Lecture 14: Scientific Visualization

Information Visualization
CPSC 533C, Fall 2006

Tamara Munzner

UBC Computer Science

26 Oct 2006

Credits

- almost unchanged from lecture by Melanie Tory (University of Victoria)
  - who in turn used resources from
  - Torsten Möller (Simon Fraser University)
  - Raghu Machiraju (Ohio State University)
  - Klaus Mueller (SUNY Stony Brook)

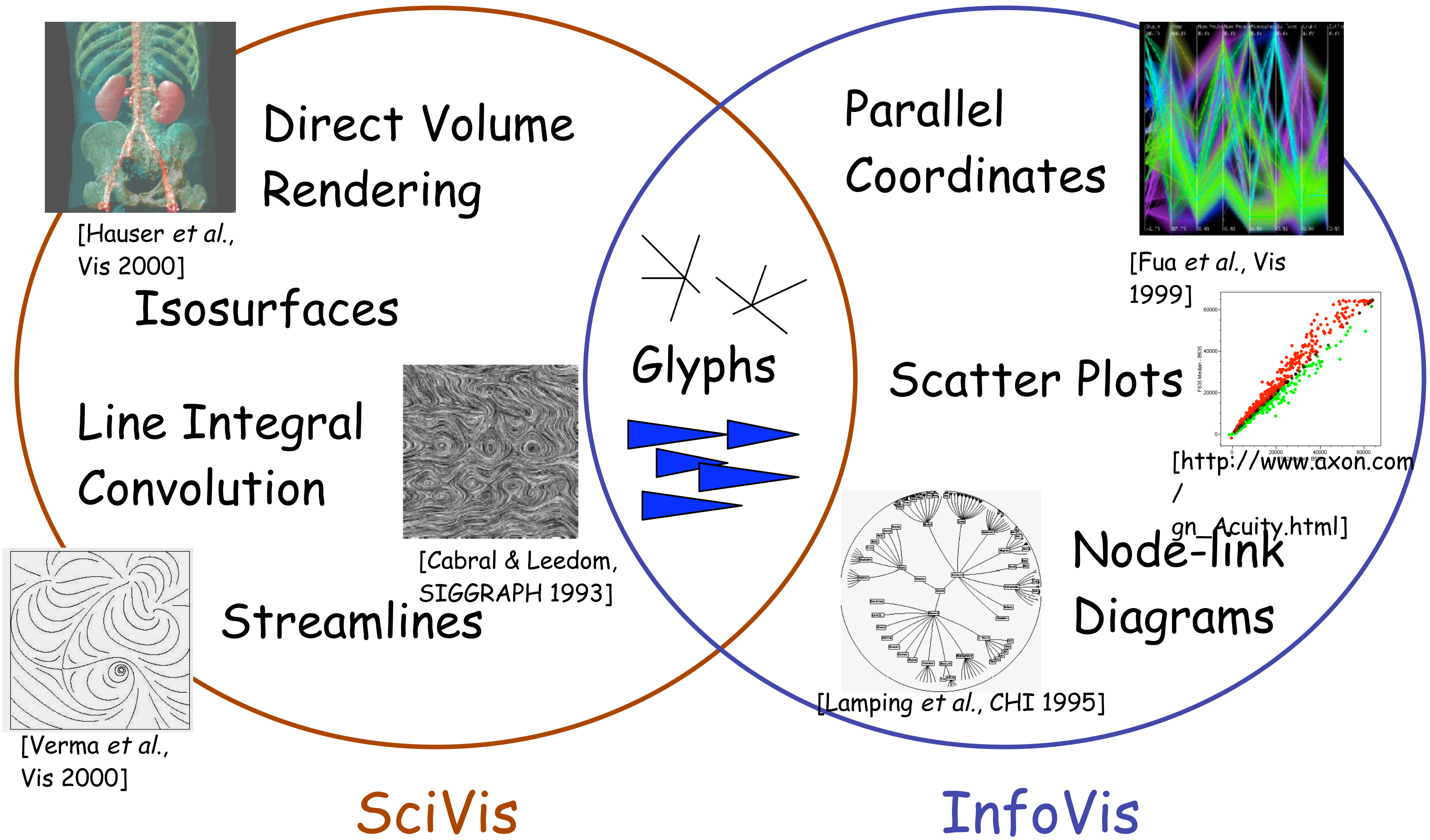
News

- Reminder: no class next week
  - I'm at InfoVis/Vis in Baltimore

Overview

- **What is SciVis?**
- Data & Applications
- Iso-surfaces
- Direct Volume Rendering
- Vector Visualization
- Challenges

Difference between SciVis and InfoVis



Difference between SciVis and InfoVis

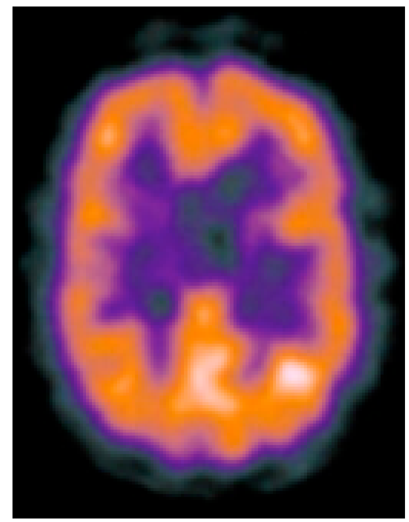
- **Card, Mackinlay, & Shneiderman:**
  - SciVis: Scientific, physically based
  - InfoVis: Abstract
- **Munzner:**
  - SciVis: Spatial layout given
  - InfoVis: Spatial layout chosen
- **Tory & Möller:**
  - SciVis: Spatial layout given + Continuous
  - InfoVis: Spatial layout chosen + Discrete
  - Everything else -- ?

Overview

- What is SciVis?
- **Data & Applications**
- Iso-surfaces
- Direct Volume Rendering
- Vector Visualization
- Challenges

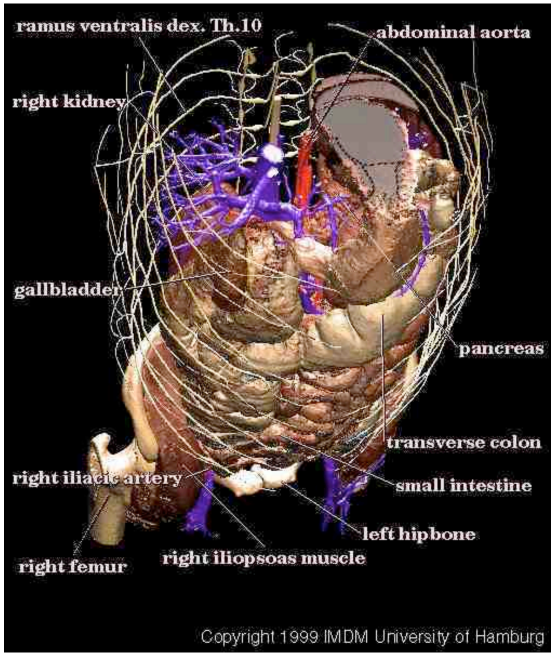
Medical Scanning

- MRI, CT, SPECT, PET, ultrasound



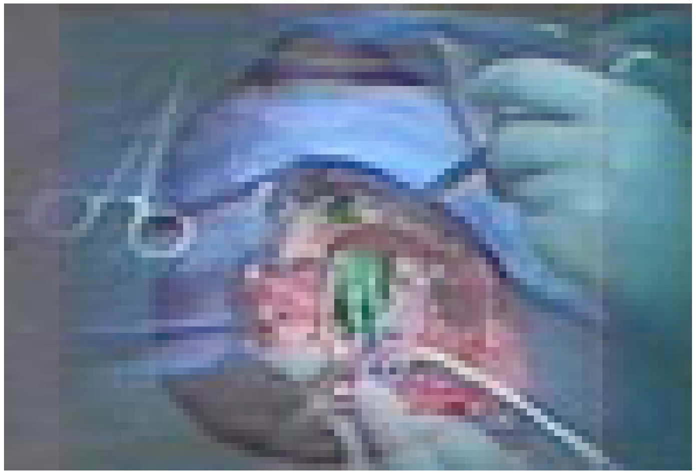
Medical Scanning - Applications

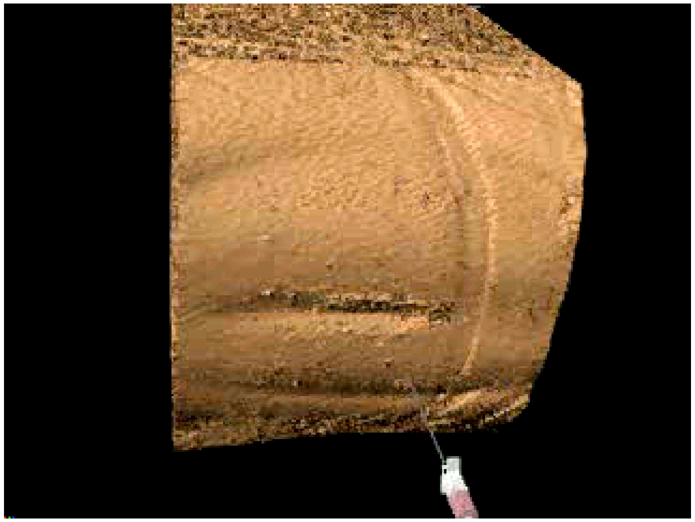
- Medical education for anatomy, surgery, etc.
- Illustration of medical procedures to the patient



Medical Scanning - Applications

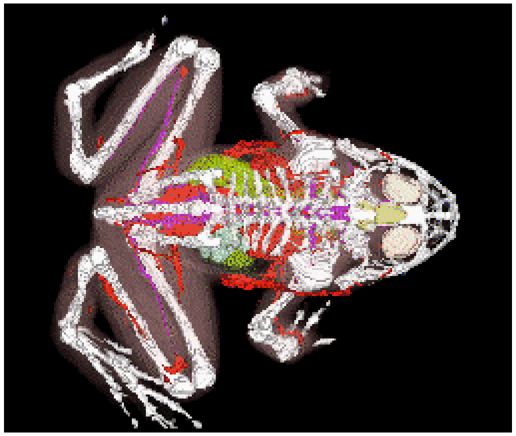
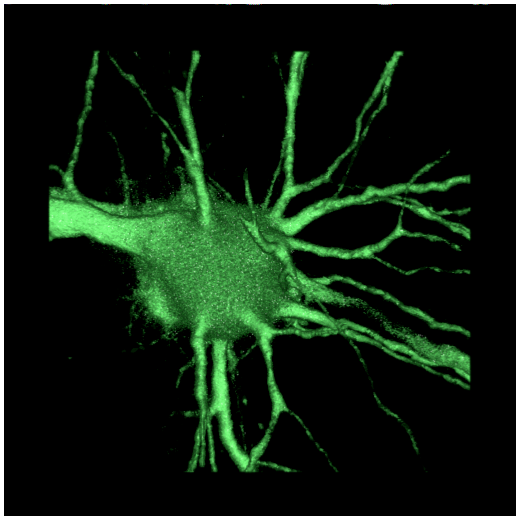
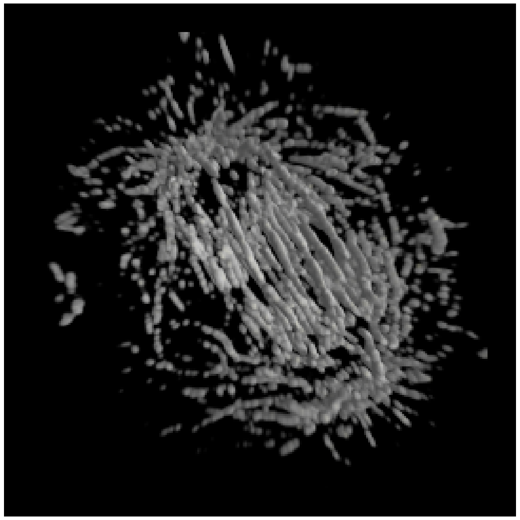
- Surgical simulation for treatment planning
- Tele-medicine
- Inter-operative visualization in brain surgery, biopsies, etc.





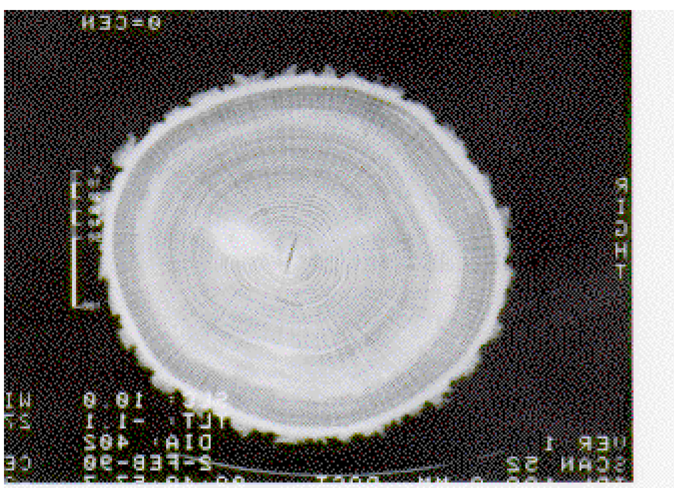
Biological Scanning

- Scanners: Biological scanners, electronic microscopes, confocal microscopes
- Apps – physiology, paleontology, microscopic analysis…

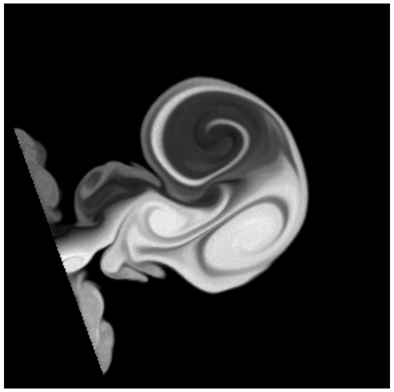
Industrial Scanning

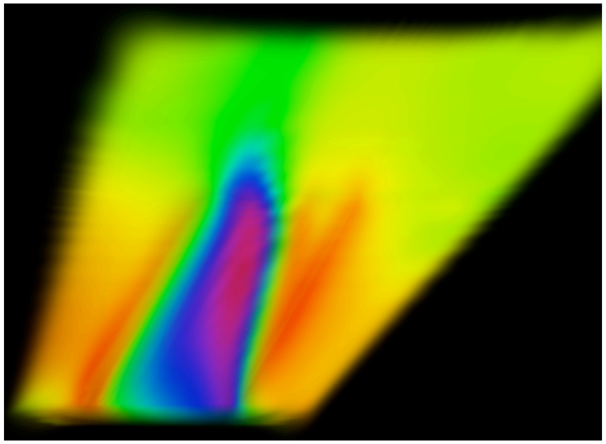
- Planning (e.g., log scanning)
- Quality control
- Security (e.g. airport scanners)



Scientific Computation - Domain

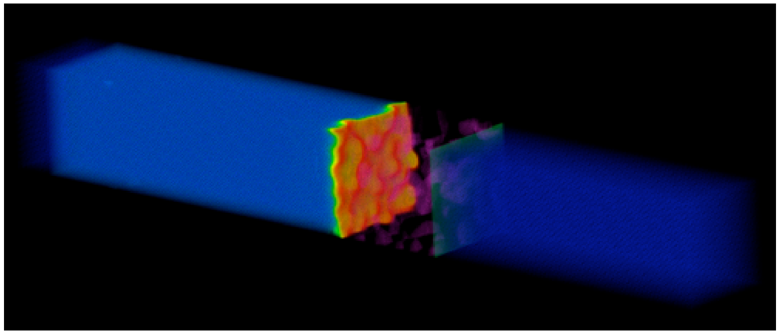
- Mathematical analysis
- ODE/PDE (ordinary and partial differential equations)
- Finite element analysis (FE)
- Supercomputer simulations

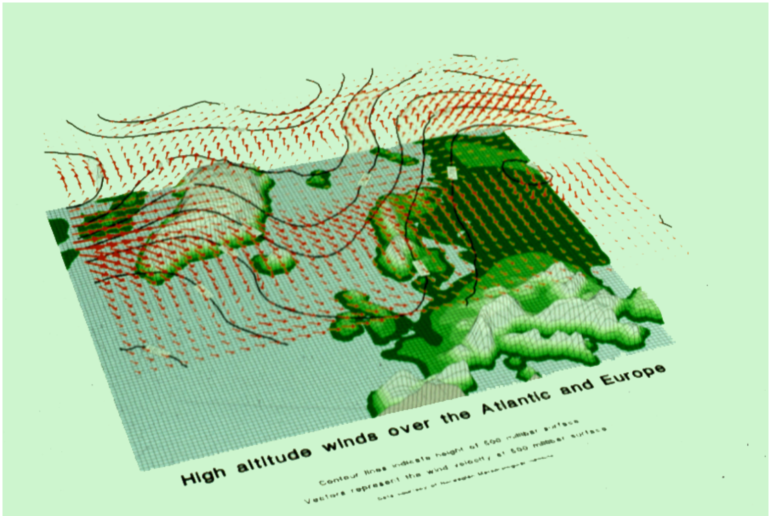




Scientific Computation - Apps

- Flow Visualization



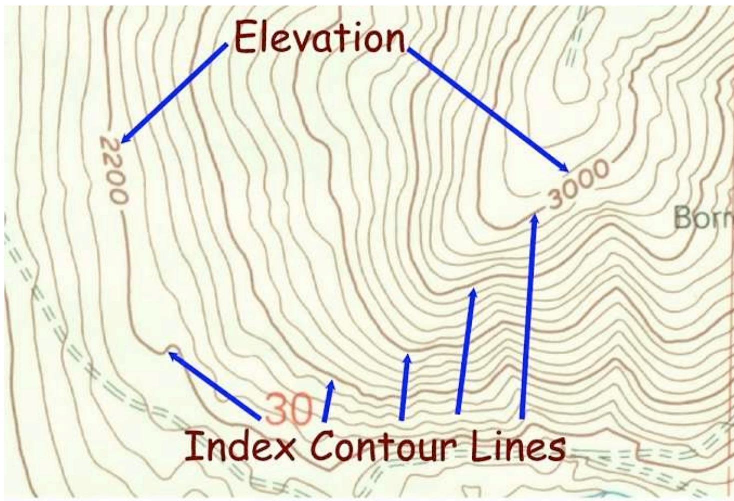


Overview

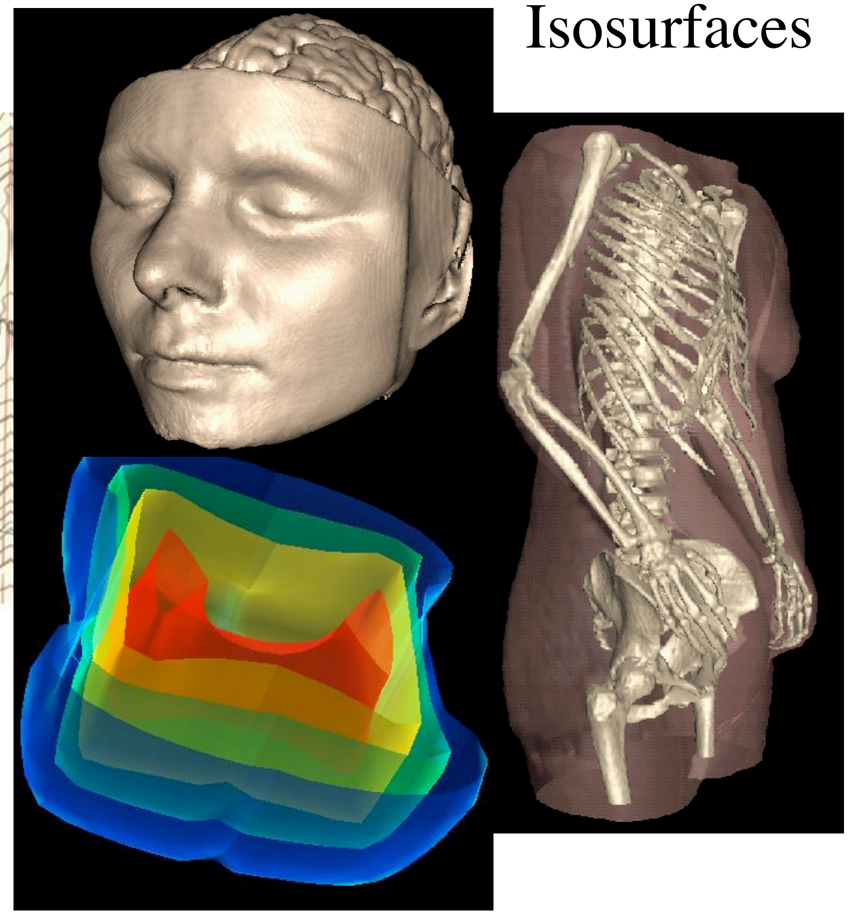
- What is SciVis?
- Data & Applications
- Iso-surfaces
- Direct Volume Rendering
- Vector Visualization
- Challenges

Isosurfaces - Examples

Isolines

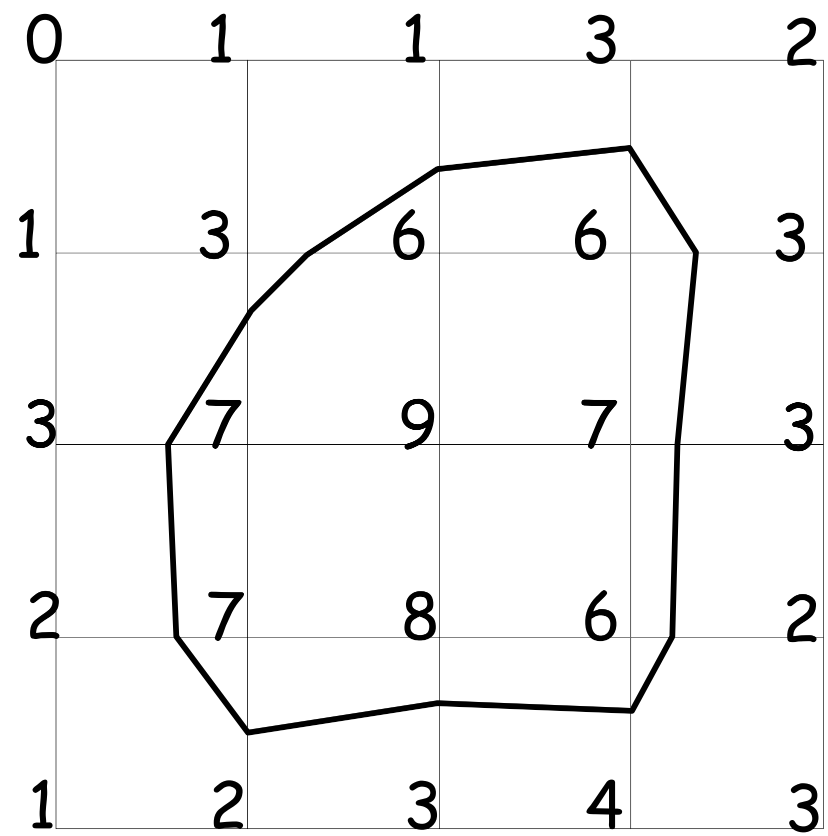


Isosurfaces



Isosurface Extraction

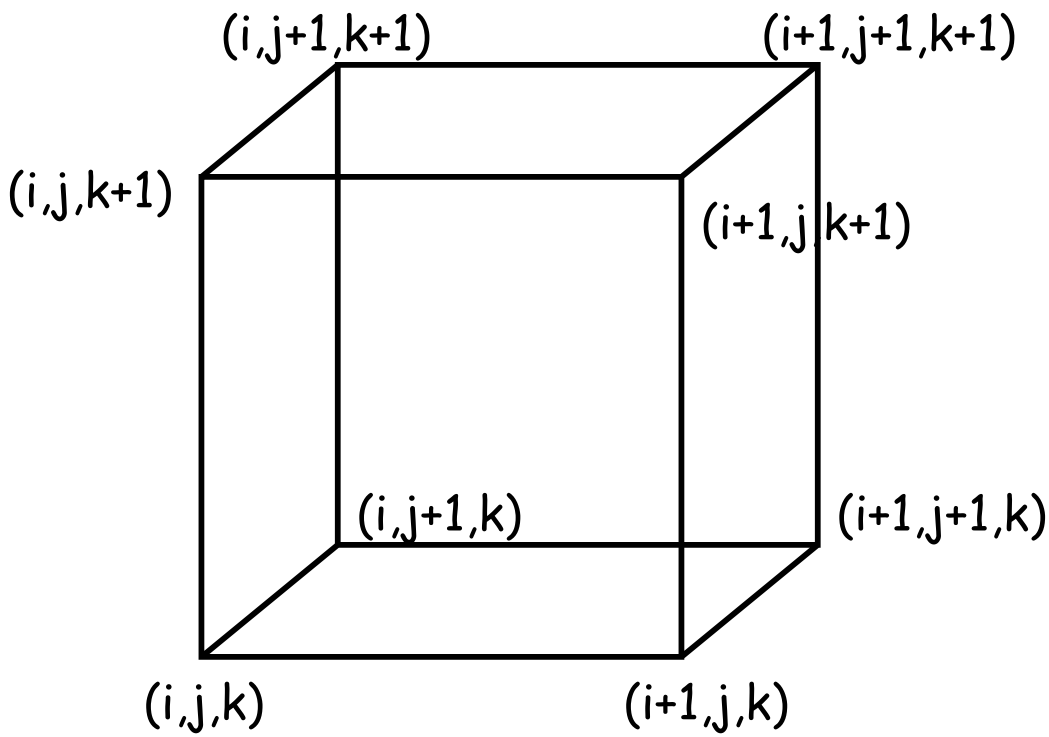
- by contouring
  - closed contours
  - continuous
  - determined by iso-value
- several methods
  - marching cubes is most common



Iso-value = 5

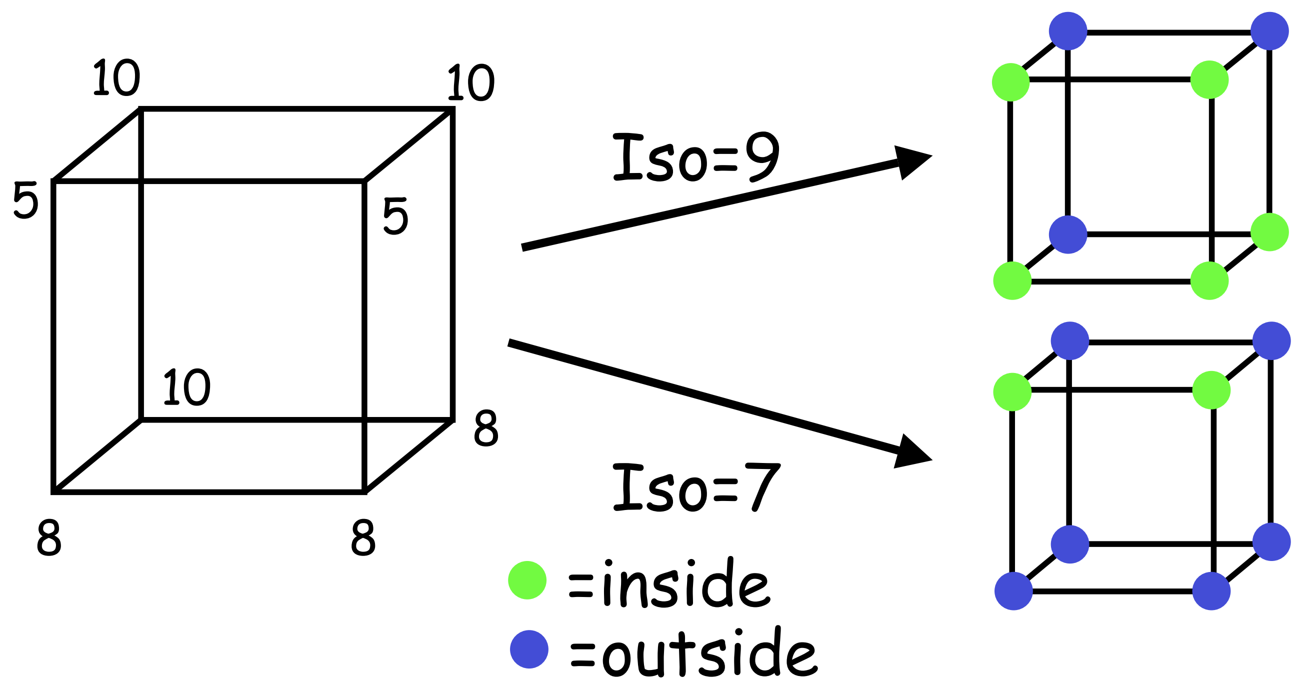
MC 1: Create a Cube

- Consider a Cube defined by eight data values:



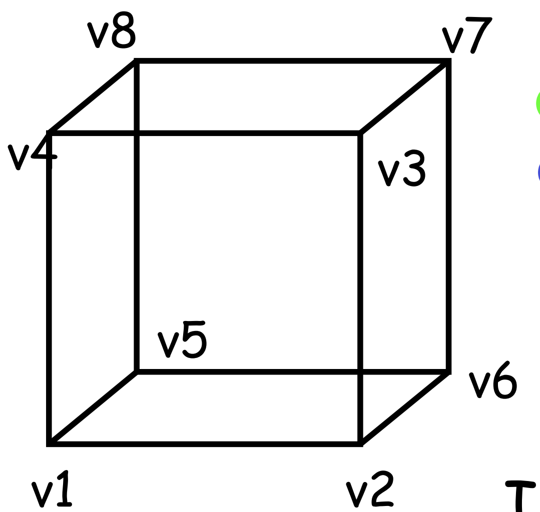
MC 2: Classify Each Voxel

- Classify each voxel according to whether it lies outside the surface (value > iso-surface value) inside the surface (value <= iso-surface value)

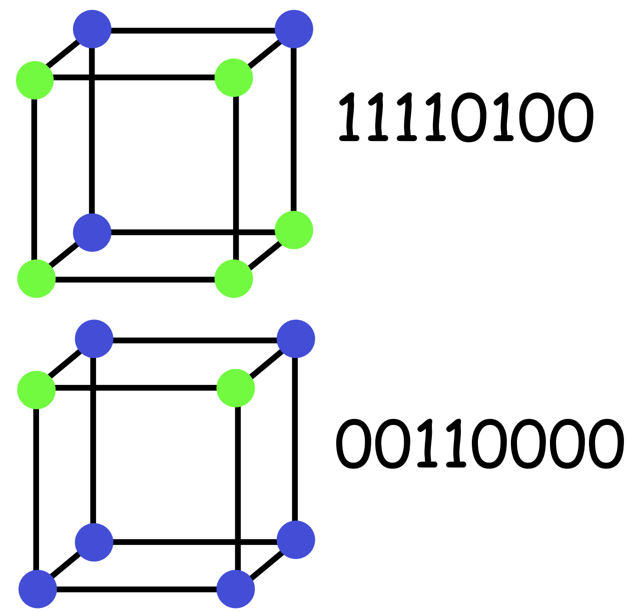


MC 3: Build An Index

- Use the binary labeling of each voxel to create an index



- inside =1
- outside=0

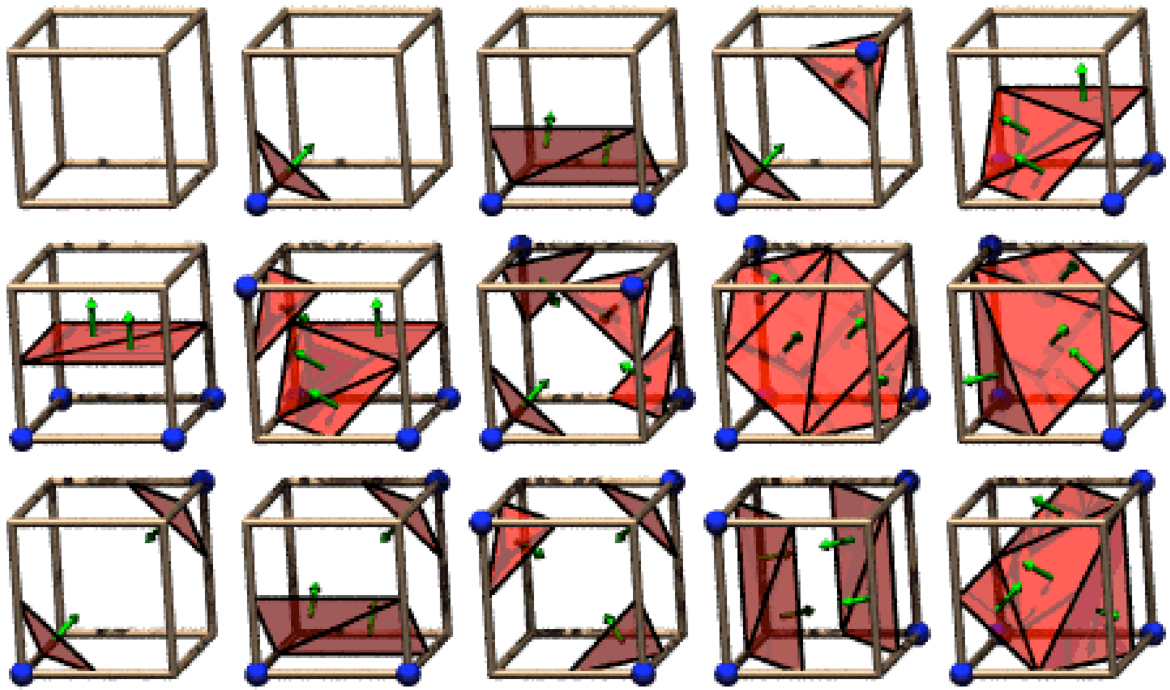


Index:

v1	v2	v3	v4	v5	v6	v7	v8

MC 4: Lookup Edge List

- For a given index, access an array storing a list of edges

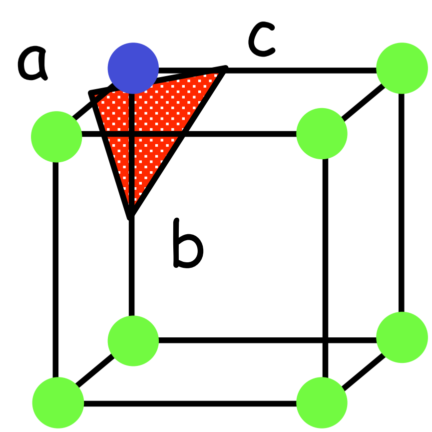


The 15 Cube Combinations

- all 256 cases can be derived from 15 base cases

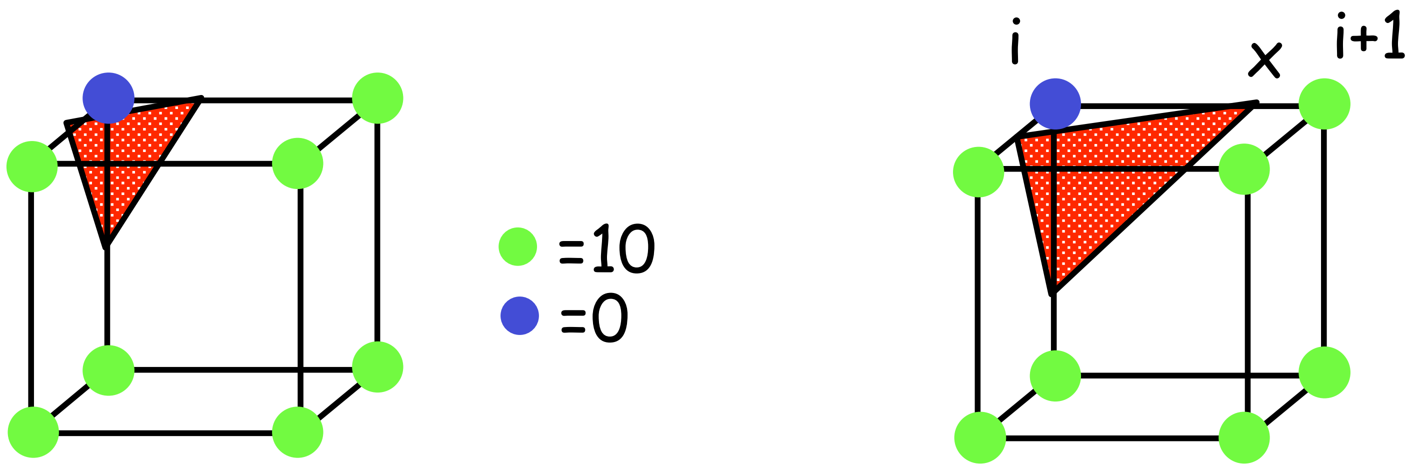
MC 4: Example

- Index = 00000001
- triangle 1 = a, b, c



MC 5: Interp. Triangle Vertex

- For each triangle edge, find the vertex location along the edge using linear interpolation of the voxel values



T=5

$$x = i + \left(\frac{T - v[i]}{v[i+1] - v[i]} \right)$$

T=8

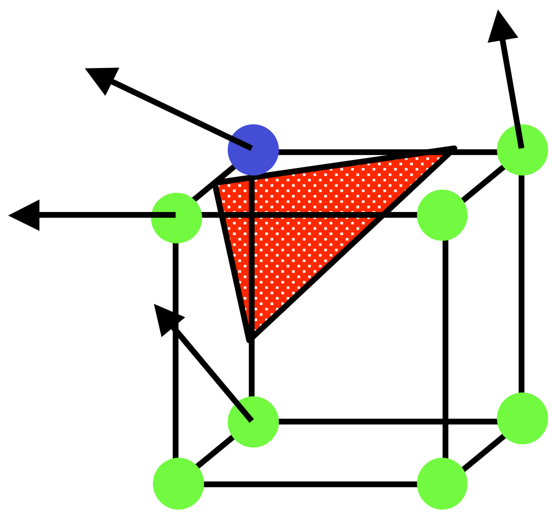
MC 6: Compute Normals

- Calculate the normal at each cube vertex

$$G_x = v_{i+1,j,k} - v_{i-1,j,k}$$

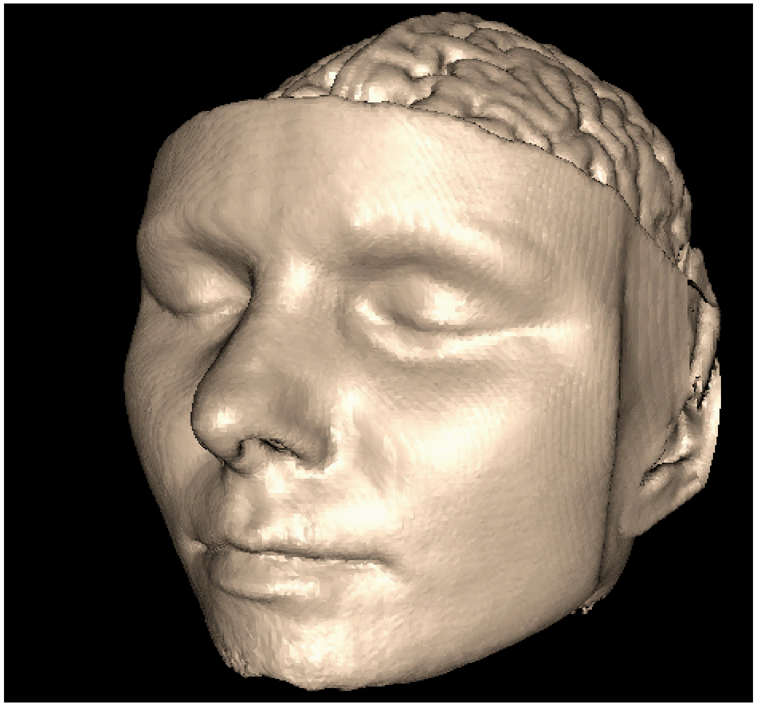
$$G_y = v_{i,j+1,k} - v_{i,j-1,k}$$

$$G_z = v_{i,j,k+1} - v_{i,j,k-1}$$



- Use linear interpolation to compute the polygon vertex normal

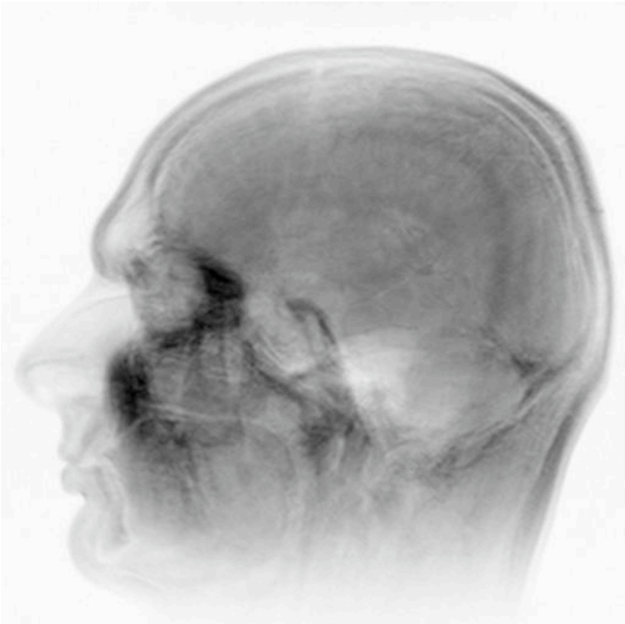
MC 7: Render!

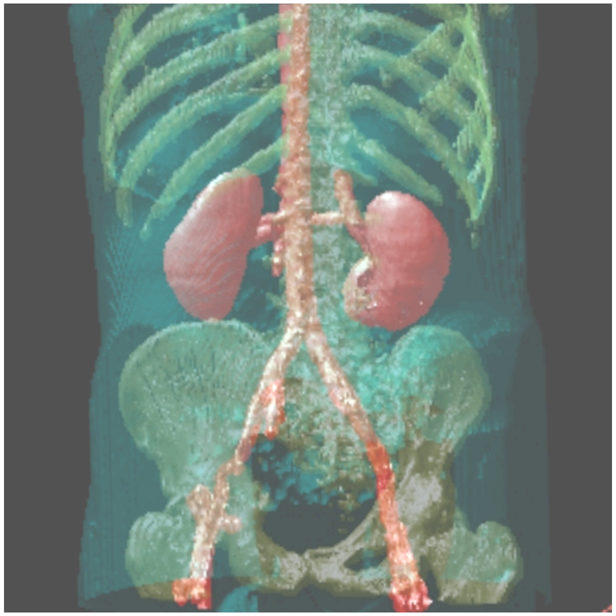


Overview

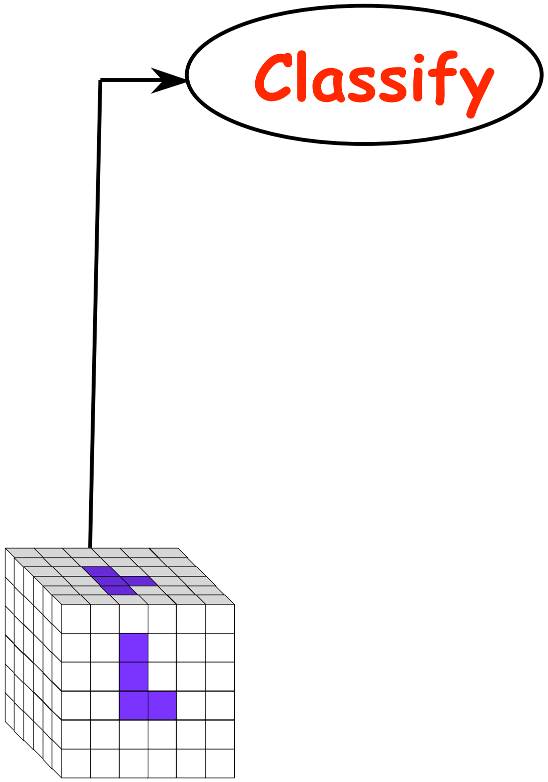
- What is SciVis?
- Data & Applications
- Iso-surfaces
- **Direct Volume Rendering**
- Vector Visualization
- Challenges

Direct Volume Rendering Examples



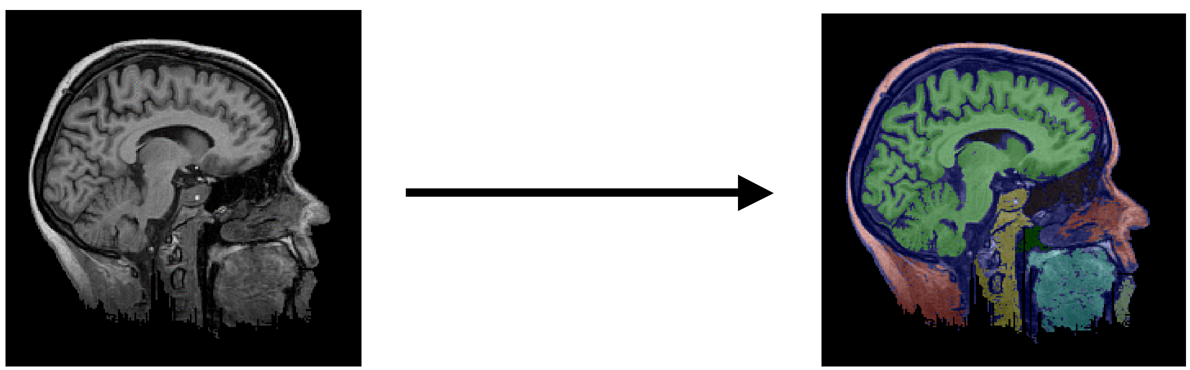


Rendering Pipeline (RP)



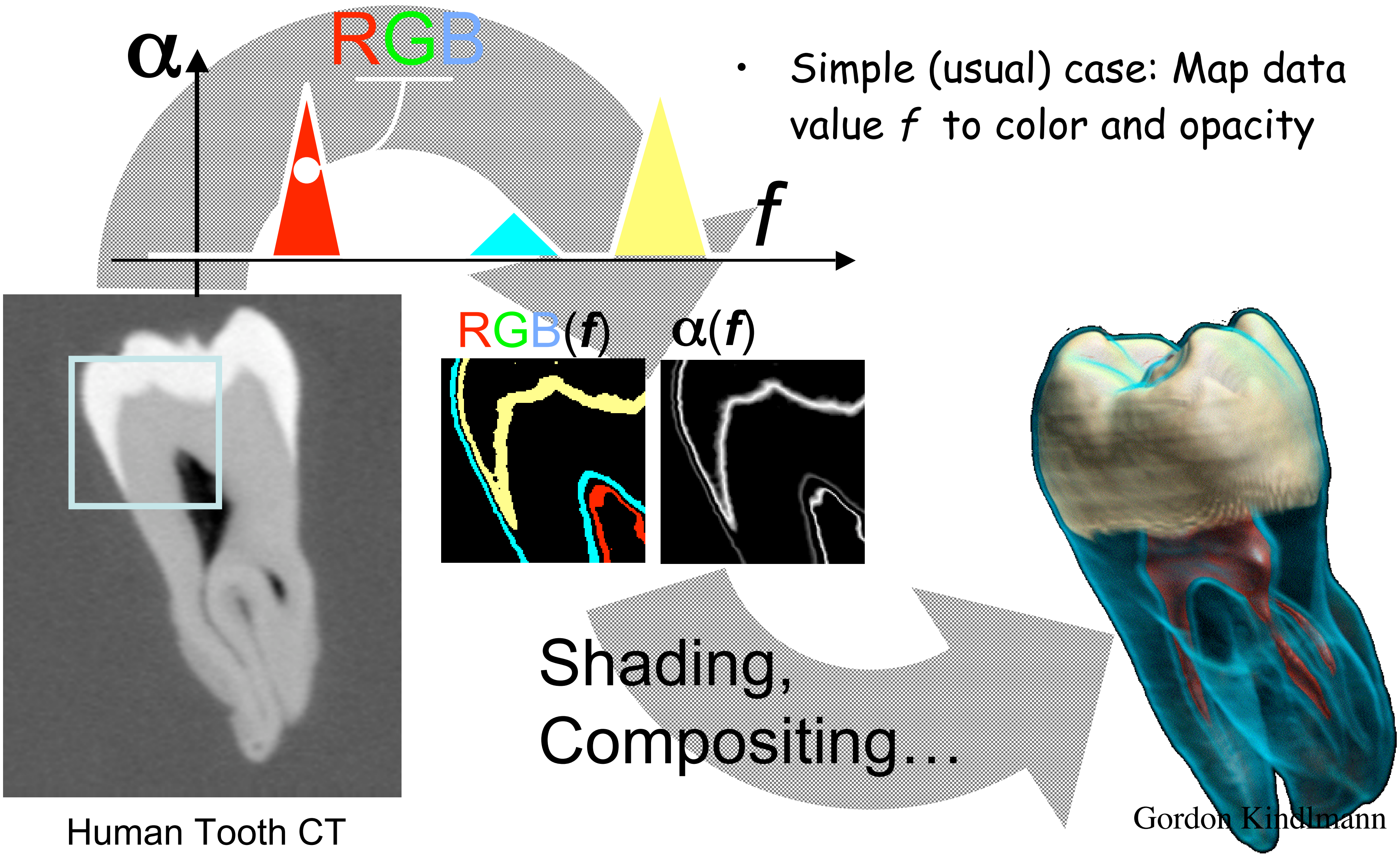
Classification

- original data set has application specific values (temperature, velocity, proton density, etc.)
- assign these to color/opacity values to make sense of data
- achieved through transfer functions



Transfer Functions (TF's)

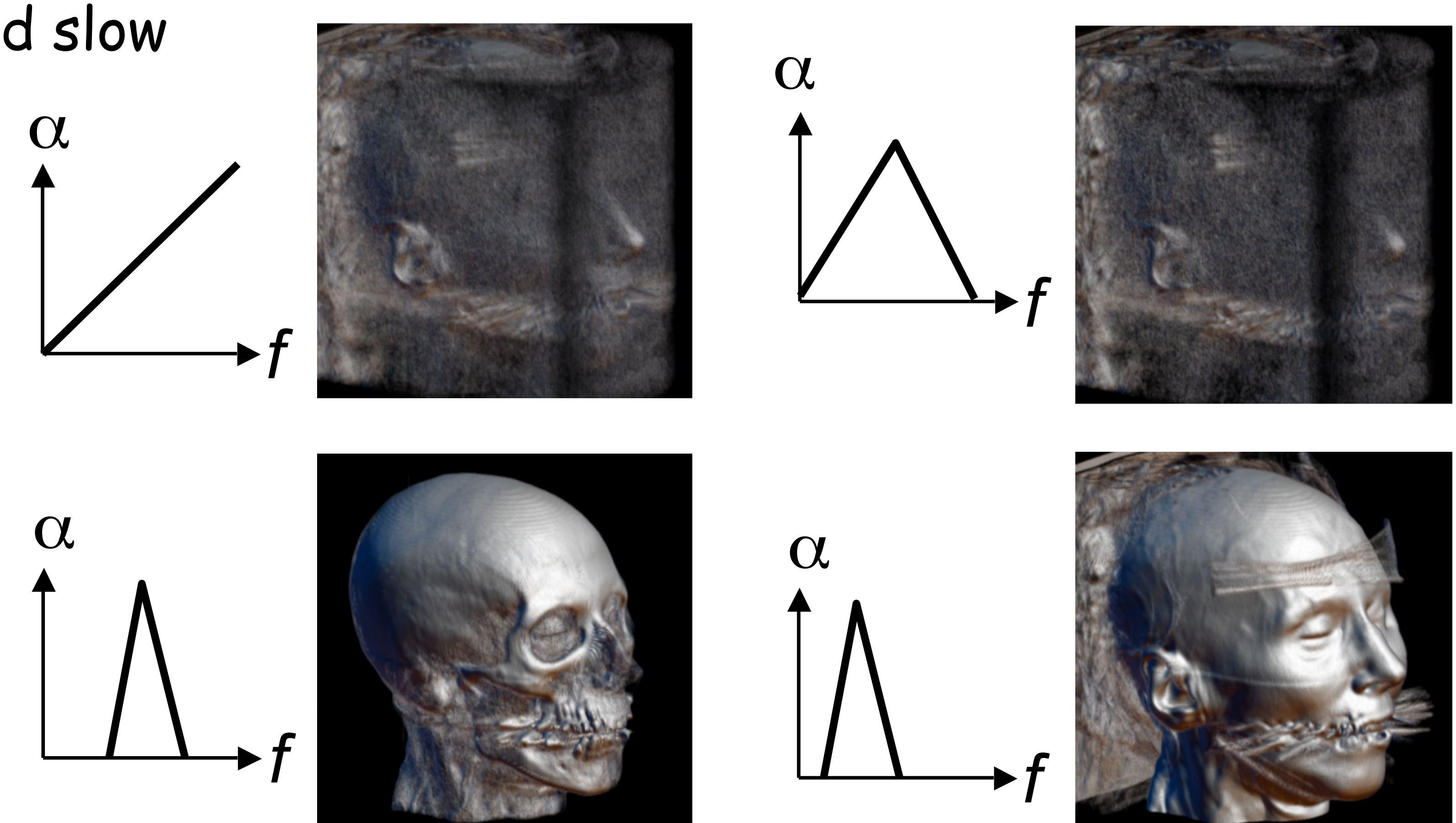
- Simple (usual) case: Map data value *f* to color and opacity



Human Tooth CT

TF's

- Setting transfer functions is difficult, unintuitive, and slow



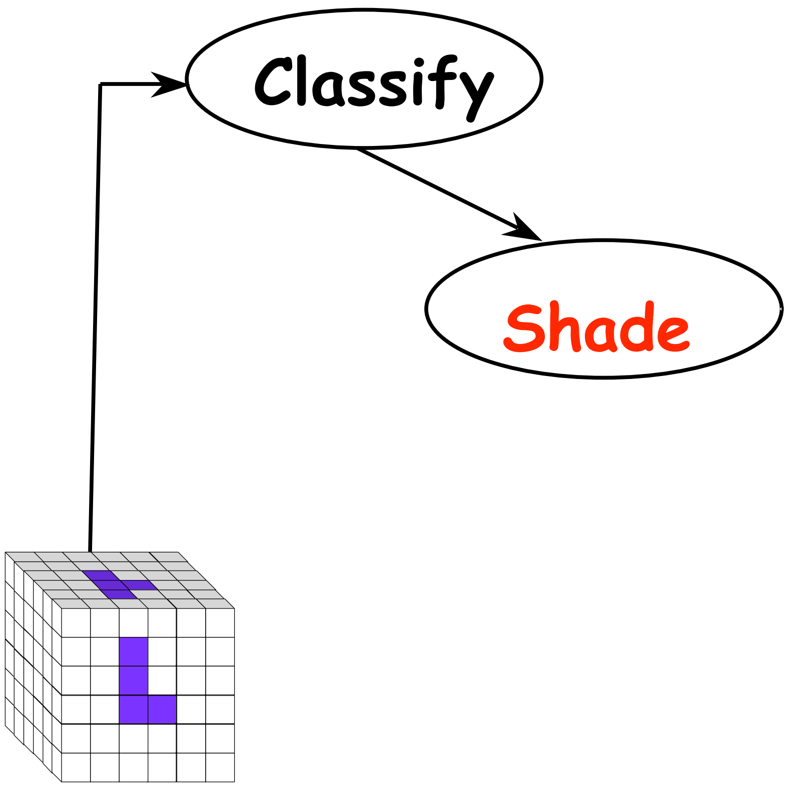
Gordon Kindlmann

Transfer Function Challenges

- Better interfaces:
  - Make space of TFs less confusing
  - Remove excess "flexibility"
  - Provide guidance
- Automatic / semi-automatic transfer function generation
  - Typically highlight boundaries

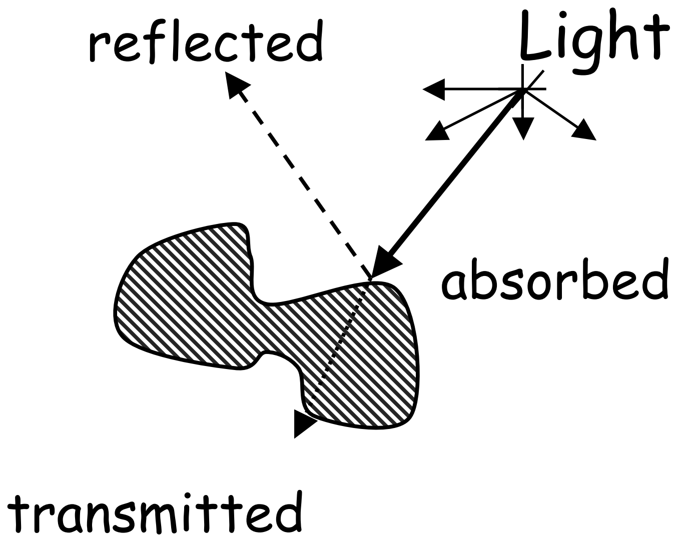
Gordon Kindlmann

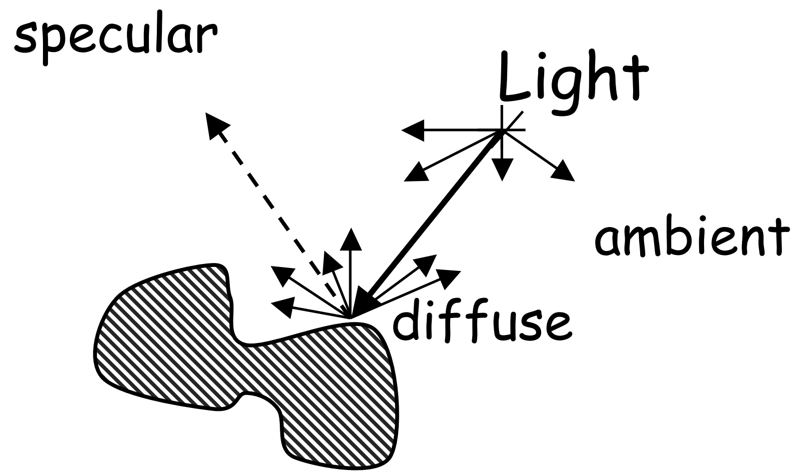
Rendering Pipeline (RP)



Light Effects

- Usually only considering reflected part



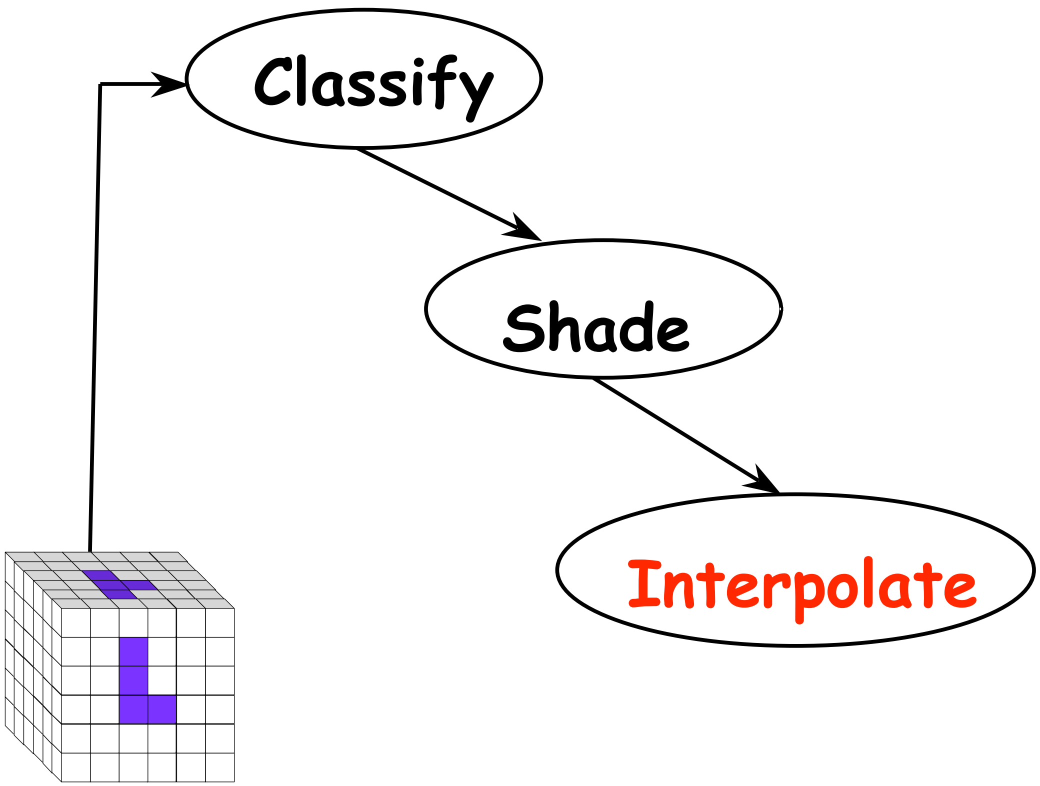


Light=refl.+absorbed+trans.

Light=ambient+diffuse+specular

$$I = k_a I_a + k_d I_d + k_s I_s$$

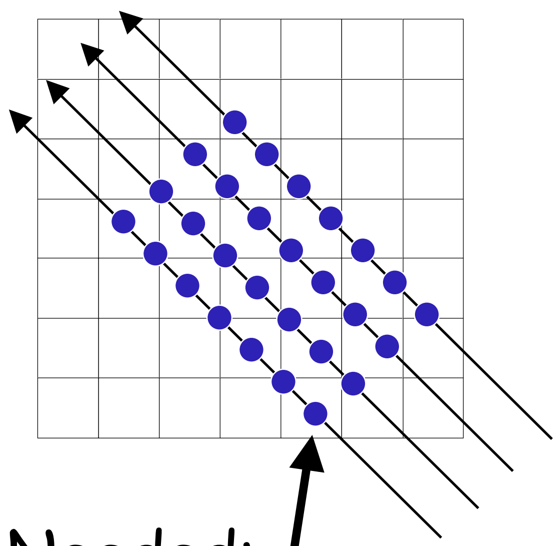
Rendering Pipeline (RP)



Interpolation

2D

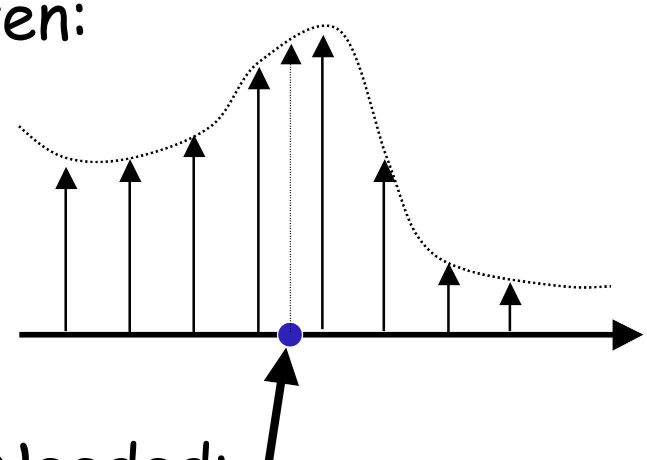
- Given:



- Needed:

1D

- Given:



- Needed:

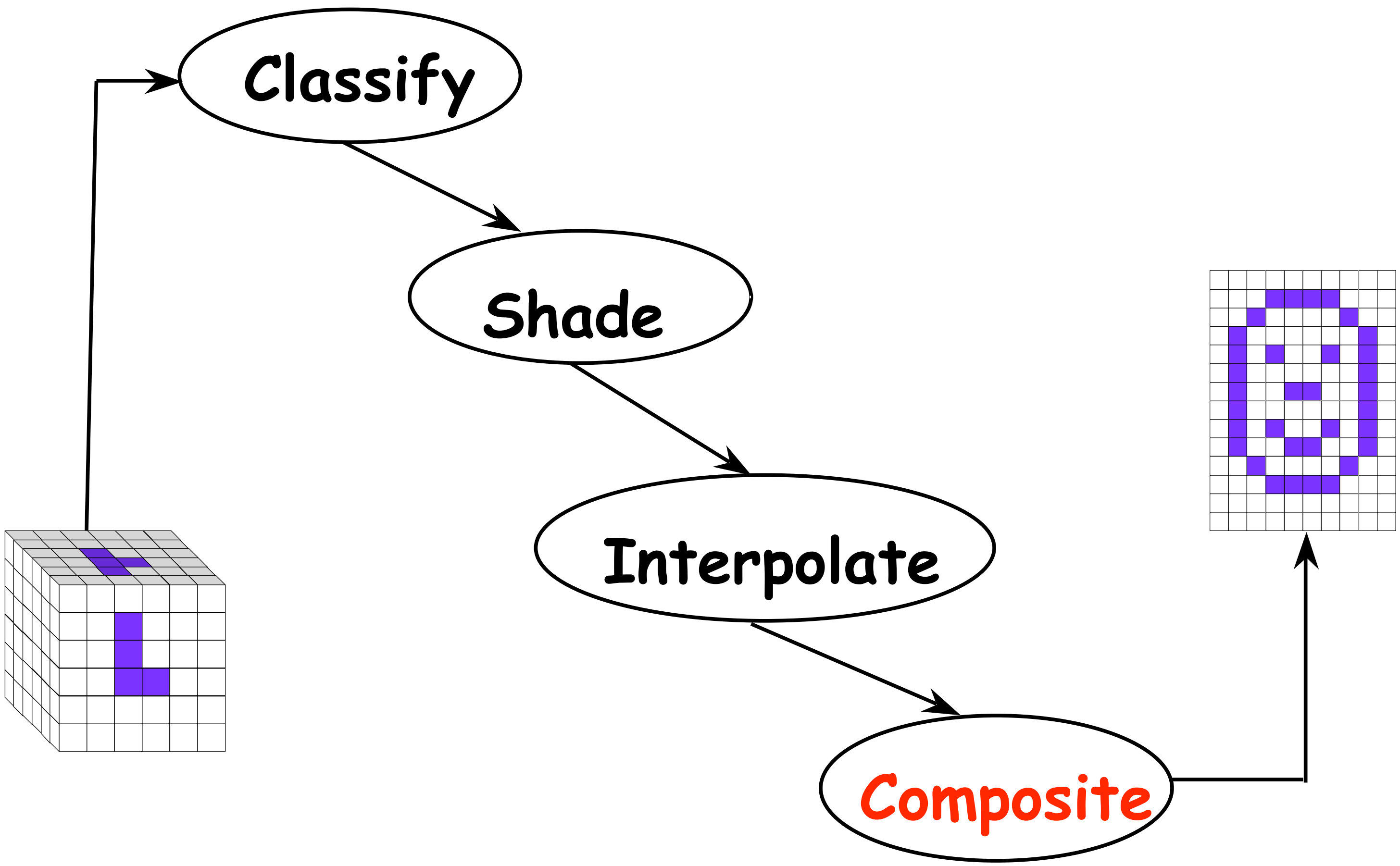
Interpolation

- Very important; regardless of algorithm
- Expensive => done very often for one image
- Requirements for good reconstruction
  - performance
  - stability of the numerical algorithm
  - accuracy

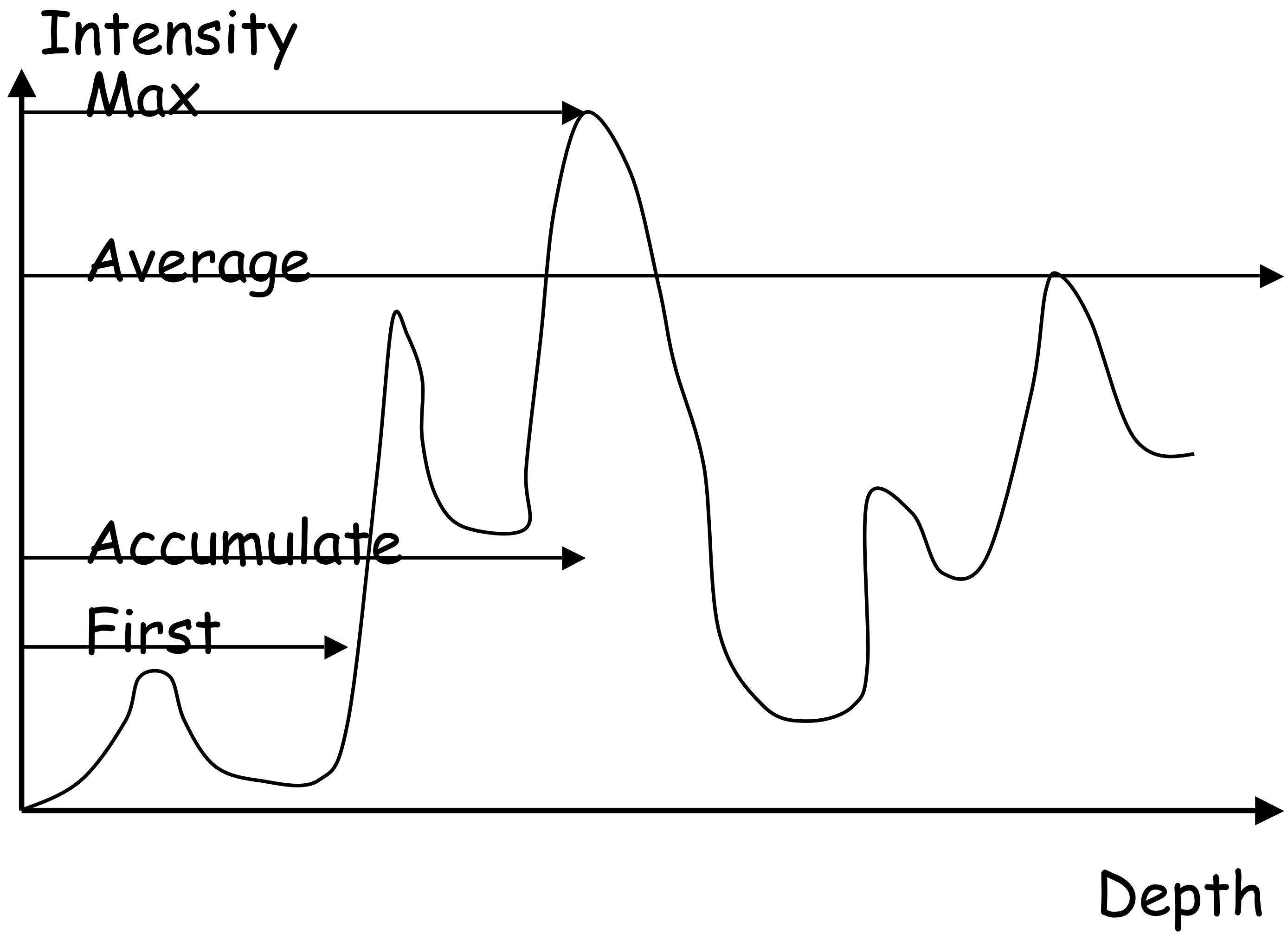
Nearest neighbor

Linear

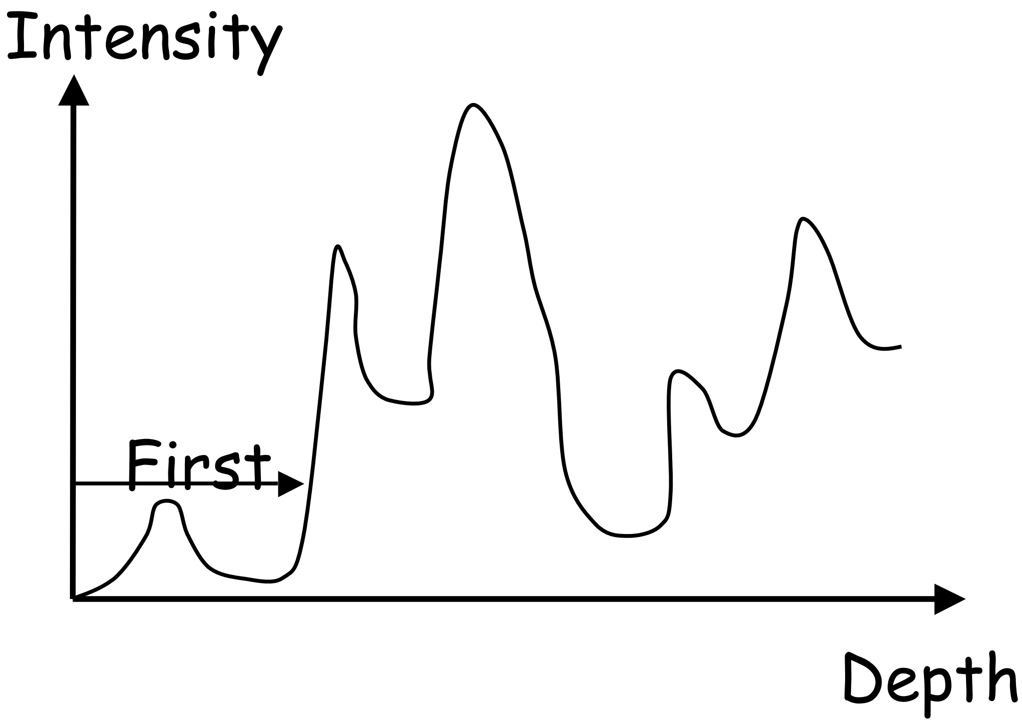
Rendering Pipeline (RP)

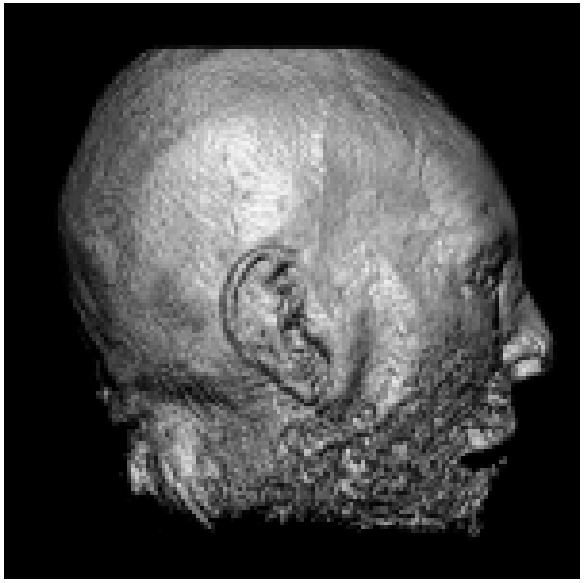


Ray Traversal Schemes



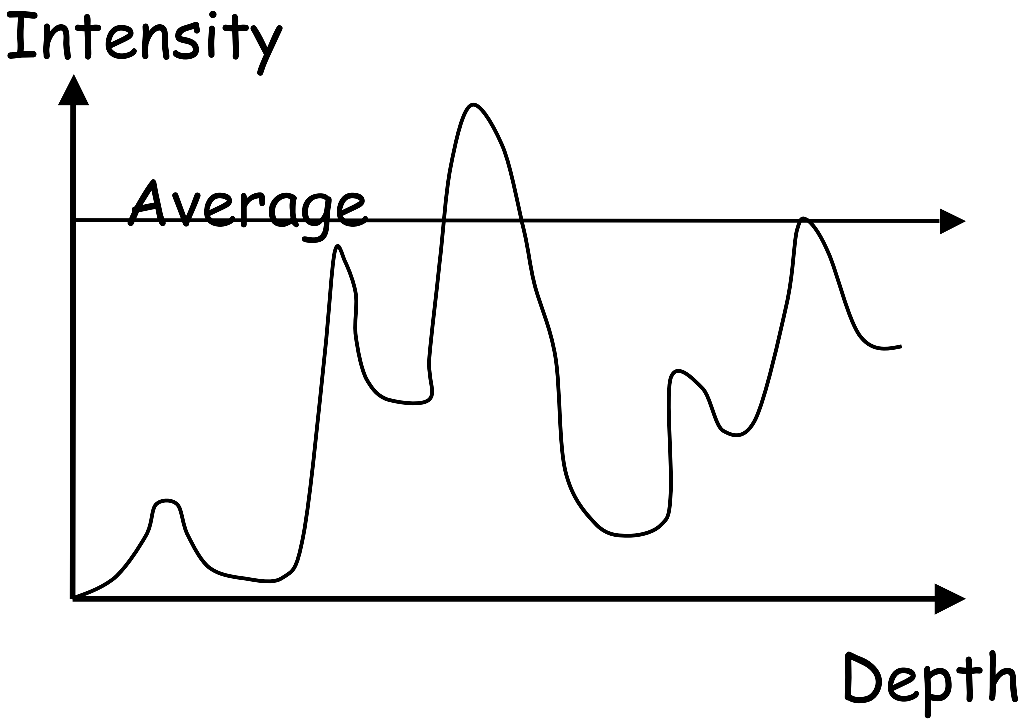
Ray Traversal - First

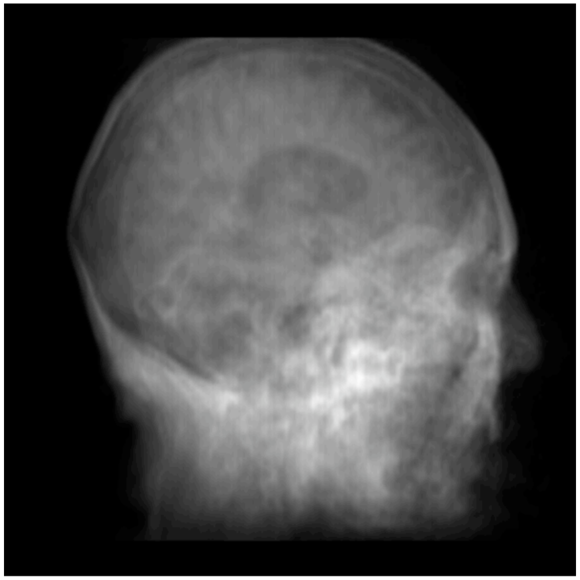




- **First**: extracts iso-surfaces (again!) done by Tuy&Tuy '84

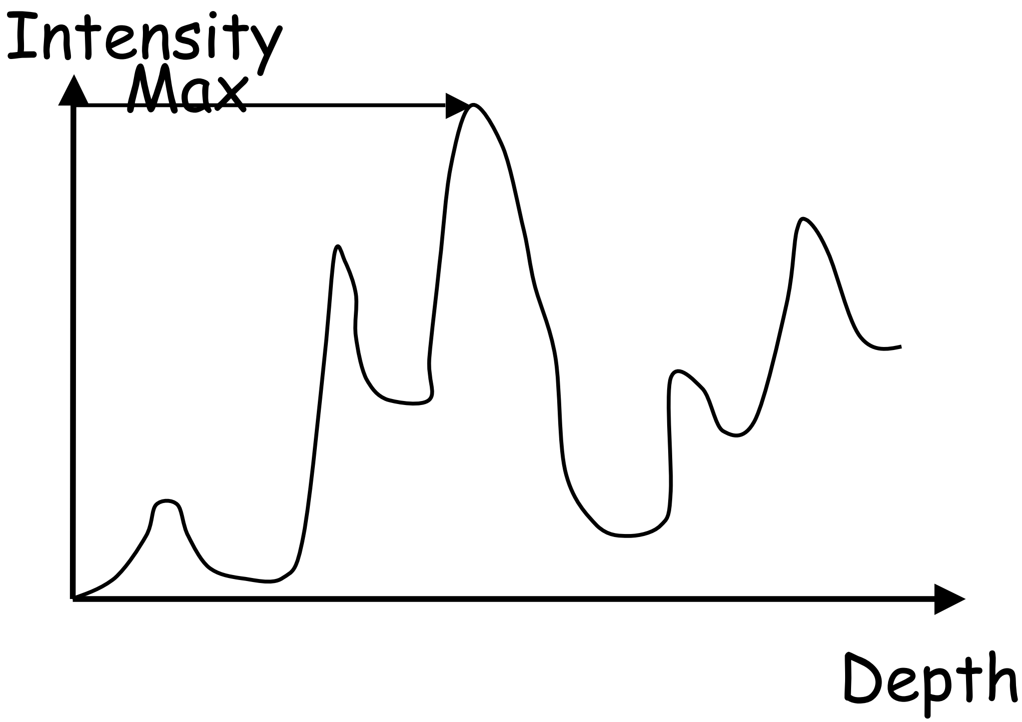
Ray Traversal - Average

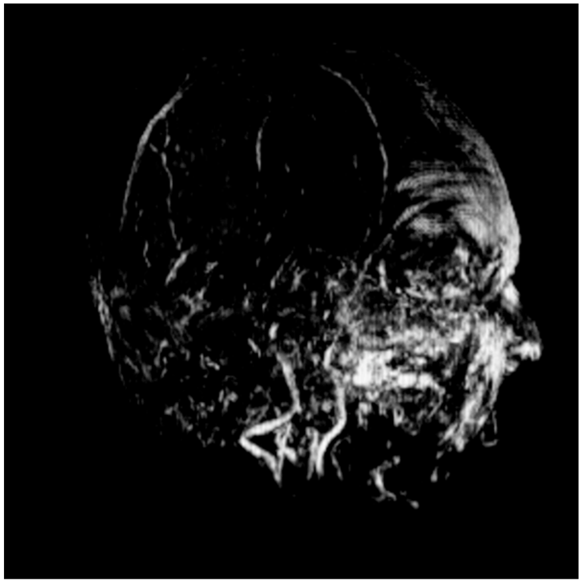




- **Average**: produces basically an X-ray picture

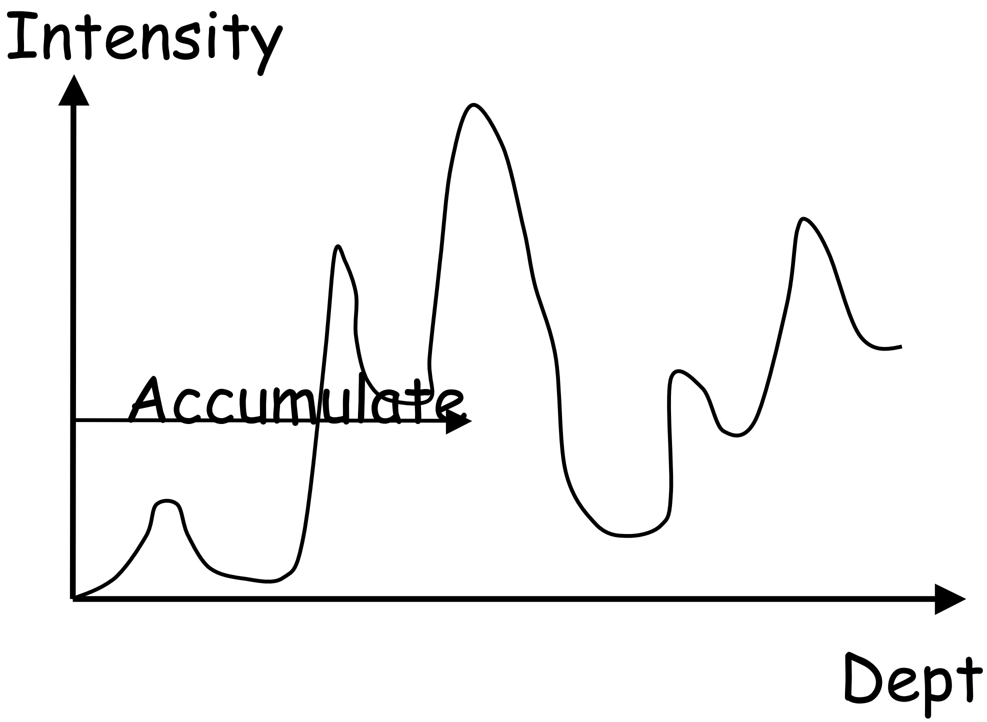
Ray Traversal - MIP

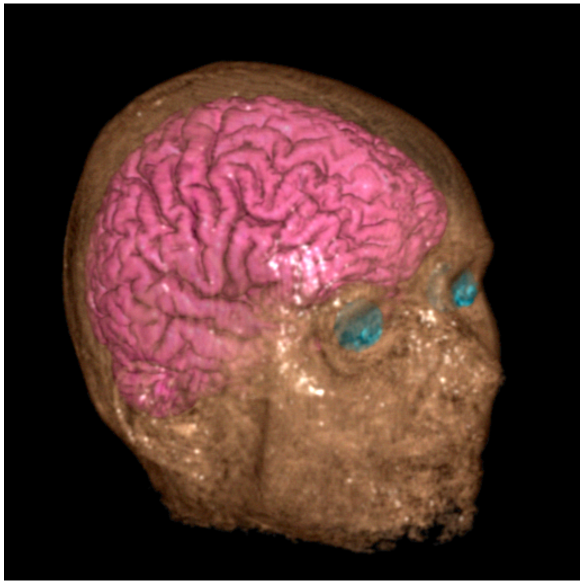




- **Max**: Maximum Intensity Projection
  used for Magnetic Resonance Angiogram

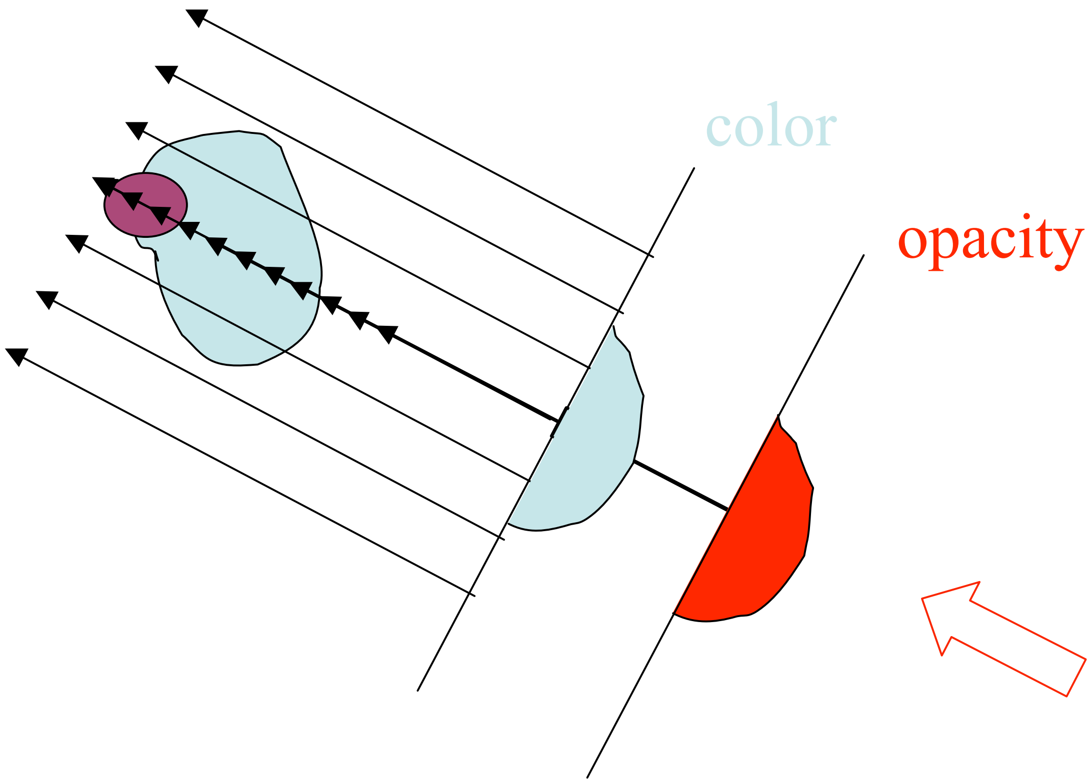
Ray Traversal - Accumulate





- **Accumulate**: make transparent layers visible!
  Levoy '88

Volumetric Ray Integration



object (color, opacity)

Overview

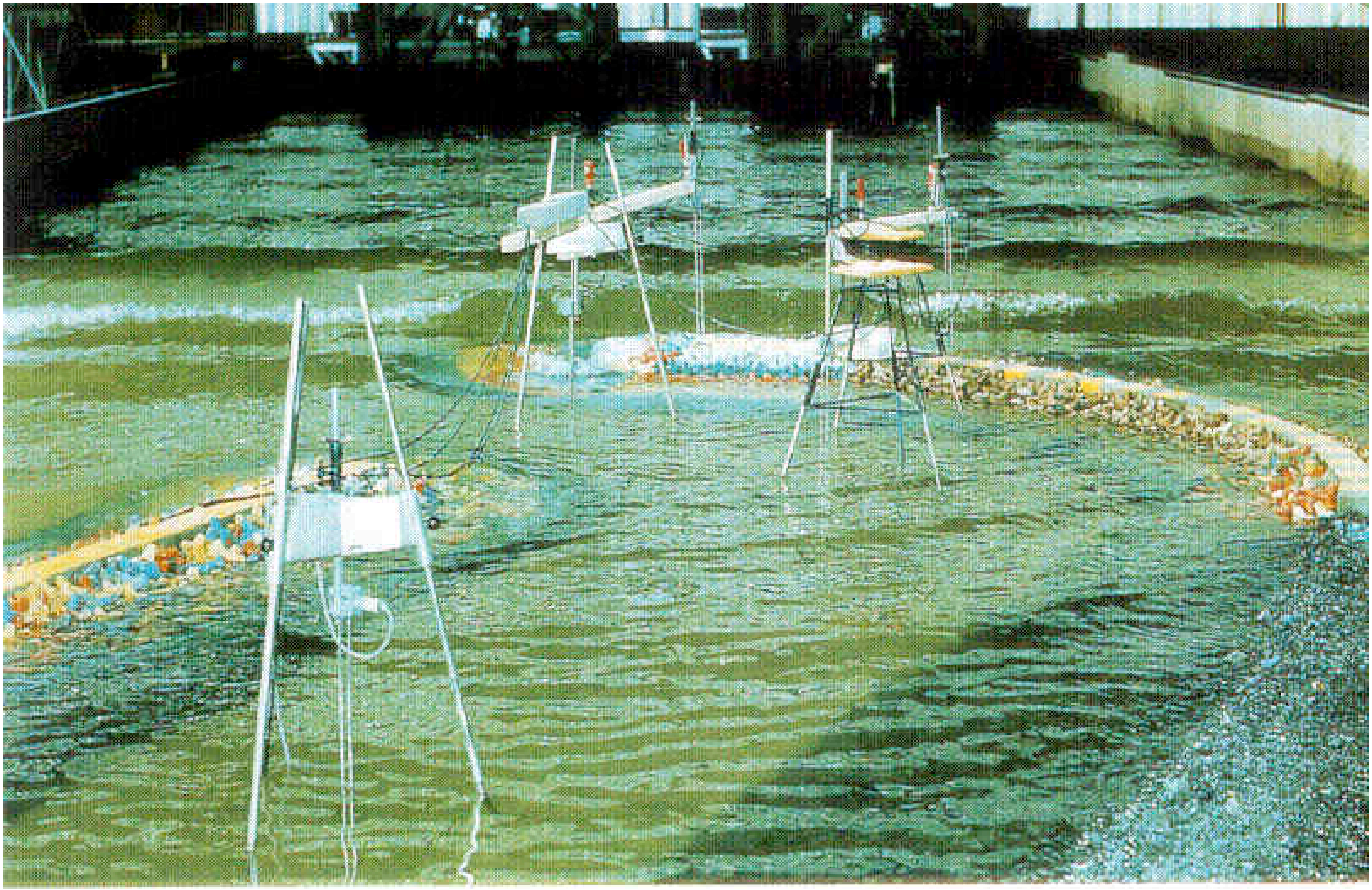
- What is SciVis?
- Data & Applications
- Iso-surfaces
- Direct Volume Rendering
- **Vector Visualization**
- Challenges

Flow Visualization

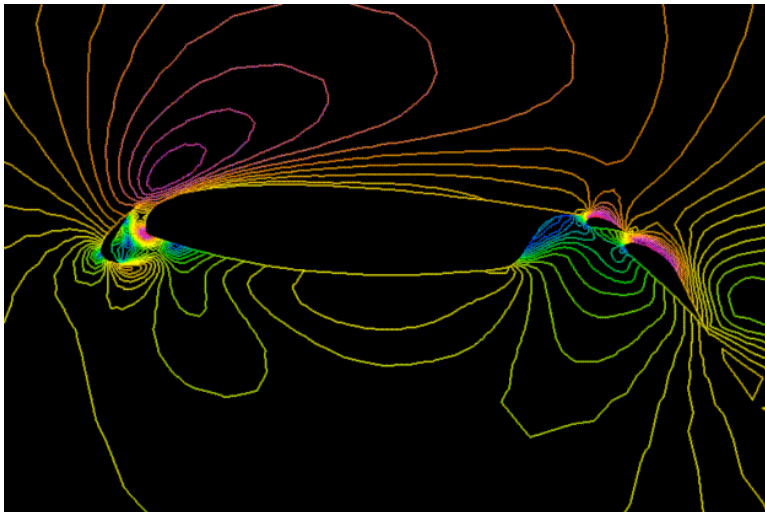
- Traditionally – Experimental Flow Vis
- Now – Computational Simulation

- Typical Applications:
  - Study physics of fluid flow
  - Design aerodynamic objects

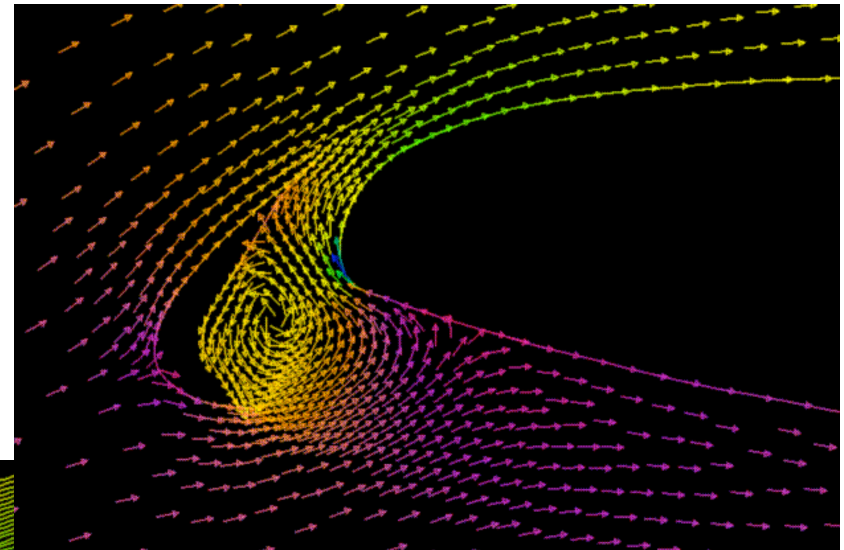
Traditional Flow Experiments



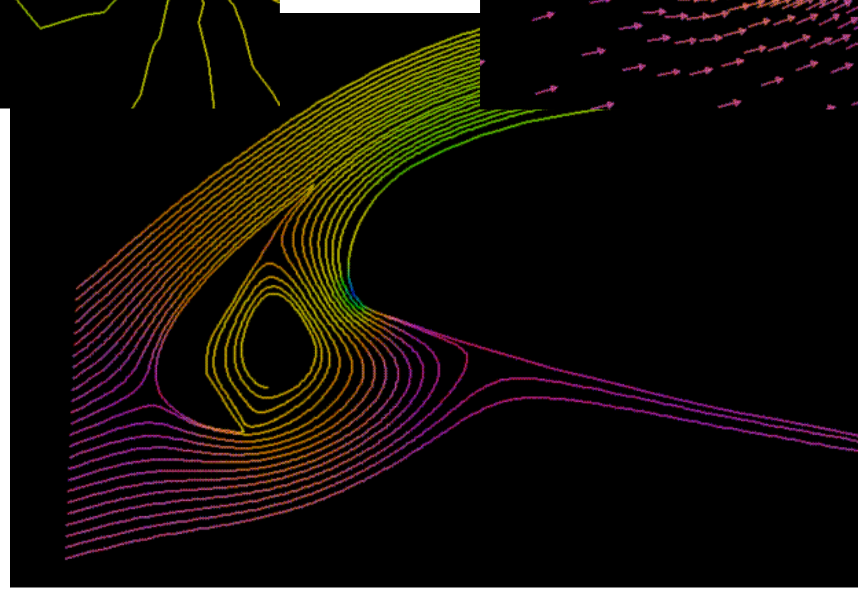
Techniques



Contours

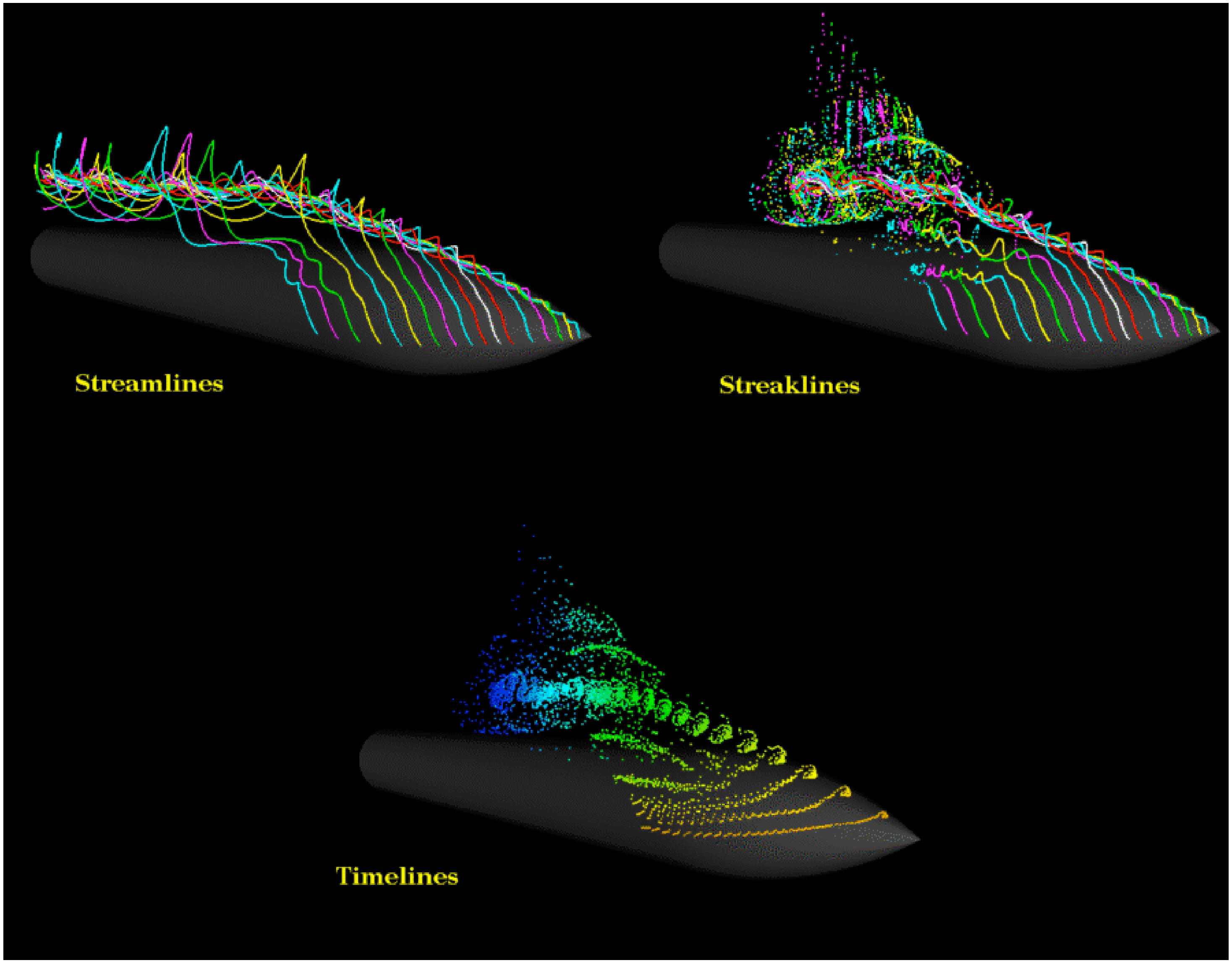


Glyphs (arrows)



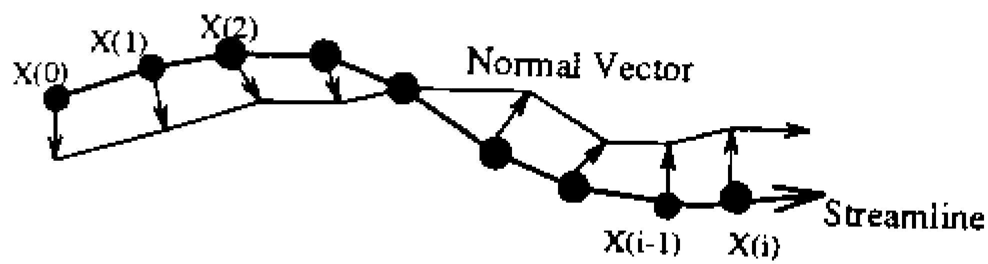
Streamlines

Techniques



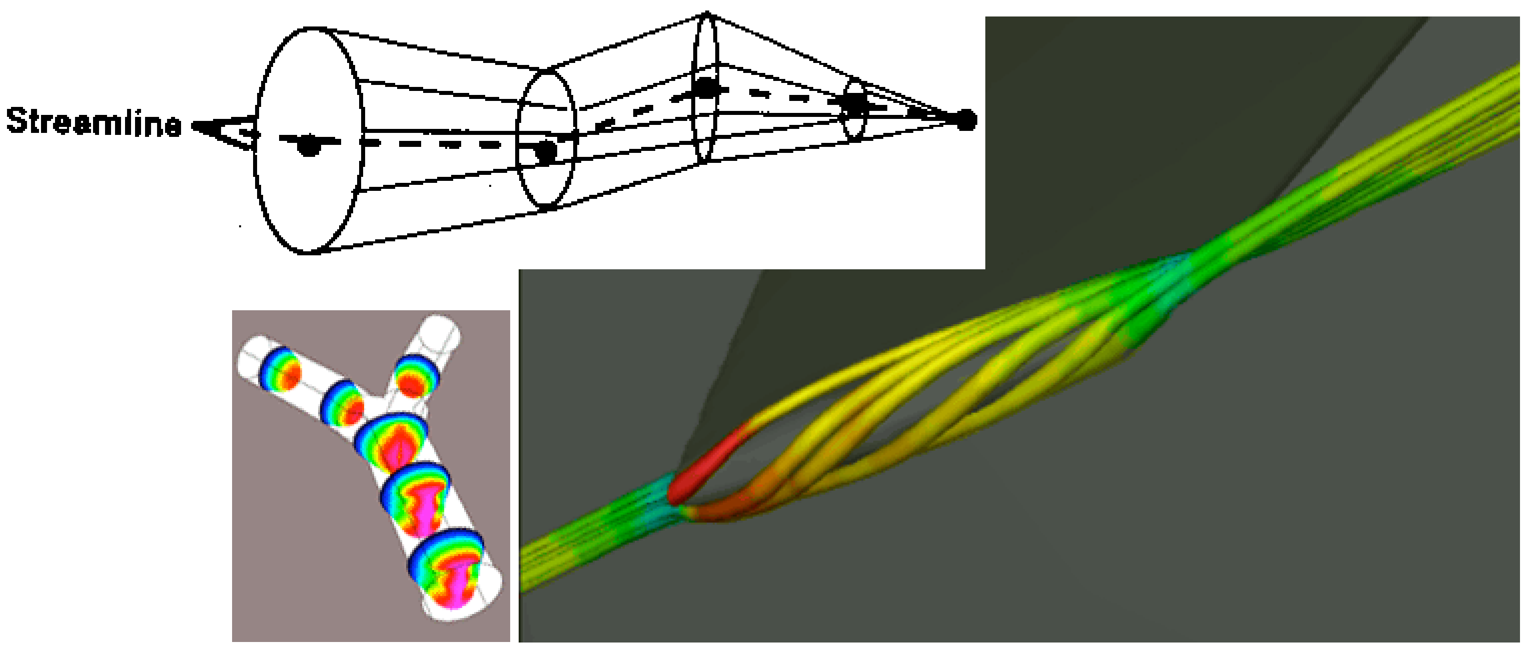
Techniques - Stream-ribbon

- Trace one streamline and a constant size vector with it
- Allows you to see places where flow twists



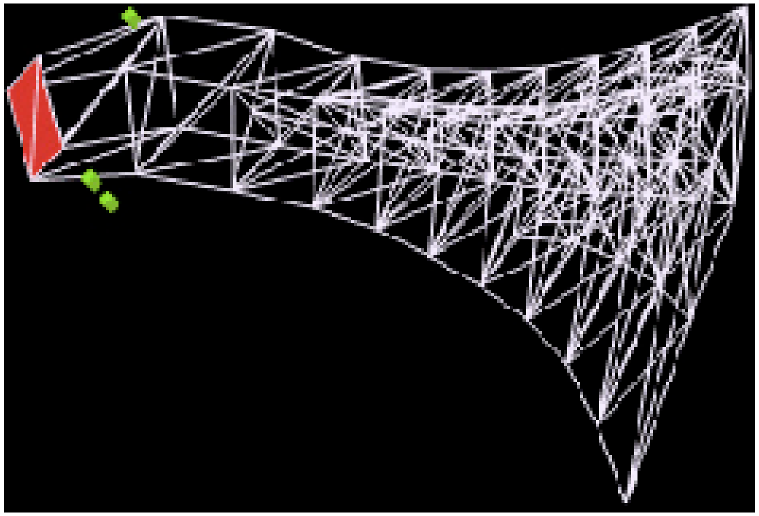
Techniques - Stream-tube

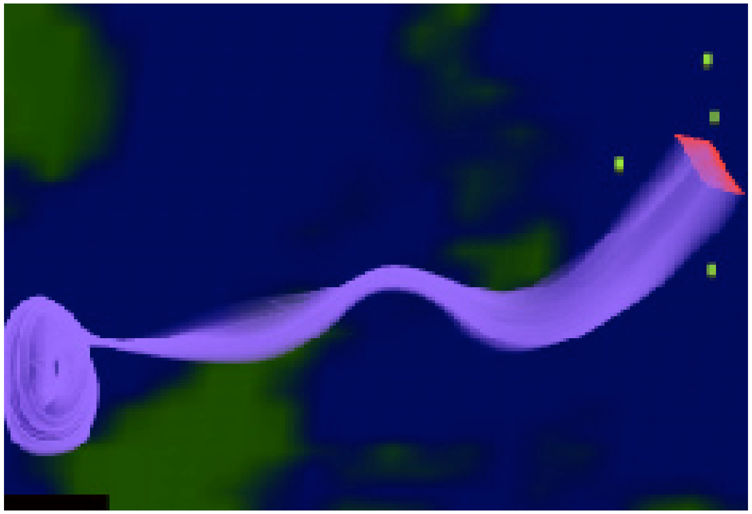
- Generate a stream-line and widen it to a tube
- Width can encode another variable



Mappings - Flow Volumes

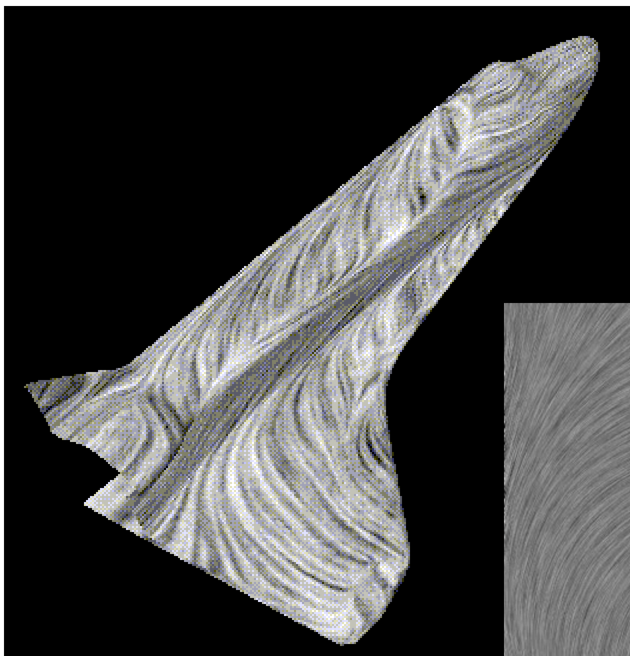
- Instead of tracing a line - trace a small polyhedron

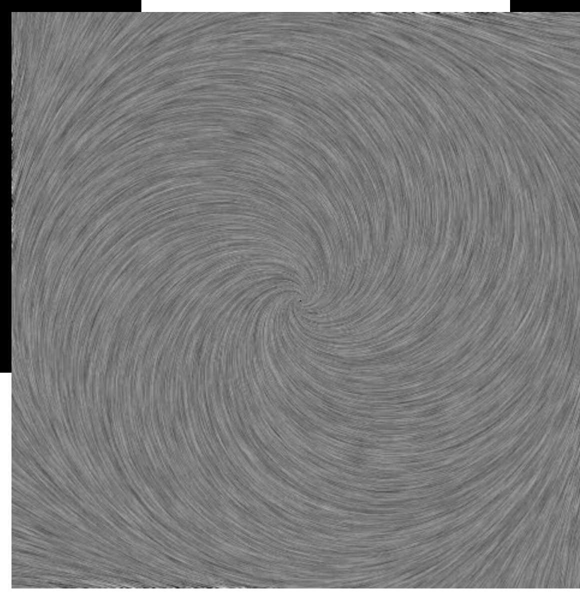


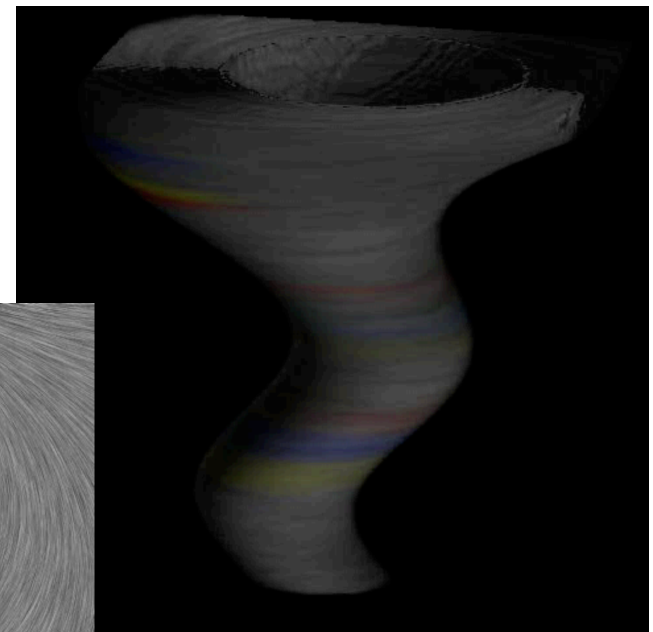


LIC (Line Integral Convolution)

- Integrate noise texture along a streamline







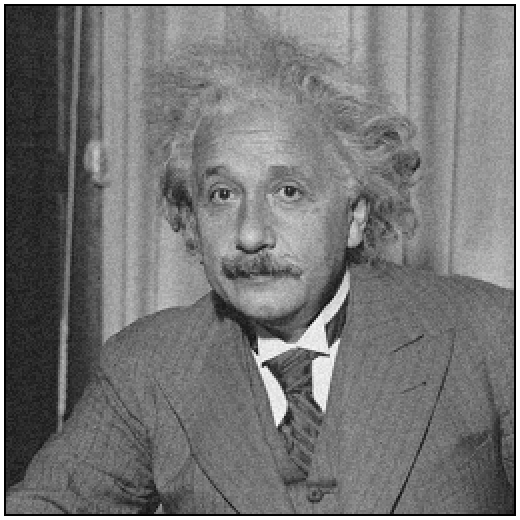
H.W. Shen

Overview

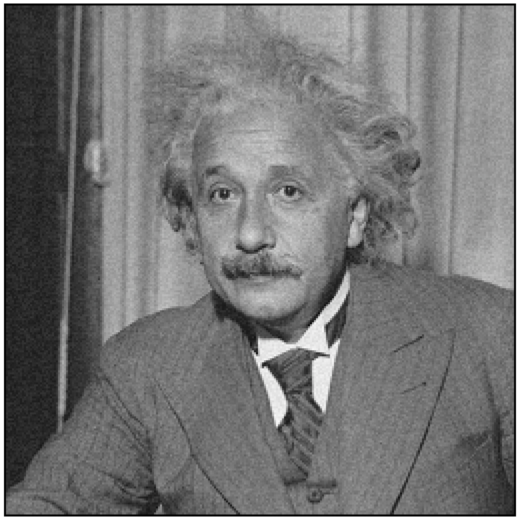
- What is SciVis?
- Data & Applications
- Iso-surfaces
- Direct Volume Rendering
- Vector Visualization
- **Challenges**

Challenges - Accuracy

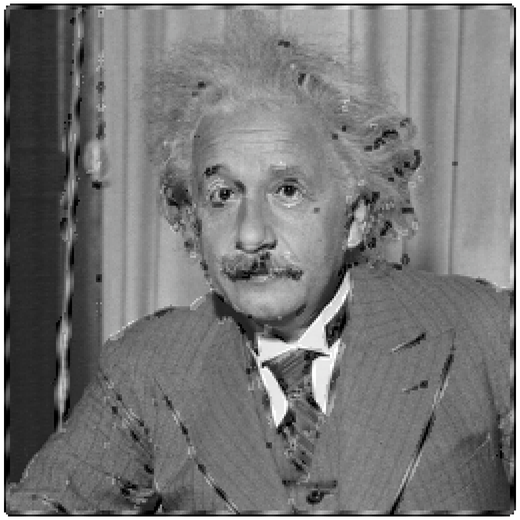
- Need metrics -> perceptual metric



(a) Original



(b) Bias-Added



(c) Edge-Distorted

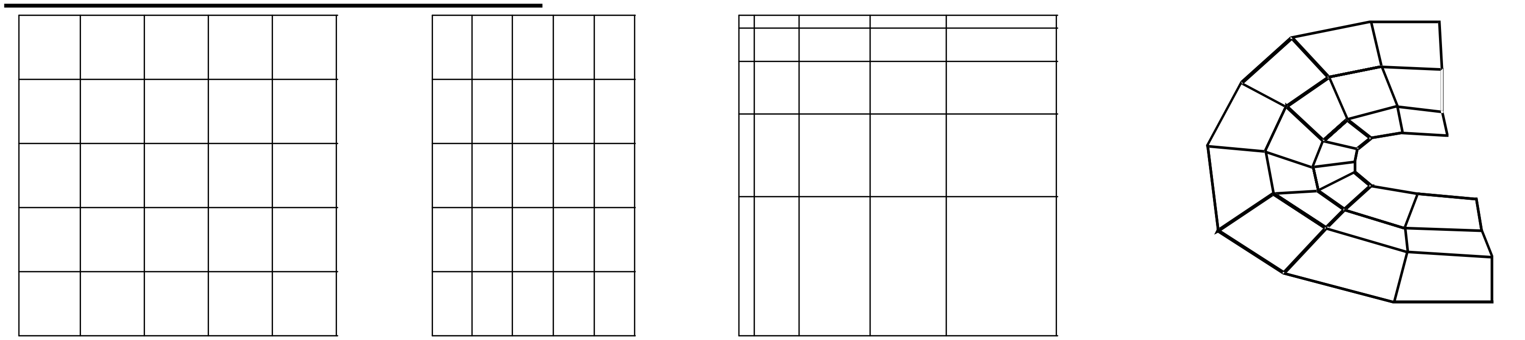
Challenges - Accuracy

- Deal with unreliable data (noise, ultrasound)

Challenges - Accuracy

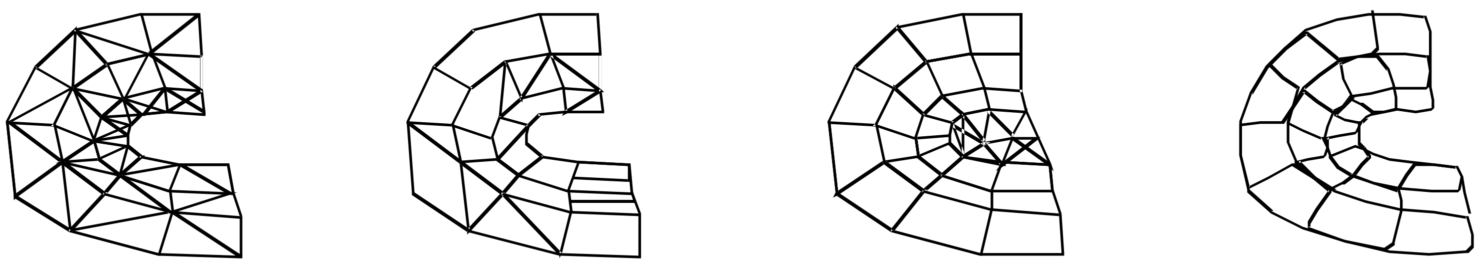
- Irregular data sets

Structured Grids:



regular uniform rectilinear curvilinear

Unstructured Grids:



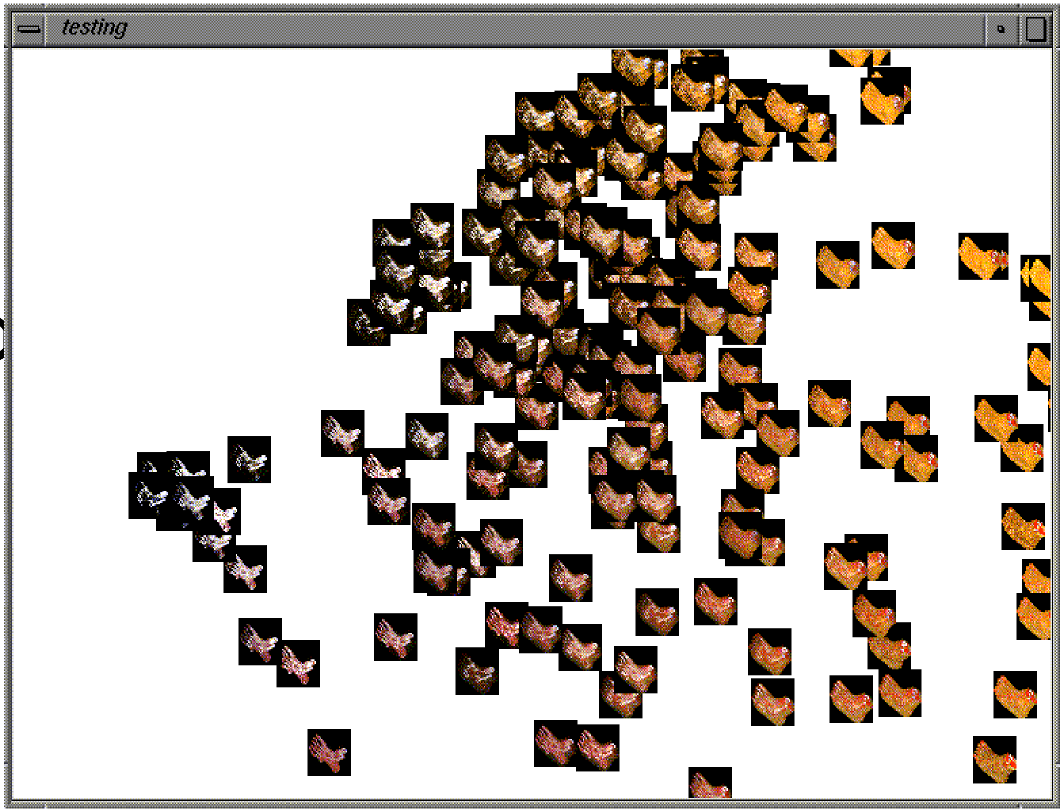
regular irregular hybrid curved

Challenges - Speed/Size

- Efficient algorithms
- Hardware developments (VolumePro)
- Utilize current hardware (nVidia, ATI)
- Compression schemes
- Terabyte data sets

Challenges - HCI

- Need better interfaces
- Which method is best?



Challenges - HCI

- "Augmented" reality
- Explore novel I/O devices