

Visualisierung: 2D Strömungsdaten

Christof Rezk-Salama

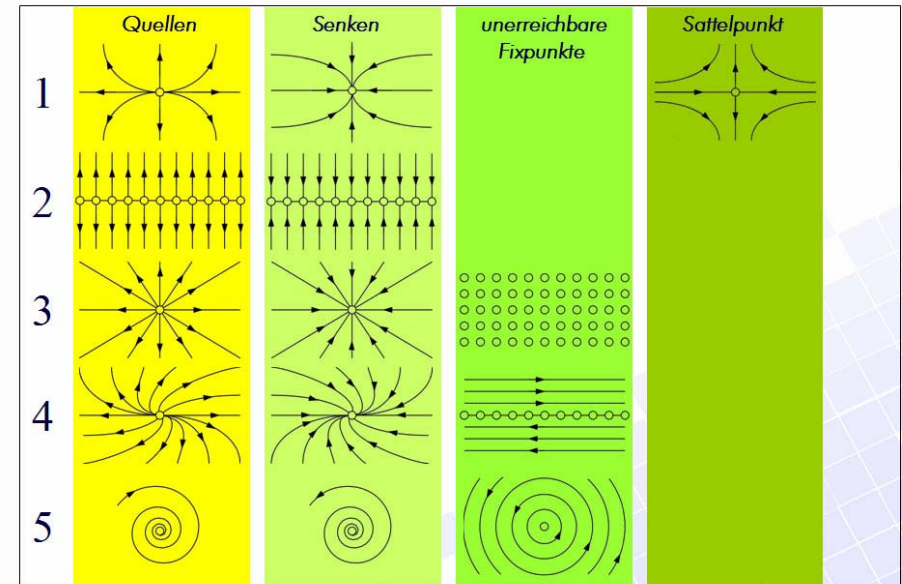
Visualisierung WS 03/04, 25.11.2003

computergraphik und multimedia systeme
universität siegen



Letzte Stunde: Vektorfeld-Topologie

2



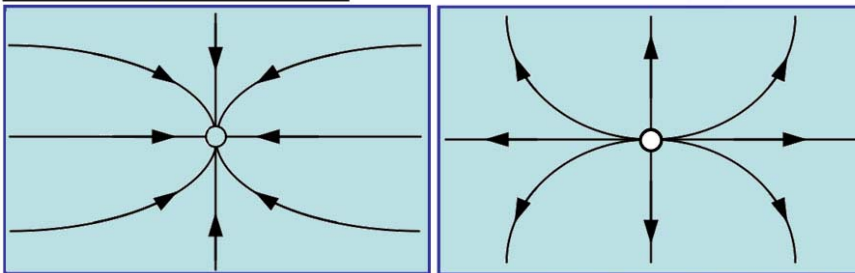
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Nachtrag

3

Kategorie 1:
Jacobimatrix besitzt
2 reelle Eigenwerte
2 reelle Eigenvektoren

Beispiel: $\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$



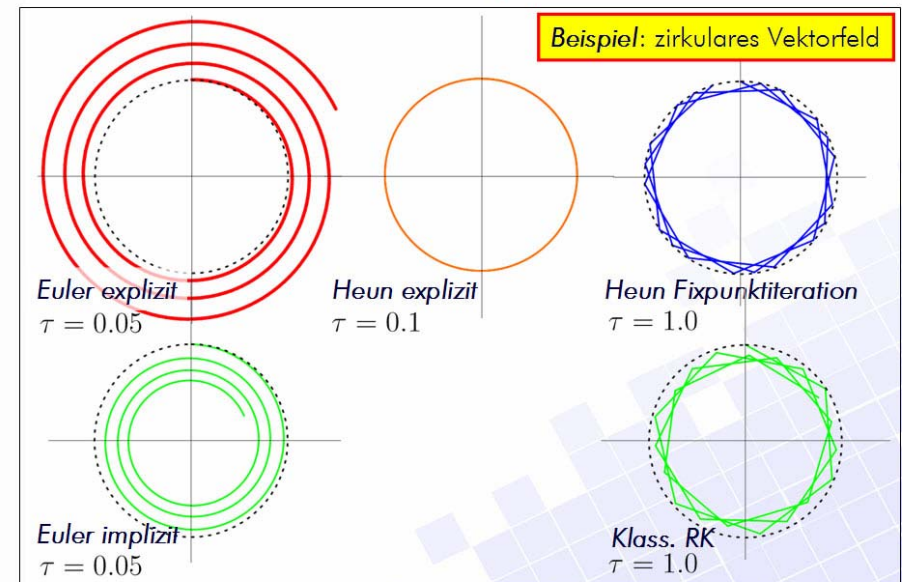
Kategorie 1A:
 $\lambda_1 < \lambda_2 < 0$

Kategorie 1B:
 $0 < \lambda_1 < \lambda_2$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Numerische Integration

4



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Simulationsdaten

5

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

6

● Direkte Verfahren

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

2D Strömungsdaten

6

● 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

7

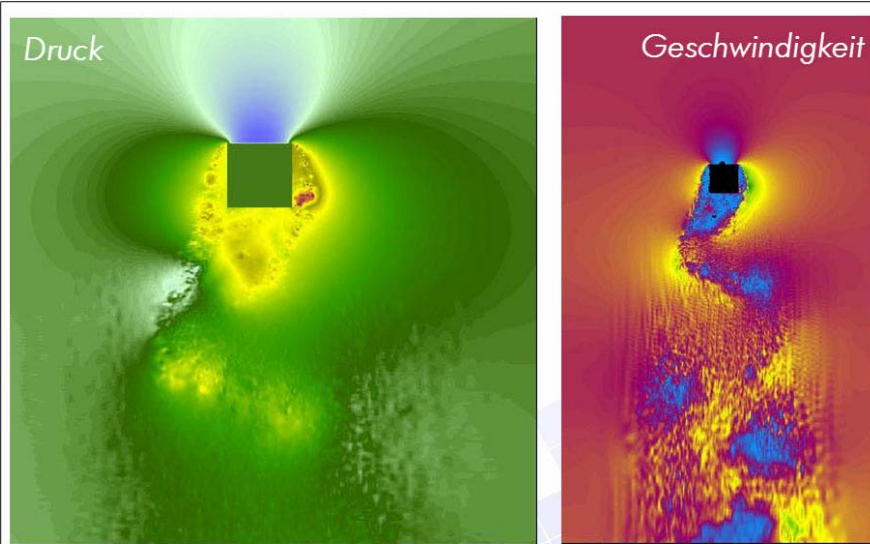
Verfahren mit möglichst direkter Transformation der Strömungsdaten in visuelle Objekte

- 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

8

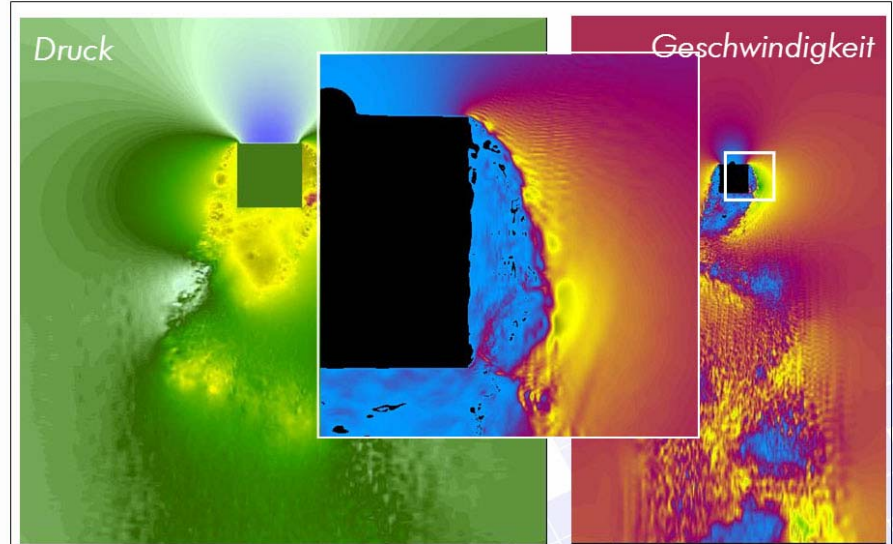


Bilder: TU Wien, Daten: A.E.P. Veldman, University of Groningen (the Netherlands)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

8



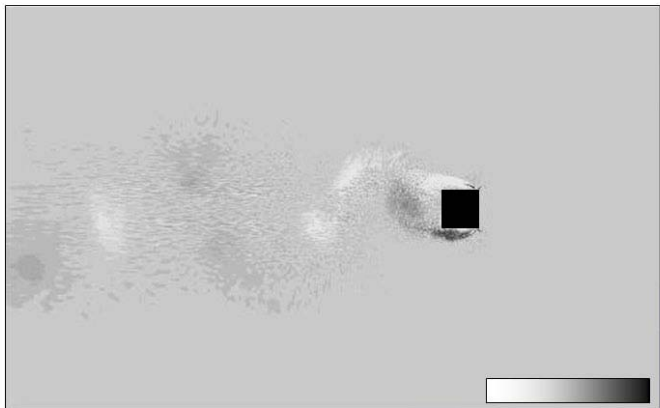
Bilder: TU Wien, Daten: A.E.P. Veldman, University of Groningen (the Netherlands)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Farbkodierung

9

● Farbkodierung der Rotation (vorticity)



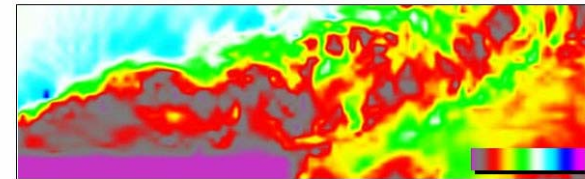
Beispiel: Strömung um einen quadratischen Block

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

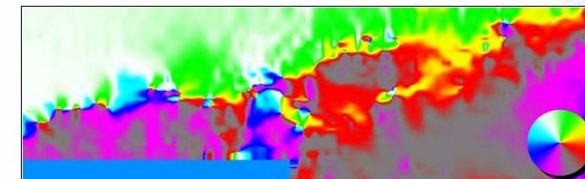
Farbkodierung

10

● Absolutbetrag der Strömungsgeschwindigkeit Vektorlänge farbkodiert



● Richtung der Strömung Richtung (Winkel in Polarkoordinaten) farbkodiert



Quelle: TU Wien



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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

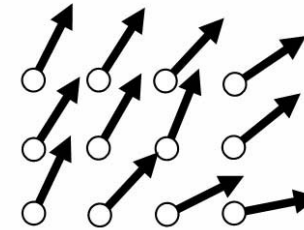
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

12

Datenpunkte werden durch kurze Linien oder Vektorpfeile dargestellt

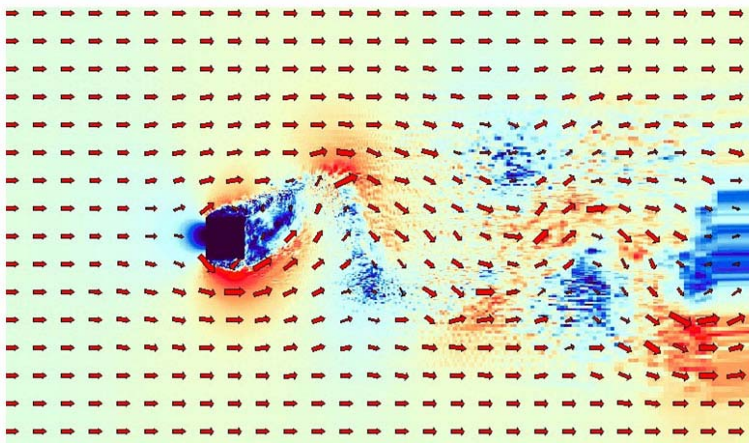
- z.B. gleichmäßig auf kartesischem Gitter
- auch *Arrow Plots* oder *Hedgehogs*.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

13

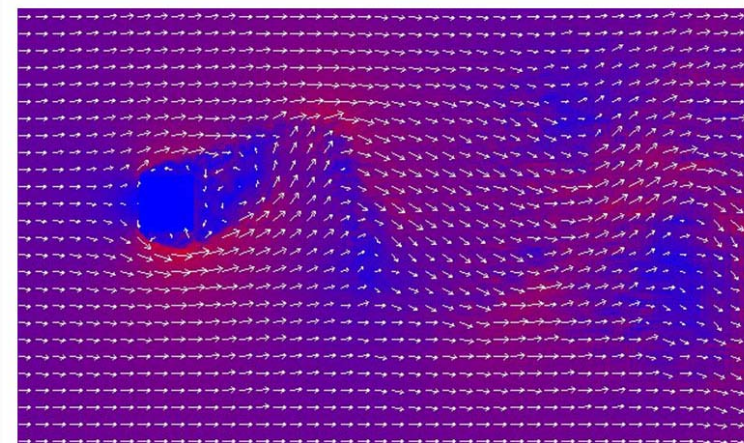


Bei relativ gleichmäßiger Strömung:
Länge/Größe des Pfeils entspricht Geschwindigkeitsbetrag

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Vektorpfeile

14



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

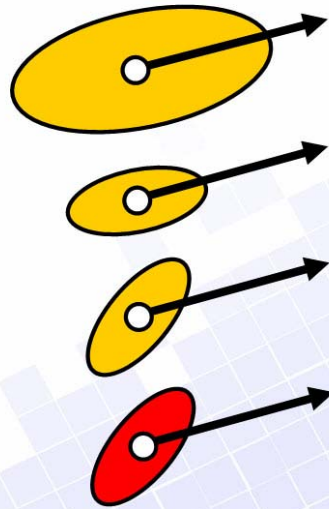
Glyphen/Icons

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



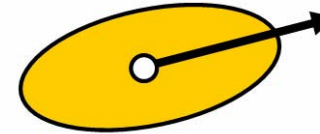
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Glyphen/Icons

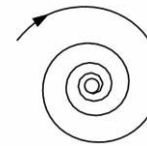
16

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.

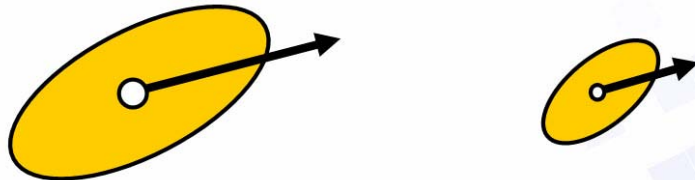
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Glyphen/Icons

17

Schwierigkeiten:

Auf Skalierung achten!



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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

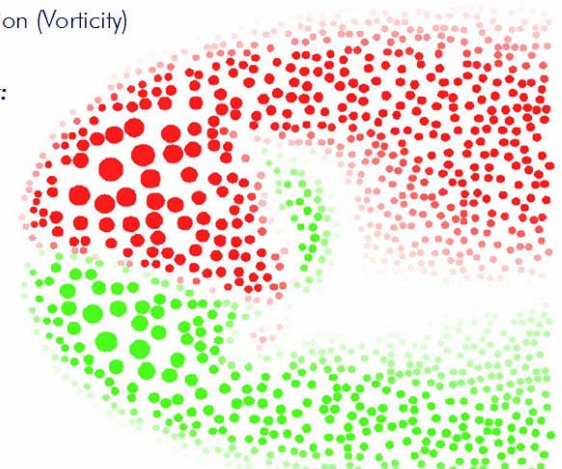
Glyphen

18

Beispiel:

Visualisierung der Rotation (Vorticity)

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



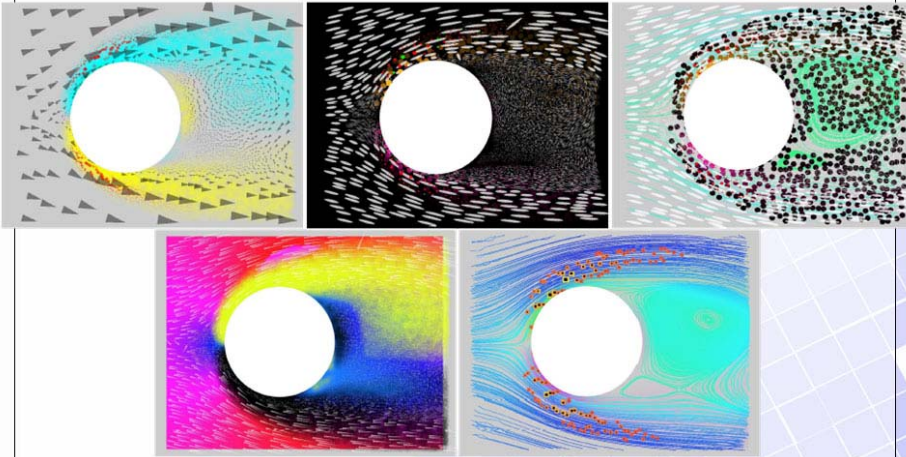
Quelle: Jason S. Sobel, Brown University, USA

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Kombinierte Verfahren

19

Farbcodierung und unterschiedliche Glyphen für verschiedene lokale Größen



Quelle: Jason S. Sobel, Brown University, USA

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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

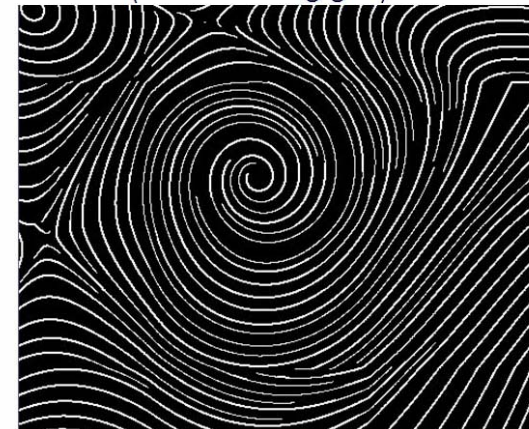
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

22

Beispiel:

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



Quelle: TU Wien

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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

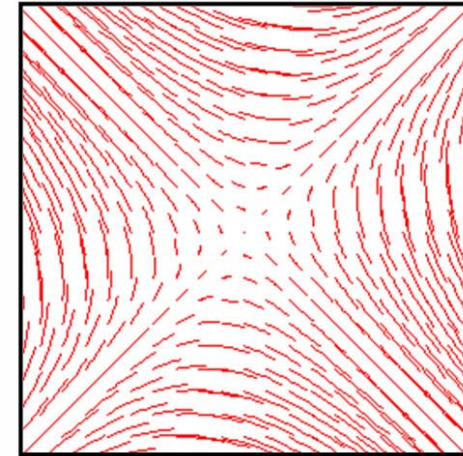
- Benutzerdefiniert (Flowfield-Probes)
- Uniformes Gitter von Startpunkten (streamlets) (wird leicht unübersichtlich und redundant)
- Zufällige Verteilung von Startpunkten
- Spezielle Algorithmen

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

24

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

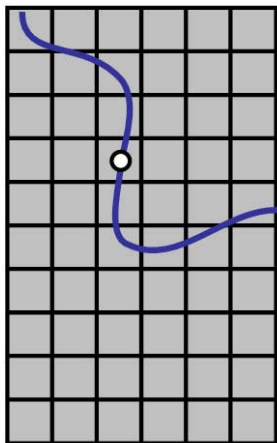


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

25

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



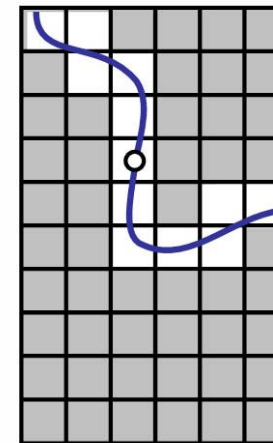
- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).
- Wähle den ersten Startpunkt (beliebig)
- Bestimme die Partikelbahn vor- und rückwärts (solang es geht)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Wahl der Startpunkte

25

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



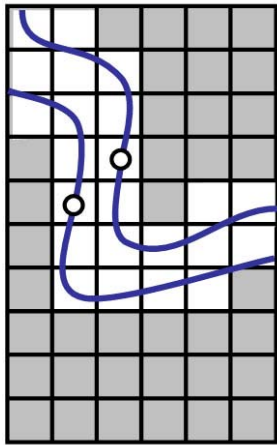
- Teile die Ebene in Zellen bestimmter Größe (unabhängig vom zugrunde liegenden Gitter).
- Wähle den ersten Startpunkt (beliebig)
- Bestimme die Partikelbahn vor- und rückwärts (solang es geht)
- Markiere alle Zellen, die die Partikelbahn schneidet

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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

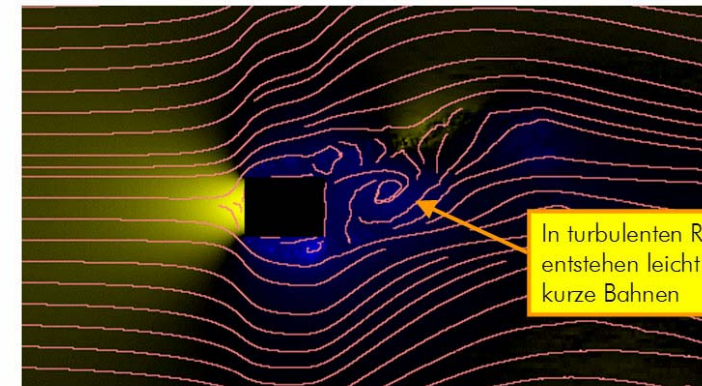
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

26

Beispiel:

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



In turbulenten Regionen entstehen leicht sehr kurze Bahnen

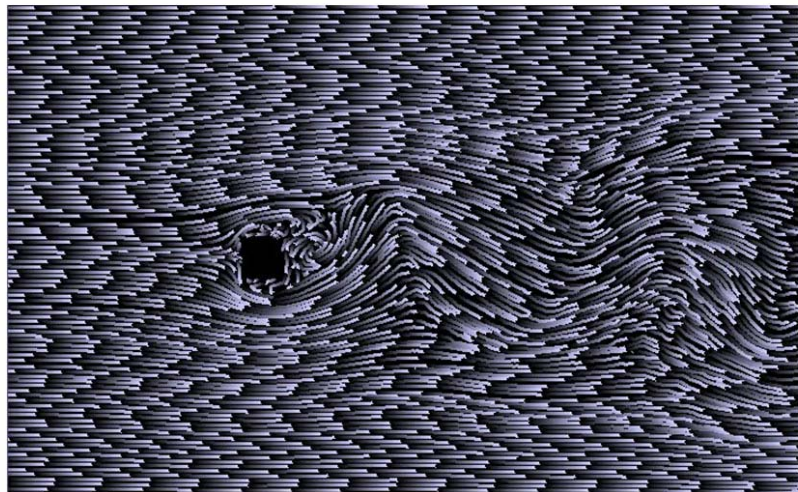
Quelle: TU Wien

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

27

- **Farbcodierung der Zeitschritte**



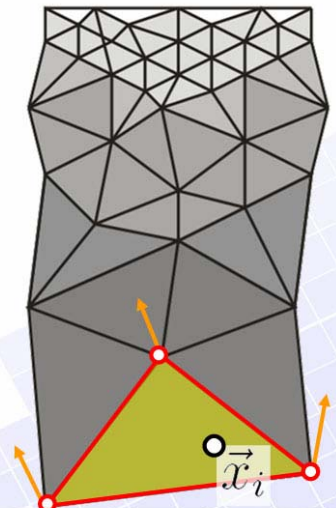
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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)



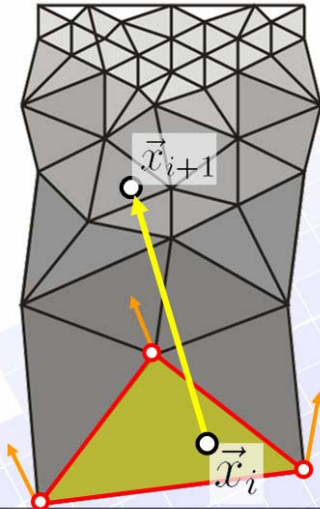
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

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.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Partikelbahnen

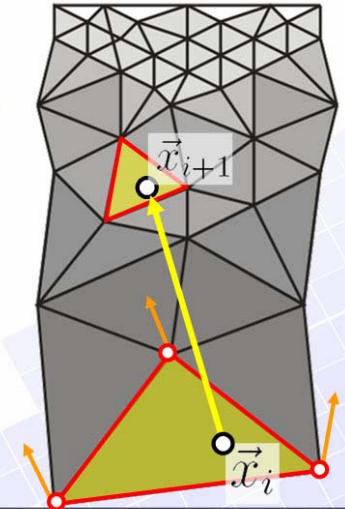
28

● 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

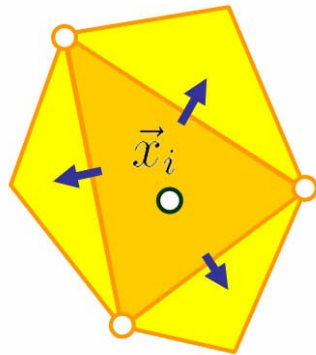


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

29

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



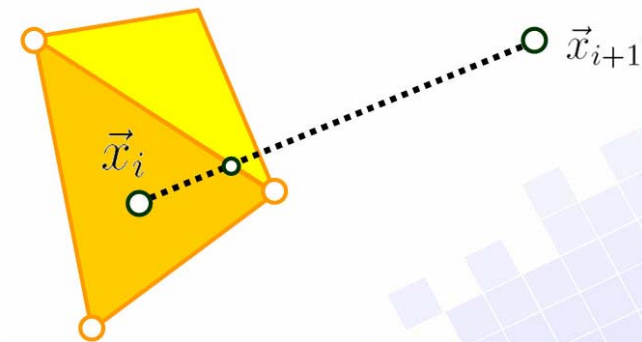
● **Voraussetzung:** Datenstruktur, die für jedes Dreieck Referenzen auf die benachbarte Dreiecke speichert (z.B. *extended winged edge*)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

30

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

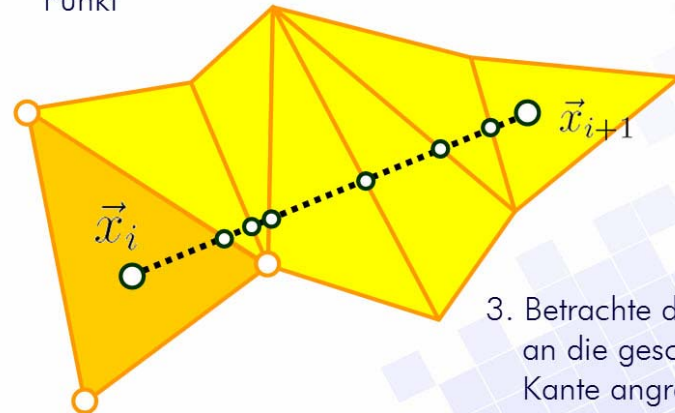


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

30

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



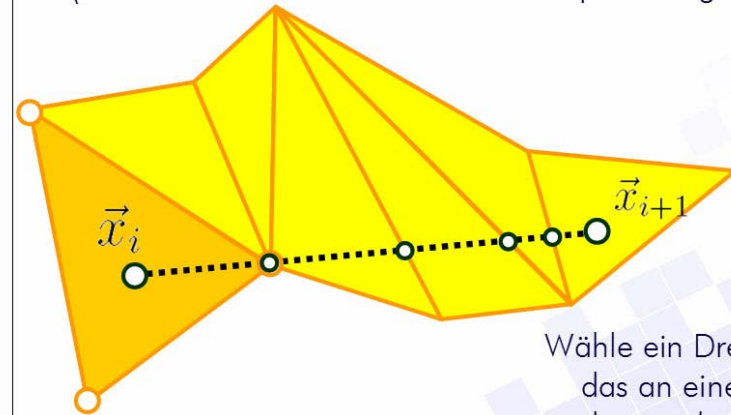
3. Betrachte die Zelle die an die geschnittene Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

31

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



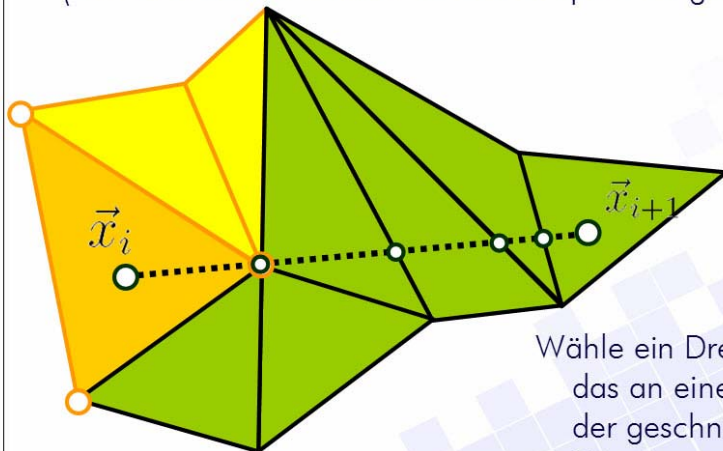
Wähle ein Dreieck, das an eine beliebige der geschnittenen Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im Dreiecksnetz

31

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



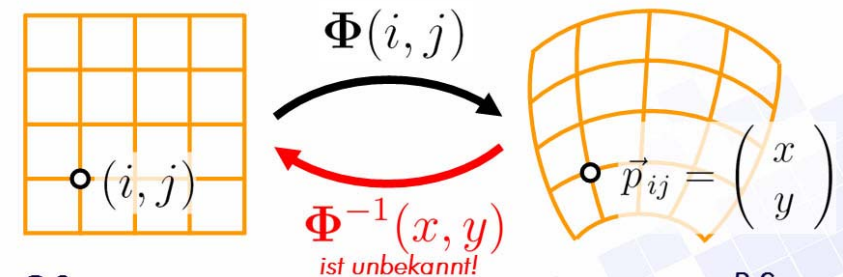
Wähle ein Dreieck, das an eine beliebige der geschnittenen Kante angrenzt.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Curvilineare Gitter

32

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



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

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

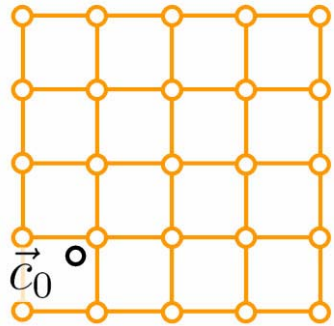
P-Space
physical space
physikalischer Raum
curvilineares Gitter
hier sind die Vektoren gültig

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Curvilineare Gitter

33

Particle Tracing auf curvilinearen Gittern



C-Space

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

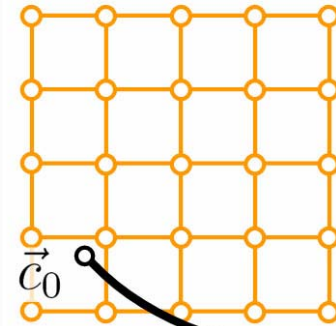
Für einen **beliebigen** Punkt im C-Space kann ich demnach **einen Datenwert und** eine dazugehörige **Position im P-Space** bilinear interpolieren.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

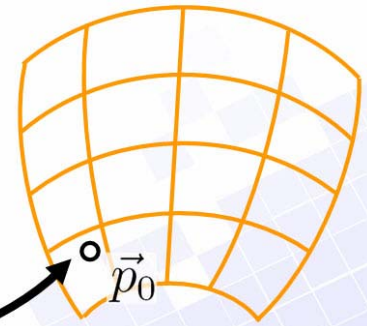
Curvilineare Gitter

33

Particle Tracing auf curvilinearen Gittern



C-Space



P-Space

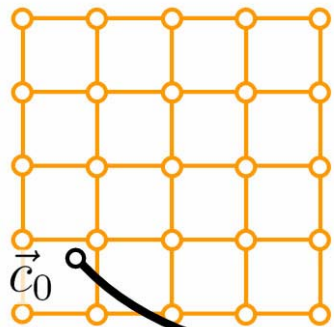
$$\Phi(\vec{c}_0) = \vec{p}_0$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Curvilineare Gitter

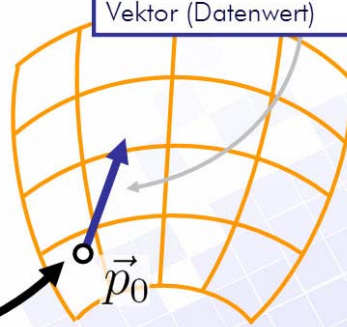
33

Particle Tracing auf curvilinearen Gittern



C-Space

Im C-Space interpolierter Vektor (Datenwert)



P-Space

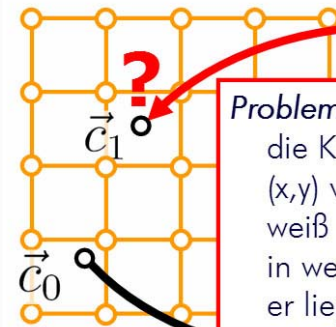
$$\Phi(\vec{c}_0) = \vec{p}_0$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

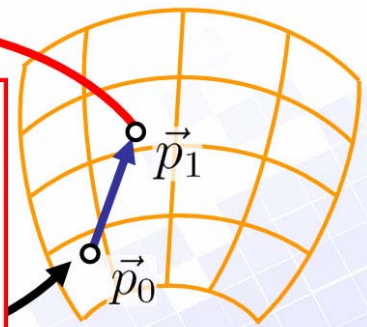
Curvilineare Gitter

33

Particle Tracing auf curvilinearen Gittern



C-Space



P-Space

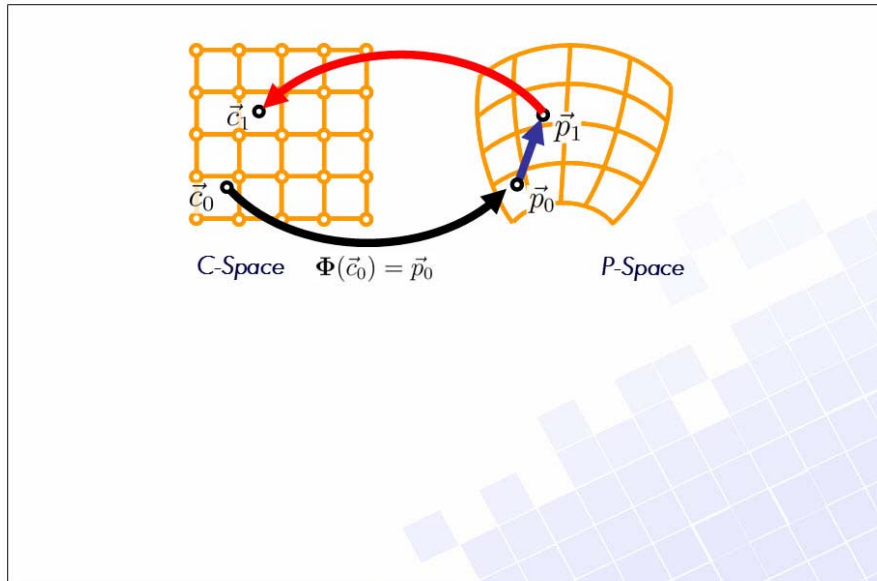
Problem: Ich kenne die Koordinaten (x,y) von $\vec{p}_1$, weiß aber nicht in welcher Zelle er liegt.

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im curvilinearen Gitter

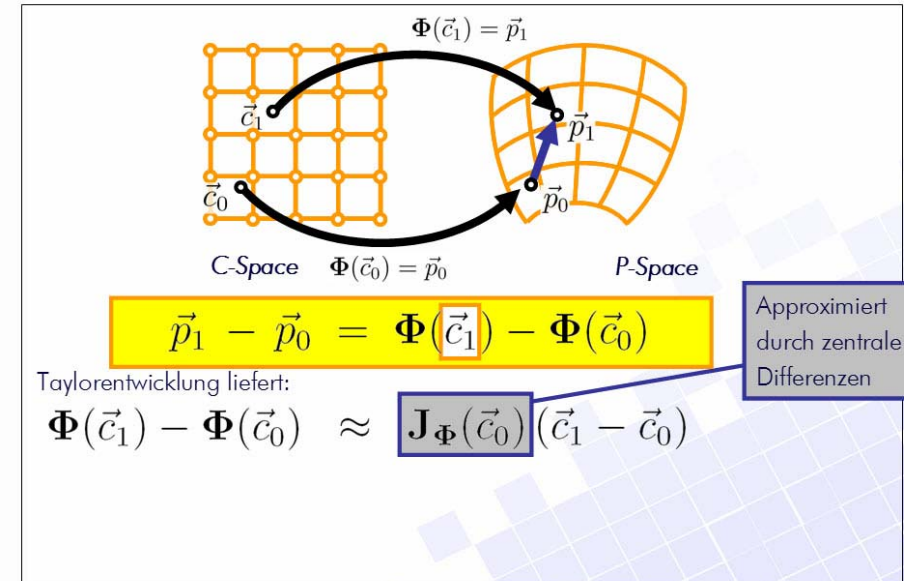
34



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im curvilinearen Gitter

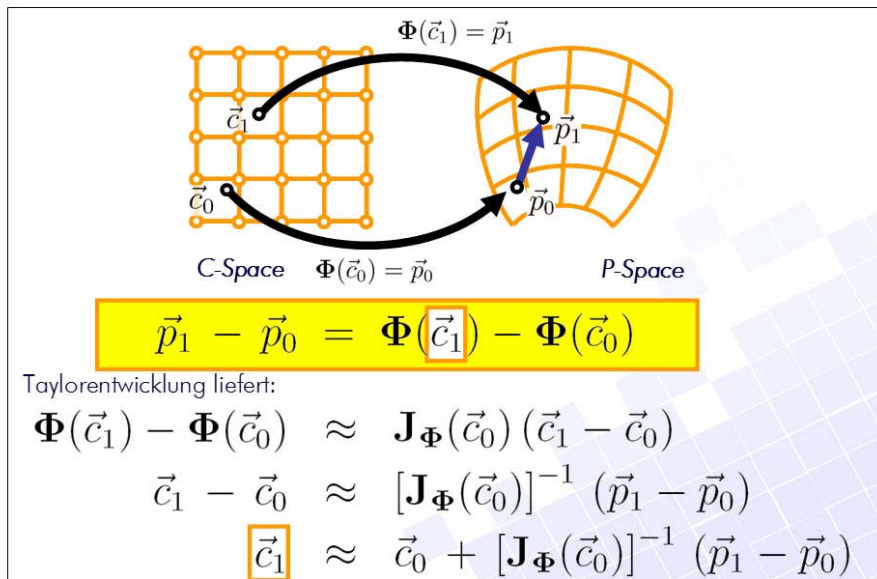
34



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im curvilinearen Gitter

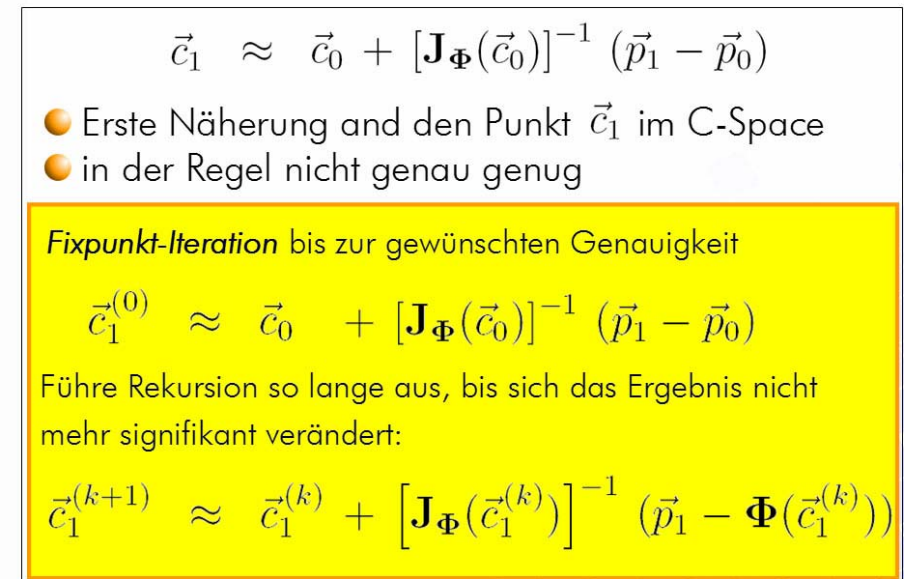
34



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im curvilinearen Gitter

35



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zellsuche im curvilinearen Gitter

36

Fixpunkt-Iteration (Pseudo-Code):

```

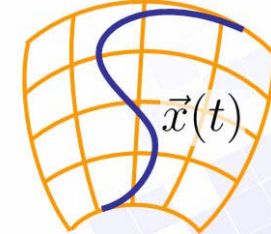
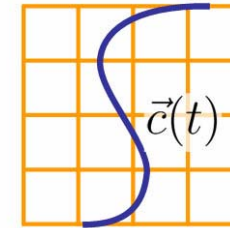
00 c = c0; p = p0; // initialize
01 do
02     dp = p1 - p;
03     dc = inverse(Jacobi(c)) * dp;
04     c = c + dc;
05     p = Φ(c);
06 until (dc < toleranz)
07 c1 = c;
    
```

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space Verfahren:

37

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



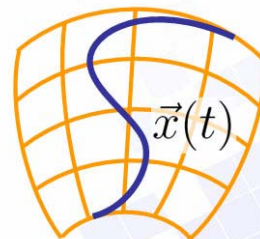
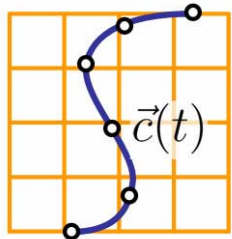
Jeder Partikelbahn $\vec{x}(t)$ im P-Space entspricht eine Partikelbahn $\vec{c}(t)$ im C-Space.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space Verfahren:

37

- 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.

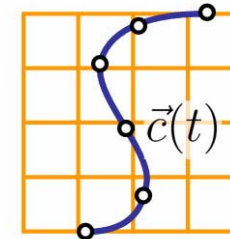
Idee: Bestimme die Partikelbahn im C-Space

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

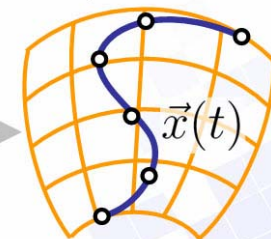
C-Space Verfahren:

37

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



$\Phi(i, j)$



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

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:

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

ableiten

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:

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

nachdifferenzieren (Kettenregel)

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:

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

40

- 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

41

- 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

42

- 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:

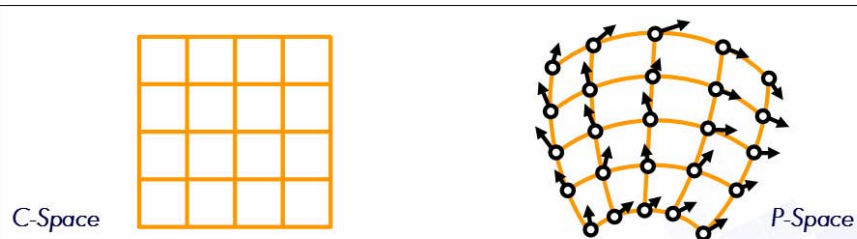
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{\vec{v}}(\vec{c}(t))$$

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

42



Differentialgleichung im C-Space:

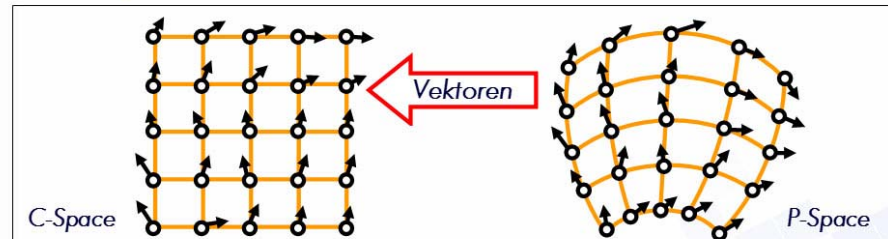
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{\vec{v}}(\vec{c}(t))$$

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

42



Differentialgleichung im C-Space:

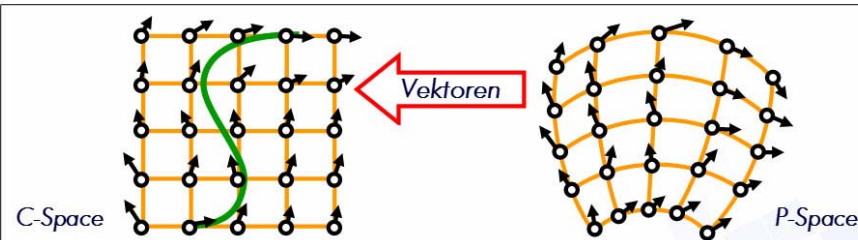
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{\vec{v}}(\vec{c}(t))$$

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

42



Differentialgleichung im C-Space:

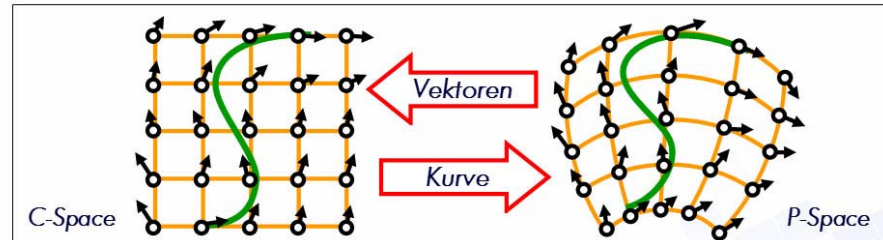
$$\frac{\partial \vec{c}(t)}{\partial t} = \tilde{v}(\vec{c}(t))$$

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

C-Space-Verfahren

42



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

43

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zeitabhängige Daten

44

- **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

45

- **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

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

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zeitabhängige Partikelbahnen

46

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

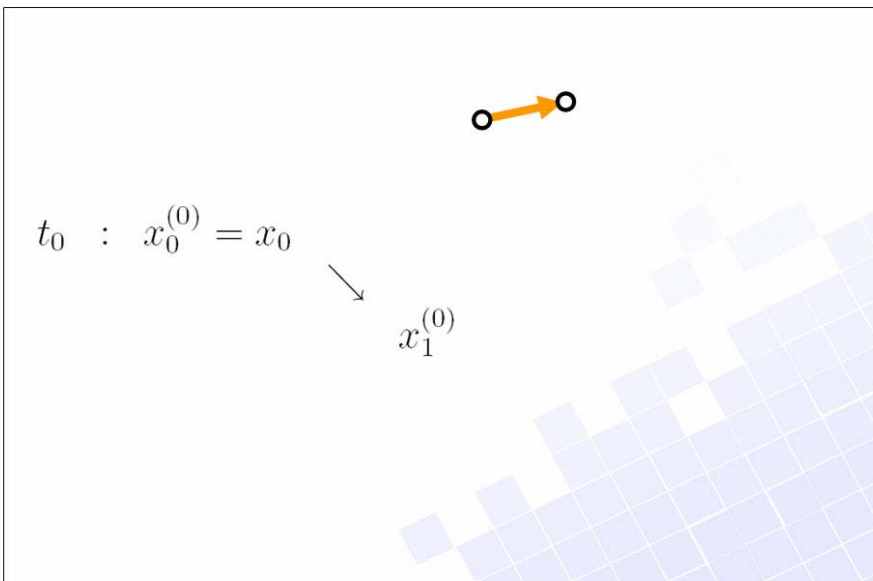
$$\frac{\partial \vec{x}(t)}{\partial t} = \vec{v}(\vec{x}(t), t) \quad \vec{x}_0 \text{ konstant, } t_0 \text{ variabel}$$

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

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

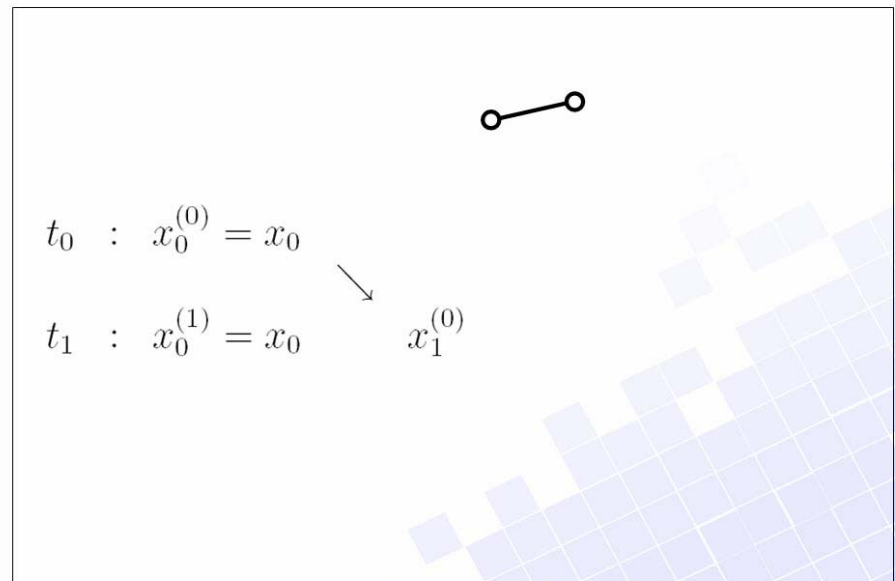
47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

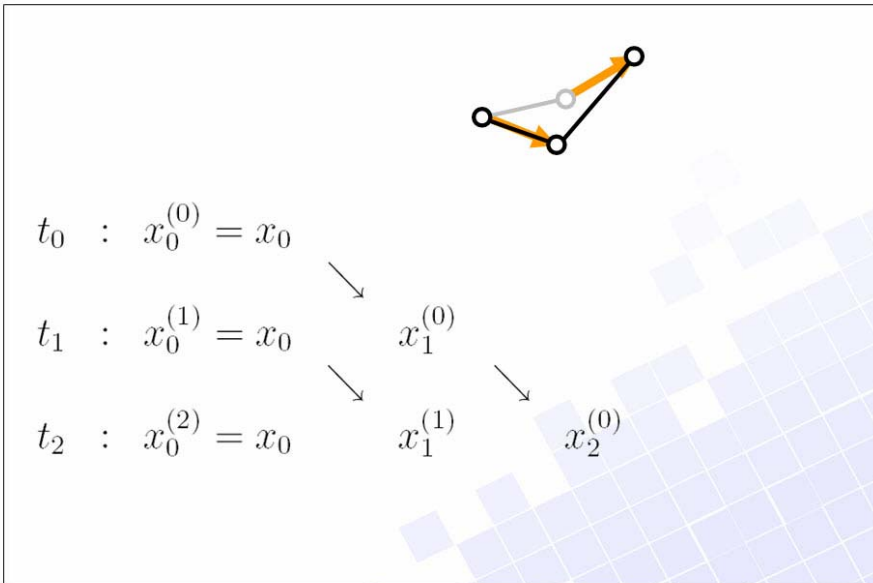
47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

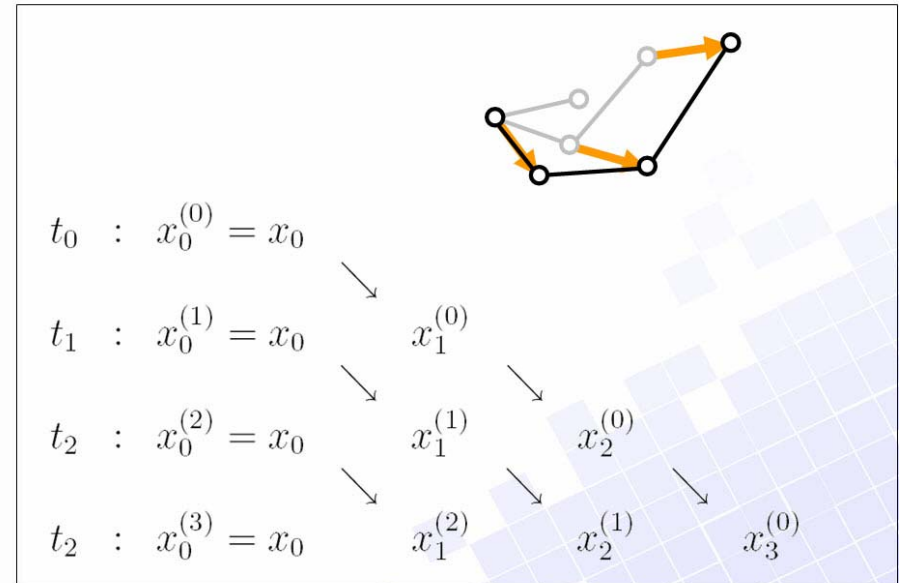
47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

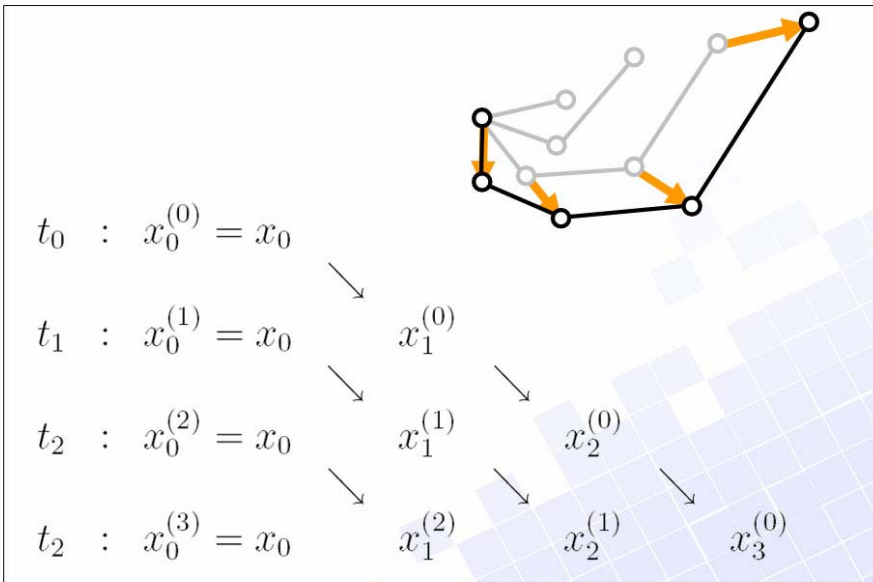
47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

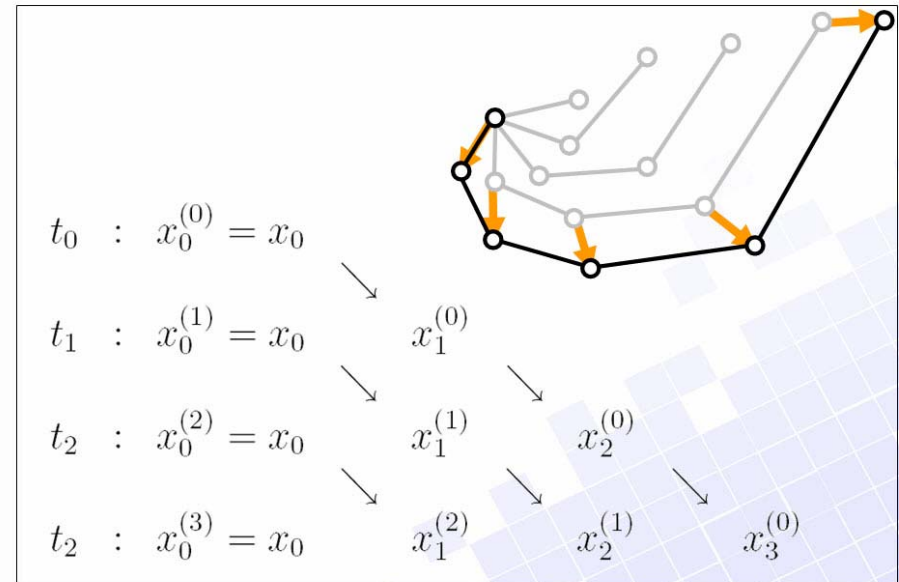
47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streaklines

47

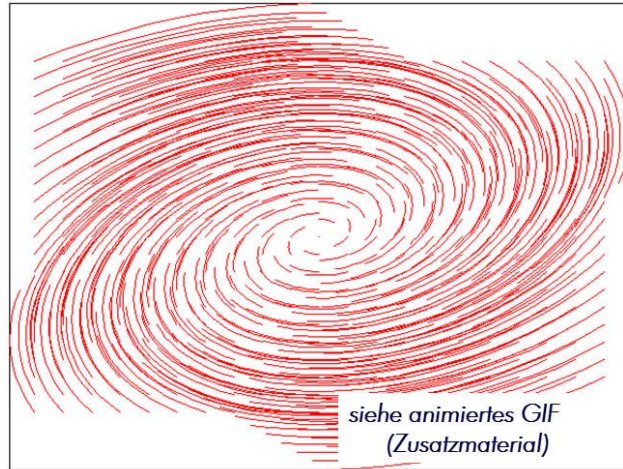


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Streamlines (Feldlinien)

48

- Feldlinien zu einem festen Zeitpunkt



siehe animiertes GIF
(Zusatzmaterial)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

49

Partikelbahnen

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

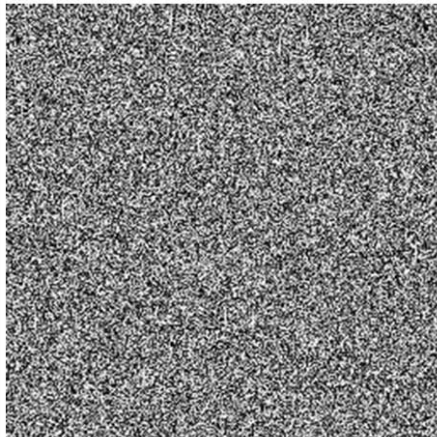
Texturbasierte Verfahren

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Exkurs: Rauschen

50

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



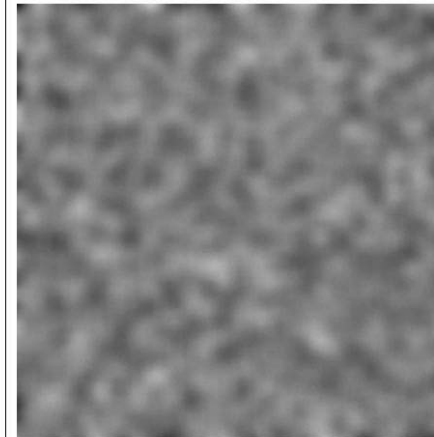
Signalverarbeitung:
In Rauschen sind **alle** Frequenzen eines bestimmten Frequenzbandes enthalten

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Exkurs: Rauschen

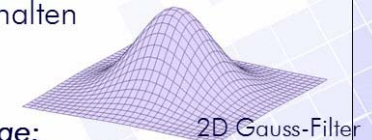
50

- 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?

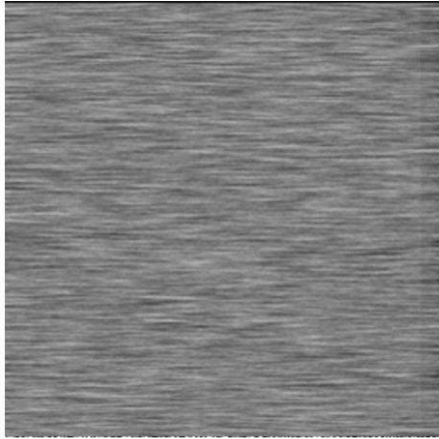


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Exkurs: Rauschen

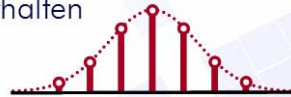
51

- 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

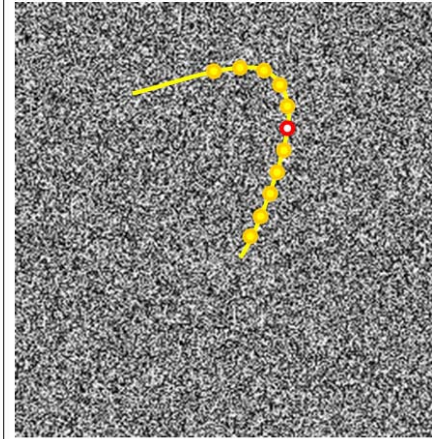
Was passiert wenn ich nur in eine Richtung (z.B. x-Richtung) glätte?

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Line Integral Convolution

52

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



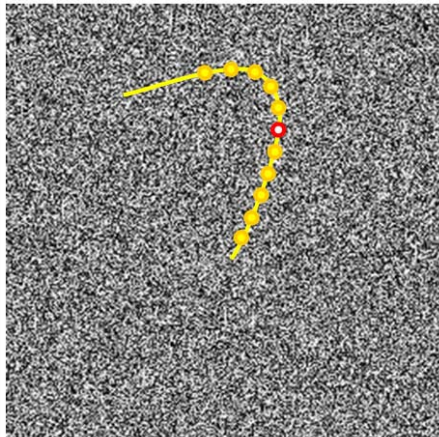
- Bestimme für jeden Pixel des Ergebnisbildes eine kurze Partikelbahn (vor- und rückwärts)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Line Integral Convolution

52

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

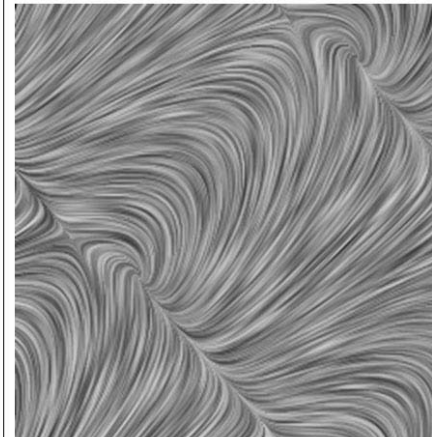


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Line Integral Convolution

52

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

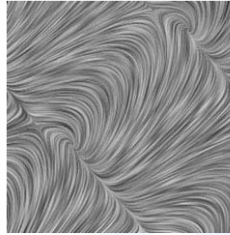
53

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

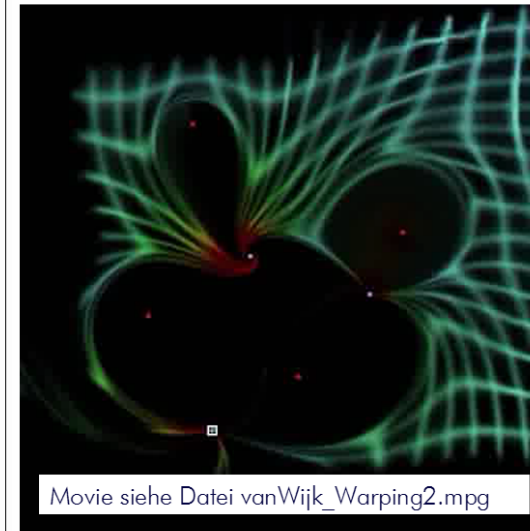
Nachteil:

- Relativ hoher Rechenaufwand
- Kein Absolutbetrag der Geschwindigkeit!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

54



Texture Advection:

„Verzerre ein Eingabebild entlang der Strömungslinien des Vektorfeldes“

Quelle:

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

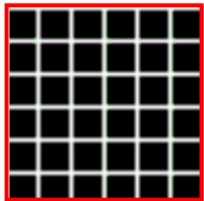
Movie siehe Datei vanWijk_Warping2.mpg

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

Texture Advection Implementierung:



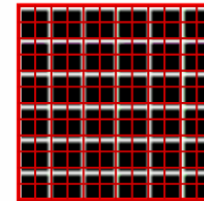
- Definiere das Eingabebild als *Polygon mit Textur*.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

Texture Advection Implementierung:



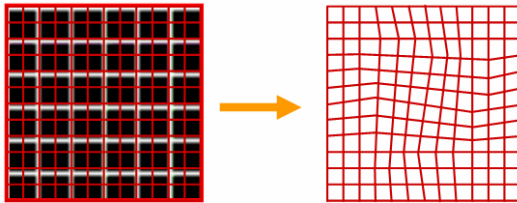
- Definiere das Eingabebild als *Polygon mit Textur*.
- Zerlege das Polygon in *viele kleine Polygone*

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

Texture Advection Implementierung:



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

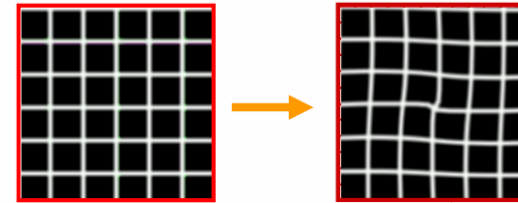


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

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

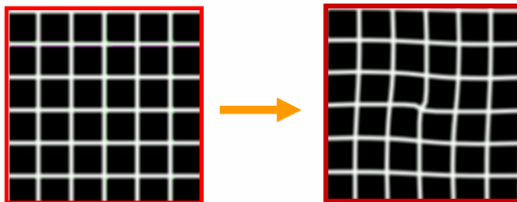


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

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

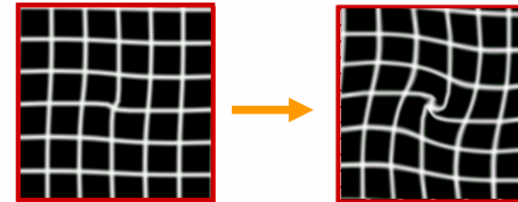


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

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

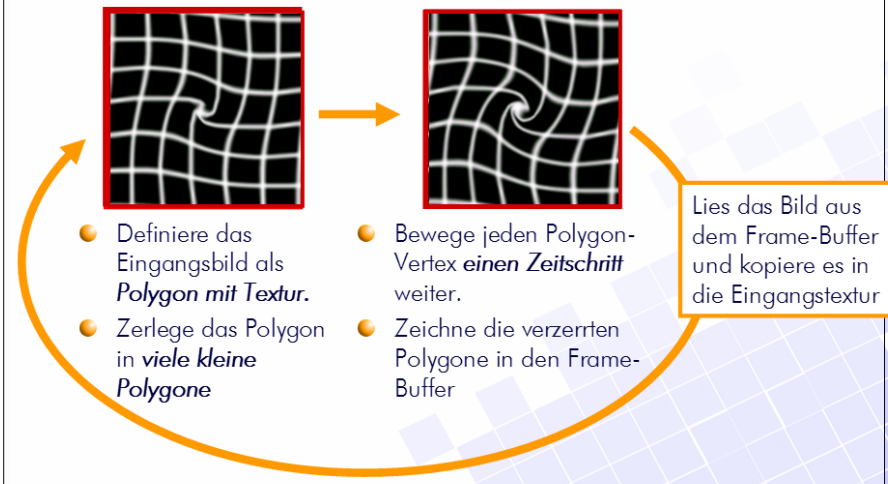


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

Texture Advection Implementierung:

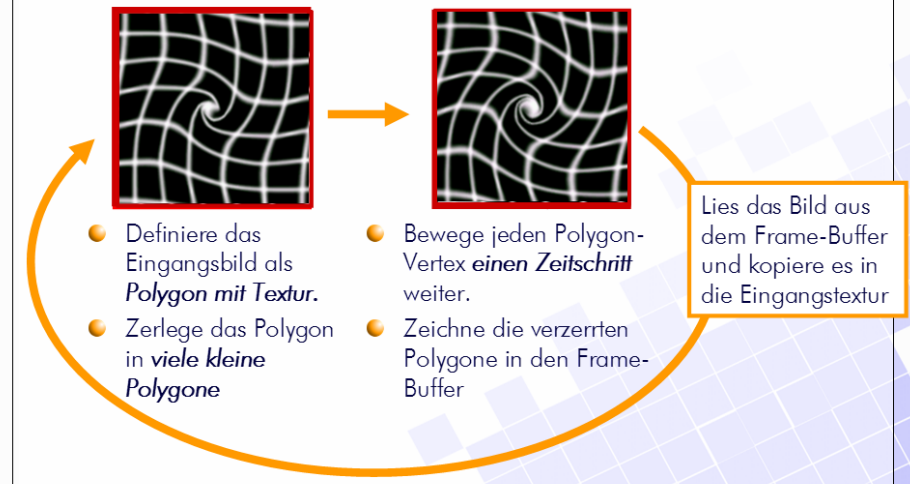


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

55

Texture Advection Implementierung:

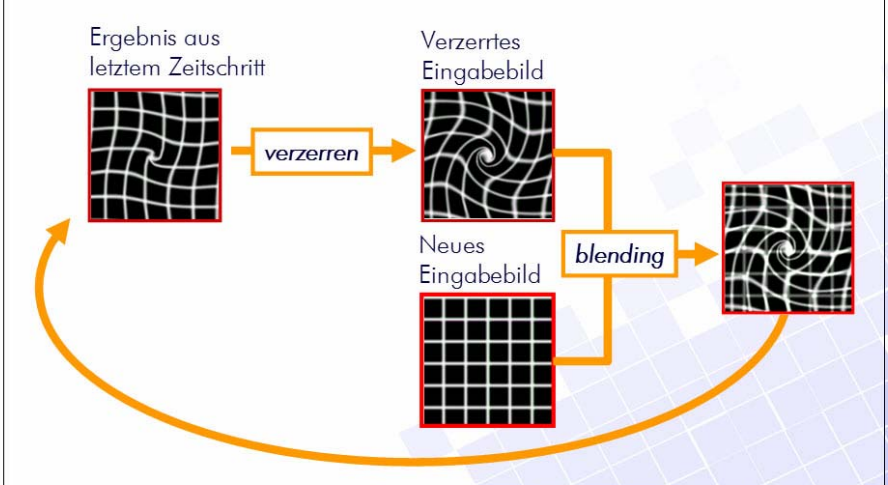


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

56

Erweiterte Implementierung:



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Techniken

57

Beispiel der erweiterten Implementierung



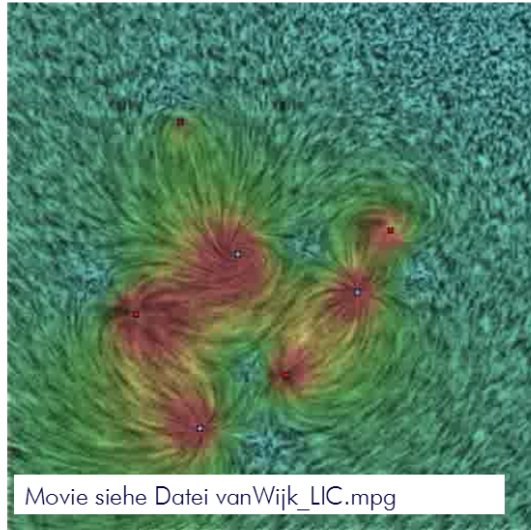
Quelle:
van Wijk, Univ. Eindhoven
Image-based flow visualization
ACM SIGGRAPH 2003

Movie siehe Datei vanWijk_Warping.mpg

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Verfahren

58



Beispiel der erweiterten Implementierung:
Simulation von LIC

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

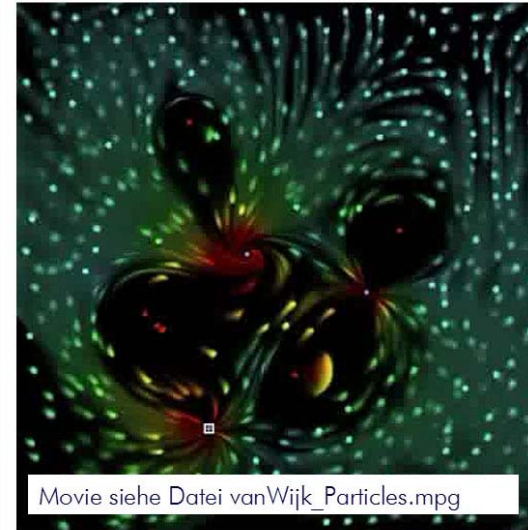
Movie siehe Datei vanWijk_LIC.mpg



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Verfahren

59



Beispiel der erweiterten Implementierung:
Partikelsysteme

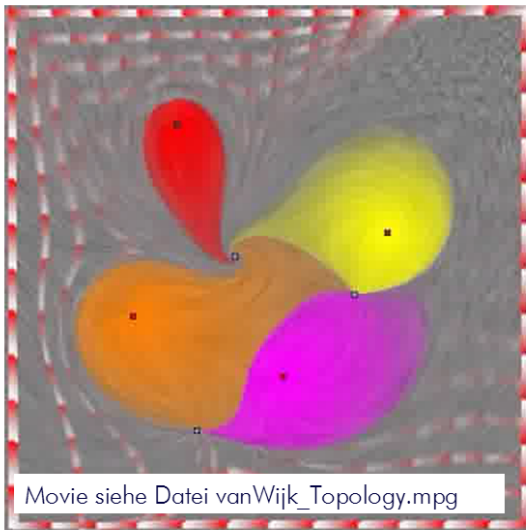
Quelle:
van Wijk, Univ. Eindhoven
Image-based flow visualization
ACM SIGGRAPH 2003

Movie siehe Datei vanWijk_Particles.mpg

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Verfahren

60



Beispiel der erweiterten Implementierung.

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

Movie siehe Datei vanWijk_Topology.mpg



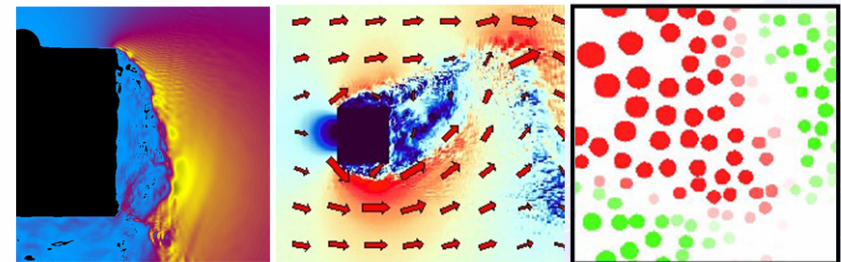
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

61

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

Zusammenfassung

62

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

Zusammenfassung

63

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

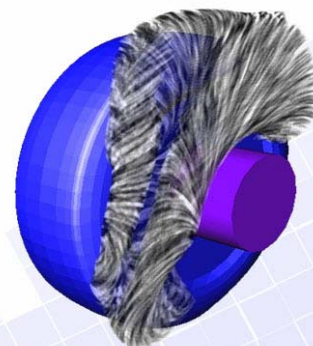
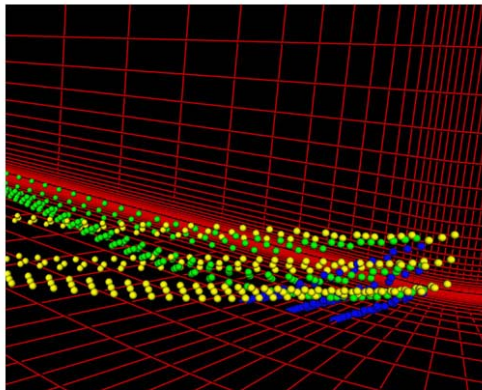
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Nächste Stunde

64

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