

# Visualisierung: 3D Strömungsdaten

Christof Rezk-Salama

Visualisierung WS 04/05, 30.11.2004

computergraphik und multimedia systeme  
universität siegen



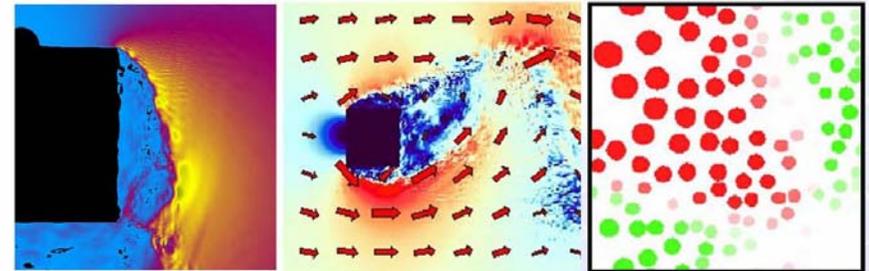
2

## Letzte Stunde: 2D Strömungsdaten

Visualisierungsverfahren für 2D Strömungsdaten

### Direkte Verfahren

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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

3

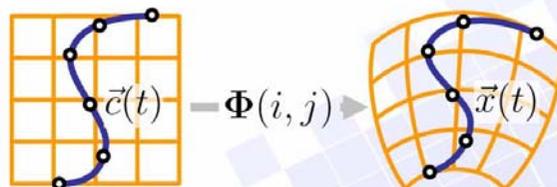
## Letzte Stunde: 2D Strömungsdaten

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*



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

4

## 2D Strömungsdaten

Visualisierungsverfahren für 2D Strömungsdaten

### Integrationsbasierte Verfahren

#### ● Texturbasierte Verfahren

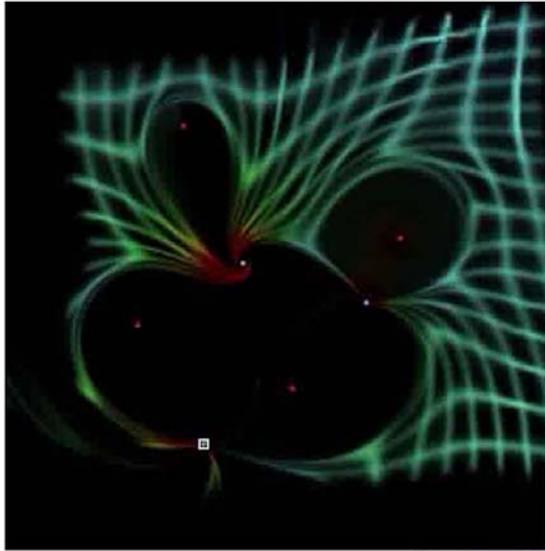
- Line Integral Convolution (LIC)  
Faltung eines Rauschfeldes entlang der Strömungslinien eines Vektorfeldes
- Texture Advection  
Bildbasierter Ansatz: Verzerre ein Ergebnisbild schrittweise in Richtung der Strömung



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

5



*Texture Advection:*  
„Verzerre ein Eingabebild entlang der Strömungslinien des Vektorfeldes“

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

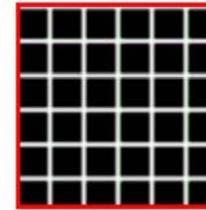
<http://www.win.tue.nl/~vanwijk/ibfv/>

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

*Texture Advection Implementierung:*



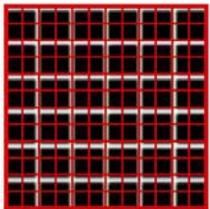
- Definiere das Eingabebild als *Polygon mit Textur.*

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

*Texture Advection Implementierung:*



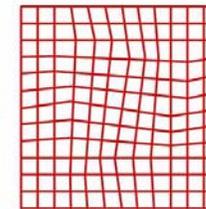
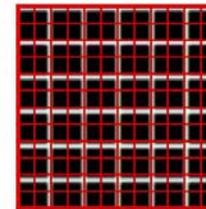
- Definiere das Eingabebild als *Polygon mit Textur.*
- Zerlege das Polygon in *viele kleine Polygone*

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

*Texture Advection Implementierung:*



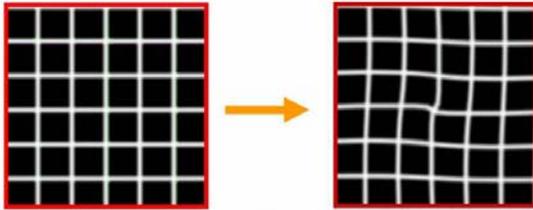
- Definiere das Eingabebild als *Polygon mit Textur.*
- Zerlege das Polygon in *viele kleine Polygone*
- Bewege jeden Polygon-Vertex *einen Zeitschritt* weiter.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

## Texture Advection Implementierung:



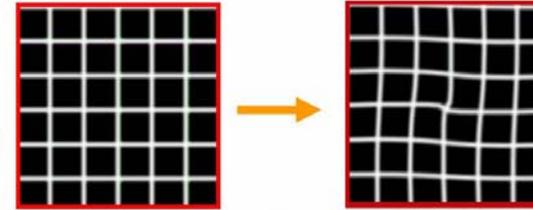
- Definiere das Eingangsbild als *Polygon mit Textur*.
- Zerlege das Polygon in *viele kleine Polygone*
- Bewege jeden Polygon-Vertex *einen Zeitschritt* weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

## 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

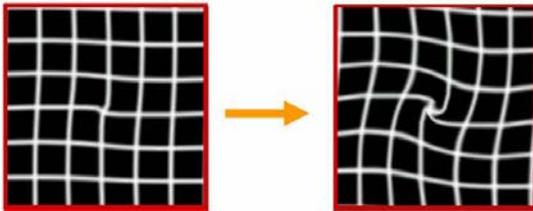
Lies das Bild aus dem Frame-Buffer und kopiere es in die Eingangstextur

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

## 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

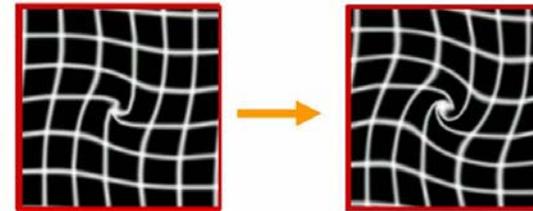
Lies das Bild aus dem Frame-Buffer und kopiere es in die Eingangstextur

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

## 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

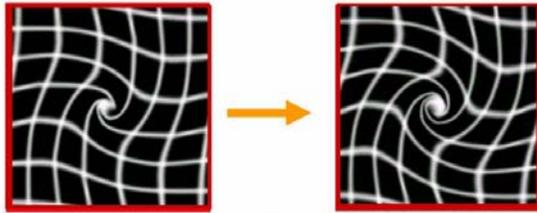
Lies das Bild aus dem Frame-Buffer und kopiere es in die Eingangstextur

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

6

## Texture Advection Implementierung:



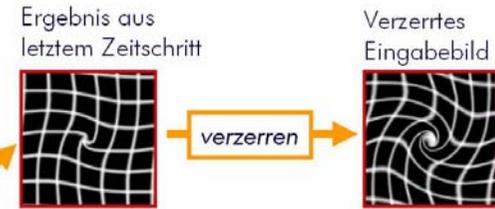
- Definiere das Eingangsbild als **Polygon mit Textur**.
- Zerlege das Polygon in **viele kleine Polygone**
- Bewege jeden Polygon-Vertex **einen Zeitschritt** weiter.
- Zeichne die verzerrten Polygone in den Frame-Buffer

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

7

## Erweiterte Implementierung:

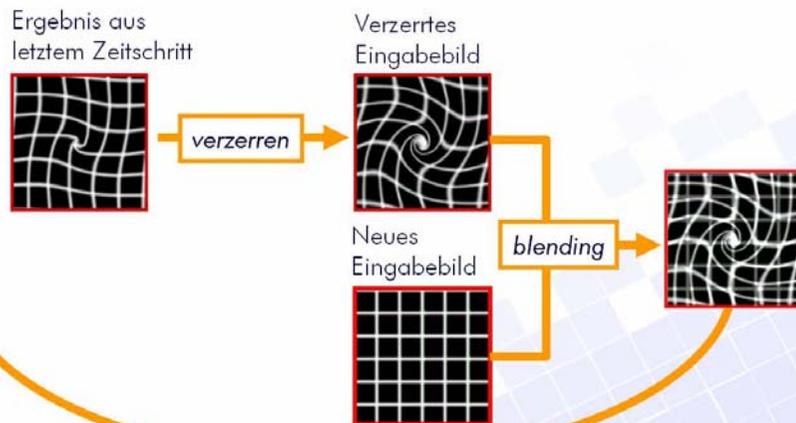


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Texturbasierte Techniken

7

## Erweiterte Implementierung:



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

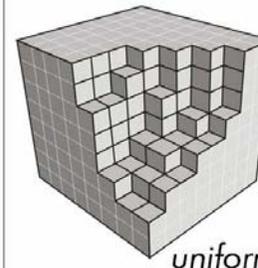
# 3D Vektorfelder

12

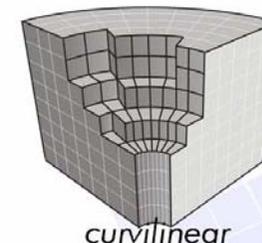
## Definition: 3D Vektorfeld

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

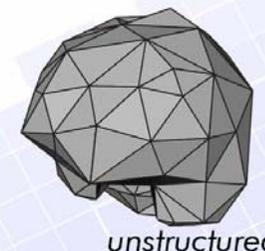
$$\vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$



uniform



curvilinear



unstructured

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Jacobi-Matrix in 3D

13

- 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  $\vec{h}$  ?

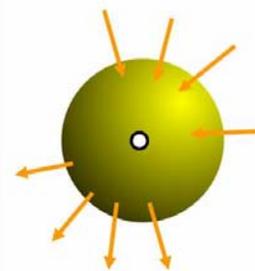
$$\frac{\partial \vec{v}(\vec{x})}{\partial \vec{h}} = \mathbf{J}_{\vec{v}}(\vec{x}) \vec{h}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Divergenz in 3D

14

- 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 infinitesimal kleines Volumen um den Punkt
- Wieviel fließt hinein, wieviel hinaus?

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

Senke, „Zusammenströmen“

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

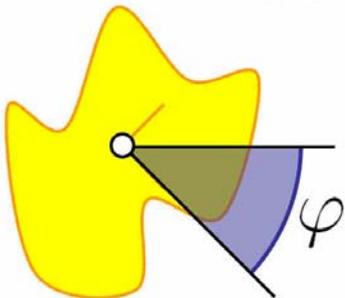
Quelle, „Auseinanderströmen“

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 2D

15

- 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:  $\omega = \frac{\varphi}{t}$

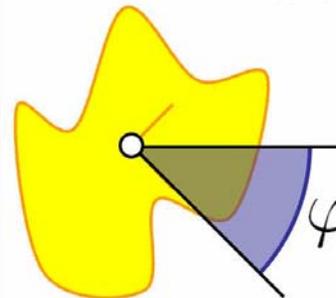
Bei zeitabhängiger Winkelgeschwindigkeit:  $\omega(t) = \frac{\partial \varphi(t)}{\partial t}$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 2D

15

- 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:

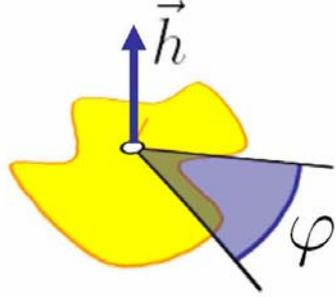
Winkelgeschwindigkeit  $\omega = \frac{1}{2} \operatorname{rot} \vec{v}(\vec{x})$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Rotation starrer Körper

16

## ● 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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Rotation in 3D

17

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Rotation in 3D

17

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Rotation in 3D

17

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

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 3D

17

- Betrachte die 2D Rotation in den einzelnen Ebenen:

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

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 3D

17

- Betrachte die 2D Rotation in den einzelnen Ebenen:

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

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

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 3D

17

- Betrachte die 2D Rotation in den einzelnen Ebenen:

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

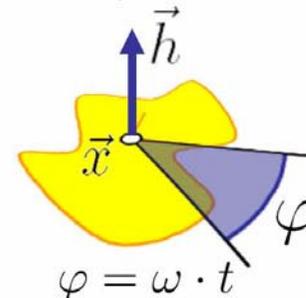
$$\text{rot } \vec{v}(\vec{x}) = \begin{pmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{pmatrix}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 3D

18

- 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:

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Rotation in 3D

19

- 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 Achse)

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

20

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

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

20

$$\begin{array}{l} \frac{\partial}{\partial x} \leftarrow \text{Differentialoperator (partielle Ableitung in x-Richtung)} \\ \frac{\partial}{\partial y} \leftarrow \text{Differentialoperator (partielle Ableitung in y-Richtung)} \\ \frac{\partial}{\partial z} \leftarrow \text{Differentialoperator (partielle Ableitung in z-Richtung)} \end{array}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

20

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

21

- 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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

22

- 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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Ableitungen in 3D

23

- 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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

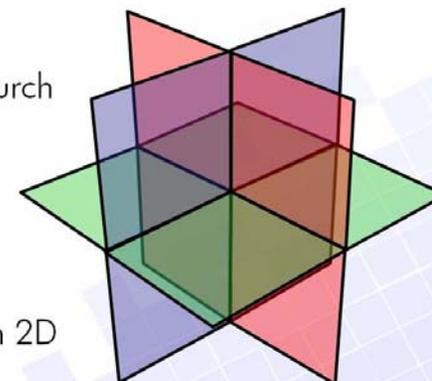
## Kritische Punkte in 3D

24

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

Prinzip:

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



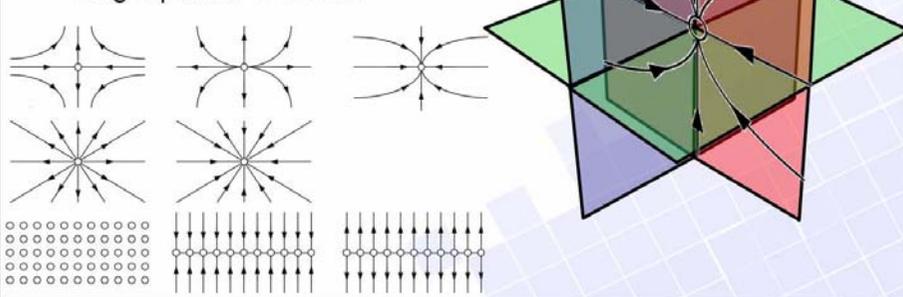
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Kritische Punkte in 3D

25

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

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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

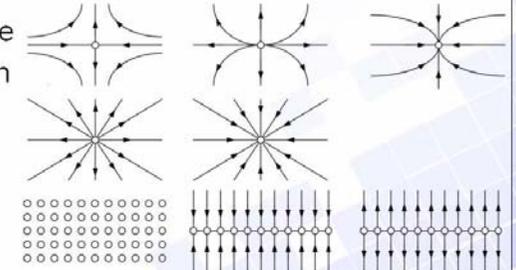
## Kritische Punkte in 3D

26

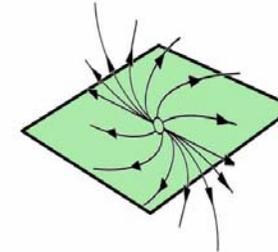
- Bestimmung und Klassifikation analog zu 2D:  
Eigenwerte und Eigenvektoren der Jacobi-Matrix

**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:



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Kritische Punkte in 3D

27

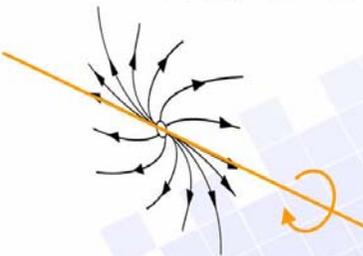
- Bestimmung und Klassifikation analog zu 2D:  
Eigenwerte und Eigenvektoren der Jacobi-Matrix

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

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



Beispiel:

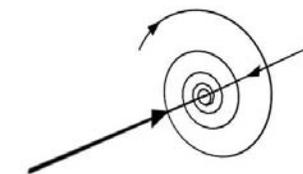
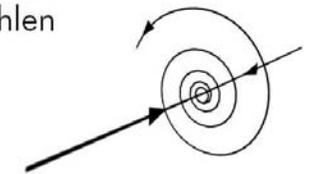
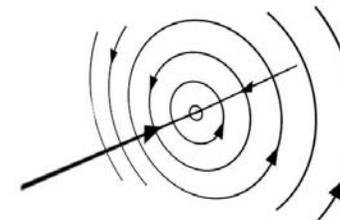


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Kritische Punkte in 3D

28

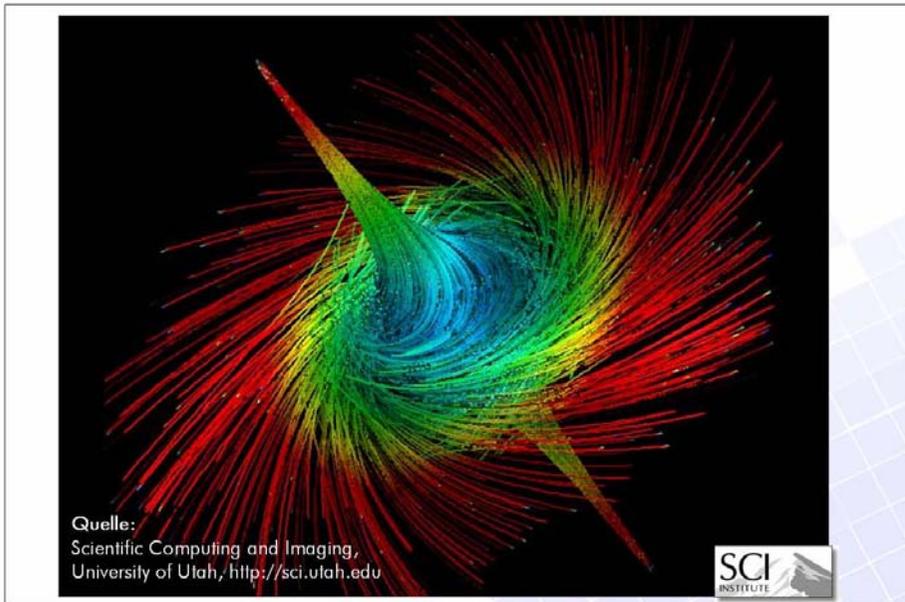
**Komplexe Eigenwerte** treten nur als Paar konjugiert komplexer Zahlen auf. Daher existiert bei einer  $3 \times 3$  Matrix immer mindestens ein reeller Eigenwert!



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Kritische Punkte in 3D



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Visualisierungsverfahren

## Direkte Verfahren

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

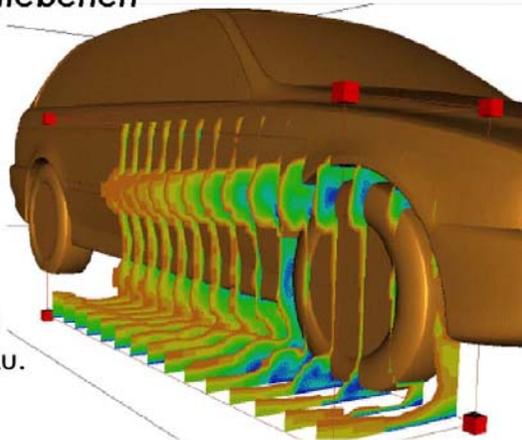
- Vektorpfeile
- Glyphen/Icons

## Integrationsbasierte Verfahren

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Farbkodierung

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



Transparenz/Opazität  
kommt als zusätzliche  
Farbkomponente hinzu.

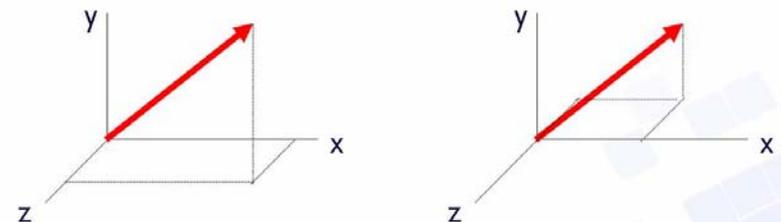
Quelle: Martin Schulz, Lehrstuhl für Visualisierung  
und Interaktive Systeme, Universität Stuttgart

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

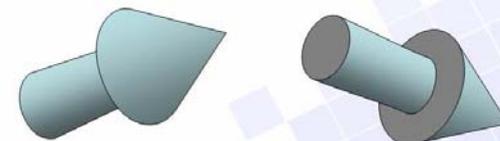
# Arrow Plots

## ● Linien und einfache Pfeile:

Darstellung wird erschwert durch Perspektive



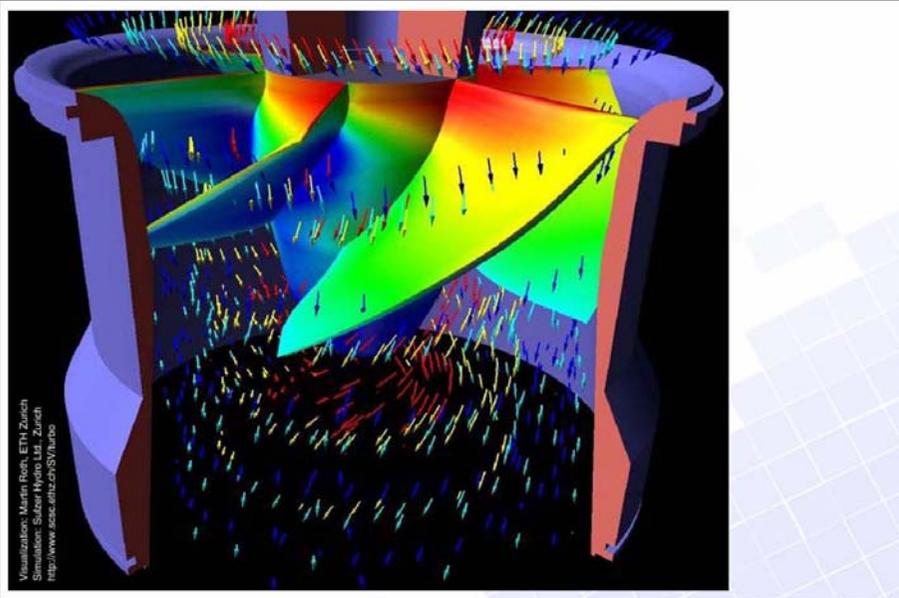
## ● Besser, aber auch aufwändiger: *Echte 3D Pfeile*



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Beispiel: Arrow Plots in 3D

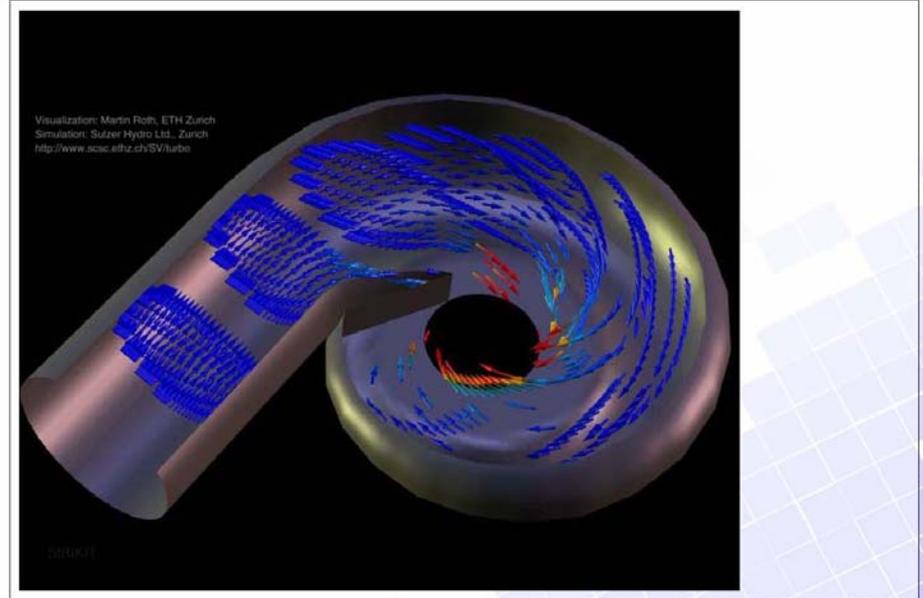
33



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Beispiel: Arrow Plots in 3D

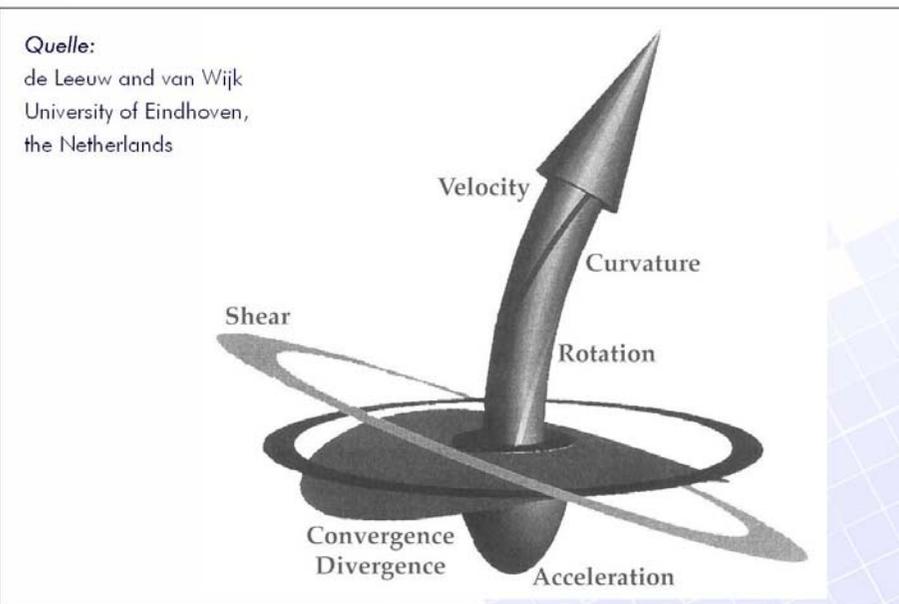
34



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Glyphen/Icons in 3D

35



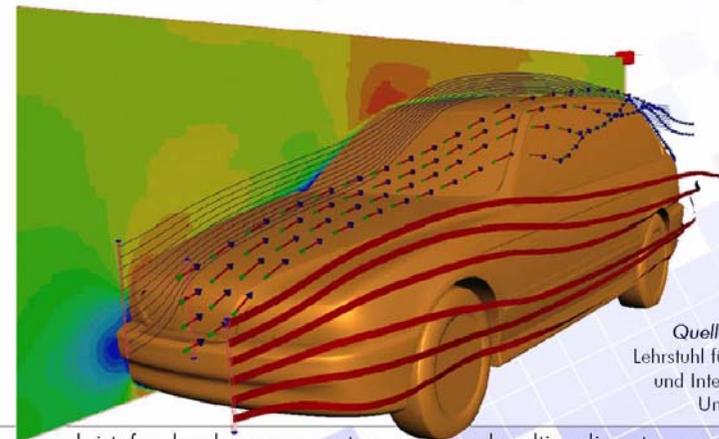
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Integrationsbasierte Verfahren

36

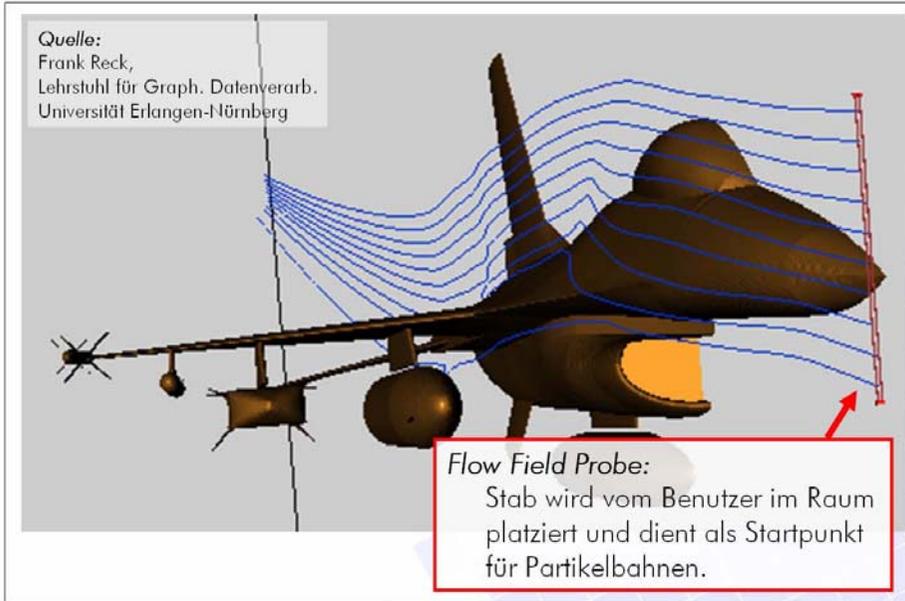
### ● Linienbasierte Techniken:

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

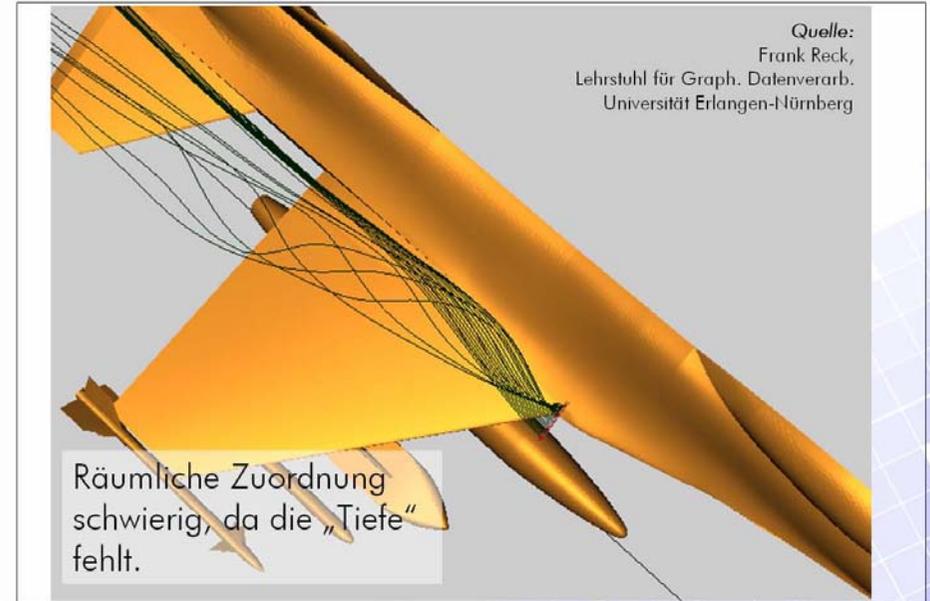


Quelle: Martin Schulz,  
Lehrstuhl für Visualisierung  
und Interaktive Systeme,  
Universität Stuttgart

christof rezk-salama, computergraphik und multimediasysteme, universität siegen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

**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:

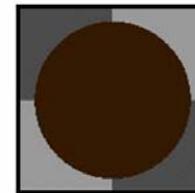
**Streamballs, Stream Ribbons, Stream Tubes**

- Beleuchtungsverfahren für Linien:

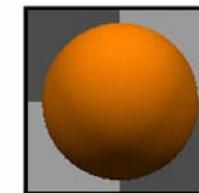
**Beleuchtete Partikelbahnen**

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

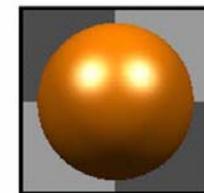
**Blinn-Phong Beleuchtungsmodell:**



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?

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

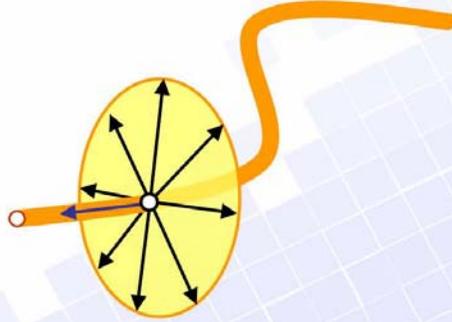
# Beleuchtete Partikelbahnen

41

Was nehme ich als Normalenvektor der Linie?

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

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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Beleuchtete Partikelbahnen

41

Was nehme ich als Normalenvektor der Linie?

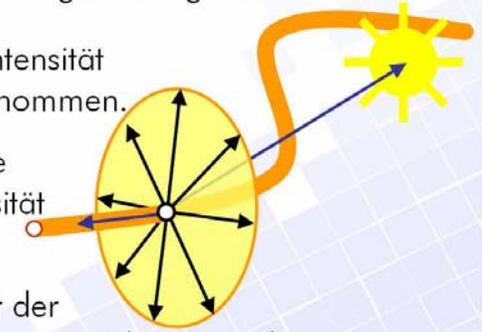
1. Interpretiere die Partikelbahn als eine *sehr dünne Röhre*:

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, der die hellste Beleuchtungsintensität erzeugt.

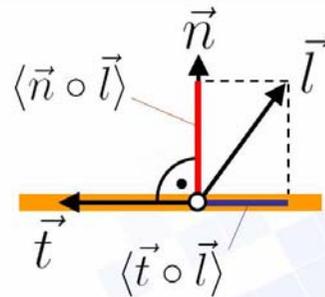
Das ist der Vektor, der mit der Lichttrichtung und der Strömungsrichtung coplanar ist.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Beleuchtete Partikelbahnen

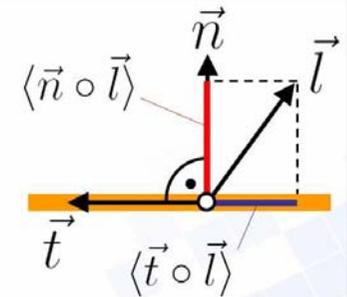
42



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Beleuchtete Partikelbahnen

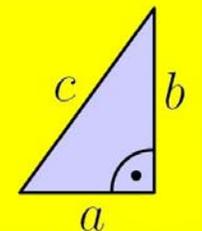
42



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

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

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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Beleuchtete Partikelbahnen

42

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

- Diffuser Term:

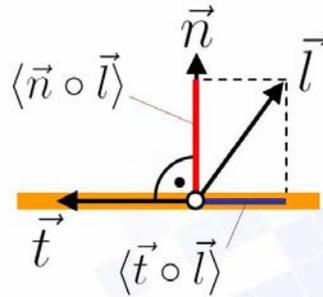
$$I_d \langle \vec{n} \circ \vec{l} \rangle = I_d \sqrt{1 - \langle \vec{t} \circ \vec{l} \rangle^2}$$

- 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:

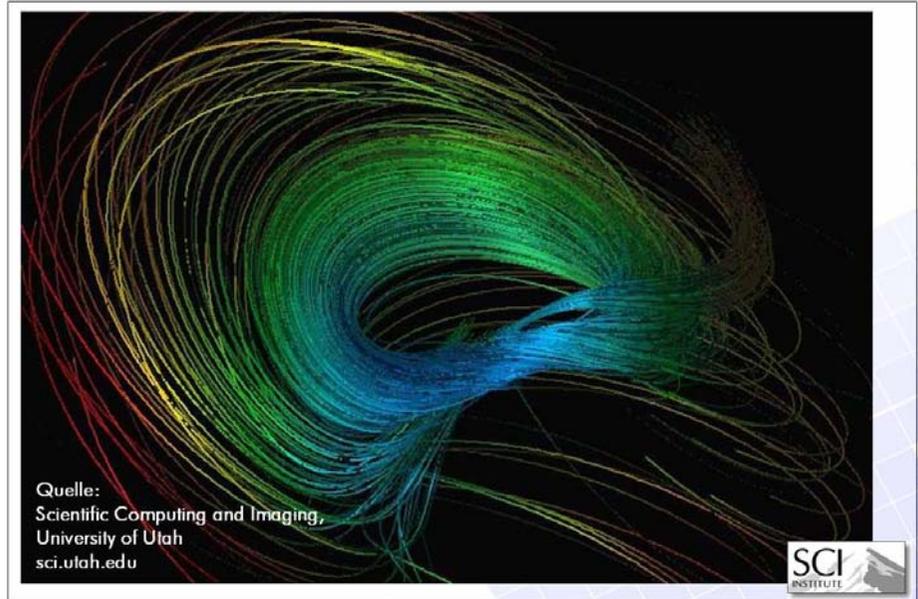
$$I = I_a + I_d \sqrt{1 - \langle \vec{t} \circ \vec{l} \rangle^2} + I_s \sqrt{1 - \langle \vec{t} \circ \vec{h} \rangle^2}^r$$



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Beleuchtete Partikelbahnen

43



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

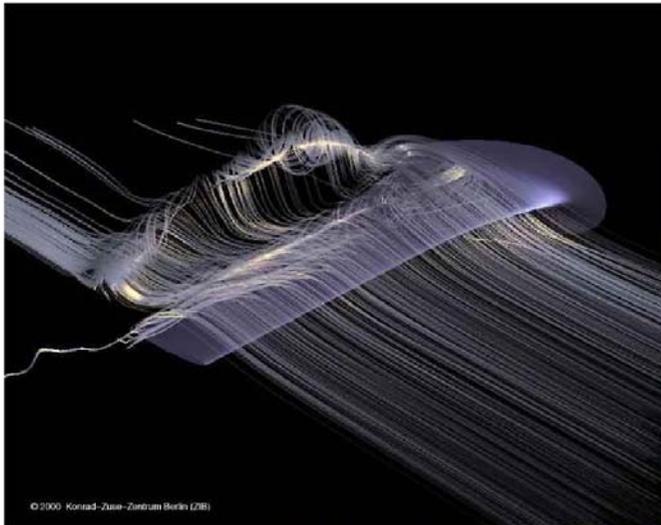


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Beleuchtete Partikelbahnen

44

- Beispiel:



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

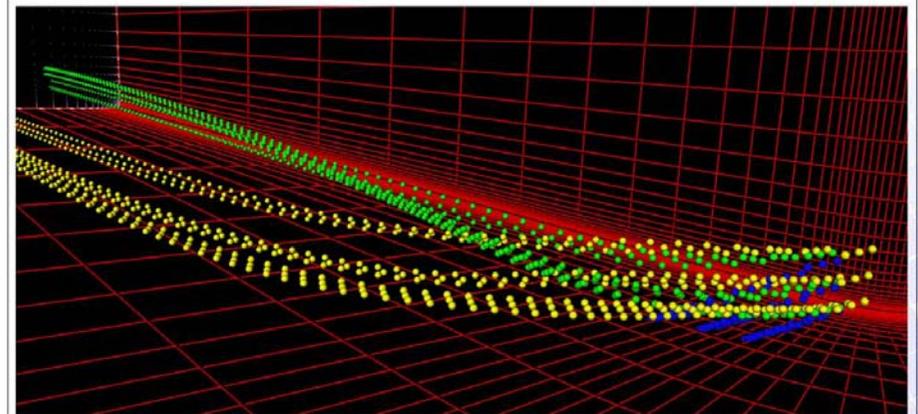
© 2000 Konrad-Zuse-Zentrum Berlin (Z3B)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Stream Balls

45

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



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Stream Balls

45

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



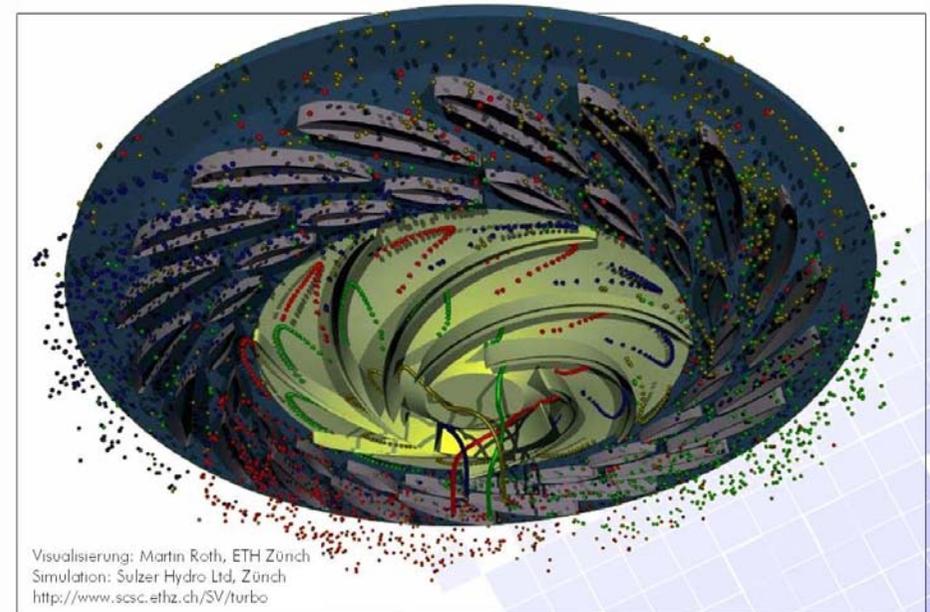
- Abstand der Kugeln gibt Information über Geschwindigkeitsbetrag.
- Farbe und Radius können weitere skalare Eigenschaften darstellen (z.B. Divergenz)

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Stream Balls

46



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Stream Balls

47

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

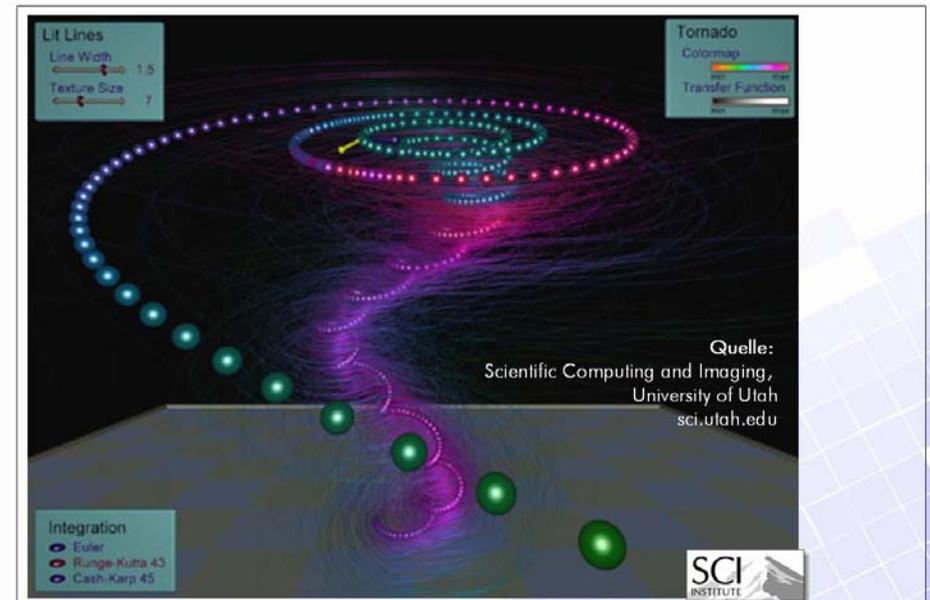


Quelle: Hans Hagen, IEEE Visualization 1994

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Stream Balls und Partikelbahnen

48



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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 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:

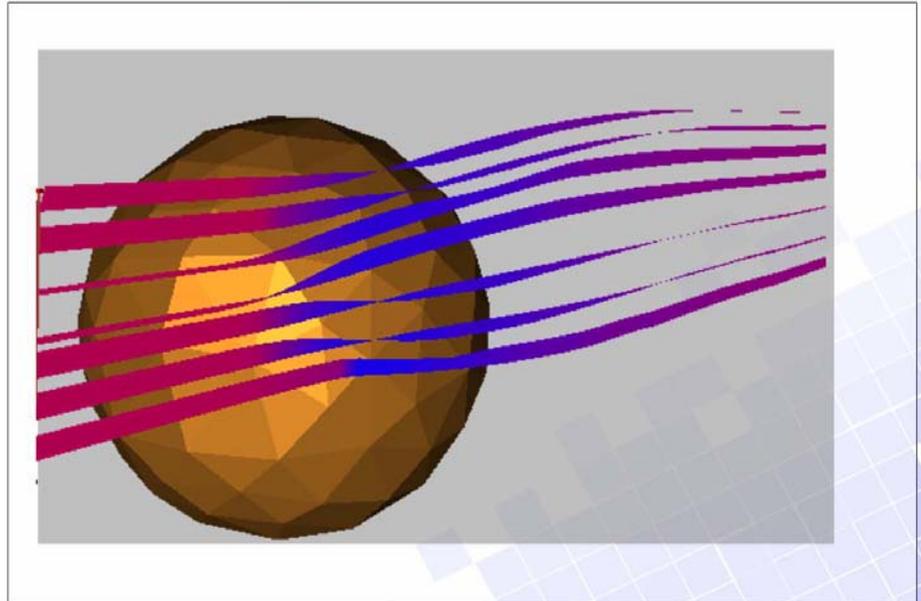
- M. Zöckler, D. Stalling, and H.-C. Hege.  
*Interactive Visualisation of 3D-Vector Fields using Illuminated Streamlines.*

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

## Stream Balls

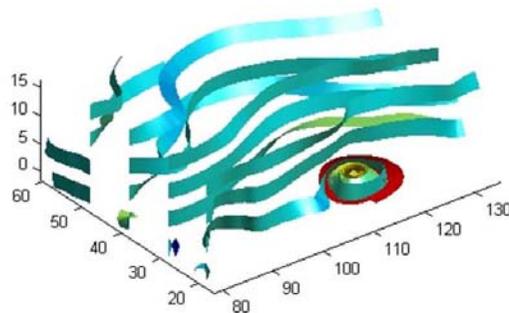
- M. Brill, H. Hagen, H. Rodrian, W. Djatschin, and S.Klimenko.  
*Stream ball techniques for flow visualization.*  
In *Proceedings Visualization '94*, pages 225-231, 1994.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

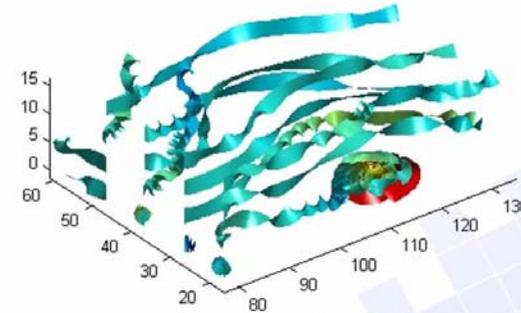
- **Idee:** Zeige die lokale Rotation der Strömung durch Darstellung eines Bandes (*stream ribbon*) mit bestimmter Breite.



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

- **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)

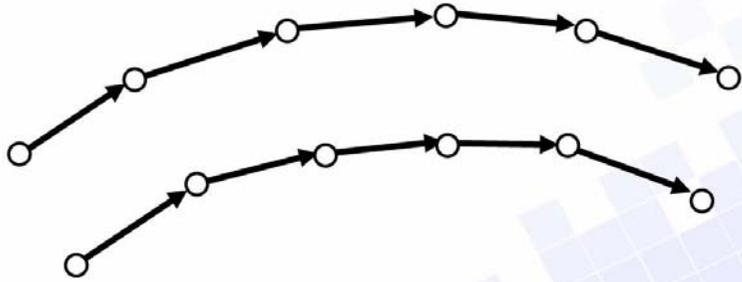
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

52

### Methode 1: Zwei Partikelbahnen.

- Wähle zwei Startpunkte, die nahe beieinander liegen.
- Bestimme zwei Partikelbahnen



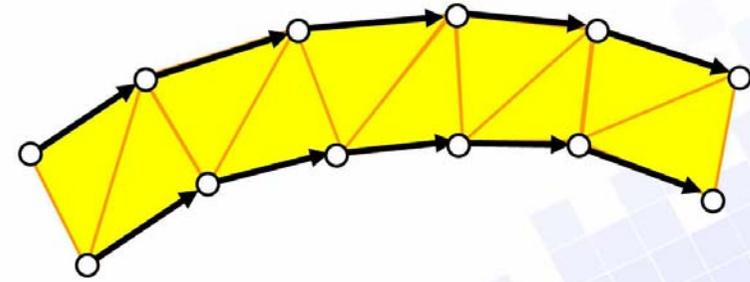
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

52

### Methode 1: Zwei Partikelbahnen.

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



- **Vorteil:** Einfache Berechnung

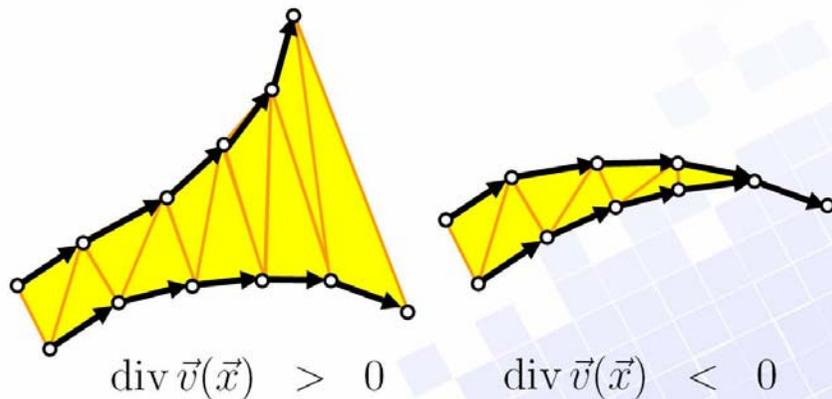
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

53

### Methode 1: Zwei Partikelbahnen.

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



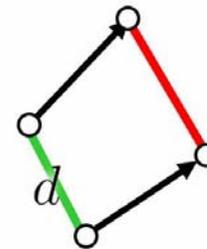
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

54

### Methode 2: Methode 1 mit Korrektur.

- Wähle zwei Startpunkte, in bestimmtem Abstand  $d$ .
- Bestimme zwei Folgepunkte



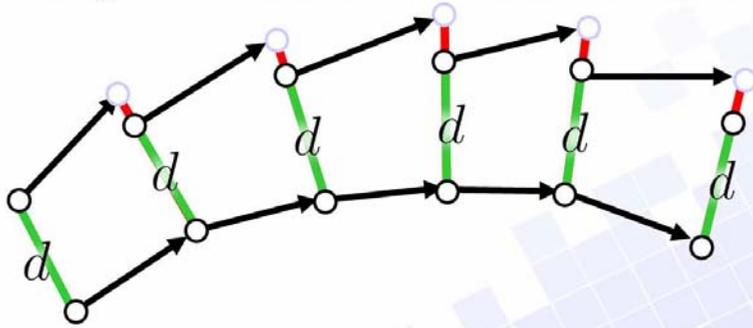
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

54

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

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



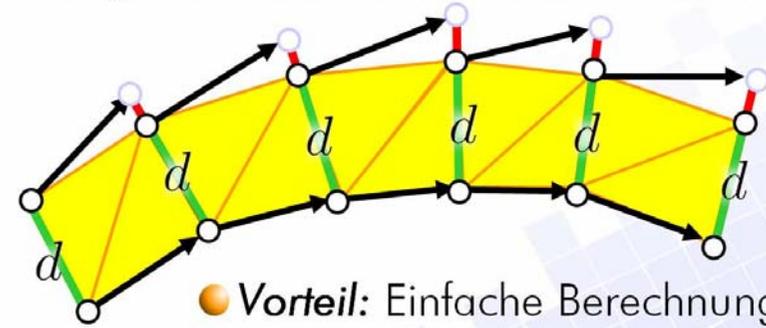
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

54

**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“

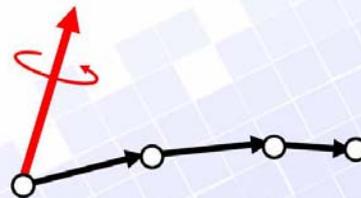
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

55

**Methode 3:** Direkte Bestimmung der Rotation

- Bestimme eine einzelne Partikelbahn.
- Für jeden Punkt
  - Bestimme den Rotationsvektor (durch Zent. Diff. + Interp.)



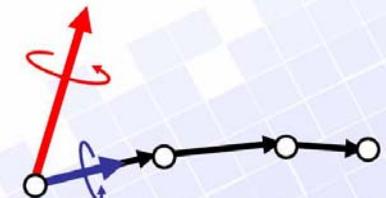
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern

55

**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

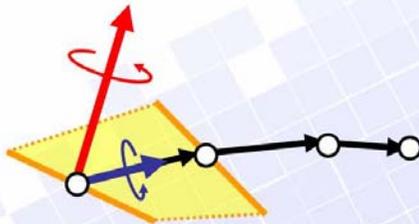


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern <sup>55</sup>

### Method 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  $\varphi = \omega_s \cdot \tau$

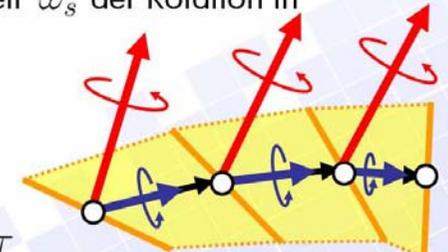


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Erzeugen von Strömungsbändern <sup>55</sup>

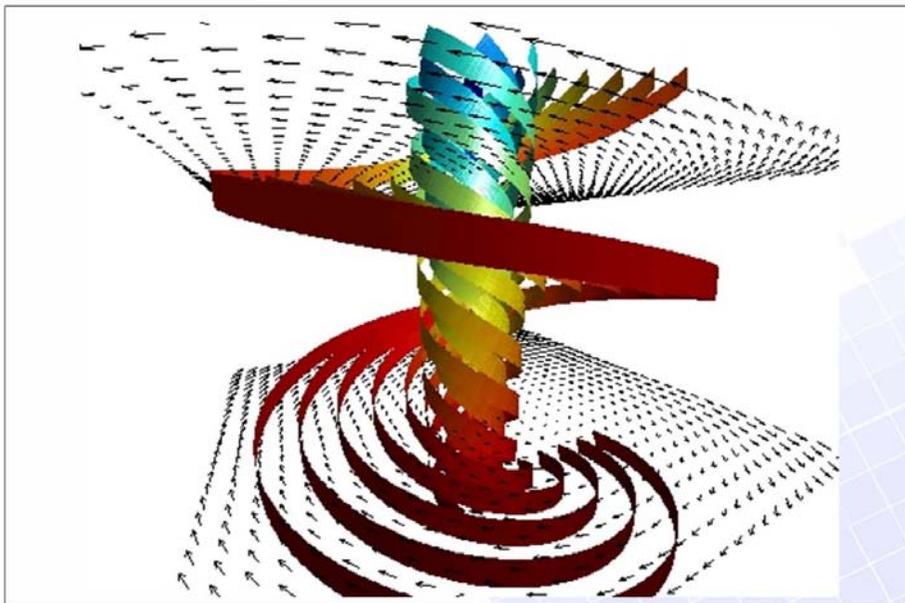
### Method 3: Direkte Bestimmung der Rotation

- 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  $\varphi = \omega_s \cdot \tau$



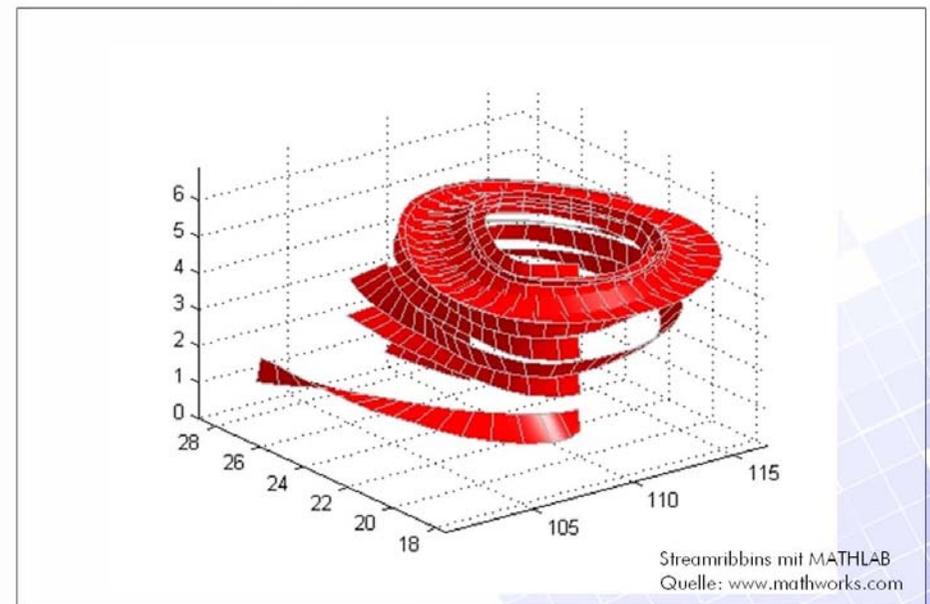
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Strömungsbänder <sup>56</sup>



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Strömungsbänder <sup>57</sup>



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Stream Tubes

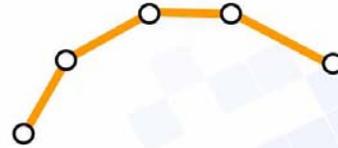
58

### ● Strömungsröhren:

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

*Bestimme eine Partikelbahn.*

*Für jeden Zeitschritt:*



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Stream Tubes

58

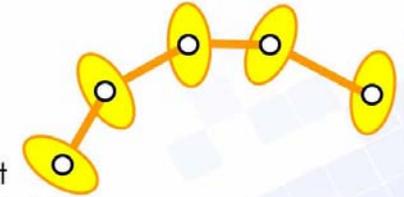
### ● Strömungsröhren:

*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.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Stream Tubes

58

### ● 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.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Stream Tubes

58

### ● Strömungsröhren:

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

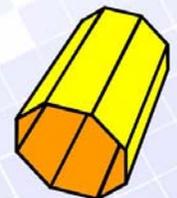
*Bestimme eine Partikelbahn.*

*Für jeden Zeitschritt:*

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



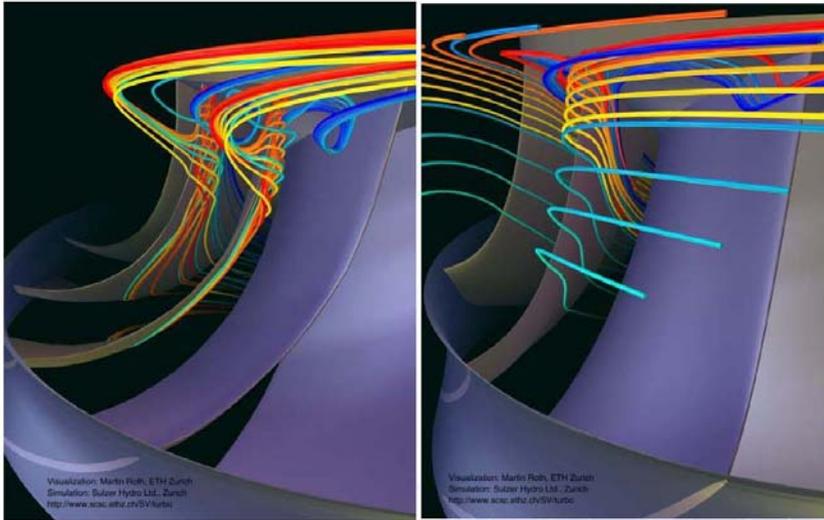
Die Zylinderflächen werden durch Polygone approximiert und gezeichnet.



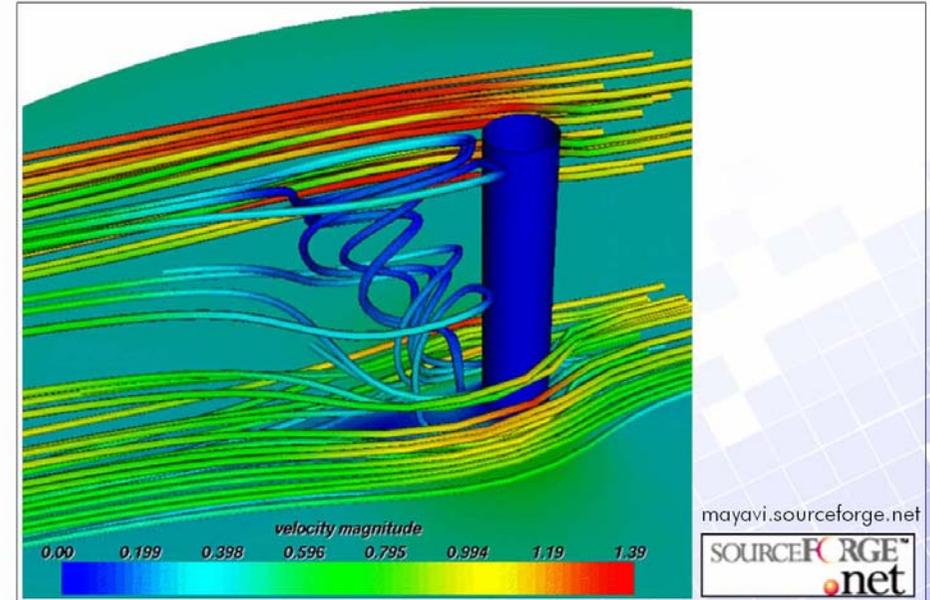
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Strömungsröhren

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



christof rezk-salama, computergraphik und multimediasysteme, universität siegen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

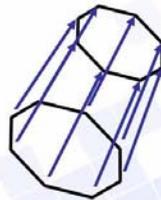
## ● Strömungsröhren:

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

*Wähle einen Kreis.*

*Für jeden Punkt des Kreises:*

- Bestimme eine Partikelbahn.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

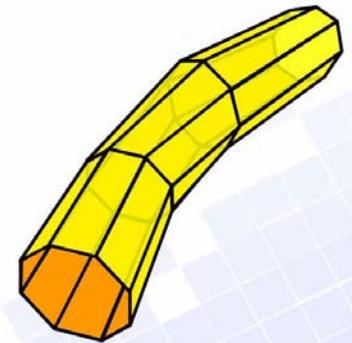
## ● Strömungsröhren:

*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.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

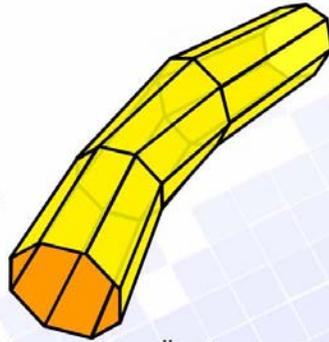
## ● Strömungsröhren:

*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.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

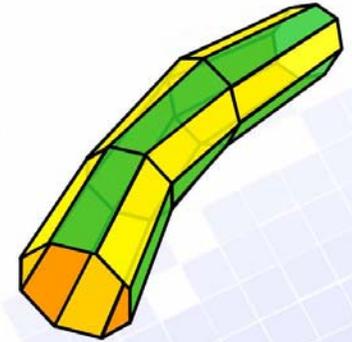
## ● Strömungsröhren:

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

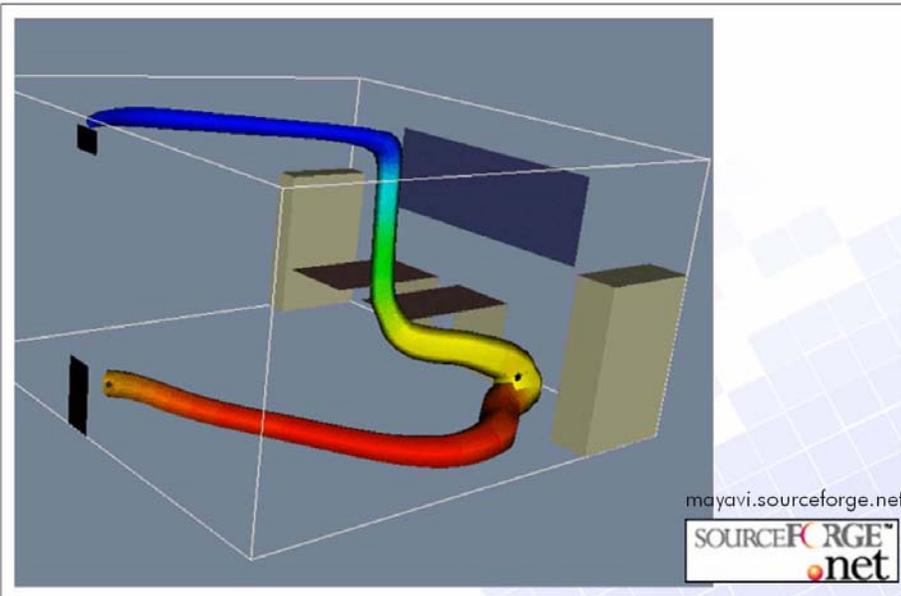
Durch *Einfärben der einzelnen Streifen* kann die lokale *Rotation in Strömungsrichtung*

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

visualisiert werden



christof rezk-salama, computergraphik und multimediasysteme, universität siegen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 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.

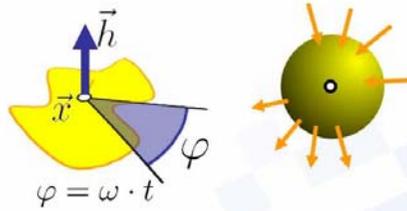
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Zusammenfassung

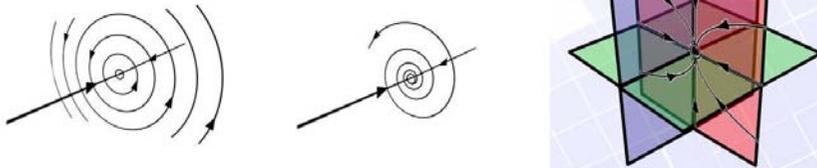
66

## Differentialrechnung für 3D Vektorfelder

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



## Kritische Punkte in 3D:



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Zusammenfassung

67

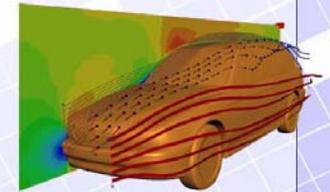
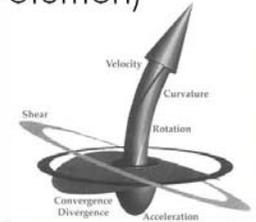
## Visualisierungsverfahren

### Direkte Verfahren

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

### Integrationsbasierte Verfahren

- Beleuchtete Partikelbahnen
- Stream Balls
- Strömungsbänder
- Stream Tubes



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

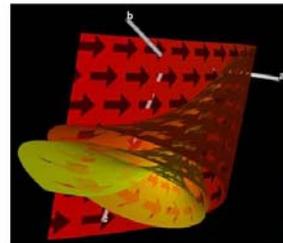
# Nächste Stunde

68

## Visualisierungsverfahren

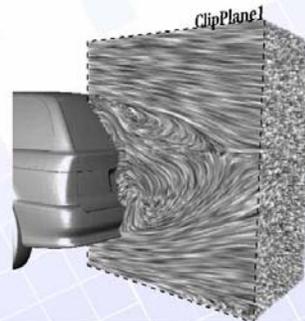
### Flächenbasierte Verfahren

- Stream Surfaces
- Time Surfaces



### Volumen-basierte Verfahren

- Flow Volumes
- 3D LIC



christof rezk-salama, computergraphik und multimediasysteme, universität siegen