

Vortrag über das Paper:
"Large Steps in Cloth Simulation"

von

David Baraff, Andrew Witkin



gehalten von

Martin Hauser

im Rahmen des Seminars

"Aktuelle Themen der graphischen Datenverarbeitung"

der Abteilung für Informatik der ETH Zürich

Sommersemester 1999

INHALTSVERZEICHNIS

1. ÜBERSICHT	1
2. PHYSIKALISCHES MODELL	1
3. MATHEMATIK	2
3.1. EXPLIZITER ANSATZ	2
3.2. IMPLIZITER ANSATZ	2
4. KRÄFTE/CONDITIONS	3
4.1. STRECKUNGSKRÄFTE	4
4.2. SCHERUNGSKRÄFTE	4
4.3. BIEGUNGSKRÄFTE	5
4.4. WEITERE KRÄFTE	5
4.5. DÄMPFUNG	5
5. CONSTRAINTS	6
5.1. VERSCHIEDENE METHODEN	6
5.1.1. Reduced Coordinates	6
5.1.2. Penalty Method	6
5.1.3. Lagrange Multipliers	6
5.2. MASSENMODIFIKATION	6
6. KOLLISION	7
7. LÖSUNG DES GLEICHUNGSSYSTEMS	8
7.1. VORBEMERKUNGEN	8
7.2. PSEUDO-CODE	9
8. SCHRITTWEITE ANPASSEN	9
9. ERGEBNISSE	9

1. Übersicht

Seit mehr als einem Jahrzehnt stellt die physikalisch-basierte Stoffanimation eines Herausforderung für die Computer-Graphik dar. Im Allgemeinen verwendeten bisherige Systeme zur Stoffsimulation zur Lösung der zu Grunde liegenden Differentialgleichung (→ 2. Physikalisches Modell) explizite Methoden wie Euler oder Runge-Kutta. Der Flaschenhals dieser Methoden sind die kleinen Zeitschritte, die zur Vermeidung von numerischen Instabilitäten verwendet werden müssen. Das Paper von David Baraff und Andrew Witkin beschreibt nun ein Stoffsimulationssystem, das numerische Stabilität auch bei grossen Schrittweiten bietet. Der Titel "Large Steps in Cloth Simulation" darf deshalb sowohl wörtlich als auch im übertragenen Sinn verstanden werden: Die grossen Schrittweiten bei der Lösung der Differentialgleichung sind ein grosser Schritt in der Stoffsimulation.

2. Physikalisches Modell

In physikalisch-basierter Stoffsimulation wird Stoff durch ein Masse-Federn-System modelliert. Dabei wird die Position der einzelnen Masse-Knoten als zeitabhängige, partielle Differentialgleichung formuliert:

$$\ddot{\mathbf{x}} = \mathbf{M}^{-1} \left(-\frac{\partial E}{\partial \mathbf{x}} + \mathbf{F} \right)$$

Gleichung 1

Dabei beschreiben der Vektor \mathbf{x} den geometrischen Zustand und die Diagonalmatrix \mathbf{M} die Masseverteilung des Stoffes, E – eine skalare Funktion von \mathbf{x} – beinhaltet die innere Energie des Stoffes, und \mathbf{F} (als Funktion von \mathbf{x} und $\dot{\mathbf{x}}$) beschreibt zusätzliche Kräfte (Wind, Kontaktkräfte, innere Dämpfung, usw.), die auf den Stoff einwirken.

Fasst man alle Kräfte zusammen, lässt sich Gleichung 1 vereinfacht schreiben als

$$\ddot{\mathbf{x}} = \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$$

Gleichung 2

mit $\mathbf{M} = \text{diag}(m_1, m_1, m_1, m_2, m_2, m_2, \dots, m_n, m_n, m_n)$ (n = Anzahl der Knoten). \mathbf{x} und $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$ sind Vektoren mit n Komponenten, jede Komponente ist wiederum ein 3dimensionaler Vektor.

Jedem Masseelement wird eine feste Koordinate (u, v) in der Ebene zugewiesen. Diese Koordinaten beschreiben die Position im Ruhezustand des Stoffes und können z.B. für Texturen verwendet werden (→ Abbildung 1).

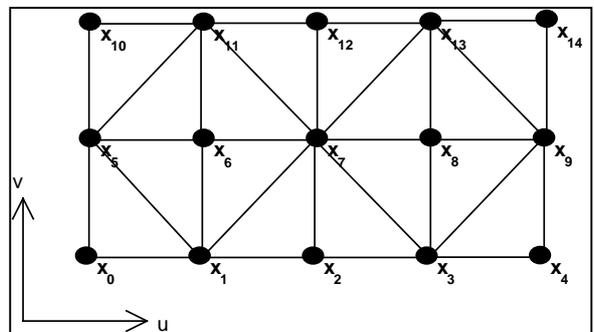


Abbildung 1

Stoff lässt sich leicht scheren und biegen, kann jedoch nur schwer gestreckt werden (→ Abbildung 2). Dies führt zu einer "steifen" zugrundeliegenden Differentialgleichung. Explizite Methoden sind schlecht geeignet zur Lösung von steifen Gleichungen, da diese viele kleine Zeitschritte benötigen, um numerisch stabil in der Zeit fortzuschreiten. Der durch diese kleinen Zeitschritte notwendige Rechenaufwand beschränkt in der Praxis die Anzahl der Masseknoten im Stoffmodell. Stoffanimation erfordert zur realistischen Darstellung der Faltung jedoch weit höher Gitterauflösungen.

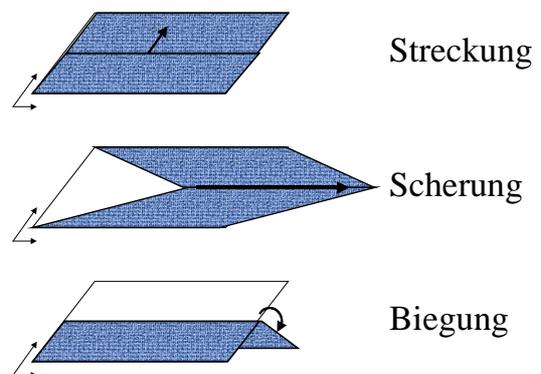


Abbildung 2

Die Autoren des Papers verwenden deshalb ein implizites Verfahren, dass bei steifen Gleichungssystemen trotzdem grosse Schrittweiten zulässt.

3. Mathematik

Ausgehend von der bekannten Position $\mathbf{x}(t_0)$ und der bekannten Geschwindigkeit $\dot{\mathbf{x}}(t_0)$ zur Zeit t_0 sollen die neue Position $\mathbf{x}(t_0 + h)$ und die neue Geschwindigkeit $\dot{\mathbf{x}}(t_0 + h)$ zur Zeit $t_0 + h$ bestimmt werden.

Zur Lösung mit einer impliziten Methode muss Gleichung 2 zuerst in eine Differentialgleichung erster Ordnung umgewandelt werden:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}, \mathbf{v}) \end{pmatrix}$$

Gleichung 3

Zur Vereinfachung der Schreibweise definieren wir $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\mathbf{v}_0 = \mathbf{v}(t_0)$, $\Delta \mathbf{x} = \mathbf{x}(t_0 + h) - \mathbf{x}(t_0)$, $\Delta \mathbf{v} = \mathbf{v}(t_0 + h) - \mathbf{v}(t_0)$ und $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0, \mathbf{v}_0)$

3.1. EXPLIZITER ANSATZ

Beim expliziten (Vorwärts-) Lösungsverfahren werden $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ durch folgendes Gleichungssystem beschreiben:

$$\begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}_0, \mathbf{v}_0) \end{pmatrix}$$

Gleichung 4

Hier genügt zur Berechnung von $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ die einmalige Auswertung von \mathbf{f} . Die Schrittweite h ist dabei aufgrund der in 2. Physikalisches Modell beschriebenen Steifheit der Differentialgleichung klein zu halten.

3.2. IMPLIZITER ANSATZ

Beim impliziten (Rückwärts-) Lösungsverfahren werden $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ durch folgendes Gleichungssystem beschrieben:

$$\begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 + \Delta \mathbf{v} \\ \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) \end{pmatrix}$$

Gleichung 5

Im Unterschied zum expliziten Ansatz hängt $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ nicht nur von Bedingungen zur Zeit t_0 ab, sondern $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ werden bestimmt durch Bedingungen am Schluss des Zeitschrittes. Dieses Gleichungssystem muss nun nach $\Delta \mathbf{x}$ und $\Delta \mathbf{v}$ aufgelöst werden. Gleichung 5 ist nicht-linear. Anstelle einer exakten Lösung wird eine Taylor-Reihe erster Ordnung verwendet:

$$\mathbf{f}(\mathbf{x}_0 + \Delta \mathbf{x}, \mathbf{v}_0 + \Delta \mathbf{v}) = \mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \Delta \mathbf{v}$$

Die untere Zeile von Gleichung 5 wird durch diese Taylor-Entwicklung und Substitution von $\Delta \mathbf{x} = h(\mathbf{v}_0 + \Delta \mathbf{v})$ zu:

$$\Delta \mathbf{v} = \mathbf{M}^{-1} \left(\mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} h(\mathbf{v}_0 + \Delta \mathbf{v}) + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \Delta \mathbf{v} \right)$$

Durch Umstellen erhält man

$$\left(\mathbf{I} - h\mathbf{M}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2 \mathbf{M}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \Delta \mathbf{v} = h\mathbf{M}^{-1} \left(\mathbf{f}_0 + h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v}_0 \right)$$

Gleichung 6

mit \mathbf{I} der Einheitsmatrix

4. Kräfte/Conditions

Wie aus Gleichung 1 ersichtlich, wird das Verhalten von Stoff durch eine Energiefunktion $E(\mathbf{x})$ beschrieben. Die davon herrührende Kraft ist $\mathbf{f} = -\frac{\partial E}{\partial \mathbf{x}}$. In Gleichung 6 wird sowohl \mathbf{f} als auch $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ benötigt. Anstelle der Beschreibung von $E(\mathbf{x})$ als einer allgemeinen Funktion, die alle Aspekte des inneren Verhaltens des Stoffes umfasst, ist in Hinblick auf den Berechnungsaufwand eine Aufteilung in mehrere Energiefunktionen vorzuziehen, d.h. man schreibt $E(\mathbf{x}) = \sum_{\alpha} E_{\alpha}(\mathbf{x})$, wobei die einzelnen $E_{\alpha}(\mathbf{x})$ von möglichst wenigen Partikeln abhängig sein sollen, und somit auch unabhängig von möglichst vielen Elementen von \mathbf{X} .

Die Autoren des Papers sind jedoch der Ansicht, dass auch diese Aufteilung der Energiefunktion kein geeigneter Ansatz ist, da geeignete Dämpfung nicht aus der Energiefunktion hergeleitet werden kann. Stattdessen definieren sie das interne Verhalten des Stoffes durch eine Vektorcondition $\mathbf{C}(\mathbf{x})$, die Null sein soll. Die dazugehörige Energie wird definiert als $E_C(\mathbf{x}) = \frac{k}{2} \mathbf{C}(\mathbf{x})^T \mathbf{C}(\mathbf{x})$, wobei k eine frei gewählte Steifheitskonstante ist. Unter der Voraussetzung, dass es sich bei \mathbf{C} um eine dünnbesetzte Matrix handelt, führt \mathbf{C} zu einem dünnbesetzten Kraftvektor \mathbf{f} . Wie in 2. Physikalisches Modell beschrieben, ist jede Komponente \mathbf{f}_i von \mathbf{f} ein 3dimensionaler Vektor. Es gilt:

$$\mathbf{f}_i = \begin{cases} -\frac{\partial E_C}{\partial \mathbf{x}_i} = -k \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \mathbf{C}(\mathbf{x}) & \mathbf{C} \text{ ist abhängig von Partikel } i \\ 0 & \text{sonst} \end{cases}$$

Gleichung 7

Auch die Ableitung $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ von \mathbf{f} ist dünnbesetzt. \mathbf{K} ist eine $n \times n$ -Matrix, jede Komponente ist wiederum eine 3×3 -Matrix. Aus Gleichung 7 gilt:

$$\mathbf{K}_{ij} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j} = -k \left(\frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \frac{\partial \mathbf{C}(\mathbf{x})^T}{\partial \mathbf{x}_j} + \frac{\partial^2 \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \mathbf{C}(\mathbf{x}) \right)$$

Gleichung 8

\mathbf{K} ist symmetrisch, da $\mathbf{K}_{ij} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j} = \frac{\partial^2 E}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_i} = \mathbf{K}_{ji}$

Innere Kräfte können nun leicht beschrieben werden, indem die entsprechende Condition \mathbf{C} verwendet wird. Kräfte und deren Ableitungen werden aus Gleichung 7 und Gleichung 8 hergeleitet.

4.1. STRECKUNGSKRÄFTE

Für die Berechnung der Streckung wird eine Funktion $\mathbf{w}(u, v)$ definiert, die die Flächenkoordinaten (u_i, v_i) des Partikels i in die Raumkoordinaten \mathbf{x}_i abbildet. Streckung kann nun durch Betrachtung der Ableitungen $\mathbf{w}_u = \frac{\partial \mathbf{w}}{\partial u}$ und $\mathbf{w}_v = \frac{\partial \mathbf{w}}{\partial v}$ bestimmt werden. Der Betrag von \mathbf{w}_u bzw. \mathbf{w}_v beschreibt die Streckung in u - bzw. v -Richtung. In ungestrecktem Zustand gilt $\|\mathbf{w}_u\| = 1$ bzw. $\|\mathbf{w}_v\| = 1$.

Die Streckung wird wie folgt auf ein Dreieck des Meshes mit Eckknoten i, j, k angewendet:

Man definiert $\Delta \mathbf{x}_1 = \mathbf{x}_j - \mathbf{x}_i$, $\Delta \mathbf{x}_2 = \mathbf{x}_k - \mathbf{x}_i$, $\Delta u_1 = u_j - u_i$, $\Delta u_2 = u_k - u_i$, $\Delta v_1 = v_j - v_i$ und $\Delta v_2 = v_k - v_i$ und approximiert $\mathbf{w}(u, v)$ als lineare Funktion über jedem Dreieck (\mathbf{w}_u bzw. \mathbf{w}_v sind konstant über jedem Dreieck). Das führt zu $\Delta \mathbf{x}_1 = \mathbf{w}_u \Delta u_1 + \mathbf{w}_v \Delta v_1$ und $\Delta \mathbf{x}_2 = \mathbf{w}_u \Delta u_2 + \mathbf{w}_v \Delta v_2$. Aufgelöst nach \mathbf{w}_u und \mathbf{w}_v ergibt dies:

$$\begin{pmatrix} \mathbf{w}_u & \mathbf{w}_v \end{pmatrix} = \begin{pmatrix} \Delta \mathbf{x}_1 & \Delta \mathbf{x}_2 \end{pmatrix} \begin{pmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{pmatrix}^{-1}$$

Gleichung 9

$\Delta \mathbf{x}_{1i}$ und $\Delta \mathbf{x}_2$ variieren während der Simulation, nicht jedoch die Matrix, so dass \mathbf{w}_u und \mathbf{w}_v als Funktionen von \mathbf{X} geschrieben werden können, die nur von \mathbf{X}_i , \mathbf{X}_j und \mathbf{X}_k abhängen, so dass sich als Condition ergibt:

$$\mathbf{C}(\mathbf{x}) = a \begin{pmatrix} \|\mathbf{w}_u(\mathbf{x})\| - b_u \\ \|\mathbf{w}_v(\mathbf{x})\| - b_v \end{pmatrix}$$

Gleichung 10

dabei ist a die Dreiecksfläche in uv -Koordinaten. b_u und b_v werden im allgemeinen auf 1 gesetzt. Sie können zur Modellierung von speziellen Stoffeigenschaften verwendet werden (Erhöhung von b_u in einem Ärmel \rightarrow Falten quer zur u -Richtung, Verringerung von b_v am Ärmelabschluss \rightarrow steifen Abschluss) (\rightarrow Abbildung 3).

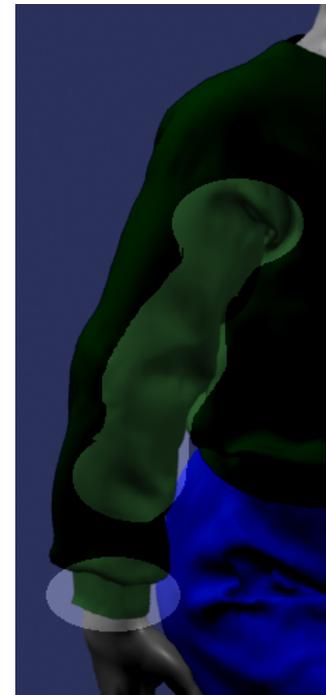


Abbildung 3

4.2. SCHERUNGSKRÄFTE

Der Grad der Scherung eines Dreieckes kann mit dem inneren Produkt $\mathbf{w}_u^T \mathbf{w}_v$ gemessen werden. Im Ruhezustand ist dieses Produkt gleich null. Durch die Terme für die Streckung sind \mathbf{w}_u und \mathbf{w}_v beschränkt, so dass auf einen Normierung verzichtet werden kann. Für kleine Winkel $\mathbf{w}_u^T \mathbf{w}_v$ ist annähernd für den Scherungswinkel. Die Condition für Scherung ist

$$\mathbf{C}(\mathbf{x}) = a \mathbf{w}_u(\mathbf{x})^T \mathbf{w}_v(\mathbf{x})$$

Gleichung 11

4.3. BIEGUNGSKRÄFTE

Biegung wird zwischen Paaren von aneinandergrenzenden Dreiecken gemessen. Die Bedingung für die Biegung wird also von vier Punkten abhängen. Wenn \mathbf{n}_1 und \mathbf{n}_2 die normierten Normalen auf den zwei Dreiecken und \mathbf{e} ein normierter Vektor parallel zur gemeinsamen Kante der beiden Dreiecke sind, so ist der Winkel θ zwischen den Flächen beschrieben durch $\sin(\theta) = (\mathbf{n}_1 \times \mathbf{n}_2) \cdot \mathbf{e}$ und $\cos(\theta) = \mathbf{n}_1 \cdot \mathbf{n}_2$. Die Condition für Biegung ist nun

$$C(\mathbf{x}) = \theta$$

Gleichung 12

4.4. WEITERE KRÄFTE

Beispiele für zusätzliche Kräfte, die aufgrund dieser Technik leicht hinzugefügt werden können, sind Schwerkraft, Luftwiderstand oder "Maus"-Kräfte durch Benutzerinteraktion.

4.5. DÄMPFUNG

Wie zu Beginn dieses Kapitels angeführt, kann die Dämpfung nur schlecht aus der Energiefunktion hergeleitet werden, weshalb die Autoren auch die verschiedenen, vorgängig beschriebenen Conditions verwenden. Wie die von der Energie herführenden Kraft \mathbf{f} soll auch die Dämpfung nur in Richtung von $\frac{\partial C(\mathbf{x})}{\partial \mathbf{x}}$ wirken. Zudem soll die Dämpfung von der Geschwindigkeitskomponente in dieser Richtung abhängen, also von $\left(\frac{\partial C(\mathbf{x})}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \dot{C}(\mathbf{x})$. Dies zusammengenommen ergibt für die zu einer Condition C gehörende Dämpfung \mathbf{d} die Form

$$\mathbf{d} = -k_d \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}} \dot{C}(\mathbf{x})$$

Gleichung 13

Wie vorher sind auch hier nur die Komponenten von \mathbf{d} ungleich null für diejenigen Partikel, von denen C abhängt.

In Gleichung 6 werden zusätzlich noch die partiellen Ableitungen benötigt:

$$\frac{\partial d_i}{\partial x_j} = -k_d \left(\frac{\partial C(\mathbf{x})}{\partial x_i} \frac{\partial \dot{C}(\mathbf{x})}{\partial x_j} + \frac{\partial^2 C(\mathbf{x})}{\partial x_i \partial x_j} \dot{C}(\mathbf{x}) \right)$$

Gleichung 14

$\frac{\partial \mathbf{d}}{\partial \mathbf{x}}$ hat das gleiche Belegungsmuster wie $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$. Man beachte, dass $\frac{\partial \mathbf{d}}{\partial \mathbf{x}}$ im Gegensatz zu Gleichung 8 nicht symmetrisch ist. In Gleichung 14 wird diese Asymmetrie durch den Term $\left(\frac{\partial C(\mathbf{x})}{\partial x_i}\right) \left(\frac{\partial \dot{C}(\mathbf{x})}{\partial x_j}\right)^T$ verursacht.

Wie später gezeigt wird, ist es jedoch wichtig, die Symmetrie zu wahren und deshalb diesen Term zu vernachlässigen. Die Vereinfachung ist nicht physikalisch gerechtfertigt, zeigt jedoch keine negativen Effekte (das Weglassen aller Terme in Gleichung 14 hingegen führt zu Problemen).

Für die partielle Ableitung nach \mathbf{v} verwenden wir $\dot{C}(\mathbf{x}) = \left(\frac{\partial C(\mathbf{x})}{\partial \mathbf{x}}\right)^T \mathbf{v}$, so dass gilt:

$$\frac{\partial \dot{\mathbf{C}}(\mathbf{x})}{\partial \mathbf{v}} = \frac{\partial}{\partial \mathbf{v}} \left(\frac{\partial \mathbf{C}(\mathbf{x})^T}{\partial \mathbf{x}} \mathbf{v} \right) = \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}}$$

Wir erhalten so

$$\frac{\partial \mathbf{d}_i}{\partial \mathbf{v}_j} = -k_d \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \frac{\partial \dot{\mathbf{C}}(\mathbf{x})^T}{\partial \mathbf{v}_j} = -k_d \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i} \frac{\partial \mathbf{C}(\mathbf{x})^T}{\partial \mathbf{x}_j}$$

Gleichung 15

Diese Gleichung ist ohne Weglassen von Termen symmetrisch.

5. Constraints

Die einzelnen Partikel in einer Stoffsimulation können verschiedenen Constraints unterworfen sein. Zum einen sind dies durch den Benutzer gegebene Bedingungen, wie z.B. fixe geometrische Position oder Verbindung mit einem anderen Objekt, zum andern gibt es verschiedene Bedingungen, die durch das System erzeugt werden, z.B. aufgrund von Kollisionen. Diese Constraints schränken die Freiheitsgrade der Partikel ein, indem sie die Bewegung in einer oder mehreren Richtungen bestimmen. Die Autoren vergleichen verschiedene Methoden, diese Constraints in das Simulationssystem einzubinden.

5.1. VERSCHIEDENE METHODEN

5.1.1. Reduced Coordinates

Die Zahl der Koordinaten wird durch die Constraints reduziert. Ein Partikel mit Constraints in 1 Richtung hat noch 2 Koordinaten, ein solches mit Constraints in 3 Richtungen hat keine Koordinaten mehr. Dieser Ansatz ist wohl eine Möglichkeit, dadurch werden die Berechnungen jedoch stark verkompliziert, so dass diese Methode schnell verworfen wurde.

5.1.2. Penalty Method

Die Penalty Methode verbietet gewisse Werte und korrigiert sie. Diese Methode berücksichtigt die Constraints jedoch nicht exakt, so dass auch auf diese Methode verzichtet wurde.

5.1.3. Lagrange Multipliers

Hier werden zusätzliche Kräfte eingeführt, die die Constraint-Bedingungen erzwingen. Die dadurch nötigen Änderungen in Gleichung 6 erschweren jedoch die Lösung der Gleichung unzulässig fest, so dass auch von dieser Methode abgesehen wurde.

5.2. MASSENMODIFIKATION

Die Autoren gehen bei ihrem Modell der Constraints von folgender Idee aus:

Die Beschleunigung eines Partikels i ist durch $\ddot{\mathbf{x}}_i = \frac{1}{m_i} \mathbf{f}_i$ gegeben. Die Beschleunigung kann nun durch die inverse Masse $\frac{1}{m_i}$ beeinflusst werden. Wird die inverse Masse auf 0 gesetzt, so kann keine Beschleunigung erfolgen. Für Constraints in einer oder zwei Richtungen muss die inverse Masse nicht als skalar sondern als Diagonalmatrix betrachtet

werden. Angenommen, wir schreiben $\ddot{\mathbf{x}}_i = \begin{pmatrix} \frac{1}{m_i} & 0 & 0 \\ 0 & \frac{1}{m_i} & 0 \\ 0 & 0 & 0 \end{pmatrix}_i \mathbf{f}_i$. Hier muss $\ddot{\mathbf{x}}_i$ in der xy-Ebene liegen, eine Beschleunigung in z-Richtung ist nicht möglich. Constraints sind nicht auf Richtungen der Koordinatenachsen

beschränkt. Soll die Beschleunigung in Richtung des Vektors \mathbf{p}_i verunmöglicht werden, so wird die inverse Masse ausgedrückt durch $\frac{1}{m_i} (\mathbf{I} - \mathbf{p}_i \mathbf{p}_i^T)$, bei zwei orthogonalen Vektoren \mathbf{p}_i und \mathbf{q}_i , in deren Richtungen die Beschleunigung unterdrückt werden soll, ist die inverse Masse $\frac{1}{m_i} (\mathbf{I} - \mathbf{p}_i \mathbf{p}_i^T - \mathbf{q}_i \mathbf{q}_i^T)$. Die Beschleunigung kann nun jedoch nicht nur auf null gesetzt werden, sondern sie kann beliebig beeinflusst werden. Sei \mathbf{z}_i die gewünschte Beschleunigung in den beschränkten Richtungen, so hält man durch ergänzen von Gleichung 6

$$\left(\mathbf{I} - h\mathbf{W} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2 \mathbf{W} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \Delta \mathbf{v} = h\mathbf{W} \left(\mathbf{f}_0 + h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v}_0 \right) + \mathbf{z}$$

Gleichung 16

Dabei ist \mathbf{W} eine Block-Diagonal-Matrix mit den Blöcken in der Diagonale $\mathbf{W}_{ii} = \frac{1}{m_i} \mathbf{S}_i$ mit

$$\mathbf{S}_i = \begin{cases} \mathbf{I} & dof(i) = 3 \\ (\mathbf{I} - \mathbf{p}_i \mathbf{p}_i^T) & dof(i) = 2 \\ (\mathbf{I} - \mathbf{p}_i \mathbf{p}_i^T - \mathbf{q}_i \mathbf{q}_i^T) & dof(i) = 1 \\ \mathbf{0} & dof(i) = 0 \end{cases}$$

6. Kollision

Kollisionen können bei der Stoffsimulation zwischen Stoff/Stoff oder zwischen Stoff/Objekt erfolgen. Stoff/Stoff-Kollisionen werden detektiert, indem Paare von zwei Dreiecksanten bzw. von einem Knoten und einer Dreiecksfläche betrachtet werden. Ausgehend vom bisherigen, legalen Zustand wird lineare Bewegung zum neuen Zustand angenommen und auf Durchdringung von zwei Dreiecksanten bzw. einem Knoten und einem Dreieck geprüft (\rightarrow Abbildung 4). Wird eine Kollision festgestellt, so wird eine stark gedämpfte Sprungkraft eingeführt, die die Stoffschichten wieder auseinander drückt. Zusätzlich wird eine Reibungskraft eingesetzt, die der Gleitbewegungen entgegenwirkt. Genaugenommen ist diese Kraft keine Reibungskraft: Da sie proportional zur Gleitgeschwindigkeit eingesetzt wird, handelt es sich eher um eine Dämpfung, die jedoch die dynamische Reibung auf vernünftige Art emuliert. Statische Reibungskräfte sind bedeutend komplizierter und ein bis jetzt ungelöstes Problem.

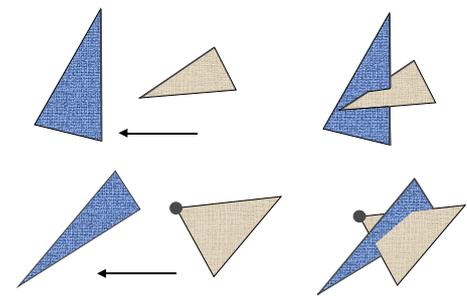


Abbildung 4

Kollisionen mit Objekten werden detektiert, indem jeder Knoten des Stoffgitters mit den Flächen jedes Objektes geprüft wird. Dazu wird ein hierarchischer Baum von Bounding Boxes verwendet (\rightarrow Abbildung 5). Für Knoten, bei denen eine Kollision festgestellt wurde, werden entsprechende Constraints eingefügt.

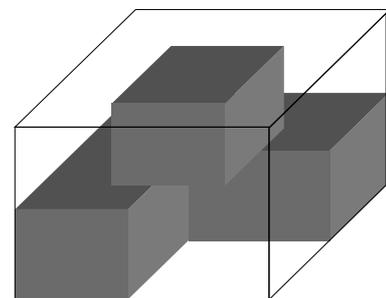


Abbildung 5

Da die Simulation in diskreten Schritten arbeitet, kann es sowohl bei der Kollision zwischen Stoffen als auch bei der Kollision mit Objekten geschehen, dass die Durchdringung bis zur Detektion bereits bis zu einer gewissen Tiefe erfolgt ist. Bei Stoff/Stoff-Kollisionen stellt dies kein Problem dar, da die Sprungkraft die Stoffschichten wieder auseinander drückt. Bei der Kollision mit festen Objekten wird durch die eingefügten Constraints nur die Geschwindigkeit der Stoffpartikels an diejenige des Objektes angeglichen, so dass das Partikel im Innern des Objektes bleibt. Der Ansatz, die Beschleunigung so zu beeinflussen, dass der Fehler in der Position und Geschwindigkeit behoben wird, hat sich nicht bewährt. Als nächstes versuchten die Autoren eine einfachere Lösung, bei der die fehlerhafte Position Partikels auf einen legalen Wert gesetzt wurde. Dabei entstehen jedoch grosse Streckungskräfte, die zu sichtbaren Artefakten führen.

Diese Kräfte kommen daher, dass die Nachbarschaft eines "verschobenen" Partikels über dessen Positionsänderung nicht "informiert" wird und so zu Beginn des nächsten Schrittes mit dieser Änderung konfrontiert wird. Das führt zu folgender Lösung: Die obere Zeile der Gleichung 5 wird ersetzt durch

$$\Delta \mathbf{x}_i = h(\mathbf{v}_{0i} + \Delta \mathbf{v}_i) + \mathbf{y}_i$$

Gleichung 17

wobei \mathbf{y} ein beliebiger Korrekturterm unserer Wahl ist, ausschliesslich eingefügt, um ein bestimmtes Partikel beim Rückwärts-Eulerschritt an eine gewünschte Position zu bringen. Durch Weiterziehen dieser Änderungen durch die Ableitung und die Umformungen der vorhergehenden Kapitel erhält man

$$\left(\mathbf{I} - h\mathbf{W} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2 \mathbf{W} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \Delta \mathbf{v} = h\mathbf{W} \left(\mathbf{f}_0 + h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{y} \right) + \mathbf{z}$$

Gleichung 18

Diese Änderungen geben uns vollständige Kontrolle über Position und Geschwindigkeit ohne zusätzliche Berechnungen.

7. Lösung des Gleichungssystems

7.1. VORBEMERKUNGEN

Zur Lösung des Gleichungssystems wollen die Autoren die Conjugate Gradient Methode verwenden. CG ist gut geeignet zur Lösung spärlich besetzter Gleichungssysteme, wie sie bei unserem Problem anfallen, benötigt jedoch symmetrische Matrizen. Gleichung 16 ist jedoch nicht symmetrisch. Ohne Constraints gibt es keine Probleme, da Gleichung 6 durch Multiplikation mit \mathbf{M} von links in das symmetrische System

$$\left(\mathbf{M} - h \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2 \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \Delta \mathbf{v} = h \left(\mathbf{f}_0 + h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v}_0 \right)$$

Gleichung 19

umgewandelt werden kann. Selbe Transformation kann jedoch nicht auf Gleichung 16 angewandt werden, da \mathbf{W} singular ist. Stattdessen wird CG so modifiziert, dass die Methode auf Gleichung 19 arbeitet, während die in der Matrix \mathbf{W} enthaltenen Constraints prozedural hinzugefügt werden. Wir definieren

$$\mathbf{A} = \left(\mathbf{M} - h \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - h^2 \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)$$

$$\mathbf{b} = h \left(\mathbf{f}_0 + h \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v}_0 \right)$$

$$\mathbf{r} = \mathbf{A} \Delta \mathbf{v} - \mathbf{b}$$

Gleichung 20

Gegeben \mathbf{A} , \mathbf{b} , \mathbf{z} und Constraints auf den Partikeln, die modifizierte CG Methode sucht ein $\Delta \mathbf{v}$, das folgende zwei Bedingungen erfüllt:

1. für jedes Partikel i , die Komponenten von \mathbf{r}_i in Richtungen ohne Constraints sind gleich 0 (eine genügende Anzahl Iteration vorausgesetzt)
2. für jedes Partikel i , die Komponenten von $\Delta \mathbf{v}_i$ in Richtungen mit Constraints sind gleich \mathbf{z}_i (unabhängig von der Anzahl Iterationen)

7.2. PSEUDO-CODE

<pre> procedure filter(a) for i=1 to n $\hat{\mathbf{a}}_i = S_i \mathbf{a}_i$ return $\hat{\mathbf{a}}$ </pre>	<pre> procedure modified - pcg $\Delta \mathbf{v} = \mathbf{z}$ $\delta_0 = \text{filter}(\mathbf{b})^T \mathbf{P} \text{filter}(\mathbf{b})$ $\mathbf{r} = \text{filter}(\mathbf{b} - \mathbf{A} \Delta \mathbf{v})$ $\mathbf{c} = \text{filter}(\mathbf{P}^{-r} \mathbf{r})$ </pre>	<pre> $\delta_{new} = \mathbf{r}^T \mathbf{c}$ while $\delta_{new} > \varepsilon^2 \delta_0$ $\mathbf{q} = \text{filter}(\mathbf{A} \mathbf{c})$ $\alpha = \delta_{new} / (\mathbf{c}^T \mathbf{q})$ $\Delta \mathbf{v} = \Delta \mathbf{v} + \alpha \mathbf{c}$ </pre>	<pre> $\mathbf{r} = \mathbf{r} - \alpha \mathbf{q}$ $\mathbf{s} = \mathbf{P}^{-1} \mathbf{r}$ $\delta_{old} = \delta_{new}$ $\delta_{new} = \mathbf{r}^T \mathbf{s}$ $\mathbf{c} = \text{filter}\left(\mathbf{s} + \frac{\delta_{new}}{\delta_{old}} \mathbf{c}\right)$ </pre>
---	---	---	---

\mathbf{P} ist ein einfacher Preconditioner zur Verbesserung der Konvergenz. \mathbf{P} ist eine Diagonalmatrix mit $\mathbf{P}_{ii} = 1/\mathbf{A}_{ii}$

8. Schrittweite anpassen

Die beschriebene Methode erlaubt beträchtliche Schritte vorwärts in der Zeit ohne Stabilitätsverlust. Trotzdem gibt es Situationen, in denen die Schrittweite reduziert werden muss. Dazu wird nach jedem Schritt $\Delta \mathbf{x}$ als Vorschlag für eine Zustandsänderung betrachtet. Nun werden die Streckungsterme für alle Dreiecke in diesem neuen Zustand untersucht. Unterliegt die Streckung in einem Dreieck starken Änderungen, so wird $\Delta \mathbf{x}$ verworfen, die Schrittweite reduziert und anschliessend eine neues $\Delta \mathbf{x}$ berechnet. Nach einigen erfolgreichen Berechnungen wird versucht, die Schrittweite wieder zu erhöhen.

9. Ergebnisse



Bild 1



Bild 2



Bild 3



Bild 4

Die Autoren zeigen mit zwei Untersuchungen am Bild 1 die Güte ihres Verfahrens auf:

Als erstes wurde der Steifheitsfaktor des Bildes 1 auf Werte von 0.1, 1, 10, 100, 1'000 relativ zum Wert in Bild 1 gesetzt. Obwohl sich dabei die Steifheit in einem Bereich von 10'000 bewegt, bleibt die Änderung bei der Rechenzeit unter 5%.

Als nächstes wurden die Anzahl der Knoten in Bild 1 verändert. Die Knotenzahlen und die Rechenzeiten können der folgenden Tabelle entnommen werden

Anzahl Knoten	Rechenzeit (in Sekunden/Frame)
500	0.23
899	0.46
2602 (Bild 1)	2.23
7359	10.3

Das ist leicht besser als $O(n^{1.5})$