

Universität Karlsruhe (TH)
Fakultät für Informatik
Institut für Betriebs- und Dialogsysteme
Prof. Dr. A. Schmitt

Fotorealistische Wasseranimation bei Brunnenanlagen

Matthias Baas, Michael Fautz, Alfred Schmitt

baas@ira.uka.de, fautz@ira.uka.de, aschmitt@ira.uka.de

Interner Bericht 2000-13

ISSN 1432 - 7864

Fotorealistische Wasseranimation bei Brunnenanlagen

Kurzfassung

Ziel dieser Arbeit ist es, ein Simulationsverfahren zu entwickeln, mit dem Brunnenanlagen in der Computergrafik animiert werden können. Dazu werden existierende Verfahren zur Flüssigkeitssimulation auf ihre Tauglichkeit für diese Aufgabe untersucht und, wo notwendig, auch Modifikationen an den ursprünglichen Methoden vorgenommen.

Da die in einem Brunnen auftretenden Effekte zu vielseitig sind, um von einem einzelnen Verfahren effizient gelöst werden zu können, wird in dieser Arbeit ein Verfahren entwickelt, mit dem sich eine Wassersimulation nach dem „Baukastenprinzip“ aus mehreren spezialisierten Methoden zusammensetzen lässt. Auf diese Weise können die Stärken der einzelnen Verfahren ausgespielt und deren Schwächen durch andere Verfahren überdeckt werden.

Einer dieser Bausteine ist ein spezielles Oberflächenpartikelsystem, welches Fließbewegungen auf einer Oberfläche nachbildet, wobei Eigenschaften wie Adhäsion und Rauheit der Oberfläche berücksichtigt werden. Am Beispiel eines Brunnens, der aus keilförmig eingeschnittenen Würfelementen zusammengesetzt ist, wird die Leistungsfähigkeit des Verfahrens demonstriert.

Photo-Realistic Animation of Water for Fountain Installations

Abstract

The aim of this work is the development of a simulation method for the animation of fountain arrangements with computer graphics methods. For that purpose existing methods for the simulation of liquids are investigated to see if they are suitable for this task and are modified where necessary.

Since the effects appearing on a fountain are too manifold to be solved efficiently by a single simulation method, a method is developed in this report, which compounds a whole water simulation using a module system of specialized simulation techniques. By this means the potency of a particular technique can be exploited and its drawbacks can be compensated by other methods.

One of this modules is a specialized surface-particle-system, which can imitate the flow of fluids on a surface. Surface properties like adhesion and roughness are taken into account. The capacity of this method is demonstrated by a simulation of a fountain assembled from wedge shaped cubes.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Literaturüberblick	1
2	Bekannte Verfahren	3
2.1	Wellengleichung	3
2.2	Methode nach Foster/Metaxas	4
2.3	Smoothed Particle Hydrodynamics (SPH)	5
2.4	Fazit	6
3	Oberflächenpartikelsystem	7
3.1	Aktualisierung der Geschwindigkeiten	8
3.2	Bewegung der Partikel	10
3.3	Zusätzliche Effekte	13
4	Kombination von Methoden	16
4.1	Oberfläche – 3D-Partikelsystem	17
4.2	3D-Partikelsystem – Oberfläche	18
4.3	Partikelsystem – Wellengleichung	19
5	Experimentelle Ergebnisse	20
	Literaturverzeichnis	22

1 Einleitung

Die realitätsnahe Nachbildung natürlicher Phänomene wie Wolken, Dampf, Nebel, Feuer oder Wasser hat schon eine Vielzahl von Computergrafikern zu regen Forschungen angeregt. Im Bereich der Flüssigkeitssimulation existiert bereits eine Reihe von Veröffentlichungen, die alle eine realistische Bewegung der Flüssigkeit zum Ziel haben. Dabei entstanden die unterschiedlichsten Verfahren, da alle die Simulation unter einem anderen Aspekt betrachteten und mit anderen Anforderungen begannen. Daran kann man schon erkennen, dass es nicht eine einzige optimale Methode gibt, um Flüssigkeit zu simulieren. Flüssigkeit kann in unterschiedlichsten Ausprägungen auftreten, wie z.B. einzelne Tropfen, spritzende Fontänen, Rinnsale, Bäche, Flüsse, Seen, Ozeane. Je spezieller man eine Methode gestaltet, desto effizienter lässt sich eine vorgegebene Aufgabe lösen.

Dieser Artikel beschäftigt sich mit der Flüssigkeitssimulation bei Brunnenanlagen. Solche Anlagen bestehen in der Regel aus mehreren Komponenten wie Fontänen, Bereiche, in denen ein Wasserfilm auf einer Oberfläche fließt und einem Auffangbecken. In dem hier vorgestellten Simulationsverfahren wird deshalb eine Kombination aus mehreren Simulationsmethoden verwendet: ein Partikelsystem zur Modellierung von Spritzwasser, ein spezielles Oberflächenpartikelsystem zur Simulation von Adhäsion und Fließbewegungen auf Oberflächen und ein Höhenfeld zur Repräsentation eines gefüllten Wasserbeckens, welches mittels der Wellengleichung moduliert wird.

Für das angestrebte Ziel, die Bewegung von Wasser in einem Brunnen nachzuempfinden, so dass man einen, aus graphischer Sicht realistischen optischen Eindruck erhält, liefert diese Methode gute Ergebnisse. Aufgrund der fehlenden physikalischen Korrektheit des Simulationsmodells ist das Verfahren allerdings ungeeignet, um eine zu errichtende Brunnenanlage exakt zu simulieren. Die Evaluation von Design-Änderungen am Brunnen, um ein gewünschtes Fließverhalten zu erreichen, ist somit nicht Sinn und Zweck der Simulation.

Die folgenden Abschnitte geben zunächst einen Überblick über Forschungsarbeiten zum Thema Flüssigkeitssimulation sowie eine Einführung in Simulationsverfahren, die für die Brunnensimulation besonders geeignet scheinen. Anschließend folgt eine Beschreibung des neu entwickelten Oberflächenpartikelsystems und der in dieser Arbeit entwickelten Methodik der Verknüpfung unterschiedlicher Simulationsmethoden in einer Gesamtsimulation. Am Beispiel eines Keilwürfelbrunnens wird die Leistungsfähigkeit der vorgestellten Methoden verdeutlicht.

1.1 Literaturüberblick

In der Vergangenheit wurden bereits einige Verfahren entwickelt, um Flüssigkeiten, jeweils in bestimmten unterschiedlichen Ausprägungen, für die Computergrafik zu simulieren.

In [Fournier & Reeves '86] und [Peachey '86] werden Ozeanwellen durch parametrisierte Oberflächen angenähert. Die Oberflächen sind zeitabhängig, so dass sich die

Wellen auch animieren lassen.

In [Kass & Miller '90] können Oberflächenwellen durch Lösen der Wellengleichung erzeugt werden. Das Verfahren benutzt eine Höhenfeld-Repräsentation, wodurch solche Effekte wie spritzendes Wasser nicht mehr darstellbar sind. Dafür hat das Verfahren den Vorteil, dass es mit wenig Speicher und wenig Rechenkapazität auskommt. Für leichte Wellen auf einer Wasseroberfläche liefert die Methode sehr realistische Ergebnisse.

Einen ähnlichen Ansatz wird in [O'Brien & Hodgins '95] (im Internet auch unter [Hodgins '99]) verfolgt, zusätzlich wird hier aber noch ein Partikelsystem integriert, um spritzendes Wasser simulieren zu können.

In [Miller & Pearce '89] wird die Flüssigkeit vollständig als Partikelsystem repräsentiert, bei dem die einzelnen Partikel gegenseitig Kräfte aufeinander ausüben. Damit kann ein viskoses Verhalten der simulierten Flüssigkeit erreicht werden.

Die aus der Hydrodynamik bekannten Bewegungsgleichungen für Flüssigkeiten (die sogenannten Navier-Stokes-Gleichungen) werden in [Chen & Lobo '95] angewendet. Um Rechenzeit zu sparen, werden sie allerdings nur in 2D gelöst und die Flüssigkeit wird als Höhenfeld dargestellt, wobei der Druck die Höhe bestimmt.

In [Foster & Metaxas '96] (im Internet auch unter [Foster '99]) werden die Navier-Stokes-Gleichungen innerhalb eines dreidimensionalen Volumens gelöst. Mit diesem Verfahren lassen sich neben den Wellen nun auch Wirbel und spritzendes Wasser darstellen. Allerdings erkaufte man sich dies mit einer hohen Rechenzeit und hohem Speicherbedarf. Die Methode aus diesem Artikel wurde in dieser Arbeit etwas genauer untersucht (siehe Abschnitt 2.2), da sie für eine Brunnensimulation wegen ihrer Allgemeinheit recht vielversprechend klang.

Ein Ansatz, bei dem nur sehr wenig Flüssigkeit simuliert wird (nämlich einzelne Tropfen) wird in [Fournier et al. '98] vorgestellt. Die Tropfen können sich über eine Oberfläche bewegen, die durch Dreiecke approximiert wird, sie können miteinander kollidieren, wobei sie sich vereinigen und sie können auf der Oberfläche Streifen hinterlassen. Die Grundidee dieser Methode wird auch in dieser Arbeit benutzt (siehe Abschnitt 3).

Eine in der Computergrafik recht neue Methode findet in [Desbrun & Gascuel '96] Verwendung. Wie in [Miller & Pearce '89] wird auch hier die Flüssigkeit als Partikelsystem repräsentiert, bei dem die Partikel gegenseitig Kräfte aufeinander ausüben. Allerdings sind diese Kräfte nun besser an die tatsächlichen physikalischen Gegebenheiten angepasst (beispielsweise wird der Druck innerhalb der Flüssigkeit berücksichtigt). Die grundlegenden Prinzipien dieser Methode haben ihren Ursprung in der Astrophysik. Abschnitt 2.3 geht näher auf die prinzipielle Methode ein.

Bei der letztjährigen Siggraph-Konferenz gab es gleich drei Beiträge zum Thema Flüssigkeitssimulation. In [Weimer & Warren '99] wird die Simulation anhand eines Unterteilungsalgorithmus für Vektorfelder gemacht. Vom Benutzer wird dazu ein grobes Geschwindigkeitsfeld vorgegeben und der Algorithmus generiert durch Unterteilung ein beliebig feines Geschwindigkeitsfeld. Das unterteilte Vektorfeld nähert dabei

die Lösung einer linearisierten Form der Navier-Stokes-Gleichung an. Das bedeutet, das Verfahren kann für Flüssigkeiten ohne Viskosität oder für langsame Strömungen verwendet werden.

Die Methode in [Stam '99] löst ebenfalls die Navier-Stokes-Gleichung, hier sogar ohne Einschränkungen und in Echtzeit. Die Methode benutzt eine Langrangesche Beschreibung und eine implizite Lösungsmethode, wodurch größere Zeitschritte als bei expliziten Methoden möglich sind und das Verfahren somit schneller wird. Die Autoren wenden das Verfahren allerdings nur bei Bewegungen an, die denen von Gas ähnlich sind. Soll das Verfahren auch für Simulationen angewendet werden, bei denen freie Oberflächen enthalten sind (wie z. B. bei Wasser), so muss zuerst noch weitere Forschungsarbeit geleistet werden.

Eine Methode, die zur Unterstützung von Zeichentrickfilmen entwickelt wurde, ist in [Witting '99] beschrieben. Dabei handelt es sich um eine zweidimensionale Methode, bei der erneut die Navier-Stokes-Gleichung eingesetzt wird. Hier allerdings mit dem Unterschied, dass die Flüssigkeit kompressibel ist. Ziel ist es, eine Bildsequenz zu erzeugen, in denen der Eindruck von Rauch oder fließendem Wasser entsteht. Beispielsweise kann dazu ein Bild vorgegeben werden, in dem unterschiedliche Helligkeitswerte als unterschiedliche Temperatur interpretiert wird. Aufgrund der Temperatur- und Dichteunterschiede im dem Bild kommt es zur Bewegung und Verwirbelung des Bildes. Diese Methode wurde für den Zeichentrickfilm *Der Prinz von Ägypten* [DreamWorks '98] in über 20 Szenen benutzt.

Die letzten drei Verfahren hatten allerdings aufgrund ihrer Aktualität keinen Einfluss mehr auf diese Arbeit.

2 Bekannte Verfahren

2.1 Wellengleichung

Eine Methode, die im Prinzip eine Lösung der Wellengleichung ist, wird in [Kass & Miller '90] vorgestellt. Die Wasseroberfläche wird hierbei als sogenanntes *Höhenfeld* repräsentiert (siehe Abbildung 1). Solch ein Höhenfeld ist ein zweidimensionales Gitter, an dessen Gitterpunkte die Auslenkung der Wasseroberfläche aus der Ruhelage gespeichert wird.

Bei der Methode muss lediglich pro Schritt für jede Spalte und jede Zeile des Höhenfeldes ein tridiagonales Gleichungssystem gelöst werden. Da die Flüssigkeit nur als zweidimensionales Gebilde gegeben ist, ist das Verfahren sehr schnell (bei passender Auflösung auch in Echtzeit durchführbar). Das Verfahren ist im Prinzip spezialisiert auf Wasseroberflächen, die relativ sanfte Wellen bilden. Durch die Repräsentation als Höhenfeld ist es nicht möglich, spritzendes Wasser oder sich überschlagende Wellen zu simulieren. Die Ausbildung von Wirbel ist ebenfalls nicht möglich, da man sich für die Wellengleichung als zugrundeliegendes physikalisches Modell entschieden hat (mit der Wirbel bekanntlich nicht beschrieben werden können).

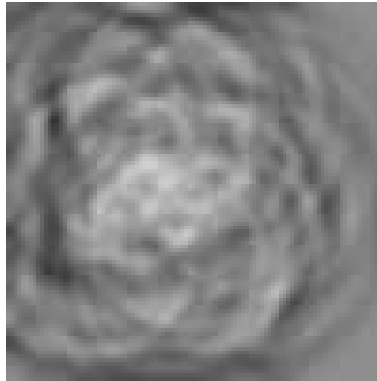


Abbildung 1: Höhenfeld, das eine Wasseroberfläche beschreibt, auf der sich bereits einige Wellen ausgebildet haben. Die Grauwerte geben die Höhe an dem entsprechenden Punkt an. Je heller, desto höher der Punkt.

Für eine Brunnensimulation ist das Verfahren allein also nicht geeignet. Seine Stärke spielt es bei sich sanft bewegendem Wasser aus, worin es sehr effizient ist und realistische Ergebnisse liefert.

2.2 Methode nach Foster/Metaxas

In [Foster & Metaxas '96], [Foster & Metaxas '97] (oder im Internet unter [Foster '99]) werden zur Simulation einer Flüssigkeit die Navier-Stokes-Gleichungen ([Zierep '97], [Guyon et al. '97]) innerhalb des gesamten Flüssigkeitsvolumens gelöst. Durch diesen Ansatz kann ein hoher Grad an Realismus erreicht werden. Spritzendes Wasser, Wasserfontänen, Wirbel, Wellen, all diese Effekte sind mit der Methode theoretisch machbar.

Der betroffene Raum wird dazu zunächst in ein Raumgitter unterteilt. Jede Zelle speichert die Geschwindigkeiten und den Druck innerhalb der Flüssigkeit. Der Druck ist dabei im Zentrum der Zelle definiert, die einzelnen Geschwindigkeitskomponenten jeweils in der Mitte der Seitenflächen. Anhand dieser Diskretisierung können später die Geschwindigkeiten und der Druck mit Hilfe der Navier-Stokes-Gleichungen aktualisiert werden. Der alleinige Zweck des Gitters ist es, die Geschwindigkeit der Flüssigkeit an einem beliebigen Punkt innerhalb des Gitters zu ermitteln. Dazu wird die Geschwindigkeit anhand der im Gitter nächstgelegenen Geschwindigkeitskomponenten interpoliert.

Für die Darstellung wird das Gitter nicht benötigt. Stattdessen wird die eigentliche Flüssigkeit getrennt vom Gitter gespeichert, z. B. als Partikelsystem oder auch als Höhenfeld. Die Kopplung zwischen Flüssigkeit und Gitter ist dadurch gegeben, dass die Flüssigkeitsrepräsentation die Zellen definiert, in denen sich Flüssigkeit befindet (siehe Abbildung 2). Dazu wird jede Zelle, je nach ihrem Inhalt, in eine von 6 Kategorien eingeteilt: Leere Zelle, volle Zelle (d. h. sie ist vollständig mit Flüssigkeit gefüllt), Oberflächenzelle (sie enthält die Grenzschicht zwischen Flüssigkeit und Luft), festes Hindernis, Quelle und Abfluss.

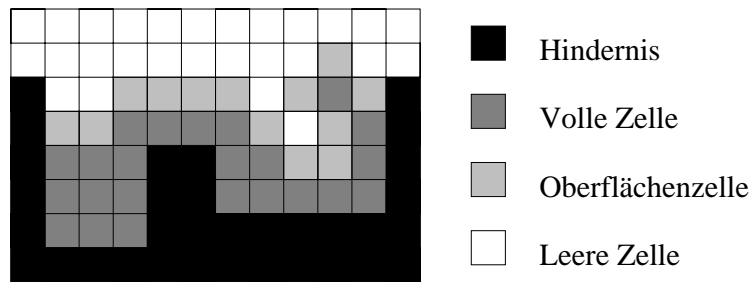


Abbildung 2: Einteilung des Gitters in unterschiedliche Zelltypen.

Zu Beginn der Simulation müssen die Zellen in eine dieser Kategorien eingeteilt und die Anfangsgeschwindigkeiten, sowie der Anfangsdruck müssen initialisiert werden. Danach läuft ein einzelner Simulationsschritt grob nach folgendem Schema ab:

1. Neueinteilung der Zellen in leere und Flüssigkeitszellen.
2. Randbedingungen an Hindernis- und Oberflächenzellen setzen. Hier wird die freie Oberfläche der Flüssigkeit behandelt, sowie die Bedingung, dass keine Flüssigkeit durch ein Hindernis hindurchtreten kann.
3. In den Flüssigkeitszellen die Geschwindigkeiten anhand der Navier-Stokes-Gleichungen aktualisieren.
4. Den Druck anhand einer Iteration aktualisieren.
5. Flüssigkeitsposition anhand der Geschwindigkeiten im Gitter aktualisieren.

2.3 Smoothed Particle Hydrodynamics (SPH)

Die Anfänge von SPH stammen aus der Astrophysik, in der die Methode dazu benutzt wurde, Sterne, Nebel und Galaxien zu simulieren. Einführungen in SPH sind u. a. in [Benz '90], [Monaghan '92] oder [Morris '96] nachzulesen, zusätzliche Referenzen sind im WWW unter [Armitage '00] zu finden. In der Computergrafik wurde das Verfahren in [Desbrun & Gascuel '96] angewandt.

Smoothed Particle Hydrodynamics ist eine Lagrangesche Methode, um Fluide (im weitesten Sinne) zu simulieren, d.h. die Flüssigkeit wird durch Teilchen (Partikel) repräsentiert, die ein Volumenelement der Flüssigkeit darstellen und die mit der Zeit verfolgt werden. Man kann diese Partikel als „Messpunkt“ ansehen, die in dem zu simulierenden Material verteilt sind und an deren Position eine gesuchte Größe (wie z. B. der Druck) exakt bekannt ist. Zwischen diesen Messpunkten wird eine Größe durch Interpolation der in der Nähe liegenden Punkten angenähert. Somit kann das Verfahren als ein stochastisches Verfahren angesehen werden, bei dem eine Messgröße durch stochastisch verteilte Stützstellen approximiert wird.

Um die angenäherte Größe zu erhalten, wird eine Teilchengröße (z. B. die Masse) nicht als in einem Punkt konzentriert angesehen, sondern sie wird um diesen Punkt herum

verteilt. Im Falle der Masse erhält man dadurch eine Dichtefunktion. An der Partikelposition selbst ist die Dichte dabei am größten, während sie in wachsender Entfernung abnimmt. Die ursprüngliche Masse erhält man dadurch wieder, dass man die Dichtefunktion über den gesamten Raum integriert. Man kann sich die Masse also über den Raum „verschmiert“ vorstellen, deshalb auch der Name *Smoothed Particle Hydrodynamics* (siehe Abbildung 3).



Abbildung 3: (a) Bei herkömmlichen Partikelsystemen ist eine Teilcheneigenschaft (z. B. die Masse) meist in einem Punkt konzentriert. (b) Bei SPH wird die Teilcheneigenschaft über den Raum „verschmiert“.

Dadurch, dass eine Teilchengröße nun nicht mehr in einem einzigen Punkt konzentriert, sondern als (differenzierbare) Funktion im Raum verteilt ist, kann man diese Größe für jede beliebige Position annähern. Dies geschieht einfach dadurch, dass man die jeweiligen Beiträge jedes Partikels an dem betrachteten Punkt aufsummiert. Zusätzlich ist es auch möglich, den Gradienten einer Teilchengröße zu berechnen, wie z. B. den Druckgradienten, der ja bekanntlich in der Bewegungsgleichung von Flüssigkeiten auftaucht.

Der Vorteil gegenüber Eulerschen Methoden ist der, dass man eine intuitivere Beschreibung der Flüssigkeit hat. Die Flüssigkeit ist da, wo die Partikel sind. Oder andersherum: die Partikel sind immer da, wo die Flüssigkeit ist. Man hat also keine „leeren Zellen“ (wie es in Abschnitt 2.2 vorkommen kann), die nur Speicherplatz und Rechenzeit verbrauchen, sondern die Berechnung erfolgt zwangsläufig nur an denjenigen Stellen, wo sie auch benötigt wird.

2.4 Fazit

Beim Einsatz in einer Brunnensimulation tauchen bei den eben vorgestellten Verfahren unterschiedliche Probleme auf.

Wie bereits erwähnt, ist die Wellengleichung eine sehr spezialisierte Methode, die damit nur bedingt einsetzbar ist (beispielsweise für ein Sammelbecken, das ein größeres Wasservolumen enthält).

Bei der Implementierung der Methode aus Abschnitt 2.2 zeigte sich, dass es

hauptsächlich die numerischen Probleme sind, die die Methode für eine Brunnensimulation letztendlich unbrauchbar machen. Durch eine Erhöhung der Viskosität lässt sich die Simulation zwar stabiler machen, das Ergebnis sieht aber zu dickflüssig aus, als dass man es für Wasser halten könnte. Eine geringere Viskosität verlangt aber ein feineres Gitter und kleinere Zeitschritte, wodurch die Rechenzeit, sowie der Speicherbedarf, beträchtlich ansteigen. Gelegentliche „Explosionen“ der Flüssigkeit konnten aber auch durch diese Maßnahme nicht verhindert werden. Zusätzlich hat auch die Partikelrate einen Einfluss auf die Stabilität des Verfahrens, wodurch es insgesamt sehr schwierig wird, zu einem gewünschten Ergebnis zu kommen. Ein weiterer Nachteil ist der, dass die Brunnengeometrie an das Raumraster des Algorithmus' ausgerichtet sein muss. Gekrümmte Oberflächen können also nur grob approximiert werden.

Ein reines 3D-Partikelsystem, in dem beispielsweise der SPH-Formalismus eingesetzt wird, hat den Vorteil, dass es sich an beliebige Geometrien anpassen kann. Diese Allgemeinheit erkaufte man sich allerdings auch mit einem hohen Rechenaufwand, da für jedes Partikel ein Kollisionstest mit der Geometrie gemacht werden muss. Zusätzlich zu diesem Aufwand kommt noch die eigentliche Bewegung der Flüssigkeit, wobei sich die Partikel untereinander beeinflussen können. Des Weiteren wird im SPH-Formalismus üblicherweise nur eine vereinfachte Bewegungsgleichung für Flüssigkeiten verwendet, da für den allgemeinen Einsatz der Navier-Stokes-Gleichungen zweite Ableitungen berechnet werden müssten. Dies ist aber nicht so leicht zu bewerkstelligen.

Wie man sieht, hat also jedes Verfahren seine Stärken und Schwächen. Je nach Anwendungsbereich kann einmal das eine, einmal das andere Verfahren am besten geeignet sein. In einer Brunnensimulation kann das Wasser in unterschiedlichster Weise in Erscheinung treten, so dass ein einzelnes Verfahren hier versagen würde. Deshalb liegt es nahe, die Verfahren zu kombinieren, so dass man die Stärken ausnutzen kann, wobei die Schwächen durch andere Verfahren überdeckt werden. Man erzeugt sich eine Simulation sozusagen nach dem „Baukastenprinzip“.

Das im folgenden Abschnitt vorgestellte Oberflächenpartikelsystem ist ein Baustein eines solchen Baukastens. Es ist auf die Simulation eines dünnen Wasserfilms spezialisiert, der sich auf einer Oberfläche (hier die Brunnenoberfläche) bewegt.

3 Oberflächenpartikelsystem

Für manche Zwecke lässt sich die Komplexität eines Partikelsystems um eine Dimension reduzieren, wenn man die Bewegung der Partikel auf eine Oberfläche beschränkt. Diese Vereinfachung ist dann gerechtfertigt, wenn das zu simulierende Wasser keine größeren zusammenhängende Volumina mehr bildet, sondern als schmaler Wasserfilm oder einzelner Tropfen auf einer Oberfläche vorhanden ist. Neben dem Vorteil des geringeren Speicherverbrauchs und der niedrigeren Rechenzeit, ist in solch einem Modell auch automatisch die Adhäsion, also die Anziehung zwischen Oberfläche und Flüssigkeit, vorhanden.

In [Fournier et al. '98] wird diese Idee bereits zur Simulation von einzelnen Wassertropfen verwendet. Jeder Tropfen wird von einem Partikel repräsentiert, das von Gravitation und Reibung beeinflusst wird und mit anderen Partikeln kollidieren und sich vereinigen kann. Tropfen können sich von der Oberfläche ablösen, wenn die Gravitation stärker als die Adhäsion wird. Eine Rauheit der Oberfläche wird dadurch simuliert, dass die tangentielle Kraft um einen bestimmten Faktor verringert wird. Zur Darstellung wird an jeder Partikelposition ein „Tropfen-Objekt“ erzeugt, dessen Form mit der Bewegung des Partikels und der Orientierung der Fläche korreliert ist. In diesem Abschnitt wird eine ähnliche Methode vorgestellt, die jedoch einen gesamten Wasserfilm anstelle einzelner Tropfen simuliert. Die Bewegungsgleichung ist dabei im Prinzip die gleiche wie in dem genannten Artikel, jedoch wird eine völlig andere Darstellungsart benutzt und es werden einige zusätzliche Effekte eingebaut.

Hier besteht das Oberflächenpartikelsystem aus zwei Komponenten,

- einem Dreiecksnetz,
- und den Partikeln mit einer Position im 3D-Raum.

Die Partikel besitzen als Attribut neben der 3D-Position eine Geschwindigkeit und eine Masse. Zusätzlich ist jedem Partikel dasjenige Dreieck aus dem Dreiecksnetz zugeordnet, auf dem es sich gerade befindet.

Die Simulation gliedert sich in zwei Schritte:

1. Aktualisieren der Geschwindigkeiten in 3D.
2. Bewegen der Partikel in 2D.

Eine nähere Erklärung dieser beiden Schritte erfolgt in den nächsten Abschnitten.

3.1 Aktualisierung der Geschwindigkeiten

Der Einfachheit halber werden die Partikelpositionen, die Geschwindigkeiten sowie die wirkenden Kräfte im dreidimensionalen Raum beschrieben. Damit sind alle für eine Geschwindigkeitsaktualisierung benötigten Komponenten im gleichen Koordinatensystem gegeben und es muss nicht für jedes Partikel bzw. jedes Dreieck eine Basis-Transformation durchgeführt werden.

Das Verfahren kann bereits mit einer sehr einfachen Bewegungsgleichung angewendet werden. Es genügt schon, wenn lediglich die Gravitation und Reibung berücksichtigt werden.

Ausgangspunkt ist also die Gleichung:

$$\mathbf{F} = m \cdot \mathbf{a}$$

Die Kraft auf der linken Seite setzt sich aus den in den folgenden Unterabschnitten beschriebenen Kräfte zusammen.

Gravitation

Die Gravitation ist als Vektor gegeben, der die Richtung und den Betrag der Gravitationskraft angibt. Da das Partikel sich nur auf der Oberfläche bewegen kann, trägt nur die Tangentialkomponente der Gravitation zu einer Impulsänderung des Partikels bei (siehe Abbildung 4).

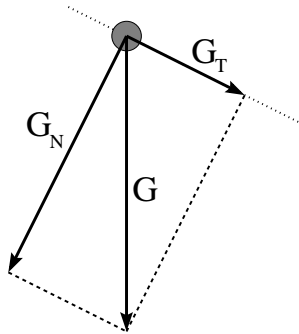


Abbildung 4: Die Gravitation muss in Normal- und Tangentialkomponente zerlegt werden. Nur die Tangentialkomponente bewirkt eine Impulsänderung an dem Partikel.

Die Normalkomponente \mathbf{G}_N und Tangentialkomponente \mathbf{G}_T der Gravitation \mathbf{G} sind gegeben durch:

$$\mathbf{G}_N = \frac{\mathbf{N} \cdot \mathbf{G}}{\mathbf{N} \cdot \mathbf{N}} \mathbf{N}$$

und

$$\mathbf{G}_T = \mathbf{G} - \mathbf{G}_N$$

\mathbf{N} ist hierbei der Normalenvektor der Fläche, auf der sich das Partikel befindet (dabei spielt es keine Rolle, ob er nach innen oder außen zeigt).

Reibung

Die Reibungskraft wirkt immer der aktuellen Geschwindigkeit entgegen und versucht somit, das Partikel abzubremesen. Der Betrag der Reibung ist proportional zum Betrag der Geschwindigkeit (siehe Abbildung 5):

$$\mathbf{F}_{\text{Reibung}} = -k_r \cdot \mathbf{v}$$

Mit dem Reibungsfaktor k_r kann eingestellt werden, ob es sich um eine eher glatte oder raue Oberfläche handelt.

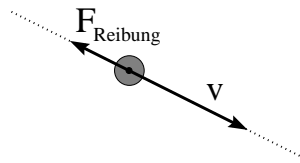


Abbildung 5: Die Reibung wirkt immer der aktuellen Geschwindigkeit entgegen und ist proportional zu deren Betrag.

3.2 Bewegung der Partikel

Beim Bewegen der Partikel sind zwei Dinge zu berücksichtigen:

- Die Bewegung darf nur auf der Oberfläche stattfinden.
- Die Geschwindigkeit muss immer tangential zur Oberfläche sein.

Solange sich ein Partikel nur innerhalb eines Dreiecks bewegt, macht es keine Schwierigkeiten, diese Punkte einzuhalten. Problematisch wird es erst, wenn das Partikel auf ein anderes Dreieck überwechselt. In diesem Fall ist die grobe Vorgehensweise wie folgt:

- Übergangskante ermitteln, über die das Partikel das Dreieck verlässt.
- Projizieren der Position und Geschwindigkeit auf das Nachbardreieck.
- Eventuell rekursiver Aufruf dieser Prozedur, falls das Partikel das Nachbardreieck im gleichen Schritt schon wieder verlässt.

Hierbei werden einige Anforderungen an die verwendeten Datenstrukturen gestellt. Zum einen muss für jedes Partikel bekannt sein, auf welchem Dreieck es sich gerade befindet und zum anderen müssen die Nachbarschaftsbeziehungen der Dreiecke untereinander bekannt sein, so dass benachbarte Dreiecke schnell ausfindig gemacht werden können.

u,v-Parameter

Im Algorithmus wird an verschiedenen Stellen die Koordinate eines Partikels bezüglich seines Dreiecks benötigt. Man wählt also zwei beliebige Kanten des Dreiecks aus und stellt die Position des Partikels unter Verwendung dieser Basis dar (siehe Abbildung 6). Die resultierende Koordinaten werden hier *u, v-Parameter* genannt. Es handelt sich hierbei um nichts anderes als zwei der Komponenten der baryzentrischen Koordinate des Partikels bezüglich der Eckpunkte des Dreiecks.

Ein Punkt liegt innerhalb eines Dreiecks, wenn sämtliche Komponenten der baryzentrischen Koordinate positiv sind. Oder nur anhand der *u, v-Parameter* ausgedrückt muss also gelten:

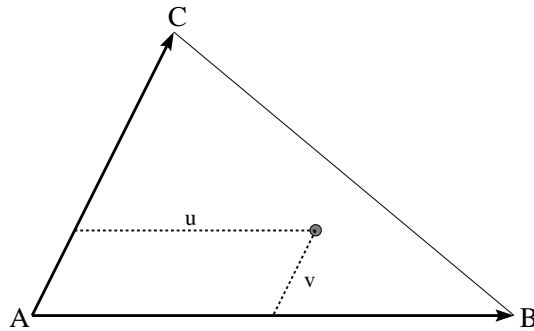


Abbildung 6: Zwei beliebige Achsen des Dreiecks dienen als Basis für die u, v -Parameter. Es sind zwei Komponenten der baryzentrischen Koordinate des Partikels bezüglich der Eckpunkte des Dreiecks.

$$\begin{aligned} u &> 0 \\ v &> 0 \\ u + v &< 1 \end{aligned}$$

Anhand der u, v -Parameter kann auch leicht festgestellt werden, über welche Kante ein Partikel das Dreieck verlässt. Man hat in diesem Fall eine Gerade im u, v -Parameterbereich, die durch den Anfangspunkt und den Zielpunkt geht (siehe Abbildung 7). Die Gerade wird geschrieben als:

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} + t \begin{pmatrix} r_u \\ r_v \end{pmatrix}$$

r_u und r_v geben die Richtung der Bewegung im Parameterbereich an, $r_u = u_2 - u_1$, $r_v = v_2 - v_1$.

Der Schnittpunkt mit Kante AB ergibt sich durch nullsetzen von \tilde{v} und ist beim Parameter t_{AB} zu finden:

$$t_{AB} = -\frac{v_1}{r_v}$$

Analog der Schnittpunkt mit der Kante AC:

$$t_{AC} = -\frac{u_1}{r_u}$$

Beim Schnittpunkt mit der Kante BC muss $\tilde{u} + \tilde{v} = 1$ gelten, damit ist der Schnitt bei:

$$t_{BC} = \frac{1 - u_1 - v_1}{r_u + r_v}$$

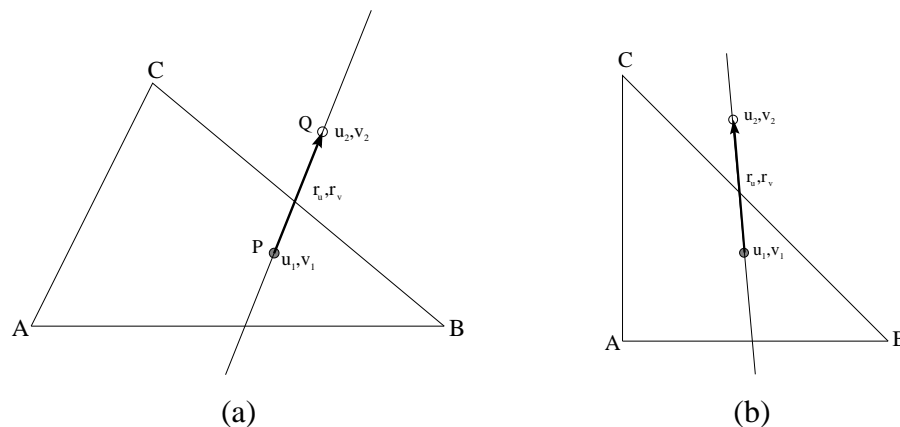


Abbildung 7: Ein Partikel bewegt sich von Punkt P nach Q und überquert dabei die Kante BC. (a) Im 3D-Raum. (b) Im u, v -Parameterbereich.

Falls der Nenner in diesen Ausdrücken gleich null ist, so verläuft die Gerade parallel zu der entsprechenden Kante und es existiert kein Schnittpunkt.

Eine weitere Anwendung der u, v -Parameter besteht darin, sie als Pixel-Index in einem Bild zu benutzen, in dem Informationen über nasse und trockene Bereiche gespeichert werden (siehe Abschnitt 3.3).

Bewegungsalgorithmus

Ein Partikel, welches sich auf dem Dreieck T befindet, soll von seiner ursprünglichen Position P zu dem Zielpunkt $Q = P + \Delta t \cdot v$ bewegt werden (siehe Abbildung 7). Q liegt unter Umständen nicht mehr auf der Oberfläche und muss demnach angepasst werden.

Eingabe für den Bewegungsalgorithmus ist die Anfangsposition P und der Verschiebungsvektor dr (tangential zur Oberfläche), um den das Partikel verschoben werden soll. Die Bewegung des Partikels läuft dann nach folgendem Schema ab:

1. u, v -Parameter von P und $Q = P + dr$ bestimmen $\rightarrow u_P, v_P$ und u_Q, v_Q .
2. Liegt Q noch innerhalb von T ? Dann ist Q die neue Partikelposition und es kann abgebrochen werden.
3. Andernfalls muss die Kante ermittelt werden, über die das Partikel das Dreieck T verlässt (wie oben beschrieben). Als Ergebnis erhält man den Schnittpunkt S und das Nachbardreieck T_n (siehe Abbildung 8).
4. Bestimmen des Winkels zwischen dem Dreieck T und seinem Nachbar T_n (anhand der Normalen).
5. Falls es sich um eine *konvexe* Kante handelt (d.h. vom Dreieck T aus gesehen geht das Nachbardreieck nach „unten“ weg, so wie in Abbildung 8 dargestellt),

so muss getestet werden ob die Geschwindigkeit des Partikels groß genug ist, damit es die Oberfläche verlassen kann.

6. Q auf die Fläche projizieren, in der das Nachbardreieck T_n liegt, daraus ergibt sich der Schnittpunkt R .
7. Die Richtung SR gibt nun die Richtung der neuen Geschwindigkeit und die Richtung der verbleibenden Bewegung $d\vec{r}$ an. Die Länge von $d\vec{r}$ wird mit der Länge von SQ gleichgesetzt.
8. Rekursiver Aufruf mit den Eingabeparametern S und $d\vec{r}$.

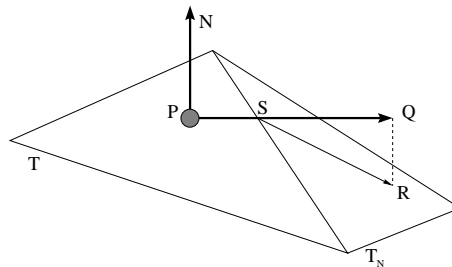


Abbildung 8: Ein Partikel bewegt sich von P nach Q und wechselt dabei das Dreieck. S ist der Schnittpunkt mit der Übergangskante, R die Projektion von Q auf das Nachbardreieck T_n .

3.3 Zusätzliche Effekte

Variation der Normalen

Bis jetzt wirkt auf die Partikel allein die Gravitation und eine Reibung mit konstantem Reibungsfaktor. Die Reibung simuliert zwar schon eine etwas raue Oberfläche, jedoch wird das Teilchen dadurch nur gebremst, seine Richtung wird sich aufgrund der Reibung nicht ändern. Die Bewegung der Partikel ist damit noch sehr gleichförmig. Eine etwas zufälliger Bewegung kann man dadurch erhalten, dass man den Normalenvektor der Oberfläche an jedem Punkt etwas variiert (siehe Abbildung 9). Das Prinzip ist das gleiche, das auch bei computergenerierten Bildern benutzt wird, um kleine Unebenheiten in der Oberfläche zu simulieren und ist dort unter dem Namen *Bump-Mapping* bekannt ([Blinn '78]).

Die Variation des Normalenvektors wird in dieser Arbeit anhand einer sogenannten *Noise-Funktion* ([Ebert et al. '98]) bewerkstelligt. Die *Noise-Funktion* liefert einem zu einem beliebigen Punkt im Raum einen „Zufallsvektor“. Die Funktion ist dabei so beschaffen, dass sie immer den gleichen Vektor zurückgibt, wenn sie mit demselben Eingabepunkt aufgerufen wird. Weicht der Eingabepunkt nur leicht vom ursprünglichen Punkt ab, so weicht auch der Ausgabevektor nur leicht vom ursprünglichen Ausgabevektor ab. Sind zwei Eingabepunkte jedoch weit voneinander entfernt, so besitzen

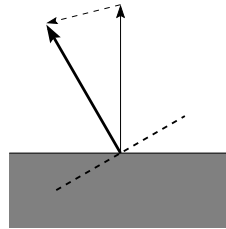


Abbildung 9: Um Unebenheiten in der Oberfläche zu simulieren, wird der Normalenvektor an jedem Punkt zufällig variiert.

sie auch völlig unabhängige Ausgabevektoren. Mit solch einer Funktion lässt sich also an einem beliebigen Punkt ein Zufallsvektor erzeugen, den man auf die Oberflächennormale an diesem Punkt aufaddiert. Dadurch ergibt sich eine unebene Oberfläche und das Partikel verändert seine Bewegungsrichtung und wird abgebremst oder beschleunigt.

Adhäsion

Ein weiterer Zusatz ist die Einführung einer Adhäsion. Bisher ist die Adhäsion im Modell unendlich hoch, da die Partikel gezwungenermaßen die Oberfläche nicht verlassen können. Im Bewegungsalgorithmus im letzten Abschnitt ist bereits eine Stelle vorgesehen, an dem die Partikel unter Umständen die Oberfläche verlassen können, nämlich dann, wenn sie ein Dreieck über eine konvexe Kante verlassen. Im Prinzip bedeutet das auch schon, dass die Adhäsion nicht stark genug ist, um das Partikel auf der Oberfläche zu halten. Jedoch wird dieser Fall anders behandelt als der hier neu eingeführte Test.

Die Partikel verlassen die Oberfläche auch dann, wenn die in Normalenrichtung wirkende Kraft größer als ein bestimmter Betrag wird und das Partikel sich damit nicht mehr auf der Oberfläche halten kann (siehe Abbildung 10). Die Kraft, die das Partikel an der Oberfläche hält (die Adhäsionskraft) wirkt in Normalenrichtung und ist abhängig von der Kontaktfläche πr^2 und einem Proportionalitätsfaktor k_a :

$$F_{\text{Reibung}} = k_a \pi r^2$$

Das Partikel fällt von der Oberfläche, wenn für die Beschleunigung a_n in Normalenrichtung gilt:

$$a_n > \frac{k_a \pi r^2}{m}$$

Der Radius r des Partikels kann aus seiner Masse m berechnet werden (wenn man das Partikel als Kugel betrachtet):

$$r = \sqrt[3]{\frac{3}{4\pi} \frac{m}{\rho}}$$

ρ ist hierbei die Dichte der Flüssigkeit (bei Wasser also $1000 \frac{\text{kg}}{\text{m}^3}$).

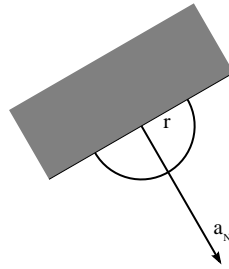


Abbildung 10: Das Partikel löst sich von der Oberfläche ab, wenn die Kraft in Normalenrichtung größer wird als die Adhäsionskraft.

Nass/Trocken-Textur

Bestimmte Materialien sehen in der Realität in nassem Zustand anders aus, als in trockenem. Beispielsweise wird Sandstein deutlich dunkler, wenn man ihn mit Wasser übergießt. Für eine realitätsnahe Computeranimation sollte man diese Tatsache nicht außer Acht lassen.

Bei jedem Punkt auf der Oberfläche muss also ermittelt werden, ob bzw. wie nass er ist. Dies wird anhand einer Textur gemacht, in der genau diese Information gespeichert ist. Dazu wird jedem einzelnen Dreieck eine Graustufenbild zugewiesen, dessen Bildpunkte jeweils einer bestimmten Position auf dem Dreieck entsprechen. Ist der Bildpunkt gesetzt, so ist diese Stelle auf dem Dreieck nass, ist er gelöscht, ist sie trocken (siehe Abbildung 11).

Glücklicherweise gibt es dabei gar nichts neues mehr zu berechnen, da im Bewegungsalgorithmus schon die benötigten Größen ermittelt wurden, nämlich die u, v -Parameter des Partikels auf dem Dreieck. Genau diese Parameter kann man als Index in die Textur verwenden und den Bildpunkt an dieser Stelle setzen. Das Trocknen der Oberfläche kann dadurch simuliert werden, dass die Intensität der Bildpunkte in der Textur mit der Zeit abgeschwächt wird. Diejenigen Stellen auf der Oberfläche, die also nicht kontinuierlich mit Partikeln „benetzt“ werden, werden dadurch nach einiger Zeit wieder trocken.

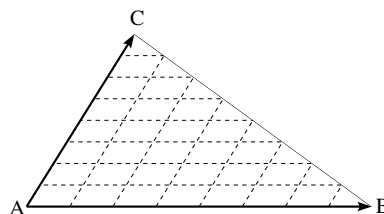


Abbildung 11: Lage der Nass/Trocken-Textur innerhalb eines Dreiecks. Die Zellen, die von den gestrichelten Linien gebildet werden, entsprechen den Bildpunkten in der Textur (hier eine 8x8-Bitmap).

Für die Darstellung muss das Bild wieder auf die gleiche Weise auf das Dreieck projiziert werden. Für die RenderMan-Schnittstelle stellt dies aber überhaupt kein Problem dar, da dort jedem Eckpunkt eine Texturkoordinate zugewiesen werden kann. Die

unterschiedliche Darstellung der nassen und trockenen Bereiche geschieht dann über einen speziellen Surface-Shader, der seine Darstellung abhängig von der Information in der Nässe-Textur macht.

Außer für die Darstellung kann die Nass/Trocken-Textur auch wieder in der Bewegungsgleichung berücksichtigt werden, indem an nassen Stellen der Reibungskoeffizient vermindert wird. Dadurch gleitet das Wasser auf einem bereits bestehenden Wasserfilm schneller als auf trockenen Stellen.

4 Kombination von Methoden

Da das Wasser, wie bereits festgestellt, bei einer Brunnensimulation in unterschiedlichster Weise in Erscheinung treten kann, würde hier eine einzelne Methode versagen. Im Folgenden werden deshalb drei Modelle zur Flüssigkeitssimulation miteinander verknüpft, nämlich

- die Wellengleichung,
- das 3D-Partikelsystem,
- und das Oberflächenpartikelsystem.

Prinzipiell ist ein Übergang von jedem Flüssigkeitsmodell in jedes der anderen denkbar. Die häufigsten Übergänge sind jedoch:

- (1) Oberflächenpartikelsystem → 3D-Partikelsystem
- (2) 3D-Partikelsystem → Oberflächenpartikelsystem
- (3) Oberflächenpartikelsystem → Wellengleichung
- (4) 3D-Partikelsystem → Wellengleichung

Der Übergang (1) tritt immer dann auf, wenn die Adhäsionskraft zu schwach ist, um ein Partikel an der Oberfläche zu halten oder wenn ein Partikel über eine konvexe Kante hinausschießt. Der Übergang (2) tritt einfach dann auf, wenn die Flüssigkeit wieder auf die dem Oberflächenpartikelsystem zugeordnete Oberfläche auftrifft. Bei (3) und (4) treffen ebenfalls Partikel auf eine Oberfläche auf, jedoch ist hier die Oberfläche nicht der Brunnen selbst, sondern die Flüssigkeit in dem Brunnen. Die beiden fehlenden Übergänge von der Wellengleichung in eines der beiden Partikelsysteme werden in dieser Arbeit nicht berücksichtigt, da sie in dem Beispiel-Brunnen aus Kapitel 5 eine untergeordnete Rolle spielen. Möglich sind sie aber allemal. Zum Beispiel könnten starke Wellen, bei denen die senkrechte Geschwindigkeitskomponente eine gewisse Schwelle überschreitet, dazu führen, dass Partikel im 3D-Partikelsystem erzeugt

werden, mit denen die davonspritzenden Tropfen simuliert werden. Ein Übergang in das Oberflächenpartikelsystem könnte immer dann stattfinden, wenn am Beckenrand die Flüssigkeit überschwappt.

Der Übergang von Flüssigkeit A nach Flüssigkeit B muss unter zwei Aspekten betrachtet werden:

1. *Wann* tritt der Übergang ein?
2. *Wie* geht der Übergang vonstatten?

Zur Beantwortung der ersten Frage muss jedes Partikel in jedem Schritt für jeden möglichen Übergang einem Test unterworfen werden, mit dem festgestellt wird, ob das Partikel in die andere Flüssigkeit übergeht oder nicht. Die Bedingung, die zu einem Wechsel führt, kann entweder eine *Austrittsbedingung* aus A oder eine *Eintrittsbedingung* in B sein. Im ersten Fall übernimmt A den aktiven Part, im zweiten Fall ist B die aktive Flüssigkeit. Ein Beispiel verdeutlicht den Unterschied: Bei Flüssigkeit A soll es sich nun um ein Oberflächenpartikelsystem handeln, bei Flüssigkeit B um ein 3D-Partikelsystem. Ein Partikel macht den Wechsel von A nach B, wenn die Austrittsbedingung erfüllt ist, dass die Adhäsionskraft schwächer als die Erdanziehung ist. Beim Wechsel von B nach A ist die Eintrittsbedingung, dass ein Partikel aus B auf die Oberfläche des Oberflächenpartikelsystems auftrifft. In beiden Fällen ist das Oberflächenpartikelsystem die aktive Flüssigkeit, da nur sie den Test durchführen kann. Wie solch ein Test nun in jedem einzelnen Fall aussieht, hängt von der jeweiligen Flüssigkeit ab.

Die Beantwortung der zweiten Frage hängt sehr davon ab, wie eine Methode die Flüssigkeit intern repräsentiert. Im Falle der obigen drei Simulationsmethoden kann man feststellen, dass zwei der Methoden ein Partikelsystem verwenden und die dritte Methode ein Höhenfeld. Die Brunnengeometrie wird im Oberflächenpartikelsystem als Dreiecksnetz spezifiziert. Das Höhenfeld in der Wellengleichung kann ebenfalls als Dreiecksnetz angesehen werden (die Gitterpunkte bilden dabei die Eckpunkte der Dreiecke). Damit ergeben sich also im wesentlichen nur zwei unterschiedliche Übergänge:

Partikelsystem → Partikelsystem

Partikelsystem → Dreiecksnetz

Die restlichen Abschnitte erläutern jeweils den Übergang zwischen zwei Flüssigkeiten in einer Richtung.

4.1 Oberfläche – 3D-Partikelsystem

Die beiden Fragen nach dem *wann* und dem *wie* sind beim Übergang vom Oberflächenpartikelsystem in das 3D-Partikelsystem recht unproblematisch und führen kaum zu

einem zusätzlichen Rechenaufwand. Der Vorteil ist hier, dass beide Methoden auf ähnliche Weise funktionieren und sogar ähnliche Datenstrukturen verwenden. Beide benutzen Partikelsysteme zur Repräsentation der Flüssigkeit und beide führen ihre Berechnungen direkt anhand der Partikel aus (im Gegensatz zur Methode aus Abschnitt 2.2, bei der das Partikelsystem lediglich die momentane Position der Flüssigkeit definiert).

Die Bedingung, die zu einem Wechsel führt, ist in diesem Fall eine Austrittsbedingung. Nur das Oberflächenpartikelsystem weiß, wann der Wechsel stattfindet, nämlich immer dann, wenn entweder die Adhäsionskraft zu schwach ist, um ein Partikel an der Oberfläche zu halten oder wenn das Partikel beim Übergang über eine konvexe Kante eine so große Geschwindigkeit besitzt, dass es über die Kante hinauschießt (auch in diesem Fall ist die Adhäsion zu klein, um das Partikel auf der Oberfläche zu halten, jedoch wird der Test an anderer Stelle gemacht). Beide Fälle werden innerhalb des Oberflächenpartikelsystems behandelt. Die Frage nach dem *wann* wäre somit geklärt. Bezüglich der Frage nach dem *wie* wurde oben schon angedeutet, dass der eigentliche Wechsel keine Probleme macht, da beide Methoden auf ähnliche Datenstrukturen basieren, nämlich dem Partikel. Im Falle einer erfüllten Austrittsbedingung muss das Oberflächenpartikelsystem also lediglich das Partikel an das 3D-Partikelsystem „übergeben“, wobei eventuell eine Konvertierung von einer Partikelrepräsentation in eine andere gemacht werden muss. Die Attribute wie Position, Geschwindigkeit und Masse werden dazu einfach kopiert, für eventuell fehlende Attribute werden Default-Werte benutzt.

4.2 3D-Partikelsystem – Oberfläche

Dieser Übergang ist der umgekehrte Fall wie im letzten Abschnitt, ein Partikel wechselt vom 3D-Partikelsystem in das Oberflächenpartikelsystem über. Der Test, ob der Wechsel stattfindet oder nicht, führt diesmal allerdings zu einem zusätzlichen Rechenaufwand. Offensichtlich findet der Wechsel ja immer dann statt, wenn Flüssigkeit auf die Oberfläche auftrifft, die dem Oberflächenpartikelsystem zugeordnet ist. Damit handelt es sich hier um eine Eintrittsbedingung in das Oberflächenpartikelsystem. Das bedeutet, dass in jedem Bewegungsschritt geprüft werden muss, ob ein Partikel mit dem Dreiecksnetz kollidiert oder nicht. Findet eine Kollision statt, muss das Partikel wieder, wie bereits im letzten Abschnitt, konvertiert werden, wobei diesmal zu beachten ist, dass die Position auf die Oberfläche projiziert und die Geschwindigkeit so angepasst wird, dass sie tangential zur Oberfläche ist.

Dadurch, dass es bei diesem Übergang eine *Eintrittsbedingung* gibt, muss das Oberflächenpartikelsystem (welches die Bedingung testet) Zugriff auf Attribute eines Partikels im 3D-Partikelsystem erhalten und muss das 3D-Partikelsystem veranlassen können, das betreffende Partikel abzugeben. Das 3D-Partikelsystem selbst spielt hier nur eine passive Rolle.

4.3 Partikelsystem – Wellengleichung

Bei diesem Übergang trifft Flüssigkeit von einem Partikelsystem (es spielt dabei keine Rolle, ob es sich um das Oberflächen- oder das 3D-Partikelsystem handelt) auf die Wasseroberfläche auf, auf der die Wellengleichung angewandt wird. Die Frage nach dem Zeitpunkt des Wechsels ist damit klar, nämlich immer dann, wenn ein Partikel mit dem Höhenfeld kollidiert. Wie im letzten Abschnitt, handelt es sich hier wieder um eine Eintrittsbedingung, die von der Wellengleichung getestet werden muss.

Ein Höhenfeld kann als spezielle Repräsentation eines Dreiecksnetzes angesehen werden, bei dem die Gitterpunkte die Eckpunkte der Dreiecke bilden. Somit würde der Kollisionstest nach dem gleichen Schema wie im letzten Abschnitt ablaufen. Damit hätte man allerdings nicht die regelmäßige Struktur des Höhenfeldes ausgenutzt. Abgesehen davon, dass die Eckpunkte in einer Dimension ausgelenkt sind (je nach Amplitude der Flüssigkeit), sind sie in einem regelmäßigen (zweidimensionalen) Gitter mit jeweils gleichen Gitterabständen angeordnet. Es kann also leicht getestet werden, „in“ (bzw. über oder unter) welcher Zelle sich ein Partikel befindet (siehe Abbildung 12). Nun muss nur noch geprüft werden, ob sich die aktuelle Höhe in dieser Zelle über oder unter dem Partikel befindet. Liegt die Höhe über dem Partikel, so fand bereits eine Kollision statt (das Partikel befindet sich bereits im Flüssigkeitsvolumen), andernfalls liegt das Partikel noch über der Wasseroberfläche.

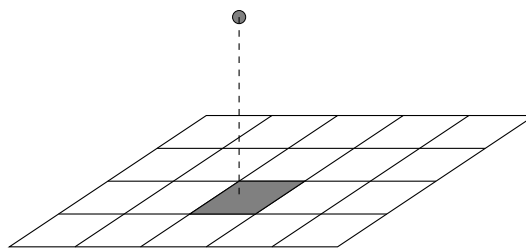


Abbildung 12: Ein Partikel befindet sich im Höhenfeld über der grauen Zelle. Liegt die Höhe der Flüssigkeit in dieser Zelle unterhalb des Partikels, so hat keine Kollision stattgefunden. Liegt sie jedoch oberhalb, so befindet sich das Partikel bereits im Flüssigkeitsvolumen.

Ist also die Eintrittsbedingung erfüllt, so muss das Partikel aus seinem ursprünglichen Partikelsystem gelöscht werden. Es kann nun nicht in ein anderes Partikelsystem eingefügt werden, da die Wellengleichung nicht mit Partikeln arbeitet. Damit sich aber von der Position, an der das Partikel in die Oberfläche eingetreten ist, eine Welle ausbilden kann, wird das Höhenfeld an dieser Stelle (die ja bereits für den Kollisionstest ermittelt wurde) um einen bestimmten Betrag „eingedrückt“ (kann in Abhängigkeit von der Masse und der Geschwindigkeit des Partikels gemacht werden). Im Höhenfeld ist dann eine Störung enthalten, von der sich eine Welle ausbreitet.

5 Experimentelle Ergebnisse

Die Animation eines Keilwürfelbrunnens demonstriert die Leistungsfähigkeit des gewählten Ansatzes. Im Beispiel wird eine Kombination aus drei unterschiedlichen Simulationsmodellen verwendet, dem Oberflächenpartikelsystem, einem 3D-Partikelsystem und der Wellengleichung.

Der Brunnen wurde von dem Künstler Andreas Helmling entworfen und besteht aus würfelförmigen Elementen mit jeweils einer Kantenlänge von einem halben Meter, aus denen zwei Keile aus gegenüberliegenden Seiten jeweils versetzt herausgeschnitten wurden. Sechs von diesen Elementen sind wiederum jeweils versetzt übereinander gestapelt und bilden das Zentrum des Brunnens (siehe Abbildung 13). Aus den Kanten in der Mitte eines Würfelements kann unterschiedlich viel Wasser hervorquellen.

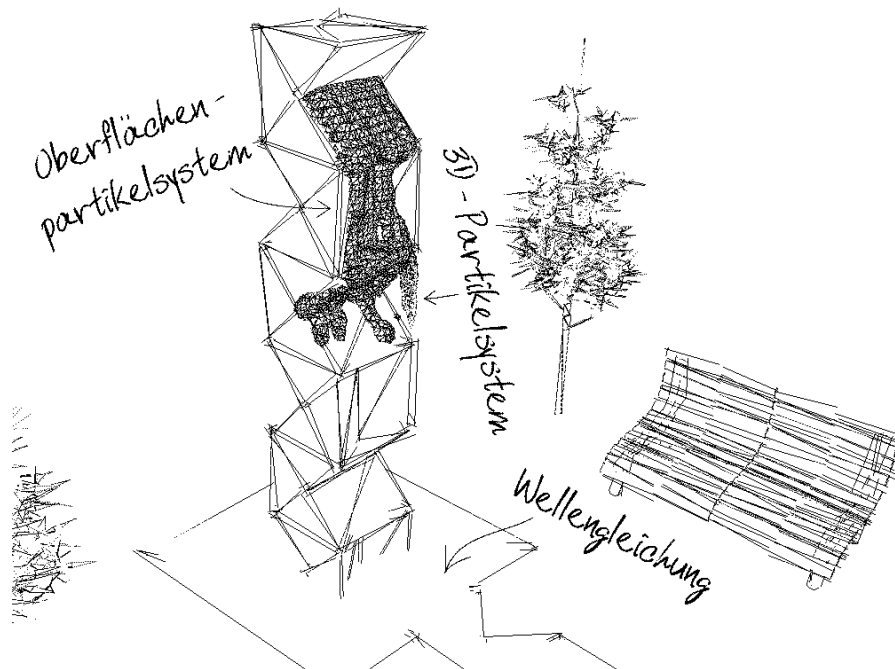


Abbildung 13: Skizze des Brunnens mit den verwendeten Simulationen.

Diese Anordnung steht nun wiederum in einem kleinen Bassin, von dem in einer Richtung ein Kanal abgeht. Der Kanal wird nach dem gleichen Muster wie schon die Brunnensäule gemacht und besitzt daher die gleiche Silhouette wie die Säule.

In der Animation tritt lediglich Wasser aus dem obersten Würfelement des Brunnens. Die Stärke, mit der das Wasser austritt, kann reguliert werden. Das gewünschte Verhalten ist nun so, dass das Wasser bei geringer Intensität an der Brunnenoberfläche „hinuntergleitet“, sich in dem Becken sammelt und im Kanal abfließen kann. Dabei spielt die Adhäsion eine große Rolle, da das Wasser auch an den nach innen laufenden Flächen entlang gleiten soll. Wird die Intensität des austretenden Wassers erhöht, so soll das Wasser an den Kanten als Wasservorhang herunterfallen.

In Abbildung 13 sind die drei verwendeten Simulationen bereits eingezeichnet.

net. Für das Wasser auf der Brunnensäule ist das Oberflächenpartikelsystem aus Abschnitt 3 verantwortlich. Von der Oberfläche kann sich Flüssigkeit ablösen und geht dann in ein 3D-Partikelsystem über. Wenn Flüssigkeit von dem Brunnen herunterfällt ist es hauptsächlich der freie Fall (also die Gravitation), der die Bewegung der Flüssigkeit beeinflusst. Aus diesem Grund wurde auf die Anwendung des SPH-Formalismus' aus Abschnitt 2.3 verzichtet und lediglich die Gravitation berücksichtigt. Schließlich wird im Becken und im Kanal die Wellengleichung an einem Höhenfeld angewendet. Das Höhenfeld hat dabei eine rechteckige Form und richtet sich der Einfachheit halber *nicht* nach der Geometrie des Brunnens. Lediglich in der Darstellung der Szene sieht es so aus, als wäre das Wasser nur in dem Becken und in dem Kanal enthalten. Diese zugegeben sehr grobe Vereinfachung liefert dennoch ein recht überzeugendes Bild eines Beckens und der darin entstehenden Wellen, auch wenn die physikalische Situation im Prinzip völlig verkehrt ist und die Wellen nicht an den korrekten Stellen reflektiert werden. Dadurch, dass aber zum einen die Aufmerksamkeit hauptsächlich auf die Brunnensäule gelenkt wird und zum anderen kein direkter Vergleich mit einem realen Brunnen dieser Art vorhanden ist, reicht die Methode aus, um die Illusion eines Wasserbeckens zu erzeugen.

Die Übergänge zwischen den einzelnen Simulationsmodellen, wie sie in Kapitel 4 erläutert wurden, sind in der Simulation enthalten, d.h. alle Übergänge sind möglich, außer von der Wellengleichung zurück zu einem Partikelsystem.

Zusammenfassung

Es hat sich herausgestellt, dass es kein Verfahren zur Flüssigkeitssimulation in der Computergrafik gibt, das in der Lage ist, alle bei Flüssigkeiten auftretenden Effekte überzeugend zu simulieren. Jedes der bekannten Verfahren ist mehr oder weniger auf die Simulation einer bestimmten Ausprägung von Flüssigkeiten in einem bestimmten Umfeld spezialisiert.

Verfahren, die ein Raumraster verwenden, eignen sich zur Flüssigkeitssimulation des Beispielbrunnens nur bedingt, da sich die Brunnengeometrie nur schlecht in das Raster einpassen lässt. Um eine ausreichend genaue Repräsentation des Brunnens erreichen zu können, müsste das Raster sehr fein gewählt werden. Auch bei einem noch relativ groben Raster erreicht der Speicherverbrauch schnell die Dimension mehrerer 100 MB.

Weiterhin hat sich gezeigt, dass das Lösen von Navier-Stokes-Gleichungen rechenintensiv ist und aufgrund der geringen Viskosität von Wasser numerische Instabilitäten auftreten können, die nur schwierig zu beherrschen sind. Eine physikalische Eigenschaft, die von den Navier-Stokes-Gleichungen nicht berücksichtigt wird, ist die Adhäsion. Gerade aber die Adhäsion spielt bei vielen Brunnenanlagen eine bedeutende Rolle.

Als eine Lösung dieses Problems hat sich die Kombination mehrerer spezialisierter Simulationsmethoden herausgestellt. Solche Methoden sind in der Regel sehr effizient und eignen sich zur Darstellung von Flüssigkeiten in einer bestimmten Form, z.B. die Wellengleichung zur Modellierung eines mit Wasser gefüllten Beckens. Die

Übergänge zwischen den einzelnen Methoden ergeben sich automatisch.

Eine solche Simulationemethode ist das Oberflächenpartikelsystem. Es entspricht im Ansatz zwar dem Verfahren von [Fournier et al. '98] ist aber zum Zweck der Brunnenanimation auf eine Wasseroberfläche erweitert. Durch die zusätzliche Berücksichtigung der Oberflächenrauheit (vgl. Abbildung 14 links oben) wird eine unregelmäßige Fließbewegung des Wassers bewirkt. Zu einer besonders realistischen Darstellung führt die Verwendung von Nass/Trocken Texturen, mit denen die Brunnenoberfläche überzogen wird. In Abbildung 14 sind einige Einzelbilder aus der erstellten Animation zu sehen.



Abbildung 14: Einzelbilder der Animation. Im Bild links oben sieht man den Effekt der Nass/Trocken Texturen; links unten ist der Übergang vom Oberflächen- in das 3D-Partikelsystem zu beobachten.

Literatur

[Armitage '00] Philip Armitage. Smooth Particle Hydrodynamics. WWW-Seite: <http://www.mpa-garching.mpg.de/~armitage/sph.html>, Jan. 2000 (gesehen Feb. 2000).

- [Benz '90] W. Benz. Smooth Particle Hydrodynamics: A Review. In J. Robert Buchler, Hrsg., *The numerical modelling of nonlinear stellar pulsations*, Seiten 269–288. Kluwer, März 1990.
- [Blinn '78] James F. Blinn. Simulation of Wrinkled Surfaces. *Computer Graphics*, 12(3):286–292, August 1978.
- [Chen & Lobo '95] Jim X. Chen und Niels Da Vitoria Lobo. Toward interactive-rate simulation of fluids with moving obstacles using Navier-Stokes equations. *Graphical models and image processing: GMIP*, 57(2):107–116, März 1995.
- [Desbrun & Gascuel '96] Mathieu Desbrun und Marie-Paule Gascuel. Smoothed Particles: A new paradigm for animating highly deformable bodies. WWW-Seite: <http://w3imagis.imag.fr/~Mathieu.Desbrun/smoothed.html>, 1996.
- [DreamWorks '98] DreamWorks. The Prince of Egypt. WWW-Seite: <http://www.prince-of-egypt.com>, 1998.
- [Ebert et al. '98] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin und Steven Worley. *Texturing & Modeling: A Procedural Approach*. Academic Press, 2. Auflage, 1998.
- [Foster & Metaxas '96] Nick Foster und Dimitri Metaxas. Realistic Animation of Liquids. *Graphical models and image processing: GMIP*, 58(5):471–483, September 1996.
- [Foster & Metaxas '97] Nick Foster und Dimitri Metaxas. Controlling Fluid Animation. In *Proceedings CGI '97*, Seiten 178–188, 1997.
- [Foster '99] Nick Foster. Nick Foster's Homepage. WWW-Seite: <http://www.cis.upenn.edu/~fostern>, Okt. 1997 (gesehen Jan. 1999).
- [Fournier & Reeves '86] Alain Fournier und William T. Reeves. A Simple Model of Ocean Waves. *Computer Graphics*, 20(4):75–84, August 1986.
- [Fournier et al. '98] Patrick Fournier, Arash Habibi und Pierre Poulin. Simulating the Flow of Liquid Droplets. In *Graphics Interface*, Seiten 133–142, Juni 1998.
- [Guyon et al. '97] Etienne Guyon, Jean-Pierre Hulin und Luc Petit. *Hydrodynamik*. vieweg, 1997.
- [Hodgins '99] J. K. Hodgins. Graphics, Visualization & Usability Center. WWW-Seite: <http://www.cc.gatech.edu/gvu/animation/index.html>, Dez. 1998 (gesehen Jan. 1999).
- [Kass & Miller '90] Michael Kass und Gavin Miller. Rapid, Stable Fluid Dynamics for Computer Graphics. *Computer Graphics*, 24(4):49–57, August 1990.

- [Miller & Pearce '89] Gavin Miller und Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *Computers and Graphics*, 13(3):305–309, 1989.
- [Monaghan '92] J. J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Reviews of Astronomy & Astrophysics*, 30:543–574, 1992.
- [Morris '96] Joseph P. Morris. *Analysis of Smoothed Particle Hydrodynamics with Applications*. Dissertation, Department of Mathematics, Monash University, Juli 1996.
- [O'Brien & Hodgins '95] J. F. O'Brien und J. K. Hodgins. Dynamic Simulation of Splashing Fluids. In *Computer Animation '95*, Seiten 198–205, April 1995.
- [Peachey '86] Darwyn R. Peachey. Modeling Waves and Surf. *Computer Graphics*, 20(4):65–74, August 1986.
- [Stam '99] Jos Stam. Stable Fluids. In *Proceedings of the ACM SIGGRAPH Conference (SIGGRAPH-99)*, Seiten 121–128, 1999.
- [Weimer & Warren '99] Henrik Weimer und Joe Warren. Subdivision Schemes for Fluid Flow. In *Proceedings of the ACM SIGGRAPH Conference (SIGGRAPH-99)*, Seiten 111–120, 1999.
- [Witting '99] Patrick Witting. Computational Fluid Dynamics in a Traditional Animation Environment. In *Proceedings of the ACM SIGGRAPH Conference (SIGGRAPH-99)*, Seiten 129–136, 1999.
- [Zierep '97] Jürgen Zierep. *Grundzüge der Strömungslehre*. Springer-Verlag, 6. Auflage, 1997.