

Simulating Visual Geometry

Patrick Bayer


23. Januar 2017

Inhalt

- 1 Konstruktion eines Simulationsnetzes
- 2 Die Simulationmethode
- 3 Deformation der Primitive
- 4 Das Visualisierungsnetz
- 5 Plastische Deformation
- 6 Brüche und Risse

Konstruktion eines Simulationsnetzes

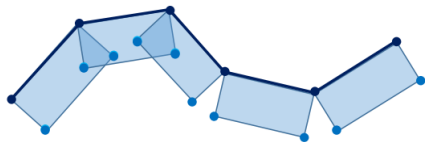
Input:


- Netz zur Visualisierung bestehend aus drei- oder viereckigen Elementen 
- Physikalische Eigenschaften des Objektes (Materialdicke, ...)

Idee:

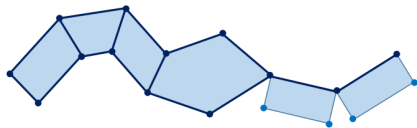
- Konvexe Formen als Primitive für das Simulationsnetz

Konstruktion eines Simulationsnetzes



(1) Elemente werden nach innen
erweitert. 

⇒ volumetrische, konvexe Primitive



(2) Bereits vorhandene konvexe
Formen werden direkt in Primitive
transformiert

Konstruktion eines Simulationsnetzes

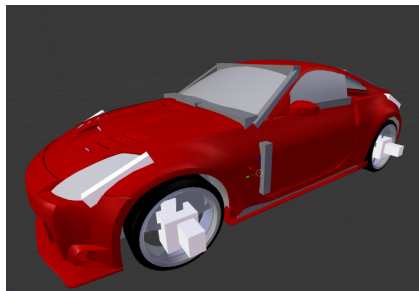
Verbinden der Primitive:

- Ausgangssituation: Ruhelage des Objektes
- Graph mit Primitiven als Knoten und Verbindungen als Kanten
- Zwei Primitive i, j sind verbunden \Leftrightarrow Die Primitive i, j berühren oder überlappen sich


Konstruktion eines Simulationsnetzes

Verbinden von Subnetzen:

- Verwendung von Joints definiert durch Boxen
- Ball joint, Hinge-Joint, Slider Joints



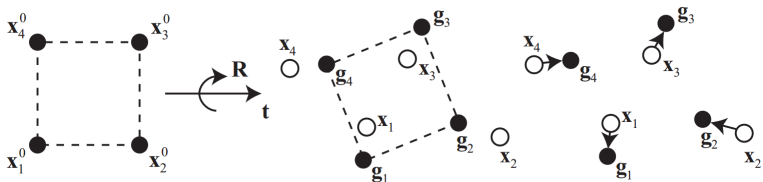
Die Simulationemethode

- **Oriented particle method**
- Einfach und schnell 
- Partikel haben neben den linearen Attributen (Position, Geschwindigkeit) eine Rotation und Winkelgeschwindigkeit
- Basiert auf Shape Matching und Position Based Dynamics

Die Simulationmethode

Generalized Shape Matching:

- Arbeitet mit Partikeln ohne Verbindungen
- Simulation von Elastizität



Die Simulationsmethode

- Berechne \mathbf{R} , $\bar{\mathbf{t}}$ und \mathbf{t} durch Minimieren von

$$\sum_i \mathbf{m}_i (\mathbf{R}(\bar{\mathbf{x}}_i - \bar{\mathbf{t}}) + \mathbf{t} - \mathbf{x}_i)^2$$

- Es stellt sich heraus, dass

$$\bar{\mathbf{t}} = \bar{\mathbf{x}}_{cm} = \frac{\sum_i \mathbf{m}_i \bar{\mathbf{x}}_i}{\sum_i \mathbf{m}_i} \text{ and } \mathbf{t} = \mathbf{x}_{cm} = \frac{\sum_i \mathbf{m}_i \mathbf{x}_i}{\sum_i \mathbf{m}_i}$$

- Seien $\mathbf{q}_i = \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_{cm}$ und $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}$, dann minimiere

$$\sum_i \mathbf{m}_i (\mathbf{A} \mathbf{q}_i - \mathbf{p}_i)^2$$

wobei \mathbf{A} die optimale lineare Transformation ist.

Die Simulationsmethode

- Es stellt sich heraus, dass

$$\mathbf{A} = (\sum_i \mathbf{m}_i \mathbf{p}_i \mathbf{q}_i^T) (\sum_i \mathbf{m}_i \mathbf{q}_i \mathbf{q}_i^T)^{-1} = \mathbf{A}_{pq} \mathbf{A}_{qq}$$

- \mathbf{A}_{qq} ist symmetrisch, enthält also keine Rotation
- Eine Polarzerlegung von \mathbf{A}_{pq} liefert

$$\mathbf{A}_{pq} = \mathbf{R} \mathbf{S} \text{ wobei } \mathbf{S} = \sqrt{\mathbf{A}_{pq}^T \mathbf{A}_{pq}} \text{ und } \mathbf{R} = \mathbf{A}_{pq} \mathbf{S}^{-1}$$

- Berechne die Zielposition jedes Partikels

$$\mathbf{g}_i = \mathbf{R}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_{cm}) + \mathbf{x}_{cm}$$

Die Simulationsmethode

- Co-Planarität oder Co-Linearität der Punkte \Rightarrow Unstabilität
- \mathbf{A}_{pq} für zwei Partikelmengen mit Moment Matrizen \mathbf{A}_1 und \mathbf{A}_2

$$\mathbf{A}_{pq} = \sum_i \mathbf{m}_i \mathbf{x}_i \bar{\mathbf{x}}_i^T - M \mathbf{x}_{cm} \bar{\mathbf{x}}_{cm}^T$$

- \mathbf{A}_{pq} mit Moment Matrizen \mathbf{A}_i für jedes einzelne Partikel

$$\mathbf{A}_{pq} = \sum_i (\mathbf{A}_i + \mathbf{m}_i \mathbf{x}_i \bar{\mathbf{x}}_i^T) - M \mathbf{x}_{cm} \bar{\mathbf{x}}_{cm}^T$$

wobei M die Summe der Massen \mathbf{m}_i ist

Die Simulationemethode

- Berechne die Moment Matrizen der einzelnen Partikel mit orthonormaler Orientierungsmatrix \mathbf{R} für Kugeln mit Radius r Volumen V_r und Ellipsoide mit Radien a, b und c

$$\mathbf{A}_{sphere} = \int_{V_r} \rho(\mathbf{R}\mathbf{x})\mathbf{x}^T dV = \frac{1}{5}mr^2\mathbf{R}$$

$$\mathbf{A}_{ellipsoid} = \frac{1}{5}m \begin{bmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{bmatrix} \mathbf{R}$$

Generalized Position Based Dynamics

- Integrationsmechanismus zur Simulation von Partikeln
- Partikel haben eine Position \mathbf{x} , eine Geschwindigkeit \mathbf{y} , eine Rotation \mathbf{q} und eine Winkelgeschwindigkeit ω
- Schritt 1: Vorhersage von \mathbf{x} und \mathbf{q}

$$\mathbf{x}_p \leftarrow \mathbf{x} + \mathbf{v}\Delta t$$

$$\mathbf{q}_p \leftarrow \left[\frac{\omega}{|\omega|} \sin\left(\frac{|\omega|\Delta t}{2}\right), \cos\left(\frac{|\omega|\Delta t}{2}\right) \right] \mathbf{q}$$

- Schritt 2: Korrektur der Positionen \mathbf{x} anhand gegebener

Einschränkungen



Die Simulationsmethode

- Schritt 3: Update von \mathbf{x} , \mathbf{v} , \mathbf{q} , ω

$$\mathbf{v} \leftarrow \frac{\mathbf{x}_p - \mathbf{x}}{\Delta t}$$

$$\mathbf{x} \leftarrow \mathbf{x}_p$$

$$\mathbf{q} \leftarrow \mathbf{q}_p$$

$$\omega \leftarrow \frac{\text{axis}(\mathbf{q}_p \mathbf{q}^{-1}) * \text{angle}(\mathbf{q}_p \mathbf{q}^{-1})}{\Delta t}$$

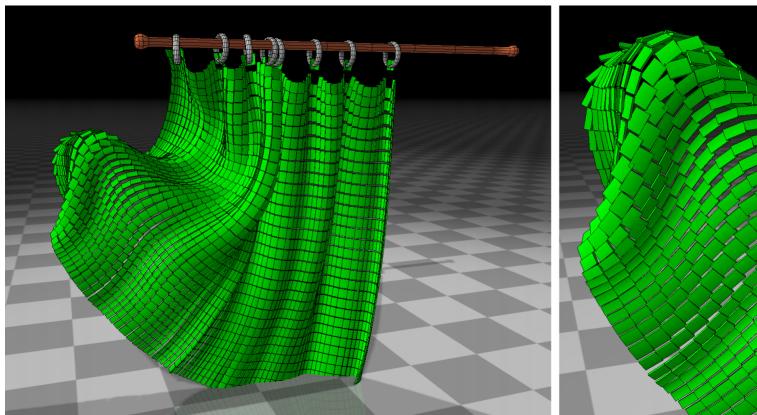
Die Simulationemethode

Collision Handling:

- Finde Überschneidungen von Objektgrenzen
- Identifiziere alle beteiligten Primitive und prüfe für jedes Paar, ob eine Kollision auftritt
- Seperating Axis Theorem

Deformation der Primitive

- Oriented Particle Method führt zu visuellen Fehlern (Lücken, ...)



Deformation der Primitive

- Konvexe Eigenschaft der Primitive muss erhalten bleiben \Rightarrow Nutze die Local Affine Deformation Matrix \mathbf{A}_{pq}
- Berechne die Deformationsmatrix \mathbf{D}

$$\mathbf{D} = \mathbf{A}\bar{\mathbf{A}}^{-1}$$

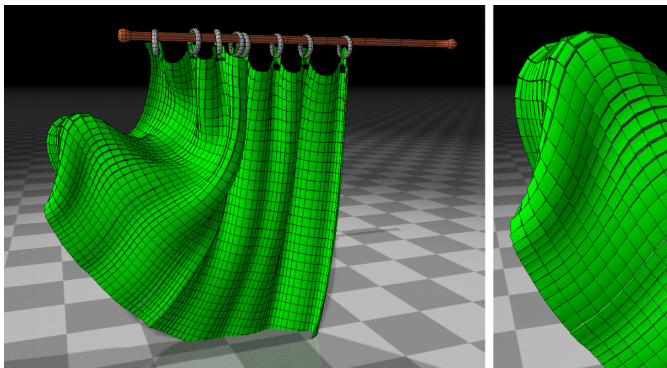
wobei

$$\bar{\mathbf{A}} = \sum_i (\bar{\mathbf{A}}_i + \mathbf{m}_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T) - M \bar{\mathbf{x}}_{cm} \bar{\mathbf{x}}_{cm}^T$$

und

$$\mathbf{A}_i = \frac{1}{5} m r^2 \mathbf{I}$$

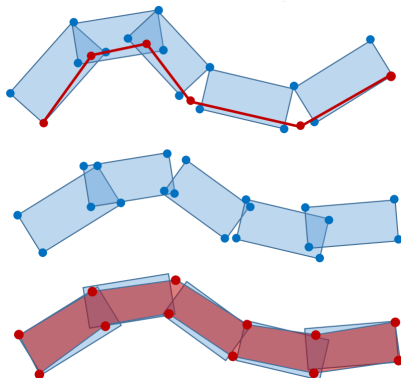
Das Visualisierungsnetz



- Auch nach der Deformation der Primitive sind kleine Lücken zu finden

Das Visualisierungsnetz

- Gruppiere die Eckpunkte der Primitive nach ihren Positionen
- Berechne den Durchschnitt der Positionen
- Die visuelle Position aller beteiligten Primitive ist der Durchschnitt der Positionen



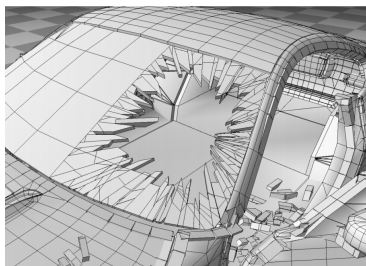
Plastische Deformation

- Handle Objekte wie Starrkörper bis eine Deformation eintritt
- Schwellenwert für die Geschwindigkeit an den Kontaktpunkten
- $\text{Deformationsoffset} = \text{Geschwindigkeit} * \text{Time Step Size}$
- Primitive, die nach einer Deformation einen Joint überschneiden, werden als fix betrachtet

Brüche und Risse

Trennbrüche:

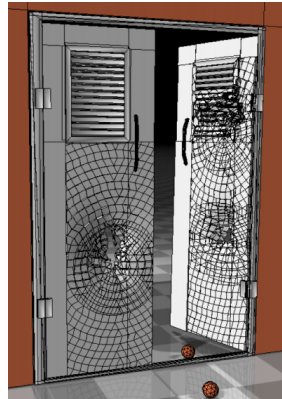
- Menge von verbundenen konvexen Polyedern als Bruchmuster
bestehend aus Zellen
- Finde Überschneidungen der
Objektgrenzen mit dem Bruchmuster
und alle beteiligten Primitive
⇒ Neues, unabhängiges Objekt
bestehend aus allen Primitiven
innerhalb einer Zelle



Brüche und Risse

Verformungsbrüche und Risse:

- Behalte alle resultierenden Teile nach einem Bruch in demselben Objekt
⇒ Das verfeinerte Netz erlaubt detaillierte Deformationen



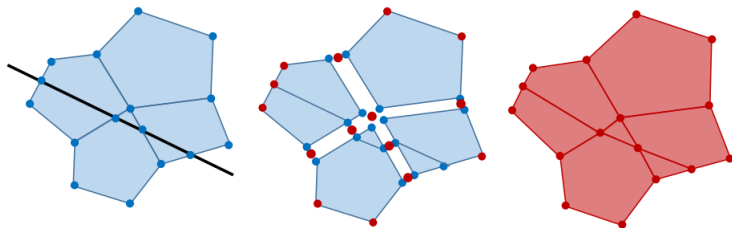
Brüche und Risse

- Schwellenwert für die Distanz zwischen zwei Primitiven zur Identifizierung von Rissen
- Nutzer kann Risslinien vordefinieren, indem nur eine Teilmenge der Verbindungen erlaubt ist zu reißen

Brüche und Risse

Visualisierung von Brüchen:

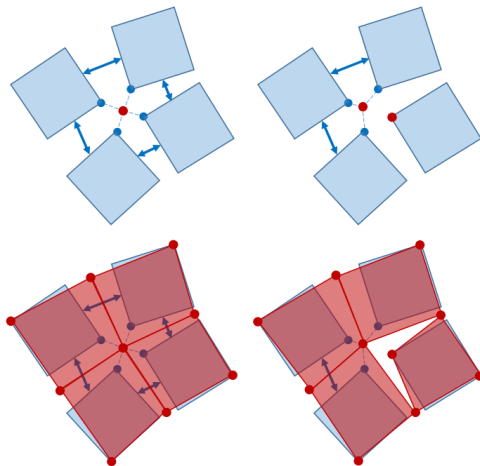
- Das Bruchmuster wird im nicht deformierten Zustand ausgerichtet
⇒ Sowohl die alten als auch die neuen Primitive erhalten dieselbe *id*





Visualisierung von Rissen:

- Berechne $val(id)$ als Anzahl der Primitive mit derselben id
- Besuche alle Nachbarn mit derselben id im Verbindungsgraphen
- Falls die Anzahl der besuchten Eckpunkte kleiner als $val(id)$ ist, ist ein Riss aufgetreten
- Die besuchten Eckpunkte erhalten eine neue id

Brüche und Risse



Zusammenfassung

- Methode zur Simulation und Visualisierung von Deformationen, Brüchen und Rissen 
- Simulationsnetz, das direkt aus einem Visualisierungsnetz hervor geht 
- Convexe Polyeder als Primitive
- Oriented Particle Method

Alle Informationen und Bilder stammen aus folgenden Quellen:

- Matthias Müller, Nuttapong Chentanez, Miles Macklin: Simulating Visual Geometry, Proceedings of the 9th International Conference on Motion in Games, 2016
- MÜLLER M., CHENTANEZ N.: Solid simulation with oriented particles. ACM Trans. Graph. 30, 4 (July 2011), 92:1–92:10
- MÜLLER M., HEIDELBERGER B., TESCHNER M.: Meshless deformations based on shape matching. In Proc. SIGGRAPH 2005 (2005), pp. 471–478