

Visualisierung: 2D Strömungsdaten

Christof Rezk-Salama

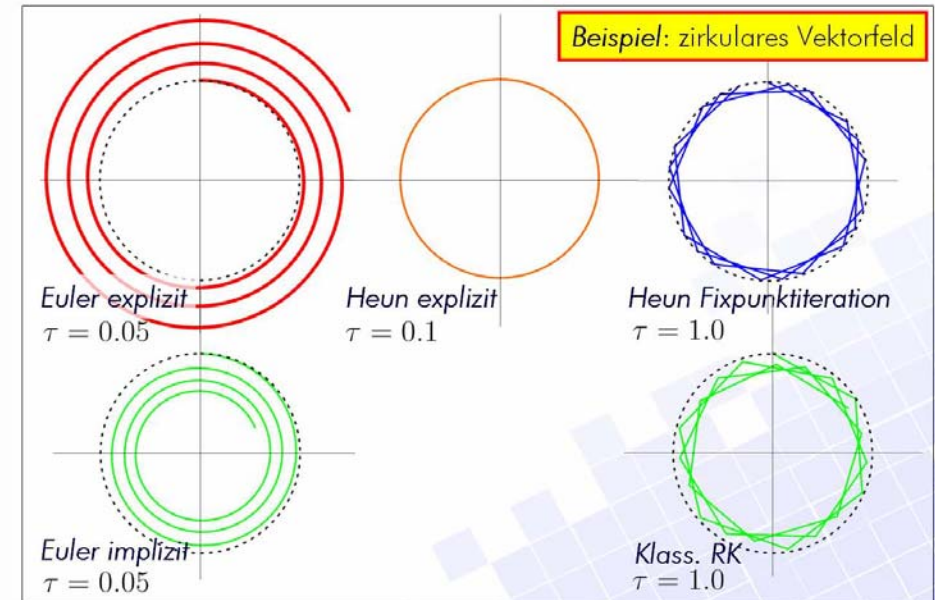
Visualisierung WS 04/05, 23.11.2004

computergraphik und multimedia systeme
universität siegen



2

Numerische Integration



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Simulationsdaten

3

Strömungsfelder (Flow Fields)

üblicherweise sind mehrere Datenfelder gegeben,

- Geschwindigkeit (Vektorfeld)
- Druck (Skalarfeld)
- Dichte (Skalarfeld)
- Temperatur (Skalarfeld)

bzw. können errechnet werden:

- Absolutbetrag der Geschwindigkeit
- Divergenz und Rotation (Skalarfelder)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

2D Strömungsdaten

4

● Direkte Verfahren

- Farbcodierung (color coding)
- Vektorpfeile, Arrow Plots (*hedgehogs*)
- Glyphen/Icons

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

2D Strömungsdaten

4

- **Direkte Verfahren**
- **Integrations-basierte Verfahren**
 - Wahl der Startpunkte
 - Particle Tracing im unstrukturierten Gitter
 - Particle Tracing im curvilinearen Gitter
 - Streamlets
 - Streamlines, Streaklines, Pathlines
 - Texturbasierte Verfahren

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Direkte Verfahren

5

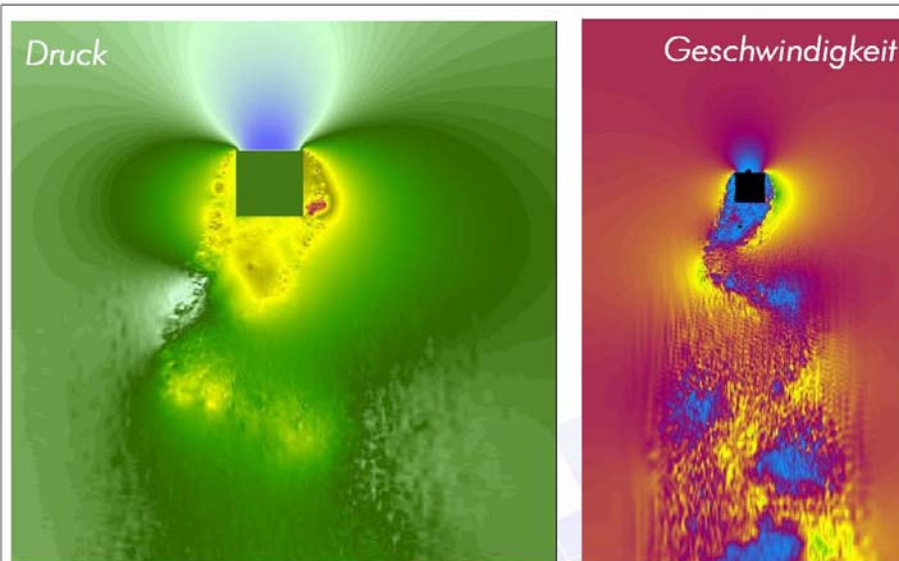
Verfahren mit möglichst direkter Transformation der Strömungsdaten in visuelle Objekte

- schnelle, unmittelbare Darstellung ohne aufwändige Vorverarbeitung
- nur lokale Eigenschaften, Zusammenhänge und Langzeit-Verhalten wird nicht explizit dargestellt

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

6

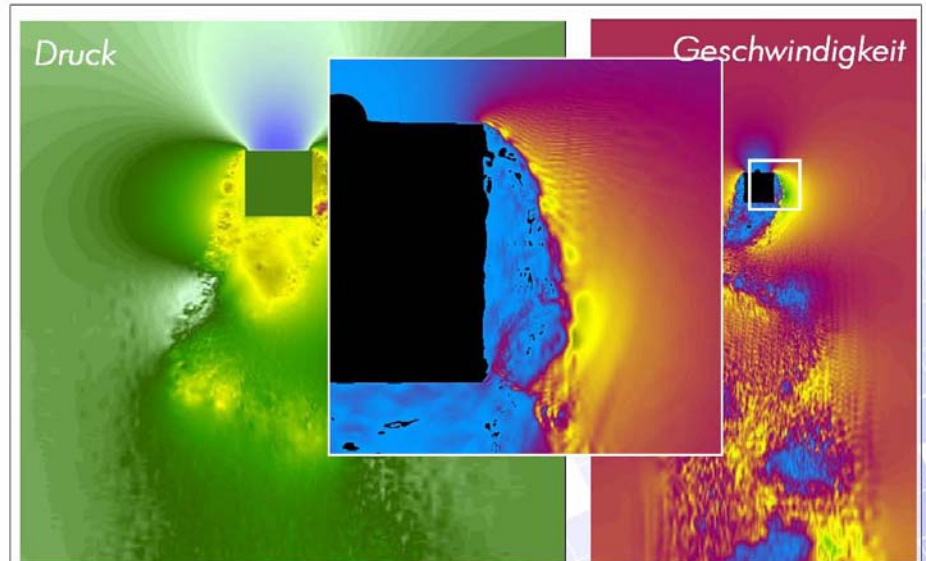


Bilder: TU Wien, Daten: A.E.P. Veldman, University of Groningen (the Netherlands)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

6



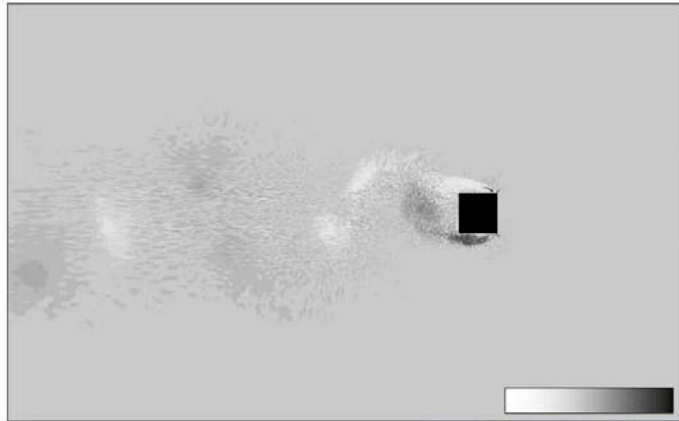
Bilder: TU Wien, Daten: A.E.P. Veldman, University of Groningen (the Netherlands)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

7

● Farbkodierung der Rotation (vorticity)



Beispiel: Strömung um einen quadratischen Block

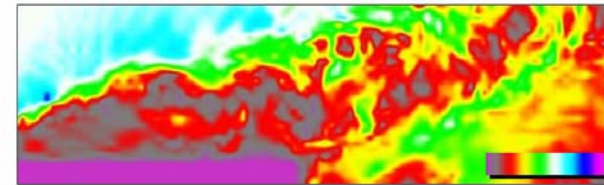
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

8

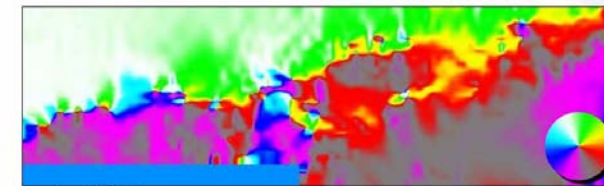
● **Absolutbetrag der Strömungsgeschwindigkeit**

Vektorlänge farbkodiert



● **Richtung der Strömung**

Richtung (Winkel in Polarkoordinaten) farbkodiert



Quelle: TU Wien



z.B. Hue-Winkel

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

9

Attribute der Strömung werden farblich dargestellt

- Geschwindigkeitsbetrag,
- Druck, Dichte, Temperatur

Vorteil:

- Sehr einfache Verfahren,
- gut geeignet auch für zeitabhängige Daten

Nachteil:

- nur skalare Größen, keine Orientierung
- keine zeitlichen Zusammenhänge

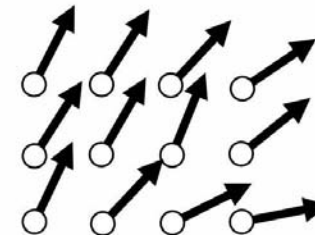
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

10

Datenpunkte werden durch kurze Linien oder Vektorpfeile dargestellt

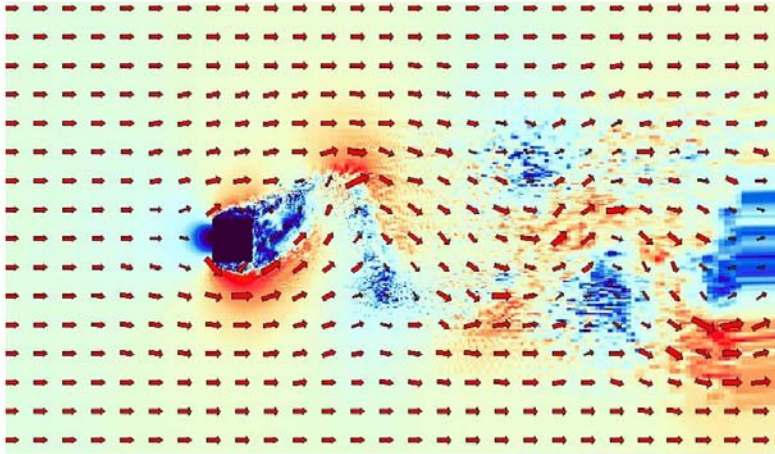
- z.B. gleichmäßig auf kartesischem Gitter
- auch *Arrow Plots* oder *Hedgehogs*.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

11

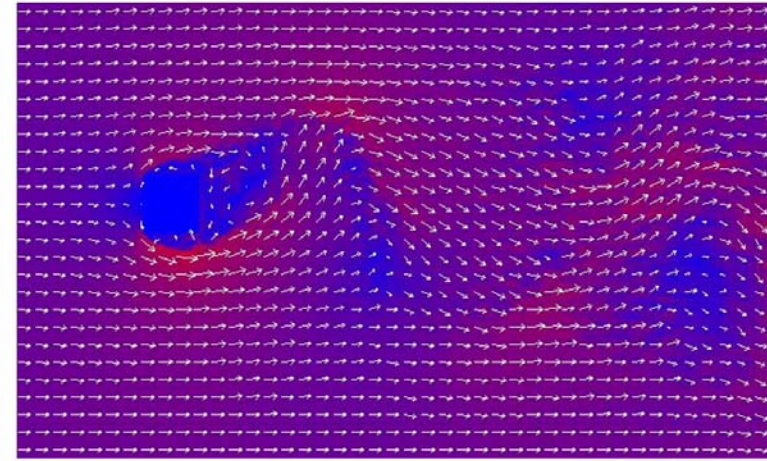


Bei relativ gleichmäßiger Strömung:
Länge/Größe des Pfeils entspricht Geschwindigkeitsbetrag

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

12



Bei stark variierender Geschwindigkeit:
Länge des Pfeils einheitlich, Absolutbetrag über Farbkodierung

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

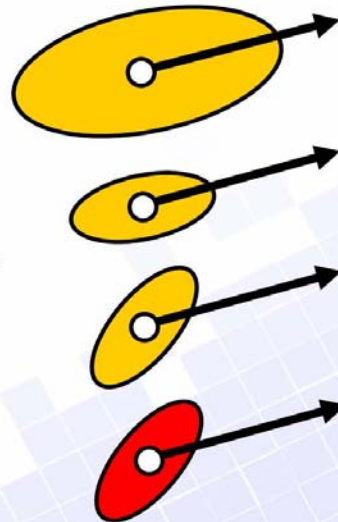
Glyphen/Icons

13

Verallgemeinerung der Vektorpfeile:
Geometrische Objekte, die lokale Eigenschaften der Strömung darstellen.

Beispiel: Vektorpfeil mit Ellipse

- Pfeil zeigt in *Strömungsrichtung*
- Fläche der Ellipse: *Geschwindigkeitsbetrag*
- Verdrehung der Ellipse: *Rotation*
- Farbcodierung: *Divergenz*



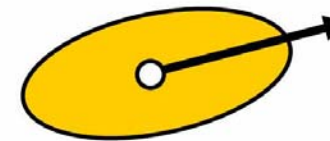
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Glyphen/Icons

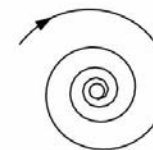
14

Schwierigkeiten:

Geometrische Eigenschaften sind nicht unabhängig!



Pfeil: Strömungsrichtung
Fläche: Geschwindigkeit
Verdrehung: Rotation
Farbe: Divergenz



Ellipsen-Beispiel: Wirbelsenke (kritischer Punkt)

- *Keine Geschwindigkeit: Ellipse verschwindet*
- *Rotation nicht mehr erkennbar.*
- *Divergenz nicht mehr erkennbar.*

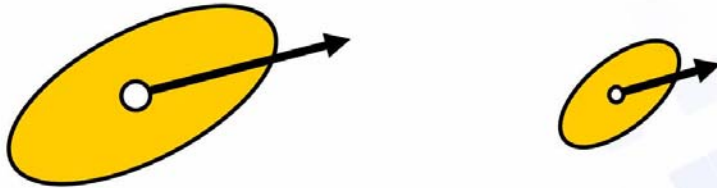
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Glyphen/Icons

15

Schwierigkeiten:

Auf **Skalierung** achten!



Länge skaliert linear.
Fläche skaliert quadratisch.
Drehung ist invariant gegenüber Skalierung.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

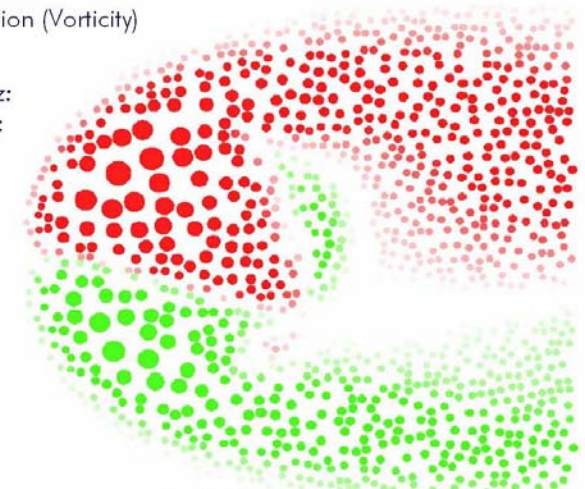
Glyphen

16

Beispiel:

Visualisierung der Rotation (Vorticity)

Radius und Transparenz:
Betrag der Rotation:
Farbe (rot und grün):
Rotationsrichtung



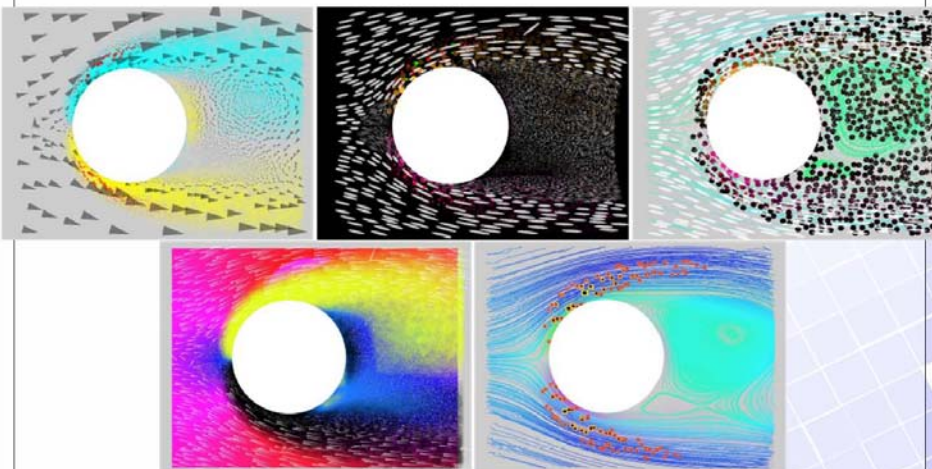
Quelle: Jason S. Sobel, Brown University, USA

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Kombinierte Verfahren

17

Farbcodierung und unterschiedliche Glyphen für verschiedene lokale Größen



Quelle: Jason S. Sobel, Brown University, USA

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

18

Vektorpfeile/Glyphen/Icons

Vorteile:

- Skalare und vektorielle Information wird dargestellt
- Direkte Darstellung, geringer Rechenaufwand

Nachteile:

- Dichte der Vektorpfeile ist begrenzt (sonst wird das Bild schnell unübersichtlich)
- Nur lokale Information (keine Zusammenhänge)
- Es ist schwierig, intuitiv verständliche Glyphen zu finden

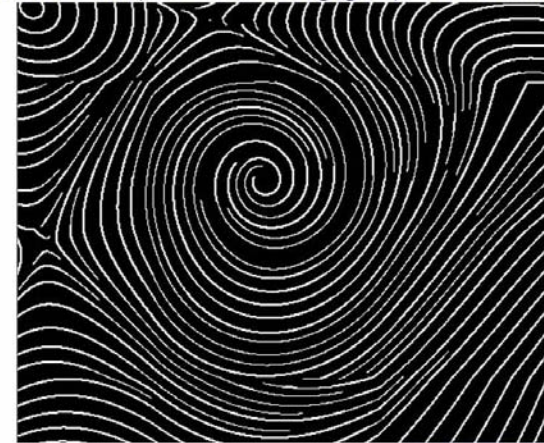
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Verfahren, die auf Integration des Vektorfelds (Partikelbahnen) basieren:

- erfordert die Bestimmung von Partikelbahnen.
- Numerisches Integrationsverfahren notwendig (z.B. Euler, Heun, Runge-Kutta)
- Vorteil gegenüber direkten Verfahren: Zeitliche Zusammenhänge werden dargestellt

Beispiel:

Feldlinien, Strömungslinien (zeitlicher Zusammenhang) im statischen (zeitunabhängigen) Vektorfeld



Quelle: TU Wien

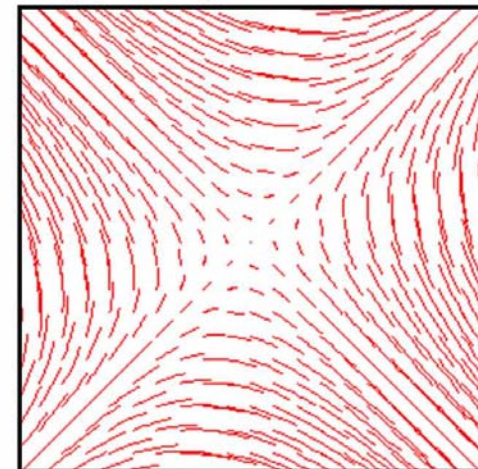
Berechnung der Partikelbahn

- Schrittweise mit numerischem Integrationsverfahren (siehe letzte Stunde)
- Einfach auf uniformem Gitter
- Für curvilineare und unstrukturierte Gitter ist Zellsuche erforderlich.

Wahl der Startpunkte (seed points) für Partikelbahnen

- Benutzerdefiniert (Flowfield-Probes)
- Uniformes Gitter von Startpunkten (streamlets) (wird leicht unübersichtlich und redundant)
- Zufällige Verteilung von Startpunkten
- Spezielle Algorithmen

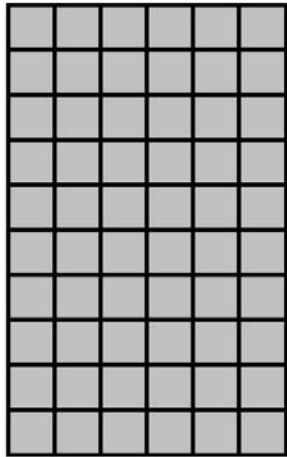
- Streamlets: Zeichne für jeden Gitterpunkt eine kurze Partikelbahn (nur wenige Zeitschritte vor- bzw. rückwärts)



Wahl der Startpunkte

23

- **Seed Point Placement Algorithms:** Ziel ist eine möglichst gleichmäßige Verteilung der Partikelbahnen



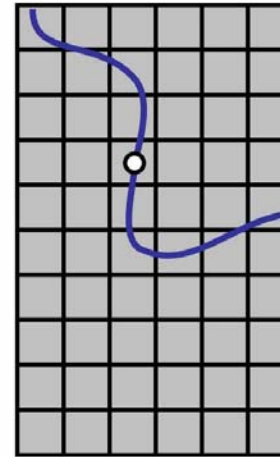
- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

23

- **Seed Point Placement Algorithms:** Ziel ist eine möglichst gleichmäßige Verteilung der Partikelbahnen



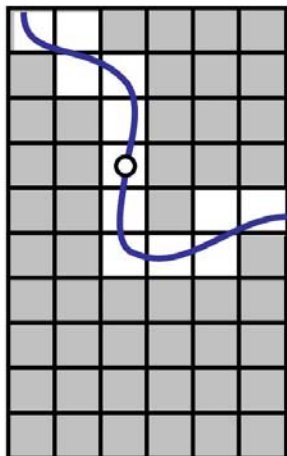
- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).
- Wähle den ersten Startpunkt (beliebig)
- Bestimme die Partikelbahn vor- und rückwärts (solang es geht)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

23

- **Seed Point Placement Algorithms:** Ziel ist eine möglichst gleichmäßige Verteilung der Partikelbahnen



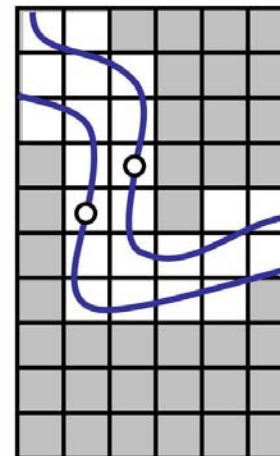
- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).
- Wähle den ersten Startpunkt (beliebig)
- Bestimme die Partikelbahn vor- und rückwärts (solang es geht)
- Markiere alle Zellen, die die Partikelbahn schneidet

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

23

- **Seed Point Placement Algorithms:** Ziel ist eine möglichst gleichmäßige Verteilung der Partikelbahnen



- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).
- Wähle den ersten Startpunkt (beliebig)
- Bestimme die Partikelbahn vor- und rückwärts (solang es geht)
- Markiere alle Zellen, die die Partikelbahn schneidet
- Wähle einen neuen Startpunkt in einer freien (unmarkierten) Zelle
- Breche die Berechnung der neuen Partikelbahnen ab, wenn eine bereits markierte Zelle geschnitten wird

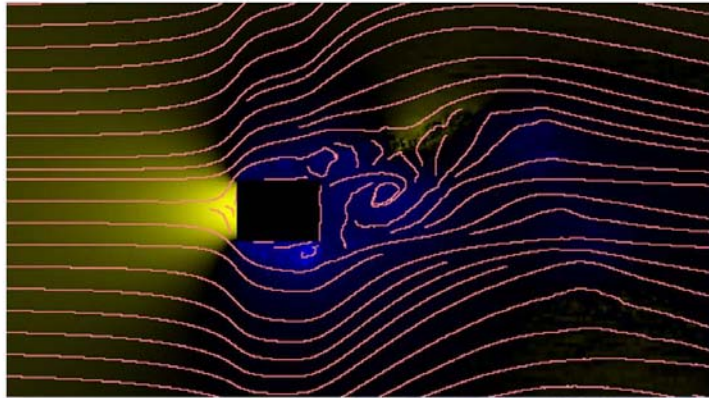
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

24

Beispiel:

Feldlinien, Strömungslinien (zeitlicher Zusammenhang) im statischen (zeitunabhängigen) Vektorfeld



Quelle: TU Wien

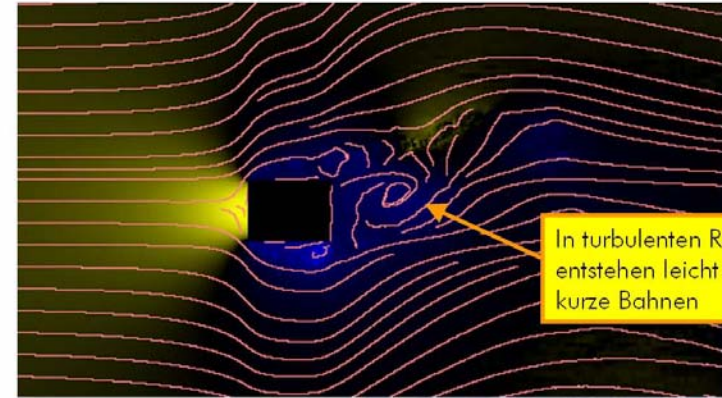
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

24

Beispiel:

Feldlinien, Strömungslinien (zeitlicher Zusammenhang) im statischen (zeitunabhängigen) Vektorfeld



In turbulenten Regionen entstehen leicht sehr kurze Bahnen

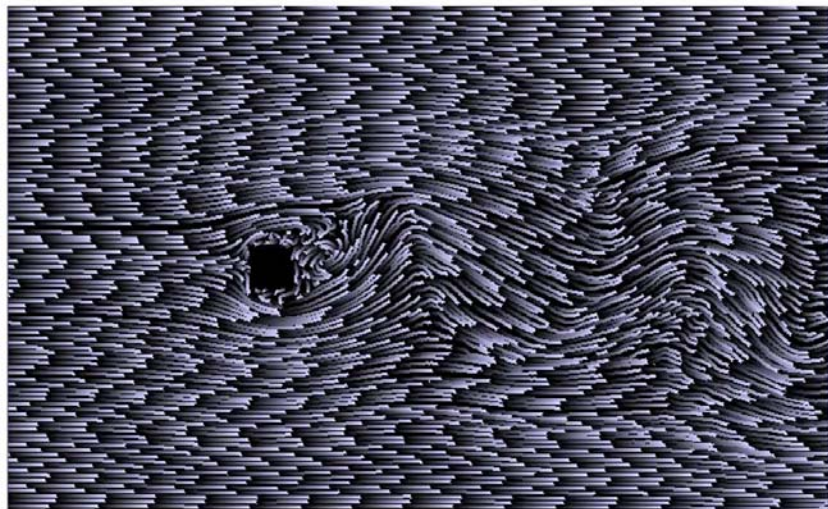
Quelle: TU Wien

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

25

Farbcodierung der Zeitschritte



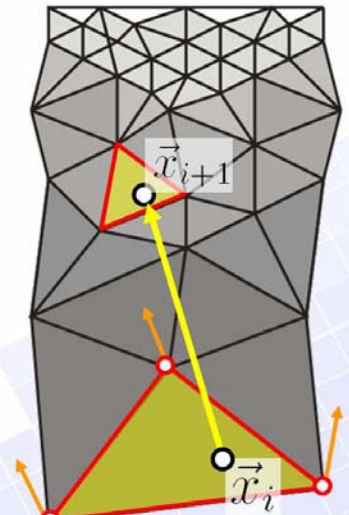
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

26

Gegeben sei ein reguläres Dreiecksnetz

1. Wähle einen Startpunkt \vec{x}_i (die Zelle sei hier bekannt)
2. Interpoliere die Geschwindigkeit (baryzent. Koord)
3. Bestimme den Folgepunkt \vec{x}_{i+1} durch numerisches Verfahren.
4. Suche die Zelle, in der der Folgepunkt liegt.



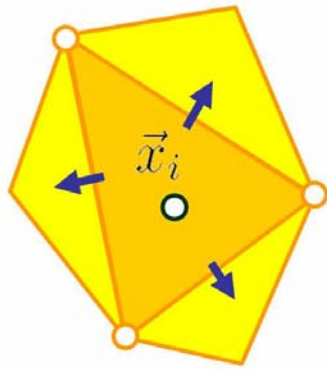
Zellsuche-Algorithmus

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

27

- **Idee: Iterativer Ansatz.** Suche eines der benachbarten Dreiecke, das näher an dem Folgepunkt liegt.



\vec{x}_{i+1}

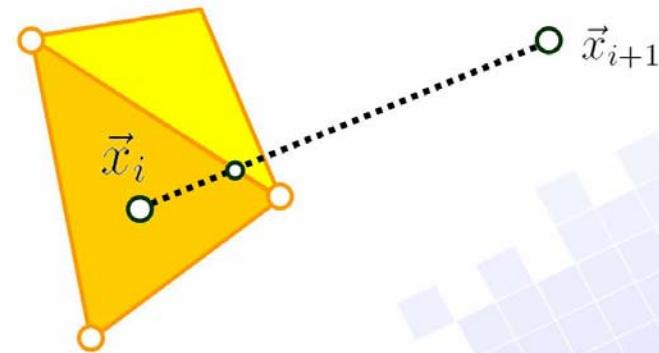
- **Voraussetzung:** Datenstruktur, die für jedes Dreieck Referenzen auf die benachbarte Dreiecke speichert (z.B. *extended winged edge*)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

28

1. Betrachte die (gerichtete) Verbindungslinie zwischen dem alten und dem neuen Punkt
2. Bestimme Schnittpunkte mit den Kanten der aktuellen Zelle

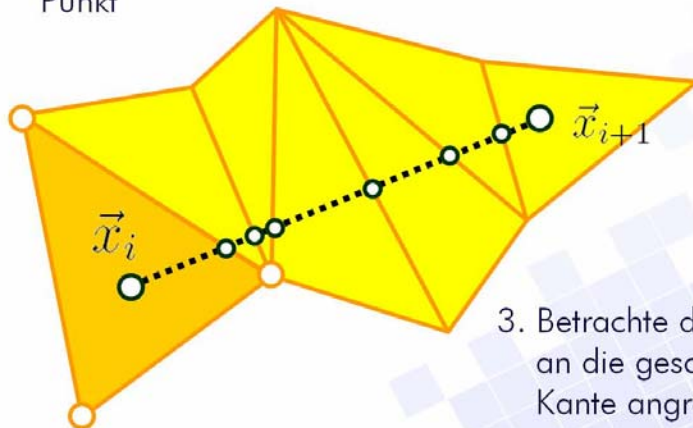


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

28

1. Betrachte die (gerichtete) Verbindungslinie zwischen dem alten und dem neuen Punkt
2. Bestimme Schnittpunkte mit den Kanten der aktuellen Zelle



3. Betrachte die Zelle die an die geschnittene Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

29

- Sonderfall:** Verbindungslinie geht durch einen Vertex (d.h. zwei Kanten werden an den Endpunkten geschnitten)



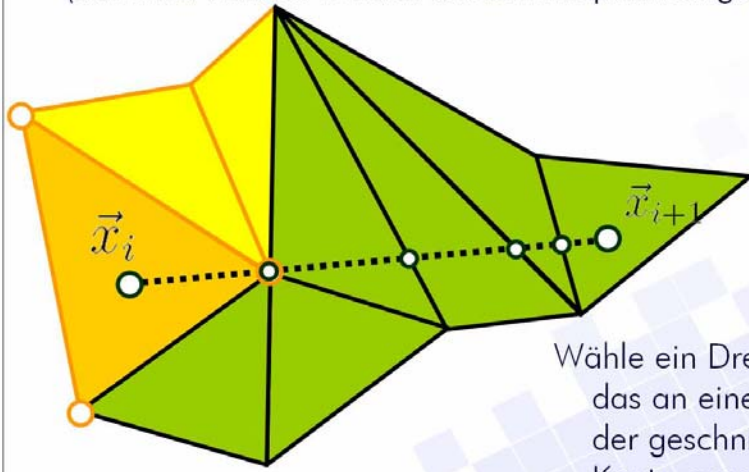
- Wähle ein Dreieck, das an eine beliebige der geschnittenen Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

29

Sonderfall: Verbindungslinie geht durch einen Vertex
(d.h. zwei Kanten werden an den Endpunkten geschnitten)



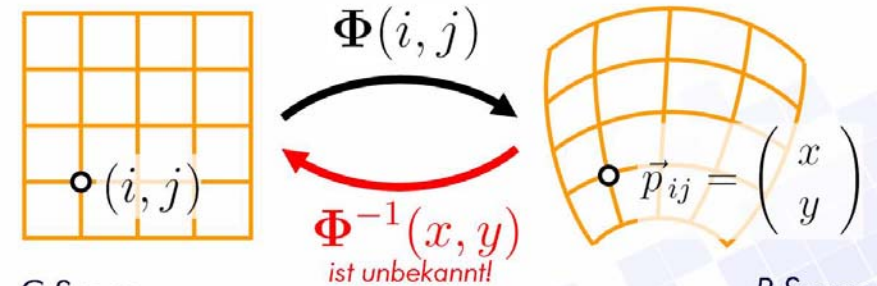
Wähle ein Dreieck,
das an eine beliebige
Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Curvilineare Gitter

30

Bei curvilinearen Gittern unterscheidet man zwischen P-Space und C-Space



C-Space
computational space
Parameterraum
uniformes Gitter
hier kann ich leicht interpolieren

$$\Phi(i, j) = \vec{p}_{ij}$$

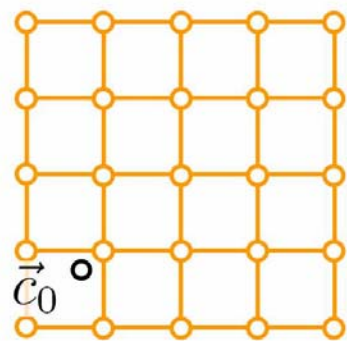
P-Space
physical space
physikalischer Raum
curvilineares Gitter
hier sind die Vektoren gültig

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Curvilineare Gitter

31

Particle Tracing auf curvilinearen Gittern



Für jeden Gitterpunkt im C-Space habe ich einen Datenwert (z.B. Geschwindigkeitsvektor) und einen dazugehörigen Punkt im P-Space gegeben.

Für einen beliebigen Punkt im C-Space kann ich demnach einen Datenwert und eine dazugehörige Position im P-Space bilinear interpolieren.

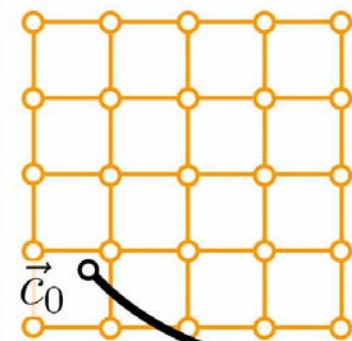
C-Space

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

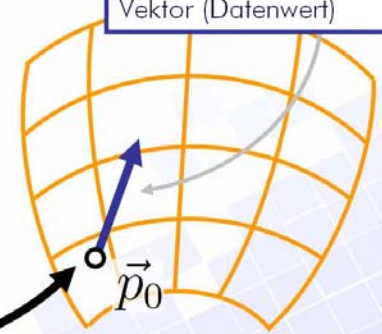
Curvilineare Gitter

31

Particle Tracing auf curvilinearen Gittern



Im C-Space interpolierter Vektor (Datenwert)



\vec{c}_0

\vec{p}_0

$$\Phi(\vec{c}_0) = \vec{p}_0$$

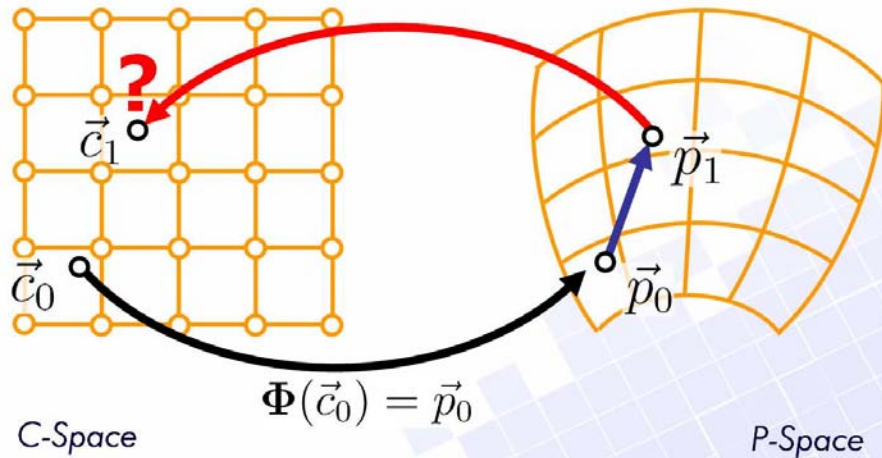
C-Space

P-Space

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

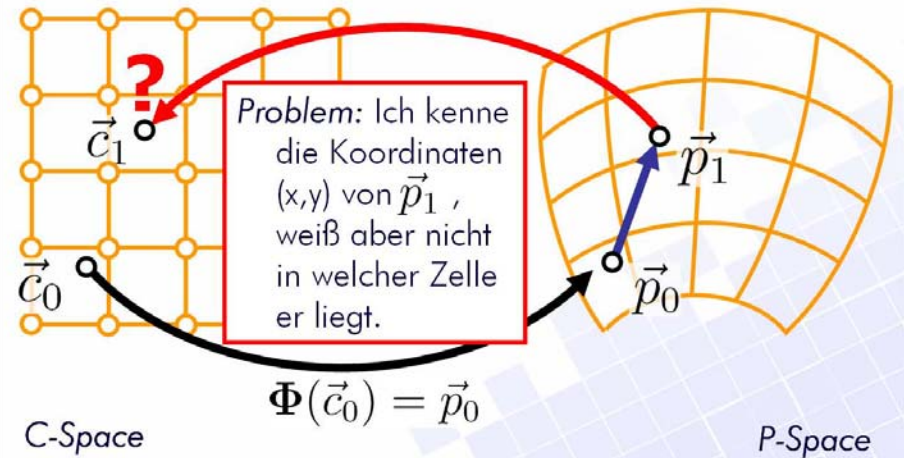
Curvilineare Gitter

Particle Tracing auf curvilinearen Gittern

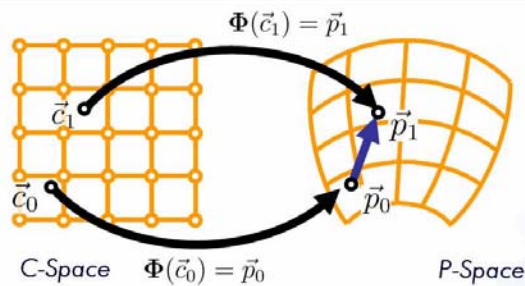


Curvilineare Gitter

Particle Tracing auf curvilinearen Gittern



Zellsuche im curvilinearen Gitter



$$\vec{p}_1 - \vec{p}_0 = \Phi(\vec{c}_1) - \Phi(\vec{c}_0)$$

Taylorentwicklung liefert:

$$\Phi(\vec{c}_1) - \Phi(\vec{c}_0) \approx \mathbf{J}_\Phi(\vec{c}_0) (\vec{c}_1 - \vec{c}_0)$$

$$\vec{c}_1 - \vec{c}_0 \approx [\mathbf{J}_\Phi(\vec{c}_0)]^{-1} (\vec{p}_1 - \vec{p}_0)$$

$$\vec{c}_1 \approx \vec{c}_0 + [\mathbf{J}_\Phi(\vec{c}_0)]^{-1} (\vec{p}_1 - \vec{p}_0)$$

Zellsuche im curvilinearen Gitter

$$\vec{c}_1 \approx \vec{c}_0 + [\mathbf{J}_\Phi(\vec{c}_0)]^{-1} (\vec{p}_1 - \vec{p}_0)$$

- Erste Näherung and den Punkt \vec{c}_1 im C-Space
- in der Regel nicht genau genug

Fixpunkt-Iteration bis zur gewünschten Genauigkeit

$$\vec{c}_1^{(0)} \approx \vec{c}_0 + [\mathbf{J}_\Phi(\vec{c}_0)]^{-1} (\vec{p}_1 - \vec{p}_0)$$

Führe Rekursion so lange aus, bis sich das Ergebnis nicht mehr signifikant verändert:

$$\vec{c}_1^{(k+1)} \approx \vec{c}_1^{(k)} + [\mathbf{J}_\Phi(\vec{c}_1^{(k)})]^{-1} (\vec{p}_1 - \Phi(\vec{c}_1^{(k)}))$$

- Fixpunkt-Iteration, „Stencil Walk“ (Pseudo-Code):

```

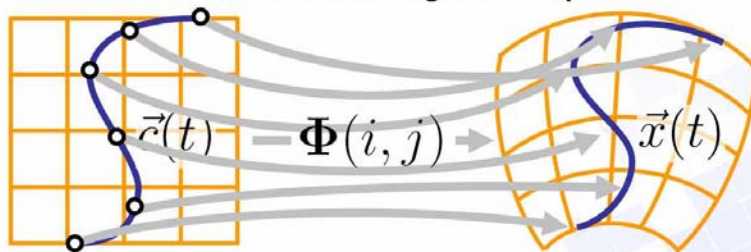
00 c = c0; p = p0; // initialize
01 do
02     dp = p1 - p;
03     dc = inverse(Jacobi(c)) * dp;
04     c = c + dc;
05     p = Φ(c);
06 until (dc < toleranz)
07 c1 = c;
    
```

- Zellsuche im curvilinearen Gitter kann ziemlich aufwändig werden
- Alternative: Particle Tracing im C-Space



Jeder Partikelbahn $\vec{x}(t)$ im P-Space entspricht eine Partikelbahn $\vec{c}(t)$ im C-Space.

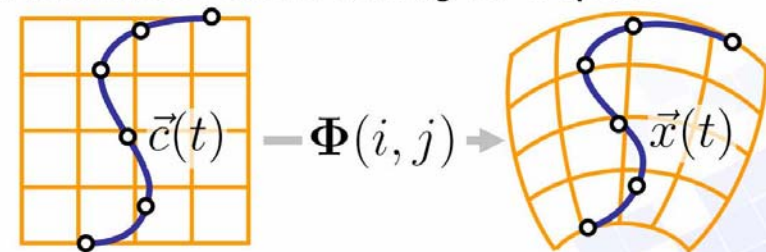
- Zellsuche im curvilinearen Gitter kann ziemlich aufwändig werden
- Alternative: Particle Tracing im C-Space



Jeder Partikelbahn $\vec{x}(t)$ im P-Space entspricht eine Partikelbahn $\vec{c}(t)$ im C-Space.

Idee: Bestimme die Partikelbahn im C-Space und transformiere die Punkte zurück in den P-Space

- Zellsuche im curvilinearen Gitter kann ziemlich aufwändig werden
- Alternative: Particle Tracing im C-Space



Jeder Partikelbahn $\vec{x}(t)$ im P-Space entspricht eine Partikelbahn $\vec{c}(t)$ im C-Space.

Idee: Bestimme die Partikelbahn im C-Space und transformiere die Punkte zurück in den P-Space

C-Space-Verfahren

36

- Wie transformiere ich die Geschwindigkeitsvektoren in den C-Space?

Geschwindigkeitsvektoren sind im P-Space gegeben:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t))$$

Transformation vom C-Space in den P-Space:

$$\vec{x}(t) = \Phi(\vec{c}(t))$$

Differenzieren:

$$\frac{\partial \vec{x}(t)}{\partial t} = \mathbf{J}_{\Phi}(\vec{c}(t)) \frac{\partial \vec{c}(t)}{\partial t}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

38

- Wie transformiere ich die Geschwindigkeitsvektoren in den C-Space?

Geschwindigkeitsvektoren sind im P-Space gegeben:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t))$$

Transformation vom C-Space in den P-Space:

$$\vec{x}(t) = \Phi(\vec{c}(t))$$

Differenzieren:

$$[\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \frac{\partial \vec{x}(t)}{\partial t} = \frac{\partial \vec{c}(t)}{\partial t}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

38

- Wie transformiere ich die Geschwindigkeitsvektoren in den C-Space?

Geschwindigkeitsvektoren sind im P-Space gegeben:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t))$$

Transformation vom C-Space in den P-Space:

$$\vec{x}(t) = \Phi(\vec{c}(t))$$

Differenzieren:

$$[\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\vec{x}(t)) = \frac{\partial \vec{c}(t)}{\partial t}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

39

- Wie transformiere ich die Geschwindigkeitsvektoren in den C-Space?

Geschwindigkeitsvektoren sind im P-Space gegeben:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t))$$

Transformation vom C-Space in den P-Space:

$$\vec{x}(t) = \Phi(\vec{c}(t))$$

Differenzieren:

$$[\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t))) = \frac{\partial \vec{c}(t)}{\partial t}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

40

- Wie transformiere ich die Geschwindigkeitsvektoren in den C-Space?

$$[\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t))) = \frac{\partial \vec{c}(t)}{\partial t}$$

Differentialgleichung im C-Space:

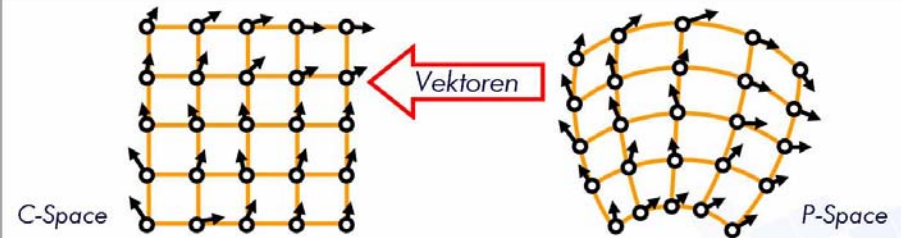
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{v}(\vec{c}(t))$$

$$\tilde{v}(\vec{c}(t)) = [\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t)))$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

40



Differentialgleichung im C-Space:

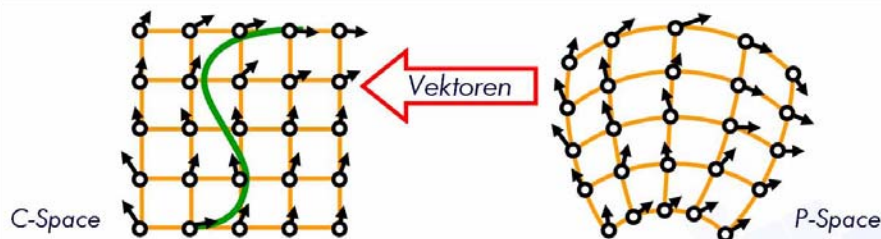
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{v}(\vec{c}(t))$$

$$\tilde{v}(\vec{c}(t)) = [\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t)))$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

40



Differentialgleichung im C-Space:

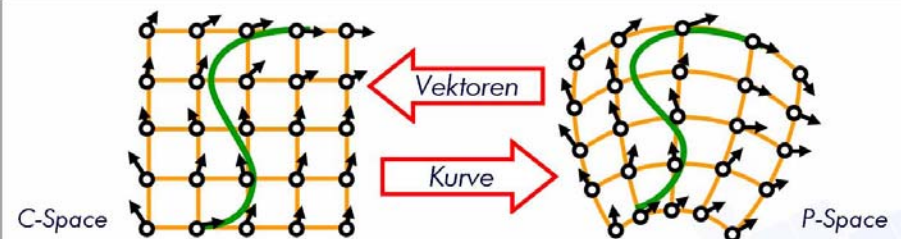
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{v}(\vec{c}(t))$$

$$\tilde{v}(\vec{c}(t)) = [\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t)))$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

40



Differentialgleichung im C-Space:

$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{v}(\vec{c}(t))$$

$$\tilde{v}(\vec{c}(t)) = [\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t)))$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

41

P-Space Verfahren

- Interpoliere im C-Space
- Transformiere den Punkt in den P-Space
- Numerischer Integrations-schritt im P-Space
- Zellsuche zur Bestimmung des Folgepunktes im C-Space

C-Space Verfahren

- Transformiere die Vektoren in den C-Space
- Bestimme Partikelbahn durch num. Integration im C-Space
- Transformiere die Ergebnis-Kurve zurück in den P-Space

- Jacobi-Matrix wird nur für Zellsuche verwendet
- Höhere Genauigkeit

- Vektoren werden transformiert (ungenau)
- Einfach und schnell

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zeitabhängige Daten

42

- *Bisher:* Particle Tracing in zeitlich konstanten (statischen) Vektorfeldern:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t))$$

- *Jetzt:* Particle Tracing in zeitabhängigen (dynamischen) Vektorfeldern:

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t), t)$$

Was ändert sich an den numerischen Verfahren?

Zeit muss als zusätzlicher Parameter mitgeführt werden.

z.B. Eulerschritt $t_{i+1} = t_i + \tau$

$$\vec{x}(t_{i+1}) = \vec{x}(t_i) + \tau \vec{v}(\vec{x}(t_i), t_i + \tau)$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zeitabhängige Partikelbahnen

43

- *Pathlines (Pfadlinie):* Die Bahn eines Partikels im zeitabhängigen Vektorfeld

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t), t) \quad \vec{x}_0 = \vec{x}(t_0)$$

Experimentelle Bestimmung: „Injiziere zum Zeitpunkt t_0 einen einzigen Tropfen Farbstoff in die Strömung und mache ein Foto mit langer Belichtungszeit.“

- *Streamlines (Feldlinie):* zeitlicher Schnappschuß des Vektorfeldes

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t), T) \quad T = const.$$

Experimentelle Bestimmung: nicht möglich, da die Zeit nicht abgehalten werden kann.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zeitabhängige Partikelbahnen

44

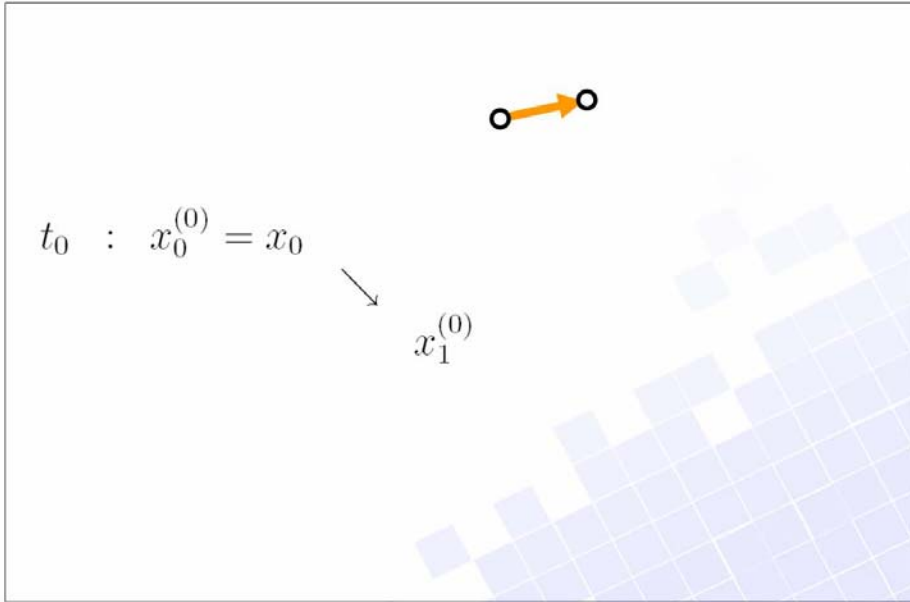
- *Streak Line:* Die Positionen von Partikeln, die zu verschiedenen Zeitpunkten am gleichen Ort gestartet sind.

$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t), t) \quad \vec{x}_0 \text{ konstant, } t_0 \text{ variabel}$$

Experimentelle Bestimmung: „Injiziere an einem bestimmten Punkt kontinuierlich Farbstoff in die Strömung“

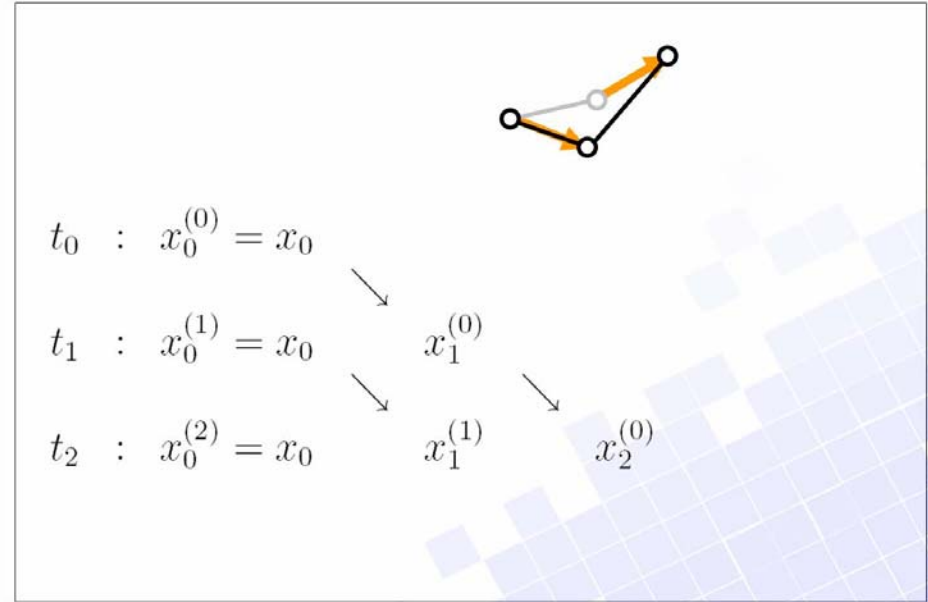
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines (Streichlinien)



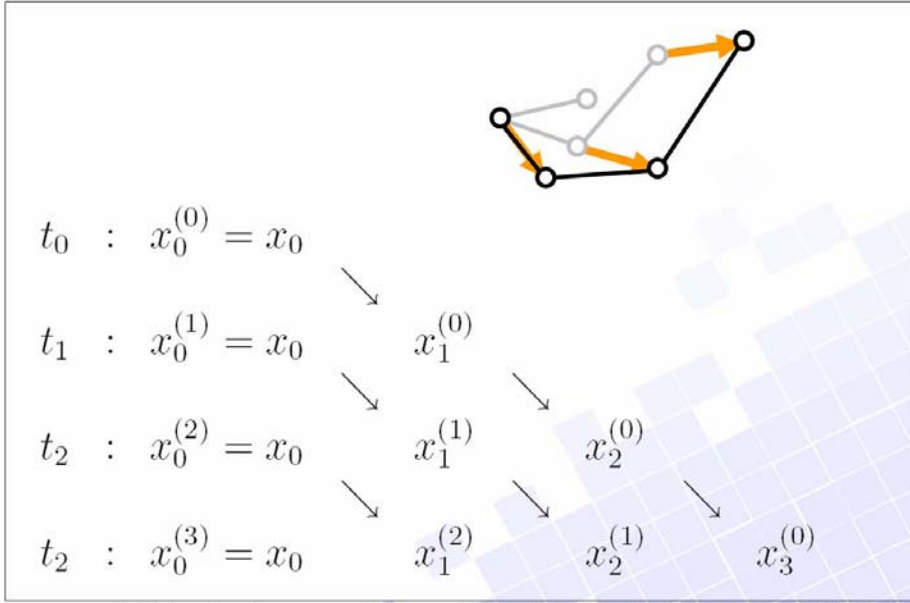
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines (Streichlinien)



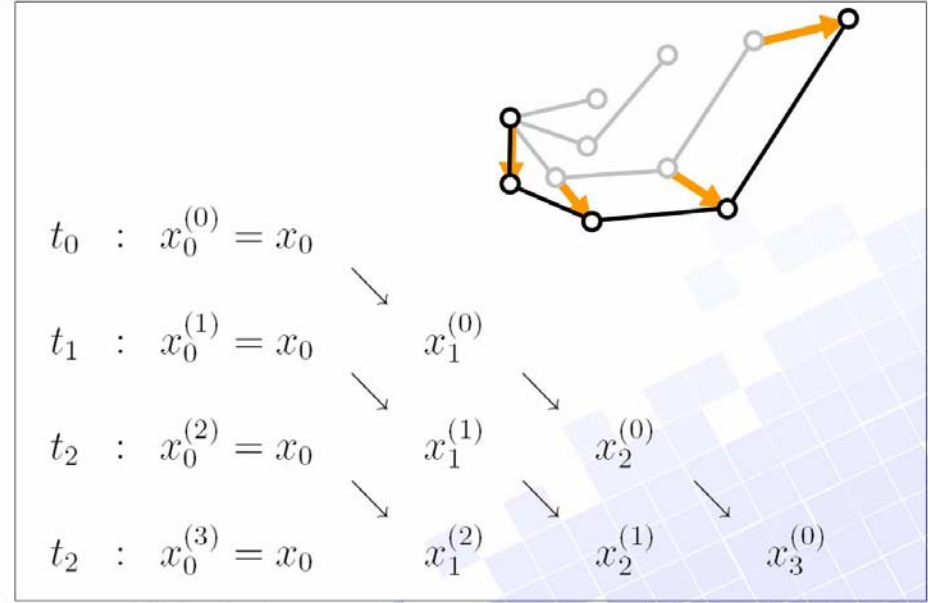
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines (Streichlinien)



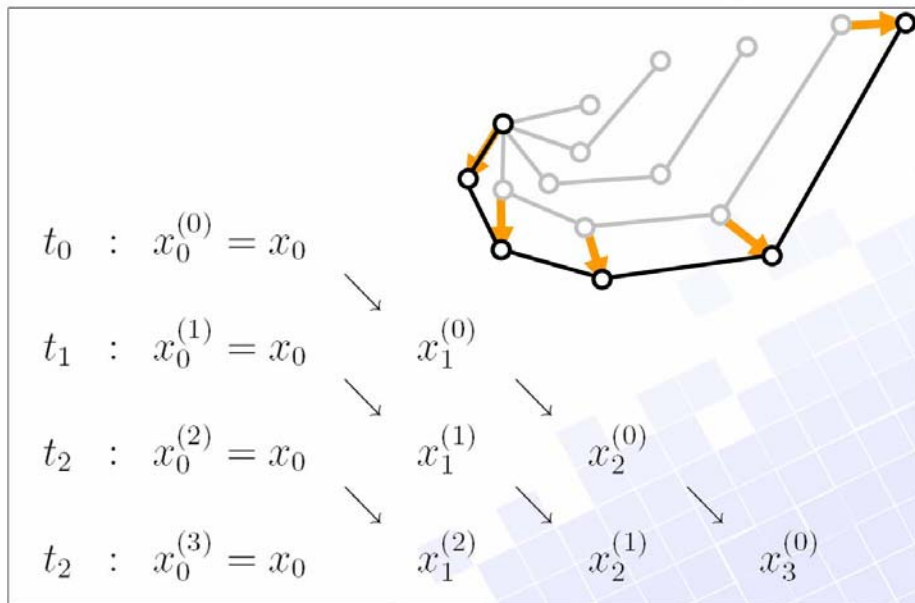
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines (Streichlinien)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines (Streichlinien)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

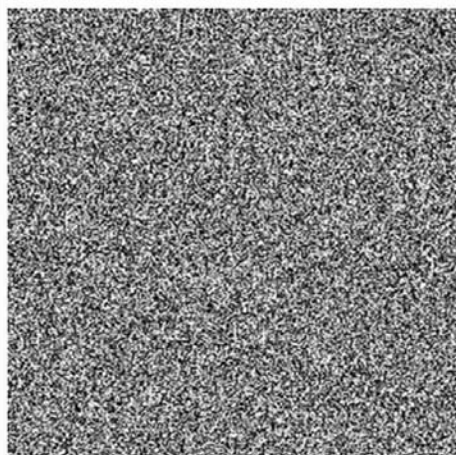
Partikelbahnen

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Exkurs: Rauschen

- Ein Bild (2D Skalarfeld) mit zufälligen (Grau-) Werten bezeichnet man als „Rauschen“

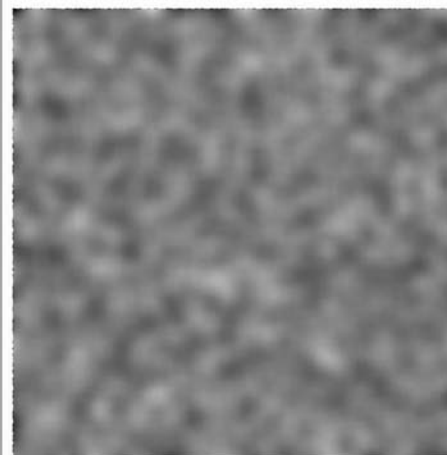


Signalverarbeitung:
In Rauschen sind *alle* Frequenzen eines bestimmten Frequenzbandes enthalten

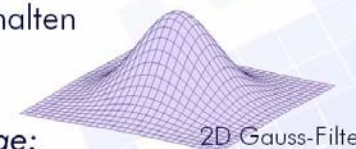
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Exkurs: Rauschen

- Ein Bild (2D Skalarfeld) mit zufälligen (Grau-) Werten bezeichnet man als „Rauschen“



Signalverarbeitung:
In Rauschen sind *alle* Frequenzen eines bestimmten Frequenzbandes enthalten



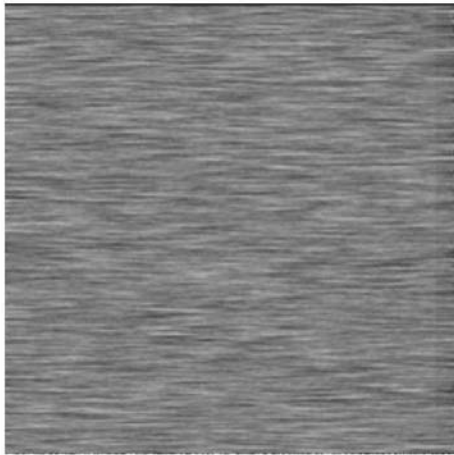
Frage:
Was passiert wenn ich Rauschen mit einem Glättungsfilter bearbeite?

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

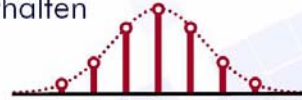
Exkurs: Rauschen

48

- Ein Bild (2D Skalarfeld) mit zufälligen (Grau-) Werten bezeichnet man als „Rauschen“



Signalverarbeitung:
In Rauschen sind *alle* Frequenzen eines bestimmten Frequenzbandes enthalten



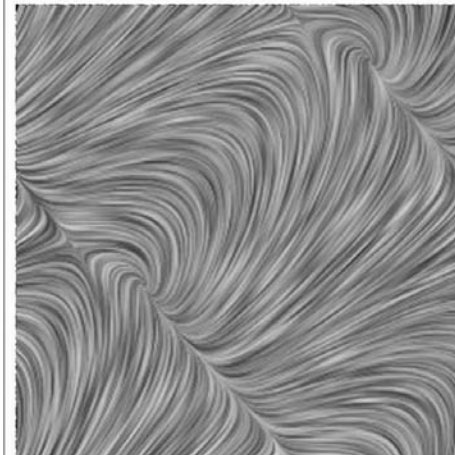
Frage: 1D Gauss-Filter
Was passiert wenn ich nur in eine Richtung (z.B. x-Richtung) glätte?

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

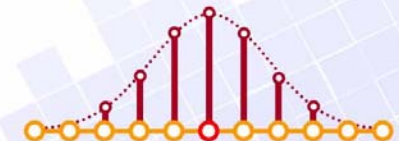
Line Integral Convolution

49

Idee: Filtere ein 2D Rauschen entlang der Feldlinien eines 2D Vektorfeldes



- Bestimme für jeden Pixel des Ergebnisbildes eine kurze Partikelbahn (vor- und rückwärts)
- Betrachte die Werte des Rauschbildes an den Positionen
- Filtere die Werte mit einem Glättungsfilter

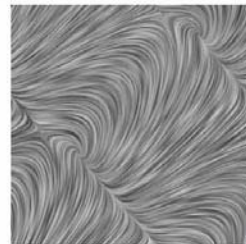


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Line Integral Convolution

50

Line-Integral: Linienintegral = Partikelbahn
Convolution: Faltung = Linearer Filter
„Faltung entlang der Linienintegrale“



In der Praxis: Box-Filter (Mittelung)

Vorteil:

- Hohe Dichte an Partikelbahnen,
- sehr intuitiv verständlich

Nachteil:

- Relativ hoher Rechenaufwand
- Kein Absolutbetrag der Geschwindigkeit!

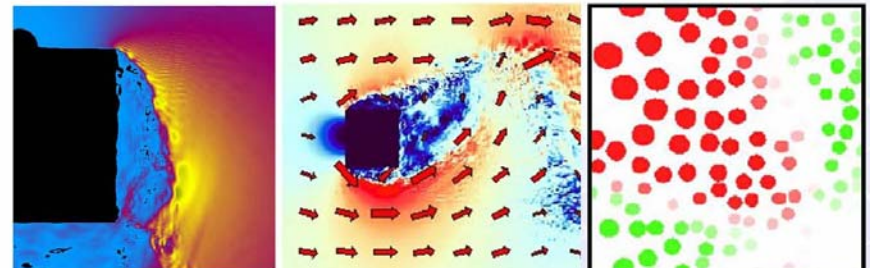
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

51

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

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 Daten:
Streamlines, Streaklines & Pathlines

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Visualisierungsverfahren für 2D Strömungsdaten

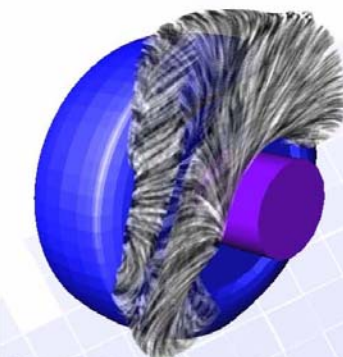
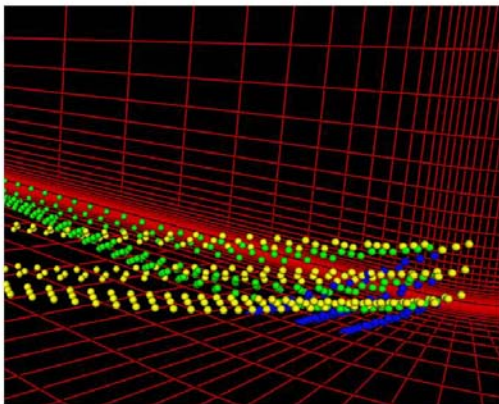
Integrationsbasierte Verfahren

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

3D Vektorfelder

Stream Balls, -Ribbons, -Tubes
Stream Surfaces, Time Surfaces, 3D LIC



Quelle: C. Teitzel, Universität Erlangen-Nürnberg

Daten: BMW, München
Quelle: Rezk-Salama, IEEE Visualization 99

christof rezk-salama, computergraphik und multimediasysteme, universität siegen