Steffen Heinrich und Stephan Brumme

2D Line Integral Convolution

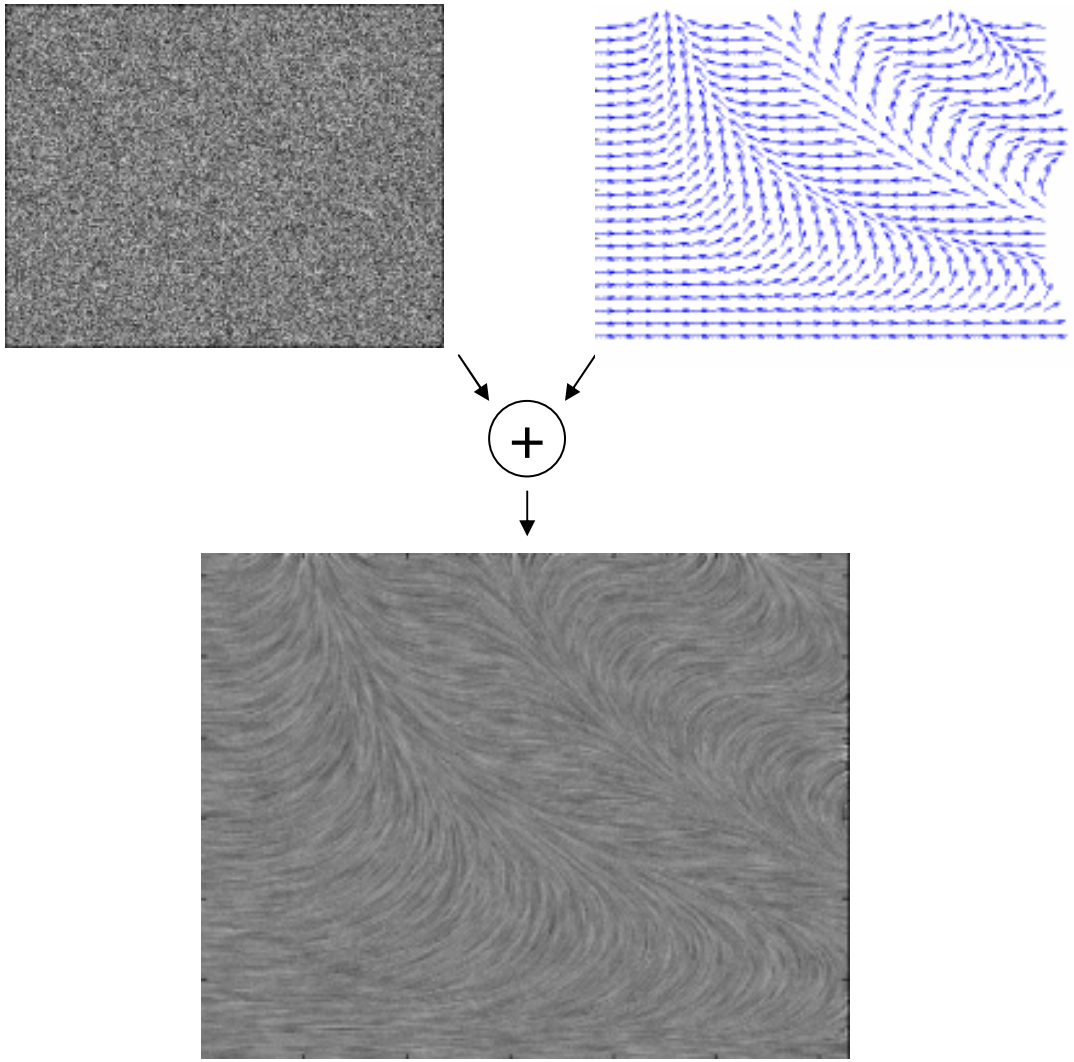
Arbeitsweise und Techniken

2. Juli 2003

Lehrstuhl für
Computergrafische Systeme

Agenda

- Überblick
- Arbeitsweise
  - Theorie
  - Praxis
- Beispiele
- Ausblick
- Quellen

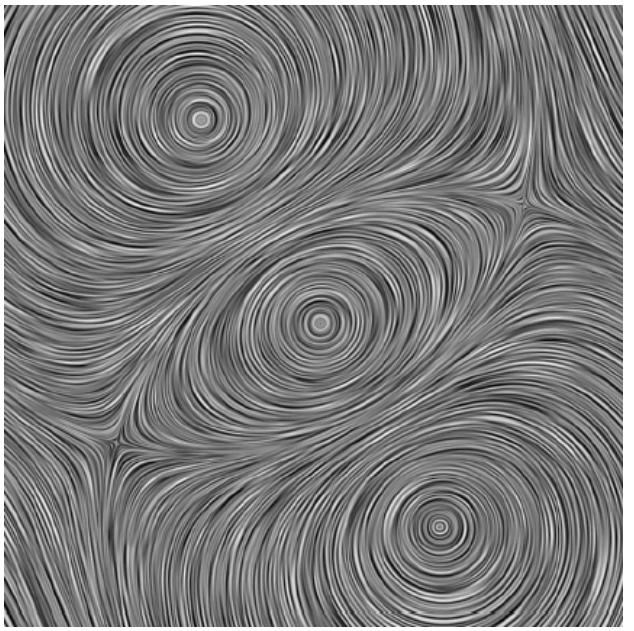


Überblick Line Integral Convolution

- entwickelt von Brian Cabral und Leith Casey Leedom
  - „Imaging Vector Fields Using Line Integral Convolution“ (1993)
  - stetige Weiterentwicklung, z.B. am ZIB in Berlin

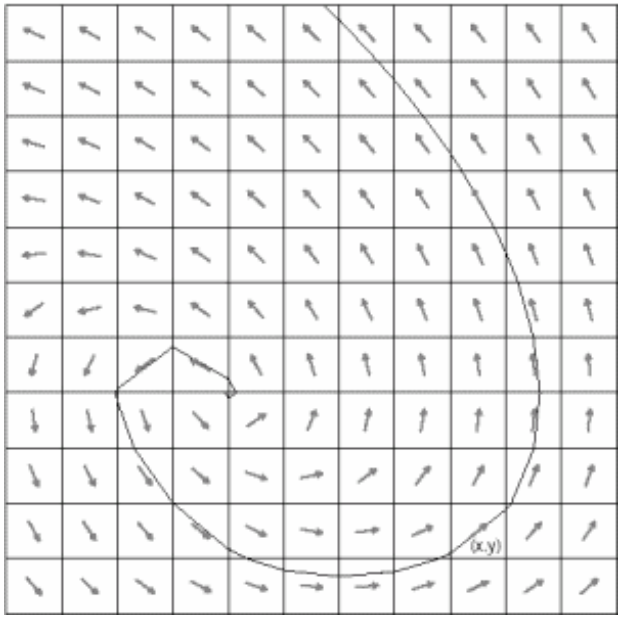
- Idee: Darstellung von Vektorfelder auf Basis von Texturalgorithmen

- Eigenschaften:
  - lokale Operationen
    - hoch parallelisierbar !
  - Erweiterbar auf 3D
  - Nutzung moderner Grafikhardware
    - Shader



Arbeitsweise (Theorie, I)

- Aufteilung des Vektorfeldes in ein (reguläres) Gitter
- Iteratives Verfahren



$$P_i = P_{i-1} + direction \cdot length$$

Δs_{i-1}

normalisiert, entstammt dem Vektorfeld

Abstand bis zur nächsten Zelle im Gitter

$$\frac{V(\lfloor P_{i-1} \rfloor)}{\|V(\lfloor P_{i-1} \rfloor)\|}$$

$$(e,c) \in \begin{Bmatrix} (top, y) \\ (bottom, y) \\ (left, x) \\ (right, x) \end{Bmatrix}$$

$$\Delta s_i = \min(s_{top}, s_{bottom}, s_{left}, s_{right})$$

$$s_e = \begin{cases} \infty & if\ V \| e \\ 0 & if\ \frac{\lfloor P_c \rfloor - P_c}{V_c} < 0 \\ \frac{\lfloor P_c \rfloor - P_c}{V_c} & otherwise \end{cases}$$

Arbeitsweise (Theorie, II)

- Verfolgung des Vektorfeldes auch rückwärts

$$P_i' = P_{i-1}' + direction' \cdot length'$$

- Bestimmung eines Ausgabepixels:

$$out(x,y) = \frac{\sum_{i=0}^{l} in(\lfloor P_i \rfloor) \cdot weight_i + \sum_{i=0}^{l} in(\lfloor P_i' \rfloor) \cdot weight_i'}{\sum_{i=0}^{l} weight_i + \sum_{i=0}^{l} weight_i'}$$

  - wobei $s_l \le L < s_{l+1}$

$$weight_i = \int_{s_i}^{s_i + \Delta s_i} k(w)\, dw$$

normalisierender convolution kernel, z.B. Hanning-Filter

Arbeitsweise (Theorie, III)

- Wahl einer sinnvollen Pfadlänge L in Pixel notwendig
  - z.B. L=10
  - entscheidend für die Bildung einer local stream line

von oben links
nach unten rechts:
L=0, L=5,
L=10, L=20

Arbeitsweise (Praxis, I)

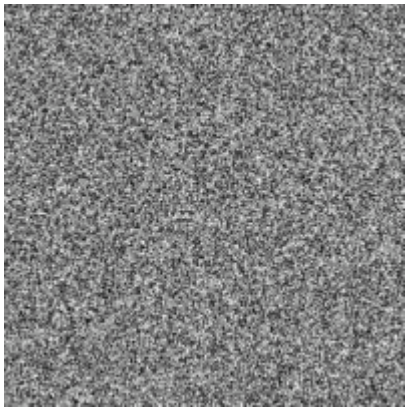
- Pseudocode

```
lade Vektorfeld und Noise-Textur
für jeden Pixel (x,y)
    verfolge entlang des Pfades vorwärts, dann rückwärts
        addiere gewichtete Farbwerte aus der Noise-Textur
    schreibe normalisierte Summe in Ergebnisbild
```

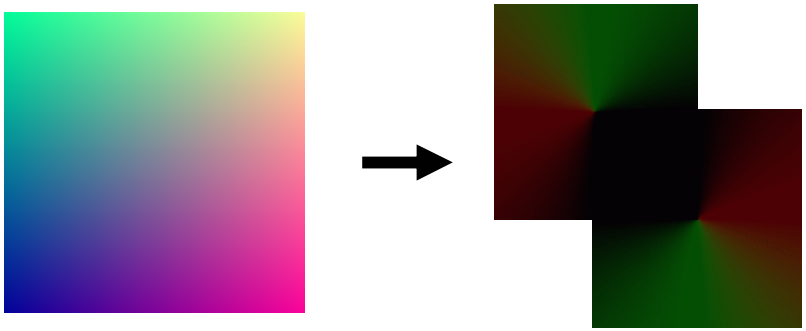
- kritische Operationen
  - Pfadverfolgung
    → Dependent Texture Lookup
  - Convolution Kernel
    → Wahl einer konstanten Funktion

Arbeitsweise (Praxis, II)

- Vorberechnung von Texturen
  - Noise als Luminanztextur



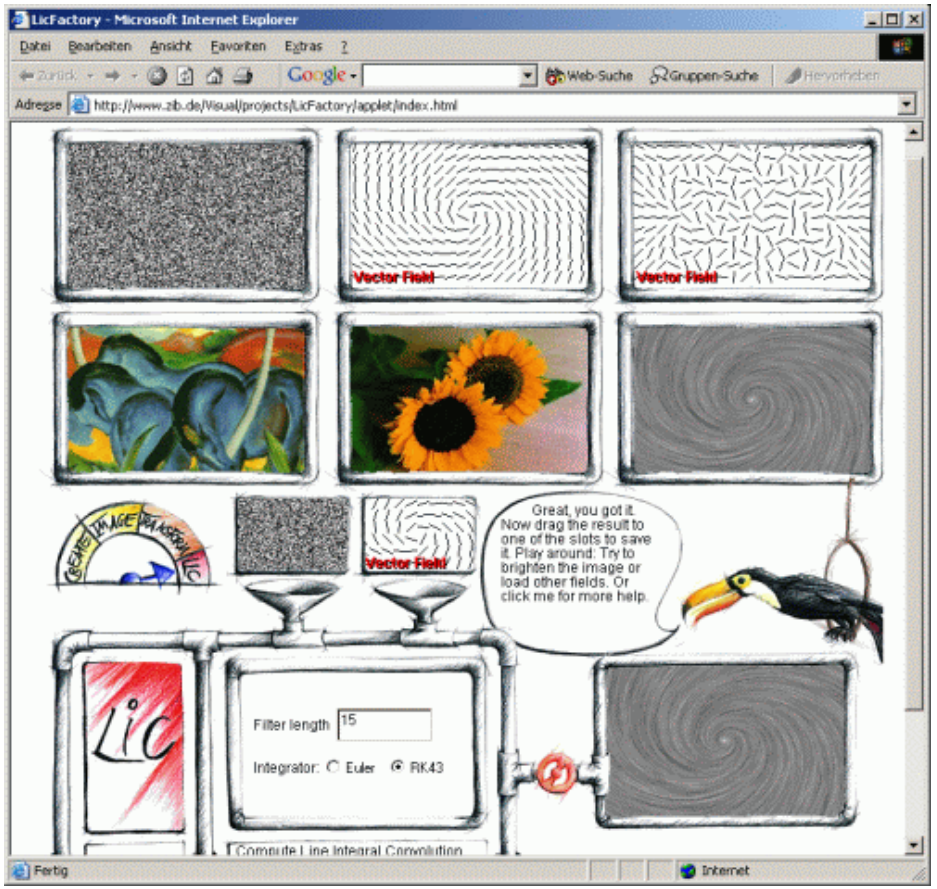
  - 2D-Vektorfeld im R- und G-Teil einer RGB-Textur
    - Abbildung von x,y aus [-1..1] auf zwei Texturen mit jeweils R,G aus [0..1]



negative Werte entstehen durch subtraktives Blending

Beispiele, I

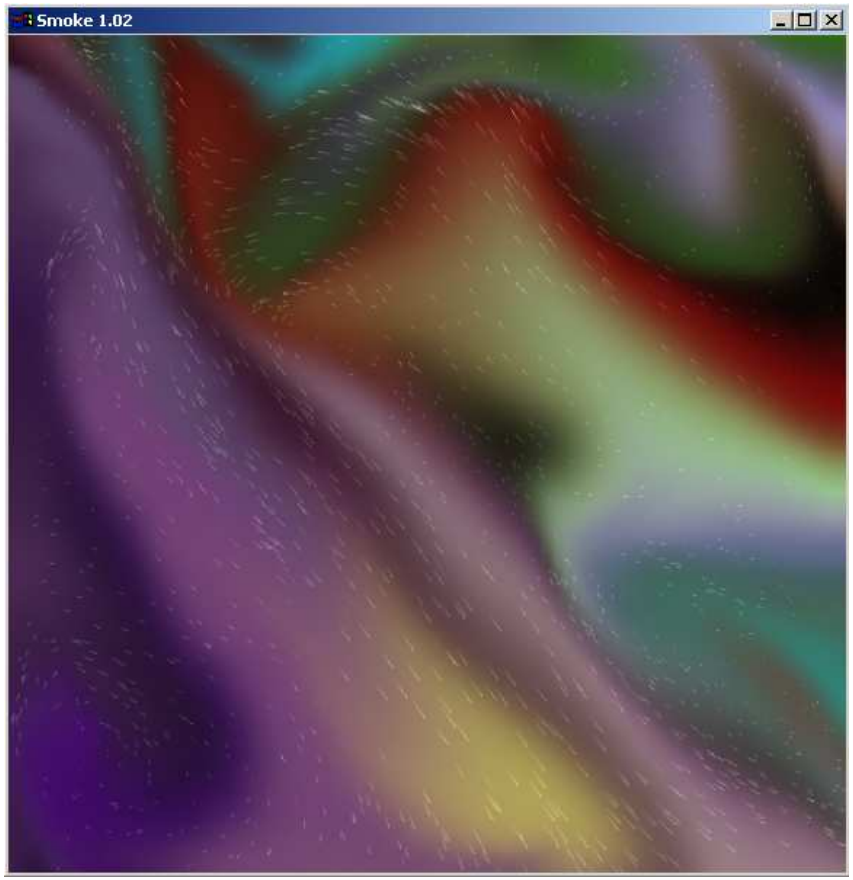
- LIC-Factory
  - Java-Applet
  - interaktives Erforschen von LIC
  - entwickelt am ZIB Berlin



http://www.zib.de/Visual/projects/LicFactory/applet/

Beispiele, II

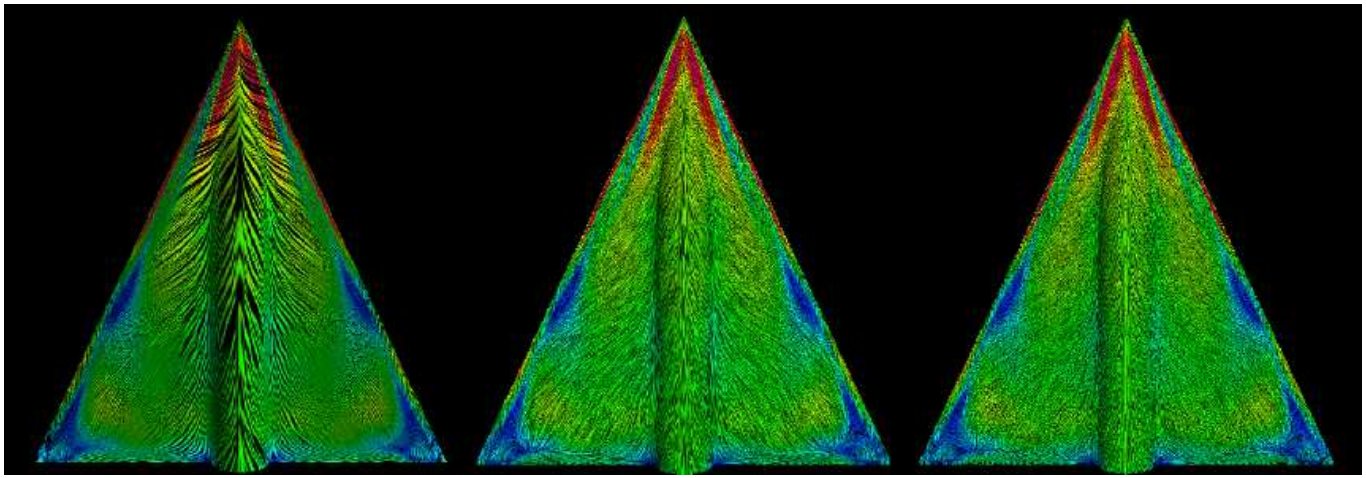
- Smoke PlugIn für WinAmp
  - Freeware von Ryan Geiss (GeissWerks)



http://www.nullsoft.com/free/smoke/

Ausblick, I

- UFLIC
  - unsteady flow line integral convolution
- PLIC
  - combining streamlines and LIC
- ELIC
  - enhanced LIC



Ausblick, II

- verstärkter Einsatz von Hardwarebeschleunigung
  - (Pixel-) Shader
  - fortgeschrittene Textur- und Speichermechanismen
    - P-Buffer
- Level-of-Detail
- 3D LIC
  - ebenfalls hardware-beschleunigt

Quellen

Brian Cabral, Leith (Casey) Leedom:
"Imaging Vector Fields Using Line Integral Convolution", 1995

Heidrich, Westermann, Seidel, Ertl:
"Applications of Pixel Textures in Visualization and Realistic Image Synthesis", 1999

Daniel Weiskopf, Matthias Hopf, Thomas Ertl:
"Hardware-Accelerated Visualization of Time-Varying 2D and 3D Vector Fields by Texture Advection via Programmable Per-Pixel Operations", 2001

Shuang Zhang: http://www.caip.rutgers.edu/~zhang/class/ece562/project3/