ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

computer graphics lab

Practical Animation of Liquids

N. Foster, R. Fedkiw, *Proceedings of SIGGRAPH '01*

Seminar "Physically-based methods for 3D games and medical applications"

Wintersemester 02/03
Marco Fischer

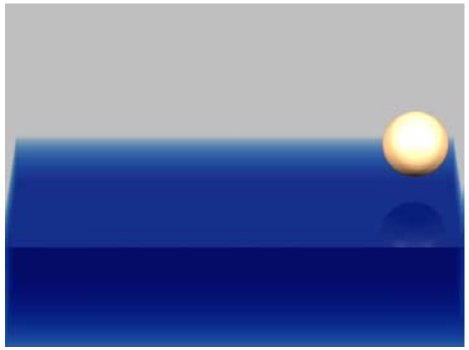
Practical Animation of Liquids

- Proceedings of SIGGRAPH '01
- Die Autoren
  - Nick Foster, PDI/Dreamworks

  - Ronald Fedkiw, Stanford University

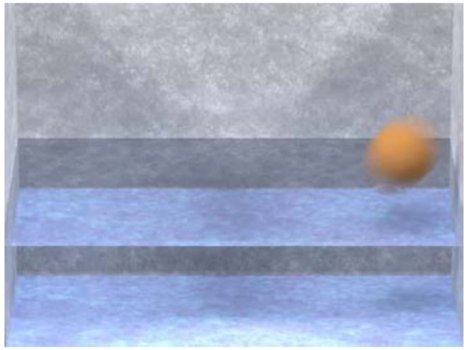
Inhalt

Motivation



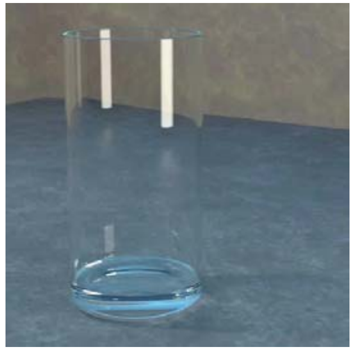
N. Foster, R. Fedkiw. „Practical Animation of Liquids", *Proceedings of SIGGRAPH '01*

Motivation



D. Enright, S. Marschner, R. Fedkiw. „Animation and Rendering of Complex Water Surfaces“, *SIGGRAPH '02*

Motivation

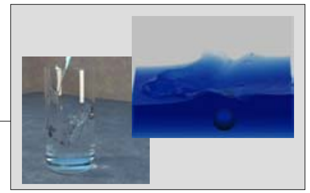


D. Enright, S. Marschner, R. Fedkiw. „Animation and Rendering of Complex Water Surfaces“, *SIGGRAPH '02*

Motivation

PDI/Dreamworks, Shrek

Inhalt



Einführung

- Physikalisch basierte Animationstools
  - Gesetze aus der Physik, um die reale Welt zu modellieren
- Ziel der Computergrafik sind coole und schnelle Animationen
  - Manchmal auf Kosten der physikalischen Korrektheit
  - Trotzdem muss es plausibel aussehen
- Im Bereich Flüssigkeiten ist die Entwicklung in vollem Gang
  - Komplex, rechenintensiv
  - Nicht direkt umsetzbar (wie bspw. Feder-Masse-Modelle)
- Verhalten von Flüssigkeiten modellieren
  - Aus den Gesetzen der Fluiddynamik

9

Einführung - Flüssigkeitsmodelle

- Erste Flüssigkeitsmodelle in der Computergrafik berücksichtigten nur die Oberfläche
  - Die Bewegung der Flüssigkeit darunter wurde ignoriert
- Wirklichkeitsnahe Simulation von Flüssigkeiten ist rechenintensiv!
  - Zwei Alternativen zur Modellierung:
    - » Partikel, Kräfte aus der Physik: Lagrange
    - » Unterteilung in Raumelemente, Differentialgleichungen beschreiben den Fluss zwischen benachbarten Elementen: Euler
  - Berechnungsaufwand mindestens proportional zur Auflösung^3!
- Fluiddynamik (Computational Fluid Dynamics, CFD)
  - Wie werden diese Differentialgleichungen gelöst?
- Computergrafik
  - Die Resultate müssen plausibel aussehen, dabei ist aber die Geschwindigkeit meist wichtiger als physikalisch korrekte Berechnung
  - Diskretisierung, Approximation

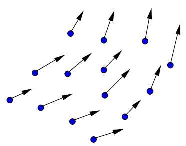
10

Flüssigkeitsmodelle – Lagrange oder Euler?

- Zwei Ansätze:

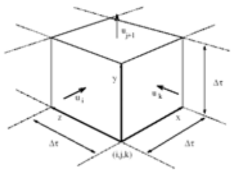
Lagrange

Fluid als Menge von Partikeln
Jedes Partikel besitzt eine Geschwindigkeit



Euler

Raum in Voxel unterteilt
Geschwindigkeitsvektoren zu jedem Nachbarn
Druckvariable im Zentrum



11

Überblick – Inhalt des Papers

- Allgemeines Verfahren
  - Modellierung und Animation von Flüssigkeiten
- Auf die Bedürfnisse der Computeranimation ausgelegt
  - Flüssigkeiten unterschiedlicher Viskosität (Wasser bis dicker Schlamm)
  - Bewegen sich in einer 3D-Umgebung
  - Interagieren mit graphischen Primitiven (parametrische Kurven, bewegte Polygone)
  - Verhalten lässt sich (eingeschränkt) beeinflussen
- Neue Ideen
  - Hybrider Ansatz, um die Bewegung der Oberfläche zu verfolgen:
  - Partikel in Kombination mit einer impliziten Oberfläche (Level Set)

12

Inhalt

Motivation

Einführung

Physik
Computational Fluid Dynamics (CFD)

Computergrafik
Alternativen der Modellierung:
Lagrange(Partikel) oder Euler(Raumelemente)

Hybrider Ansatz (Foster, Fedkiw)
Kombination Lagrange, Euler

Fluiddynamik
Auszüge aus den Slides von Matthias Müller

Algorithmus, Foster & Fedkiw

Computeranimation

Zusammenfassung

Aussschnitt aus ‚Shrek'

13

Zwei kontinuierliche Grössen

Dichtefeld

Skalarfeld:

$\rho(\mathbf{x},t)$ [kg/m³]

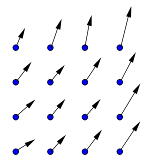
Inkompressible Fluids: $\rho(\mathbf{x},t) \equiv 1$

Geschwindigkeitsfeld

Vektorfeld:

$$\mathbf{v}(x,y,z,t) = \begin{bmatrix} u(x,y,z,t) \\ v(x,y,z,t) \\ w(x,y,z,t) \end{bmatrix}$$

[m/s]



14

Navier-Stokes Gleichungen

An jedem Punkt **x**:

Massenerhaltung

$$\underbrace{\frac{\partial \rho}{\partial t}}_{\text{Lokale Änderung der Dichte}} + \overbrace{\nabla \cdot (\rho \mathbf{v})}^{\text{Fluss aus dem Element hinaus}} = 0$$

Inkompressible Fluids

$$\nabla \cdot \mathbf{v} = 0$$

$$\rho(\mathbf{x},t) \equiv 1$$

Impulserhaltung

$$\underbrace{\rho \frac{D\mathbf{v}}{Dt}}_{\frac{m \cdot a}{Volumen}} = \underbrace{-\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}}_{\frac{Kraft}{Volumen}}$$

15

∇ Nabla Operator $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$

Vektorfeld

$$\mathbf{v} = \begin{pmatrix} u(x,y,z,t) \\ v(x,y,z,t) \\ w(x,y,z,t) \end{pmatrix}$$

Skalarfeld

$$p = (x,y,z,t)$$

Divergenz

$$\nabla \cdot \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

Gradient

$$\nabla p = \begin{pmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{pmatrix}$$

$$\nabla \mathbf{v} = \left(\frac{\partial \mathbf{v}}{\partial x}, \frac{\partial \mathbf{v}}{\partial y}, \frac{\partial \mathbf{v}}{\partial z}\right)$$

Laplace Operator:

$$\Delta p = \nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial y^2}$$

16

Navier-Stokes Gleichungen

Newton (pro Volumenelement): $\frac{m \cdot a}{Volumen} = \frac{Kraft}{Volumen}$

$$\rho \boxed{\frac{D\mathbf{v}}{Dt}} = -\nabla p + \rho\mathbf{g} + \mu\nabla^2\mathbf{v}$$

Beschleunigung: (verallgemeinerte Kettenregel)

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial}{\partial t}\mathbf{v}(x(t), y(t), z(t), t) = \frac{\partial \mathbf{v}}{\partial x}\cdot\frac{\partial x}{\partial t} + \frac{\partial \mathbf{v}}{\partial y}\cdot\frac{\partial y}{\partial t} + \frac{\partial \mathbf{v}}{\partial z}\cdot\frac{\partial z}{\partial t} + \frac{\partial \mathbf{v}}{\partial t}\cdot\frac{\partial t}{\partial t}$$
$$= \frac{\partial \mathbf{v}}{\partial x}\cdot u + \frac{\partial \mathbf{v}}{\partial y}\cdot v + \frac{\partial \mathbf{v}}{\partial z}\cdot w + \frac{\partial \mathbf{v}}{\partial t}$$
$$= \mathbf{v}\cdot\nabla\mathbf{v} + \frac{\partial v}{\partial t}$$

Navier-Stokes Gleichungen

Newton (pro Volumenelement): $\frac{m \cdot a}{Volumen} = \frac{Kraft}{Volumen}$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \boxed{\rho\mathbf{g}} + \mu\nabla^2\mathbf{v}$$

Externe Kräfte:

$$Gravitationskraft = \frac{Masse \cdot Beschleunigung}{Volumen}$$

Navier-Stokes Gleichungen

Newton (pro Volumenelement): $\frac{m \cdot a}{Volumen} = \frac{Kraft}{Volumen}$

$$\rho \frac{D\mathbf{v}}{Dt} = \boxed{-\nabla p} + \rho\mathbf{g} + \mu\nabla^2\mathbf{v}$$

Druck:

$$\nabla p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Für isotherme Fluids:

$$pV = const$$
$$p = \frac{k}{V} = k\rho$$

Navier-Stokes Gleichungen

Newton (pro Volumenelement): $\frac{m \cdot a}{Volumen} = \frac{Kraft}{Volumen}$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \rho\mathbf{g} + \boxed{\mu\nabla^2\mathbf{v}}$$

Vereinfachte Viskosität für inkompressible Fluids:

$$\mu\nabla^2\mathbf{v} = \mu\begin{bmatrix} u_{xx} + u_{yy} + u_{zz} \\ v_{xx} + v_{yy} + v_{zz} \\ w_{xx} + w_{yy} + w_{zz} \end{bmatrix}$$

μ (skalar): Viskosität des Fluids

Schliesslich…

$$\nabla \cdot \mathbf{v} = 0$$

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -k\nabla\rho + \rho\mathbf{g} + \mu\nabla^2\mathbf{v}$$

Umformen:

$$\nabla \cdot \mathbf{v} = 0$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\mathbf{v} \cdot \nabla \mathbf{v} - \frac{k}{\rho}\nabla\rho + \mathbf{g} + \frac{\mu}{\rho}\nabla^2\mathbf{v}$$

Inhalt

Motivation

Einführung

Fluiddynamik

Auszüge aus den Slides von Matthias Müller

Algorithmus, Foster & Fedkiw

Computeranimation

Zusammenfassung

Aussschnitt aus ‚Shrek'

Navier-Stokes Gleichungen

$$\nabla \cdot \mathbf{v} = 0$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\mathbf{v} \cdot \nabla \mathbf{v} - \frac{k}{\rho}\nabla\rho + \mathbf{g} + \frac{\mu}{\rho}\nabla^2\mathbf{v}$$

Algorithmus – Überblick

- Gleichungen werden über die Zeit gelöst
- Verhalten einer Flüssigkeit (Volumen) kann modelliert werden

6 Schritte:

I. Statische Umgebung als Voxel grid modellieren

II. Modellierung der Flüssigkeit mit einer Kombination aus Partikeln und einer impliziten Oberfläche

Für jeden Zeitschritt:

III. Geschwindigkeitsfeld updaten: Gleichung lösen

IV. Geschwindigkeiten anpassen aufgrund bewegter Objekte

V. Inkompressibilität (Massenerhaltung) gewährleisten $\nabla \cdot \mathbf{v} = 0$

VI. Mit dem neuen Geschwindigkeitsfeld die Position der Flüssigkeit updaten

Voxel grid

Die ‚Welt', in der sich die Flüssigkeit bewegt, wird in ein Grid von Voxeln mit Seitenlänge $\Delta\tau$ unterteilt

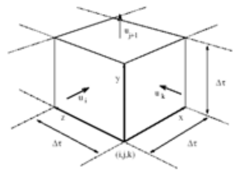
Nicht zwingend senkrecht, aber geringer Overhead bei ‚leeren' Zellen

Druckvariable im Zentrum jeder Zelle

Zusammen mit jeder Nachbarzelle ein Geschwindigkeitsvektor
- Im Zentrum jeder Seitenfläche definiert
- Repräsentiert die normale Komponente der Fliessgeschwindigkeit

Jede Zelle ist entweder leer oder vollständig gefüllt mit einem undurchlässigen, statischen Objekt

Trotz dieser groben Auflösung des Flüssigkeitsvolumens kann eine glatte Oberfläche realisiert werden



Modellierung der Flüssigkeit

- 4 Teile
  - Partikel
  - Isofläche
  - Dynamische Level Sets
  - Hybrides Oberflächenmodell

Flüssigkeitsverteilung im Grid repräsentiert durch implizite Oberfläche

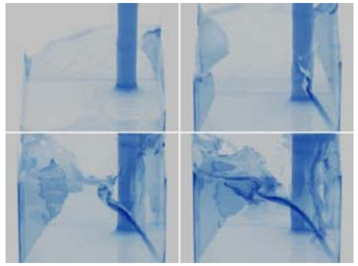
Implizite Funktion hergeleitet aus Kombination von Partikeln und dynamischer Isofläche

Modellierung der Flüssigkeit

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeits-feld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Isofläche liefert glatte Oberfläche

Partikel bieten Detail für Spritzer, Tropfen



N. Foster, R. Fedkiw. „Practical Animation of Liquids", *Proceedings of SIGGRAPH '01*

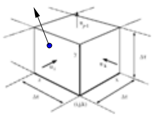
Partikel

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeits-feld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

- Im Grid plaziert (oder von einer Quelle zugefügt) gemäss einer initialen Flüssigkeitsverteilung
- Partikelgeschwindigkeit direkt aus dem Geschwindigkeitsgrid berechnet (trilineare Interpolation)
- Jedes Partikel wird nach folgender Gleichung bewegt:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_x$$

  - v_x: Fluid-Geschwindigkeit an der Stelle x_p
- Vorteile
  - Relativ geringer Berechnungsaufwand
  - Repräsentieren das sich ändernde Geschwindigkeitsfeld
- Nachteile
  - Fürs Rendering brauchen wir die Oberfläche
  - Alle Partikel zu Dreiecken verbinden
    - » Schwierig, die Konnektivität und die Oberflächendreiecke zu finden



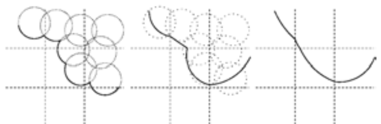
Isofläche

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeits-feld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

- Oberfläche der Flüssigkeit kann aus Isofläche einer impliziten Funktion gewonnen werden:
- Implizite Funktion
  - Implizite Funktion an der Stelle x_p mit Radius r:

$$\phi_p(\mathbf{x}) = \sqrt{(x_i - x_{pi})^2 + (x_j - x_{pj})^2 + (x_k - x_{pk})^2} - r$$

  - Jedes Partikel als Zentrum einer implizit definierten Kugeloberfläche wo $\phi_p(\mathbf{x}) = 0$
  - Implizite Funktion über alle Partikel, $\phi(\mathbf{x})$ jeweils der Wert des Partikels mit dem kleinsten Abstand zu x.



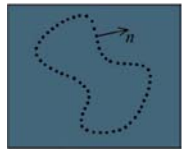
Isofläche

- Möglichkeit: polygonale Oberfläche mit Marching Cubes
- Ausgleichen, glätten um Ecken abzuschwächen

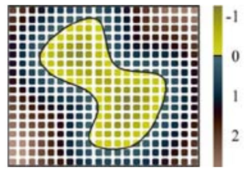
- Oberflächennormale: $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$
- Glatte Isofläche für jedes Frame: besseres Oberflächenmodell im Vergleich zu Partikeln
- Trotzdem Nachteile:
  - » Hohe Partikeldichte an der Oberfläche (bei $\phi(\mathbf{x}) = 0$) notwendig, damit die Oberfläche glatt aussieht
  - » Partikel werden auch im Rest des Volumens benötigt, obwohl sie keinen Einfluss auf das Aussehen der Oberfläche haben
- Lösungsansatz:
  - » Funktion ϕ einmal erstellen und dann die Bewegung verfolgen mit demselben Geschwindigkeitsfeld wie die Partikel
  - » **Temporally smoothed dynamic isosurface: A level set**

Level Sets anschaulich...

Zwei Gebiete, getrennt durch Grenzfläche (-linie)
Ziel: die Grenzfläche mit variabler Geschwindigkeit in Richtung n bewegen
Dreiecksnetze ungeeignet für grosse Veränderungen



Grenzfläche durch Partikel beschrieben
Keine Nachbarschaft definiert
Normale muss heuristisch gefunden werden
Jedes Partikel bewegt sich unabhängig in seine Normalenrichtung
Partikelkonzentration kann schwanken



Levelset Methode: implizite Beschreibung der Grenzfläche
Fläche bei Funktion = 0 „zero level"
Bewegung der Funktion beschrieben als PDE in Abh. der Zeit
Levelset Funktion im Raum definiert
Gesamplet auf regulärem Grid

C. Sigg. „Framework for Levelset Methods", *Diploma Thesis, ETH Zurich*

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Dynamische Level Sets

- ϕ direkt über die Zeit entwickeln mit dem Geschwindigkeitsfeld der Flüssigkeit
  - ϕ updaten gemäss Physik von Flüssigkeiten:

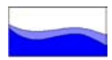
$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0$$

  - Verschiedene Techniken, um diese Gleichung zu lösen
  - Nachteile:
    - » Volumenverlust kann auftreten
    - » Tropfen, die sich vom Rest der Flüssigkeit lösen, sind zu klein, um vom Level Set aufgelöst zu werden
    - » Für Gas oder Rauch kein Problem, für Flüssigkeiten unbrauchbar!
  - Die Level Set Methode muss angepasst werden, um Volumenerhaltung zu gewährleisten

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Hybrides Oberflächenmodell

- Partikel: Lagrange
- Level Set: Euler
  - Komplementäre Stärken und Schwächen



    - Level Set: Volumenverlust
    - Partikel: Artefakte, wenn Anzahl Partikel zu gering

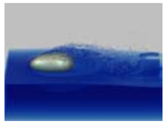
    + Level Set: Immer glatt
    + Partikel: genügend Detail unabhängig von Komplexität
  - Vorschlag: Kombination der beiden Ansätze
  - Ideen:
    - » Partikel innerhalb der Flüssigkeit und genügend weit von der Oberfläche entfernt: Partikel löschen
    - » Zellen nahe bei $\phi = 0$, mit zu wenigen Partikeln: Partikel erzeugen

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Hybrides Oberflächenmodell

- Kombination der beiden Ansätze
  - » Krümmung der Oberfläche berechnen
  - » Kleine Krümmung: glatte Oberfläche: Level Set Partikel ignorieren

  - » Grosse Krümmung: Spritzer, Tropfen: Partikel können ϕ lokal verändern



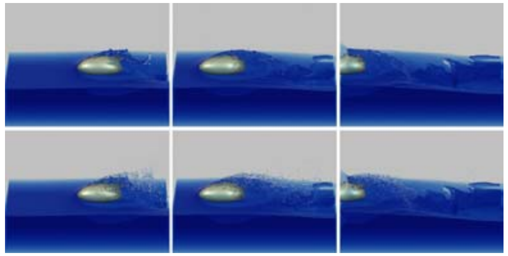
  - » Einzelne Partikel verlassen die Oberfläche
  - » Können direkt als kleine Tropfen gerendert werden

33

Hybrides Oberflächenmodell

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

- Partikel verlassen die Oberfläche



unten: Partikel als kleine Kugeln gerendert
150 x 75 x 90 Zellen, Berechnungszeit ca. 4 Minuten für ein Frame!

N. Foster, R. Fedkiw. „Practical Animation of Liquids", *Proceedings of SIGGRAPH '01*

34

Hybrides Oberflächenmodell

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

- Movie
  - Partikel als kleine Kugeln gerendert

N. Foster, R. Fedkiw. „Practical Animation of Liquids", *Proceedings of SIGGRAPH '01*

35

Hybrides Oberflächenmodell

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

- Movie
  - Partikel als kleine Kugeln gerendert

N. Foster, R. Fedkiw. „Practical Animation of Liquids", *Proceedings of SIGGRAPH '01*

36

Geschwindigkeitsfeld updaten

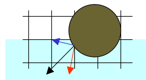
$$\mathbf{u}_t = \nu\nabla\cdot(\nabla\mathbf{u}) - (\mathbf{u}\cdot\nabla)\,\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{g}$$ Navier Stokes

- Auf Geschwindigkeitsfeld anwenden
  - Euler-Integrationsschritt
- 3 Schritte
  I. Δt berechnen CFL (Courant-Friedrichs-Levy)-Bedingung
     Δt muss kleiner sein als min. Zeit, in welcher "etwas signifikantes passiert"
     Hier: ein Partikel soll keine Zelle des Grids überspringen können,
     $$\Delta t < \Delta\tau / |\mathbf{u}|$$
  II. $(\mathbf{u}\cdot\nabla)\,\mathbf{u}$ 'integrieren' (semi-Lagrange-Methode)
  III. Restliche Terme 'integrieren' (Zentrale Differenzen)

→ Geschwindigkeitsfeld zum Zeitpunkt $t + \Delta t$
…leider ohne Massenerhaltung!

Bewegte Objekte

- Existierende Methoden erfordern ein sehr feines Grid
  - Für Animation nicht geeignet
- Wenn ein Objekt in eine Zelle mit Flüssigkeit eindringt:
  I. Die Flüssigkeit soll nicht ins Objekt hinein fliessen
     » Relative Geschwindigkeit zwischen Oberflächennormale und Flüssigkeitsgeschwindigkeit >= 0



  II. Kein Einfluss auf Komponente tangential zur Oberfläche des Objekts
  III. Weder Partikel noch Level Set Fläche dürfen die Oberfläche des Objekts durchdringen
     Partikel werden 'geschoben'
     Isofläche wird durch das Geschwindigkeitsfeld angepasst

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Massenerhaltung

- Nach Integration der Navier-Stokes Gleichung
  - Geschwindigkeitsfeld erfüllt die Bedingung $\nabla\cdot\mathbf{v} = 0$ nicht mehr (ist nicht divergenzfrei)
  - Einzige Möglichkeit:
    » Inkompressibilitätsbedingung für jede Grid-Zelle erfüllen
  - Lineares Gleichungssystem:
    » Lokale Druckänderung gekoppelt mit Divergenz jeder Zelle

$$\nabla^2 p = \frac{\rho\nabla\cdot\mathbf{u}}{\Delta t}$$

  - Diskretisieren, Lösen
  - Preconditioned Conjugate Gradient-Methode
  - Resultierendes Geschwindigkeitsfeld erhält die Masse (ist divergenzfrei) und erfüllt die Navier-Stokes Gleichungen

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Flüssigkeit updaten

Aktualisiertes Geschwindigkeitsfeld ist berechnet

Position des Flüssigkeitsvolumens updaten
(einen Zeitschritt weiter bewegen)

Partikel und implizite Oberfläche

Ende der Schleife.

I. Statische Umgebung als Voxel grid
II. Flüssigkeit aus Partikeln und impliziter Oberfläche
Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität (Massenerhaltung)
VI. Flüssigkeit updaten

Inhalt

Motivation

Einführung

Fluiddynamik

Auszüge aus den Slides von Matthias Müller

Algorithmus, Foster & Fedkiw

Computeranimation

Zusammenfassung

Ausschnitt aus ‚Shrek'

6 Schritte

I. Voxel grid
II. Partikel und implizite Oberfläche

Loop:
III. Geschwindigkeitsfeld updaten
IV. Bewegte Objekte
V. Inkompressibilität
VI. Flüssigkeit updaten

41

Computeranimation – Steuerung, Kontrolle

- Einfluss wünschenswert für Animationsanwendungen
  - Physikalisch basierte Animation
  - Steuerung vs. realistisches Verhalten
  - Gleichungen geben vor, dass sich alles bewegt, dreht und mischt solange die Kräfte nicht im Gleichgewicht sind
  - Nur beschränkten Einfluss auf den Ablauf
  - Natur der nichtlinearen Simulation
- Ansatz
  - Geschwindigkeiten statt Kräfte
  - Bspw. Unsichtbare Flächen oder Punkte
    - » Normalen und Geschwindigkeiten, steuern die Flüssigkeit
  - Oder parametrische Kurven in 3D
    - » Geschwindigkeit an jedem Punkt
    - » Einfluss durch Ändern der Position, Normalen oder Geschwindigkeit

PDI/Dreamworks, ‚Shrek', 3 min/frame

42

Inhalt

Motivation

Einführung

Fluiddynamik

Auszüge aus den Slides von Matthias Müller

Algorithmus, Foster & Fedkiw

Computeranimation

Zusammenfassung

Ausschnitt aus ‚Shrek'

Steuerung nur bedingt möglich

Natur der nichtlinearen Simulation

Kontrolle vs. realistisches Verhalten

bspw. parametrische Kurven in 3D

43

Zusammenfassung – Practical Animation of Liquids

N. Foster, R. Fedkiw, *Proceedings of SIGGRAPH '01*

- Physikalisch basierte Animation – Flüssigkeiten
- Computational Fluid Dynamics, Navier Stokes Gleichungen
- Computeranimation
  - Kombination aus Partikeln und impliziter Oberfläche, Level Sets
  - Interaktion mit bewegten Objekten
- Wie geht es weiter...
  - Animation and Rendering of Complex Water Surfaces

Enright, Marschner, Fedkiw, *SIGGRAPH '02*

44

So viel Geduld soll belohnt werden...

45