

Visualisierung: 2D Strömungsdaten

Andreas Kolb und Martin Lambers

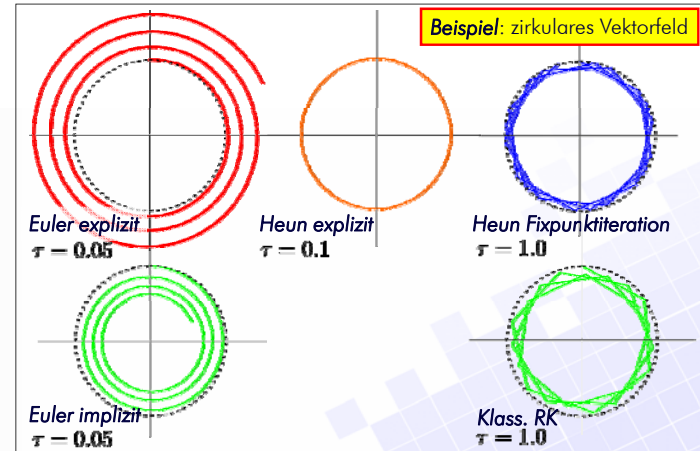
computergraphik und multimedia systeme
universität siegen



Numerische Integration

2

Beispiel: zirkulares Vektorfeld



Andreas Kolb, Martin Lambers

Visualisierung

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)

Andreas Kolb, Martin Lambers

Visualisierung

2D Strömungsdaten

4

Direkte Verfahren

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

Andreas Kolb, Martin Lambers

Visualisierung

2D Strömungsdaten

5

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

Andreas Kolb, Martin Lambers

Visualisierung

Direkte Verfahren

6

Verfahren mit möglichst direkter Transformation der Strömungsdaten in visuelle Objekte

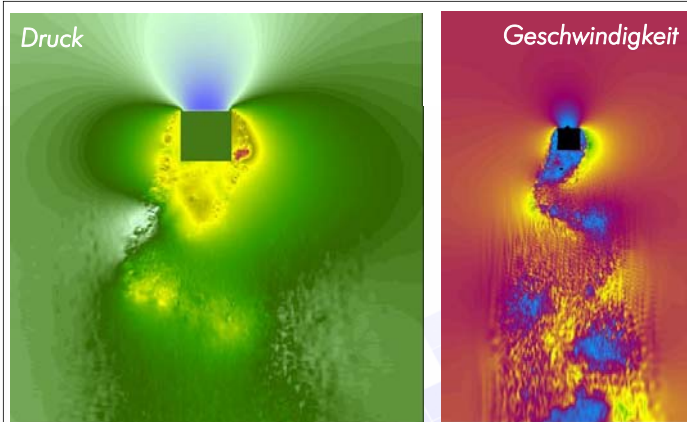
- schnelle, unmittelbare Darstellung ohne aufwändige Vorverarbeitung
- nur lokale Eigenschaften darstellbar
- schwer explizit darstellbar sind
 - Zusammenhänge zwischen den skalaren Größen und
 - Zeitverhalten

Andreas Kolb, Martin Lambers

Visualisierung

Farbkodierung

7



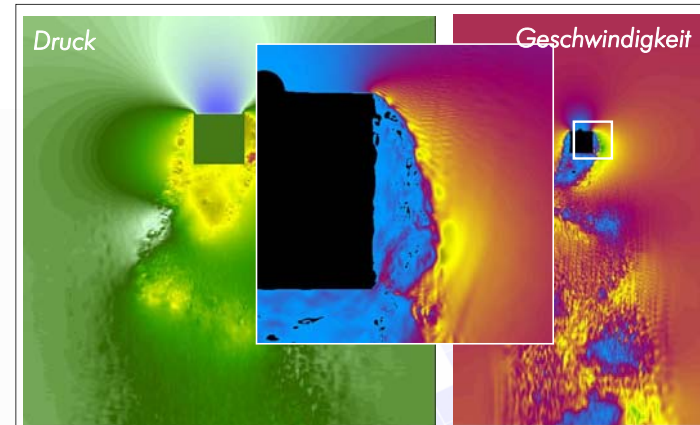
Bilder: TU Wien, Daten: A.E.P. Veldman, University of Groningen (the Netherlands)

Andreas Kolb, Martin Lambers

Visualisierung

Farbkodierung

8



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

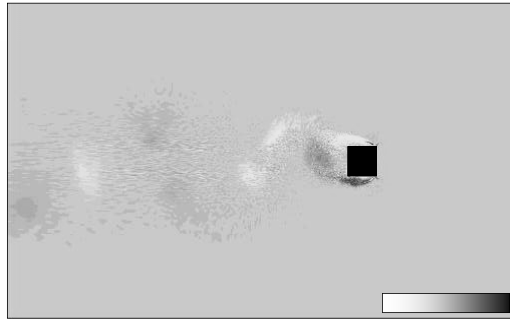
Andreas Kolb, Martin Lambers

Visualisierung

Farbkodierung

9

Farbkodierung der Rotation (vorticity)



Beispiel: Strömung um einen quadratischen Block

Andreas Kolb, Martin Lambers

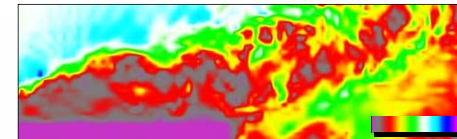
Visualisierung

Farbkodierung

10

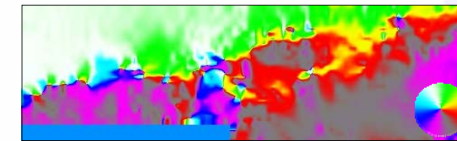
Absolutbetrag der Strömungsgeschwindigkeit

Vektorlänge farbkodiert



Richtung der Strömung

Richtung (Winkel in Polarkoordinaten) farbkodiert



Quelle: TU Wien



z.B. Hue-Winkel

Andreas Kolb, Martin Lambers

Visualisierung

Farbkodierung

11

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

Andreas Kolb, Martin Lambers

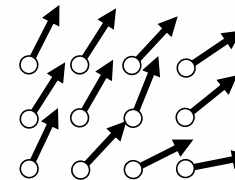
Visualisierung

Vektorpfeile

12

Datenpunkte werden durch kurze Linien oder Vektorpfeile dargestellt

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

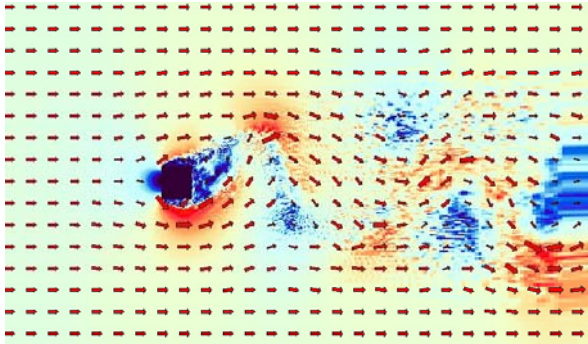


Andreas Kolb, Martin Lambers

Visualisierung

Vektorpfeile

13



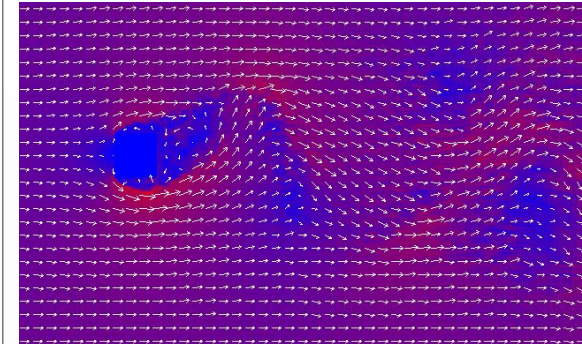
Bei relativ gleichmäßiger Strömung:
Länge/Größe des Pfeils entspricht Geschwindigkeitsbetrag

Andreas Kolb, Martin Lambers

Visualisierung

Vektorpfeile

14



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

Andreas Kolb, Martin Lambers

Visualisierung

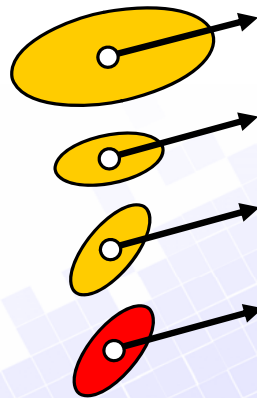
Glyphen/Icons

15

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



Andreas Kolb, Martin Lambers

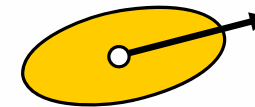
Visualisierung

Glyphen/Icons

16

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.

Andreas Kolb, Martin Lambers

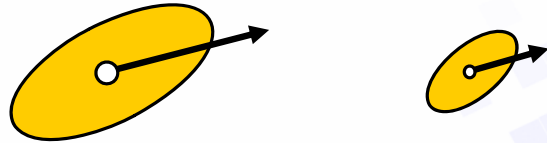
Visualisierung

Glyphen/Icons

17

Schwierigkeiten:

Auf *Skalierung* achten!



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

Andreas Kolb, Martin Lambers

Visualisierung

Glyphen

18

Beispiel:

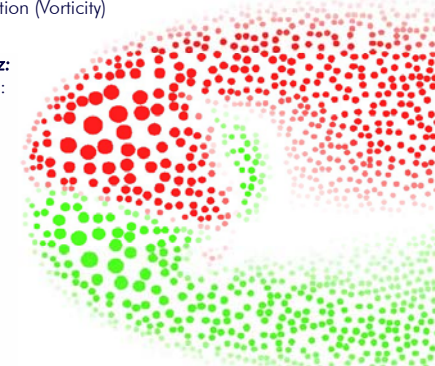
Visualisierung der Rotation (Vorticity)

Radius und Transparenz:

Betrag der Rotation:

Farbe (rot und grün):

Rotationsrichtung



Quelle: Jason S. Sobel, Brown University, USA

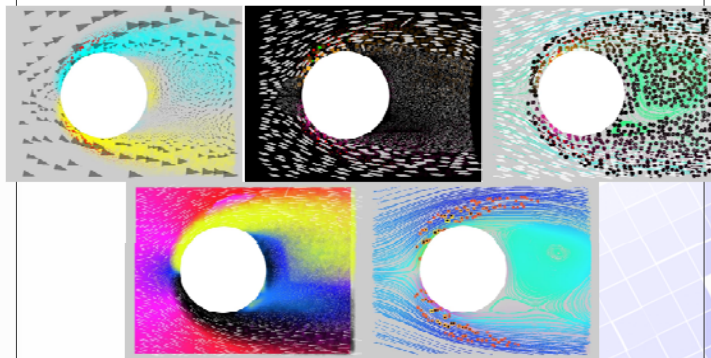
Andreas Kolb, Martin Lambers

Visualisierung

Kombinierte Verfahren

19

Farbcodierung und unterschiedliche Glyphen für verschiedene lokale Größen



Quelle: Jason S. Sobel, Brown University, USA

Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung

20

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

Andreas Kolb, Martin Lambers

Visualisierung

Verfahren mit Integration

21

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

Andreas Kolb, Martin Lambers

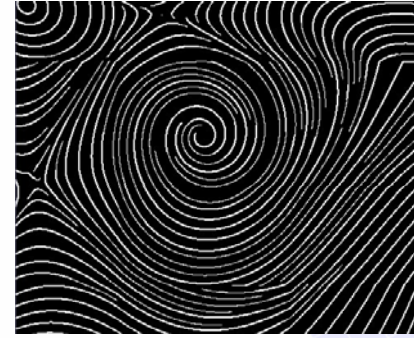
Visualisierung

Partikelbahnen

22

Beispiel:

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



Quelle: TU Wien

Andreas Kolb, Martin Lambers

Visualisierung

Übersicht Partikelbahnen

23

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

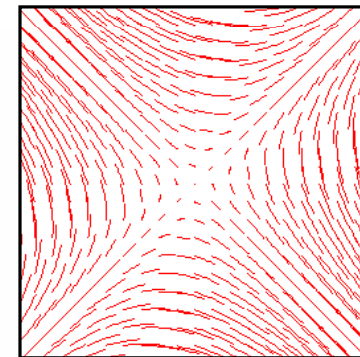
Andreas Kolb, Martin Lambers

Visualisierung

Wahl der Startpunkte

24

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



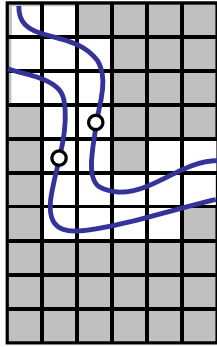
Andreas Kolb, Martin Lambers

Visualisierung

Wahl der Startpunkte

25

- **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 (solange 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

Andreas Kolb, Martin Lambers

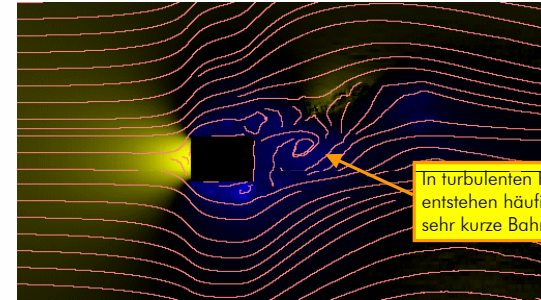
Visualisierung

Partikelbahnen

26

Beispiel:

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



In turbulenten Regionen entstehen häufig viele, sehr kurze Bahnen

Quelle: TU Wien

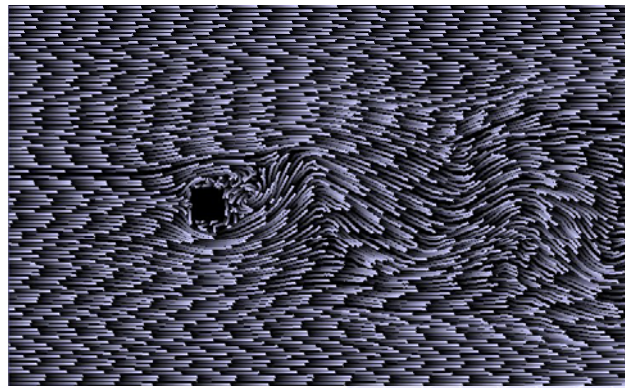
Andreas Kolb, Martin Lambers

Visualisierung

Partikelbahnen

27

Farbcodierung der Zeitschritte



Andreas Kolb, Martin Lambers

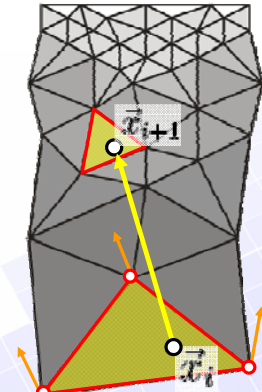
Visualisierung

Partikelbahnen

28

- 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

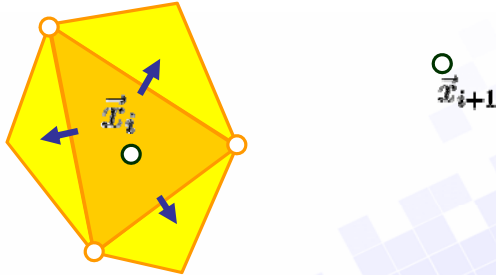
Andreas Kolb, Martin Lambers

Visualisierung

Zellsuche im Dreiecksnetz

29

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



Voraussetzung: Datenstruktur, die für jedes Dreieck Referenzen auf die benachbarten Dreiecke speichert (z.B. erweiterte *half edge* Struktur)

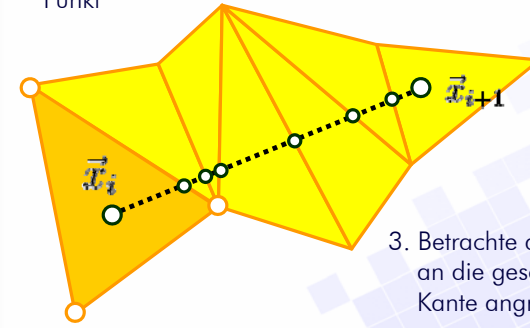
Andreas Kolb, Martin Lambers

Visualisierung

Zellsuche im Dreiecksnetz

30

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.



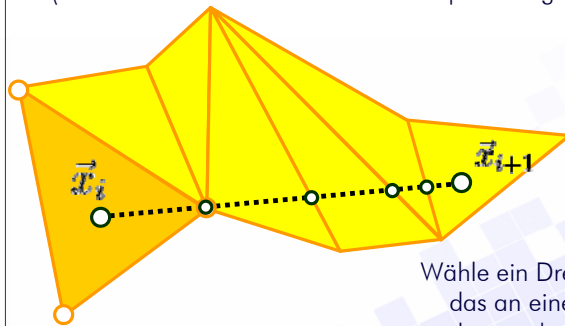
Andreas Kolb, Martin Lambers

Visualisierung

Zellsuche im Dreiecksnetz

31

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.

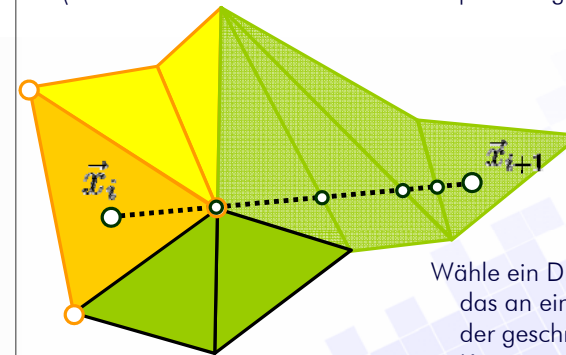
Andreas Kolb, Martin Lambers

Visualisierung

Zellsuche im Dreiecksnetz

32

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.

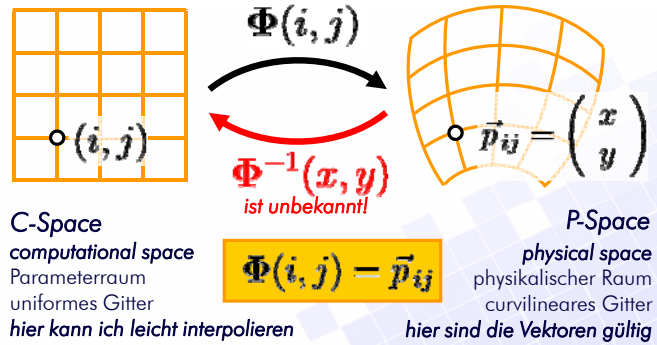
Andreas Kolb, Martin Lambers

Visualisierung

Curvilineare Gitter

33

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



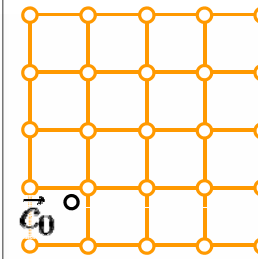
Andreas Kolb, Martin Lambers

Visualisierung

Curvilineare Gitter

34

- Particle Tracing auf curvilinearen Gittern



C-Space

Andreas Kolb, Martin Lambers

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

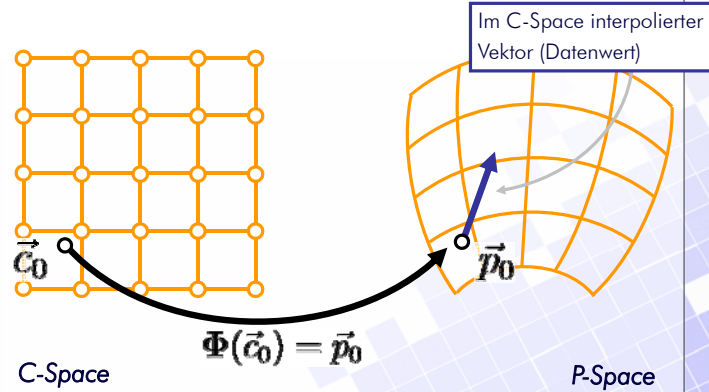
Für einen **beliebigen** Punkt im C-Space kann demnach ein **Datenwert und** eine dazugehörige **Position im P-Space** bilinear interpolieren.

Visualisierung

Curvilineare Gitter

35

- Particle Tracing auf curvilinearen Gittern



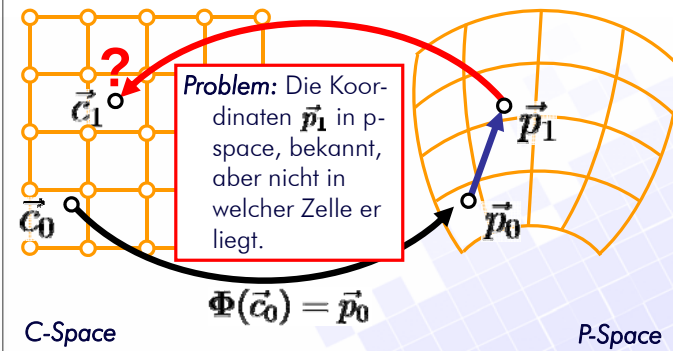
Andreas Kolb, Martin Lambers

Visualisierung

Curvilineare Gitter

36

- Particle Tracing auf curvilinearen Gittern

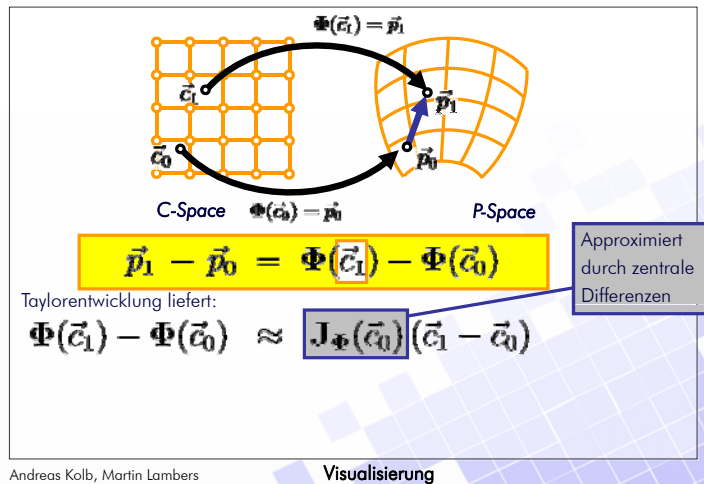


Andreas Kolb, Martin Lambers

Visualisierung

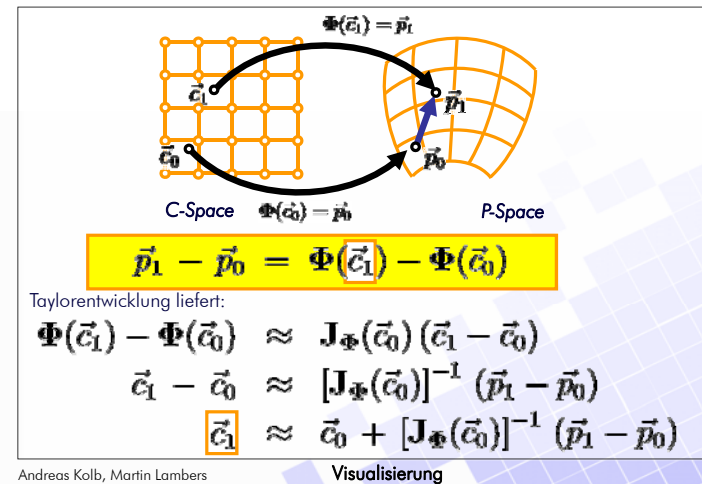
Zellsuche im curvilinearen Gitter

37



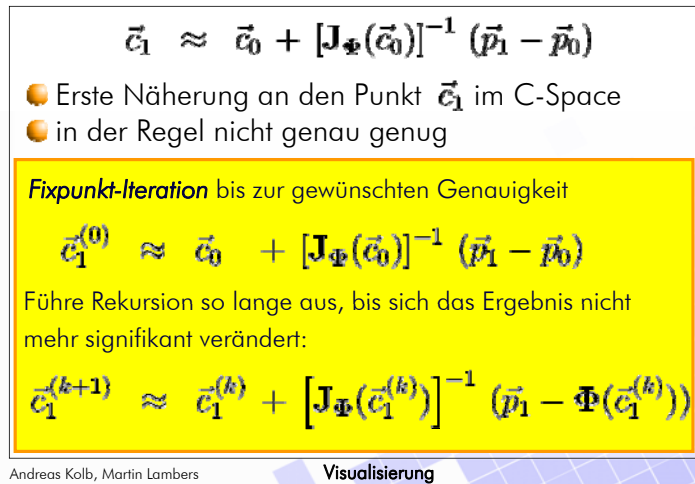
Zellsuche im curvilinearen Gitter

38



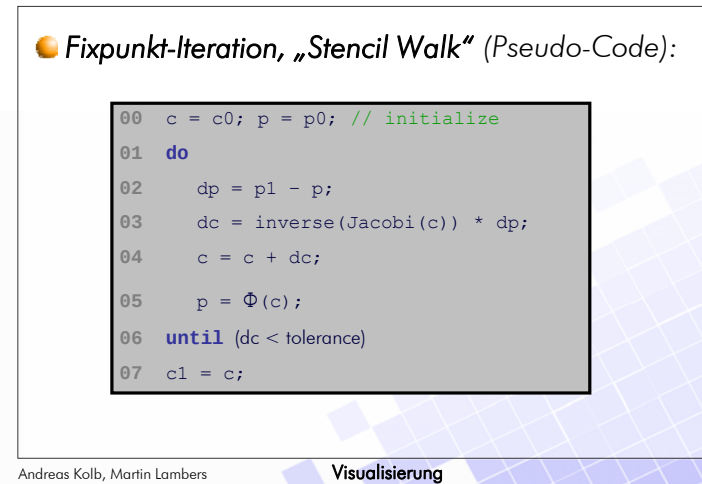
Zellsuche im curvilinearen Gitter

39



Zellsuche im curvilinearen Gitter

40

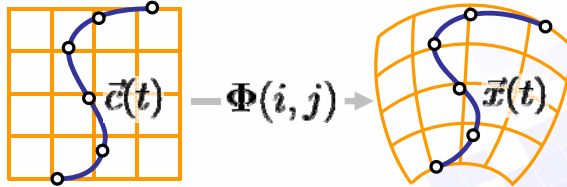


C-Space Verfahren:

41

- 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

Andreas Kolb, Martin Lambers

Visualisierung

C-Space-Verfahren

42

- 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:

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

nachdifferenzieren (Kettenregel)

Andreas Kolb, Martin Lambers

Visualisierung

C-Space-Verfahren

43

- 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:

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

Andreas Kolb, Martin Lambers

Visualisierung

C-Space-Verfahren

44

- 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:

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

Andreas Kolb, Martin Lambers

Visualisierung

C-Space-Verfahren

45

- 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:

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

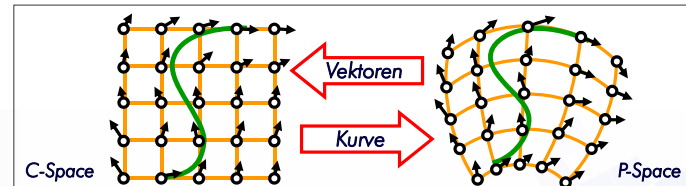
$$\tilde{\vec{v}}(\vec{c}(t)) = [\mathbf{J}_{\Phi}(\vec{c}(t))]^{-1} \vec{v}(\Phi(\vec{c}(t)))$$

Andreas Kolb, Martin Lambers

Visualisierung

C-Space-Verfahren

46



Differentialgleichung im C-Space:

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

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

Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung

47

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

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

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

- Vektoren werden transformiert (ungenau)
- Einfach und schnell

Andreas Kolb, Martin Lambers

Visualisierung

Zeitabhängige Daten

48

- 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 $t_{i+1} = t_i + \tau$

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

Andreas Kolb, Martin Lambers

Visualisierung

Zeitabhängige Partikelbahnen

49

- **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 Styroporkügelchen in die Strömung und mache ein Foto mit langer Belichtungszeit.“

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

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

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

Andreas Kolb, Martin Lambers

Visualisierung

Zeitabhängige Partikelbahnen

50

- **Streak Line (Streichlinien):** 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“

Andreas Kolb, Martin Lambers

Visualisierung

Streaklines (Streichlinien)

51

$$t_0 : x_0^{(0)} = x_0$$

$$x_1^{(0)}$$



Andreas Kolb, Martin Lambers

Visualisierung

Streaklines (Streichlinien)

52

$$t_0 : x_0^{(0)} = x_0$$

$$t_1 : x_0^{(1)} = x_0$$

$$x_1^{(0)}$$

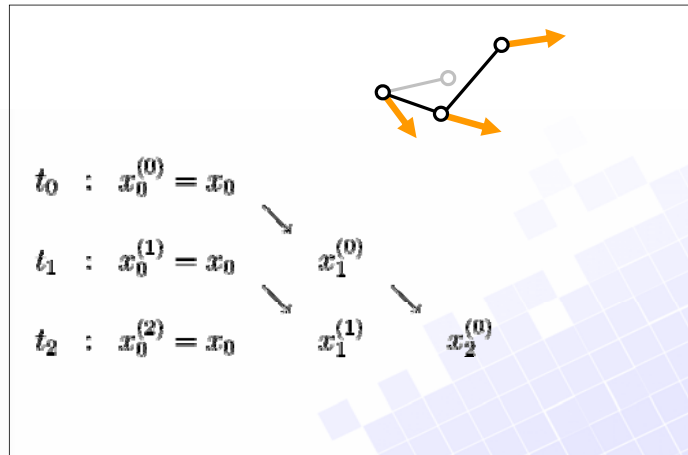


Andreas Kolb, Martin Lambers

Visualisierung

Streaklines (Streichlinien)

53

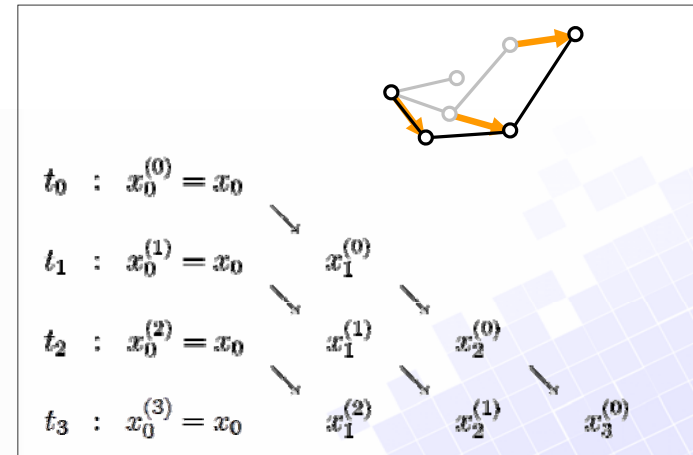


Andreas Kolb, Martin Lambers

Visualisierung

Streaklines (Streichlinien)

54

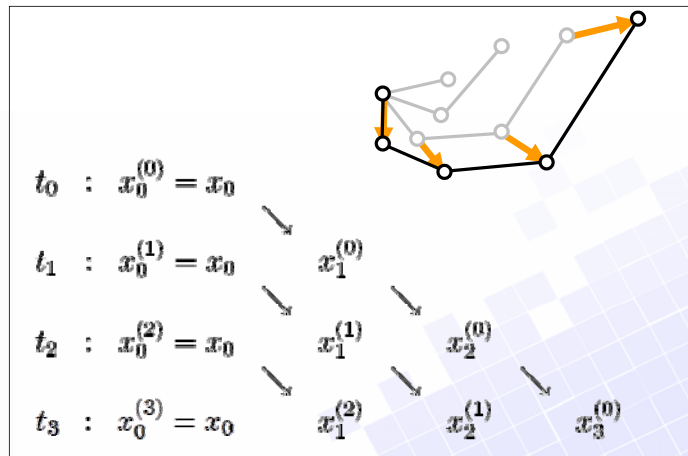


Andreas Kolb, Martin Lambers

Visualisierung

Streaklines (Streichlinien)

55



Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung

56

Partikelbahnen

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

Texturbasierte Verfahren

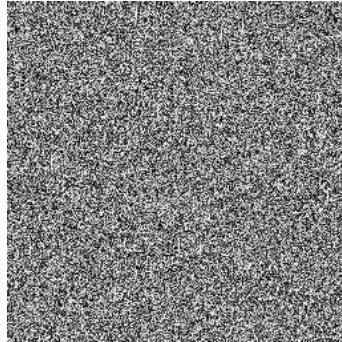
Andreas Kolb, Martin Lambers

Visualisierung

Exkurs: Rauschen

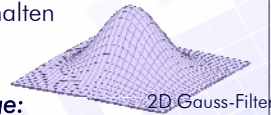
57

- 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?

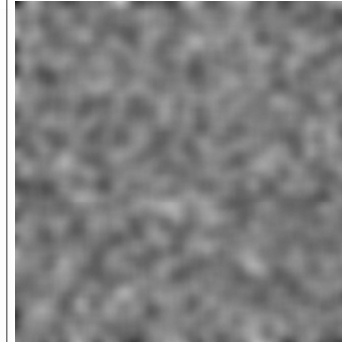
Andreas Kolb, Martin Lambers

Visualisierung

Exkurs: Rauschen

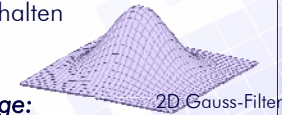
58

- 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?

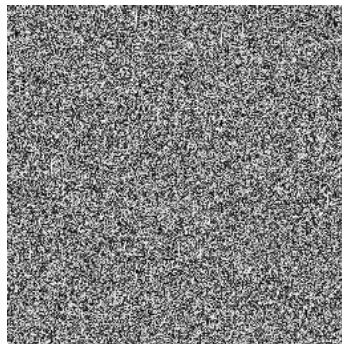
Andreas Kolb, Martin Lambers

Visualisierung

Exkurs: Rauschen

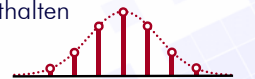
59

- 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 nur in eine Richtung (z.B. x-Richtung) glätte?

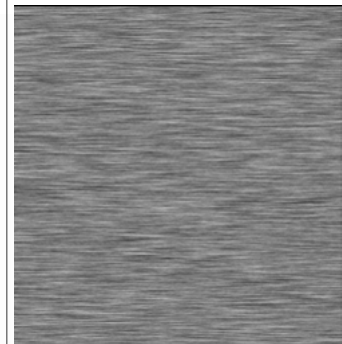
Andreas Kolb, Martin Lambers

Visualisierung

Exkurs: Rauschen

60

- 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 nur in eine Richtung (z.B. x-Richtung) glätte?

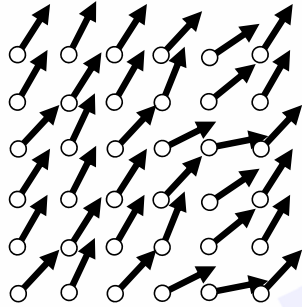
Andreas Kolb, Martin Lambers

Visualisierung

Line Integral Convolution

61

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



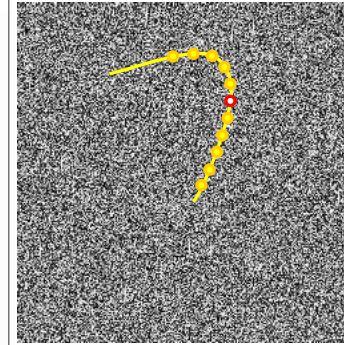
Andreas Kolb, Martin Lambers

Visualisierung

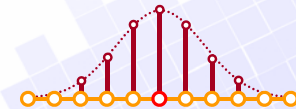
Line Integral Convolution

62

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 Glättungsfiler



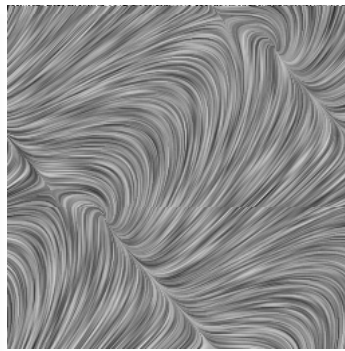
Andreas Kolb, Martin Lambers

Visualisierung

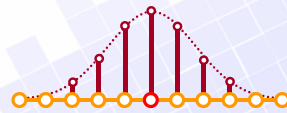
Line Integral Convolution

63

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 Glättungsfiler



Andreas Kolb, Martin Lambers

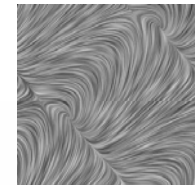
Visualisierung

Line Integral Convolution (LIC) ⁶⁴

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!

Andreas Kolb, Martin Lambers

Visualisierung

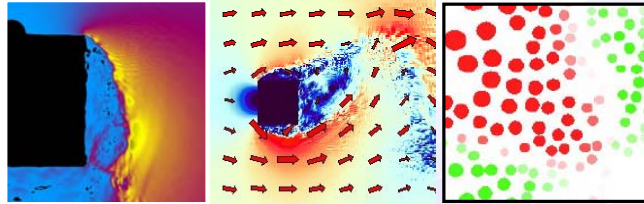
Zusammenfassung

65

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

Zusammenfassung

66

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

Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung

67

Visualisierungsverfahren für 2D Strömungsdaten

Integrationsbasierte Verfahren

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

Andreas Kolb, Martin Lambers

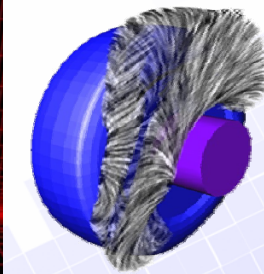
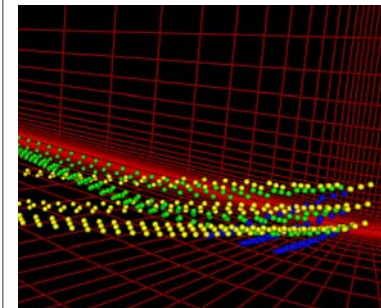
Visualisierung

Nächste Stunde

68

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

Andreas Kolb, Martin Lambers

Visualisierung