

Einfacher SPH Flüssigkeitssimulator mit Beschleunigter Nachbarschafts Suche

Thierry Meiers

Bachelor Projekt

Albert-Ludwigs-Universität Freiburg
Technische Fakultät
Graphische Datenverarbeitung

9. August 2024

Inhaltsverzeichnis

1	Einführung	1
2	Navier-Stokes-Gleichung	1
2.1	Impulsgleichung	1
2.2	Kontinuitätsgleichung	2
2.3	Nicht komprimierbare Fluide	3
3	Smoothed Particle Hydrodynamics	3
3.1	Grundlagen	3
3.2	Kontinuitätsgleichung	4
3.3	Impulsgleichung	4
3.4	Berechnung der Partikeleigenschaften	5
4	Simulationsschritt	5
5	Nachbarschaft Suche	6
5.1	Einfügen von Partikel	6
5.2	Updaten der Partikel Positionen	7
5.3	Ermitteln von Nachbarn	7
6	Implementierung	7
6.1	Tests	7
6.2	TODO	7
7	Analysen	7
8	Schlussfolgerung	7
9	Bibliographie	7

1 Einführung

Fluidsimulationen haben sich zu einem unverzichtbaren Werkzeug in zahlreichen Bereichen unserer Gesellschaft entwickelt. Sie ermöglichen die präzise Simulation von Flüssigkeiten und Gasen in verschiedenen Umgebungen, wodurch das Verhalten dieser Stoffe und ihre Interaktionen mit der Umgebung untersucht werden können. Aufgrund ihrer Vielseitigkeit kommen Fluidsimulationen in einer großen Bandbreite von Anwendungsgebieten zum Einsatz, darunter die Optimierung von Fahrzeugen und Gebäuden, Wettervorhersagen, medizinische Forschung, die Erzeugung realistischer Effekte in Filmen und Spielen, industrielle Prozesse, Energieerzeugung und die Entwicklung von Sportgeräten. Diese Simulationen tragen wesentlich dazu bei, Designs zu verbessern, die Effizienz zu steigern und komplexe Strömungen besser zu verstehen.

Die Allgegenwärtigkeit und Bedeutung dieser Technologie in verschiedenen Branchen haben mein Interesse an diesem Thema geweckt. Insbesondere die Verbindung von Physik und Informatik ist eine Motivation zur Arbeit an diesem Projekt. Das Hauptziel ist die Implementierung der Smoothed Particle Hydrodynamics (SPH) zur Simulation von Fluiden, um ein tieferes Verständnis für die Komponenten und Mechanismen von SPH zu entwickeln. Analysen werden dabei helfen, die Parameter von SPH richtig zu setzen um die Stabilität der Simulation zu optimieren und Fehler in der Implementierung zu finden.

In diesem Bericht werden wir zuerst in Abschnitt 2 die Navier-Stokes-Gleichung analysieren, die die Grundlage aller Methoden zur Simulation von Fluiden und Gasen bilden. In Abschnitt 3 folgt eine Übersicht aller Teile der SPH-Methode, die zur Implementierung der Simulation verwendet wurden. Wie SPH schlussendlich implementiert wurde, wird in Abschnitt 4 beschrieben, zusammen mit einer Übersicht der einzelnen Komponenten. Der wichtigste Teil wird die Analyse und Interpretation der Daten der Simulation in Abschnitt 5 sein.

2 Navier-Stokes-Gleichung

Die Navier-Stokes-Gleichungen sind fundamentale Gleichungen in der Fluidmechanik, die das Verhalten von strömenden Flüssigkeiten und Gasen beschreiben und bildet ein System von nichtlinearen partiellen Differentialgleichungen zweiter Ordnung. Sie basieren auf den Prinzipien der Erhaltung von Masse, Impuls und Energie. Sie modellieren die Bewegung des Fluids, indem sie die Einflüsse von Druck, Viskosität und externen Kräften auf das Strömungsverhalten berücksichtigen.

2.1 Impulsgleichung

$$\rho(\nabla \vec{v} + (\vec{v} \cdot \nabla) \vec{v}) = -\vec{\nabla} p + \eta \vec{\nabla}^2 \vec{v} + (\lambda + \eta) \vec{\nabla}(\vec{\nabla} \cdot \vec{v}) + \rho \vec{f} \quad (1)$$

Gleichung (1) beschreibt die Impulserhaltung für komprimierbare Fluide. Diese entspricht dem zweiten Newtonsches Gesetz und kann dementsprechend hergeleitet

werden:

$$\begin{aligned}\vec{F} &= \Delta m \cdot a = \Delta m \cdot \nabla \vec{v} \\ \vec{F} &= \rho \cdot \Delta V \cdot \nabla \vec{v} \\ \vec{F} &= \rho \cdot \Delta V (\nabla \vec{v} + (\vec{v} \cdot \vec{\nabla}) \vec{v})\end{aligned}\tag{2}$$

Die Masse eines Objektes kann durch $m = \rho \cdot V$ bestimmt werden. Die Geschwindigkeitsänderung ergibt sich aus der Lokalen zeitlichen Beschleunigung $\nabla \vec{v}$ und der Konvektiven Beschleunigung $(\vec{v} \cdot \vec{\nabla}) \vec{v}$, also die die Änderung der Geschwindigkeit aufgrund der Bewegung der Flüssigkeit selbst.

Wir benötigen nun die Kräfte \vec{F} . Diese sind aus verschiedenen Kräfte welche auf einen Fluid auswirken. Diese bestehen aus der Druck Kraft (3), der Reibungs Kraft (4), der Kompressions Kraft (5) und der Kräfte die von außen wirken (6).

$$\vec{F}_{Druck} = -\vec{\nabla} p \cdot \Delta V \tag{3}$$

$$\vec{F}_{Reib} = \eta \vec{\nabla}^2 \vec{v} \cdot \Delta V \tag{4}$$

$$\vec{F}_{Kompress} = (\lambda + \eta) \vec{\nabla} (\vec{\nabla} \cdot \vec{v}) \cdot \Delta V \tag{5}$$

$$\vec{F}_{ext} = \rho \vec{f} \cdot \Delta V \tag{6}$$

$$\vec{F} = \vec{F}_{Druck} + \vec{F}_{Reib} + \vec{F}_{Kompress} + \vec{F}_{ext} \tag{7}$$

Wir setzen (7) in die Gleichung (2) ein und erhalten somit:

$$-\vec{\nabla} p \cdot \Delta V + \eta \vec{\nabla}^2 \vec{v} \cdot \Delta V + (\lambda + \eta) \vec{\nabla} (\vec{\nabla} \cdot \vec{v}) \cdot \Delta V + \rho \vec{f} \cdot \Delta V = \rho \cdot \Delta V (\nabla \vec{v} + (\vec{v} \cdot \vec{\nabla}) \vec{v})$$

Durch das dividieren beider Seiten mit ΔV erhält man schließlich die Gleichung (1).

$$-\vec{\nabla} p + \eta \vec{\nabla}^2 \vec{v} + (\lambda + \eta) \vec{\nabla} (\vec{\nabla} \cdot \vec{v}) + \rho \vec{f} = \rho (\nabla \vec{v} + (\vec{v} \cdot \vec{\nabla}) \vec{v})$$

2.2 Kontinuitätsgleichung

Des weiterem folgt die Kontinuitätsgleichung (8) welche die Erhaltung der Masse innerhalb eines Fluides beschreibt.

$$\Delta \rho + \vec{\nabla} \cdot (\rho \vec{v}) = 0 \tag{8}$$

In ihrer klassischen Form sind sowohl die Impulsgleichung (1) als auch die Kontinuitätsgleichung (8) spezifisch für Newtonsche Fluide formuliert. Newtonsche Fluide zeichnen sich durch ein lineares viskoses Fließverhalten aus, bei dem die Viskosität unabhängig von der Schergeschwindigkeit ist.

2.3 Nicht komprimierbare Fluide

Für nicht komprimierbare Fluide vereinfachen sich die Impulsgleichung (1) und die Kontinuitätsgleichung (8) zu

$$\rho(\nabla \vec{v} + (\vec{v} \cdot \vec{\nabla})\vec{v}) = -\vec{\nabla} p + \eta \vec{\nabla}^2 \vec{v} + \rho \vec{f} \quad (9)$$

da $(\lambda + \eta)\vec{\nabla}(\vec{\nabla} \cdot \vec{v}) = 0$ für nicht komprimierbare Fluide, und

$$\vec{\nabla} \vec{v} = 0 \quad (10)$$

da $\Delta \rho = 0$ und $\vec{\nabla} \rho = 0$

3 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) ist eine numerische Methode zur Approximation der Navier Stokes Gleichungen. Im Verlauf des Kapitels werden die einzelnen Bestandteile von SPH erläutert und die Verbindung von SPH und den Navier Stokes Gleichungen beleuchtet.

3.1 Grundlagen

Das Konzept von SPH lässt sich allgemein als eine Methode zur Diskretisierung von räumlichen Feldgrößen und räumlichen Differentialoperatoren verstehen. Die zentrale Idee von SPH ist es, Partikel einzusetzen, die Proben von Feldgrößen enthalten, und eine Kernel-Funktion zu nutzen, um kontinuierliche Felder zu approximieren. Ein Partikel besitzt Eigenschaften wie die Masse m_i , die Position x_i , die Geschwindigkeit v_i und den Druck p_i .

Die Kernel-Funktion $W_{ij} = W(x_i - x_j, h)$ ist eine Funktion zur Glättung der Räumlichen Diskretisierung wobei h als die Glättungslänge des Kernels bezeichnet wird. Diese steuert, wie stark der Wert an Position x_j Einfluss auf den Wert an Position x_i nimmt. Dabei soll die Kernel-Funktion zweifach Ableitbar sein und folgende Eigenschaften besitzen: $\forall x_i, x_j \in \mathbb{R}^d, h \in \mathbb{R}^+ \rightarrow$

$$\int_{\mathbb{R}^d} W(x_i - x'_j, h) dx'_j = 1$$

$$\lim_{h' \rightarrow 0} W(x_i - x_j, h') = \delta(r)$$

$$W(x_i - x_j, h) \geq 0$$

$$W(x_i - x_j, h) = W(-(x_i - x_j), h)$$

$$W(x_i - x_j, h) = 0 : |x_i - x_j| \geq h$$

wobei $\delta(x)$ die Delta Verteilung ist. In diesem Projekt wird die Cubic Spline Kernel Funktion verwendet, wie in [Monaghan1992] beschrieben:

$$W(q) = \alpha \begin{cases} (2-q)^3 - 4(1-q)^3 & 0 \leq q \leq 1 \\ (2-q)^3 & 1 \leq q \leq 2 \\ 0 & q \geq 2 \end{cases}$$

$$q = \frac{\|x_j - x_i\|}{h}$$

Dabei ist α ein Normalisierungsfaktor, der in verschiedenen Dimensionen wie folgt definiert ist:

- 1D: $\alpha = \frac{1}{6h}$
- 2D: $\alpha = \frac{5}{14\pi h^2}$
- 3D: $\alpha = \frac{1}{4\pi h^3}$

3.2 Kontinuitätsgleichung

Die Kontinuitätsgleichung beschreibt die Erhaltung der Masse und wird in der SPH-Methode wie folgt diskretisiert:

$$\frac{d\rho_i}{dt} = \sum_j m_j (v_i - v_j) \cdot \nabla_i W_{ij}$$

Dabei ist ρ_i die Dichte des Partikels i , m_j die Masse des Partikels j , v_i und v_j die Geschwindigkeiten der Partikel i und j , und $\nabla_i W_{ij}$ das Gradientenfeld der Kernel-Funktion bezüglich des Partikels i .

3.3 Impulsgleichung

Die Impulsgleichung beschreibt die Erhaltung des Impulses und wird in der SPH-Methode wie folgt diskretisiert:

$$\frac{dv_i}{dt} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} + 2\nu \sum_j \frac{m_j}{\rho_j} \frac{(v_i - v_j) \times (x_i - x_j)}{(x_i - x_j) \times (x_i - x_j) + 0.01 \cdot h^2} \nabla_i W_{ij} + f_i$$

3.3.1 Druckbeschleunigung

Die Beschleunigung aufgrund des Drucks wird durch die folgende Gleichung beschrieben:

$$a_{druck} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} \quad (11)$$

Dabei sind p_i und p_j die Drücke der Partikel i und j , ρ_i und ρ_j die Dichten der Partikel i und j , und $\nabla_i W_{ij}$ der Gradient der Kernel-Funktion bezüglich des Partikels i .

3.3.2 Viskositätsbeschleunigung

Die Beschleunigung aufgrund der Viskosität wird durch die folgende Gleichung beschrieben:

$$a_{viskosität} = 2\nu \sum_j \frac{m_j}{\rho_j} \frac{(v_i - v_j) \cdot (x_i - x_j)}{(x_i - x_j) \cdot (x_i - x_j) + 0.01 \cdot h^2} \nabla_i W_{ij} \quad (12)$$

Hier ist ν die kinematische Viskosität, v_i und v_j sind die Geschwindigkeiten der Partikel i und j , x_i und x_j sind die Positionen der Partikel i und j , und $\nabla_i W_{ij}$ ist der Gradient der Kernel-Funktion bezüglich des Partikels i .

3.3.3 Externe Kräfte

Die externen Kräfte, die auf ein Partikel wirken, werden durch f_i dargestellt. Dies kann beispielsweise die Gravitation oder andere externe Einflüsse umfassen:

$$a_{\text{extern}} = f_i \quad (13)$$

3.4 Berechnung der Partikeleigenschaften

3.4.1 Dichteberechnung

Die Dichte jedes Partikels i wird durch die Summe der Massen der benachbarten Partikel j gewichtet durch die Kernel-Funktion berechnet:

$$\rho_i = \sum_j m_j W_{ij} \quad (14)$$

3.4.2 Druckberechnung

Der Druck p_i wird häufig durch eine Zustandsgleichung berechnet, z.B.:

$$p_i = k \left(\frac{\rho_i}{\rho_0} - 1 \right) \quad (15)$$

wobei k die Steifigkeit Konstante ist und ρ_0 die Ruhe-Dichte.

4 Simulationsschritt

Der Simulationsschritt ist in der SPH Simulation von Zentraler Bedeutung. Dort werden alle nötigen Beschleunigungen ermittelt, die zum simulieren eines Fluides mithilfe von Partikeln, nötigen sind. In diesem Projekt wurde die Methode in 3 Schleifen unterteilt wobei die erste als einzige alle Partikel berücksichtigt. In den Restlichen Schleifen werden nur Partikel welche das Fluid bilden beachtet und Partikel zum Simulieren der Grenze Ignoriert.

Eigenschaften wie die Partikelgröße und die Ruhedichte der Flüssigkeit werden vor dem Starten der Simulation festgelegt. Aus diesen wird eine Masse für die Partikel ermittelt welche im laufe der Simulation konstant bleibt. Weitere Werte werden im Kapitel TODO erläutert.

Darauf hin werden in der ersten Schleife die restlichen locale Eigenschaften ermittelt. Dazu müssen die Nachbar Partikel ermittelt werden. Mit Hilfe dieser wird die Lokale Dichte und der Lokale Druck bestimmt. Diese werden für die Berechnungen der nächste Schleife benötigt. In dieser werden, für alle Partikel die das Fluid bilden, die Viskosität- und Druckbeschleunigungen berechnet. Anhand dieser Beschleunigungen werden die Positionen der Partikel in der letzten schleife Aktualisiert.

Algorithm 1 Simulationsschritt

1: **For each** particle **in** particles

2: FindNeighbors()

3: ComputeLocalDensity() (14)

4: ComputeLocalPressure() (15)

5: **For each** particle **in** noBoundaryParticles

6: GetViscosityAcceleration() (12)

7: GetPressureAcceleration() (11)

8: **For each** particle **in** noBoundaryparticles

9: acceleration = viscosityAcceleration + pressureAcceleration + gravitation

10: particle.Velocity+ = timeSteps · acceleration

11: particle.Position+ = timeSteps · particle.Velocity

5 Nachbarschaft Suche

Die Nachbarschaft Suche ist das Fundament der Simulation. Diese erlaubt das ermitteln der Nachbar Partikel in einem feste Radius um eine Position zum Berechnen lokaler Werte. Ein naiver Ansatz wäre, für alle Partikel den Abstand zur Position zu ermitteln und zu prüfen ob sich diese im Radius zu befinden. Diese Methode weist jedoch eine quadratische Laufzeitkomplexität auf, da für jedes Partikel der Abstand zu allen anderen Partikeln berechnet werden muss. Dies führt zu einer Laufzeit von $O(n^2)$, wobei n die Anzahl der Partikel ist. Bei großen Partikelzahlen wird dieser Ansatz daher ineffizient und rechenintensiv.

Für das Projekt wurde dementsprechend das Räumliche Hashing verwendet. Dabei handelt es sich um eine Datenstruktur die den Raum in einzelne Blöcke unterteilt. Beim ermitteln von Nachbar Partikel müssen dadurch nicht mehr alle Partikel geprüft werden sondern nur diese die sich in dem selbem Block oder Nachbar Block befinden. Dies führt zu einer effizienteren und weniger rechenintensiven Methode zum bestimmen der Nachbarn.

5.1 Einfügen von Partikel


Um ein Partikel in die Datenstruktur einzufügen wird als erst ein Hashwert bezüglich der Position errechnet. Dazu wird der Position Vektor durch die Blockgröße geteilt und das Ergebnis abgerundet.

Algorithm 2 Hash Funktion

Require: Vector $\mathbf{v} = (v_x, v_y)$, scalar CellSize

Ensure: Hashed vector $\mathbf{h} = (h_x, h_y)$

1: $\mathbf{h} \leftarrow \frac{\mathbf{v}}{\text{CellSize}}$

2: $\mathbf{h} \leftarrow \lfloor \mathbf{h} \rfloor$ 

Schließlich wird der Partikel in eine Liste innerhalb der Hashtabelle Eingefügt. Das löschen eines Partikels aus der Datenstruktur funktioniert Analog.

5.2 Updaten der Partikel Positionen

Wurden die Positionen der Partikel im Simulationsschritt aktualisiert müssen ebenfalls die Einträge in der Hashtabelle Aktualisiert werden. Dazu werden die Partikel vor dem Aktualisieren der Position aus der Datenstruktur entfernt und nach der Aktualisierung wieder eingefügt.


5.3 Ermitteln von Nachbarn

Das ermitteln von Nachbar Partikel ist das Fundament des Simulationsschrittes. Zu erst werden die Blöcke, welche Teil des Radiuses sind und somit durchsucht werden müssen, ermittelt. Des weiterem werden die Partikel in den Blöcke überprüft ob sich diese in dem Radius um die Position befinden.

Algorithm 3 Partikel im Radius

Require: Position \mathbf{p} , Radius r

Ensure: Liste der Partikel im Radius: `partikelImRadius`

```
1:  $startX \leftarrow \lfloor (\mathbf{p}_x - r) / CellSize \rfloor$ 
2:  $endX \leftarrow \lceil (\mathbf{p}_x + r) / CellSize \rceil$ 
3:  $startY \leftarrow \lfloor (\mathbf{p}_y - r) / CellSize \rfloor$ 
4:  $endY \leftarrow \lceil (\mathbf{p}_y + r) / CellSize \rceil$ 
5: for ( $x \leftarrow startX$  to  $endX$ ) do
6:   for ( $y \leftarrow startY$  to  $endY$ ) do
7:     if  $((x, y)$  existiert in Hashtabelle) then
8:       for all (Partikel in Hashtabelle[hash]) do
9:         if  $(Distance(Partikel.Position, \mathbf{p}) \leq r)$  then
10:          partikelImRadius.add( $q$ )
10:  partikelImRadius
```

Dies erlaubt es die Anzahl an Partikel welche überprüft werden müssen zu beschränken und die Laufzeit zu verringern.

6 Implementierung

Die Implementierung ist zentraler Bestandteil des Projektes. Implementiert wurde in C# mit Hilfe des Frameworks Monogame. In diesem Kapitel wird die Implementierung von Tests, essenziell für die Korrektheit der Simulation, und die Implementierung Spezieller Hilfsmittel vorgestellt.

6.1 Tests

6.2 TODO

7 Analysen

8 Schlussfolgerung

9 Bibliographie