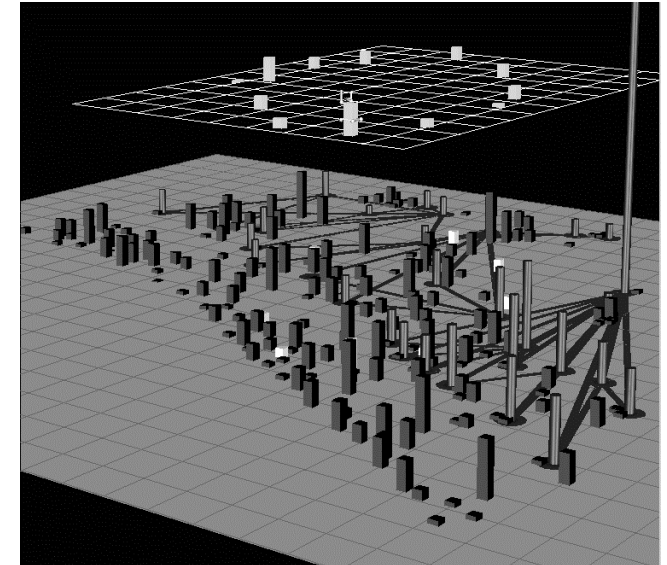
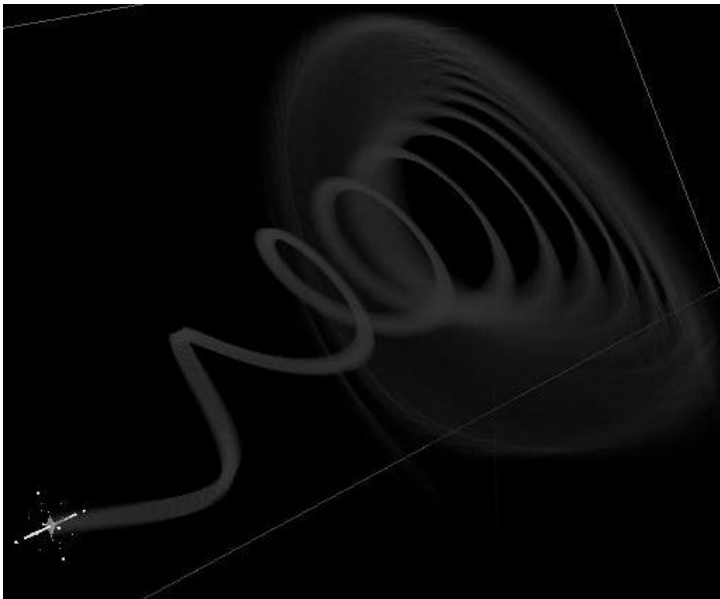


# Visualisierung

## Dr. Helwig Hauser

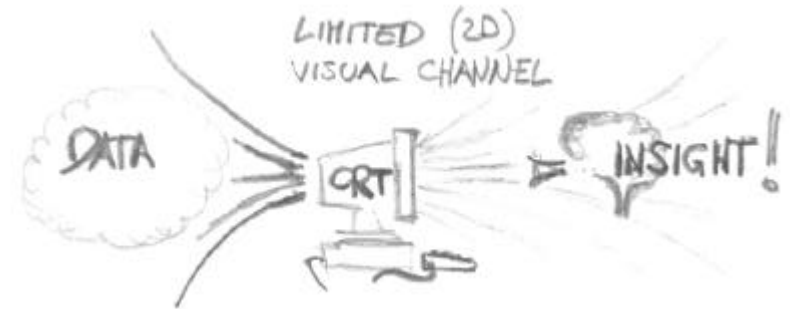
VO 186.004 + LU 186.703,  
Wintersemester 2000/2001,  
Sem186, Fav.-str. 9, 5. Stock,  
<http://www.cg.tuwien.ac.at/courses/Visualisierung/>



# Visualisierung – Definition

**The purpose of computing  
is insight, not numbers**

[R. Hamming, 1962]



## Visualisierung:

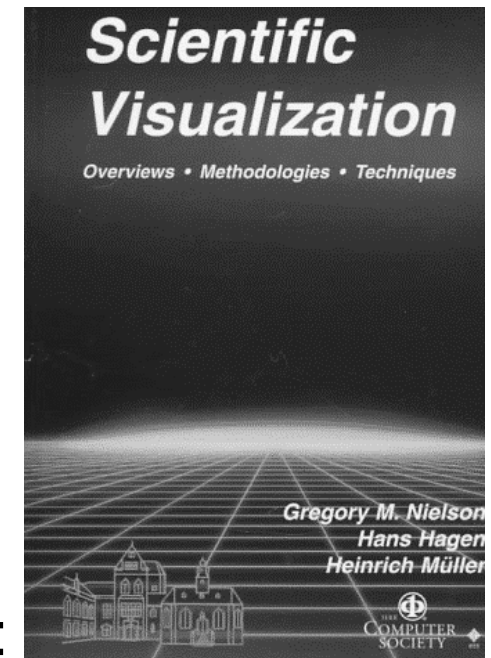
- ◆ Tool, um User Einblick in Daten zu ermögl.
- ◆ to form a mental vision, image, or picture of (something not visible or present to the sight, or of an abstraction); to make visible to the mind or imagination [Oxford Engl. Dict., 1989]
- ◆ Computer Graphik, aber nicht photo-realistisches Rendern

# Visualisierung – Hintergrund

## Hintergrund:



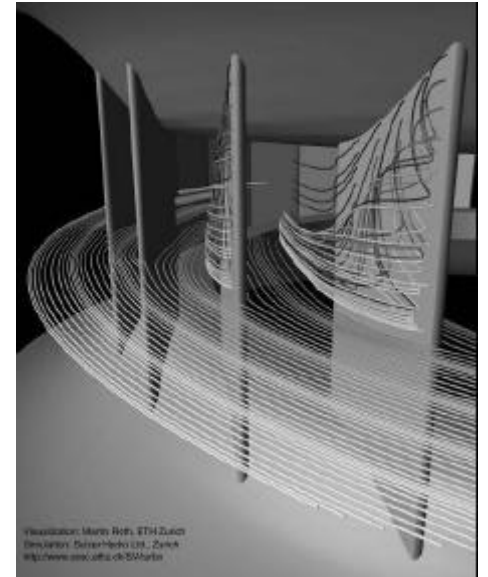
- ◆ Visualisierung = sehr alt
- ◆ Oft intuitiver Schritt: graph. Verdeutlichen
- ◆ Daten immer öfter in sehr großer Menge gegeben  $\Rightarrow$  graphischer Ansatz notwendig
- ◆ Einfache Ansätze bekannt von business graphics (Excel, etc.)
- ◆ Visualisierung = eigene Wissenschaft seit gut 10 Jahren
- ◆ Erste eigene Konfs.: 1990



# Visualisierung – Teilbereiche

## Visualisierung von ...

- ◆ medizinischer Daten ⇒ VolViz!
- ◆ Strömungsdaten ⇒ FlowViz!
- ◆ abstrakten Daten ⇒ InfoViz!
- ◆ GIS-Daten
- ◆ historische Daten (Archäologie)
- ◆ mikroskopischer Daten (Molekularphysik),  
makroskopischer Daten (Astronomie)
- ◆ extrem großer Datenmengen
- usw. ...

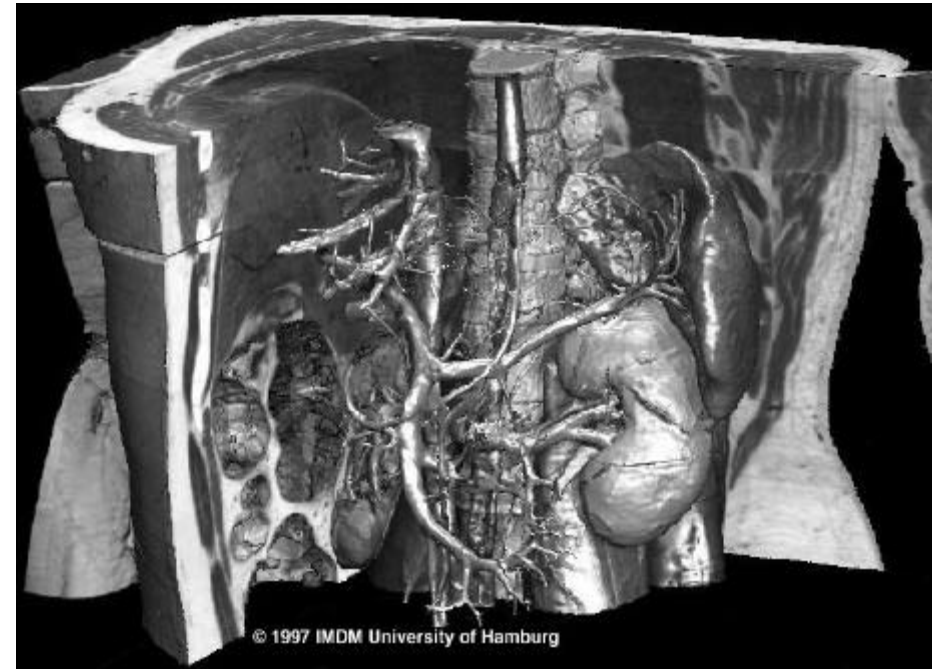


# Visualisierung – Beispiele

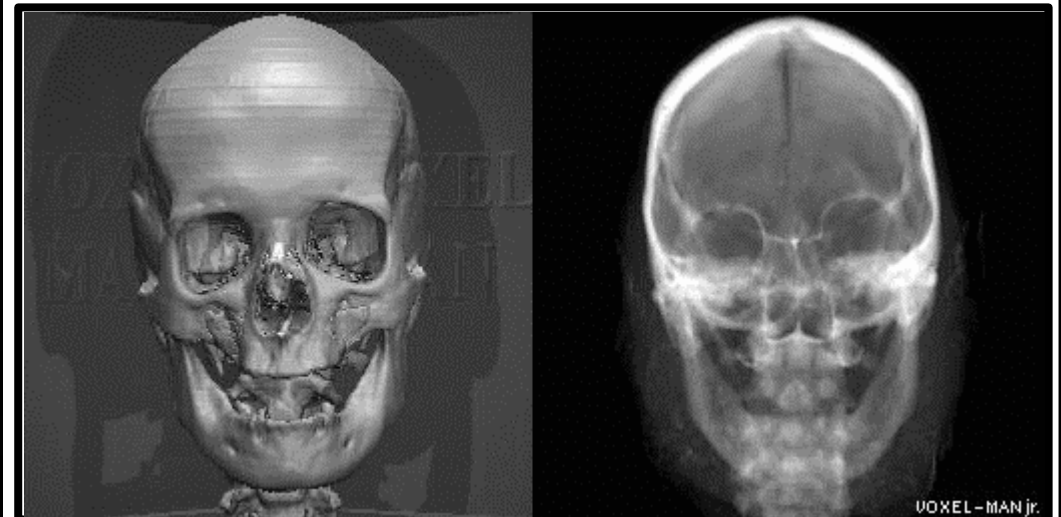
## Medizinische Daten



Helwig Hauser



© 1997 IMDM University of Hamburg

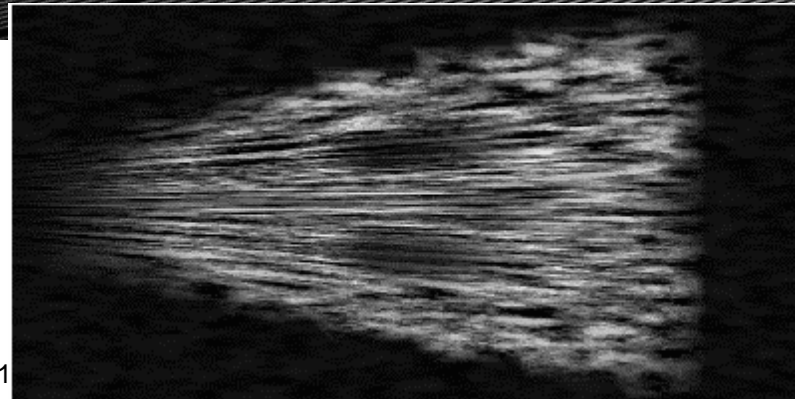
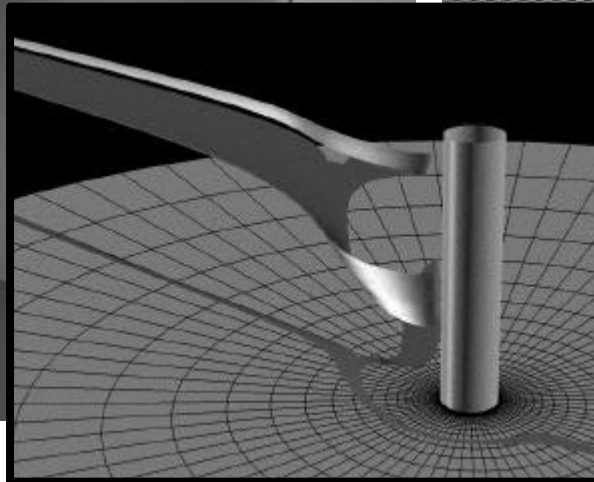
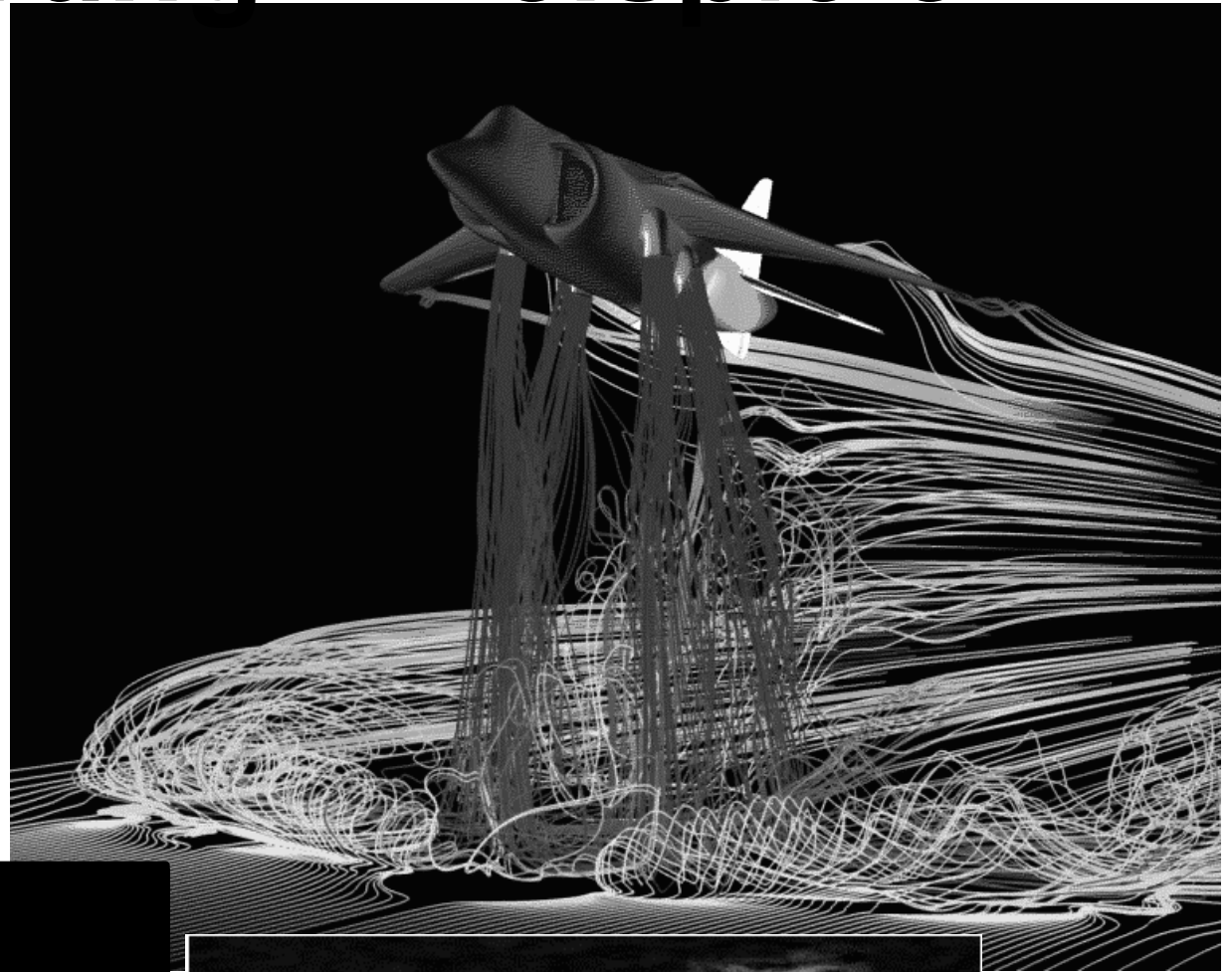
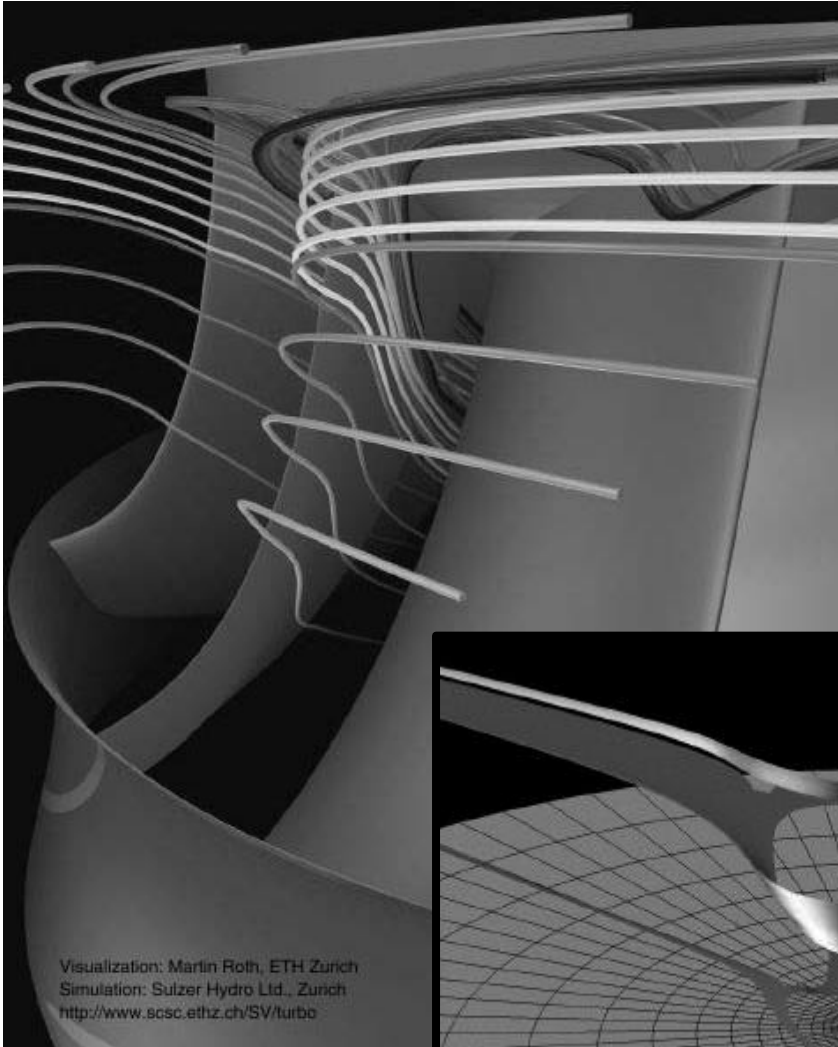


VOXEL-MAN jr.

VisVO 2000/2001, kurzer Auszug

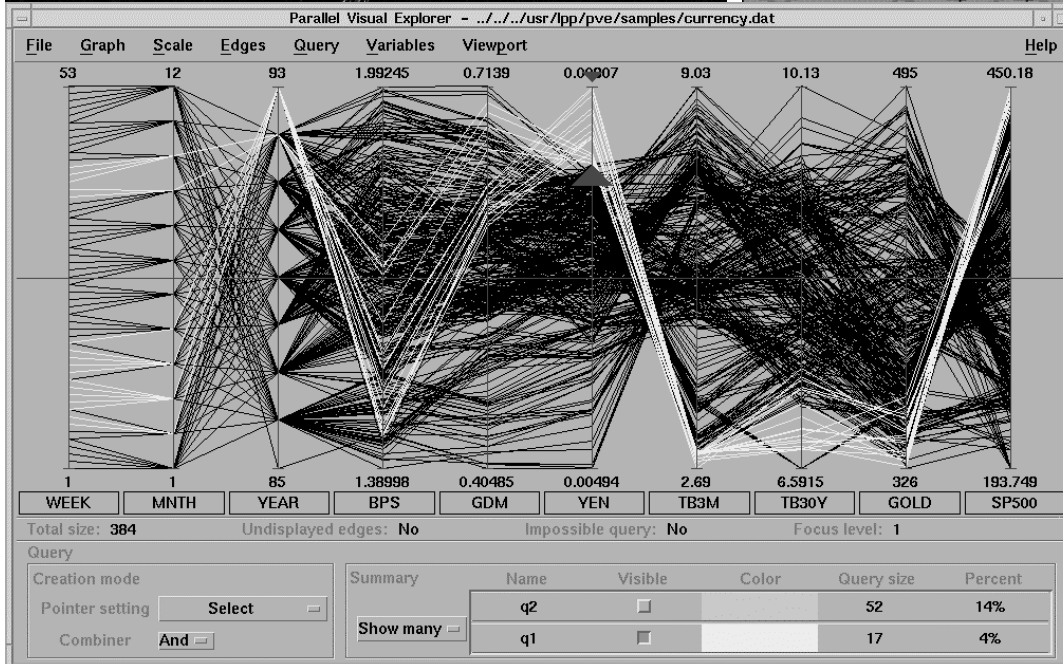
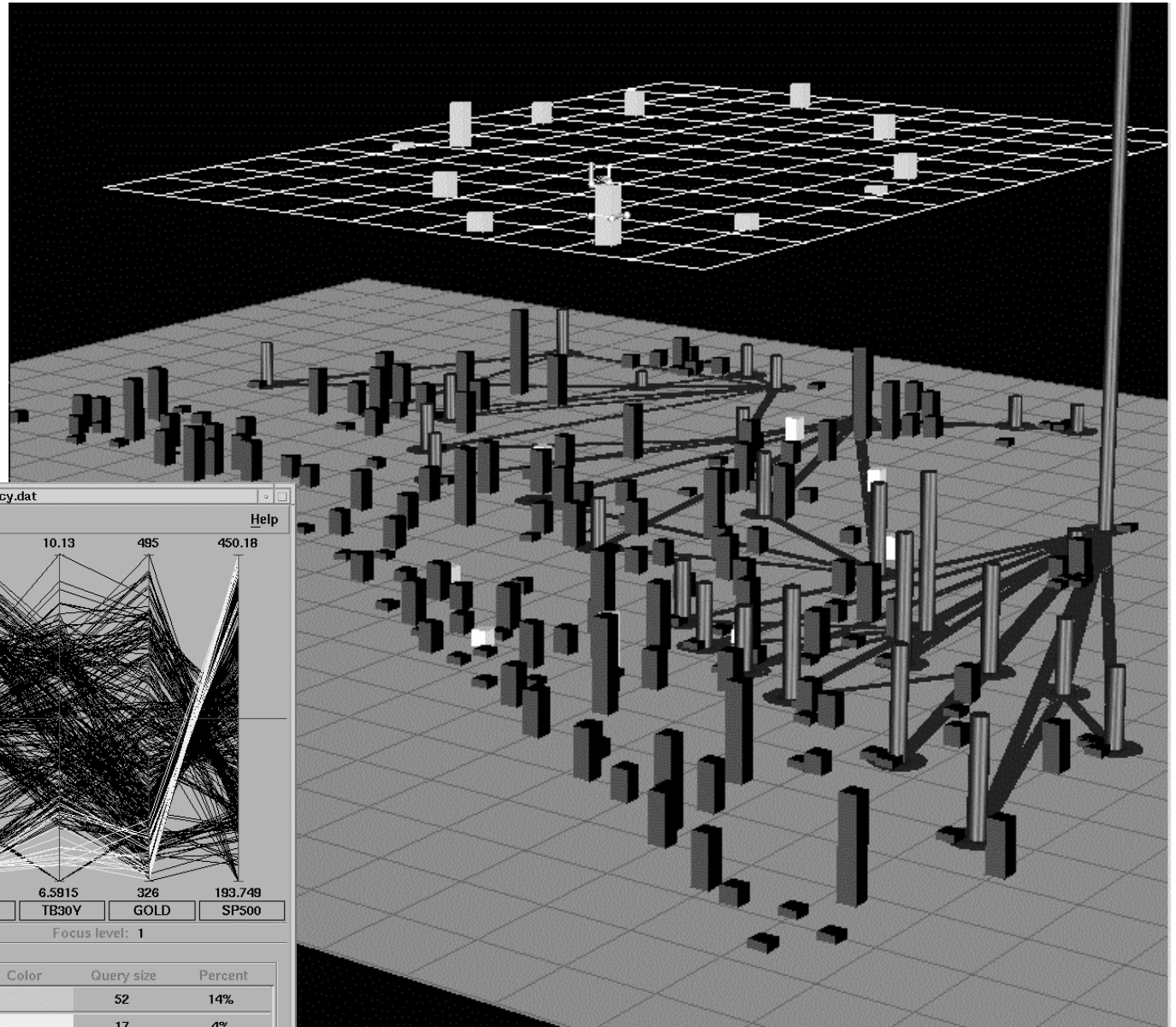
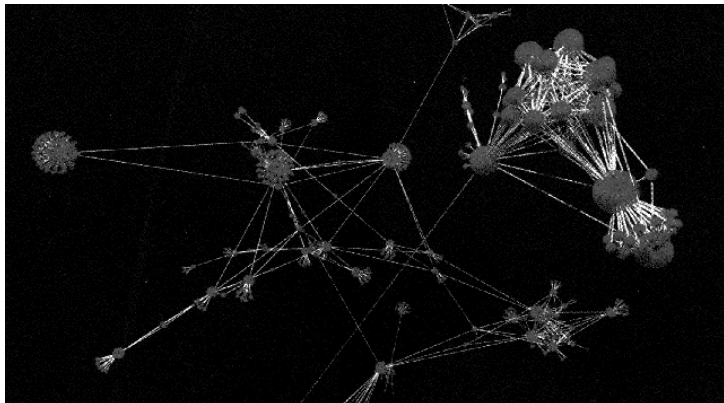
# Visualisierung – Beispiele

## Strömungsdaten





# Visualisierung – Beispiele

## Abstr. Daten



# Visualisierung – drei Ansätze

## Visualisierung, um ...

- ◆ ... zu erforschen
  - ◆ nichts ist bekannt,  
Vis. dient zur neuen Erforschung der Daten
- ◆ ... zu analysieren
  - ◆ es gibt Hypothesen, ?!  
Vis. dient zur Bestätigung bzw. Widerlegung
- ◆ ... zu präsentieren
  - ◆ “alles” über die Daten bekannt, ?!  
Vis. dient zur Kommunikation v. Ergebnissen



# Visualisierung – 3 Schwerpunkte

## Drei Hauptbereiche:

◆ Volume  
Visualization

◆ Flow  
Visualization

Scientific  
Visualization

Inherenter  
Raumbezug

3D

---

◆ Information  
Visualization

nD

meist kein Raumbezug

# **Volumensvisualisierung**

## **Visualisierung von skalaren Daten im 3D**

# Volumensvisualisierung

## Einleitung:

- ◆ VolViz = Visualisierung v. Volumensdaten
  - ◆ Abbildung 3D→2D
  - ◆ Projektion (MIP), Schnitt, vol. Rend., etc.
- ◆ VolData =
  - ◆ 3D×1D Daten
  - ◆ Skalare Daten, 3D Datenraum, raumfüllend
- ◆ User goals:
  - ◆ Einblick in 3D Daten gewinnen
  - ◆ Strukturen von spez. Interesse + Kontext

# Volumensdaten

## Woher kommen die Daten?

- ◆ Medizinische Anwendung
  - ◆ Computertomographie (CT)
  - ◆ Magnetresonanztomographie (MR)
- ◆ Materialprüfung
  - ◆ Industrie-CT
- ◆ Simulation
  - ◆ Finite element methods (FEM)
  - ◆ Computational fluid dynamics (CFD)
- ◆ etc.

# 3D Datenraum

## Wie sind Volumensdaten organisiert?

- ◆ Kartesisches bzw. reguläres Gitter:
  - ◆ CT/MR: oft  $dx=dy<dz$ ,  
z.B. 35 Schichten (z) á 256<sup>2</sup> Werten (Pixel)
  - ◆ Data enhancement: iso-stack-Berechnung =  
Interpolation von zusätzl. Schichten, sodaß  
 $dx=dy=dz$ , 256<sup>3</sup> Voxel
  - ◆ Daten: Zellen (Quader), Ecken: Voxel
- ◆ Curvi-linear grid bzw. unstrukturiert:
  - ◆ Daten als Tetraeder bzw. Hexaeder org.
  - ◆ Oft: Umrechnung auf Tetraeder

# VolViz – Herausforderungen

## Challenges:

- ◆ rendering projection,  
so viel Information und so wenig Pixel!
- ◆ große Datenmengen, z.B.  
 $512 \times 512 \times 512$  Voxel á 16 Bit = 256 Mbytes
- ◆ Geschwindigkeit,  
Interaktion ist sehr wichtig, aber  $>10$  fps!

# Surfaces vs. Volume Rendering

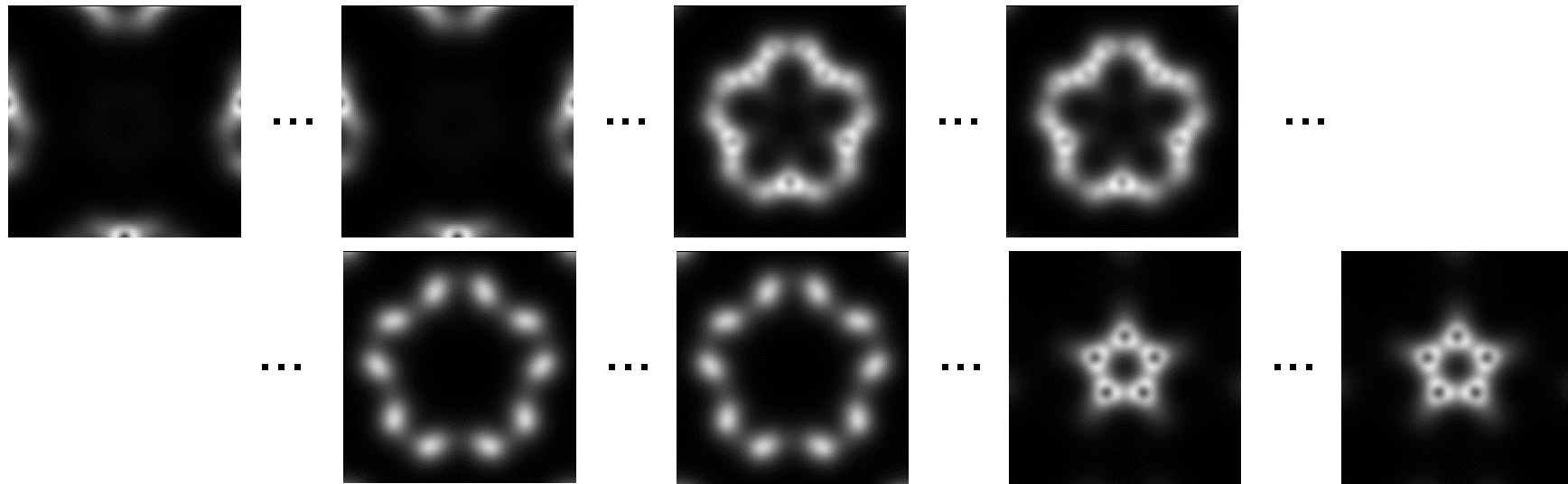
## Surface rendering:

- ◆ indirekte Volumensvisualisierung
- ◆ Zwischenrepräsentation: Iso-fläche, “3D”
- ◆ Pros: Shading→Shape!, HW-rendering

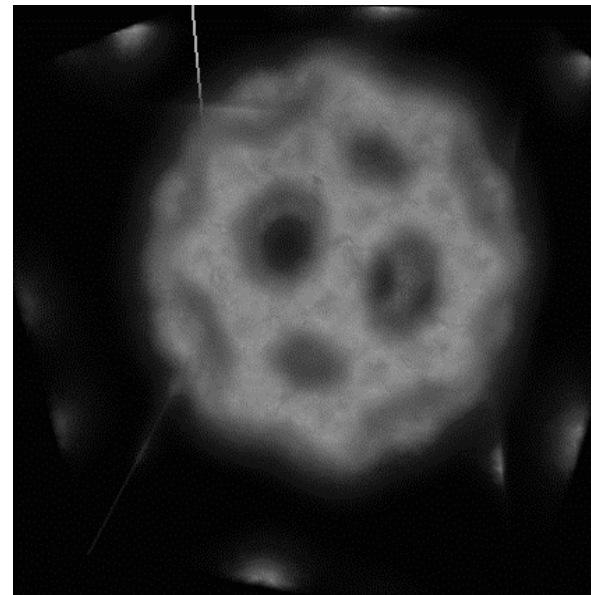
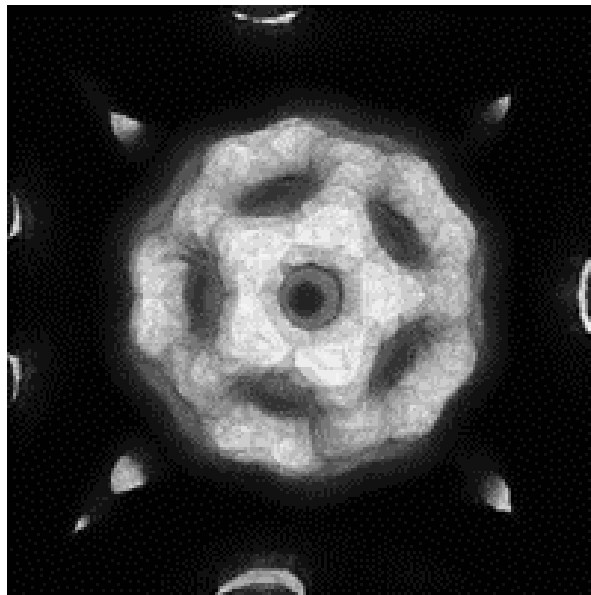
## Volume rendering:

- ◆ direkte Volumensvisualisierung
- ◆ Verwendung von Transferfunktionen
- ◆ Pros: Blick in's Innere, Semi-Transparenz

# Surfaces vs. Volume Rendering



hybrid  
rendering  
=  
surfaces  
+volumes



volume  
rendering



# VolViz-Techniken – Überblick

## Einfache Methoden:

- ◆ slicing, MPR (multi-planar reco.)

## Direkte Volumensvisualisierung:

- ◆ ray casting
- ◆ shear-warp factorization
- ◆ splatting
- ◆ 3D-texture mapping

## Surface-fitting methods:

- ◆ marching cubes (tetrahedra)

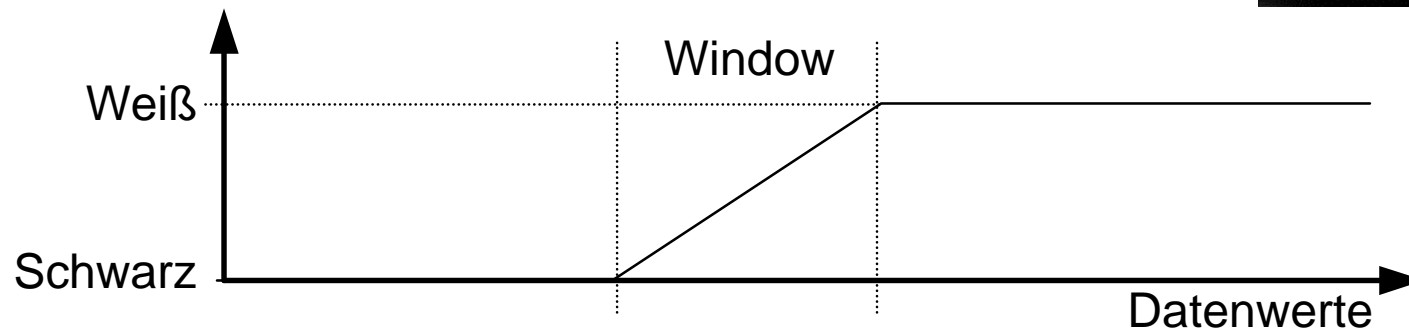
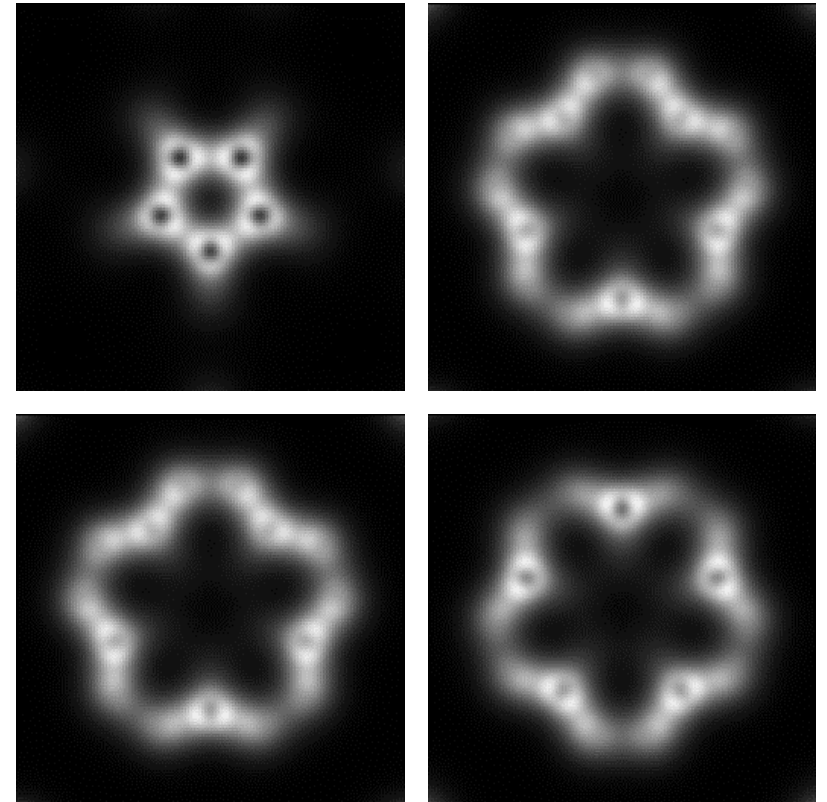
# Einfache Methoden

**Slicing, etc.**

# Slicing

## Slicing:

- ◆ Axen-parallele Schnitte
- ◆ Reguläre Gitter: einfach
- ◆ ohne Transferfunkt.  
keine Farbe
- ◆ Windowing:  
Kontrast einstellen



# Slicing

## Nicht so einfach:

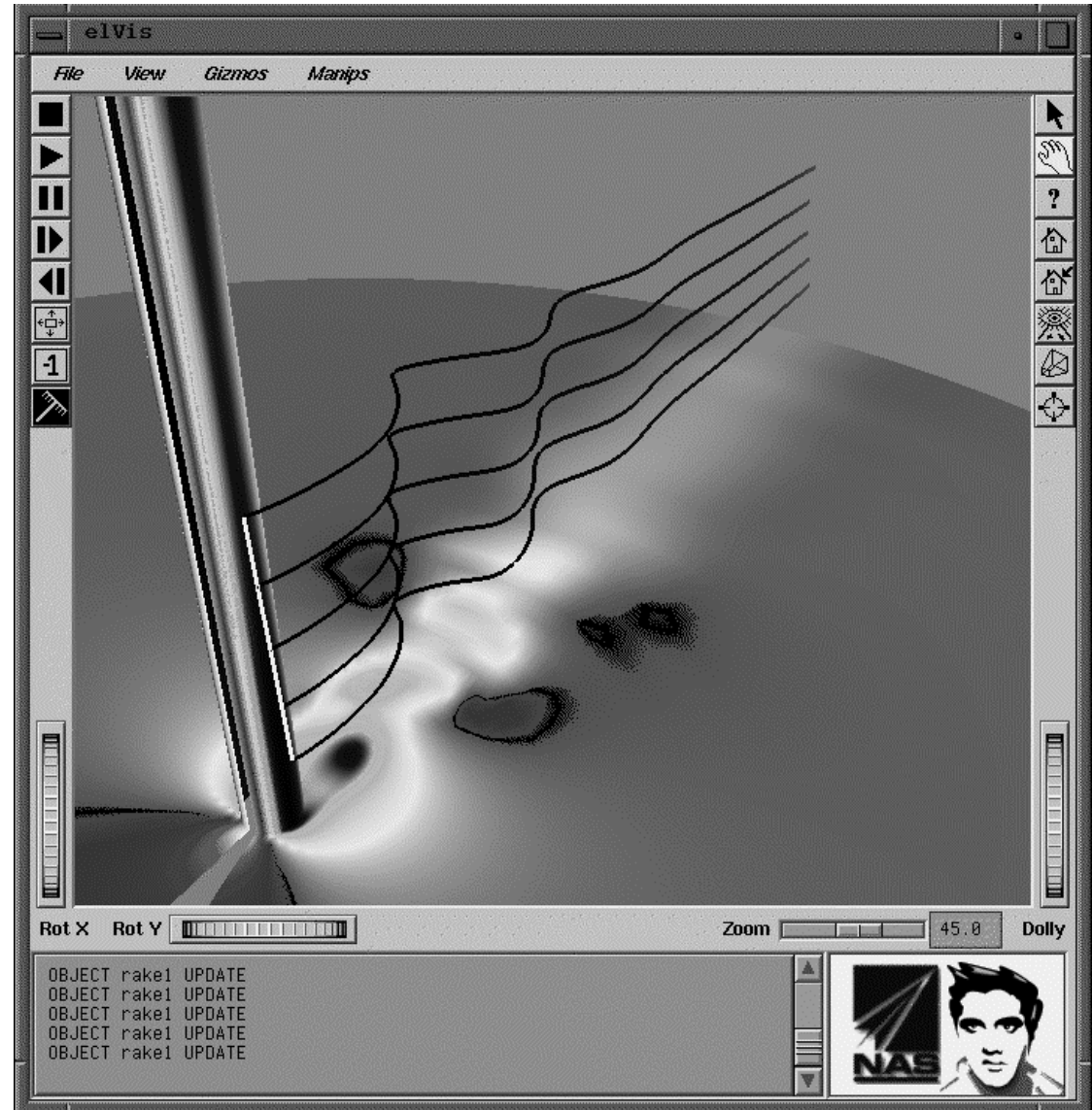
- ◆ Slicing durch allgem. Gitter
- ◆ Interpolation notwendig

## Slicing:

- ◆ gut kombinierbar mit 3D-Vis.

## MPR:

- ◆ versch. Axen, 3D



# **Direkte Volumsvisualisierung, Einführung**

## **Klassifikation – Transferfunktionen,**

# Direkte Volumensvisualisierung

## Überblick:

- ◆ keine Zwischenrepräsentation
- ◆ “real 3D”
- ◆ Integration von so viel Information: schwierig
- ◆ object-order vs. image-order rendering
- ◆ versch. Techniken (ray casting, splatting, shear-warp, texture mapping, etc.)
- ◆ versch. Kombinationsformen (compositing, MIP, first-hit, average, etc.)

# Kombinationsformen

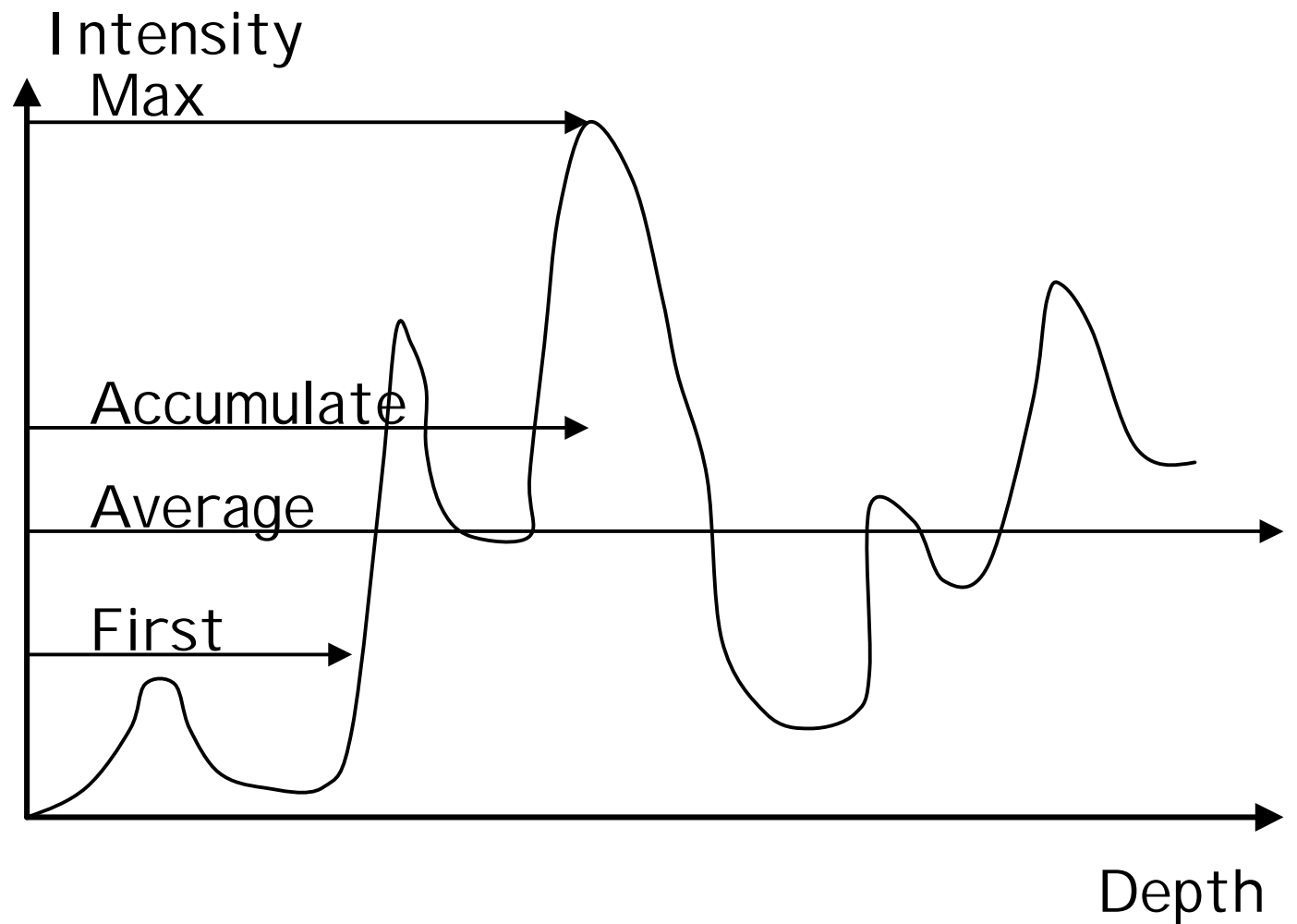
## Überblick:

◆ MIP  $\Rightarrow$

◆ Compositing  $\Rightarrow$

◆ X-Ray  $\Rightarrow$

◆ First hit  $\Rightarrow$



# Klassifikation

## Zuordnung Daten $\Rightarrow$ Semantik:

- ◆ Zuordnung zu Objekten, z.B. Knochen, Haut, Muskel, etc.
- ◆ Verwendung von Datenwerten, Gradienteninformation
- ◆ Ziel: Segmentierung
- ◆ Oft: semi-automatisch bzw. manuell
- ◆ Automatische Approximation: Transferfunktionen

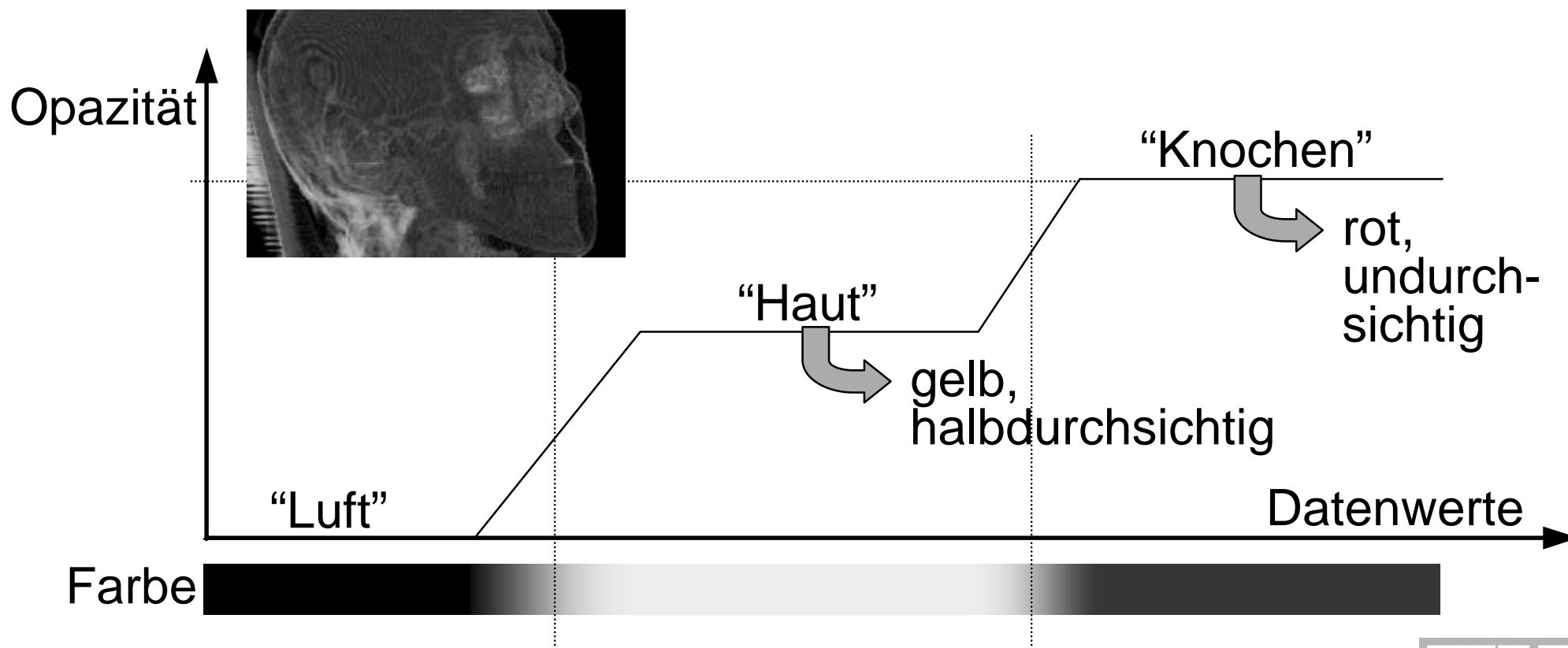
Beispiel



# Transferfunktionen

## Abbildung Daten → "Darstellbares":

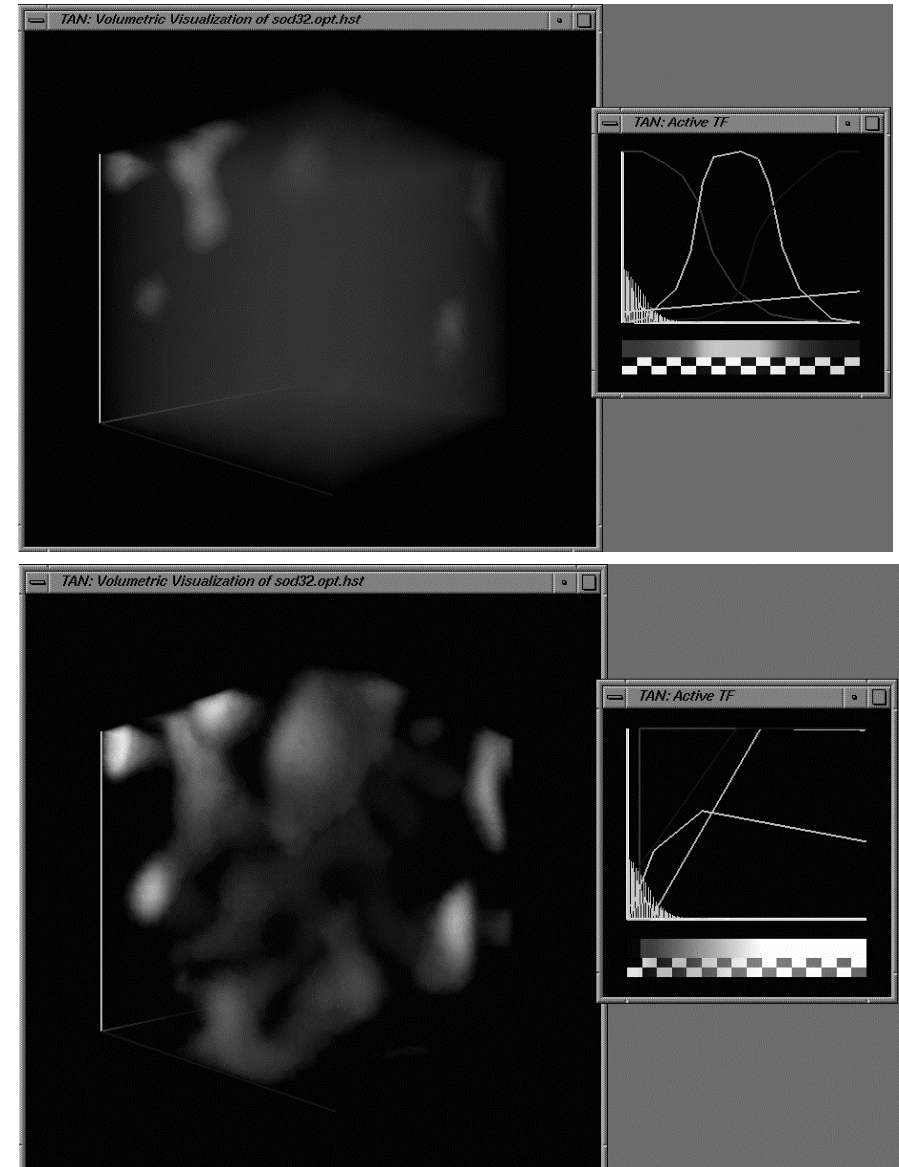
- ◆ 1.) Daten → Farbe
- ◆ 2.) Daten → Opazität (Nichtdurchsichtigkeit)



# Versch. Transferfunktionen

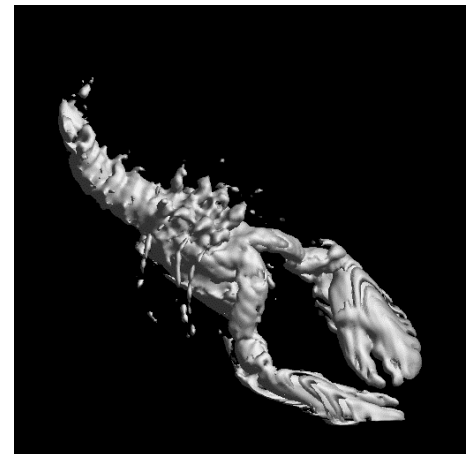
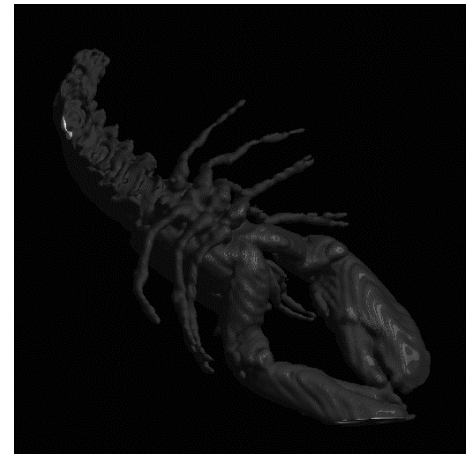
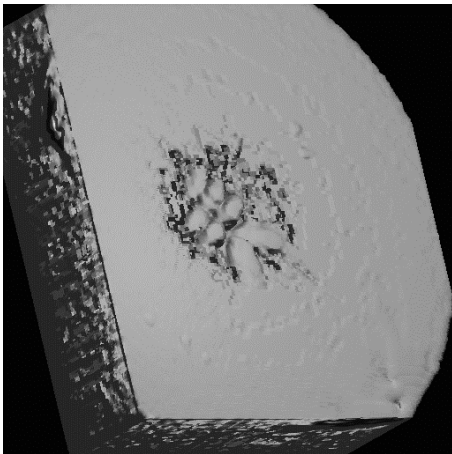
## Ergebnisse:

- ◆ starke Abh. von Transferfunktionen
- ◆ nicht-triviale Einstellung
- ◆ Segmentierung nur bedingt möglich



# Lobster – unterschiedl. TF

Drei Objekte: Medium, Schale, Fleisch



# Fußdaten – ein bzw. zwei TF



# Ray casting / compositing

## Klassische image-order Methode

# Ray Tracing vs. Ray Casting

**Ray Tracing: Methode der Bildgenerierung**

**In Vol. Rend.: nur Primärstrahlen**

**⇒ deswegen Ray Casting**

**Klassische image-order Methode**

**Ray Tracing: Strahl-Objekt Schnitte**

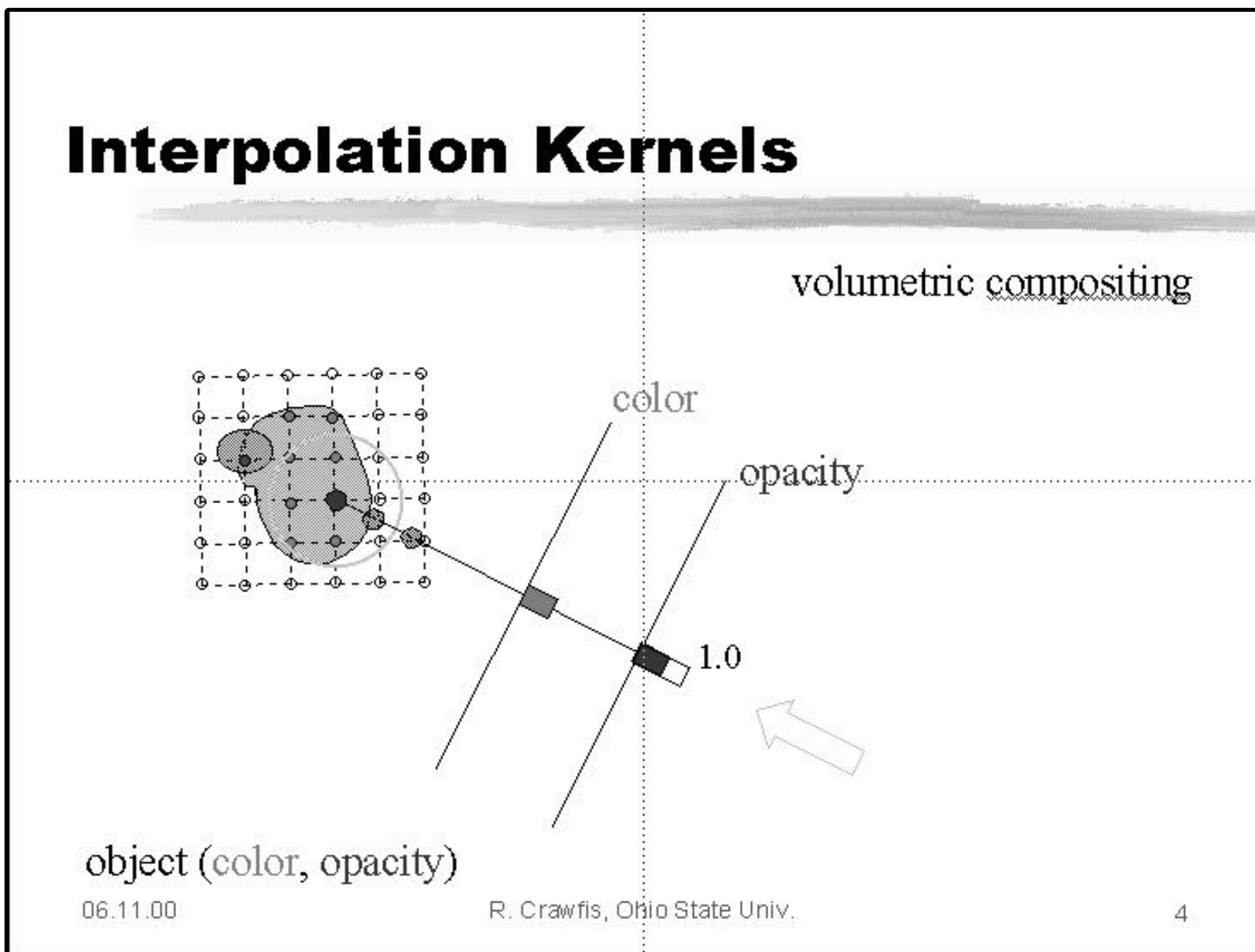
**Ray Casting: keine Objekte, Dichtewerte!**

**Theorie: alle Dichtewerte berücksichtigen!**

**Praxis: Volumen Schritt für Schritt travers.**

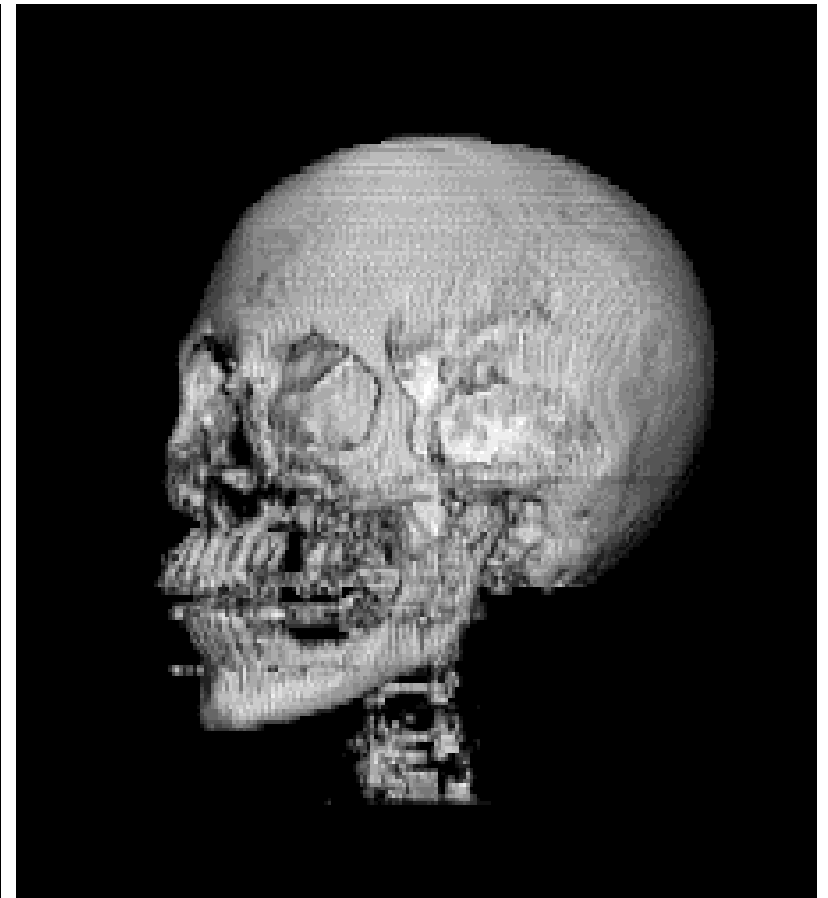
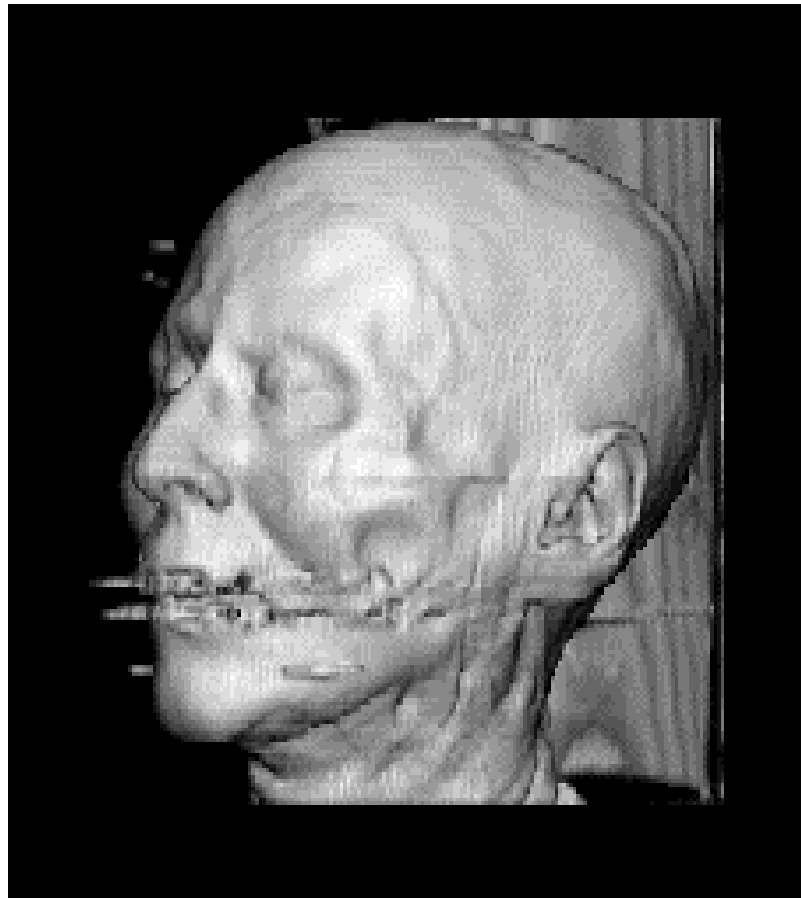
**Interpolation pro Schritt notwendig!**

# Front-to-back Compositing



# Ray Casting – Beispiele

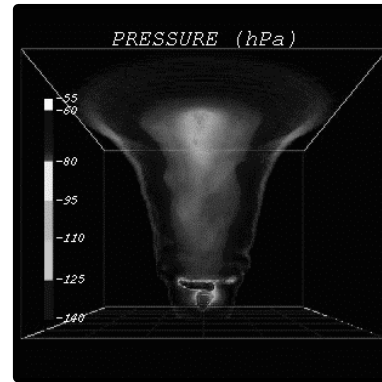
Unterschiedliche TF (quasi-surf. rend.),  
 $256^2 \times 113$  CT-Daten



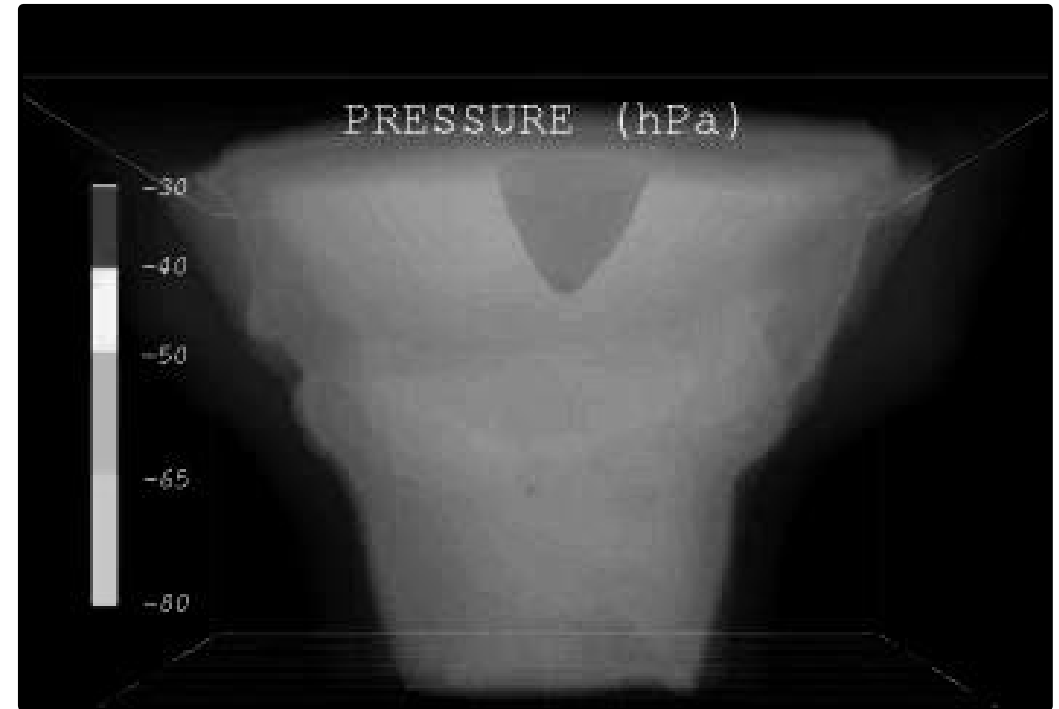
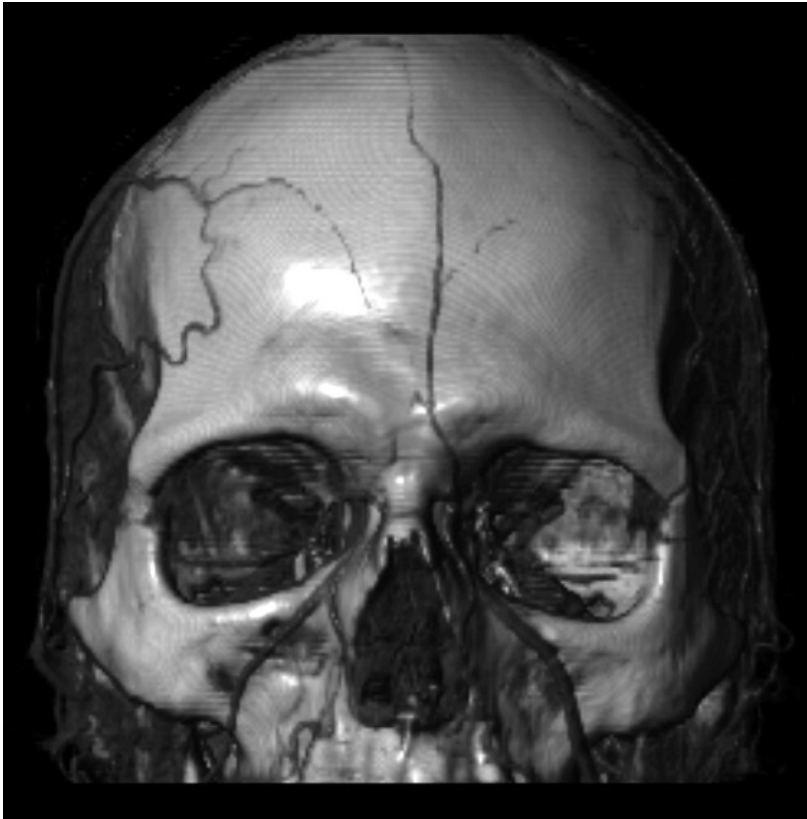


# Ray Casting – weitere Beispiel

Tornado Viz:



Kopf Data:



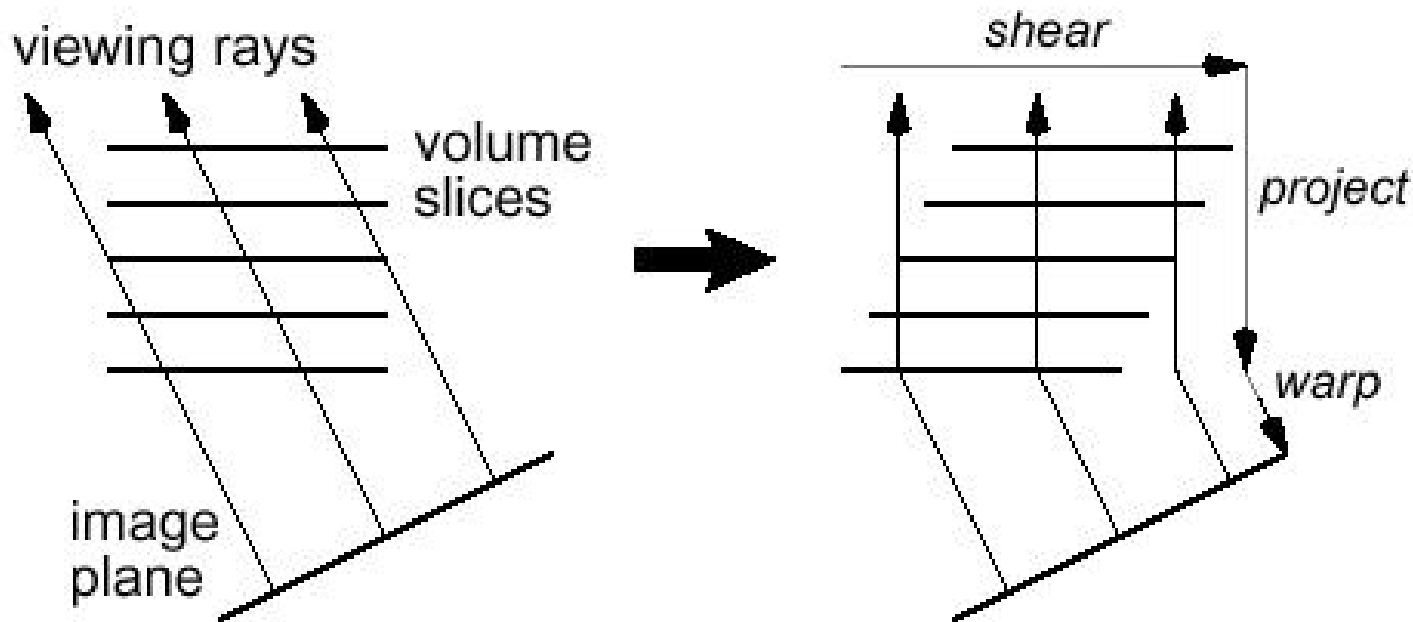
# Shear-warp factorization

## Fast object-order rendering

# Shear-warp Factorization

## Faktorisierung der viewing transformation:

- ◆  $M_{\text{view}} = P \cdot S \cdot M_{\text{warp}}$  (perm, shear, proj., warp)
- ◆ Ziel: parallele Strahlen, voxel:pixel=1:1, einfaches compositing



# Shear, project, warp

## 1.: shear-step

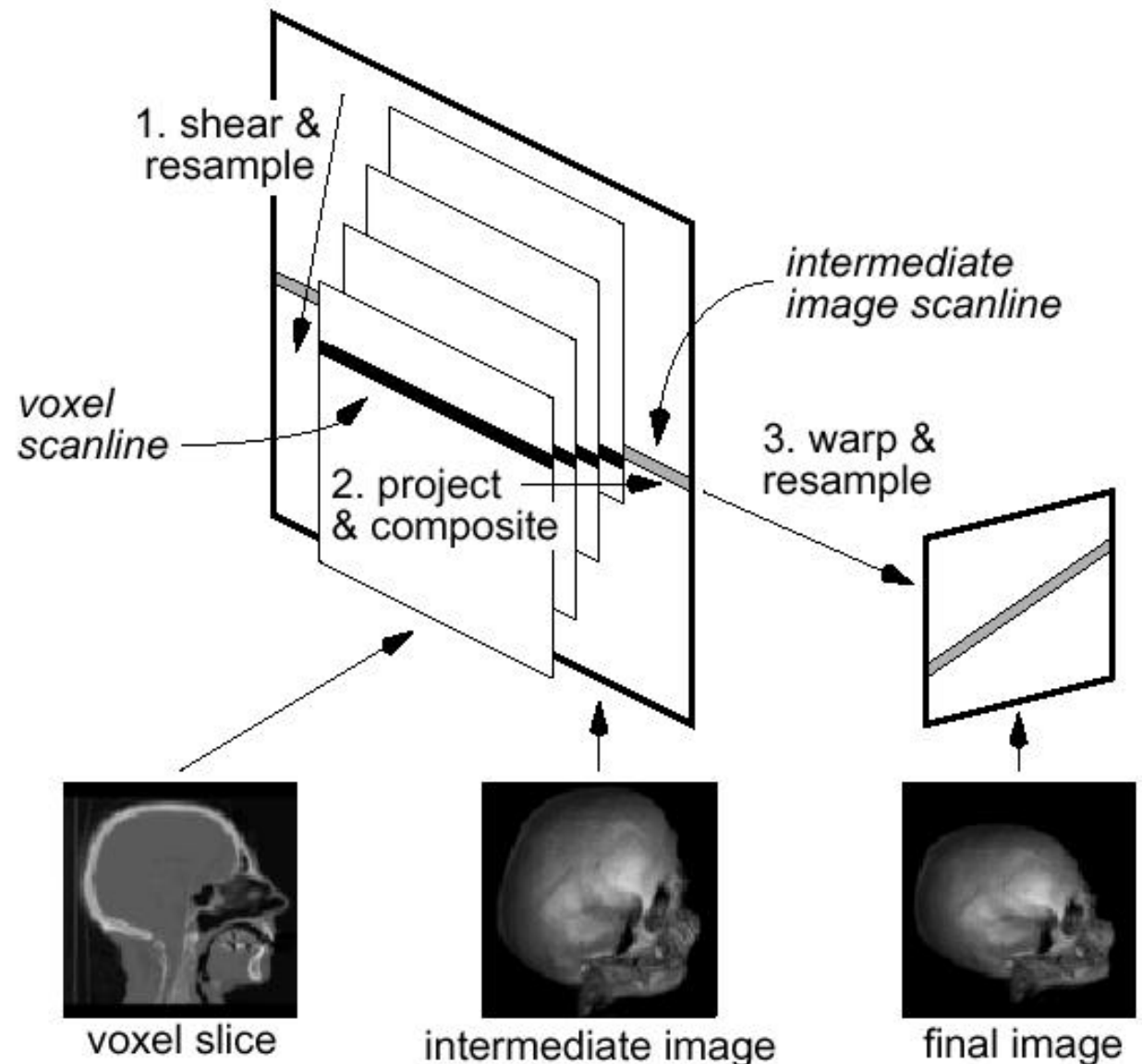
- ◆ 2 shears
- ◆ bi-linear reco.

## 2.: project-step

- ◆ compositing
- ◆ 1 voxel/pixel
- ◆ Erg.: Zw.-Bild

## 3.: warp-step

- ◆ Zw.-Bild auf Bild abbilden



# Shear-Warp – Abschluß

## Pros:

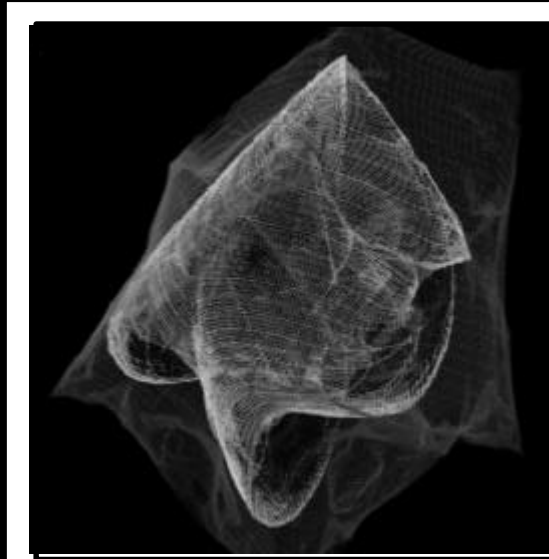
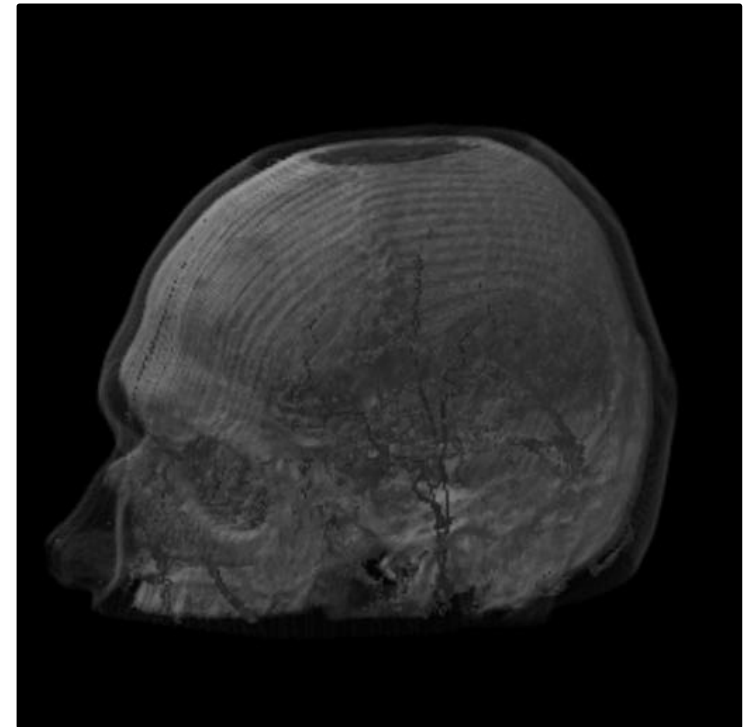
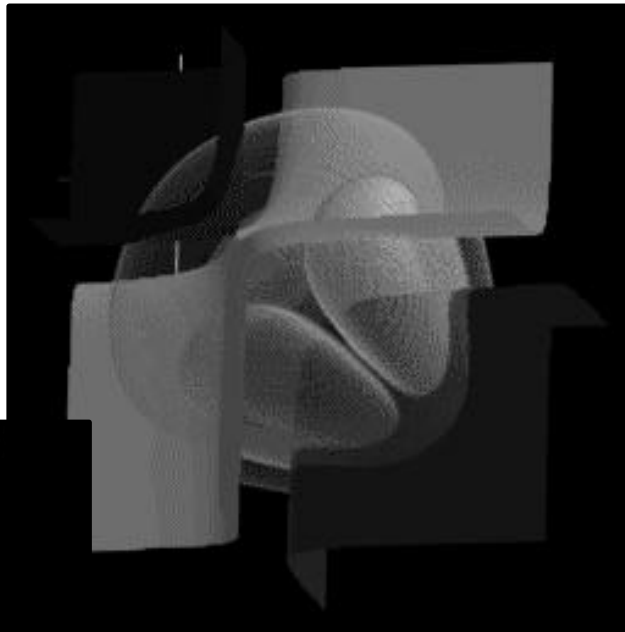
- ◆ schnell!
- ◆ einfach!
- ◆ perspektivische Projektion möglich

## Cons:

- ◆ reco. nur bi-linear (innerhalb Schichten)
- ◆ voxel/pixel(Zw.-Bild!!) = 1  $\Rightarrow$   
Probleme beim Vergrößern!

# Shear-warp – Bilder

Bilder:



# Marching Cubes

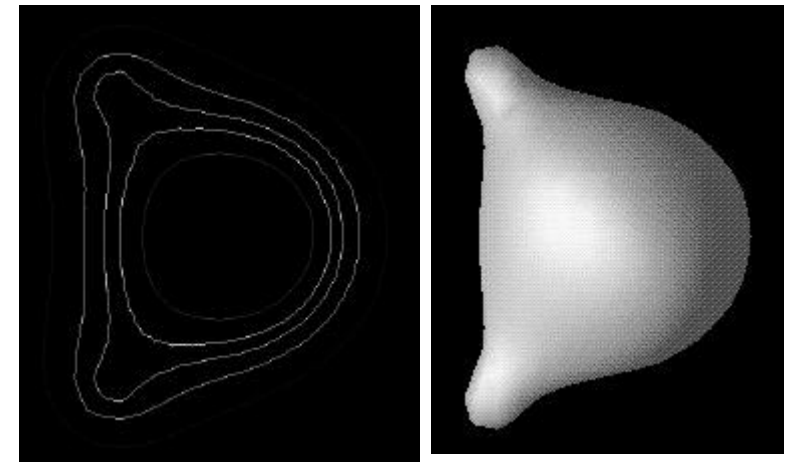
## Iso-Flächen-Darstellung

# Iso-Flächen

## Zwischenrepräsentation

### Aspekte:

- ◆ Voraussetzungen:
  - ◆ aussagekräftiger Iso-Wert, Iso-Wert trennt Materialien
  - ◆ Interesse: in Übergängen
- ◆ sehr selektiv (binäres Auswählen/Weglassen)
- ◆ nützt traditionelle HW
- ◆ shading  $\Rightarrow$  3D-Eindruck!





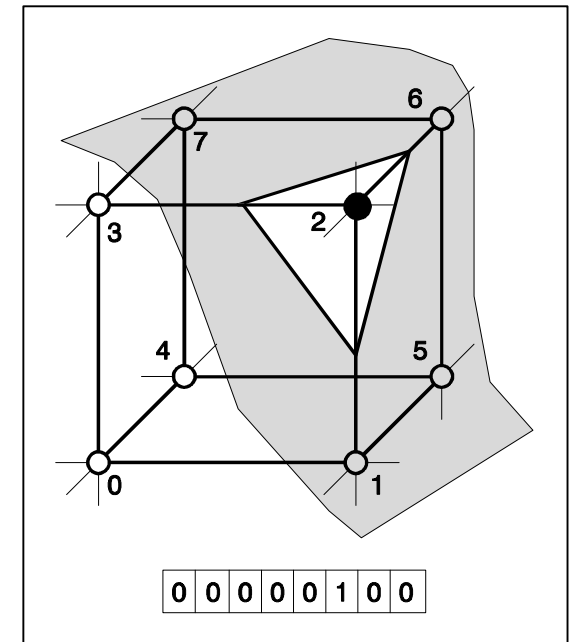
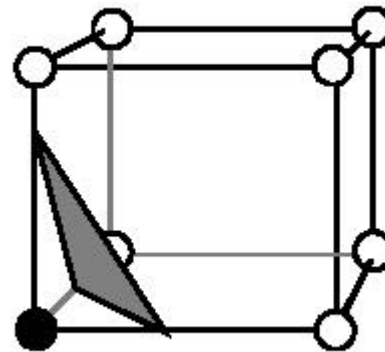
# Annäherung der Iso-Fläche

## Ansatz:

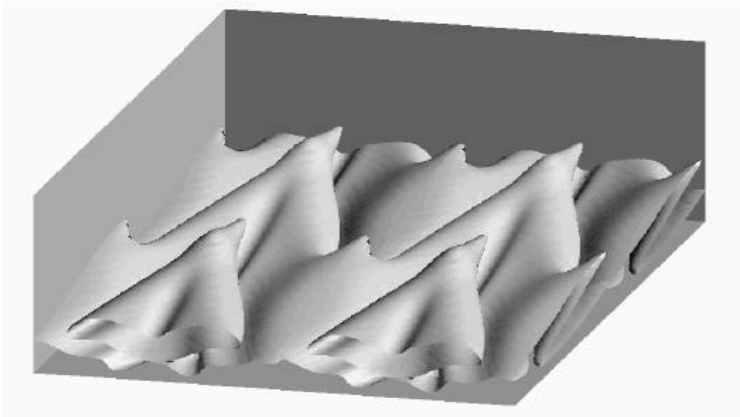
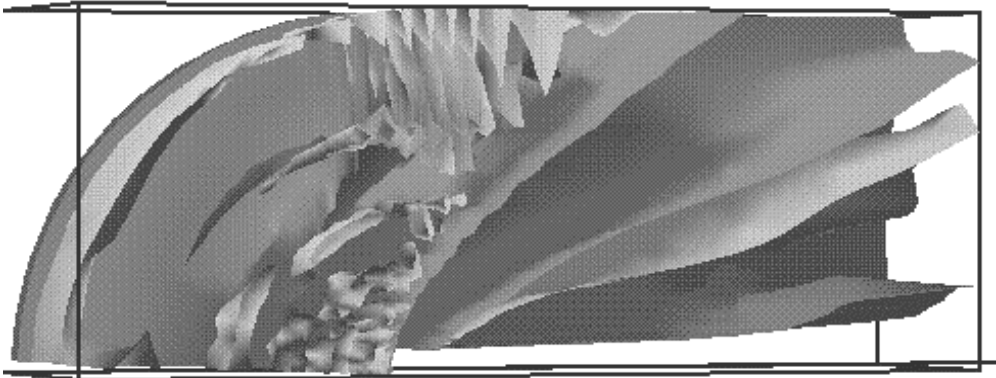
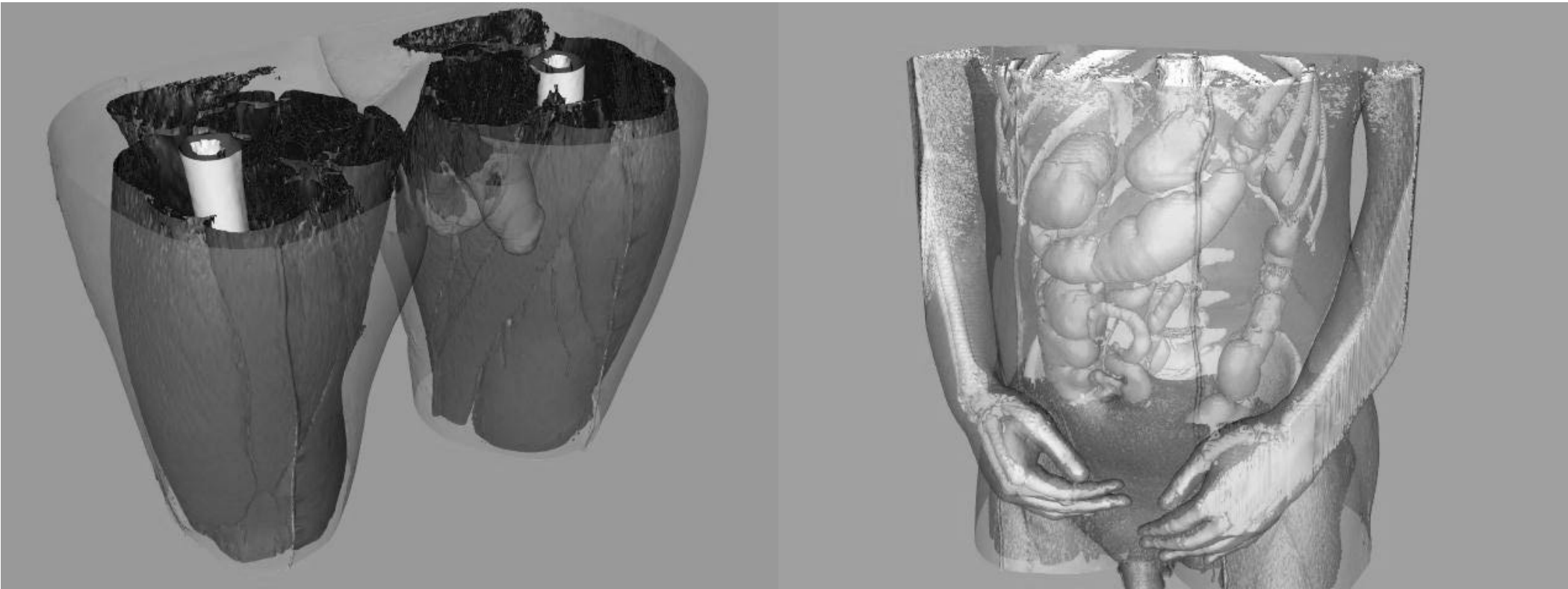
- ◆ Iso-Fläche schneidet Volumen = Menge aller Zellen

## Idee:

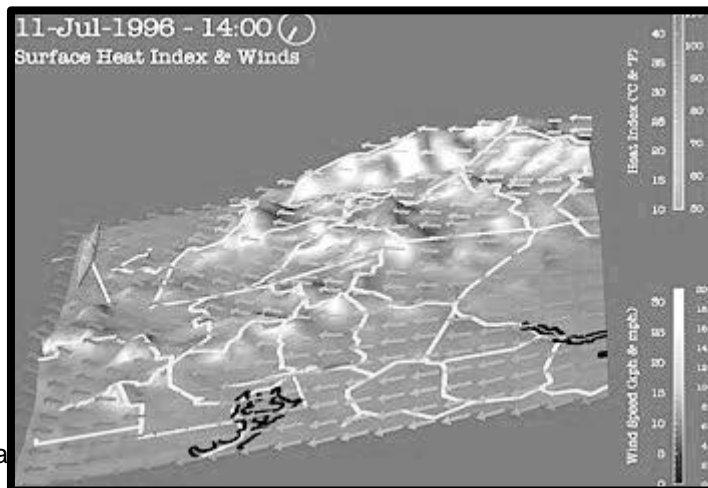
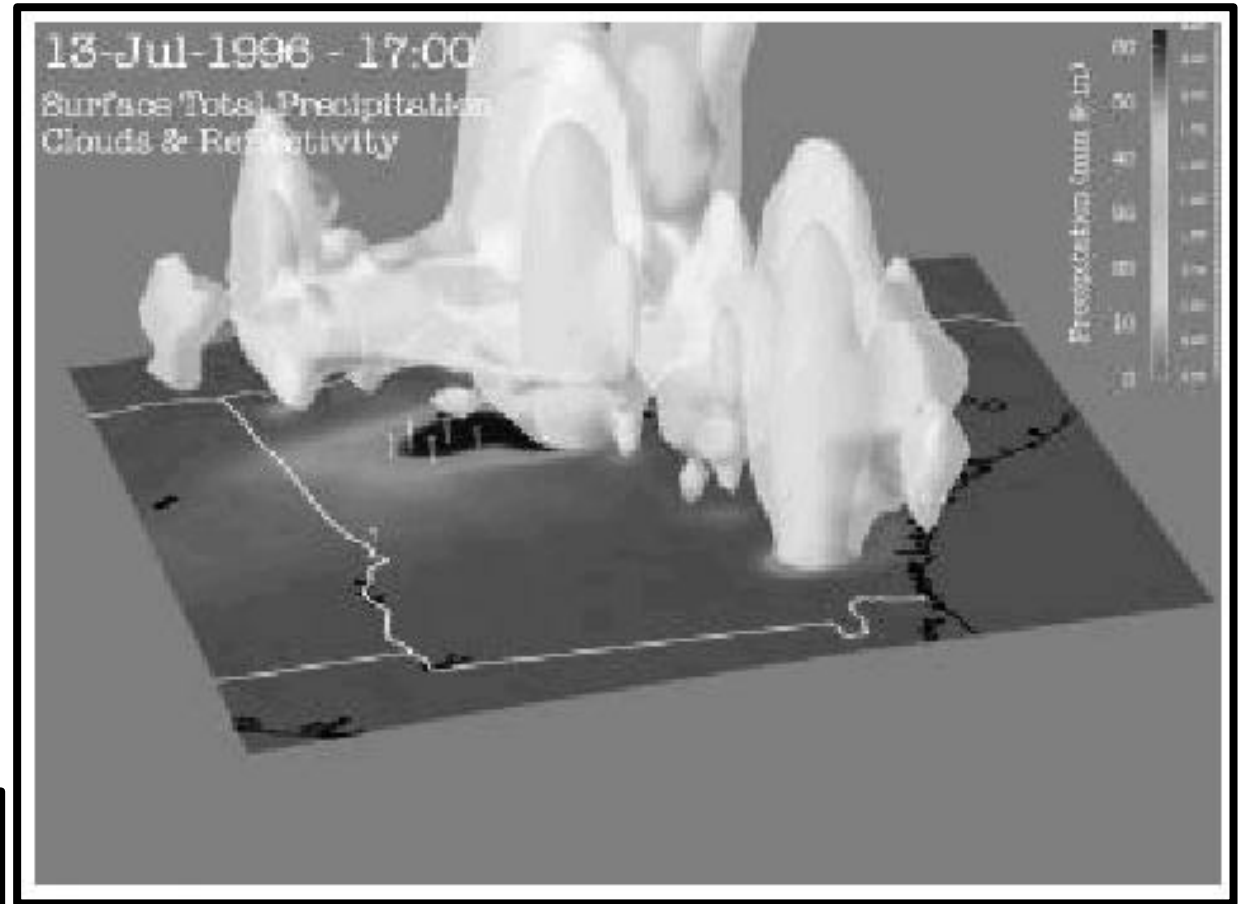
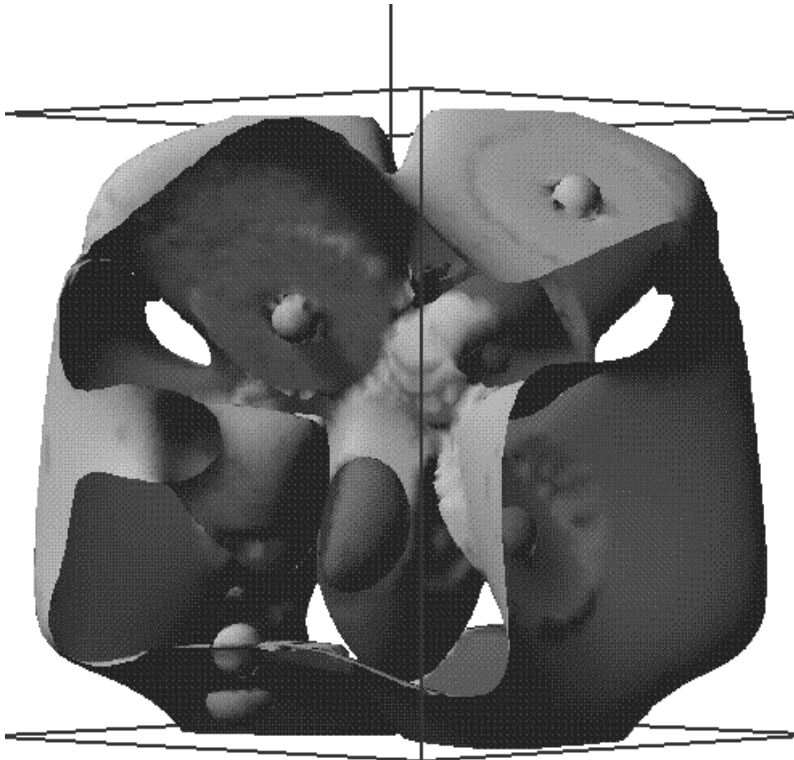
- ◆ Teile der Iso-Fläche pro geschnittener Zelle repräsentieren
- ◆ Möglichst einfach: Verwendung von Dreiecken



# Beispiele



# Mehr Beispiele



# Strömungsvisualisierung

## Einleitung, Überblick

# Strömungsvisualisierung

## Einleitung:

- ◆ FlowViz = Visualisierung von Strömungen
  - ◆ Visualisierung von Änderungsinformation
  - ◆ Normal: mehr als 3 Daten-Dimensionen
  - ◆ Allgemeiner Überblick: noch schwieriger
- ◆ Strömungsdaten:
  - ◆  $nD \times nD$  Daten,  $1D^2 / 2D^2 / nD^2$  (Modelle),  $2D^2 / 3D^2$  (Simulationen, Messungen)
  - ◆ Vektorielle Daten ( $nD$ ) im  $nD$  Datenraum
- ◆ User goals:
  - ◆ Überblick vs. Details (mit Kontext)

# Strömungsdaten

## Woher kommen die Daten:

- ◆ Strömungssimulation:
  - ◆ Flugzeug- / Schiff- / Auto-Design
  - ◆ Wettersimulation (Luft-, Meeresströmungen)
  - ◆ Medizin (Blutströmungen, etc.)
- ◆ Strömungsmessung:
  - ◆ Windkanal, Wasserkanal
  - ◆ Schlieren-, Schatten-Technik
- ◆ Strömungsmodelle:
  - ◆ Differenzialgleichungssysteme (dynamische Systeme)

# Strömungsdatendefinition

## Simulation:

- ◆ flow: set of samples,  
z.B. auf curvi-linear grid gegeben
- ◆ wichtigstes Primitiv: Tetraeder

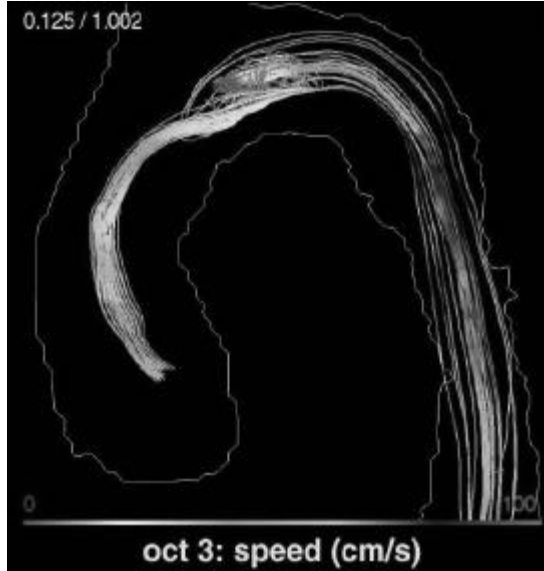
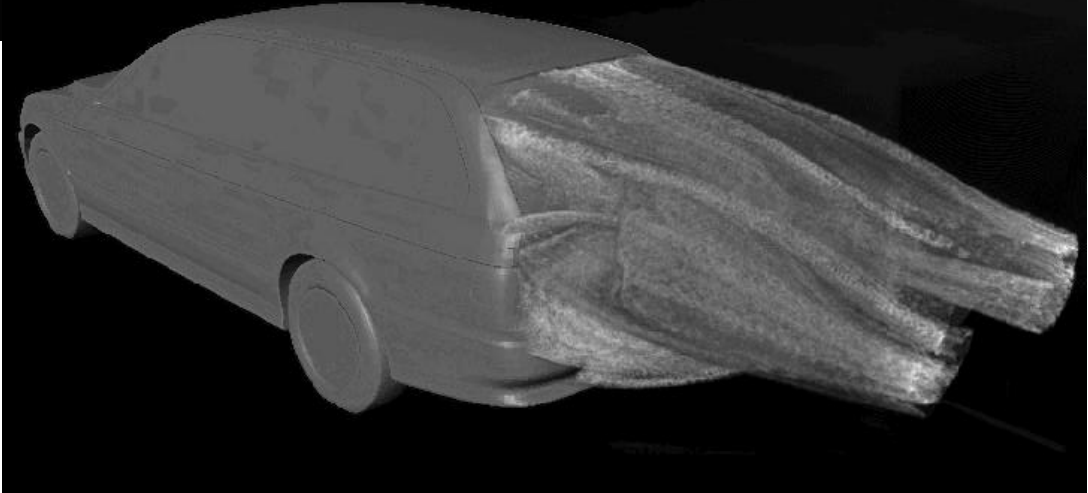
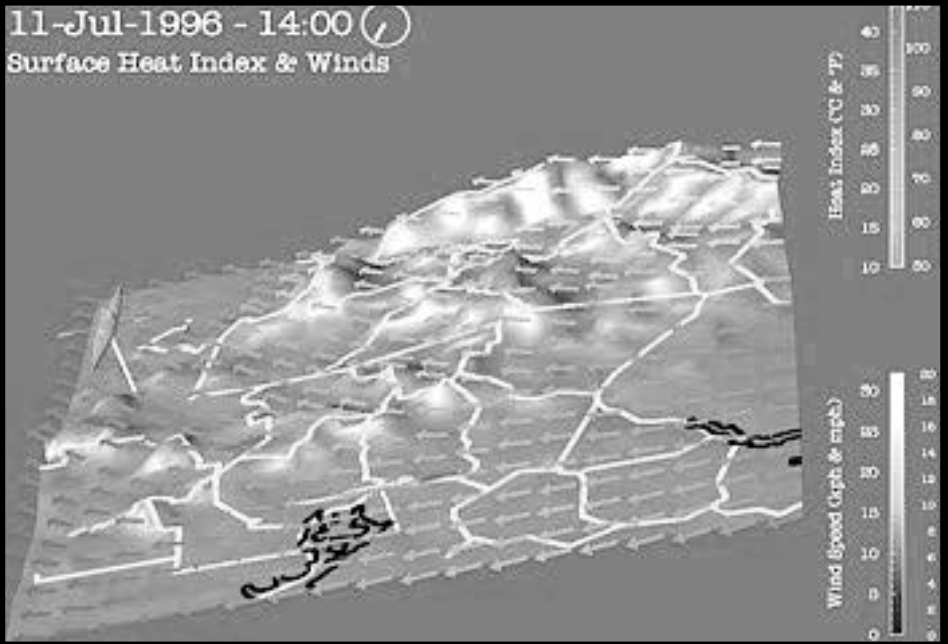
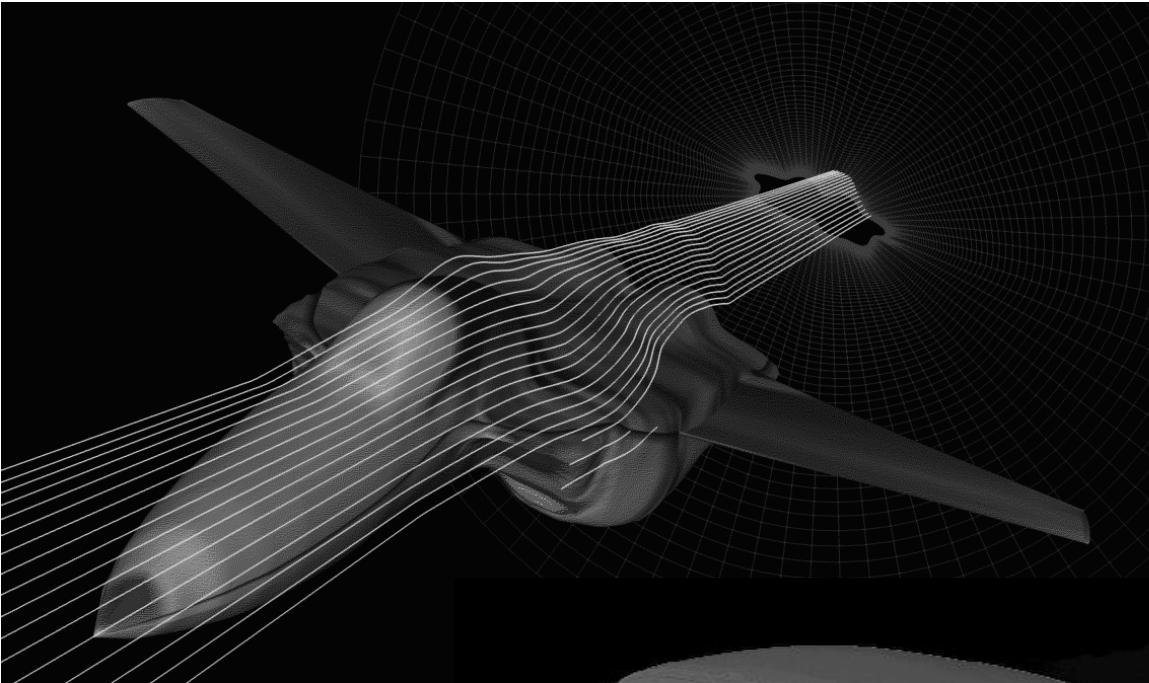
## Messung:

- ◆ flow: Rekonstruktion aus Korelationsdaten,  
oft auf regulären Gittern berechnet

## Modellierung:

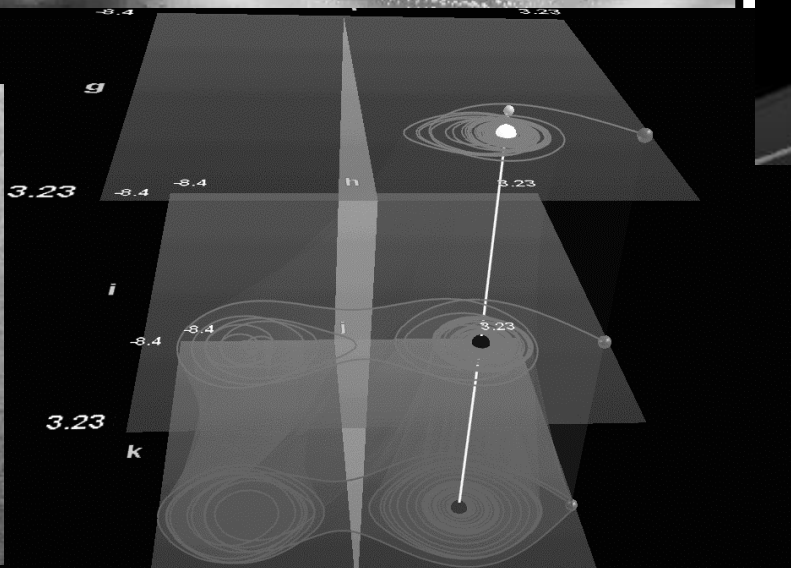
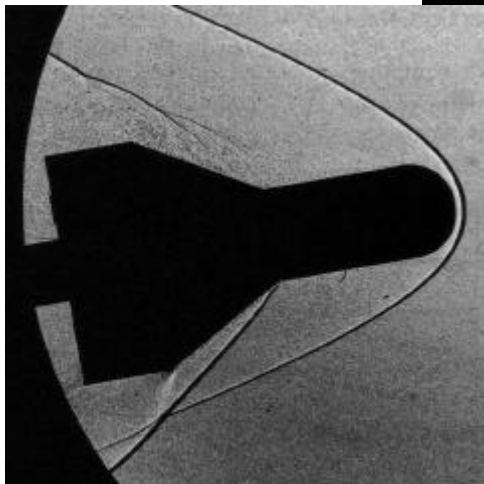
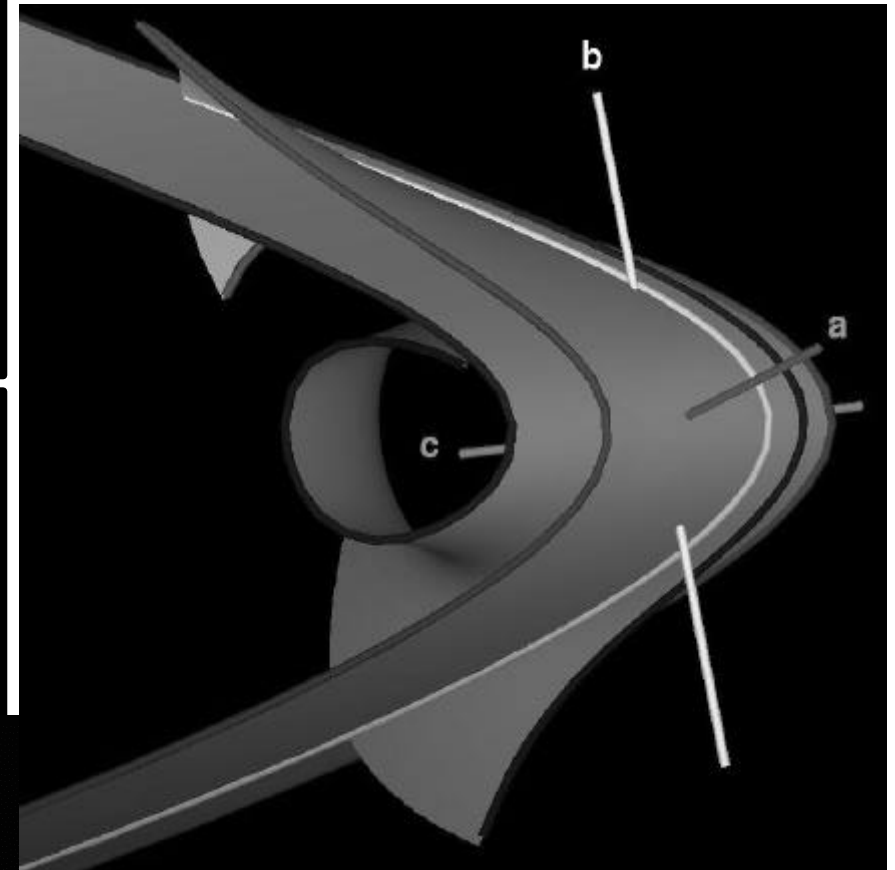
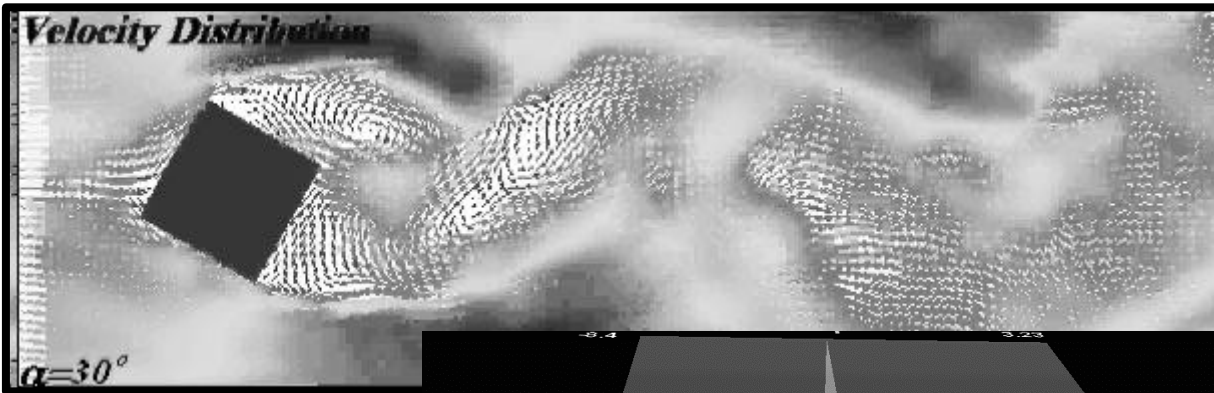
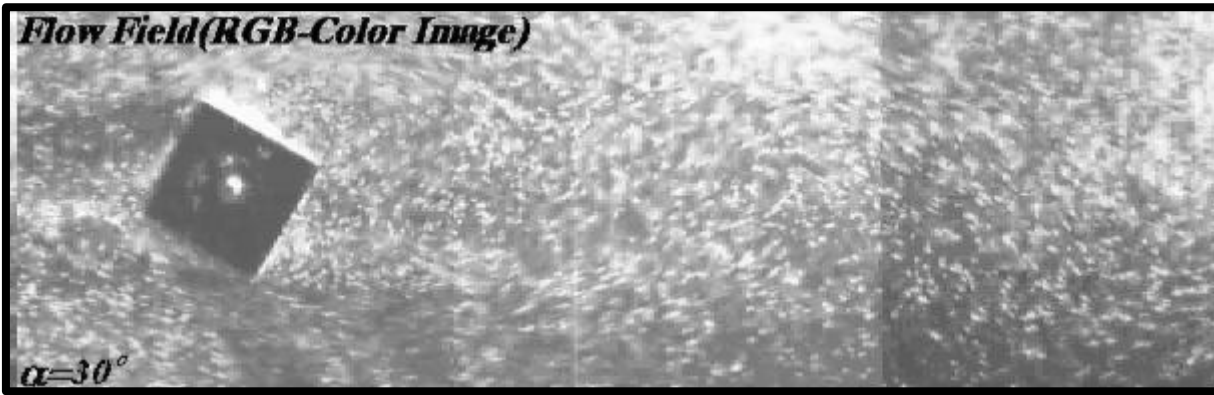
- ◆ flow: analytische Formel,  
“überall” auszuwerten

# Datenursprung – Beispiele 1/2





# Datenursprung – Beispiele 2/2



# Sim. vs. Messung vs. Modell

## Simulation:

- ◆ Raum der Strömung mit Gitter modellieren
- ◆ FEM (Finite Elemente Methode),  
CfD (computational fluid dynamics)

## Messungen:

- ◆ Optische Methoden + Bilderkennung,  
z.B.: PIV (Particle Image Velocimetry)

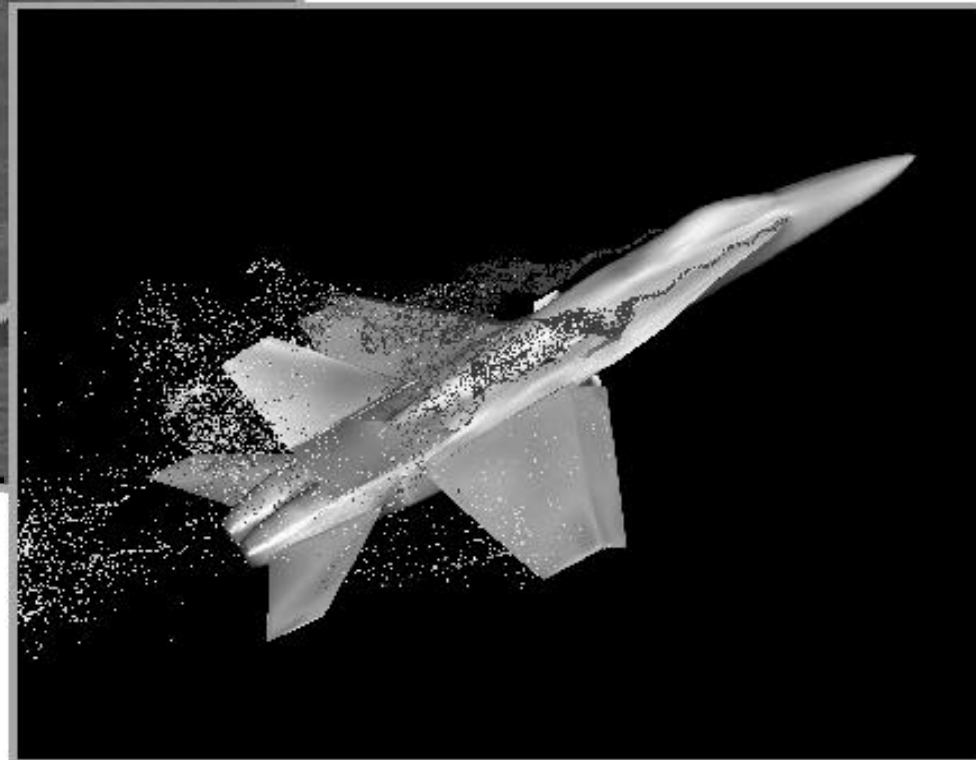
## Modelle:

- ◆ Differenzialgleichungssysteme  $dx/dt$

# Vergleich mit Wirklichkeit



Experiment



Simulation

# 2D vs. Flächen vs. 3D

## 2D-Strömungsvisualisierung

- ◆ 2D×2D-Strömungen
- ◆ Modelle, Schichtströmungen (2D aus 3D)

## Visualisierung von Oberflächenströmungen

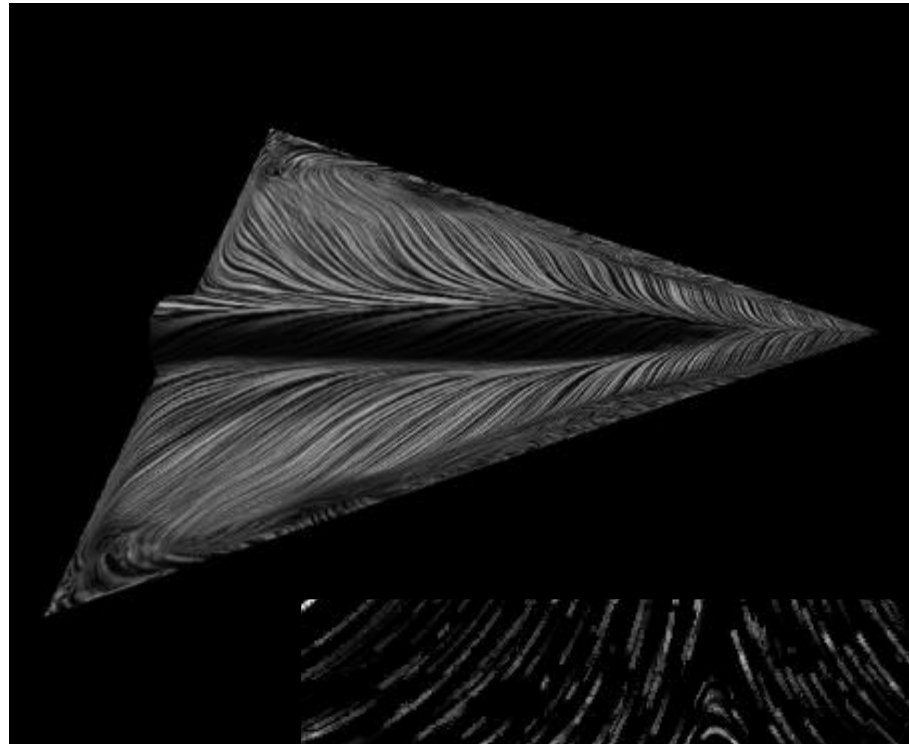
- ◆ 3D-Strömungen rund um “Hindernisse”
- ◆ Randströmungen auf Oberflächen (2D)

## 3D-Strömungsvisualisierung

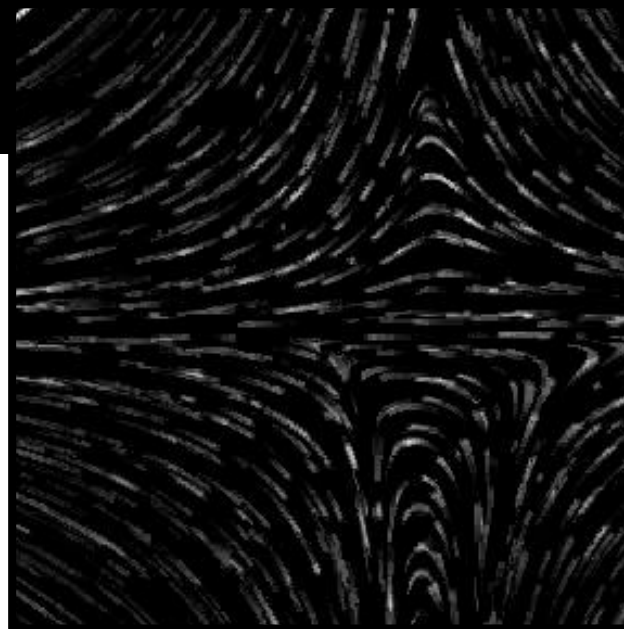
- ◆ 3D×3D-Strömungen
- ◆ Simulationen, 3D-Modelle

# 2D/Flächen/3D – Beispiele

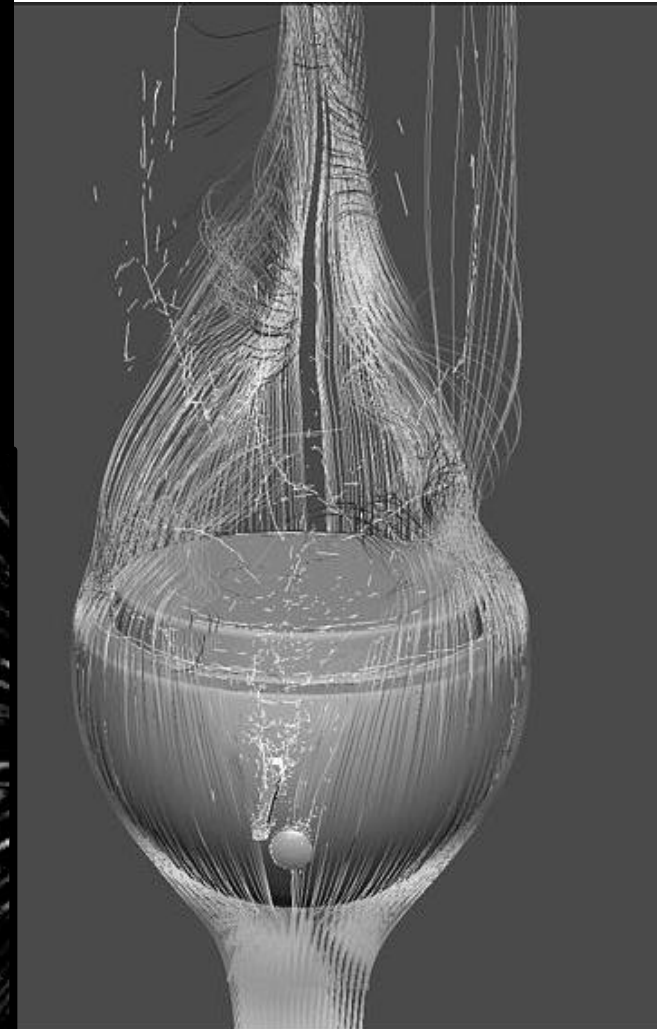
Fläche



2D



3D



# Steady vs. Time-dependent

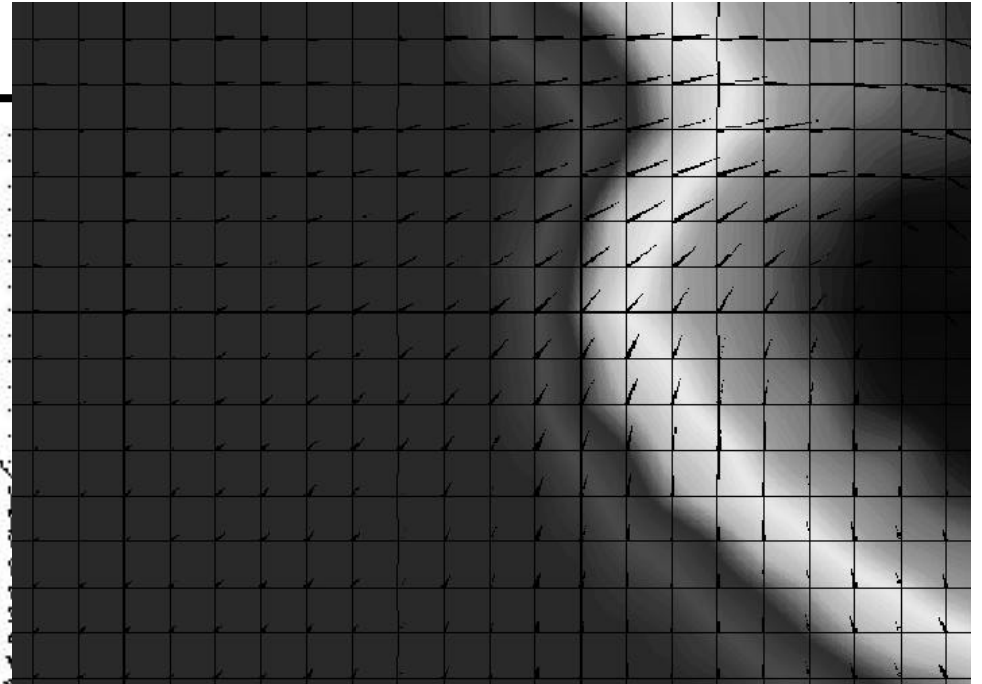
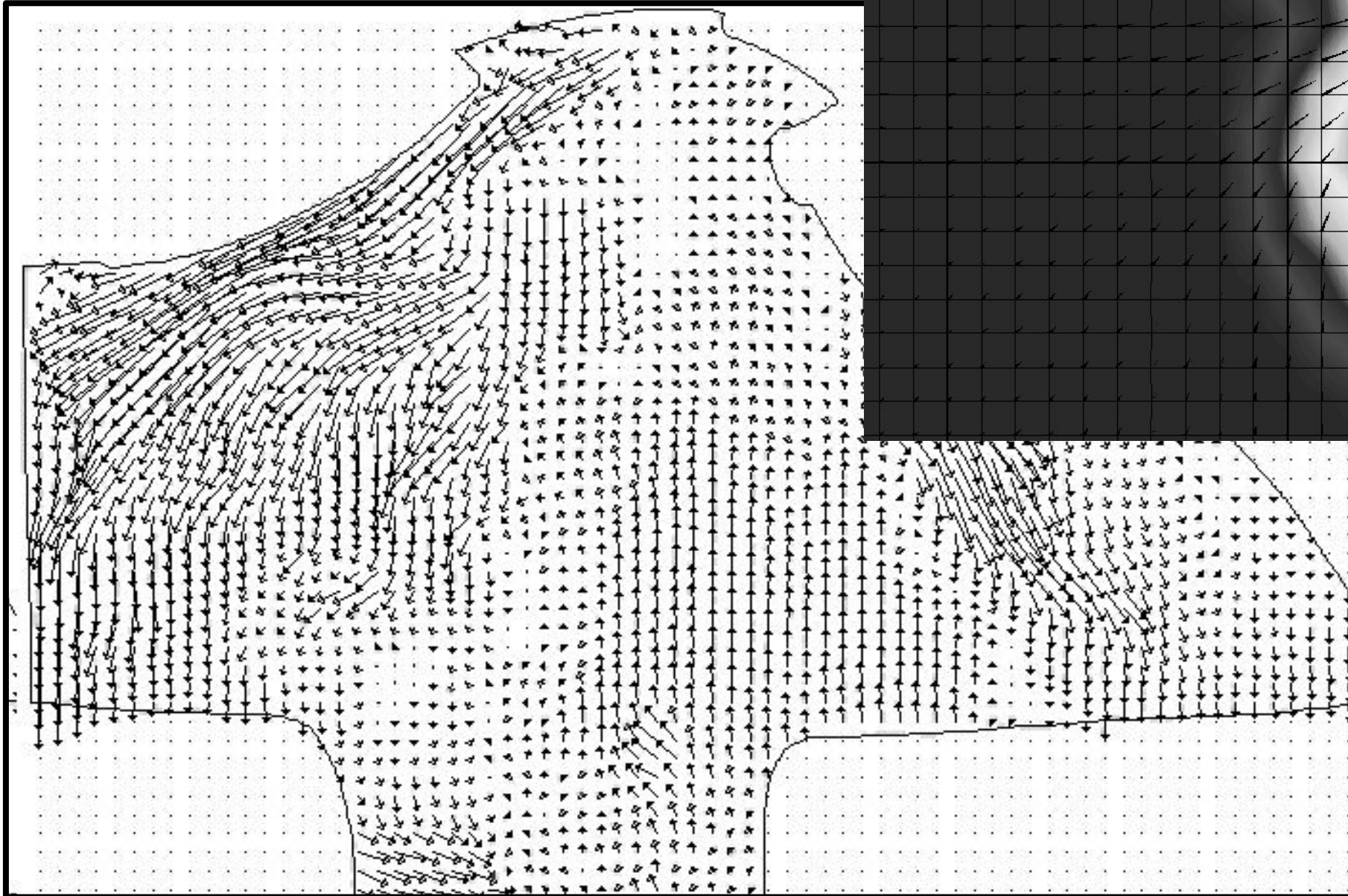
## Steady (time-independent) flows:

- ◆ Strömung über Zeit unveränderlich
- ◆  $\mathbf{v}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^n$ , z.B. laminare Strömungen
- ◆ einfacherer Zusammenhang

## Time-dependent (unsteady) flows:

- ◆ Strömung ändert sich über Zeit selbst
- ◆  $\mathbf{v}(\mathbf{x}, t): \mathbb{R}^n \times \mathbb{R}^1 \rightarrow \mathbb{R}^n$ , z.B. turbolente Str.
- ◆ komplexerer Zusammenhang

# Time-dependent vs. steady



# Direkte vs. indirekte FlowViz

## Direkte Strömungsvisualisierung:

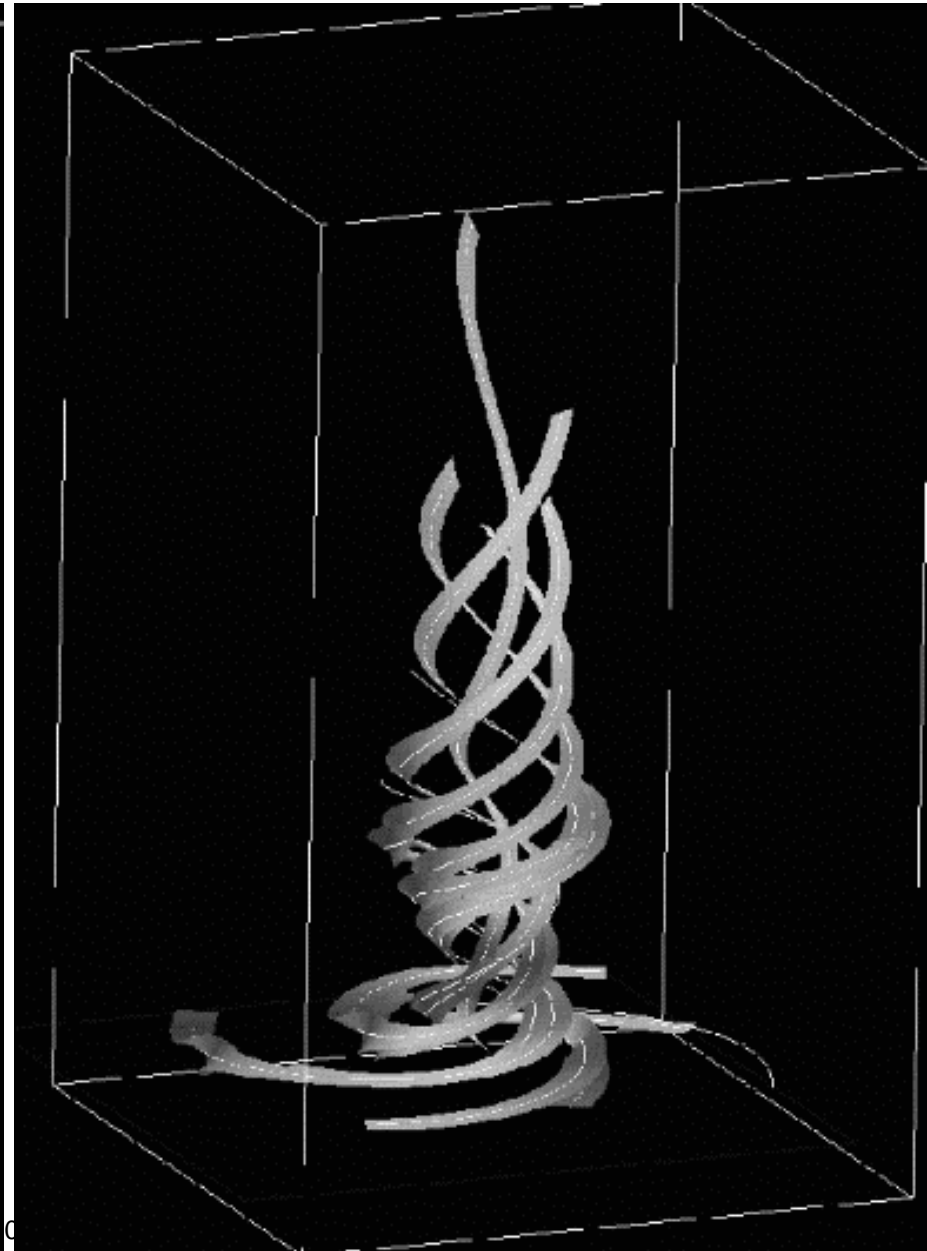
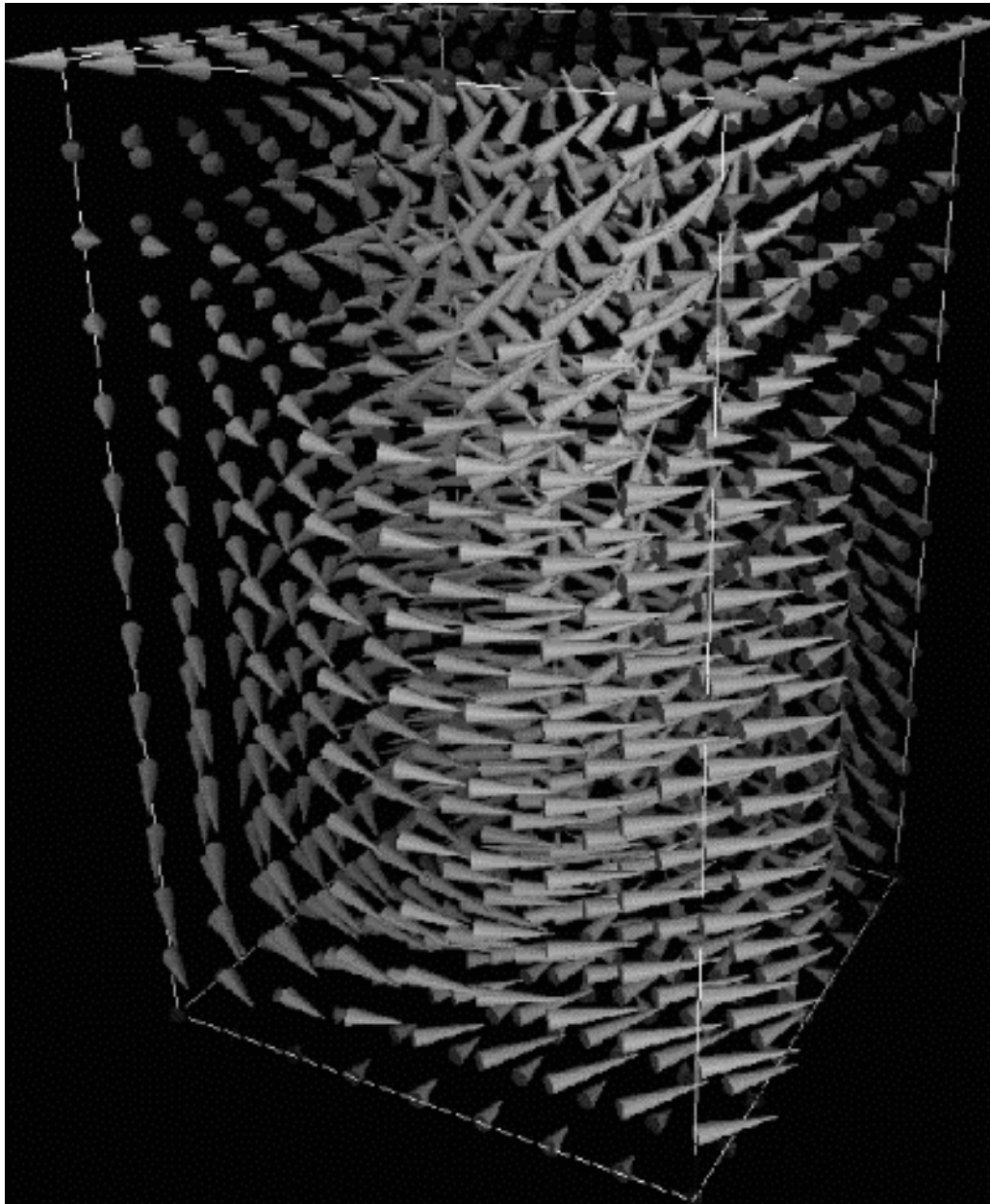
- ◆ Überblick über Jetztzustand der Strömung
- ◆ Visualisierung der Vektoren
- ◆ Pfeildarstellungen, Verwischtechniken

## Indirekte Strömungsvisualisierung:

- ◆ Verwendung einer Zwischenrepräsentation: Vektorfeldintegration über Zeit
- ◆ Visualisierung der Zeitentwicklung
- ◆ Strömungslinien, Strömungsflächen



# Direkt vs. indirekt – Beispiel



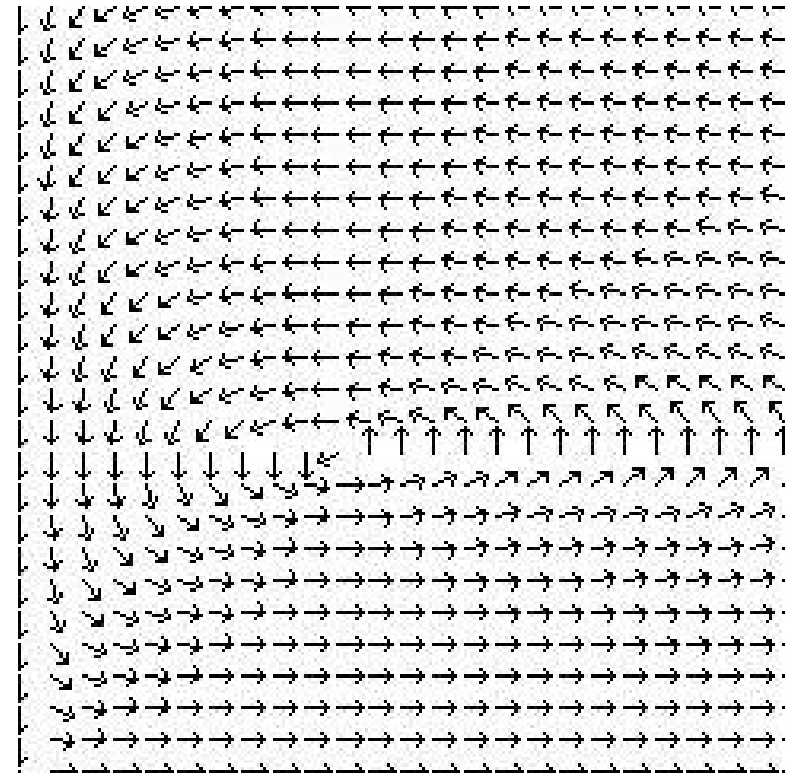
# **Strömungsvisualisierung mit Pfeilen**

**Hedgehog plots, etc.**

# FlowViz mit Pfeilen

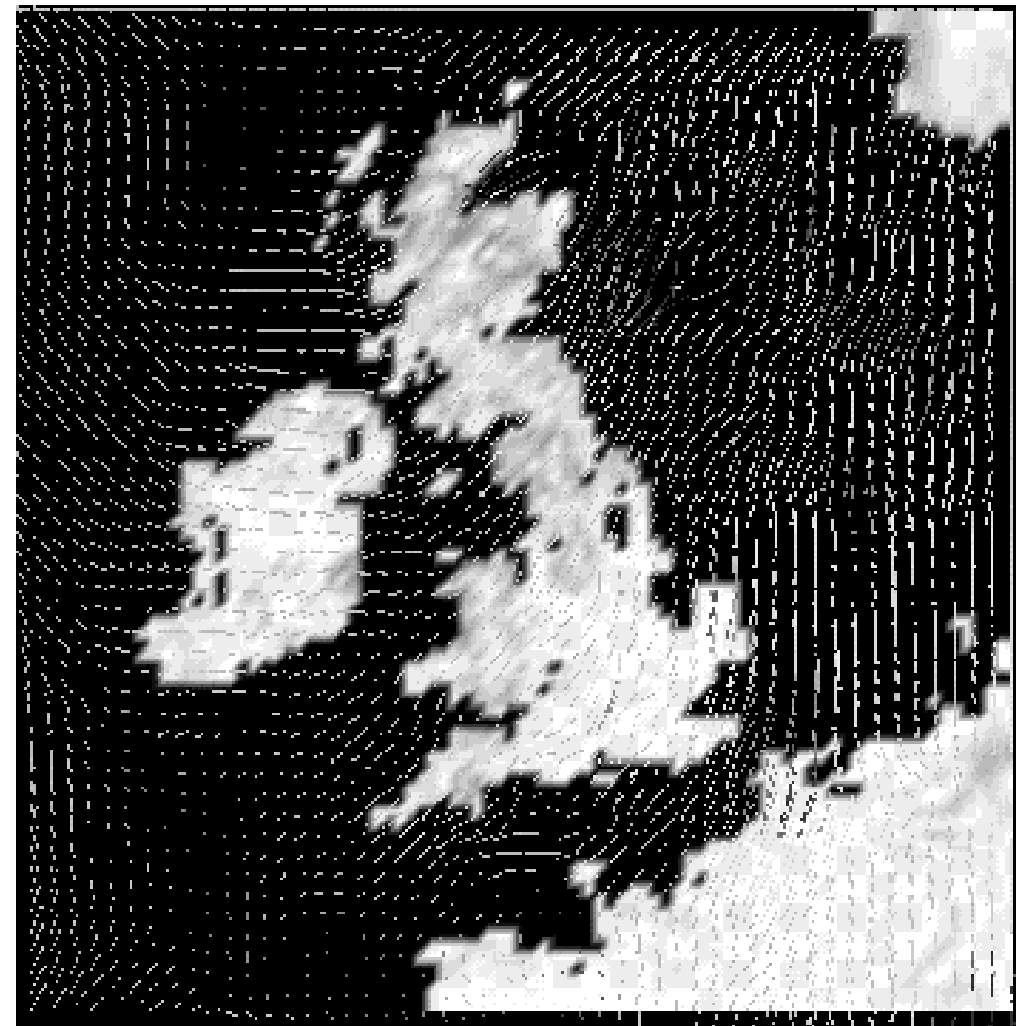
## Aspekte:

- ◆ Direkte FlowViz
- ◆ normierte Pfeile vs. Skalierung mit Geschwindigkeit
- ◆ 2D: ganz gut brauchbar, 3D: meist problematisch
- ◆ oft nur bedingt verständlich (zeitliche Komponente fehlt)
- ◆ oft in Verwendung!



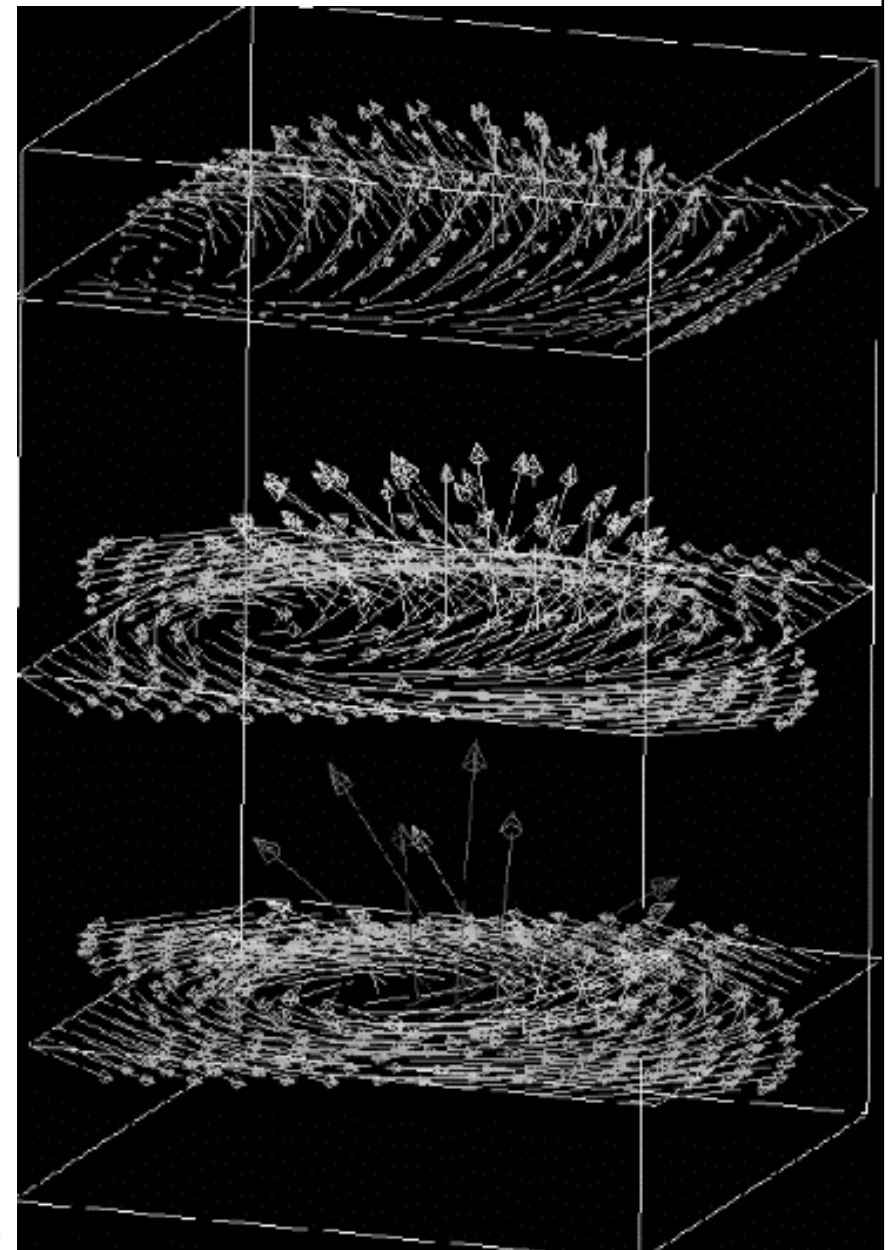
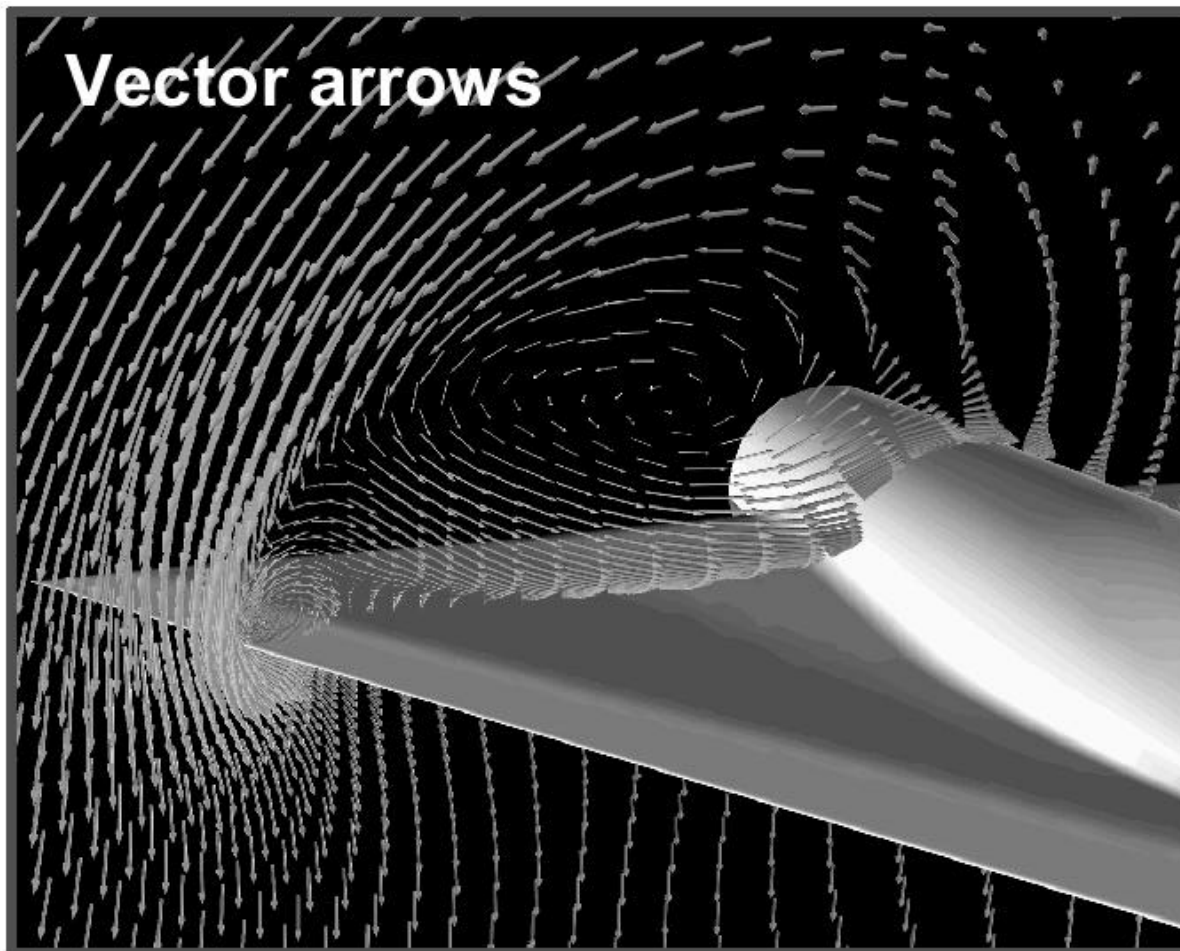
# Pfeile im 2D

## Skalierte Pfeile vs. farb-codierte Pfeile



# Pfeile im 3D

**Kompromiß:  
Pfeile nur in Schichten**

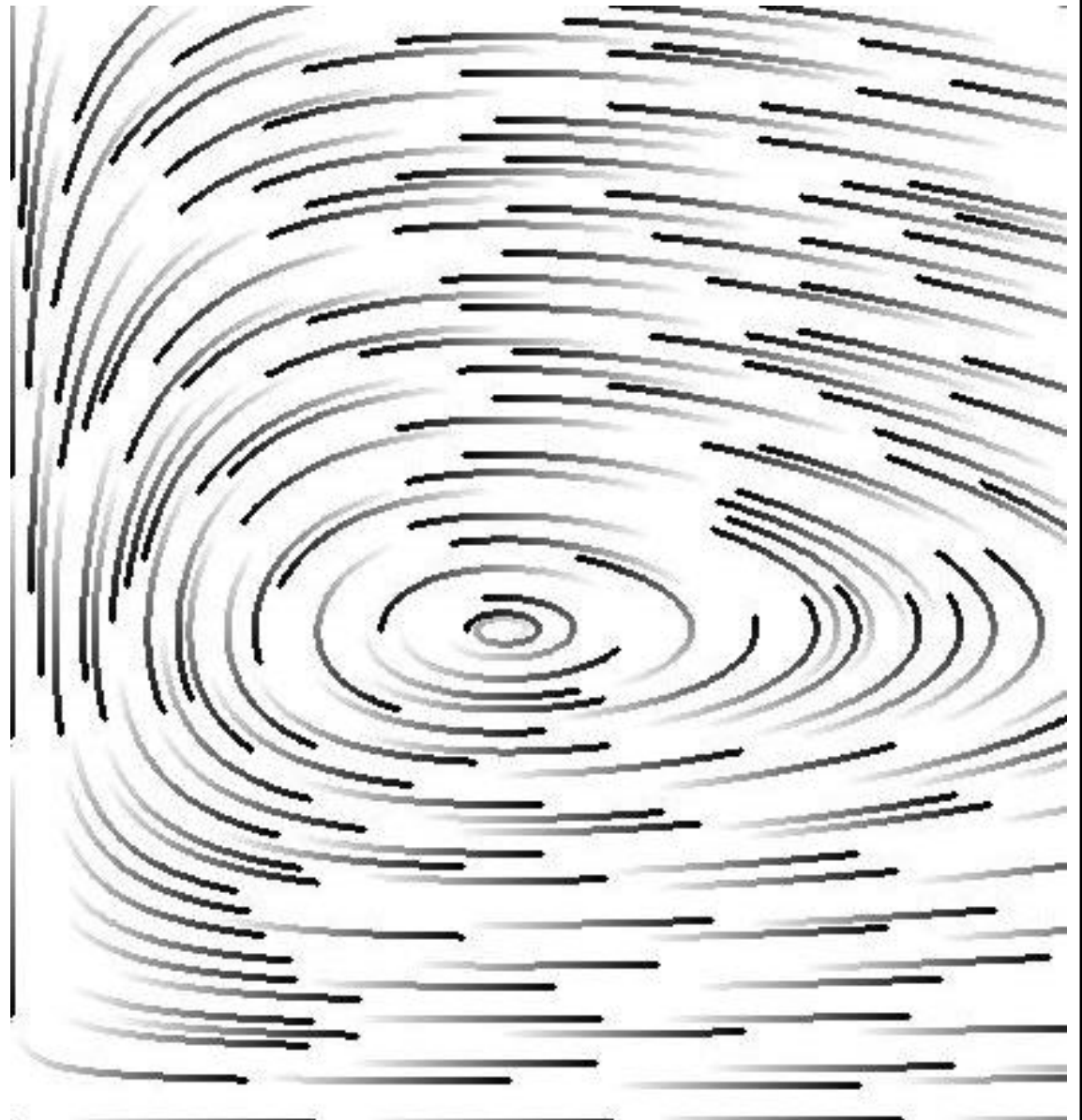
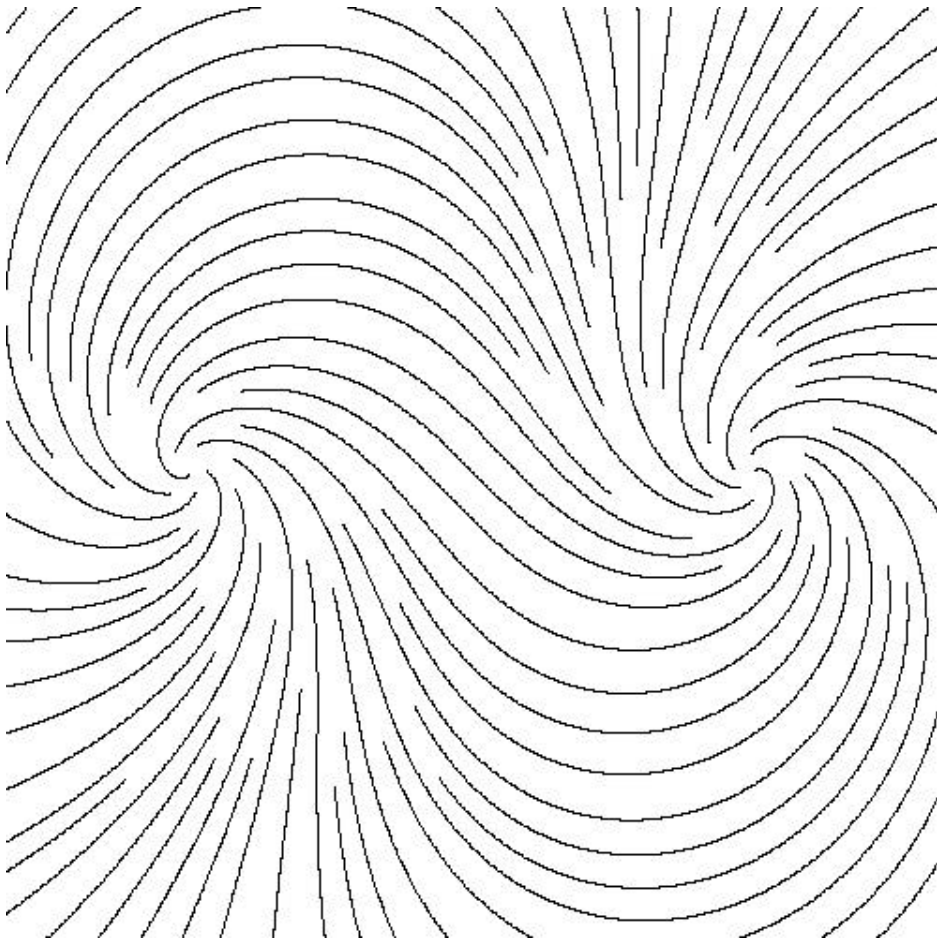


# **Strömungsvisualisierung mit Strömungslinien**

**Strömungslinien,  
Partikelbahnen, etc.**

# Strömungslinien im 2D

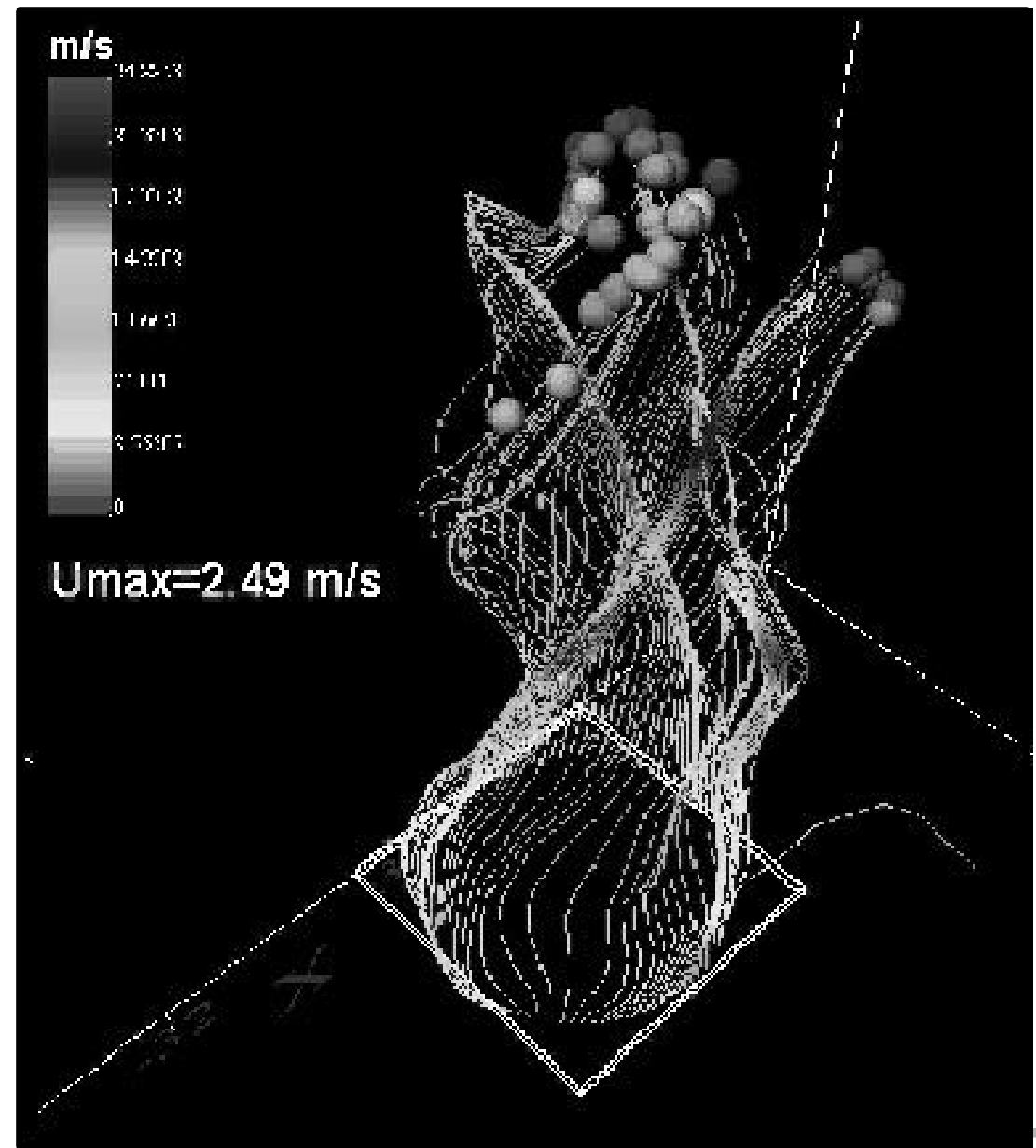
**Gut geeignet, um  
Überblick zu geben**



# Visualisierung mit Partikel

**Partikelbahnen =  
Strömungslinien  
Varianten (time-  
dependent data):**

- ◆ streak lines:  
immer neue  
Partikel los-  
lassen
- ◆ path lines:  
Langzeitweg  
eines Partikels

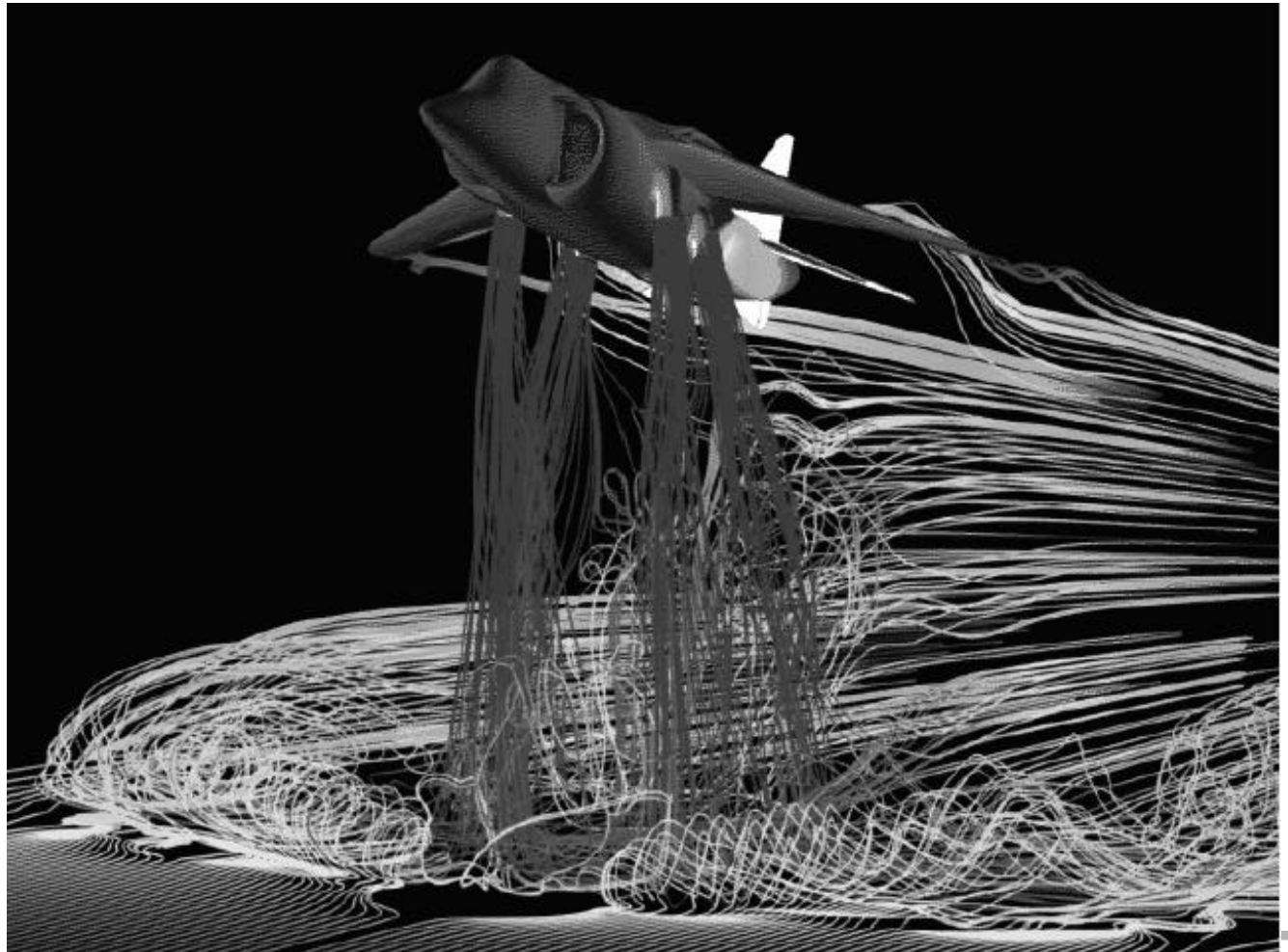




# Strömungslinien im 3D

**Farbcodierung:  
Geschwindigkeit**

**Selektive  
Platzierung**

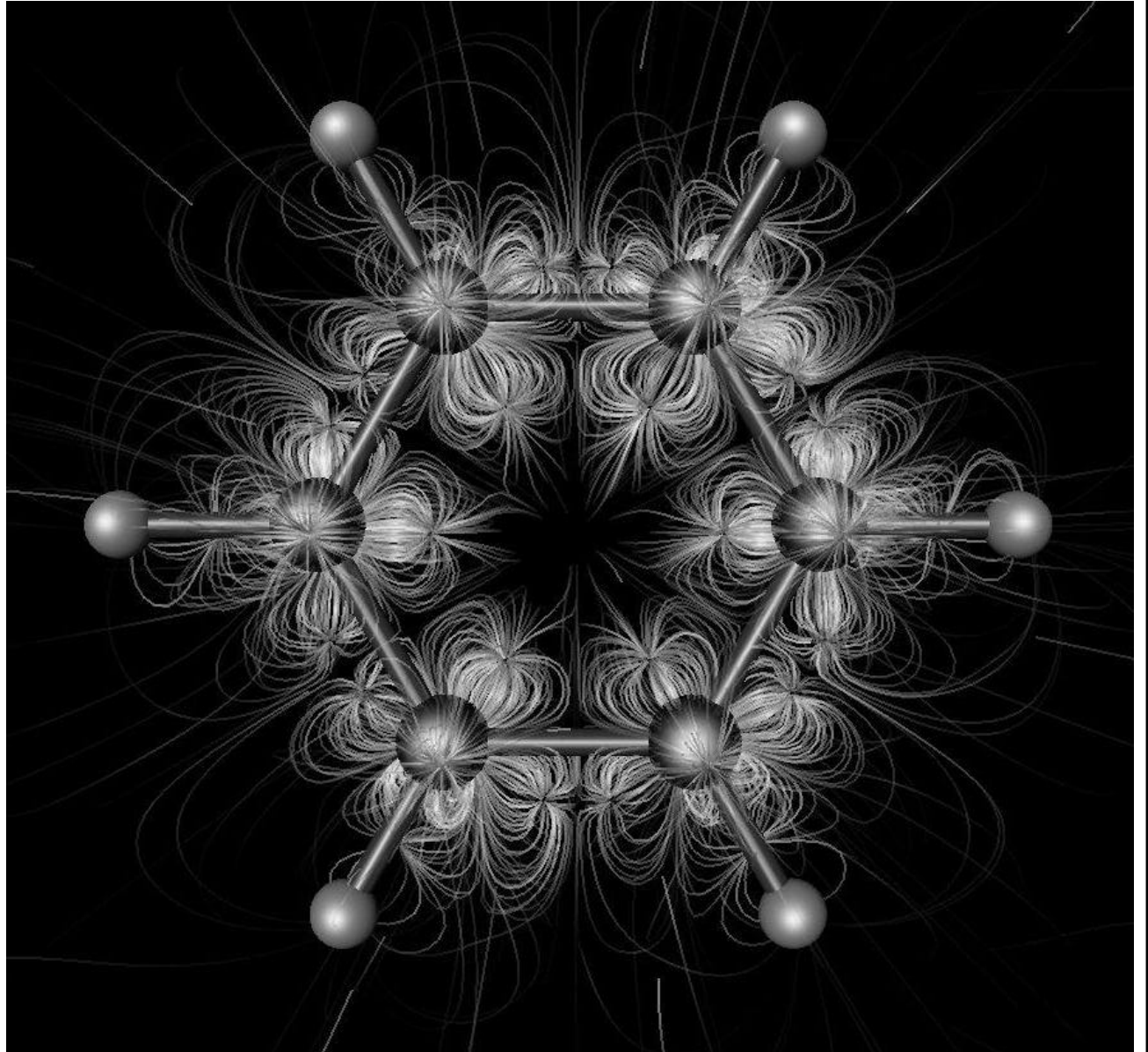


# Illuminated Stream Lines

**Beleuchtung  
von 3D Kurven**



**bessere Wahr-  
nehmung!**



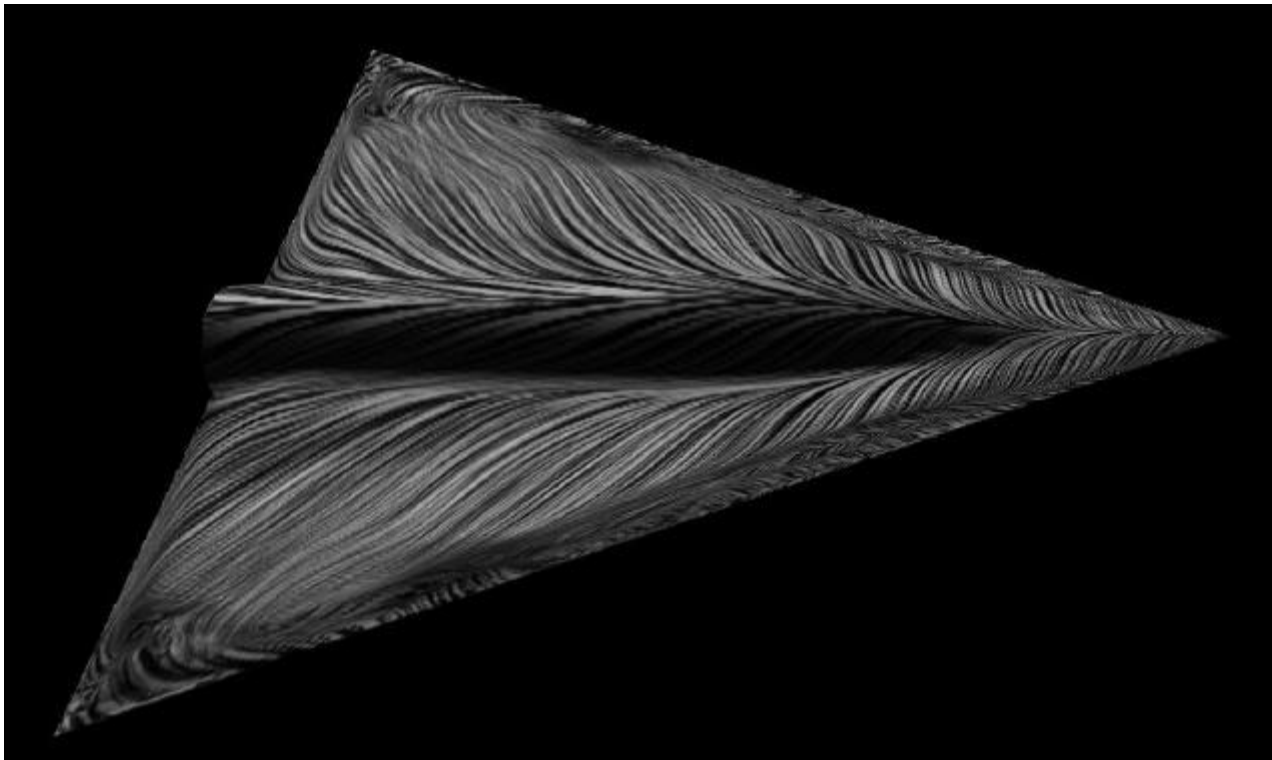
# Line Integral Convolution

**Strömungsvisualisierung  
im 2D oder auf Flächen**

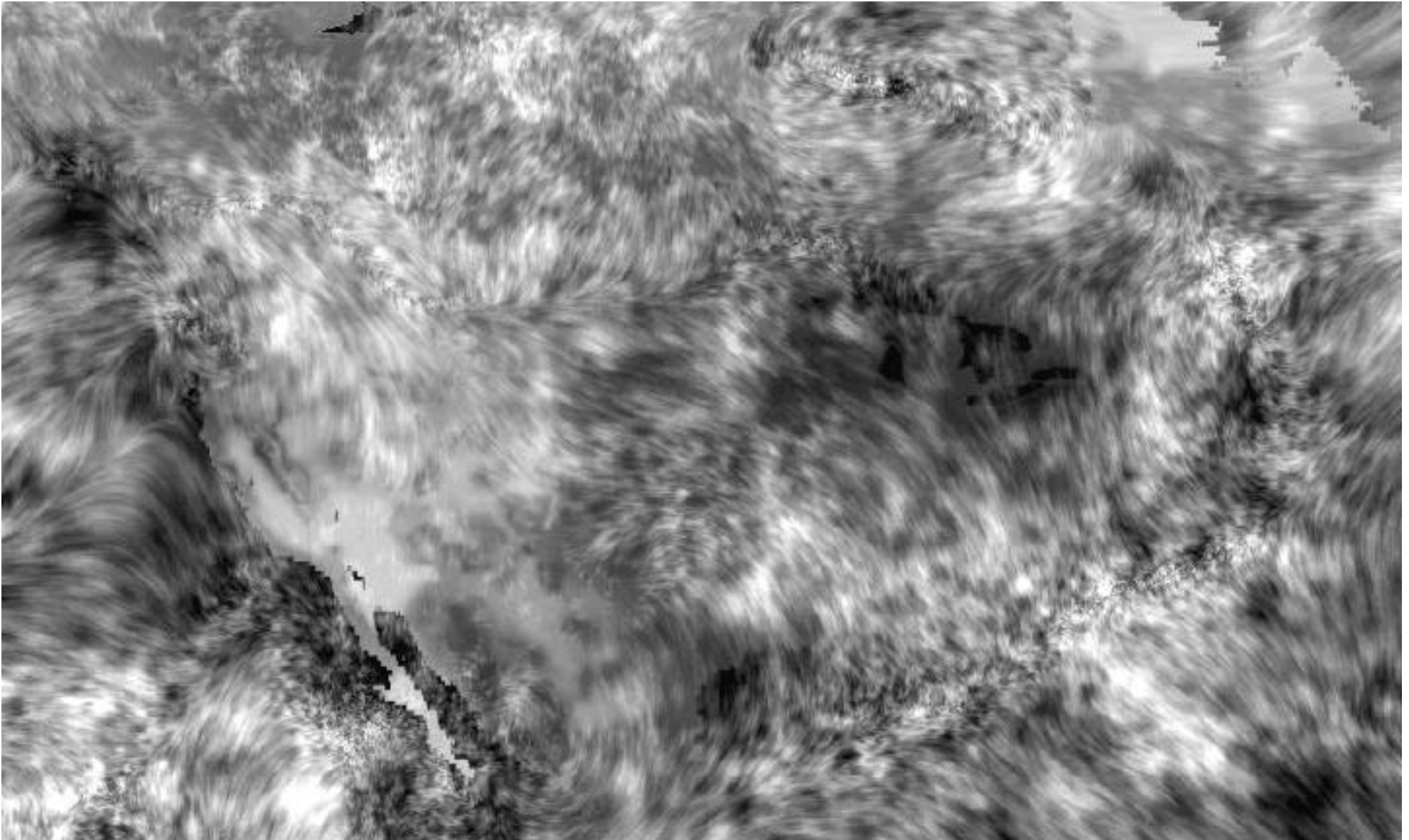
# LIC – Einleitung

## Aspekte:

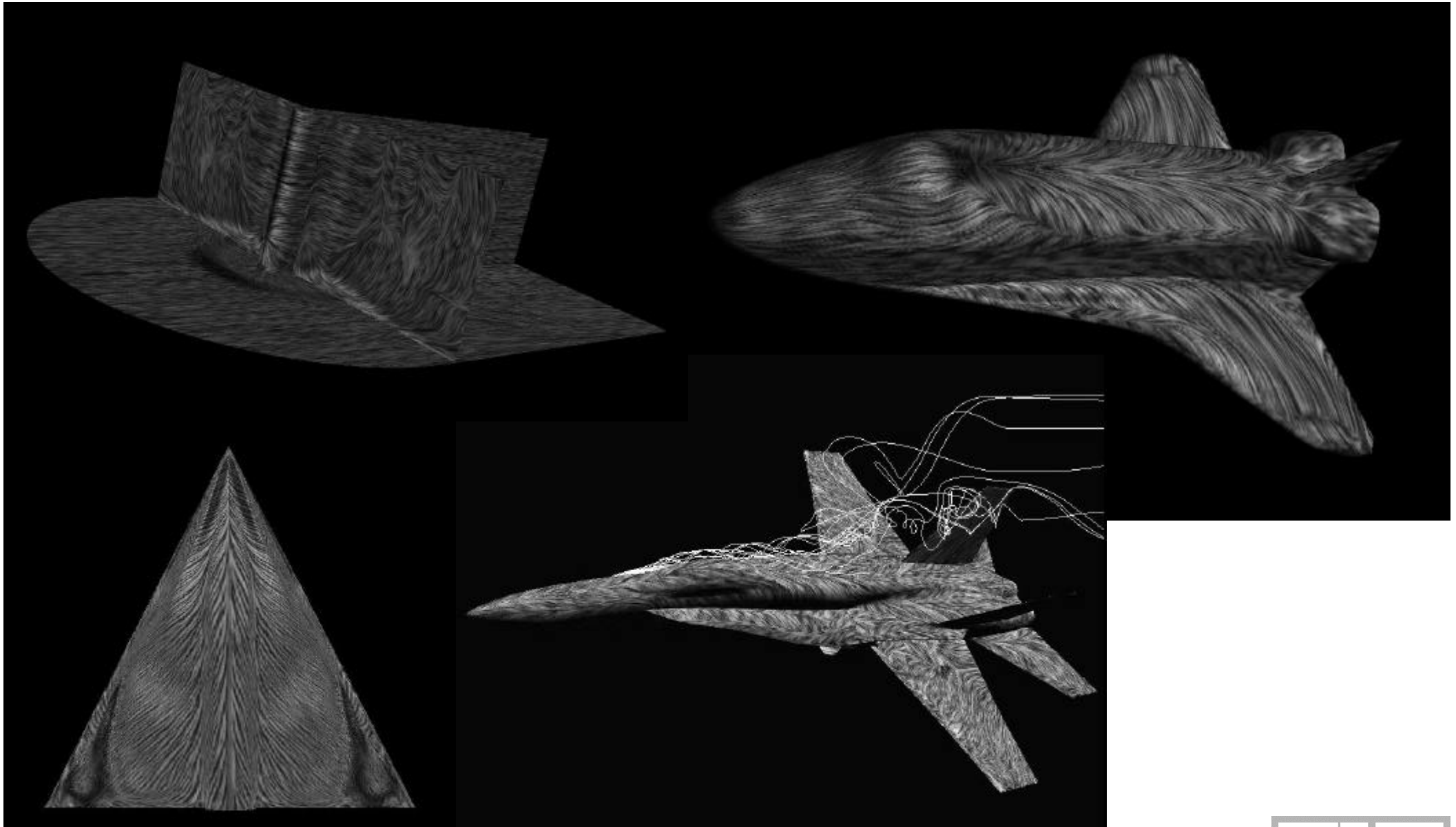
- ◆ Ziel: Gesamtüberblick über Strömung
- ◆ Ansatz: Verwendung von Texturen
- ◆ Idee: Strömung  $\Leftrightarrow$  visuelle Korelation
- ◆ Beispiel:



# LIC im 2D – Beispiel

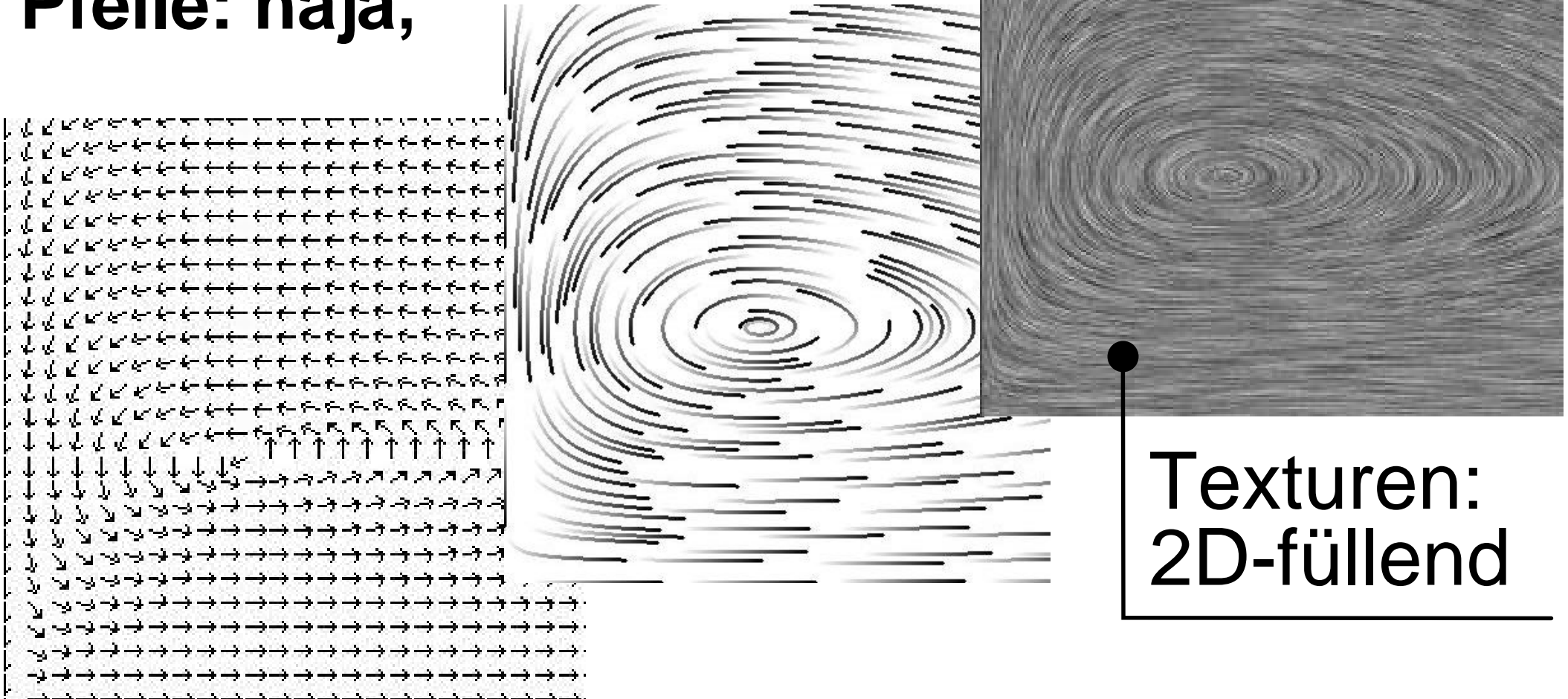


# LIC – Beispiele auf Flächen



# Pfeile vs. Str.-Linien vs. Texturen

**Strömungslinien: selektiv,**  
**Pfeile: naja,**



**Texturen:  
2D-füllend**

# Information Visualization

## InfoViz-Kontext, Allgemeines



# Information Visualization

**Rather new branch of viz, next to:**

- ◆ volume visualization
  - ◆ flow visualization
- } scientific viz

**Deals with:**

- ◆ abstract data
- ◆ multi-dimensional data
- ◆ very large data-sets

# InfoViz vs. SciViz

**usually no inherent spatial arrangement in general n-dimensional data**

**prime goals:**

- ◆ useful visual metaphors
- ◆ flexible interaction mechanisms
- ◆ useful tools for exploration

**inherent spatial arrangement (2D, 3D) often 2- or 3-dimensional data**

**prime goals:**

- ◆ fast visualization and rendering
- ◆ interactive applications
- ◆ useful tools for analysis

# Visual metaphors & interaction

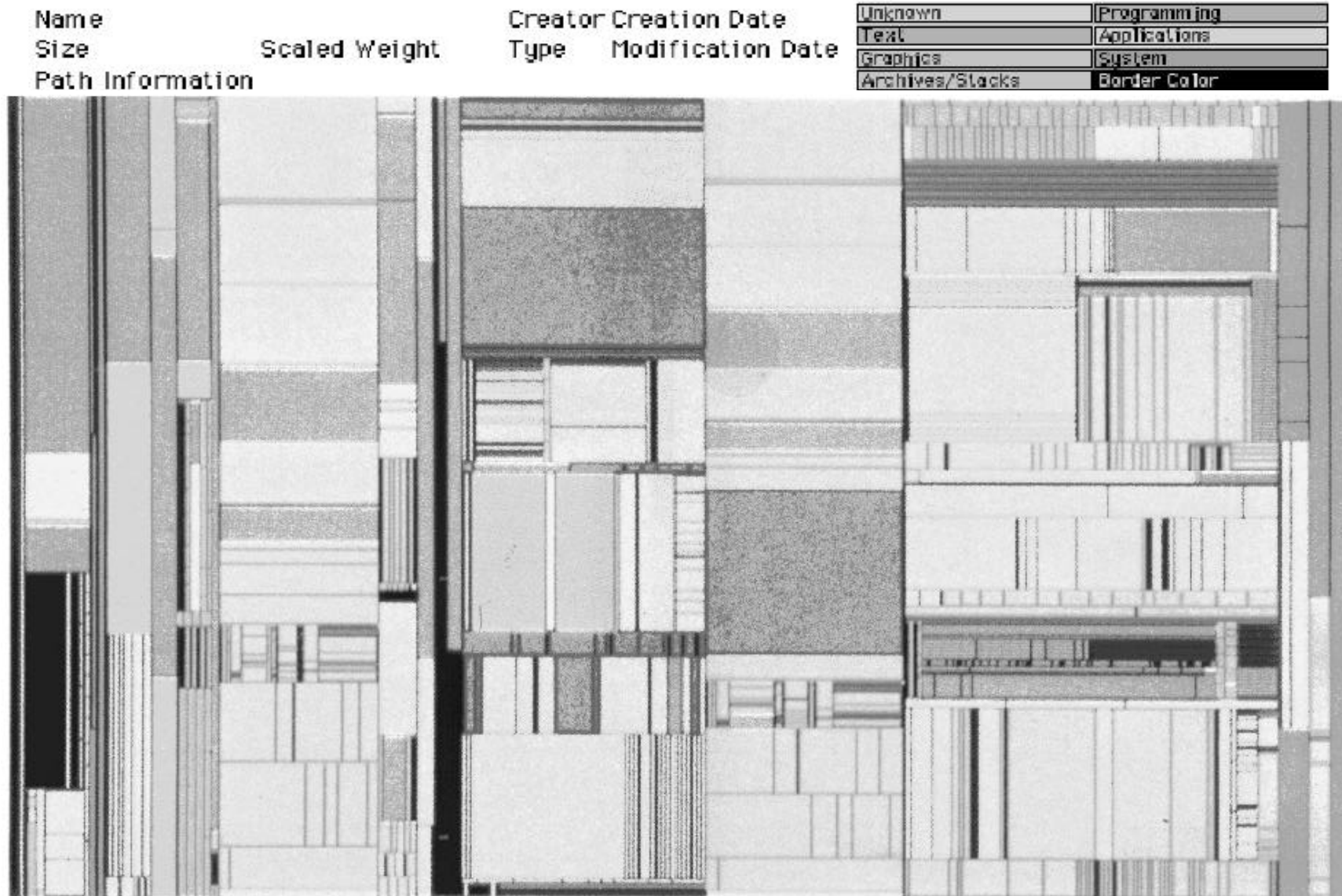
## Req. 1: useful visual metaphors

- ◆ how to represent abstract data, n-dimensional data, very large data-sets?
- ◆ how to locate data items?
- ◆ 2D or 3D representation?

## Req. 2: flexible interaction techniques

- ◆ changing between different views
- ◆ changing the focus
- ◆ zooming, panning, sub-setting, ...

# Tree map: 1500 files, 190 dirs.



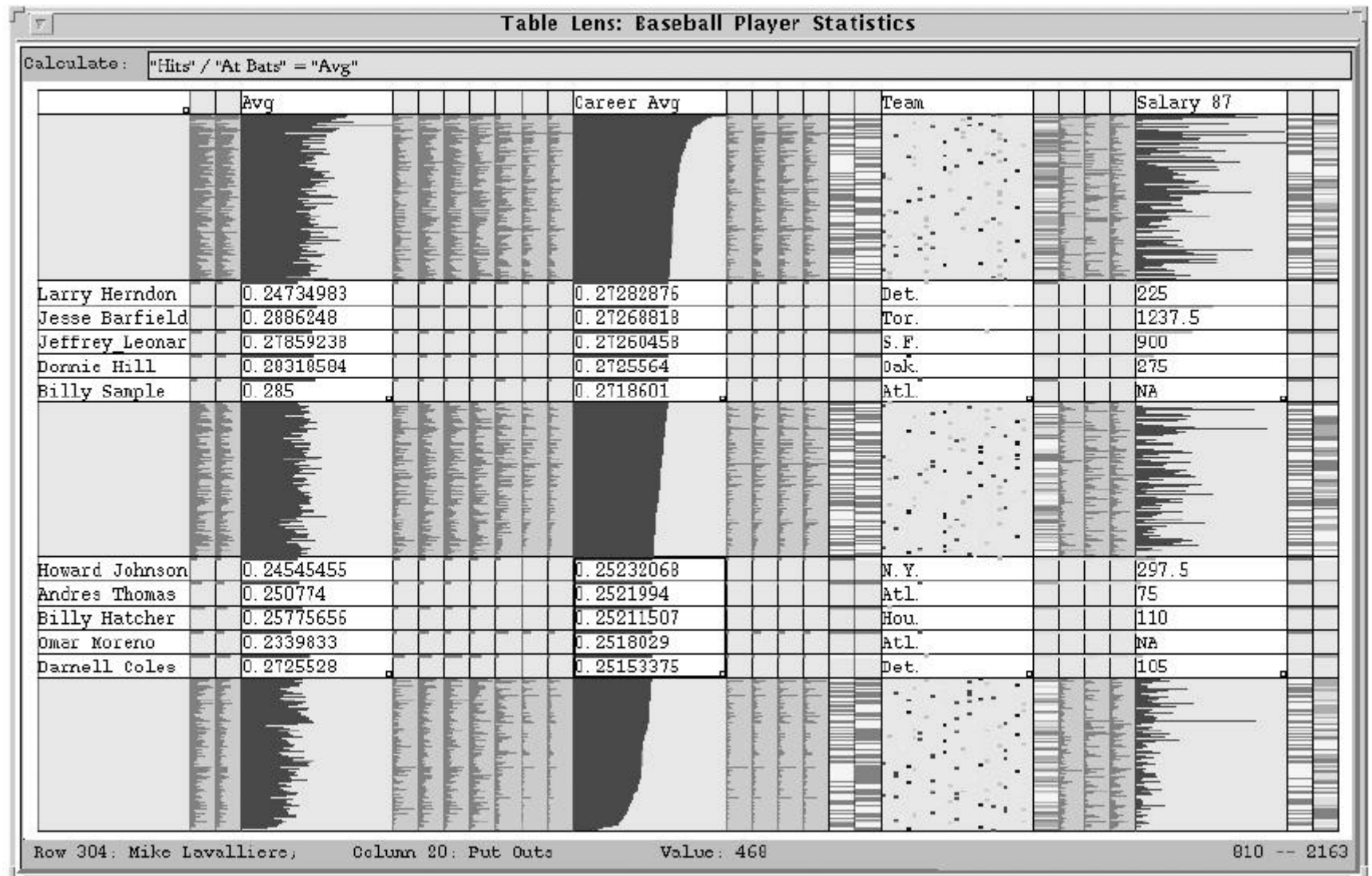
**Figure 8: Slice-and-Dice Treemap**

Macintosh file hierarchy with 1500 files and 190 directories.

Notice the duplicate directory at the top level which contains a copy of the system file.

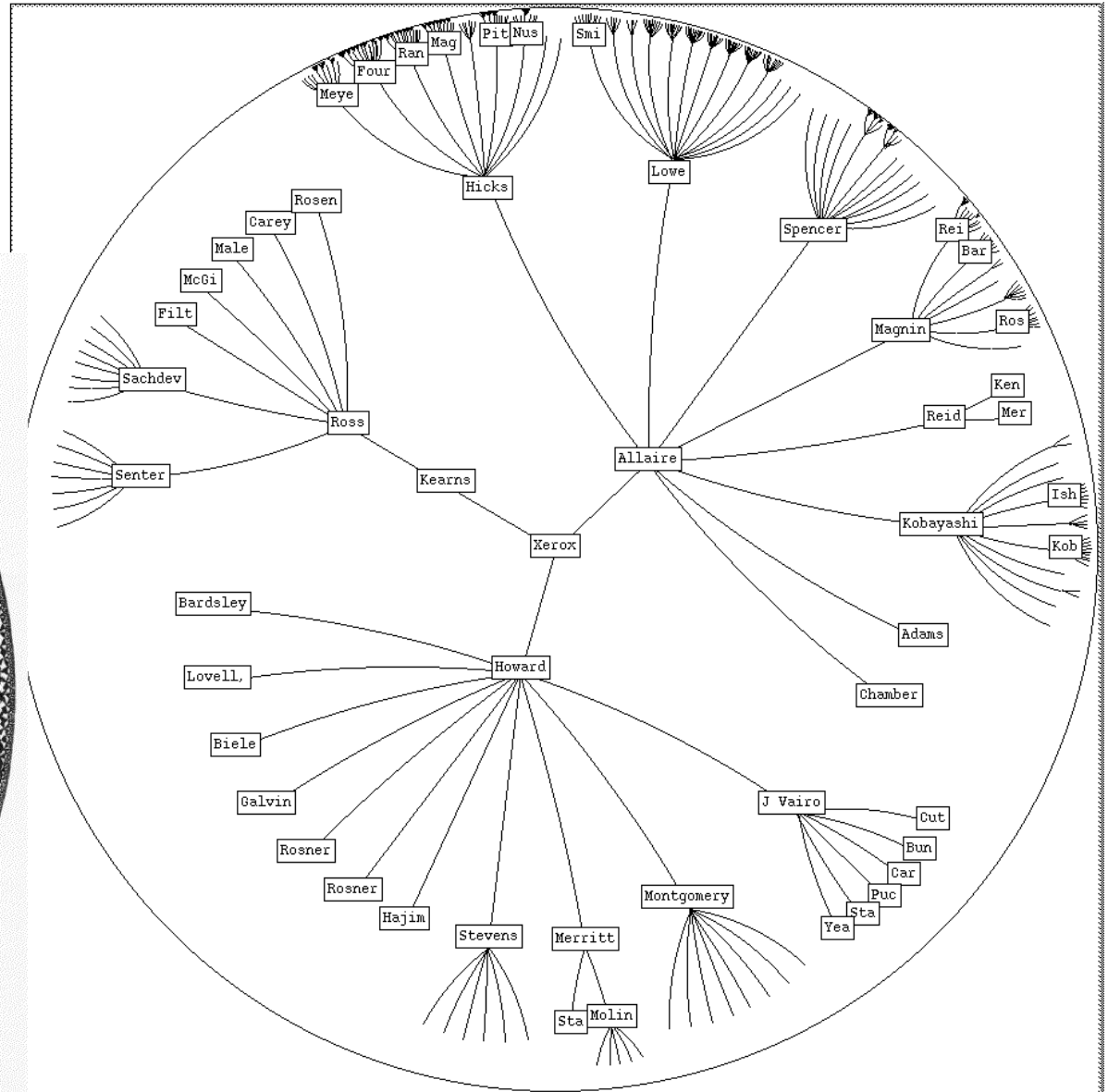
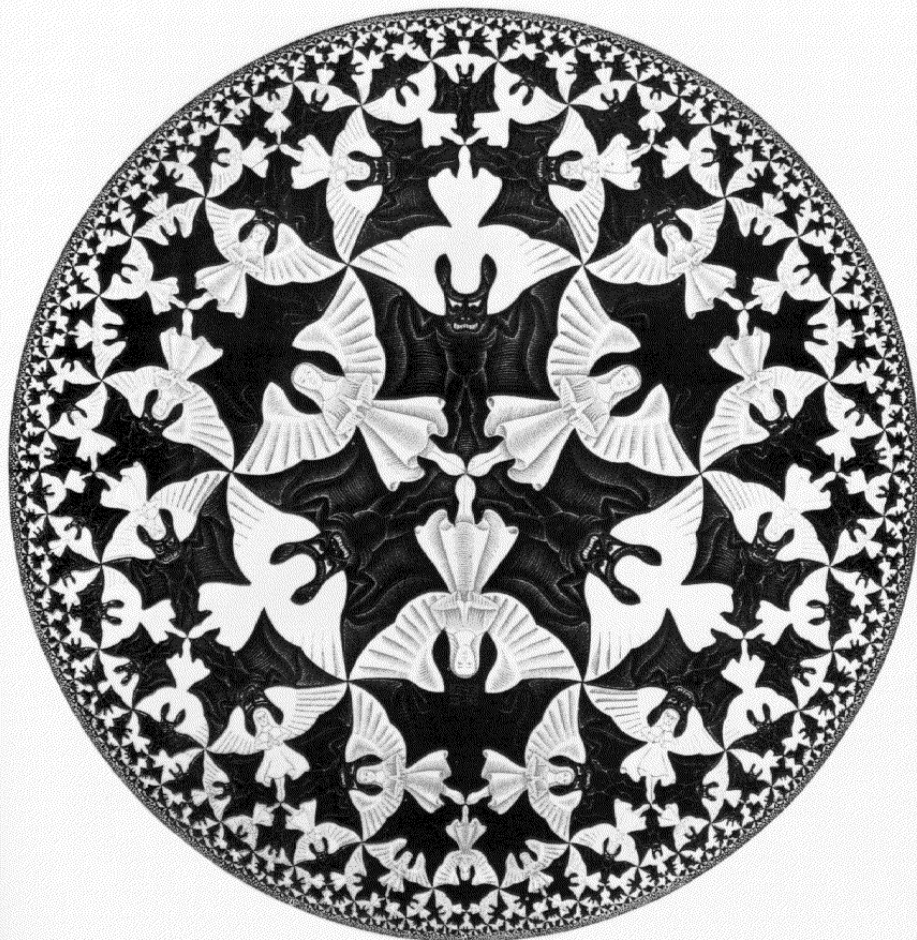
# Table lens

## Demo



# Hyperbolic trees – idea

## Art vs. InfoViz



T. Kearns, CEO, Chairman, Chief Executive Officer

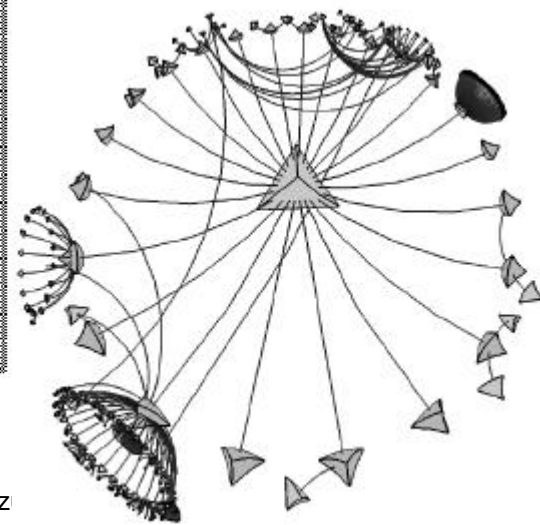
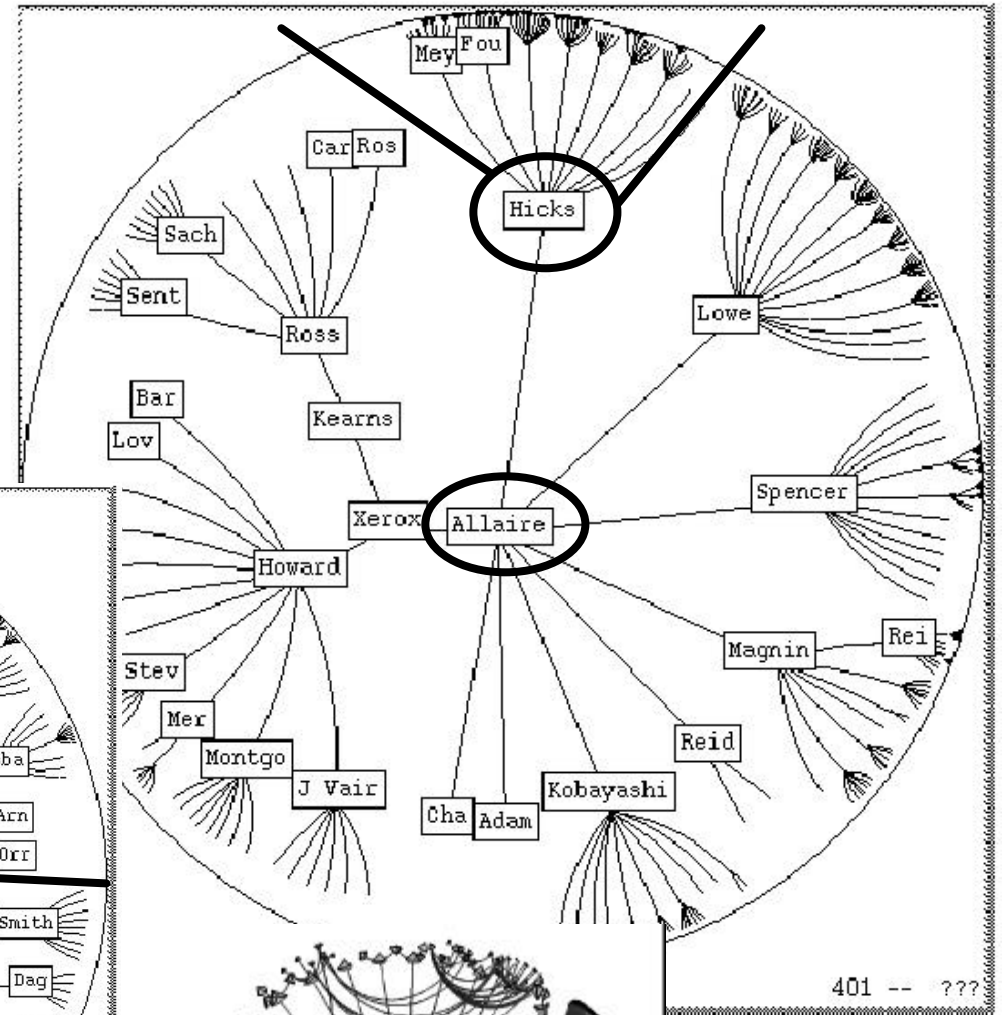
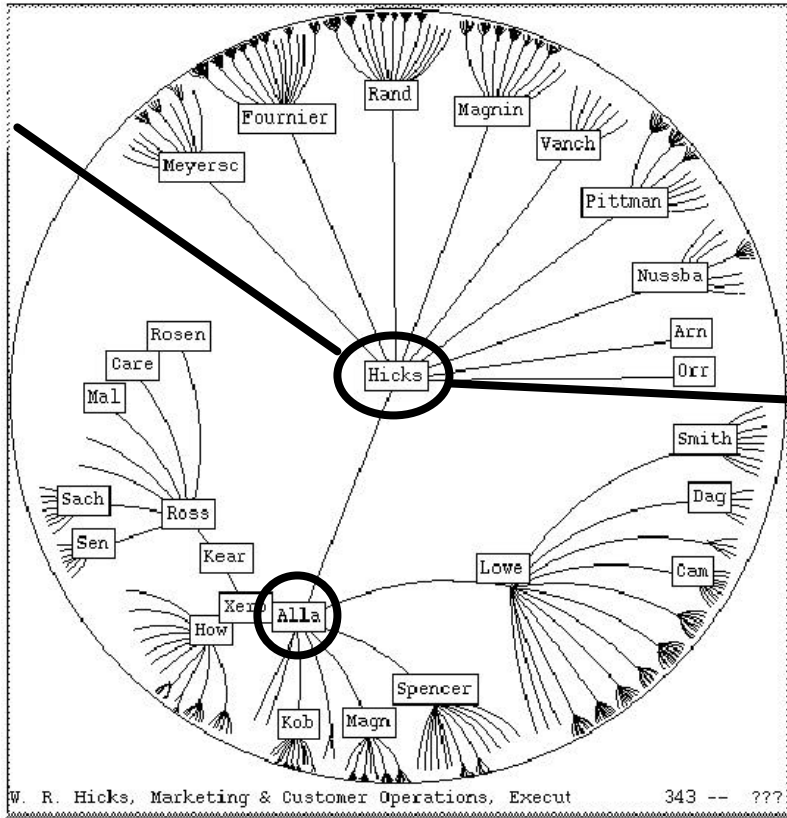
667 -- 135

# Hyperbolic trees

## Using hyperbolic geometry

## Focus change through pan

## Works in 3D also



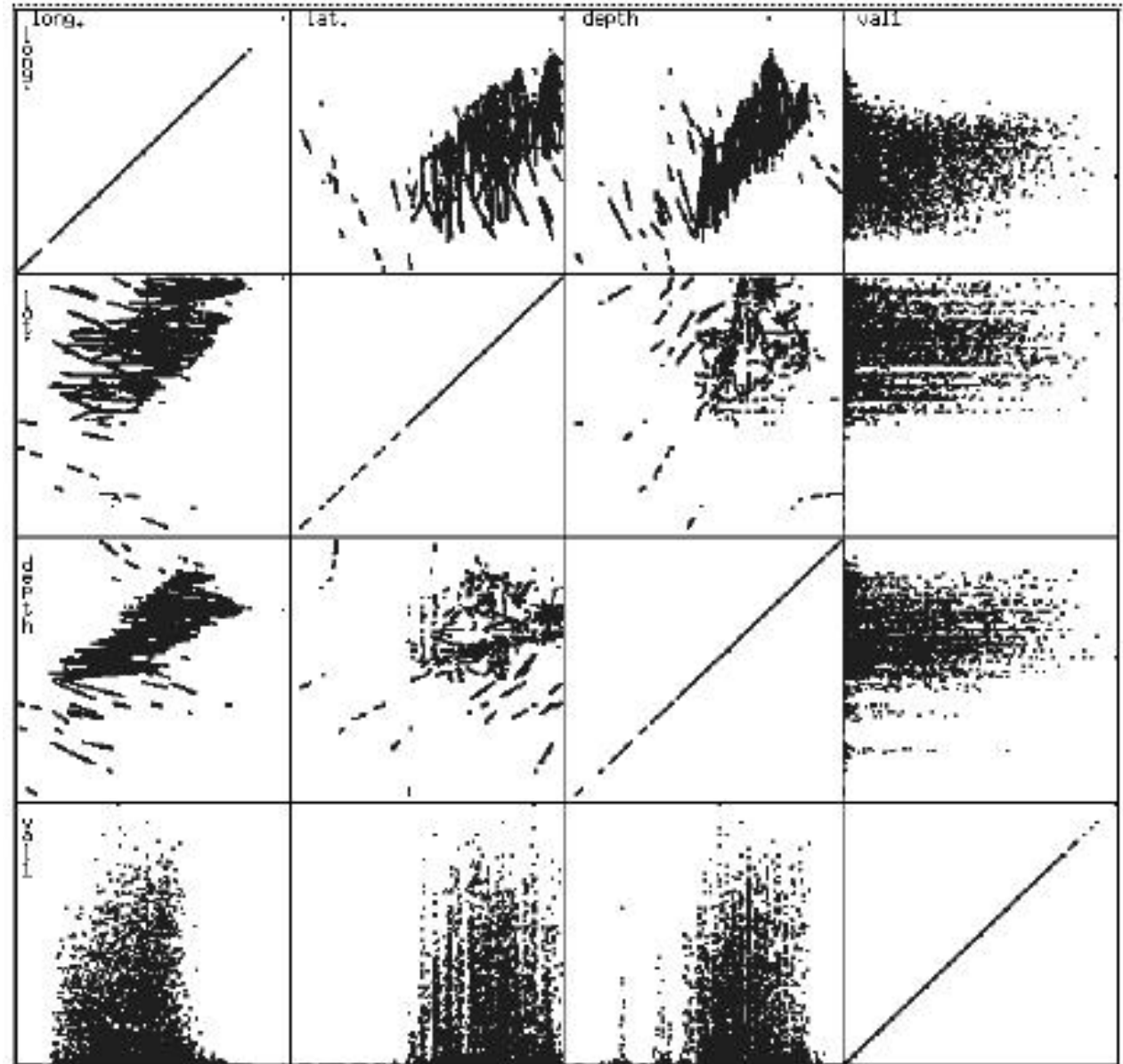
# Scatterplots, Scatterplot-matrices

## Scatterplot:

- ◆ 2 variables
- ◆ data records: single points

## Scatterplot-matrices:

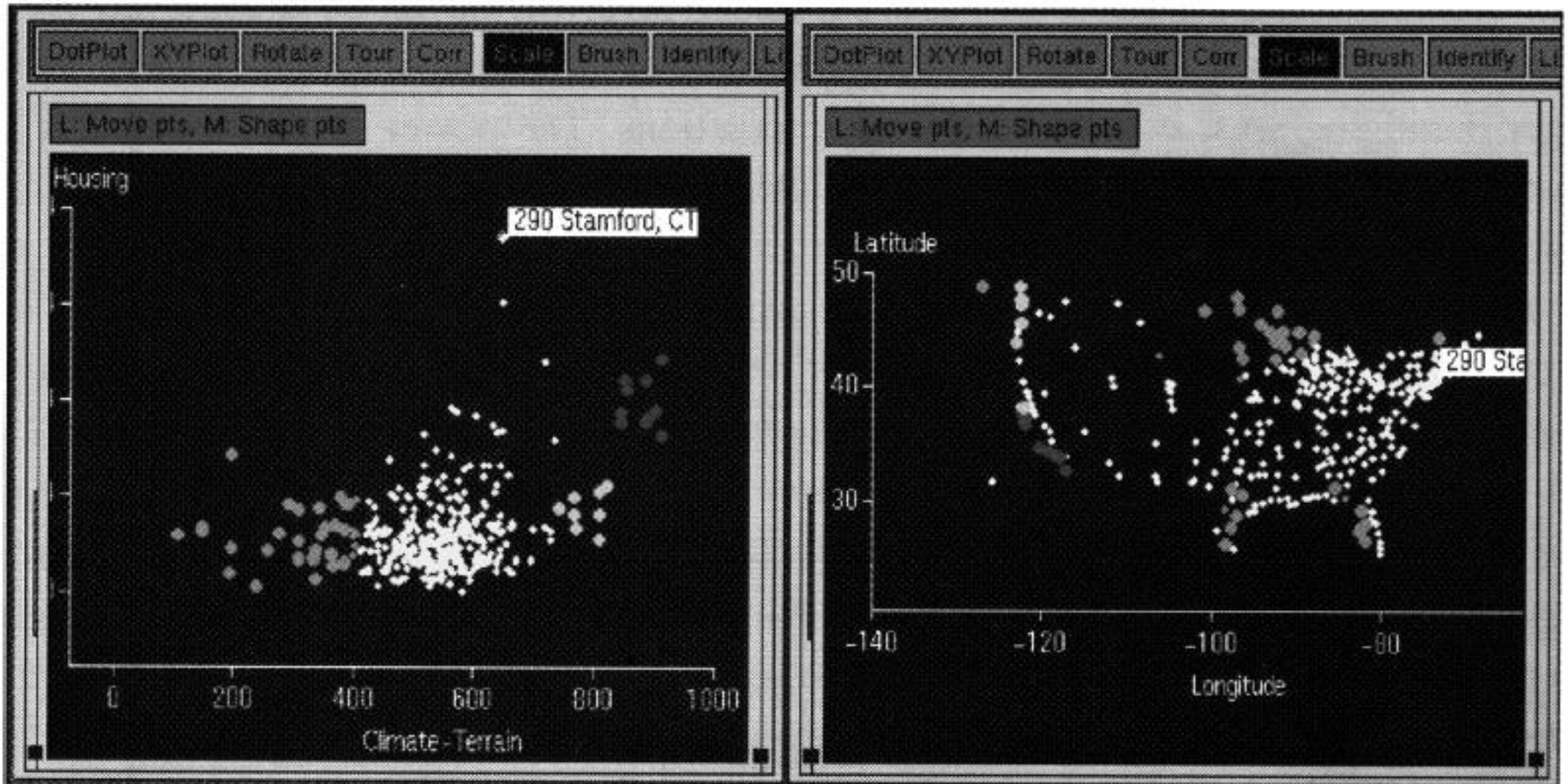
- ◆ all variables vs. all others





# Linking & Brushing (XGobi)

## Linked displays:



# WEAVE – SciViz+InfoViz

## Linking of SciViz- and InfoViz-views:

- ◆ 3D view  
(SciViz)
- ◆ scatterplot
- ◆ histogram

## Brushing!

