

Polygonnetz-freie Deformationen mittels Formangleich

Christian Georg Wehrberger
christian@wehrberger.de

Seminar Computergraphik
Institut für Informatik
Georg-August-Universität Göttingen

24.01.2012



Gliederung

- 1 Motivation
- 2 Grundlagen
- 3 Deformationen
- 4 Erweiterungen
- 5 Ergebnisse
- 6 Referenzen



Motivation



Einleitung

- Vielfältige Ansätze zur Deformations-Beschreibung
 - Physikalisch akkurat
 - Auf Stabilität fokussiert
- Keine Verwendung in Computerspielen
 - Stabil, aber rechenintensiv
 - Ungeignet für degenerierte Geometrien
 - Geringe Anpassbarkeit

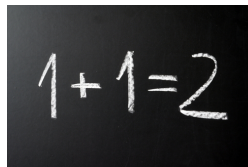


Verwandte Arbeiten

- Elastisch deformierbare Modelle
- Masse-Feder-Modelle
- Randelement-Methode
- Finite-Elemente-Methode
- Finite-Volumen-Methode
- Polygonnetz-freie Partikel-Systeme



Grundlagen


$$1+1=2$$



Numerische Integration

- Implizite Integration: $G(X(t), X(t + \Delta t)) = 0$
 - Unbedingt numerisch stabil
- Explizite Integration: $X(t + \Delta t) = F(X(t))$
 - Schnellere Berechnung
 - Bedingte numerische Stabilität
 - Abhängigkeit der Konvergenzgeschwindigkeit und Stabilität von Spektralradius ρ der System-Matrix A :

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i(A)| \stackrel{!}{\leq} 1 \quad (\lambda_i \text{ Eigenwerte von } A)$$



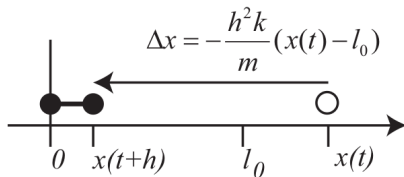
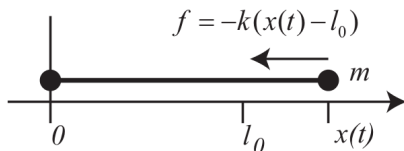
Explizite vs. implizite Integration

- Beispiel: $f(x) = x^2$ mittels Euler-Verfahren
- Explizite Integration
 - $f(x+h) = f(x) + 2h\sqrt{f(x)}$
- Implizite Integration
 - $f(x+h) = f(x) + 2h\sqrt{f(x+h)}$
 - Man erhält: $f(x+h) = 2h^2 \pm \sqrt{4h^2 - f(x)^2}$



Stabilität numerischer Integration

- Beispiel: modifiziertes Euler-Schema

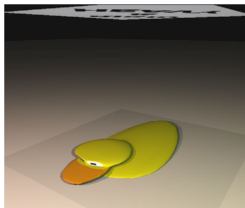


Stabilität numerischer Integration

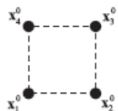
- Beispiel: modifiziertes Euler-Schema
- $v(t+h) = v(t) + h \frac{-k(x(t)-l_0)}{m}$, $x(t+h) = x(t) + hv(t+h)$
- Gleichungssystem:
$$\begin{pmatrix} 1 & -\frac{hk}{m} \\ h & 1 - \frac{h^2k}{m} \end{pmatrix} \begin{pmatrix} v(t) \\ x(t) \end{pmatrix} = \begin{pmatrix} v(t+h) \\ x(t+h) \end{pmatrix}$$
- Eigenwerte der System-Matrix:
$$\lambda_{1/2} = 1 - \frac{1}{2m}(h^2k \pm \sqrt{-4mh^2k + h^4k^2})$$
- Forderung zur Konvergenz der iterativen Berechnung:
$$\rho(A) \leq 1, \text{ d. h. } h \leq 2\sqrt{\frac{m}{k}}$$



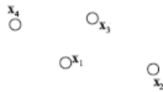
Deformationen



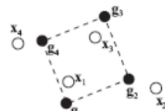
Algorithmus



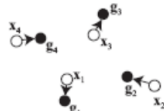
(1)



(2)



(3)



(4)

- 1 Initiale Konfiguration
- 2 Deformierte Konfiguration
- 3 Formangleich
- 4 "Hinziehen" der Partikel zu formangepasster Konfiguration



Formangleich

- Gegeben: Punktemengen x_i^0 und x_i
- Gesucht: Rotationsmatrix R und Verschiebungen t_0 , t , sodass der Ausdruck $\sum_i w_i (R(x_i^0 - t_0) - (x_i - t))^2$ minimiert wird (w_i Gewichtung)
- Ergebnis: $t_0 = \frac{\sum_i w_i x_i^0}{\sum_i w_i} := x_{cm}^0$, $t = \frac{\sum_i w_i x_i}{\sum_i w_i} := x_{cm}$,
$$R = A_{pq} \sqrt{A_{pq}^T A_{pq}}^{-1} \quad \text{mit} \quad A_{pq} := \sum_i w_i (x_i - x_{cm})(x_i^0 - x_{cm}^0)^T$$
- Zielpositionen $g_i = R(x_i^0 - x_{cm}^0) + x_{cm}$



Integration

- Wissen über Zielpositionen ermöglicht bedingungslose numerische Stabilität
- $v_i(t+h) = v_i(t) + \alpha \frac{g_i(t) - x_i(t)}{h} + h \frac{f_{\text{ext}}(t)}{w_i}$
($\alpha \in [0 \dots 1]$ Steifigkeits-Parameter)
- $x_i(t+h) = x_i(t) + hv_i(t+h)$
- System-Matrix $\begin{pmatrix} 1 & -\frac{\alpha}{h} \\ h & 1 - \alpha \end{pmatrix}$
- Eigenwerte der System-Matrix $\lambda_{1/2} = (1 - \frac{\alpha}{2}) \pm \frac{i}{2} \sqrt{\alpha(4 - \alpha)}$,
also $|\lambda_{1/2}| = 1$

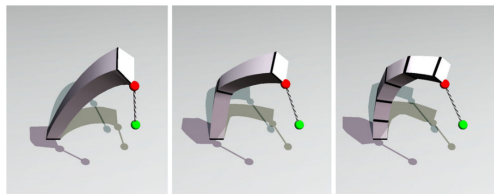


Diskussion des Ansatzes

- Berechnung:
 - Vorberechnung der x_{cm}^0 und $x_i^0 - x_{cm}^0$
 - In jedem Zeitschritt Berechnung von A_{pq} und des Ausdrucks
$$\sqrt{A_{pq}^T A_{pq}}^{-1}$$
- Ansatz sinnvoll für kleine Deformationen
- Physikalischer Realismus:
 - Impulserhaltung
 - Drehimpulserhaltung
- Simulation starrer Körper bei $\alpha = 1$



Erweiterungen



Lineare Deformationen

- Lineare Transformation $\beta A + (1 - \beta)R$ (A allgemeine Matrix) ersetzt Rotationsmatrix R
- Weiterhin Rotationsmatrix zur Formerverhaltung
- Keine höhere Berechnungsintensität, da analog (teilweise) vorberechenbar

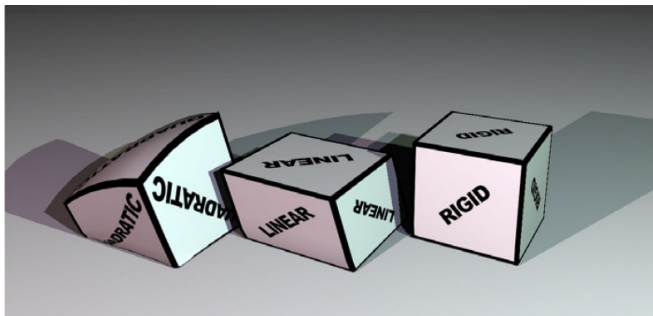


Quadratische Deformationen

- Quadratische Transformation $\beta \tilde{A} + (1 - \beta) \tilde{R}$
 $(\tilde{A} = [A \ Q \ M] \in \mathbb{R}^{3 \times 9})$
- Rotationsmatrix $\tilde{R} = [R \ 0 \ 0] \in \mathbb{R}^{3 \times 9}$ zwecks Formerhaltung
- Zielpositionen $\tilde{g}_i = \tilde{A} \tilde{q}_i$ mit
 $\tilde{q} = (q_x \ q_y \ q_z \ q_x^2 \ q_y^2 \ q_z^2 \ q_x q_y \ q_y q_z \ q_z q_y)^T$ und
 $q_{i\alpha} = (x_i^0 - x_{cm}^0)_\alpha \ (\alpha \in \{x, y, z\})$
- Resultat $\tilde{A} = (\sum_i w_i (x_i - x_{cm}) \tilde{q}_i^T) (\sum_i w_i \tilde{q}_i \tilde{q}_i^T)$

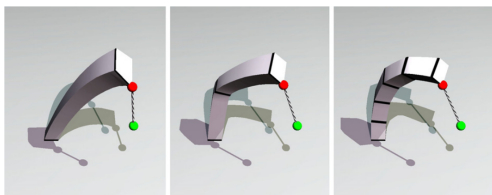


Vergleich linearer und quadratischer Deformation



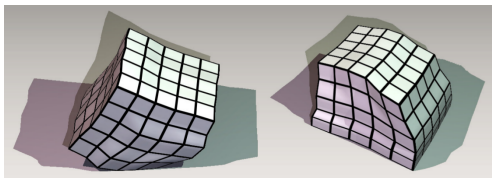
Deformation von Punktgruppen

- Kubische Punktgruppen (Cluster)
- In jedem Zeitschritt Geschwindigkeitsaddition in Richtung Punktgruppen-Zielpunkt g_i^c : $\Delta v_i = \alpha \frac{g_i^c - x_i}{h}$
- Höherer Aufwand, größere Flexibilität



Plastizität

- Simulation plastischen Verhaltens
- Polarzerlegung von Transformationsmatrix A in Rotations- und Deformationsanteil R bzw. S
- In jedem Schritt, in dem $\|S - I\|_2 > c_{yield}$, aktualisiere Status der plastischen Deformation durch $S_{n+1}^P = (I + hc_{creep}(S - I))S_n^P$ mit Zeitschritt h und Kontrollparameter c_{creep}
- Anwendung auf $q_i = S^P(x_i^0 - x_{cm}^0)$

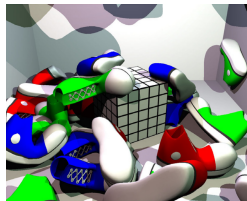


Vergleich mit anderen Deformationsmodellen

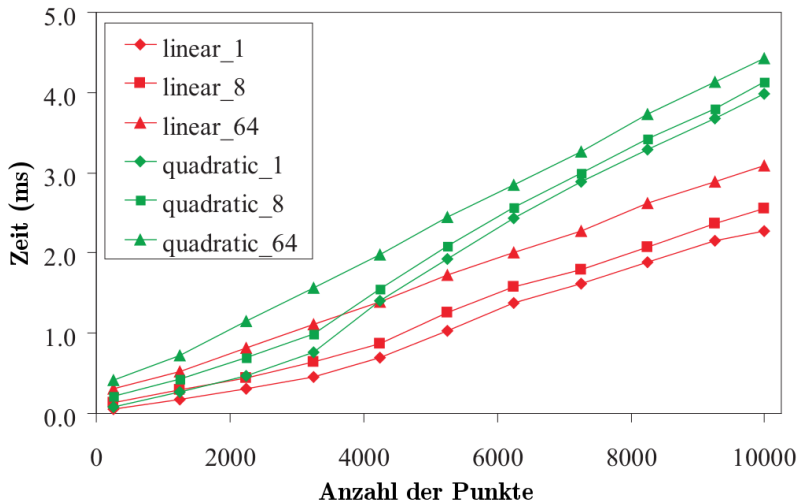
- Hohe Effizienz durch einfache Berechnung
- Unbedingte numerische Stabilität
- Anpassbarkeit durch linearen/quadratischen Formangleich
- Keine aufwendigen Vorberechnungen
- Physikalisch näherungsweise realistisch



Ergebnisse



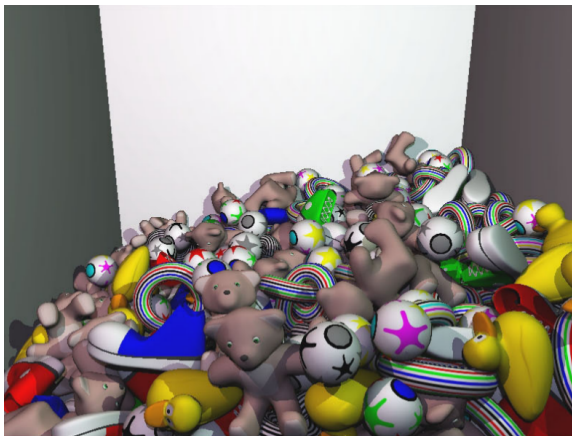
Zeitverhalten



Weitere Szenarien



Weitere Szenarien



Ergebnisse

- Höhere Genauigkeit nicht durch höhere Deformationsmoden garantiert
- Kein physikalischer Realismus
- Kollisionserkennung fehlt
 - Flaschenhals im Zeitverhalten
- Keine Vorbearbeitung der Daten notwendig
- Keine Hilfsstrukturen nötig
- Unbedingte numerische Stabilität der Simulation
- Schnelle Berechnung
- Einfache Implementierung



Referenzen

- 1 **M. Müller, B. Heidelberger, M. Teschner, M. Gross:** Meshless deformations based on shape matching, ACM Transactions on Graphics (TOG), Band 24 (3), Seiten 471–478, 2005
- 2 **M. Müller, B. Heidelberger, M. Teschner, M. Gross:** Meshless deformations based on shape matching (Report), SIGGRAPH 2005
- 3 **D. Terzopoulos, J. Platt, A. Barr, K. Fleischer:** Elastically deformable models, ACM Siggraph Computer Graphics, Band 21 (4), Seiten 205–214, 1987
- 4 **Wolfram MathWorld (by Eric Weisstein):** Euler Forward Method,
mathworld.wolfram.com/EulerForwardMethod.html
(Zuletzt zugegriffen am 22.01.2012)



Bildquellen

- **Folie 6:** <http://sme-blog.com/files/2010/09/Business-Basics.jpg>
(Zuletzt zugegriffen am 22.01.2012)
- **Folie 11:** [2]
- **Folie 28:** http://www.n24.de/media/_fotos/bildergalerien/wulff/wu7.jpg (Zuletzt zugegriffen am 22.01.2012)
- **Weitere Graphiken:** [1]



Bemerkungen und Fragen



Bitte klären Sie alle Ihre Fragen möglichst direkt. Bemerkungen und Fragen können aber auch im Nachhinein gerichtet werden an:

christian@wehrberger.de

Vielen Dank für Ihre Aufmerksamkeit!

