser Abweichung zu jedem Zeitpunkt t_i erhält man ein Maß W für die Übereinstimmung der Verformung und somit für die Genauigkeit. Als Berechnungsvorschrift ergibt sich beispielsweise für eine Iteration

$$W(t_i) = \|\mathbf{u}^{k=1}(t_i) - \mathbf{u}^{k=5}(t_i)\| \quad (4.40)$$

Der Verlauf von W ist im Bild 4.10 für eine und im Bild 4.11 zwei Newton-Iterationen dargestellt. Mit einer Newton-Iteration kann man einen geringen Unterschied zur Referenzlösung erkennen, während man mit zwei Iterationen kaum noch einen Unterschied feststellen kann. Mit drei und vier Iterationen befindet man sich bereits unterhalb der Darstellungsauflösung.

Tabelle 4.4: Rechenzeit für einen Zeitschritt des Newmark-Verfahrens in ms

Matrixdimension	Linear	Korotiert	Nichtlinear (k Iterationen)				
			$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
261	0,1	4	4	8	12	16	20
489	0,2	9	9	18	27	35	44
849	0,8	21	21	43	64	86	107
1317	1,5	38	39	79	118	158	198
1920	2,4	66	67	135	203	269	337
2835	4,0	108	111	223	335	446	557
5097	8,6	262	269	540	808	1079	1349
9597	21,2	734	759	1490	2235	2976	3734

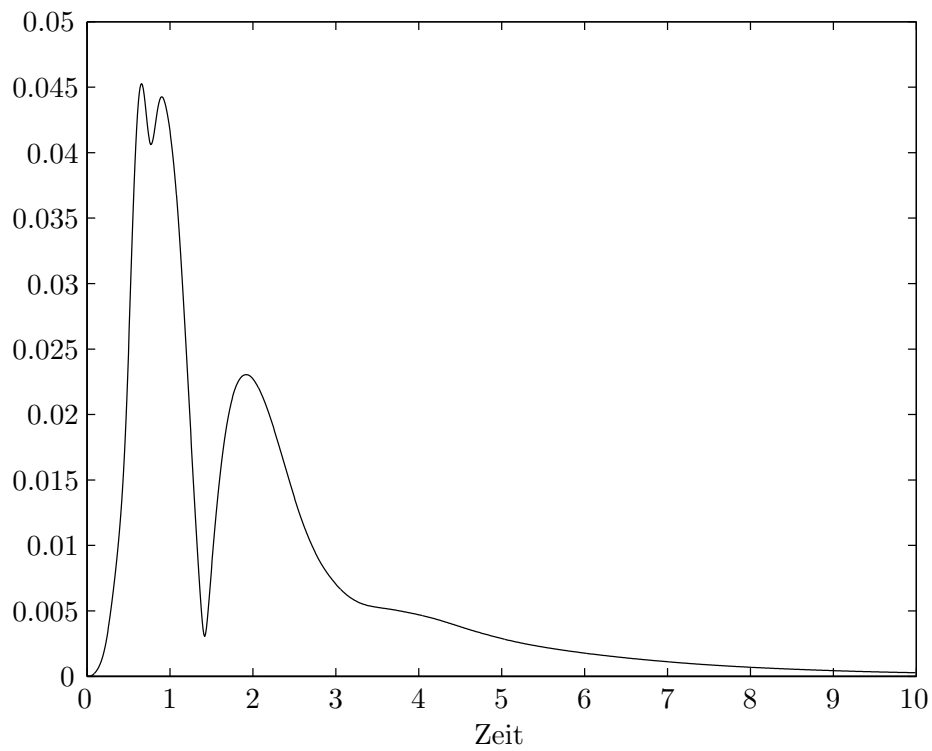


Bild 4.10: Maß der Verformungsabweichung W für eine Newton-Iteration

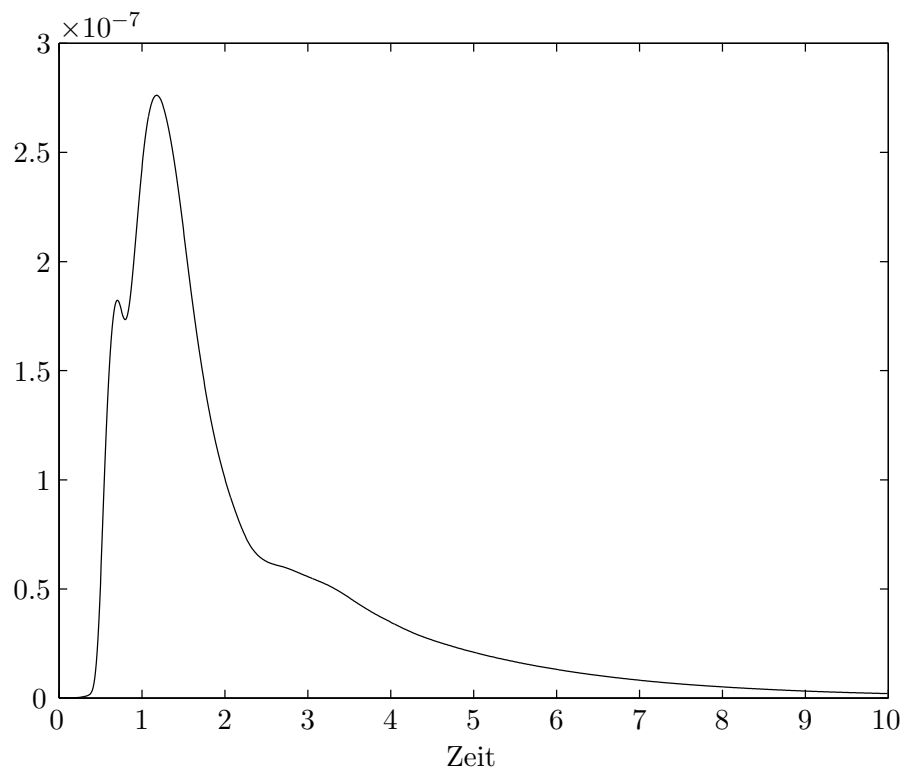


Bild 4.11: Maß der Verformungsabweichung W für zwei Newton-Iterationen

5 Haptisches Rendering

In den vorherigen Kapiteln wurde die Simulation von deformierbaren Körpern ausführlich untersucht. Eine Anwendung davon ist das haptische Rendering. Haptisches Rendering ist die interaktive Simulation einer virtuellen Umgebung mittels eines haptischen Geräts, welches eine Krafrückkopplung ermöglicht. Ein Einsatzgebiet ist die virtuelle Realität, um realistischere Szenarien zu schaffen, wie zum Beispiel Trainingssimulatoren für die Chirurgie. Die Anforderungen beim haptischen Rendering sind höher als beim visuellen Rendering, bei dem eine Bildrate von etwa 30 Hz ausreichend ist, um eine flüssige Animation zu erzeugen. Beim haptischen Rendering ist eine Taktrate bis 1000 Hz notwendig, um Krafteindrücke realistisch darzustellen. Die Berechnung der Kraft in der Simulation muss daher dementsprechend schnell erfolgen.

Im Folgenden wird die Simulation der deformierbaren Körper mit einem haptischen Gerät gekoppelt, wobei die Simulation für die Berechnung der Kräfte in Echtzeit erfolgen soll. Bei großen Simulationssystemen kann die Kraftberechnung viel Zeit beanspruchen, wodurch die Echtzeitanforderung verletzt werden kann. Daher wird ein Ansatz vorgestellt, der auch bei längeren Rechenzeiten dem menschlichen Benutzer noch eine realistische Kraftwahrnehmung eines deformierbaren Körpers ermöglicht.

Der Systemaufbau ist im Bild 5.1 dargestellt. Die Position des haptischen Geräts wird dabei an ein Interaktionsmodul gesendet. Dieses Modul ist dafür verantwortlich, die Indizes und Verschiebungen der Knoten für die Interaktion mit dem deformierbaren Körper zu ermitteln. Im einfachsten Fall entspricht die Position des haptischen Geräts genau der Verschiebung eines einzelnen Knotens des Netzes. Bei komplexeren Szenarien kann das Interaktionsmodul eine Kollisionserkennung oder auch eine Mehrkörpersimulation beinhalten. Auf der anderen Seite hat das Modul die Aufgabe, aus den Reaktionskräften an den ermittelten Knoten genau eine dreidimensionale Kraft abzubilden, die an das haptische Gerät gesendet wird. Das Interaktionsmodul dient außerdem als Verbindung zwischen den verschiedenen Taktraten, mit denen das haptische Gerät und die Simulation arbeiten.

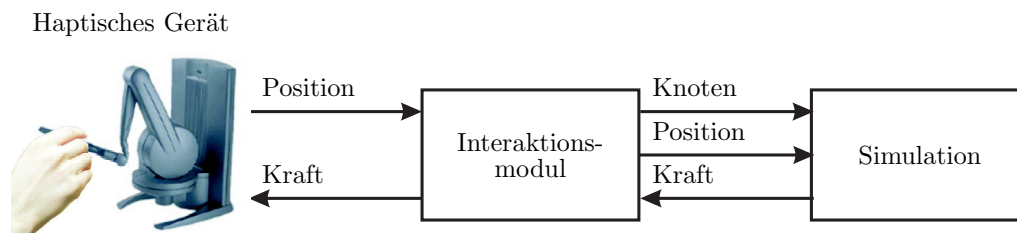


Bild 5.1: Systemaufbau für haptisches Rendering

5.1 Menschliche Wahrnehmung

Das Ziel des haptischen Rendering ist, die Manipulation des deformierbaren Körpers möglichst realistisch darzustellen. Diese Arbeit beschränkt sich auf die realistische Darstellung der Reaktionskräfte bei weichen Kontakten.

Die Krafrückkopplung zum Menschen erfolgt über ein Gerät, dessen Taktfrequenz vom darzustellenden Effekt abhängt. So wird für eine Krafrückmeldung eine Frequenz von 300 Hz empfohlen und für eine Vibrationsrückmeldung eine Frequenz von 3 kHz. In virtuellen Umgebungen hängt die notwendige Taktfrequenz von der Umgebungssteifigkeit ab und reicht von 100 Hz bis 1kHz. Oftmals wird eine Taktfrequenz von 1kHz realisiert, was einen guten Kompromiss zwischen Darstellungstreue, Rechenzeit und auch Kosten ermöglicht.

Eine funktionelle Beschreibung für die menschliche Kraftwahrnehmung ist das *Weber'sche Gesetz*. Es kann auch bei anderen Sinnesmodalitäten angewendet werden und lautet

$$\frac{F - F_{\text{Ref}}}{F_{\text{Ref}}} = c_{\text{Weber}} \quad (5.1)$$

Hierbei ist F der neue wahrgenommene Sinnesreiz, in diesem Fall eine Kraft, F_{Ref} ist die Referenzkraft und c_{Weber} ist eine Konstante. Der Zähler wird auch *just noticeable difference* (JND) $JND = F - F_{\text{Ref}}$ genannt und bezeichnet die gerade noch wahrnehmbare Kraftänderung. In Worten bedeutet das Weber'sche Gesetz, dass die wahrnehmbare Kraftänderung in Relation zur Referenzkraft ein konstantes Verhältnis bilden. Mehrere experimentelle Untersuchungen zeigten, dass bei Kräften die Weber'sche Konstante c_{Weber} zwischen 7% und 10% liegt. Wird beispielsweise eine Referenzkraft von $F_{\text{Ref}} = 10 \text{ N}$ und eine Konstante $c_{\text{Weber}} = 10\%$ angenommen, so wird nach Gleichung (5.1) davon ausgegangen, dass der Mensch eine Kraftänderung erst wahrnehmen kann, wenn die Kraft kleiner 9 N oder größer 11 N ist.

Beim haptischen Rendering muss für die Krafrückkopplung die Reaktionskraft mittels Simulation berechnet werden. Im vorherigen Kapitel wurde gezeigt, dass bei deformierbaren Körpern die Berechnung in der Regel länger dauert als die für eine Krafrückkopplung zulässigen 1 – 3 ms. Die Simulation der Kräfte steht somit nicht in Echtzeit zur Verfügung und durch die größere Rechenzeit entsteht eine Diskretisierung des sonst glatten Kraftverlaufs. Zur Illustration dieses Problems wird zunächst ein linearer eindimensionaler Zusammenhang zwischen Verschiebung x und Kraft F angenommen

$$F = cx \quad (5.2)$$

mit einer Steifigkeit c . Das kontinuierliche Kraftsignal ist im Bild 5.2 veranschaulicht. Aufgrund der Rechenzeit ergibt sich eine Diskretisierung des Kraftsignals. Das mit einer fiktiven Rechenzeit von 50 ms diskretisierte Kraftsignal ist im Bild 5.3 gezeichnet.

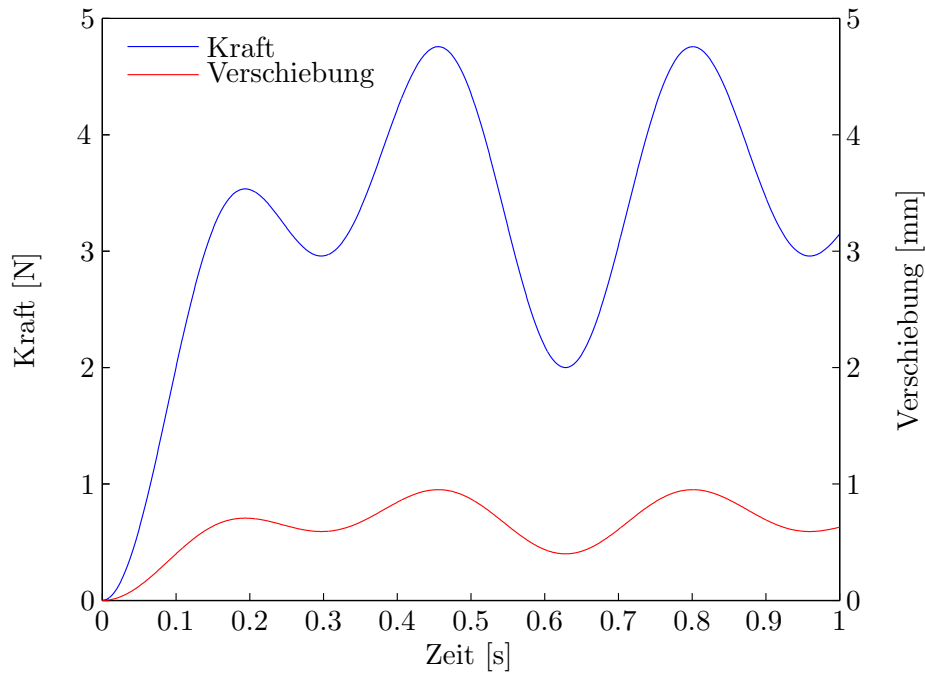


Bild 5.2: Kontinuierliches Kraftsignal

Das diskretisierte Kraftsignal wird vom menschlichen Benutzer als störend und unrealistisch wahrgenommen. Der Benutzer spürt Kraftsprünge, obwohl er einen kontinuierlichen Kraftverlauf erwartet. Dies kann zu Irritationen beim Benutzer, zu ungenauen oder unkontrollierten Bewegungen und auch zur Instabilität des Gesamtsystems führen. Daher sollte der Kraftverlauf stets so glatt sein, wie ein Benutzer es mit seiner Erfahrung erwartet.

5.2 Glättung von diskretisierten Kraftsignalen

Ein Verbesserungsansatz ist der Einsatz eines Filters zur Glättung des diskretisierten Kraftsignals. Dieses kann in das Interaktionsmodul (siehe Bild 5.1) integriert werden. Einfache, aber auch effiziente Filter in Form von PT1, PT2 und allgemeinen PTn-Übertragungsgliedern werden im Folgenden untersucht und für die Anwendung der Krafterückkopplung ausgelegt. Ein großer Vorteil dieser Filter ist die geringe Anzahl von Parametern, die einfach und schnell variiert werden können. Dies ist insbesondere wichtig, wenn die Filter nach dem subjektiven Empfinden von Personen ausgelegt werden und deswegen auf eine kurze Experimentierzeit geachtet werden muss.

Ein einfaches Tiefpassfilter ist das PT1-Glied. Es ist ein proportional wirkendes Verzögerungsglied 1. Ordnung. Die Differentialgleichung lautet

$$T_1 \dot{y}(t) + y(t) = ku(t) \quad (5.3)$$

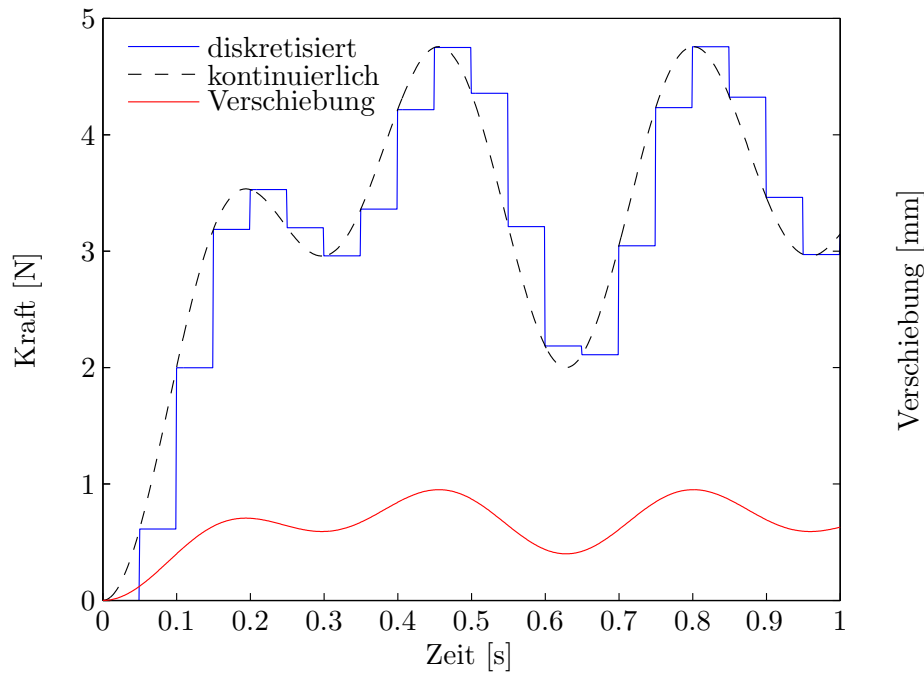


Bild 5.3: Diskretisiertes Kraftsignal

mit dem Ausgang $y(t)$, dem Eingang $u(t)$, der Zeitkonstanten T_1 und dem Verstärkungsfaktor k . Im Folgenden wird $k = 1$ verwendet um stationäre Genauigkeit sicherzustellen.

Bild 5.4 zeigt die Filterung der diskretisierten Kraftsignale aus Bild 5.3 mit unterschiedlichen Zeitkonstanten T_1 .

Die Differentialgleichung eines PT2-Glieds lautet

$$T_2^2 \ddot{y}(t) + 2dT_2 \dot{y}(t) + y(t) = u(t) \quad (5.4)$$

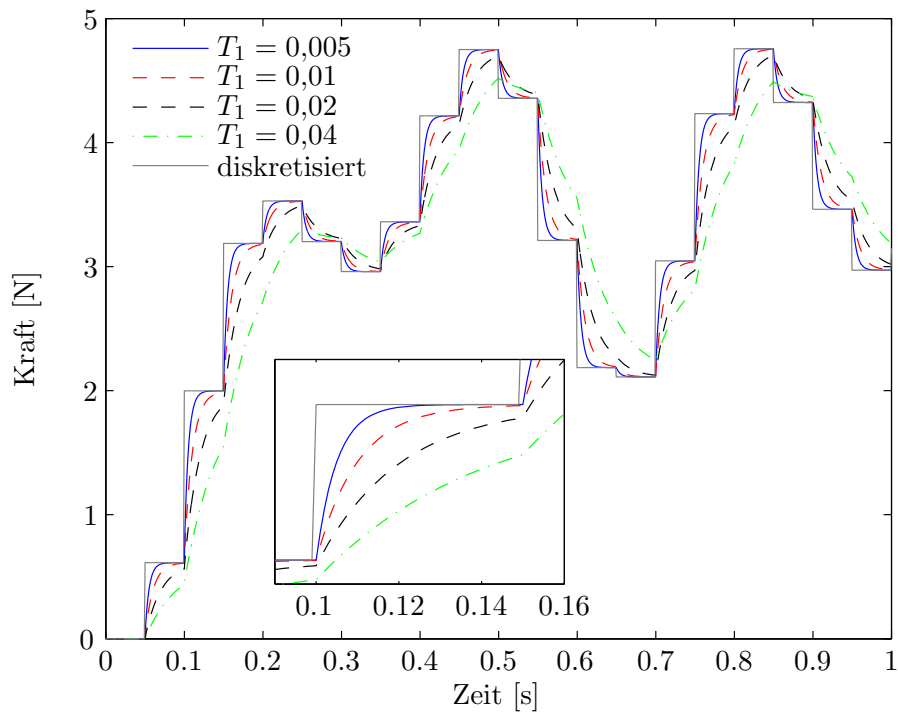
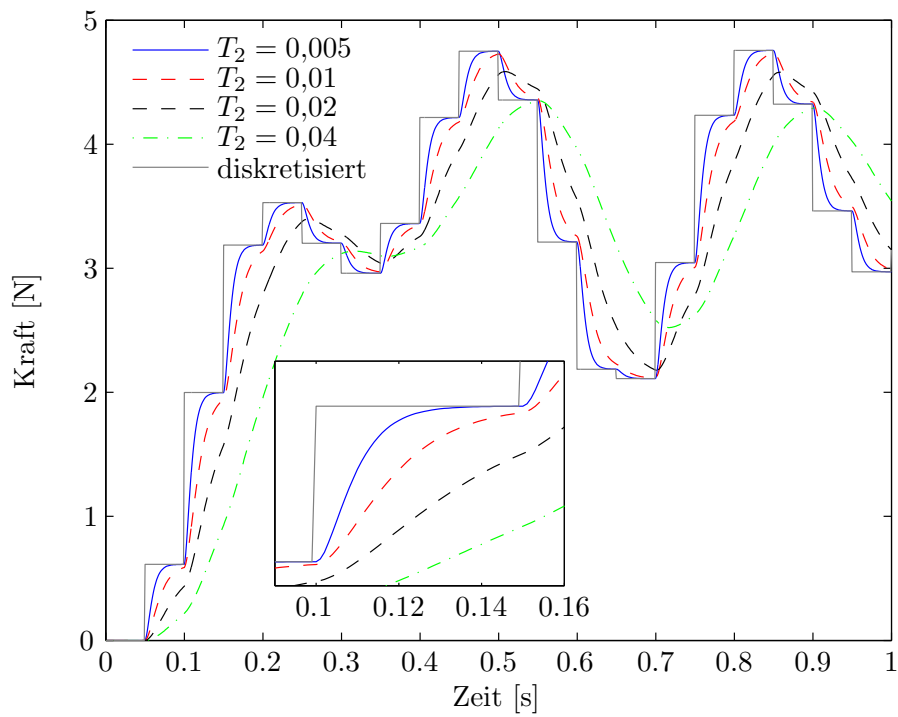
mit der Zeitkonstanten T_2 und der Dämpfung d . Im Folgenden wird eine kritische Dämpfung $d = 1$ vorausgesetzt und das PT2-Glied entspricht damit der Reihenschaltung von zwei PT1-Gliedern. Im Bild 5.5 sind die mit einem PT2-Glied gefilterten Kraftsignale eingezeichnet, wobei die Zeitkonstante T_2 variiert wird.

Ein PTn-Glied kann als Reihenschaltung von n PT1-Gliedern konstruiert werden. Die zugehörige Differentialgleichung lautet

$$\sum_{k=0}^n \binom{n}{k} T_n^k y^{(k)}(t) = u(t) \quad (5.5)$$

mit der k -ten Ableitung $y^{(k)}(t)$ und der Zeitkonstanten T_n .

Die gefilterten Kraftsignale mit PTn-Gliedern bis zur Ordnung 4 sind im Bild 5.6 dargestellt. Es wurde dabei jeweils die gleiche Zeitkonstante T_i , $i = 1..4$ verwendet.

**Bild 5.4:** PT1-Glied zur Glättung des diskretisierten Signals**Bild 5.5:** PT2-Glied zur Glättung des diskretisierten Signals

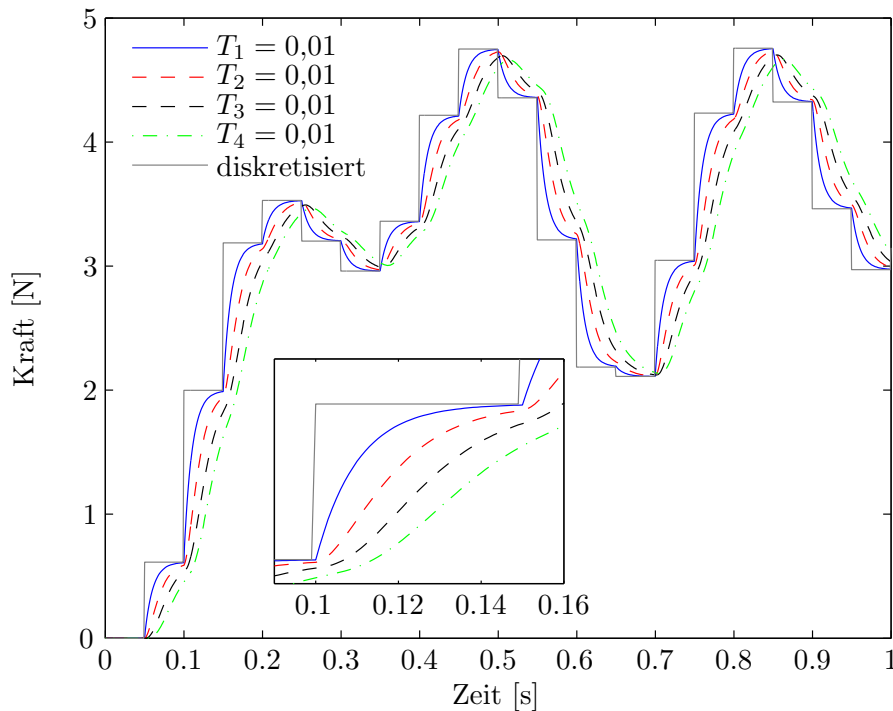


Bild 5.6: Vergleich der Filterordnung bei gleicher Zeitkonstante

In den Bildern 5.4 und 5.5 kann man erkennen, dass mit einer kleinen Zeitkonstante das diskretisierte Signal genauer angenähert wird und die Stufencharakteristik des Eingangssignals weitgehend erhalten bleibt. Dagegen bewirkt eine große Zeitkonstante ein flacheres Ausgangssignal und auch eine größere Zeitverzögerung. Bei dem Vergleich der Filterordnungen im Bild 5.6 zeigt sich, dass das Ausgangssignal bei Filtern höherer Ordnung glatter ist und eine höhere Differenzierbarkeitsordnung besitzt, was sich auch mit der Gleichung (5.5) theoretisch bestätigen lässt.

Die Auswahl und Auslegung des Filters soll mit den Anforderungen der menschlichen Wahrnehmung geschehen. Dadurch ergeben sich widersprüchliche Vorgaben an das Filter. Auf der einen Seite soll das Filter möglichst schnell sein, um keine zusätzliche Zeitverzögerung zu erzeugen, das heißt die Zeitkonstante T_i soll klein sein. Auf der anderen Seite soll das Signal so glatt sein, dass der Mensch das Kraftsignal als kontinuierlich wahrnimmt. Dies entspricht einer großen Zeitkonstanten T_i oder einer hohen Ordnung des Filters.

Die Wahl der Zeitkonstante hängt von der diskretisierten Kraftsprunghöhe ab, die geglättet werden soll. Die Kraftsprünge selbst hängen von mehreren Faktoren ab. Als erstes von der Modellierung des deformierbaren Körpers und der damit verbundenen Rechenzeit. Bei hoher Rechenzeit ist nicht nur die Zeit zwischen den Kraftsprüngen größer, auch die Höhe der Kraftsprünge kann größer sein. Außerdem hängen die Kraftsprünge vom Material des Objekts ab. Bei hoher Steifigkeit bewirken auch kleine Verformungsänderungen eine große Änderung der Kraft. Ein weiterer Faktor ist die Eingangsbewegung. So ändert sich bei einer schnellen Bewegung die Reaktionskraft entsprechend schnell und es treten dadurch größere Kraftsprünge auf.

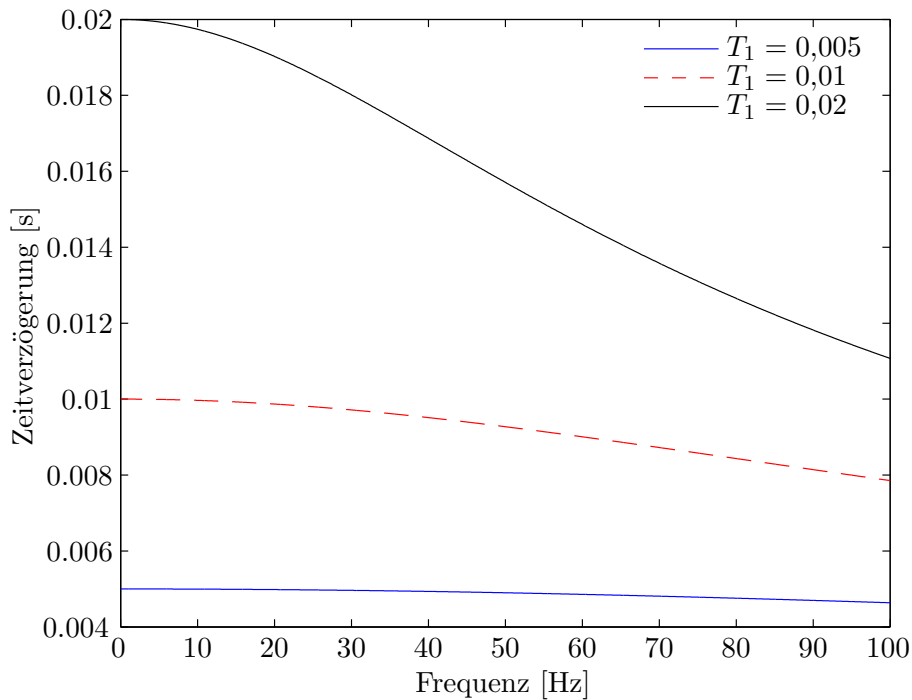


Bild 5.7: Zeitverzögerung eines PT1-Gliedes in s

Die Zeitverzögerung, die durch die Filterung entsteht, kann durch den Phasenwinkel φ des Übertragungsglieds abgeschätzt werden. Beim PT1-Glied lautet er

$$\varphi = \arctan(-\omega T_1) \quad (5.6)$$

wobei ω die Frequenz des Eingangssignals ist. Die frequenzabhängige Zeitverzögerung Δt durch das Filter ist somit

$$\Delta t = -\frac{\varphi}{\omega} = -\frac{\arctan(-\omega T_1)}{\omega} \quad (5.7)$$

In Bild 5.7 sind die Zeitverzögerungen eingezeichnet, die sich durch ein PT1-Glied mit verschiedenen Zeitkonstanten T_1 ergeben.

Man erkennt, dass ein niederfrequentes Signal eine größere Zeitverzögerung als ein hochfrequentes besitzt. Die maximale Zeitverzögerung ist gleich der Zeitkonstanten des PT1-Glieds. Dies ergibt sich aus der Grenzwertbetrachtung

$$\lim_{\omega \rightarrow 0} \Delta t = T_1 \quad (5.8)$$

Die Berechnung der Zeitverzögerung durch Glieder höherer Ordnung erfolgt analog und es resultiert die maximale Zeitverzögerung für ein PTn-Glied

$$\lim_{\omega \rightarrow 0} \Delta t = nT_n \quad (5.9)$$

Die bisherigen Betrachtungen der Filter erfolgten mit kontinuierlichen Signalen. Für

einen Rechneinsatz müssen diese nach der Auslegung noch diskretisiert werden. Dazu wird die Ableitung durch den Differenzenquotienten angenähert

$$\dot{y}(t_k) = \dot{y}_k \approx \frac{y_k - y_{k-1}}{h} \quad (5.10)$$

mit der Zeitschrittweite h des Filters. Setzt man diese Approximation in die Differentialgleichungen der Übertragungsglieder ein, erhält man eine zeitdiskretisierte Form. Wenn man diese Darstellung nach dem aktuellen Ausgang y_k auflöst, ergibt sich ein Filtergesetz, in dem der Ausgang nur vom Eingang und von zurückliegenden Ausgängen abhängt.

Für das PT1-Glied mit der Differentialgleichung (5.3) lautet das digitale Filtergesetz

$$y_k = \frac{h}{T_1 + h} u_k + \frac{T_1}{T_1 + h} y_{k-1} \quad (5.11)$$

Approximiert man die Differentialgleichung (5.4) des PT2-Glieds mit (5.10), ergibt sich das digitale Filtergesetz

$$y_k = \left(\frac{h}{T_2 + h} \right)^2 u_k + 2 \frac{T_2}{T_2 + h} y_{k-1} - \left(\frac{T_2}{T_2 + h} \right)^2 y_{k-2} \quad (5.12)$$

Für das allgemeine PTn-Glied erhält man aus der Gleichung (5.5) das Filtergesetz

$$y_k = \left(\frac{h}{T_n + h} \right)^n u_k + \sum_{i=1}^n c_i y_{k-i} \quad (5.13)$$

mit den Koeffizienten

$$c_i = (-1)^{i-1} \binom{n}{i} \left(\frac{T_n}{T_n + h} \right)^i \quad (5.14)$$

5.3 Experimente und Ergebnisse

Im Folgenden werden zwei Experimente präsentiert. Das erste Experiment zeigt die Auslegung des Filters für haptische Anwendungen. Dadurch erhält man Werte für die Filterparameter, um diese mit einem haptischen Gerät einsetzen zu können. Das zweite Experiment ist ein Beispiel für das haptische Rendering. Dabei wird eine Simulation eines virtuellen Objekts durchgeführt, wobei der Benutzer interaktiv das Modell verformen kann und die Reaktionskräfte spürt.

Filterauslegung Das Ziel der Filterauslegung ist eine obere Grenze für die Zeitkonstante eines PTn-Filters zu ermitteln, so dass sich ein diskretisiertes Signal glatt anfühlt. Als Versuchsumgebung wird ein PHANToM Desktop verwendet und die Krafterückkopplung ist proportional zur Eingangsposition. Der Versuch entspricht

somit dem Ziehen an einer virtuellen linearen Feder. Es werden mehrere Steifigkeiten eingestellt, um deren Auswirkungen auf die Kraftsprünge zu untersuchen. Für die Diskretisierung des Kraftsignals wird eine künstliche Rechenzeit von 100 ms eingestellt. Die interne Taktrate des Geräts und somit auch die Zeitschrittweite des Filters ist 1000 Hz beziehungsweise 1 ms. Die Aufgabe für die Versuchspersonen ist, die Zeitkonstante für die Filter anzupassen, so dass sich das gefilterte Kraftsignal kontinuierlich anfühlt.

In Tabelle 5.1 sind die experimentell ermittelten Zeitkonstanten bis zur Filterordnung 4 eingetragen. Es wird dabei der Mittelwert der Ergebnisse von 8 Versuchspersonen gebildet.

Tabelle 5.1: Experimentell ermittelte Zeitkonstanten

Filter	Zeitkonstante
PT1	0.025
PT2	0.013
PT3	0.008
PT4	0.007

In den Bildern 5.8 und 5.9 sind aufgezeichnete Kraftsignale dargestellt. In dem Experiment wird das kontinuierliche, das diskretisierte und das gefilterte Signal abgespeichert. Die Zeitkonstante für das PT1-Glied ist $T_1 = 30$ ms und für das PT2-Glied $T_2 = 15$ ms. Das mit diesen Einstellungen gefilterte Kraftsignal wird von den Versuchspersonen als glatt wahrgenommen.

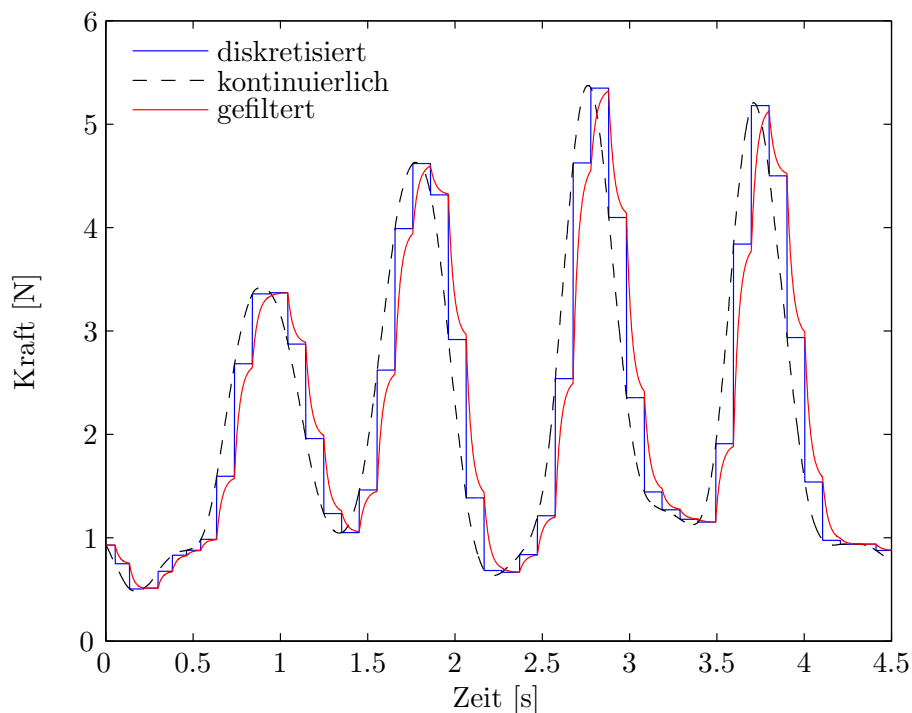


Bild 5.8: Filterung mit PT1-Glied mit $T_1 = 0.03$ s

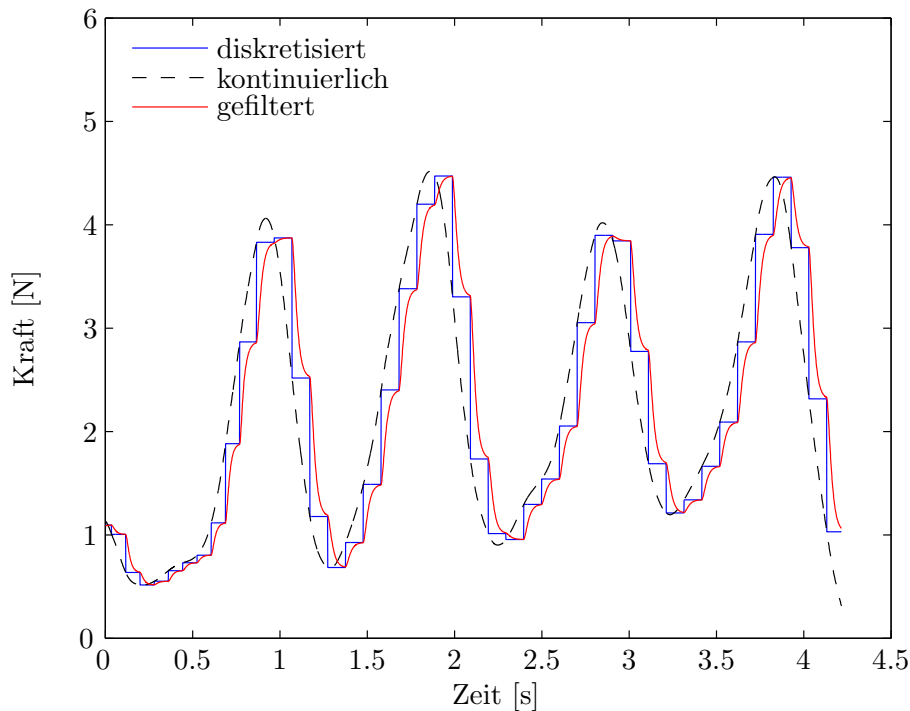


Bild 5.9: Filterung mit PT2-Glied mit $T_2 = 0.015$ s

Echtzeit Manipulation eines virtuellen Objekts Die vorgestellten Techniken werden nun in einem virtuellen Szenario demonstriert. Das virtuelle Objekt, das manipuliert beziehungsweise verformt werden kann, ist ein Hase (siehe Bild 5.10). Der Hase ist Teil des *Stanford 3D Scanning Repository* [118]. Das ursprüngliche Modell wurde reduziert und besteht hier aus 571 Punkten und 1873 Tetraedern. Die Modellierung erfolgt unter Berücksichtigung der geometrischen Nichtlinearitäten und das System wird dynamisch betrachtet. Die Zeitintegration erfolgt mit dem Newmark-Verfahren gemäß Algorithmus 4.6, wobei die Zeitschrittweite der Rechenzeit eines Integrationsschritts angepasst wurde. Dadurch ist die Simulationszeit in etwa synchron zur realen Zeit. In diesem Beispiel beträgt die Zeitschrittweite 100 ms. Als Interaktionsart wird die Verschiebungsinteraktion gewählt. Das Eingabegerät ist ein PHANToM Desktop und der Benutzer kontrolliert über den Eingabestift einen Knotenpunkt des Modells. Für diesen Knoten wird die Position vorgegeben und die Reaktionskraft berechnet. Zur Glättung der durch die Berechnung diskretisierten Signale wird ein PT2-Glied mit einer Zeitkonstanten $T_2 = 0.015$ s integriert.

Einige Momentaufnahmen der Simulation sind im Bild 5.11 zu sehen. Der Knotenpunkt P , an dem die Verschiebungsinteraktion stattfindet, ist ebenfalls eingezeichnet. Im Bild 5.12 ist der Kraftverlauf für eine Bewegung dargestellt. Es sind zum einen die diskretisierten Kräfte aus der Simulation eingezeichnet und zum anderen die durch das PT2-Filter geglätteten Kräfte. Die gefilterten Kräfte werden dabei vom Benutzer als glatt empfunden.

Mit der vorgestellten Simulationsarchitektur besteht die Möglichkeit ein haptisches Rendering von nahezu beliebigen Objekten durchzuführen. Die notwendige Geome-

triebsbeschreibung des Modells kann aus vielen Konstruktions- und Modellierungsprogrammen mit standardisierten Dateiformaten exportiert werden. Anschließend müssen für die vollständige Beschreibung die Materialparameter und die Simulationseigenschaften gewählt werden. Diese Wahl bestimmt die Simulationskomplexität und somit die Rechenzeit. Für die Echtzeitfähigkeit muss die Simulationszeit mit der realen Zeit synchronisiert werden. Bei einer konstanten Anzahl von Iteration innerhalb eines Simulationsschrittes, lassen sich sogar harte Echtzeitanforderungen erfüllen, da die Anzahl der Rechenoperationen gleich bleibt.

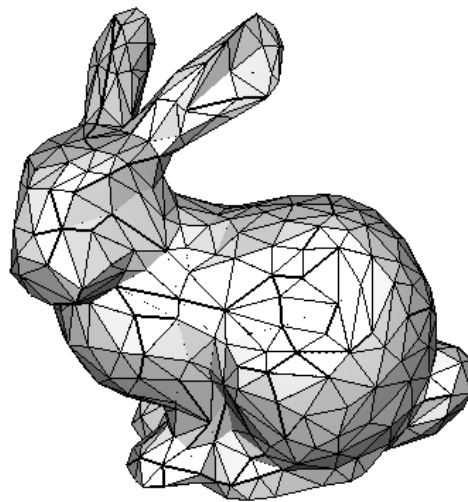


Bild 5.10: Modell des Hasen

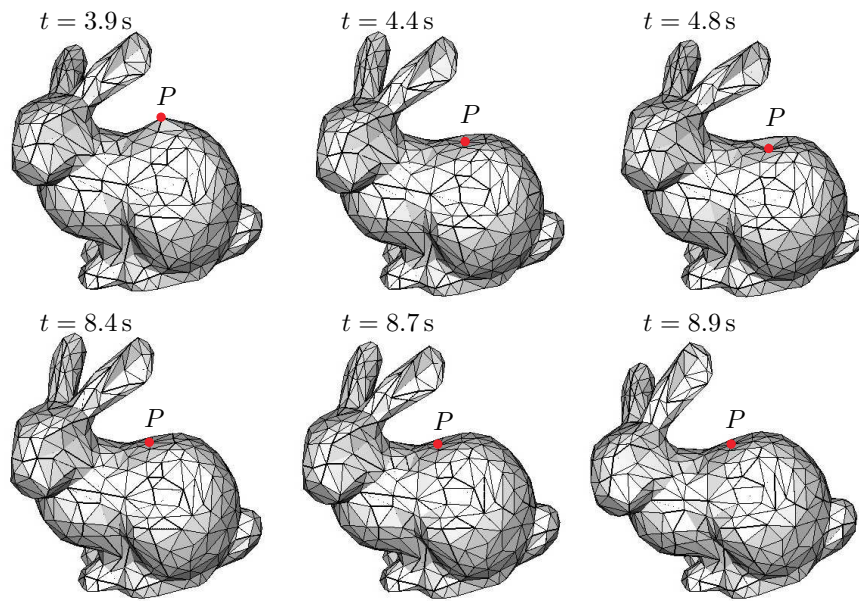


Bild 5.11: Dynamische Simulation

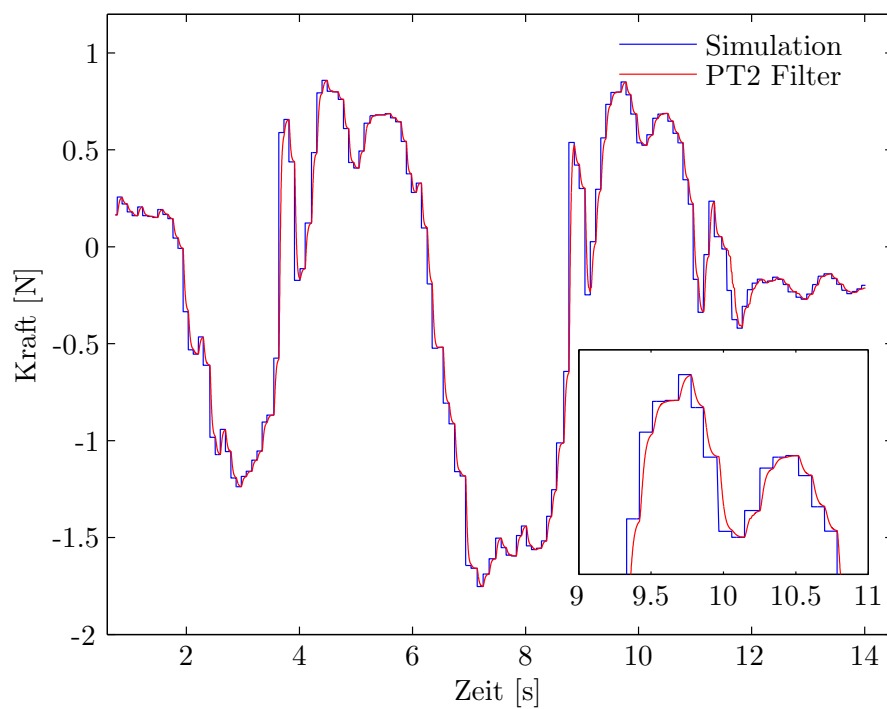


Bild 5.12: Kraftverlauf

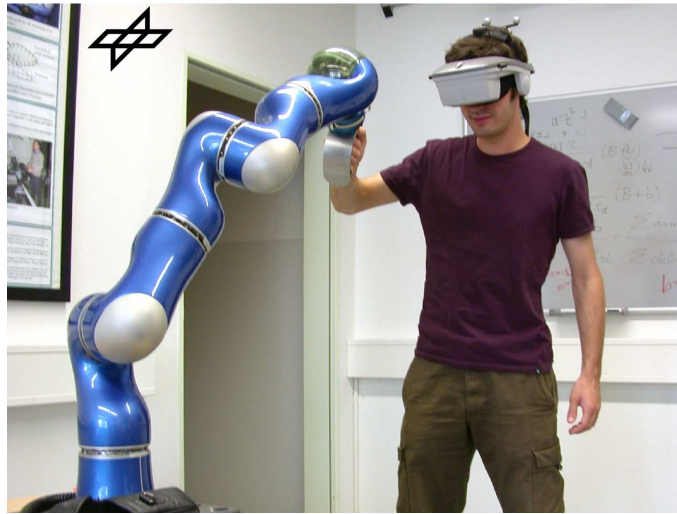
6 Kraftprädiktion in der Telepräsenz

Die zweite Anwendung ist die Kraftprädiktion in Telepräsenzsystemen. Telepräsenz bedeutet, einem Menschen mit technischen Mitteln ein Empfinden zu geben, so dass er sich in einer entfernten Umgebung präsent fühlt. Dies kann mit visuellen, auditorischen und haptischen Eindrücken geschehen. Angewendet in der Telerobotik ist das Ziel dabei, die Durchführung von Manipulationsaufgaben in entfernten, meist nicht zugänglichen Umgebungen durch den Menschen. Als Anwendungsbeispiele können hier Satellitenreparatur im Weltraum, Minenentschärfung oder chirurgische Operationen genannt werden. Im Allgemeinen sind das Umgebungen, in denen es zu gefährlich, zu teuer oder zu aufwendig ist einen Menschen vor Ort zu haben. Ein Telepräsenzsystem besteht dabei auf der einen Seite aus einem menschlichen Operator, der mit entsprechenden Eingabe- und Rückkopplungsgeräten ausgestattet ist. In der entfernten Umgebung befindet sich ein Teleoperator, mit dessen Aktorik, Sensorik und Werkzeugen man Arbeiten verrichten kann. Der Operator steuert dabei den Teleoperator und gemessene Größen werden vom Teleoperator an den Operator gesendet. Der Operator und der Teleoperator bilden einen geschlossenen Regelkreis. Die Kommunikation zwischen der lokalen Operatorumgebung und der entfernten Teleoperatorumgebung erfolgt über Übertragungswege, wie zum Beispiel LAN, Internet oder Funk.

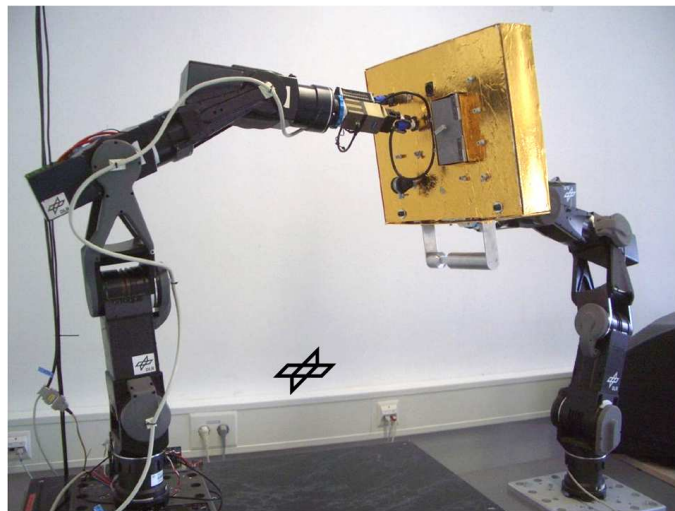
Als Beispiel eines Telepräsenzsystems ist hier ein System eines Teilprojekts des Sonderforschungsbereichs 453 „Wirklichkeitsnahe Telepräsenz und Teleaktion“ aufgeführt. Das Szenario wurde am Deutschen Zentrum für Luft- und Raumfahrt (DLR) im Institut für Robotik und Mechatronik realisiert. Verschiedene Algorithmen zur Kraftsimulation und Kraftprädiktion aus dieser Arbeit wurden mit diesem System getestet. Das Szenario entspricht einer Satellitenreparatur im Weltraum, bei der der Satellit zunächst eingefangen werden muss und anschließend mehrere Manipulationsaufgaben durchzuführen sind, wie zum Beispiel Kabelverbindungen öffnen und umstecken. Die Operatorseite ist im Bild 6.1 oben zu sehen. Der Operator bedient einen Leichtbauroboter, der als haptisches Gerät dient, über den er den Teleoperator ansteuert und die Kräfte rückgekoppelt werden. Das Sichtsystem ist ein Head-Mounted Display (HMD), welches ein Videobild der Umgebung wiedergibt. Die entfernte Umgebung besteht aus einem Teleoperator und einer Versuchsfläche (Bild 6.1 unten), die einen Teil eines Satelliten darstellt. Der Teleoperator ist ebenfalls ein Leichtbauroboter mit einem Greifer als Werkzeug.

Im Allgemeinen besteht das Telepräsenzsystem aus einer haptischen, visuellen und auditorischen Signalarückführung. Im Folgenden wird der haptische Kanal des Systems genauer untersucht und erweitert. Im Bild 6.2 ist der haptische Kanal mit Zeitverzögerungen durch die Übertragung dargestellt.

Operator



Teleoperator

**Bild 6.1:** Telepräsenzplattform am DLR

Die unvermeidlichen Zeitverzögerungen durch die Übertragungsstrecke wirken sich störend auf das Telepräsenzsystem aus. Ist die Zeitverzögerung zu groß, ist ein stabiles Arbeiten sogar unmöglich. Zur Lösung dieses Problems existieren verschiedene regelungstechnische Ansätze, die bereits erfolgreich demonstriert wurden [91], [92], [9]. Diese Ansätze funktionieren bei kleinen Zeitverzögerungen sehr gut, bei größeren Zeitverzögerungen verringert sich aber die Agilität des Systems stark. Ein neuer Ansatz zur Kompensation der Latenzzeiten ist die Anwendung von lokalen Simulationsmodellen in Telepräsenzsystemen. Mit den Simulationsmodellen können die Kräfte, die in der entfernten Umgebung auftreten, prädiziert werden.

Ein zusätzlicher Ansatz zur Kraftprädiktion ist die Bewegungsprädiktion zur Vorhersage der Operatorbewegung. Die Bewegungsprädiktion reduziert dabei die Differenz zwischen Operator- und Teleoperatorbewegung und verringert somit den Einfluss

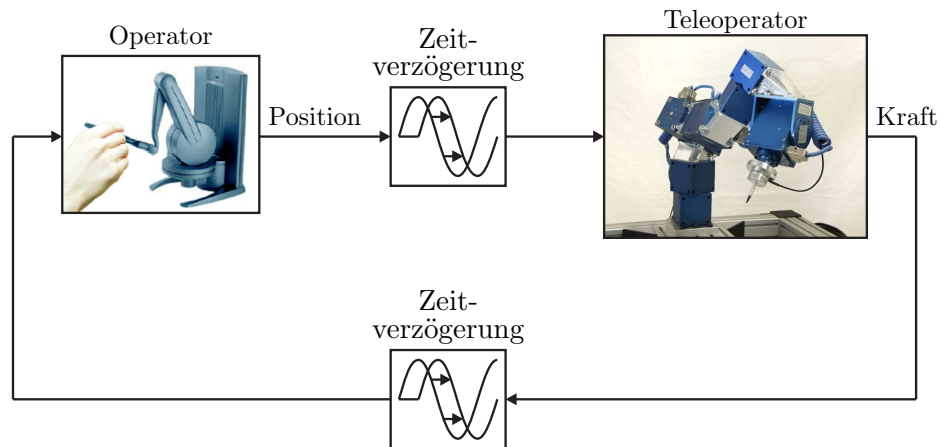


Bild 6.2: Telepräsenzsystem

der Zeitverzögerung im Übertragungskanal vom Operator zum Teleoperator. Dieser Ansatz ist mit den Algorithmen in [26] genauer beschrieben.

6.1 Kraftprädiktion zur Kompensation von Latenzzeiten

Die Idee der Kraftprädiktion ist eine Erweiterung des Telepräsenzsystems um ein Prädiktionsmodul, das die Kontaktkräfte simuliert (siehe Bild 6.3). Die simulierten Kräfte werden anstelle der zeitverzögerten, gemessenen Kräfte zum Operator rückgeführt und sind somit zeitsynchron zur Operatorbewegung. Die Kraftprädiktion entspricht dabei einem lokalen Simulationsmodell der entfernten Umgebung. Das Simulationsmodell muss somit in einem ersten Schritt erstellt werden. Bei einer unbekanntenen Umgebung kann dies in einer experimentellen Lernphase erfolgen, in der die zeitverzögerten Messdaten zur Modellerzeugung und zur Parameteridentifikation verwendet werden. Nach der Lernphase werden die Messdaten zur Modellvalidierung genutzt, und bei einer Abweichung des Modells von den Messungen erfolgt eine Modellanpassung oder eine erneute Identifikation.

In dieser Arbeit wird der Ansatz der Kraftprädiktion insbesondere auf weiche Kontakte, die in der entfernten Umgebung auftreten, angewendet. Der weiche Kontakt wird mit den Algorithmen aus den vorherigen Kapiteln simuliert. Ein Vorgehen zur Modellidentifikation, das sich für unbekannte Umgebungen eignet, wurde in Kapitel 3 präsentiert. Dabei wird angenommen, dass der Teleoperator mit einer Kamera und einem Kraft-Momentensensor ausgestattet ist. Damit kann das deformierbare Objekt erfasst und parametrisiert werden. Anschließend dient die Simulation der deformierbaren Körper als Kraftprädiktion im Telepräsenzsystem. Die Wahl des Modellierungsansatzes – nichtlinear, linear oder korotiert – hängt von der Umgebung und dem Objekt ab, und ebenso, ob eine statische oder dynamische Betrachtung erfolgt.

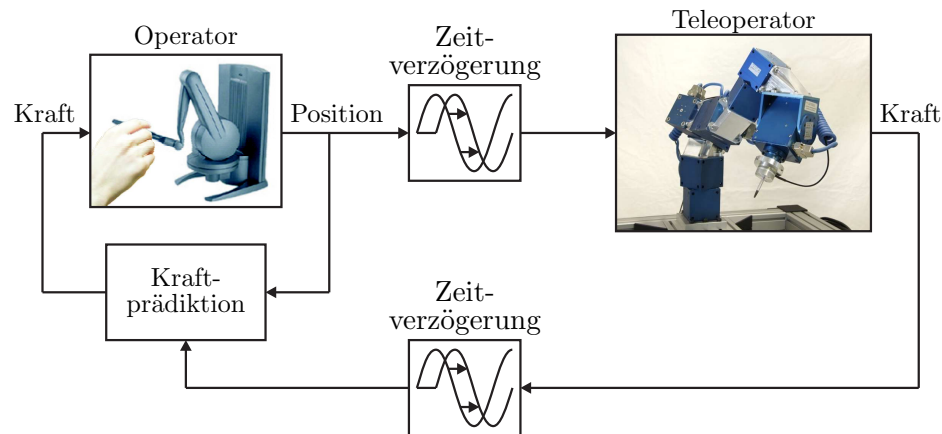


Bild 6.3: Kraftprädiktion

6.2 Experimente und Ergebnisse

Das folgende Experiment soll den Einsatz der Kraftprädiktion demonstrieren. Dazu wird ein Telepräsenzsystem aufgebaut, das die Strukturen aus den Bildern 6.2 und 6.3 realisiert. Der menschliche Operator steuert über das haptische Gerät, einem PHANTOM Desktop, den Teleoperator. Der Teleoperator ist der Roboter mit einem Kraft-Momentensensor an der Spitze, der auch in den Experimenten in Kapitel 3 verwendet wurde. Der Operator und der Teleoperator sind über ein lokales Netzwerk verbunden und die Kommunikation findet mit dem UDP-Protokoll statt. Da die Zeitverzögerung in diesem lokalen System gering ist, besteht die Möglichkeit eine simulierte Zeitverzögerung über einen Ringpuffer in das System zu integrieren. Es wird eine Zeitverzögerung von 200 ms in jede Signalrichtung eingestellt, wodurch sich eine Zeitverzögerung von insgesamt 400 ms zwischen Operatorbewegung und rückgekoppelter Kraft ergibt. Das Objekt in der entfernten Umgebung ist ein deformierbarer Kunststoffball, der im Bild 6.4 dargestellt ist. Die Geometrie und die Materialparameter sind bereits identifiziert und der Ball wird mit der linearen FEM modelliert. Die in diesem Experiment einfache Aufgabe des Operators ist, mit dem Teleoperator auf den Ball zu drücken.

Der erste Versuch wird ohne Zeitverzögerung und ohne Kraftprädiktion durchgeführt. Die Position des Operators und die Kraft sind im Bild 6.5 dargestellt. Es ist jeweils nur die Komponente normal zur Balloberfläche aufgetragen. Ist die Operatorposition kleiner als Null, befindet sich der Roboter in Kontakt mit dem Ball und die Position entspricht der Verformung des Balles. Die Kraft verschwindet, wenn der Roboter nicht in Kontakt ist, das heißt wenn die Position größer Null ist. Beim zweiten Versuch ist die simulierte Zeitverzögerung von 400 ms in das System integriert. Im Bild 6.6 ist der zeitliche Versatz zwischen Position- und Kraftverlauf deutlich zu erkennen. Ein Kontakt wird als Kraftsignal verzögert übermittelt und somit vom Operator zu spät wahrgenommen. Aufgrund der Latenzzeit ist es dem Operator nicht möglich, stabil mit dem Teleoperator auf den Ball zu drücken. Man erkennt das in den starken Signalschwankungen im Bild 6.6. Im dritten Versuch ist

die Kraftprädiktion zur Kompensation der Zeitverzögerung aktiviert. Die simulierten Kräfte werden anstelle der gemessenen an den Operator gesendet. Im Bild 6.7 sind die prädizierten und die gemessenen Kräfte eingezeichnet. Man kann erkennen, dass die simulierten Kräfte zeitsynchron zur Operatorposition sind und es dadurch dem Operator möglich ist, stabil auf den Ball zu drücken.

Dieses Experiment zeigt die Leistungsfähigkeit des Ansatzes der Kraftprädiktion. Dabei wird nach einer Identifikationsphase das haptische Rendering zur Vorhersage der Kräfte eingesetzt. Für eine Funktionalität des Systems in dieser Form ist eine gute Übereinstimmung des Simulationsmodells mit der entfernten Umgebung notwendig. Von dieser Architektur ausgehend lassen sich Varianten des Systemaufbaus ableiten, die robuster in unbekanntem Umgebungen sind. So können die zeitverzögerten Messsignale verarbeitet werden und nach einem detektierten Ereignis kann eine Modellanpassung erfolgen. Darauf basierend können problemangepasste Explorations- und Lernverfahren für unbekanntem Umgebungen entwickelt werden.

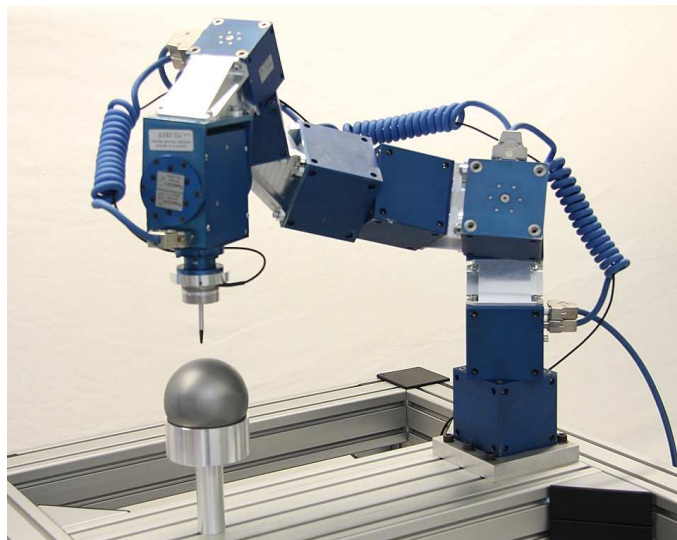


Bild 6.4: Laboraufbau

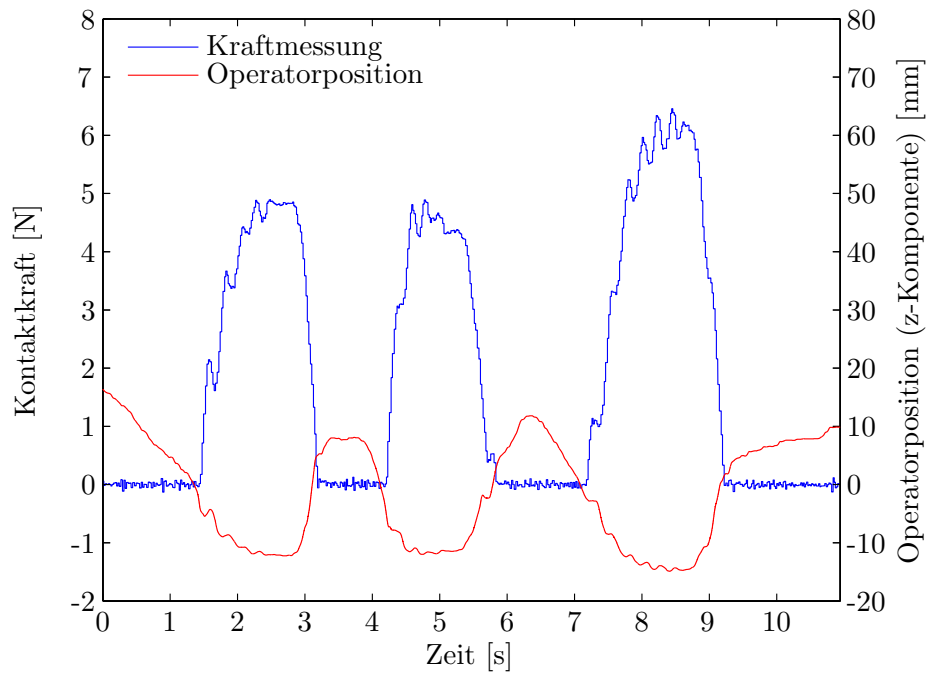


Bild 6.5: Messung ohne Zeitverzögerung

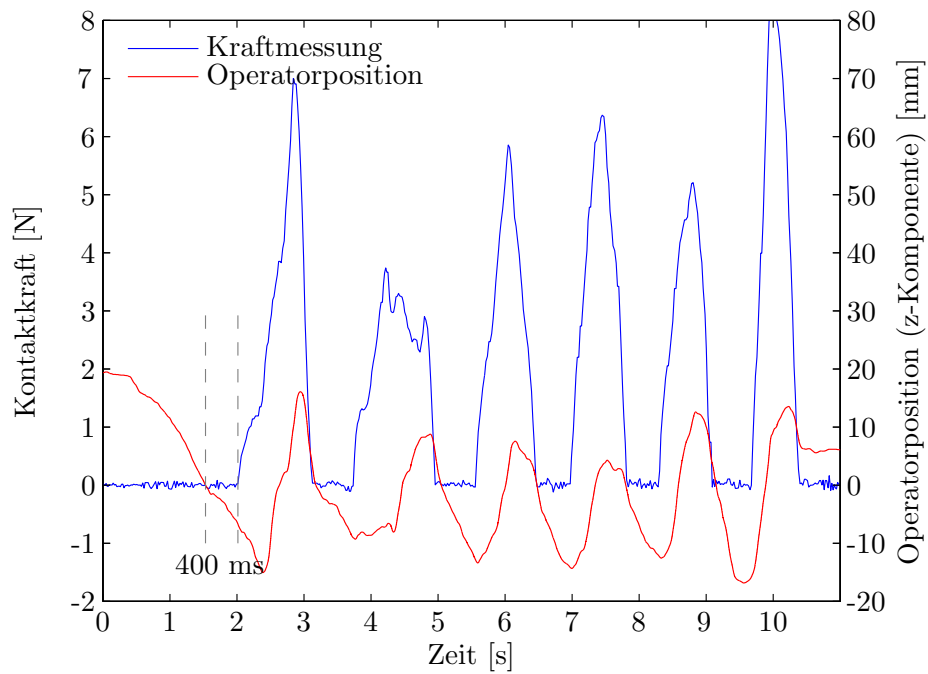


Bild 6.6: Messung mit 400 ms Zeitverzögerung ohne Kraftprädiktion

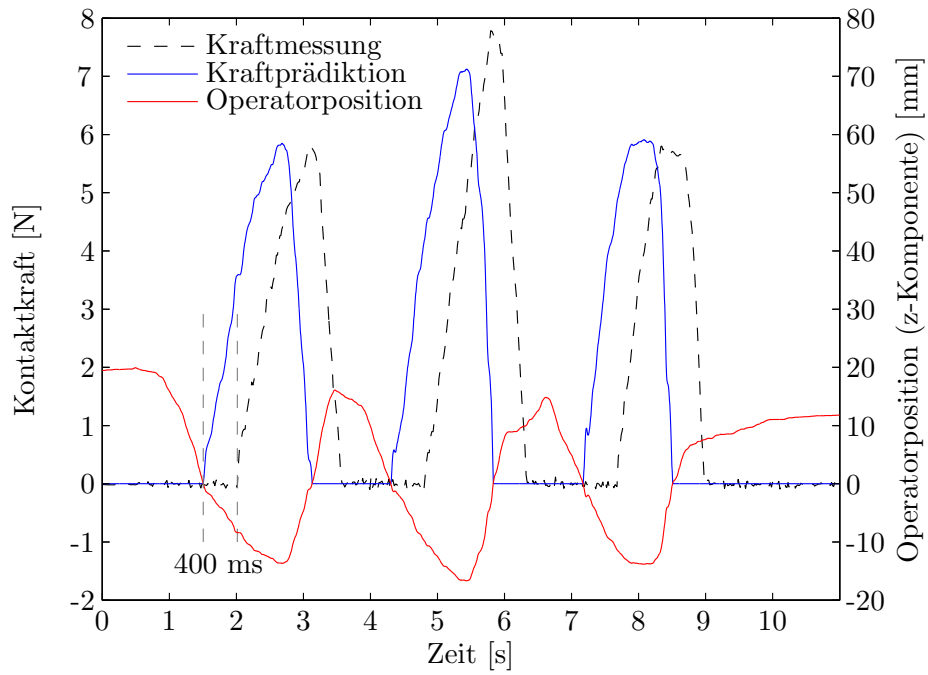


Bild 6.7: Messung mit 400 ms Zeitverzögerung und Kraftprädiktion

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Der Einsatz von Simulationen von Systemen mit deformierbaren Körpern ist in vielen Fachgebieten ein bewährtes Mittel. Benötigt man aber schnelle oder sogar interaktive Simulationen müssen die Modelle hinsichtlich einer schnellen Auswertung aufbereitet und mit effizienten Strukturen und Algorithmen implementiert werden. In dieser Arbeit werden Algorithmen zusammengestellt, die eine interaktive Simulation von deformierbaren Körpern ermöglichen. Je nach Anforderung entsteht ein Simulationsablauf, der auf unterschiedlichen Modellierungsarten, Simulationsarten und Interaktionsarten basieren kann. Von der Idee der Kraftprädiktion in Telepräsenzsystemen inspiriert, wird ein geschlossenes Vorgehen zur experimentellen Modellidentifikation, Simulationsaufbau und Systemintegration präsentiert. Im Folgenden werden die einzelnen Aspekte der Arbeit zusammengefasst.

In Kapitel 2 werden die theoretischen Grundlagen aus der Kontinuumsmechanik ausgearbeitet. Es werden drei verschiedene Möglichkeiten zur Modellierung eines deformierbaren Körpers mittels der Finite-Elemente-Methode vorgestellt. Die lineare Finite-Elemente-Methode eignet sich für kleine Verformungen des Körpers, während der nichtlineare und korotierte Modellierungsansatz auch große Verformungen realistisch abbilden kann. Die Vernetzung des Körpers erfolgt in dieser Arbeit mit Tetraederelementen.

Die Vorstellung eines Vorgehens für eine experimentelle Modellidentifikation, die speziell für Telepräsenzscenarien geeignet ist, erfolgt in Kapitel 3. Die Geometrieronstruktion verwendet eine an einem Roboterarm montierte Kamera. Mit Aufnahmen des Objekts aus mehreren Ansichten kann ein dreidimensionales Modell erstellt werden. Mit dem zu Grunde liegenden Geometriemodell können anschließend die Materialparameter durch einen Druckversuch und eine Parameteroptimierung identifiziert werden.

In Kapitel 4 werden die Simulationsalgorithmen für eine schnelle Auswertung der Systemgleichungen aufbereitet und analysiert. Es werden dabei sowohl statische als auch dynamische Probleme berücksichtigt. Außerdem werden Schnittstellen zur Interaktion mit der Simulation in die Systemgleichungen integriert. Ein Rechenzeitvergleich gibt Anhaltspunkte, welche Simulation für eine Anwendung am besten geeignet ist.

Die erste Anwendung der Simulation der deformierbaren Körper wird in Kapitel 5 präsentiert. Es handelt sich um das haptische Rendering, bei dem für eine Kraftrückkopplung die Reaktionskräfte in Echtzeit berechnet werden müssen. Da die hohen

Rechenzeitanforderungen oftmals nicht erfüllt werden können, wird ein Tiefpassfilter als Glättungsalgorithmus vorgeschlagen und für haptische Anwendungen mit Versuchen ausgelegt.

In Kapitel 6 erfolgt die zweite Anwendung der Simulation der deformierbaren Körper. Sie wird zur Kompensation der Zeitverzögerung in Telepräsenzsystemen eingesetzt. Dazu wird das System um eine operatorseitige Simulation der entfernten Umgebung erweitert. Diese dient zur zeitsynchronen Kräfteberechnung bei auftretenden weichen Kontakten. Die Leistungsfähigkeit der Kraftprädiktion wird in einem Telepräsenzsystem experimentell demonstriert.

7.2 Ausblick

Die behandelten Forschungsfragen besitzen noch Potential für Verbesserungen zur Fortführung der Arbeit. Dies geschieht bereits zum Teil in einem Nachfolgeprojekt des Sonderforschungsbereichs 453 „Wirklichkeitsnahe Telepräsenz und Teleaktion“. Im Folgenden werden Vorschläge zur Erweiterung gegeben, die im Laufe der Arbeit entstanden sind.

Eine Erweiterung der Modellierung kann die Berücksichtigung weiterer physikalischer Effekte sein. So könnten Reibung, Stöße und Strukturbrüche mehr Möglichkeiten in der Manipulation der deformierbaren Körper bieten. Die dafür benötigten zusätzlichen Modellparameter könnten dann auch experimentell identifiziert werden.

Das Vorgehen zur Objektrekonstruktion ist im Moment auf statische Umgebungen beschränkt. Eine Erweiterung wäre die Erfassung von zeitabhängigen Objekten und dynamischen Umgebungen. Eine weitere Verbesserung wäre eine erhöhte Automatisierung der 3D Rekonstruktion, was durch eine Verknüpfung der einzelnen Rekonstruktionsschritte erreicht werden könnte.

Eine Effizienzsteigerung der numerischen Auswertung könnte neben einer weiteren Programmoptimierung auch durch eine Parallelisierung der Algorithmen geschehen. Die Nutzung von Grafikprozessoren könnte die Rechenzeit verringern, ebenso ein Mehrgitterverfahren zur Lösung der Systemgleichungen, das sich in anderen Forschungsarbeiten bereits als effizient erwiesen hat. Dadurch könnten detailliertere Modelle mit mehr Freiheitsgraden simuliert werden.

Die Interaktion zwischen dem haptischen Gerät und der Simulation erfolgt über Knotenindizes, die in der Implementierung als bekannt angenommen werden. Um die Knoten während der Simulation bestimmen zu können, müsste das System um eine Kollisionserkennung für deformierbare Körper erweitert werden. Damit wäre es möglich, den Kontakt eines beliebig geformten Starrkörpers mit einem deformierbaren Körper zu simulieren.

Der vorgestellte Ansatz der Kraftprädiktion eignet sich für statische beziehungsweise langsam veränderliche Umgebungen, weil dann das Simulationsmodell mit der

Realität übereinstimmt. Weicht das Modell zu stark von der Realität ab, können die Kräfte nicht mehr richtig vorhergesagt werden. Durch Erweiterungen könnte man auch veränderliche Systeme abbilden und das lokale Simulationsmodell mit den zusätzlichen Informationen anpassen. So könnte man ein vorhandenes Videosignal nutzen um Objekte zu erkennen und zu verfolgen. Mit der Berücksichtigung starrer Körper in der Kraftprädiktion könnte ein breiteres Spektrum an Anwendungsszenarien abgedeckt werden. Erste Architektur Erweiterungen mit einem Online-Identifikationsverfahren und einer Modellanpassung zur Laufzeit wurden bereits getestet. Diese könnte man spezifischer für ein Szenario anpassen, um nicht nur einen Funktionstest im Labor zu verifizieren, sondern auch eine Validierung mit einem realistischen Telemanipulationsvorgang durchführen zu können. Ein weiterer Punkt wäre die Entwicklung fortgeschrittener Regelungsstrategien, die die Simulation beinhalten, wie beispielsweise ein nichtlinearer Beobachter, der die Simulation zur Zustandsschätzung verwendet.

A Nomenklatur

Es folgt eine kurze Beschreibung der verwendeten Schreibweise. Die Darstellung ist dabei auf die Ausdrücke beschränkt, die in der vorliegenden Arbeit Verwendung finden. Die Schreibweise entspricht der Notation aus dem Buch von PARISCH [96] und dem Vorlesungsskript von WALL [121] und ist dort ausführlich dargestellt.

Indizes a, b, c, A, B, C laufen, wenn nicht anders angegeben, von 1 bis 3.

Es gilt die Einsteinsche Summenregel

$$X^A \mathbf{E}_A = \sum_{A=1}^3 X^A \mathbf{E}_A \quad (\text{A.1})$$

Die Definition des Kronecker-Delta lautet

$$\delta^a_A = \begin{cases} 1, & \text{falls } a = A \\ 0, & \text{falls } a \neq A \end{cases} \quad (\text{A.2})$$

Die Definition des Permutationssymbols ϵ ist

$$\epsilon^a_{bc} = \begin{cases} +1, & \text{falls } \{a, b, c\} = \{1, 2, 3\}, \{3, 1, 2\}, \{2, 3, 1\} \\ -1, & \text{falls } \{a, b, c\} = \{3, 2, 1\}, \{1, 3, 2\}, \{2, 1, 3\} \\ 0 & \text{sonst} \end{cases} \quad (\text{A.3})$$

Das äußere oder dyadische Produkt zweier Vektoren $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ lautet

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix} \quad (\text{A.4})$$

und es gilt

$$(\mathbf{a} \otimes \mathbf{b}) \cdot \mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c}) \quad (\text{A.5})$$

Die Basisdarstellung eines Tensors \mathbf{T} 2.Ordnung lautet

$$\mathbf{T} = T^a_b \mathbf{e}_a \otimes \mathbf{e}^b \quad (\text{A.6})$$

$$\mathbf{T} = T_a^b \mathbf{e}^a \otimes \mathbf{e}_b \quad (\text{A.7})$$

$$\mathbf{T} = T^{ab} \mathbf{e}_a \otimes \mathbf{e}_b \quad (\text{A.8})$$

$$\mathbf{T} = T_{ab} \mathbf{e}^a \otimes \mathbf{e}^b \quad (\text{A.9})$$

Die Matrixdarstellung eines Tensors ist

$$[T^a_b] = \begin{bmatrix} T^1_1 & T^1_2 & T^1_3 \\ T^2_1 & T^2_2 & T^2_3 \\ T^3_1 & T^3_2 & T^3_3 \end{bmatrix} \quad (\text{A.10})$$

Die Tensormultiplikation in der Basisdarstellung ist definiert als

$$\mathbf{T} \cdot \mathbf{S} = T^a_b S^b_c \mathbf{e}_a \otimes \mathbf{e}^c \quad (\text{A.11})$$

und lautet in der Matrixdarstellung

$$[T^a_b] \cdot [S^b_c] = [T^a_b S^b_c] = \begin{bmatrix} T^1_1 & T^1_2 & T^1_3 \\ T^2_1 & T^2_2 & T^2_3 \\ T^3_1 & T^3_2 & T^3_3 \end{bmatrix} \begin{bmatrix} S^1_1 & S^1_2 & S^1_3 \\ S^2_1 & S^2_2 & S^2_3 \\ S^3_1 & S^3_2 & S^3_3 \end{bmatrix} \quad (\text{A.12})$$

Das innere Produkt zweier Tensoren lautet in der Basisdarstellung

$$\mathbf{T} : \mathbf{S} = T^{ab} S_{ab} \quad (\text{A.13})$$

und in der Matrixdarstellung

$$[T^{ab}] : [S_{ab}] = \text{spur}([T^{ab}]^T [S_{ab}]) \quad (\text{A.14})$$

Der räumliche Gradient eines materiellen Tensorfeldes \mathbf{X} ist

$$\text{grad} \mathbf{X} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \quad \text{oder} \quad [X^A_{,a}] = \begin{bmatrix} \frac{\partial X^1}{\partial x^1} & \frac{\partial X^1}{\partial x^2} & \frac{\partial X^1}{\partial x^3} \\ \frac{\partial X^2}{\partial x^1} & \frac{\partial X^2}{\partial x^2} & \frac{\partial X^2}{\partial x^3} \\ \frac{\partial X^3}{\partial x^1} & \frac{\partial X^3}{\partial x^2} & \frac{\partial X^3}{\partial x^3} \end{bmatrix} \quad (\text{A.15})$$

und der materielle Gradient eines räumlichen Tensorfeldes \mathbf{x} lautet

$$\text{Grad} \mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad \text{oder} \quad [x^a_{,A}] = \begin{bmatrix} \frac{\partial x^1}{\partial X^1} & \frac{\partial x^1}{\partial X^2} & \frac{\partial x^1}{\partial X^3} \\ \frac{\partial x^2}{\partial X^1} & \frac{\partial x^2}{\partial X^2} & \frac{\partial x^2}{\partial X^3} \\ \frac{\partial x^3}{\partial X^1} & \frac{\partial x^3}{\partial X^2} & \frac{\partial x^3}{\partial X^3} \end{bmatrix} \quad (\text{A.16})$$

Die räumliche Divergenz eines räumlichen Tensorfeldes \mathbf{f} ist definiert als

$$\text{div} \mathbf{f} = \text{spur}(\text{grad} \mathbf{f}) = \frac{\partial f^a}{\partial x^b} \delta_a^b = f^a_{,a} = \frac{\partial f^1}{\partial x^1} + \frac{\partial f^2}{\partial x^2} + \frac{\partial f^3}{\partial x^3} \quad (\text{A.17})$$

und die materielle Divergenz eines materiellen Tensorfeldes \mathbf{F} ist

$$\text{Div} \mathbf{F} = \text{spur}(\text{Grad} \mathbf{F}) = \frac{\partial F^A}{\partial X^B} \delta_A^B = F^A{}_{,A} = \frac{\partial F^1}{\partial X^1} + \frac{\partial F^2}{\partial X^2} + \frac{\partial F^3}{\partial X^3} \quad (\text{A.18})$$

Literaturverzeichnis

- [1] *LAPACK – Linear Algebra PACKage*. <http://www.netlib.org/lapack/>.
Version: 2007
- [2] *OpenCV - Open Computer Vision Library*.
<http://sourceforge.net/projects/opencvlibrary>. Version: 2007
- [3] ALLARD, Jérémie ; COTIN, Stéphane ; FAURE, François ; BENSOUSSAN, Pierre-Jean ; POYER, François ; DURIEZ, Christian ; DELINGETTE, Hervé ; GRISONI, Laurent: SOFA an Open Source Framework for Medical Simulation. In: *Proceedings of Medicine Meets Virtual Reality (MMVR'15)*. Long Beach, USA, February 2007
- [4] AMENTA, Nina ; CHOI, Sunghee ; KOLLURI, Ravi: The Power Crust. In: *Proceedings of Sixth ACM Symposium on Solid Modeling and Applications*, 2001, S. 249–260
- [5] ANDERSON, E. ; BAI, Z. ; BISCHOF, C. ; BLACKFORD, S. ; DEMMEL, J. ; DONGARRA, J. ; CROZ, J. D. ; GREENBAUM, A. ; HAMMARLING, S. ; MCKENNEY, A. ; SORENSEN, D.: *LAPACK Users' Guide*. Third. Philadelphia, PA : Society for Industrial and Applied Mathematics, 1999
- [6] ANDERSON, Robert J. ; SPONG, Mark W.: Bilateral Control of Teleoperators with Time Delay. In: *IEEE Transactions on Automatic Control* 34 (1989), Nr. 5, S. 494–501
- [7] ARTIGAS, J. ; PREUSCHE, C. ; HIRZINGER, G.: Wave variables based bilateral control with time delay model for space robot applications. In: *Proceedings of ROBOTIK 2004*. Munich, Germany, 2004, S. 101–108
- [8] ARTIGAS, Jordi ; KREMER, Philipp ; PREUSCHE, Carsten ; HIRZINGER, Gerd: Testbed for Telepresent On-Orbit Satellite Servicing. In: *Proceedings of the 2nd International Workshop on Human-Centered Robotic Systems (HCRS 06)*. Munich, Germany, 2006
- [9] ARTIGAS, Jordi ; VILANOVA, Jordi ; PREUSCHE, Carsten ; HIRZINGER, Gerd: Time Domain Passivity Control-based Telepresence with Time Delay. In: *Proceedings of the International Conference on Intelligent Robots and Systems*. Beijing, China, October 9-15 2006
- [10] BARBAGLI, F. ; PRATTICHIZZO, D. ; SALISBURY, K.: Multi-Contact Haptic Interaction with Deformable Objects: a Multi-rate Approach. In: *Robotics Research XI* Bd. 15. Springer, 2005, S. 466–478

- [11] BARBAGLI, F. ; PRATTICHIZZO, D. ; SALISBURY, K.: A Multirate Approach to Haptic Interaction with Deformable Objects Single and Multipoint Contacts. In: *International Journal of Robotics Research* 4 (2005), September, Nr. 29, S. 703–715
- [12] BARBIČ, Jernej ; JAMES, Doug L.: Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. In: *ACM Transactions on Graphics (SIGGRAPH 2005)* 24 (2005), Nr. 3, S. 982–990
- [13] BASDOGAN, C. ; LAYCOCK, S.D. ; DAY, A.M. ; PATOGLU, V. ; GILLESPIE, R.B.: 3-Dof Haptic Rendering. In: LIN, Ming (Hrsg.) ; OTADUY, Miguel (Hrsg.): *Haptic Rendering: Foundations, Algorithms, and Applications*. A K Peters, 2008
- [14] BATHE, Klaus-Jürgen: *Finite - Elemente - Methoden*. Springer-Verlag, 2002
- [15] BAY, Herbert ; TUYTELAARS, Tinne ; VAN GOOL, Luc: SURF: Speeded Up Robust Features. In: *Proceedings of the ninth European Conference on Computer Vision*. Graz, Austria, May 2006, S. 404–417
- [16] BELYTSCHKO, Ted ; LIU, Wing K. ; MORAN, Brian: *Nonlinear Finite Elements for Continua and Structures*. Wiley & Sons, 2000
- [17] BIRCHFIELD, Stan ; TOMASI, Carlo: Depth Discontinuities by Pixel-to-Pixel Stereo. In: *International Journal of Computer Vision* 35 (1999), December, Nr. 3, S. 269–293
- [18] BOUGUET, Jean-Yves: *Pyramidal Implementation of the Lucas Kanade Feature Tracker, Description of the algorithm*. Technical Report, Intel Corporation, 1999
- [19] BREMER, Hartmut ; PFEIFFER, Friedrich: *Dynamik und Regelung mechanischer Systeme*. Teubner Verlag, 1988
- [20] BREMER, Hartmut ; PFEIFFER, Friedrich: *Elastische Mehrkörpersysteme*. Teubner Verlag, 1992
- [21] BRO-NIELSEN, Morten ; COTIN, Stephane: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. 15 (1996), Nr. 3, S. 57–66
- [22] CAVUSOGLU, M. C. ; TENDICK, F.: Multirate Simulation for High Fidelity Haptic Interaction with Deformable Objects in Virtual Environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*. San Francisco, CA, USA, April 24-28 2000, S. 2458–2465
- [23] CHANTHASOPEEPHAN, Teeranoot ; DESAI, Jaydev P. ; LAU, Alan C. W.: Modeling Soft-Tissue Deformation Prior to Cutting for Surgical Simulation: Finite Element Analysis and Study of Cutting Parameters. In: *Transactions on Biomedical Engineering* 54 (2007), Nr. 3, S. 349–359
- [24] CHEN, Yanqing ; DAVIS, Timothy A. ; HAGER, William W. ; RAJAMANICKAM, Sivasankaran: Algorithm 8xx: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. In: *ACM Trans. Math. Software* (2006)

- [25] CIGNONI, Paolo ; CALLIERI, Marco ; CORSINI, Massimiliano ; DELLEPIANE, Matteo ; GANOVELLI, Fabio ; RANZUGLIA, Guido: MeshLab: an Open-Source Mesh Processing Tool. In: *Sixth Eurographics Italian Chapter Conference*, 129–136
- [26] CLARKE, Stella ; SCHILLHUBER, Gerhard ; ZAEH, Michael F. ; ULBRICH, Heinz: Prediction-based methods for teleoperation across delayed networks. In: *Multimedia Systems* 13 (2007), Januar 2008, Nr. 4, S. 253–261
- [27] CORTESAO, R. ; PARK, Jaeheung ; KHATIB, O.: Real-time adaptive control for haptic telemanipulation with Kalman active observers. In: *IEEE Transactions on Robotics* 22 (2006), Nr. 5, S. 987–999
- [28] COTIN, Stéphane ; DELINGETTE, Hervé ; AYACHE, Nicholas: Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation. In: *Transactions on Visualization and Computer Graphics* 5 (1999), Nr. 1, S. 62–73
- [29] COTIN, Stéphane ; DELINGETTE, Hervé ; AYACHE, Nicholas: A hybrid elastic model for real-time cutting, deformations and force feedback for surgery training and simulation. In: *The Visual Computer* 16 (2000), Nr. 7, S. 437–452
- [30] CURLESS, Brian ; LEVOY, Marc: A volumetric method for building complex models from range images. In: *Proceedings of SIGGRAPH '96*. New Orleans, USA, 1996, S. 303–312
- [31] D'AULIGNAC, D. ; CAVUSOGLU, M. C. ; LAUGIER, Christian: Modeling the Dynamics of the Human Thigh for a Realistic Echographic Simulator with Force Feedback. In: *MICCAI '99: Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1999, S. 1191–1198
- [32] DAVIS, Timothy A.: *Direct Methods for Sparse Linear Systems*. Society for Industrial & Applied Mathematics, 2006
- [33] DAVIS, Timothy A.: *CHOLMOD*.
<http://www.cise.ufl.edu/research/sparse/cholmod>. Version: 2007
- [34] DEBUNNE, Gilles ; DESBRUN, Mathieu ; CANI, Marie-Paule ; BARR, Alan H.: Dynamic real-time deformations using space & time adaptive sampling. In: *Proceedings of SIGGRAPH '01*, 2001, S. 31–36
- [35] DEUFLHARD, Peter ; BORNEMANN, Folkmar: *Numerische Mathematik 2*. Walter de Gruyter Verlag, 1994
- [36] DEUFLHARD, Peter ; HOHMANN, Andreas: *Numerische Mathematik 1*. Walter de Gruyter Verlag, 1993
- [37] DIMAIO, S. P. ; SALCUDEAN, S. E.: Needle Insertion Modeling and Simulation. In: *IEEE Transactions on Robotics and Automation* 19 (2003), Nr. 5, S. 864–875
- [38] DIMAIO, S. P. ; SALCUDEAN, S. E.: Interactive simulation of needle insertion models. In: *IEEE Transactions on Biomedical Engineering* 52 (2005), Nr. 7, S. 1167–1179

- [39] DURIEZ, C. ; ANDRIOT, C. ; KHEDDAR, A.: A multi-threaded approach for deformable/rigid contacts with haptic feedback. In: *Proceedings of Haptics Symposium in IEEE Virtual Reality*. Chicago, USA, March 27-28 2004
- [40] DURIEZ, C. ; DUBOIS, F. ; KHEDDAR, A. ; ANDRIOT, C.: Realistic haptic rendering of interacting deformable objects in virtual environments. In: *Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 1, S. 36–47
- [41] EBERHARD, Peter: *Kontaktuntersuchungen durch hybride Mehrkörpersystem / Finite Elemente Simulationen*. Shaker Verlag, 2007
- [42] EDELSBRUNNER, Herbert ; MÜCKE, Ernst P.: Three-dimensional alpha shapes. In: *ACM Transactions on Graphics* 13 (1994), Nr. 1, S. 43–72
- [43] ENGELKE, Robert: *Modellierung und Optimierung von Robotern mit einseitigen Bindungen und lokalen Verspannungen*, Technische Universität München, Germany, Dissertation, 2008
- [44] ERICKSON, D. ; WEBER, M. ; SHARF, Inna: Contact Stiffness and Damping Estimation for Robotic Systems. In: *The International Journal of Robotics Research* 22 (2003), Nr. 1, S. 41–58
- [45] ETZMUSS, Olaf ; KECKEISEN, Michael ; STRASSER, Wolfgang: A Fast Finite Element Solution for Cloth Modelling. In: *Proceedings of Pacific Conference on Computer Graphics and Applications*, 2003, S. 244–251
- [46] FABIO, Remondino: From Point Cloud to Surface: The Modeling and Visualization Problem. In: *Proceedings of International Workshop on Visualization and Animation of Reality-Based 3D Models*, 2003
- [47] FAUGERAS, Olivier: *Three-Dimensional Computer Vision (Artificial Intelligence)*. The MIT Press, 1993. – ISBN 0262061589
- [48] FORSYTH, David ; PONCE, Jean: *Computer Vision: A Modern Approach*. Prentice Hall, 2003. – ISBN 0131911937
- [49] GEORGII, Joachim: *Real-Time Simulation and Visualization of Deformable Objects*, Technische Universität München, Germany, Dissertation, 2007
- [50] GEORGII, Joachim ; WESTERMANN, Rüdiger: Interactive Simulation and Rendering of Heterogeneous Deformable Bodies. In: *Proceedings of Vision, Modeling and Visualization*, 2005, S. 383–390
- [51] GEORGII, Joachim ; WESTERMANN, Rüdiger: Mass-Spring Systems on the GPU. In: *Simulation Modelling Practice and Theory* 13 (2005), S. 693–702
- [52] GEORGII, Joachim ; WESTERMANN, Rüdiger: A multigrid framework for real-time simulation of deformable bodies. In: *Computers & Graphics* 30 (2006), Nr. 3, S. 408–415
- [53] GIBSON, Sarah F. F.: 3D ChainMail: a fast algorithm for deforming volumetric objects. In: *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, 1997, S. 149–154

- [54] GIBSON, Sarah F. F. ; MIRTICH, Brian: A Survey of Deformable Modeling in Computer Graphics / Mitsubishi Electric Research Laboratories. 1997. – Forschungsbericht
- [55] GOULD, Nicholas I. ; SCOTT, Jennifer A.: A Numerical Evaluation of Sparse Direct Solvers for the Solution of Large Sparse Symmetric Linear Systems of Equations. In: *ACM Transactions on Mathematical Software* 33 (2007), Nr. 2
- [56] GRINSPUN, Eitan ; KRYSL, Petr ; SCHRÖDER, Peter: CHARMS: A Simple Framework for Adaptive Simulation. In: *SIGGRAPH (ACM Transactions on Graphics)* 21 (2002), Jul, Nr. 3, S. 281–290
- [57] HARRIS, Cyril M. ; PIERSOL, Allan G.: *Shock and Vibration Handbook*. McGraw-Hill, 2002
- [58] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. – ISBN 0521540518
- [59] HAUTH, Michael: *Visual Simulation of Deformable Models*, Eberhard-Karls-Universität Tübingen, Germany, Dissertation, 2004
- [60] HAUTH, Michael ; STRASSER, Wolfgang: Corotational Simulation of Deformable Solids. In: *Proceedings of WSCG*, 2004, S. 137–144
- [61] HIGHAM, Nicholas ; SCHREIBER, Robert: Fast Polar Decomposition of an Arbitrary Matrix. In: *SIAM Journal on Scientific and Statistical Computing* 11 (1990), Nr. 4, S. 648–655
- [62] HING, James ; BROOKS, Ari ; DESAI, Jaydev P.: Reality Based Needle Insertion Simulation for Haptic Feedback in Prostate Brachytherapy. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Orlando, Florida, USA, May 15-19 2006, S. 619–624
- [63] HIRCHE, Sandra: *Haptic Telepresence in Packet Switched Communication Networks*, Technische Universität München, Germany, Dissertation, 2005
- [64] HIROTA, Koichi ; KANEKO, Toyohisa: Haptic Representation of Elastic Objects. In: *Presence: Teleoperators and Virtual Environments* 10 (2001), Nr. 5, S. 525–536
- [65] HOPPE, Hugues ; DEROSE, Tony ; DUCHAMP, Tom ; MCDONALD, John ; STUETZLE, Werner: Surface reconstruction from unorganized points. In: *Computer Graphics* 26 (1992), Nr. 2, S. 71–78
- [66] JAMES, Doug ; PAI, Dinesh: ArtDefo - Accurate Real Time Deformable Objects. In: *Proceedings of SIGGRAPH 99*, 1999, S. 65–72
- [67] JAMES, Doug L. ; PAI, Dinesh K.: A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics. In: *Haptics-e (www.haptics-e.org)* 2 (2001), Nr. 1
- [68] JERABKOVA, Lenka: *Interactive Cutting of Finite Elements based Deformable Objects in Virtual Environments*, RWTH Aachen University, Germany, Dissertation, 2007

- [69] KAUER, M. ; VUSKOVIC, V. ; DUAL, J. ; SZÉKELY, Gábor ; BAJKA, Michael: Inverse Finite Element Characterization of Soft Tissues. In: *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2001, S. 128–136
- [70] KÜHNAPFEL, Uwe G. ; ÇAKMAK, Hüseyin K. ; MAASS, Heiko: Endoscopic surgery training using virtual reality and deformable tissue simulation. In: *Computers & Graphics* 24 (2000), Nr. 5, S. 671–682
- [71] LANG, J. ; PAI, D.: Estimation of Elastic Constants from 3D Range-Flow. In: *Proceedings of International Conference on 3D Digital Imaging and Modeling*. Los Alamitos, CA, USA, 2001, S. 331–338
- [72] LANG, Jochen ; PAI, Dinesh K. ; WOODHAM, Robert J.: Acquisition of Elastic Models for Interactive Simulation. In: *The International Journal of Robotics Research* 21 (2002), Nr. 8, S. 713–734
- [73] LANG, Jochen ; PAI, D.K. ; WOODHAM, R.J.: Robotic Acquisition of Deformable Models. In: *Proceedings of International Conference on Robotics and Automation ICRA '02*, 2002, S. 933–938
- [74] LAYCOCK, S. D. ; DAY, A. M.: A Survey of Haptic Rendering Techniques. In: *Computer Graphics Forum* 26 (2007), Nr. 1, S. 50–65
- [75] LEE, Kyungno ; LEE, Doo Y.: Multirate control of haptic interface for stability and high fidelity. In: *Proceedings of International Conference on Systems, Man and Cybernetics*, 2004, S. 2542–2547
- [76] LEPETIT, Vincent ; FUA, Pascal: Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. In: *Foundations and Trends in Computer Graphics and Vision* 1 (2005), Nr. 1, S. 1–89
- [77] LOWE, David G.: Object recognition from local scale-invariant features. In: *International Conference on Computer Vision*. Corfu, Greece, September 1999, S. 1150–1157
- [78] LOWE, David G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110
- [79] LUCAS, B. ; KANADE, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proceedings of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, S. 674–679
- [80] MAASS, H. ; KÜHNAPFEL, U.: Noninvasive measurement of elastic properties of living tissue. In: *In 13th Int. Congress on Comp. Assisted Radiology (CARS)*, 1999, S. 865–870
- [81] MAHVASH, Mohsen ; HAYWARD, Vincent: Haptic Rendering of Cutting: A Fracture Mechanics Approach. In: *Haptics-e (www.haptics-e.org)* 2 (2001), Nr. 3
- [82] MAHVASH, Mohsen ; HAYWARD, Vincent: Haptic simulation of a tool in contact with a nonlinear deformable body. In: *Proceedings of International Symposium on Surgical Simulation and Soft Tissue Deformation*, 2003, S. 311–320

- [83] MAHVASH, Mohsen ; HAYWARD, Vincent ; LLOYD, John: Haptic Rendering of Tool Contact. In: *Proceedings of Eurohaptics*, 2002, S. 110–115
- [84] MIKOLAJCZYK, Krystian ; TUYTELAARS, Tinne ; SCHMID, Cordelia ; ZISSERMAN, Andrew ; MATAS, J. ; SCHAFFALITZKY, F. ; KADIR, T. ; GOOL, L. V.: A comparison of affine region detectors. In: *International Journal of Computer Vision* 65 (2005), Nr. 1/2, S. 43–72
- [85] MITRA, Probal ; NIEMEYER, Günter: Model-mediated Telemanipulation. In: *Journal of Robotics Research* 27 (2008), Nr. 2, S. 253–262
- [86] 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*, 2002, S. 49–54
- [87] MÜLLER, Matthias ; GROSS, Markus: Interactive Virtual Materials. In: *Proceedings of Graphics Interface 2004*, 2004, S. 239–246
- [88] NEALEN, Andrew ; MÜLLER, Matthias ; KEISER, Richard ; BOXERMAN, Eddy ; CARLSON, Mark: Physically Based Deformable Models in Computer Graphics. In: *Computer Graphics Forum* 25 (2005), Nr. 4, S. 809–836
- [89] NESME, Matthieu ; PAYAN, Yohan ; FAURE, François: Efficient, Physically Plausible Finite Elements. In: *Proceedings of Eurographics*, 2005
- [90] NIEMEYER, G. ; SLOTINE, J. J. E.: Stable adaptive teleoperation. In: *IEEE Journal of Oceanic Engineering* 16 (1991), Nr. 1, S. 152–162
- [91] NIEMEYER, G. ; SLOTINE, J. J. E.: Towards force-reflecting teleoperation over the Internet. In: *Proceedings of International Conference on Robotics and Automation*, 1998, S. 1909–1915
- [92] NIEMEYER, G. ; SLOTINE, J. J. E.: Telemanipulation with Time Delay. In: *Journal of Robotics Research* 23 (2004), Nr. 9, S. 873–890
- [93] NIENHUYS, Han-Wen ; STAPPEN, A. F. d.: Supporting cuts and finite element deformation in interactive surgery simulation / Utrecht University, Institute for Information and Computing Sciences. Version: June 2001. citeseer.nj.nec.com/nienhuys01supporting.html. PO Box 80.089, 3508 TB, The Netherlands, June 2001. – Forschungsbericht. – Elektronische Ressource
- [94] NIENHUYS, Han-Wen ; STAPPEN, A. F. d.: A surgery simulation supporting cuts and finite element deformation. In: *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI*. Utrecht, The Netherlands, October 2001, S. 145–152
- [95] PAI, Dinesh K. ; DOEL, Kees Van D. ; JAMES, Doug L. ; LANG, Jochen ; LLOYD, John E. ; RICHMOND, Joshua L. ; YAU, Som H.: Scanning Physical Interaction Behavior of 3D Objects. In: *In Computer Graphics, Annual Conference Series*, 2001, S. 87–96
- [96] PARISCH, Horst: *Festkörper-Kontinuumsmechanik*. Teubner Verlag, 2003

- [97] PARK, Jaeheung ; KHATIB, Oussama: A Haptic Teleoperation Approach Based on Contact Force Control. In: *Journal of Robotics Research* 25 (2006), Nr. 5-6, S. 575–591
- [98] PFEIFFER, Friedrich: *Mechanical System Dynamics*. Springer Verlag, 2008
- [99] PFEIFFER, Friedrich ; GLOCKER, Christoph: *Multibody dynamics with unilateral contacts*. John Wiley & Sons, 1996
- [100] PREUSCHE, Carsten ; HIRZINGER, Gerd: Haptics in telerobotics. In: *The Visual Computer* 23 (2007), Nr. 4, S. 273–284
- [101] PREUSCHE, Carsten ; REINTSEMA, Detlef ; LANDZETTEL, Klaus ; HIRZINGER, Gerd: Robotics Component Verification on ISS ROKVISS - Preliminary Results for Telepresence. In: *Proceedings of the International Conference on Intelligent Robots and Systems*. Beijing, China, October 9-15 2006
- [102] SALISBURY, J. K. ; SRINIVASAN, Mandayam A.: Phantom-Based Haptic Interaction with Virtual Objects. In: *IEEE Computer Graphics and Applications* 17 (1997), Nr. 5, S. 6–10
- [103] SAUPIN, Guillaume ; DURIEZ, Christian ; COTIN, Stephane: Contact Model for Haptic Medical Simulations, 2008, S. 157–165
- [104] SCHARSTEIN, Daniel ; SZELISKI, Richard: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. In: *Journal of Computer Vision* 47 (2002), Nr. 1/2/3, S. 7–42
- [105] SCHONER, Jeffrey L. ; LANG, Jochen ; SEIDEL, Hans-Peter: Measurement-Based Interactive Simulation of Viscoelastic Solids. In: *Computer Graphics Forum* 23 (2004), Nr. 3, S. 547–556
- [106] SCHREER, Oliver: *Stereoanalyse und Bildsynthese*. Springer-Verlag, 2005
- [107] SCHWARZ, Hans R.: *Numerische Mathematik*. Teubner Verlag, 1997
- [108] SCHWERTASSEK, Richard ; WALLRAPP, Oskar: *Dynamik flexibler Mehrkörpersysteme*. Vieweg Verlag, 1999
- [109] SHABANA, Ahmed A.: *Dynamics of multibody systems*. John Wiley & Sons, 1998
- [110] SI, Hang: *TetGen, A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. <http://tetgen.berlios.de/>. Version: 2007
- [111] SI, Hang ; GAERTNER, Klaus: Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In: *Proceedings of the 14th International Meshing Roundtable*, 2005, S. 147–163
- [112] STOER, Josef: *Numerische Mathematik 1*. Springer Verlag, 1999
- [113] STOER, Josef ; BULIRSCH, Roland: *Einführung in die Numerische Mathematik 2*. Springer Verlag, 1978
- [114] TANNER, Neal A. ; NIEMEYER, Günter: Improving Perception in Time-delayed Telerobotics. In: *International Journal Robotic Research* 24 (2005), Nr. 8, S. 631–644

- [115] TAYLOR, Zeike ; CHENG, Mario ; OURSELIN, Sébastien: High-speed nonlinear finite element analysis for surgical simulation using graphics processing units. In: *IEEE Transactions on Medical Imaging* 27 (2008), Nr. 5, S. 650–663
- [116] TERZOPOULOS, D. ; PLATT, J. ; BARR, A.H. ; FLEISCHER, K.W.: Elastically Deformable Models. In: *Computer Graphics (SIGGRAPH '87)* 21 (1987), Nr. 4, S. 205–214
- [117] THÜMMEL, Michael: *Modellbasierte Regelung mit nichtlinearen inversen Systemen und Beobachtern von Robotern mit elastischen Gelenken*, Technische Universität München, Germany, Dissertation, 2006
- [118] TURK, Greg ; LEVOY, Marc: Zippered Polygon Meshes from Range Images. In: *Proceedings of SIGGRAPH '94*. Orlando, Florida, USA, 1994, S. 311–318
- [119] ULBRICH, Heinz: *Maschinendynamik*. Teubner Verlag, 1996
- [120] WALL, Wolfgang A. ; BORNEMANN, Burkhard: Nichtlineare Finite-Element-Methoden / Lehrstuhl für Numerische Mechanik, Technische Universität München. 2006. – Vorlesungsskript
- [121] WALL, Wolfgang A. ; BORNEMANN, Burkhard: Grundlagenfach Mechanik: Nichtlineare Kontinuumsmechanik / Lehrstuhl für Numerische Mechanik, Technische Universität München. 2007. – Vorlesungsskript
- [122] WASFY, Tamer M. ; NOOR, Ahmed K.: Computational strategies for flexible multibody systems. In: *Applied Mechanics Reviews* 56 (2003), Nr. 6, S. 553–613
- [123] WRIGGERS, Peter: *Nichtlineare Finite-Element-Methoden*. Springer Verlag, 2001
- [124] WRIGGERS, Peter: *Computational Contact Mechanics*. John Wiley & Sons, 2002
- [125] ZIENKIEWICZ, Olgried: *Methode der finiten Elemente*. Carl Hanser Verlag, 1984