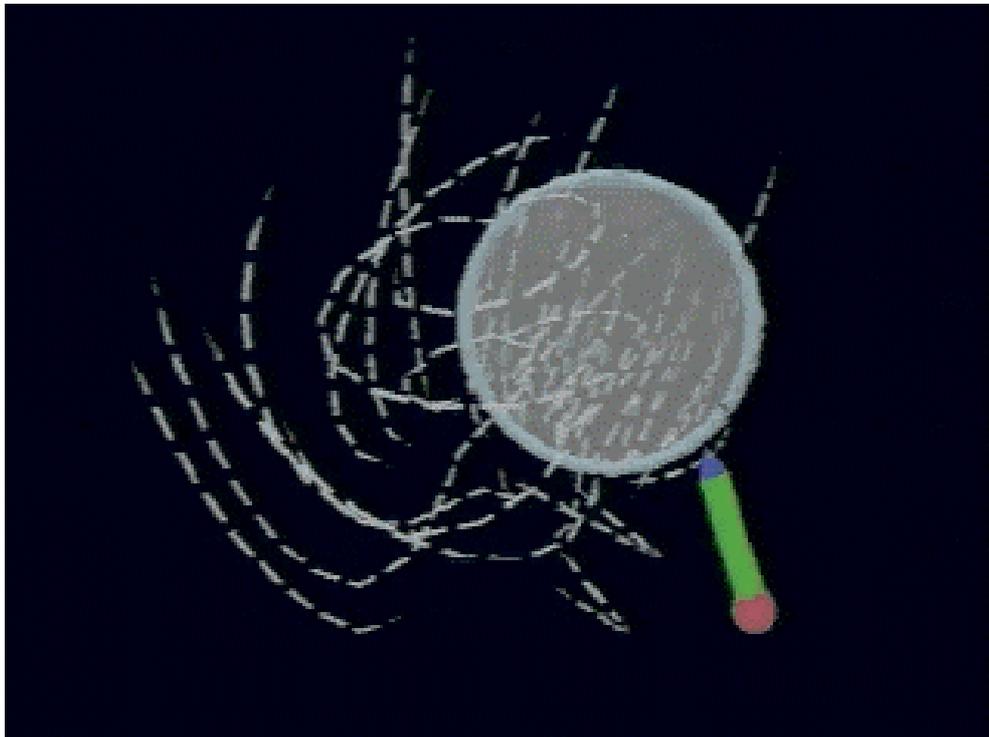


Real-Time Techniques For 3D Flow Visualisazion

Anton Fuhrmann and Eduard Gröller
Vienna University of Technology

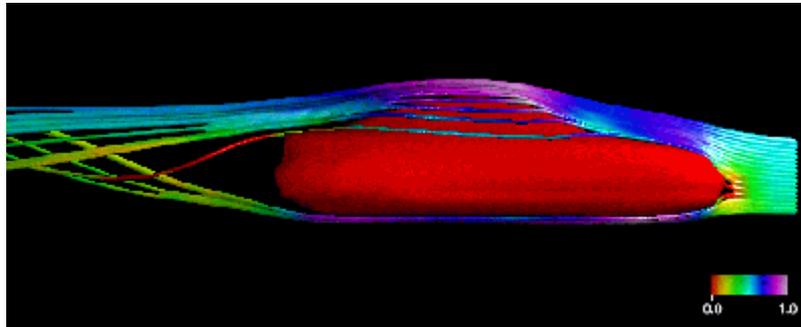


Eine Ausarbeitung von Stefan Berger im
Rahmen des Fachseminares
'Aktuelle Themen der Graphischen Datenverarbeitung'
SS 1999

VORWORT	3
1 VISUALISIERUNG	4
1.1 EINFÜHRUNG	4
1.2 STRÖMUNGSLINIEN: GUT PLATZIERT IST HALB GEWONNEN.....	4
1.3 DASHTUBES: EINE SPEZIELLE VISUALISIERUNG VON STRÖMUNGSLINIEN	6
1.3.1 Grundidee	6
1.3.2 Welche Eigenschaften müssen erfüllt werden?	6
1.3.3 Wie werden diese Eigenschaften erfüllt?	7
1.3.4 Implementierung	7
1.4 WIE WIRD DIE TEXTURIERUNG IMPLEMENTIERT?	7
1.4.1 Die Mipmap Methode.....	8
1.4.2 Die Texture-Coordinate Methode	9
2 INTERAKTION	10
2.1 DIE MAGISCHE LINSE.....	10
2.1.1 Idee.....	10
2.1.2 Implementierung	10
2.2 DER MAGISCHE WÜRFEL	11
2.2.1 Idee.....	11
2.2.2 Implementierung	11

Vorwort

Strömungen spielt in vielen Ingenieur-Entwicklungsgebieten eine wichtige Rolle. Im Fahrzeugbau zum Beispiel entscheidet die Aerodynamik einer Karosserie weitgehend den Treibstoffverbrauch und die Geräusentwicklung eines zukünftigen Fahrzeuges.



http://davinci.informatik.uni-kl.de/Projects/d4m/Html/Research/g_text04.html

Der Computer stellt eine sehr komfortable Plattform dar, solche Strömungen zu simulieren und verschiedene Parameter der Strömung rasch und einfach zu Verändern. Doch bereits bei der Darstellung 2-Dimensionaler Strömungen muss stark abstrahiert werden, wenn nur schon Richtung und Geschwindigkeit des Strömungsfeldes dargestellt werden sollen. Erweitert man das Feld um eine weitere Dimension, scheint dies aufgrund der grossen Informationsdichte eine schier unlösbare Aufgabe zu werden. Die Ausarbeitung von A.Fuhrmann und E.Gröller beschreibt eine faszinierende Möglichkeit, sowohl der Visualisierung als auch der Interaktion bezüglich stehenden,3-Dimensionalen Strömungen in Echtzeit.

1 Visualisierung

1.1 Einführung

Die Darstellung einer stehenden Strömung auf einem Bildschirm erfordert eine ‘vernünftige’ Art der Abstrahierung. Vernünftig in dem Sinne, dass die entstehende Grafik einerseits gut verständlich bleibt, andererseits die Natur der Strömung dennoch erhalten bleibt.

Strömungslinien sind eine einfache und intuitiv zu verstehende Möglichkeit einer solchen Vereinfachung.

Jedoch treten gerade im 3-dimensionalen Fall unerwünschte Nebenerscheinungen auf, die das Erkennen von interessanten Stellen erschwert.

Werden die Strömungslinien auch als Linien dargestellt, ist es sehr schwer, Informationen über die Tiefe der Linie zu entnehmen. Ebenfalls kann die Richtung der Strömung nur durch separates Anbringen von Richtungsmarkierungen herausgelesen werden.

Werden die Strömungslinien als Röhren gerendert, können Tiefeninformationen realistisch dargestellt werden. Auch die Strömungsrichtung kann durch direktes Aufbringen von Markierungen auf der Textur einfach angezeigt werden. Sehr nahe gelegene Röhren verdecken jedoch einen grossen Teil der dahinterliegenden Szene.

Beide Verfahren haben zudem den Nachteil, dass die Darstellung der Strömungsgeschwindigkeit nicht intuitiv erfolgen kann. Eine der Geschwindigkeit entsprechende Dichte der Linien oder Farbgebung der Röhren ist eine oft verwendete Methode. Beide Methoden führen jedoch im 3-dimensionalen Fall zu komplexen, schwer lesbaren Grafiken.

Das Verfahren von Fuhrmann und Gröller basiert auf solchen Strömungslinien. Daher vorerst ein paar Bemerkungen zu ihnen.

1.2 Strömungslinien: Gut platziert ist halb gewonnen

“A **streamline** is a path traced out by a massless particle as it moves with the flow.”¹

Das Hauptproblem bei Strömungslinien besteht in der Platzierung der einzelnen Linien. Die resultierende Grafik sollte trotz möglichst genauer Abbildung des Vektorfeldes nicht zu viele Linien enthalten, welche die Betrachterin nur unnötig verwirren und eventuel wichtige Stellen verdecken würden. Auch sollten die Strömungslinien möglichst gleichmässig über dem Feld verteilt werden.

Fuhrmann und Gröller verwenden dazu eine auf 3 Dimensionen erweiterte Methode von [Bruno Jobard](#) und [Wilfrid Lefer](#). Das Prinzip des Algorithmus ist einfach zu verstehen und funktioniert folgendermassen:

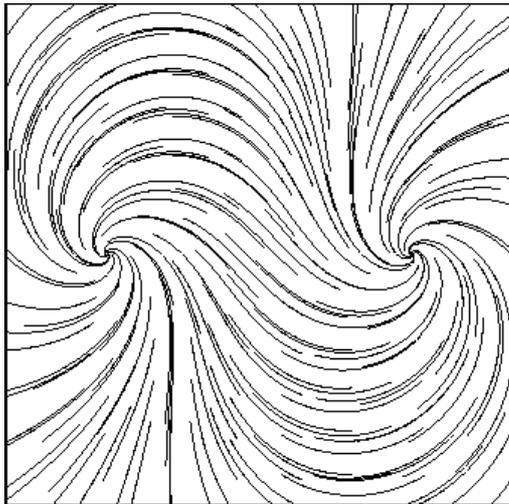
¹ Tom Benson, Definition of Streamlines, Glenn Research Center.
<http://www.grc.nasa.gov/WWW/IFMD/airplane/stream.html>

```

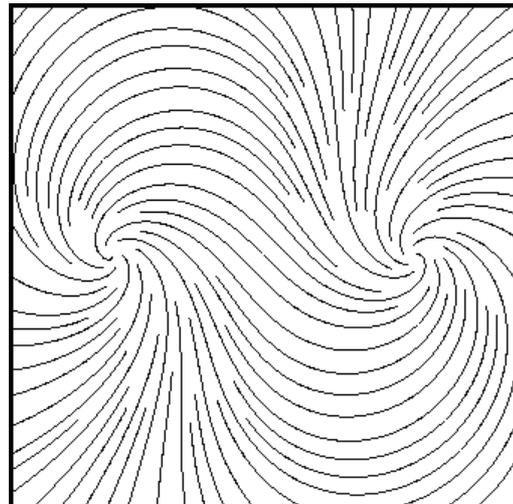
Wähle einen beliebigen Punkt und berechne daraus eine Strömungslinie ;
Füge diese Linie in die Warteschlange ein ;
Fertig := False ;
REPEAT
  REPEAT
    Wähle einen beliebigen Punkt mit gegebenem Abstand  $d$  von der aktuellen Linie
    (die Vorderste in der Warteschlange) ;
    // Die nun entstehende Strömungslinie ist regulär, wenn sie beidseitig den
    Wertebereich verlässt, ohne den Minimalabstand  $d*k$  (  $k$  ist wie  $d$  sinnvoll zu
    wählen) an irgendeine Linie zu unterbieten.
  UNTIL die entstehende Strömungslinie ist regulär oder es lohnt sich nicht mehr
  weiterzusuchen ;
  IF eine reguläre Linie wurde gefunden
    Füge diese Linie in die Warteschlange ein
  ELSE // es lohnt sich nicht mehr weiterzusuchen
    IF Warteschlange ist leer
      Fertig := True ;
    ELSE
      Entferne die erste Linie aus der Warteschlange ;
    END IF ;
  END IF ;
UNTIL Fertig = True ;

```

Anhand der folgenden zwei Abbildungen² ist einfach zu sehen, welchen visuellen Vorteil gleichmässig verteilte Strömungslinien erzeugen.



Startpunkte auf regulärem Netz verteilt



Den Algorithmus von Jobard und Lefer angewendet

Ein wichtiger Punkt hierbei ist, dass für die Betrachtung von stehenden Strömungen diese Strömungslinien vor der eigentlichen Echtzeit-Visualisierung berechnet werden können. Eine Vereinfachung bringt es, wenn die Startpunkte auf einem festen Gitter zu liegen kommen, auf dem auch die Abstandstests durchgeführt werden.

²[Bruno Jobard und Wilfrid Lefer, Creating Evenly-Spaced Streamlines of Arbitrary Density : http://www-lil.univ-littoral.fr/~jobard/Research/Publications/EGW-ViSC97/ViSC97.html](http://www-lil.univ-littoral.fr/~jobard/Research/Publications/EGW-ViSC97/ViSC97.html)

1.3 Dashtubes: Eine spezielle Visualisierung von Strömungslinien

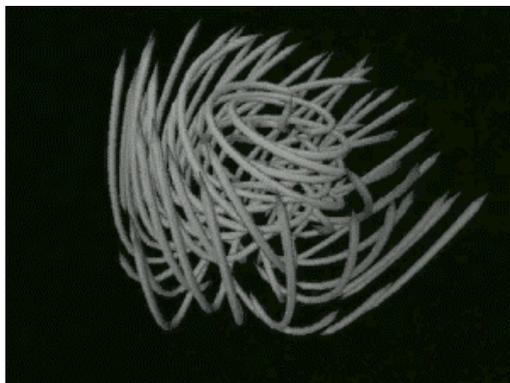
Während der Simulation müssen diese Strömungslinien nun als 'Masse' gerendert werden, und genau in diesem Rendering steckt des Pudels Kern. Fuhrmann und Gröller setzen eine interessante Idee auf eine noch interessantere Methode um. Doch alles schön der Reihe nach.

1.3.1 Grundidee

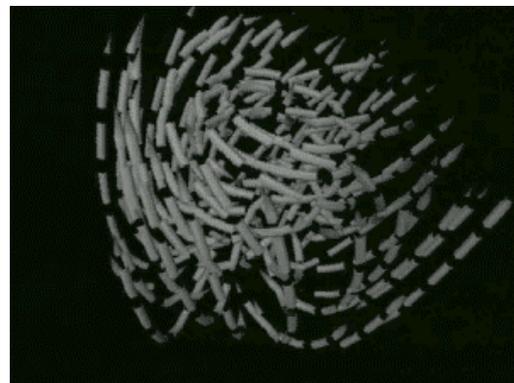
Um eine 3-dimensionale Wirkung zu erhalten, müssen die Strömungslinien voluminös gerendert werden, idealerweise als Zylinder (Röhren) mit einem vorgegebenen Radius. Diese Darstellung besitzt jedoch einen entscheidenden Nachteil: Nahe am Betrachter liegende Strömungslinien verdecken aufgrund ihrer Grösse einen grossen Teil der Szene.

Fuhrmann und Gröller verwenden Dashtubes zur Visualisierung der Strömungslinien. Dashtubes sind kurze, undurchsichtige Zylinder, die entlang der Strömungslinie angeordnet sind. Sie werden sogar animiert, um Richtung und Geschwindigkeit der Strömung anzuzeigen. Zusätzlich können durch diese abwechselnd durchsichtig und undurchsichtig gerenderten Strömungslinien weiter hinten liegende Strömungen besser erkannt werden.

Die folgenden zwei Abbildungen³ verdeutlichen diese Darstellung:



Strömungslinien 'normal' gerendert



Strömungslinien mit Dashtubes gerendert

1.3.2 Welche Eigenschaften müssen erfüllt werden?

Aufgrund guter Erkennbarkeit der Grafik müssen diese Dashtubes also folgende Forderungen erfüllen :

- A. **Volume-filling properties:** Die Dashtubes sollten gleichmässig über den Raum verteilt sein.
- B. **Reduced occlusion:** Weiter hinten liegende Strömungen sollten so wenig wie möglich von nahen Strömungen verdeckt werden.
- C. **Animation of flow:** Geschwindigkeit und Richtung der Strömungen sollte gut sichtbar gemacht werden. Die Geschwindigkeit der Tubes sollte proportional zur Geschwindigkeit der Strömung sein.
- D. **Visibility of dashes:** Die Länge der Tubes sollte unabhängig von der Geschwindigkeit sein, da sonst unschöne Effekte (sehr lange oder kurze Tubes) entstehen können.
- E. **Fast Rendering:** Da Strömungen in Echtzeit dargestellt werden wollen, müssen die Tubes so schnell wie möglich gerendert werden. Idealerweise mit Unterstützung von Grafik-Hardware.

³[Anton L. Fuhrmann](#) und Eduard Gröller, [Real-Time Techniques For 3D Flow Visualization](#)

1.3.3 Wie werden diese Eigenschaften erfüllt?

- A. Durch die gleichmässig verteilten Strömungslinien werden die Tubes ebenfalls gleichmässig verteilt.
- B. Durch die Verwendung einer Textur, die abwechselnd transparent und undurchsichtig ist, entstehen die gewünschten Tubes, deren Zwischenräume die Verdeckung von Details reduzieren.
- C. Wird die Textur auf den Strömungslinien verschoben, entsteht der Effekt einer Bewegung. Diese Verschiebung wird aus Performance-Gründen nicht in den Textur-Daten vorgenommen, sondern wird durch eine Veränderung in der Transformationsmatrix vorgenommen.
- D. Da die Texturierung nicht über gleichlange Linienintegrale, sondern direkt mit der Strömungsfunktion geschieht, wird die Texturierung an 'schnellen' Stellen auseinandergezogen und an langsamen zusammengedrückt. Dies wollen wir vermeiden! Die Forderung nach einigermaßen gleichlangen Tubes wird durch spezielle Texturen erreicht, die sich der Geschwindigkeit der Strömung 'anpassen'. Im nächsten Abschnitt wird diese Texturierung näher erläutert.
- E. Die Änderungen in der Transformationsmatrix und das Texturieren der Strömungslinien werden durch Grafik-Hardware effizient unterstützt.

1.3.4 Implementierung

Idealerweise wird die Schnittfläche solcher Tubes aus einem Kreis bestehen. Zur Vereinfachung der Berechnungen kann diese Fläche jedoch auch n-Eck implementiert werden, 3 bis 6 Ecken liefern bei geeignetem Shading jedoch ein durchaus akzeptables Resultat.

Werden nun beim Rendern die einzelnen Eckpunkte dieser Querschnittsflächen zu einem Gitter verbunden, können unschöne Effekte entstehen. Verdrehen sich diese Flächen gegeneinander, verkleinert sich der Durchmesser des Tubes zwischen zwei Querschnitten. Dieser Effekt kann bei gebogenen Tubes besonders stark auftreten.

Ein geeigneter Algorithmus zur Erzeugung solcher Gitter verringert diese Torsion jedoch, durch Angleichen der einzelnen Flächen, auf ein Minimum.

1.4 **Wie wird die Texturierung implementiert?**

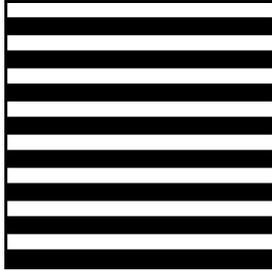
Wie bereits gesehen fordern wir, dass die Länge der einzelnen Tubes nicht allzustark variieren, sich also nicht der Geschwindigkeit entsprechend stark ändern soll. Dies hat zusätzlich den Effekt, Aliasing-Phänomene bei sehr kleinen Geschwindigkeiten auszuschliessen.

Auch weit entfernte oder vom Betrachter wegführende Strömungslinien können Aliasing-Effekte verursachen. Daher sollte die Texturierung so funktionieren, dass alle Tubes ungefähr gleich gross sind, und sich mit zunehmender Entfernung nicht zu stark verkleinern.

Zur Lösung dieses Problems bieten sich die folgenden zwei Methoden an.

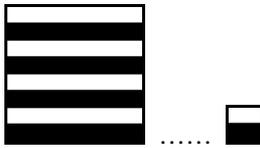
1.4.1 Die Mipmap Methode

Die hier als bekannt angenommene Mip-Map Methode zur Texturierung diens als Basis, um unser Ziel zu erreichen. Sei die folgende Textur gegeben:



Wir 'verkleinern' unsere Textur nun nicht entsprechend der Mip-Map Methode, indem wir sie filtern und auf jeweils halbe grösse interpolieren. Dies hätte nicht den gewünschten Effekt, und würde zu grauen Stellen führen, die nicht mehr eindeutig als sichtbar oder transparent definiert wären. Auch würden sehr kleine Stellen im Bildbereich nicht durch weniger Tubes repräsentiert. Wir behalten einfach die Grösse der Strukturierung in den verkleinerten Maps bei.

Die so entstehenden Maps sehen dann wie folgt aus:



Dies hat den gewünschten Effekt, dass im Bildbereich sehr kurze Stellen, sei dies aufgrund grosser Entfernung zur Betrachterin oder wegen sehr kleiner Fliesgeschwindigkeit, nur noch durch einen Tube dargestellt werden, während längere Segmente durch mehrere Tubes dargestellt werden.

Ein weiterer Vorteil dieser Methode ist, dass die einzelnen Maps in einem Prä-Prozess berechnet werden können, also zum Zeitpunkt der Visualisierung keine zusätzliche Rechenleistung erfordern. Doch diese Methode hat auch Nachteile, so steigt zum Beispiel der Speicherbedarf für jede weitere, grössere Map um das Doppelte. Weit unschöner ist die Tatsache, dass zwischen den einzelnen Maps nicht interpoliert werden darf, dies auch wieder aufgrund der eindeutigen Unterscheidbarkeit zwischen Sichtbar und Transparent. Dies führt zu unschönen Sprüngen in den Längen der Tubes, wenn die nächste Map-Stufe erreicht wird.

Mit Hilfe der nächsten Methode können solche Sprünge elegant und einfach umgangen werden. Sie werden staunen...

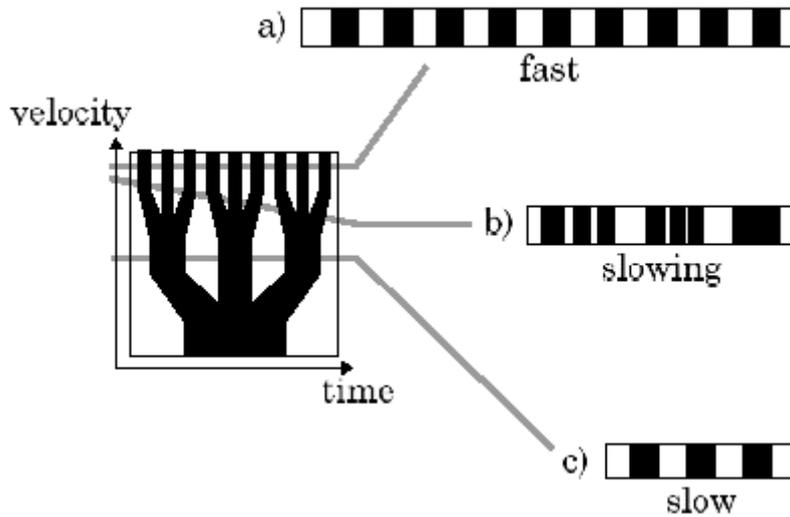
1.4.2 Die Texture-Coordinate Methode

Die Texture-Coordinate Methode besteht aus einem speziellen Zusammenfügen einzelner Mip-Maps zu einer einzigen Textur.

Da die Dastubes im ganzen Umfang die gleiche Textur erhalten, ist es nicht nötig, in dieser Richtung ebenfalls eine Parametrisierung vorzunehmen. Wir benötigen also keine Rechtecksfläche zur Texturierung, sondern nur einen Streifen dieser Textur.

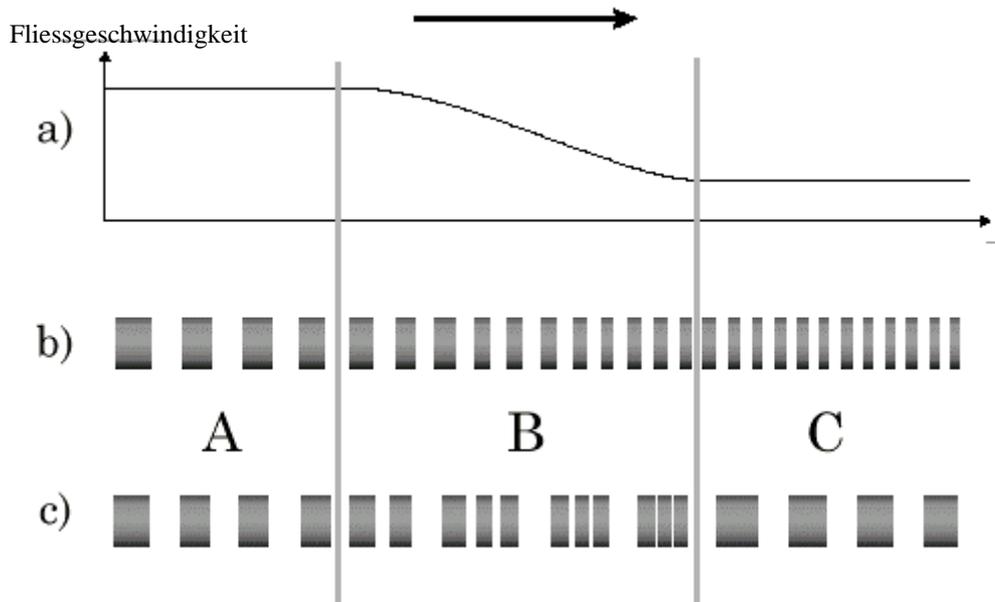
Die so entstehende Freiheit in einer Richtung der Textur-Koordinaten kann nun für die Länge des zu füllenden Segmentes verwendet werden.

Die folgende Grafik verdeutlicht diesen Sachverhalt:



Verwendete Textur für die Texture-Coordinate Methode⁴

An Stellen, an denen sich die Strömung verlangsamt oder weiter vom Betrachter entfernt, verkleinern sich die Tubes zwar, sie verschmelzen aber regelmässig zu grösseren, langsamer Bewegenden Tubes.



a) Abnahme der Flieessgeschwindigkeit b) Normale Texturierung c) Texture-Coordinate Methode⁴

⁴ [Anton L. Fuhrmann](#) und Eduard Gröller, [Real-Time Techniques For 3D Flow Visualization](#)

2 Interaktion

Ein generelles Problem bei der Darstellung 3-dimensionaler Strömungen ist die Dichte der anzuzeigenden Informationen. Werden sehr wenige Informationen der Strömung dargestellt, ist die Untersuchung komplexer Stellen nicht möglich. Werden hingegen von Anfang an sehr viele Informationen angezeigt, wird das Auffinden und Betrachten solcher Stellen zur Suche der Nadel im Heuhaufen. Nahe gelegene Objekte verdecken einen grossen Teil der Darstellung.

In unserer Visualisierung mit Dash-Tubes ist dieser Umstand besonders hinderlich. Auch die Ausmasse der einzelnen Tubes und deren Zwischenräume spielen dabei eine grosse Rolle. Diese Veränderung kann einfach durch wechseln der Textur-Datei und einzelner Parameter der Map-Funktion erreicht werden.

Die Dichte der zuvor berechneten Strömungslinien, die durch die Parameter d und k (s. 1.2) gegeben ist, möchte man oft an gewissen Stellen erhöhen können, um Details genauer betrachten zu können.

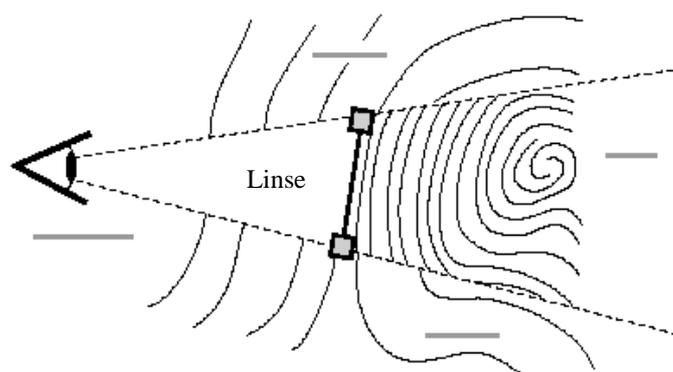
Die hier vorgeschlagenen Lösungen basieren auf einer solchen lokalen Erhöhung der gerenderten Strömungslinien.

Dem Benutzer wird ein virtuelles Instrument zur Verfügung gestellt, mit dessen Hilfe er interaktiv in die Darstellung der Strömung eingreifen kann.

2.1 Die magische Linse

2.1.1 Idee

Die magische Linse ist eine im Bildbereich liegende, planare Fläche, die vom Benutzer frei im Raum platziert werden kann. Diese Linse erfüllt zwei Aufgaben:



1. Sie vergrössert die Zahl der gerenderten Strömungslinien 'hinter' ihr
2. Sie schneidet die vor ihr liegenden Teilstücke der Strömungslinien heraus

Das Arbeiten mit einer solchen Linse ist für die Benutzerin sehr intuitiv. Doch leider hat diese Methode einen grossen Nachteil: Das vergrösserte Teilgebiet hängt sehr stark von der Betrachtungsrichtung ab. Wird diese Richtung geändert, muss die Linse jeweils wieder neu ausgerichtet werden.

Auch werden sämtlich Stellen hinter der Linse detaillierter dargestellt, eine spezifische Fokussierung einer ganz bestimmten Stelle ist also nicht möglich.

2.1.2 Implementierung

Die Implementierung dieser Linse geschieht ebenfalls auf eine sehr interessante Art und Weise.

Wird die Linse benutzt, wird zuerst nur der Teil des Bildes gerendert, der ausserhalb des Sichtbarkeits-Kegels der Linse liegt. Danach wird die Strömung ein zweites Mal etwas dichter gerendert, und zwar nur an den Stellen, die hinter der Linse liegen.

Diese Linse stellt also eine Art 'Fenster in eine andere Welt' dar.

Dieser Mechanismus wurde von [Gernot Schaufler](#) und [Dieter Schmalstieg](#) entwickelt, und nennt sich **SEAM** (Spatially Extended Anchor Mechanism) (<http://www.cg.tuwien.ac.at/research/vr/seams>)

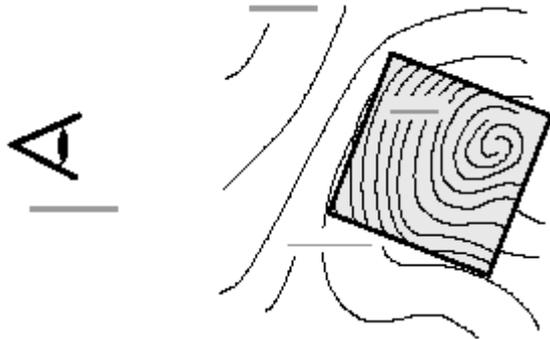
Ein grosser Vorteil besteht darin, dass die verschiedenen dichten Visualisierungen nur einmal berechnet werden müssen. Durch nachträgliches Clippen wird der benötigte Teil einfach ausgeschnitten.

2.2 *Der magische Würfel*

2.2.1 Idee

Dieser semitransparente Würfel kann vom Benutzer frei im Raum bewegt werden. Der darin enthaltene Raumbereich wird dann dichter gerendert. Auch kann Grösse und Form diese Würfels zur Laufzeit verändert werden.

Diese Methode hat den Vorteil, dass die Blickrichtungen der Betrachter keine Auswirkungen auf den vergrösserten Bereich hat. Wurde der Würfel an einer interessanten Stelle platziert, kann diese Stelle nun von verschiedenen Positionen aus betrachtet werden, ohne dass weitere Anpassungen vorgenommen werden müssten.



2.2.2 Implementierung

Auch dieser Würfel wird mit Hilfe von SEAMs implementiert. Der einzige Unterschied besteht darin, dass man jetzt sechs Schnitt-Ebenen benötigt.