

Realistische Darstellung von Flüssigkeiten und Gasen

Oliver Franzke*

Technische Universität Dresden

Abriss

Die realistische Darstellung von natürlichen Phänomenen war schon immer ein wichtiges Ziel der Computergrafik. Dies ist jedoch keine leichte Aufgabe. Dabei müssen unzählige Effekte beachtet werden, von denen einige noch nicht einmal vollständig untersucht wurden. Die mathematischen Formulierungen dieser Phänomene sind zudem meist so kompliziert, dass sie durch eine mehr oder weniger grobe Approximation ersetzt werden müssen, um sie praxistauglich zu machen.

So stellt insbesondere die Darstellung von natürlichen Szenen, wie zum Beispiel Wäldern, Ozeanen und anderen Landschaften ein ganz besonderes Problem dar. Häufig ist die geometrische Komplexität sehr groß, so dass gewisse Vereinfachungen in Kauf genommen werden müssen. So ist es zum Beispiel noch nicht möglich, einen Wald mit einer vollständigen Beleuchtungssimulation – in sinnvoller Rechenzeit - darzustellen.

Ein weiteres Beispiel stellen die gasförmigen Phänomene dar, die gegenüber Festkörpern den zusätzlichen Nachteil besitzen, dass sie lichtdurchlässig sind und häufig eine inhomogene Struktur aufweisen. Dadurch sind viele Vereinfachungen, die im Zusammenhang mit Festkörpern gemacht werden können nicht mehr möglich.

Diese Arbeit fasst die Entwicklungen der letzten Jahre zusammen, die im Bereich der gasförmigen Objekte gemacht wurden. Dabei wird insbesondere auf die Möglichkeiten der Darstellung dieser Phänomene eingegangen.

1 Einführung

Wir haben jeden Tag mit einer Vielzahl von Flüssigkeiten und Gasen zu tun bzw. sind von ihnen umgeben. Das Spektrum der Flüssigkeiten reicht von einem Glas voll mit Wasser bis hin zu ganzen Ozeanen. Die Anzahl, der uns bekannten Gase ist nicht minder umfangreich und umfasst sowohl die Flamme einer Kerze, als auch die Atmosphäre der Erde.

Diese Phänomene erscheinen auf den ersten Blick so unterschiedlich, dass man nicht vermuten würde, dass sie auf dasselbe physikalische Fundament aufbauen. Wasser und Rauch sind also grundsätzlich fast identisch. Die Ähnlichkeit von Gasen und Flüssigkeiten, werden im Rahmen dieser Arbeit noch häufiger deutlich werden.

Doch die Analyse dieser Effekte erfolgt nicht zum reinen Selbstzweck, es existiert eine Vielzahl von Anwendungsgebieten. Als erstes seien hier die Medizin, Architektur, Physik, Astronomie und die Auto- und Flugzeugindustrie genannt. In diesen Gebieten wird eine sehr genaue Beschreibung der gasförmigen Phänomene benötigt. Es wäre schließlich fatal, wenn zum Beispiel die Tragfläche eines Flugzeugs auf Grund einer fehlerhaften oder ungenauen Simulation abbrechen würde.

*E-Mail: of641854@inf.tu-dresden.de

Deswegen werden in diesen Anwendungsgebieten meist die exakten physikalischen Formulierungen verwendet. Um damit jedoch überhaupt Berechnungen durchführen zu können, ist entweder eine hohe Rechenleistung oder eine große Rechenzeit nötig. Dies kann allerdings meist ohne Probleme in Kauf genommen werden.

Ausgehend von diesen sehr genauen Beschreibungen wird anschließend nach approximativen Methoden gesucht, die dann zum Beispiel in der Unterhaltungsindustrie eingesetzt werden können. Es gilt, die ursprünglichen Formulierungen so weit zu vereinfachen, dass sie auch auf einem Standard PC berechnet werden können. Häufig gibt es sogar Lösungen, die sich in Echtzeit darstellen lassen. Diese Vorgehensweise ist legitim, weil es in der Unterhaltungsindustrie häufig nicht darauf ankommt, dass die Ergebnisse zu einhundert Prozent dem physikalischen Vorbild entsprechen. Vielmehr möchte man eine Lösung haben, die gut aussieht und eine möglichst kleine Rechenzeit besitzt.

Wenn man somit über gasförmige Phänomene in der Computergrafik spricht, dann müssen zwei grundsätzlich verschiedene Teilaspekte betrachtet werden. Erst einmal gilt es, die Bewegungen von Flüssigkeiten und Gasen zu simulieren. In diesem Gebiet ist seit 1999 eine große Weiterentwicklung zu verzeichnen, die insbesondere durch die Arbeit von Jos Stam [Stam 1999] eingeleitet wurde.

Der zweite Aspekt stellt das Rendering dar, also das Übersetzen der durch die Simulation gewonnenen Daten in ein digitales Bild. Hierbei ist vor allem die Forschung von Henrik Wann Jensen [Jensen und Christensen 1998] zu nennen.

Der Rest dieser Arbeit ist wie folgt aufgebaut: Im zweiten Abschnitt wird auf die dynamische Simulation, also die Berechnung der Bewegung von gasförmigen Phänomenen eingegangen. Das dritte Kapitel erläutert approximative Methoden zur Darstellung von Gasen. Der vierte Abschnitt beschreibt die physikalischen Hintergründe des Renderings und leitet die Volumen - Rendering - Gleichung her. Schlussendlich fasst Kapitel Fünf die Erkenntnisse zusammen und gibt einen Ausblick auf künftige Entwicklungen.

2 Dynamiksimulation von Flüssigkeiten & Gasen

Die Bewegung von Flüssigkeiten und Gasen gehorcht denselben physikalischen Gesetzen und ist damit - mathematisch gesehen - fast identisch. Wie wir später noch sehen werden, gilt diese Aussage auch für die optischen Eigenschaften dieser Objekte.

Flüssigkeiten und Gase bestehen aus einer Ansammlung von Teilchen, die sich, bei ihrer Bewegung gegenseitig beeinflussen. Im Fall von Wasser besteht die Flüssigkeit hauptsächlich aus H_2O -Molekülen, Luft besteht aus Stickstoff, Sauerstoff und weiteren Elementen und Rauch ist eine Ansammlung von vielen Staubpartikeln. Dabei ist die Bewegungsfreiheit eines einzelnen Partikels äußerst eingeschränkt und ordnet sich dem Fluss der anderen Teilchen unter. Der Einfluss eines Partikels auf seine

Umgebung ist jedoch auf die unmittelbaren Nachbarn beschränkt, die dann wieder auf ihre eigenen Nachbarn wirken usw. Die Bewegungsrichtung und -stärke ist somit stark orts- und zeitabhängig. Dadurch werden zum Beispiel Strudel möglich, die nur in einem kleinen Raumabschnitt wirken, während der Rest der Flüssigkeit oder des Gases eine ganz andere Bewegung ausführt. Diese Bewegung der Teilchen kann mathematisch durch die so genannte Navier – Stokes – Gleichung beschrieben werden.

Wie kommt es aber dann, dass sich zähflüssige Magma doch anders bewegt als Wasser? Die Antwort auf diese Frage ist einfach: Magma besitzt einfach andere Parameterwerte als Wasser. Das offensichtlichste Beispiel dürfte wohl die Viskosität des Mediums sein.

2.1 Die Navier - Stokes - Gleichung

Die Dynamik von Flüssigkeiten und Gasen – nachfolgend als Fluid bezeichnet – wird durch die so genannte Navier - Stokes - Gleichung beschrieben. Das strömende Medium wird dabei als eine Kombination aus einem zwei bzw. dreidimensionalen (Geschwindigkeits-)Vektorfeld \mathbf{u} und einem skalaren Druckfeld p beschrieben. Wenn wir zusätzlich annehmen, dass die Temperatur in dem Medium nahezu konstant ist, dann kann die Dynamik eines Fluids mit den folgenden zwei Gleichungen berechnet werden:

$$\nabla \circ \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \circ \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (2)$$

wobei ν die kinematische Viskosität des Fluids beschreibt und mit \mathbf{f} eine externe Kraft gemeint ist.

Es ist wichtig anzumerken, dass die Navier – Stokes – Gleichung sowohl die Masse (Gleichung (1)), als auch das Moment (Gleichung (2)) des Fluids erhält. Diese Arbeit hat sich nicht zum Ziel genommen, die obigen Formulierungen zu interpretieren, dafür sei auf die entsprechende Fachliteratur der Physik verwiesen.

Eine rein analytische Lösung ist auf Grund der Kompliziertheit nicht möglich. Die numerische Berechnung durch den Computer erfordert jedoch eine diskretisierte Beschreibung des Problems, die sich dann mit Standard – Lösungsverfahren berechnen lässt. Bei der Navier – Stokes – Gleichung wird das Fluid auf einem diskreten Gitter berechnet. Somit ist es zum Beispiel möglich das Geschwindigkeitsvektorfeld als eine N^2 - bzw. N^3 -dimensionale Datenstruktur (Abbildung 1) zu beschreiben. Die Darstellung des Dichtefelds erfolgt analog.

Diese Art der Beschreibung hat zudem den Vorteil, dass sie beim Rendering wieder verwendet werden kann. Denn wie wir noch sehen werden, wird dabei die Existenz eines Dichte- bzw. Partikelfelds vorausgesetzt, welches uns an jedem Punkt im Raum die vorhandene Flüssigkeits- bzw. Gasdichte angibt. Das bedeutet, dass die Simulation der Dynamik sehr gut mit dem Rendering kombiniert werden kann.

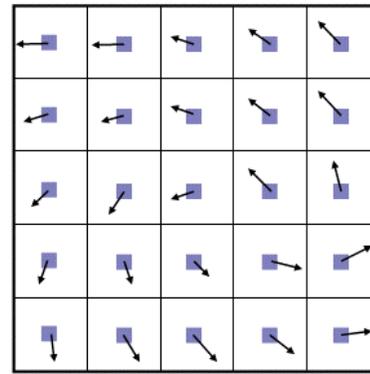


Abbildung 1. Ein diskretes Gitter.

2.2 Weitere Darstellungsmöglichkeiten

Neben der „realistischen“ Beschreibung durch die Navier – Stokes – Gleichung und einem diskreten Gitter existieren noch einige andere Darstellungsmöglichkeiten um gasförmige Objekte geometrisch zu beschreiben.

Gerade in den Anfängen der Computergrafik wurden häufig Turbulenzfunktionen oder so genannte implizite Oberflächenbeschreibungen (Blobs) verwendet. Die Dynamik musste in diesen Fällen allerdings per Hand erstellt werden, indem ein Grafiker mehr oder weniger intuitive Parameter manipulierte und dadurch eine Animation der Oberfläche beschrieb. In einer solchen Form kann natürlich nicht von einer realistischen Dynamik die Rede sein. Allerdings ist es möglich diese Art der Darstellung ebenfalls auf ein diskretes Gitter mit Dichtewerten abzubilden und damit bekannte Renderalgorithmen zu verwenden. Eine mögliche Anwendung wäre zum Beispiel für Objekte (Wolken etc.), die sich im Hintergrund befinden. Bei solchen Gebilden ist eine realistische Dynamik meist nicht so wichtig und auch beim Rendern können approximative Methoden benutzt werden. Insbesondere bei interaktiven Anwendungen wird fast ausschließlich eine direkte Partikelbeschreibung verwendet. Dabei wird die Dynamik von Fluidobjekten auf die Bewegungssimulation von einzelnen wenigen Teilchen beschränkt. Der Vorteil dieser Darstellungsform ist, dass es wesentlich weniger berechnungsintensiv ist als die Evaluation der Navier – Stokes – Gleichung. Auch das Rendering erweist sich als vereinfacht, denn es wird meist vollkommen auf physikalischen Realismus verzichtet. Auf das Rendering von Partikeln wird im dritten Abschnitt näher eingegangen.

3 Approximative Modelle

In diesem Abschnitt soll es um approximative Methoden zum Rendern von gasförmigen Phänomenen gehen. Diese sind in erster Linie auf die Echtzeitdarstellung ausgelegt. Modelle dieser Art sind hauptsächlich für interaktive Anwendungen interessant, wie zum Beispiel Computerspiele. Neuere Methoden greifen dabei fast ausschließlich auf das so genannte Partikelrendering zurück. Deswegen wird im nachfolgenden Abschnitt zunächst diese Darstellungsart beschrieben. Wie wir aber noch sehen werden, gibt es auch andere Möglichkeiten, die es ermöglichen, die Lösung der Navier – Stokes – Gleichung direkt zu visualisieren.

3.1 Partikelrendering

Wenn Fluide durch Partikel modelliert werden, dann wird das Phänomen durch eine Menge von Massepunkten beschrieben, die sich auf Grund von Kräfteinwirkungen durch den Raum bewegen. Ein Massepunkt repräsentiert ein Cluster von Fluidteilchen, von denen angenommen wird, dass sie alle gleichartig fließen.

Die einfachste Möglichkeit, diese Massepunkte zu visualisieren, ist die Partikel direkt als ein Punkt in einem Bild darzustellen. Der Nachteil dieser Vorgehensweise ist allerdings, dass eine große Anzahl von simulierten Fluidteilchen benötigt wird, um eine geschlossene und flächige Darstellung zu erzielen.

Um diesem Problem zu entgehen, werden oft so genannte Billboards verwendet, die an die Position des Partikels gesetzt werden. Ein Billboard besteht im einfachsten Falle aus zwei Dreiecken, die ein Rechteck bilden. Es können auch kompliziertere Objekte benutzt werden, aber aus Gründen der Performanz beschränkt man sich auf relativ einfache Geometrien.

Nachfolgend wollen wir davon ausgehen, dass Billboards zur Darstellung des Fluids verwendet werden.

Da ein (Teilchen-) Cluster lediglich durch ein Rechteck bzw. zwei Dreiecke repräsentiert wird, muss die Frage der Ausrichtung der Fläche - zur Visualisierung für den Beobachter - geklärt werden. Die einfachste Möglichkeit besteht darin, dem Billboard einfach eine feste Orientierung zu geben. Sollte sich allerdings die Kamera relativ zu dem Partikeln bewegen, kann es passieren, dass die von der Kamera aus gesehene Ausdehnung des gasförmigen Phänomens stark variiert und dann irgendwann gar nicht mehr zu sehen ist. Das geschieht genau dann, wenn die Orientierung der Kamera senkrecht zu der des Billboards ist. In diesem konkreten Fall betrachtet man somit die Rechtecke von der Seite aus. Da ein Billboard aber kein Volumen besitzt, kann es auch nicht mehr visuell erfasst werden.

In der Praxis werden deswegen häufig die Billboards anhand der Kamera ausgerichtet, so dass folgende Bedingung gilt:

$$\vec{v}_{\text{Kam}} \circ \vec{n}_{\text{BB}} = 1 \quad (3)$$

wobei mit \vec{v}_{Kam} der normierte Richtungsvektor der Kamera und mit \vec{n}_{BB} die Normale eines Billboards gemeint ist.

Intuitiv kann man Gleichung (3) folgendermaßen verstehen: Die Billboards, die die Partikel repräsentieren, sehen die Kamera stets direkt an.

Bis jetzt wurde allerdings noch nicht geklärt, wie ein Billboard dargestellt wird. Einfaches Gouraudshading der Dreiecke kann sicherlich nur recht eingeschränkt eine realistische Darstellung von gasförmigen Phänomenen ermöglichen. Eine gängige Methode stellt die Verwendung von transparenten Texturen dar, die mit Hilfe von Alpha – Blending übereinander gerendert werden. Dabei muss allerdings beachtet werden, dass die Partikel von hinten nach vorne gezeichnet werden, da sonst der Z-Pufferalgorithmus eine realistische Darstellung verhindert. Da eine Sortierung der Partikel - anhand ihres Abstands zur Kamera - aber relativ rechenaufwendig ist, existiert ein kleiner Trick, um sich diesen Schritt zu ersparen. Bei den modernen 3D – APIs ist es nämlich möglich, dass schreiben der aktuellen Tiefe in den Z-Puffer zu unterbinden. Ein Billboard, das sich sehr nah an der Kamera befindet, verdeckt dadurch keine anderen Partikel, die sehr viel weiter entfernt von der Kamera positioniert sind. Das eigentliche Blending wird davon nicht beeinflusst, somit können die Partikel in einer beliebigen Reihenfolge auf den Bildschirm gezeichnet werden, ohne dass es zu Flackereffekten kommt.

Nun soll noch ein kurzer Blick auf das eigentliche Alpha – Blending geworfen werden. Es existieren verschiedene Möglichkeiten, wie der Hintergrund mit dem aktuellen Billboard kombiniert werden kann. Dies geschieht im Rahmen einer einheitlichen Beschreibung, durch die es möglich wird, sowohl opake Körper, als auch Fluide (und weitere Objekte) durch dieselbe Gleichung darzustellen. Ganz grundsätzlich wird die Farbe eines Pixels anhand der folgenden Formel berechnet:

$$C_{\text{Res}}(\vec{x}) = C_{\text{Src}}(\vec{x}) \cdot B_{\text{Src}}(\vec{x}) + C_{\text{Dst}}(\vec{x}) \cdot B_{\text{Dst}}(\vec{x}) \quad (4)$$

Wobei mit C_{Res} die Ergebnisfarbe, mit C_{Src} und C_{Dst} die jeweilige Vorder- und Hintergrundfarbe gemeint ist. B_{Src} und B_{Dst} sind die so genannten Blendingparameter, die beschreiben, wie die zwei Operandenfarben (Vorder- und Hintergrundfarbe) verrechnet werden. Eine Farbe C wird dabei als ein Vektor mit vier Komponenten aufgefasst: $C = (\text{Rot}, \text{Grün}, \text{Blau}, \text{Alpha})$.

Formel (4) ist auch als die Blending - Gleichung bekannt. Sie beschreibt wie aus zwei Quellbildern ein neues komponiert wird. Im Rahmen der modernen Grafik – APIs wie OpenGL oder DirectX haben sich - für die wichtigsten Spezialfälle - die Abkürzungsbegriffe ONE, ZERO, SOURCE_ALPHA oder INVERSE_SOURCE_ALPHA in der Computergrafik eingebürgert. Diese Begriffe stehen jeweils für einen konkreten Farbvektor: So steht ONE für $(1,1,1,1)$ und ZERO für $(0,0,0,0)$. Ein wenig komplizierter ist SOURCE_ALPHA, das bedeutet nämlich: $(C.\alpha, C.\alpha, C.\alpha, C.\alpha)$. INVERSE_SOURCE_ALPHA steht für: $(1 - C.\alpha, 1 - C.\alpha, 1 - C.\alpha, 1 - C.\alpha)$. Die restlichen Blendingparameter erschließen sich analog.

Mit Hilfe der Blendinggleichung können jedoch nicht nur gasförmige Phänomene beschreiben werden. Vielmehr ist es möglich durch Formel (4) das Aussehen von allen Objekten beim Rendering zu beschreiben. Um einen opaken Körper darzustellen erhält B_{Src} einfach den Wert ONE und B_{Dst} ZERO. Intuitiv wird sofort klar, dass der Hintergrund nicht durch den Vordergrund zu sehen ist. Für ein lichtdurchlässiges Medium – und diese sind Thema dieser Arbeit – macht diese Kombination allerdings nicht viel Sinn, da man ja zum Beispiel durch einen Dunstschleier auch noch die Objekte sehen kann, die sich hinter oder im Dunst befinden.

Für ein Fluid mit einer hohen Lichtdurchlässigkeit (hoher Albedo), wählt man am Besten die Kombination ONE und ONE für die beiden Blendingparameter. Die Billboards werden somit einfach auf den Hintergrund aufaddiert.

Eine Wolke mit einem niedrigen Albedo, die also nur eine geringe Lichtdurchlässigkeit besitzt, wird sehr gut durch die Parameterwerte SOURCE_ALPHA und INVERSE_SOURCE_ALPHA repräsentiert. Je dichter das Fluid, desto höher ist der Alpha-Wert der Billboards, wodurch immer weniger vom Hintergrund zu sehen ist.

Abbildung 2 fasst verschiedene Kombinationen zusammen und veranschaulicht deren Wirkung im finalen Bild.

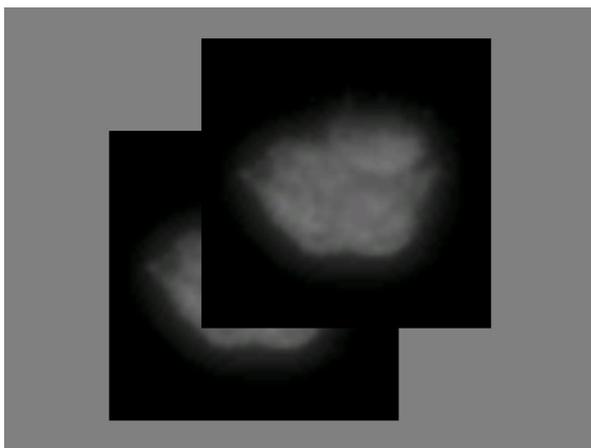


Abbildung 2.a) **Opake** Objekte: ONE, ZERO

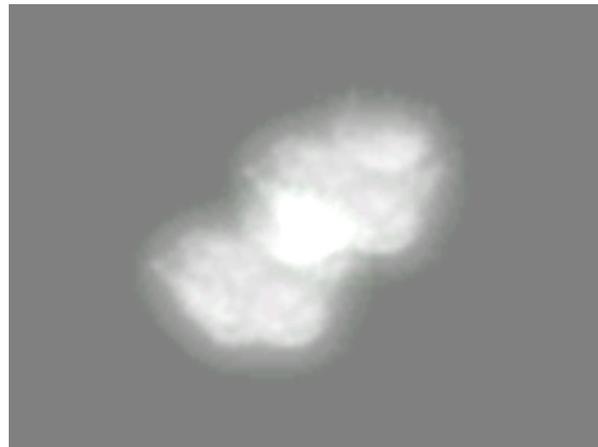


Abbildung 2.b) Einfaches Blending: ONE, ONE



Abbildung 2.c) Rauch: SOURCE_ALPHA, INVERSE_SOURCE_ALPHA

3.2 Slicing

Durch das Partikelrendering wurde es möglich eine dynamische und punktförmige Beschreibung eines gasförmigen Phänomens sehr effizient darzustellen. Dieser Ansatz kann allerdings nur sehr schwer mit anderen Repräsentationsformen kombiniert werden, wie zum Beispiel mit einem diskreten Dichtegitter. Das liegt vor allem daran, dass in einem solchen Gitter die Position der Samplepunkte fixiert ist, d.h. das Gitter bewegt oder verformt sich in der Regel nicht. Es wird somit ein anderes Verfahren benötigt, um die Repräsentation von Fluiden durch ein diskretes Dichtegitter grafisch darzustellen.

Eine Möglichkeit, die dieses Problem löst ist das so genannte Slicing. Dabei wird unser Gitter mit mehreren parallelen Ebenen geschnitten. Diese Schnittebenen werden Slices genannt, weil sie Scheiben der Daten repräsentieren. Wie beim Partikel-rendering mit Billboards, gilt es auch hier die Frage zu klären, welche Orientierung und Position diese Ebenen besitzen sollen. Im ersten Moment scheint der Ansatz, der bei dem Partikelsystem gemacht wurde auch an dieser Stelle zu funktionieren. Ganz grundsätzlich ist das auch richtig, aber es erweist sich als sehr unpraktisch, da

bei einer beliebigen Orientierung der Slices für jeden Vertex der Schnittebene eine trilineare Interpolation der nächsten acht Dichtewerte durchgeführt werden müsste. Diese Operation ist jedoch rechenintensiv und wird deshalb in der Regel nicht verwendet. Man assoziiert viel mehr jeden Vertex eines Slice mit einem Dichtesample im Fluidgitter. In diesem Fall sind aber lediglich drei Kombinationen von Sliceorientierungen sinnvoll - eine für jede Koordinaten-achse. Doch welche der drei Slicemengen soll für eine gegebene Ausrichtung der Kamera verwendet werden? Es wird die Menge i von Schnittebenen dargestellt, für die folgende Bedingung gilt:

$$(\vec{v}_{Kam} \circ \vec{e}_i) = \max$$

Wenn sich die Kamera relativ zum Gitter bewegt, dann kann es natürlich passieren, dass zu einem bestimmten Zeitpunkt ein Wechsel von einer Slicemenge zu einer anderen stattfindet. Diese ruckartige Änderung kann dem Betrachter negativ auffallen, deswegen ist es möglich die zwei Mengen ineinander überzublenzen. Allerdings müssen bei einem solchen Vorgehen mehr Dreiecke pro Bild dargestellt werden, was unter Umständen in einem langsameren Rendering endet.

Die Darstellung der Slices [Jensen et al 2001] erfolgt auf eine ähnliche Art und Weise wie bei dem Partikelrendering. Wieder nutzt man das Alphablending aus, um den Eindruck eines lichtdurchlässigen Mediums zu erzeugen. Allerdings werden diesmal keine Texturen verwendet. Vielmehr benutzt man direkt die Vertexfarben, die ja ebenfalls einen Alphawert enthalten können. Die jeweiligen Farben der konkreten Vertices der Slices werden anhand des diskreten Fluidgitters berechnet. Wie dies genau erfolgt ist vom Anwendungsfall abhängig. Ein einfacher Weg stellt die trilineare Interpolation der acht nächsten Dichtewerte dar. Je höher die Gesamtdichte nach der Interpolation ist, desto „heller“ wählt man die Vertexfarbe.

Beim Partikelrendering lag das Hauptaugenmerk auf der performanten Darstellung eines Fluids. Somit verzichtete man auch auf jegliche Art der Beleuchtung des Phänomens. Natürlich ist der erreichte Realismusgrad durch ein solches Vorgehen sehr beschränkt.

Beim Slicing wurde der Fokus jedoch stärker auf einen hohen Realismus verlagert. Somit sollten auch Beleuchtungseffekte mit berücksichtigt werden. Jedoch verwendet man dazu nicht die Grafikhardware, sondern einen Ansatz der in Software implementiert ist. Dieser Algorithmus funktioniert wie folgt: Wenn sich Photonen durch ein lichtdurchlässiges Medium bewegen, dann fällt deren Leuchtstärke exponentiell entlang des Weges ab. Dieses Verhalten kann man sich folgendermaßen erklären: Die Photonen stoßen innerhalb des Fluids gegen andere Teilchen und werden dadurch von ihrem Weg abgelenkt. Es bewegen sich also immer weniger Photonen entlang des ursprünglichen Wegs. Somit erreicht auch nur ein Bruchteil des ausgesandten Lichts das direkte Ziel. Diese Aussage ist auch als das Lambert – Beer – Gesetz bekannt.

Glücklicherweise lässt sich dies relativ einfach implementieren. Man schießt einen Lichtstrahl von jeder Lichtquelle zu jeder Zelle in unserem diskreten Fluidgitter. Um alle geschnittenen Voxel zu finden verwendet man einen 3D – Bresenham Algorithmus, der in dem Gitter eine diskrete (Licht-)Linie zeichnet. Je nach Dichte in den Zellen entlang des Weges wird die Intensität des Lichts abgeschwächt. In Abbildung 3 kann man diesen Ansatz recht gut nachvollziehen.

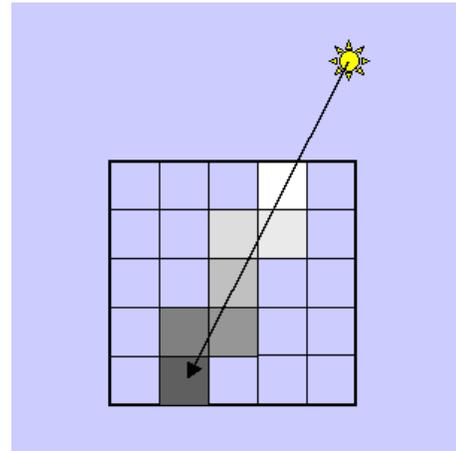


Abbildung 3. Diskrete Beleuchtung

Um die abnehmende Leuchtstärke zu berechnen, assoziiert man mit jedem Strahl einen Transparenzwert, der mit Eins initialisiert wird. Das Photon besitzt also - bildlich gesprochen – am Anfang die volle Leuchtstärke.

Auf dem Weg von der Lichtquelle zur Zielzelle wird dieser Wert schrittweise verringert. Jedes mal wenn der Strahl eine neue Zelle durchquert, sinkt die Transparenz proportional zur Dichte des aktuellen Voxels. Es ist einfach zu sehen, dass die Leuchtstärke des Photons umso kleiner wird, je mehr Dichte es auf seinem Weg passiert.

Die Helligkeit der Zielzelle wird anhand des Transparenzwertes berechnet, die der Strahl am Ende seines Weges noch besitzt. Je höher der Wert ist, desto heller ist die entsprechende Zelle. Hat ein Photon allerdings viel Dichte auf dem Weg zu einem Voxel durchquert, dann ist auch der Transparenzwert des Strahls relativ gering und somit wird die Farbe der Zelle auch sehr dunkel ausfallen.

In Pseudocode sieht der Algorithmus wie folgt aus:

```

Für jede Zelle X im Gitter:
  Für jede Lichtquelle L:

    Ray = Strahl von L nach X
    TRay = 1

    Für alle Zellen entlang Ray:
      TVox = exp( -h*d )
      LVox = a*LLight* (1-TVox) *TRay
      TRay = TRay * TVox
  
```

Wobei mit **T** jeweils immer die Transparenz und mit **L** die Radianz (Leuchtstärke) gemeint ist. Der Faktor **h** repräsentiert die Größe einer Zelle, **d** gibt die aktuelle Fluiddichte an und mit **a** ist der Albedo gemeint. Ein hoher Albedo steht für ein helles Gas. Wasserdampf und Wolken sind zwei Beispiele für Medien mit einem relativ hohen Albedo. Dementsprechend repräsentiert ein niedriger Albedo ein dunkles Fluid, also zum Beispiel Rauch. In Abbildung 4 ist zu sehen wie eine entsprechende Implementierung aussehen könnte. Dabei sind die einzelnen Slices sehr gut zu sehen. Trotzdem wirkt das Ergebnis „voluminös“.

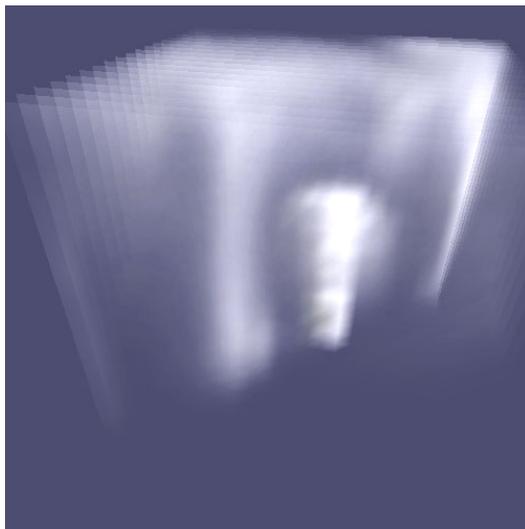


Abbildung 4. Slicing in Aktion.

4 Photorealistische Modelle

Absolut photorealistische Bilder künstlich zu erzeugen setzt voraus, dass man sich zuerst mit den physikalischen Gegebenheiten auseinandersetzt, die in Flüssigkeiten und Gasen vorherrschen. Wie wir bald sehen werden, können die optischen Eigenschaften eines Fluids durch die so genannte Volumen – Rendering – Gleichung physikalisch korrekt beschrieben werden. Dadurch wird es dann auch möglich, das gesamte Spektrum von Flüssigkeiten und Gasen darzustellen. Extrem realitätsnahe Bilder werden zum Beispiel in der Filmproduktion benötigt, schließlich sollte der Zuschauer möglichst nicht erkennen, dass die entsprechenden Effekte aus dem Computer stammen.

4.1 Die Volumen - Rendering – Gleichung

Aus einem physikalischen Standpunkt heraus besteht ein Fluid aus einer Menge von einzelnen Teilchen. Es ist dabei egal ob es sich um Makropartikel - wie zum Beispiel Rauchteilchen - handelt, die zwischen einigen Millimetern und Zentimetern groß sind oder um Mikropartikel, die lediglich aus einigen Molekülen bestehen. Wichtig für die Herleitung der Volumen – Rendering – Gleichung ist allerdings, dass im Raum eine Funktion definiert werden kann, die in Abhängigkeit des Orts ($\rho(\vec{x})$) eine momentane Partikeldichte angibt. Natürlich ist es möglich, diese Funktion in unserem diskreten Gitter zu definieren, indem zum Beispiel die acht nächsten Dichtewerte trilinear interpoliert werden.

Weiterhin sind eine Reihe von optischen Parametern zu definieren. Der so genannte Absorptionsquerschnitt σ_a beschreibt wie viel Licht von den Fluidteilchen absorbiert wird. Diese Energie geht dabei nicht verloren, sondern wird in eine andere Energieform transformiert, zum Beispiel in Wärmeenergie.

Auf Basis des Absorptionsquerschnitts kann der Absorptionskoeffizient bestimmt werden:

$$\mathbf{a}(\vec{x}) = \sigma_a \cdot \rho(\vec{x}). \quad (5)$$

Dieser Koeffizient gibt an, wie viel Licht an jedem Punkt im Raum vom Fluid absorbiert wird. Analog zu dem eben genannten Koeffizient wird der Scatterquerschnitt σ_s eingeführt, der angibt wie viel Licht von dem Medium gestreut wird. Er beschreibt also quasi die optische Rauheit des Mediums. Somit kann der Scatterkoeffizient folgendermaßen definiert werden:

$$\mathbf{s}(\vec{x}) = \sigma_s \cdot \rho(\vec{x}). \quad (6)$$

Diese beiden Effekte werden häufig zum so genannten Auslöschkoeffizient (Extinction) zusammengefasst:

$$\mathbf{e}(\vec{x}) = \mathbf{a}(\vec{x}) + \mathbf{s}(\vec{x}). \quad (7)$$

Auf Basis dieser Grundlagen kann die Volumen – Rendering – Gleichung hergeleitet werden. Dabei wird die Änderung der Radianz entlang eines kleinen Wegabschnitts $d\mathbf{s}$ betrachtet. Wir wollen dabei annehmen, dass das lichtdurchlässige Medium entlang dieses Wegabschnitts homogen ist, d.h. es finden keinerlei Änderungen der Dichte oder anderer optischer Parameter statt. In Abbildung 5 ist dies schematisch dargestellt.

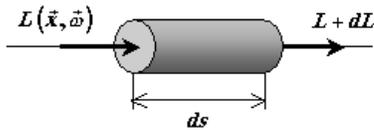


Abbildung 5. Ein Volumenstück des Mediums.

Vor der Durchquerung von $d\mathbf{s}$ soll die Radianz \mathbf{L} vorherrschen und danach $\mathbf{L} + d\mathbf{L}$. Nun gilt es also die Änderung der Radianz $d\mathbf{L}$ zu bestimmen. Durch die Absorption von Licht wird die Radianz verringert:

$$d\mathbf{L} = -\mathbf{a}(\vec{x}) \cdot \mathbf{L}(\vec{x}, \vec{\omega}) \cdot d\mathbf{s}. \quad (8)$$

Neben der Absorption „verliert“ das Fluid auch noch Radianz durch das so genannte Outscattering. Dabei werden Photonen in eine beliebige Richtung aus dem Zylinder gestreut. Die Radianz dieser Photonen ist jedoch am Ende des Wegabschnitts nicht mehr vorhanden, somit gilt:

$$d\mathbf{L} = -\mathbf{s}(\vec{x}) \cdot \mathbf{L}(\vec{x}, \vec{\omega}) \cdot d\mathbf{s}. \quad (9)$$

Abbildung 6 veranschaulicht das Outscattering grafisch.

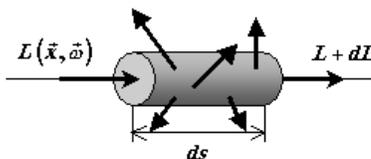


Abbildung 6. Gestreutes Licht.

Da die Absorption und das Outscattering die Radianz verringern, können sie zur Auslöschung zusammengefasst werden. Die zwei Formeln (8) und (9) werden zu der folgenden Formulierung kombiniert:

$$d\mathbf{L} = -\mathbf{e}(\vec{x}) \cdot \mathbf{L}(\vec{x}, \vec{\omega}) \cdot d\mathbf{s}. \quad (10)$$

Es existieren aber auch Effekte, die die Radianz entlang des Weges wieder erhöhen. Die Emission stellt das Gegenstück der Absorption dar und kann mathematisch durch die folgende Gleichung beschrieben werden:

$$d\mathbf{L} = \mathbf{a}(\vec{x}) \cdot \mathbf{L}_e(\vec{x}, \vec{\omega}) \cdot d\mathbf{s}. \quad (11)$$

Wie man sieht ist die Ähnlichkeit zur Absorption auch mathematisch vorhanden. \mathbf{L}_e steht in diesem Kontext für das Eigenleuchten des Fluids.

Ein wenig komplexer ist das Gegenstück zum Outscattering. Es ist natürlich sofort klar, dass wenn Licht den betrachteten Zylinder in eine beliebige Richtung verlässt, es dadurch nicht spurlos verschwindet, sondern in der Nachbarschaft wieder eingestreu wird. Dies wird durch

$$d\mathbf{L} = \mathbf{s}(\vec{x}) \cdot \int_{\Omega} \mathbf{P}(\vec{x}, \vec{\omega}, \vec{\omega}') \cdot \mathbf{L}(\vec{x}, \vec{\omega}') d\vec{\omega}' \cdot d\mathbf{s} \quad (12)$$

beschrieben.

Da beim Outscattering das Licht den Zylinder in eine beliebige Richtung verlässt, muss beachtet werden, dass es somit auch aus einer beliebigen Richtung wieder eingestreu wird. In der Formel (12) ist dieser Effekt durch das Integral über die gesamte Einheitskugel inbegriffen. Die einfallende Radianz wird allerdings noch durch die so genannte Phasenfunktion \mathbf{P} gewichtet. Diese Funktion stellt eine Wahrscheinlichkeitsverteilung über der Einheitskugel dar, die folgendes bedeutet: Wenn Licht unter einem Winkel $\vec{\omega}'$ in unseren Zylinder eingestreu wird, wie wahrscheinlich ist es dann, dass sich dieses Licht in die Richtung $\vec{\omega}$ weiterbewegt und somit unsere Ursprungsradianz \mathbf{L} erhöht. (Siehe Abbildung (5).)

Da die Phasenfunktion ein sehr wichtiges Element der Volumen – Rendering- Gleichung ist, soll sie nachfolgend noch ein wenig genauer untersucht werden.

\mathbf{P} ist eine Wahrscheinlichkeitsverteilung, deswegen ist sie auch über der Einheitskugel normalisiert:

$$\int_{\Omega} \mathbf{P}(\vec{\omega}, \vec{\omega}') d\vec{\omega}' = \mathbf{1}. \quad (13)$$

Mit anderen Worten: In unseren Zylinder wird nicht mehr Radianz eingestreu, als überhaupt aus der Umgebung kommt. Auf Basis dieser Funktion sind die unterschiedlichsten Verteilungen denkbar. Man trennt sie in die Kategorien Isotrop, Vorwärts - Scattering, Rückwärts – Scattering und Hybride Verteilungen. In einem isotropen

Material wird das Licht aus allen Richtungen gleichmäßig stark eingestreu. Ein typisches Beispiel für Vorwärts – Scattering ist Wasserdampf. Wenn zum Beispiel ein Photon auf einen Wassertropfen trifft, dann wird dieses in die Richtung weitergestreut, in die es sich sowieso schon vorher bewegte. Dies liegt vor allem daran, dass Wassertropfen lichtdurchlässig sind, aber auf Grund der Refraktion kann es eben zu einer Richtungsänderung kommen. Im Gegensatz dazu sind Staubteilchen in einer Rauchwolke ein recht gutes Beispiel für ein Rückwärts – Scattering – Medium. Wenn Licht auf ein Staubpartikel auftrifft wird es von diesem reflektiert, damit bewegen sich die Photonen aber genau in die entgegengesetzte Richtung, aus der sie ursprünglich kamen. Die Staubpartikel sind nur selten lichtdurchlässig.

Mit diesem Wissen kann man sich daran machen, diese Effekte in mathematische Formeln umzusetzen. Durch das Scattering geht Licht nie „verloren“, es kann aber in eine beliebige Richtung gestreut werden. Somit ist es möglich das Streuverhalten als eine Wahrscheinlichkeitsfunktion über der Sphäre zu definieren. Dabei wird jedem Richtungspaar ein Wert zwischen Null und Eins zugeordnet. Dieser Wert gibt an, wie wahrscheinlich es ist, dass ein Photon aus der (eingehenden) Richtung $\vec{\omega}$ in die (ausgehende) Richtung $\vec{\omega}'$ gestreut wird.

Das Isotrope Scattering kann durch eine einfache Gleichverteilung beschrieben werden:

$$P(\vec{\omega}, \vec{\omega}') = \frac{1}{4\pi}. \quad (14)$$

In der Computergrafik hat sich ganz besonders stark die so genannte Henvey – Greenstein – Phasenfunktion etabliert. Sie wurde ursprünglich für die Astronomie entwickelt [Henvey und Greenstein 1941] und repräsentiert lediglich empirische Untersuchungen. Auf Grund ihrer Einfachheit und der Tatsache, dass sie über einen Asymmetrieparameter kontrolliert werden kann, ist sie dennoch die am häufigsten verwendete Phasenfunktion.

$$P(\vec{\omega}, \vec{\omega}') = \frac{1 - g^2}{4\pi \cdot (1 + g^2 - 2 \cdot g \cdot [\vec{\omega} \circ \vec{\omega}'])^2}. \quad (15)$$

g ist der oben genannte Asymmetrieparameter. Sein Wertebereich umfasst das offene Intervall $] -1, 1 [$

Der Parameter g hat dabei die folgende Bedeutung: Für $g=0$ ist P eine Isotrope Phasenfunktion. Für Werte größer Null repräsentiert sie das Vorwärts – Scattering und bei Werten kleiner Null das Rückwärts – Scattering. Tatsächlich kann man die Henvey – Greenstein - Phasenfunktion sogar noch weiter vereinfachen und damit Rechenzeit sparen.

Bevor wir endgültig die Volumen – Rendering – Gleichung herleiten wollen, müssen wir allerdings zuvor noch eine Eigenschaft von Fluiden analysieren. Bis jetzt gingen wir ja davon aus, dass unser betrachtetes Medium entlang des Weges homogen ist. Wir müssen uns allerdings von dieser Einschränkung lösen, anderenfalls könnten wir nur ein relativ kleines Spektrum von Fluiden beschreiben. Die meisten Flüssigkeiten und Gase sind inhomogen, d.h. es variiert zum Beispiel die Dichte oder andere Parameter wie Farbe, Absorption usw. im Medium. Um solche Effekte zu beschreiben, zerlegt man das lichtdurchlässige Medium in kleine Volumina, bei denen man dann aber davon ausgehen kann, dass unser Fluid in diesem Raumabschnitt homogen ist. Wenn über alle diese Teile integriert wird, ergibt sich wieder das ursprüngliche Medium. Dieses Vorgehen soll am Beispiel der so genannten optischen Tiefe erläutert werden. Sie beschreibt, wie viel Radianz entlang eines inhomogenen Weg-abschnitts „verloren geht“, also welcher Teil absorbiert oder gestreut wird. Wir gehen davon aus, dass entlang eines Strahls lediglich der oben definierte Auslöschkoeffizient (vgl. Formel (7)) variiert, der die Absorption und Streuung zusammenfasst. Dabei betrachten wir die Änderung dieses Koeffizienten entlang des Strahls, der die zwei Punkte s und s' verbindet. Dieser Wegabschnitt wird in unendlich viele kleine Teilabschnitte zerlegt, die jeweils die Länge dt besitzen. Wir können folgendermaßen über den gesamten Weg integrieren:

$$\tau(s, s') = \int_s^{s'} e^{(\vec{x} + t \cdot \vec{\omega})} dt. \quad (16)$$

Abbildung 7 visualisiert dieses Vorgehen.

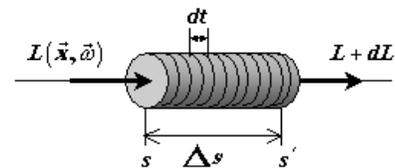


Abbildung 7. Die optische Tiefe.

Da wir (16) aber nicht so einfach analytisch lösen können, zerlegen wir den Weg in endlich viele Teile, die jeweils dt lang sind und addieren den Auslöschkoeffizienten dementsprechend auf. Genau so werden wir dann später auch die Volumen – Rendering – Gleichung im Rahmen des so genannten Raymarchings lösen.

Doch zurück zur Volumen – Rendering – Gleichung: Wenn wir die Formeln (10), (11) und (12) kombinieren, dann kommen wir zu der folgenden Formulierung:

$$(\vec{\omega} \circ \nabla) \mathbf{L}(\vec{x}, \vec{\omega}) = \mathbf{a}(\vec{x}, \vec{\omega}) \cdot \mathbf{L}_e(\vec{x}, \vec{\omega}) - \mathbf{e}(\vec{x}, \vec{\omega}) \cdot \mathbf{L}(\vec{x}, \vec{\omega}) + \mathbf{s}(\vec{x}, \vec{\omega}) \cdot \int_{4\pi} \mathbf{P}(\vec{x}, \vec{\omega}, \vec{\omega}') \cdot \mathbf{L}(\vec{x}, \vec{\omega}') d\vec{\omega}' \quad (17)$$

Gleichung (17) fasst lediglich das zusammen, was wir bereits oben Schritt für Schritt hergeleitet haben. Es soll jedoch noch kurz erläutert werden, wie die linke Seite von Gleichung (17) zu verstehen ist. Das Ergebnis von Gl. (17) ist die Änderung (∇) der Radianz ($\mathbf{L}(\vec{x}, \vec{\omega})$) entlang einer beliebigen Richtung ($\vec{\omega}$) im Raum.

Um nun endgültig die Volumen – Rendering – Gleichung herzuleiten, muss man (17) entlang des Weges integrieren. Mit einigen zusätzlichen Umformungen erhalten wir die vollständige Volumen – Rendering – Gleichung:

$$\mathbf{L}(\vec{x}, \vec{\omega}) = e^{-\tau(\vec{x}, \vec{x} + \vec{s} \cdot \vec{\omega})} \cdot \mathbf{L}(\vec{x} + \vec{s} \cdot \vec{\omega}, \vec{\omega}) + \int_0^s e^{-\tau(\vec{x}, \vec{x}')} \cdot \mathbf{a}(\vec{x}, \vec{\omega}) \cdot \mathbf{L}_e(\vec{x}, \vec{\omega}) d\vec{x}' + \left\{ \int_0^s e^{-\tau(\vec{x}, \vec{x}')} \cdot \mathbf{s}(\vec{x}, \vec{\omega}) \cdot \int_{4\pi} \mathbf{P}(\vec{x}, \vec{\omega}, \vec{\omega}') \cdot \mathbf{L}(\vec{x}, \vec{\omega}') d\vec{\omega}' d\vec{x}' \right\} \quad (18)$$

Auf den ersten Blick mag (18) sehr kompliziert aussehen, es kann allerdings eine recht einfache intuitive Erklärung dafür gefunden werden. Dies soll an dem zweiten der drei Teilterme in der Gleichung (18) erläutert werden, welcher das Eigenleuchten des Fluids repräsentiert.

Wie bereits weiter oben erläutert wurde, muss ein inhomogenes Stück Medium in viele kleine homogene Teile zerlegt werden, dieses Vorgehen wird durch das Integral über den Weg dargestellt. In jedem Abschnitt dieses Weges wird das Eigenleuchten des Mediums aufaddiert, allerdings wird es jeweils durch $e^{-\tau(\vec{x}, \vec{x}')}$ gewichtet. Es ist ganz klar, dass von einem Punkt der sich nur einen Millimeter innerhalb des Mediums befindet mehr Licht zum Betrachter kommt, als von einem weiteren Punkt, der zum Beispiel einhundert Meter Abstand zum Betrachter besitzt. Denn das Licht, welches an dieser Stelle emittiert wird, muss ja schließlich noch den gesamten Weg - also einhundert Meter - durch das Medium zurücklegen, bis es überhaupt zum Betrachter kommt. Dabei treten natürlich wieder Absorption und Outscattering auf, die die Energie des emittierten Lichts verringern. Und genau diese Aussage wird durch $e^{-\tau(\vec{x}, \vec{x}')}$ repräsentiert. Die anderen Teilterme erschließen sich analog.

Trotz dieser einfachen intuitiven Erklärung handelt es sich bei (18) um eine rekursive Integralgleichung, für die keine analytische Lösung möglich ist. Es werden also numerische Ansätze benötigt.

4.2 Punktweise Lösung: Raymarching

Ähnlich wie bei der Rendering – Gleichung existieren unterschiedliche Möglichkeiten um Formel (18) zu lösen. Durch das so genannte Raymarching wird eine Punktweise Lösung der Volumen – Rendering – Gleichung durchgeführt. Dabei wandert man in kleinen Schritten durch die Szene und löst für jeden Schritt die Volumen – Rendering – Gleichung. Abbildung 8 veranschaulicht dieses Vorgehen.

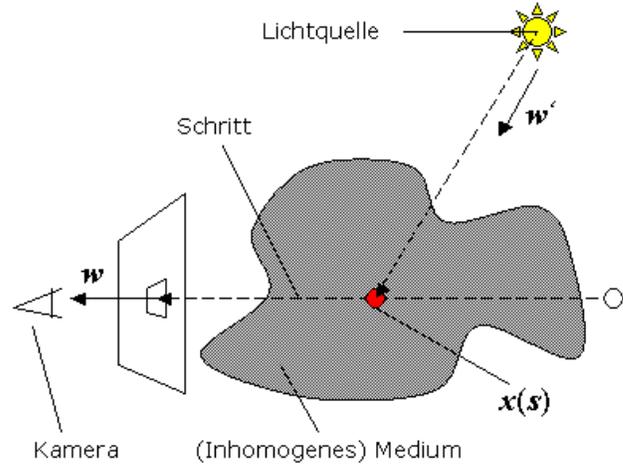


Abbildung 8. Raymarching

Die Volumen – Rendering – Gleichung beschreibt natürlich alle Beleuchtungseffekte, die innerhalb eines solchen lichtdurchlässigen Mediums auftreten können. Auf Grund der Einfachheit wollen wir für einen Moment mal annehmen, dass keine indirekte Beleuchtung auftritt, d.h. unser Fluid wird lediglich durch Lichtquellen direkt beleuchtet.

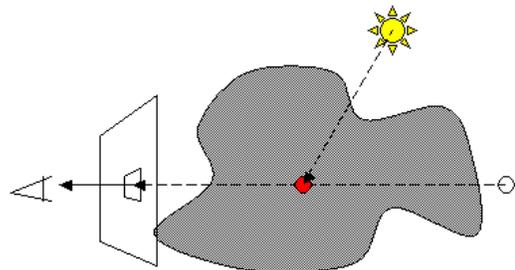


Abbildung 9. Direkte Beleuchtung.

In diesem Fall kann Formel (18) vereinfacht werden:

$$\mathbf{L}_{n+1}(\vec{x} + \Delta \mathbf{s} \cdot \vec{\omega}, \vec{\omega}) = \exp(-\mathbf{e}(\vec{x}) \cdot \Delta \mathbf{s}) \cdot \mathbf{L}_n(\vec{x}, \vec{\omega}) + \sum_I^{\text{all Lights}} \mathbf{L}_I(\vec{x}, \vec{\omega}') \cdot \mathbf{P}(\vec{\omega}, \vec{\omega}') \cdot \mathbf{s}(\vec{x}) \cdot \Delta \mathbf{s} \quad (19)$$

Die Formel (19) kann auch wieder intuitiv interpretiert werden. In dem ersten Teilterm wird zunächst die Radianz, die im vorherigen Schritt berechnet wurde, anhand des aktuellen Auslöschungsfaktors verringert. Schließlich wird durch den zweiten Term der Anteil aller Lichtquellen der Szene hinzuaddiert. Da unser Medium inhomogen ist, muss allerdings ein zweiter Raymarchingvorgang durchgeführt werden, um die eingehende Radianz der Lichtquellen zu berechnen. Somit wandert man nicht nur von einem Anfangspunkt zur Kamera, sondern auch von jeder Lichtquelle zu dem aktuell betrachteten Punkt im Medium. Nun wollen wir eine große Einschränkung beseitigen, nämlich dass wir lediglich direkte Beleuchtung berücksichtigen. In diesem Fall müssen allerdings sehr viel mehr Lichtwege berücksichtigt werden. Durch die Teilchen des Fluids werden Photonen gestreut. Dadurch wird es möglich, dass zwei Photonen, die von der selben Lichtquelle stammen, sich in gänzlich verschiedenen Richtungen bewegen. Wenn wir also einen konkreten Punkt im Fluid betrachten, dann kommt das einfallende Licht nicht nur direkt von der Quelle, sondern auch aus allen anderen Richtungen. Abbildung 10 zeigt ein Beispiel.

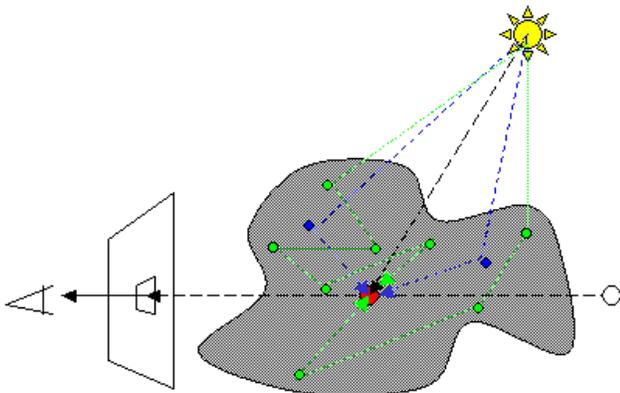


Abbildung 10. Indirekte Beleuchtung.

Wie man sehen kann sind Pfade der unterschiedlichsten Komplexität denkbar, die aber unseren aktuell betrachteten Punkt mit erhellen. Die blauen Pfade stellen das so genannte Single Scattering dar. Es beschreibt die Photonen, die ursprünglich nicht in die Richtung des momentanen Schritts ausgesandt wurden, dann aber so gestreut wurden, so dass sie sich schließlich doch noch in die Richtung des aktuellen Schritts bewegen. Die grünen Pfade zeigen das Multi Scattering, also Lichtwege beliebiger Komplexität. Diesen indirekten Lichtanteil zu berechnen benötigt eine Erweiterung von (19).

$$\begin{aligned}
 L_{n+1}(\vec{x} + \Delta s \cdot \vec{\omega}, \vec{\omega}) &= \exp(-\mathbf{e}(\vec{x}) \cdot \Delta s) \cdot L_n(\vec{x}, \vec{\omega}) + \\
 &\left\{ \frac{1}{M} \sum_{m=1}^M L_m(\vec{x}, \vec{\omega}_m) \cdot \mathbf{P}(\vec{\omega}, \vec{\omega}_m) \right\} \cdot \mathbf{s}(\vec{x}) \cdot \Delta s + \\
 &\sum_I^{all\ Lights} L_I(\vec{x}, \vec{\omega}') \cdot \mathbf{P}(\vec{\omega}, \vec{\omega}') \cdot \mathbf{s}(\vec{x}) \cdot \Delta s. \quad (20)
 \end{aligned}$$

Um die indirekte Beleuchtung zu berechnen, verwendet man M Strahlen, die gleichmäßig in alle Richtungen ausgesandt werden. Allerdings muss für diese Strahlen wieder ein Raymarching ausgeführt werden, wobei letztendlich erneut die Gleichung (20) entlang dieses neuen Weges berechnet werden muss. Wie man sich sicherlich vorstellen kann, ist diese hochgradig rekursive Berechnung sehr aufwendig, weil eine sehr große Anzahl von Lichtpfaden berücksichtigt werden muss.

Es ist jedoch wichtig zu erwähnen, dass der Einfluss der indirekten Beleuchtung niemals unterschätzt werden sollte. In der Abbildung 11 kann man einmal den Vergleich zwischen unterschiedlichen Berechnungstiefen sehen.

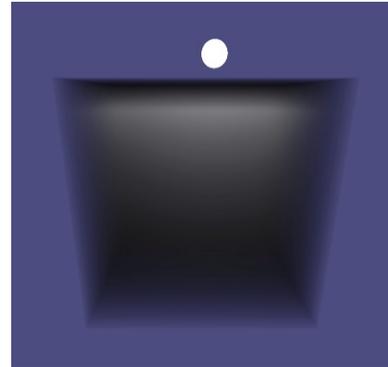


Abbildung 11.a) Direkte Beleuchtung.

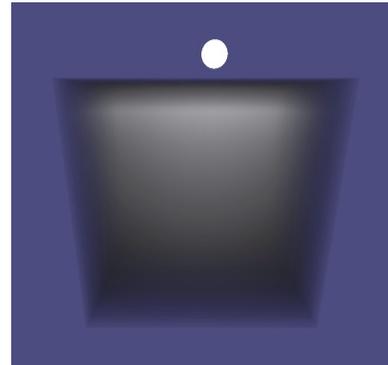


Abbildung 11.b) Einmaliges Scattering.

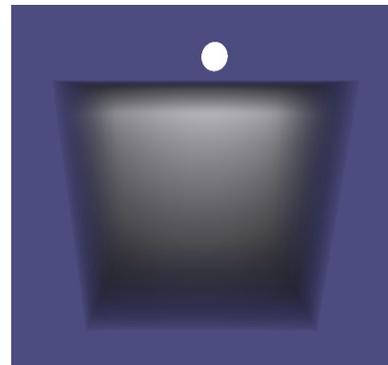


Abbildung 11.c) Zweimaliges Scattering.

Kombiniert man das Raymarching mit einem „gewöhnlichen“ Monte – Carlo – Raytracer, dann ist man in der Lage fast alle Effekte zu beschreiben, die in der Realität auch auftreten können. Wie bei allen Varianten des Raytracings, leidet auch das Raymarching an einigen Problemen. Um Bilder zu synthetisieren, die nur wenig Rauschen enthalten, werden sehr viele Strahlenanfragen nötig, was wiederum zu einer sehr hohen Renderdauer führt. Mit anderen Worten, dieses Verfahren besitzt eine schwache Konvergenz.

Glücklicherweise können aber fast alle Verbesserungen des Raytracings auch auf das Raymarching angewendet werden, wodurch auch mit relativ wenigen Sichtstrahlen eine schnelle Konvergenz des Ergebnisses erreicht werden kann. Als Beispiel sollen hier nur die Verfahren Metropolis – Light – Transport, Pathtracing oder Bidirectional – Raytracing genannt werden. In diesen Verfahren werden (optimale) Lichtpfade von den Lichtquellen zur Kamera (und anders herum) gesucht.

In Abbildung 12 sind Bilder zu sehen, die durch einen - um das Raymarching erweiterten – Standard-Raytracer [Franzke 2003] berechnet wurden.

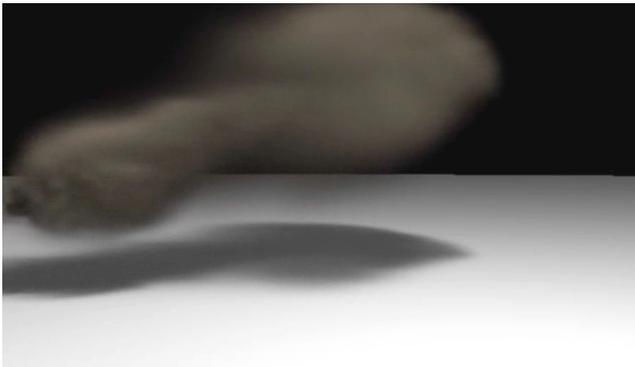


Abbildung 12.a) Raymarching Beispiel 1

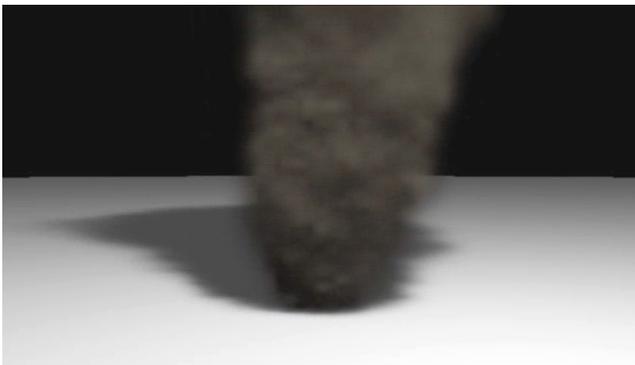


Abbildung 12.b) Raymarching Beispiel 2

4.3 Weitere Lösungsmöglichkeiten

Neben dem Raymarching existieren noch einige weitere Möglichkeiten, die Volumen – Rendering – Gleichung zu lösen. Als erstes sei dabei der Finite – Elemente – Ansatz zu nennen, der im Grunde dem Radiosity bei der Rendering Gleichung entspricht. Dabei wird die Gleichung (18) in eine Matrixrepräsentation überführt und dann mit Hilfe einer Relaxierung gelöst. Diese Herangehensweise hat natürlich dieselben Vor- und Nachteile wie das Radiosity. Die Matrix ist häufig sehr groß, insbesondere dann, wenn man sehr große Phänomene beschreiben möchte. Daraus folgen gleich zwei negative Eigenschaften. Erst einmal wird sehr viel Speicherplatz für die Matrix benötigt. Dies kann sogar so weit gehen, dass die Matrix nicht einmal mehr vollständig in den Arbeitsspeicher eines Supercomputers passt.

Auf der anderen Seite dauert die Relaxierung um so länger, je größer die Matrix ist. Außerdem hat der Finite – Elemente – Ansatz den Nachteil, dass die Lösung der Volumen – Rendering – Gleichung bei jeder noch so kleinen Änderung des Fluids wiederholt werden muss. Das heißt, dieses Verfahren ist insbesondere schlecht für dynamische Szenen geeignet. Nun sind aber Flüssigkeiten und Gase der Inbegriff der Dynamik, das heißt erst ihre Bewegung lässt sie wirklich realistisch erscheinen. Eine Wasseroberfläche ohne Wellen lässt sich in der Natur wohl kaum finden. Damit ist die Finite – Elemente – Methode nicht wirklich geeignet, um die Volumen – Rendering – Gleichung zu lösen. Sie wird in der Praxis deswegen auch nicht verwendet.

Eine weitere Möglichkeit, Formel (18) zu lösen, besteht darin, ein Mehr – Pass – Raytracing [Jensen und Christensen 1998] zu verwenden. Im ersten Pass verfolgt man Photonen von den Lichtquellen aus durch die Szene. Tritt ein Photon in ein lichtdurchlässiges Medium ein, dann wird auf Basis von stochastischen Verfahren ein Streueffekt berechnet. An jedem Punkt (Scatterpunkt), wo ein solches Ereignis auftritt wird ein Photon in eine räumliche Datenstruktur eingefügt. (In der Praxis wird dafür häufig ein kd – Baum verwendet, weil dieser sich sehr gut für Range – Search - Aufgaben eignet.) Entlang des Weges berücksichtigt man zusätzlich den Energieabfall der Photonen, in Abhängigkeit zu dem durchquerten Medium. Abbildung 13 veranschaulicht den ersten Pass.

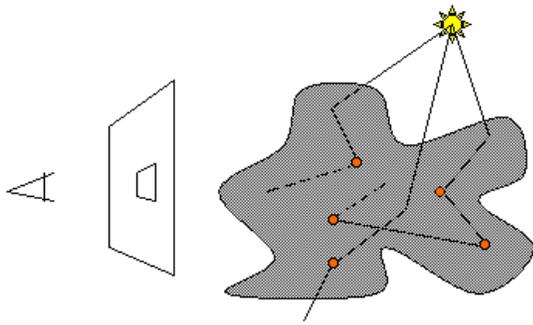


Abbildung 13. Pass 1: Photonverfolgung

Der zweite Pass besteht nun lediglich darin, ein einfaches Raymarching durchzuführen. Dabei sammelt man eine gewisse Anzahl von Photonen um den aktuell betrachteten Punkt ein, die letztendlich die indirekte Beleuchtung repräsentieren. Dies ist in Abbildung 14 zu sehen.

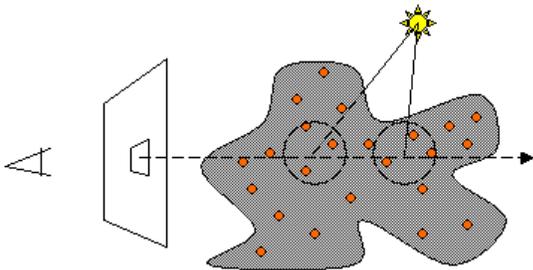


Abbildung 14. Pass 2: Raymarching

Der Vorteil von diesem Verfahren liegt natürlich auf der Hand: Man erspart sich den aufwendigsten Teil von Gleichung (20), nämlich die Berechnung der indirekten Beleuchtung. Stattdessen muss man lediglich eine gewisse Anzahl von Photonen in der Nähe unseres momentanen Schritts finden. Dies geht im Normalfall sehr viel schneller. Man kann sogar noch weitere Vereinfachungen einführen. Zum Beispiel kann man die Anzahl der Photonen, die man in der räumlichen Datenstruktur sucht, in Abhängigkeit des Abstands zum Betrachter reduzieren. Das heißt je tiefer sich der Raymarcher in dem Medium befindet, desto ungenauer berechnet man die indirekte Beleuchtung. Dieses so genannte Two – Pass – Raytracing wird auch tatsächlich in der Praxis verwendet. Einige Renderer die es einsetzen sind zum Beispiel mental ray© von mental images oder FinalRender© von cebas.

5 Zusammenfassung und Ausblick

Mittlerweile existieren sehr viele unterschiedliche Ansätze, um Flüssigkeiten und Gase zu simulieren und darzustellen. Von sehr einfachen Vorgehensweisen, wie das Partikelrendering bis hin zu hyperrealistischen Methoden, wie zum Beispiel ein Navier – Stokes – Löser im Zusammenspiel mit dem Raymarching. Je nach Qualitätsanforderungen oder Performanzentscheidungen kann man den Weg wählen, der am geeigneten erscheint. Es ist auch ein

verbesserter Workflow denkbar, indem mit einer approximativen Methode zunächst eine schnelle Vorschau des Ergebnisses gegeben wird. Beim Rendering wird das Bild dann schließlich mit einer hochqualitativen Vorgehensweise berechnet.

In Zukunft werden sich wohl - gerade im Bereich der Simulation der Dynamik von Fluiden – hybride Methoden durchsetzen. Außerdem geht die Forschung im Bereich der Steuerung und Animierbarkeit voran. Einige Arbeit sollte noch in die Interaktion der Fluide mit der Welt gesteckt werden. Die aktuellen Lösungen sind mitunter mehr als unbefriedigend.

Im Bereich des Renderings geht die Forschung vor allem im Rahmen der Performanzverbesserung und der Reduzierung von Rauschen voran. Außerdem gibt es sicherlich noch einiges zu tun, um das so genannte temporale Aliasing zu verringern. Dies tritt insbesondere bei Algorithmen auf, die sich in irgendeiner Art und Weise auf stochastische Werte beziehen. Es kann nämlich nicht garantiert werden, dass in zwei aufeinander folgenden Bildern einer Animation dieselben Lichtpfade gewählt werden. Dies kann aber unter Umständen zu einem zeitlichen Rauschen während der Animation führen.

Man kann also zusammenfassen, dass in den letzten Jahren ein sehr großer Schritt nach Vorne gemacht wurde, dass aber noch längst nicht alle Probleme gelöst sind. Es steht uns eine aufregende Zeit in der Computergrafik bevor, in der es immer realistischere Flüssigkeiten und Gasen zu sehen gibt.

Referenzen

- FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual Simulation of Smoke. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Annual Conference Series, ACM, 15–22.
- FRANZKE, O. 2003. Flexible Raytracerarchitekturen zur Berechnung komplexer optischer Phänomene. Technische Universität Dresden, *Großer Beleg, Sommersemester 2003*.
- HARRIS, M. J., HOFFMAN, N., PREETHAM, A., AND PREMOSZE, S. 2003. Light and Color in the Outdoors. ACM SIGGRAPH 2003 Course #1 Notes, July.
- HENVEY, L., AND GREENSTEIN, J. 1941. Diffuse Radiation in the Galaxy. *Astrophysics Journal* 93, 70-83.
- JENSEN, H. W., AND CHRISTENSEN, P. 1998. Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. In *Proceedings of SIGGRAPH 1998*, ACM Press / ACM SIGGRAPH, Annual Conference Series, ACM, 311–320.
- STAM, J. 1999. Stable Fluids. In *Proceedings of SIGGRAPH 1999*, ACM Press / ACM SIGGRAPH, Annual Conference Series, ACM, 121–128.