

## Visualisierung: 3D Strömungsdaten

Andreas Kolb und Martin Lambers

computergraphik und multimedia systeme  
universität siegen



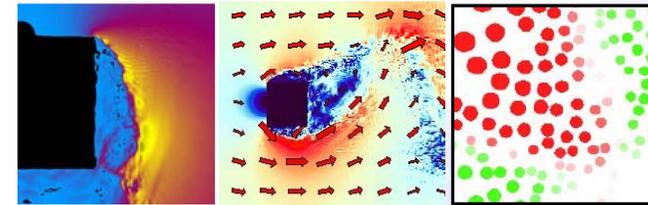
## Letzte Stunde: 2D Strömungsdaten

2

Visualisierungsverfahren für 2D Strömungsdaten

### Direkte Verfahren

- Farbkodierung (nur für skalare Größen)
- Vektorpfeile (nur für vektorielle Größen)
- Glyphen/Icons (für skalare und vektorielle Größen)



Andreas Kolb, Martin Lambers

Visualisierung

## Letzte Stunde: 2D Strömungsdaten

3

Visualisierungsverfahren für 2D Strömungsdaten

### Integrationsbasierte Verfahren

- Partikelbahnen
  - Wahl der Startpunkte (seed point algorithms)
  - Zellsuche im unstrukturierten und curvilinearen Gitter
  - C-Space- und P-Space -Verfahren
  - Zeitabhängige Vektorfelder:  
Streamlines, Streaklines & Pathlines



Andreas Kolb, Martin Lambers

Visualisierung

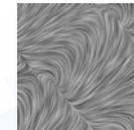
## 2D Strömungsdaten

4

Visualisierungsverfahren für 2D Strömungsdaten

### Integrationsbasierte Verfahren

- Texturbasierte Verfahren
  - Line Integral Convolution (LIC)  
Faltung eines Rauschfeldes entlang der  
Strömungslinien eines Vektorfeldes



Andreas Kolb, Martin Lambers

Visualisierung

## 2D Strömungsdaten

5

Visualisierungsverfahren für 2D Strömungsdaten  
**Integrationsbasierte Verfahren**

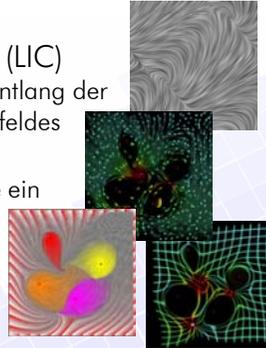
- **Texturbasierte Verfahren**

- **Line Integral Convolution (LIC)**

Faltung eines Rauschfeldes entlang der Strömungslinien eines Vektorfeldes

- **Texture Advection**

Bildbasierter Ansatz: Verzerre ein Ergebnisbild schrittweise in Richtung der Strömung

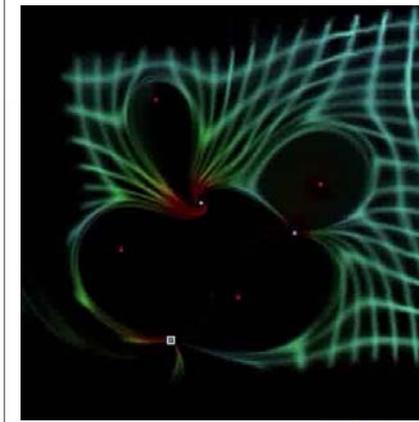


Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

6



**Texture Advection:**

„Verzerre ein Eingabebild entlang der Strömungslinien des Vektorfeldes“

**Quelle:**

van Wijk, Univ. Eindhoven  
 Image-based flow visualization  
 ACM SIGGRAPH 2003

<http://www.win.tue.nl/~vanwijk/ibfv/>

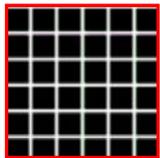
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

7

**Texture Advection Implementierung:**



- Definiere das Eingangsbild als **Polygon mit Textur.**

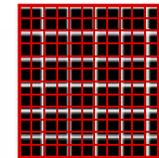
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

8

**Texture Advection Implementierung:**



- Definiere das Eingangsbild als **Polygon mit Textur.**
- Zerlege das Polygon in **viele kleine Polygone**

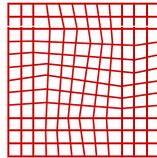
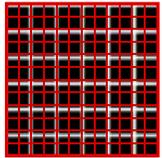
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

9

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.

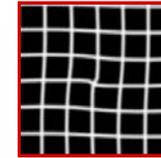
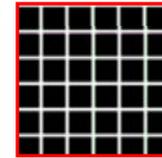
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

10

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer.

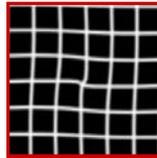
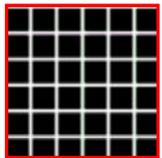
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

11

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer.

Lies das Bild aus dem Frame-Buffer und kopiere es in die Eingangstextur

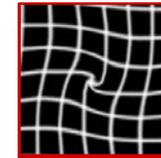
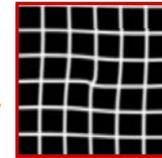
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

12

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer.

Lies das Bild aus dem Frame-Buffer und kopiere es in die Eingangstextur

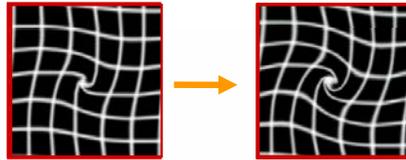
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

13

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer.

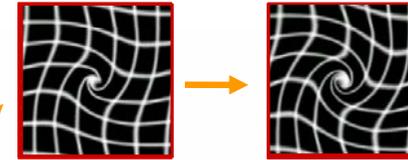
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

14

### Texture Advection Implementierung:



- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**.
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer.

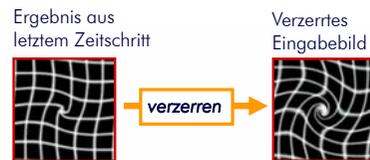
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

15

### Erweiterte Implementierung:



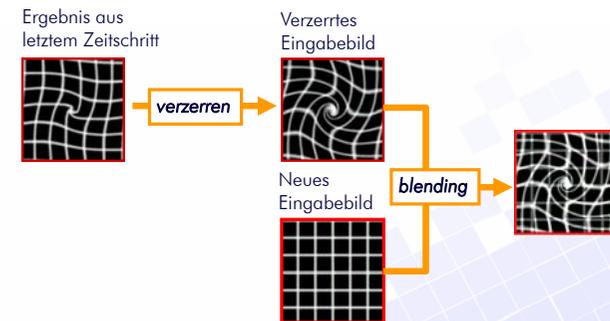
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

16

### Erweiterte Implementierung:



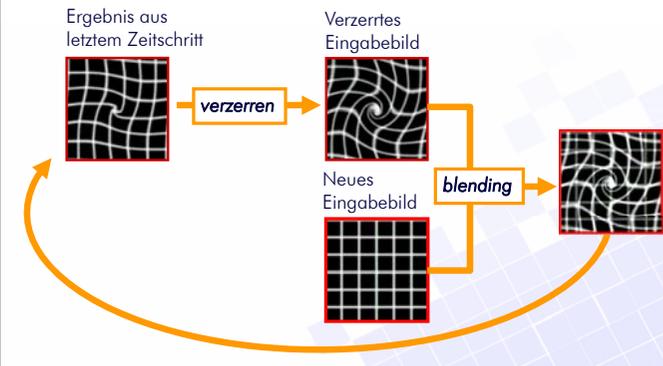
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

17

Erweiterte Implementierung:



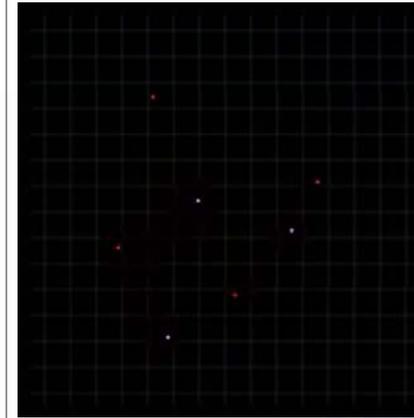
Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Techniken

18

Beispiel der erweiterten Implementierung



Quelle:  
van Wijk, Univ. Eindhoven  
Image-based flow visualization  
ACM SIGGRAPH 2003  
<http://www.win.tue.nl/~vanwijk/ibfv/>

Andreas Kolb, Martin Lambers

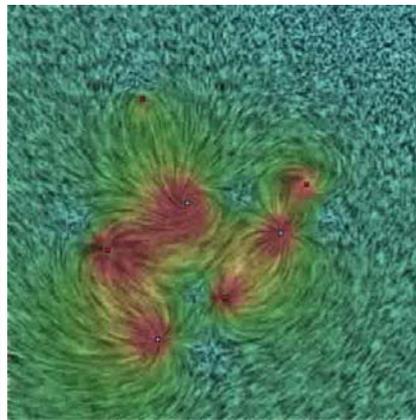
Visualisierung

## Texturbasierte Verfahren

19

Beispiel der erweiterten Implementierung:

Simulation von LIC



Quelle:  
van Wijk, Univ. Eindhoven  
Image-based flow visualization  
ACM SIGGRAPH 2003  
<http://www.win.tue.nl/~vanwijk/ibfv/>

Andreas Kolb, Martin Lambers

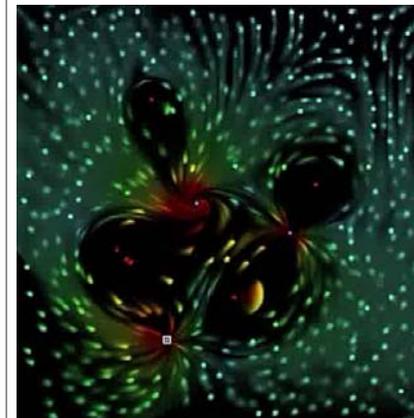
Visualisierung

## Texturbasierte Verfahren

20

Beispiel der erweiterten Implementierung:

Partikelsysteme



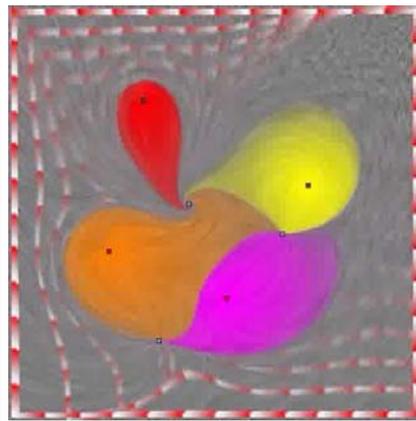
Quelle:  
van Wijk, Univ. Eindhoven  
Image-based flow visualization  
ACM SIGGRAPH 2003  
<http://www.win.tue.nl/~vanwijk/ibfv/>

Andreas Kolb, Martin Lambers

Visualisierung

## Texturbasierte Verfahren

21



Beispiel der erweiterten Implementierung.

Quelle:  
van Wijk, Univ. Eindhoven  
Image-based flow visualization  
ACM SIGGRAPH 2003  
<http://www.win.tue.nl/~vanwijk/ibfv/>

Andreas Kolb, Martin Lambers

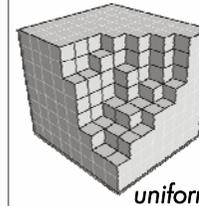
Visualisierung

## 3D Vektorfelder

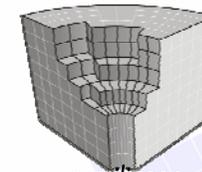
22

### Definition: 3D Vektorfeld

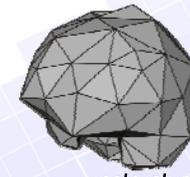
$$\vec{v}(\vec{x}) = \begin{pmatrix} v_x(x, y, z) \\ v_y(x, y, z) \\ v_z(x, y, z) \end{pmatrix} \quad \vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



uniform



curvilinear



unstructured

Andreas Kolb, Martin Lambers

Visualisierung

## Jacobi-Matrix in 3D

23

- Wie stark ändert sich das Vektorfeld an der Stelle  $\vec{x}$ ?

Totales Differential: Jacobi-Matrix (analog zu 2D)

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

Richtungsableitung: Wie stark ändert sich das Vektorfeld in Richtung  $\vec{h}$ ?

$$\frac{\partial \vec{v}(\vec{x})}{\partial \vec{h}} = \mathbf{J}_{\vec{v}}(\vec{x}) \vec{h}$$

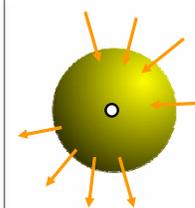
Andreas Kolb, Martin Lambers

Visualisierung

## Divergenz in 3D

24

- Analog zu 2D: Summe der Elemente der Hauptdiagonale der Jacobi-Matrix



$$\operatorname{div} \vec{v}(\vec{x}) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

Interpretation:

- Betrachte ein infinitesimales kleines Volumen um den Punkt
- Wieviel fließt hinein, wieviel hinaus?

$$\operatorname{div} \vec{v}(\vec{x}) < 0$$

Senke, „Zusammenströmen“

$$\operatorname{div} \vec{v}(\vec{x}) > 0$$

Quelle, „Auseinanderströmen“

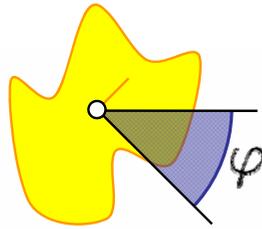
Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 2D

25

### Aus der Schulphysik: Rotation starrer Körper



Ein Körper dreht sich in der Zeit  $t$  um den Winkel  $\varphi$ .

Der Körper dreht sich mit der Winkelgeschwindigkeit  $\omega$ :

Bei konstanter Winkelgeschwindigkeit:  $\omega = \frac{\varphi}{t}$

Bei zeitabhängiger Winkelgeschwindigkeit:  $\omega(t) = \frac{\partial \varphi(t)}{\partial t}$

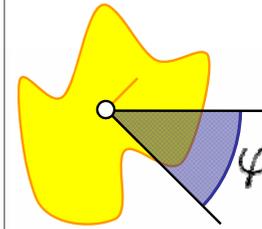
Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 2D

26

### Aus der Schulphysik: Rotation starrer Körper



Ein Körper dreht sich in der Zeit  $t$  um den Winkel  $\varphi$ .

Der Körper dreht sich mit der Winkelgeschwindigkeit  $\omega$ :

Im 2D Vektorfeld:

Winkelgeschwindigkeit  $\omega = \frac{1}{2} \text{rot } \vec{v}(\vec{x})$

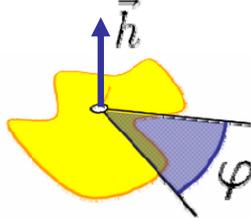
Andreas Kolb, Martin Lambers

Visualisierung

## Rotation starrer Körper

27

### Rotation in 3D:



Ein Körper dreht sich in der Zeit  $t$  um den Winkel  $\varphi$  um die Achse  $\vec{h}$ .

Der Körper dreht sich mit der Winkelgeschwindigkeit  $\omega$  um die Achse  $\vec{h}$ .

Die Rotation  $\text{rot } \vec{v}(\vec{x})$  für 3D Vektorfelder ist kein Skalar, sondern ein Vektor

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

28

### Betrachte die 2D Rotation in den einzelnen Ebenen:

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

29

- Betrachte die 2D Rotation in den einzelnen Ebenen:

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

$$\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

30

- Betrachte die 2D Rotation in den einzelnen Ebenen:

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

$$\frac{\partial v_x}{\partial y} - \frac{\partial v_y}{\partial z}$$

$$\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

31

- Betrachte die 2D Rotation in den einzelnen Ebenen:

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

$$\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}$$

$$\frac{\partial v_z}{\partial z} - \frac{\partial v_z}{\partial x}$$

$$\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

32

- Betrachte die 2D Rotation in den einzelnen Ebenen:

$$\mathbf{J}_{\vec{v}}(\vec{x}) = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

$$\text{rot } \vec{v}(\vec{x}) = \begin{pmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{pmatrix}$$

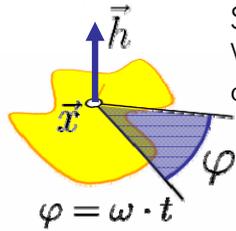
Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

33

- **Interpretation:** An der Stelle  $\vec{x}$  dreht sich die Strömung lokal mit der Winkelgeschwindigkeit  $\omega$  um die Achse  $\vec{h}$ :



$$\vec{\omega} = \frac{1}{2} \text{rot } \vec{v}(\vec{x})$$

Winkelgeschwindigkeit:

$$\omega = \|\vec{\omega}\|$$

Rotationsachse:

$$\vec{h} = \frac{\vec{\omega}}{\|\vec{\omega}\|}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Rotation in 3D

34

- Wie stark ist die lokale Rotation der Strömung um eine *bestimmte gegebene Achse*  $\vec{r}$ ?

$$\text{rot}_{\vec{r}} \vec{v} = \frac{\langle \text{rot } \vec{v} \circ \vec{r} \rangle}{\|\vec{r}\|}$$

(Projektion des Rotationsvektors auf die Rotationsachse des Vektorfelds)

- **Spezialfall „streamwise rotation“:**  
Lokale Rotation um die Strömungsrichtung:

$$\omega_s = \frac{\langle \text{rot } \vec{v} \circ \vec{v} \rangle}{\|\vec{v}\|}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

35

$\frac{\partial}{\partial x}$  ← Differentialoperator  
(partielle Ableitung in  
x-Richtung)

Angewendet auf eine Funktion  $f(x, y, z)$  liefert er die partielle Ableitung in x-Richtung:

$$\frac{\partial}{\partial x} f(x, y, z)$$

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

36

$\frac{\partial}{\partial x}$  ← Differentialoperator  
(partielle Ableitung in  
x-Richtung)

$\frac{\partial}{\partial y}$  ← Differentialoperator  
(partielle Ableitung in  
y-Richtung)

$\frac{\partial}{\partial z}$  ← Differentialoperator  
(partielle Ableitung in  
z-Richtung)

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

37

- Jacobi-Matrix, Divergenz und Rotation können mit Hilfe des *Nabla-Operators* ausgedrückt werden.

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

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

38

- Jacobi-Matrix, Divergenz und Rotation können mit Hilfe des *Nabla-Operators* ausgedrückt werden

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

Divergenz

$$\begin{aligned} \operatorname{div} \vec{v} &= \langle \nabla \circ \vec{v} \rangle \\ &= \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \end{aligned}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

39

- Jacobi-Matrix, Divergenz und Rotation können mit Hilfe des *Nabla-Operators* ausgedrückt werden

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

Rotation

$$\operatorname{rot} \vec{v}(\vec{x}) = \nabla \times \vec{v}(\vec{x}) = \begin{pmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{pmatrix}$$

Andreas Kolb, Martin Lambers

Visualisierung

## Ableitungen in 3D

40

- Jacobi-Matrix, Divergenz und Rotation können mit Hilfe des *Nabla-Operators* ausgedrückt werden

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

Jacobi-Matrix

$$\mathbf{J}_{\vec{v}} = \vec{v} \cdot \nabla^T = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

Andreas Kolb, Martin Lambers

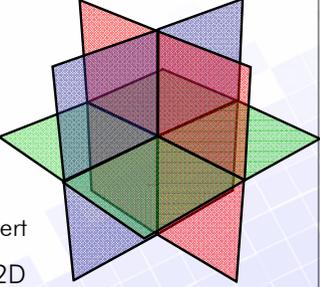
Visualisierung

### Kritische Punkte in 3D 41

- Bestimmung und Klassifikation analog zu 2D: Eigenwerte und Eigenvektoren der Jacobi-Matrix

**Prinzip:**

- Betrachte die orthogonalen Ebenen, die durch Eigenvektoren aufgespannt werden
- Betrachte die (Strömungs-) Vektoren auf die Ebene projiziert
- Klassifiziere die Ebene wie in 2D



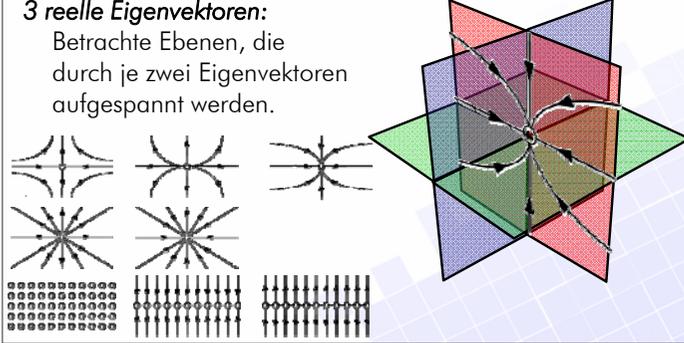
Visualisierung

Andreas Kolb, Martin Lambers

### Kritische Punkte in 3D 42

- Bestimmung und Klassifikation analog zu 2D: Eigenwerte und Eigenvektoren der Jacobi-Matrix

**3 reelle Eigenvektoren:**  
Betrachte Ebenen, die durch je zwei Eigenvektoren aufgespannt werden.



Visualisierung

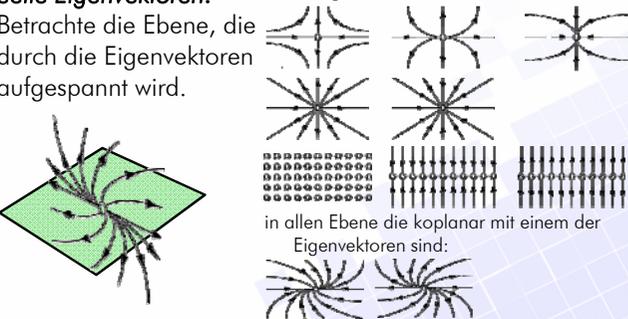
Andreas Kolb, Martin Lambers

### Kritische Punkte in 3D 43

- Bestimmung und Klassifikation analog zu 2D: Eigenwerte und Eigenvektoren der Jacobi-Matrix

**2 reelle Eigenvektoren:**  
Betrachte die Ebene, die durch die Eigenvektoren aufgespannt wird.

in der Eigenvektor-Ebene:



in allen Ebene die koplanar mit einem der Eigenvektoren sind:

Visualisierung

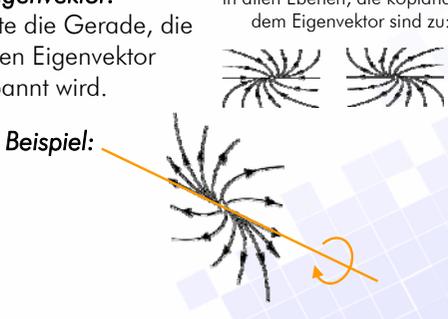
Andreas Kolb, Martin Lambers

### Kritische Punkte in 3D 44

- Bestimmung und Klassifikation analog zu 2D: Eigenwerte und Eigenvektoren der Jacobi-Matrix

**1 reeller Eigenvektor:**  
Betrachte die Gerade, die durch den Eigenvektor aufgespannt wird.

in allen Ebenen, die koplanar mit dem Eigenvektor sind zu:



**Beispiel:**

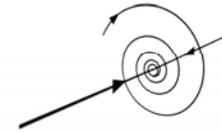
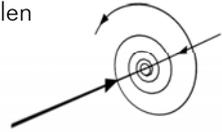
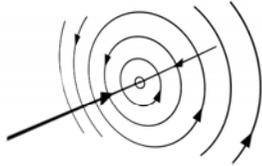
Visualisierung

Andreas Kolb, Martin Lambers

## Kritische Punkte in 3D

45

**Komplexe Eigenwerte** treten nur als Paar konjugiert komplexer Zahlen auf. **Daher existiert bei einer 3x3 Matrix immer mindestens ein reeller Eigenwert!**



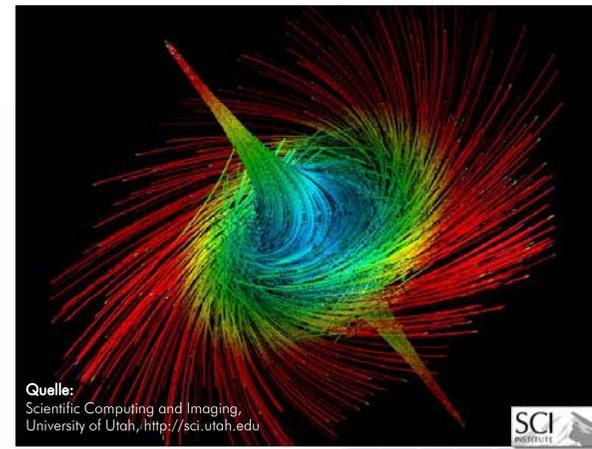
Je nach Vorzeichen des reellen Eigenwerts führt die Strömung zu dem kritischen Punkt hin oder weg.

Andreas Kolb, Martin Lambers

Visualisierung

## Kritische Punkte in 3D

46



Quelle:  
Scientific Computing and Imaging,  
University of Utah, <http://sci.utah.edu>



Andreas Kolb, Martin Lambers

Visualisierung

## Visualisierungsverfahren

47

### Direkte Verfahren

- Farbkodierung
  - auf Schnittebenen
  - im Volumen (siehe Volume Rendering)
  - Isoflächen (siehe Volume Rendering)

- Vektorpfeile
- Glyphen/Icons

### Integrationsbasierte Verfahren

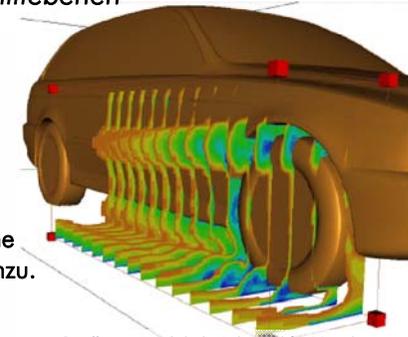
Andreas Kolb, Martin Lambers

Visualisierung

## Farbkodierung

48

3D Skalarfelder z.B. Dichte, Druck, Geschwindigkeitsbetrag  
**Abgebildet auf Schnittebenen**



Transparenz/Opazität  
kommt als zusätzliche  
Farbkomponente hinzu.

Quelle: Martin Schulz, Lehrstuhl für Visualisierung  
und Interaktive Systeme, Universität Stuttgart

Andreas Kolb, Martin Lambers

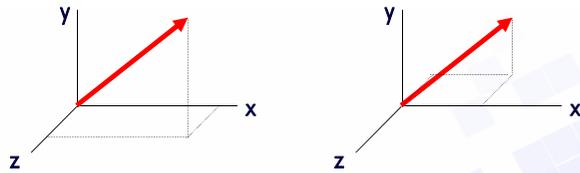
Visualisierung

## Arrow Plots

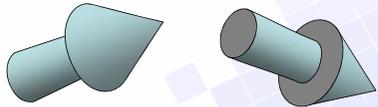
49

### Linien und einfache Pfeile:

Darstellung wird erschwert durch Perspektive



### Besser, aber auch aufwändiger: *Echte 3D Pfeile*

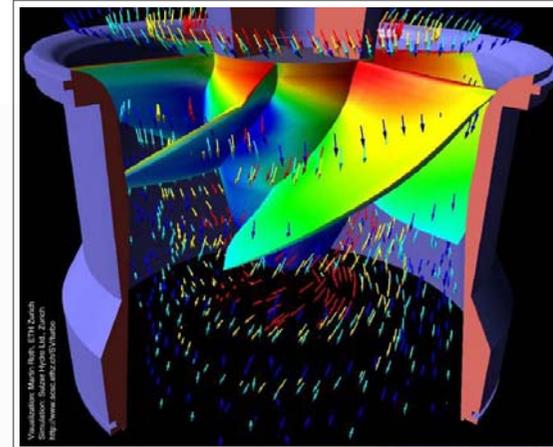


Andreas Kolb, Martin Lambers

Visualisierung

## Beispiel: Arrow Plots in 3D

50

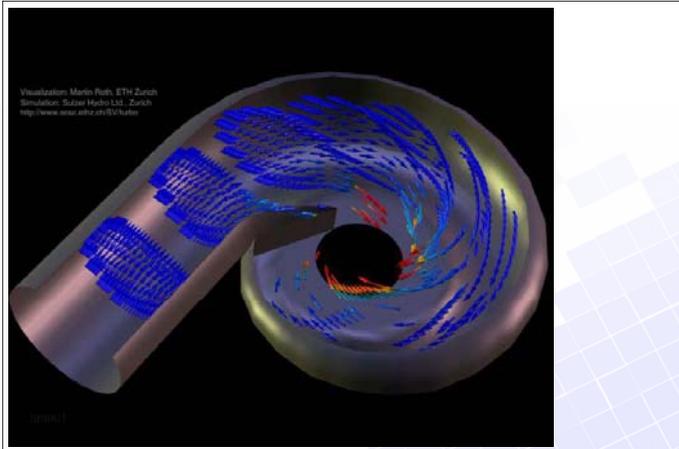


Andreas Kolb, Martin Lambers

Visualisierung

## Beispiel: Arrow Plots in 3D

51



Andreas Kolb, Martin Lambers

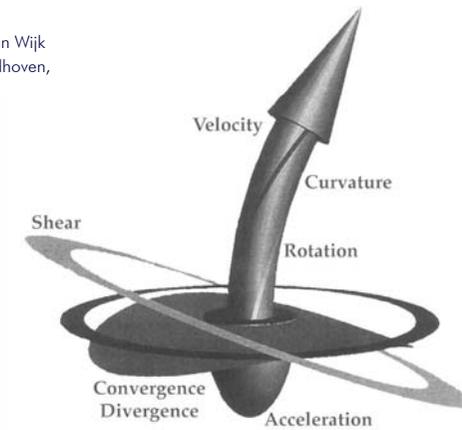
Visualisierung

## Glyphen/Icons in 3D

52

### Quelle:

de Leeuw and van Wijk  
University of Eindhoven,  
the Netherlands



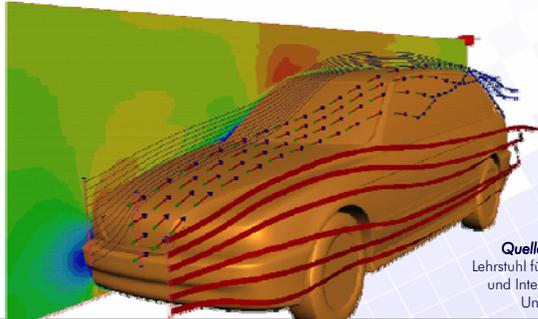
Andreas Kolb, Martin Lambers

Visualisierung

## Integrationsbasierte Verfahren

53

**Linienbasierte Techniken:**  
Wie bisher: Partikelbahnen (analog 2D),  
 Streamlines, Streaklines, Pathlines



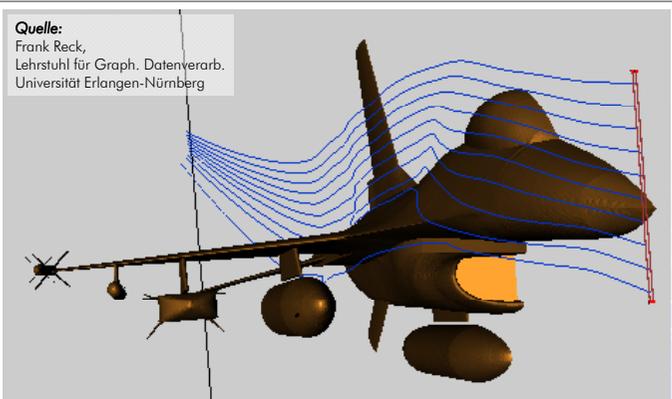
**Quelle:** Martin Schulz,  
 Lehrstuhl für Visualisierung  
 und Interaktive Systeme,  
 Universität Stuttgart

Andreas Kolb, Martin Lambers Visualisierung

## Partikelbahnen

54

**Quelle:**  
 Frank Reck,  
 Lehrstuhl für Graph. Datenverarb.  
 Universität Erlangen-Nürnberg

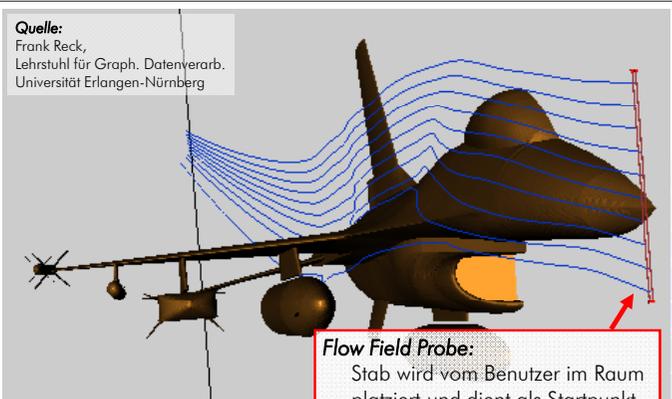


Andreas Kolb, Martin Lambers Visualisierung

## Partikelbahnen

55

**Quelle:**  
 Frank Reck,  
 Lehrstuhl für Graph. Datenverarb.  
 Universität Erlangen-Nürnberg



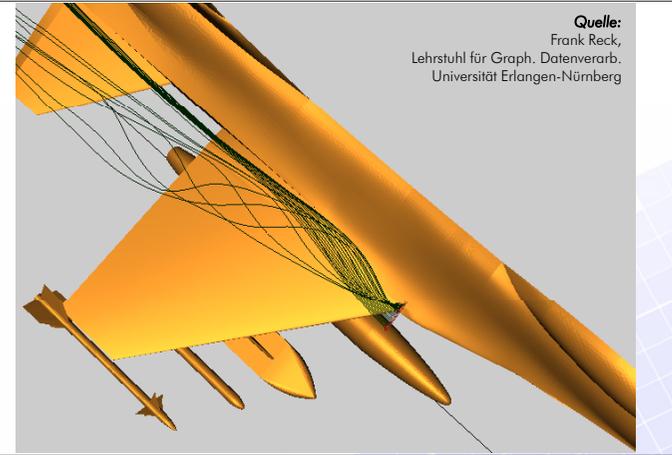
**Flow Field Probe:**  
 Stab wird vom Benutzer im Raum platziert und dient als Startpunkt für Partikelbahnen.

Andreas Kolb, Martin Lambers Visualisierung

## Partikelbahnen

56

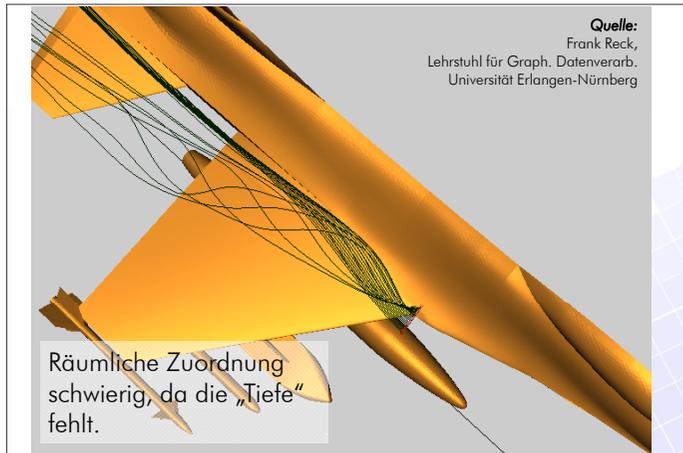
**Quelle:**  
 Frank Reck,  
 Lehrstuhl für Graph. Datenverarb.  
 Universität Erlangen-Nürnberg



Andreas Kolb, Martin Lambers Visualisierung

## Partikelbahnen

57



Andreas Kolb, Martin Lambers

Visualisierung

## Partikelbahnen in 3D

58

**Problem:** Bei 3D Partikelbahnen, die als einfache Linien dargestellt werden, fehlt der „räumliche Tiefeneindruck“

### Möglichkeiten:

#### ● Beleuchtungseffekte:

Räumliche Tiefe kann sehr gut durch *lokale Beleuchtung* erzeugt werden.

#### ● Erzeuge echte 3D Geometrie:

*Streamballs, Stream Ribbons, Stream Tubes*

#### ● Beleuchtungsverfahren für Linien:

*Beleuchtete Partikelbahnen*

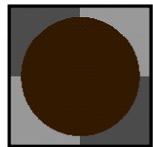
Andreas Kolb, Martin Lambers

Visualisierung

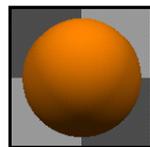
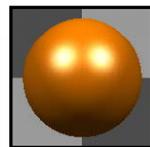
## Lokale Beleuchtung

59

### Blinn-Phong Beleuchtungsmodell:



ambient

diffuse  
(Lambert)spekular  
(spiegelnd)

$$I = I_a + I_d \langle \vec{n} \circ \vec{l} \rangle + I_s \langle \vec{n} \circ \vec{h} \rangle^r$$

#### ● Für Partikelbahnen:

Was ist die Oberflächennormale einer Linie?

Andreas Kolb, Martin Lambers

Visualisierung

## Beleuchtete Partikelbahnen

60

### Was ist ein geeigneter Normalenvektor einer Linie?

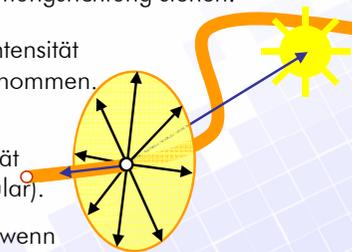
#### 1. Interpretiere die Partikelbahn als eine *sehr dünne Röhre*:

Die Normalenvektoren an einem Punkt sind alle Vektoren, die senkrecht auf die Strömungsrichtung stehen.

#### 2. Die hellste Beleuchtungsintensität wird am stärksten wahrgenommen.

Wähle den Vektor mit der max. Beleuchtungsintensität (separat für diffus & spekulär).

Diffus: Max. Beleuchtung wenn Normale coplanar mit Licht- und Strömungsrichtung.



Andreas Kolb, Martin Lambers

Visualisierung

### Beleuchtete Partikelbahnen 61

$$a^2 + b^2 = c^2$$

$$a = \sqrt{c^2 - b^2}$$

$$\langle \vec{n} \circ \vec{l} \rangle = \sqrt{1 - \langle \vec{t} \circ \vec{l} \rangle^2}$$

Andreas Kolb, Martin Lambers Visualisierung

### Beleuchtete Partikelbahnen 62

- Diffuser Term:  

$$I_d \langle \vec{n} \circ \vec{l} \rangle = I_d \sqrt{1 - \langle \vec{t} \circ \vec{l} \rangle^2}$$
- Spekularer Term (analog):  

$$I_s \langle \vec{n} \circ \vec{h} \rangle^r = I_s \sqrt{1 - \langle \vec{t} \circ \vec{h} \rangle^2}^r$$
- Blinn-Phong Modell für Linien:  

$$I = I_a + I_d \sqrt{1 - \langle \vec{t} \circ \vec{l} \rangle^2} + I_s \sqrt{1 - \langle \vec{t} \circ \vec{h} \rangle^2}^r$$

Andreas Kolb, Martin Lambers Visualisierung

### Beleuchtete Partikelbahnen 63

Quelle:  
Scientific Computing and Imaging,  
University of Utah  
sci.utah.edu

SCI INSTITUTE

Andreas Kolb, Martin Lambers Visualisierung

### Beleuchtete Partikelbahnen 64

● Beispiel:

Quelle:  
H.-C. Hege  
Konrad-Zuse-Zentrum  
Berlin

© 2000 Konrad-Zuse-Zentrum Berlin (Z3B)

Andreas Kolb, Martin Lambers Visualisierung

## Stream Balls

65

- Räumliche Tiefe durch „echte 3D Geometrie“  
Zeichne Partikelbahnen als Folge *kleiner Kugeln*



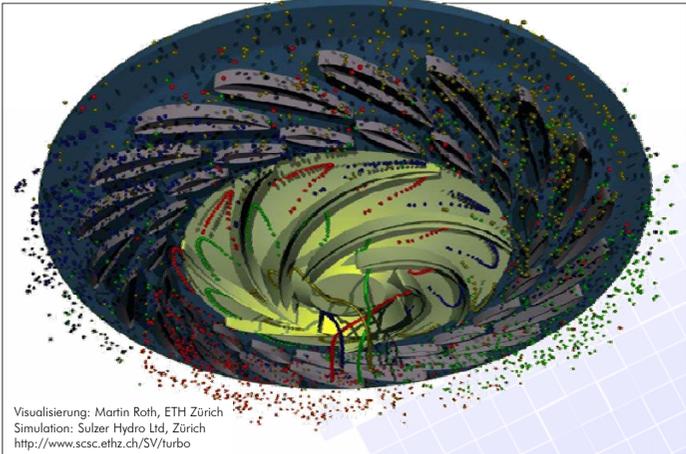
- Abstand der Kugeln gibt Information über Geschwindigkeitsbetrag.
- Farbe und Radius können weitere skalare Eigenschaften darstellen (z.B. Divergenz)

**Quelle:** Christian Teitzel, Graph. Datenverarb., Universität Erlangen-Nürnberg

Andreas Kolb, Martin Lambers Visualisierung

## Stream Balls

66



Visualisierung: Martin Roth, ETH Zürich  
Simulation: Sulzer Hydro Ltd, Zürich  
<http://www.spsc.ethz.ch/SV/turbo>

Andreas Kolb, Martin Lambers Visualisierung

## Stream Balls

67

Kugeln können miteinander verschmelzen (*Meta-Balls*) und komplexere Objekte (Röhren, Flächen) bilden.

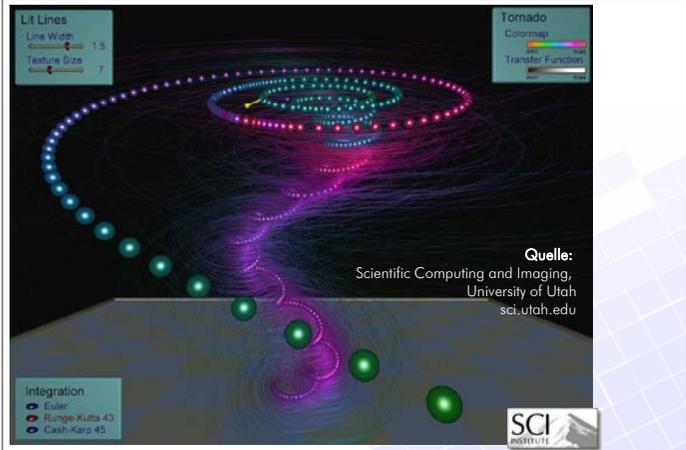


**Quelle:** Hans Hagen, IEEE Visualization 1994

Andreas Kolb, Martin Lambers Visualisierung

## Stream Balls und Partikelbahnen

68



**Lit Lines**

Line Width: 1.0

Texture Size: 7

**Tornado**

Colormap: [gradient]

Transfer Function: [gradient]

**Quelle:** Scientific Computing and Imaging, University of Utah, [sci.utah.edu](http://sci.utah.edu)

Andreas Kolb, Martin Lambers Visualisierung

## Literatur

69

### Glyphen:

- W.L. de Leeuw, J.J. van Wijk.

*A Probe for Local Flow Field Visualization.*

In *IEEE Visualization '93 Proceedings*, pages 39-45, 1993.

### Beleuchtete Partikelbahnen:

- M. Zöckler, D. Stalling, and H.-C. Hege.

*Interactive Visualisation of 3D-Vector Fields using Illuminated Streamlines.*

In *IEEE Visualization '96 Proceedings*, pages 107-113, 1996.

### Stream Balls

- M. Brill, H. Hagen, H. Rodrian, W. Djatschin, and S. Klimenko.

*Stream ball techniques for flow visualization.*

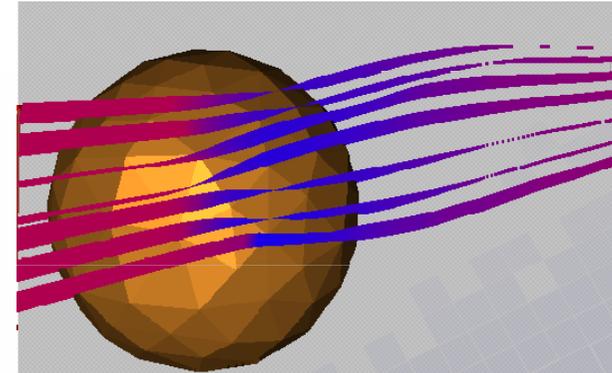
In *Proceedings Visualization '94*, pages 225-231, 1994.

Andreas Kolb, Martin Lambers

Visualisierung

## Strömungsbänder

70



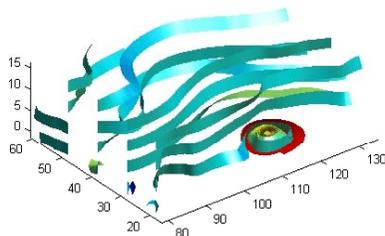
Andreas Kolb, Martin Lambers

Visualisierung

## Strömungsbänder

71

- Idee:** Zeige die lokale Rotation der Strömung durch Darstellung eines Bandes (*stream ribbon*) mit bestimmter Breite.



Streamribbins mit MATLAB  
Quelle: [www.mathworks.com](http://www.mathworks.com)

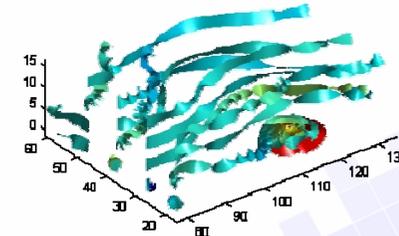
Andreas Kolb, Martin Lambers

Visualisierung

## Strömungsbänder

72

- Idee:** Zeige die lokale Rotation der Strömung durch Darstellung eines Bandes (*stream ribbon*) mit bestimmter Breite.



Streamribbins mit MATLAB  
Quelle: [www.mathworks.com](http://www.mathworks.com)

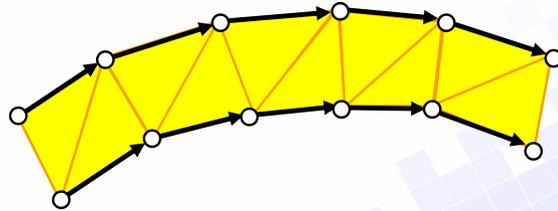
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>73</sup>

**Methode 1:** Zwei Partikelbahnen.

- Wähle zwei Startpunkte, die nahe beieinander liegen.
- Bestimme zwei Partikelbahnen
- Verbinde die beiden Bahnen durch Polygone (Dreiecke)



- **Vorteil:** Einfache Berechnung

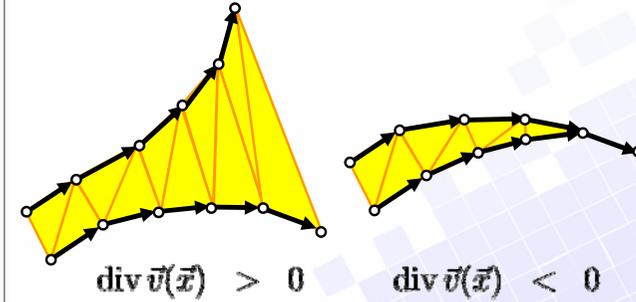
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>74</sup>

**Methode 1:** Zwei Partikelbahnen.

- **Nachteil:** Gutes Ergebnis nur bei sehr geringer Divergenz.



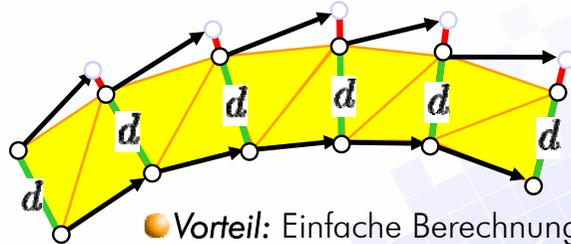
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>75</sup>

**Methode 2:** Methode 1 mit Korrektur.

- Wähle zwei Startpunkte, in bestimmtem Abstand  $d$ .
- Bestimme zwei Folgepunkte
- Korrigiere den Abstand durch Verschieben eines Punktes



- **Vorteil:** Einfache Berechnung
- **Nachteil:** „Nur eine Kante stimmt“

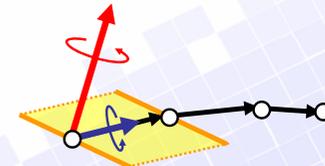
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>76</sup>

**Methode 3:** Direkte Bestimmung der Rotation

- Bestimme eine einzelne Partikelbahn.
- Für jeden Punkt
  - Bestimme den Rotationsvektor (durch Zent. Diff. + Interp.)
  - Projiziere den Rotationsvektor auf die Strömungsrichtung = Winkelgeschwindigkeit  $\omega_s$  der Rotation in Strömungsrichtung
- Lege eine Kante in den Startpunkt



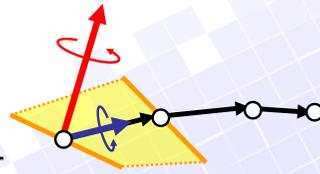
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>77</sup>

**Methode 3:** Direkte Bestimmung der Rotation

- Bestimme eine einzelne Partikelbahn.
- Für jeden Punkt
  - Bestimme den Rotationsvektor (*durch Zent. Diff. + Interp.*)
  - Projiziere den Rotationsvektor auf die Strömungsrichtung  
= Winkelgeschwindigkeit  $\omega_s$  der Rotation in Strömungsrichtung
  - Lege eine Kante in den Startpunkt
  - Drehe die Kante bei jedem Zeitschritt  $\tau$  um den Winkel  $\varphi = \omega_s \cdot \tau$



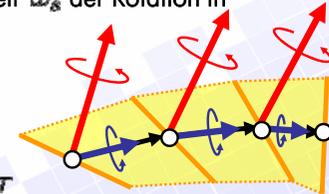
Andreas Kolb, Martin Lambers

Visualisierung

## Erzeugen von Strömungsbändern <sup>78</sup>

**Methode 3:** Direkte Bestimmung der Rotation

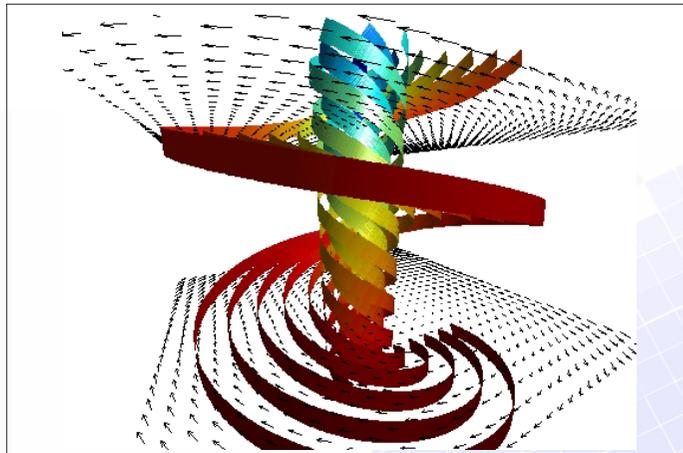
- Bestimme eine einzelne Partikelbahn.
- Für jeden Punkt
  - Bestimme den Rotationsvektor (*durch Zent. Diff. + Interp.*)
  - Projiziere den Rotationsvektor auf die Strömungsrichtung  
= Winkelgeschwindigkeit  $\omega_s$  der Rotation in Strömungsrichtung
  - Lege eine Kante in den Startpunkt
  - Drehe die Kante bei jedem Zeitschritt  $\tau$  um den Winkel  $\varphi = \omega_s \cdot \tau$



Andreas Kolb, Martin Lambers

Visualisierung

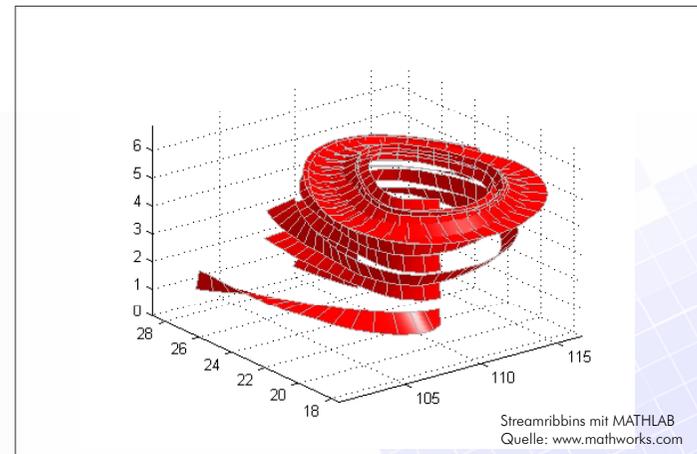
## Strömungsbänder <sup>79</sup>



Andreas Kolb, Martin Lambers

Visualisierung

## Strömungsbänder <sup>80</sup>



Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

81

### Strömungsröhren:

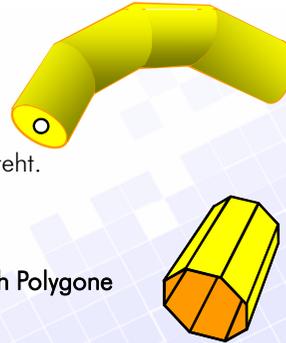
**Einfache Variante:** Zeiche die Partikelbahn als Röhre mit festem Radius.

**Bestimme eine Partikelbahn.**

**Für jeden Zeitschritt:**

- Bestimme einen Kreis mit festem Radius, der senkrecht auf der Strömungsrichtung steht.
- Verbinde die Kreise durch zylinderförmige Flächen.

Die Zylinderflächen werden durch Polygone approximiert und gezeichnet.



Andreas Kolb, Martin Lambers

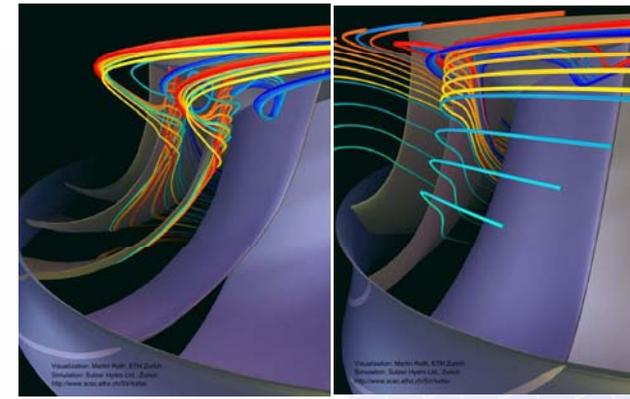
Visualisierung

## Stream Tubes

82

### Strömungsröhren

Visualisierung: Martin Roth, ETH Zürich  
Simulation: Sulzer Hydro Ltd, Zürich  
<http://www.scs.ethz.ch/SV/turbo>

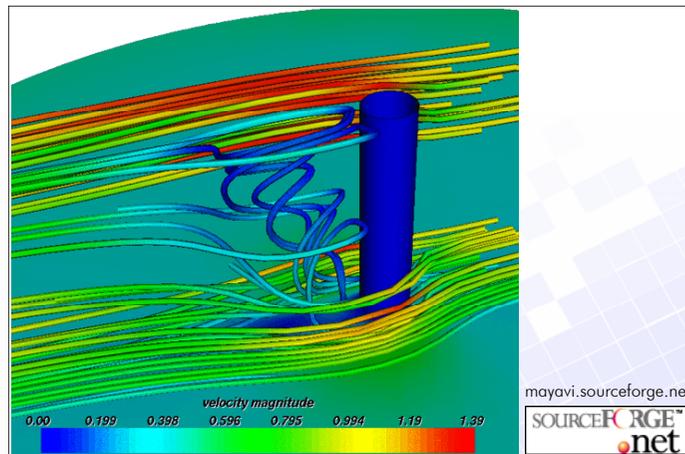


Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

83



Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

84

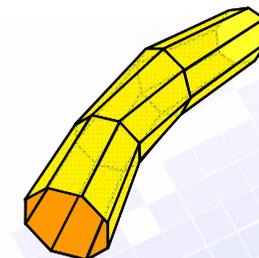
### Strömungsröhren:

**Aufwändigere Variante:** Zeiche eine Röhre, die aus mehreren Partikelbahnen besteht.

**Wähle einen Kreis.**

**Für jeden Punkt des Kreises:**

- Bestimme eine Partikelbahn.
- Verbinde die Kreise aufeinanderfolgender Zeitschritte durch zylinderförmige Flächen.



Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

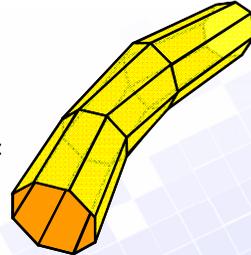
85

### ● Strömungsröhren:

**Aufwändigere Variante:** Zeiche eine Röhre, die aus mehreren Partikelbahnen besteht.

#### Physikalische Eigenschaften dieser Variante:

- Die „Wand“ der Röhre ist **impermeabel** (=undurchdringlich): Partikel innerhalb der Röhre verlassen die Röhre niemals!
- Bei **inkompressibler Strömung:** Aufgrund der Massenerhaltung bedeutet eine Änderung des Durchmessers der Röhre eine lokale Änderung der Geschwindigkeit.



Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

86

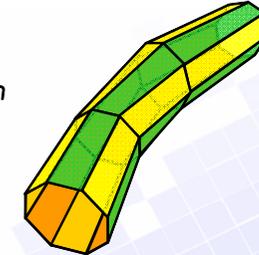
### ● Strömungsröhren:

**Aufwändigere Variante:** Zeiche eine Röhre, die aus mehreren Partikelbahnen besteht.

Durch **Einfärben der einzelnen Streifen** kann die lokale **Rotation in Strömungsrichtung**

$$\omega_s = \frac{\langle \text{rot } \vec{v} \circ \vec{v} \rangle}{\|\vec{v}\|}$$

visualisiert werden

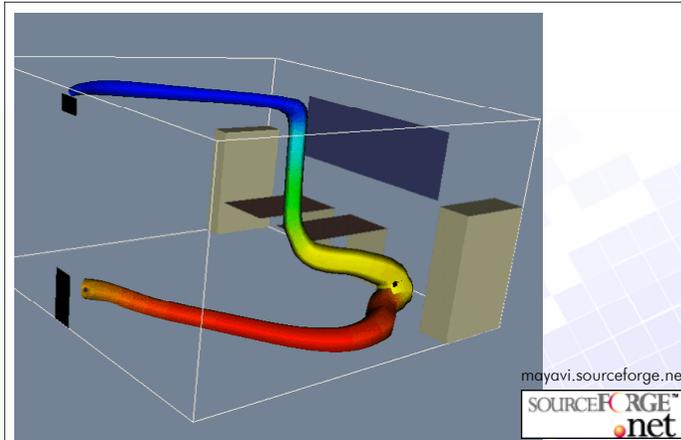


Andreas Kolb, Martin Lambers

Visualisierung

## Stream Tubes

87



mayavi.sourceforge.net  
SOURCEFORGE  
net

Andreas Kolb, Martin Lambers

Visualisierung

## Literatur

88

### Stream Ribbons & Stream Tubes:

- S.K. Ueng, C. Sikorski, K.L. Ma.  
*Efficient Streamline, Streamribbon and Streamtube Construction on Unstructured Grids.*  
In *IEEE Trans. on Comp. graph & Vis. (TVCG)*, 2(2), pages 100-110, June 1996.

Andreas Kolb, Martin Lambers

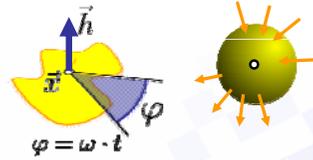
Visualisierung

## Zusammenfassung

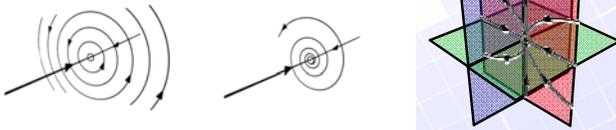
89

### Differentialrechnung für 3D Vektorfelder

- Jacobi-Matrix
- Divergenz
- Rotation
- Nabla-Operator



### Kritische Punkte in 3D:



Andreas Kolb, Martin Lambers

Visualisierung

## Zusammenfassung

90

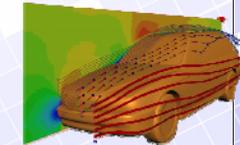
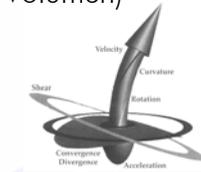
### Visualisierungsverfahren

#### Direkte Verfahren

- Farbkodierung (Schnittebenen/Volumen)
- Isoflächen
- Vektorpfeile
- Glyphen/Icons

#### Integrationsbasierte Verfahren

- Beleuchtete Partikelbahnen
- Stream Balls
- Strömungsbänder
- Stream Tubes



Andreas Kolb, Martin Lambers

Visualisierung

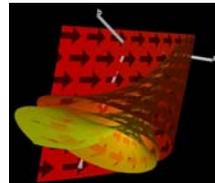
## Nächste Stunde

91

### Visualisierungsverfahren

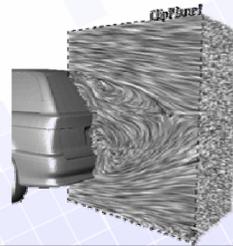
#### Flächenbasierte Verfahren

- Stream Surfaces
- Time Surfaces



#### Volumen-basierte Verfahren

- Flow Volumes
- 3D LIC



Andreas Kolb, Martin Lambers

Visualisierung