

Deformationsanimation mittels Masse-Feder-Systemen in der Computergrafik

Henning Tjaden

2. Januar 2012

Zusammenfassung

In dieser Arbeit wird eine Einführung in die Thematik der Deformation von virtuellen Objekten gegeben. Es werden verschiedene Problemstellungen in diesem Bereich diskutiert und ein Überblick über Verfahren zu deren Lösung präsentiert. Im Speziellen wird auf das Masse-Feder-System (engl. Mass-Spring-System) und dessen Anwendungsgebiete in der Computergrafik eingegangen. Dabei handelt es sich um ein Verfahren, bei dem die physikalischen Eigenschaften der Objekte durch eine endliche Anzahl von Massepunkten, die über gedämpfte Feder mit einander verbunden sind, approximiert werden. Es wird im Detail erklärt, wie die Kräfte innerhalb des Systems berechnet werden und wie es aufgebaut ist.

1 Einleitung

Seit mehr als 20 Jahren beschäftigen sich Menschen damit Modelle zu finden, die das Deformationsverhalten von Materialien unter physikalischen Kräfteinflüssen beschreiben. Besonders im Bereich der Computergrafik werden derartige Modelle häufig benötigt, um virtuelle Objekte anhand ihrer physikalischen Materialeigenschaften dynamisch zu animieren. Da diese Vorgänge in der Realität extrem komplex sind, können sie nicht eins zu eins in einer Computersimulation abgebildet werden. Allein durch die Tatsache, dass jedes Atom seine Nachbaratome beeinflusst wird deutlich, dass so viele Faktoren berücksichtigt werden müssten, dass dies von keinem derzeitigen Computer geleistet werden kann. Die Modelle, die in der Computergrafik verwendet werden, versuchen daher immer unter Vereinfachung der Realität visuell möglichst überzeugende Ergebnisse zu erzielen.

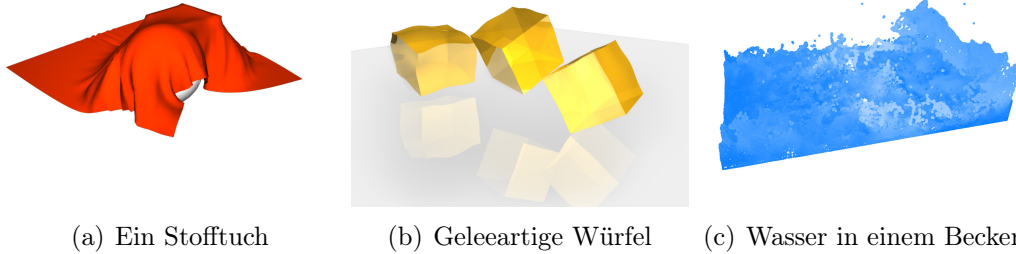


Abbildung 1: *Verschiedene Einsatzgebiete für Deformationsmodelle. Von links nach rechts: Ein virtuelles Stofftuch, das über eine Kugel und einen Würfel drapiert wurde. Drei virtuelle Würfel aus geleeartigem Material, die mit einer Bodenfläche kollidieren. Wasser, das in einem quaderförmigen Becken simuliert wird.*

Diese Arbeit gibt zunächst einen Einstieg in das Themengebiet der Deformationsanimation und deren üblichen Anwendungsgebiete. Es wird näher auf Masse-Feder-Systeme eingegangen, um ein gängiges Deformationsmodell im Detail zu veranschaulichen. Abschnitt 2 erläutert die Idee des Masse-Feder-Systems. Es wird gezeigt, wie die Kräfte innerhalb des Systems berechnet und auf verschiedene Arten zeitlich integriert werden können. Außerdem wird darauf eingegangen, wie die Masse-Feder-Struktur bei planaren, zweidimensionalen und räumlichen, dreidimensionalen Objekten aufgebaut werden kann. Abschnitt 3 stellt Verfahren vor, die Schwächen und Grenzen von konventionellen Masse-Feder-Systemen beheben, indem sie das Ursprungsmodell erweitern. Abschnitt 4 fasst die präsentierten Inhalte abschließend zusammen.

1.1 Anwendungsgebiete in der Computergrafik

Deformationsmodelle werden in der Computergrafik immer dann benötigt, wenn das Verhalten von virtuellen Objekten unter der Wirkung von physikalischen Kräften simuliert werden soll. Die Anwendungsgebiete sind dabei weit gestreut. Sie finden zum Beispiel in der Medizin (Organe bei Operationssimulationen oder virtuelle Muskeln), in der Werkstoffkunde (Autositze, Brückenträger, Crashtest-Simulationen, etc.) oder in der Film- und Videospiegelindustrie (Physik-basierte Animationen) Einsatz.

Die Problemstellungen lassen sich dabei, anhand der Art der zu animierenden Objekte, in drei grobe Kategorien einteilen: Kleidung, Festkörper und Flüssigkeiten. Zu der Kleidungskategorie zählen praktisch alle Objekte, die

eine sehr geringe Dicke haben und daher als planare Strukturen approximiert werden können. Abbildung 1(a) zeigt beispielsweise ein Stofftuch, das über zwei andere Objekte gelegt wurde und dabei einen realistischen Faltenwurf aufweist. Es ist nicht gefordert, dass die Objekte selbst flach sein müssen. Daher gehören zu dieser Kategorie beispielsweise auch Kleidungsstücke und Plastiktüten, obwohl sie eine räumliche Ausdehnung haben. Zu Festkörpern gehören daher alle Objekte, die aufgrund ihrer Dicke in der Realität eine innere Struktur aufweisen würden, die bei der Deformation nicht vernachlässigt werden kann. In Abbildung 1(b) sieht man beispielhaft drei Würfel, die aus einem geleeartigen Material bestehen und sich bei einer Kollision mit der Umgebung entsprechend verformen. Zu der dritten Kategorie gehören neben Flüssigkeiten, wie sie in Abbildung 1(c) zu sehen sind, noch gasförmige und schmelzende Objekte, sowie die Animation von Feuer.

1.2 Überblick über Deformationsmodelle

Je nach Problemstellung können der Deformationsberechnung verschiedene Modelle zu Grunde liegen. Diese können wiederum in Hauptkategorien unterteilt werden. Dabei wird zwischen Lagrangeschen Verfahren (Netz-basiert und nicht Netz-basiert) und Eulerschen Verfahren unterschieden [NMK⁺06].

Netz-basierte Lagrangesche Verfahren werden für die Deformation von Objekten mit einer wohldefinierten Oberfläche verwendet (zum Beispiel Kleidung oder Festkörper). Hierbei wird wiederum unterschieden, ob das Modell von einer kontinuierlichen Masse des Objektes ausgeht oder diese diskretisiert. Populäre Verfahren auf dem Gebiet der Kontinuumsmechanik sind beispielsweise FEM (Finite Element Method) [ZT00], FDM (Finite Difference Method) [TPBF87], FVM (Finite Volume Method) [TBHF03] oder BEM (Boundary Element Method) [Hun05]. Diese Verfahren liefern physikalisch sehr exakte Ergebnisse, sind allerdings auch relativ aufwändig zu berechnen und zu implementieren. Sie werden meist eingesetzt, wenn physikalische Präzision eine übergeordnete Rolle spielt.

Bei diskontinuierlicher/diskreter Massenverteilung können Verfahren wie das Masse-Feder-System eingesetzt werden [ChZK98]. Wie im Folgenden beschrieben vereinfacht dies die physikalischen Zusammenhänge zwar stark, liefert aber dennoch optisch sehr eindrucksvolle und überzeugende Ergebnisse bei relativ geringem Rechen- und Implementieraufwand. Dadurch kann es auch für dynamische Echtzeitberechnungen in interaktiven VR-Anwendungen verwendet werden. Obwohl Masse-Feder-Systeme schon sehr lange existieren, bieten sie immer noch eine schnelle und effiziente Lösung oder zumindest die Grundlage für viele Problemlösungen auf diesem Gebiet [THMG04].

Nicht Netz-basierte Lagrangesche Verfahren werden verwendet, wenn die Objekte keine wohldefinierte Oberfläche aufweisen (zum Beispiel brechende Wellen, Sand oder Wolken). Grundlage dieser Modelle sind sogenannte Partikelsysteme [Ree83, Ton98, LL02]. Für dieselben Probleme können auch Eulersche Verfahren benutzt werden. Diese beschreiben die zu deformierenden Objekte aus einem anderen Blickwinkel. Bei den Lagrangeschen Verfahren wurden die Objekte durch eine Menge von sich bewegenden Punkten beschrieben, die ihre Position abhängig von der Zeit ändern und ihre Materialeigenschaften mit sich tragen. Bei Eulerschen Verfahren betrachtet man eine endliche Anzahl von Punkten mit festen Positionen (zum Beispiel in einem Voxelgitter) und berechnet die Veränderung der Materialeigenschaften an jedem dieser Punkte abhängig von der Zeit [FM96, Car04]. Das Problem, das sich hieraus ergibt, ist, dass dadurch die Grenzen des Objekts nicht mehr explizit definiert sind.

2 Masse-Feder-Systeme

Das Masse-Feder-System geht von einer diskreten Verteilung der Masse eines Objekts aus. Das heißt, dabei wird ein kontinuierlicher Körper durch eine endliche Anzahl von Massepunkten approximiert.

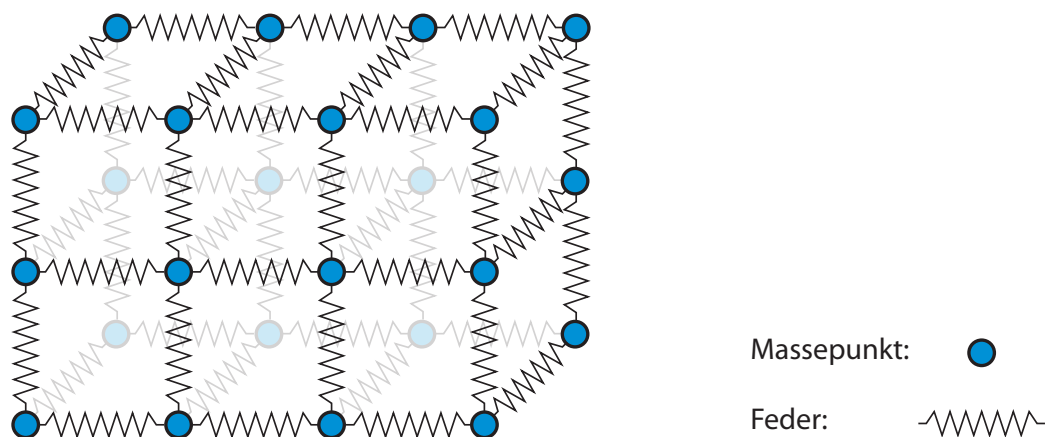


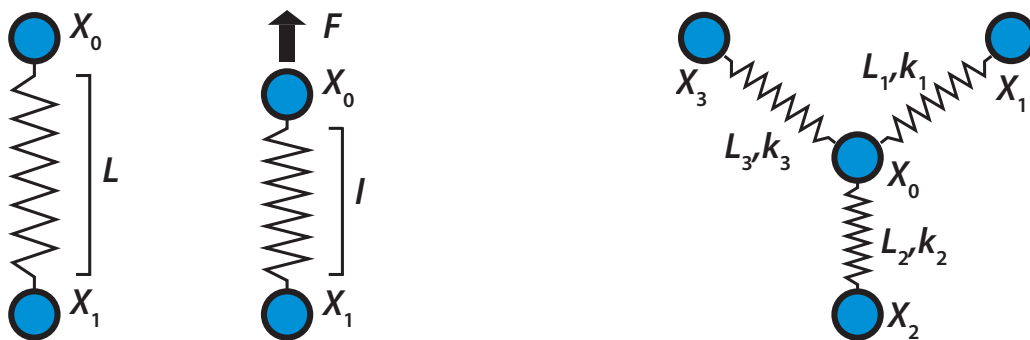
Abbildung 2: Die generelle Struktur eines Masse-Feder-Systems. Benachbarte Massepunkte sind über Federn miteinander verbunden.

Diese Massepunkte sind, wie in Abbildung 2 zu sehen ist, über imaginäre Federn mit ihren Nachbarn verbunden. Die Nachbarschaft ergibt sich

hierbei zum Beispiel aus den topologischen Informationen einer Datenstruktur für ein Polygonnetz. Verschiedene Arten die Punkte mit den Federn zu verbinden werden in Abschnitt 2.3 und 2.4 erläutert. Somit besteht ein Masse-Feder-System, wie der Name schon erkennen lässt, aus Massepunkten und Federn. Wenn nun eine Kraft auf dieses System ausgeübt wird, entstehen im Inneren des Systems Bewegungsvorgänge in Abhängigkeit der Eigenschaften der Federn. Wie diese Kräfte berechnet werden können, wird im folgenden Abschnitt beschrieben.

2.1 Berechnung der Kräfte

Die Kräfte, die in einem Masse-Feder-System entstehen, beruhen auf dem Hookeschen Gesetz, mit dem sich das elstatische Verhalten von Federn beschreiben lässt.



(a) Veranschaulichung des Hookeschen Gesetzes

(b) Kräftewirkung auf einen inneren Massepunkt

Abbildung 3: Kräftewirkung zwischen den Massepunkten in einem Masse-Feder-System.

Abbildung 3(a) veranschaulicht die Kräftewirkung, die zwischen zwei Massepunkten (X_0 und X_1) entsteht, wenn die Feder durch Positionsänderung der Massepunkte gestaucht wird. Aus dem Hookeschen Gesetz ergibt sich für die resultierende Kraft F die Formel:

$$F = k(L - l) \tag{1}$$

Hierbei ist k die Federkonstante (oder auch Steifigkeit), L die initiale Länge der Feder im Ruhezustand und l die Länge der gestauchten Feder. Die Federkraft F verhält sich somit proportional zur Differenz zu ihrer Ausgangslänge.

Daraus lässt sich die folgende Formel für die internen Kräfte F^{int} herleiten, die auf einen Massepunkt X_0 innerhalb des Masse-Feder-Systems wirken, wie in Abbildung 3(b) zu sehen ist:

$$F_0^{int} = \sum_{i \in N(0)} k_i (|X_i - X_0| - L_i) \frac{X_i - X_0}{|X_i - X_0|} \quad (2)$$

Dabei ist $N(0)$ die Menge der Massepunkte in der Nachbarschaft zum Punkt X_0 . Zu den Kräften, die durch Federn verursacht werden, kommen noch externe Kräfte F^{ext} hinzu. Dabei kann es sich beispielsweise um die Schwerkraft oder Kräfte handeln, die durch Kollisionen oder manuelle Manipulation von Massepunkten zustande kommen. Daher berechnet sich die Kraft F_i , die auf einen Massepunkt X_i wirkt, als $F_i = F^{int} + F^{ext}$.

2.2 Zeitliche Integration

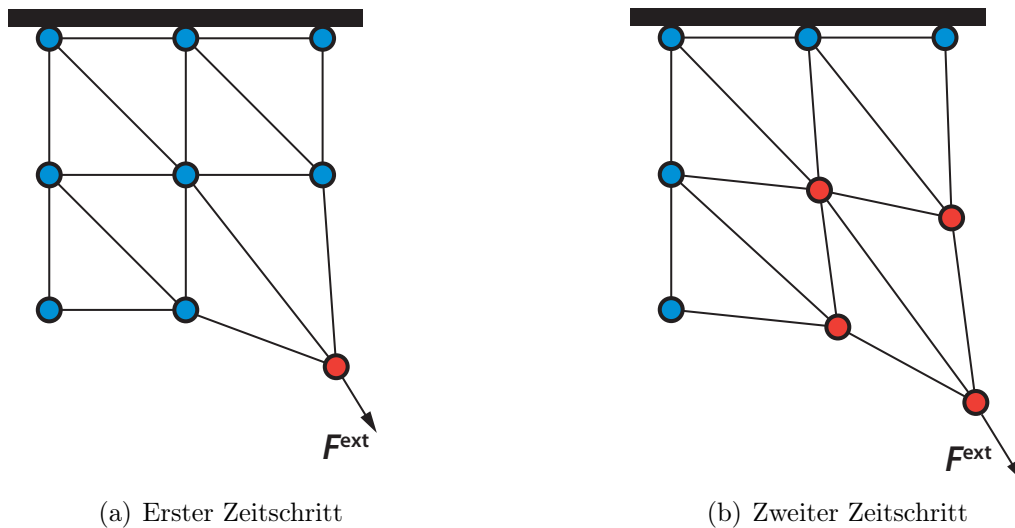
Da es sich bei dem Masse-Feder-System um eine dynamische Struktur handelt, bewegen sich die Massepunkte in Abhängigkeit von der Zeit und den auf sie wirkenden Kräften. Die Bewegung der einzelnen Punkte innerhalb des Systems ergibt sich aus dem zweiten Newtonschen Gesetz ($ma = F$ „Kraft gleich Masse mal Beschleunigung“). Da physikalische Körper sich nicht perfekt elastisch verhalten, geht bei einer Deformation Energie verloren. Um diesen Aspekt mit einzubeziehen, wird zu jeder Feder eine Dämpfung γ modelliert, die der Bewegungsgeschwindigkeit der Massepunkte entgegen wirkt. Daraus ergibt sich die Lagrangesche Bewegungsgleichung:

$$m_i \frac{d^2 X_i(t)}{dt^2} + \gamma \frac{dX_i(t)}{dt} = F_i \quad (3)$$

Dabei handelt es sich um eine Differentialgleichung zweiter Ordnung, wobei $X_i(t)$ die Position des Punktes zum Zeitpunkt t und m_i dessen Masse ist. Diese lässt sich wiederum in zwei verbundene Differentialgleichungen erster Ordnung aufteilen. Demnach gelten für die Geschwindigkeit $v_i(t)$ und die Beschleunigung $\frac{dv_i(t)}{dt}$ eines Punktes die folgenden Zusammenhänge:

$$\frac{dX_i(t)}{dt} = v_i(t) \quad \frac{dv_i(t)}{dt} = \frac{F_i(t) - \gamma v_i(t)}{m_i} \quad (4)$$

Mit dieser Kenntnis kann nun die Position jedes Massepunktes durch ein Integrationsverfahren in Abhängigkeit von der Zeit berechnet werden, da



(a) Erster Zeitschritt

(b) Zweiter Zeitschritt

Abbildung 4: Die ersten beiden Zeitschritte der zeitlichen Integration eines Masse-Feder-Systems unter Einwirkung einer externen Kraft F^{ext} . Die obersten drei Punkte sind dabei unbeweglich.

die initialen Positionen $X(t_0)$ und Geschwindigkeiten $v(t_0)$ der Massepunkte bekannt sind. Das einfachste Integrationsverfahren ist die explizite Euler-Integration. Dabei berechnet sich nach einem Zeitschritt h die Position eines Massepunktes $X_i(t_0 + h)$ wie folgt:

$$X_i(t_0 + h) = X_i(t_0) + hv_i(t_0) \quad (5)$$

Das heißt, die neue Position ergibt sich durch Addition des aktuellen Geschwindigkeitsvektors auf den aktuellen Positionsvektor. Die Geschwindigkeit im nächsten Zeitschritt $v_i(t_0 + h)$ berechnet sich daher folgendermaßen:

$$v_i(t_0 + h) = v_i(t_0) + h \frac{F_i(t_0) - \gamma v_i(t_0)}{m_i} \quad (6)$$

Die Geschwindigkeit im folgenden Zeitschritt berechnet sich also durch Addition der Beschleunigung in diesem Punkt auf den aktuellen Geschwindigkeitsvektor. So lassen sich die Positionen und Geschwindigkeiten für jeden Zeitschritt berechnen. Abbildung 4 zeigt die Auswirkung einer externen Kraft F^{ext} auf ein Masse-Feder-System für die ersten beiden Zeitschritte.

Im ersten Zeitschritt wirkt die externe Kraft auf den Punkt unten rechts (beispielsweise durch Verschiebung dieses Punktes). Daraus ergeben sich die

Geschwindigkeit und die neue Position für diesen Punkt nach Gleichung 7 und 6. Die anderen Positionen bleiben im ersten Schritt noch unverändert. Für sie werden allerdings mit der Gleichung 6 bereits Geschwindigkeiten für den zweiten Zeitschritt berechnet, die sich aus den nun wirkenden Kräften ergeben. Im zweiten Zeitschritt verändern sich dementsprechend dann auch die Positionen der Nachbarn des Punktes, auf den die externe Kraft ausgeübt wird.

Das explizite Euler-Verfahren ist relativ einfach zu berechnen, da in jedem Zeitschritt nur eine Ableitung für jeden Punkt berechnet werden muss. Allerdings hängt die Genauigkeit extrem von der Größe des Zeitschritts ab und wird sehr schnell instabil. Der numerische Fehler, den man bei diesem Verfahren macht, kann leicht dazu führen, dass das System divergiert. Hierbei gilt zum einen: je größer der Zeitschritt, desto größer die Ungenauigkeit. Gleichzeitig aber auch: Je kleiner der Zeitschritt, desto größer der Rechenaufwand. Es gibt verschiedene Ansätze die Stabilität des Integrationsverfahrens zu verbessern. Eine optimale Lösung gibt es allerdings nicht. Es muss immer ein Kompromiss zwischen Präzision und Rechenzeit gefunden werden, je nach dem, was für die jeweilige Anwendung wichtiger ist.

Ein Ansatz ist, anstelle des expliziten das implizite Euler-Verfahren zu verwenden. Dabei berechnet man die Positionen quasi rückwirkend aus den Geschwindigkeiten des nächsten Zeitschritts:

$$X_i(t_0 + h) = X_i(t_0) + hv_i(t_0 + h) \quad (7)$$

Dadurch wird die Stabilität besonders bei großen Federkonstanten verbessert. Eine weitere Möglichkeit ist die Verwendung der Runge-Kutta-Methode. Bei der Runge-Kutta-Methode 2. Ordnung (auch als Mittelpunkt-Methode bekannt) berechnet man zunächst die Geschwindigkeit und die Beschleunigung zum Zeitpunkt t . Daraus werden dann die Position, Geschwindigkeit und die Beschleunigung zum Zeitpunkt $t + h/2$ (dem Mittelpunkt) berechnet. Davon ausgehend werden dann wiederum die Position und die Geschwindigkeit zum Zeitpunkt $t + h$ mit der Geschwindigkeit und der Beschleunigung zum Zeitpunkt $t + h/2$ berechnet. Dies führt zu einer Erhöhung der Genauigkeit der Integration, da der eigentliche Zeitschritt in zwei kleinere aufgeteilt wird. Gleichzeitig steigt dadurch natürlich auch der Berechnungsaufwand.

2.3 Die Struktur im 2D-Fall

Wie die Struktur eines Masse-Feder-Systems aufgebaut ist, hängt zunächst von der Art des Objekts ab. Für Deformationsmodelle besteht ein genereller

Unterschied zwischen räumlichen, dreidimensionalen und planaren, zweidimensionalen Objekten. Als Planar werden dabei Objekte angesehen, die eine so geringe Dicke aufweisen, dass diese für die Simulation physikalischen Verformung eine zu vernachlässigende Rolle spielt. Dazu gehören zum Beispiel Kleidungsstücke, Tischdecken, Papier oder Luftballons. Also alle Objekte, die sich in einer Ebene ausbreiten lassen. Im einfachsten Fall handelt es sich bei zweidimensionalen Objekten um ein rechteckiges Gitter, bestehend aus $m \times n$ Massepunkten, die mit ihren Nachbarpunkten über Federn verbunden sind (Abbildung 1(a)).

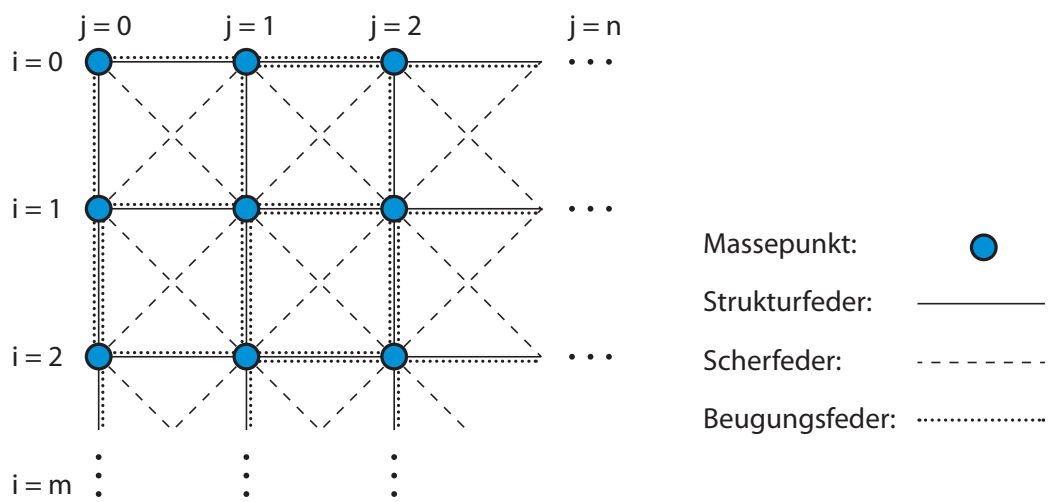


Abbildung 5: Struktur eines Masse-Feder-Systems für planare Objekte.

Abbildung 5 zeigt, dass dabei zwei Massepunkte auf drei verschiedene Arten mit einander verbunden sein können [Pro95]:

- Federn, die Massen an $[i, j]$ mit $[i + 1, j]$ und $[i, j]$ mit $[i, j + 1]$ verbinden, werden Strukturfedern genannt.
- Federn, die Massen an $[i, j]$ mit $[i + 1, j + 1]$ und $[i + 1, j]$ mit $[i, j + 1]$ verbinden, werden Scherfedern genannt.
- Federn, die Massen an $[i, j]$ mit $[i + 2, j]$ und $[i, j]$ mit $[i, j + 2]$ verbinden, werden Beugungsfedern genannt.

Hierbei ist noch zu beachten, ob das Objekt räumlich verformt werden soll oder nicht (beispielsweise bei zweidimensionalen Animationen). Beugungsfedern werden nur beansprucht, wenn das Objekt räumlich gebeugt wird (zum Beispiel, wenn eine Blatt Papier gerollt wird). Daher kann auf diese Federn je nach Anwendungsfall verzichtet werden.

2.4 Die Struktur im 3D-Fall

Dreidimensionale, räumliche Strukturen entstehen bei Objekten, die aufgrund ihrer Dicke auch im Inneren physikalische Eigenschaften aufweisen, die die Deformation beeinträchtigen würden (Abbildung 1(b)). Dabei unterscheidet man hauptsächlich zwischen Tetraeder- und Hexaedernetzen, wie sie in Abbildung 6 zu sehen sind [BpC00].

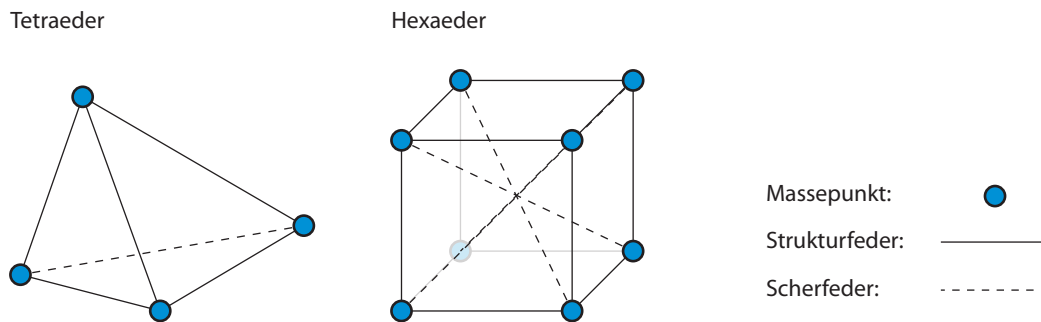


Abbildung 6: Volumenelemente aus Tetraeder- und Hexaedernetzen.

Viele 3D-Modellieranwendungen erzeugen ein Dreiecksnetz der Oberfläche der Objekte. Um hieraus eine räumliche Struktur für Masse-Feder-Systeme zu generieren, bieten sich daher Tetraedernetze an. Die Struktur kommt hierbei durch ein Schichtenmodell zustande. Dazu wird eine zweite Schicht der Oberfläche durch eine kleiner skalierte Kopie im Inneren des Objekts erstellt, wie Abbildung 7 veranschaulicht.

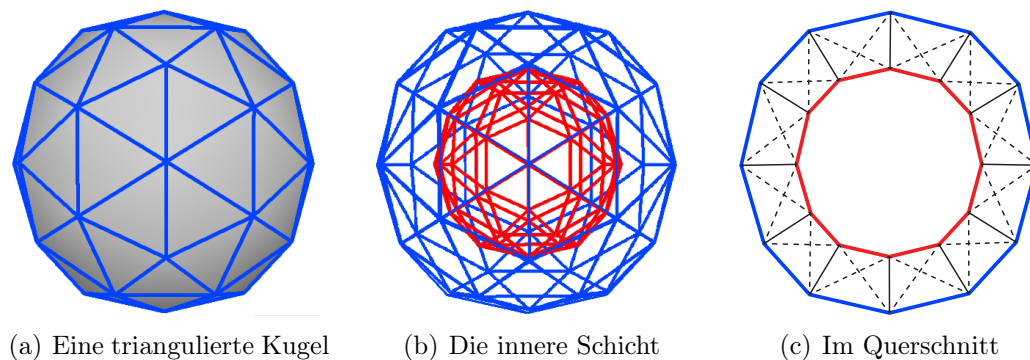


Abbildung 7: Schichtenmodell für das Dreiecksnetz einer Kugel. Von links nach rechts: Das Polygonnetz der Kugel. Die Kugel (blau) und die innere Schicht (rot) in der Gitternetzdarstellung. Die Kugel im Querschnitt mit den Verbindungen zwischen den Schichten.

Anschließend können die Schichten über eine Tetraedernetz miteinander verbunden werden. Die Massepunkte liegen dabei immer auf den Knoten und die Federn entlang der Kanten des Tetraedernetzes. Dadurch wird das Objekt um zusätzliche Massepunkte und Federn ergänzt, die die Struktur im Inneren annähern. Die Anzahl der Schichten richtet sich nach dem Anwendungsfall. Hierbei gilt aber natürlich: je mehr Massepunkte und Federn, desto höher der Berechnungsaufwand.

Der Gebrauch von Hexaedernetzen ist nicht so weit verbreitet, da die Geometrie, die durch sie definiert werden kann, begrenzt ist. Dennoch können sie nützlich sein, wenn man volumetrische Daten in Form eines Voxelgitters animieren möchte [ChZK98].

3 Verbesserungen und Erweiterungen des Systems

Die Stabilität von Masse-Feder-Systemen hängt direkt von der Größe des gewählten Zeitschritts für die Integration ab. Je kleiner dieser Zeitschritt gewählt wird, desto stabiler und präziser ist das System. Gleichzeitig muss auch dementsprechend häufiger der Systemzustand berechnet werden. Betrachtet man allerdings einen Ball im freien Fall, so benötigt diese Animation keinen besonders kleinen Zeitschritt. Jede Kollision mit der Umgebung wiederum würde einen deutlich kleineren Zeitschritt für die Kräfteintegration erfordern, damit sich das System stabil verformen kann.

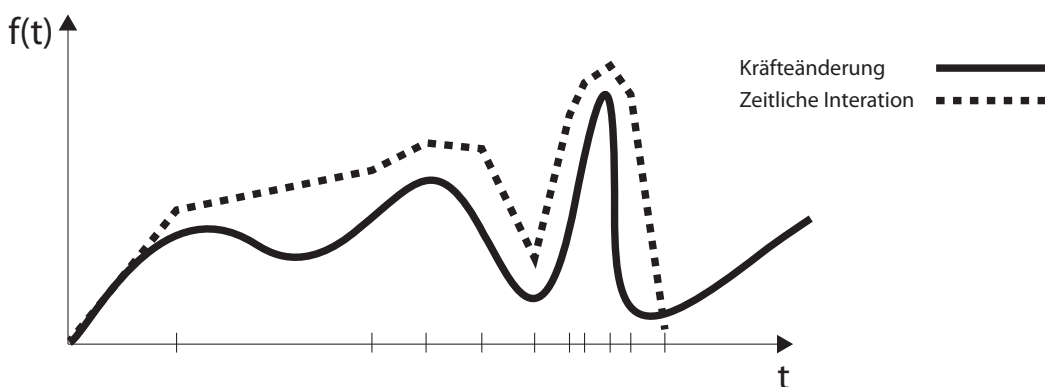


Abbildung 8: Zeitliche Integration mit adaptivem Zeitschritt.

Daher ist ein erster Ansatz Masse-Feder-Systeme zu optimieren, die Einführung eines adaptiven Zeitschritts. Wie in Abbildung 8 zu sehen ist, wird die

ser kleiner, sobald es zu großen Kraftänderungen kommt, und größer, solange die Änderungen eher gering sind. Dadurch entsteht allerdings das Problem, dass zusätzliche Berechnungen anfallen, um die Kräfteänderungen frühzeitig erkennen zu können.

In den folgenden Abschnitten werden weitere Grenzen der Masse-Feder-Systeme aufgezeigt und Lösungsansätze vorgestellt, um die Probleme, die dadurch auftreten, zu beheben.

3.1 Adaptive räumliche Verfeinerung

Bei Masse-Feder-Systemen mit einer eher groben Verteilung der Massepunkte treten die meisten Ungenauigkeiten bei großen Beugungen der Oberfläche auf. Da die Verformungen ausschließlich entlang der Federn wirken, werden diese häufig in Form von Kanten sichtbar (Abbildung 9). Allerdings bedeutet eine allgemeine Verfeinerung der Massenverteilung einen deutlich größeren Berechnungsaufwand. Eine Möglichkeit dieses Problem zu umgehen ist, das Masse-Feder-System an Stellen, an denen es zu großen Beugungen kommt, adaptiv zu verfeinern [HPH96].

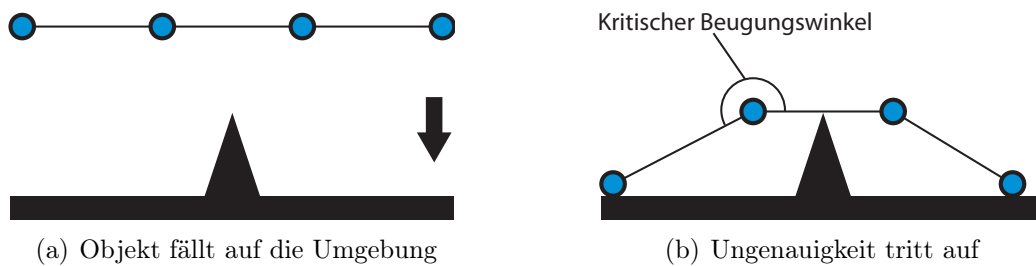


Abbildung 9: Veranschaulichung der Ungenauigkeiten, die bei großer Beugung grober Masse-Feder-Systeme auftreten können.

Als Bedingung für die Verfeinerung dient der Beugungswinkel an einem Massepunkt zu seinen Nachbarn, wie Abbildung 9(b) verdeutlicht. Der Schwellwert für diesen Winkel kann beispielsweise durch den Animator festgelegt werden.

Für die adaptive Verfeinerung wird das Masse-Feder-System in einer flachen Datenstruktur gespeichert, die bereits die Massepunkte für alle Verfeinerungsstufen enthält, wie Abbildung 10 zeigt.

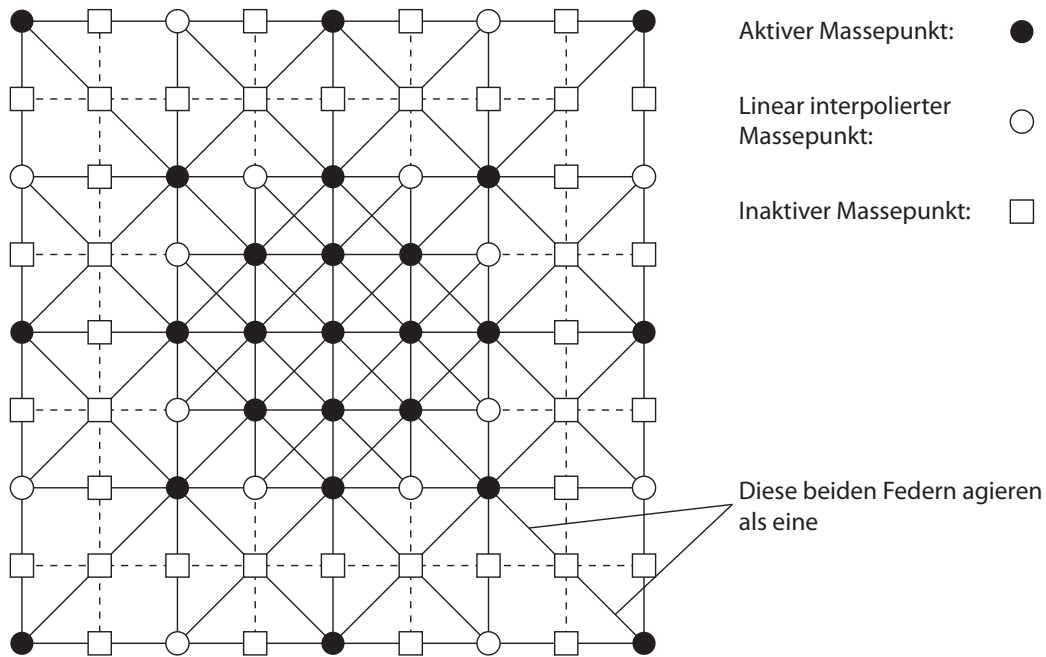


Abbildung 10: Ein verfeinertes Massengitter für zwei Stufen der Verfeinerung. Am mittleren Punkt wurde das System zweifach verfeinert.

Auf jeder Detailstufe werden durch lineare Interpolation neue Massepunkte in der Mitte zwischen zwei benachbarten Punkten der vorherigen Stufe erzeugt. Kommt es bei der Simulation an einem Massepunkt zu einer Überschreitung des kritischen Beugungswinkels, so werden die Massepunkte aus der nächst höheren Verfeinerungsstufe um ihn herum aktiviert. Die Verbindungen dieser Punkte werden dabei jeweils zu den linear interpolierten Nachbarpunkten auf derselben Stufe geknüpft. Bei der Verfeinerung ist es wichtig, dass die hinzukommenden Massepunkte, das Verhalten des Objekts nicht verändern. Daher wird die Federkonstante auf jeder Stufe verdoppelt und der Zeitschritt halbiert. Sei n also die Verfeinerungsstufe und $h(0)$ der Zeitschritt bei $n = 0$, so berechnet sich der neue Zeitschritt in Abhängigkeit der Verfeinerung als $h(n) = \frac{h}{2^n}$.

3.2 Begrenzung der Elastizität

Besonders bei Kleidungsanimationen mit Masse-Feder-Systemen kann es zu unrealistischen Ergebnissen durch sehr große Dehnungen kommen. Aufgrund der Natur dieser Systeme erscheint Kleidung oft als „zu elastisch“ oder gummiartig, wie Abbildung 11 veranschaulicht.

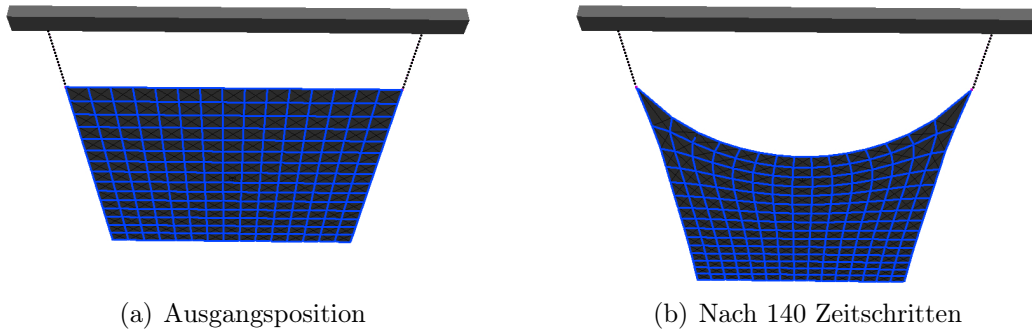


Abbildung 11: Simulation eines Stofftuches unter Einwirkung der Schwerkraft, das an zwei Eckpunkten aufgehängt wurde.

Der erste Ansatz, um diesen Effekt zu vermeiden, wäre einer Erhöhung der Federkonstanten. Numerisch lässt sich allerdings zeigen, dass eine Erhöhung der Federkonstante gleichzeitig zu einer Verringerung des Zeitschritts führen muss, da das System andernfalls divergiert. Wie bereits erwähnt, erhöht eine Verringerung des Zeitschritts aber gleichzeitig den Rechenaufwand und ist daher zu vermeiden.

Eine bessere Lösung für dieses Problem ist die Einführung eines Schwellwertes für die maximale Dehnung der Federn [Pro95]. In jedem Integrationsschritt wird die Dehnung für jede Feder berechnet. Liegt diese oberhalb eines zuvor festgelegten Schwellwertes s , würden die beiden Endpunkte nach diesem Zeitschritt zu weit auseinander liegen, und es muss zu einer Korrektur kommen. Das heißt, wenn $s = 0,1$ gewählt wird, dürfen die Federn nicht um mehr als 10% ihrer Ausgangslänge gedehnt werden.

Bei dem nun folgenden Korrekturschritt geht man davon aus, dass die berechnete Richtung der Feder korrekt ist, jedoch nicht der Abstand der beiden Endpunkte. Die Endpunkte müssen nun also so positioniert werden, dass die Dehnung der Feder nicht größer als s ist, während die Richtung der Feder unverändert bleibt. Dabei wird unterschieden, ob sich beide Endpunkte der Feder frei bewegen oder einer unbeweglich befestigt ist. Wenn beide Endpunkte frei beweglich sind, werden sie gleichmäßig in Richtung ihres Mittelpunktes verschoben. Ist einer der beiden Punkte unbeweglich, so wird der bewegliche Punkt in Richtung des anderen verschoben.

Abbildung 12 zeigt die Auswirkungen des Schwellwerts auf das Deformationsverhalten. Dabei wurde der Schwellwert im ersten Fall nur auf die Strukturfedern (Abbildung 12(a)) und im zweiten Fall zusätzlich auf die Scherfedern (Abbildung 12(b)) angewendet.

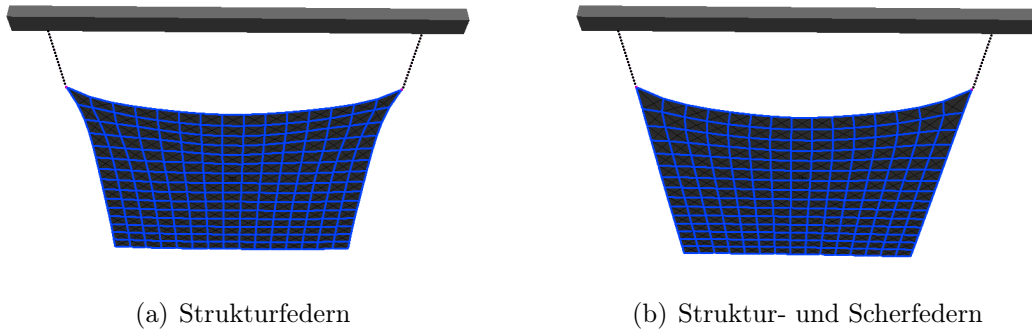


Abbildung 12: Anwendung des Schwellwerts auf die Simulation. Im ersten Fall wurde die Begrenzung nur auf die Strukturfedern angewendet und im zweiten Fall zusätzlich auf die Scherfedern.

Wie zu erkennen ist, verhält sich das Stofftuch jetzt deutlich realistischer in Anbetracht seines simulierten Materials.

4 Zusammenfassung

Zunächst wurde das Masse-Feder-System als Deformationsmodell eingeordnet, das zu den Lagrangeschen, Netz-basierten Verfahren gehört. Es wurde gezeigt, wie die Kräfte innerhalb des Systems berechnet und in Abhängigkeit von der Zeit integriert werden. Wie die Struktur aus Massepunkten und Federn aufgebaut wird, wurde für den zweidimensionalen und den dreidimensionalen Fall im Detail beschrieben. Des Weiteren wurden Verfahren aufgezeigt, die Schwächen des ursprünglichen Masse-Feder-Systems durch Modifizierungen beheben. Es wurde erläutert, dass Masse-Feder-Systeme aufgrund ihrer geringen Komplexität eine schnelle Lösung für physikalische Deformationssimulationen bieten, bei denen es eher auf geringe Laufzeiten und visuell überzeugende Resultate als auf Korrektheit und Präzision ankommt. Es wurde allerdings auch gesagt, dass ihre Stabilität direkt von der Größe des Zeitschritts und der Dichteverteilung der Massepunkte abhängt.

Die neusten Verfahren auf diesem Gebiet unterscheiden sich teilweise grundsätzlich von Masse-Feder-Systemen. Aufgrund der verbesserten Rechenleistung (zum Beispiel durch die Möglichkeiten der GPGPU-Programmierung) gehen heutige Echtzeitverfahren mehr in Richtung Kontinuumsmechanik, und sind daher physikalisch wesentlich genauer als Masse-Feder-Systeme. Hierbei werden beispielsweise Ansätze von Partikelsystemen und Netz-basierten Verfahren miteinander verbunden [GBFP11]. Auch bieten diese Modelle die

Möglichkeit Objekte zu simulieren, die aus verschiedenen Materialien zusammengesetzt sind, wie beispielsweise ein T-Bone Steak (aus Knochen und Fleisch) [FGBP11]. Auch gibt es Verfahren, die dazu verwendet werden können, um künstlerische Effekte bei der Deformation zu erzielen. Das heißt, Objekte verformen sich auf zuvor festgelegte Arten und Weisen, die keinen physikalischen Gesetzen mehr folgen müssen [MTGG11].

Allerdings erfordern diese Verfahren auch immer mehr Grundlagenwissen in Bereichen der Physik, Differenzialrechnung und der Lösung von Gleichungssystemen und sind im Vergleich zu den hier vorgestellten Masse-Feder-Systemen wesentlich aufwendiger zu implementieren.

Literatur

- [BpC00] BOURGUIGNON, DAVID und MARIE PAULE CANI: *Controlling Anisotropy in Mass-Spring Systems*. In: *In Eurographics, Eurographics Assoc*, Seiten 113–123. Springer-Verlag, 2000.
- [Car04] CARLSON, MARK THOMAS: *Rigid, Melting, and Flowing Fluid*, 2004.
- [ChZK98] CHEN, YAN, QING HONG ZHU und ARIE KAUFMAN: *Physically-based animation of volumetric objects*. In: *In Proceedings of IEEE Computer Animation 1998*, Seiten 154–160, 1998.
- [FGBP11] FAURE, FRANÇOIS, BENJAMIN GILLES, GUILLAUME BOUSQUET und DINESH K. PAI: *Sparse meshless models of complex deformable solids*. *ACM Trans. Graph.*, 30:73:1–73:10, August 2011.
- [FM96] FOSTER, NICK und DIMITRI METAXAS: *Practical animation of liquids*. In: *Graphical Models and Image Processing*, Seiten 23–30, 1996.
- [GBFP11] GILLES, BENJAMIN, GUILLAUME BOUSQUET, FRANCOIS FAURE und DINESH K. PAI: *Frame-based elastic models*. *ACM Trans. Graph.*, 30:15:1–15:12, April 2011.
- [HPH96] HUTCHINSON, DAVE, MARTIN PRESTON und TERRY HEWITT: *Adaptive Refinement for Mass/Spring Simulations*. In: *In 7th Eurographics Workshop on Animation and Simulation*, Seiten 31–45. Springer-Verlag, 1996.
- [Hun05] HUNTER, PETER: *FEM/BEM Notes*. Technischer Bericht, 2005.
- [LL02] LI, SHAOFAN und WING KAM LIU: *Meshfree and particle methods and their applications*. *Applied Mechanics Review*, 55:1–34, 2002.
- [MTGG11] MARTIN, SEBASTIAN, BERNHARD THOMASZEWSKI, EITAN GRINSPUN und MARKUS GROSS: *Example-based elastic materials*. *ACM Trans. Graph.*, 30:72:1–72:8, August 2011.
- [NMK⁺06] NEALEN, ANDREW, MATTHIAS MUELLER, RICHARD KEISER, EDDY BOXERMAN und MARK CARLSON: *Physically Based Deformable Models in Computer Graphics*. *Computer Graphics Forum*, 25(4):809–836, Dezember 2006.

- [Pro95] PROVOT, XAVIER: *Deformation constraints in a mass-spring model to describe rigid cloth behavior*. In: *IN GRAPHICS INTERFACE*, Seiten 147–154, 1995.
- [Ree83] REEVES, W. T.: *Particle Systems - a Technique for Modeling a Class of Fuzzy Objects*. *ACM Trans. Graph.*, 2(2):91–108, April 1983.
- [TBHF03] TERAN, J., S. BLEMKER, V. NG THOW HING und R. FEDKIW: *Finite Volume Methods for the Simulation of Skeletal Muscle*, 2003.
- [THMG04] TESCHNER, MATTHIAS, BRUNO HEIDELBERGER, MATTHIAS MULLER und MARKUS GROSS: *A Versatile and Robust Model for Geometrically Complex Deformable Solids*. In: *Proceedings of the Computer Graphics International*, Seiten 312–319, Washington, DC, USA, 2004. IEEE Computer Society.
- [Ton98] TONNESEN, DAVID LOVE: *Dynamically coupled particle systems for geometric modeling, reconstruction, and animation*. Doktorarbeit, Toronto, Ont., Canada, Canada, 1998. AAINQ41520.
- [TPBF87] TERZOPOULOST, DEMETRI, JOHN PLATT, ALAN BARR und KURT FLEISCHERT: *Elastically Deformable Models*. *Computer Graphics*, 21:205–214, 1987.
- [ZT00] ZIENKIEWICZ, O. C. und R. L. TAYLOR: *Finite Element Method: Volume 1, Fifth Edition (Finite Element Method Ser)*. Butterworth-Heinemann, 5 Auflage, September 2000.