



Institut für Medizin

*Konstruktion von Oberflächen
mit aktiven Konturmodellen*

Herwig Zilken

***Konstruktion von Oberflächen
mit aktiven Konturmodellen***

Herwig Zilken

Berichte des Forschungszentrums Jülich ; 3649
ISSN 0944-2952
Institut für Medizin Jül-3649
D 82 (Diss. RWTH Aachen)

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
D-52425 Jülich · Bundesrepublik Deutschland
02461/61-6102 · Telefax: 02461/61-6103 · e-mail: zb-publikation@fz-juelich.de

Konstruktion von Oberflächen mit aktiven Konturmodellen

In der medizinischen Diagnostik spielen anatomische 3D-Volumendatensätze eine große Rolle. Oftmals lassen sich wichtige Eigenschaften von in diesen Datensätzen dargestellten menschlichen Organen durch eine Beschreibung ihrer Oberfläche charakterisieren. Solche Eigenschaften sind z. B. Lage im menschlichen Körper, Lage relativ zu anderen Organen, Größe, Form und Volumen, aber auch Krümmung der Oberfläche. Zur genauen Bestimmung dieser Werte ist es nötig, eine exakte Darstellung der gesuchten Oberfläche zu finden. In dieser Arbeit wird das Konzept der aktiven Oberflächen verwendet um solch eine Darstellung zu berechnen. Durch die Anwendung vom Benutzer vorgegebener Kräfte wird ein deformierbares Template in einem iterativen Prozeß automatisch einer durch Voxeldaten implizit vorgegebenen Oberfläche angepaßt. Der Deformationsprozeß wird gelenkt durch innere und äußere Energieterme. Die innere Energie ist ein Maß für Dehnungen und Krümmungen und wird benötigt, um die beim Deformationsprozeß auftretenden Verzerrungen des Templates zu minimieren. Die äußere Energie wird aus den Voxeldaten berechnet und sorgt für die Anpassung des Templates an die Meßdaten. Es werden verschiedene Möglichkeiten diskutiert, die Energien und die daraus abgeleiteten Kräfte zu definieren. Ein diskreter Ansatz sowie ein kontinuierlicher Ansatz, der auf kontinuumsmechanischen Theorien beruht, werden vorgestellt. Letzterer wird über die Verwendung von finiten Elementen diskretisiert. Des weiteren wird ein Konzept zur Parallelisierung des Verfahrens auf einem heterogenen, über ein Netzwerk gekoppelten Rechnernetzwerk vorgestellt. Der Algorithmus wurde getestet und eingesetzt zur Berechnung der Abbildung der menschlichen Gehirnoberfläche auf eine sphärische Fläche. Diese Abbildung ist nichtlinear und weist jedem Punkt der Gehirnoberfläche eindeutig einen Punkt auf einer Kugeloberfläche zu. Mittels dieser Transformation können verschiedene Eigenschaften der Gehirnoberfläche auf einer übersichtlichen Karte visualisiert werden.

Construction of Surfaces with Active Contour-Models

Three-dimensional anatomical data are very important for medical diagnostics. It is often possible to characterize important properties of human organs represented in these data by a description of their surface. Examples of such properties are the location of the organs in the human body, location relative to other organs, size, shape and volume, and also the curvature of the surface. For a proper determination of these values an exact representation of the surface must be found. In this thesis, the concepts of active surfaces are used to calculate such a representation. By the application of user-given forces, a deformable template is fitted to a surface in an iterative process. This surface is implicitly given by the voxel-data. The process of deformation is guided by inner and outer energy terms. The inner energy is a measure of the stretching and the curvature of the template and is needed to minimize the distortion of the template in the deformation process. The outer energy is calculated from the measured voxel-data and ensures the fitting of the template to the data. Different possibilities of defining the energies and the derived forces are discussed. A discrete method as well as a method based on the principles of continuum mechanics are presented. The latter is discretized by the finite-element method. Furthermore a concept for the parallelization of the algorithm on a heterogeneous computer-system, coupled by a network, is presented. The algorithm has been tested and applied to the calculation of the mapping from the surface of the human brain onto the surface of a sphere. This transformation is non-linear and uniquely maps every point of the brain-surface to a point of a sphere. By means of this transformation it is possible to visualize several properties of the human brain-surface on a clearly arranged general map.

Inhaltsverzeichnis

Einleitung	1
1 Einführung	5
1.1 Oberflächenmodell	5
1.2 Berechnung des Energieminimums	6
1.2.1 Finite Differenzen	7
2 Diskretes Modell	9
2.1 Diskretes Oberflächennetz	9
2.2 Diskrete Kräfte	12
2.2.1 Diskretes Dehnungsmaß	13
2.2.1.1 Längenerhalt	13
2.2.1.2 Flächenerhalt	13
2.2.2 Diskretes Krümmungsmaß	14
2.2.2.1 Winkelerhalt	14
2.2.3 Bestimmung der Wichtungsfaktoren	16
2.2.3.1 Wichtungsfaktor der längenerhaltenden Kraft	17
2.2.3.2 Wichtungsfaktor der flächenerhaltenden Kraft	17
2.2.3.3 Wichtungsfaktor der winkelerhaltenden Kraft	19
3 Kontinuierliches Modell	21
3.1 Einführung in die finite Elemente Methode	21
3.1.1 Zerlegung in Elemente	23
3.1.2 Abbildung auf ein Referenzelement	23
3.1.3 Verschiebungsansatz	24
3.1.4 Lineares System	25
3.1.5 Nichtlineares System	26
3.2 Kontinuumsmechanische Grundlagen	27
3.3 Variationsprinzip	30
3.4 Verzerrungsmaße am Linienelement	31
3.4.1 Lineare Verzerrung	31
3.4.2 Greensche Verzerrung	31
3.4.3 Almansische Verzerrung	31
3.4.4 Logarithmische Verzerrung	32
3.5 Kräfte am Linienelement	32

3.5.1	Kraft bei linearer Verzerrung	32
3.5.2	Kraft bei Greenscher Verzerrung	32
3.5.3	Kraft bei Almansischer Verzerrung	34
3.5.4	Kraft bei logarithmischer Verzerrung	34
3.6	Verzerrungsmaße im Kontinuum	34
3.6.1	Green-Lagrangescher Deformationstensor	35
3.6.2	Green-Lagrangescher Deformationstensor in Koordinaten der Ver- schiebung u	36
3.6.3	Lineare Näherung	37
3.6.4	Polares Zerlegungstheorem	37
3.7	Realisierung	38
3.7.1	Berechnung der Eigenwerte einer 2×2 Matrix	39
3.7.2	Berechnung der Eigenwerte von $F^{tr} F$	40
3.7.3	Finite Elemente Approximation	40
3.7.4	Verzerrungsenergie	42
3.8	Spezialfälle der Eigenwertberechnung	45
3.8.1	$F^{tr} F$ ist Kugeltensor	45
3.8.2	$F^{tr} F$ ist Diagonalmatrix	47
3.8.3	Winkel der Eigenvektoren ist 45° zur Koordinatenachse	47
3.9	Lokales Koordinatensystem	48
3.10	Alternative Energieterme	51
3.10.1	Flächenerhalt	51
3.10.2	Winkelerhalt	52
3.10.3	Gesamtenergie	52
4	Äußere Kraft	53
4.1	Abstandsbaasierte Kraft	54
4.2	Problematik der abstandsbaasierten Kraft	54
4.3	Druckkraft	57
4.4	Netzüberschneidungen	58
4.5	Einbettung in das finite Elemente Konzept	60
4.6	Minimierung der Energie	60
5	Parallelisierung	63
5.1	Konzepte von pvm	63
5.1.1	Programmierschnittstelle	64
5.2	Parallelisierungsstrategien	65
5.2.1	Parallele Prozeßstrukturen	65
5.2.1.1	Dynamische Prozeßstruktur	65
5.2.1.2	Statische Prozeßstruktur	66
5.2.2	Lastaufteilung	67
5.2.2.1	Datenaufteilung	67
5.2.2.2	Funktionenaufteilung	67
5.3	Realisierung	67
5.3.1	Gebietsaufteilung	68

5.3.2	Prozeßerzeugung	75
5.3.3	Lastbalancierung	75
5.3.4	Datenorganisation	77
5.3.4.1	Organisation des Versendens der Randpunktkoordinaten .	78
5.3.4.2	Organisation der Lastbalancierung	78
5.3.5	Laufzeitmessungen	80
6	Anwendung: 'Cortical Flattening'	83
6.1	Grundlagen der NMR	83
6.1.1	Kodierung der Ortsinformation	86
6.2	Cortical Flattening	88
6.2.1	Vorgehensweise	92
6.2.2	Datenmaterial und Datenvorverarbeitung	94
6.3	Ergebnisse	96
7	Zusammenfassung	103

Einleitung

Obwohl in dieser Arbeit ein sehr allgemeines Verfahren zur Rekonstruktion von in Schichtdatensätzen vorhandenen Oberflächen vorgestellt wird, wurde ihr Inhalt doch maßgeblich beeinflusst von einer bestimmten Anwendung, die aus dem Bereich der medizinischen Forschung, genauer aus dem Bereich der Hirnforschung, stammt. Das medizinische Umfeld, in welchem diese Arbeit angesiedelt ist, soll daher an dieser Stelle geschildert werden.

Die Erforschung des menschlichen Gehirns basiert auf Grundlagen aus vielen Bereichen der Wissenschaft und ist somit eine interdisziplinäre Aufgabe. Dem Bereich der Medizin zugeordnet ist zum einen die Erforschung der Anatomie des menschlichen Gehirns. Neben zellphysiologischen Studien, welche anhand von mikroskopischen Untersuchungen histologischer Gehirnschnitte vorgenommen werden, ist die Beschreibung der Gehirnanatomie aufgrund größerer Strukturen, wie Faltung des Gehirns und innerer Gehirnstrukturen, bedeutsam. Durch medizintechnisch hochmoderne Untersuchungsmethoden ist ein Studium dieser anatomischen Strukturen am lebenden Menschen möglich. Durch sogenannte bildgebende, tomographische Verfahren, wie z. B. die Computertomographie (CT) und die Magnet-Resonanz-Tomographie (MRT, NMR), können Aufnahmen beliebiger Regionen des menschlichen Körpers mit wenig invasiven Methoden, d. h. mit einer geringen Belastung des Probanden, gewonnen werden.

Neben der Anatomie spielt auch die Beschreibung der Funktion des Gehirns eine immer wichtigere Rolle. Ziel ist es hierbei, eine Erklärung der Arbeitsweise des Gehirns zu finden. Aufschluß über die Hirnfunktionen kann durch die sogenannte funktionelle Bildgebung gewonnen werden. Beispiele dieser Verfahren sind die funktionelle Magnet-Resonanz-Tomographie (fNMR), die Positronen-Emissions-Tomographie (PET), die Single-Photonen-Emissions-Computertomographie (SPECT) und auch die Magnetenzephalographie (MEG). Die dabei entstehenden Bilder des Gehirns spiegeln weniger dessen Anatomie wieder, sondern geben vielmehr Aufschluß über Regionen, die bei der Erfüllung einer bestimmten Gehirnleistung aktiv werden. In der Praxis werden Messungen an Probanden durchgeführt, die während eines Versuches eine bestimmte kognitive oder motorische Leistung erbringen müssen. Durch die Verwendung von radioaktiv markierten Substanzen ist bei der PET und der SPECT zudem die Erforschung des Hirnstoffwechsels möglich. Dies ist bedeutsam, da bestimmte Gehirnerkrankungen nicht in der Anatomie, sondern nur durch einen gestörten Stoffwechsel erkennbar sind.

Alle oben genannten Methoden befinden sich in einer stetigen Entwicklung, welche angetrieben wird von neuen Erkenntnissen aus der Physik und deren Umsetzung im technischen Bereich. Um das anfallende Datenmaterial sinnvoll auswerten zu können, muß ein vielfältiges Repertoire an mathematischen Methoden vorhanden sein. Zum einen muß der Meßvorgang mit mathematisch exakt formulierten Gesetzen der Physik so genau wie

möglich beschrieben werden. Dazu sind oft stochastische Methoden anzuwenden, z. B. läßt sich der radioaktive Kernzerfall mit Mitteln aus der Stochastik beschreiben. Aus den Beschreibungen des Abbildungsvorganges ergeben sich die Möglichkeiten der Bildrekonstruktion durch Bildung der (Pseudo-) Inversen der Abbildungsfunktion. Hier ist z. B. die auf der Radon-Transformation ([Rad17]) basierende, gefilterte Rückprojektion zu nennen, aber auch iterative Rekonstruktionsalgorithmen, wie der bei der PET verwendete Maximum-Likelihood-Expectation-Maximization Algorithmus (ML-EM Algorithmus) ([SV82]), fallen darunter. Statistische Methoden helfen auch dabei, die bei Versuchsreihen anfallenden Daten verschiedener Probanden besser auswerten und vereinheitlichen zu können.

Neuerdings werden mathematische Methoden auch verwendet, um ein globales Modell der Gehirnfunktionen zu bilden ([Hor91], [Hor94]). Mit diesem Modell wird versucht, die funktionellen Zusammenhänge, welche zwischen verschiedenen Regionen des Gehirns bestehen, zu erfassen. Es sollen die zwischen funktionellen Einheiten des Gehirns bestehenden logischen Verbindungen rekonstruiert werden.

Auch zur Bearbeitung des anfallenden Bilddatenmaterials bedarf es spezieller Methoden. Darunter fallen die vielfältigen Möglichkeiten der Datennachbearbeitung zum Zwecke der Rauschunterdrückung und der Kontrastverbesserung. Aber auch andere Transformationen sind sinnvoll. Um funktionelle Daten mit anatomischen Daten vergleichen zu können, muß eine räumliche Zuordnung der funktionellen zu den anatomischen Daten des gleichen Probanden gefunden werden ([PCS89]). Auch an Abbildung der Daten unterschiedlicher Probanden wird gearbeitet. Dazu bedient man sich den Gesetzen der Kontinuumsmechanik, um eine elastische ([BLR83], [MCAG93], [CRM94]) oder auf Flüssigkeitsströmungen basierende ([CRM93]) Abbildung einer Gehirnanatomie auf eine andere vorzunehmen und somit eine Art standardisierte Gehirnanatomie zu berechnen.

In dieser Arbeit wird die für Zwecke der Forschung sinnvolle Transformation der stark gefalteten Gehirnoberfläche, in deren sogenannter *grauer Substanz* ein Großteil der an einer Gehirnleistung beteiligten Neuronen sitzen, auf die Oberfläche einer Kugel vorgestellt. Dadurch entsteht eine Karte der Gehirnoberfläche, eine Art Globus, mit dessen Hilfe sich verschiedene Fragestellungen, die sich auf Areale in der Großhirnrinde beziehen, leichter beantworten lassen. Die gesuchte Transformation soll die Gehirnoberfläche so originalgetreu wie möglich auf die Kartenoberfläche abbilden. Daraus ergeben sich gewisse Nebenbedingungen, die bei der Transformation beachtet werden müssen.

Um die gesuchte Transformation zu berechnen, wird das Konzept der aktiven Konturen, genauer gesagt der aktiven Oberflächen, verwendet. Dieses Konzept wird beiden Anforderungen gerecht, nämlich daß zum einen die Berechnung der Transformation der Kartenoberfläche auf die in einem Datensatz implizit vorhandene Gehirnoberfläche möglich ist, und zum anderen lassen sich die Nebenbedingungen leicht berücksichtigen über sogenannte Energieterme, die das Verhalten des Modells beschreiben. Physikalisch entspricht das Auffinden der Transformation der Deformation einer Membran, durch welche die Kartenoberfläche repräsentiert wird. Geleitet wird diese Deformation über innere und äußere Kräfte, die über die bereits oben erwähnten Energieterme definiert werden. Die inneren Kräfte sorgen für die Einhaltung der Nebenbedingungen, und die aus den Bilddaten berechneten äußeren Kräfte sorgen für die Anpassung der Membran an die im Bilddatensatz vorhandene Gehirnoberfläche.

An dieser Stelle soll aber betont werden, daß die Abbildung der Gehirnoberfläche auf eine Karte nur eine mögliche Anwendung für aktive Oberflächenmodelle darstellt. Mit diesen Modellen lassen sich ganz allgemein Oberflächen in medizinischen Datensätzen rekonstruieren. Dabei läßt sich ein Vorwissen über die Geometrie des gesuchten Objektes, wie z. B. Krümmung der Oberfläche oder Objektgröße, über spezielle Energieterme in das Modell aufnehmen. Wird für den Deformationsprozeß eine Ausgangsoberfläche gewählt, welche mit der Form eines Durchschnichts-Objektes initialisiert wurde, so ist eine modellbasierte Objekterkennung möglich. Darauf soll in dieser Arbeit allerdings nicht näher eingegangen werden.

In Kapitel 1 dieser Arbeit wird zunächst das Modell der aktiven Oberflächen vorgestellt, so wie es aus der Literatur bekannt ist. Dadurch wird eine Einführung in die allgemeinen Konzepte dieser Thematik einschließlich der Bedeutung der Energieterme gegeben. Dem in diesem Kapitel vorgestellten Modell liegt eine kontinuierliche Beschreibung des Begriffs der Oberfläche zugrunde. In Kapitel 2 wird eine Abwandlung des Modells vorgestellt, welche auf einem diskreten Oberflächennetz und, darauf basierend, auf unmittelbar im Diskreten definierten Energietermen beruht. Dieses Modell zeichnet sich durch eine hohe Flexibilität und gute Handhabbarkeit der Energieterme aus. Sowohl das in dieser Arbeit verwendete diskrete Oberflächennetz als auch die inneren Kräfte, welche sich aus den diskreten Energietermen ergeben, werden hier vorgestellt.

Kapitel 3 geht wieder zurück in den kontinuierlichen Raum. Hier werden die kontinuumsmechanischen Grundlagen der für die Membrandeformation wichtigen Verzerrungsmaße untersucht. Am Ende dieser Analyse wird die Möglichkeit aufgezeigt, die im Kontinuum definierten Verzerrungsmaße sowohl einem gewünschten Materialverhalten anpassen zu können, als auch die Maße aufzuspalten in die Anteile, durch die Längen-, Flächen- und Winkeländerungen beschrieben werden. Auch werden Sonderfälle bei der Berechnung der Membranverzerrung untersucht. Es wird weiterhin die Realisierung dieses Modells mit Hilfe der finiten Elemente Methode besprochen. Bis hierhin wurden verschiedene Wege aufgezeigt, die Membranverzerrungen zu definieren, woraus immer die Definition der inneren Kraft resultiert. Die zur Anpassung an die im Bilddatensatz vorhandene Oberfläche notwendige äußere Kraft wird in Kapitel 4 vorgestellt. Es werden zwei verschiedene Möglichkeiten diskutiert, eine äußere Kraft aus den Bilddaten zu gewinnen.

Da der Rechenaufwand aufgrund der verwendeten, großen Oberflächennetze sehr hoch ist, wird in Kapitel 5 eine parallele Version des zur Berechnung der Deformation der aktiven Oberfläche verwendeten Algorithmus vorgestellt. Es wird eine Methode der zur Parallelisierung notwendigen Aufteilung des Gesamtnetzes in Teilnetze erläutert, sowie die Prozeßkommunikation beschrieben. Das Verfahren wurde realisiert, so daß Untersuchungen des Speedups wiedergegeben werden.

In Kapitel 6 wird die bereits in dieser Einleitung erwähnte Anwendung, nämlich die Abbildung der Gehirnoberfläche auf eine Kartenoberfläche, zunächst ausführlich beschrieben. Da das zugrundeliegende anatomische Datenmaterial aus einer kernspintomographischen Messung stammt, wird hier auch eine kurze Einführung in die Grundlagen dieses bildgebenden Verfahrens gegeben. An einem Beispiel wird gezeigt, daß die hier vorgestellte Methode in der Lage ist, die gewünschte Transformation zu liefern. Ebenfalls wird gezeigt, daß die Oberfläche eines im Kernspintomographen gemessenen Kopf-Phantoms mit dem in Kapitel 3 vorgestellten finite Elemente Verfahren rekonstruiert werden kann.

te. Kapitel 7 schließlich enthält eine Zusammenfassung mit abschließender Diskussion und Ausblicken.

Kapitel 1

Einführung

In diesem Kapitel wird zunächst das aus der Literatur bekannte Modell der aktiven Oberfläche dargestellt. Auch wird der in der Literatur am häufigsten gewählte Weg der Diskretisierung, nämlich über finite Differenzen ([Coh91], [KWT87], [Ter88], [WS92], [DP95]), kurz vorgestellt werden.

1.1 Oberflächenmodell

Das von Kaas, Witkin und Terzopoulos 1987 vorgestellte Modell der aktiven Kontur [KWT87], welche ein eindimensionales Modell ist, kann in kanonischer Weise zu einem zweidimensionalen Modell für aktive Oberflächen erweitert werden [CC93]. Eine Oberfläche wird definiert mittels einer Abbildung v :

$$\begin{aligned} v : \Omega = [0, 1] \times [0, 1] &\rightarrow \mathbb{R}^3 \\ (r, s) &\rightarrow v(r, s) = (v_1(r, s), v_2(r, s), v_3(r, s)). \end{aligned}$$

Einer so definierten, zweimal stetig ableitbaren Oberfläche v wird folgender Term als innere Energie zugeordnet:

$$E(v) = \int_{\Omega} \omega_{10} \left| \frac{\partial v}{\partial s} \right|^2 + \omega_{01} \left| \frac{\partial v}{\partial r} \right|^2 + 2\omega_{11} \left| \frac{\partial^2 v}{\partial s \partial r} \right|^2 + \omega_{20} \left| \frac{\partial^2 v}{\partial s^2} \right|^2 + \omega_{02} \left| \frac{\partial^2 v}{\partial r^2} \right|^2 dr ds. \quad (1.1)$$

Die erste Ableitung stellt ein Maß für die Dehnung und die zweite Ableitung ein Maß für die Krümmung der Oberfläche v dar. ω_{10} und ω_{01} sind somit Wichtungsfaktoren für die Dehnung, mit ω_{20} und ω_{02} wird der Widerstand der Oberfläche gegen Biegung gewichtet, und schließlich wird mit ω_{11} die Gewichtung der Verwindung der Oberfläche vorgenommen. Der Energieterm kann nur dann Null werden, wenn die Dehnungen $\left| \frac{\partial v}{\partial s} \right|^2$ und $\left| \frac{\partial v}{\partial r} \right|^2$

verschwinden. Diese Ableitungen werden Null, wenn die Koordinaten der Oberfläche sowohl in s - als auch in r -Richtung konstant sind. Wenn dies der Fall ist, dann besteht die Oberfläche nur aus einem Punkt im Raum. Das bedeutet, daß jede Fläche, welche nicht auf einen Raumpunkt zusammengeschrumpft ist, eine innere Energie größer als Null besitzt. Zudem entspricht die in (1.1) definierte Energie nicht der Intuition, falls die Parameter r und s nicht in Bogenlänge angegeben sind. Dann ist es z. B. möglich, daß eine glatte Oberfläche eine Krümmung größer als Null aufweist. So hat zum Beispiel das durch

$$\tilde{v}(r, s) = (r^2, s, 0), \quad (r, s) \in [0, 1] \times [0, 1]$$

gegebene ebene Quadrat $\tilde{v}$ der Seitenlänge 1 eine Krümmung von 2 in Richtung r .

Das durch Gleichung (1.1) definierte Modell weist einige Nachteile auf. Zum einen hat die so definierte Oberfläche, wie bereits oben erwähnt, keine natürliche Ausgangsgröße, sondern schrumpft auf einen Raumpunkt zusammen, wenn die äußere Kraft fehlt. Es ist daher nicht ohne weiteres möglich, eine Ausgangsoberfläche mit einer endlichen Größe vorzugeben und Verzerrungen in Bezug auf diese Ausgangsoberfläche zu messen. Ferner ist nicht genau ersichtlich, wie die in (1.1) definierte Energie modelliert, d. h. speziellen Anforderungen angepaßt werden kann. Die einzige Einflußmöglichkeit besteht in der Vergabe der Wichtungsfaktoren. Für die in Kapitel 6 vorgestellte Anwendung ist insbesondere eine genaue Modellierung des Dehnungsmaßes (weniger der Krümmung) erforderlich. Solche eine Modellierung ist in (1.1) nicht möglich und soll in dieser Arbeit in Kapitel 2 und Kapitel 3 an abgewandelten Modellen entwickelt werden.

1.2 Berechnung des Energieminimums

Die Oberfläche $v(r, s)$ soll eine Form annehmen, welche den Energieterm $E(v)$ minimiert. Eine notwendige Bedingung für ein Minimum ist durch die Euler-Lagrange Gleichung ([BS79]) gegeben:

$$\begin{aligned} -\frac{\partial}{\partial s}(\omega_{10} \frac{\partial v}{\partial s}) - \frac{\partial}{\partial r}(\omega_{01} \frac{\partial v}{\partial r}) + 2 \frac{\partial^2}{\partial s \partial r}(\omega_{11} \frac{\partial^2 v}{\partial s \partial r}) \\ + \frac{\partial^2}{\partial s^2}(\omega_{20} \frac{\partial^2 v}{\partial s^2}) + \frac{\partial^2}{\partial r^2}(\omega_{02} \frac{\partial^2 v}{\partial r^2}) = F(v), \end{aligned} \quad (1.2)$$

wobei $F(v)$ die an der Oberfläche $v(r, s)$ angreifenden äußeren Kräfte sind. Die durch (1.2) gegebene partielle Differentialgleichung kann nicht analytisch, sondern nur numerisch gelöst werden. Dazu muß sie in geeigneter Weise diskretisiert werden. Der Übergang vom Kontinuierlichen zum Diskreten kann durch die Methode der finiten Differenzen ([KWT87]) oder durch Verwendung von finiten Elementen ([CC93]) geschehen. In beiden Fällen ergibt sich ein großes, dünn besetztes lineares Gleichungssystem mit Punktkoordinaten, oder auch, im Falle der finiten Elemente, mit partiellen Ableitungen als Knotenparameter. Auf die finite Elemente Methode wird in Kapitel 3.1 ausführlich eingegangen, die Methode der finiten Differenzen soll nachfolgend kurz dargestellt werden.

1.2.1 Finite Differenzen

Die erste Ableitung einer Funktion $f(x)$ im Punkt x_0 ist definiert durch den Grenzwert des Differenzenquotienten

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Wird der Grenzübergang nicht ausgeführt, bleibt also h finit, so stellt der Differenzenquotient eine Näherung der ersten Ableitung dar. In analoger Weise lassen sich auch höhere und partielle Ableitungen als Grenzwerte von Differenzenquotienten darstellen. Seien nun die Funktionswerte von f nur an diskreten Stützstellen $x_i = i\Delta x$, $i \in \{0, 1, 2, \dots, n\}$ gegeben und sei $f_i := f(x_i)$, so erhält man durch Approximation mit Taylorpolynomen:

$$f_{i+1} = f_i + \Delta x f'_i + \frac{1}{2!} \Delta x^2 f''_i + \frac{1}{3!} \Delta x^3 f'''_i + \dots \quad (1.3)$$

$$f_{i-1} = f_i - \Delta x f'_i + \frac{1}{2!} \Delta x^2 f''_i - \frac{1}{3!} \Delta x^3 f'''_i + \dots \quad (1.4)$$

Aus (1.3) ergibt sich

$$f'_i = \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \quad (\text{Vorwärtsdifferenzenquotient}),$$

aus (1.4) analog

$$f'_i = \frac{f_i - f_{i-1}}{\Delta x} + O(\Delta x) \quad (\text{Rückwärtsdifferenzenquotient}),$$

und (1.3) und (1.4) kombiniert ergeben

$$f'_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \quad (\text{zentraler Differenzenquotient}).$$

Für die in (1.2) definierte Differentialgleichung ergibt sich bei Verwendung von diskreten Funktionswerten $v_{i,j} := v(i\Delta s, j\Delta r)$ und unter der Voraussetzung, daß $\omega_{10}, \omega_{01}, \omega_{11}, \omega_{20}$ und ω_{02} Konstanten sind:

$$\begin{aligned} \frac{\partial^2 v}{\partial s^2} &\approx \frac{1}{\Delta x^2} (v_{i+1,j} - 2v_{i,j} + v_{i-1,j}) \\ \frac{\partial^2 v}{\partial r^2} &\approx \frac{1}{\Delta x^2} (v_{i,j+1} - 2v_{i,j} + v_{i,j-1}) \\ \frac{\partial^4 v}{\partial r \partial s \partial r \partial s} &\approx \frac{1}{\Delta x^4} (v_{i+2,j+2} - 2v_{i+2,j+1} - 2v_{i+1,j+2} + 4v_{i+1,j+1} + v_{i+2,j} \\ &\quad + v_{i,j+2} - 2v_{i+1,j} - 2v_{i,j+1} + v_{i,j}) \\ \frac{\partial^4 v}{\partial s^4} &\approx \frac{1}{\Delta x^4} (v_{i-2,j} + v_{i+2,j} - 4v_{i-1,j} - 4v_{i+1,j} + 6v_{i,j}) \\ \frac{\partial^4 v}{\partial r^4} &\approx \frac{1}{\Delta x^4} (v_{i,j-2} + v_{i,j+2} - 4v_{i,j-1} - 4v_{i,j+1} + 6v_{i,j}). \end{aligned}$$

Obige Gleichungen ergeben zusammen mit den an den Netzknoten angreifenden äußeren Kräften $\vec{f}$ ein dünn besetztes lineares Gleichungssystem der Form $A\vec{v} = \vec{f}$. Die Einträge in der Matrix A setzen sich aus den Wichtungsfaktoren $\omega_{10}, \omega_{01}, \dots$ und den in den obigen finiten Differenzengleichungen vorkommenden Faktoren zusammen. $\vec{v}$ bezeichnet den Vektor der Knotenpunktkoordinaten. Ausgehend von solch einem linearen Gleichungssystem wird der Weg der Energieminimierung ausführlich in Abschnitt 4.6 beschrieben.

Kapitel 2

Diskretes Modell

Anstatt die Oberfläche v als Funktion über kontinuierliche Parameter r und s zu beschreiben und dieses Modell in einem späteren Schritt im Zuge der Berechnung des Energieminimums zu diskretisieren, kann auch ein anderer Weg beschritten werden. Wird die Oberfläche unmittelbar als diskreter Polygonzug definiert, so ist das Problem von Anfang an durch eine endliche Anzahl von Parametern, nämlich den Punktkoordinaten, beschreibbar. Die in Kapitel 1 von der Mathematik klar definierten Begriffe der Dehnung und Krümmung müssen in diesem Modell durch geeignet zu wählende Maße ersetzt werden. Geeignet meint hier, daß wichtige Eigenschaften eines Dehnungs- und Krümmungsmaßes mit in das diskrete Modell übernommen werden. Es lassen sich nicht alle Eigenschaften, die das kontinuierliche Modell aufweist, mittels diskreter Parameter beschreiben. Die diskrete Beschreibung ist immer unvollständig, soll jedoch so genau wie möglich sein und insbesondere der intuitiven Bedeutung der Begriffe „Dehnung“ und „Krümmung“ entsprechen. Da die diskreten Dehnungs- und Krümmungsmaße an den Elementen eines Oberflächennetzes entwickelt werden, wird zunächst das verwendete Oberflächennetz erklärt.

2.1 Diskretes Oberflächennetz

Die Wahl eines geeigneten Oberflächennetzes hängt im wesentlichen von der konkreten Anwendung des Konturmodells ab. Eine Abbildung des Oberflächenmodells auf die in einem Datensatz implizit gegebene, gesuchte Oberfläche ist prinzipiell nur möglich, wenn die beiden Oberflächen bzgl. ihrer Topologie übereinstimmen. In der in Kapitel 6 beschriebenen Anwendung ist die gesuchte Oberfläche die Oberfläche des Gehirns bzw. einer Gehirnhälfte. Im folgenden wird davon ausgegangen, daß die Gehirnoberfläche in sich geschlossen ist und aus einer zusammenhängenden Oberfläche besteht. Genauer gesagt soll sie als topologisch äquivalent zu einer Kugeloberfläche angenommen werden.

Aufgrund dieser Annahme wird das zur Deformation benutzte Ausgangstemplate als diskrete Oberfläche einer Kugel gewählt. Eine Diskretisierung erfolgt durch Zerlegung in Dreieckselemente. Durch die Wahl von auf der Kugeloberfläche liegenden Netzknoten wird das Oberflächennetz definiert. Je mehr Netzknoten vorhanden sind, desto besser kann sowohl die Referenzoberfläche (=Kugel), als auch die Oberfläche des im Volumendatensatz vorliegenden Objektes approximiert werden. Durch die Netzknoten wird das gesuchte Objekt sozusagen abgetastet. Die Abtastung soll an allen Stellen des Objek-

tes ungefähr gleich sein. Aus diesem Grund hat die Verteilung der Netzkpunkte auf dem Ausgangstemplate so gleichmäßig wie möglich zu geschehen.

Um bei der Berechnung Fallunterscheidungen, welche aufgrund der Geometrie und des Zusammenhanges der Dreieckselemente auftreten können, so weit wie möglich zu vermeiden, soll das Oberflächennetz möglichst an allen Stellen eine ähnliche Elementanordnung aufweisen. Dies wird erreicht, indem ein in eine Kugel eingebetteter, gleichmäßiger geometrischer Körper schrittweise verfeinert wird. Als Ausgangskörper wurde der Ikosaeder (Abbildung 2.2) gewählt. Er besteht aus 12 Netzknoten, 30 Kanten und 20 gleichseitigen Dreiecken. Jeder Netzknoten des Ikosaeders hat 5 Nachbarpunkte, mit denen er verbunden ist. Um eine Verfeinerung des Netzes zu erreichen, werden neue Netzknoten auf den Mittelpunkten der Kanten eingefügt. Die neuen und die alten Punkte werden daraufhin neu vernetzt, wie in Abbildung 2.1 (oben) dargestellt. Da eine Kugeloberfläche approximiert werden soll, müssen die neu eingefügten Punkte abschließend auf die Kugeloberfläche projiziert werden. Dies wird in Abbildung 2.1 (unten) verdeutlicht.

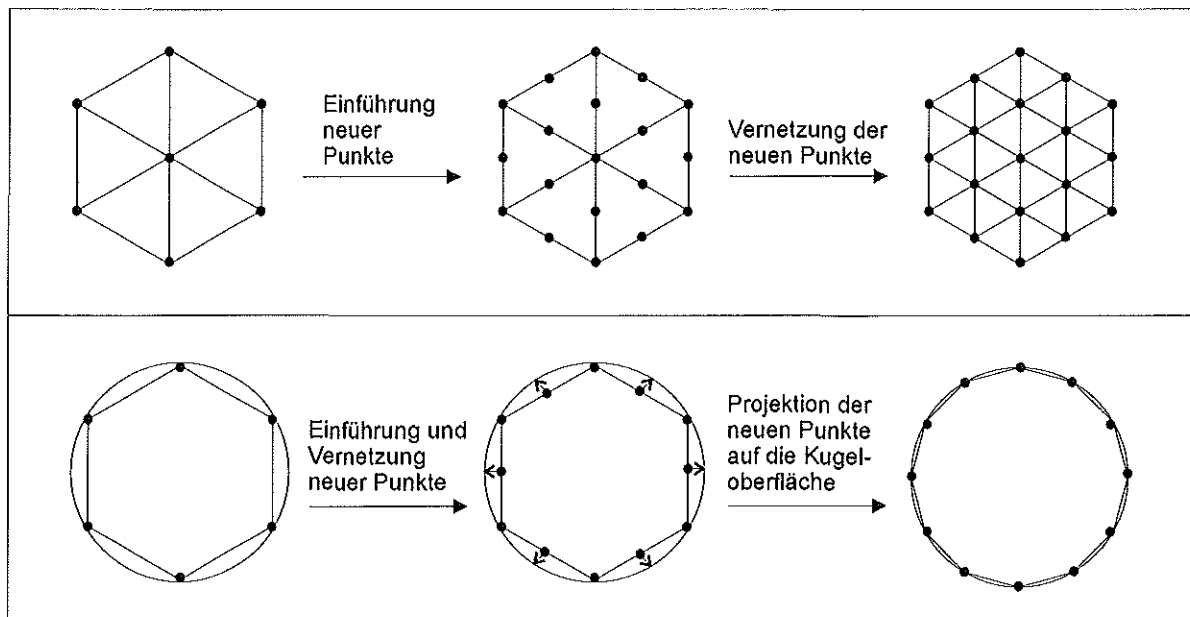


Abbildung 2.1: Das Oberflächennetz wird schrittweise verfeinert, indem neue Punkte eingeführt, diese neu vernetzt und dann auf die Kugeloberfläche projiziert werden. Oben ist eine Aufsicht auf ein Netzsegment dargestellt, unten ein Schnitt durch das in eine Kugel eingebettete Netz.

Der Verfeinerungsprozeß kann beliebig oft durchgeführt werden. Allgemeine Formeln für die Anzahl der Punkte, Kanten und Dreieckselemente im Iterationsschritt n sind gegeben durch

- Punkte: $10 \cdot 4^n + 2$,
- Kanten: $30 \cdot 4^n$ und
- Dreieckselemente: $20 \cdot 4^n$.

Dabei wird für den Ikosaeder als Ausgangsobjekt die Iterationsstufe $n = 0$ angenommen. Die Formeln können leicht durch vollständige Induktion bewiesen werden.

Alle neu eingefügten Punkte haben jeweils 6 Nachbarpunkte, mit denen sie verbunden sind. Die ursprünglichen 12 Punkte des Ikosaeders behalten anzahlmäßig 5 Nachbarpunkte. Somit entsteht ein Netz, welches eine sehr gleichmäßige Punktverbindungsstruktur aufweist. Lediglich die 12 Ursprungspunkte besitzen ein leicht anderes Verbindungsmuster als die später eingefügten Punkte, nämlich 5 Nachbarn im Vergleich zu 6 Nachbarn. Da die Punktzahl mit jedem Verfeinerungsschritt steigt, wird diese Unregelmäßigkeit, bezogen auf das Gesamtnetz, mit jedem Iterationsschritt geringer.

Die oben beschriebene, schrittweise Verfeinerung des Netzes bietet sich auch während des Deformationsprozesses an. Ein zunächst grobes Oberflächennetz kann schnell an ein Objekt angepaßt werden. Die Form des Objektes wird aufgrund des groben Netzes nur schlecht wiedergegeben. Nachdem sich das grobe Netz an das Objekt angepaßt hat, wird es verfeinert, so daß immer feinere Strukturen des gesuchten Objektes wiedergegeben werden. Der Verfeinerungsprozeß während der Berechnung wird analog zur oben erklärten initialen Verfeinerung durchgeführt. Die Projektion der neu eingeführten Punkte auf eine Kugeloberfläche entfällt hier, da das zu verfeinernde Netz schon deformiert ist und eine lokale Anpassung an eine Kugeloberfläche nun keinen Sinn mehr macht.

Eine örtlich angepaßte Verfeinerung des Kugelnetzes ist ebenfalls denkbar. Dies stellt eine Optimierung des Verfeinerungsprozesses im Hinblick auf die Rechenzeit dar. Mit einer solchen Vorgehensweise könnte die Abtastrate der Granularität der Objektoberfläche angepaßt werden. An den Stellen mit einer feinen Struktur würde die Punktdichte im Netz lokal erhöht, um dort eine höhere Abtastrate zu erreichen. Eine beliebig hohe Abtastrate kann auch über die oben beschriebene globale Netzverfeinerung erreicht werden. Dann ergibt sich an Stellen der Objektoberfläche, welche eine grobe Struktur haben, eine Überabtastung. Diese Überabtastung ist zwar für die eigentliche Objekterkennung kein Nachteil, führt aber wegen der überhöhten Anzahl an Netzpunkten zu unnötig hohen Rechenzeiten. Da die Stellen im Oberflächennetz, an denen eine höhere Punktdichte nötig ist, nur schwer und insbesondere wegen der Dynamik der Netzdeformation nicht im vorhinein zu bestimmen sind, wurde in dieser Arbeit auf eine lokale Verfeinerung des Netzes verzichtet. Zudem muß bei einer lokalen Verfeinerung des Netzes beachtet werden, daß die Konformität der Netzelemente nicht verletzt wird. Dies ist an Stellen der Fall, wo zwei Dreieckselemente mit unterschiedlichem Verfeinerungsgrad aneinanderstoßen. An solchen Stellen müßten zusätzliche Nebenbedingungen bezüglich der Position der Gitterpunkten so definiert werden, daß die Konformität erhalten bliebe.

Es soll an dieser Stelle noch einmal betont werden, daß die Wahl des Ausgangsnetzes, hier die Approximation einer Kugel, von der Anwendung abhängt. In der Literatur werden häufig ebene Netze verwendet ([CC93]). Solch ein Netz kann analog zu einem kugelförmigen erzeugt werden, indem eine Ebene in Dreieckselemente aufgeteilt wird. Auch eine iterative Verfeinerung des ebenen Netzes bietet sich an.

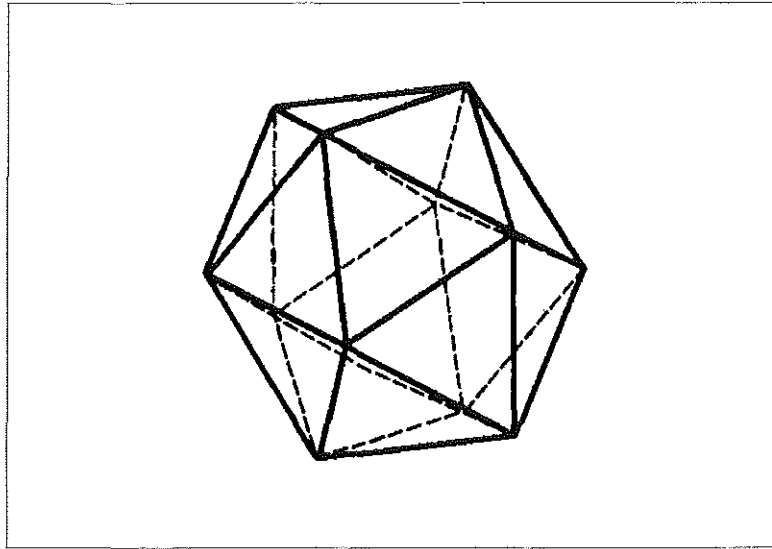


Abbildung 2.2: Ikosaeder.

2.2 Diskrete Kräfte

Im diskreten Modell lassen sich Eigenschaften der Oberfläche, wie z. B. Widerstand gegen Flächenverzerrungen, relativ leicht über Energieterme modellieren.

Da das Oberflächennetz vollständig durch die Koordinaten der Netzkpunkte definiert ist, kann hier auch bei komplizierten Netzgeometrien die Berechnung der inneren Kräfte auf einfache Weise unter ausschließlicher Verwendung dieser Punktkoordinaten erfolgen, ähnlich wie es bei der Benutzung der finiten Differenzen Methode möglich ist. Eine Verwendung von Ableitungen zur Beschreibung der Energieterme, wie es in (1.1) nötig ist, entfällt hier von vornherein. Es ist auch nicht nötig, auf den komplexen Ansatz der finiten Elemente Methode mit deren hohen Berechnungsaufwand zurückzugreifen. Nachfolgend werden diskrete Dehnungs- und Krümmungsmaße entwickelt.

Im folgenden seien

$$v_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad \text{und} \quad v_j = \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix}$$

zwei Punkte im $\mathbb{R}^3$. v_{ij} bezeichne den Differenzvektor aus v_i und v_j , also $v_{ij} = v_i - v_j$. Mit $v_i, v_j, v_k, \dots$ werden im folgenden die Netzkpunkte des diskreten Oberflächennetzes angegeben. v_{ij} steht somit für die durch die Punkte v_i und v_j gegebene Seite eines Dreieckselementes.

2.2.1 Diskretes Dehnungsmaß

Das Maß für die Dehnung eines Dreieckselementes soll hier zum einen durch den Abstand der Netzknoten untereinander und zum anderen durch den Flächeninhalt des Elementes definiert werden. Zwar wird die Geometrie eines Dreiecks schon durch die Angabe der Längen seiner drei Seiten vollständig beschrieben, doch es hat sich als nützlich erwiesen, den Flächeninhalt als eigenständiges Maß einzuführen. Der Flächeninhalt beschreibt ein Dreieck nicht eindeutig und ist daher als alleiniges Verzerrungsmaß ungeeignet. Längen- und flächenerhaltende Kräfte ergänzen sich gegenseitig, da durch eine Aufspaltung des Dehnungsmaßes in diese beiden Anteile eine unterschiedliche Gewichtung dieser Anteile in der Gesamtenergie erfolgen kann.

2.2.1.1 Längenerhalt

Die Kraft, welche für den Erhalt eines möglichst gleichmäßigen Punktabstandes im Netz zuständig ist, wird über folgenden Energieterm definiert:

$$E_d(v_j) = \frac{1}{2} \frac{(d - |v_{ij}|)^2}{d^2}, \quad (2.1)$$

wobei d der momentane Durchschnittsabstand benachbarter Punkte im Netz ist ($d \neq 0$). Durch die Division mit d^2 im obigen Term wird eine Skalierungsinvarianz der Energie E_d erreicht. Als am Punkt v_j angreifende Kraft ergibt sich:

$$F_d(v_j) = -\frac{\partial E_d(v_j)}{\partial v_j} = -\frac{d - |v_{ij}|}{d^2} \frac{v_{ij}}{|v_{ij}|}. \quad (2.2)$$

Anschaulich bedeutet (2.2), daß die längenerhaltende Kraft in Richtung der Verbindungsgeraden der Punkte v_i und v_j wirkt. Der Betrag der Kraft hängt linear von der Abweichung der Länge des Vektors v_{ij} von der Durchschnittslänge d ab. Es handelt sich somit um die Kraft einer Feder der Länge d , welche die beiden Punkte v_i und v_j verbindet. Nach dem Hookschen Gesetz ist die Kraft proportional zur Auslenkung.

2.2.1.2 Flächenerhalt

Der folgende Energieterm mißt die Abweichung der Fläche des von v_i , v_j und v_k aufgespannten Dreieckselementes von einer Durchschnittsfläche a . Die Punkte v_j und v_k sollen als Konstante angenommen werden; somit hängt der Energieterm nur von v_i ab:

$$E_a(v_i) = \frac{\left(a - \frac{|(v_j - v_i) \times (v_k - v_i)|}{2}\right)^2}{2a^2}.$$

Auch hier wird durch die Division durch a^2 eine Skalierungsinvarianz erreicht. Als Kraft ergibt sich (mit Herleitung):

$$F_a(v_i) = -\frac{\partial E_a(v_i)}{\partial v_i}$$

$$\begin{aligned}
&= \frac{a - \frac{|(v_j - v_i) \times (v_k - v_i)|}{2}}{a^2} \frac{1}{2} \frac{((v_j - v_i) \times (v_k - v_i))^{tr}}{|(v_j - v_i) \times (v_k - v_i)|} \\
&\quad \left(\frac{\partial((v_j - v_i) \times (v_k - v_i))}{\partial(v_j - v_i)} \frac{\partial(v_j - v_i)}{\partial v_i} + \frac{\partial((v_j - v_i) \times (v_k - v_i))}{\partial(v_k - v_i)} \frac{\partial(v_k - v_i)}{\partial v_i} \right) \\
&= \frac{a - \frac{|(v_j - v_i) \times (v_k - v_i)|}{2}}{a^2} \frac{1}{2} \frac{((v_j - v_i) \times (v_k - v_i))^{tr}}{|(v_j - v_i) \times (v_k - v_i)|} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \\
&\quad \left[\begin{pmatrix} y_k - y_i & 0 & -z_k + z_i \\ -x_k + x_i & z_k - z_i & 0 \\ 0 & -y_k + y_i & x_k - x_i \end{pmatrix} + \begin{pmatrix} -y_j + y_i & 0 & z_j - z_i \\ x_j - x_i & -z_j + z_i & 0 \\ 0 & y_j - y_i & -x_j + x_i \end{pmatrix} \right]^{tr} \\
&= \frac{a - \frac{|(v_j - v_i) \times (v_k - v_i)|}{2}}{a^2} \frac{1}{2} \frac{((v_j - v_i) \times (v_k - v_i))^{tr}}{|(v_j - v_i) \times (v_k - v_i)|} \\
&\quad \begin{pmatrix} y_j - y_k & x_k - x_j & 0 \\ 0 & z_j - z_k & y_k - y_j \\ z_k - z_j & 0 & x_j - x_k \end{pmatrix}. \tag{2.3}
\end{aligned}$$

2.2.2 Diskretes Krümmungsmaß

Die Kraft, welche für eine möglichst glatte Oberfläche des Netzes sorgt, leitet sich aus nachfolgendem Energieterm her. Die Krümmung der durch die Dreieckselemente gegebenen diskreten Oberfläche wird hier durch den Winkel zwischen Netzknoten definiert, wobei spitze Winkel einer großen Krümmung entsprechen (siehe Abbildung 2.3).

2.2.2.1 Winkelerhalt

Der Cosinus des Winkels w zwischen zwei Vektoren, welche über die Punkte v_i, v_j und v_k definiert sind, berechnet sich durch $\cos(w) = \frac{v_{ji}^{tr} v_{ik}}{|v_{ji}| |v_{ik}|}$. Es bietet sich an, einen Energieterm über den Cosinus des Winkels aufzustellen ([FCC96]), was zu folgender Formel führt:

$$E(v_i) = \frac{1}{2} \left(1 - \frac{v_{ji}^{tr} v_{ik}}{|v_{ji}| |v_{ik}|} \right).$$

Dieser Energieterm erreicht bei einem zwischen den Vektoren v_{ji} und v_{ki} liegenden Winkel von 0° („Spitze“) seinen absoluten Maximalwert von 2 und bei einem Winkel von 180° („Gerade“) seinen absoluten Minimalwert von 0. Zwischen 0° und 180° wächst die Energie streng monoton. Als Kraftvektor ergibt sich:

$$F(v_i) = -\frac{\partial E(v_i)}{\partial v_i} = \frac{(v_{ji} + v_{ki})|v_{ji}||v_{ik}| - (v_{ji}^{tr} v_{ik})(v_{ji} \frac{|v_{ik}|}{|v_{ji}|} + v_{ki} \frac{|v_{ji}|}{|v_{ik}|})}{2v_{ji}^2 v_{ik}^2}.$$

Dieser Kraftterm hat den Nachteil, daß er bei sehr spitzem Winkel zwischen den Vektoren kaum noch zu einer Winkeländerung führt. Der Winkel ändert sich nur wenig bzw. im

Extremfall, daß $v_{ji} = \lambda v_{ik}$, $\lambda > 0$ ist, gar nicht mehr, da der Kraftvektor dann ganz verschwindet:

$$\begin{aligned} F(v_i)(v_{ik} = \lambda v_{ji}) &= \frac{(1 + \lambda)v_{ji}\lambda|v_{ji}|^2 + \lambda|v_{ji}|^2(-\lambda v_{ji} - \frac{v_{ki}}{\lambda})}{2v_{ji}^2 v_{ik}^2} \\ &= \frac{(1 + \lambda)v_{ji}\lambda|v_{ji}|^2 + \lambda|v_{ji}|^2(-\lambda v_{ji} - v_{ji})}{2v_{ji}^2 v_{ik}^2} \\ &= \frac{(1 + \lambda)v_{ji}\lambda|v_{ji}|^2 + \lambda|v_{ji}|^2(-\lambda - 1)}{2v_{ji}^2 v_{ik}^2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Somit ist obiges Maß für große (diskrete) Krümmungen unbrauchbar. Dieses unerwünschte Verhalten kann geändert werden, indem in den Energieterm nicht der Cosinus des Winkels, sondern der Winkel selber eingeht:

$$E_{wi}(v_i) = \frac{1}{2} \arccos \left(\frac{v_{ji}^{tr} v_{ik}}{|v_{ji}||v_{ik}|} \right)^2. \quad (2.4)$$

In diesen Energieterm geht der Winkel zwischen den Vektoren v_{ji} und v_{ik} quadratisch ein, wodurch die Kraft nun linear auf den Winkel einwirkt. Als Kraft ergibt sich (mit $\partial \arccos(y)/\partial y = -1/\sqrt{1-y^2}$ ([JoSch]):)

$$\begin{aligned} F_{wi}(v_i) = -\frac{\partial E(v_i)}{\partial v_i} &= \arccos \left(\frac{v_{ji}^{tr} v_{ik}}{|v_{ji}||v_{ik}|} \right) \frac{1}{\sqrt{1 - \left(\frac{v_{ji}^{tr} v_{ik}}{|v_{ji}||v_{ik}|} \right)^2}} \\ &\quad \frac{(v_j + v_k - 2v_i)|v_{ji}||v_{ik}| - v_{ji}^{tr} v_{ik} \left(\frac{v_{ik}}{|v_{ik}|}|v_{ji}| - \frac{v_{ji}}{|v_{ji}|}|v_{ik}| \right)}{v_{ji}^2 v_{ik}^2}. \end{aligned} \quad (2.5)$$

Damit der Winkel als Maß für die Oberflächenkrümmung genommen werden kann, darf nicht der Winkel zwischen den Punkten innerhalb eines Dreieckselementes, sondern der Winkel zwischen gegenüberliegenden Punkten genommen werden, wie Abbildung 2.3 verdeutlicht.

Die Punkte v_j und v_k liegen in *verschiedenen* Dreieckselementen, welche als gemeinsamen Punkt genau v_i besitzen.

Durch die in (2.4) definierte Energie wird nicht nur die Abweichung der Oberfläche von einer Ebene gemessen. Auch eine ebene Verzerrung der Elemente geht in den Energieterm ein. Soll dies vermieden werden, so kann die durch (2.5) definierte Kraft auf die im Punkt v_i vorliegende Oberflächennormale projiziert werden. Dann geht nur der senkrecht zur Oberfläche liegende Kraftanteil in die Verschiebung der Netzknoten ein.

Falls auch eine Messung der bei der Verzerrung auftretenden Winkeländerungen innerhalb eines Elementes gewünscht ist, kann dies geschehen, indem die Abweichung der drei in einem Element vorliegenden Innenwinkel von jeweils 60° gemessen wird. Da sich der entsprechende Energieterm ganz analog zu (2.4) ergibt, wird an dieser Stelle auf eine ausführliche Darstellung verzichtet. Das daraus resultierende Maß ist eher ein Dehnungs-, als ein Krümmungsmaß, da es auch die Verzerrung innerhalb einer Ebene mißt.

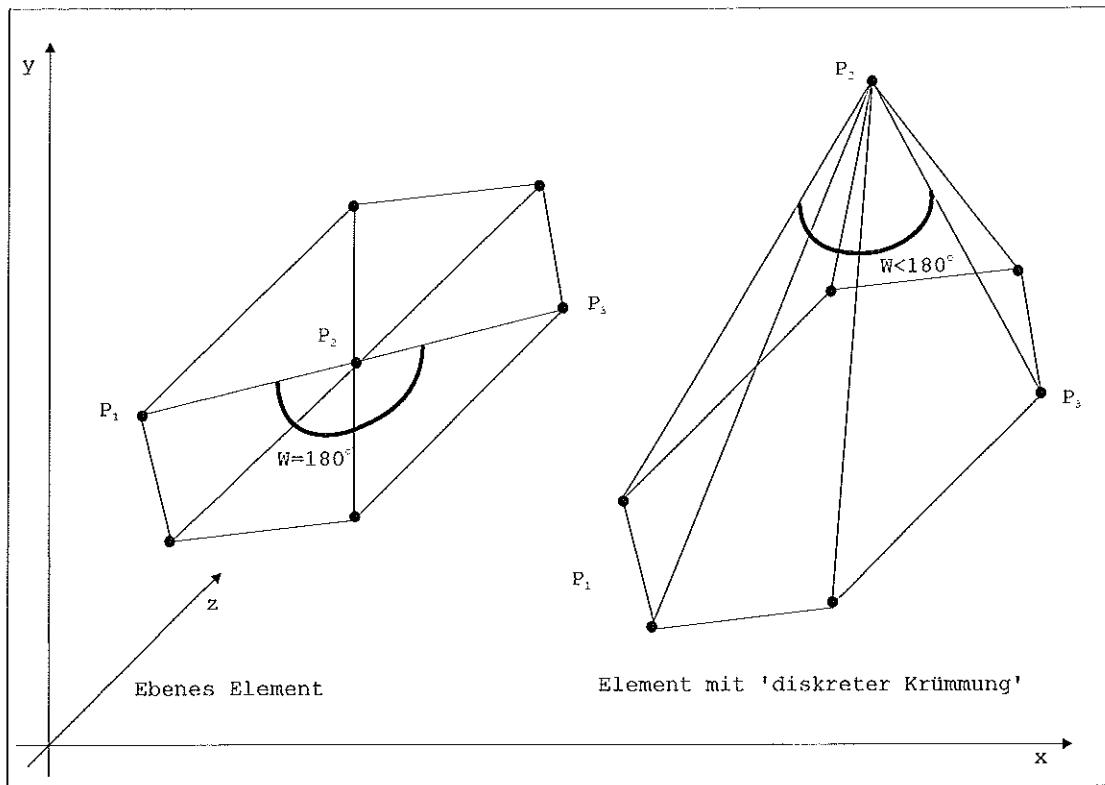


Abbildung 2.3: Eine diskrete Form des Krümmungsmaßes kann über den Winkel zwischen $P_1 - P_2$ und $P_3 - P_2$ definiert werden.

2.2.3 Bestimmung der Wichtungsfaktoren

Als an einem Punkt v_i angreifende Gesamtkraft wird die gewichtete Summe aus den in (2.2), (2.3) und (2.5) definierten Teilkräften genommen:

$$F_{ges}(v_i) = \lambda_1 F_d(v_i) + \lambda_2 F_a(v_i) + \lambda_3 F_{wi}(v_i). \quad (2.6)$$

Bisher wurde noch nichts über die Größe der Wichtungsfaktoren λ_1 , λ_2 und λ_3 gesagt. Im folgenden werden einige Überlegungen dazu angestellt, wie die Wichtungsfaktoren sinnvollerweise gewählt werden sollten. Dazu wird ein Zusammenhang zwischen einer vom Benutzer vorgegebenen, maximal zulässigen Verzerrung (Längen-, Flächen- oder Winkelabweichung von einem Sollwert) und der daraus resultierenden inneren Kraft bestimmt. Wird darüber hinaus vorausgesetzt, daß die an den Netzknoten angreifende äußere Kraft betragsmäßig im Mittel konstant ist, so können sinnvolle Werte für die Wichtungsfaktoren berechnet werden. Eine konstante äußere Kraft wird in der in dieser Arbeit vorgestellten Anwendung in Form einer Druckkraft benutzt. Siehe dazu Kapitel 4. Es wird also folgendes Problem betrachtet: wie groß muß ein Wichtungsfaktor sein, damit die gewichtete innere Kraft bei vorgegebener maximal-zulässiger Verzerrung gerade eine durchschnittliche äußere Kraft ausgleicht?

2.2.3.1 Wichtungsfaktor der längenerhaltenden Kraft

Betrachtet man ein zum Vektor v_{ij} feststehendes Koordinatensystem, so vereinfacht sich (2.1). Wird mit $l = |v_i - v_j|$ der Abstand zwischen Punkt v_i und v_j bezeichnet, so ergibt sich folgender Energieterm:

$$E_d(l) = \frac{1}{2} \frac{(d-l)^2}{d^2},$$

welcher nur noch von der einen Variablen l abhängt. Dieser Energieterm wird nun verwendet, um den Zusammenhang zwischen dem Betrag der Kraft $F_d(l)$ und dem Punktabstand l herzustellen. Es gilt:

$$|F_d(l)| = \left| -\frac{\partial E_d(l)}{\partial l} \right| = \left| \frac{l-d}{d^2} \right| = \begin{cases} \frac{l-d}{d^2}, & l > d \\ \frac{d-l}{d^2}, & d > l \end{cases}.$$

Die Kraft $F_d(l)$ hängt also linear vom Punktabstand l ab. Soll die innere Kraft eine äußere Kraft, die den Betrag c hat, ausgleichen, so folgt:

$$\begin{aligned} \lambda_1 \left| \frac{l-d}{d^2} \right| &= c \\ \Leftrightarrow \lambda_1 &= c \left| \frac{d^2}{l-d} \right|, \text{ für } l \neq d. \end{aligned}$$

Sei z. B. $c = 1.0$ gewählt, so ergibt sich aus $l = d/2$ folgendes λ'_1 :

$$\Leftrightarrow \lambda'_1 = \frac{2d^2}{d} = 2d.$$

2.2.3.2 Wichtungsfaktor der flächenerhaltenden Kraft

Im folgenden soll aus Gründen der Anschaulichkeit ein zweidimensionales Dreieckselement mit zweidimensionalen Punktkoordinaten betrachtet werden. Man stelle sich dazu die Punktkoordinaten dargestellt in einem zum Element lokalen zweidimensionalen Koordinatensystem vor. Es seien also nun

$$v_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad v_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix} \quad \text{und} \quad v_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$$

drei Punkte im $\mathbb{R}^2$. Das Vektorprodukt $v_{ji} \times v_{ki}$ vereinfacht sich damit zu

$$v_{ji} \times v_{ki} = \begin{pmatrix} (x_j - x_i)(y_k - y_i) - (y_j - y_i)(x_k - x_i) \\ (y_j - y_i)(x_k - x_i) - (x_j - x_i)(y_k - y_i) \end{pmatrix} =: \begin{pmatrix} k \\ -k \end{pmatrix}.$$

Wegen

$$|v_{ji} \times v_{ki}| = \left| \begin{pmatrix} k \\ -k \end{pmatrix} \right| = \sqrt{2}|k|$$

vereinfacht sich die Kraft (2.3) zu

$$F_a(v_i) = \frac{1}{\sqrt{2}} \left(\frac{a - \frac{|k|}{\sqrt{2}}}{a^2} \right) \frac{k}{|k|} \begin{pmatrix} y_j - y_k \\ x_j - x_k \end{pmatrix}.$$

Als Betrag der Kraft ergibt sich somit unter Beachtung, daß $|\frac{k}{|k|}| = 1$:

$$|F_a(v_i)| = \frac{1}{\sqrt{2}} \left| \left(\frac{a - \frac{|k|}{\sqrt{2}}}{a^2} \right) \right| \left| \begin{pmatrix} y_j - y_k \\ x_j - x_k \end{pmatrix} \right|.$$

In diesem zweidimensionalen Modell ist sehr schön zu sehen, in welche Richtung die flächenerhaltende Kraft auf den Punkt v_i wirkt. Sie wirkt in Richtung des Vektors $((y_j - y_k) \quad (x_j - x_k))^tr$. Dieser Vektor steht gerade senkrecht auf der Dreiecksseite, die dem Punkt v_i gegenüberliegt. Dies ist auch zu erwarten, da Kraftanteile, welche parallel zur gegenüberliegenden Dreiecksseite liegen, keine Änderung des Flächeninhaltes bewirken. Die Größe der Kraft hängt somit sowohl von der Differenz aus Soll- und Ist-Fläche $\Delta a = |(a - \frac{k}{\sqrt{2}})/a^2|$, aber auch von der Geometrie des Elementes selber ab.

Ein auflösbarer Zusammenhang zwischen Betrag der Kraft und Flächenverzerrung Δa kann nicht für jede beliebige Geometrie aufgestellt werden. Deshalb wird im folgenden als Referenzgeometrie ein gleichschenkliges Dreieck verwendet. Dies ist insbesondere sinnvoll, da durch die Definition der längenerhaltenden Kraft gleichschenklige Dreiecke bevorzugt werden. Es wird nun ein gleichschenkliges Dreieckselement mit Seitenlänge l betrachtet. Dazu werden folgende spezielle Punktkoordinaten gewählt:

$$v_1 = \begin{pmatrix} l/2 \\ \frac{l}{2}\sqrt{3} \end{pmatrix}, v_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{und} \quad v_3 = \begin{pmatrix} l \\ 0 \end{pmatrix}.$$

Daraus ergibt sich für die resultierende Kraft:

$$\begin{pmatrix} k \\ -k \end{pmatrix} = \begin{pmatrix} -l/2 \\ -\frac{l}{2}\sqrt{3} \end{pmatrix} \times \begin{pmatrix} l/2 \\ -\frac{l}{2}\sqrt{3} \end{pmatrix} = \begin{pmatrix} \frac{l^2}{2}\sqrt{3} \\ -\frac{l^2}{2}\sqrt{3} \end{pmatrix},$$

$$b := \left| \begin{pmatrix} \frac{l^2}{2}\sqrt{3} \\ -\frac{l^2}{2}\sqrt{3} \end{pmatrix} \right| = \sqrt{\frac{3}{2}} l^2 \quad \text{und damit}$$

$$F_a(v_1) = \begin{pmatrix} 0 \\ \frac{a-b'/2}{2a^2b'}(k'_0 l - k'_1 l) \end{pmatrix}.$$

Soll der Betrag der flächenerhaltenden Kraft $F_a(v_1) = \begin{pmatrix} F_a^X(v_1) \\ F_a^Y(v_1) \end{pmatrix}$ für ein gleichschenkliges Dreieck der Fläche F gleich einer gegebenen Größe c sein, so muß für den Wichtungsfaktor λ_2 gelten:

$$\lambda_2 = \frac{2ca^2b'}{(a - b'/2)(k'_0 l - k'_1 l)}, \quad \text{wobei}$$

$$l = 2\sqrt{\frac{F}{\sqrt{3}}} \text{ ist.}$$

2.2.3.3 Wichtungsfaktor der winkelerhaltenden Kraft

Auch hier soll wieder ein zweidimensionales Element betrachtet werden. Bei einer vorgegebenen Seitenlänge $|v_{ji}| = |v_{ik}| = l$ und gegebenem Winkel ω sei die Geometrie des Elementes gegeben durch $v_{ji} = (l \cos(\omega/2), l \sin(\omega/2))^tr$ und $v_{ki} = (l \cos(\omega/2), -l \sin(\omega/2))^tr$. Als Kraft erhält man in diesem speziellen Fall:

$$\begin{aligned}
 F_{wi}(v_i) &= \omega \frac{1}{\sqrt{1 - \cos(\omega)^2}} \frac{(v_{ji} - v_{ik})l^2 - l^2(\cos(\omega/2)^2 - \sin(\omega/2)^2)(v_{ik} - v_{ji})}{l^4} \\
 &= \omega \frac{1}{\sqrt{1 - \cos(\omega)^2}} \frac{(v_{ji} + v_{ki})2 \cos(\omega/2)^2}{l^2} \\
 &= \omega \frac{1}{\sqrt{1 - \cos(\omega)^2}} \frac{2 \cos(\omega/2)^2 \begin{pmatrix} 2l \cos(\omega/2) \\ 0 \end{pmatrix}}{l^2}.
 \end{aligned}$$

Soll der Betrag der gewichteten Kraft gleich einer Konstanten c sein, so ergibt sich als Wichtungsfaktor λ_3 :

$$\lambda_3 = \frac{cl\sqrt{1 - \cos(\omega)^2}}{4\omega \cos(\omega/2)^3}.$$

Es bietet sich an, als Länge l den durchschnittlichen Punktabstand im Oberflächennetz zu wählen. Somit ist garantiert, daß im vorhandenen Netz die winkelerhaltende Kraft eine äußere Kraft mit Betrag c bei vorgegebenem Winkel ω gerade ausgleicht.

Kapitel 3

Kontinuierliches Modell

Obwohl mit dem in Kapitel 2 vorgestellten, unmittelbar diskret definierten Modell sehr gute Ergebnisse erzielt werden können, wie in Kapitel 6 zu sehen ist, wird in diesem Abschnitt eine Möglichkeit aufgezeigt, ein kontinuierliches Modell zu verwenden, in welchem das Maß für die Dehnung vom Benutzer weitgehend modellierbar wird. Darunter soll hier die Möglichkeit verstanden werden, das Dehnungsmaß aufzuspalten in separate Maße für Längen-, Flächen- und Winkeländerung. Jedes dieser Maße kann, je nach Bedarf, unterschiedlich gewichtet werden. Auf die genaue Untersuchung des Krümmungsmaßes soll hier nicht weiter eingegangen werden, da dieses Maß insbesondere für die in Kapitel 6 vorgestellte Anwendung eine untergeordnete Rolle spielt.

3.1 Einführung in die finite Elemente Methode

Die Gesetzmäßigkeiten physikalischer Systeme können in Form von partiellen Differentialgleichungen bzgl. einer das System beschreibenden Funktion u gegeben sein. Alternativ dazu kann eine Systembeschreibung auch mittels eines Energiepotentials erfolgen, welches gegeben ist durch ein Integral über einen Differentialoperator $L(u)$:

$$\Theta = \int_V L(u) dv. \quad (3.1)$$

Bei einer konkreten Problemstellung ist es die Aufgabe, die unbekannte, nur implizit beschriebene Funktion u aus den Gleichungen, die das System beschreiben, zu ermitteln. Eine Lösung dieser Gleichungen kann oft nicht analytisch erfolgen. In diesen Fällen muß nach einer numerischen Approximation gesucht werden. In den folgenden Ausführungen wird davon ausgegangen, daß die Integralform (3.1) der Problembeschreibung vorliegt. Eine Systembeschreibung durch Differentialgleichungen kann durch die Methode der gewichteten Residuen auf Integralform gebracht werden (siehe z. B. [DT85]).

Bei der Methode der finiten Elemente wird das Gebiet, in welchem die Funktion u gesucht wird, in diskrete, genau definierte Teilgebiete aufgeteilt. Solch ein Teilgebiet wird (finites) Element genannt. Auf jedem Teilgebiet wird die gesuchte Funktion durch eine Summe von endlich vielen, gewichteten Formfunktionen approximiert. Die Gewichte dieser Funktionen werden auch Knotenparameter genannt. Nach dieser Diskretisierung hängt die Systembeschreibung nicht mehr von der Funktion u , sondern von endlichen vielen Parametern

$u_i, i = 1, \dots, n$ ab. Da das Extremum des Energiepotentials gesucht wird (normalerweise das Minimum), wird das Potential nach den Parametern u_i differenziert. Das Integral kann dann ausgewertet werden, da es nur von den bekannten Formfunktionen abhängt. Diese Auswertung geschieht bei einfachen Formfunktionen analytisch, bei komplexeren Funktionen numerisch, z. B. durch Gauss-Quadratur.

Hängt das Energieintegral quadratisch von den Knotenparametern u_i ab, so ergibt sich ein lineares Gleichungssystem mit n Gleichungen, welches durch geeignete Methoden gelöst werden kann.

Hängt das Potential in höherer Ordnung von den u_i ab, so ist das entstehende Gleichungssystem nichtlinear. In diesem Fall kann eine Lösung der Gleichungen näherungsweise bestimmt werden. Dazu wird das System unter Verwendung der zweiten Ableitungen nach den u_i zunächst durch ein lineares Gleichungssystem angenähert. Diese Näherung wird benutzt, um das Ausgangssystem in einem iterativen Verfahren, z. B. mit der modifizierten Newton Methode (siehe [Cri91], Seiten 6–8), zu lösen.

Im folgenden sollen einige wichtige Punkte der finiten Elemente Methode näher beleuchtet werden. Für eine vollständige, ausführliche Beschreibung sei z. B. auf [Bat90], [DT85] oder [Cri91] verwiesen.

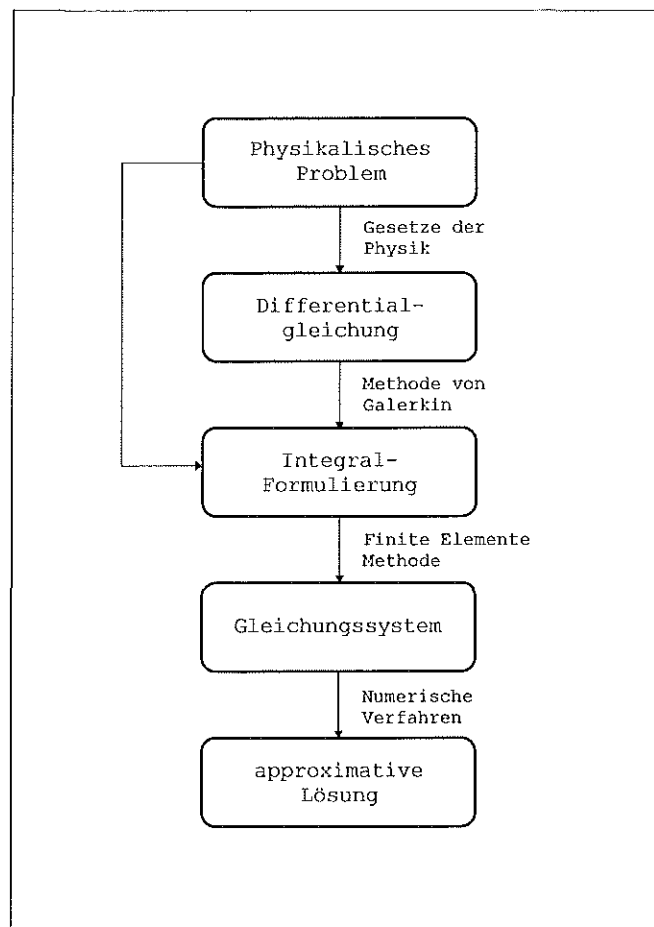


Abbildung 3.1: Der Weg von einem physikalischen Problem über die finite Elemente Methode zur Lösung.

3.1.1 Zerlegung in Elemente

Das dem Potential (3.1) zugrundeliegende Grundgebiet V wird im ersten Schritt der finite Elemente Methode in einzelne Elemente zerlegt. Dies kann im zweidimensionalen Fall z. B. durch eine Zerlegung in Dreiecke, Vierecke oder Parallelogramme geschehen. Um numerische Probleme zu vermeiden, sollten keine Elemente mit allzu stumpfen Winkeln gebildet werden ([Sch84]). Die für die hier verwendete Oberfläche bereits im Abschnitt 2.1 vorgestellte Diskretisierung in Form von Dreieckselementen soll auch bei der Anwendung der finiten Elemente Methode weiterverwendet werden.

3.1.2 Abbildung auf ein Referenzelement

Um eine effiziente Berechnung des Energieintegrals über die Formfunktionen durchführen zu können, wird die spezielle Geometrie eines Elementes E auf ein Standardelement S mittels einer Koordinatentransformation τ abgebildet. τ bildet das Element E , welches durch x - und y -Koordinaten gegeben sein soll, auf ein Referenzelement ab, das durch (dimensionslose) Koordinaten ξ und ν definiert ist. Da in dieser Arbeit ausschließlich Dreieckselemente verwendet werden, soll dieses Prinzip auch anhand eines Dreieckselementes erklärt werden.

In τ gehen ein Satz von drei linearen Funktionen¹ sowie die Punktkoordinaten der Knotenpunkte von E ein. Die Transformation wird also von der Geometrie von E bestimmt. Es ergeben sich auf diese Weise verschiedene Vorteile:

- Die Formfunktionen N_i brauchen nur auf dem Referenzelement definiert zu sein. Eine Anpassung der Formfunktionen an eine spezielle Elementgeometrie ist nicht nötig.
- Für jedes Netzelement ist das Integral über der gleichen (Referenz-) Fläche auszuwerten, was die Berechnung stark vereinheitlicht.

Der im Integral (3.1) enthaltene Differentialoperator L bezieht sich auf die ursprünglichen Koordinaten x und y . Nach Transformation auf das Referenzelement soll sich (3.1) auf die über ξ und ν definierten Formfunktionen beziehen. Daher sind alle Ableitungen nach x und y durch Ableitungen nach ξ und ν zu ersetzen. Der Zusammenhang der ersten Ableitungen ergibt sich durch

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \nu} \end{pmatrix} = J \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix},$$

mit Jacobimatrix

$$J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \nu} & \frac{\partial y}{\partial \nu} \end{pmatrix} =: \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix}.$$

¹Werden Funktionen höherer Ordnung verwendet, so können auch krummlinige Elemente definiert werden.

Zu beachten ist, daß J über die Transformation $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x(\xi, \nu) \\ y(\xi, \nu) \end{pmatrix}$ explizit gegeben ist. Die eigentlich benötigte Umkehrung $\begin{pmatrix} \xi \\ \nu \end{pmatrix} = \begin{pmatrix} \xi(x, y) \\ \nu(x, y) \end{pmatrix}$ liegt somit nur implizit vor, eine explizite Formulierung ist umständlich zu finden. Wegen

$$\begin{pmatrix} \frac{\partial x}{\partial y} \end{pmatrix} = \frac{1}{|J|} \begin{pmatrix} j_{22} & -j_{12} \\ -j_{21} & j_{11} \end{pmatrix} \begin{pmatrix} \frac{\partial \xi}{\partial \nu} \end{pmatrix}$$

braucht die Umkehrung jedoch nicht explizit formuliert zu werden. Höhere Ableitungen können auf ähnliche Weise umformuliert werden. Eine explizite Herleitung für zweite Ableitungen kann in [DT85], Seite 46–47 gefunden werden. Nach der Transformation auf ein Einheitsselement S kann das Integral (3.1) somit umgeschrieben werden zu:

$$\Theta(u) = \int_{V_E} L_{(x,y)}(u) dx dy = \int_{V_S} L_{(\xi,\nu)} |J| d\xi d\nu. \quad (3.2)$$

$L_{(x,y)}$ meint hier, daß sich der Differentialoperator L auf Ableitungen nach x und y bezieht. Entsprechendes gilt für $L_{(\xi,\nu)}$.

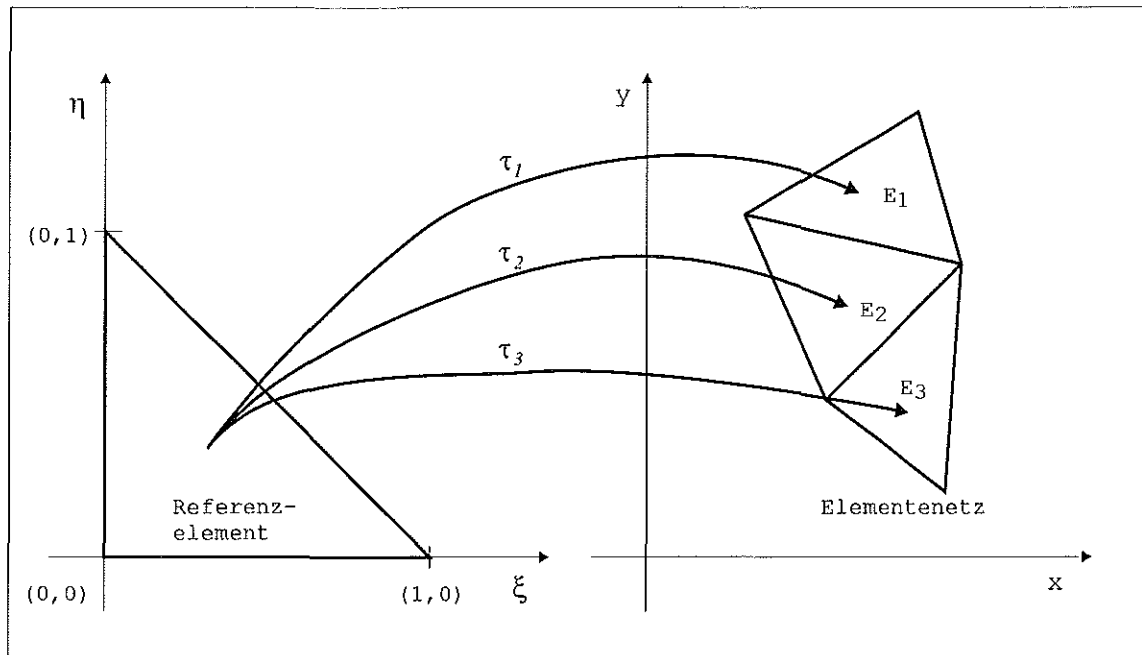


Abbildung 3.2: Das Referenzelement S wird über verschiedene Transformationen τ_1, τ_2 und τ_3 auf die Netzelemente E_1, E_2 und E_3 abgebildet.

3.1.3 Verschiebungsansatz

Gleichung 3.2 ist immer noch infinit, in dem Sinne, daß Θ von der Funktion u abhängt. Das Integral (3.1) wurde lediglich umgeschrieben auf ein Standardelement S mit Koordinatensystem ξ und ν . In diesem Schritt werden die Formfunktionen N_i eingeführt.

Diese Formfunktionen müssen gewissen Stetigkeitsbedingungen genügen. Die Stetigkeitsbedingungen lassen sich mathematisch begründen und gehen auch aus den physikalischen Gegebenheiten hervor. Bedeutet u z. B. die Verschiebung eines kontinuierlichen Körpers, so muß u stetig sein, damit die Kontinuität des Materials gewährleistet ist. Geht in u auch die Biegung des Materials ein, so müssen sogar die ersten partiellen Ableitungen stetige Funktionen sein.

Ansatzfunktionen, welche die erforderlichen Eigenschaften besitzen, werden *konform* genannt. Im folgenden wird vorausgesetzt, daß alle verwendeten Formfunktionen konform sind.

Häufig werden Polynome als Formfunktionen verwendet. Der Funktionsverlauf innerhalb eines Elementes läßt sich nun mit Hilfe von Funktionswerten und evtl. von partiellen Ableitungen an bestimmten Stützstellen innerhalb des Elementes darstellen. Diese Stützstellen werden *Knotenpunkte* genannt, die Werte der Funktion und die Ableitungswerte an diesen Stellen sind die *Knotenvariablen* oder auch *Knotenparameter*.

u läßt sich somit approximativ darstellen als Linearkombination aus Knotenparametern u_i und Formfunktionen $N_i(\xi, \nu)$:

$$u(\xi, \nu) \approx \sum_{i=1}^n u_i N_i(\xi, \nu) = (u_1 u_2 \dots u_n) \begin{pmatrix} N_1(\xi, \nu) \\ N_2(\xi, \nu) \\ \vdots \\ N_n(\xi, \nu) \end{pmatrix}. \quad (3.3)$$

Nach Einsetzen von (3.3) in (3.2) ergibt sich ein von endlich vielen Parametern u_i abhängendes Potential. Ein Extremum dieses Potentials kann über Differenzieren nach den u_i bzw. über die Variation des Potentials bestimmt werden.

3.1.4 Lineares System

Ist L ein linearer Differentialoperator, d. h. kommen die Funktion u und deren Ableitungen im Term $L(u)$ nur in erster Potenz vor, und hat das Potential die Form $\Theta = \int_{V_E} \frac{1}{2} L(u)^2 dV$, so kann die Variation von Θ nach Diskretisierung von u als lineares Gleichungssystem geschrieben werden.

L^2 hat in diesem Fall die Form

$$L^2 = \left(u(x, y) \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} \frac{\partial^2 u}{\partial x \partial y} \dots \right) D \begin{pmatrix} u(x, y) \\ \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial^2 u}{\partial x \partial y} \\ \vdots \end{pmatrix},$$

wobei D eine $n \times n$ Matrix ist, welche durch Parameter bestimmt wird, welche Eigenschaften des Materials beschreiben. D wird auch Materialmatrix genannt. Mit diesem Ansatz

ergibt sich für das Potential:

$$\begin{aligned}
 \Theta &= \frac{1}{2} \int_{V_E} \left(u(x, y), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x \partial y}, \dots \right) D \left(u(x, y), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x \partial y}, \dots \right)^{tr} dx dy \\
 &= \frac{|J|}{2} \int_{V_S} \left(u(\xi, \nu), \frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \nu}, \frac{\partial^2 u}{\partial \xi \partial \nu}, \dots \right) T^{tr} D T \left(u(\xi, \nu), \frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \nu}, \frac{\partial^2 u}{\partial \xi \partial \nu}, \dots \right)^{tr} d\xi d\nu \\
 &= \frac{|J|}{2} \int_{V_S} (u_1, u_2, \dots, u_n) B^{tr} T^{tr} D T B (u_1, u_2, \dots, u_n)^{tr} d\xi d\nu,
 \end{aligned} \tag{3.4}$$

mit

$$B = \begin{pmatrix} N_1(\xi, \nu) & \dots & N_n(\xi, \nu) \\ \frac{\partial N_1(\xi, \nu)}{\partial \xi} & \dots & \frac{\partial N_n(\xi, \nu)}{\partial \xi} \\ \vdots & & \vdots \end{pmatrix}.$$

Die Matrix T beschreibt den Übergang vom finiten Element auf das Standardelement E . Mit der Matrix B , welche die Formfunktionen $N_i(\xi, \nu)$ und deren Ableitungen enthält, und dem Parametervektor $(u_1, \dots, u_n)$ wird die Diskretisierung vorgenommen.

Die Matrix

$$K = \frac{|J|}{2} \int_{V_S} B^{tr} T^{tr} D T B d\xi d\nu$$

wird *Elementsteifigkeitsmatrix* genannt. Mit den Gesetzen der Variationsrechnung (siehe Abschnitt 3.3) gilt

$$\delta\Theta = (\delta u_1, \dots, \delta u_n) K \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}.$$

Das Minimum der Energie wird unter Einbeziehung der verallgemeinerten Kräfte $(f_1, \dots, f_n)$ gefunden durch das Lösen des linearen Gleichungssystems

$$K \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}.$$

Die Inverse der Abbildung K kann entweder durch Berechnung der Inversen Matrix K^{-1} oder durch Anwendung eines iterativen Verfahrens, wie z. B. der konjugierten Gradienten Methode ([EJ93]), gewonnen werden.

3.1.5 Nichtlineares System

Der im obigen Abschnitt vorgestellte Fall des linearen Systems ist ein sehr einfaches Modell für ein physikalisches System. Häufig, so auch in dem hier behandelten Problem, ist der Zusammenhang zwischen Knotenparametern und innerer Kraft nicht linear. Dann ergibt

sich ein System nicht linearer Gleichungen G_i , $i = 1, \dots, n$, welches die Knotenparameter mit den inneren Kräften verknüpft:

$$G(u_1, \dots, u_n) = \begin{pmatrix} G_1(u_1, \dots, u_n) \\ G_2(u_1, \dots, u_n) \\ \vdots \\ G_n(u_1, \dots, u_n) \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}. \quad (3.5)$$

Die f_i , $i = 1, \dots, n$, sind, wie oben, die Komponenten der äußeren Kraft. Die Invertierung dieses Systems ist normalerweise analytisch nicht möglich. Es bieten sich aber numerische Methoden, wie z. B. das *modifizierte Newton-Verfahren*, an ([Cri91]). Dazu wird das Gleichungssystem (3.5) durch Bildung der sogenannten *Tangentensteifigkeitsmatrix* K_T um einen Wert $(\tilde{u}_1, \dots, \tilde{u}_n)$ herum linearisiert:

$$G(u_1, \dots, u_n) \approx G(\tilde{u}_1, \dots, \tilde{u}_n) + K_T((u_1, \dots, u_n) - (\tilde{u}_1, \dots, \tilde{u}_n))$$

mit

$$K_T(u_1, \dots, u_n) = \begin{pmatrix} \frac{\partial G_1}{\partial u_1} & \cdots & \frac{\partial G_1}{\partial u_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial G_n}{\partial u_1} & \cdots & \frac{\partial G_n}{\partial u_n} \end{pmatrix}.$$

Zu beachten ist, daß die Tangentensteifigkeitsmatrix K_T im allgemeinen von den Parametern $u_1, \dots, u_n$ abhängt. Die Inverse des Systems (3.5) kann in einem iterativen Prozeß gefunden werden. t bezeichne hier die Iterationsstufe:

$$\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}_{t+1} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}_t + K_T^{-1}((u_1, \dots, u_n)_0) \left(\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} - \begin{pmatrix} G_1((u_1, \dots, u_n)_t) \\ \vdots \\ G_n((u_1, \dots, u_n)_t) \end{pmatrix} \right).$$

In der obigen Darstellung wird während des gesamten iterativen Prozesses die Tangentensteifigkeitsmatrix K_T verwendet, in welche die Startparameter $(u_1, \dots, u_n)_0$ eingehen. Dies erspart ein erneutes Berechnen (und Invertieren) der Matrix während des Iterationsprozesses. Ein Neuberechnen der Matrix K_T während des Iterierens ist auch möglich. Das Verfahren ist in Abbildung 3.3 für den eindimensionalen Fall dargestellt.

3.2 Kontinuumsmechanische Grundlagen

In diesem Kapitel werden die kontinuumsmechanischen Grundlagen bereitgestellt, die zur Beschreibung der Bewegungen und Deformationen eines elastischen Körpers unter Einwirkung einer äußeren Kraft nötig sind. 1987 stellten Terzopoulos, Platt, Barr und Fleischer ein Modell vor ([TPBF87]), welches auf der Elastizitätstheorie basiert. Ein elastischer Körper Ω , gegeben durch die Parametrisierung $x(u, t) = (x_1(u, t), x_2(u, t), x_3(u, t))^2$, verhält sich gemäß folgender Differentialgleichung:

$$\frac{\partial}{\partial t} \left(\rho \frac{\partial x}{\partial t} \right) + \theta \frac{\partial x}{\partial t} + \frac{\partial \varepsilon(x)}{\partial x} = f(x, t). \quad (3.6)$$

² $u = (u_1, u_2, u_3)$ für einen Körper, $u = (u_1, u_2)$ für eine Oberfläche.

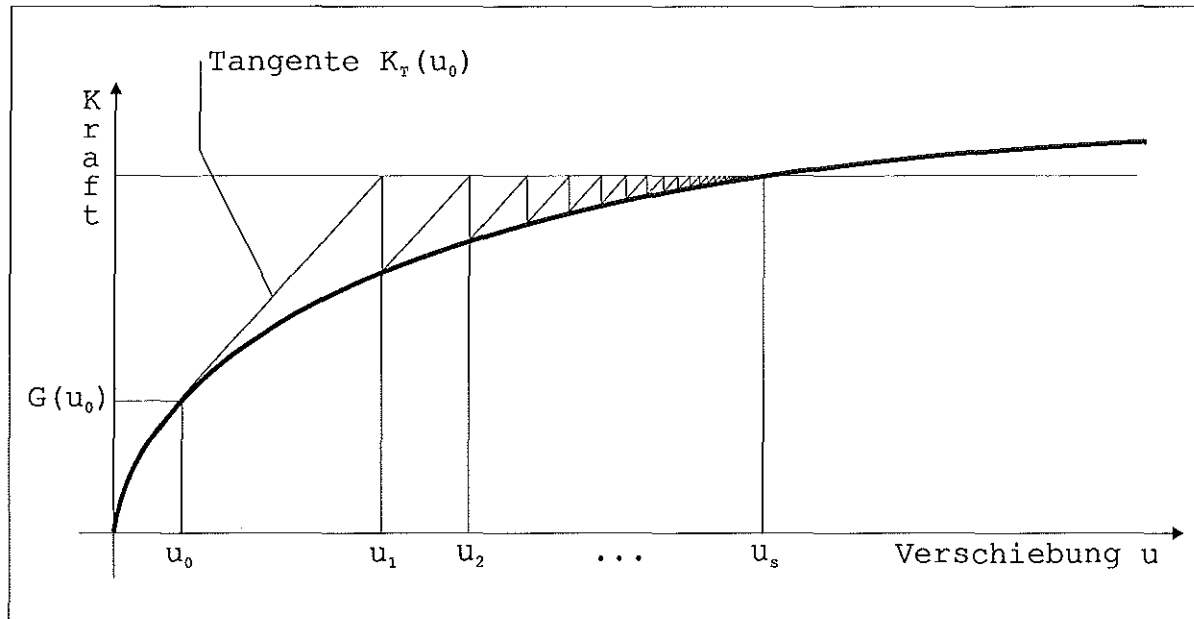


Abbildung 3.3: Ist der funktionale Zusammenhang G zwischen Knotenparameter und Kraft nicht linear, so kann der zu einer vorgegebenen Kraft gehörende Knotenparameter durch das modifizierte Newton-Verfahren, in welchem G linearisiert wird, numerisch berechnet werden.

Hierbei bedeuten

- $\frac{\partial}{\partial t} \left(\rho \frac{\partial x}{\partial t} \right)$ die Beschleunigung des Körpers,
- $\theta \frac{\partial x}{\partial t}$ die durch Dämpfung entstehende Kraft,
- $\frac{\partial \varepsilon(x)}{\partial x}$ die elastische Antwort des Körpers. $\varepsilon(x)$ entspricht dem Minimum des Energiepotentials, welches durch die elastische Deformation gegeben ist.
- $f(x, t)$ ist die Gesamtheit der äußeren Kräfte.

In dem hier vorgestellten Modell soll die Dynamik der Deformation keine Rolle spielen. Lediglich der Endzustand des Systems, in welchem sich alle Kräfte ausgleichen und daher keine Beschleunigung mehr an den Punkten angreift, ist von Interesse. Daher soll im folgenden nur eine vereinfachte Gleichung (3.6) betrachtet werden, in welcher die beiden ersten Terme vernachlässigt werden. In den nun folgenden Abschnitten werden verschiedene Verzerrungsmaße untersucht. Eine Definition der äußeren Kräfte $f(x, t)$ wird in Kapitel 4 gegeben.

Es gibt verschiedene Möglichkeiten, die Bewegung eines kontinuierlichen Körpers im Raum zu beschreiben. Insbesondere kann mit verschiedenen Koordinatensystemen gearbeitet werden, in denen die zu einem Körper gehörenden Größen dargestellt werden. Die hier interessierende Größe sind die Koordinaten des Körpers bzw. das daran gekoppelte

Verschiebungsvektorfeld u , welches die Bewegung des Körpers, hier einer Oberfläche, und damit auch seine Deformation charakterisiert.

Die Verschiebung u kann durch *körperbezogene* Koordinaten beschrieben werden durch

$$u_i = u_i(a_1, a_2, a_3, t), \quad i = 1, 2, 3.$$

Hierin sind a_1, a_2 und a_3 die sogenannten *Lagrangeschen Koordinaten*, welche fest einem Teilchen des betrachteten Körpers zugeordnet sind. u beschreibt somit die Verschiebung des Körpers zum Zeitpunkt t bei körperfesten Koordinaten, die sich sozusagen mit dem Körper mitbewegen. Bei festem a_1, a_2, a_3 und variierendem t gibt u die Verschiebung zu immer dem gleichen Punkt des Körpers zu den verschiedenen Zeitpunkten an.

u kann auch durch *raumbezogene* Koordinaten ausgedrückt werden:

$$u = u(x_1, x_2, x_3, t).$$

x_1, x_2 und x_3 sind die sogenannten *Eulerschen Koordinaten*, welche fest dem Raum, in dem sich der Körper bewegt, zugeordnet sind. Befindet sich ein Teilchen des Körpers zum Zeitpunkt t an der Stelle x_1, x_2, x_3 , so gibt u die Verschiebung gerade dieses Teilchens an. Neben dem Verschiebungsvektorfeld lassen sich auch die Raumkoordinaten selber in Lagrangescher Form ausdrücken:

$$x_i = x_i(a_p, t), \quad i, p = 1, 2, 3.$$

Wie auch das Verschiebungsvektorfeld, so beschreibt obiger Ausdruck die Bewegung des Körpers und kann als Abbildung von Ausgangskordinaten ($t=0$) auf aktuelle Koordinaten zum Zeitpunkt $t > 0$ gedeutet werden. Ist die Jacobi Determinante

$$J := \det \left(\frac{\partial x_i}{\partial a_j} \right)$$

positiv, so ist auch die Inversion

$$a_i = a_i(x_p, t), \quad i, p = 1, 2, 3$$

möglich.

Zum Zeitpunkt $t = 0$ sind beide Darstellungen identisch:

$$x_i = x_i(a_p, t = 0) = a_i, \quad a_i = a_i(x_p, t = 0) = x_i.$$

Sind Anfangskordinaten a_i und aktuelle Koordinaten x_i zum Zeitpunkt $t > 0$ gegeben, so ergibt sich für die Verschiebung u_i unmittelbar

$$u_i = x_i(a_p, t) - a_i.$$

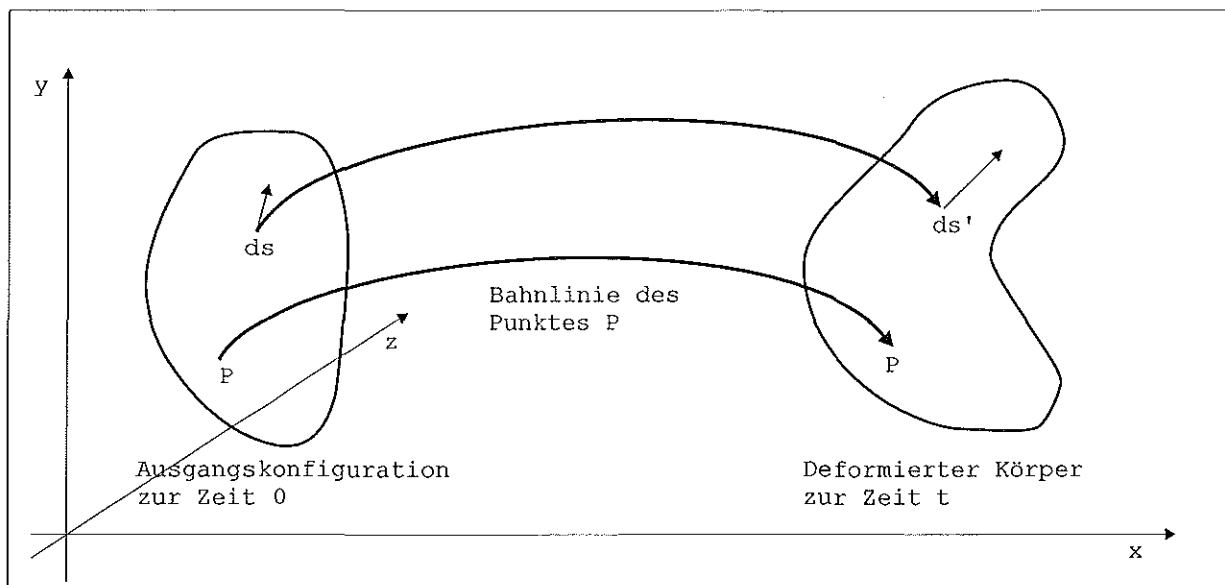


Abbildung 3.4: Ein kontinuierlicher Körper geht von seiner Ausgangskonfiguration zu einem Zustand zum Zeitpunkt t über. Das Linienelement ds wird zu ds' verzerrt.

3.3 Variationsprinzip

Bei gegebenem Verzerrungsmaß $\varepsilon(x)$ kann dieser Verzerrung eine Deformationsenergie Θ zugeordnet werden. Dazu wird die an jedem Punkt x des betrachteten Körpers gegebene Verzerrung ε aufintegriert.

$$\Theta = \int \frac{1}{2} C \varepsilon(x)^2 dV.$$

Das Extremum des Potentials Θ kann über das Variationsprinzip gefunden werden:

$$\begin{aligned} \delta\Theta &= \delta \int \frac{1}{2} C \varepsilon(x)^2 dV \\ &= \int C \varepsilon(x) \delta\varepsilon(x) dV = 0. \end{aligned} \quad (3.7)$$

Ausdruck (3.7) ist äquivalent mit dem Prinzip der kleinsten virtuellen Arbeit. Er beschreibt die Wirkung von Kräften ε entlang sogenannter 'virtueller Verrückungen' $\delta\varepsilon$. C ist im allgemeinen Fall ein sogenannter *Elastizitätstensor* und beschreibt den Zusammenhang zwischen Spannung und Verzerrung im Kontinuum. Zu beachten ist, daß das Extremum des Potentials eines diskreten Systems, also eines Systems, dessen Potential Θ nur von diskreten Zustandsgrößen u_i abhängt, durch die Ableitung des Potentials nach den Zustandsgrößen gegeben ist:

$$\begin{aligned} \delta\Theta(u_i) &= \frac{\partial\Theta(u_i)}{\partial u_i} \delta u_i = 0, \quad \delta u_i \text{ beliebig} \\ \Rightarrow \frac{\partial\Theta(u_i)}{\partial u_i} &= 0. \end{aligned}$$

Ein kontinuierliches System geht z. B. durch Diskretisierung mittels finiter Elemente in ein diskretes über. Die u_i entsprechen in diesem Fall den Knotenparametern.

3.4 Verzerrungsmaße am Linienelement

Die verschiedenen Möglichkeiten, Maße für die Verzerrung eines Körpers aufzustellen, werden im folgenden an einem einfachen, eindimensionalen Linienelement diskutiert. Im nächsten Abschnitt folgt dann die Verallgemeinerung einiger ausgesuchter Verzerrungsmaße auf das Kontinuum.

Das Linienelement habe die Ausgangslänge l_0 und nach Verzerrung die Länge l_n . Schon von der Intuition her ist klar, daß das Maß für die Verzerrung des Linienelementes mit der Änderung der Länge von l_0 auf l_n zusammenhängt. Die absolute Längenänderung Δl ist jedoch kein vernünftiges Maß, da die Dehnung eines kleinen Linienelementes offenbar eine größere Verzerrung hervorrufen muß als die Dehnung eines großen Linienelementes um den gleichen Betrag. Das Maß muß daher relativ definiert sein. Relativ kann zum einen relativ zur Ausgangslänge l_0 und zum anderen relativ zur Endlänge l_n bedeuten.

3.4.1 Lineare Verzerrung

Unter *linearer Verzerrung* ist die Differenz aus Ausgangs- und Endlänge, bezogen auf die Ausgangslänge, zu verstehen.

$$\varepsilon_E = \frac{l_n - l_0}{l_0} = \frac{l_n}{l_0} - 1.$$

In dieses Verzerrungsmaß geht die Endlänge l_n linear ein.

3.4.2 Greensche Verzerrung

Hier wird als Verzerrungsmaß nicht die Längendifferenz, sondern die Differenz der Längenquadrate, bezogen auf die quadrierte Ausgangslänge l_0 , verstanden:

$$\varepsilon_G = \frac{l_n^2 - l_0^2}{2l_0^2} = \frac{1}{2} \left(\left(\frac{l_n}{l_0} \right)^2 - 1 \right).$$

Zwischen diesem Maß und der linearen Verzerrung besteht folgender Zusammenhang:

$$\varepsilon_E = \frac{l_n - l_0}{l_n} = \frac{(l_n - l_0)(l_n + l_0)}{l_0(l_n + l_0)} = \frac{l_n^2 - l_0^2}{l_0^2(2 + \varepsilon_E)}.$$

Die Greensche Verzerrung ist nur bei kleinen Verzerrungen annähernd gleich der linearen Verzerrung.

3.4.3 Almansische Verzerrung

Dieses Maß ist ähnlich definiert wie die Greensche Verzerrung, jedoch bezieht sich die Differenz der Längenquadrate hier auf die aktuelle Elementlänge:

$$\varepsilon_A = \frac{l_n^2 - l_0^2}{2l_n^2}.$$

3.4.4 Logarithmische Verzerrung

Bei diesem Maß wird die infinitesimale Längenänderung bei der Dehnung des Stabes von seiner Ausgangslänge l_0 bis zu seiner Endlänge l_n integriert:

$$\varepsilon_L = \int_{l_0}^{l_n} \frac{dl}{l} = \ln \left(\frac{l_n}{l_0} \right).$$

Dieses logarithmische Maß wird in der Literatur oft als 'echte' Verzerrungsmaß bezeichnet, da es sich nicht auf eine fixe Ausgangsgeometrie bezieht, sondern die Verzerrung stetig während des gesamten Deformationsprozesses mißt. Insbesondere wird es auch für große Deformationen angewandt.

3.5 Kräfte am Linienelement

In diesem Abschnitt wird die Variation des Energiepotentials für die oben angegebenen Verzerrungsmaße berechnet. Aus diesen Überlegungen kann die Tauglichkeit der Maße für eine Verwendung in einem aktiven Konturmodell beurteilt werden. Zu beachten ist, daß alle obigen Verzerrungsmaße unabhängig vom betrachteten Punkt auf dem Linienelement sind. In diesem einfachen Modell wird die Verzerrung als konstant auf dem ganzen Element angenommen. Eine Integration ist daher sehr einfach vorzunehmen.

3.5.1 Kraft bei linearer Verzerrung

Da dieses Verzerrungsmaß relativ zur Ausgangslänge l_0 berechnet wird, wird hier zur Berechnung der Energie über das Ausgangsvolumen integriert:

$$\begin{aligned} \delta\Theta(\varepsilon_E) &= \int_0^{l_0} C \frac{l_n - l_0}{l_0} \delta \frac{l_n - l_0}{l_0} dl \\ &= l_0 C \frac{l_n - l_0}{l_0} \frac{\partial}{\partial l_n} \frac{l_n - l_0}{l_0} \delta l_n \\ &= C \frac{l_n - l_0}{l_0} \delta l_n = 0. \end{aligned} \tag{3.8}$$

Somit ergibt sich als Kraft:

$$F_E(l_n) = C \frac{l_n - l_0}{l_0}.$$

3.5.2 Kraft bei Greenscher Verzerrung

Analog zur linearen Verzerrung ergibt sich für die Greensche Verzerrung folgender Zusammenhang zwischen Kraft und Endlänge l_n :

$$F_G(l_n) = C \frac{l_n^3 - l_0^2 l_n}{2l_0^3} = \frac{C}{2} \left(\left(\frac{l_n}{l_0} \right)^3 - \frac{l_n}{l_0} \right)$$

Diese Funktion besitzt drei Nullstellen l_1, l_2 und l_3 , und zwar bei $l_1 = l_0$, $l_2 = -l_0$ und $l_3 = 0$. Die Nullstellen l_1 und l_2 entsprechen der Intuition, wann die Kraft infolge einer Dehnung verschwinden soll: wenn das Element gar nicht gedehnt ist (Nullstelle l_1) und wenn das Element um 180° gedreht, aber ebenfalls nicht gedehnt ist (Nullstelle l_2). Die Nullstelle an l_3 ist hingegen nicht mit einem realen Kraft-Dehnungsgesetz vereinbar. Bei einem bis auf die Länge Null zusammengedrückten Element entsteht keine Rückstellkraft. Wie in Abbildung 3.5 ersichtlich, verhält sich das Element bei einer Stauchung so, daß es der Stauchung zwar zunächst eine (betragsmäßig) immer größer werdende Kraft entgegensetzt, die Kraft dann aber wieder abnimmt und bei vollständigem Zusammendrücken sogar Null wird. Nur in einem kleinen Bereich um l_0 herum, also bei kleiner Verzerrung, verhält das Element sich annähernd linear. Für große Stauchungen ist das Greensche Verzerrungsmaß daher nicht zu verwenden.

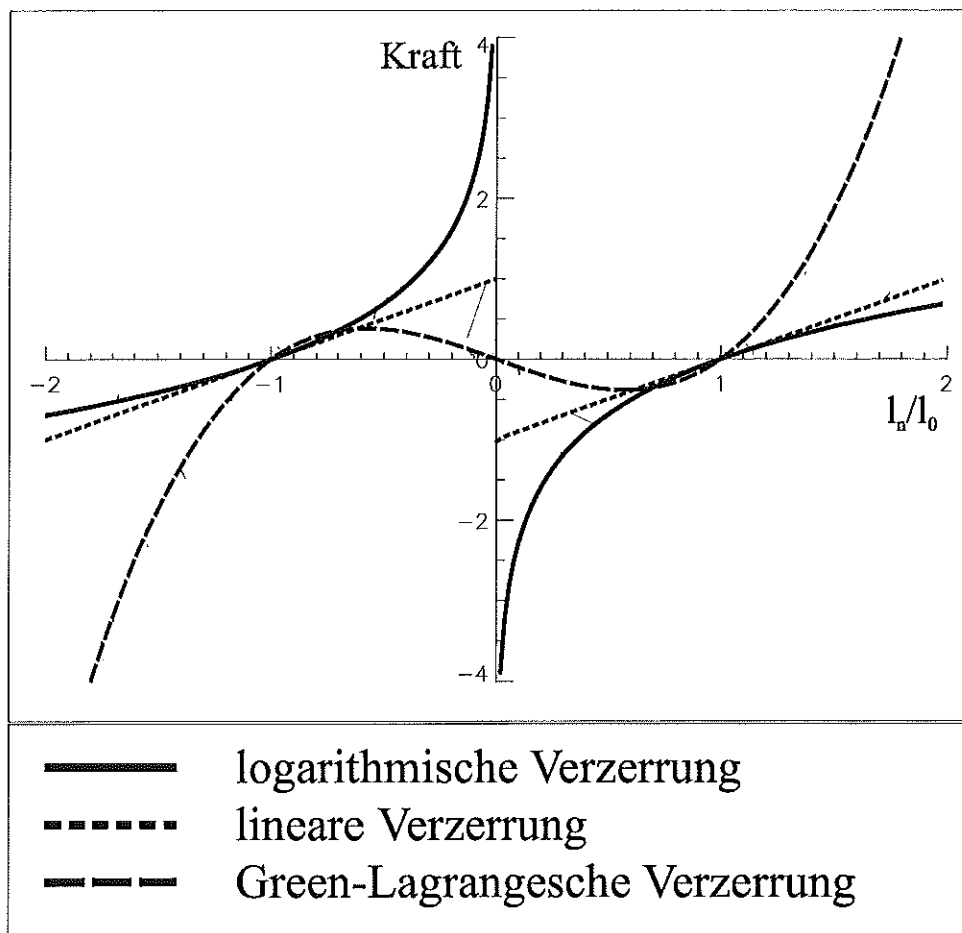


Abbildung 3.5: Verzerrungs-Kraft-Relation für verschiedene Verzerrungsmaße. Die x-Achse entspricht der relativen Längenänderung l_n/l_0 .

3.5.3 Kraft bei Almansischer Verzerrung

Bei einer Dehnung auf die Länge l_n entwickelt das Linienelement folgende Kraft:

$$F_A(l_n) = C \frac{l_0^2}{l_n^2} - \frac{l_0^4}{l_n^4}.$$

Da $\lim_{l_n \rightarrow \infty} F_A(l_n) = 0$ gilt, ist auch dieses Maß nicht gut geeignet, um größere Verzerrungen zu beschreiben.

3.5.4 Kraft bei logarithmischer Verzerrung

Als Kraft erhält man hier folgende Funktion:

$$F_L(l_n) = C \ln \left(\frac{l_n}{l_0} \right) l_0.$$

Auch diese Kraft verhält sich in der Nähe von $l_n = l_0$ annähernd linear. Bei einer Stauchung steigt sie stark an, bei einer Dehnung jedoch wird der Kraftzuwachs bei steigender Dehnung immer geringer und verhält sich nicht linear. Als Kraft für ein aktives Konturmodell ist sie daher nur bedingt geeignet, da die Kräfte in Gebieten mit starken Dehnungen nicht proportional zur Dehnung selber ansteigen, was eine über die Gesamtoberfläche ungleichmäßige Verzerrung zur Folge hat. Sie bietet sich allerdings an, wenn starke Elementstauchungen vermieden werden sollen, da die Kraft bei Stauchung sehr schnell ansteigt.

Oben wurden verschiedene Energieterme vorgestellt, mit denen die Verzerrung eines Linienelementes gemessen werden kann. Aus den verschiedenen Energietermen resultieren auch unterschiedliche Kraft-Verzerrungs-Funktionen, die jeweils andere Eigenschaften haben. Dies kann bei der Modellierung spezieller Deformationseigenschaften, welche sich ja gerade über die Kraft-Verzerrungsfunktionen definieren lassen, behilflich sein. Wie sich gewünschte Eigenschaften im kontinuierlichen Raum bilden lassen, wird in den folgenden Abschnitten entwickelt.

3.6 Verzerrungsmaße im Kontinuum

Im folgenden sei ein zweidimensionaler, kontinuierlicher Körper über folgende Variablen definiert:

Ausgangskoordinaten des Objekts:

$$a(r, s) := \begin{pmatrix} a_1(r, s) \\ a_2(r, s) \end{pmatrix}.$$

Aktuelle Koordinaten des Objekts:

$$x(r, s) := \begin{pmatrix} x_1(r, s) \\ x_2(r, s) \end{pmatrix}.$$

Verschiebungsvektorfeld:

$$u(r, s) := \begin{pmatrix} u_1(r, s) \\ u_2(r, s) \end{pmatrix}.$$

Da das Objekt in seiner aktuellen Lage x gerade um u bezüglich seiner Ausgangslage a verschoben ist, gilt:

$$x = a + u. \quad (3.9)$$

Der *Deformationsgradient* F ist gegeben durch:

$$F := \frac{\partial x}{\partial a} = \begin{pmatrix} \frac{\partial x_1}{\partial a_1} & \frac{\partial x_1}{\partial a_2} \\ \frac{\partial x_2}{\partial a_1} & \frac{\partial x_2}{\partial a_2} \end{pmatrix}. \quad (3.10)$$

Wird ein Linienelement da des Objektes in seiner Ausgangslage mit demselben Linienelement in der aktuellen Lage dx verglichen, so gilt:

$$dx = \frac{\partial x}{\partial a} da = \begin{pmatrix} \frac{\partial x_1}{\partial a_1} & \frac{\partial x_1}{\partial a_2} \\ \frac{\partial x_2}{\partial a_1} & \frac{\partial x_2}{\partial a_2} \end{pmatrix} da = F da.$$

Der Deformationsgradient F verknüpft also infinitesimale Elemente des Ausgangsobjektes mit denen des aktuellen Objektes.

3.6.1 Green-Lagrangescher Deformationstensor

Eine Metrik des deformierten Raumes kann über den *Green-Lagrangeschen Deformationstensor* eingeführt werden. Dieser vergleicht, wie beim eindimensionalen Linienelement auch, die quadrierte Länge eines Linienelementes in Ausgangs- und aktueller Lage.

Es gilt dabei:

$$\begin{aligned} dx^2 - da^2 &= dx^{tr} dx - da^{tr} da = da^{tr} F^{tr} F da - da^{tr} da \\ &= da^{tr} \begin{pmatrix} \frac{\partial x_1}{\partial a_1} & \frac{\partial x_1}{\partial a_2} \\ \frac{\partial x_2}{\partial a_1} & \frac{\partial x_2}{\partial a_2} \end{pmatrix}^{tr} \begin{pmatrix} \frac{\partial x_1}{\partial a_1} & \frac{\partial x_1}{\partial a_2} \\ \frac{\partial x_2}{\partial a_1} & \frac{\partial x_2}{\partial a_2} \end{pmatrix} da - da^{tr} I da \\ &= da^{tr} (F^{tr} F - I) da. \end{aligned} \quad (3.11)$$

Der Green-Lagrangesche Deformationstensor E wird daher definiert durch

$$E = \frac{1}{2} (F^{tr} F - I).$$

3.6.2 Green–Lagrangescher Deformationstensor in Koordinaten der Verschiebung u

Wird die Ableitung des Verschiebungsvektorfeldes u nach den Ausgangskoordinaten a betrachtet, so ergibt sich

$$D := \frac{\partial u}{\partial a} = \begin{pmatrix} \frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} \\ \frac{\partial u_2}{\partial a_1} & \frac{\partial u_2}{\partial a_2} \end{pmatrix}.$$

Unter Beachtung von (3.9) folgt

$$\frac{\partial x}{\partial a} = \frac{\partial(a + u)}{\partial a} = I + \frac{\partial u}{\partial a}.$$

Somit lässt sich eine Verbindung zwischen dem Deformationsgradienten F und dem Verschiebungsgradienten D herstellen:

$$F = \begin{pmatrix} \frac{\partial x_1}{\partial a_1} & \frac{\partial x_1}{\partial a_2} \\ \frac{\partial x_2}{\partial a_1} & \frac{\partial x_2}{\partial a_2} \end{pmatrix} = \begin{pmatrix} 1 + \frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} \\ \frac{\partial u_2}{\partial a_1} & 1 + \frac{\partial u_2}{\partial a_2} \end{pmatrix} = D + I. \quad (3.12)$$

Für den Green-Lagrangeschen Tensor E folgert man daraus:

$$E = \frac{1}{2}(F^{tr}F - I) = \frac{1}{2}((D + I)^{tr}(D + I) - I) = \frac{1}{2}((D^{tr} + D) + D^{tr}D).$$

Ausgerechnet ergibt sich für E folgender Ausdruck:

$$\begin{aligned} 2E &= \begin{pmatrix} 1 + \frac{\partial u_1}{\partial a_1} & \frac{\partial u_2}{\partial a_1} \\ \frac{\partial u_1}{\partial a_2} & 1 + \frac{\partial u_2}{\partial a_2} \end{pmatrix} \begin{pmatrix} 1 + \frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} \\ \frac{\partial u_2}{\partial a_1} & 1 + \frac{\partial u_2}{\partial a_2} \end{pmatrix} - I \\ &= \begin{pmatrix} (1 + \frac{\partial u_1}{\partial a_1})^2 + (\frac{\partial u_2}{\partial a_1})^2 & (1 + \frac{\partial u_1}{\partial a_1})\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1}(1 + \frac{\partial u_2}{\partial a_2}) \\ \frac{\partial u_1}{\partial a_2}(1 + \frac{\partial u_1}{\partial a_1}) + (1 + \frac{\partial u_2}{\partial a_2})\frac{\partial u_2}{\partial a_1} & (\frac{\partial u_1}{\partial a_2})^2 + (1 + \frac{\partial u_2}{\partial a_2})^2 \end{pmatrix} - I \end{aligned} \quad (3.13)$$

$$\begin{aligned}
&= \begin{pmatrix} 1 + 2\frac{\partial u_1}{\partial a_1} + \left(\frac{\partial u_1}{\partial a_1}\right)^2 + \left(\frac{\partial u_2}{\partial a_1}\right)^2 & \frac{\partial u_1}{\partial a_2} + \frac{\partial u_1}{\partial a_1}\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} + \frac{\partial u_2}{\partial a_1}\frac{\partial u_2}{\partial a_2} \\ \frac{\partial u_1}{\partial a_2} + \frac{\partial u_1}{\partial a_1}\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} + \frac{\partial u_2}{\partial a_1}\frac{\partial u_2}{\partial a_2} & \left(\frac{\partial u_1}{\partial a_2}\right)^2 + 1 + 2\frac{\partial u_2}{\partial a_2} + \left(\frac{\partial u_2}{\partial a_2}\right)^2 \end{pmatrix} - I \\
&= \begin{pmatrix} 2\frac{\partial u_1}{\partial a_1} + \left(\frac{\partial u_1}{\partial a_1}\right)^2 + \left(\frac{\partial u_2}{\partial a_1}\right)^2 & \frac{\partial u_1}{\partial a_2} + \frac{\partial u_1}{\partial a_1}\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} + \frac{\partial u_2}{\partial a_1}\frac{\partial u_2}{\partial a_2} \\ \frac{\partial u_1}{\partial a_2} + \frac{\partial u_1}{\partial a_1}\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} + \frac{\partial u_2}{\partial a_1}\frac{\partial u_2}{\partial a_2} & \left(\frac{\partial u_1}{\partial a_2}\right)^2 + 2\frac{\partial u_2}{\partial a_2} + \left(\frac{\partial u_2}{\partial a_2}\right)^2 \end{pmatrix}.
\end{aligned}$$

3.6.3 Lineare Näherung

Unter der Annahme, daß die Verzerrungen des Körpers unendlich klein sind, ergibt sich aus dem Green-Lagrangeschen Verzerrungstensor unmittelbar der sogenannte klassische Verzerrungstensor ε . Er ergibt sich aus E , indem alle quadratischen Anteile weggelassen werden:

$$\varepsilon = \begin{pmatrix} 2\frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} \\ \frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1} & 2\frac{\partial u_2}{\partial a_2} \end{pmatrix}.$$

Er ist nur bei kleinen Verzerrungen und Bewegungen des Elementes gültig. Da hier ein aktives Konturmodell mit großen Verzerrungen und Dehnungen und insbesondere auch mit großen Starrkörperbewegungen entwickelt werden soll, wird der klassische Verzerrungstensor ε im folgenden nicht weiter verwendet.

3.6.4 Polares Zerlegungstheorem

Mit Hilfe des polaren Zerlegungstheorems kann ein Deformationsmaß definiert werden, welches der linearen Verzerrung des im vorherigen Abschnitt besprochenen Linienelementes entspricht. Im kontinuierlichen Fall ist es ein lineares Maß, welches die Längenänderung eines infinitesimalen Linienelementes im Verhältnis zu seiner Ausgangslänge beschreibt. Dies geschieht hier über den Deformationsgradient F , welcher neben der eigentlichen (Dreh-)Streckung auch noch eine reine Drehung enthält. F läßt sich jedoch nach dem polaren Zerlegungstheorem aufspalten in den Streck- und in den Drehanteil. Es gibt danach Matrizen U , V und R , so daß gilt:

$$F = RU = VR.$$

Hier ist U der sogenannte *Rechtsstrecktensor*, V der *Linksstrecktensor* und die Rotationsmatrix R ein *eigentlich orthogonaler Tensor*, d. h.

$$R^{tr}R = RR^{tr} = I, \text{ sowie } \det(R) = 1. \quad (3.14)$$

Wegen der in (3.14) angegebenen Eigenschaften von R kann U (und ebenso V) berechnet werden mittels

$$F^{tr}F = U^{tr}R^{tr}RU = UU = U^2, \text{ also } U = (F^{tr}F)^{\frac{1}{2}}.$$

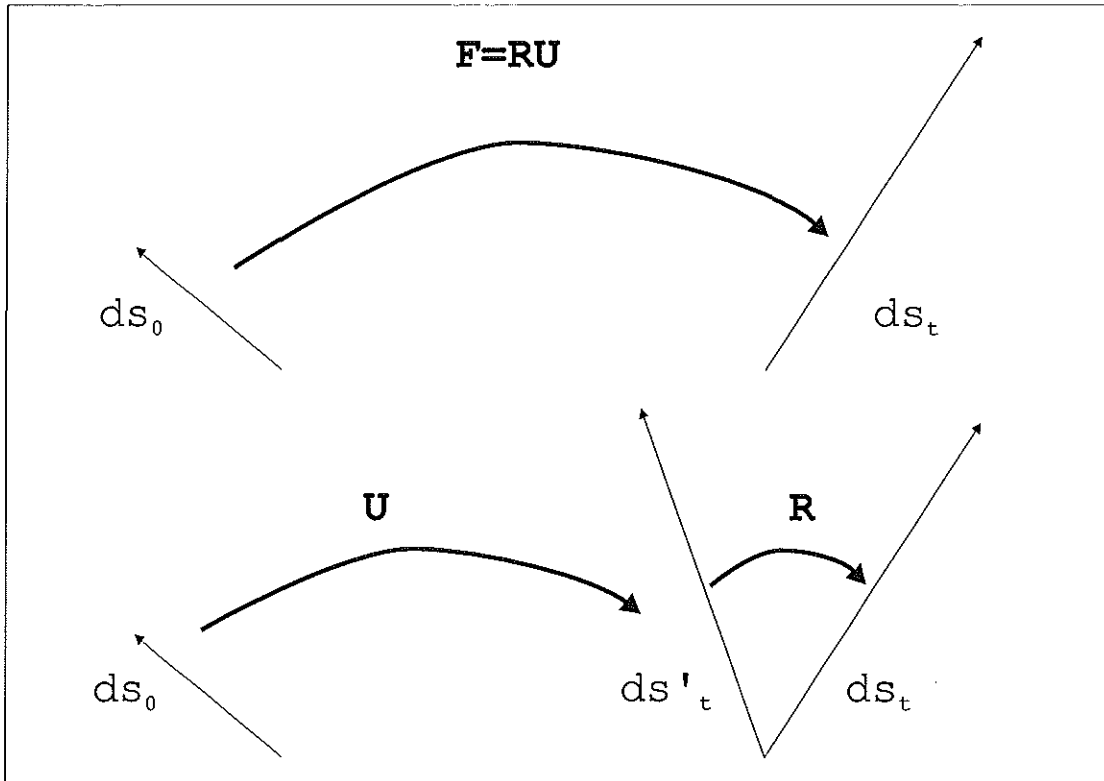


Abbildung 3.6: Das Linienelement ds_0 geht mittels F unmittelbar in ds_t über. Eine Zerlegung von F in R und U erlaubt die Unterscheidung zwischen der eigentlichen (Dreh-) Streckung U und der reinen Starrkörperdrehung R .

Die Berechnung von U ist im Eigensystem von $F^{tr}F$ möglich. Sind c_1 und c_2 die Eigenwerte von $F^{tr}F$, so gilt:

$$U = \begin{pmatrix} \sqrt{c_1} & 0 \\ 0 & \sqrt{c_2} \end{pmatrix}.$$

$\mu_1 := \sqrt{c_1}$ und $\mu_2 := \sqrt{c_2}$ sind somit ein Maß für die Streckung eines Linienelementes in Richtung der Hauptverzerrungsachsen. Eine Schiebung (Scherung) gibt es in Richtung der Hauptachsen nicht.

3.7 Realisierung

Für das hier verwendete aktive Konturmodell wurde das oben beschriebene lineare Verzerungsmaß verwendet. Konkret wurden zweidimensionale Dreieckselemente benutzt, deren Verschiebung u im finite-Elemente Ansatz mit Funktionen erster Ordnung angenähert wurde. Die Verwendung von Funktionen erster Ordnung ermöglicht eine einfache Berechnung der resultierenden Kräfte. Das die Kräfte beschreibende Integral enthält nämlich erste Ableitungen der Ansatzfunktionen. Diese Ableitungen sind bei Funktionen erster Ordnung Konstanten, welche nicht mehr vom Ort abhängen. Das Integral kann somit direkt, ohne numerische Approximation (z. B. durch Gauss-Quadratur), ausgewertet werden. Ein Nachteil dieses einfachen Ansatzes besteht darin, daß mit ihm nicht die Krümmung der

Oberfläche beschrieben werden kann. Soll die Krümmung berücksichtigt werden, so ist entweder ein Ansatz höherer Ordnung zu verwenden, oder aber ein hybrides Kräftemodell zu benutzen, in welchem als Verzerrungsmaß das nachfolgend konkret vorgestellte Verfahren und als Krümmungsmaß die im vorherigen Kapitel diskutierten diskreten Winkelkräfte dienen.

Alle zur Berechnung der aus einer Verzerrung resultierenden Kräfte notwendigen Formeln werden nun vorgestellt. Dazu gehören die Berechnung der Eigenwerte des Deformationsgradienten, die verwendete Energiegleichung und schließlich die Diskretisierung über finite Elemente.

3.7.1 Berechnung der Eigenwerte einer 2×2 Matrix

Im allgemeinen erfolgt die Berechnung der Eigenwerte c_1 und c_2 einer 2×2 Matrix

$$A := \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

über die Bestimmung der Nullstellen des charakteristischen Polynoms

$$p(c) := \det(A - cI) = \det \begin{pmatrix} a_{11} - c & a_{12} \\ a_{21} & a_{22} - c \end{pmatrix} = (a_{11} - c)(a_{22} - c) - a_{12}a_{21}.$$

Es ergibt sich folgender Rechenweg:

$$\begin{aligned} \det(A - cI) &= (a_{11} - c)(a_{22} - c) - a_{12}a_{21} \\ &= c^2 - c(a_{11} + a_{22}) + a_{11}a_{22} - a_{12}a_{21} \end{aligned}$$

und somit

$$\begin{aligned} c_1 &= \frac{a_{11} + a_{22}}{2} + \sqrt{\frac{(a_{11} + a_{22})^2}{4} - a_{11}a_{22} + a_{12}a_{21}}, \\ c_2 &= \frac{a_{11} + a_{22}}{2} - \sqrt{\frac{(a_{11} + a_{22})^2}{4} - a_{11}a_{22} + a_{12}a_{21}}. \end{aligned}$$

Im Falle einer symmetrischen Matrix A ist $a_{12} = a_{21}$, und es ergibt sich folgende Vereinfachung:

$$c_1 = \frac{a_{11} + a_{22}}{2} + \sqrt{\frac{(a_{11} + a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}, \quad (3.15)$$

$$c_2 = \frac{a_{11} + a_{22}}{2} - \sqrt{\frac{(a_{11} + a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}. \quad (3.16)$$

3.7.2 Berechnung der Eigenwerte von $F^{tr}F$

Nach Gleichung (3.12) und (3.13) ergibt sich für $A = F^{tr}F$:

$$a_{11} = \left(1 + \frac{\partial u_1}{\partial a_1}\right)^2 + \left(\frac{\partial u_2}{\partial a_1}\right)^2 \quad (3.17)$$

$$a_{12} = a_{21} = \left(1 + \frac{\partial u_1}{\partial a_1}\right)\frac{\partial u_1}{\partial a_2} + \frac{\partial u_2}{\partial a_1}\left(1 + \frac{\partial u_2}{\partial a_2}\right) \quad (3.18)$$

$$a_{22} = \left(\frac{\partial u_1}{\partial a_2}\right)^2 + \left(1 + \frac{\partial u_2}{\partial a_2}\right)^2. \quad (3.19)$$

Somit hängen die Eigenwerte vom Verschiebungsvektorfeld u ab und können berechnet werden, wenn dieses bekannt ist.

3.7.3 Finite Elemente Approximation

Das Verschiebungsvektorfeld u sowie die Koordinaten des deformierten Objektes werden über finite Elemente approximiert. Im folgenden wird eine lineare Approximation über Dreieckselemente betrachtet. Mittels sogenannter *Shape-Funktionen* werden die Körperkoordinaten a und x und die Koordinaten u des Verschiebungsvektorfeldes linear parametrisiert, wobei die Koordinatenwerte an den Eckpunkten des Dreieckselementes gegeben sein sollen. Ist ein Dreieckselement mit den Punkten p_1 , p_2 und p_3 gegeben, wird im weiteren folgende Notation verwendet:

Ausgangskoordinaten im Punkt p_i :

$$a(p_i) = \begin{pmatrix} a_1^i \\ a_2^i \end{pmatrix}, \quad i = 1, 2, 3.$$

Aktuelle Koordinaten im Punkt p_i :

$$x(p_i) = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}, \quad i = 1, 2, 3.$$

Verschiebung im Punkt p_i :

$$u(p_i) = \begin{pmatrix} u_1^i \\ u_2^i \end{pmatrix}, \quad i = 1, 2, 3.$$

Der obere Index bezeichnet die Nummer des Punktes und der untere die Koordinatenrichtung. Mittels einer linearen Shape-Funktion lassen sich die obigen Werte stetig auf dem ganzen Dreieckselement definieren:

$$a(\xi, \nu) := \begin{pmatrix} a_1(\xi, \nu) \\ a_2(\xi, \nu) \end{pmatrix} = \begin{pmatrix} (1 - \xi - \nu)a_1^1 + \xi a_1^2 + \nu a_1^3 \\ (1 - \xi - \nu)a_2^1 + \xi a_2^2 + \nu a_2^3 \end{pmatrix}, \quad 0 \leq \xi, \nu \leq 1, \quad \xi + \nu \leq 1. \quad (3.20)$$

Ganz analog definiert man $x(\xi, \nu)$ und $u(\xi, \nu)$.

Was nun benötigt wird, ist die Ableitung des Verschiebungsvektorfeldes u nach den Ausgangskordinaten a , also ist $\partial u / \partial a$ gesucht. Diese Ableitung berechnet sich nach der Kettenregel durch

$$\begin{aligned} \frac{\partial u}{\partial a} &= \begin{pmatrix} \frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} \\ \frac{\partial u_2}{\partial a_1} & \frac{\partial u_2}{\partial a_2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial u_1}{\partial \xi} \frac{\partial \xi}{\partial a_1} + \frac{\partial u_1}{\partial \nu} \frac{\partial \nu}{\partial a_1} & \frac{\partial u_1}{\partial \xi} \frac{\partial \xi}{\partial a_2} + \frac{\partial u_1}{\partial \nu} \frac{\partial \nu}{\partial a_2} \\ \frac{\partial u_2}{\partial \xi} \frac{\partial \xi}{\partial a_1} + \frac{\partial u_2}{\partial \nu} \frac{\partial \nu}{\partial a_1} & \frac{\partial u_2}{\partial \xi} \frac{\partial \xi}{\partial a_2} + \frac{\partial u_2}{\partial \nu} \frac{\partial \nu}{\partial a_2} \end{pmatrix}. \end{aligned} \quad (3.21)$$

Die partiellen Ableitungen von u_1 und u_2 nach ξ und ν , ausgedrückt in Punktkoordinaten, ergeben sich unter Berücksichtigung der Shape-Funktionen (3.20) wie folgt:

$$\frac{\partial u_1}{\partial \xi} = u_1^2 - u_1^1 \quad (3.22)$$

$$\frac{\partial u_1}{\partial \nu} = u_1^3 - u_1^1 \quad (3.23)$$

$$\frac{\partial u_2}{\partial \xi} = u_2^2 - u_2^1 \quad (3.24)$$

$$\frac{\partial u_2}{\partial \nu} = u_2^3 - u_2^1. \quad (3.25)$$

Die partiellen Ableitungen von ξ und ν nach a_1 und a_2 lassen sich ebenfalls berechnen:

$$da = \begin{pmatrix} da_1 \\ da_2 \end{pmatrix} = \begin{pmatrix} \frac{\partial a_1}{\partial \xi} & \frac{\partial a_1}{\partial \nu} \\ \frac{\partial a_2}{\partial \xi} & \frac{\partial a_2}{\partial \nu} \end{pmatrix} \begin{pmatrix} d\xi \\ d\nu \end{pmatrix}.$$

Mit Hilfe der Jacobi-Determinanten J

$$J = \det \begin{pmatrix} \frac{\partial a_1}{\partial \xi} & \frac{\partial a_1}{\partial \nu} \\ \frac{\partial a_2}{\partial \xi} & \frac{\partial a_2}{\partial \nu} \end{pmatrix} = \det \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix} = (a_1^2 - a_1^1)(a_2^3 - a_2^1) - (a_1^3 - a_1^1)(a_2^2 - a_2^1)$$

ergibt sich

$$\begin{aligned} \frac{\partial \xi}{\partial a_1} &= \frac{a_2^3 - a_2^1}{J}, & \frac{\partial \nu}{\partial a_1} &= \frac{a_2^2 - a_2^1}{J}, \\ \frac{\partial \xi}{\partial a_2} &= \frac{a_1^3 - a_1^1}{J}, & \frac{\partial \nu}{\partial a_2} &= \frac{a_1^2 - a_1^1}{J}. \end{aligned}$$

Man beachte, daß der Wert der Jacobi-Determinanten J gerade der doppelte Flächeninhalt des Dreieckselementes ist.

Nun kann man die Gleichungen (3.22)–(3.25) in (3.21) einsetzen. Dies ergibt

$$\begin{aligned}
\frac{\partial u}{\partial a} &= \begin{pmatrix} \frac{\partial u_1}{\partial a_1} & \frac{\partial u_1}{\partial a_2} \\ \frac{\partial u_2}{\partial a_1} & \frac{\partial u_2}{\partial a_2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial u_1}{\partial \xi} \frac{\partial \xi}{\partial a_1} + \frac{\partial u_1}{\partial \nu} \frac{\partial \nu}{\partial a_1} & \frac{\partial u_1}{\partial \xi} \frac{\partial \xi}{\partial a_2} + \frac{\partial u_1}{\partial \nu} \frac{\partial \nu}{\partial a_2} \\ \frac{\partial u_2}{\partial \xi} \frac{\partial \xi}{\partial a_1} + \frac{\partial u_2}{\partial \nu} \frac{\partial \nu}{\partial a_1} & \frac{\partial u_2}{\partial \xi} \frac{\partial \xi}{\partial a_2} + \frac{\partial u_2}{\partial \nu} \frac{\partial \nu}{\partial a_2} \end{pmatrix} \\
&= \begin{pmatrix} (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_1} & (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_2} \\ (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1} & (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2} \end{pmatrix}
\end{aligned} \tag{3.26}$$

Durch die oben vorgestellte Approximation des Verschiebungsvektorfeldes u auf einem finiten Dreieckselement läßt sich $F^{tr}F$ für dieses Element so darstellen, daß die einzigen unbekannten Parameter die Verschiebungen in den Knotenpunkten sind (6 Parameter). Habe $F^{tr}F$ die Form

$$F^{tr}F = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

so ergibt sich nach (3.17)–(3.19):

$$\begin{aligned}
a_{11} &= \left(1 + (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_1}\right)^2 \\
&\quad + \left((u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1}\right)^2
\end{aligned} \tag{3.27}$$

$$\begin{aligned}
a_{12} = a_{21} &= \left(1 + (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_1}\right) \left((u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_2}\right) \\
&\quad + \left((u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1}\right) \\
&\quad \left(1 + (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2}\right)
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
a_{22} &= \left((u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_2}\right)^2 \\
&\quad + \left(1 + (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2}\right)^2.
\end{aligned} \tag{3.29}$$

3.7.4 Verzerrungsenergie

Sei ein Dreieckselement V gegeben, so sind die Wurzeln der Eigenwerte c_1 und c_2 von $F^{tr}F$ ein Maß für die Verzerrung des Elementes. Im verzerrungsfreien Zustand haben beide Parameter den Wert 1. Daher kann man eine Verzerrungsenergie E wie folgt definieren:

$$\begin{aligned}
E &= \int_V \frac{1}{2} ((\sqrt{c_1(x,y)} - 1)^2 + (\sqrt{c_2(x,y)} - 1)^2) dV \\
&= \int_{\xi} \int_{\nu} \frac{1}{2} ((\sqrt{c_1(\xi,\nu)} - 1)^2 + (\sqrt{c_2(\xi,\nu)} - 1)^2) J d\nu d\xi.
\end{aligned} \tag{3.30}$$

Aufgrund der linearen Shape-Funktionen sind c_1 und c_2 nicht ortsabhängig, sondern überall auf der Oberfläche des Dreieckselementes gleich. Daher lassen sich die Terme mit c_1 und c_2 vor das Integral ziehen, und unter Beachtung von $0 \leq \xi, \nu \leq 1$, $\xi + \nu \leq 1$ ergibt sich:

$$\begin{aligned}
E &= \int_{\xi} \int_{\nu} \frac{1}{2} ((\sqrt{c_1} - 1)^2 + (\sqrt{c_2} - 1)^2) J d\nu d\xi \\
&= \frac{1}{2} ((\sqrt{c_1} - 1)^2 + (\sqrt{c_2} - 1)^2) \int_{\xi} \int_{\nu} J d\nu d\xi \\
&= \frac{1}{2} ((\sqrt{c_1} - 1)^2 + (\sqrt{c_2} - 1)^2) \frac{1}{2} J \\
&= \frac{J}{4} ((\sqrt{c_1} - 1)^2 + (\sqrt{c_2} - 1)^2).
\end{aligned} \tag{3.31}$$

Die an den Knotenpunkten des Dreieckselementes angreifenden Kräfte erhält man, indem man die Energie E minimiert. Dazu bestimmt man die Ableitungen von E nach den sechs Parametern $u_1^1, u_2^1, \dots, u_2^3$.

$$\frac{\partial E}{\partial u_i^j} = \frac{J}{4} \left(\frac{(\sqrt{c_1} - 1)}{\sqrt{c_1}} \frac{\partial c_1}{\partial u_i^j} + \frac{(\sqrt{c_2} - 1)}{\sqrt{c_2}} \frac{\partial c_2}{\partial u_i^j} \right). \tag{3.32}$$

Gesucht sind also $\frac{\partial c_1}{\partial u_i^j}$ und $\frac{\partial c_2}{\partial u_i^j}$ für $i = 1, 2$ und $j = 1, 2, 3$. Nun hängen c_1 und c_2 nach (3.15) und (3.16) zunächst von a_{11}, a_{22} und a_{12} ab. Daher wird im folgenden $\frac{\partial c_i}{\partial a_{jk}}$ gebildet.

$$\frac{\partial c_1}{\partial a_{11}} = \frac{1}{2} + \frac{\frac{1}{2}(a_{11} + a_{22}) - a_{22}}{2\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}} \tag{3.33}$$

$$\frac{\partial c_1}{\partial a_{22}} = \frac{1}{2} + \frac{\frac{1}{2}(a_{11} + a_{22}) - a_{11}}{2\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}} \tag{3.34}$$

$$\frac{\partial c_1}{\partial a_{12}} = \frac{a_{12}}{\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}} \tag{3.35}$$

$$\frac{\partial c_2}{\partial a_{11}} = \frac{1}{2} - \frac{\frac{1}{2}(a_{11} + a_{22}) - a_{22}}{2\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}} \tag{3.36}$$

$$\frac{\partial c_2}{\partial a_{22}} = \frac{1}{2} - \frac{\frac{1}{2}(a_{11} + a_{22}) - a_{11}}{2\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}} \quad (3.37)$$

$$\frac{\partial c_2}{\partial a_{12}} = -\frac{a_{12}}{\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2}}. \quad (3.38)$$

Die Ableitungen (3.33) bis (3.38) sind nur dann definiert, wenn der Nenner des in den Gleichungen vorkommenden Bruches ungleich Null, also wenn $\sqrt{\frac{(a_{11}+a_{22})^2}{4} - a_{11}a_{22} + a_{12}^2} = \sqrt{\frac{(a_{11}-a_{22})^2}{4} + a_{12}^2} \neq 0$ ist³. Die im folgenden vorgestellten Formeln gelten nur unter dieser Voraussetzung. Es ist der am häufigsten vorkommende Fall. Der Fall, daß dieser Nenner Null wird, wird in Abschnitt 3.8.1 behandelt.

Nach (3.27), (3.28) und (3.29) hängen a_{11}, a_{22} und a_{12} direkt von $u_1^1, \dots, u_2^3$ ab. Daher werden die folgenden 18 Ableitungen gebildet:

$$\frac{\partial a_{11}}{\partial u_1^1} = 2(1 + (u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_1})(-\frac{\partial \xi}{\partial a_1} - \frac{\partial \nu}{\partial a_1}) \quad (3.39)$$

$$\frac{\partial a_{11}}{\partial u_2^1} = 2((u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_1})(-\frac{\partial \xi}{\partial a_1} - \frac{\partial \nu}{\partial a_1}) \quad (3.40)$$

$$\frac{\partial a_{11}}{\partial u_1^2} = 2(1 + (u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_1})\frac{\partial \xi}{\partial a_1} \quad (3.41)$$

$$\frac{\partial a_{11}}{\partial u_2^2} = 2((u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_1})\frac{\partial \xi}{\partial a_1} \quad (3.42)$$

$$\frac{\partial a_{11}}{\partial u_1^3} = 2(1 + (u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_1})\frac{\partial \nu}{\partial a_1} \quad (3.43)$$

$$\frac{\partial a_{11}}{\partial u_2^3} = 2((u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_1})\frac{\partial \nu}{\partial a_1} \quad (3.44)$$

$$\frac{\partial a_{22}}{\partial u_1^1} = 2((u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_2})(-\frac{\partial \xi}{\partial a_2} - \frac{\partial \nu}{\partial a_2}) \quad (3.45)$$

$$\frac{\partial a_{22}}{\partial u_2^1} = 2(1 + (u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_2})(-\frac{\partial \xi}{\partial a_2} - \frac{\partial \nu}{\partial a_2}) \quad (3.46)$$

$$\frac{\partial a_{22}}{\partial u_1^2} = 2((u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_2})\frac{\partial \xi}{\partial a_2} \quad (3.47)$$

$$\frac{\partial a_{22}}{\partial u_2^2} = 2(1 + (u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_2})\frac{\partial \xi}{\partial a_2} \quad (3.48)$$

$$\frac{\partial a_{22}}{\partial u_1^3} = 2((u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_2})\frac{\partial \nu}{\partial a_2} \quad (3.49)$$

$$\frac{\partial a_{22}}{\partial u_2^3} = 2(1 + (u_2^2 - u_2^1)\frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1)\frac{\partial \nu}{\partial a_2})\frac{\partial \nu}{\partial a_2} \quad (3.50)$$

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_1^1} = & (-\frac{\partial \xi}{\partial a_1} - \frac{\partial \nu}{\partial a_1})((u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_2}) \\ & + (1 + (u_1^2 - u_1^1)\frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1)\frac{\partial \nu}{\partial a_1})(-\frac{\partial \xi}{\partial a_2} - \frac{\partial \nu}{\partial a_2}) \end{aligned} \quad (3.51)$$

³Falls der Nenner Null wird, handelt es sich bei $F^{tr}F$ um einen Kugeltensor.

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_1^2} &= \frac{\partial \xi}{\partial a_1} ((u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_2}) \\ &\quad + (1 + (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_1}) \frac{\partial \xi}{\partial a_2} \end{aligned} \quad (3.52)$$

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_1^3} &= \frac{\partial \nu}{\partial a_1} ((u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_2} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_2}) \\ &\quad + (1 + (u_1^2 - u_1^1) \frac{\partial \xi}{\partial a_1} + (u_1^3 - u_1^1) \frac{\partial \nu}{\partial a_1}) \frac{\partial \nu}{\partial a_2} \end{aligned} \quad (3.53)$$

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_2^1} &= (-\frac{\partial \xi}{\partial a_1} - \frac{\partial \nu}{\partial a_1}) (1 + (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2}) \\ &\quad + ((u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1}) (-\frac{\partial \xi}{\partial a_2} - \frac{\partial \nu}{\partial a_2}) \end{aligned} \quad (3.54)$$

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_2^2} &= \frac{\partial \xi}{\partial a_1} (1 + (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2}) \\ &\quad + ((u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1}) \frac{\partial \xi}{\partial a_2} \end{aligned} \quad (3.55)$$

$$\begin{aligned} \frac{\partial a_{12}}{\partial u_2^3} &= \frac{\partial \nu}{\partial a_1} (1 + (u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_2} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_2}) \\ &\quad + ((u_2^2 - u_2^1) \frac{\partial \xi}{\partial a_1} + (u_2^3 - u_2^1) \frac{\partial \nu}{\partial a_1}) \frac{\partial \nu}{\partial a_2}. \end{aligned} \quad (3.56)$$

Somit ergibt sich für die Ableitungen $\frac{\partial c_i}{\partial u_j^k}$ nach der Kettenregel:

$$\frac{\partial c_1}{\partial u_1^1} = \frac{\partial c_1}{\partial a_{11}} \frac{\partial a_{11}}{\partial u_1^1} + \frac{\partial c_1}{\partial a_{22}} \frac{\partial a_{22}}{\partial u_1^1} + \frac{\partial c_1}{\partial a_{12}} \frac{\partial a_{12}}{\partial u_1^1}. \quad (3.57)$$

Analog geht man für die anderen 11 Ableitungen vor. (3.57) läßt sich mit den Gleichungen (3.33)–(3.38) und (3.39)–(3.56) bestimmen.

3.8 Spezialfälle der Eigenwertberechnung

Bei der Berechnung der Eigenwerte von $F^{tr} F$ können einige Spezialfälle auftreten. Eine dieser Besonderheiten, nämlich der Fall, daß $F^{tr} F$ einen sogenannten Kugeltensor bildet, muß getrennt behandelt werden, da sonst die Ableitungen (3.33) bis (3.38) mathematisch nicht definiert sind. Es werden zudem noch drei andere Sonderfälle betrachtet, deren Unterscheidung zwar nicht zwingend notwendig, aber aus Effizienzgründen ratsam ist.

3.8.1 $F^{tr} F$ ist Kugeltensor

Ein Tensor der Dimension n (hier $n = 2$) wird als Kugeltensor bezeichnet, wenn er einen Eigenwert mit Vielfachheit n besitzt. Ist $F^{tr} F$ ein Kugeltensor, so hat $F^{tr} F$ die Gestalt

$$F^{tr} F = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} = aI. \quad (3.58)$$

Der Unterschied zwischen einem Kugeltensor und einem Tensor mit verschiedenen Eigenwerten läßt sich anschaulich über die Verzerrung des Einheitskreises verdeutlichen (Abbildung 3.7). Ein Kugeltensor verzerrt den Einheitskreis gleichmäßig in alle Richtungen. Daher ist insbesondere auch jeder Vektor ein Eigenvektor des Kugeltensors, wohingegen ein nicht-Kugeltensor zwei ausgezeichnete Eigenvektorrichtungen besitzt.

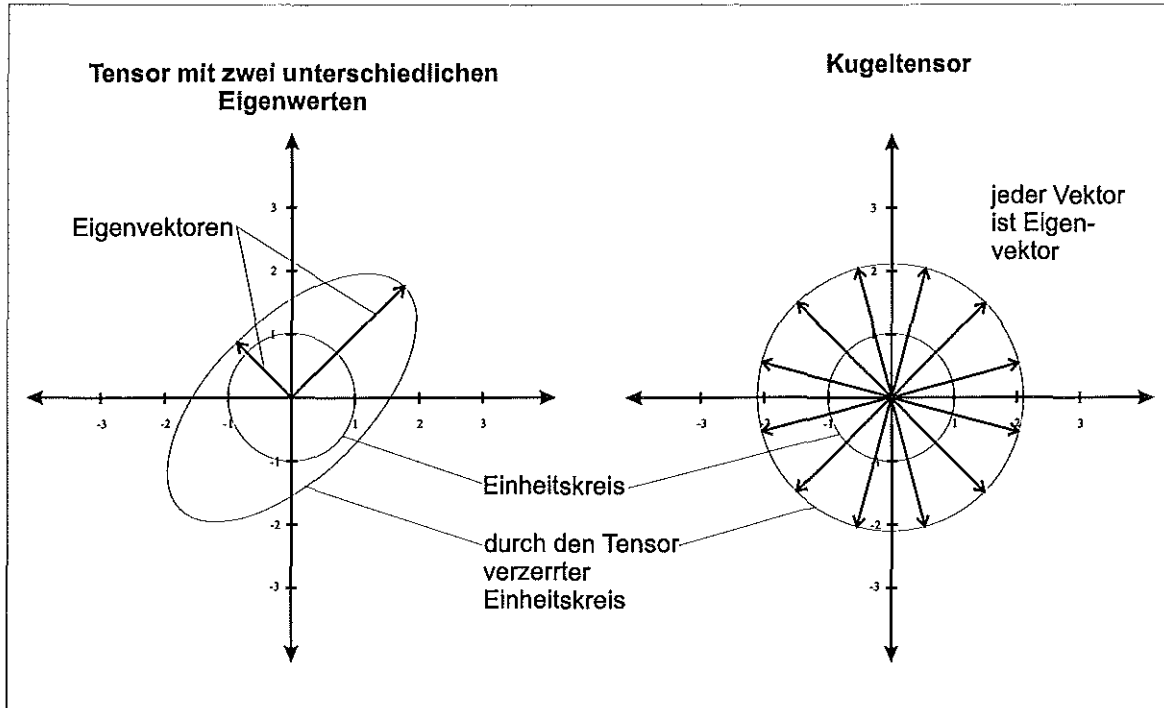


Abbildung 3.7: Ein Tensor mit zwei unterschiedlichen Eigenvektoren verzerrt den Einheitskreis unterschiedlich stark in die zwei Eigenvektorrichtungen (links). Ein Kugeltensor verzerrt den Einheitskreis gleichmäßig in alle Richtungen (rechts).

Wird die allgemeine Form der Eigenwertberechnung einer symmetrischen 2×2 Matrix betrachtet, wie es in (3.15) und (3.16) der Fall ist, so wird klar, wann die beiden Eigenwerte c_1 und c_2 gleich sind: bei Verschwinden des Ausdrucks $\sqrt{\frac{(a_{11}-a_{22})^2}{4} + a_{12}^2}$. Durch Umformung ergeben sich die Bedingungen, unter denen dies geschieht:

$$\begin{aligned}
 & \sqrt{\frac{(a_{11}-a_{22})^2}{4} + a_{12}^2} = 0 \\
 \Rightarrow & \frac{(a_{11}-a_{22})^2}{4} + a_{12}^2 = 0 \\
 \Rightarrow & \frac{(a_{11}-a_{22})^2}{4} = 0 \quad \wedge \quad a_{12}^2 = 0 \\
 \Rightarrow & a_{11} - a_{22} = 0 \quad \wedge \quad a_{12} = 0 \\
 \Rightarrow & a_{11} = a_{22} \quad \wedge \quad a_{12} = 0
 \end{aligned}$$

Diese Bedingungen zeigen, daß ein 2×2 Tensor genau dann ein Kugeltensor ist, wenn er die Gestalt (3.58) hat. Eine weitere Eigenschaft eines Kugeltensors $T = aI$ ist seine

Invarianz gegenüber einem Basiswechsel. Hat der Tensor bzgl. einer Basis B die Gestalt (3.58), und wird er mit den Basiswechselmatrizen ${}_{B'}W_B$ und ${}_BW_{B'}$ auf die Basis B' umgeschrieben, so ergibt sich:

$$\begin{aligned} {}_{B'}T_{B'} &= ({}_{B'}W_B)({}_BT_B)({}_BW_{B'}) \\ &= ({}_{B'}W_B)(aI)({}_BW_{B'}) \\ &= a({}_{B'}W_B)({}_BW_{B'}) \\ &= aI = {}_BT_B. \end{aligned}$$

Ist $F^{tr}F$ ein Kugeltensor, so läßt sich der Eigenwert direkt an der Matrix ablesen, und mit obigen Notationen ergibt sich $c_1 = c_2 = a$. Der Energieterm (3.30) vereinfacht sich zu

$$\begin{aligned} E &= \int_{\xi} \int_{\nu} \frac{1}{2} ((\sqrt{a} - 1)^2 + (\sqrt{a} + 1)^2) J \, d\nu d\xi \\ &= \int_{\xi} \int_{\nu} (\sqrt{a} - 1)^2 J \, d\nu d\xi. \end{aligned}$$

Die Ableitungen (3.33) bis (3.38) werden eingespart, und die Anzahl der Ableitungen (3.39) bis (3.56) reduziert sich von 18 auf 6.

3.8.2 $F^{tr}F$ ist Diagonalmatrix

Diese und die nachfolgend vorgestellte Optimierung der Eigenwertberechnung sind nicht zwingend nötig, um die Verzerrungsenergie korrekt berechnen zu können. Sie bieten sich jedoch an, da sich die Berechnung der Kräfte unter Berücksichtigung dieser Spezialfälle vereinfachen läßt.

Hat die Matrix $F^{tr}F$ bereits Diagonalgestalt, etwa

$$F^{tr}F = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}, \quad a_{11} \neq a_{22},$$

so gilt für die Eigenwerte c_1 und c_2 sofort $c_1 = a_{11}$ und $c_2 = a_{22}$. Somit kann auch in diesem Fall auf die Berechnung der Ableitungen $\frac{\partial c_i}{\partial a_{ij}}$, $i = 1, 2$, $(i, j) \in \{(1, 1), (1, 2), (2, 2)\}$, welche in (3.33) bis (3.38) dargestellt sind, verzichtet werden. Die Anzahl der Ableitungen (3.39) bis (3.56) verringert sich von 18 auf 12.

3.8.3 Winkel der Eigenvektoren ist 45° zur Koordinatenachse

Ein anderer Spezialfall ergibt sich, wenn $a_{11} = a_{22} = a$, aber anders als bei einem Kugeltensor $a_{12} \neq 0$ ist. Dann hat $F^{tr}F$ die Gestalt

$$F^{tr}F = \begin{pmatrix} a & a_{12} \\ a_{12} & a \end{pmatrix}.$$

Es läßt sich leicht zeigen, daß die zugehörigen zwei Eigenvektorräume im Winkel von 45° und -45° zur x -Achse stehen. Die Eigenvektorräume sind nämlich gegeben durch $\lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ und $\lambda \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, da

$$\begin{pmatrix} a & a_{12} \\ a_{12} & a \end{pmatrix} \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix} = (a + a_{12}) \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

und

$$\begin{pmatrix} a & a_{12} \\ a_{12} & a \end{pmatrix} \lambda \begin{pmatrix} 1 \\ -1 \end{pmatrix} = (a - a_{12}) \lambda \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

gilt. Für die Eigenwerte kann sofort $c_1 = a + a_{12}$ und $c_2 = a - a_{12}$ abgelesen werden. Damit vereinfachen sich die Ableitungen (3.33) bis (3.38), und die Ableitungen (3.39) bis (3.56) reduzieren sich von 18 auf 12 Stück.

3.9 Lokales Koordinatensystem

Die Berechnung der an den Dreieckselementen angreifenden Kräfte wird vereinfacht, indem jedes im dreidimensionalen Raum liegende Element in ein lokales, zweidimensionales Koordinatensystem eingebettet wird. Dies ist mit einer Transformation τ möglich. Zu beachten ist hierbei, daß die dem Element zugeordnete innere Verzerrungsenergie E unabhängig vom Koordinatensystem ist. Daher können die Kräfte, welche sich aus der Energie E ergeben, sowohl im lokalen als auch im globalen Koordinatensystem berechnet werden, ohne daß sich deren Betrag und deren Richtung relativ zum Element ändert. Um die im lokalen Koordinatensystem berechneten Kräfte im globalen System anzuwenden, d. h. mit den an anderen Elementen angreifenden Kräften kombinieren zu können, müssen sie lediglich mit der Umkehrtransformation τ^{-1} in das globale System übertragen werden. Dieser Vorgang ist in Abbildung 3.8 dargestellt.

Wird das lokale Koordinatensystem geschickt gewählt, so ergibt sich zudem eine vereinfachte Berechnung der notwendigen Transformation auf das Einheitsselement. Bevor darauf eingegangen wird, soll zunächst die Abbildung des globalen auf das lokale Koordinatensystem besprochen werden. Dazu soll ein Dreieckselement zum Zeitpunkt t mit den Punktkoordinaten $P_i(t) = (x_i(t), y_i(t), z_i(t))$, $i = 1, 2, 3$ betrachtet werden. Gesucht ist eine Transformation τ , welche die zu den drei Eckpunkten gehörenden dreidimensionalen Koordinaten auf zweidimensionale Koordinaten abbildet, also

$$\begin{aligned} \tau : \mathbb{R}^3 &\rightarrow \mathbb{R}^2 \\ P_i(t) = (x_i(t), y_i(t), z_i(t)) &\rightarrow P'_i(t) = (x'_i(t), y'_i(t)). \end{aligned}$$

Das zweidimensionale, lokale Koordinatensystem wird so gewählt, daß der Punkt P_0 im Nullpunkt und eine Achse des Koordinatensystems parallel zu der Seite $P_0(t) - P_1(t)$ des Elementes liegt. Daher gilt $P'_0(t) = (0, 0)$ und $P'_1(t) = (|P_0(t) - P_1(t)|, 0)$. Um die Koordinaten von $P'_2(t)$ zu berechnen, wird der Vektor $(P_2(t) - P_0(t))$ auf den Vektor $(P_1(t) - P_0(t))$ projiziert. Die Länge dieser Projektion gibt die Koordinate in x' -Richtung an. Um die Koordinate in y' -Richtung zu erhalten, wird der zu der Projektion senkrecht

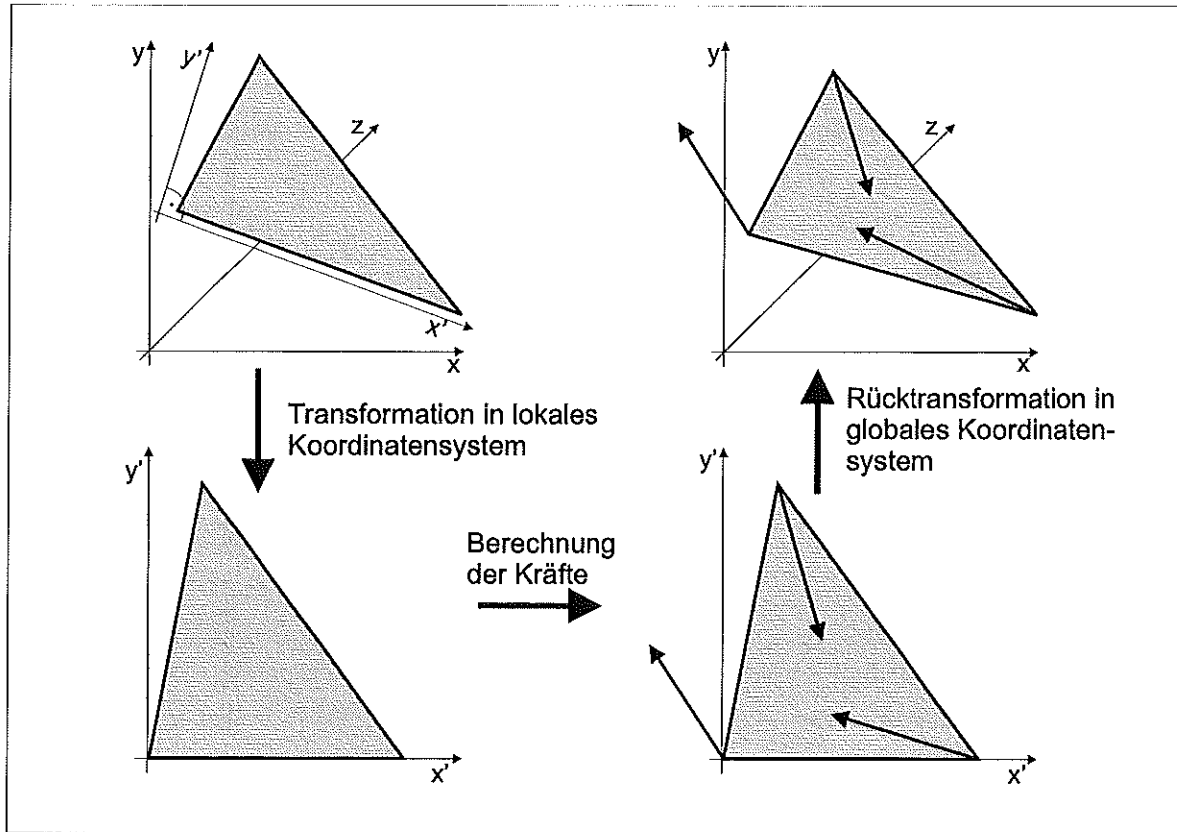


Abbildung 3.8: Anstatt die durch die Verzerrungsenergie implizit definierten Kräfte in globalen Koordinaten zu berechnen, kann diese Berechnung in einem zum Flächenelement lokalen System erfolgen. Für die Berechnung der Gesamtkräfte sind die lokalen Kräfte wieder in das globale System zurück zu transformieren.

stehende Anteil des Vektors $(P_2(t) - P_0(t))$ ermittelt. Die Einheitsvektoren $e_1(t)$, welcher parallel zu $(P_1(t) - P_0(t))$ liegt, und $e_2(t)$, welcher senkrecht zu $e_1(t)$ und in der Ebene des Dreieckselement liegt, berechnen sich durch

$$e_1(t) = \frac{P_1(t) - P_0(t)}{|P_1(t) - P_0(t)|} \quad (3.59)$$

$$e_2(t) = \frac{(P_2(t) - P_0(t)) - [(P_2(t) - P_0(t))^{\text{tr}} e_1(t)] e_1(t)}{|(P_2(t) - P_0(t)) - [(P_2(t) - P_0(t))^{\text{tr}} e_1(t)] e_1(t)|}. \quad (3.60)$$

Mit diesen Überlegungen ergibt sich

$$P'_2(t) = \begin{pmatrix} (P_2(t) - P_0(t))^{\text{tr}} e_1(t) \\ (P_2(t) - P_0(t))^{\text{tr}} e_2(t) \end{pmatrix}.$$

Die eigentliche Verzerrung wird nun durch die Verschiebung der Eckpunkte des Elementes relativ zu der unverzerrten Ausgangsform des Elementes berechnet, so wie sie zum Zeitpunkt $t = 0$ gegeben ist. Die unverzerrte Ausgangsform ist ein gleichschenkliges Dreieck. Dieses wird geometrisch ebenfalls so in das lokale Koordinatensystem gelegt, daß der

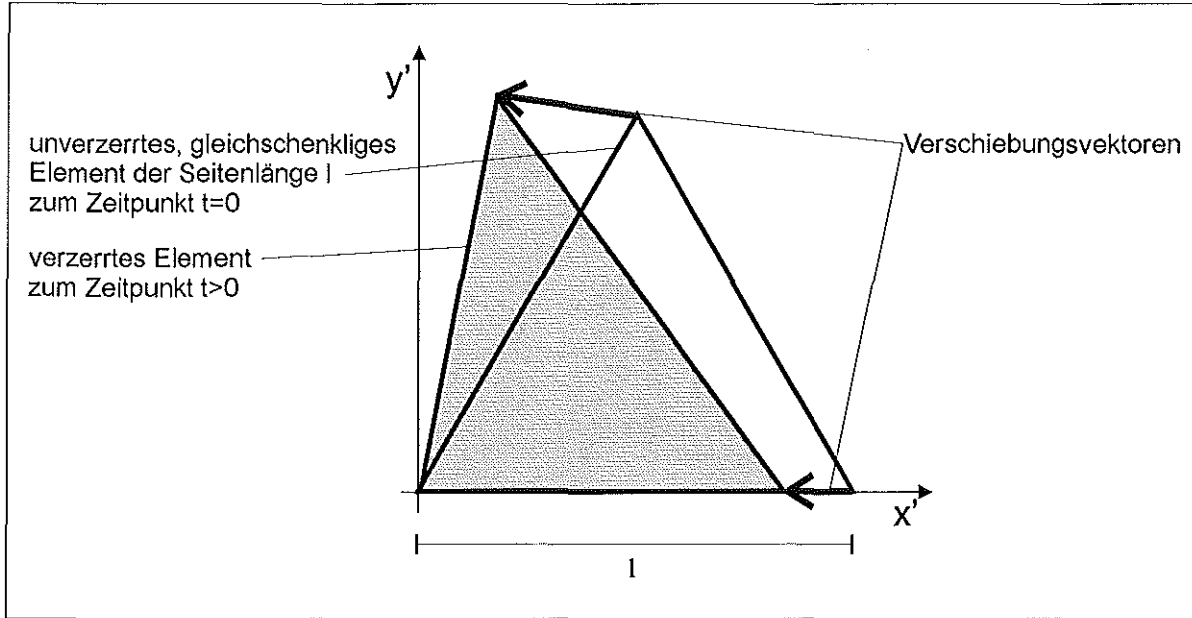


Abbildung 3.9: Das lokale Koordinatensystem für das Element zum Zeitpunkt $t = 0$ und $t > 0$ wird so gewählt, daß jeweils ein Punkt des Elementes im Ursprung und die x' -Achse parallel zu einer Elementseite liegt. Damit ergibt sich eine vereinfachte Darstellung der Verschiebungsvektoren.

Punkt $P_0(0)$ im Ursprung des lokalen Koordinatensystems und die x' -Achse parallel zur Linie $(P_1(0) - P_0(0))$ liegt, siehe dazu Abbildung 3.9.

Hat das unverzerrte Element die Seitenlänge l , so ergeben sich folgende Koordinaten:

$$\begin{pmatrix} P_0(0) \\ P_1(0) \\ P_2(0) \end{pmatrix} = \begin{pmatrix} (x'_0(0), y'_0(0)) \\ (x'_1(0), y'_1(0)) \\ (x'_2(0), y'_2(0)) \end{pmatrix} = \begin{pmatrix} (0, 0) \\ (l, 0) \\ (l/2, l/2\sqrt{3}) \end{pmatrix}.$$

Die Verschiebungsvektoren $u_0(t)$, $u_1(t)$ und $u_2(t)$ an den drei Punkten $P_0(0)$, $P_1(0)$ und $P_2(0)$ vereinfachen sich somit zu

$$\begin{pmatrix} u_0(t) \\ u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} (0, 0) \\ (x'_1(t) - l, 0) \\ (x'_2(t) - l/2, y'_2(t) - l/2\sqrt{3}) \end{pmatrix}.$$

Nach Berechnung der drei Kraftvektoren $k'_i(t) = (kx'_i(t), ky'_i(t))$, $i = 1, 2, 3$, müssen diese letztendlich wieder zurück in das globale, dreidimensionale Koordinatensystem transformiert werden. Die x' -Achse des lokalen Koordinatensystems wurde parallel zur Seite $(P_1(t) - P_0(t))$ gelegt. Die y' -Achse liegt senkrecht dazu, in der Ebene des Dreieckselementes. Daher ergibt sich mit den aus (3.59) und (3.60) gegebenen Definitionen folgende Rücktransformation:

$$k_i(t) = \begin{pmatrix} kx_i(t) \\ ky_i(t) \\ kz_i(t) \end{pmatrix} = kx'_i(t)e_1(t) + ky'_i(t)e_2(t).$$

Aufgrund der hier verwendeten speziellen Transformation auf ein lokales Koordinatensystem ergeben sich bei der Kraftberechnung zwei Vereinfachungen:

- Die bei der Transformation auf das Referenzelement auftauchende Ableitung $\partial\nu/\partial x'$ ist konstant Null.
- Die Verschiebung im Punkt P_0 ist für jeden Zeitpunkt t konstant Null, da der Ursprung des lokalen Koordinatensystems immer in diesem Punkt liegt.

3.10 Alternative Energieterme

Der in (3.30) definierte Energieterm spiegelt zunächst nur die Dehnung der Seiten eines Elementarquaders in Richtung der Hauptdehnungsachsen wieder. Der im Integral stehende Ausdruck $\frac{1}{2}((\sqrt{c_1(\xi, \nu)} - 1)^2 + (\sqrt{c_2(\xi, \nu)} - 1)^2)$ ist somit ein Maß für die Längenänderungen der Seiten eines im Dreieckselement an der Stelle (ξ, ν) liegenden infinitesimalen Quaders, dessen Seiten parallel zu den Eigenvektoren ausgerichtet sind. Integriert über die Oberfläche des Elementes ergibt sich die dem Element zugeordnete Gesamtverzerrungsenergie. Dieser Energieterm wurde in Abschnitt 3.7.4 vorgestellt, da ein Elementarquader eindeutig über die Längen seiner Seiten beschrieben wird. Somit führt schon die alleinige Verwendung von (3.30) dazu, daß sich die Elemente des Oberflächennetzes elastisch verhalten und bei fehlender äußerer Kraft in ihre Ausgangsform zurückkehren. Aus diesem Grund wurde dieser Energieterm bei der Implementierung des kontinuierlichen Modells verwendet.

In diesem Kapitel sollen zwei weitere Energieterme vorgestellt werden, die dazu gedacht sind, das Verhalten des Netzes bzgl. Eigenschaften zu steuern, welche nicht direkt mit der Längenänderung korrespondieren. Das sind einmal der Flächenerhalt und zum anderen der Winkelerhalt. Genau wie in Abschnitt 2.2.1 für das diskrete Modell beschrieben, gilt auch hier, daß zwar durch den Längenerhalt auch ein Flächen- und Winkelerhalt bedingt ist, aber nicht umgekehrt. Soll auf einen Längenerhalt (teilweise) verzichtet werden, nicht aber auf konstante Flächen und Winkel, so müssen die nun beschriebenen Energieterme verwendet werden.

3.10.1 Flächenerhalt

Die Flächenverzerrung eines zu den Hauptachsen parallel liegenden Elementarquaders läßt sich durch die Wurzeln der beiden Eigenwerte c_1 und c_2 beschreiben. Mit der Notation aus Kapitel 3.2 sind dies die Eigenwerte von $F^{tr}F$.⁴ Hat ein Elementarquader vor seiner Verzerrung den Flächeninhalt A , so beträgt der Flächeninhalt nach Verzerrung $\sqrt{c_1 c_2} A$. Der Flächeninhalt bleibt gleich, wenn $\sqrt{c_1 c_2} = 1$ ist. Die Differenz zwischen $\sqrt{c_1 c_2}$ und 1 ist ein Maß für die Änderung des Flächeninhaltes. Man beachte, daß $\sqrt{c_1 c_2}$ gerade die Determinante der in Kapitel 3.6.4 vorgestellten Matrix U ist. Da $F = RU$ ist mit $\det(R) = 1$ ⁵, folgt sofort $\det(F) = \det(R)\det(U) = \det(U)$. Dies ist auch zu erwarten, da die Determinante des Deformationsgradienten F ein Maß für die Flächenverzerrung

⁴ F ist der Deformationsgradient.

⁵ R ist eine eigentlich orthogonale Matrix, also gilt $\det(R)=1$.

ist. Daraus läßt sich schließen, daß nicht der Umweg über die Berechnung der Matrix U gegangen werden braucht, wenn ein Energieterm gewünscht ist, in welchen ausschließlich die Flächenänderung eingehen soll. Soll auch die Längenänderung berücksichtigt werden, so müssen ohnehin die Eigenwerte der Matrix U bestimmt werden, welche dann auch verwendet werden können, um die Flächenänderung zu berechnen. Als Energieterm, welcher die Flächenänderung mißt, ergibt sich mit diesen Überlegungen:

$$\begin{aligned} E_{fl} &:= \int_V \frac{1}{2} (\sqrt{c_1(x,y)c_2(x,y)} - 1)^2 dV \\ &= \int_V \frac{1}{2} (\det(F)(x,y) - 1)^2 dV. \end{aligned}$$

3.10.2 Winkelerhalt

Wie in Kapitel 3.6.4 bereits beschrieben, gibt es in Richtung der Hauptachsen keine Winkeländerung (Schiebung). Linienelemente, welche nicht entlang der Hauptachsen liegen, werden allerdings gedreht, und zwar in Richtung des Eigenvektors mit dem größten Eigenwert. Nur wenn der Tensor $F^{tr}F$ ein Kugeltensor ist, dann ist jeder Vektor auch Eigenvektor, und es erfolgt keine Winkeländerung. Wie in Abschnitt 3.8.1 erläutert, ist dies dann der Fall, wenn die Eigenvektoren c_1 und c_2 (bzw. $\sqrt{c_1}$ und $\sqrt{c_2}$ für die Matrix U) gleich sind. Dies wird über den Ausdruck $|\sqrt{c_1} - \sqrt{c_2}|$ gemessen. Dieser Term geht quadratisch in die Energie ein, und es ergibt sich somit

$$\begin{aligned} E_{wi} &:= \int_V \frac{1}{2} |\sqrt{c_1(x,y)} - \sqrt{c_2(x,y)}|^2 dV \\ &= \int_V \frac{1}{2} (\sqrt{c_1(x,y)} - \sqrt{c_2(x,y)})^2 dV \end{aligned}$$

als Maß für die Winkeländerungen im Gesamtelement.

3.10.3 Gesamtenergie

Die Terme für den Längen-, Flächen- und Winkelerhalt können zu einer Gesamtenergie addiert werden:

$$\begin{aligned} E_{ges} &:= \int_V \frac{1}{2} [\lambda_{le} \left((\sqrt{c_1(x,y)} - 1)^2 + (\sqrt{c_2(x,y)} - 1)^2 \right) + \\ &\quad \lambda_{fl} \left(\sqrt{c_1(x,y)c_2(x,y)} - 1 \right)^2 + \\ &\quad \lambda_{wi} \left(\sqrt{c_1(x,y)} - \sqrt{c_2(x,y)} \right)^2] dV. \end{aligned}$$

Hierbei sind λ_{le} , λ_{fl} und λ_{wi} Wichtungsfaktoren für die Längen-, Flächen- und Winkelmaße.

Kapitel 4

Äußere Kraft

In den vorherigen Kapiteln wurden nur die inneren Netzkräfte behandelt, d. h. es wurde beschrieben, mit welchen Kräften das Netz auf eine Deformation reagiert. Da das Netz elastische Eigenschaften aufweist, behält es seine Ausgangsform, wenn keine äußere Kraft an ihm angreift. Wie z. B. in Gleichung (3.6) kenntlich wird, befinden sich die inneren und äußeren Kräfte bei einem Körper immer im Gleichgewicht. Ziel der Anwendung von aktiven Oberflächen ist es, eine durch einen 3D Datensatz implizit vorgegebene Oberfläche zu finden. Daher ist die äußere Kraft so zu definieren, daß das Kräftegleichgewicht möglichst genau dann erreicht wird, wenn die gesuchte Oberfläche gefunden wurde. Bildlich gesprochen muß eine Kraft wirken, welche das Oberflächennetz an die Oberflächenpunkte des im Datensatz liegenden Objektes heranzieht bzw. herandrückt. Wie auch im Fall der inneren Kräfte können physikalische Analogien für diese Vorgänge gefunden werden, wie im weiteren gezeigt wird. Es werden im folgenden zwei unterschiedliche Möglichkeiten aufgeführt, wie äußere Kräfte definiert werden können.

Das Thema dieses Kapitels ist nur die Definition einer äußeren Kraft von Bildpunkten auf Netzknoten. Auf die Vorverarbeitung des Volumendatensatzes wird hier nicht eingegangen. Stattdessen wird an dieser Stelle vorausgesetzt, daß die Objektgrenzschicht, an welche sich das Oberflächennetz anpassen soll, klar definiert ist. D. h. es wird vorausgesetzt, daß eine Klassifikation der Bildpunkte bereits stattgefunden hat. Für die Definition der ersten Kraft, welche auf einer euklidischen Abstandstransformation beruht, reicht eine Einteilung in Randpunkte und Nicht-Randpunkte. Für die Definition der Druckkraft muß eine Einteilung in Randpunkte sowie in innere und äußere Objektpunkte vorliegen. Ferner sei angenommen, daß die im Datensatz vorliegenden Voxel isotrope sind, d. h. daß Voxelhöhe, -breite und -tiefe gleich sind. Aus einem nicht-isotropen Datensatz kann durch Interpolation immer ein isotroper Datensatz gewonnen werden. Als Notation wird hier verwendet:

- $I(x, y, z) \in [0, \dots, 2^n - 1]$: Wert des Voxels an Position x, y und z . Die Intensitäten der Voxel werden durch n Bit kodiert.
- $R = \{(x, y, z) : \text{Voxel an Position } (x, y, z) \text{ ist Randpunkt}\}$: Die Menge der Objektrandpunkte.

4.1 Abstandsasierte Kraft

Hier wird eine Kraft definiert, welche für eine Minimierung des Abstandes zwischen Netzknoten und Objektpunkten sorgt. Um diese Kraft zu berechnen, wird im ersten Schritt ein dem Datensatz zugeordnetes Energiepotential $\Theta(X)$, $X = (x, y, z)$ erzeugt. Dieses Potential weist jedem Datenpunkt (Voxel) X einen Wert zu, in welchen die Abstände dieses Voxels zu den Objektpunkten eingehen. Allgemein ergibt sich

$$\Theta(x, y, z) = \sum_{(x', y', z') \in R} f(|(x, y, z) - (x', y', z')|). \quad (4.1)$$

Mit der Funktion $f(x)$ ($x \simeq$ Abstand) kann Einfluß genommen werden, wie der Abstand in die Energie Θ eingeht. Wird auf das physikalische Gesetz der Abstoßung zweier Körper zurückgegriffen, so ist die Energie umgekehrt proportional zum Abstand, also $f(x) = \frac{1}{x}$. Die Massen der Punkte sind hier willkürlich auf Eins gesetzt. Als Anziehungskraft K ergibt sich somit

$$K(x, y, z) = \begin{pmatrix} \frac{\partial \Theta(x, y, z)}{\partial x} \\ \frac{\partial \Theta(x, y, z)}{\partial y} \\ \frac{\partial \Theta(x, y, z)}{\partial z} \end{pmatrix} \approx \begin{pmatrix} \Theta(x+1, y, z) - \Theta(x, y, z) \\ \Theta(x, y+1, z) - \Theta(x, y, z) \\ \Theta(x, y, z+1) - \Theta(x, y, z) \end{pmatrix}.$$

Die Kraft ist oben nur für ganzzahlige x, y und z definiert. Sie greift aber an den Netzknoten an. Die Koordinaten der Netzknoten werden im Rechner durch Fließkommazahlen dargestellt. Daher wird ein Netzknoten im allgemeinen nicht genau auf einem Gitterpunkt des Voxelraumes liegen. Die an dem Netzknoten angreifende Kraft muß daher interpoliert werden. Dies kann über eine trilineare Interpolation über die Kräfte geschehen, die an den sechs den Netzknoten einschließenden Gitterpunkten angreifen ([Coh91]). Siehe dazu Abbildung 4.1 für den zweidimensionalen Fall (bilineare Interpolation).

Die so definierten äußeren Kräfte werden im allgemeinen nicht senkrecht zur Oberflächennormalen des Oberflächennetzes sein. Der Kraftanteil, der zur Netzoberfläche tangential ist, bewirkt eine Verschiebung der Netzknoten innerhalb der Netzoberfläche. Dies ist häufig unerwünscht, da sich die tangentielle Kraftkomponente ungünstig auf die gewünschten Eigenschaften wie Längen- und Flächenerhalt auswirkt. Zudem ist wegen des sogenannten Blendenproblems [Jäh97] der tangentielle Anteil eines Verschiebungsvektors unbestimmt, wenn der Verschiebungsvektor aufgrund eines lokalen Operators bestimmt wurde. Aus diesen Gründen ist es sinnvoll, die Kraft nur in Richtung der Oberflächennormalen $N(x, y, z)$ wirken zu lassen. Somit ergibt sich eine Kraft $K'(x, y, z)$ gemäß:

$$K'(x, y, z) = N(x, y, z)(K(x, y, z)^{tr} N(x, y, z)).$$

4.2 Problematik der abstandsasierten Kraft

Die aus den obigen Abstandsbetrachtungen gewonnene Kraft hat drei Nachteile. Dabei ist ein Nachteil genereller Natur, nämlich der große Rechenaufwand, und ein Nachteil ergibt sich durch die in Kapitel 6 vorgestellte Anwendung.

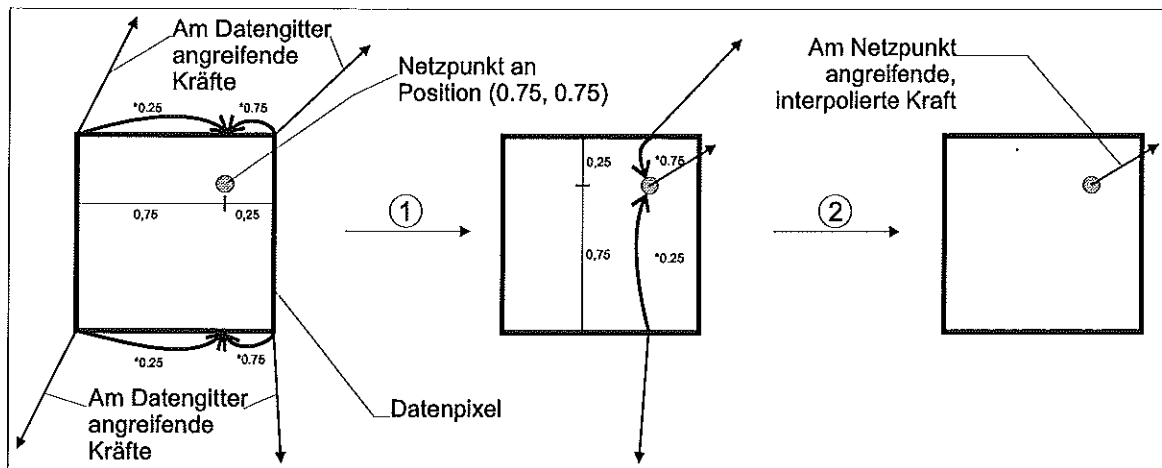


Abbildung 4.1: Bilineare Interpolation: der abgebildete Netzpunkt liegt im Datenpixel an Position (0.75, 0.75). Daher werden die an den Pixelecken angreifenden Kräfte in zwei Schritten bilinear interpoliert.

Rechenaufwand

Wie aus Gleichung (4.1) hervorgeht, wird zur Berechnung des Energiepotentials der Abstand eines jeden Voxels zu allen Objektrandpunkten benötigt. Bei einem Datensatz bestehend aus n^3 Punkten und m Randpunkten liegt der Aufwand des Verfahrens in $\mathcal{O}(n^3m)$. Die Randpunktkoordinaten sind aber nicht explizit gegeben. Sie liegen vielmehr im Datensatz vor und können durch ihre Klassifizierung von anderen Punkten unterschieden werden. Um sie explizit aufzuführen, muß in einem ersten Verarbeitungsschritt der Datensatz komplett durchlaufen werden, um die Koordinaten der Randpunkte in einer Liste explizit zu speichern. Das führt wiederum zu einem hohen Rechen- und, falls die Menge der Randpunkte groß ist, Speicheraufwand. Der Gesamtrechenaufwand beläuft sich damit auf $\mathcal{O}(n^3(m+1))$.

Der Aufwand läßt sich verringern, wenn weit entfernte Randpunkte bei der Potentialberechnung vernachlässigt werden. Dieses Vorgehen wird gerechtfertigt, da $1/x$ für große x gegen Null strebt. Es kann ein dreidimensionales Suchfenster der Größe r^3 um jedes Voxel eingerichtet werden. Nur in dem Suchfenster wird nach Randpunkten gesucht. Als Aufwand ergibt sich damit $\mathcal{O}(n^3r^3)$. Je kleiner das Suchfenster, desto geringer ist der Aufwand. Trotzdem ist der Aufwand auch bei einem kleinen Suchfenster (in der Größenordnung 8^3 Voxel) hoch.

Schnelle Kraftabnahme

Werden die Gesetze der Physik verwendet, so ist die Kraft, mit welcher sich zwei Punkte mit Abstand x im Raum anziehen, proportional zu $1/x^2$. Sie nimmt mit größer werdendem Abstand also schnell ab. Da zudem bei der Kraftberechnung, wie oben beschrieben, weit entfernte Punkte durch die Verwendung eines Suchfensters vernachlässigt

werden, kann es im Volumendatensatz Regionen geben, in denen die Kraft verschwindet. Die in diesen Regionen liegenden Netzpunkte sind somit keiner direkten Anziehung ausgesetzt. Sie können sich nur mittelbar auf das gesuchte Objekt zu bewegen, indem sie (durch die inneren Kräfte gesteuert) Punkten folgen, an denen Anziehungskräfte vorliegen. Im Extremfall, daß das Oberflächennetz sehr schlecht initialisiert ist und alle Netzpunkte weit von Objektrandpunkten entfernt liegen, kann es sogar vorkommen, daß kein Netzpunkt von einer Anziehungskraft erfaßt wird.

Lokale Maxima

Ist das im Datensatz gesuchte Objekt konkav, dann kann es im Energiepotential zu lokalen Maxima (Abbildung 4.2) kommen. Im lokalen Maximum verschwindet die Kraft, die am Netz angreift. Netzpunkte bleiben in lokalen Maxima hängen.

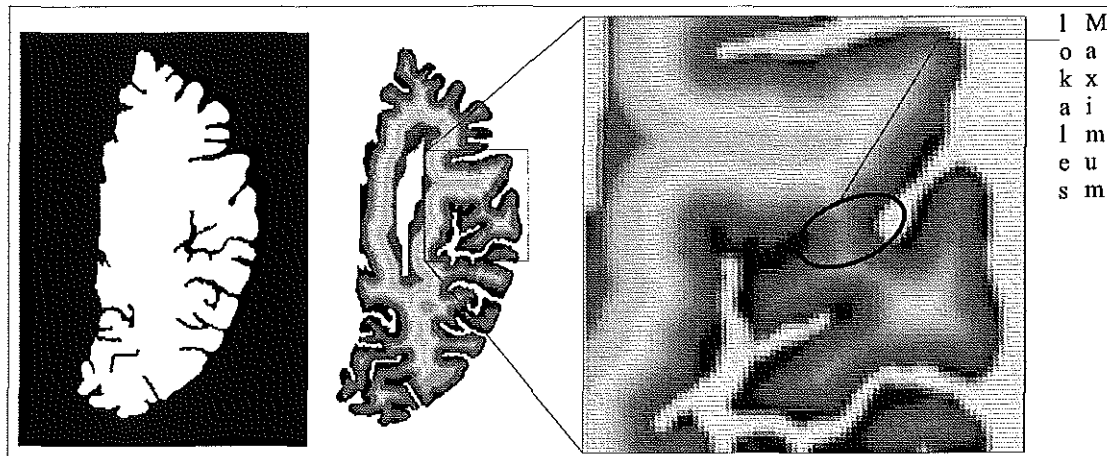


Abbildung 4.2: Liegt eine abstands-basierte Kraft zugrunde, so kommt es an vielen Stellen des Datensatzes zu lokalen Maxima, an denen das Oberflächennetz während des Deformationsprozesses hängen bleibt. Links ist das Ausgangsbild dargestellt, das Gehirnninnere in Weiß und das Gehirnäußere in Schwarz. In der Mitte ist das Abstandspotential in Graustufen dargestellt. In der Ausschnittsvergrößerung, rechts, wurde ein lokales Maximum markiert.

Dies ist insbesondere für die in Kapitel 6 vorgestellte Anwendung nachteilig. Das hier gesuchte Objekt, die Oberfläche der Großhirnrinde, ist durch seine vielfältige Faltung extrem konkav geformt. An vielen Stellen des Datensatzes kommt es daher bei der Berechnung des Potentials zur Bildung von lokalen Extremalstellen. Das Oberflächennetz bleibt bei Anwendung der abstands-basierten Kraft an vielen Stellen in den Falten hängen. Die äußere Oberfläche des Gehirns kann nicht gefunden werden, zumindest nicht, wenn die abstands-basierte Kraft als alleinige äußere Kraft genommen wird.

4.3 Druckkraft

Wegen des Blendproblems und der Forderung nach gleichmäßigem Punktabstand ist es sinnvoll, die äußeren Kräfte senkrecht zur Netzoberfläche angreifen zu lassen. Die bei der Anziehungskraft auftretende Problematik der lokalen Extrema kann beseitigt werden, indem eine konstante Kraft in Richtung des Normalenvektors N der Netzoberfläche definiert wird. Diese Kraft entspricht einer Druckkraft. Sie hängt nicht von den Abständen zwischen Netz- und Oberflächenrandpunkten ab, sondern drückt die Netzpunkte in Richtung gesuchte Objektoberfläche. Eine an einem Netzpunkt P angreifende Druckkraft D mit konstantem Betrag c wird definiert über

$$D(P) = \begin{cases} cN(P), & \text{falls } P \text{ innerhalb des Objektes liegt,} \\ -cN(P), & \text{falls } P \text{ außerhalb des Objektes liegt,} \\ 0, & \text{falls } P \text{ auf dem Objektrand liegt.} \end{cases}$$

$N(P)$ bezeichnet hier die nach außen gerichtete Oberflächennormale am Punkt P .

Um die Kraft zu bestimmen, muß also die bereits oben erwähnte Klassifizierung der Datenvoxel in innere, äußere und Randvoxel vorliegen. Netzpunkte innerhalb des Objektes werden nach außen und Netzpunkte außerhalb des Objektes werden nach innen gedrückt. Es ist also sichergestellt, daß die Netzpunkte in Richtung Objektrand bewegt werden. Der Objektrand ist durch die Menge der Randvoxel gegeben. Somit ist dieser so definierte Rand keine Oberfläche im eigentlichen Sinne, sondern vielmehr eine Menge von kleinen Volumenelementen. Es wurde bis hierhin noch nicht gesagt, was darunter zu verstehen ist, daß ein 'Netzpunkt auf dem Objektrand liegt', also wann genau die Druckkraft an einem Netzpunkt auf Null gesetzt wird. Es bieten sich zwei Definitionen an:

1. Es kann eine im betrachteten Randvoxel liegende Oberfläche als Berandung konstruiert werden. Diese Randoberfläche soll das Voxel volumenmäßig in zwei gleich große Hälften aufteilen. Da der wahre Verlauf der Objektoberfläche nicht bekannt ist, orientiert sich der konstruierte Rand an der Netzoberfläche. Er wird senkrecht zur Normalen der Netzoberfläche definiert. Es kann also im Datenvoxel eine mathematisch genau definierte Fläche konstruiert werden, die das Voxel in gleiche Hälften teilt, und welche tangential zur Netzoberfläche verläuft. Mit dieser 'Hilfsfläche' kann das Innere und Äußere des Objektes auch innerhalb des Randvoxels beschrieben werden. Ein Netzpunkt liegt dann auf dem Rand, wenn er zu dieser Fläche einen Abstand $< \varepsilon$ hat. Das Verfahren ist in Abbildung 4.3 illustriert.
2. Eine einfach zu implementierende Lösung besteht darin, daß das ganze Randvoxel als Objektrand angesehen wird. Ein Netzpunkt, der an einer beliebigen Stelle *innerhalb* eines Randvoxels liegt, erfährt somit keine Druckkraft.

Der Nachteil des ersten Verfahrens ist sein großer Rechenaufwand. Für jeden in einem Randvoxel liegenden Netzpunkt muß die oben beschriebene Randfläche konstruiert werden. Dies hat in jedem Iterationsschritt neu zu geschehen, da die Form der Netzoberfläche sich ändert. Da die wirkliche Objektoberfläche unbekannt ist, ist durch dieses Verfahren keine merkliche Verbesserung der Oberflächenfindung zu erwarten. Der Mehraufwand an Rechenzeit gegenüber dem zweiten Verfahren ist also nicht gerechtfertigt. Zudem führt

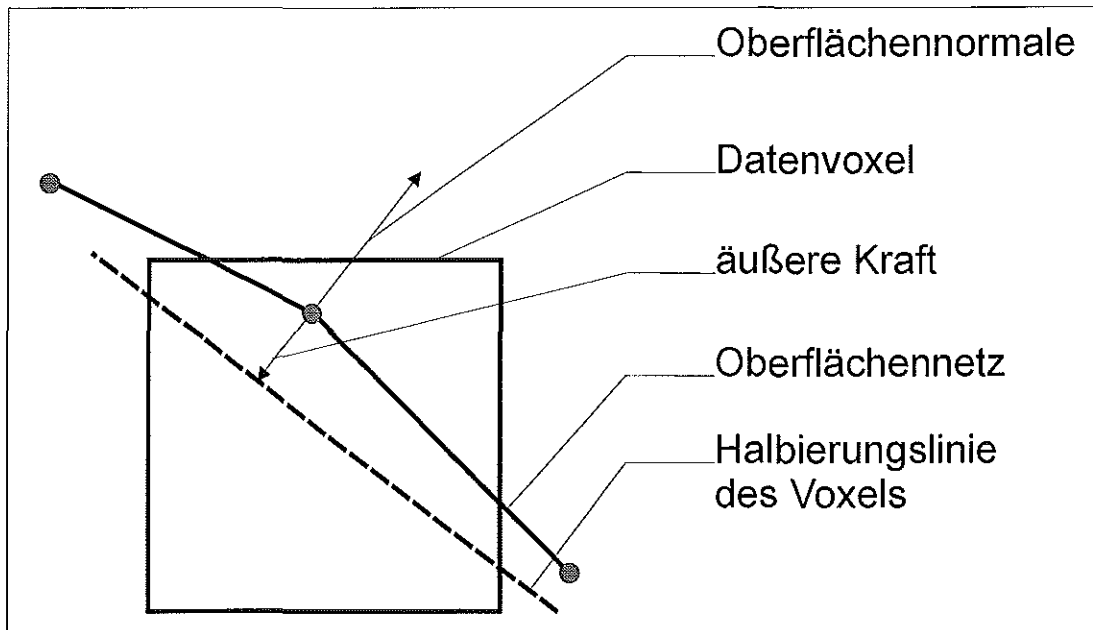


Abbildung 4.3: Die an einem Netzknoten angreifende Druckkraft zieht ihn in Richtung einer zum Netz parallelen Fläche, welche das Randvoxel in zwei gleich große Hälften teilt.

das Fehlen einer äußeren Kraft innerhalb eines Randvoxels dazu, daß sich Netzknoten innerhalb eines Voxels gemäß den inneren Kräften bewegen können. Da durch die inneren Kräfte gewünschte Netzeigenschaften vorgegeben werden, können erwünschte Eigenschaften wegen der Möglichkeit, daß sich ein Netzknoten innerhalb eines Randvoxels alleine gemäß der inneren Kräfte bewegen kann, leicht optimiert werden.

4.4 Netzüberschneidungen

An Stellen, an denen Grenzflächen im Datensatz aneinanderliegen, und die nicht (manuell) durch eine künstliche Grenzlinie getrennt werden, kann es zu Regionen kommen, an denen das Netz sich selber überschneidet (Abbildung 4.4). Eine einmal stattgefunden Überschneidung verstärkt sich im Laufe der Berechnung, da die Druckkraft das Netz immer weiter nach außen drückt.

Um dies abzufangen, muß

1. die Überschneidung detektiert und
2. die Überschneidung rückgängig gemacht werden.

Um zu erkennen, ob die durch die Punkte P_0 und P_1 gegebene Dreiecksseite ein anderes, durch die Punkte P'_0 , P'_1 und P'_2 definiertes Dreieck schneidet, ist folgende Gleichung nach t , u und v aufzulösen:

$$P_0 + t(P_1 - P_0) = P'_0 + u(P'_1 - P'_0) + v(P'_2 - P'_0).$$

Eine Überschneidung hat dann stattgefunden, wenn obige Gleichung eine eindeutige Lösung hat mit $0 \leq t \leq 1$, $u \geq 0$, $v \geq 0$ und $u + v \leq 1$.

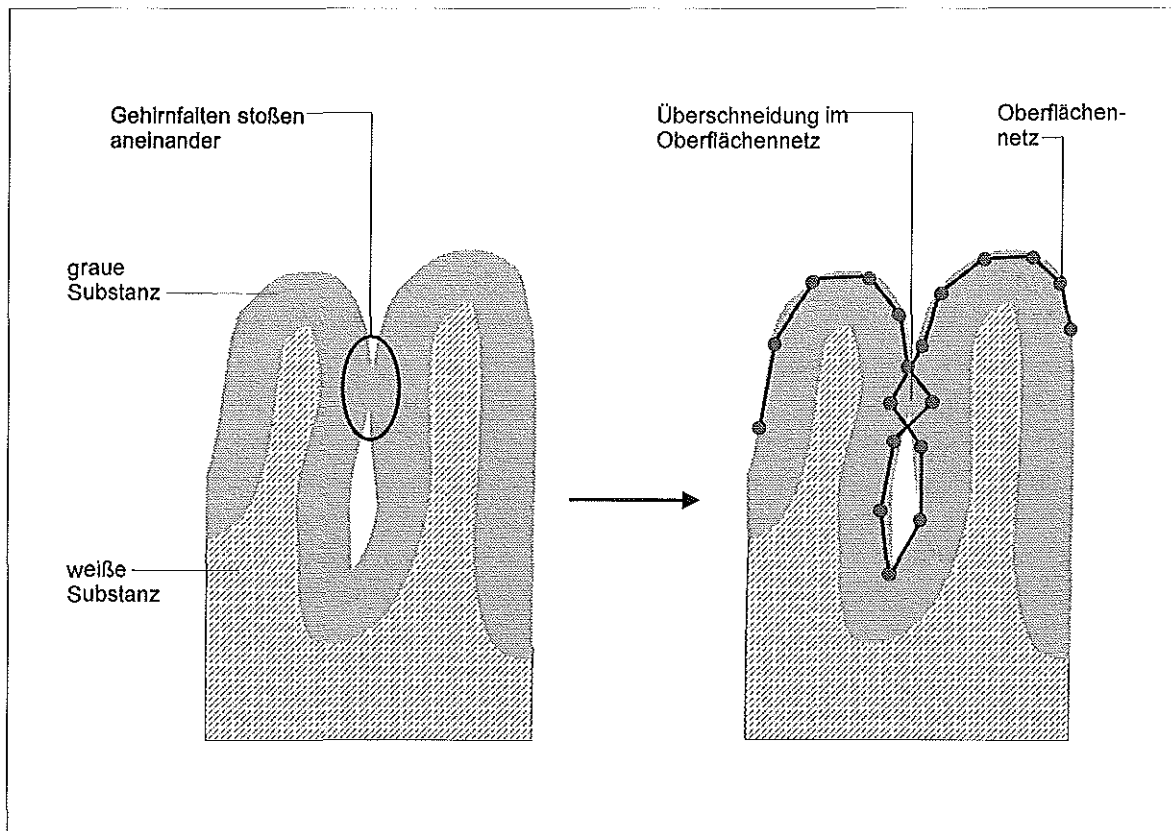


Abbildung 4.4: An Stellen, an denen sich Gehirnfalten berühren (links), kann es zur Überschneidung des Oberflächennetzes kommen (rechts).

Die Detektion von Überschneidungen ist sehr rechenintensiv und kann nicht lokal auf dem Netz geschehen, da Elemente, die auf dem Netz weit entfernt voneinander liegen, durch die Faltung des Netzes im Raum dicht benachbart sein können. Damit nicht jedes Element auf Überschneidung mit jedem anderen Element überprüft werden muß, kann eine Unterteilung des Netzes aufgrund der Lage der Netzpunkte im Raum vorgenommen werden. Das Datenvolumen wird dazu in Volumenquader der Seitenlänge l aufgeteilt. Ein Quader Q_{ijk} bezeichne das (dreidimensionale) Intervall $[(i-1)l, il] \times [(j-1)l, jl] \times [(k-1)l, kl]$. Die Seitenlänge l der Quader wird gerade so groß wie die maximale Seitenlänge unter allen Dreieckselementen gewählt. Dann kann eine Dreiecksseite, von der ein Punkt im Quader Q_{ijk} liegt, nur eine Dreiecksfläche schneiden, die wenigstens einen Punkt in Q_{ijk} oder in den Q_{ijk} umgebenden 26 Quadern hat. Die Anzahl der auf Überschneidung zu testenden Elemente wird dadurch stark eingeschränkt. Der Rechenaufwand bleibt aber immer noch hoch.

Wurde eine Überschneidung festgestellt, so kann diese Überschneidung rückgängig gemacht werden, indem das Oberflächennetz an der Stelle der Überschneidung in einem iterativen Prozeß geglättet wird. An der Glättung nehmen alle Punkte teil, die Randpunkte eines an der Überschneidung beteiligten Elementes sind. Durch diese lokale Glättung zieht sich das Netz an der Stelle der Überschneidung wieder aus dem kritischen Bereich zurück.

4.5 Einbettung in das finite Elemente Konzept

In den obigen Abschnitten wurden äußere Kräfte definiert, welche an Punkten im Voxerraum angreifen. Diese Kräfte wirken an den Netzpunkten, indem sie diese in Richtung Kraft verschieben. Für das in Kapitel 2.2 definierte diskrete Oberflächennetz kann diese Verschiebung direkt erfolgen, da die das Netz bestimmenden Parameter die Koordinaten der Netzpunkte sind.

Wird die finite Elemente Methode zur Beschreibung der aktiven Oberfläche verwendet, so können je nach verwendeten Ansatzfunktionen für die Verschiebung auch Parameter, die nicht die Punktkoordinaten sind, in der Beschreibung des Netzes auftauchen. Diese Parameter sind z. B. partielle Ableitungen an den Netzpunkten und werden auch Knotenparameter genannt. Dann müssen sogenannte verallgemeinerte Kräfte berechnet werden, die eine Veränderung der Knotenparameter zur Folge haben. Des weiteren ist die Lage der Oberfläche des aktiven Konturmodells bei Verwendung von finiten Elementen nicht nur an den Netzpunkten selber, sondern über die parametrisierten Ansatzfunktionen überall bekannt. Somit macht es Sinn, äußere Kräfte nicht nur an den Knotenpunkten selber, sondern an der ganzen Netzoberfläche angreifen zu lassen. Die an einem Netzpunkt angreifende Gesamtkraft ergibt sich aus der Integration über die Oberflächenelemente, die an diesem Punkt anliegen. Die Integration erfolgt numerisch über Stützstellen auf der Netzoberfläche. Als verallgemeinerter, an den Knotenparametern des Elementes i angreifender Kraftvektor b_i ergibt sich mit dem Vektor der Verschiebungsfunktionen $N(x, y)$ somit [DT85]:

$$b_i = \int_{E_i} N(x, y) f(x, y) dx dy.$$

4.6 Minimierung der Energie

Das in Kapitel 1 vorgestellte aktive Oberflächenmodell wird in [CC93] über die in Kapitel 1.2.1 vorgestellte finite Differenzen Methode diskretisiert und ist daher über ein lineares Gleichungssystem beschreibbar.

Eine weitere Methode, ein Variationsproblem zu lösen, ist die dynamische Programmierung ([AWJ90]). Die dynamische Programmierung sucht das globale Energieminimum, indem alle Netzpunkte in einem Suchfenster verschoben und die daraus resultierenden Energien berechnet werden. Aufgrund der hohen Zeitkomplexität, welche in $\mathcal{O}(nm^2)$ liegt, und des großen Speicherbedarfs von $n \times m$, wobei n die Netzpunktzahl und m die Anzahl an Suchrichtungen ist, scheidet diese Methode hier jedoch aus.

Die in dieser Arbeit vorgestellten Modelle, nämlich das direkt diskrete Oberflächenmodell aus Abschnitt 2.2 und das auf kontinuumsmechanischen Grundlagen basierende Modell aus Kapitel 3.2, sind nicht linear. Eine Lösung kann durch Linearisierung mittels der in Kapitel 3.1.5 vorgestellten modifizierten Newton-Methode gewonnen werden.

Von dieser Vorgehensweise muß hier jedoch aus zwei Gründen abgesehen werden: zum einen ist die Tangentensteifigkeitsmatrix, welche die Linearisierung des Systems beschreibt, sehr groß. Konkret wurden Netze mit bis zu 40962 Knotenpunkten verwendet, wobei für jeden Knotenpunkt drei Koordinatenwerte gespeichert werden müssen. Die daraus resultierende Matrix hat eine Größe von $(40962 * 3) \times (40962 * 3)$ Einträgen. Obwohl

sie dünn besetzt ist, ist sie auf Grund ihrer Größe nur schwer handhabbar.

Ein weiterer Grund, das Gleichgewicht zwischen inneren und äußeren Kräften nicht direkt über die Invertierung der Tangentensteifigkeitsmatrix zu bestimmen, ergibt sich aus der Tatsache, daß die äußeren Kräfte nicht konstant sind, sondern von der Lage des Netzes im Raum abhängen. Es liegt somit ein System der Form

$$Au = f(u) \quad (4.2)$$

vor, wobei A die (Tangenten-) Steifigkeitsmatrix, u der Vektor der Knotenparameter und $f(u)$ der Vektor der an den Knotenparametern angreifenden verallgemeinerten Kräfte ist. Statt (4.2) wird daher die zugehörige sogenannte *Evolutionsgleichung* verwendet. Dazu wird in Gleichung (4.2) ein Geschwindigkeitsterm $\partial u / \partial t$ eingeführt, und zwar als finite Differenz über die Zeit t :

$$\frac{u_{t+1} - u_t}{\tau} + Au_{t+1} = f(u_t). \quad (4.3)$$

τ ist ein diskreter Zeitschritt. Wird τ hinreichend klein gewählt, so ist die in einem Iterationsschritt vorgenommene Änderung des Parametervektors u sehr klein, so daß $u_{t+1} \approx u_t$ gilt. Unter dieser Voraussetzung kann (4.3) ohne die Inversion der Matrix A nach u_{t+1} aufgelöst werden:

$$u_{t+1} = u_t + \tau(f(u_t) - Au_t). \quad (4.4)$$

Anschaulich kann (4.4) so gedeutet werden, daß die Netzknoten einen kleinen Schritt τ weit in die Richtung der Summe aus inneren und äußeren Kräften bewegt werden. Dies ist in Abbildung 4.5 dargestellt.

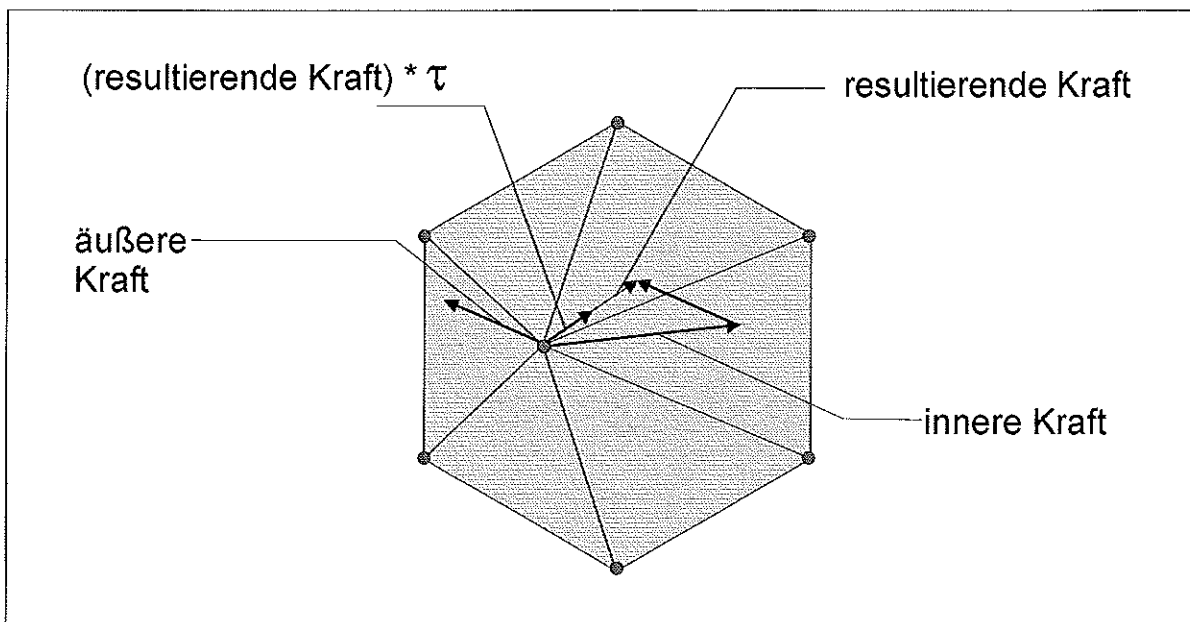


Abbildung 4.5: Zur Minimierung der Gesamtenergie werden die an den Netzknoten angreifenden äußeren und inneren Kräfte addiert. Der Netzknoten wird dann einen kleinen Schritt τ in diese Richtung bewegt.

Gleichung (4.4) kann auch als Fixpunktgleichung der Form $u = F(u)$ gelesen werden, mit $F(u) = u + \tau(f(u) - Au)$. Es ist klar, daß die Lösung u dieser Fixpunktgleichung

gerade eine Nullstelle des Ausgangsproblems $f(u) - Au = 0$ ist. Nach dem Banachschen Fixpunktsatz muß die in (4.4) definierte Funktion kontrahierend sein¹, damit die Fixpunktgleichung bei beliebigem Startwert konvergiert. Eine hinreichende Bedingung dafür, daß F eine kontrahierende Funktion ist, ist $|\partial F(u)/\partial u| < 1$. Nimmt man den Vektor der äußeren Kraft $f(u)$ nun wieder als konstant an, so ergibt sich:

$$\left| \frac{\partial F}{\partial u} \right| = |I - \tau A| < 1.$$

Habe A die n positiven Eigenwerte $a_1, \dots, a_n$ mit maximalem Eigenwert a_i , so gilt $|I - \tau A| = |1 - \tau a_i|$. Also

$$\begin{aligned} -1 &< 1 - \tau a_i < 1 \\ \Rightarrow -2 &< -\tau a_i < 0 \\ \Rightarrow 2 &> \tau a_i > 0 \\ \Rightarrow 2/a_i &> \tau > 0 \end{aligned}$$

Somit ergibt sich eine Bedingung für τ , die garantiert, daß der Vektor der Knotenparameter u in jedem Iterationsschritt näher an die Lösung herankommt. Da die Eigenwerte der Matrix A jedoch nicht bekannt sind, ist obige Bedingung von rein theoretischer Natur. Sie zeigt jedoch, daß der Zeitschritt τ umgekehrt proportional zur Kraft sein muß, d. h. je größer die Kraft ist, desto kleiner muß τ sein.

Ist m_t das Maximum der im Iterationsschritt t berechneten Gesamtkraft, dann wird τ zweckmäßigerweise so gewählt, daß $m_t \tau = e$ ist. e ist eine vom Benutzer festgesetzte Schranke. Wird e kleiner als die Voxelbreite gewählt, dann ist garantiert, daß jeder Netzknoten in einem Iterationsschritt nicht weiter als die Voxelbreite bewegt wird.

¹Die andere Bedingung, nämlich daß F ein Gebiet D in sich selbst abbildet, ist hier trivialerweise erfüllt, da $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Kapitel 5

Parallelisierung

Aufgrund der extrem langen Rechenzeiten der sequentiellen Version des Algorithmus wird eine Beschleunigung durch eine Parallelisierung des Programms angestrebt. Dies ist besonders in einem medizinischen Umfeld wichtig, da die Auswertung von anfallenden Daten möglichst schnell erfolgen soll.

Im Vordergrund der Überlegungen steht vor allen Dingen die praktikable Umsetzung eines parallelen Konzeptes. So soll das Programm zunächst nicht für spezielle Parallelrechner zugeschnitten sein, sondern die Ressourcen von schon vorhandener Hardware ausnutzen. Aus diesem Grund erfolgt die Realisierung primär auf einem vernetzten, heterogenen Workstation Cluster. Durch die Verwendung des weitverbreiteten Programmpaketes *pvm* bleibt die Möglichkeit der Portierung auf eine parallele Rechnerarchitektur gegeben. Die hier verwendete Rechnerarchitektur wird auch als MIMD Architektur mit verteiltem Speicher bezeichnet [OV89]. Dies bedeutet, daß zur gleichen Zeit verschiedene Operationen auf unterschiedlichen Daten ausgeführt werden können. Jeder Prozessor besitzt seinen eigenen Speicher, auf den nur er selber zugreifen kann. Die Kommunikation zwischen den Prozessen erfolgt über das Senden und Empfangen von Nachrichten. Eine Bibliothek mit entsprechenden Befehlen wird von *pvm* zur Verfügung gestellt.

5.1 Konzepte von pvm

pvm ist die Abkürzung von 'parallel virtual machine'. Das Programmpaket *pvm* wurde im Jahre 1989 vom Oak Ridge National Laboratory entwickelt. Ursprünglich war es als Tool zur Erforschung von heterogenen Computernetzwerken gedacht. Inzwischen ist *pvm* als public domain Programm frei verfügbar.

pvm ist ein Werkzeug, welches in der Lage ist, parallele Applikationen zu erzeugen und zu verwalten. Es kann dabei auf einem Verbund von Unix-Rechnern, welche lose über ein Netzwerk miteinander gekoppelt sind, operieren. Ebenso existieren *pvm* Portierungen für massiv parallele Rechnerarchitekturen. Die Kommunikation der Prozesse erfolgt über Nachrichten. Das *pvm*-System besteht aus zwei Hauptkomponenten:

- dem *pvm* Dämon, der auf jedem Prozessor gestartet sein muß, und welcher, für den Benutzer unsichtbar, die zur Laufzeit anfallende Prozeß- und Nachrichtenverwaltung übernimmt.

- Einem Programmierinterface, welches in Form einer linkbaren Bibliothek gegeben ist. Es ermöglicht die Prozeßkontrolle und den Nachrichtenaustausch.

Die Verwendung eines heterogenen Computernetzes hat gegenüber einer auf Parallelisierung spezialisierten Rechnerhardware einige Nachteile:

- Die Datenformate der Rechner, die den Verbund bilden, können zueinander inkompatibel sein. Dies ist z. B. durch die Verwendung einer Zahlenkodierung im Low-Endian oder Big-Endian Format gegeben. Die pvm-Befehle zum Versenden der Daten berücksichtigen diese Inkompatibilitäten und nehmen eine evtl. notwendige Umkodierung vor. Dies führt zu Performanceverlusten.
- Durch die Verwendung von Rechnern mit stark unterschiedlicher Rechenleistung muß eine an diese Gegebenheiten angepaßte Lastbalancierung stattfinden.
- Das Netzwerk, über das die Rechner verbunden sind, ist im allgemeinen nicht so leistungsfähig wie das eines massiv parallelen Systems. Die Netzbelastung geht als wichtiger Faktor in die Performance ein.

Den Nachteilen stehen auch Vorteile gegenüber:

- Die Kosten der Parallelisierung sind gering, da ein schon vorhandenes Rechnernetz genutzt werden kann.
- Die Programmentwicklung kann in der gewohnten Entwicklungsumgebung stattfinden. Dies beschleunigt die Programmentwicklungszeiten.

5.1.1 Programmierschnittstelle

Im folgenden werden die Prozedurengruppen, die pvm dem Benutzer in seiner Schnittstelle zur Verfügung stellt, kurz beschrieben. Eine ausführliche Darstellung kann in [GBDJMS94] gefunden werden.

- **Prozeßkontrolle**
Befehle zum Erzeugen und Löschen von Prozessen sind vorhanden. Ferner kann die Standardausgabe von Prozessen abgefangen werden.
- **Signalbearbeitung**
Die virtuelle Maschine kann auf Anfrage Signale an Prozesse übermitteln, welche die Prozesse auf bestimmte Situationen aufmerksam machen. Die Signale werden z. B. verschickt, wenn ein anderer Prozeß terminiert oder wenn ein Rechner zum Rechnerverbund hinzugefügt oder aus dem Verbund gelöscht wird.
- **Informationsbeschaffung**
Ein Prozeß kann Informationen bzgl. der Prozeß-ID anderer Prozesse sowie zur Konfiguration der virtuellen Maschine einholen.

- **Dynamische Konfiguration**

Es können zur Laufzeit Rechner aus dem Verbund herausgenommen oder in den Verbund eingefügt werden. Ferner sind Befehle vorhanden, mit denen etliche Optionen der virtuellen Maschine, wie z. B. der Routing-Mechanismus, zur Laufzeit geändert werden können.

- **Nachrichtenaustausch**

Es stehen Befehle zum Erzeugen und Löschen sogenannter Input- und Output-Buffer zur Verfügung. In diese Buffer können Daten vom Typ *byte*, *complex*, *double complex*, *float*, *double*, *integer*, *long*, *short* und *string* zum Verschicken gepackt bzw. nach Erhalt entpackt werden. Es ist möglich, Daten sowohl blockierend als auch nicht-blockierend zu verschicken und zu empfangen. Die Kommunikation kann stattfinden von Prozeß zu Prozeß, von Prozeß zu einer Prozeßgruppe, und von Prozeß zu allen anderen Prozessen (Broadcast).

- **Dynamische Prozeßgruppenbildung**

Prozesse können zur Laufzeit zu einer Gruppe zusammengefaßt werden. Eine Prozeßgruppe hat die Möglichkeit, sich an einer sogenannten Barriere zu synchronisieren.

5.2 Parallelisierungsstrategien

Im folgenden werden die verschiedenen prinzipiellen Möglichkeiten aufgezeigt, wie ein Algorithmus auf ein paralleles Programm abgebildet werden kann. Dies soll unter zwei verschiedenen Aspekten erfolgen:

1. Zum einen kann eine Unterscheidung aufgrund der Struktur des bei der parallelen Berechnung entstehenden Prozeßbaumes erfolgen. Darunter soll die Art und Reihenfolge der Erzeugung von Prozessen sowie die daraus resultierende logische Abhängigkeit der Prozesse untereinander verstanden werden.
2. Zum anderen kann eine Unterscheidung vorgenommen werden aufgrund der Art und Weise, wie die einzelnen Prozesse auf den zu verarbeitenden Daten operieren. Hierbei spielen die Möglichkeiten, ein konkretes Problem auf Teilprobleme abzubilden, eine entscheidende Rolle.

5.2.1 Parallele Prozeßstrukturen

In diesem Abschnitt sollen zwei fundamentale Möglichkeiten der Organisation eines Prozeßbaumes erläutert werden. Jedem dieser zwei Modelle kann je nach Problemstellung eine andere Art der Lastverteilung zugeordnet werden. Darauf wird im nächsten Abschnitt näher eingegangen.

5.2.1.1 Dynamische Prozeßstruktur

Werden zur Laufzeit dynamisch Prozesse erzeugt oder gelöscht, so wird von einer dynamischen Prozeßstruktur gesprochen. Häufig entsteht dabei eine baumartige Verzweigung

der Prozesse, welche immer dann auftritt, wenn ein Ausgangs- oder auch Vaterprozeß n Kindprozesse erzeugt und diese Kindprozesse wiederum ihrerseits nachfolgende Prozesse erzeugen usw. Es wird somit zur Berechnung ein Datenfluß von den Vätern zu den Kindern, also von der Prozeßbaumwurzel zu den Blättern, generiert. Nach erfolgreicher Berechnung erfolgt eine Datenübergabe in die andere Richtung, d. h. die Rechenergebnisse der Kindprozesse werden von den Vaterprozessen aufgesammelt, evtl. aufgearbeitet und dann wiederum an deren Vorgängerprozeß weitergegeben. Das Endresultat der Berechnung wird somit an der Baumwurzel aufgesammelt und kann von dem dortigen Prozeß angezeigt werden.

Dieses Prozeßmodell bietet sich insbesondere dann an, wenn die für die Kindprozesse anfallende Datenmenge nicht im vorhinein bekannt ist. Dann muß die Prozeßerzeugung dynamisch zur Laufzeit in Abhängigkeit der Eingabedaten erfolgen, und es entsteht die oben beschriebene Baumstruktur.

Als Beispiel für eine Problemstellung mit diesen Eigenschaften sei die Sortierung einer Liste mit Hilfe des Quicksort-Algorithmus genannt. Selbst wenn die Länge der Ausgangsliste bekannt ist, kann die Länge der während der Berechnung auftretenden Teillisten nicht vorhergesagt werden, da sie von den Elementen der unbekannten Ausgangsliste, genauer gesagt von der Wahl des Pivot-Elementes, abhängt. Die Entscheidung, ob weitere Kindprozesse erzeugt werden sollen oder nicht, trifft jeder Prozeß dynamisch nach Erhalt der Teilliste von seinem Vater. Somit entsteht die oben beschriebene Baumstruktur.

5.2.1.2 Statische Prozeßstruktur

In diesem Modell findet die Generierung von Prozessen nicht dynamisch, während der Laufzeit veränderbar statt, sondern es werden zu Anfang der Berechnung statisch n gleichartige Prozesse erzeugt, welche alle in ähnlicher Weise an verschiedenen Teilen der Datenmenge Berechnungen vornehmen. Ein typischer Ablauf einer Berechnung mit flacher Prozeßstruktur gliedert sich in folgende Schritte auf:

1. Starten des initialen Prozesses P . Dieser Prozeß selber kann im folgenden entweder nur administrative Zwecke erfüllen, d. h. er nimmt selber nicht an der Berechnung teil, sondern kümmert sich nur um die Initialisierung von Prozessen und die Verteilung von Daten und Ergebnissen. Dieses Modell wird *Master-Slave* Modell genannt. Oder der initiale Prozeß nimmt neben seinen organisatorischen Aufgaben ebenfalls an der Berechnung teil. Dann wird von einem *Node-Only* Modell gesprochen.
2. Generierung von n gleichen Prozessen durch P .
3. Initialisierung der Prozesse und Verteilung von Daten.
4. Berechnung des Resultates. Hierbei müssen in Abhängigkeit vom Problem evtl. periodisch Zwischenresultate ausgetauscht werden.
5. Sammeln des Endergebnisses durch P sowie Anzeige des Resultates.
Terminierung der erzeugten Prozesse.

Diese Art der Parallelisierung bietet sich immer dann an, wenn die Größe der Eingabedatenmenge bekannt ist und die Aufteilung des Gesamtproblems nicht vom Inhalt der Daten

abhängt. In diesem Fall ist eine dynamische Prozeßerzeugung nicht nötig und würde zu einem unnötigen Overhead in der Berechnung führen.

5.2.2 Lastaufteilung

Im vorherigen Abschnitt wurde eine Unterscheidung von verschiedenen Parallelisierungsstrategien aufgrund der entstehenden Prozeßstruktur getroffen. Nun soll eine Aufteilung nach anderen Gesichtspunkten getroffen werden. Maßgeblich ist hier das Prinzip, nach welchem ein Problem in Teilprobleme zerlegt wird.

5.2.2.1 Datenaufteilung

Besteht das Gesamtproblem aus einer großen Datenmenge, auf der immer die gleichen Berechnungsschritte ausgeführt werden, so kann die Berechnung parallelisiert werden, indem die Datenmenge in n Teilmengen zerlegt wird, auf denen die parallelen Prozesse jeweils gleiche Arbeitsschritte ausführen. Eine optimale Geschwindigkeitssteigerung ergibt sich dann, wenn die n Teilprobleme unabhängig voneinander bearbeitet werden können. In diesem Fall kann die Berechnung ohne den Austausch von Zwischenergebnissen erfolgen, und auch eine Synchronisation der Prozesse während der Berechnung, um Zwischenergebnisse auszutauschen, entfällt.

Benötigt ein Prozeß zur Bearbeitung von Daten im Segment i Daten aus einem anderen Segment j , so ist eine Interprozeßkommunikation mit eventueller Synchronisation nötig. Dies bedeutet einen Overhead im Vergleich zur sequentiellen Version des Programms.

5.2.2.2 Funktionenaufteilung

Oft erfolgt die Verarbeitung der Daten in verschiedenen, aufeinander aufbauenden Phasen, in denen verschiedene Funktionen auf den Daten ausgeführt werden und bei denen die Eingabedaten einer Berechnungsphase gleich den Ausgabedaten des vorherigen Berechnungsschrittes sind. In diesem Fall können die einzelnen, funktionell verschiedenen Stufen der Berechnung parallelisiert werden. Dieses Verfahren wird auch als *Pipelining* bezeichnet. Innerhalb einer Berechnungsphase kann wiederum eine Parallelisierung möglich sein, dann evtl. nach dem Prinzip der Datenaufteilung.

Für das Pipelining ist es wichtig, daß die Berechnung einer Stufe zwar auf dem Rechenergebnis der vorhergehenden Stufe basiert, ansonsten aber von dieser Stufe unabhängig sein soll. Aus Effizienzgründen sollten weiterhin die Bearbeitungszeiten der einzelnen Stufen in etwa gleich sein [OV89].

5.3 Realisierung

Um die in Kapitel 2.2 und 3 definierten Kräfte auf einem vorgegebenen Oberflächennetz zu berechnen, wurde das Node-Only Modell mit Datenaufteilung gewählt. Die Auswahl wurde aus folgenden Gründen getroffen.

Die Berechnung der inneren und äußeren Kräfte erfolgt für jeden Knotenpunkt nach dem

gleichen, iterativen Schema. Eine rekursive Bearbeitung der Daten, an der sich eine dynamische Prozeßgenerierung orientieren könnte, gibt es nicht. Die Berechnungsschritte hängen nicht von den Punktpositionen ab. Daher ist eine dynamische Aufteilung der Daten und damit ebenso eine dynamische, baumartige Prozeßstruktur nicht nötig. Dies ist insbesondere auch für die Effizienz des parallelen Algorithmus von Bedeutung, da eine dynamische Prozeßerzeugung und Datenaufteilung einen großen Overhead bedeuten würde. In dem hier vorgestellten Algorithmus geschieht die Aufteilung der Daten (=Netzknoten) sowie die Erzeugung der Prozesse nur einmal in einem initialen Schritt. Das Löschen oder Erzeugen eines Prozesses während der Berechnung ist nicht nötig.

Es wurde eine Lastaufteilung gemäß des Datenaufteilungsmodells gewählt. Eine Zerlegung der Last nach funktionellen Gesichtspunkten ist nicht sinnvoll. Vorstellbar wäre eine separate Berechnung der verschiedenen, an den Netzknoten angreifenden Kraftarten¹. Dies könnte in einem Pipelineverfahren geschehen, indem pro Knotenpunkt eine Kraft berechnet und dieses Ergebnis an die nächste Stufe weitergereicht würde. Die nächste Stufe könnte die nächste Kraft berechnen und diese zum Ergebnis der vorherigen Stufe addieren. Am Ende der Pipeline würde somit die Gesamtkraft stehen. Aus Effizienzgründen scheidet dieses Verfahren hier jedoch aus, da die Bearbeitungszeiten der einzelnen Stufen der Pipeline stark differieren würden. Zum einen ist für jede Kraftart ein anderes, unterschiedlich aufwendiges Berechnungsverfahren nötig. Zum anderen soll die Parallelisierung in einem heterogenen Rechnernetz stattfinden. Das heißt insbesondere, daß die einzelnen Stufen der Pipeline auf Prozessortypen verschiedener Leistungsfähigkeit verteilt würden, was zu einer noch stärkeren Differenz in den Bearbeitungszeiten führen könnte.

Ein weiterer Gesichtspunkt ist die verwendete Hardware. Da hier ein lose gekoppeltes Rechnernetz, also eine nicht speziell nach dem Pipelinekonzept entwickelte Rechnerarchitektur verwendet werden soll, würde der mit dem Pipelining verbundene Kommunikationsaufwand zu einer hohen Effizienzeinbuße führen. Bei einem Netz mit n Knoten und mit p Prozessen (= p Berechnungsstufen) würden pro Gesamtberechnungsschritt $n * p$ Daten transferiert. Die einzelnen, zur Parallelisierung notwendigen Schritte werden im folgenden näher beschrieben.

5.3.1 Gebietsaufteilung

Damit eine Aufteilung der Last gemäß des Datenaufteilungsmodells erfolgen kann, muß das Oberflächennetz in Teilnetze zerlegt werden. Um eine effiziente Lastbalance zu erreichen, wird bei Verwendung von p Prozessoren eine Aufteilung in n Teilnetze vorgenommen, wobei $n > p$ gilt. Einem Prozeß p_i , $1 \leq i \leq p$ werden n_i Teilnetze zur Bearbeitung zugewiesen. Trivialerweise gilt $\sum_{i=1}^p n_i = n$.

Die einzelnen Prozesse berechnen jeweils einen vollen Berechnungsschritt pro Teilnetz. Unter einem vollem Berechnungsschritt wird hier die Berechnung eines ganzen Zeitschrittes, d. h. die Berechnung aller Kräfte mit anschließender Erneuerung der Punktkoordinaten, verstanden.

Zur Berechnung der an einem Netzknoten angreifenden Kräfte müssen nicht nur die Koordinaten dieses Punktes, sondern auch die Koordinaten der Nachbarknoten gegeben

¹Innere Kräfte: Dehnungs- und Krümmungskräfte. Äußere Kraft: Druckkraft.

sein. Wird ein finites Differenzen-Schema der Euler-Lagrange Gleichung (1.2) verwendet, so müssen sogar die Koordinaten von noch weiter entfernten Punkten vorliegen. Daher ist ein Austausch der Punktkoordinaten an den Grenzen von Teilnetzen nötig.

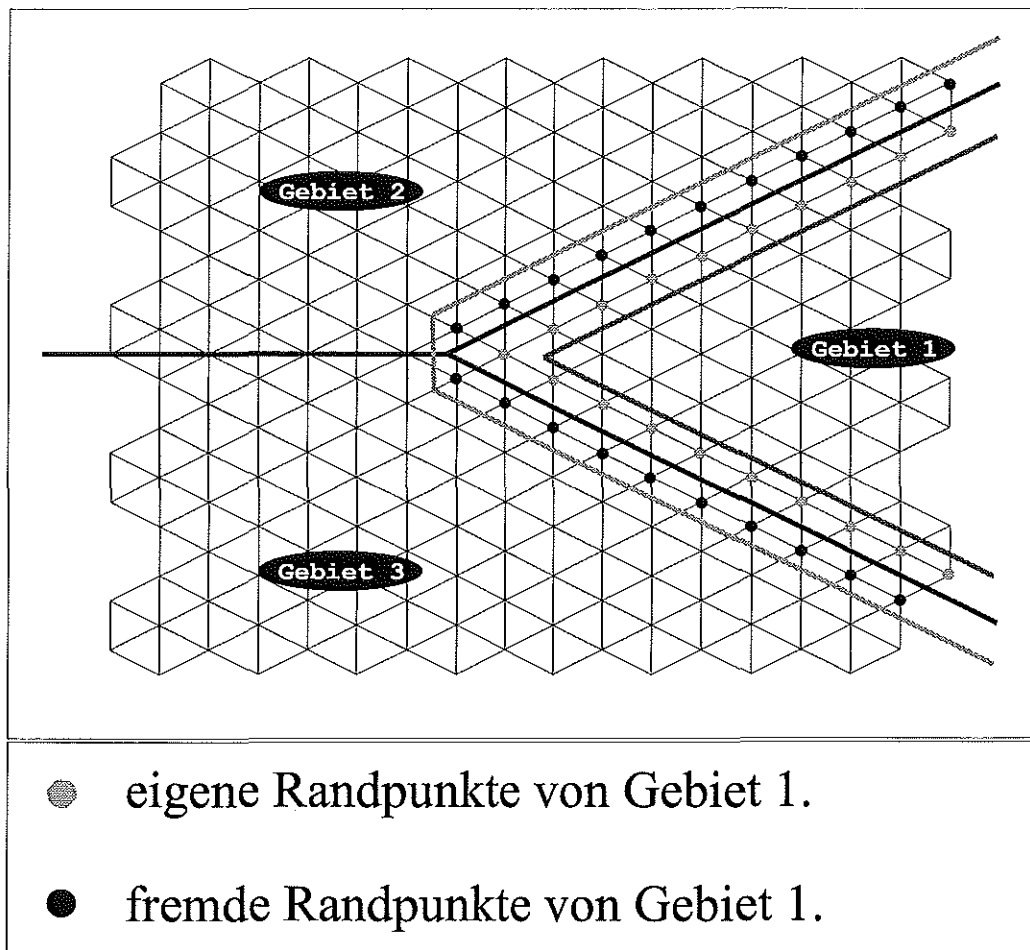


Abbildung 5.1: Netz mit Aufteilung in 3 Gebiete. Hellgrau: eigene Randpunkte von Gebiet 1. Schwarz: fremde Randpunkte von Gebiet 1.

Die Punkte, die ein Teilnetz bilden, können in drei Punktklassen eingeteilt werden. Die Einteilung orientiert sich daran, welche Operationen ein Prozeß i auf den jeweiligen Punkten ausführen muß, um die Kräfte zu berechnen. Mit Operationen sind hier insbesondere auch die zur Prozeßkommunikation notwendigen Schritte gemeint.

1. Innere Punkte

Als innerer Punkt wird ein Teilnetzpunkt bezeichnet, welcher zum eigentlichen Teilnetz gehört, und welcher vollständig mit Punkten umgeben ist, die alle ebenfalls zu

diesem Teilnetz gehören. Der Prozeß i muß für solch einen Punkt alle Kräfte berechnen sowie die Punktverschiebung durchführen. Ein Versenden der Punktkoordinaten an andere Prozesse ist nicht nötig.

2. Eigene Randpunkte

Ein eigener Randpunkt ist ein Punkt, der selber zum Teilnetz gehört, aber mindestens einen Nachbarpunkt besitzt, der einem anderen Teilnetz zugeordnet ist. Auch hier muß der Prozeß i die Kräfte und die Punktverschiebung berechnen. Zusätzlich müssen die Koordinaten der eigenen Randpunkte an die Prozesse, die diese Punktkoordinaten benötigen, verschickt werden. Die Koordinaten der eigenen Randpunkte müssen also versendet werden.

3. Fremde Randpunkte

Hierunter soll ein Punkt verstanden werden, der nicht zum Teilnetz gehört, der aber mindestens einen zum Teilnetz gehörenden Nachbarpunkt besitzt. Die Kenntnis dieser Punktkoordinaten wird benötigt, um die an den eigenen Randpunkten angreifenden Kräfte zu berechnen. Für die Kraftberechnung der fremden Randpunkte ist ein anderer Prozeß verantwortlich. Prozeß i empfängt die Koordinaten dieser Punkte nach jedem Iterationsschritt. Es ergibt sich somit, daß fremde Randpunkte von Prozeß i immer eigene Randpunkte eines anderen Prozesses sind.

Da die Effizienz der Parallelisierung maßgeblich von der Prozeßkommunikation beeinflusst wird², ist eine Aufteilung des Gesamtnetzes in gleich große Teilnetze mit möglichst kleinem Begrenzungsrand erforderlich. Um dies näherungsweise zu erreichen, wurde der im folgenden beschriebene Algorithmus zur Berechnung der Teilnetze verwendet. Ausgenutzt wurde dabei die spezielle Geometrie des Ausgangsnetzes: das Netz einer Kugeloberfläche. Verfahren zur Aufteilung von Gebieten beliebiger Geometrien können in [FL93], [Sim91] und [VZKW93] gefunden werden. Bei der hier vorgestellten Methode ist die zugrundeliegende Idee, daß bei der Aufteilung einer Kugeloberfläche in n gleich große Teilnetze möglichst kleine Begrenzungsänder entstehen, wenn die Zentren, d. h. die Schwerpunkte, der Teilnetze möglichst gleich auf der Kugeloberfläche verteilt sind. Um das Ausgangsproblem zu lösen, ergibt sich mit dieser Überlegung die Aufgabe, n Punkte möglichst gleichmäßig auf einer Kugeloberfläche zu verteilen.

Dies wird erreicht, indem die Punkte zunächst zufällig auf die Kugeloberfläche aufgebracht werden. Danach wirken Abstoßungskräfte auf diese Punkte. Dadurch, daß sich die Punkte untereinander abstoßen, wird eine gleichmäßige Verteilung auf der Kugeloberfläche erreicht. Die Koordinaten der Punkte werden in Polarnotation angegeben. Punkt P_i liegt dabei an den Polarkoordinaten $(\phi_i, \psi_i, 1)$, wobei der Radius der Kugel frei wählbar ist und hier der Radius 1 gewählt wurde. ϕ_i und ψ_i sind zwei Raumwinkel. Eine Umrechnung in kartesische Koordinaten erfolgt durch

$$\hat{x}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} \cos(\phi_i)\sin(\psi_i) \\ \sin(\phi_i) \\ \cos(\phi_i)\cos(\psi_i) \end{pmatrix}.$$

²Ein weiterer wichtiger Gesichtspunkt ist die Lastbalancierung. Siehe dazu Abschnitt 5.3.3.

Die Energie Θ bzgl. zweier Punkte P_i und P_j ist hier proportional zu '1/Punktabstand', wobei der Punktabstand auf der Kugeloberfläche gemessen werden soll. Damit ergibt sich

$$\Theta(\phi_i, \psi_i) = \frac{180}{\pi \arccos(\hat{x}_i^{\text{tr}} \hat{x}_j)}$$

und als an Punkt P_i angreifende, verallgemeinerte Kraft

$$\begin{pmatrix} \frac{\partial \Theta(\phi_i, \psi_i)}{\partial \phi_i} \\ \frac{\partial \Theta(\phi_i, \psi_i)}{\partial \psi_i} \end{pmatrix} = \begin{pmatrix} -\frac{180}{\pi \arccos(\hat{x}_i^{\text{tr}} \hat{x}_j)^2 \sqrt{1 - \hat{x}_i^{\text{tr}} \hat{x}_j}} (-\sin(\phi_i) \sin(\psi_i) x_2 + \cos(\phi_i) y_2 - \sin(\phi_i) \cos(\psi_i) z_2) \\ -\frac{180}{\pi \arccos(\hat{x}_i^{\text{tr}} \hat{x}_j)^2 \sqrt{1 - \hat{x}_i^{\text{tr}} \hat{x}_j}} (\cos(\phi_i) \cos(\psi_i) x_2 - \cos(\phi_i) \sin(\psi_i) z_2) \end{pmatrix}.$$

Die an den Punkten angreifenden Kräfte wirken nur auf die Winkel ϕ_i und ψ_i , der Radius 1 bleibt fest. Somit ist garantiert, daß die Punkte bei dem Prozeß der Krafteinwirkung immer auf der Kugeloberfläche liegen bleiben. Würden die Kräfte unmittelbar auf die Koordinaten der Punkte einwirken, so würde die Abstoßung der Punkte im dreidimensionalen Raum ein Entfernen der Punkte von der ursprünglichen Kugeloberfläche bewirken.

Analog dem Prinzip, welches auch zur Berechnung der Netzdeformation selber angewandt wird, werden die Punkte einen kleinen Schritt entlang der Kraft bewegt. In einem iterativen Prozeß verteilen sich die Punkte auf der Kugeloberfläche, wie Abbildung 5.2 zeigt. Dieser Algorithmus terminiert, wenn das Maximum der Punktverschiebung in einem Iterationsschritt unter eine vorgegebene Schranke ε fällt.

Abbildungen 5.3, 5.4 und 5.5 zeigen Gebietsaufteilungen in 6, 12 und 20 Teilgebiete. Die Aufteilung des Kugelnetzes in Teilnetze ist mit dem hier vorgestellten Algorithmus nicht eindeutig bestimmt, sondern hängt von der initialen Position der Gebietsmittelpunkte ab. Die Größe und Form der berechneten Teilgebiete schwankt daher leicht. Die Schwankungen in den Größen der Teilgebiete wurden durch mehrmaliges Ausführen des Algorithmus untersucht. Folgende Tabelle zeigt die minimale und maximale Punktzahl pro Teilgebiet sowie die zugehörigen Standardabweichungen, die sich bei 10 Durchläufen des Verfahrens ergeben haben.

Anzahl Teilgebiete	Min. Punktzahl	Max. Punktzahl	Standardabweichung
4	2556	2566	4,12
6	1678	1726	20,83
8	1267	1298	12,20
10	955	1046	33,70
12	848	857	3,12
14	709	815	34,31
16	617	694	22,67
18	521	598	21,68
20	489	548	20,26

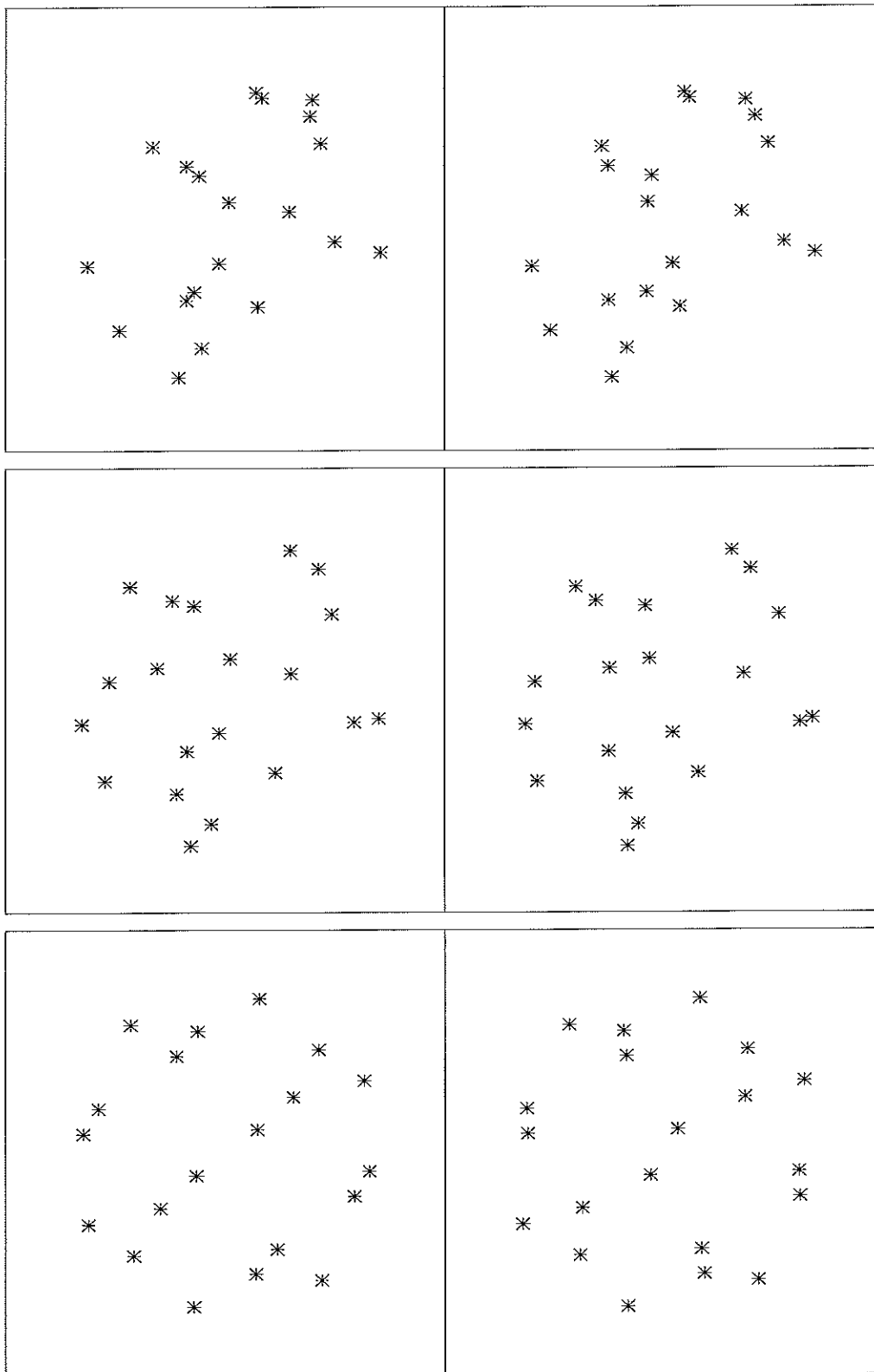


Abbildung 5.2: (Stereo-Darstellung:) 20 Punkte in drei Verteilungsphasen. In der Startkonfiguration (oben) sind sie zufällig verteilt. Über einen Abstoßungsprozeß (Mitte) erreichen sie eine gleichmäßige Verteilung (unten).

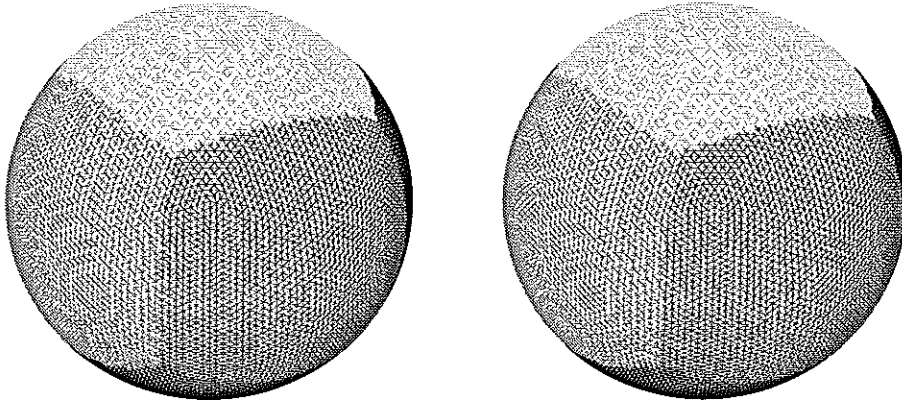


Abbildung 5.3: (Stereo-Darstellung:) Aufteilung der Kugeloberfläche in 6 Teilgebiete.

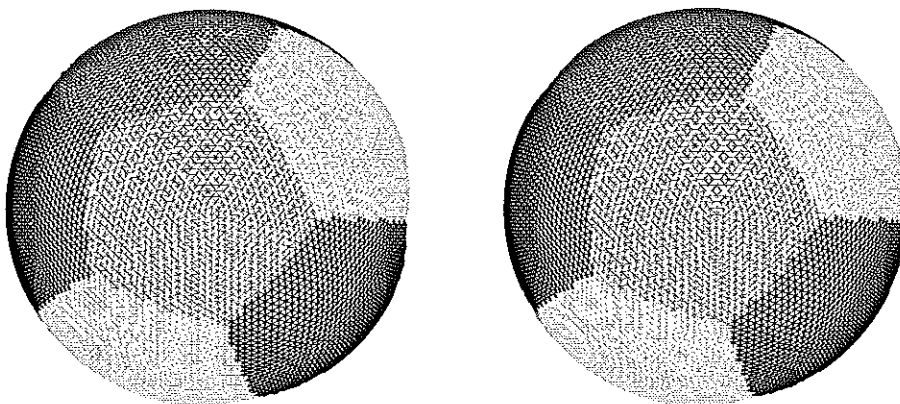


Abbildung 5.4: (Stereo-Darstellung:) Aufteilung der Kugeloberfläche in 12 Teilgebiete.

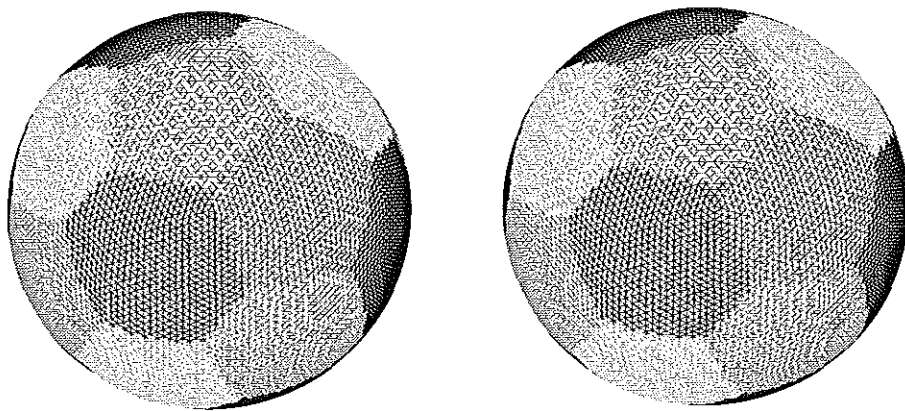


Abbildung 5.5: (Stereo-Darstellung:) Aufteilung der Kugeloberfläche in 20 Teilgebiete.

5.3.2 Prozeßerzeugung

Wie bereits oben beschrieben, wird hier ein flaches, statisches Prozeßmodell zugrundegelegt, in welchem ein Master-Prozeß p gleichartige Slave-Prozesse erzeugt. Die Slave-Prozesse führen, gesteuert vom Master-Prozeß, die Berechnung durch. Der Master selber nimmt nicht an der Berechnung teil, sondern sorgt für das Erzeugen und Löschen der Slaves sowie für die Verteilung der Daten und für die Kontrolle der dynamischen Lastverteilung (s. u.). Auf einem System mit s Prozessoren werden genau p Prozesse erzeugt, genau einer für jeden Prozessor. Das Erzeugen von $p > s$ Prozessoren wird zwar von *pvm* unterstützt, ist aber nicht sinnvoll, da sich dann mehrere Prozesse die Rechenleistung eines Prozessors teilen müssen. Dies führt zu einer Verschlechterung der Effizienz, da einem erhöhten Kommunikationsaufwand keine höhere Rechenleistung gegenübersteht.

5.3.3 Lastbalancierung

Die Lastbalancierung, also die gleichmäßige Zuweisung der Teilberechnungen zu den einzelnen Prozessen, ist ein wichtiger Faktor, durch den die Effizienz des Gesamtprogramms maßgeblich mitbestimmt wird. Die Lastbalancierung ist zu unterscheiden vom sogenannten 'Mapping', unter welchem die Abbildung von Prozessen auf Prozessoren verstanden wird. Das Mapping spielt hier keine bedeutende Rolle, da, wie bereits oben erwähnt, ein Prozeß pro Prozessor erzeugt wird und die Prozesse alle uniform arbeiten.

Prinzipiell gibt es zwei verschiedene Arten der Lastbalancierung. Erstens kann die Verteilung der Teilprobleme einmal am Anfang der Berechnung erfolgen und sich dann nicht mehr ändern. In diesem Fall wird von statischer Lastbalancierung gesprochen. Die Verteilung der Last geschieht in Relation zur Leistungsfähigkeit der jeweiligen Prozessoren, so daß die Bearbeitungszeiten der (evtl. unterschiedlich großen) Teilaufgaben aneinander angeglichen werden. Berücksichtigt wird bei der statischen Lastbalancierung nur die bei Programmstart verfügbare Rechenleistung. Dies ist nur dann sinnvoll, wenn die Rechenleistung während der Berechnung nicht schwankt. Bei der Realisierung auf einem heterogenen Netz aus Workstations ergeben sich folgende Probleme:

- Die Leistung der verwendeten Prozessoren kann stark unterschiedlich sein (statisch).
- Die zur Verfügung stehende Rechenleistung kann sich während der Berechnung ständig ändern, da die Rechner auch andere Benutzer bedienen müssen (dynamisch).

Da besonders der letzte Punkt eine wichtige Rolle für die Performance des Rechnerverbundes spielt, wurde eine dynamische Lastbalancierung realisiert, welche die Verteilung der Last während der Berechnung an die verfügbare, schwankende Rechenleistung anpassen kann.

Die Anpassung der Lastgröße erfolgt durch die Zuweisung einer unterschiedlichen Anzahl von Teilnetzen an die Prozesse entsprechend ihrer Leistungsfähigkeit. Ein Prozeß mit höherer Rechenleistung erhält mehr Teilnetze zur Bearbeitung zugewiesen als einer mit niedrigerer Performance. Dazu ist es erforderlich, daß die Anzahl der durch die Gebietsaufteilung gewonnenen Teilnetze größer als die Anzahl der Prozesse (=Prozessoren) ist. Denkbar ist auch eine Steuerung der Lastgröße über die Größe der Teilnetze. Dies würde bedeuten, daß jedem Prozeß genau ein Teilnetz zugewiesen wird, wobei die Anzahl der

Punkte in den Teilnetzen auf die Leistungsfähigkeit des jeweiligen Prozesses abgestimmt ist.

Diese Vorgehensweise hat den Vorteil, daß eine sehr feine Abstimmung auf die Rechenleistung erfolgen könnte, da Teilnetze mit genau angepaßter Punktzahl erzeugt werden könnten. Sie hat den Nachteil, daß eine komplett neue Gebietszerlegung bei einer Umschichtung der Last zu erfolgen hätte. Dies hätte den Transfer aller Teilnetze von den Slaves an den Master zur Folge. Daraufhin müßte der Master eine neue Gebietszerlegung vornehmen, und danach die angepaßten Gebiete wieder an die Slaves verteilen. Der Overhead des zweimaligen Verschickens aller Netzdaten und der erneuten Gebietszerlegung wird durch das hier angewandte Verfahren vermieden auf Kosten einer suboptimalen Lastgrößenanpassung. Im folgenden sind die Schritte, die der Master-Prozeß zur dynamischen Lastbalancierung ausführen muß, schematisch aufgeführt:

Schritte des Master-Prozesses bei der dynamischen Lastbalancierung.

1. for $i=1$ to p do spawn P_i
2. while calculation is not finished do
3. for $i=1$ to p do receive t_i from P_i
4. calculate initial n_i
5. for $i=1$ to p do send data of n_i areas to P_i
6. for $i=1$ to p do receive t_i from P_i
7. calculate n'_i
8. if $\exists i. n_i \neq n'_i$ then
9. initiate rebalancing of work

In Schritt 1. erzeugt der Master p Slaves und erhält zwecks Beurteilung der Rechenleistung t_i Rechenzeiten von ihnen (3.). Danach bestimmt der Master die Anzahl n_i von Teilnetzen pro Slave und verschickt die Daten der Teilnetze (4. und 5.). Die Slaves berechnen nun k Iterationen und senden erneut ihre Rechenzeiten an den Master (6.). Nach diesen neuen Daten errechnet der Master abermals eine optimale Teilnetzverteilung (7.) und leitet eine Umverteilung der Netze ein, falls diese erforderlich ist (9.).

Die Slaves verfahren dabei wie folgt:

Schritte des Slave-Prozesses i bei der dynamischen Lastbalancierung.

1. $t_{total} = 0$
2. for $r=1$ to k do
3. $t=time()$
4. for $s=1$ to n_i do
5. calculate $f(T_s)$
6. unblocking send data of border points
7. $t_{total} = t_{total} + time() - t$
8. blocking receive data of border points
9. send t_{total} to master

Der Slave-Prozeß mißt hier die Rechenzeit von k Iterationsschritten. Er bearbeitet dazu die ihm zugewiesenen Teilnetze (4.), indem er alle anfallenden Kräfte berechnet und dann die Punktverschiebungen durchführt. Dazu ruft er in Schritt 5. die Prozedur $f()$ auf. Danach verschickt er die Koordinaten der Randpunkte an andere Prozesse (6.) und empfängt seinerseits die Koordinaten fremder Randpunkte(8.). Da der Prozeß die Daten der Randpunkte benötigt, um mit seiner Berechnung fortzufahren, wird der Empfang im blockierenden Modus ausgeführt, d. h. der Prozeß wartet, bis er die Daten erhält. Wichtig ist daher, daß die Messung der Rechenzeiten vor dem Empfangen der Punktkoordinaten erfolgt, da an dieser Stelle eine Synchronisation der Prozesse stattfindet und die Rechenzeit des Prozesses, der die Randpunktdaten abschickt, die Laufzeitmessung stark beeinflussen könnte. Nach den k Iterationen wird die Rechenzeit an den Master-Prozeß übermittelt (9.). Die Schritte, welche zur Übermittlung eines Teilnetzes ausgeführt werden müssen, sind in Abschnitt 5.3.4.2 beschrieben.

5.3.4 Datenorganisation

Für die Effizienz des parallelen Algorithmus ist eine intelligente Organisation der Daten maßgeblich. Der wichtigste und größte Teil der Daten besteht aus den Daten des Oberflächennetzes bzw. der Teilnetze. Diese wiederum gliedern sich auf in Punktkoordinaten und Beschreibung der Netztopologie. Unter Netztopologie wird die Definition von Punktnachbarschaften verstanden, d. h. die Topologie legt fest, welcher Punkt mit welchem anderen Punkt verbunden ist. Dies geschieht durch die Definition von Dreieckslisten, in welchen alle das Teilnetz bildenden Dreieckselemente enthalten sind.

Bei der Beschreibung der Punkte selber ist neben den Punktkoordinaten auch ein globaler Index wichtig, mit dessen Hilfe ein Punkt eindeutig identifiziert werden kann. Dieser

Index wird benötigt beim Austausch der Randpunktkoordinaten, bei der Lastverteilung, sowie beim Einsammeln der Endergebnisse durch den Master-Prozeß.

Neben den Teilnetzdaten ist das Anlegen von temporären Listen notwendig, mit deren Hilfe das Versenden der Randpunktkoordinaten organisiert wird. Diese Hilfsinformationen haben solange Gültigkeit, bis eine Umbalancierung der Last stattfindet. Nach Verlagerung eines Teilnetzes auf einen anderen Prozeß müssen sie neu berechnet werden. Unter der Prämisse, daß eine Neuverteilung der Teilnetze nur selten stattfindet, ist diese Vorgehensweise gerechtfertigt. Zwar bedeutet die Berechnung dieser Listen einen Overhead, dieser findet jedoch pro Lastbalancierung nur einmal statt. Da die Prozeßkommunikation durch diese Hilfsdaten beschleunigt wird, resultiert daraus bei seltener Umbalancierung ein Performancegewinn.

5.3.4.1 Organisation des Versendens der Randpunktkoordinaten

Nachdem alle Prozesse die ihnen vom Master zugewiesenen Teilnetze erhalten haben, verschickt der Master an alle Prozesse zusätzlich eine Tabelle, durch welche die Zuordnung von Teilnetz zu Prozeß möglich wird. Der Austausch von Randpunktkoordinaten wird beschleunigt, indem ein Prozeß Listen mit Verweisen auf eigene Randpunkte anlegt. Für jeden Empfängerprozeß wird eine Liste angelegt, durch welche alle Punkte, die an diesen Prozeß geschickt werden müssen, referenziert werden. Dies ist schematisch in Abbildung 5.6 dargestellt. Durch das Generieren dieser Listen braucht die Suche der eigenen Randpunkte in den Netzdaten sowie die Zuordnung der eigenen Randpunkte zu Empfängerprozessen nur einmal, und nicht bei jedem Verschicken neu, zu geschehen. Neben den eigentlichen Punktkoordinaten wird auch die globale Punkt-ID verschickt. Mit ihrer Hilfe kann der Empfängerprozeß die erhaltenen Koordinaten dem richtigen Punkt zuordnen.

5.3.4.2 Organisation der Lastbalancierung

Eine evtl. notwendige Umverteilung der Last wird vom Master-Prozeß eingeleitet. Die Umverteilung bedeutet, daß ein Prozeß ein oder mehrere Teilnetze an einen anderen Prozeß abgibt. Der Master-Prozeß bestimmt anhand der Rechenzeiten die ID's der an dem Austausch beteiligten Prozesse und erteilt an sie den Befehl der Teilnetzabgabe bzw. -annahme. Die Übergabe eines Teilnetzes von Prozeß i an Prozeß j muß in mehreren Schritten geschehen. Die Schritte für den Prozeß i , welcher das Teilnetz abgibt, sehen wie folgt aus:

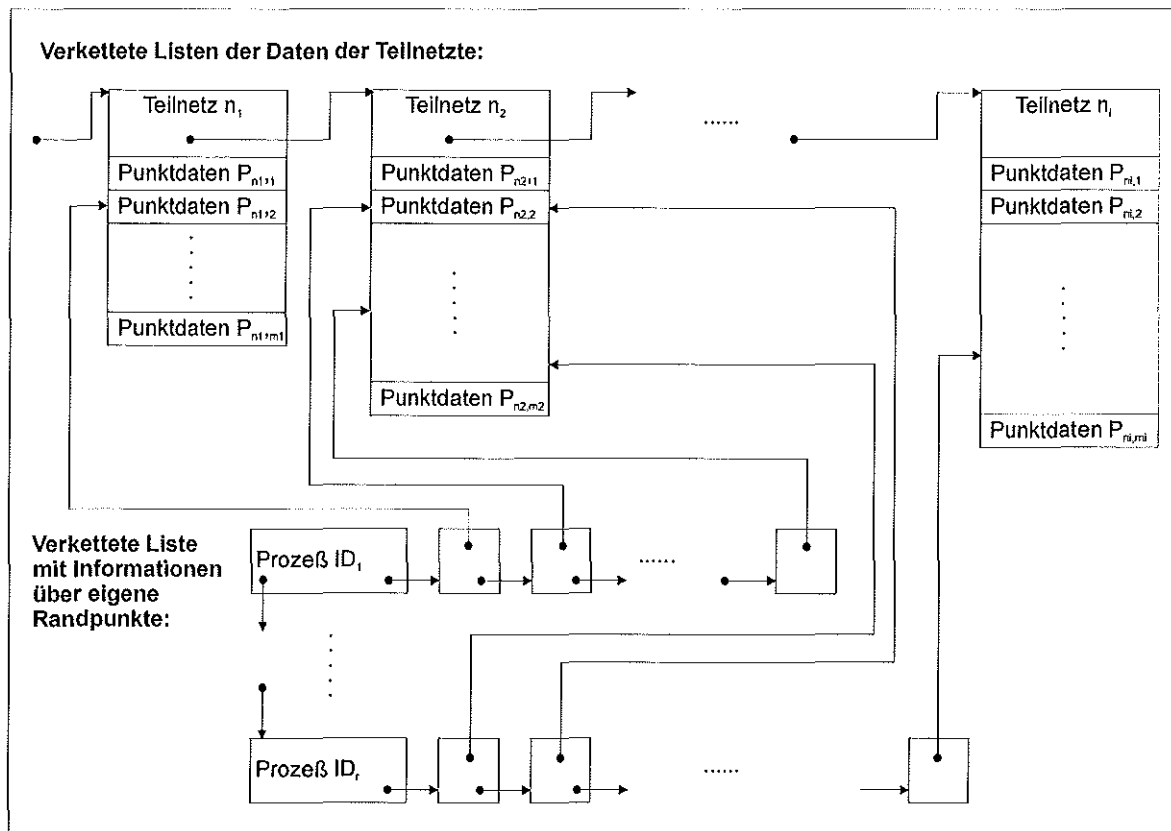


Abbildung 5.6: Die Daten der Teilnetze werden in einer verketteten Liste abgelegt (oben). Damit die Daten der eigenen Randpunkte effizient an andere Prozesse verschickt werden können, wird für jeden anderen Prozeß P eine Liste erzeugt, in welcher die Randpunkte, deren Koordinaten an diesen Prozeß geschickt werden müssen, vermerkt sind. Dies geschieht über Zeiger.

1. In der die Netztopologie beschreibenden Elementeliste werden die Punkte über Zeiger referenziert, die lokal zum Prozeß sind. Diese Zeiger werden vor dem Verschicken der Netzdaten zu globalen Punktindizes umgerechnet.
2. Die das Teilnetz beschreibende Datenstruktur wird aus der Liste der zu bearbeitenden Teilnetze ausgehängt.
3. Die Punktkoordinaten sowie die Topologiedaten werden an Prozeß j geschickt.
4. Schritte 1. bis 3. werden wiederholt, falls mehr als ein Teilnetz abgegeben werden soll.
5. Vom Master-Prozeß wird eine neue Prozeß-Teilnetzliste empfangen.
6. Die Liste, welche die eigenen Randpunkte den Prozessen zuordnet, welche diese Punktkoordinaten zur Kraftberechnung benötigen, wird erneuert.

Der Empfängerprozeß führt die dazu dualen Schritte aus:

1. Die Punktkoordinaten und die Netztopologie werden vom Prozeß i empfangen.
2. Die globalen Punktindizes in den Topologiedaten werden durch lokale Zeiger ersetzt.
3. Das neue Teilnetz wird in die Liste der zu bearbeitenden Teilnetze eingegliedert.
4. Schritt 1. bis 3. werden wiederholt, falls mehr als ein Teilnetz empfangen werden soll.
5. Vom Master-Prozeß wird eine neue Prozeß-Teilnetzliste empfangen.
6. Die Liste der zu verschickenden eigenen Randpunkte wird erneuert.

5.3.5 Laufzeitmessungen

Die parallele Version des Programms wurde getestet auf einem Rechnernetz, welches aus vier DEC-Alpha Workstations unterschiedlicher Leistungsfähigkeit besteht. Eine Workstation hatte eine Taktfrequenz von 233 MHz, eine andere Workstation eine Taktfrequenz von 200 MHz und es waren zwei Workstations mit einer Taktfrequenz von 166 MHz verfügbar. Zur Bewertung der Performance wurde ein Testprogramm ausgeführt, welches die Kraftberechnung an einem Netz mit insgesamt 40962 Punkten und 81920 Dreieckselementen durchführt. Das Gesamtnetz wurde vor der eigentlichen Berechnung in 20 Teilnetze aufgeteilt. Aufgrund der Performanceunterschiede wurden die 20 Teilnetze typischerweise wie folgt auf die vier Rechner verteilt: 233 MHz Rechner: 8 Teilnetze. 200 MHz Rechner: 5 Teilnetze. 166 MHz Rechner: 4 und 3 Teilnetze. Die Verteilung erfolgte automatisch und wurde vom Master-Prozeß aufgrund einer initialen Messung der Rechnerperformance berechnet und gesteuert.

Der erzielte Speedup ist in Abbildung 5.7 dargestellt. Wie zu erwarten war, verhindert die notwendige Prozeßkommunikation, daß der gemessene Speedup in unmittelbarer Nähe des maximal möglichen Speedups liegt. Trotzdem zeigt sich, daß die Kommunikation nicht so zeitintensiv ist, daß eine Parallelisierung von vornherein nutzlos wäre. Ein deutlicher Geschwindigkeitsgewinn konnte erzielt werden. Es soll an dieser Stelle darauf hingewiesen werden, daß der Speedup von der Feinheit des verwendeten Netzes abhängt. Es lohnt sich nicht, die Berechnung eines groben Netzes zu parallelisieren, da hier die Relation von Anzahl Teilnetzpunkten zu Anzahl Randpunkten sehr schlecht ist. Diese Relation beschreibt den Kommunikationsaufwand im Verhältnis zum Aufwand der eigentlichen Berechnung. Aus dem gleichen Grund sollte der Algorithmus auch nicht in einer hohen Parallelisierungsstufe auf sehr vielen Prozessoren laufen: das Teilnetz müsste dann in viele kleine Teilgebiete unterteilt werden, und abermals wird die Relation „Anzahl Teilnetzpunkte“ zu „Anzahl Netzrandpunkten“, ungünstig.

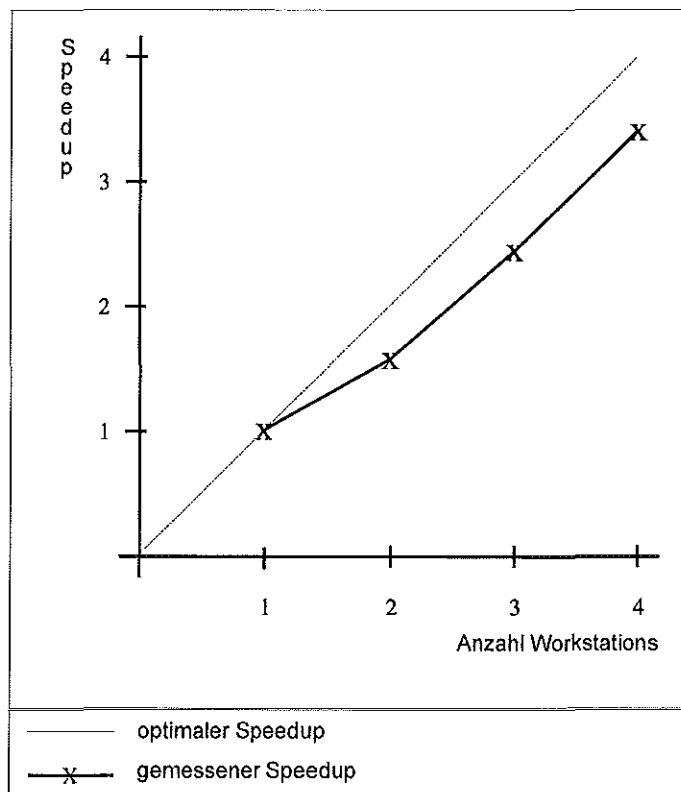


Abbildung 5.7: Der gemessene Speedup der parallelen Programmversion, gemessen an einem Workstation-Cluster mit 4 Rechnern.

Kapitel 6

Anwendung: 'Cortical Flattening'

In diesem Kapitel wird eine Anwendung des in dieser Arbeit vorgestellten Verfahrens aufgezeigt. Die Anwendung stammt aus dem Bereich der Hirnforschung und stellt einen Lösungsansatz für das im folgenden beschriebene Problem des „Cortical Flattening“ dar. An dieser Stelle soll aber auf die Universalität des Konzeptes der aktiven Oberflächen hingewiesen werden, mit deren Hilfe generell Oberflächen aus anatomischen 3D-Datensätzen rekonstruiert werden können. Dies ist von Interesse, da häufig wichtige Eigenschaften von Organen, welche im Datenmaterial abgebildet sind, durch deren Oberfläche charakterisiert werden. Solche Eigenschaften sind etwa Größe, Form, Lage im Raum und Lage zu anderen Organen. Insbesondere erscheint auch eine Anwendung bei stark verrauschtem Datenmaterial, wie es z. B. bei 3D-Ultraschalldaten vorliegt, sinnvoll. Durch die Verwendung von Glattheitsbedingungen werden verrauschte Oberflächen automatisch geglättet. Zudem spiegelt sich in der Verwendung des Membranmodells ein Vorwissen über die Oberflächenbeschaffenheit der untersuchten Objekte wieder, nämlich daß sie eine geschlossene Oberfläche besitzen. Durch Fehler bei der Datenaufnahme oder durch Rauschen entstandene „Löcher“ in der im Datenmaterial dargestellten Oberfläche werden durch Überziehen mit dem Oberflächennetz ausgefüllt.

Falls ein Vorwissen über die Art des Rauschens vorliegt, so kann dieses Wissen evtl. ebenfalls als Energieterm in das Oberflächenmodell mit hineinmodelliert werden, um eine angepaßte Rauschunterdrückung bei der Oberflächenfindung vorzunehmen.

6.1 Grundlagen der NMR

Der Effekt der sogenannten Kernspinresonanz wurde zum erstenmal 1946 von Bloch et al. [BHP46] und Purcell et al. [PTP46] entdeckt. Dieser Effekt ist bei allen Atomen zu beobachten, die eine ungerade Anzahl Nukleonen im Kern aufweisen. Dies trifft auf ca. zweidrittel aller stabilen Elemente zu. Das wohl am häufigsten vertretene Atom mit diesen Eigenschaften ist das Wasserstoffatom (^1H), dessen Kern aus einem Proton besteht, aber auch Kohlenstoff (^{13}C), Stickstoff (^{14}N), Fluor (^{19}F) und Natrium (^{23}Na) gehören dazu. Unter dem Kernspin wird der Gesamtdrehimpuls des Kerns verstanden. Dieser setzt sich aus den Bahndrehimpulsen und den Eigendrehimpulsen der Nukleonen zusammen. Die Kerne sind elektrisch geladen und induzieren einen elektrischen Kreisstrom um die Drehachse. Die bewegte Ladung wiederum induziert ein schwaches Magnetfeld, welches bei

den Kernen mit ungerader Anzahl Nukleonen nicht verschwindet. Im allgemeinen sind die magnetischen Dipolmomente in einem Material zufällig orientiert. Dies ist in Abbildung 6.1, linke Seite, wiedergegeben. Wird ein äußeres Magnetfeld der Stärke B_0 angelegt, so richten sich die Kernspins entlang den magnetischen Feldlinien auf (Abbildung 6.1, rechts). Die Stärke des Magnetfeldes wird in Tesla ('T') gemessen. Ein Tesla entspricht ungefähr der 30000-fachen Stärke des Erdmagnetfeldes. Bei der Ausrichtung der Kerne ist sowohl die parallele als auch die antiparallele Ausrichtung des Spins möglich, wobei die parallele Lage energetisch leicht begünstigt ist. Die Energiedifferenz zwischen diesen Zuständen beträgt dabei

$$\Delta E = \gamma(h/(2\pi))B_0 \quad (h : \text{Planksches Wirkungsquant}).$$

Das Verhältnis von parallelen zu antiparallelen Ausrichtungen wird bestimmt durch die Boltzmannverteilung, in welche die Temperatur T und die Magnetfeldstärke B_0 als Parameter eingehen. Unter normalen Versuchsbedingungen und bei Einwirkung eines Magnetfeldes der Stärke 1T beträgt der Überschuß an parallel ausgerichteten Wasserstoff-Kernen in einem wasserhaltigen Medium lediglich 6 auf 10 Millionen.

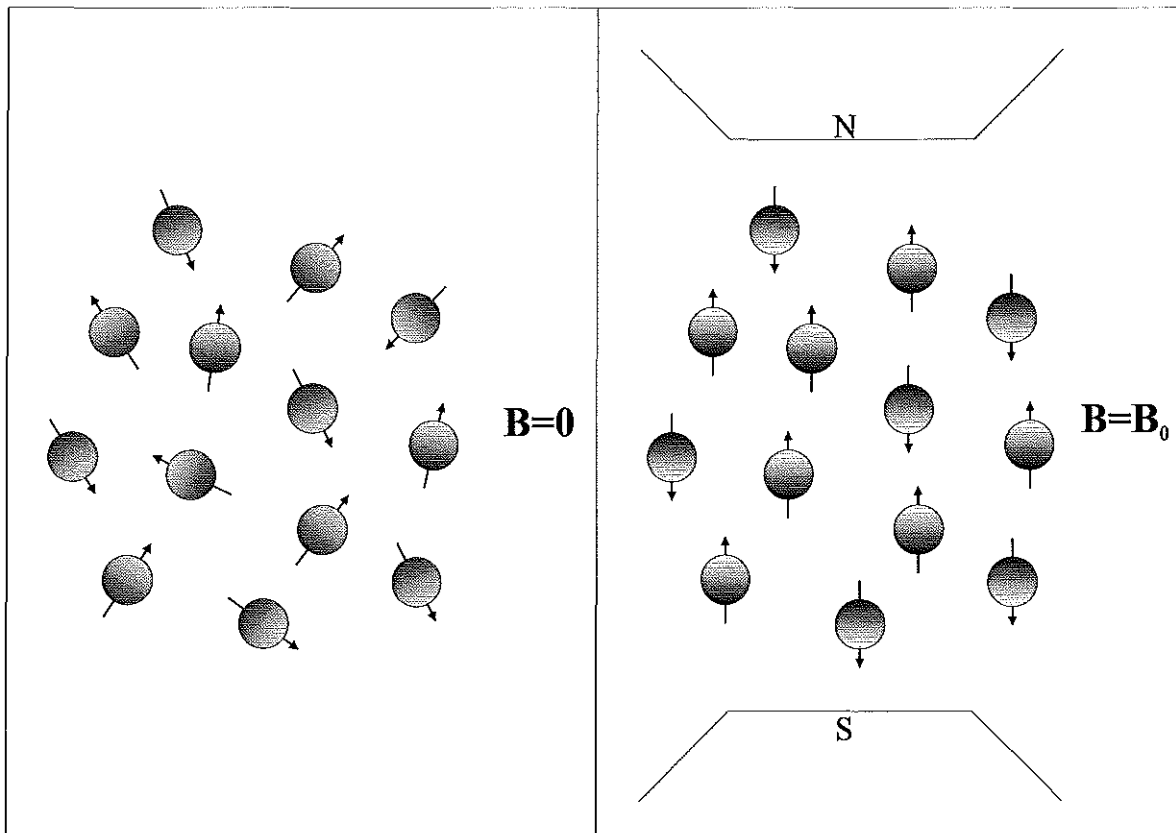


Abbildung 6.1: Ohne Einwirkung eines äußeren Magnetfeldes verteilt sich der Spin der Atomkerne zufällig (links). Nach dem Anlegen des Magnetfeldes richten sich die Kerne parallel oder antiparallel zum Magnetfeld aus (rechts).

Unter Einwirkung einer zum äußeren Magnetfeld senkrechten elektromagnetischen Strahlung einer bestimmten Frequenz werden die Kerne von dieser Position weggedreht.

Sie fangen an, um die Feldlinien des Magnetfeldes zu präzessieren, d. h. sie beschreiben eine Art Kreiselbewegung um die Feldlinien (siehe Abbildung 6.3). Die Frequenz ω_0 , mit der dies geschieht, wird bestimmt durch die Stärke B_0 des äußeren Magnetfeldes und einer von der Atomart abhängigen Konstanten γ , dem sogenannten gyromagnetischen Verhältnis. Es gilt das Gesetz $\omega_0 = \gamma B_0$. Die Frequenz ω_0 wird *Larmorfrequenz* oder auch *Präzessionsfrequenz* genannt. Im Falle des Wasserstoffatoms ist γ gerade 42.6 MHz/T. Durch das Einstrahlen von unterschiedlichen elektromagnetischen Impulsen, deren Frequenz der Larmorfrequenz entspricht, können die Kerne in beliebigen Winkeln gekippt werden. Ein Impuls, welcher die Kerne um 90° auslenkt, wird 90° Impuls genannt.

Wird die Einstrahlung der elektromagnetischen Wellen abgeschaltet, so nehmen die Kerne wieder die energetisch günstigere Position parallel oder antiparallel zu den Magnetfeldlinien ein. Dieser Vorgang wird *Relaxation* genannt. Während der Zeit der Ausrichtung, der *Relaxationszeit*, senden die Atomkerne kurzzeitig selber elektromagnetische Strahlung der Stärke M ab. Diese Strahlung kann mit einer Empfängerspule gemessen werden. Die Relaxation wird durch zwei Prozesse bestimmt, welche sich überlagern. Um dies zu erläutern soll, im folgenden das von den Atomkernen abgestrahlte Signal zum einen in Richtung des äußeren Magnetfeldes (M_z) und zum anderen in der dazu senkrechten Ebene (M_{xy}) betrachtet werden. Nach einem 90° Impuls geben die Atomkerne zunächst ein starkes Signal M_{xy} ab. Dieses Signal fällt sehr schnell ab, wobei der Abfall des (sinusförmigen) Signales über die Zeit t durch die Funktion $M_{xy} = M \exp(-t/T_2)$ beschrieben wird (Abbildung 6.2, links). T_2 ist hierbei eine Konstante, welche *transversale Relaxationszeit* oder auch Spin-Spin-Relaxationszeit genannt wird und das Abklingverhalten beschreibt. Im Gegensatz zum Feld in x, y -Richtung steigt die Stärke des von den Kernen abgestrahlten Feldes in z -Richtung mit Ausrichtung der Kerne entlang B_0 wieder an. Diese *Longitudinalrelaxation*, die auch Spin-Gitter-Relaxation genannt wird, wird durch die Funktion $M_z = M(1 - \exp(-t/T_1))$ beschrieben (Abbildung 6.2, rechts).

Die eigentliche Messung eines Objektes im Kernspintomographen wird maßgeblich beeinflusst durch die Verwendung von verschiedenen Pulssequenzen. Aufnahmen mit der gleichen Pulsfolge, die sich aber im Zeitabstand der einzelnen Impulse unterscheiden, können einen deutlichen Kontrastunterschied zwischen verschiedenen Gewebetypen aufweisen. Der Kontrast zwischen Gewebetypen kann daher durch die Verwendung von angepaßten Impulsfolgen optimiert werden. Die bei der Relaxation der Kerne gemessenen Signale können grundsätzlich aufgrund dreier Parameter analysiert werden. Dies sind die Zeitkonstanten T_1 und T_2 sowie die Spindichte. Die Bestimmung dieser Parameter erfolgt durch die Verwendung unterschiedlicher Impulssequenzen. Es sind mehrere Messungen erforderlich, um alle drei Parameter zu bestimmen. Verschiedene Gewebearten beeinflussen die drei Parameter unterschiedlich stark. Somit kann eine Unterscheidung von Gewebetypen erleichtert werden, wenn mehrere Messungen unterschiedlicher Gewichtung vorliegen. Liegen z. B. T_1 - und protonendichte-gewichtet Aufnahmen vor, so ergibt sich pro Voxel ein zweidimensionaler Merkmalsvektor, welcher sich im Gegensatz zu einem eindimensionalen Vektor besser zur Segmentierung eignet. Anschaulich läßt sich diese Verbesserung dadurch begründen, daß zwei Voxel, welche sich vom Intensitätswert in einer Messung nicht unterscheiden, evtl. in der anders gewichteten Messung unterschiedliche Intensitäten aufweisen, wodurch sie verschiedenen Gewebetypen zugeordnet und somit genauer segmentiert werden können.

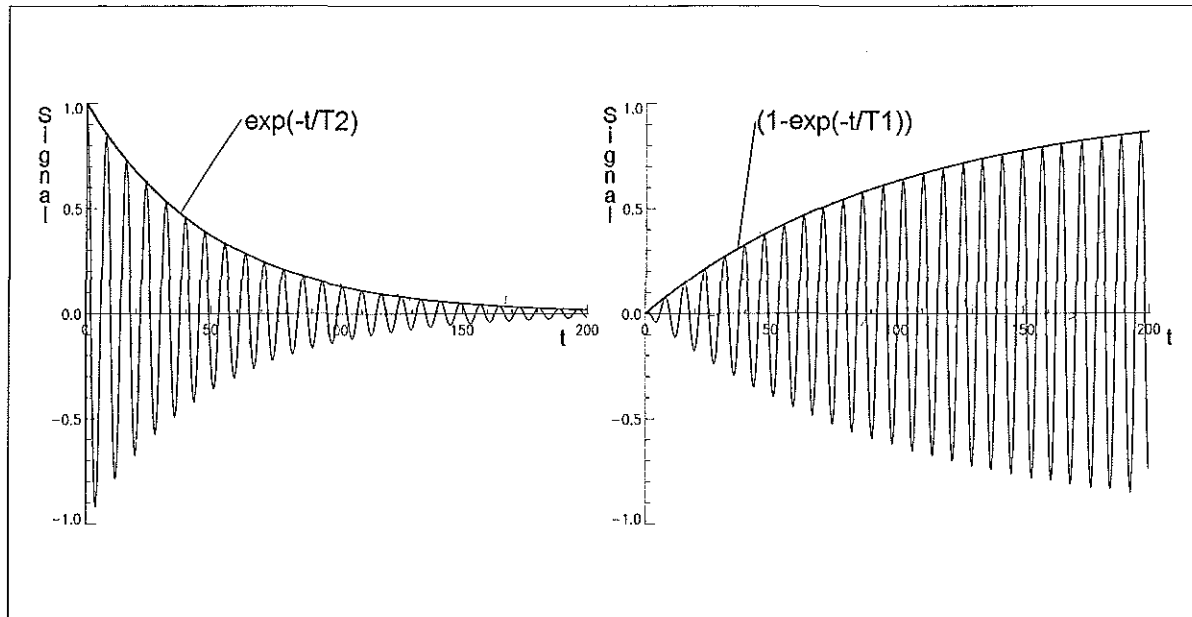


Abbildung 6.2: Das von der Präzessionsbewegung der Atomkerne erzeugte Signal klingt in xy-Richtung mit der Zeit ab (links). Parallel zum äußeren Magnetfeld B_0 steigt die Signalstärke wegen der Longitudinalrelaxation der Kerne wieder an (rechts).

6.1.1 Kodierung der Ortsinformation

Es gibt verschiedene Methoden, um den Entstehungsort der von den Empfängerspulen gemessenen Signale zu bestimmen. Alle diese Methoden basieren auf dem Prinzip, daß die Stärke des äußeren Magnetfeld ortsabhängig gestaltet werden kann und daß die Larmorfrequenz proportional zur Stärke des Magnetfeldes ist. Eine einfache Möglichkeit, ein ortsabhängiges Magnetfeld zu erzeugen, besteht darin, die das Magnetfeld erzeugenden Spulen so anzuordnen und anzusteuern, daß die Magnetfeldstärke linear vom Ort abhängt. In diesem Fall wird von einem *linearen Gradienten* gesprochen. Durch die Auswahl einer Impulsfolge eines bestimmten Frequenzspektrums kann nun selektiv eine bestimmte Schicht im zu untersuchenden Objekt angesprochen werden (Abbildung 6.4, links). Dieser Prozeß heißt *selektive Anregung* ([GGM74]).

Wird nach dem Auslenken der Atomkerne durch die elektromagnetische Impulsfolge ein zweiter magnetischer Gradient, der senkrecht zum ersten steht, angelegt, so präzisieren die Kerne mit unterschiedlichen Frequenzen, je nachdem in welcher Objektzeile sie liegen ([Lau73]). Wird das aufgrund dieser Anregung gemessene Signal mit einer eindimensionalen Fouriertransformation bearbeitet, so werden die Frequenzanteile wieder separiert, und es ergibt sich eine Projektion des gemessenen Objektes (Abbildung 6.4, rechts unten). Wird die Messung mehrfach, mit unterschiedlichen räumlichen Winkeln des zweiten Gradienten ausgeführt, so ergeben sich Projektionen unter verschiedenen Raumwinkeln. Mit diesen Projektionen kann eine komplette Schicht rekonstruiert werden, wobei der gleiche Rekonstruktionsalgorithmus verwendet werden kann, der auch für die auf Röntgenstrahlen basierende Computer-Tomographie benutzt wird.

Es gibt jedoch auch Kodierungsmethoden, welche die speziellen Eigenschaften der

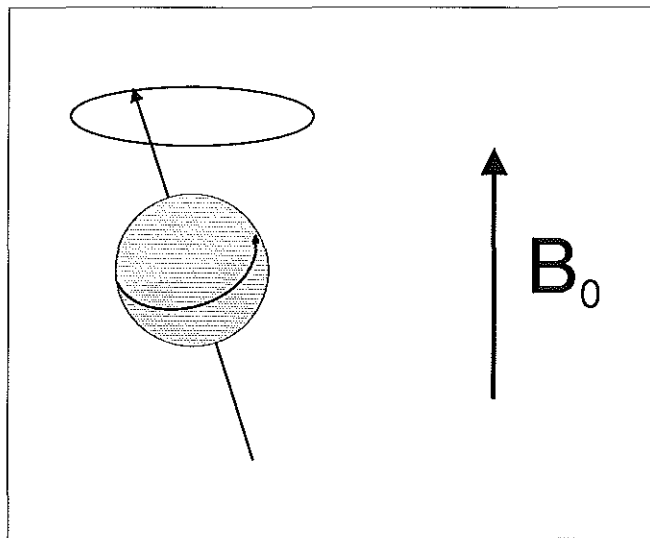


Abbildung 6.3: Wird eine zum äußeren Magnetfeld B_0 senkrecht stehende Elektromagnetische Strahlung mit der Larmorfrequenz ω_0 eingestrahlt, so präzessiert der Atomkern um die Feldlinien des äußeren Magnetfeldes, ebenfalls mit der Frequenz ω_0 .

Magnet-Resonanz-Tomographie ausnutzen. Dazu gehört die sogenannte *Fourier Tomographie* ([KWE75]). Hier wird neben den oben beschriebenen zwei Gradienten noch ein drittes Gradientenfeld, der sogenannte *Phasen-Kodierungs-Gradient*, eingesetzt. Dieser Gradient steht senkrecht zu den beiden ersten, wird nur für sehr kurze Zeit eingeschaltet und bewirkt eine Phasenverschiebung der Kern-Präzession. Bildlich gesprochen wird die Präzessionsbewegung der Kerne durch das kurzzeitige Wirken des Phasen-Kodierungs-Gradienten etwas „angeschoben“. Die Messung wird wiederholt, wobei die Anzahl der Messungen der späteren Zeilenzahl des Bildes entspricht. Bei jeder Wiederholung wird die Steigung des Phasen-Kodierungs-Gradienten um einen konstanten Wert verändert. Dies hat zur Folge, daß der von der Zeile abhängige Phasenvorlauf bei jeder Messung um einen von der Zeile abhängigen konstanten Wert variiert. Dieser Phasenunterschied kann bei einer zweidimensionalen Welle auch als Frequenzunterschied in der zur Phasenverschiebung senkrechten Richtung gedeutet werden. Dieser Frequenzunterschied wiederum kann über die Fouriertransformation auf räumliche Orte umgerechnet werden. Das in den einzelnen Messungen empfangene Signal wird dazu in einer Matrix angeordnet, deren Spalten (x-Richtung) den Zeiten t während einer Messung und deren Zeilen (y-Richtung) den unterschiedlichen Messungen entsprechen. Durch eine zweidimensionale Fouriertransformation, die bekanntermaßen separierbar ist, wird das Bild des Objektes rekonstruiert. Dabei entspricht die Fouriertransformation in x-Richtung einer Sortierung der Frequenzen nach Spalten. Durch die nachfolgende Fouriertransformation in y-Richtung wird das Signal (nun innerhalb einer Spalte) räumlich den Bildzeilen zugeordnet. Dies ist aufgrund der Phasenverschiebung, welche in y-Richtung einer Frequenz entspricht, möglich.

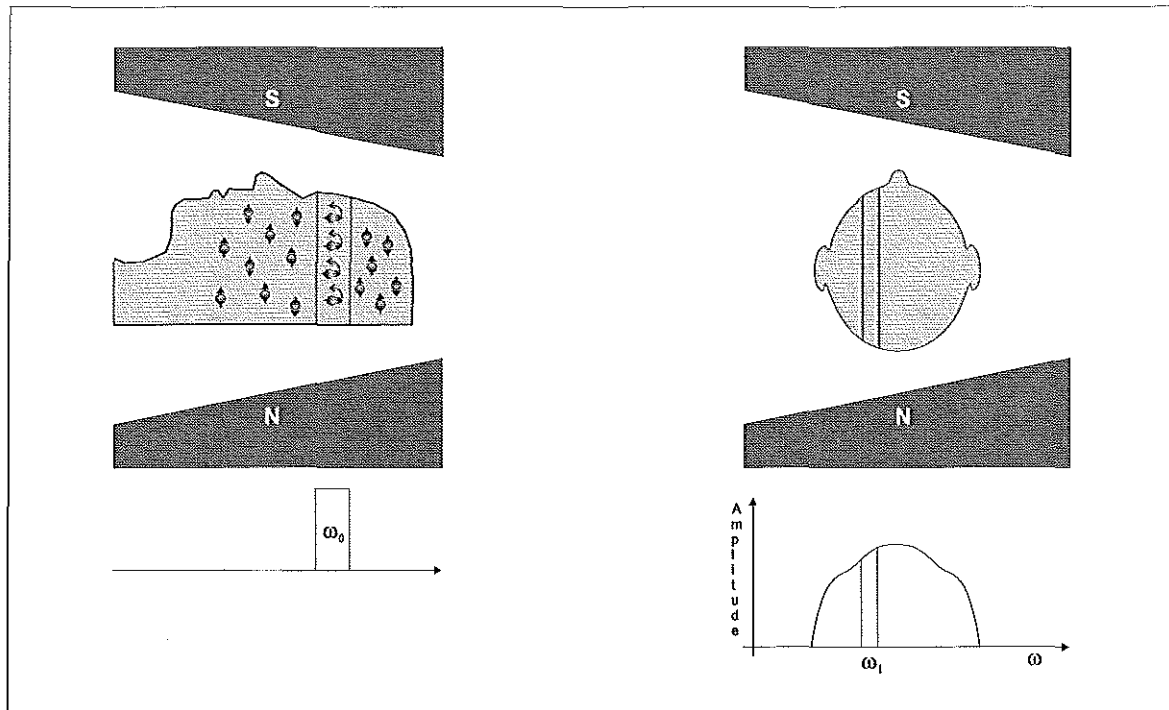


Abbildung 6.4: Nach Anlegen eines magnetischen Gradienten wird durch die Impulsfrequenz ω_0 eine Schicht selektiert (links). Wird auch in der dazu senkrechten Richtung ein magnetischer Gradient erzeugt, so kann über die Signalfrequenz auf eine Bildspalte geschlossen werden (rechts).

6.2 Cortical Flattening

Die Erforschung des Gehirns fußt im wesentlichen auf zwei Grundlagen, nämlich auf der Erforschung der Anatomie und der Erforschung der Funktion des Gehirns. Die Untersuchung beider Gesichtspunkte ist mit modernen, wenig invasiven Methoden an lebenden Probanden möglich. Eingesetzt werden dafür Meßmethoden, die Aufschluß über Struktur und Arbeitsweise des menschlichen Gehirns geben. Eine Möglichkeit, anatomische Aufnahmen des Gehirns zu gewinnen, ist die Kernspintomographie, auch NMR¹ genannt. Ein Beispiel einer solchen Aufnahme zeigt Abbildung 6.5. In dieser Aufnahme sind verschiedene Gewebetypen, wie z. B. graue und weiße Substanz und Schädelknochen, aber auch die das Gehirn umgebende Flüssigkeit (Liquor), zu erkennen. Aufnahmen, welche die Funktionsweise des Gehirns widerspiegeln, sind in Abbildung 6.6 wiedergegeben. Es handelt es sich um die Aufnahme eines Positronen-Emissions-Tomographen (PET). Bei der funktionellen Bildgebung spielt die Lokalisation von sogenannten aktivierten Gehirnarealen eine entscheidende Rolle. Dies sind Regionen des Gehirns, bei denen im funktionellen Datenmaterial eine erhöhte Gehirnaktivität aufgezeichnet ist. Durch die Korrelation von aktivierten Arealen und einer kognitiven oder auch motorischen Leistung des gemessenen Probanden können Rückschlüsse auf die Funktion des Gehirns gezogen werden.

¹nuclear magnetic resonance

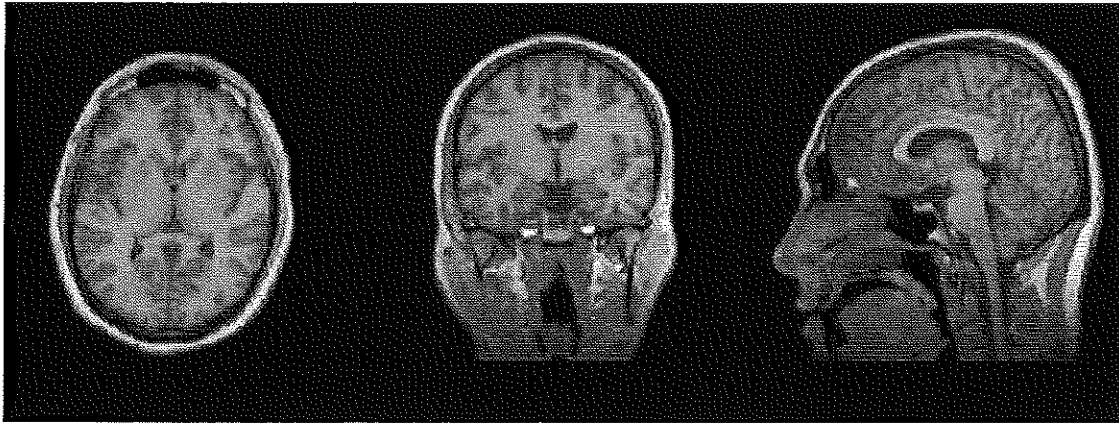


Abbildung 6.5: Die Gehirnanatomie wird im NMR-Bild wiedergegeben. Zu sehen sind, von links nach rechts, ein transversaler, coronaler und sagittaler Schnitt.

Von Interesse ist eine genaue räumliche Lokalisation von Arealen. Hierunter fallen Fragestellungen, ob Areale, die einer gewissen Funktion dienen, bei unterschiedlichen Probanden an ähnlichen Stellen des Gehirns liegen. Mit ähnlich kann eine Übereinstimmung bzgl. standardisierter Koordinaten, wie sie im Talairach Atlas definiert sind ([TS67], [TT88]), gemeint sein. Auch eine relative Ähnlichkeit kann von Interesse sein, wie z. B. die Frage, ob ein bestimmtes Areal immer in der gleichen Gehirnfalte liegt oder nicht. Da funktionelle Datensätze zunächst keine oder wenig Informationen über die zugehörige Anatomie enthalten, muß zur Beantwortung dieser Fragen neben der funktionellen Aufnahme auch ein anatomischer Datensatz erzeugt werden. Um das Problem der räumlichen Zuordnung von anatomischen und funktionellen Daten zu lösen, sind in der Literatur ([PCS89], [Ess94]) viele Ansätze vorhanden. Auf dieses sogenannte „Matching-Problem“ soll im weiteren nicht eingegangen werden, sondern es wird vorausgesetzt, daß anatomische und funktionelle Daten vorliegen, welche räumlich einander zugeordnet sind.

Obwohl es sich beim Gehirn um ein dreidimensionales Objekt handelt, sind Fragestellungen, welche sich auf zweidimensionale Flächen beziehen, sinnvoll. Dies liegt daran, daß ein Großteil der im Gehirn befindlichen Neuronen, welche bei der Erbringung einer Gehirnleistung aktiviert werden, in der sogenannten Großhirnrinde liegen. Die Großhirnrinde ist eine relativ dünne Schicht von ungefähr gleichmäßiger Dicke, welche die Oberfläche des Großhirns bedeckt. Durch Projektion aktivierter Schichten auf die äußere Oberfläche dieser Schicht erhält man ein zweidimensionales Abbild der aktivierten Areale. Dies ist in Abbildung 6.7 dargestellt. Anzumerken ist, daß das Gehirn auch innere Strukturen enthält, welche bei Gehirnleistungen aktiv werden, wie z. B. den Hippocampus. Diese Bereiche können mit der hier vorgestellten Methode nicht erfaßt werden.

Sind die in der Großhirnrinde liegenden aktiven Areale bekannt, so sind folgende, sich auf die Oberfläche beziehenden Fragestellungen von Interesse:

1. Welche Fläche nehmen Areale ein?

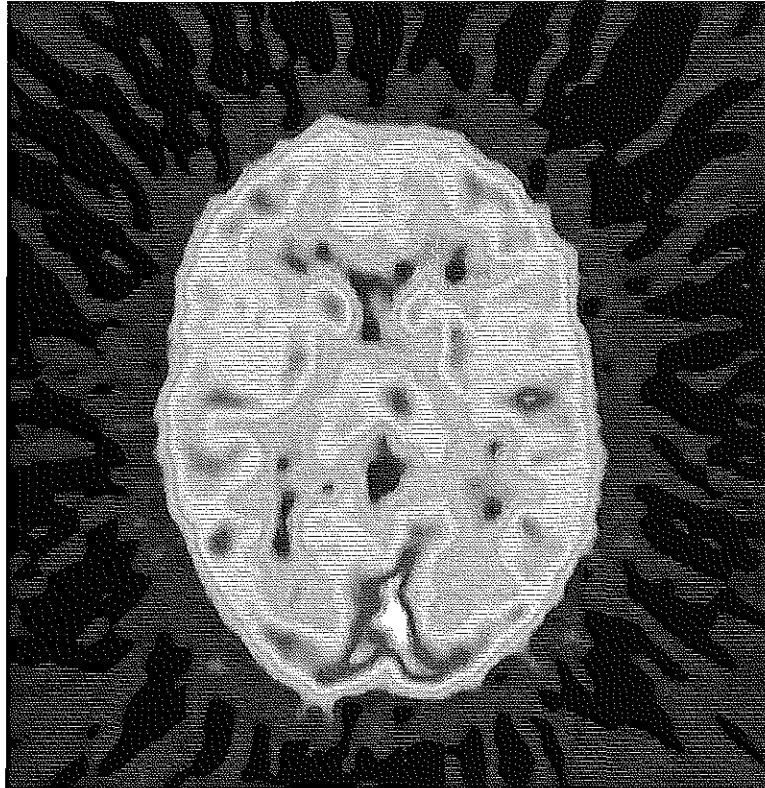


Abbildung 6.6: Die PET-Messung unter Verwendung eines radioaktiv markierten Zuckers zeigt Aktivitäten im Gehirn. Deutlich zu erkennen sind der auditorische und der visuelle Cortex.

2. Wie ist die relative Lage von Arealen untereinander? Liegen verschiedene Areale nebeneinander, oder liegen sie in einer bestimmten Anordnung vor?
3. Wie ist die relative Lage von Arealen bzgl. wichtiger anatomischer Hirnstrukturen, wie z. B. markanter Sulci oder Gyri? Z. B.: Liegen gewisse Areale bei verschiedenen Menschen immer in derselben Hirnfalte?

Die Beantwortung dieser Fragen unter Zuhilfenahme der anatomischen Volumendatensätze ist nur schwer oder auch gar nicht möglich. So läßt sich der Flächeninhalt von Oberflächenabschnitten nicht direkt im Datensatz messen, da sich der Datensatz aus Voxeln zusammensetzt, welche zunächst nur Informationen über einen Volumeninhalt wiedergeben. Neben der Größe läßt sich aber auch die relative Lage von Arealen im Originaldatensatz nur schwer und ungenau beurteilen. Dies liegt an der extremen Auffaltung und Verwindung der Gehirnoberfläche, welche eine Erfassung der geometrischen Zusammenhänge schwierig macht. Erschwert wird eine Beurteilung auch durch die allgemein übliche Visualisierung der Datensätze.

Erfolgt die Visualisierung in Form von Schnittbildern, so ist zwar das Innere der Gehirnfalten einsehbar, aber der dreidimensionale Zusammenhang der Gehirnoberfläche ist für den Betrachter gedanklich nur schwer zu rekonstruieren. Eine Schwierigkeit ergibt sich daraus, daß das Gehirn, obwohl es ein zusammenhängender Körper ist, in den unter-

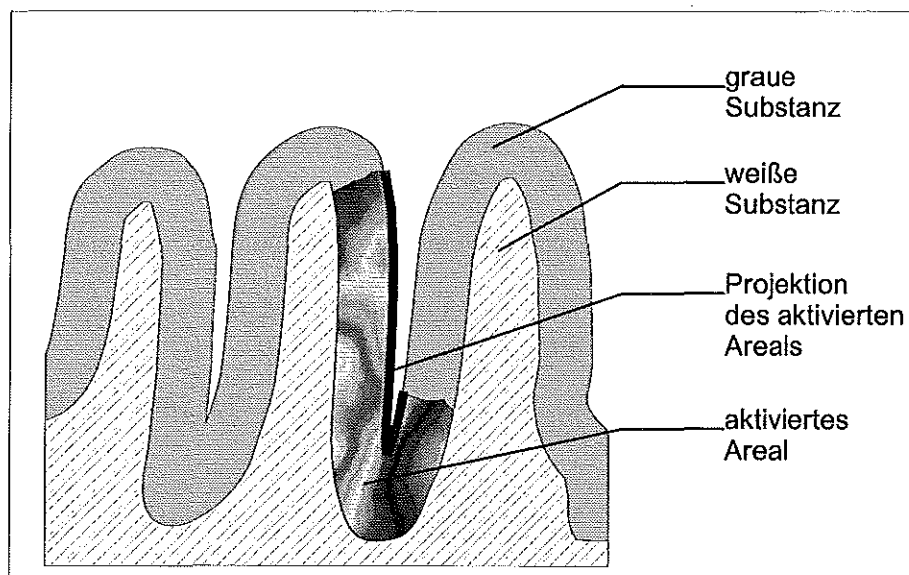


Abbildung 6.7: Durch Projektion von in der Großhirnrinde liegenden aktiven Arealen wird eine zweidimensionale Abbildung dieser Areale erzeugt.

schiedlichen Schnittbildern in mehrere, unzusammenhängende Teilobjekte zerfallen kann, wie in Abbildung 6.8 zu sehen ist.

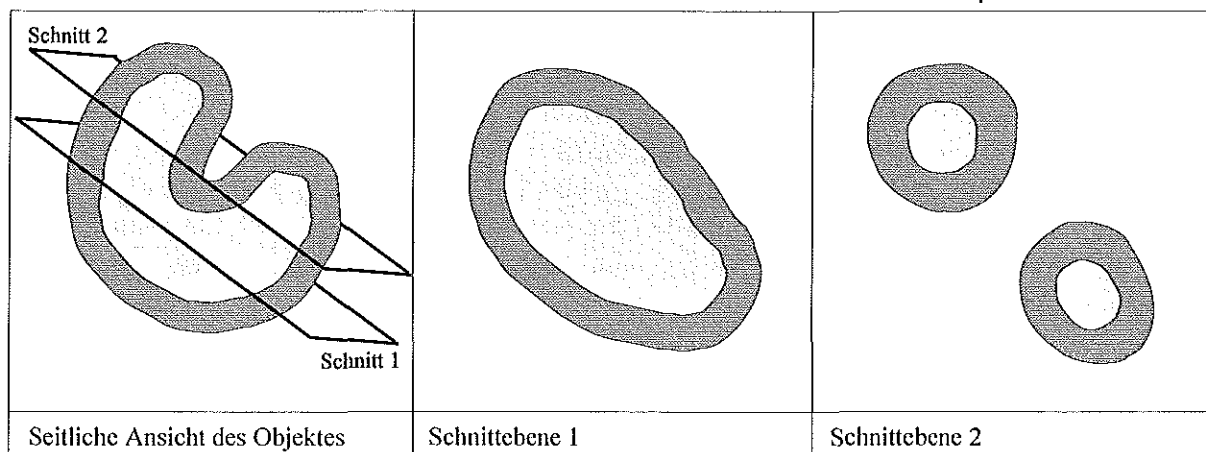


Abbildung 6.8: Ein dreidimensional zusammenhängendes Objekt zerfällt durch die Darstellung in Schnittebenen in unzusammenhängende Teilobjekte.

Erfolgt eine Visualisierung durch Darstellung einer Aufsicht auf das gesamte Hirn, etwa in Form eines Oberflächenrenderings, so ist die Oberflächengeometrie des Hirns für den Betrachter leichter vorstellbar, jedoch ist in dieser Darstellung nur ein Teil der Gehirnoberfläche wirklich sichtbar, nämlich der außen auf den Gyri liegende Oberflächenteil. Ein großer Teil der Gehirnoberfläche liegt jedoch in den Gehirnfalten verborgen und entzieht sich somit einer direkten Betrachtung.

Vereinfacht wird die Beantwortung der oben aufgeführten Fragen durch die Erstellung einer Karte der Gehirnoberfläche. Unter solch einer Karte soll die eindeutige Abbildung der gefalteten Oberfläche des Gehirns auf eine flache und somit übersichtlich darstellbare Kartenoberfläche verstanden werden. Auf der einmal berechneten Kartenoberfläche ist eine eindeutige Zuordnung von Punkten auf der Karte zu Punkten auf der Gehirnoberfläche möglich. Auf der Karte selber können sowohl

- *funktionelle Daten*, z. B. die Lage von aktivierten Arealen, als auch
- *anatomische Daten*, z. B. die Lage von Gyri und Sulci oder auch die Krümmung der Gehirnoberfläche, eingezeichnet werden.

Damit die Transformation vom Gehirn zur Karte einen Sinn ergibt, müssen zwei Nebenbedingungen eingehalten werden:

1. Nachbarschaftsbeziehungen zwischen Punkten auf der Gehirnoberfläche müssen erhalten bleiben.
2. Die bei der Transformation auftretenden Verzerrungen sind zu minimieren. Dies bedeutet, daß *Abstände*, *Flächen* und *Winkel* bei der Transformation erhalten bleiben sollen.

Die erste Nebenbedingung ist eine Art Minimalanforderung und beschreibt die Stetigkeit der Transformation. Durch sie wird garantiert, daß Areale, welche auf der Gehirnoberfläche nebeneinander liegen, auch auf der Karte direkt benachbart sind. Aneinanderliegende Areale werden somit bei der Transformation nicht auseinandergerissen, es schiebt sich keine Fläche zwischen sie.

Die zweite Bedingung soll die bei der Transformation auftretenden Verzerrungen minimieren. Eine vollkommen verzerrungsfreie Abbildung der Gehirnoberfläche auf eine Kartenoberfläche ist nicht möglich. Form und Größe von Arealen sollen auf der Karte jedoch so gut wie möglich erhalten bleiben, was durch eine Verzerrungsminimierung geschieht. Anzumerken ist an dieser Stelle, daß alle auftretenden Verzerrungen bekannt sind. Somit kann die tatsächliche Größe eines Areals jederzeit berechnet werden, auch wenn es auf der Karte verzerrt wiedergegeben wird.

6.2.1 Vorgehensweise

Um die Abbildung der Gehirnoberfläche auf eine Karte zu berechnen, gibt es zwei ähnliche, aber in ihrer Vorgehensweise genau konträre Wege. Für beide Wege ist es zunächst erforderlich, daß ein mathematisch beschreibbares Modell der im Volumendatensatz durch eine Voxelmenge beschriebenen Oberfläche gefunden wird. In solch einem Modell wird die Oberfläche beschrieben durch ein Oberflächennetz. Die gesuchte Abbildung wird anhand des Oberflächennetzes berechnet.

Das erste Verfahren, welches von D. C. van Essen et al. entwickelt wurde ([DEALCL95], [CDE95]), berechnet die Karte, indem ein von der Gehirnoberfläche entworfenes Netz auf eine ebene Gestalt gebracht wird. Es läuft in den folgenden drei Schritten ab:

1. Manuelle Bestimmung der Konturen der Gehirnoberfläche in allen NMR-Schnittbildern.
2. Bildung eines Oberflächennetzes durch Vernetzung der Konturen.
3. Berechnung der Karte durch Anordnung der Netzknoten auf eine ebene Fläche.

Im ersten Schritt werden die Voxel, welche zur gesuchten Oberfläche gehören, markiert. Dies erfolgt manuell. Schritt zwei geschieht in einem halbautomatischen Verfahren. Der in [Gei93] vorgestellte Algorithmus nimmt eine Vernetzung der Konturen vor. Diese Vernetzung ist jedoch nicht immer korrekt, so daß eine manuelle Nachbearbeitung erforderlich ist, in der falsch geknüpfte Punktverbindungen gelöst und fehlende Punktverbindungen geknüpft werden müssen. Nach Ausführung dieser Schritte liegt ein Oberflächennetz vor, welches in Schritt drei auf die gewünschte Karte transformiert werden kann. Diese Transformation entspricht einer Entfaltung des Netzes im dreidimensionalen Raum unter Berücksichtigung der Verzerrungsminimierung. In [DEALCL95] wird zudem eine optimierte Version dieser Methode vorgestellt, in welcher die Netzknoten zunächst ohne Berücksichtigung der Verzerrungen auf eine Ebene aufgebracht werden. Erst in einem späteren Schritt werden die Verzerrungen durch Verschieben der Netzknoten innerhalb der Ebene minimiert. Diese Vorgehensweise hat den Vorteil, daß die Minimierungsaufgabe um eine Dimension erniedrigt wurde, was den Rechenvorgang beschleunigt.

Die Methode, welche in dieser Arbeit vorgestellt wird, stellt eine Umkehrung des obigen Verfahrens dar. Sie führt vom Kartennetz über dessen Deformation zum Gehirnnetz. Die dazu notwendigen Schritte sind:

1. Manuelle Bestimmung der Konturen der Gehirnoberfläche in allen NMR-Schnittbildern.
2. Generierung des Kartennetzes.
3. Anpassung des Kartennetzes an die Gehirnoberfläche durch Deformation der Karte.

Bei dem hier beschriebenen Verfahren wird die Vernetzung der Gehirnkonturen eingespart, da in Schritt zwei nicht das Netz des Gehirns, sondern das Netz der Karte erzeugt wird. Da dieses Netz bekannt ist, kann es mittels eines Algorithmus automatisch, ohne manuelles Eingreifen, erzeugt werden. In Schritt drei wird das Kartennetz an die Gehirnoberfläche angepaßt. Da diese Transformation eine umkehrbare Abbildung beschreibt, ist somit eine Zuordnung von Punkten auf der Gehirnoberfläche zu Kartenpunkten möglich. Die Deformation selber vollzieht sich im dreidimensionalen Raum. Unter der Voraussetzung, daß die Oberfläche einer Gehirnhälfte topologisch äquivalent zu der Oberfläche einer Kugel ist, wird als Kartennetz das in Kapitel 2.1 beschriebene diskretisierte Netz der Kugeloberfläche genommen. Das Endresultat der Kartenanpassung stellt somit eine Art „Globus“ der Gehirnoberfläche dar.

Der Rechenaufwand der Netzdeformation ist hier größer als der Aufwand des in [DEALCL95] vorgestellten Verfahrens, da die Transformation in [DEALCL95] nur im zweidimensionalen Raum stattfindet. Es muß jedoch berücksichtigt werden, daß die hier vorgestellte Netzdeformation im dreidimensionalen Raum ein universelles Verfahren darstellt,

für welches das Cortical Flattening nur ein Anwendungsbeispiel ist. Es gewinnt seine Universalität durch die Fähigkeit, selbstständig Flächen im dreidimensionalen Raum zu finden.

6.2.2 Datenmaterial und Datenvorverarbeitung

Als Ausgangsbasis für die Anwendung des Verfahrens dienen NMR-Aufnahmen eines post-mortem Gehirnes. Das Gehirn wurde in einem 1.5 T NMR-Scanner gemessen. Die Verwendung des post-mortem Gehirnes hat den Vorteil, daß das Gehirn frei vorliegt und nicht von anderen Geweben oder Knochen umgeben ist. Ein Entfernen dieser Gewebearten aus dem Datenmaterial, das sogenannte *peeling* des Gehirns, konnte somit entfallen.

Da die Abbildung der Kugeloberfläche auf die Gehirnoberfläche für eine Gehirnhälfte berechnet werden soll, wurde zunächst die linke Seite des Gehirns aus dem Datensatz entfernt. Dies geschah mit einem semiautomatischen Verfahren. Dazu gibt der Benutzer die Parameter einer Ebene an, welche im Datensatz liegt, und die beiden Gehirnhälften so gut wie möglich trennt. Die Daten, welche auf der einen Seite dieser Ebene liegen, sind ungefähr die Voxel, welche zu einer Gehirnhälfte gehören. Diese werden ausgeblendet. Da die Trennfläche der beiden Gehirnhälften sich nur ungenau durch die Angabe einer Ebene beschreiben läßt, ist die Ebene bei ihrer Definition so zu legen, daß zumindest keine Voxel der Gehirnhälfte, welche nicht ausgeblendet werden soll, auf der falschen Ebenenseite liegen. Die Ebene wird also so gelegt, daß nach dem Löschen einer Datenhälfte im Datensatz auch Voxel zurückbleiben, die zur Gehirnhälfte gehören, die gelöscht werden sollte. Diese werden in einem zweiten Schritt manuell gelöscht. Letztendlich ergibt sich somit ein Datensatz, in welchem nur noch eine Gehirnhälfte dargestellt ist.

In Kapitel 4, in welchem die äußere, aus den Bilddaten berechnete Kraft diskutiert wurde, wurde bereits darauf hingewiesen, daß das Datenmaterial zur Verwendung der hier benutzten Druckkraft in drei Voxeltypen unterteilt vorliegen muß, nämlich in Voxeln, welche innerhalb des Objektes (hier des Gehirns), außerhalb des Objektes und auf dem Rand des Objektes liegen. Um diese Unterteilung vorzunehmen, wurde hier ein einfaches Schwellenwertverfahren benutzt, welches das Innere und Äußere des Objektes schon recht gut trennen kann. Lediglich an Stellen, an denen Gehirnfalten direkt aneinanderstoßen, ist eine manuelle Bearbeitung erforderlich. In diesem Bearbeitungsschritt muß an den Stellen, an denen Gehirnfalten durch den Schwellenwertalgorithmus zu einem Objekt „verschmolzen“ sind, eine künstliche Trennlinie im Voxelraum eingezeichnet werden. Das ist in Abbildung 6.9 dargestellt. Diese Nachbearbeitung erfolgt schichtweise. Die Schichten müssen dabei in allen möglichen Schnittrichtungen, also transversal, sagittal und coronal, begutachtet werden. Randvoxel können danach automatisch erkannt werden. Es sind die zum Gehirn gehörenden Voxel, welche mindestens ein direktes Nachbar-Voxel haben, welches außerhalb des Gehirns liegt.

Speziell im Fall des Gehirns gibt es noch eine andere Möglichkeit, die gesuchte Oberfläche zu bestimmen. Hierbei wird die spezielle Anatomie des Gehirns ausgenutzt. Diese Methode wurde nicht implementiert, da keine Segmentierung der Daten vorhanden war, soll hier aber der Vollständigkeit halber besprochen werden. Sie beruht auf der Überlegung, daß die Grenzschicht zwischen weißer und grauer Hirnsubstanz über eine Segmentierung eindeutig bestimmt werden kann und eine Art „Abbild“ der eigentlich gesuchten

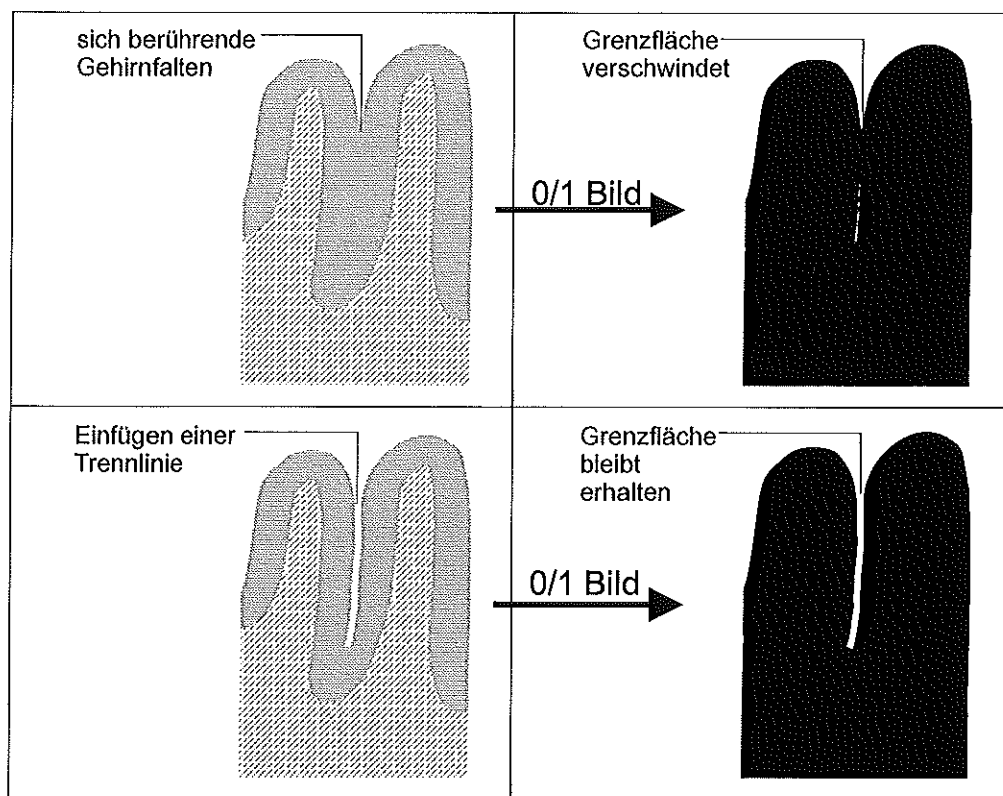


Abbildung 6.9: Bei Verwendung eines Schwellenwertverfahrens zur Unterscheidung von Gehirnninnerem und -äußerem kann es vorkommen, daß sich berührende Falten miteinander verschmelzen, ohne eine Grenzlinie zu bilden (oben). In diesen Fällen wird eine künstliche Grenzlinie von Hand eingezeichnet (unten).

äußeren Gehirnoberfläche, also der äußeren Oberfläche der grauen Substanz, darstellt. Dazu wird vorausgesetzt, daß die graue Substanz an allen Stellen des Gehirns eine (ungefähr) konstante Dicke d hat. Dann kann die Oberfläche der grauen Substanz aus der Grenzfläche weiße zu graue Substanz rekonstruiert werden, indem eine Schicht der Dicke d auf diese Grenzfläche „aufgelegt“ wird. Genauer gesagt kann ein auf der äußeren Oberfläche der grauen Substanz liegender Punkt gefunden werden, indem von einem auf der Grenzfläche weiße/graue Substanz gelegenen Punkt ausgehend ein Schritt der Länge d in Richtung der nach außen zeigenden Oberflächennormalen gegangen wird. Die Oberflächennormale ist zunächst nicht bekannt. Sie kann jedoch approximiert werden durch die Oberflächennormale des Oberflächennetzes. Hier wird angenommen, daß das Netz während der Deformation eine Approximation der gesuchten Oberfläche darstellt. Ein ähnliches Verfahren wird auch in [FCC96] verwendet, um die Zellwände von sich berührenden Nervenzellen zu trennen.

Der Vorteil dieses Verfahrens liegt darin, daß, begründet durch die spezielle Anatomie des Gehirns, die Oberflächenabschnitte der weißen Substanz auch in Gehirnfalten nie aneinander stoßen, da sie immer durch die darüberliegende Schicht der grauen Substanz voneinander getrennt werden. Die an einem Netzpunkt P angreifende äußere Kraft $D(P)$,

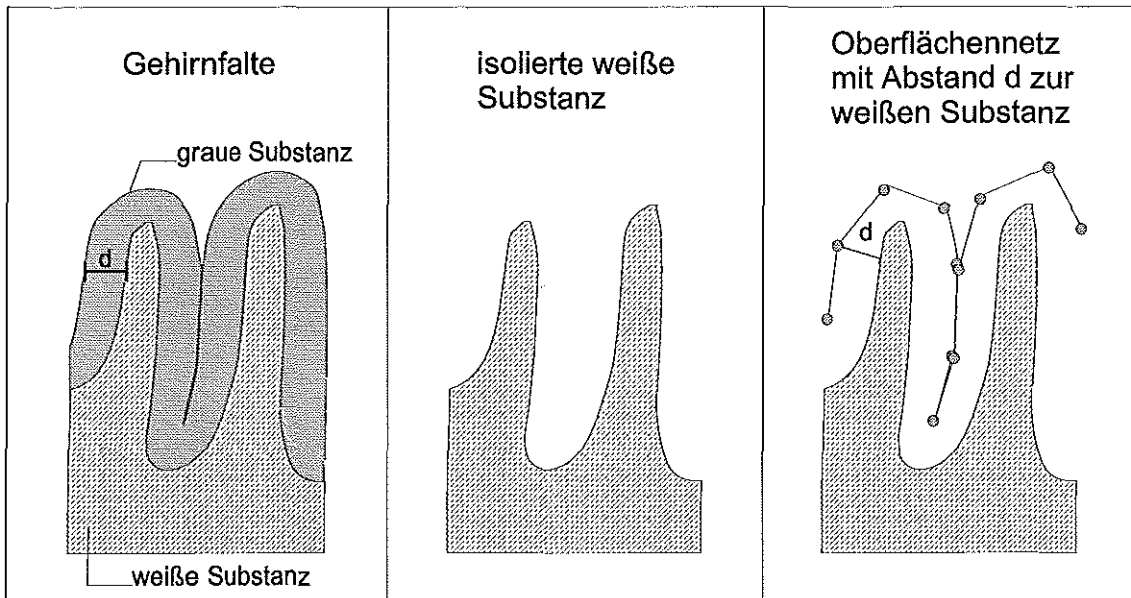


Abbildung 6.10: Liegt eine Segmentierung der Daten in graue und weiße Substanz vor, so kann die aktive Oberfläche so gelegt werden, daß sie einen Abstand d zur weißen Substanz hat. d ist hierbei gerade so zu wählen, daß es der Dicke der grauen Substanz entspricht. Mit dieser Methode können Probleme umgangen werden, die durch aneinanderstoßende Gehirnfalten entstehen.

die Druckkraft, muß in diesem Fall entsprechend angepaßt werden:

$$D(P) = \begin{cases} cN(P), & \text{falls } (P - dN(P)) \text{ innerhalb der weißen Substanz liegt,} \\ -cN(P), & \text{falls } (P - dN(P)) \text{ außerhalb der weißen Substanz liegt,} \\ 0, & \text{falls } (P - dN(P)) \text{ auf dem Rand der weißen Substanz liegt.} \end{cases}$$

$N(P)$ bezeichnet hier die nach außen gerichtete Oberflächennormale am Punkt P .

6.3 Ergebnisse

Mit dem in Kapitel 2 vorgestellten Ansatz konnte eine Karte der Oberfläche einer Gehirnhälfte erzeugt werden. Dazu wurde ein kugelförmiges Netz innerhalb der Gehirnhälfte platziert und dann unter Verwendung der Druckkraft „aufgeblasen“. Das initiale Netz hatte eine Feinheit von 2562 Punkten, was einer Elementanzahl von 5120 Dreieckselementen entspricht. Während des Deformationsprozesses wurde das Netz schrittweise verfeinert, und zwar bis zu maximal 40962 Netzpunkten (=81920 Elementen). Ein Verfeinerungsschritt auf Iterationsstufe $n + 1$ wurde immer dann durchgeführt, wenn die Form des Netzes der Iterationsstufe n durch den Deformationsvorgang nur noch wenig verändert wurde. Dies ist ein Hinweis dafür, daß sich das (grobe) Netz der Stufe n so gut wie

möglich der Gehirnoberfläche angepaßt hat, und eine noch bessere Anpassung nur durch eine Verfeinerung des Netzes erreicht werden kann.

Der Wichtungsfaktor, welcher für die Glattheit des Oberflächennetzes verantwortlich ist, muß während des Deformationsvorganges immer weiter verkleinert werden. Dies ist notwendig wegen der speziellen Form der Gehirnoberfläche, die sich durch die vielfältigen Faltungen auszeichnet, die an vielen Stellen eine sehr große Krümmung aufweisen. Am Anfang der Berechnung ist die Forderung nach Glattheit des Oberflächennetzes eine günstige Eigenschaft, um unerwünschte Auffaltungen des Netzes, welche für eine Grobanpassung nicht nötig sind, zu vermeiden. Um wirklich jede Gehirnfalte zu finden, muß jedoch im späteren Verlauf der Berechnung die Forderung nach einer glatten Oberfläche wieder (teilweise) fallengelassen werden. Zu beachten ist hierbei, daß für die Erstellung der Karte die Glattheit der Netzoberfläche ohnehin keine Rolle spielt, sondern vielmehr die Forderungen des Verzerrungsausgleiches hier die maßgeblichen Faktoren sind. Die Abbildung 6.11 zeigt einige Stufen der Netzdeformation, visualisiert durch ein Oberflächenrendering. Die Rechenzeit von der Ausgangsform bis zur Endform betrug ca. zwei Tage.

Der Sinn der Kartenerstellung liegt darin, daß anatomische und funktionelle Daten auf der Karte eingezeichnet werden können. Diese Möglichkeit wird beispielhaft in Abbildung 6.12 wiedergegeben. Zum einen können Eigenschaften der Gehirnanatomie visualisiert werden. Hier wurde die Eigenschaft, ob ein auf der Gehirnoberfläche liegender Punkt auf der von außen sichtbaren Gehirnoberfläche oder in einer Gehirnfalte verborgen liegt, durch Farben markiert. Aber auch Gebiete, welche nicht durch ihre Anatomie, sondern durch ihre Funktion ausgezeichnet sind, können auf der Karte dargestellt werden, wenn sie auf der Gehirnoberfläche markiert sind.

Deformation Kugel auf Gehirn

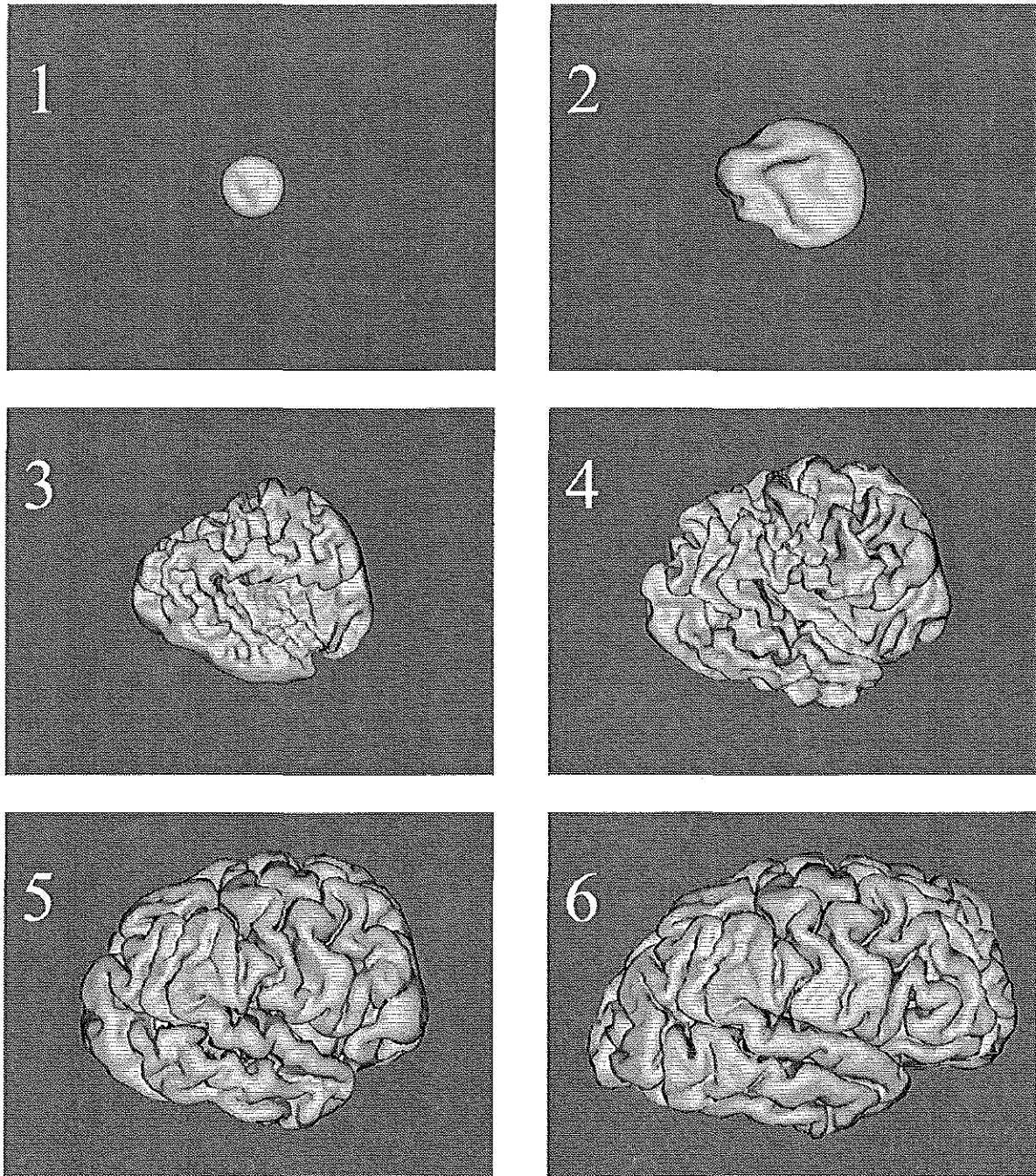


Abbildung 6.11: Sechs Schritte des Anpassungsprozesses von Kugel- auf Gehirnoberfläche. Die Kugel wird in das Innere des Gehirns gelegt (oben links). Durch Anwendung der Druckkraft stülpt sie sich immer mehr in das Gehirn ein, bis sie die Form der Gehirnoberfläche angenommen hat (unten rechts).

Projektion anatomischer oder funktionaler Daten

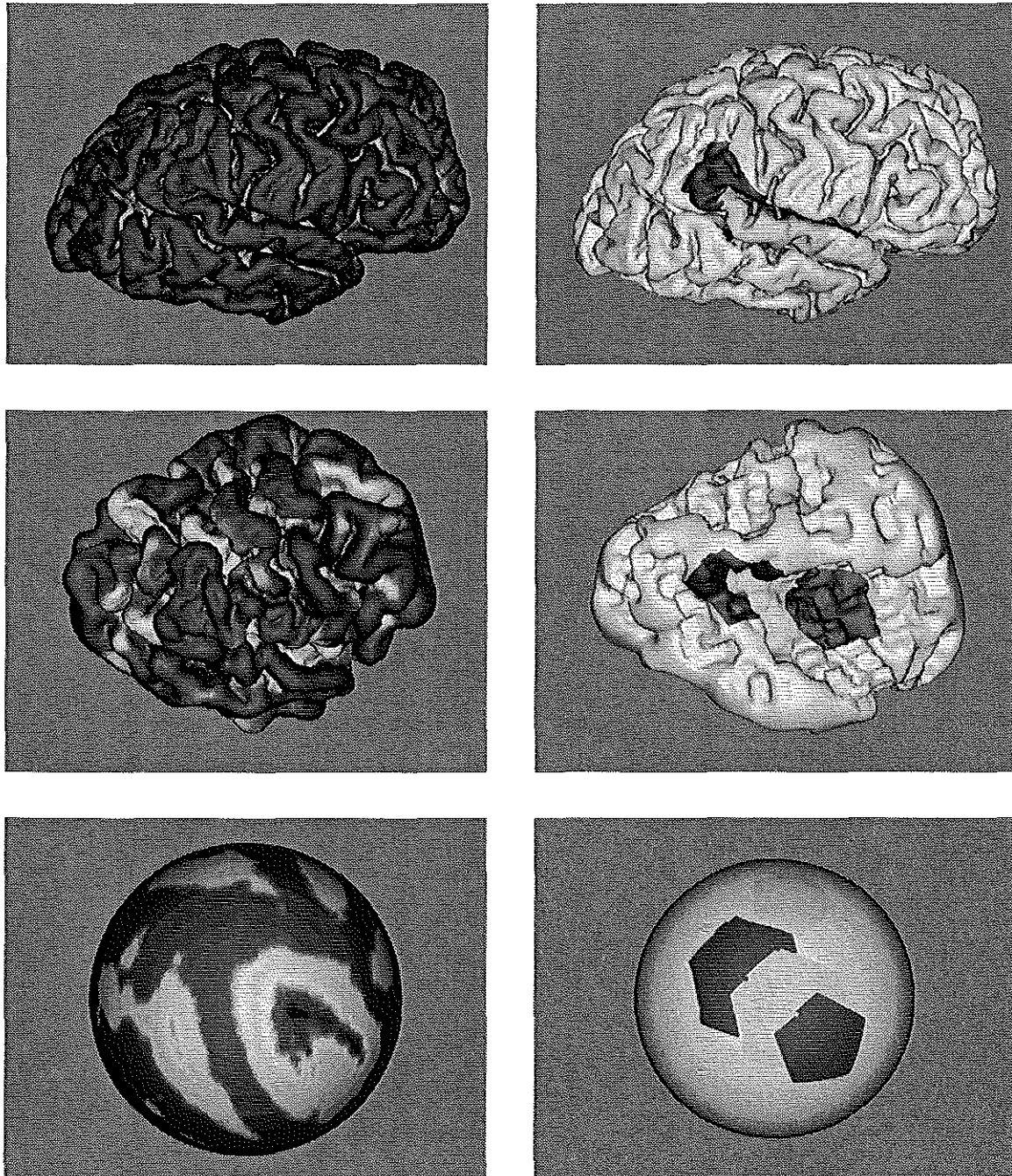


Abbildung 6.12: Links wurden anatomische Daten der Gehirnoberfläche visualisiert. Der Grauwert der Flächen spiegelt wider, wie tief ein Punkt in einer Gehirnfalte liegt. Rechts ist dargestellt, daß die Lage von markierten Arealen auf der Karte wiedergegeben werden kann.

Wie bereits erwähnt, kann eine Transformation der Gehirnoberfläche auf eine Karte nicht ohne Verzerrungen erfolgen. Die Größe der Verzerrungen wurde hier über die Flächeninhalte der Dreieckselemente gemessen und ist in Abbildung 6.13 wiedergegeben.

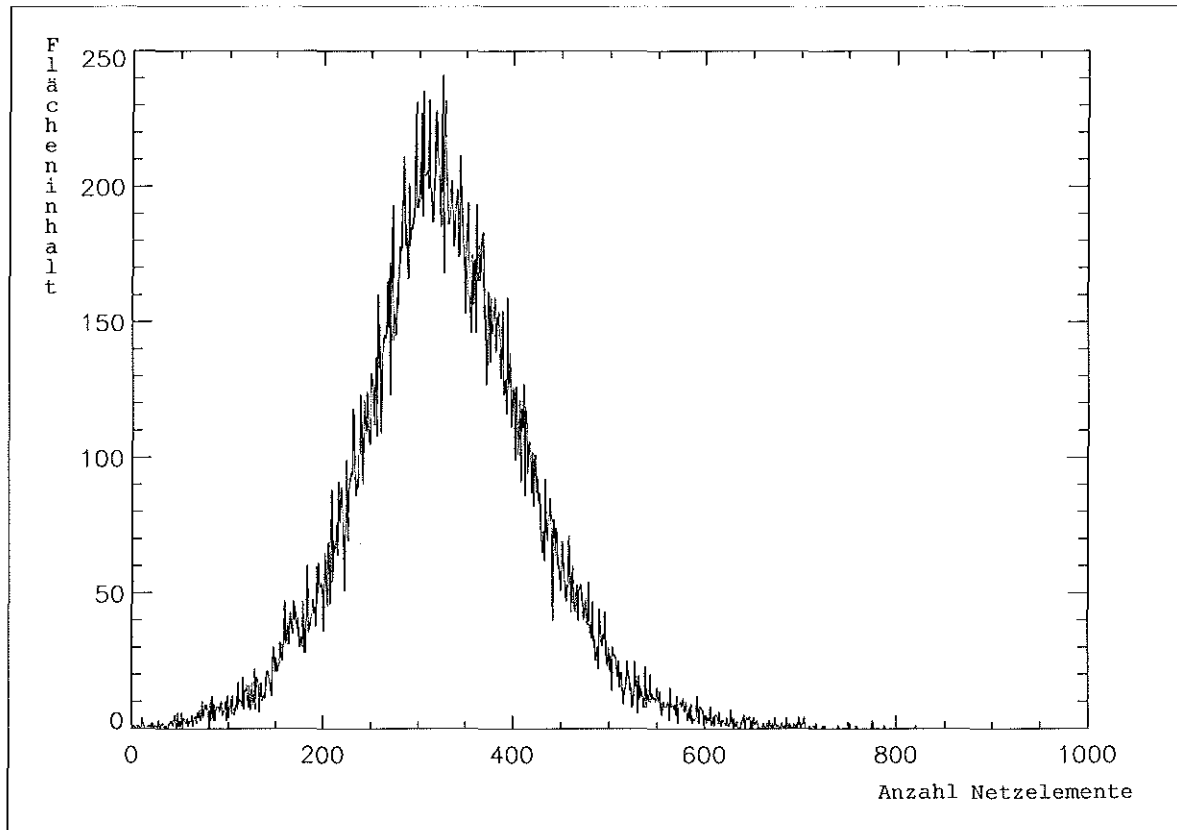


Abbildung 6.13: Die bei der Deformation auftretenden Verzerrungen können gemessen werden und sind hier in einem Histogramm dargestellt. Der mittlere Flächeninhalt beträgt 337, die Standardabweichung 91.

Auch die in Kapitel 3 vorgestellte Methode wurde implementiert. Um ihre 'Tauglichkeit' für die Rekonstruktion von Oberflächen zu demonstrieren, wurde ein Kopf-Phantom, d. h. ein Kunststoff-Kopf, welcher innen hohl ist, im Kernresonanztomographen gemessen. Die Oberfläche des Kopfes konnte mit dem implementierten Verfahren rekonstruiert werden, wie Abbildung 6.14 zeigt. Die abgeflachte Stelle links oben am Kopf ist nicht auf einen Fehler im Algorithmus zurückzuführen, sondern liegt daran, daß die vom Kopf gemessenen Schnittbilder an dieser Stelle aufhören. Über der Nase und in der Nähe der beiden Ohren sind kleine, am Kopf angebrachte Markierungsstifte zu erkennen, die ebenfalls rekonstruiert wurden. Im unteren, rechten Bild ist zu sehen, daß der Kopf von innen hohl ist: das Netz stülpt sich unter Einwirkung der Druckkraft in den offenen Halsansatz hinein.

Rekonstruktion der Oberfläche eines Kopf-Phantoms

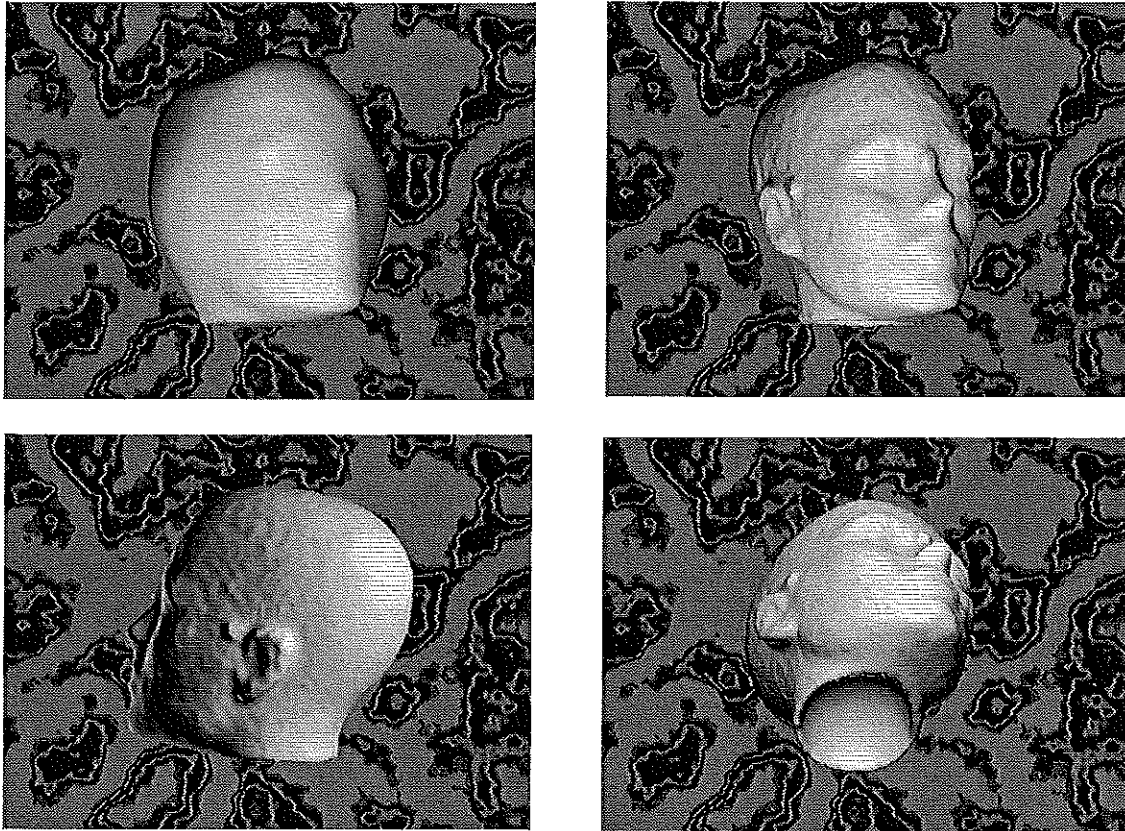


Abbildung 6.14: Die Oberfläche eines im NMR gemessenen Kopf-Phantoms wurde mit Hilfe des kontinuierlichen Oberflächenmodells rekonstruiert. Dazu wird eine Haut über den Kopf gelegt (oben links), die sich während des Deformationsprozesses zusammenzieht.

Kapitel 7

Zusammenfassung

In dieser Arbeit wurde die Möglichkeit aufgezeigt, mit Hilfe von aktiven Oberflächenmodellen eine Abbildung der Oberfläche des Gehirns auf die Oberfläche einer Karte unter Einhaltung von gewissen Nebenbedingungen zu berechnen. Es wurden zwei verschiedene Wege aufgezeigt, ein Oberflächenmodell einschließlich der notwendigen Energieterme zu definieren. Einmal wurden sowohl die Oberfläche selber als auch die Energieterme unmittelbar über diskrete Parameter definiert. Zum anderen wurde ein kontinuierliches Modell vorgestellt, in welchem die Maße für die Membranverzerrung, basierend auf kontinuumsmechanischen Theorien, analysiert wurden. Diese Analyse führte schließlich zu einer erweiterten Modellierbarkeit der Energieterme, was zu einer verbesserten Anpassungsmöglichkeit der gewünschten Netzeigenschaften führt. Die Diskretisierung dieses kontinuierlichen Modells erfolgte über finite Elemente. Beide Verfahren wurden entwickelt, implementiert und getestet.

Darüber hinaus wurde eine parallele Version der Algorithmen realisiert, um die langen Rechenzeiten zu verkürzen. Diese parallele Version wurde entwickelt unter Verwendung des Tools *pvm* und ist auf einem heterogenen Rechnernetz lauffähig.

Aus dieser Arbeit wurden Aspekte der Datenvorverarbeitung weitgehend ausgeklammert. Es wurde vorausgesetzt, daß der Datensatz des Gehirns in einer für den Algorithmus verwertbaren Form vorliegt. Einige zur Vorverarbeitung nötigen Schritte wurden in Abschnitt 6.2.2 beschrieben. Im Rahmen dieser Arbeit waren viele manuelle Schritte zur Bearbeitung des verwendeten Testdatensatzes nötig. Ein möglicher Weg zur vollautomatischen Vorverarbeitung und damit ein Schritt zur Integration des hier vorgestellten Verfahrens in ein Gesamtkonzept ist folgendermaßen denkbar: Falls eine gute Segmentierung der Daten vorliegt, die insbesondere auch den Schädelknochen gut vom restlichen Gewebe trennt, dann kann das Peeling der Daten automatisch geschehen. In dem daraus resultierenden Datensatz liegt nur das Gehirn selber vor, störende, andere Gewebe wurden entfernt. Durch eine Segmentierung in graue und weiße Substanz wäre es weiterhin möglich, die in Abschnitt 6.2.2 erklärte modifizierte Druckkraft zu verwenden. Zu der Eingliederung in ein Gesamtkonzept gehört auch die Evaluierung der Programmlaufzeiten auf einem massiv parallelen System, wie es im Forschungszentrum Jülich durch die CRAY T3E vorhanden ist. Durch eine weitere Senkung der Rechenzeiten ist eine erhöhte Akzeptanz des Verfahrens im medizinisch wissenschaftlichen Umfeld zu erwarten.

Des weiteren ist ein sinnvoller nächster Schritt, das Verfahren an einem anatomischen

Datensatz anzuwenden, zu dem auch funktionelle Daten vorliegen. Damit könnte die in Abschnitt 6.2 vorgestellte Methode, die Daten aktiver Areale auf die Gehirnoberfläche zu projizieren und auf der Karte darzustellen, an echten Daten erprobt werden.

Eine weitere offene Fragestellung besteht darin, ob es möglich und sinnvoll ist, Gehirnkarten unterschiedlicher Probanden in einem zusätzlichen Deformationsprozeß aneinander anzugleichen. Dazu ist zunächst einmal die Frage zu diskutieren, aufgrund welcher Merkmale Kräfte zu definieren sind, die diese Deformation steuern sollen. Denkbar wäre z. B. ein Abgleich unterschiedlicher Karten, basierend auf der Lage einiger ausgesuchter, wichtiger Sulci. Eventuell ist es auch sinnvoll, daß einige sich anatomisch entsprechende Punkte auf den Karten manuell markiert werden und daß diese Punkte durch die Deformation in Deckung gebracht werden.

Auch wäre es wünschenswert, andere Anwendungsgebiete für die Oberflächenrekonstruktion zu untersuchen. Besonders sinnvoll scheint die Anwendung bei stark verrauschten Daten, wie z. B. dreidimensionalen Ultraschall Datensätzen, zu sein. Hier könnten sich insbesondere modellbasierte Ansätze als gewinnbringend erweisen, die eine Objekterkennung im verrauschten Datensatz durch ein vorhandenes Wissen über die Objektform und andere Objekteigenschaften ermöglichen.

Alle diese Punkte führen über die Absicht dieser Arbeit hinaus, können aber die hier gelegten Grundlagen als Ausgangsbasis nutzen.

Literaturverzeichnis

- [AWJ90] A. A. Amini, T. E. Weymouth und R. C. Jain. *Using Dynamic Programming for Solving Variational Problems in Vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 9, September 1990.
- [Bat90] K.-J. Bathe. *Finite-Elemente-Methoden*. Springer Verlag 1990.
- [BHP46] F. Bloch, W. W. Hansen und M. Packard. *Nuclear induction*. Phys. Rev. 70: 460–474, 1946.
- [BK89] R. Bajcsy und S. Kovacic. *Multiresolution Elastic Matching*. Computer Vision, Graphics, and Image Processing, 46, 1–21, 1989.
- [BLR83] R. Bajcsy, R. Lieberman und M. Reivich. *A Computerized System for the Elastic Matching of Deformed Radiographic Images to Idealized Atlas Images*. Journal of Computer Assisted Tomography, 7(4):618–625, August 1983.
- [BS79] I. N. Bronstein und K. A. Semendjajew. *Taschenbuch der Mathematik*. BSB B. G. Teubner Verlagsgesellschaft, Leipzig, und Verlag Nauka, Moskau, 24. Auflage, 1979.
- [CC93] L. D. Cohen und I. Cohen. *Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, 1131–1147, 1993.
- [CDE95] G. J. Carman, H. A. Drury, und D. C. Van Essen. *Computational methods for reconstructing and unfolding the cerebral cortex*. Cerebral Cortex 5: 506–517, 1995.
- [CLHK96] V. Chalana, D. T. Linker, D. R. Haynor und Y. Kim. *A Multiple Active Contour Model for Cardiac Boundary Detection on Echocardiographic Sequences*. IEEE Transactions on Medical Imaging, Vol. 15, No. 3, 290–298, 1996
- [Coh91] L. D. Cohen. *Note On Active Contour Models and Balloons*. CVGIP: Image Understanding, No. 2, March, 211–218, 1991.
- [Cri91] M. A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons 1991.
- [CRM93] G. E. Christensen, R. D. Rabbitt und M. I. Miller. *A Deformable Neuroanatomy Textbook Based on Viscous Fluid Mechanics*. Invited Paper to the 1993 Conference on Information Sciences and Systems.

- [CRM94] G. E. Christensen, R. D. Rabbitt und M. I. Miller. *3D brain mapping using a deformable neuroanatomy*. Phys. Med. Biol. 39, 609–618, 1994.
- [CSS94] S. Coquillart, W. Straßer und P. Stucki (Hrsg.). *From Object Modelling to Advanced Visual Communication*. Springer-Verlag 1994.
- [DEALCL95] H. A. Drury, D. C. Van Essen, C. H. Anderson, C. W. Lee, T. A. Coogan und J. W. Lewis. *Computerized mappings of the cerebral flattening method and a surface-based coordinate system*. Journal of Cognitive Neurosciences 8: 1–28, 1995.
- [DP95] C. A. Davatzikos und J. L. Prince. *An Active Contour Model for Mapping the Cortex*. IEEE Transactions on Medical Imaging, Vol. 14, No. 1, March 1995.
- [DT85] G. Dhatt und G. Touzout. *The Finite Element Method Displayed*. John Wiley & Sons 1985.
- [EJ93] H. Esser und H. T. Jongen. *Differentialgleichungen und Numerik für Informatiker und Physiker*. Verlesungsskript, Aachener Beiträge zur Mathematik (ABM) Band 8, Verlag der Augustinus Buchhandlung, 2. Auflage, 1993.
- [Ess94] N. Esseling. *Parallele Algorithmen und Werkzeuge für die Zuordnung funktionaler und anatomischer Daten auf der Basis von PET- und MRT-Volumendaten*. Diplomarbeit, RWTH-Aachen, Lehrstuhl für Technische Informatik und Computerwissenschaften, März 1994.
- [Gei93] B. Geiger. *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*. Doktorarbeit, Ecole de Mines de Paris, April 1993.
- [FCC96] Y.-L. Fok, C. K. Chan und R. T. Chin. *Automated Analysis of Nerve-Cell Images Using Active Contour Models*. IEEE Transactions on Medical Imaging, Vol. 15, No. 3, 353–368, June 1996.
- [FL93] C. Farhat und M. Lesoinne. *Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics*. Internat. J. Numer. Meth. Engng., Vol. 36, No. 5, 745–764, 1993.
- [GBDJMS94] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek und V. Sundaram. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing*. The MIT Press, 1994.
- [GGM74] A. N. Garroway, P. K. Grannel und P. Mansfield. *Image formation in NMR by selective irradiative process*. J. Phys. [C] 7: L457–L462, 1974.
- [Hor91] B. Horwitz. *Functional Interactions in the Brain: Use of Correlations Between Regional Metabolic Rates*. Journal of Cerebral Blood Flow and Metabolism, 11:A114–A120, 1991.
- [Hor94] B. Horwitz. *Data Analysis for Metabolic-Flow Data: Combining Neural Modelling and Functional Neuroimaging*. Human Brain Mapping, 2:112–122, 1994.

- [Jäh97] B. Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, 4. Auflage, 1997.
- [JoSch] H. T. Jongen und P. G. Schmidt. *Analysis*. Skript zur Vorlesung, zweiter Teil. Verlag Augustinus Buchhandlung, ISBN: 3-925038-23-X.
- [KWE75] A. Kumar, D. Welti und R. R. Ernst. *NMR Fourier zeugmatography*. J. Magn. Reson. 18: 69–83, 1975.
- [KWT87] M. Kass, A. Witkin und D. Terzopoulos. *Snakes: Active Contour Models*. Int. J. Comput. Vision 1, 321–331, 1987.
- [Lau73] P. C. Lauterbur. *Image formation by induced local interactions: examples employing nuclear magnetic resonance*. Nature 242: 190–191, 1973.
- [LL72] L. D. Landau und E. M. Lifschitz. *Lehrbuch der Theoretischen Physik*. Akademie-Verlag Berlin, 1972.
- [MCAG93] M. I. Miller, G. E. Christensen, Y. Amit und U. Grenander. *Mathematical textbook of deformable neuroanatomies*. Proc. Natl. Acad. Sci. USA, Vol. 90, 11944–11948, Dezember 1993.
- [OV89] W. Oberschelp und G. Vossen. *Rechneraufbau und Rechnerstrukturen*. R. Oldenbourg Verlag München Wien, 1989.
- [PCS89] C. A. Pelizzari, G. T. Y. Chen und D. R. Spelbring. *Accurate three-dimensional registration of CT, PET, and/or MR images of the brain*. Journal of Computer Assisted Tomography, 13(1):20–26, 1989.
- [PTP46] E. M. Purcell, H. C. Torrey und R. V. Pound. *Resonance absorption by nuclear magnetic moments in solid*. Phys. Rev. 69: 37–38, 1946.
- [Rad17] J. Radon. *On the determination of functions from their integrals along certain manifolds*. Ber. Sachs. Akad. Wiss. 29:262–277, 1917.
- [Sim91] H. D. Simon. *Partitioning of unstructured problems for parallel processing*. Comput. Sys. Engrg., Vol. 2, No. 3, 135–148, 1991.
- [Sch84] H. R. Schwarz. *Methode der finiten Elemente*. B. G. Teubner, Stuttgart 1984.
- [SV82] L. A. Shepp und Y. Vardi. *Maximum Likelihood Reconstruction for Emission Tomography*. IEEE Transactions on Medical Imaging, 1(2):113–122, 1982.
- [Ter88] D. Terzopoulos. *The Computation of Visible-Surface Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 10, No. 4, July 1988.
- [TPBF87] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer. *Elastically Deformable Models*. Computer Graphics, 21(4), 205–214, July 1987.
- [TS67] J. Talairach und G. Szikla. *Atlas d'Anatomie Stereotaxique du Telencephale*. Masson & Cie, Paris, 1967.

- [TT88] J. Talairach und P. Tournoux. *Co-Planar Stereotaxic Atlas of the Human Brain*. Thieme Medical Publisher, Inc., New York, 1988.
- [WS92] D. J. Williams und M. Shah. *A Fast Algorithm for Active Contours and Curvature Estimation*. CVGIP: Image Understanding, Vol. 55, No. 1, 14–26, January 1992.
- [VZKW93] D. Vanderstraeten, O. Zone, R. Keunings und L. Wolsey. *Non-deterministic heuristics for automatic domain decomposition in direct parallel finite element calculations*. In R. F. Sincovec, et. al., *Parallel Processing for Scientific Computing*, SIAM, 929–932, 1993.

Danksagung

Ich danke Herrn Prof. Dr. Halling für seine hervorragende Betreuung dieser Arbeit, ohne die sie sicher nicht in dieser Form zustande gekommen wäre. Ebenso gilt mein Dank Herrn Prof. Dr. Müller-Gärtner, der es immer verstanden hat, mir die medizinischen Aspekte der hier behandelten Problematik näher zu bringen.

Auch danke ich Herrn Prof. Dr. Oberschelp für die Übernahme des Referates und für seine fachliche Unterstützung bei der Betreuung dieser externen Arbeit. Für die Übernahme des Koreferates danke ich Herrn Prof. Dr. Hoßfeld.

Gedankt sei auch meinen Kollegen, die diese Arbeit durch ihre Diskussionsbereitschaft sehr bereichern konnten.

Meiner Verlobten, Christiane Cülter, danke ich für das Korrekturlesen der Arbeit sowie für die liebevolle Unterstützung während der Promotion.

Forschungszentrum Jülich



Jül-3649
April 1999
ISSN 0944-2952