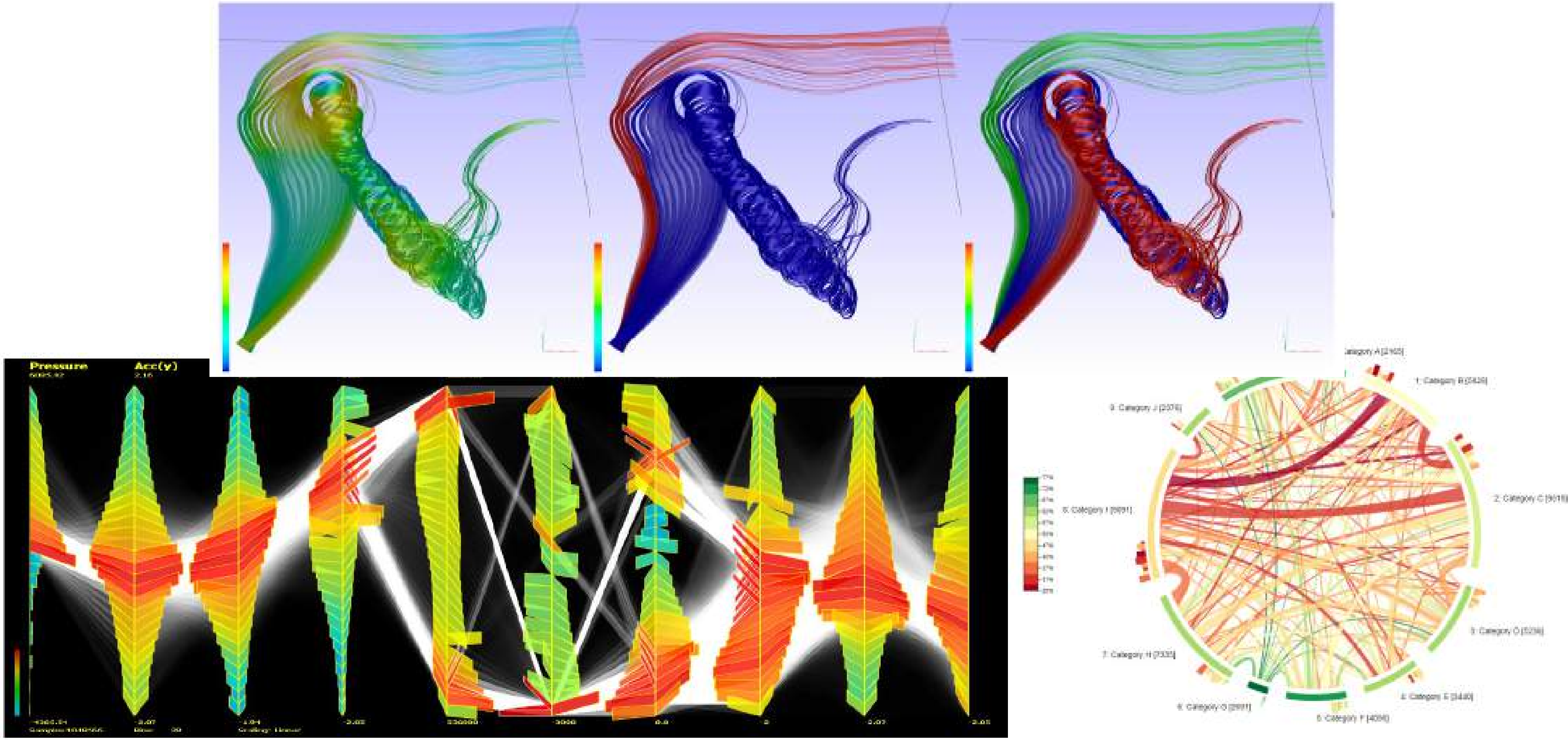
Data Visualization: An Introduction



Robert S Laramee

The Big Data Story

Sally discovers that she can collect data cheaply

Sally collects a LOT of data, likely years worth

Sally eventually wants to derive knowledge and extract insight from the data

She discovers what a challenge this is due to the size and complexity of the data

Sally passed a magic threshold and a special gap appears

The Universal Big Data Story

Computational Fluid Dynamics (CFD) Engineers: simulation data

Physicists and Astronomers: experiments & images

Marine Biologists and Bio-Chemists: accelerometers, simulations

Psychologists, Sociologists, Criminologists: EEG, Surveys

Sports Scientists: video & events

Journalists, Humanities: writing & text data

Governments and Councils: public sector data

Banks etc: customer and finance data

Call Centers: customer and employee data

Retail: customers, sales, financial, stock data

Web sites: Web traffic data, internet data

Transportation: air traffic, trains, automobile, cycling, walking data

You: photos, videos, and files

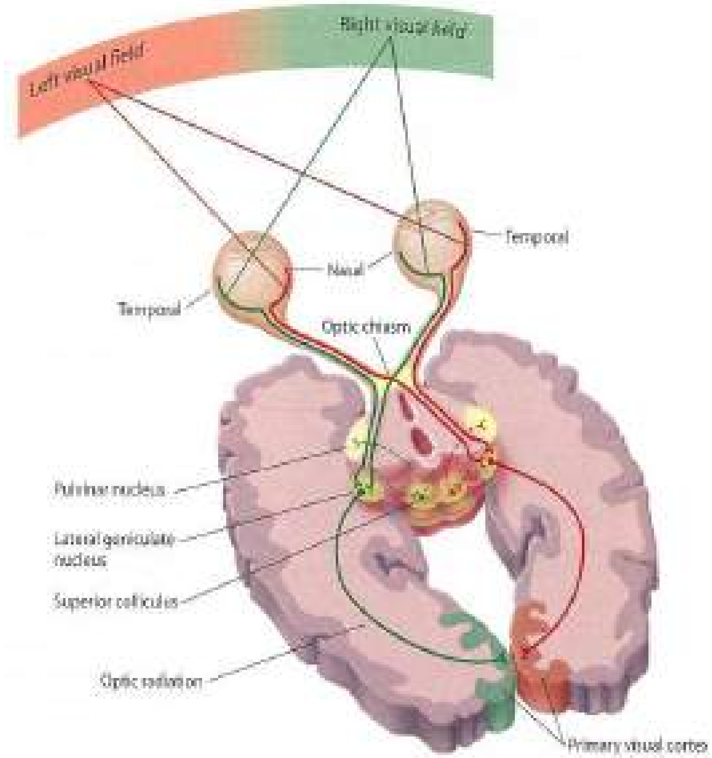
Everyone and Everywhere

People like to collect

But there's a catch...

The Visual Cortex

- Visualization exploits our powerful visual system
- 2 million nerve fibers coming from optic nerves
- Several billion neurons devoted to analyzing visual information (30% cortex)
- 8% for touch, 3% for hearing (Discover, 1993, Ware, 2013)
- Enables massively parallel processing of the visual field, i.e., incoming color, motion, texture, shapes etc.

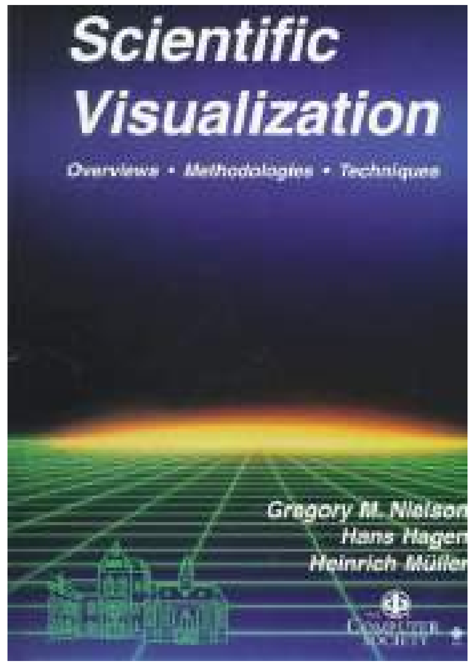


Visualization Background

Leonardo da Vinci (1452-1519)

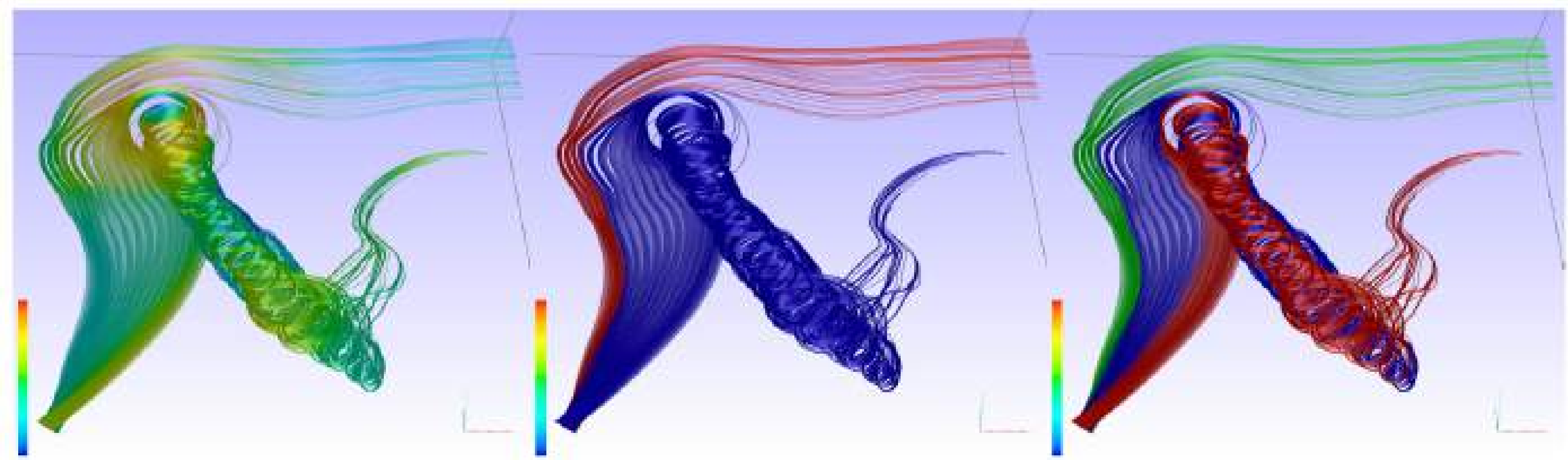
- Manual Visualization is very old
- Often an intuitive step to make something clearer, e.g., a graph
- Classical (easy) approaches known from business graphics (Spreadsheet software...)
- Classical approaches are very limited
- Computer-Aided Visualization is its own scientific discipline since 1987
- First visualization conference in 1990

1997



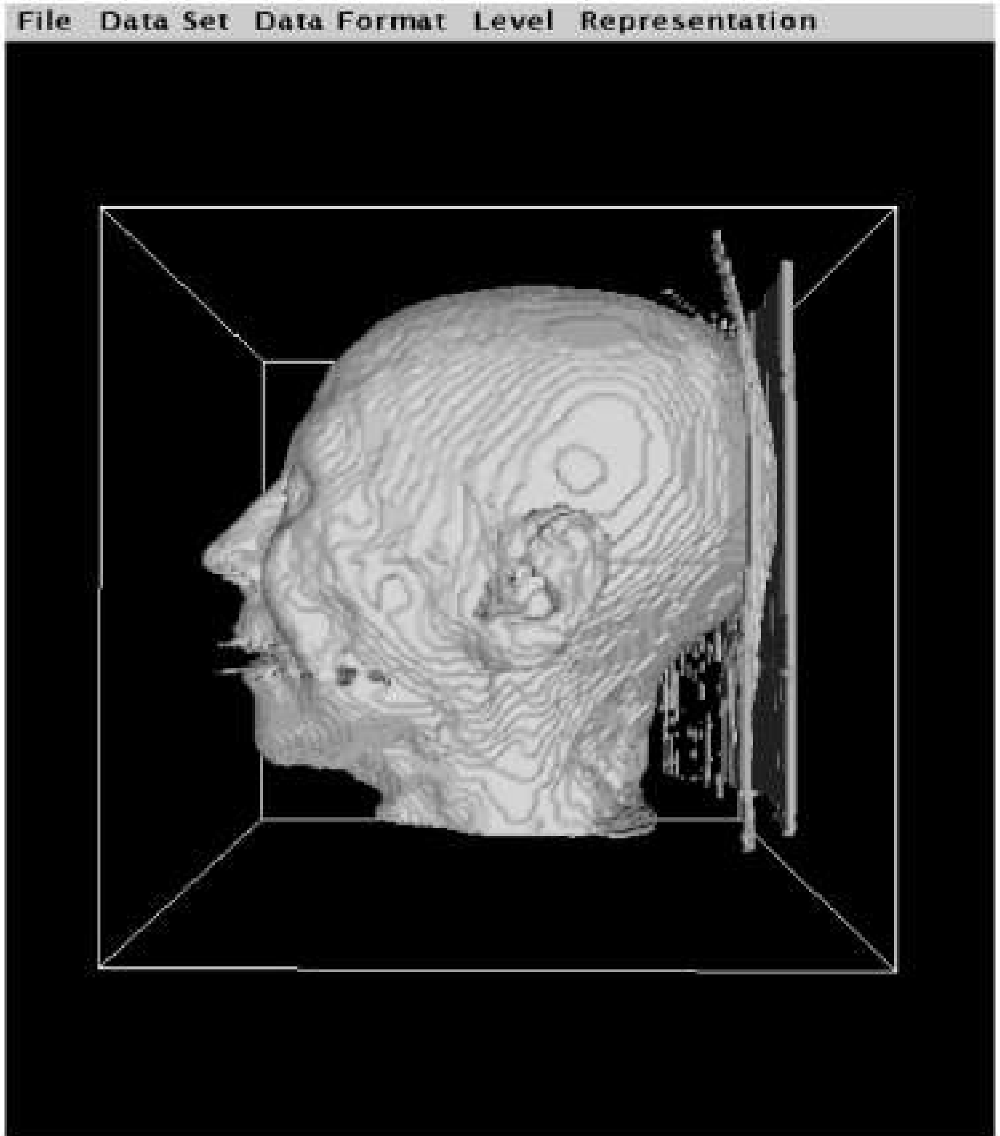
Major Visualization Subtopics

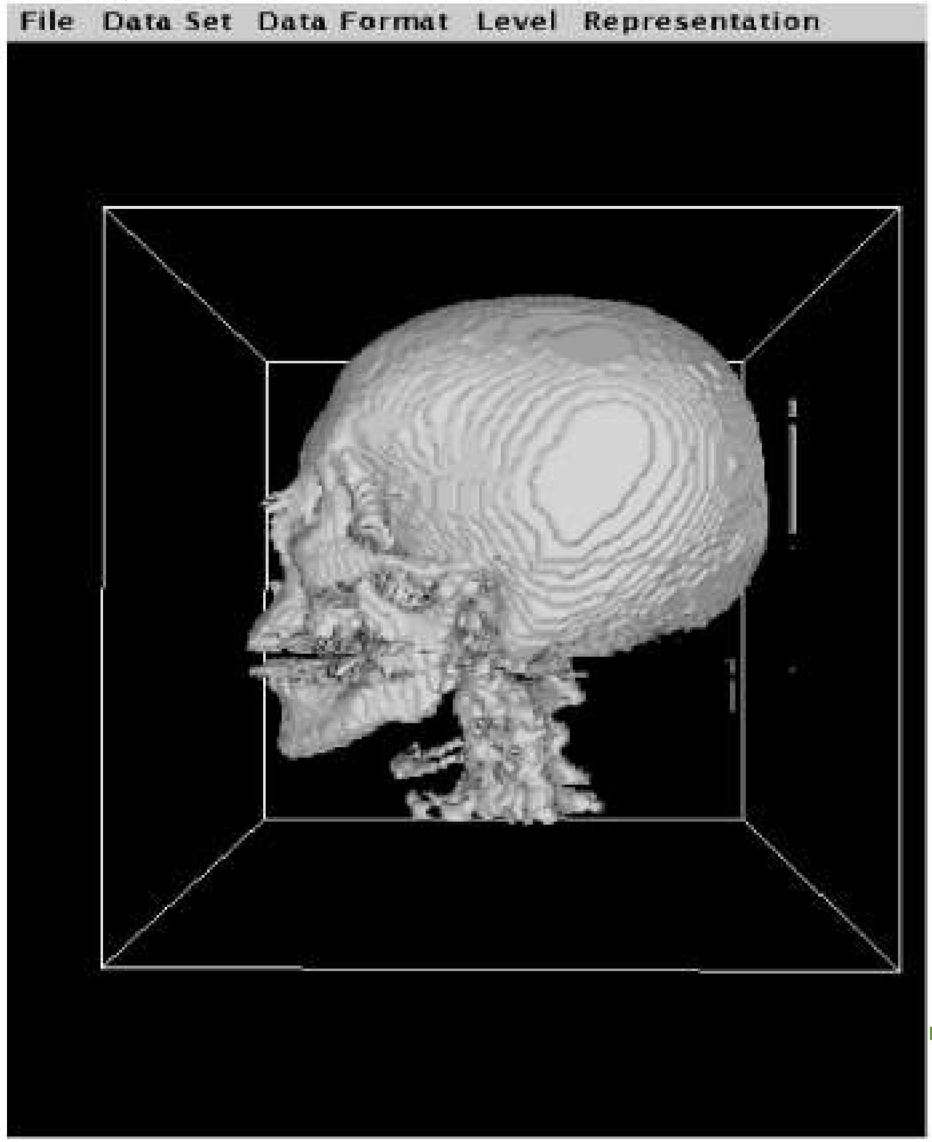
Medical Data uses Volume Visualization (VolViz)
Vector Data uses Flow Visualization (FlowViz)
Abstract Data uses Information Visualization (InfoViz)
Geographical Information Systems (GIS) Data
Historical data (Archeology, Cultural Heritage)
Microscopic data (Molecular Physics), Macroscopic Data (Astronomy)
Visual Analytics



Volume Visualization Examples

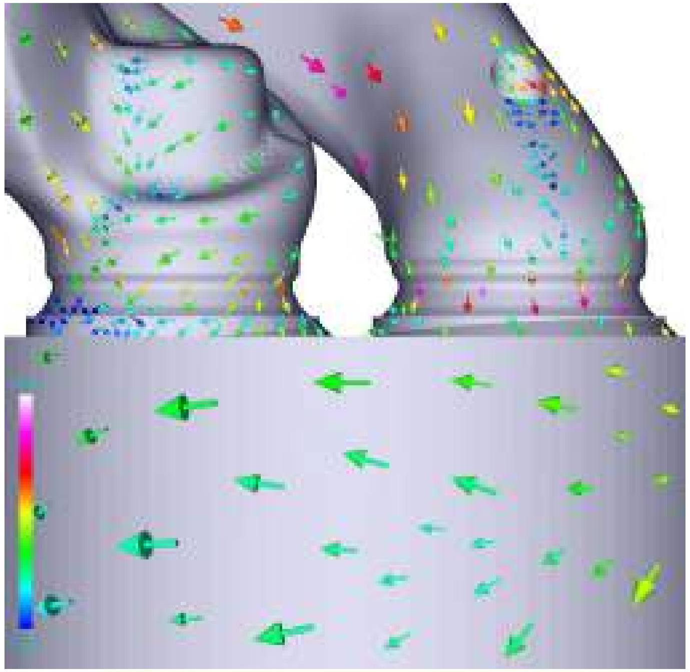
Medical Data

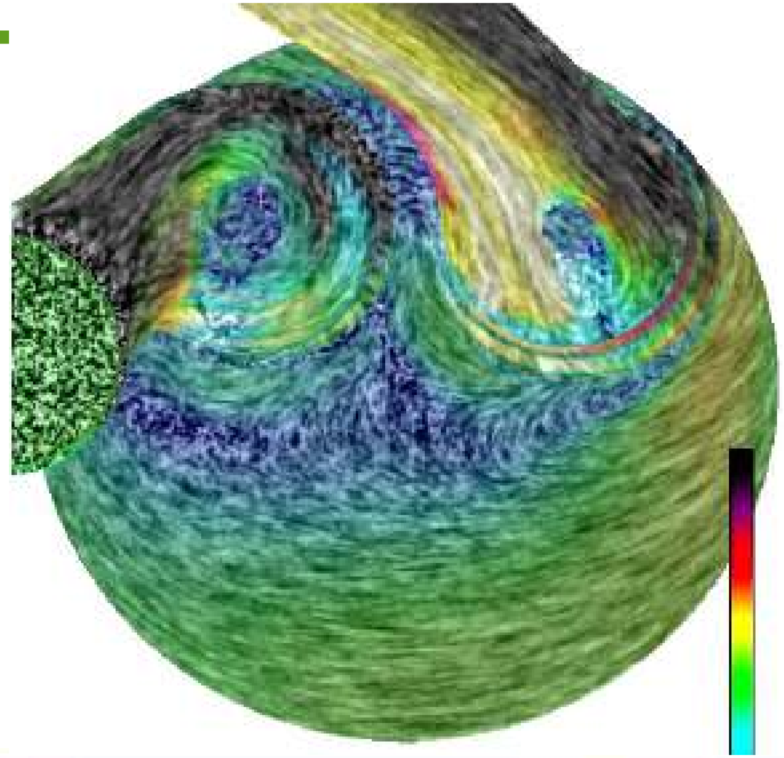


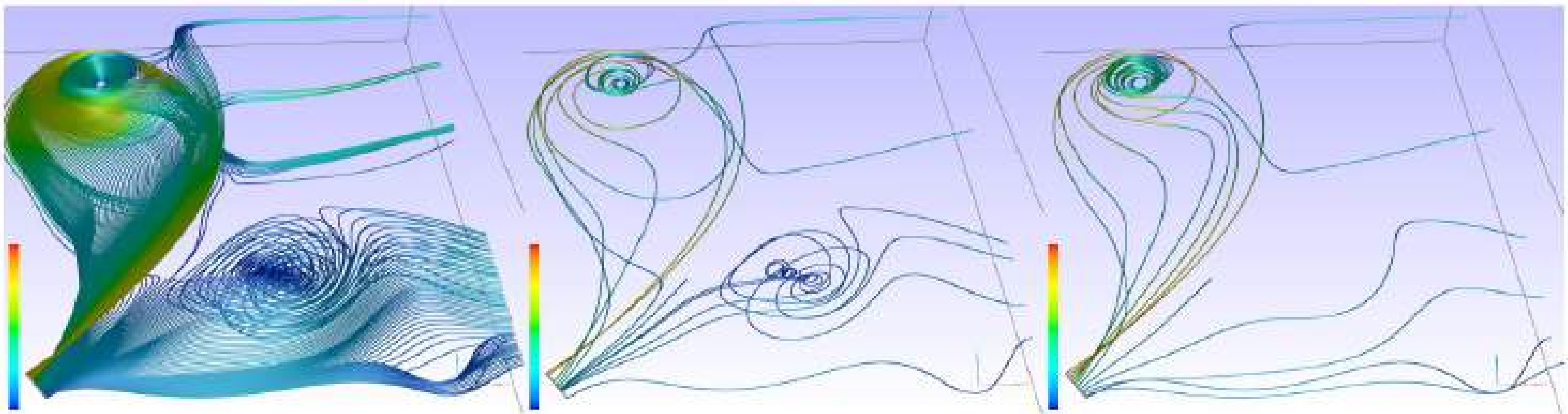


Flow Visualization Examples

Vector Data



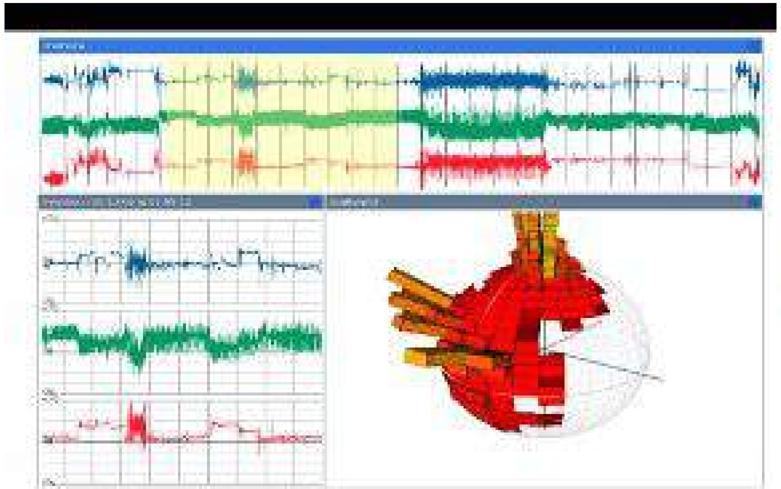


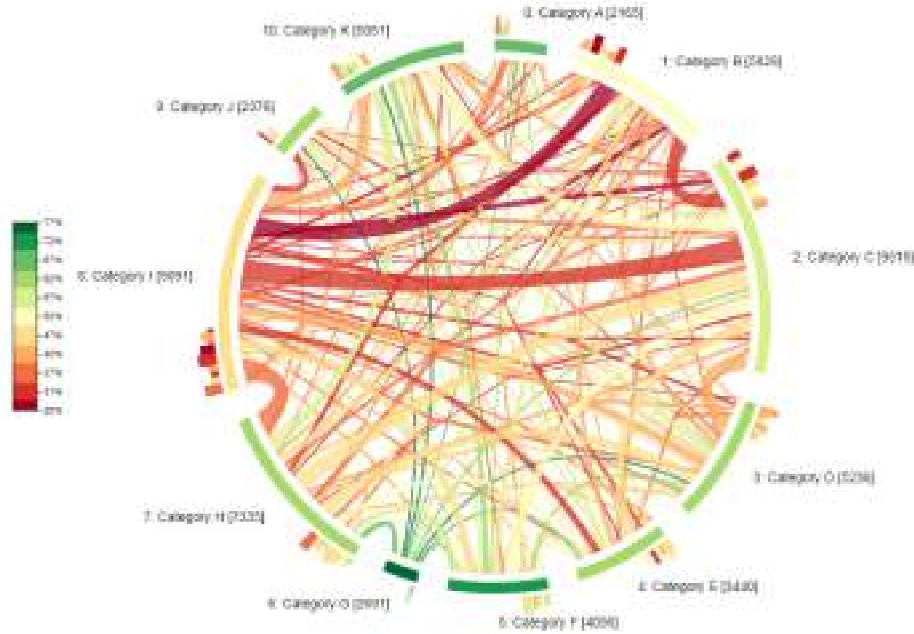


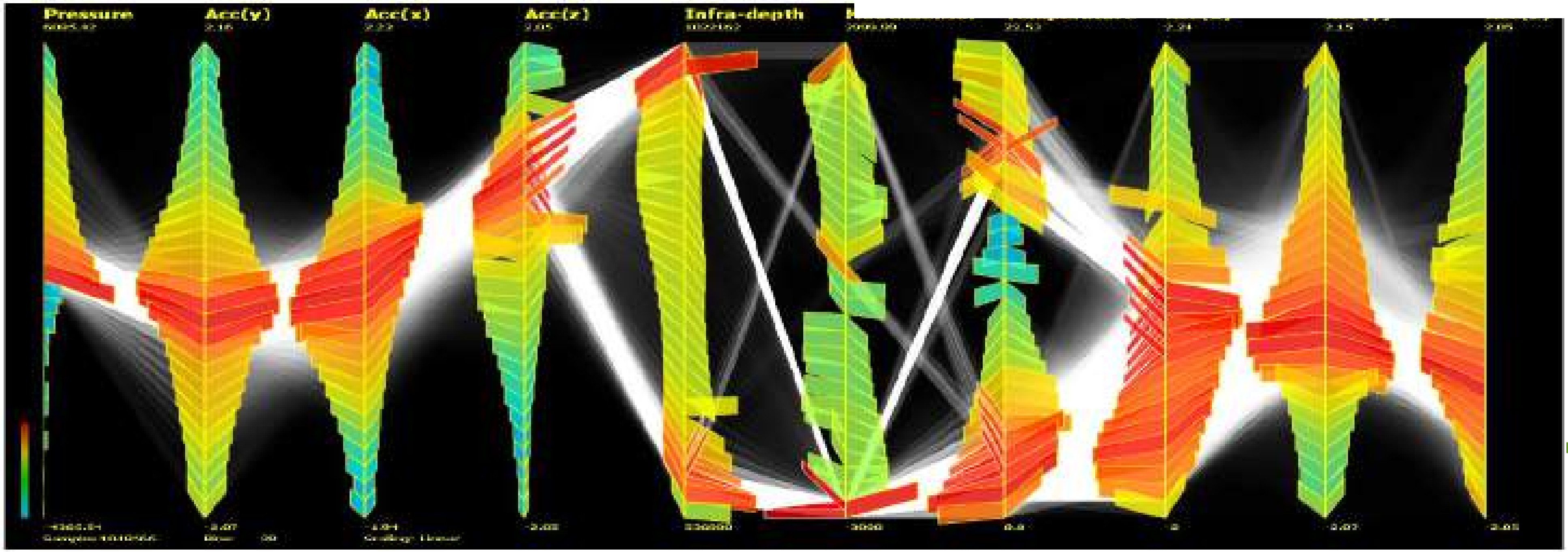
Information Visualization Examples

Abstract

Data







Visualization Goals

Visualization, in order to:

- **Explore**

no a-priori knowledge about the data,
Visualization serves to **explore** the data

- **Analyze**

There is a hypothesis,
Visualization Serves to **Confirm/Refute (Disprove)**

- **Present**

The data characteristics are well known, Visualization.
Serves to **communicate the results**

Visualization Foci

Three main sub-fields:

Inherent geometry

- Volume Visualization
- Flow Visualization

Scientific Visualization

3D

nD

- Information Visualization

Usually no inherent geometry

The Visualization Pipeline

Standard processing stages in Visualization: A data centric view

Visualization Pipeline Overview

Data acquisition → Data is produced or acquired

Data enhancement → Data is prepared (or preprocessed)

Visualization mapping → Data is mapped to geometric primitives

Rendering (3D or 2D) → Data is transformed into images

Visualization Pipeline Phase 1

Data acquisition

Data is generated (or acquired)

Data is produced and archived. This is what everyone is good at.

- Measurement, e.g., CT, MRI
- Written down, scanned in as text
- User Input, into databases or spreadsheets
- Simulation, e.g., Computational Fluid Dynamics Simulation (CFD).
- Modeling, e.g., Computer Aided Design (CAD), dynamical systems
- Videos or images are recorded

Visualization Pipeline Phase 2

Data enhancement

1. Data is acquired (or produced)
2. Data is prepared (or "preprocessed")

Data enhancement = Data preparation/preprocessing

- Filtering, e.g. smoothing (noise filtering)
- Errors are discovered and corrected
- Missing values may be handled
- Resampling or modify grid representation
- Derive new data, e.g., gradients
- Data interpolation

Visualization Pipeline Phase 3

2. Preprocessed data

Visualization mapping

3. Data is mapped to geometry

Visualization mapping results in Data being visible

Data is represented by geometric primitives: points, lines, triangles, polygons, cubes, tetrahedra, of varying size, shape, color, transparency

- Compute isosurface
- Compute glyphs or icons
- Compute graph layout
- Compute voxel attributes: color, transparency, …

Visualization Pipeline Phase 4

3. Data is mapped to geometry

Rendering (3D or 2D)

4. Data becomes an image(s)

Rendering involves representation with Computer Graphics (CG)

- Projection (3D → 2D)
- Visibility calculation
- Shading
- Compositing (accumulate transparency and color values)
- Animation

References and Further Reading

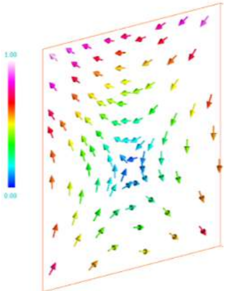
- **Interactive Data Visualization: Foundations, Techniques and Applications** by Matthew O. Ward, Georges Grinstein, and Daniel Keim, AK Peters/CRC Press, 2010
- **Data Visualization Principles and Practice**, **Second Edition** by Alexandru Telea, AK Peters/CRC Press, 2015
- **Information Visualization: Perception for Design, Third Edition** by Colin Ware, Morgan Kaufman Publishers, 2013

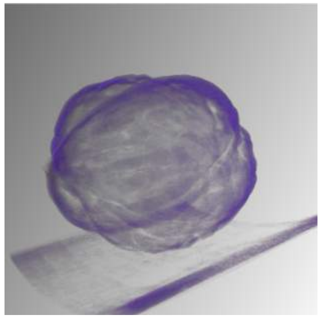
Acknowledgements

We thank the following people:

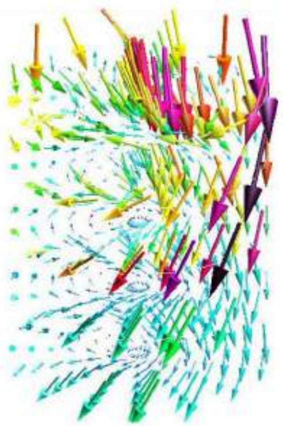
Daniel Archambault, R Daniel Bergeron, Guoning Chen, Nick Croft, Zhao Geng, M. Eduard Gröller, Edward Grundy, Charles D Hansen, Helwig Hauser, Bruno Jobard, Mark W Jones, Rami Malki, Ian Masters, Tony McLoughlin, Michael Nicholas, Zhenmin Peng, Jonathan C Roberts, Rick Walker, Rory Wilson

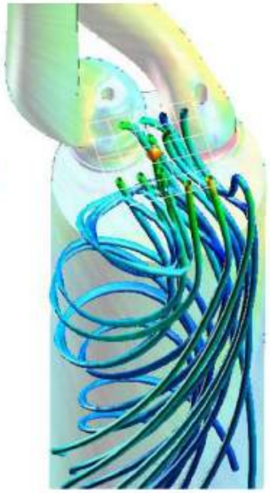
Visualization, Introduction Lecture Part II

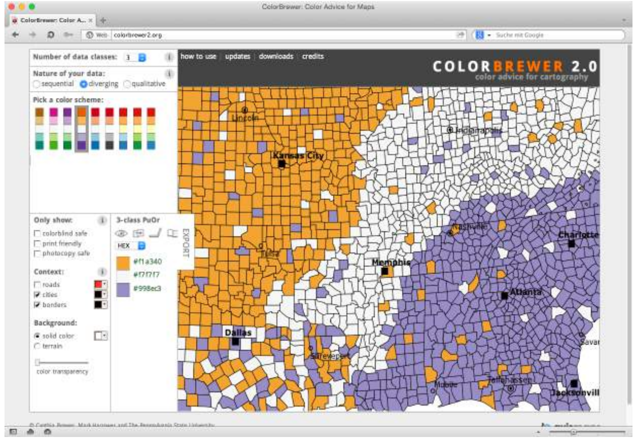


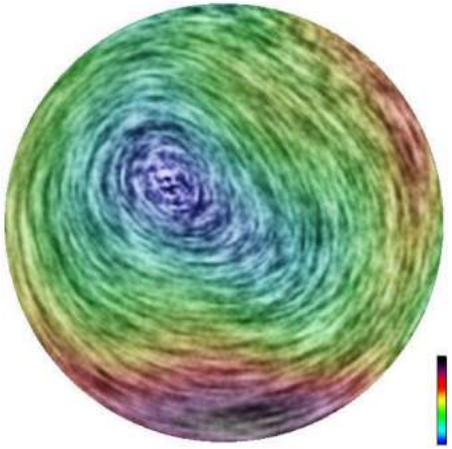


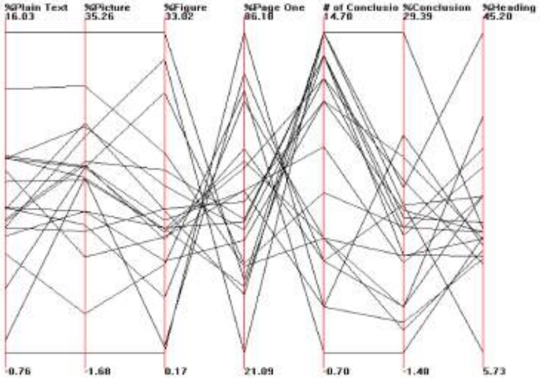
About data

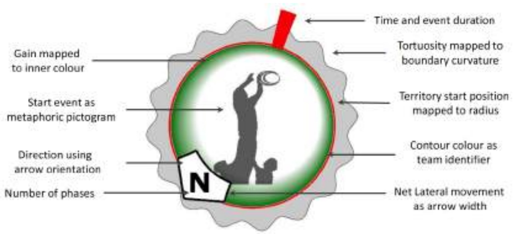


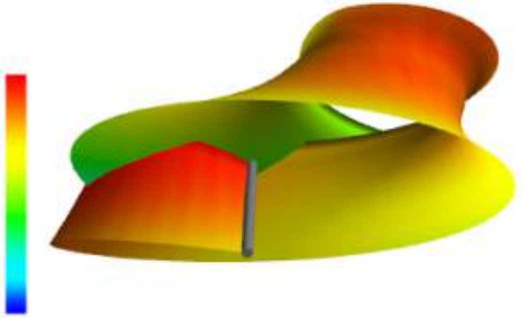












Previous Lecture

Summary of Previous Lecture

- The Data Story
- Visualization – Definition
- Application Examples
- Visualization Goals: Presentation, Analysis, Exploration
- Scientific Visualization versus Information Visualization
- The Visualization pipeline

Overview of This Lecture

Preview of this Lecture:

- About Data
- Data Characteristics and types
- Data Dimensionality
- Visualization Examples
- Some tips on color

About Data

Data characteristics,
Data attributes,
Data dimensionality

A Data Centric Point of View

Data is:

- The focal point of visualization: visualization starts with data
- Data: a “driving factor” (along with the user) with respect to the choice and attributes of the visualization method

Important questions:

- What is the dimensionality of the data?
- Where does the data come from?
- Which visualization methods make sense?
- Why did the user collect the data?
- What would the user of the data like to learn?

Data Dimensionality

What dimensionality is the data?

- Scientific Visualization: inherent spatial domain (**SciVis**):
  - 1D, 2D, 2.5D or 3D dimensionality given
  - instantaneous or time-dependent (+1D)
  - examples: medical data, data from flow simulation, GIS-data...
  - Represents a physical phenomenon
- Information Visualization: no inherent spatial reference (**InfoVis**):
  - abstract data,
    spatial arrangement is designed or derived for visualization
  - examples: databases, text, spreadsheets, surveys, monetary, call center, images, sound

Data Characteristics

What characteristics does the data have?

Data types:

- Scalar: numeric e.g. natural, integer, rational, real numbers, complex numbers
- Vectors and tensors
- non-numeric: nominal, ordinal values, text
- multi-dimensional or multi-attribute values: n-dimensional, columns in a spreadsheet, e.g. car type, manufacturer, year of manufacture, country of origin, mpg, horsepower, number of cylinders, retail price…

Characteristics: dimensionality, domain (upper and lower bounds), distribution, scale, size, resolution

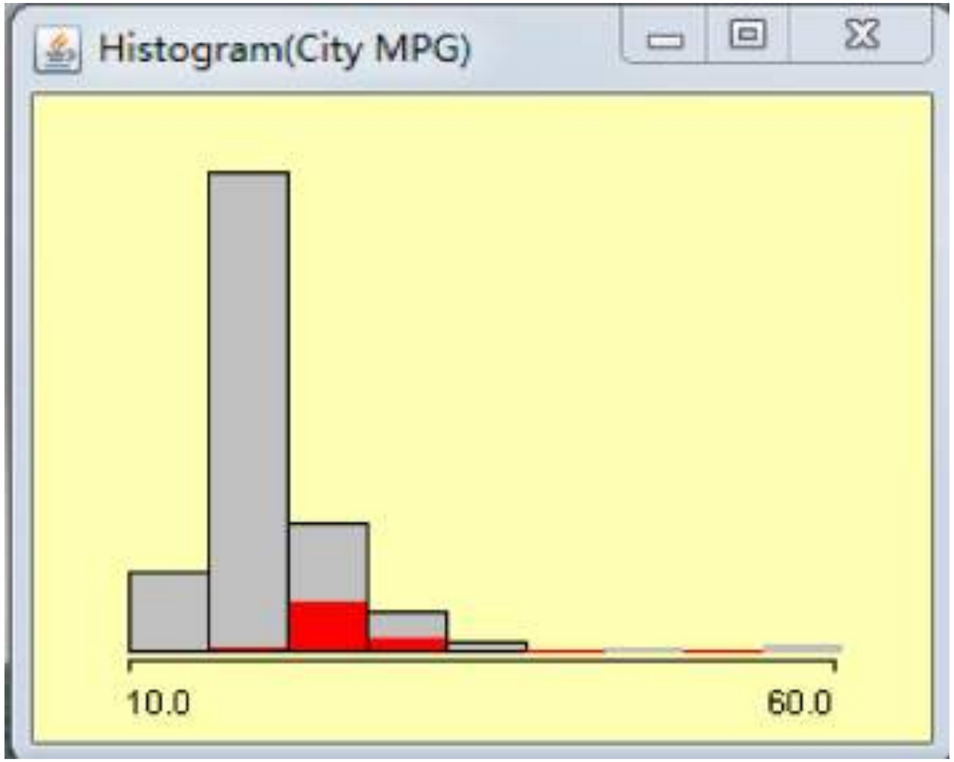
Data Presentation

How can data be presented?

- Is there an inherent spatial domain?
  - If yes then use inherent domain?
  - If no then which spatial domain?
- Which dimensions are used for what purpose?
  - What is relationship between dimensionality and data characteristics
  - Available Presentation Space (2D/3D/4D)
  - Where is the Focus?
  - Is some data more important than other?
  - What can be left out?

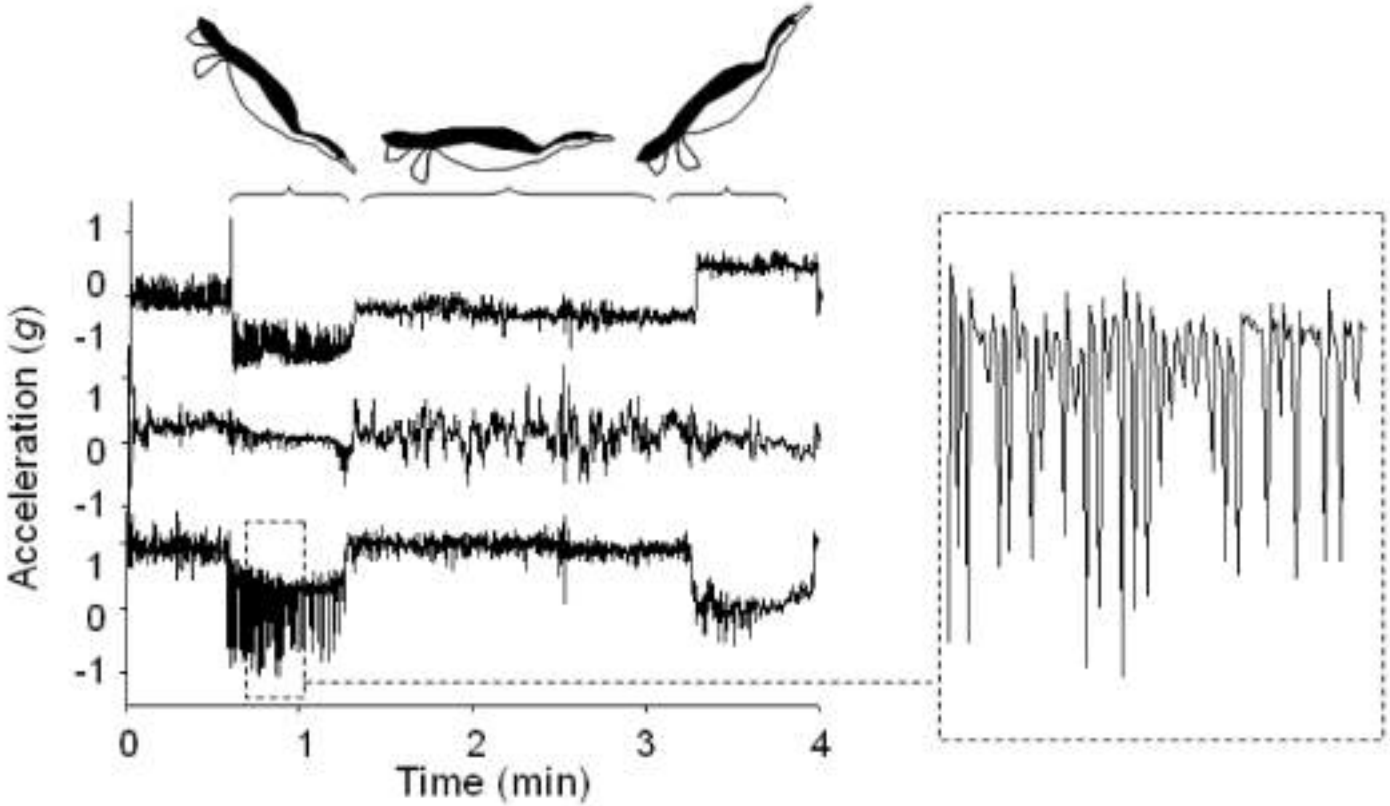
Data Dimensionality versus Visualization Dimensionality Examples

Data	Description	Visualization Example
$N^1 \rightarrow R^1$	Series of Values	bar chart, pie chart, histogram, bubble chart



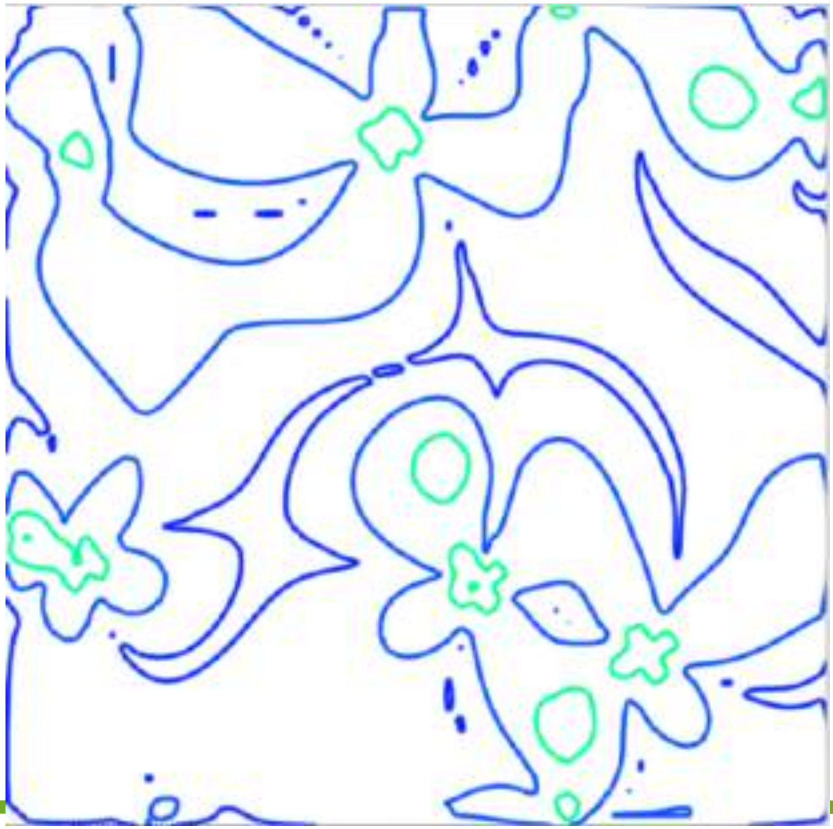
Data Dimensionality versus Visualization Dimensionality Examples

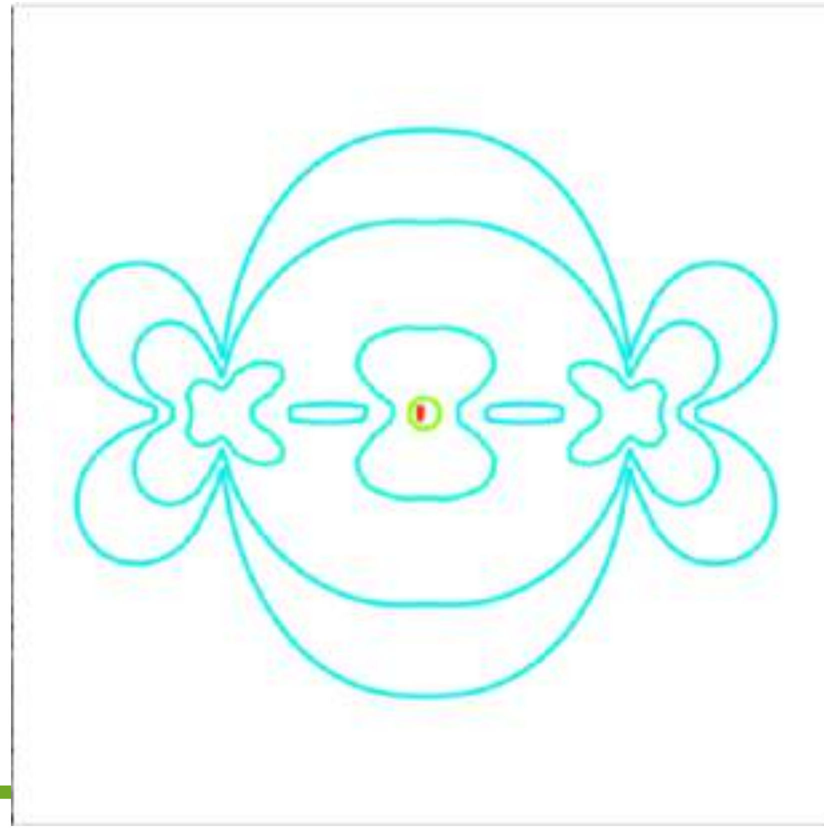
Data	Description	Visualization Examples
$R^1 \rightarrow R^1$	Function	Graph



Data Dimensionality versus Visualization Dimensionality Examples

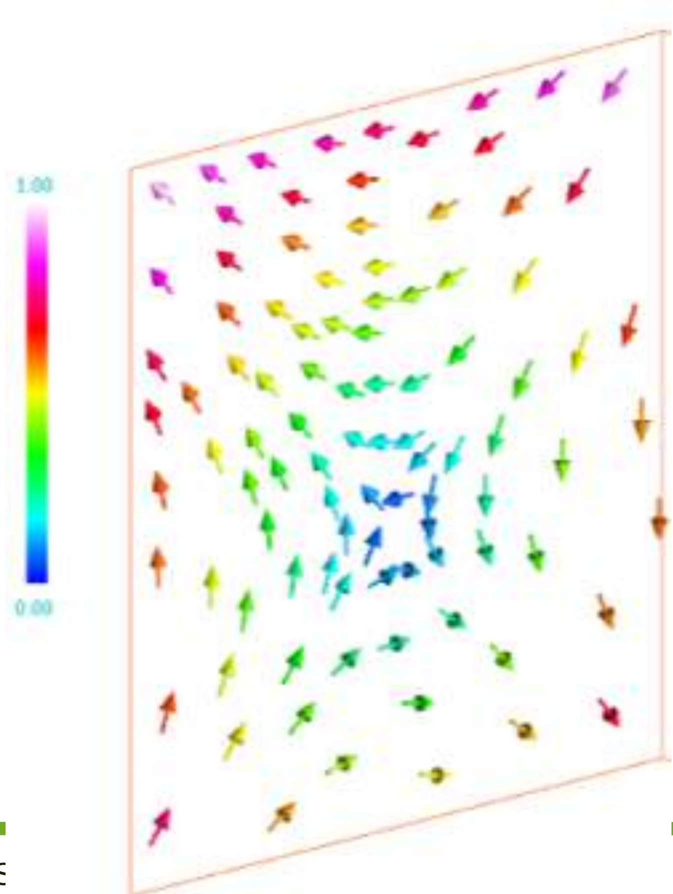
Data	Description	Visualization Examples
$R^2 \rightarrow R^1$	Function over R^2	2D Height fields in 3D, Contour lines in 2D

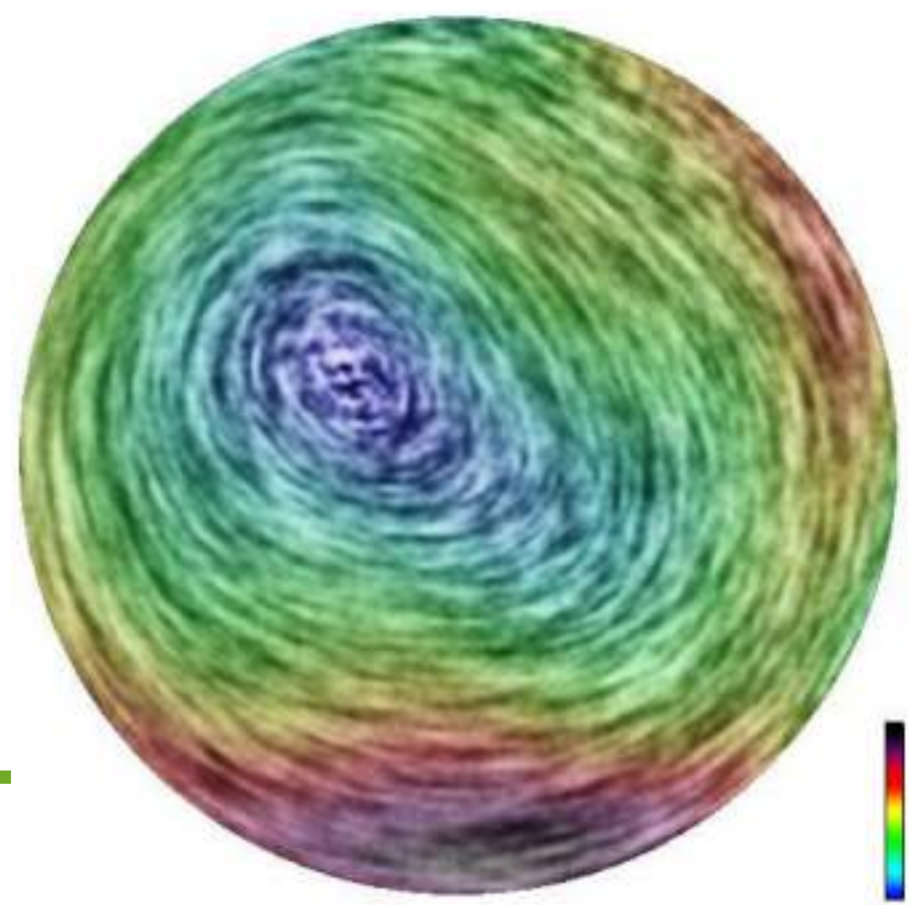




Data Dimensionality versus Visualization Dimensionality Examples

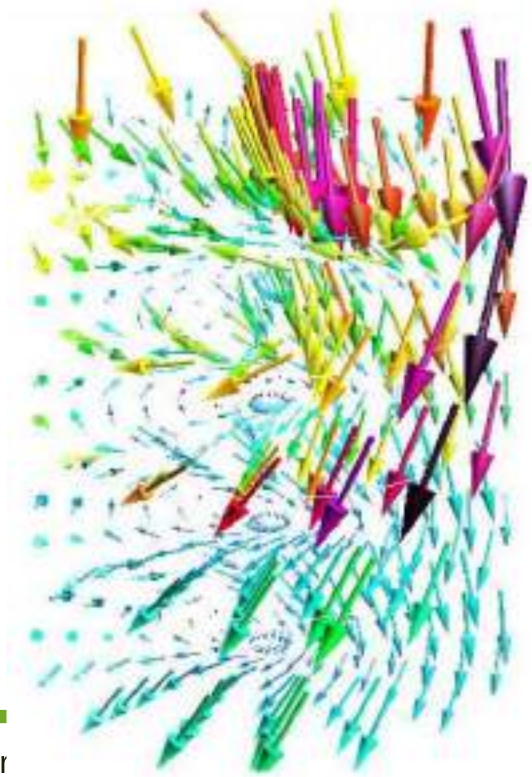
Data	Description	Visualization Examples
$R^2 \rightarrow R^2$	2D vector field	vector glyphs, texture-based flow visualization, streamlines

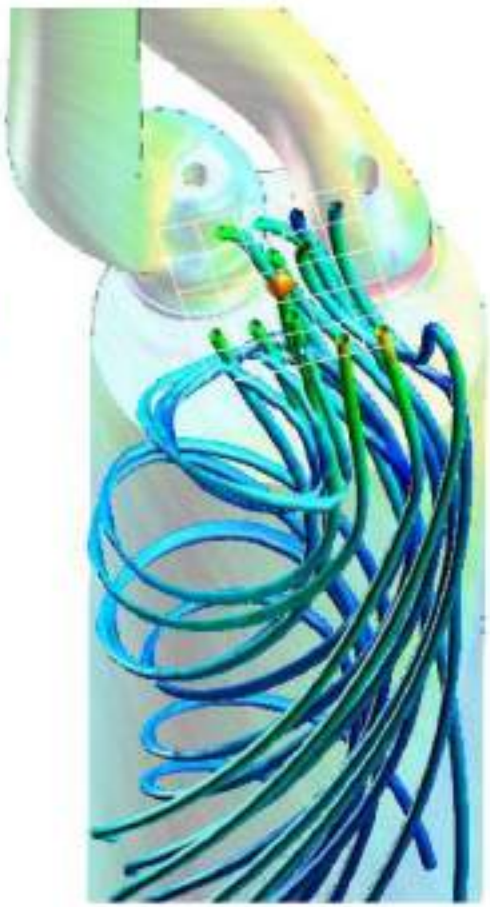


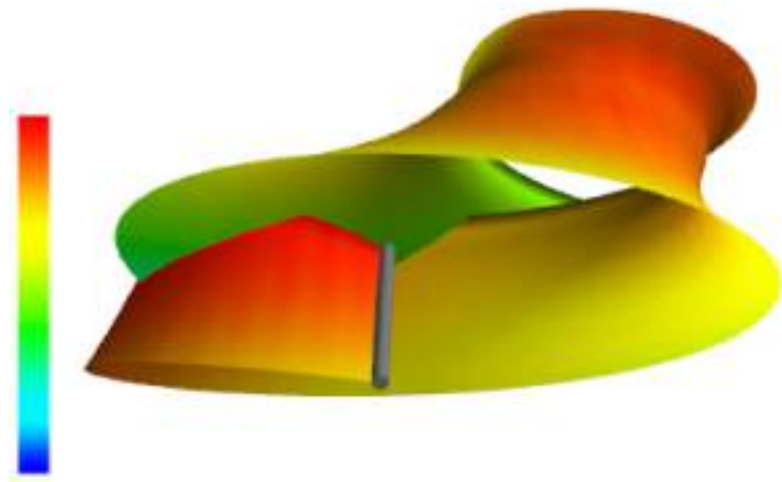


Data Dimensionality versus Visualization Dimensionality Examples

Data	Description	Visualization Examples
$R^3 \rightarrow R^3$	3D Flow	Streamlines, Stream surfaces

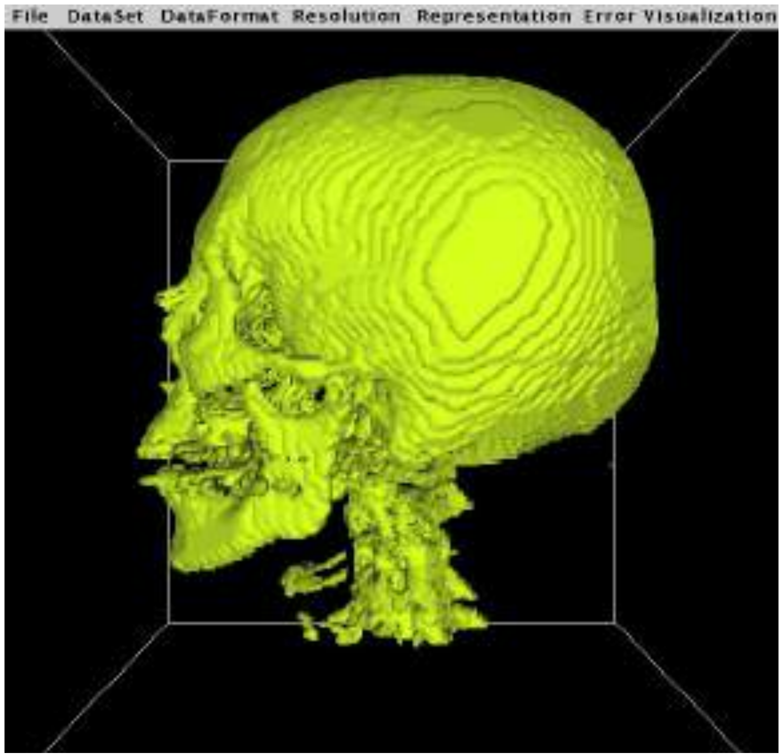


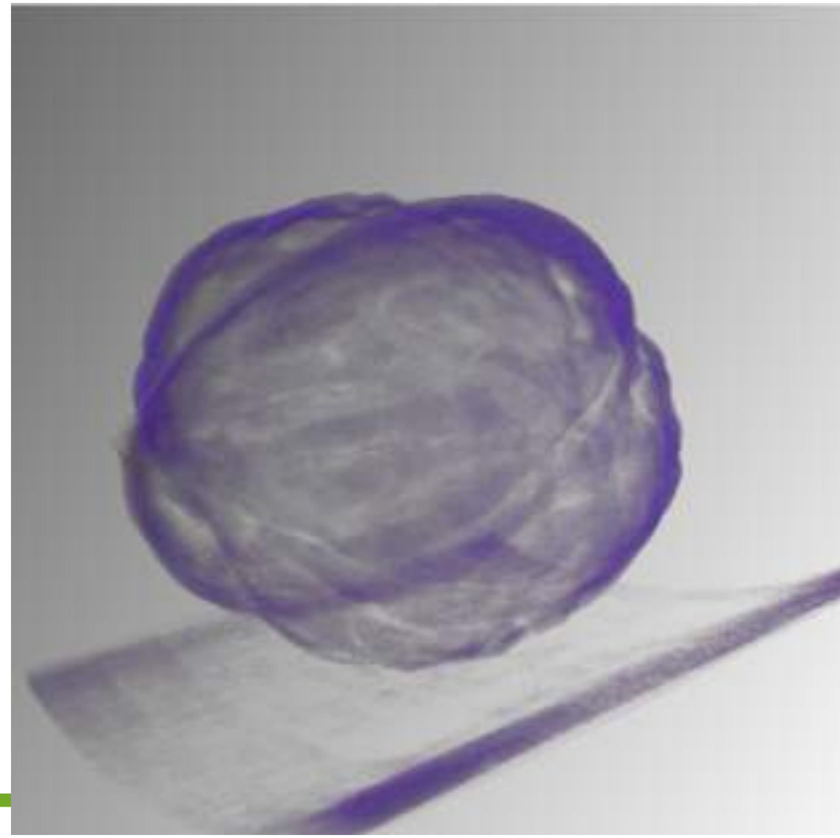




Data Dimensionality versus Visualization Dimensionality Examples

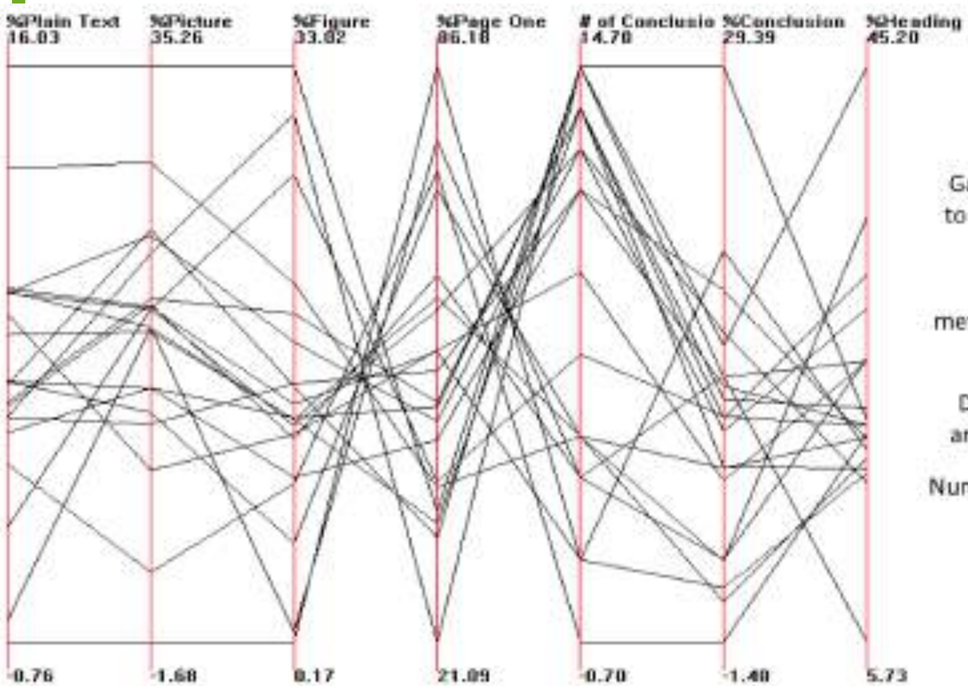
Data	Description	Visualization Examples
$R^3 \rightarrow R^1$	3D-Density Values	Isosurfaces in 3D, volume rendering

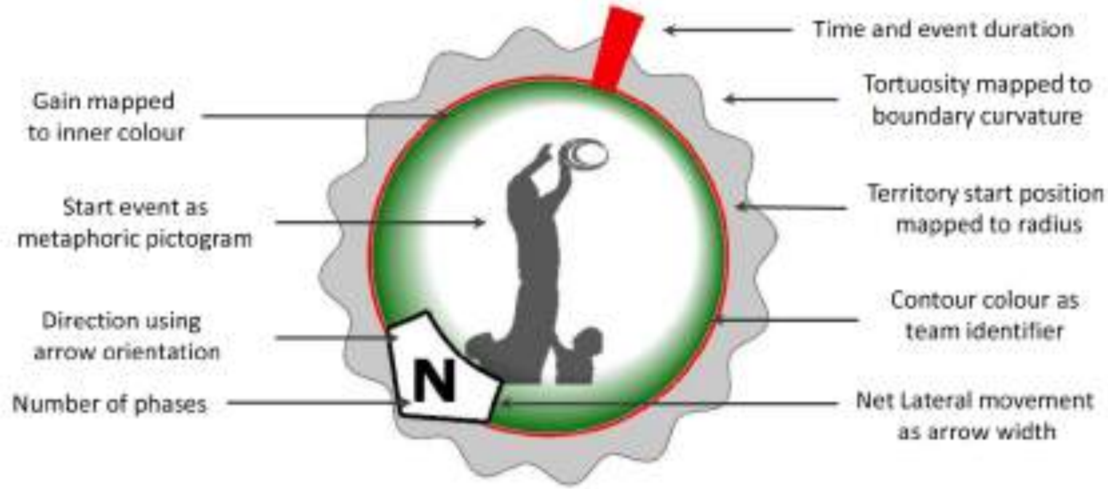




Visualization Examples

Data	Description	Visualization Example
$(N^1 \rightarrow) R^n$	Tuple Quantities (multi-attribute data)	parallel coordinates, glyphs, icons





Visualization and Color

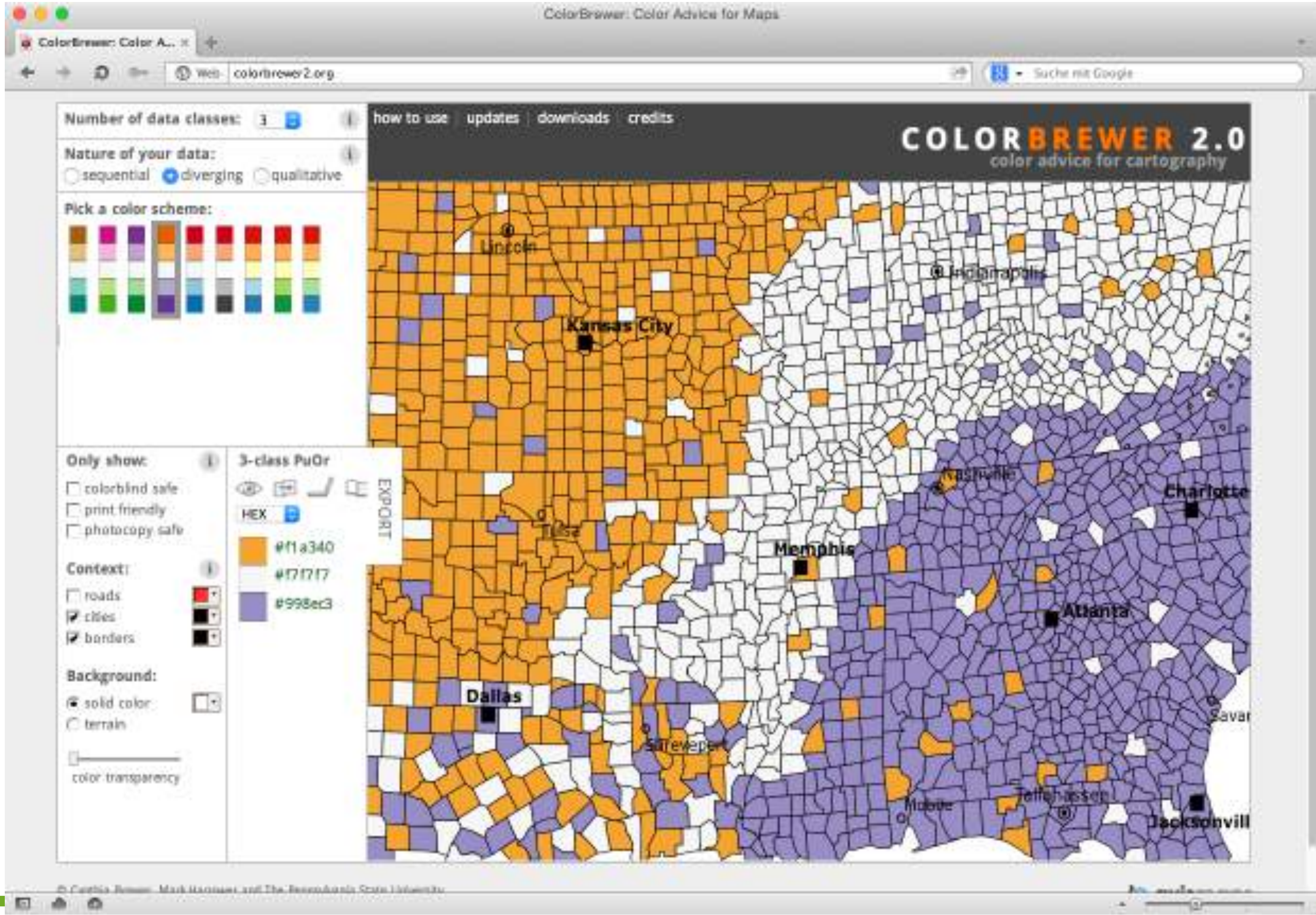
Tips for the Application of
Color in the Visualization

Application of Colors

Consider:

- color can strengthen information
- the maximum number of colors to use for categorical data 7±2
- approximately 50–300 distinguishable color shades (different depending on color)
- rainbow of colors ≠ linear
- color perception is depends strongly on context
- some users are color blind
- colors have associations

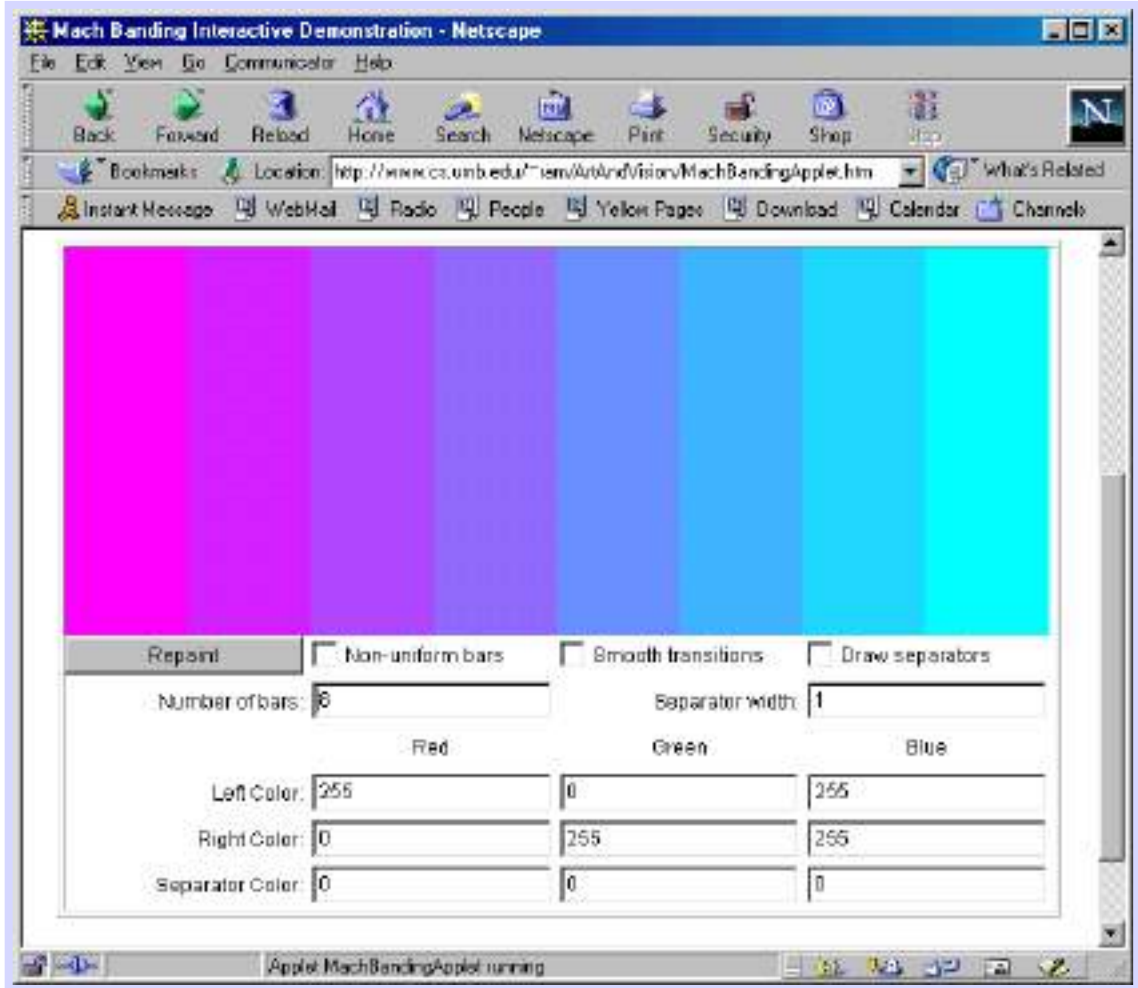
Choosing Colors with Color Brewer



Mach-Band Effect

Characteristics:

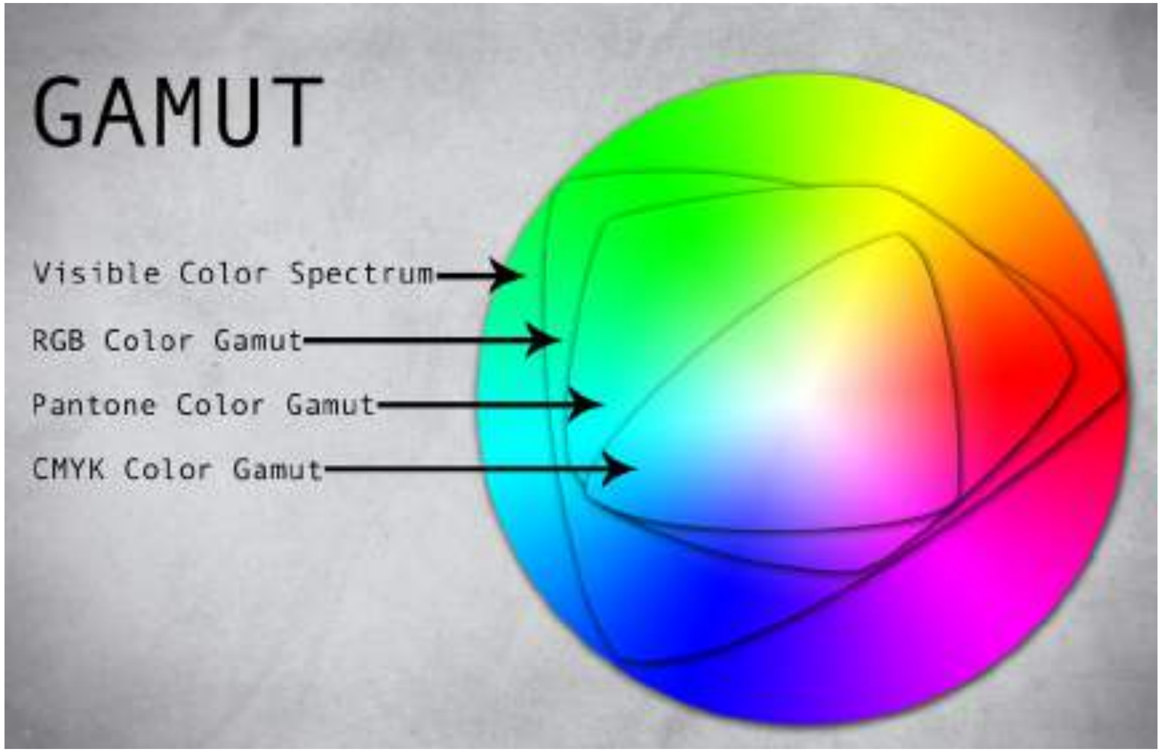
- Eyes strengthen boundaries
- This can introduce mistakes in discretion
- Attention should be paid to color intensity



Color Gamuts

Display devices are non-uniform:

- Color space non-uniform
- Color correction



Tips for Application of Color

Useful:

- gray scale lines as color boundaries
- avoid dark blue for details, animations
- dark blue and red do not mix
  (Why? That's why.)
- avoid high color frequency
- context, association is important
- good: color for qualitative visualization (as opposed to quantitative
- color can be used to strengthen boundaries

Further Reading

Interactive Data Visualization: Foundations, Techniques and Applications by Matthew O. Ward, Georges Grinstein, and Daniel Keim, 2010, AK Peters/CRC Press

Data Visualization Principles and Practice, **Second Edition** by Alexandru Telea, AK Peters/CRC Press, 2015

Information Visualization: Perception for Design, Third Edition by Colin Ware, Morgan Kaufman Publishers, 2013

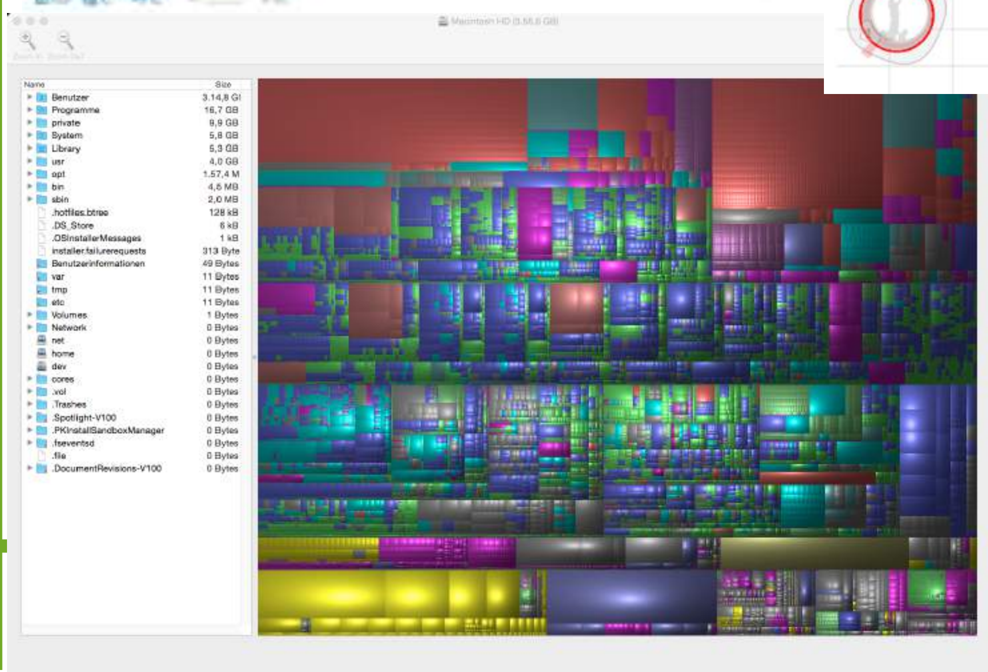
Acknowledgements

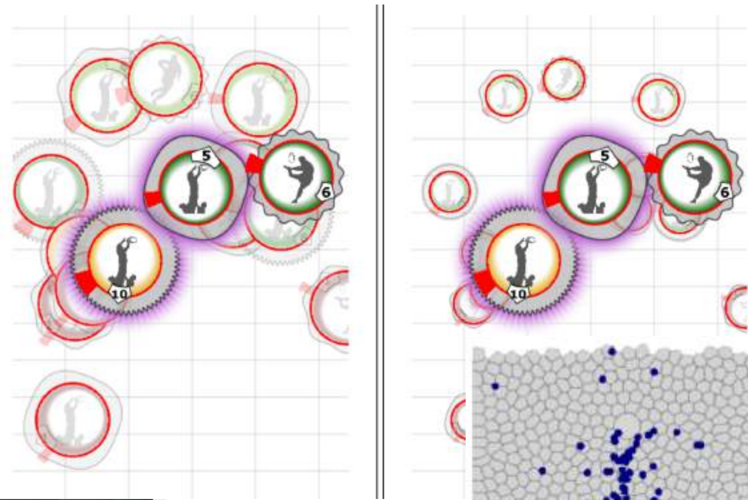
We thank the following:

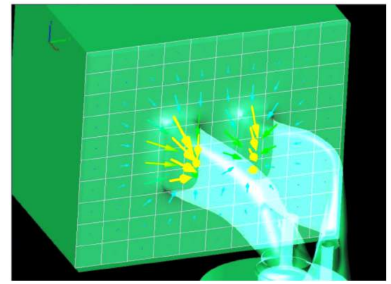
R Daniel Bergeron, Rhodri Bown, Guoning Chen, Min Chen, David Chung, Nick Croft, Iwan Griffiths, M. Eduard Gröller, Edward Grundy, Helwig Hauser, Mark Jones, Phillip A Legg, Matthew L Parry, Zhenmin Peng, Juergen Schneider, Inge Tastl, Daniel Weiskopf, Rory Wilson, Eugene Zhang

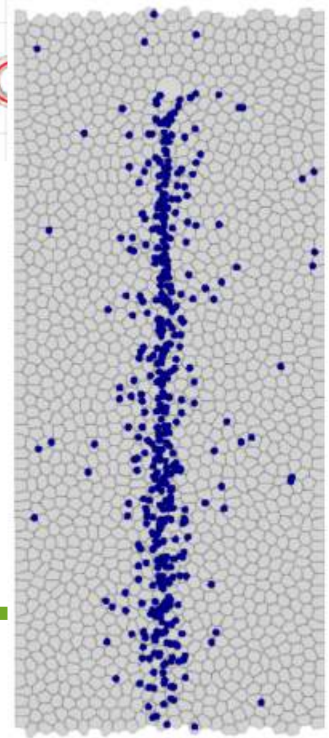
Data Visualization Lecture

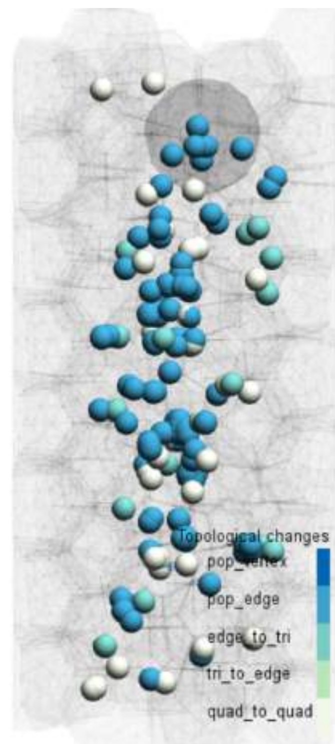
Focus on Information Visualization, Part 1











Review of Previous Lecture

Data Visualization Introduction:

- Data characteristics
- Data dimensionality
- Visualization dimensionality
- Examples
- Guidelines on Color

Overview of Current Lecture

Information visualization, part 1:

- introduction
- examples
- hierarchical data
  - trees, conventional and standard representations
  - tree maps, modern approaches, examples, linking with other views
- Focus and Context (F+C) visualization
  - conventional approaches
  - state of the art approaches
  - examples from various applications

Introduction to Information Visualization

Context, General Remarks

Recall: Visualization

Three main fields:

inherent spatial relation

- Volume Visualization
- Flow Visualization

$\Bigg\}$ Scientific Visualization

3D

---

*n*D

- Information Visualization

no inherent spatial relation

Information Visualization versus Scientific Visualization

- Abstract Data
- *n*-dimensional
- Very important:
  - – visual metaphor
  - – user interaction
  - – **exploration**, analysis (visual analytics), presentation

- Spatial data
- 1, 2- or 3-dimensional, time-dependent
- Very important:
  - – 3D rendering
  - – fast rendering
  - – **analysis**, exploration, presentation

Two Important Goals

An intuitive visual metaphor:

- How can abstract information be represented visually?
- How can high-dimensional data be depicted?
- 2D or 3D visualization?
- How can we position or place the data in space?
- Integration of Focus and Context?

A proper interaction scheme:

- Modification of visual parameters and metaphors
- Change of focus

Special Challenges

Often large amounts of data. How can we enable visualization and:

- selection of data subsets
- aggregation of data
- extraction of meta data

Extended data structures, complex data inter-dependencies

- deep hierarchies, large graphs, multi-modal data
- different kinds of inter-dependencies, i.e., relationships
- *pattern finding capabilities are very important*

Dealing with Large Amounts of Data e.g., Big Data

Subsampling/Subsetting techniques

- sampling
- querying

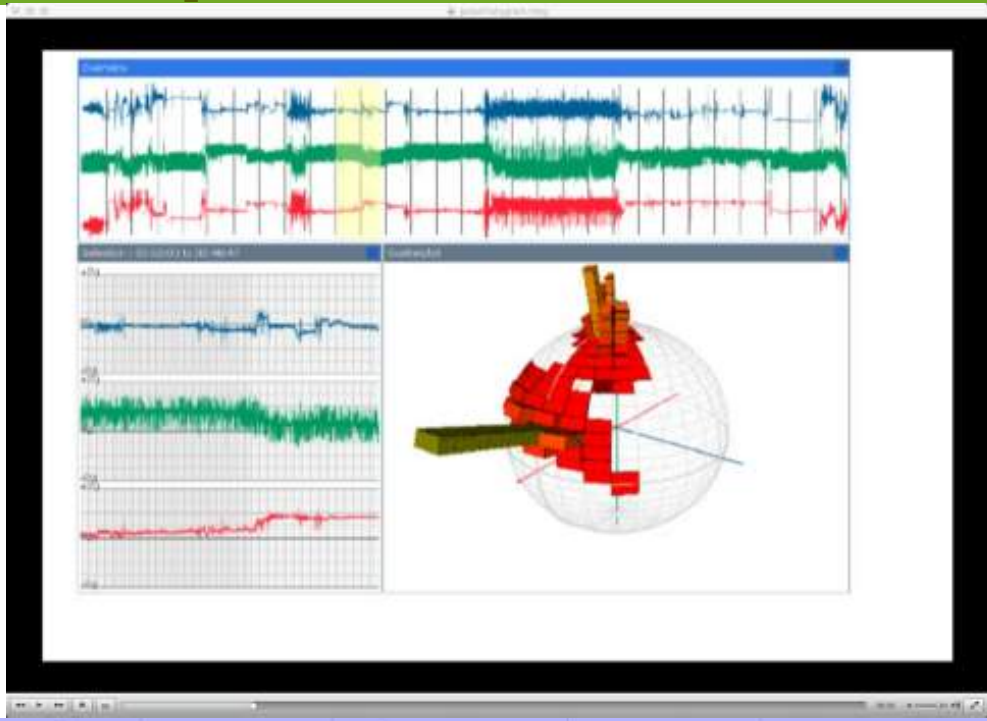
Segmentation techniques

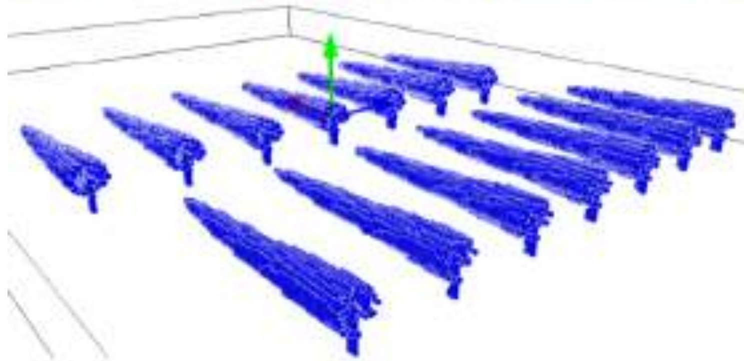
- segmentation: separation into subsets

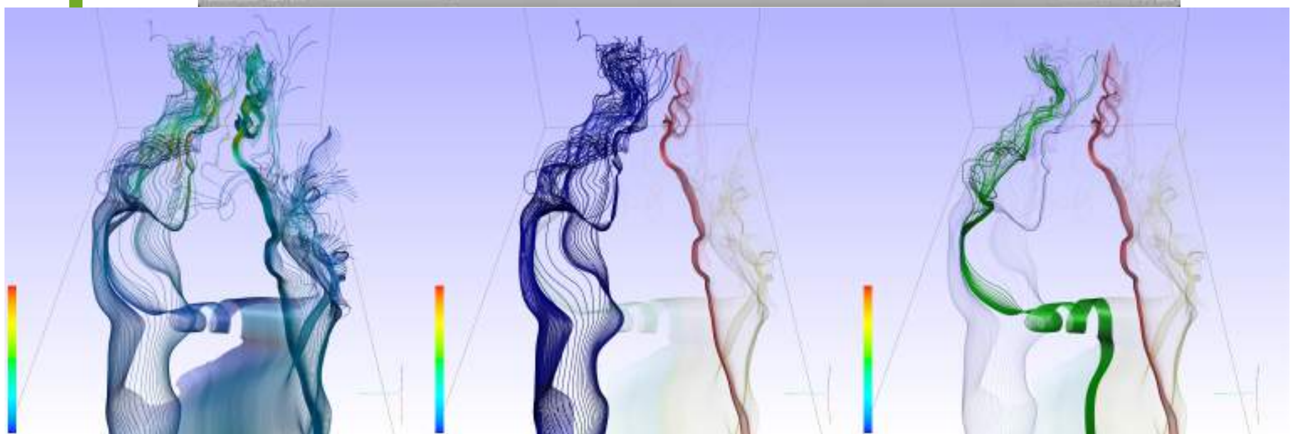
Aggregation techniques

- aggregation:
  sum, count, minimum, maximum, average...
- frequency-based techniques

Information and Scientific Visualization Examples







Hierarchical Data

Visualization of Hierarchies

Hierarchical Data

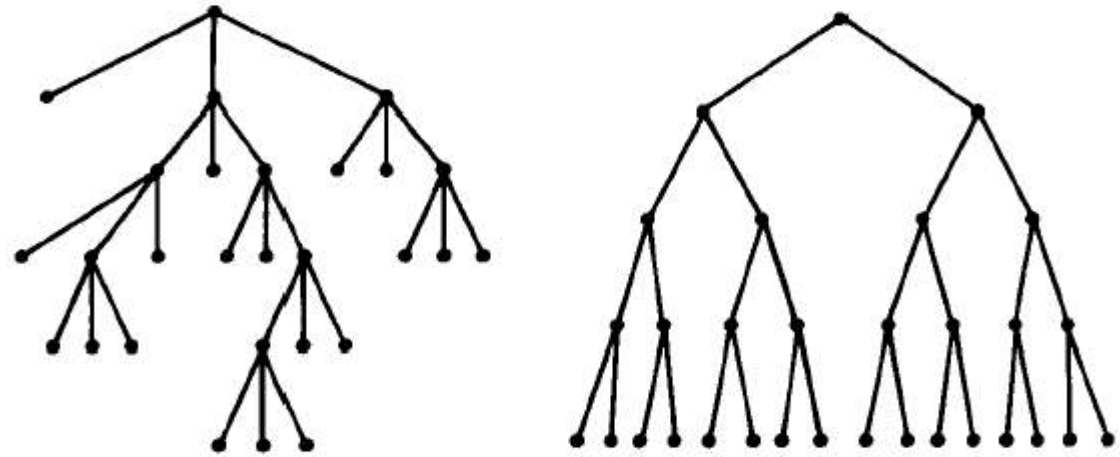
Visualization techniques:

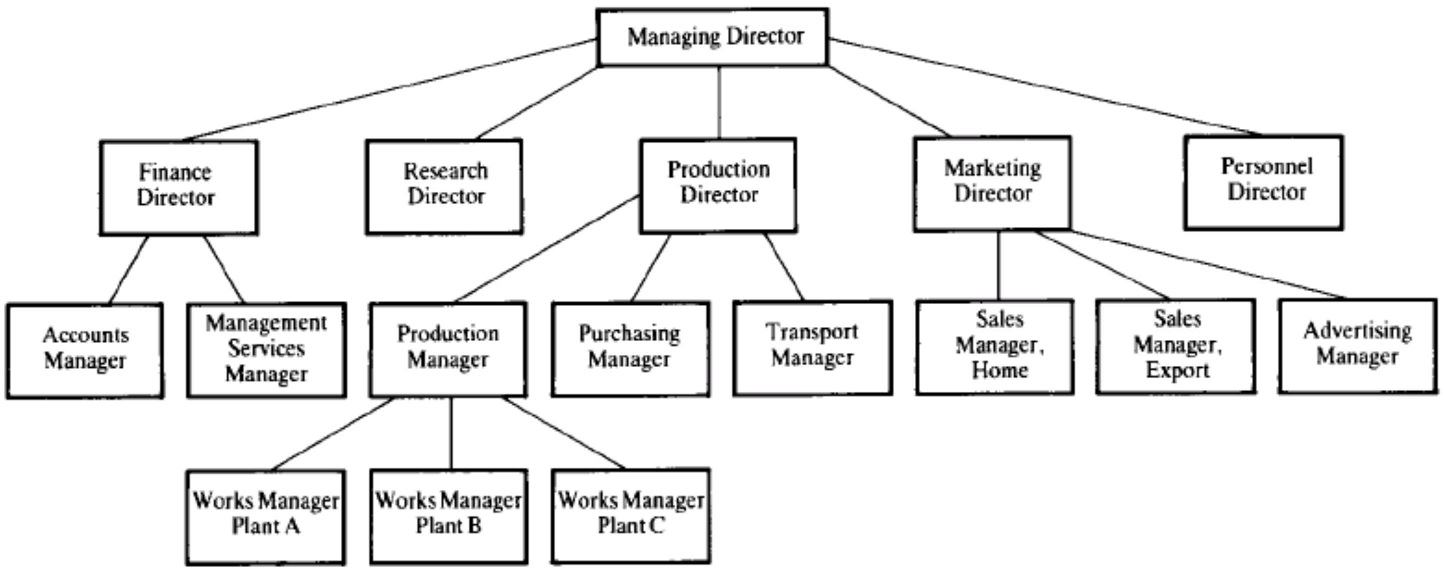
- conventional approaches such as Trees, File System Windows, Venn Diagrams
- Tree maps
- Examples

Hierarchical Data

Trees:

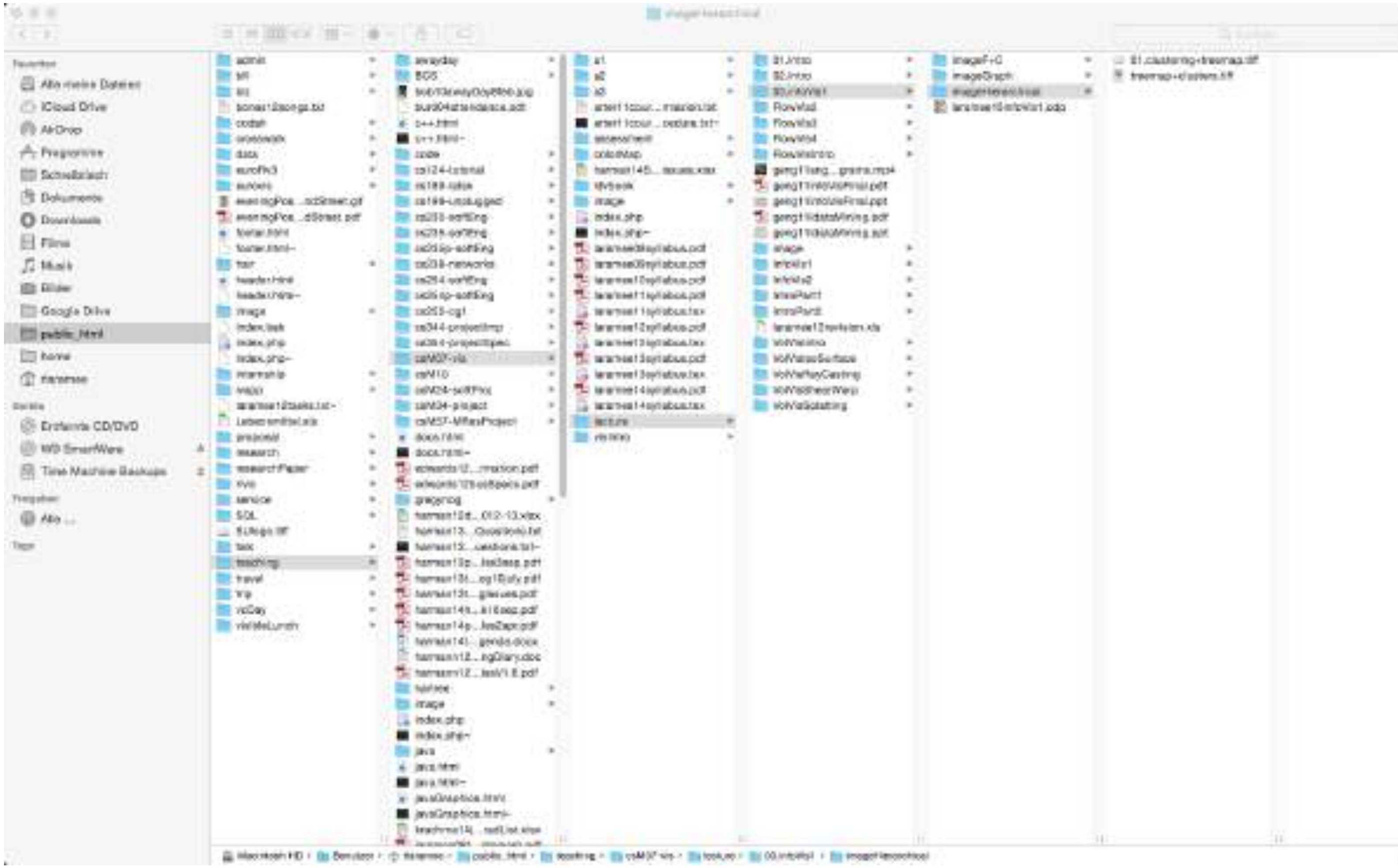
- ordered
- acyclic
- hierarchical





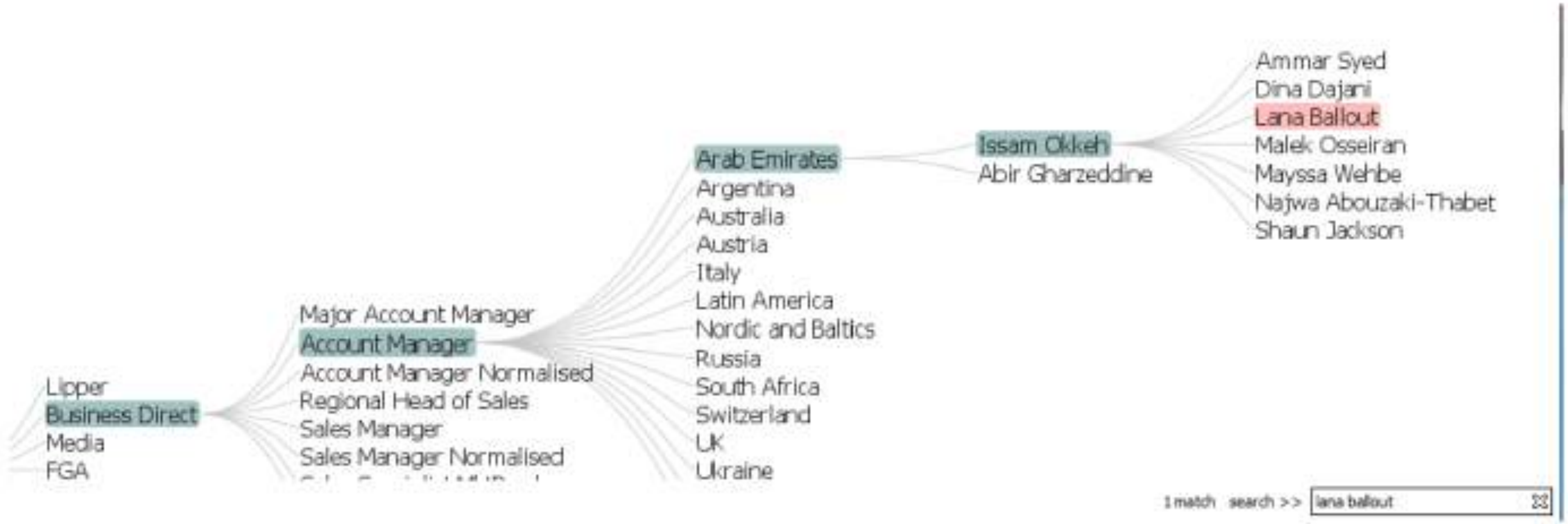
Horizontal Versus Vertical Layout

Vertical is traditional (previous slide). Horizontal aligns well with text.

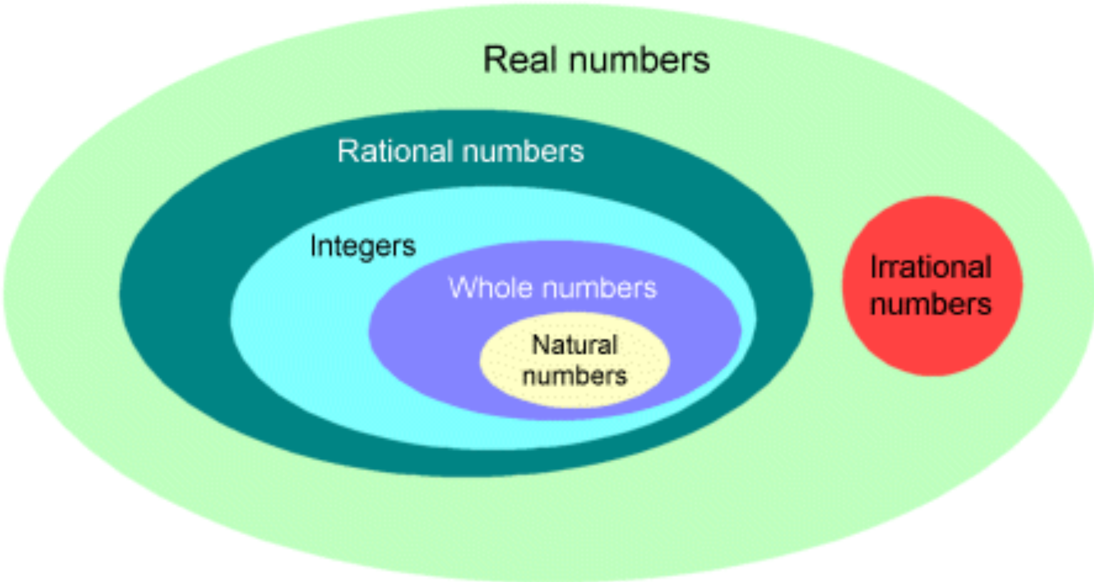


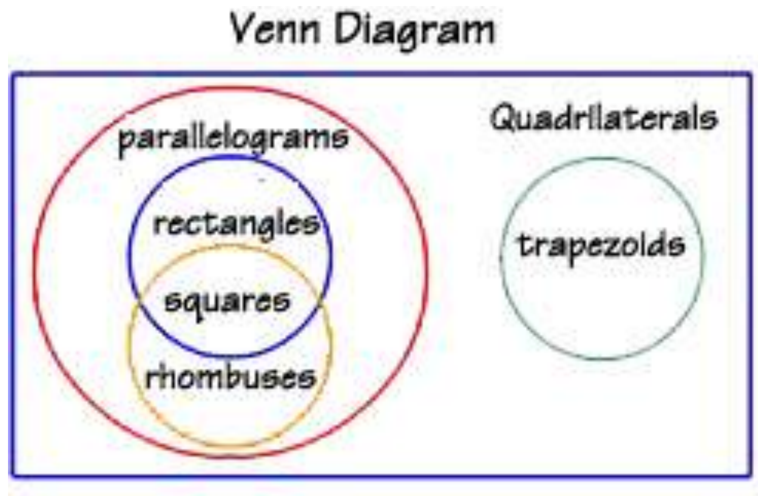
Word Tree Hierarchy

An interactive hierarchy of text (Geng et al., 2011)



Venn Diagrams





Author: Christopher J. Wells

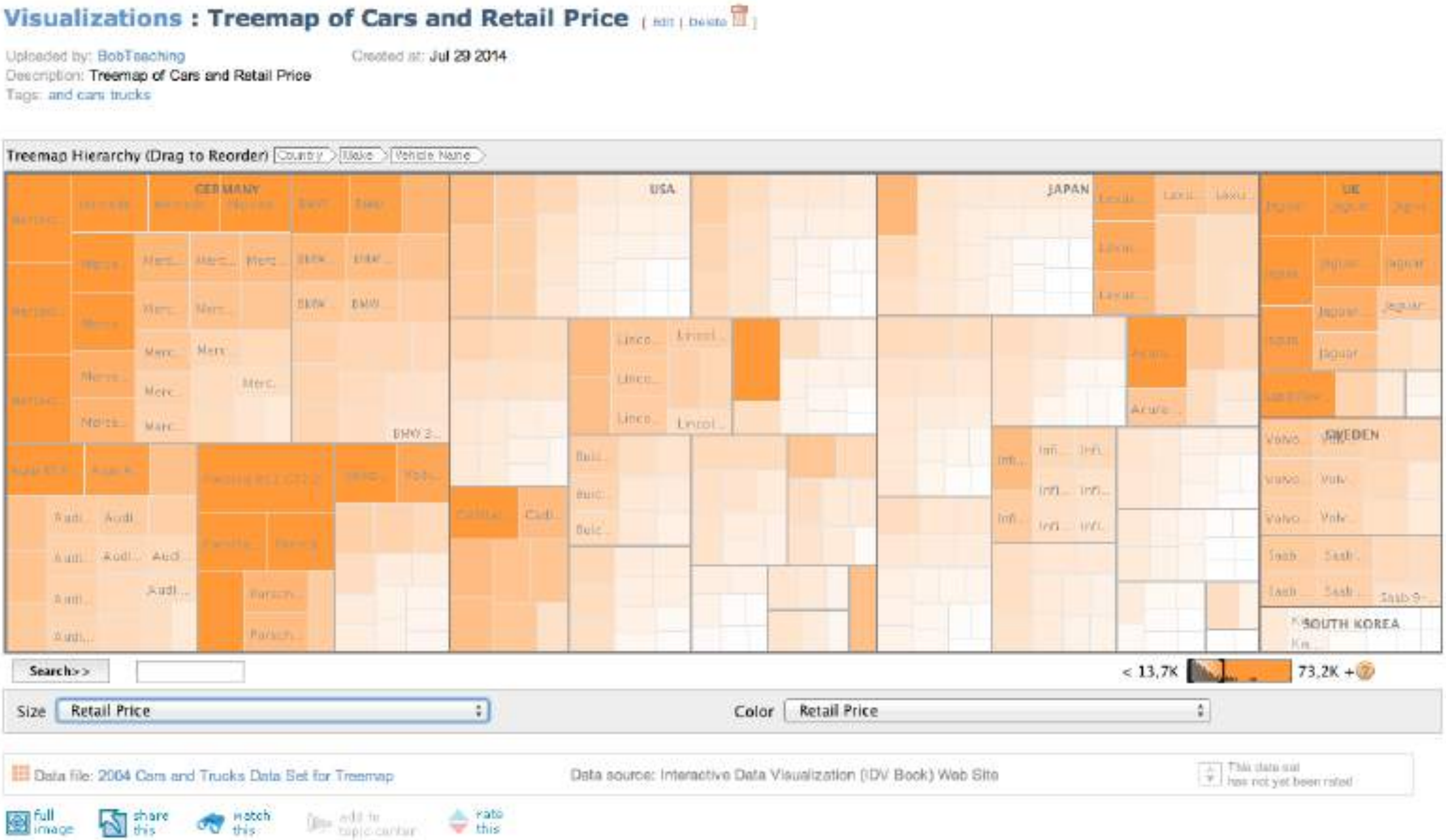
Website: TechnologyUK

Page title: Mathematics - About Numbers - Real Numbers

URL: http://www.technologyuk.net/mathematics/about_numbers/real_numbers.shtml

Accessed: 2015

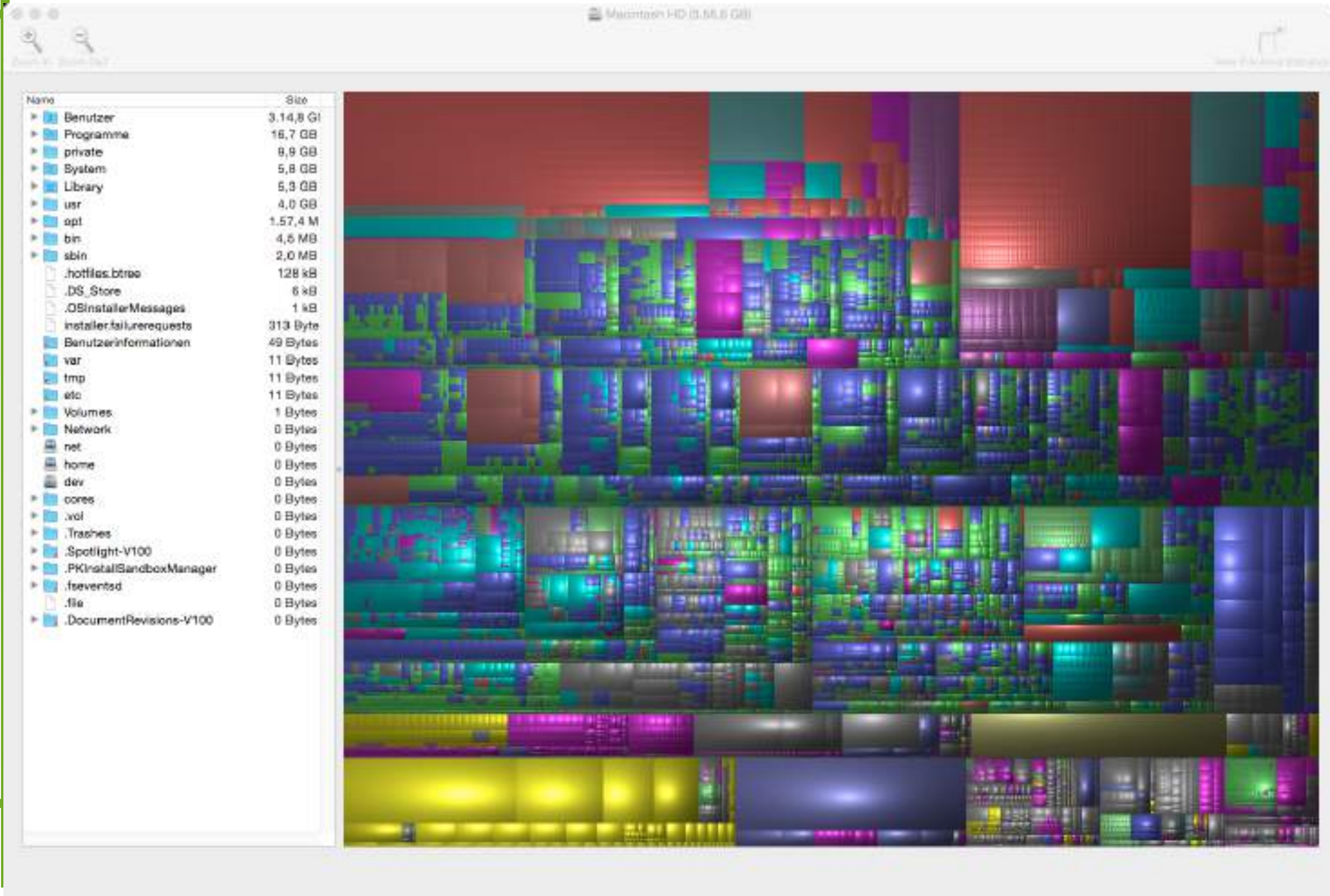
Tree Map (Improved Venn-Diagram)



Treemap Construction

- Construction demo

Tree Map: File System Example (Demo)



Ordered Treemap Layout (Demo)

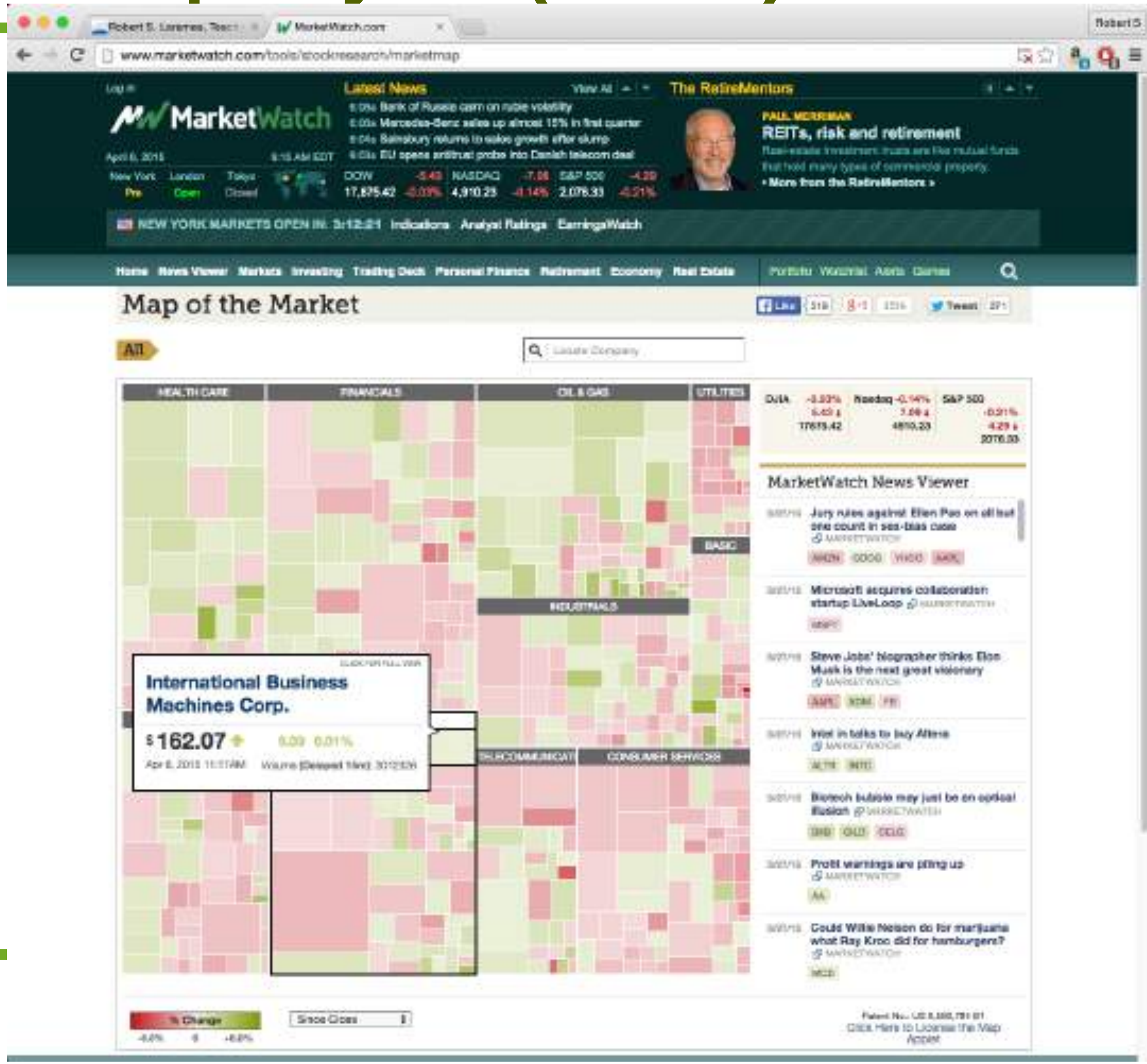
Wattenberg, 2001,

interaction

- semantic

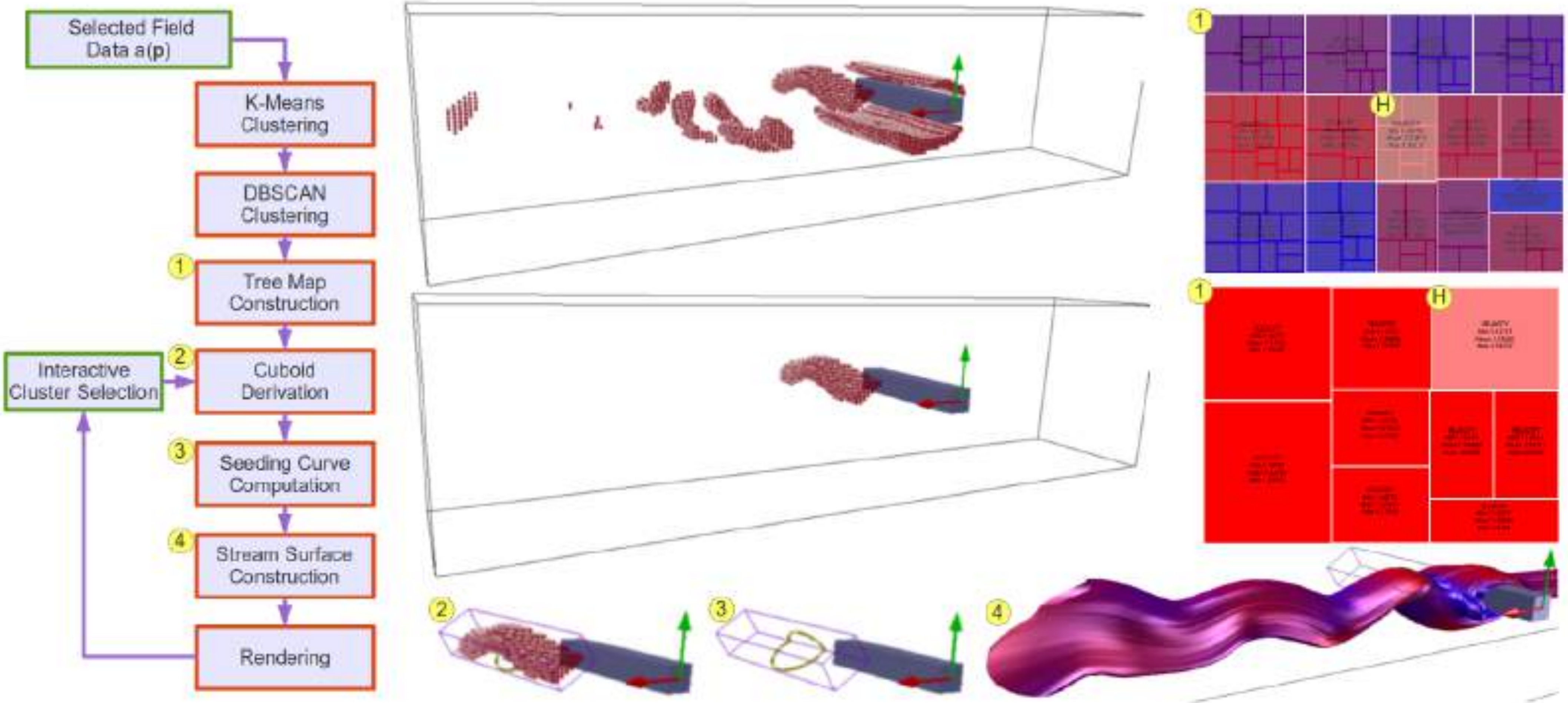
aggregation

- time dependent



Treemaps Combined with Flow Visualization

- Treemaps support cluster selection (Edmunds et al, 2014)



Word Tree Combined and Linked with Treemap

- Snapshots from interaction (Geng et al, 2011)

Focus and Context (F+C) Visualization

Emphasize Features,
Visualize the Rest as Context

Focus+Context Visualization

Problem and Motivation:

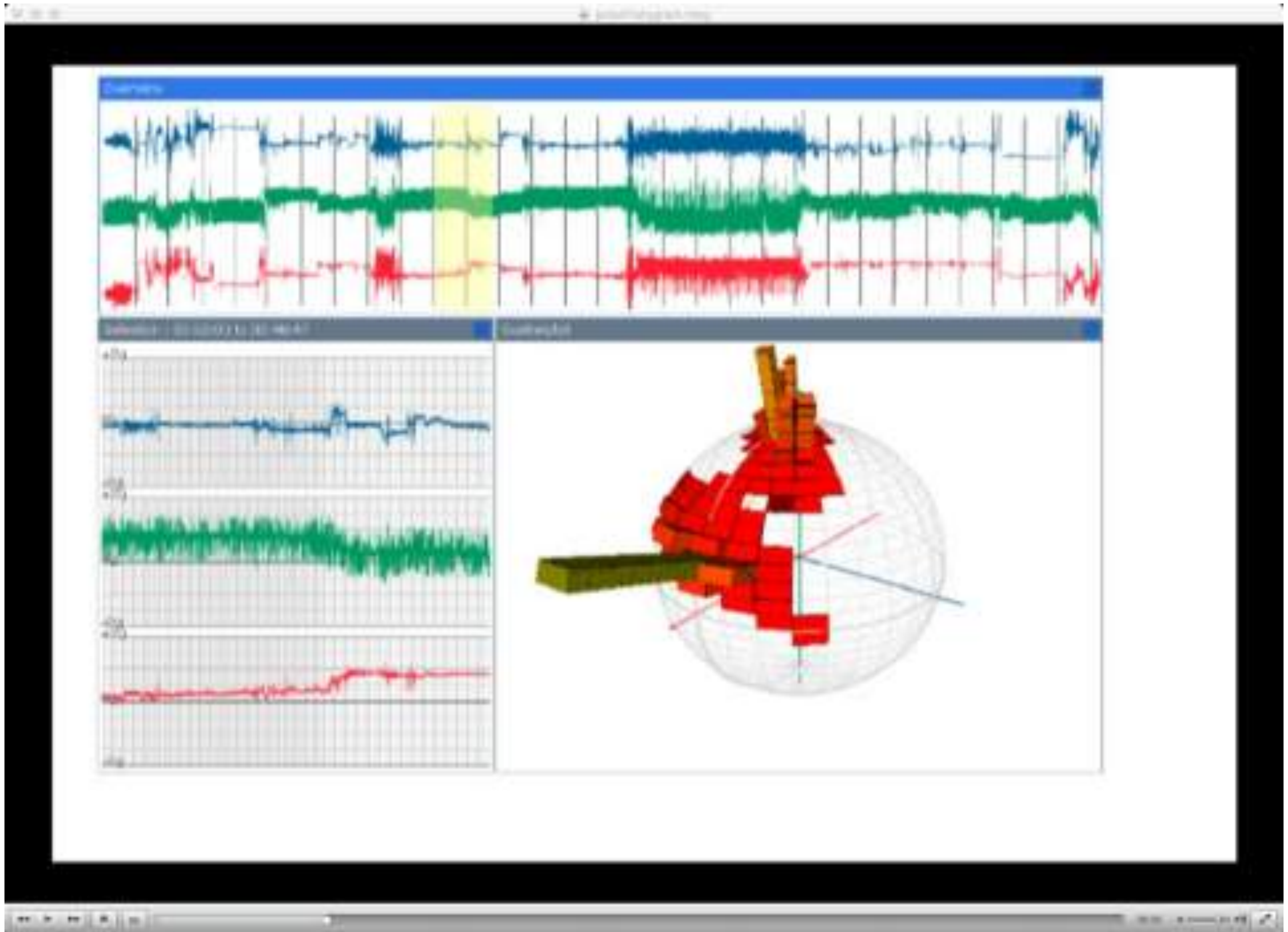
- large amounts of data, i.e. number of data samples exceeds display space
- varying importance and focus
- if only data in focus is shown, then problems stem from orientation and navigation
- if only overview is provided then no details can be obtained

Can both be included?

- multiple views, view linking
- Focus + context techniques, combinations of colormapping, varying opacity, varying zoom levels

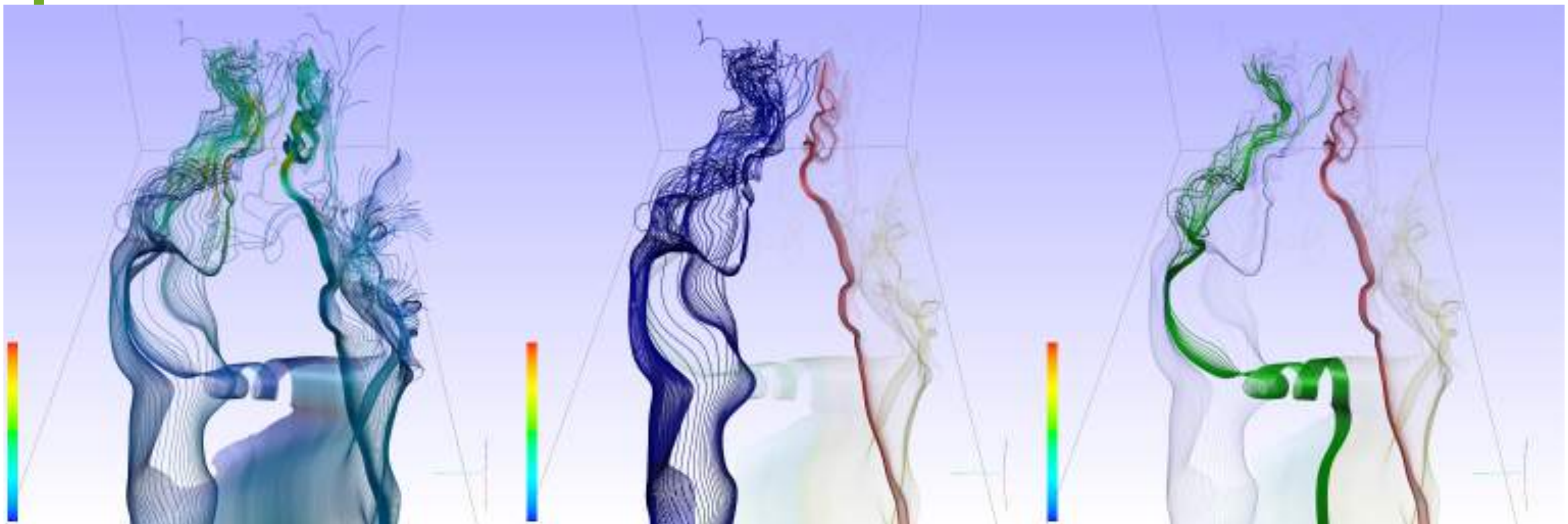
F+C for Time-Series Data

- Focus is shown bottom left and right. Context is shown above (Grundy et al., 2009)



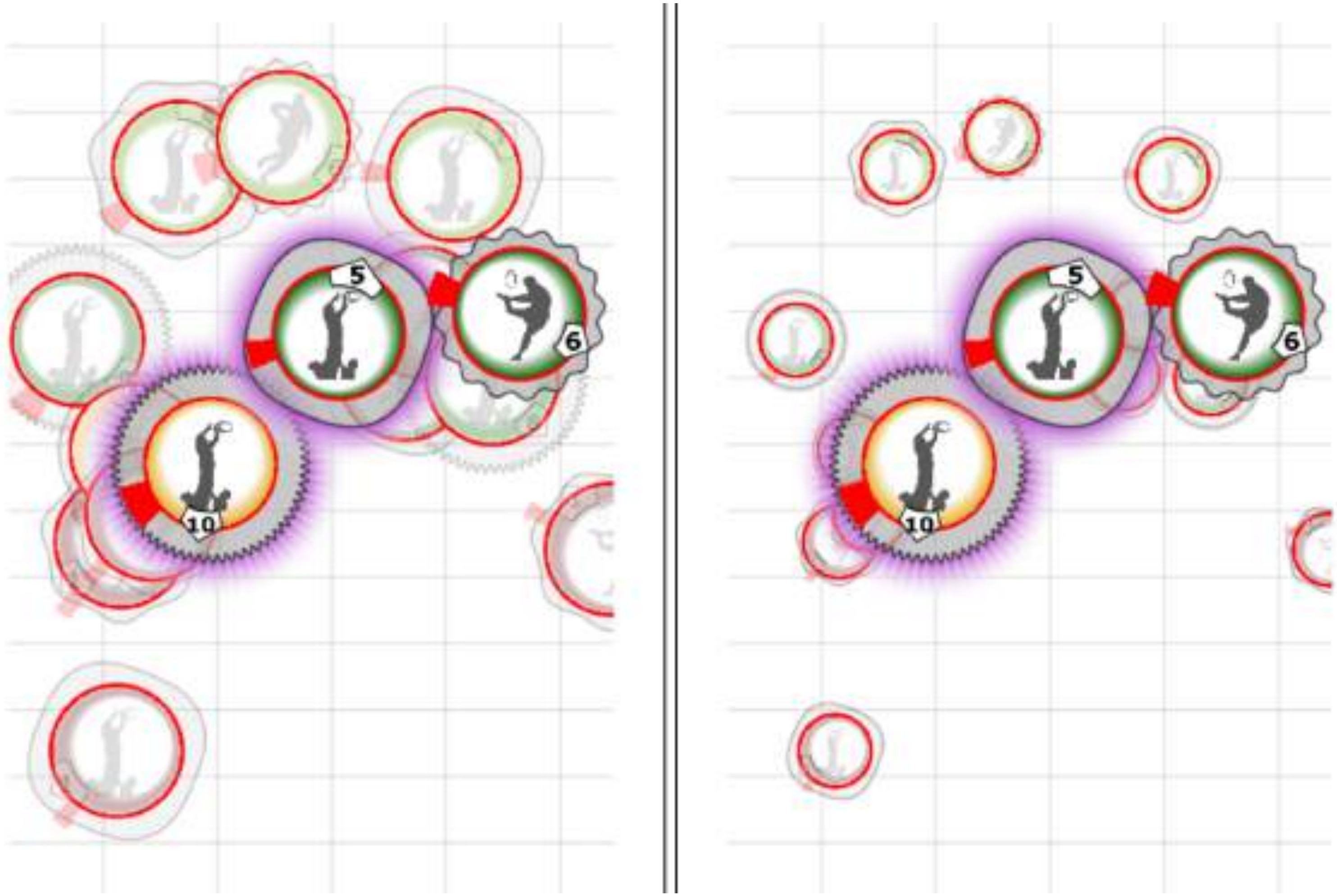
F+C for Visualizing Flow

- F+C streamlines (McLoughlin et al, 2013)



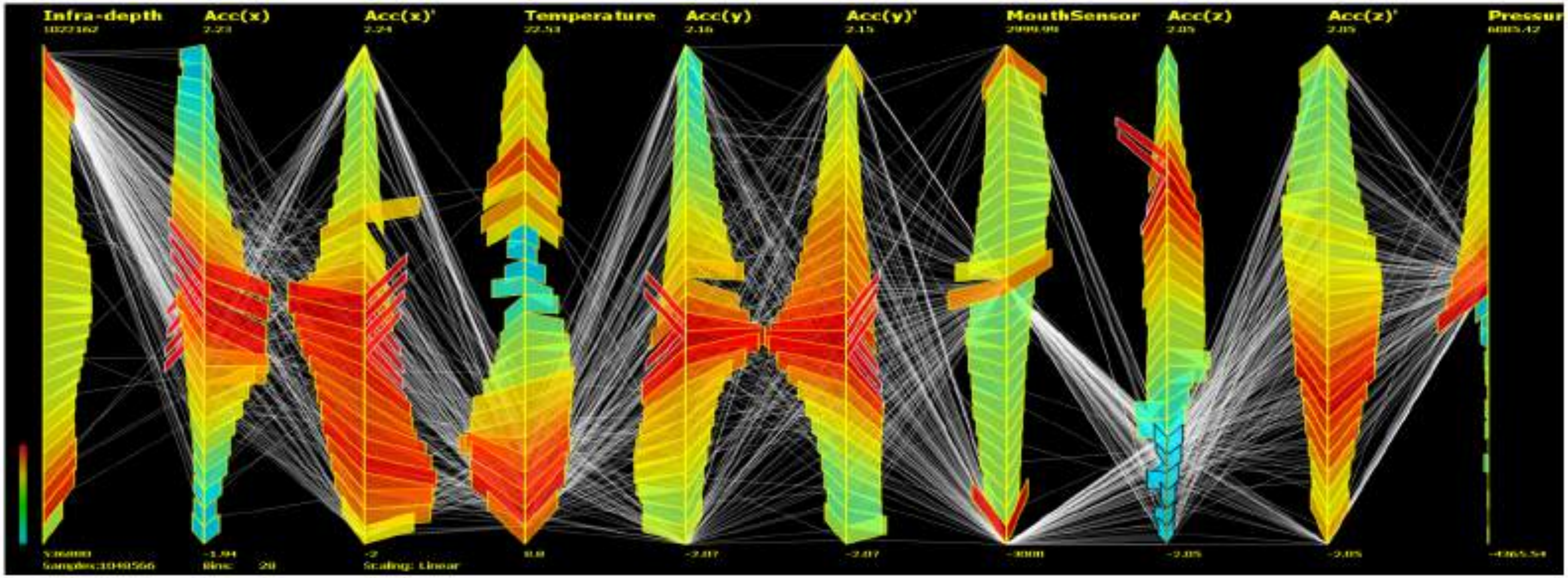
Event-based Glyphs

F+C Glyphs for visualizing rugby events (Chung et al, forthcoming)



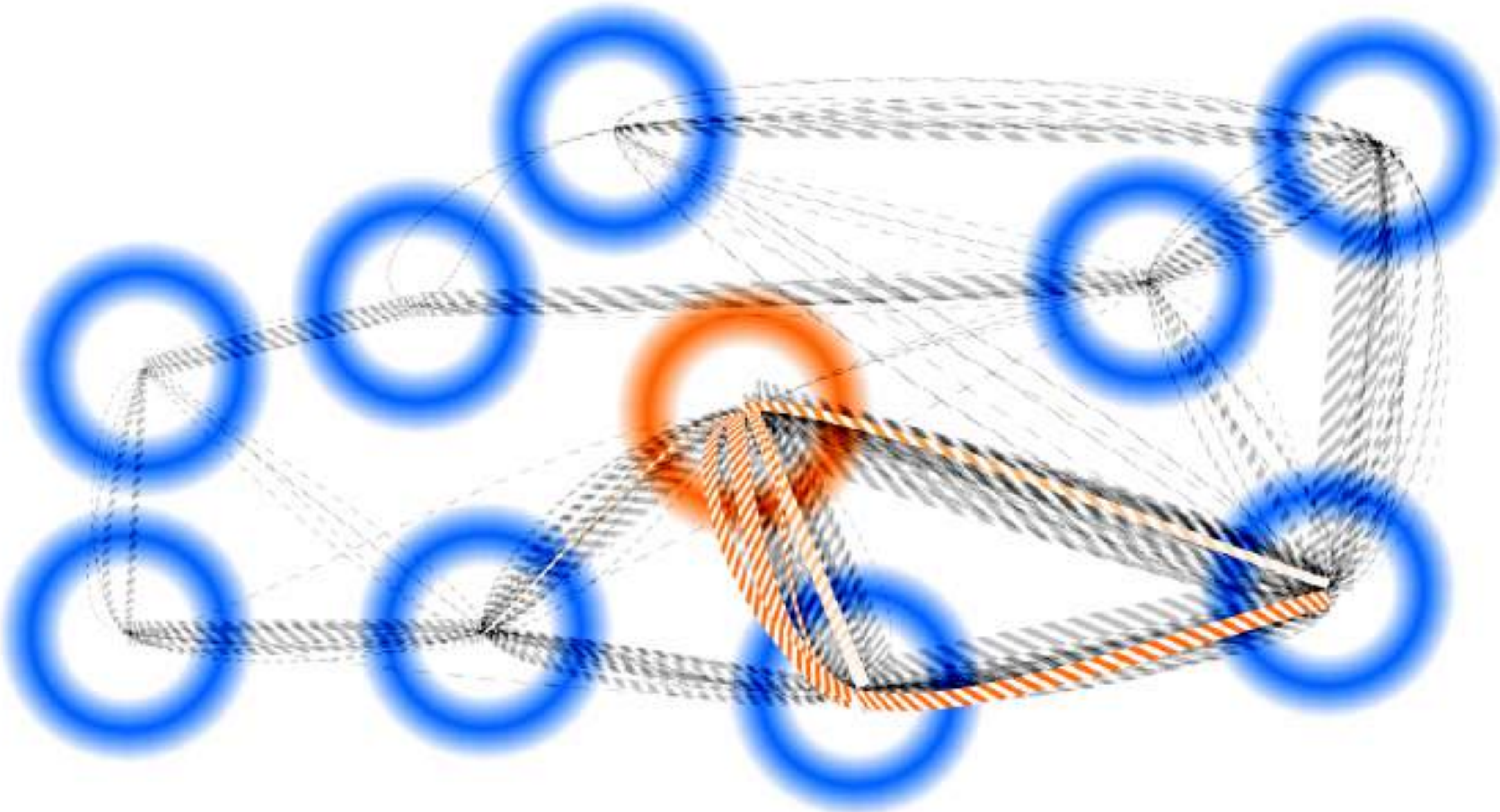
F+C Parallel Coordinates

F+C parallel coordinates and glyphs for visualizing time-series data (Geng et al, 2011)



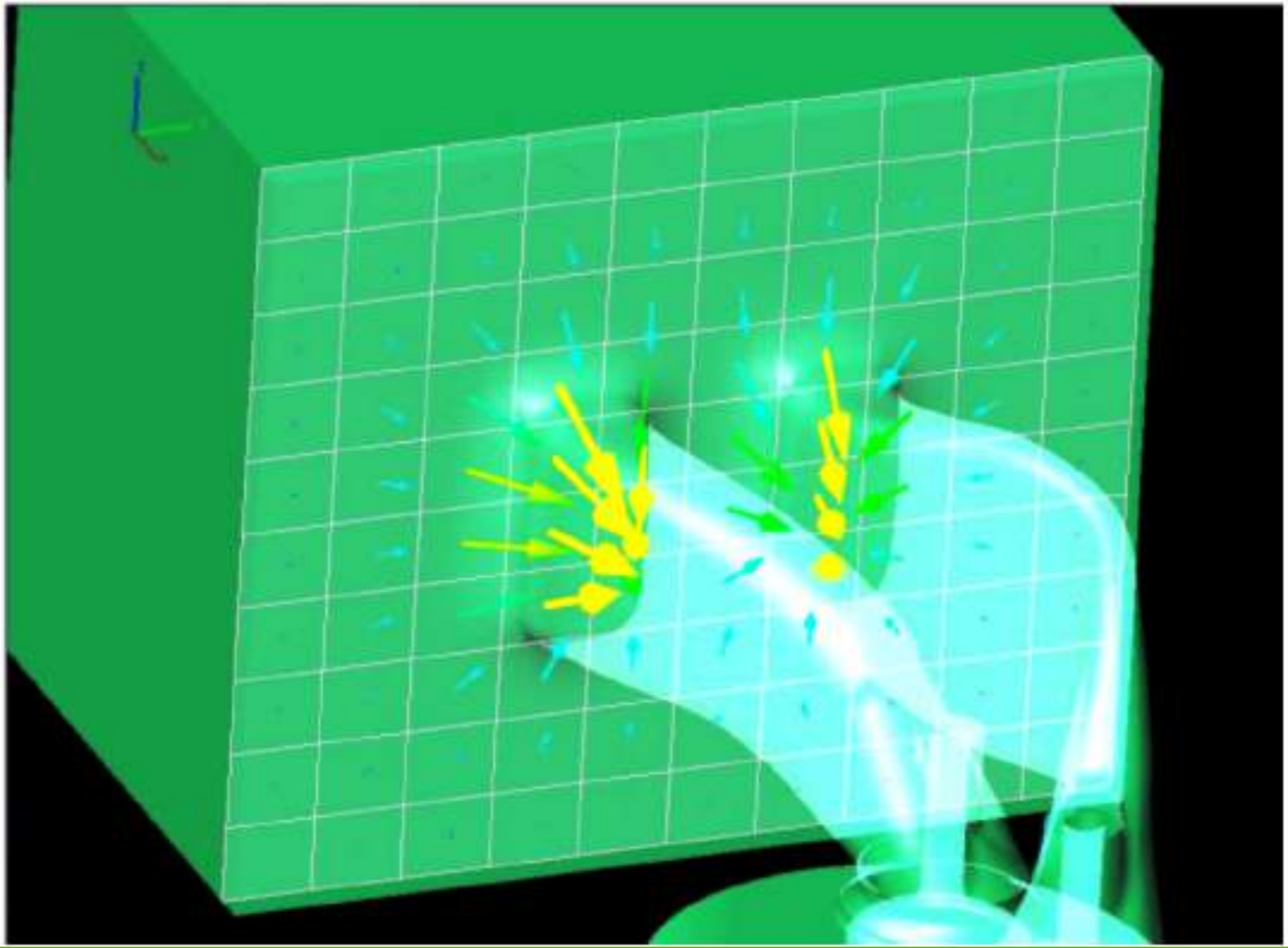
F+C Graphs

Context and focus graph edges (Blaas et al 2009)



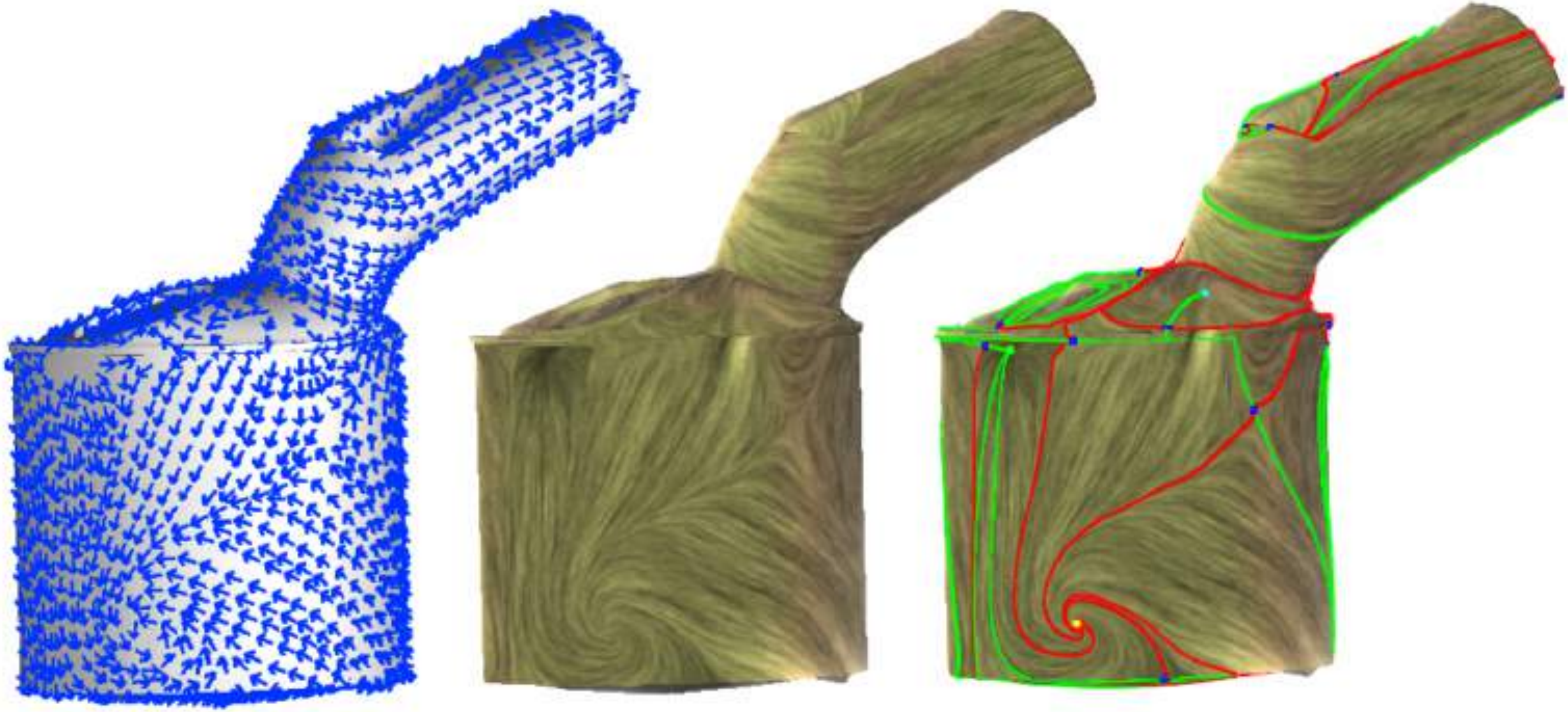
F+C Vector Glyphs

Focus vector glyphs with surface geometry context (Laramee 2003)



F+C Flow Visualization (Flow Topology)

Focus on flow features with surface flow context (Chen et al, 2007)

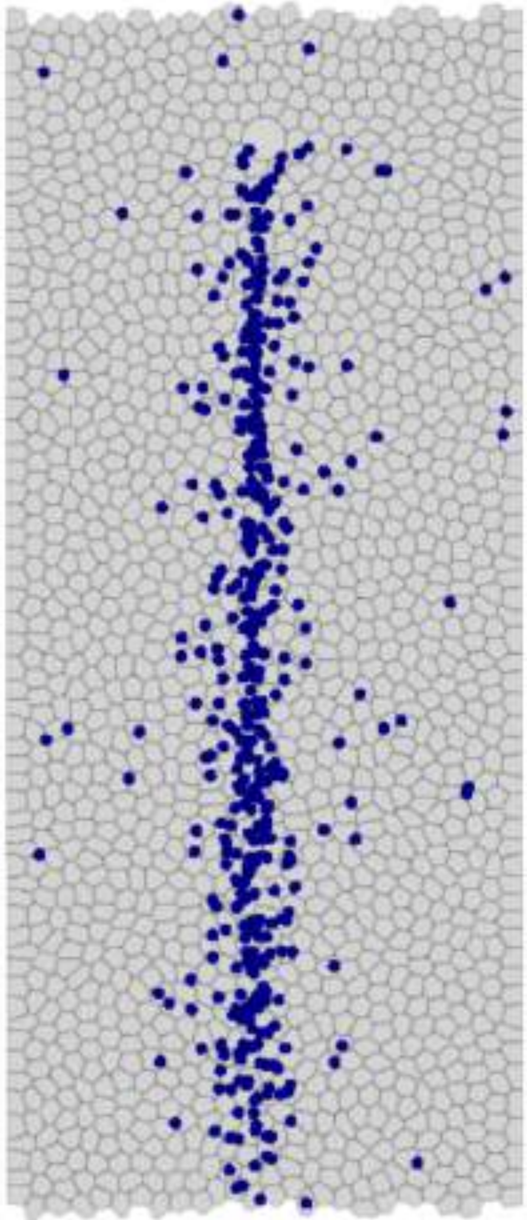


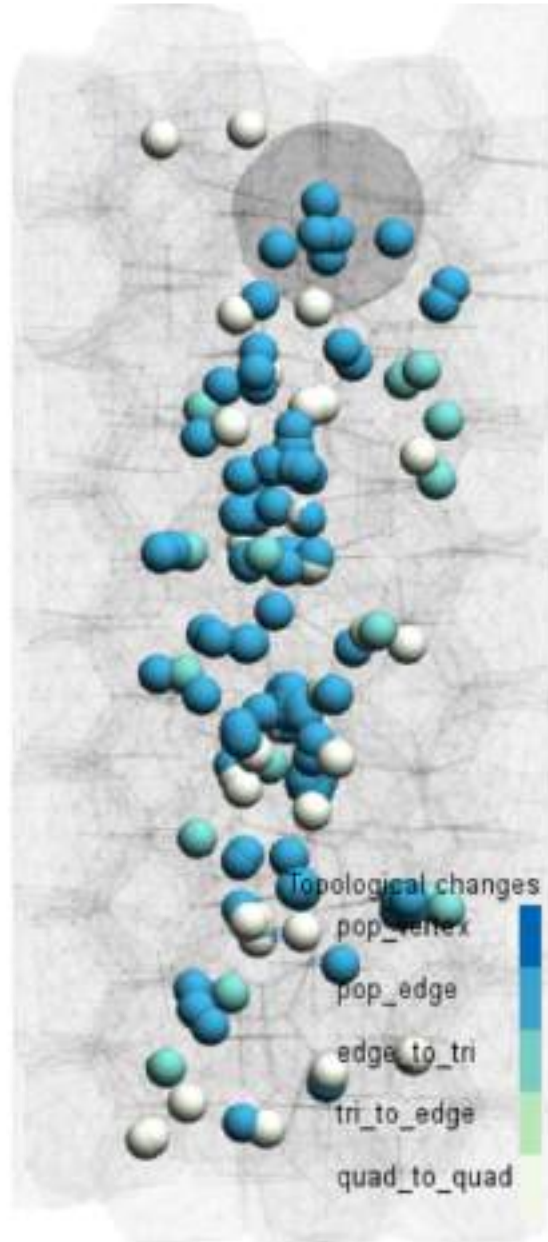
F+C Flow Visualization (Stream Surface)

Focus on flow features with surface geometry context (Garth et al, 2005)

F+C Foam Visualization (Foam Topology)

Focus on foam topology with bubbles as context (Lipsa et al, 2013)





Further Reading

For more information, see:

Interactive Data Visualization: Foundations, Techniques, and Applications, Chapter 8: **Visualization Techniques for Trees, Graphs, and Networks**, by Ward et al, 2010, A K Peters

All available online:

Zhao Geng, Gaurav Gathania, Robert S.Laramee, Zhenmin Peng, **Visual Analysis of Hierarchical Management Data**, EUROGRAPHICS 2011 Posters, 11-15 April 2011, Llandudno, Wales

Matthew Edumunds, Robert S. Laramee, Rami Malki, Ian Masters, Yunai Wang, Guoning Chen, Eugene Zhang, and Nelson Max, **Interactive Stream Surface Placement: A Hybrid Clustering Approach Supported by Treemaps**, International Conference on Information Visualization Theory and Applications (IVAPP) 2014, pages 347-355, 5-8 January, 2014, Lisbon, Portugal

Edward Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily L. C. Shepard, **Visualisation of Sensor Data from Animal Movement**, Computer Graphics Forum (CGF), Vol. 28, No. 3, 2009, pages 815-822

Tony McLoughlin, Mark W Jones, Robert S. Laramee, Rami Malki, Ian Masters, and Charles D. Hansen, **Similarity Measures for Enhancing Interactive Streamline Seeding**, IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG), Vol 19, No 8, pages 1342-1353 (August 2013)

David H.S. Chung, Phillip A. Legg, Matthew L. Parry, Rhodri Bown, Iwan W Griffiths, Robert S. Laramee, and Min Chen, **Glyph Sorting: Interactive Visualization for Multi-dimensional Data**, Information Visualization, forthcoming

Further Reading

Zhao Geng, Zhenmin Peng, Robert S.Laramee, Rick Walker, and Jonathan C. Roberts, **Angular Histograms: Frequency-Based Visualizations for Large, High-Dimensional Data**, IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG), Vol. 17, No. 12, December 2011, pages 2572-2580

Jorik Blaas, Charl Botha, Edward Grundy, Mark W. Jones, Robert S. Laramee, and Frits H. Post, **Smooth Graphs for Visual Exploration of Higher-Order State Transitions**, IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG), Vol. 15, No. 6, November/December 2009, pages 969-976

Robert S. Laramee, **FIRST: A Flexible and Interactive Resampling Tool for CFD Simulation Data** in Computers & Graphics, Vol. 27, No. 6, pages 905-916, 2003

Guoning Chen, Robert S. Laramee, and Eugene Zhang, **Advanced Visualization of Engine Simulation Data Using Texture Synthesis and Topological Analysis**, NAFEMS World Congress (NWC) Conference Proceedings, The International Association for the Engineering Analysis Community

Robert S. Laramee, Christoph Garth, Helmut Doleisch, Juergen Schneider, Helwig Hauser, and Hans Hagen, **Visual Analysis and Exploration of Fluid Flow in a Cooling Jacket**, in Proceedings of IEEE Visualization (IEEE Vis 2005), pages 623-630, October 23-28, 2005, Minneapolis, Minnesota

Dan R. Lipşa, Robert S. Laramee, Tudur Davies, and Simon Cox, **Visualizing 3D, Time-dependent Foam Simulation Data**, Advances in Visual Computing, Lecture Notes in Computer Science LNCS, Volume 8033 (Proceedings of the 9th International Symposium on Visual Computing (ISVC) 2013 29-31 July 2013, Crete, Greece), pages 255-265

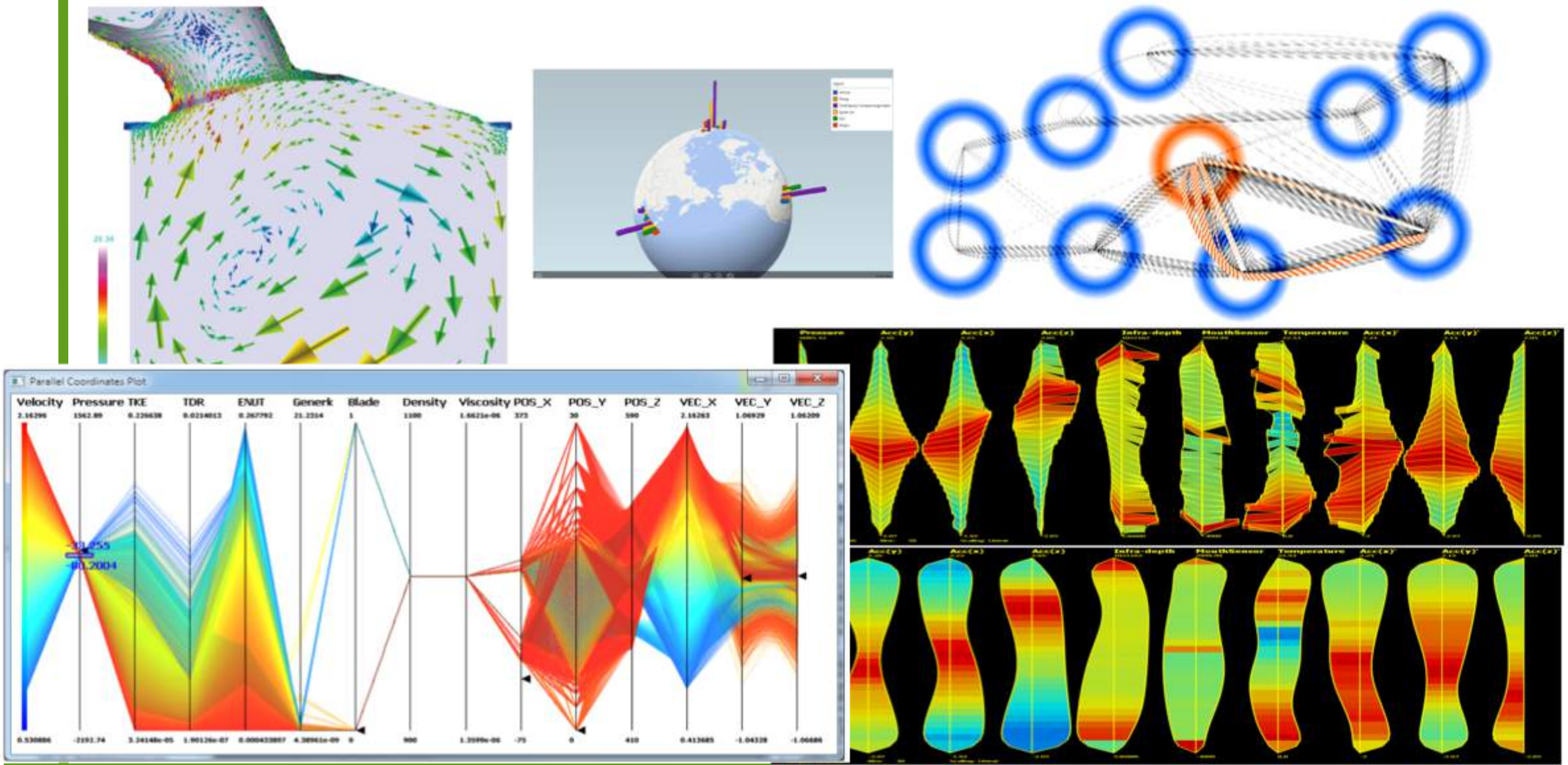
Acknowledgements

We thank the following:

Jorik Blaas, Charl Botha, Rhodri Bown, Guoning Chen, Min Chen, David Chung, Simon Cox, Tudur Davies, Helmut Doleisch, Matthew Edmunds, Christoph Garth, Zhao Geng, Gaurav Gathania, Eduard Grundy, Hans Hagen, Iwan Griffiths, Charles Hansen, Helwig Hauser, Daniel Keim, Mark Jones, Phillip Legg, Dan Lipsa, Rami Malki, Ian Masters, Nelson Max, Tony McLoughlin, Wolfgang Meyer, Matthew Parry, Frits Post, Jonathan Roberts, Juergen Schneider, Rick Walker, Yunai Wang, Rory Wilson, Eugene Zhang

Visualization Lecture

With a Focus on Information Visualization, Part 2



Review of Previous Lecture

Focus on Information Visualization, part 1:

- Introduction + examples
- Hierarchical data
- Conventional representations, venn diagrams
- Tree maps, word trees
- Various examples and applications

Overview of Current Lecture

Information visualization, part 2:

- graphs
- optimizations, high-order transitions, examples
- *n*-dimensional data
- scatterplots, scatterplot-matrices, parallel coordinates, chord diagrams and visualizations
- icons
- Chernoff-faces, high-dimensional glyphs
- examples from event-based data
- interaction techniques
- Filtering, brushing, selection

Graphs: A Short and Convenient Introduction

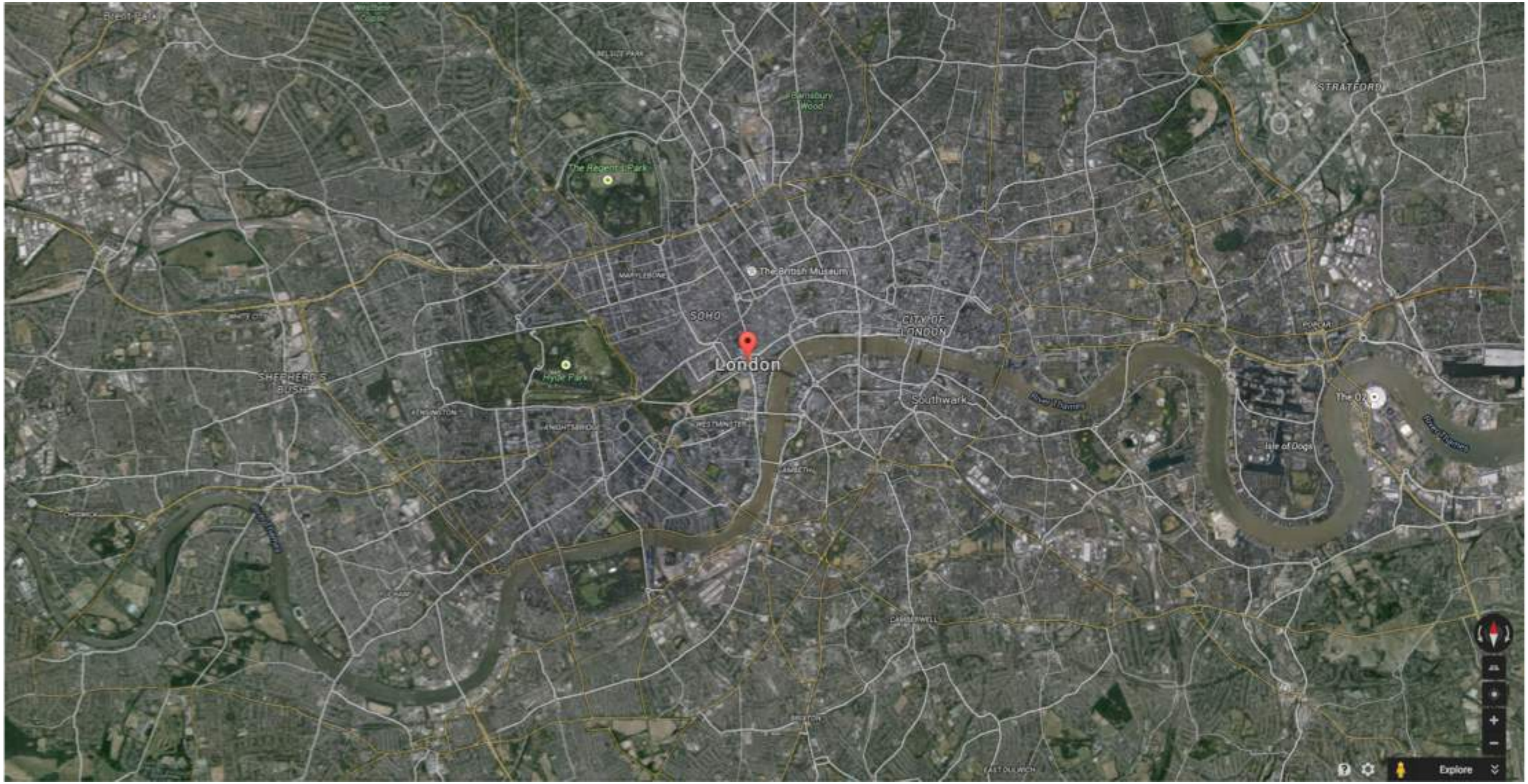
Graph Optimizations and Challenges

Maximize Graph Legibility
- Optimize Length of Edges
- Minimize Edge Crossings
- Maximize Edge Crossing Angles (right angles)

Some Challenges:
- Large Graphs
- Time-Dependent Graphs
- Hierarchcical Graphs
- Label Placement
- Multivariate Graphs

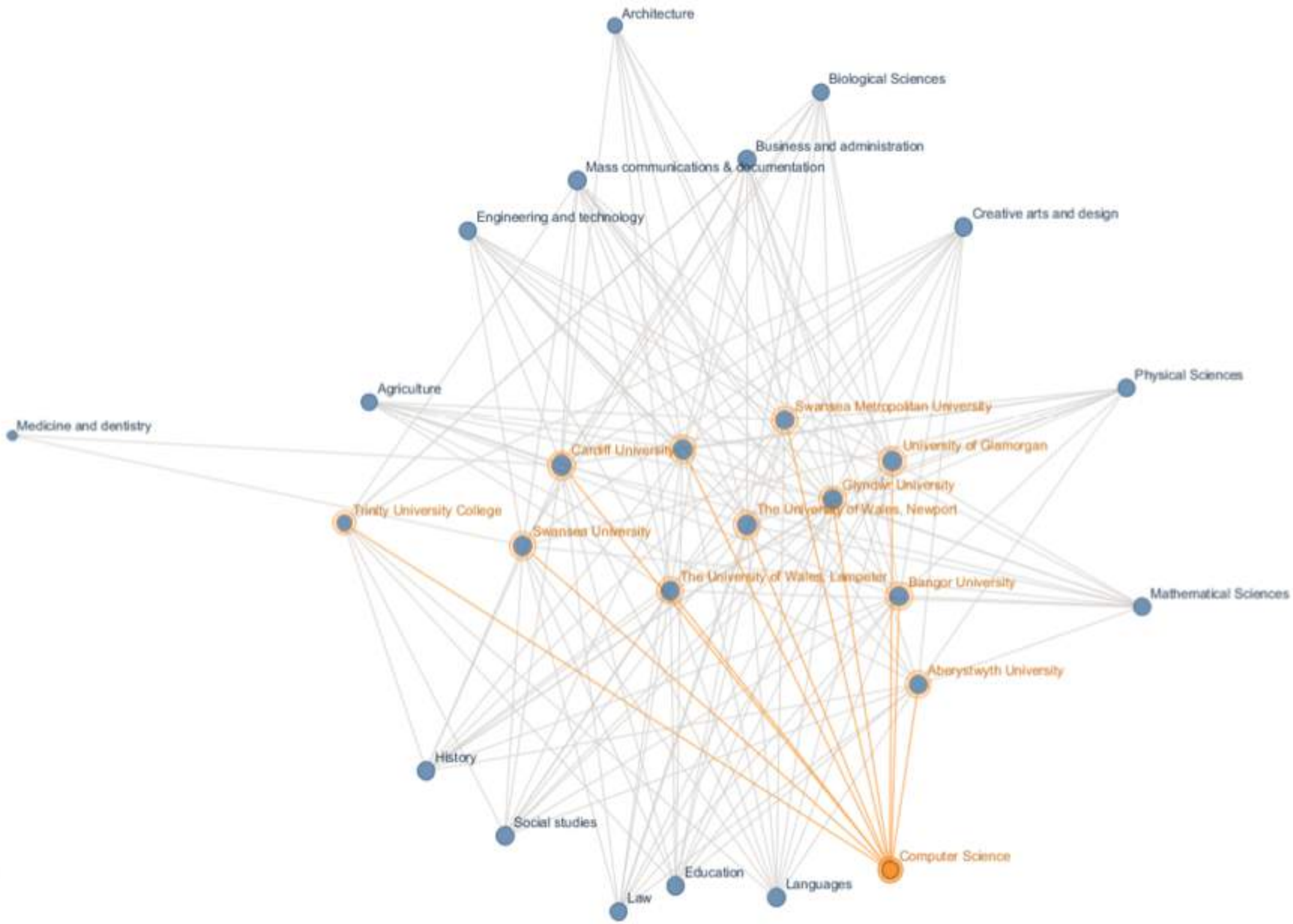
A Famous Graph, Manual Layout

London, Satellite View



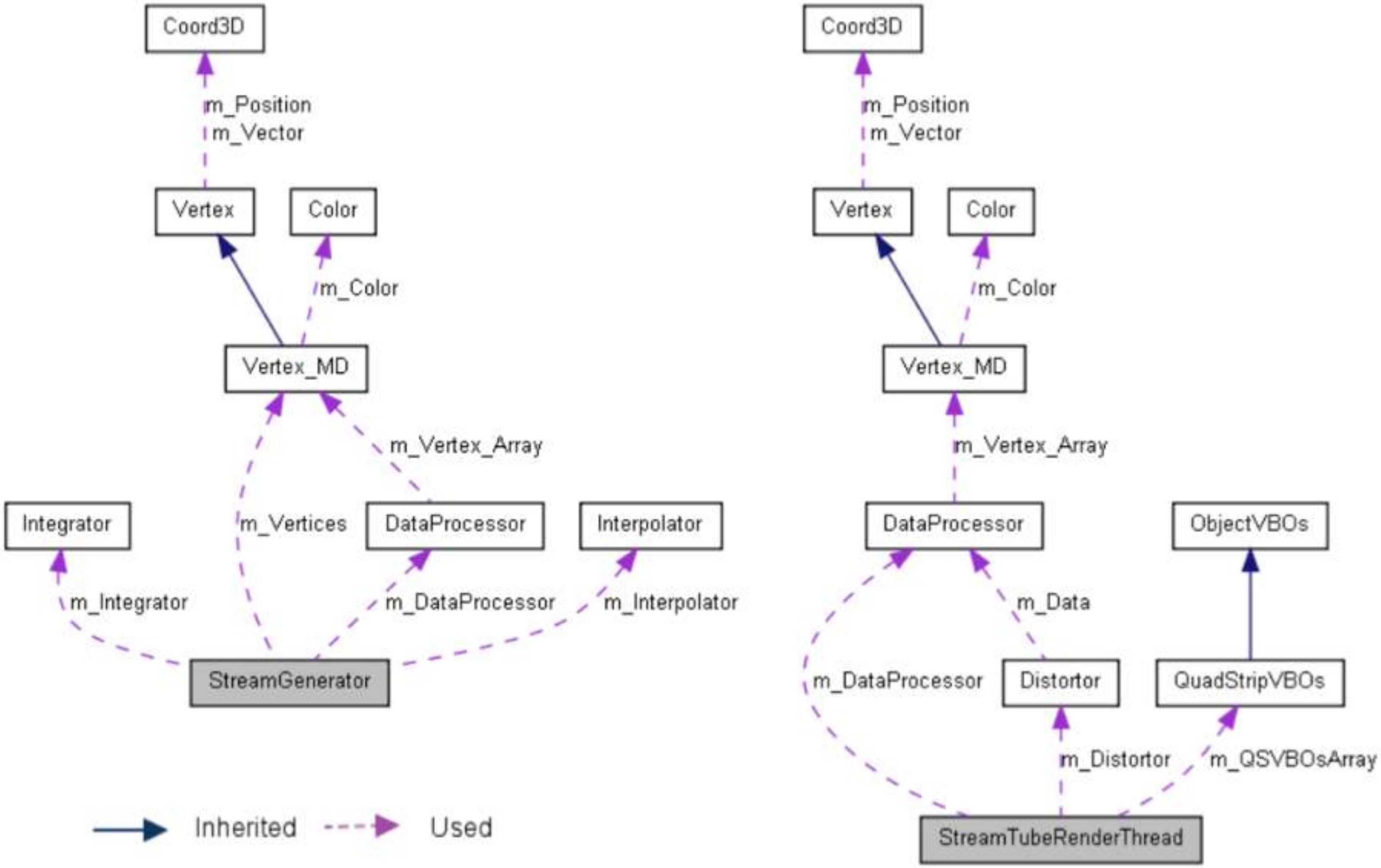
Graph Layout Algorithms

Automatic layout from ManyEyes: node labels and interaction



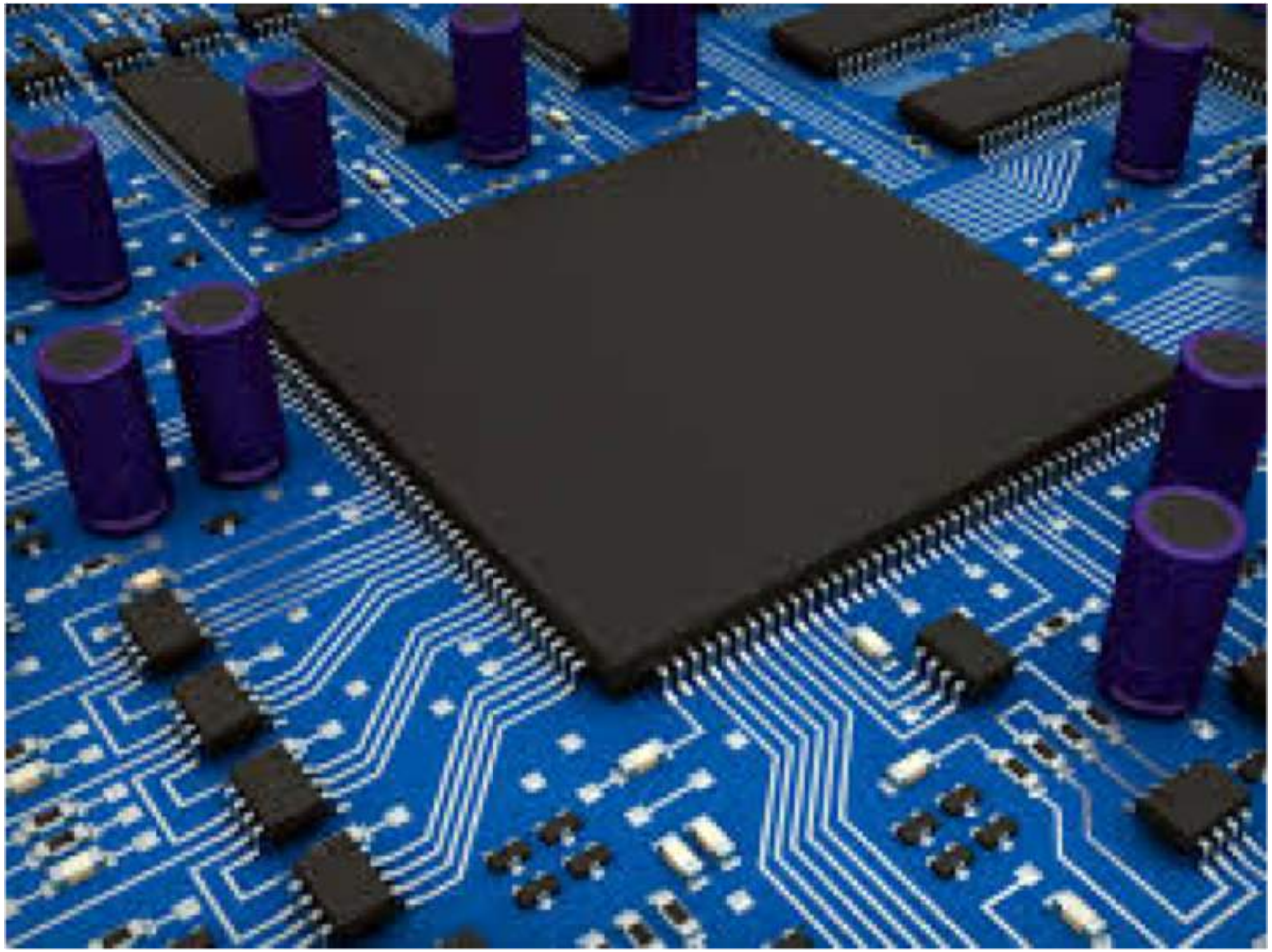
Graph Layout Algorithms

Automatic layout from Doxygen with node and edge labels



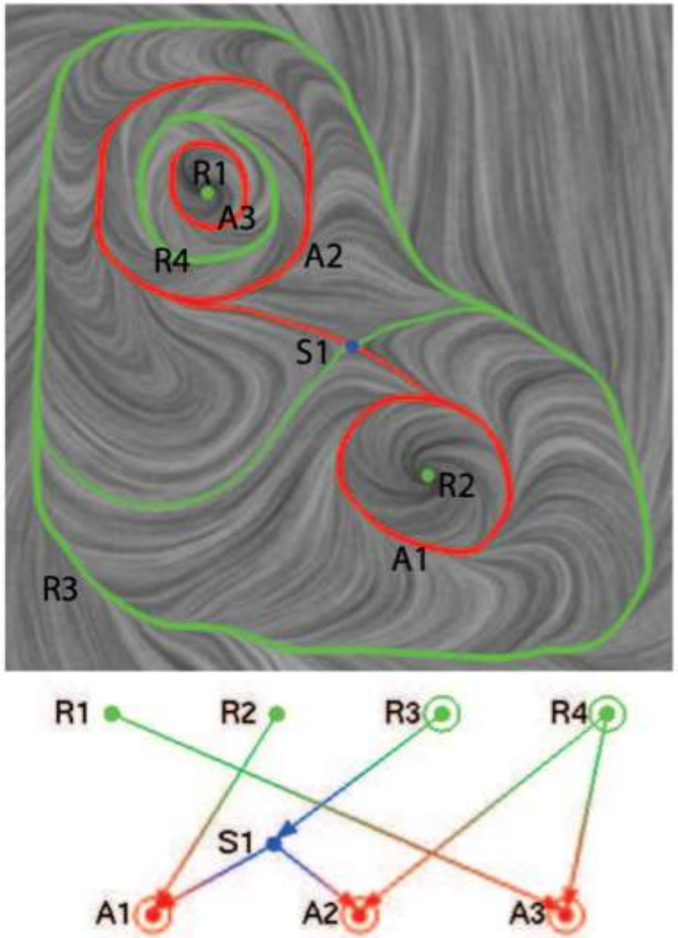
Example from VLSI

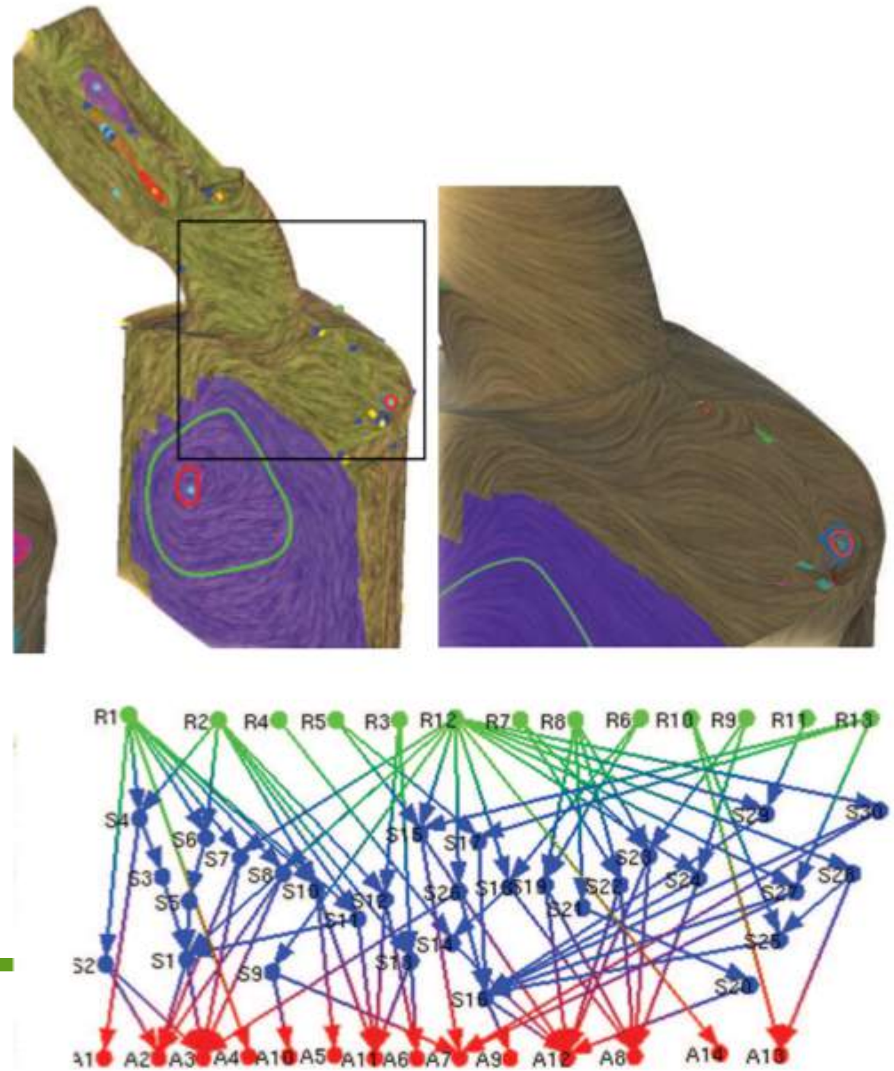
Very Large Scale Integration



Example and Application to Flow Visualization

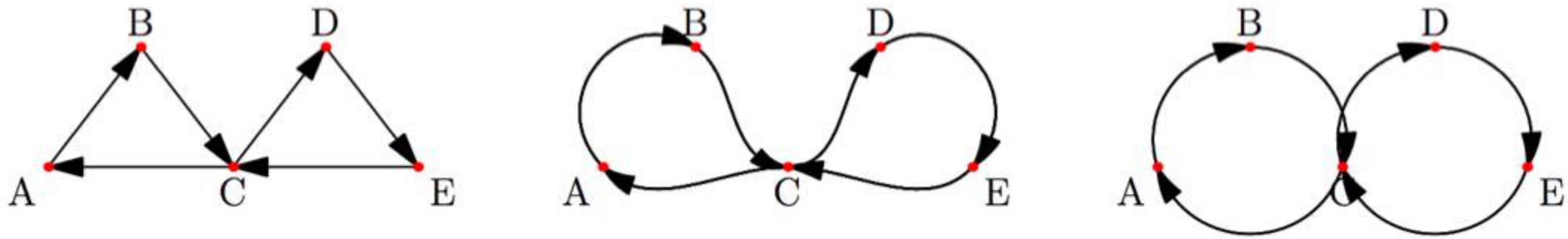
Graphs and Flow Topology (Chen et al., 2008)





Higher Order Graphs

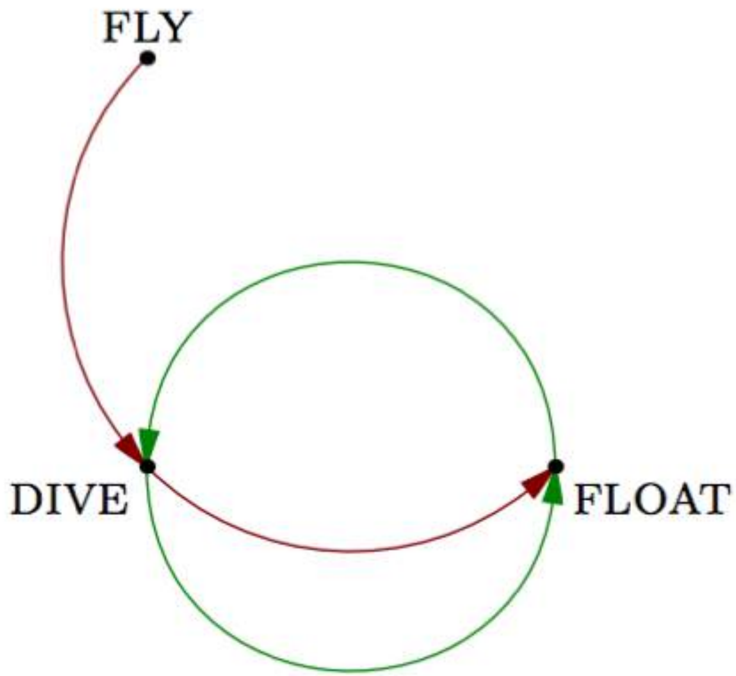
Higher Order edge transitions (Blaas et al., 2009)

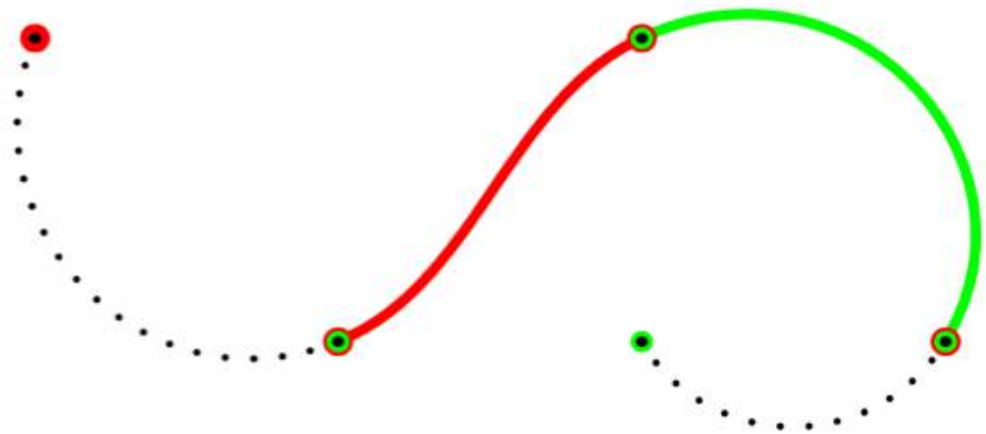


First Order | Higher Order (#1) | Higher Order (#2)

Case Study: Higher Order Graphs for Marine Biology

Applied to Penguin Dive. Case study from Blaas et al 2009.



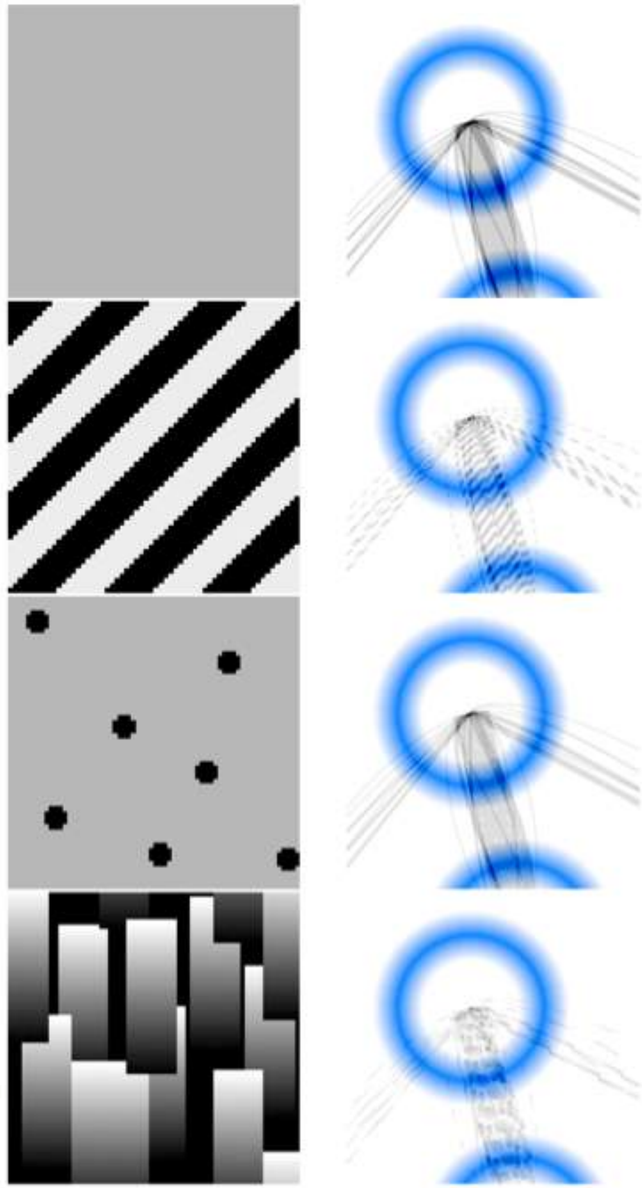


Texture Mapped Graph Edges

Different texture maps add information.

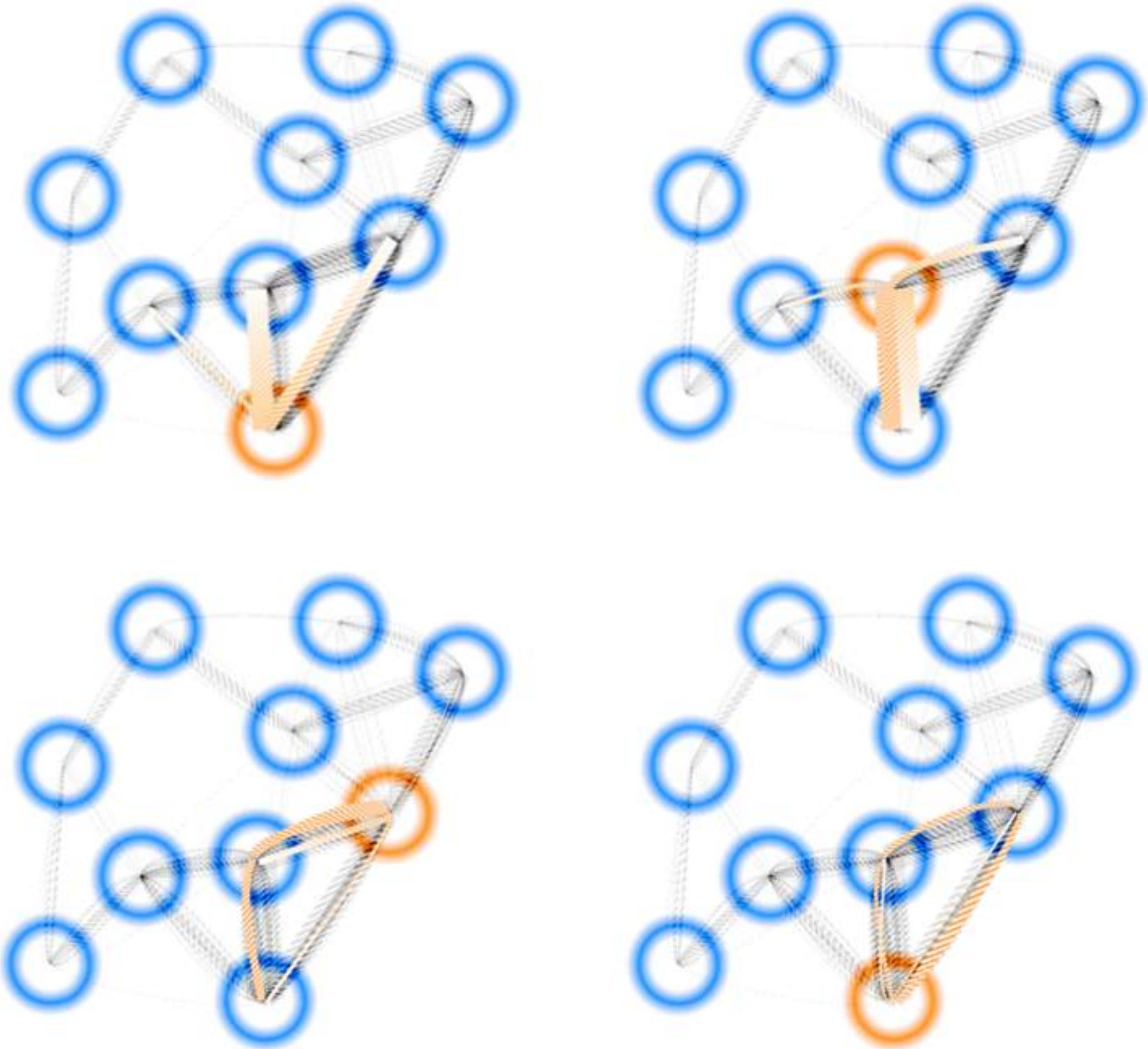
Width mapped to transition frequency

Animation shows direction of movement

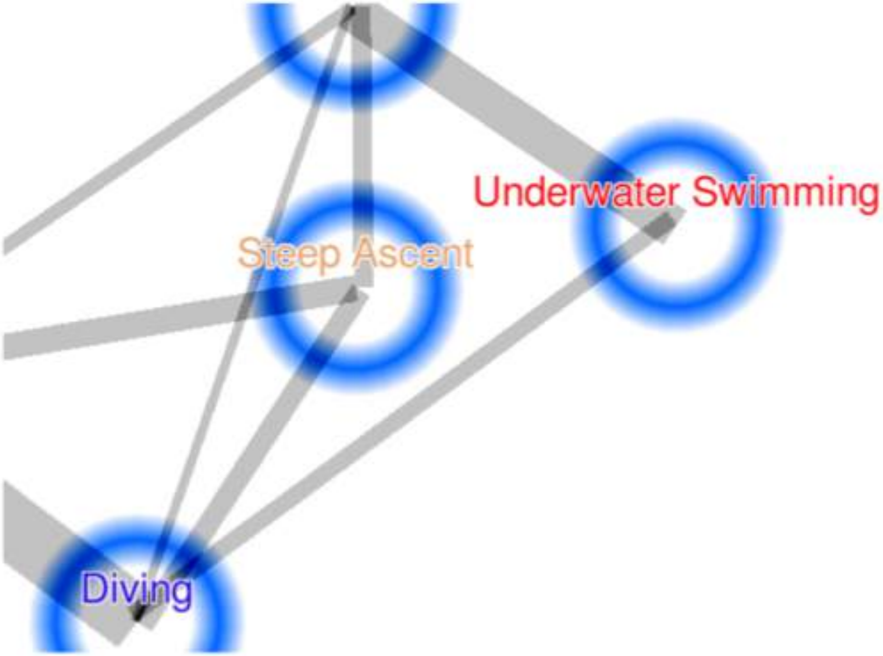


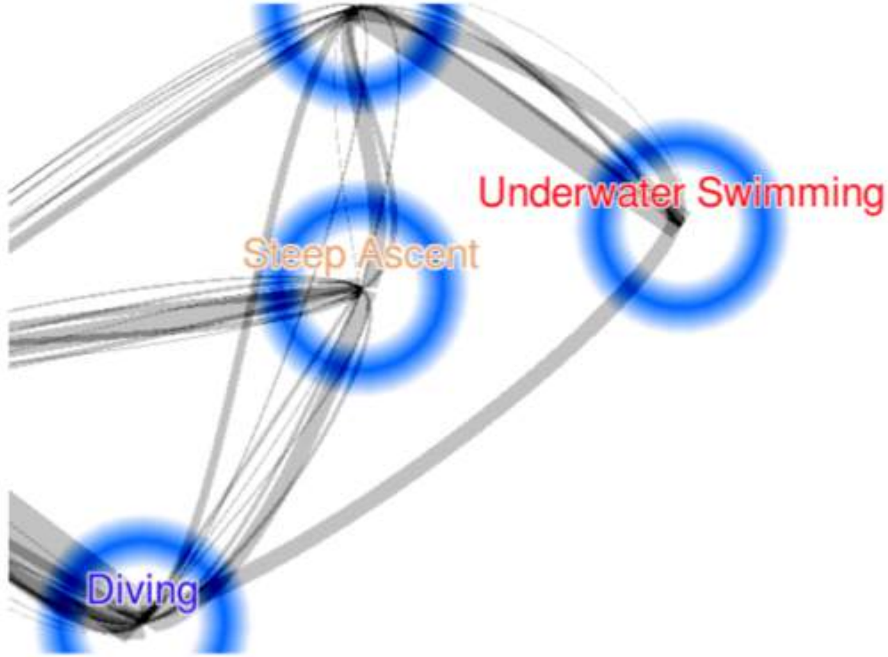
Interactive Selection of Graph Nodes

- Edges connecting selected nodes in order become part of selection, and are visually highlighted



Comparison of First and Third Order





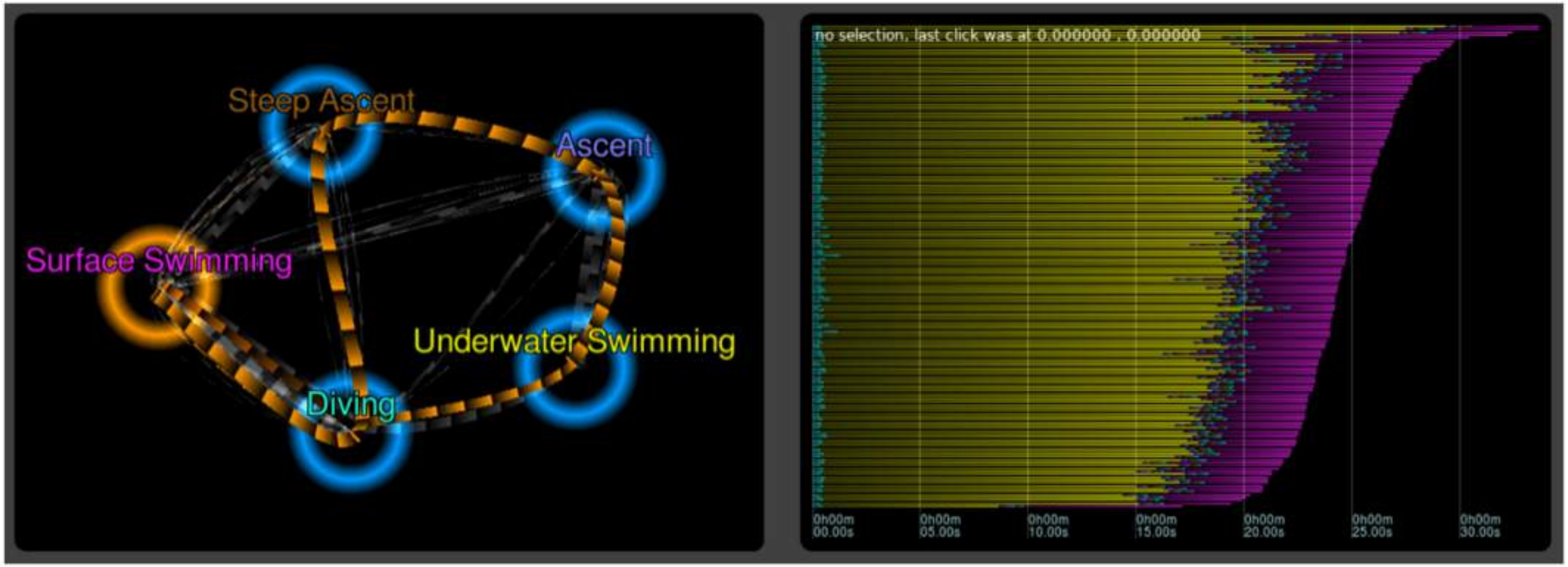
BioExplorer System

(1) Graph view, (2) selection view: x-axis-->time, y-axis-->transition, (3) scatter plot / heat map: x-axis-->y acceleration, y-axis-->x acceleration, (4) time-series view

BioExplorer System

A typical transition cycle lasts about 15-20 seconds

Demo video on YouTube: https://youtu.be/XRtGaiYZyTA

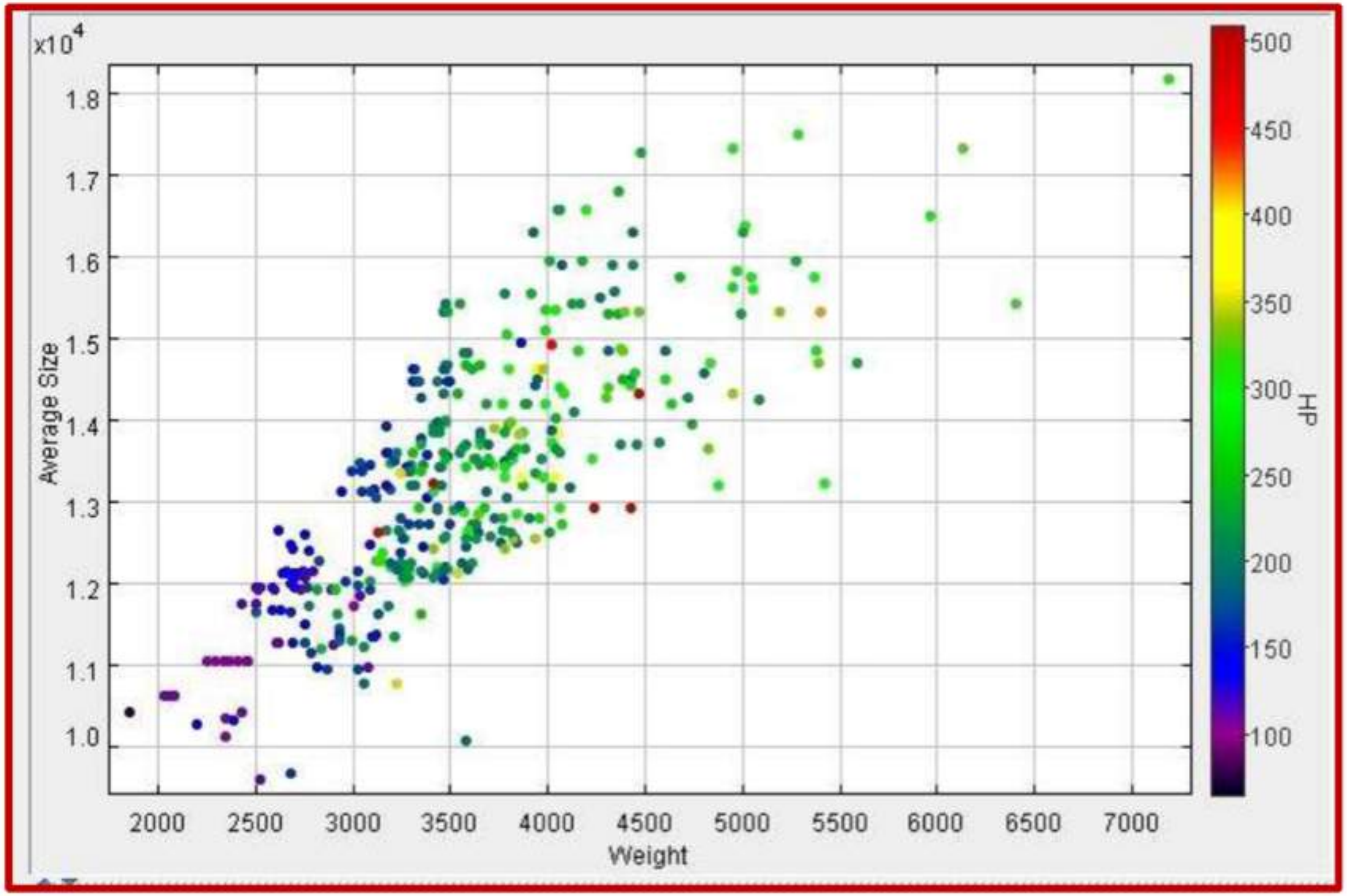


n-dimensional Data

Also called
multi-variate data
high-dimensional data

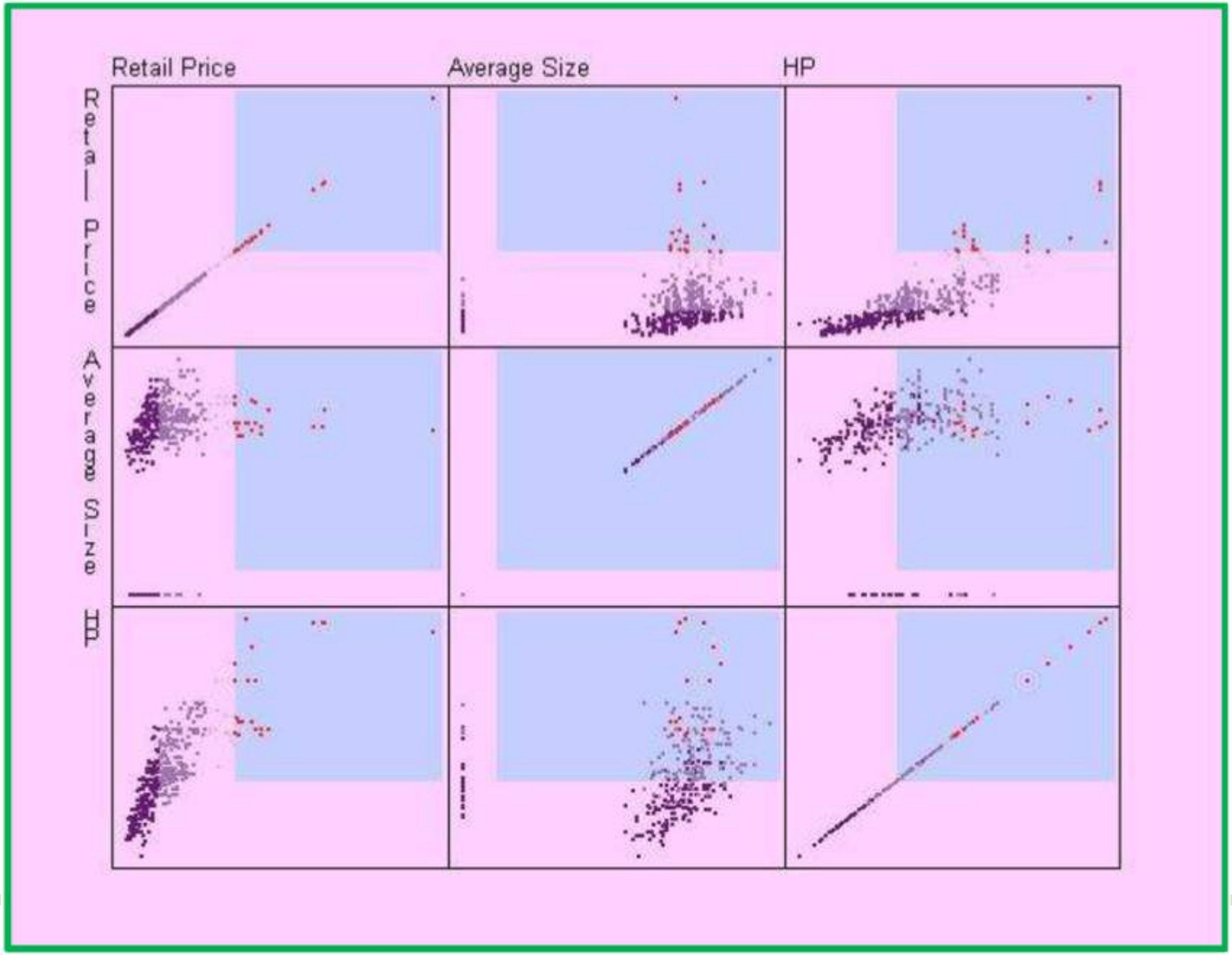
Scatterplot

2-3 variables, one ellipse per data item



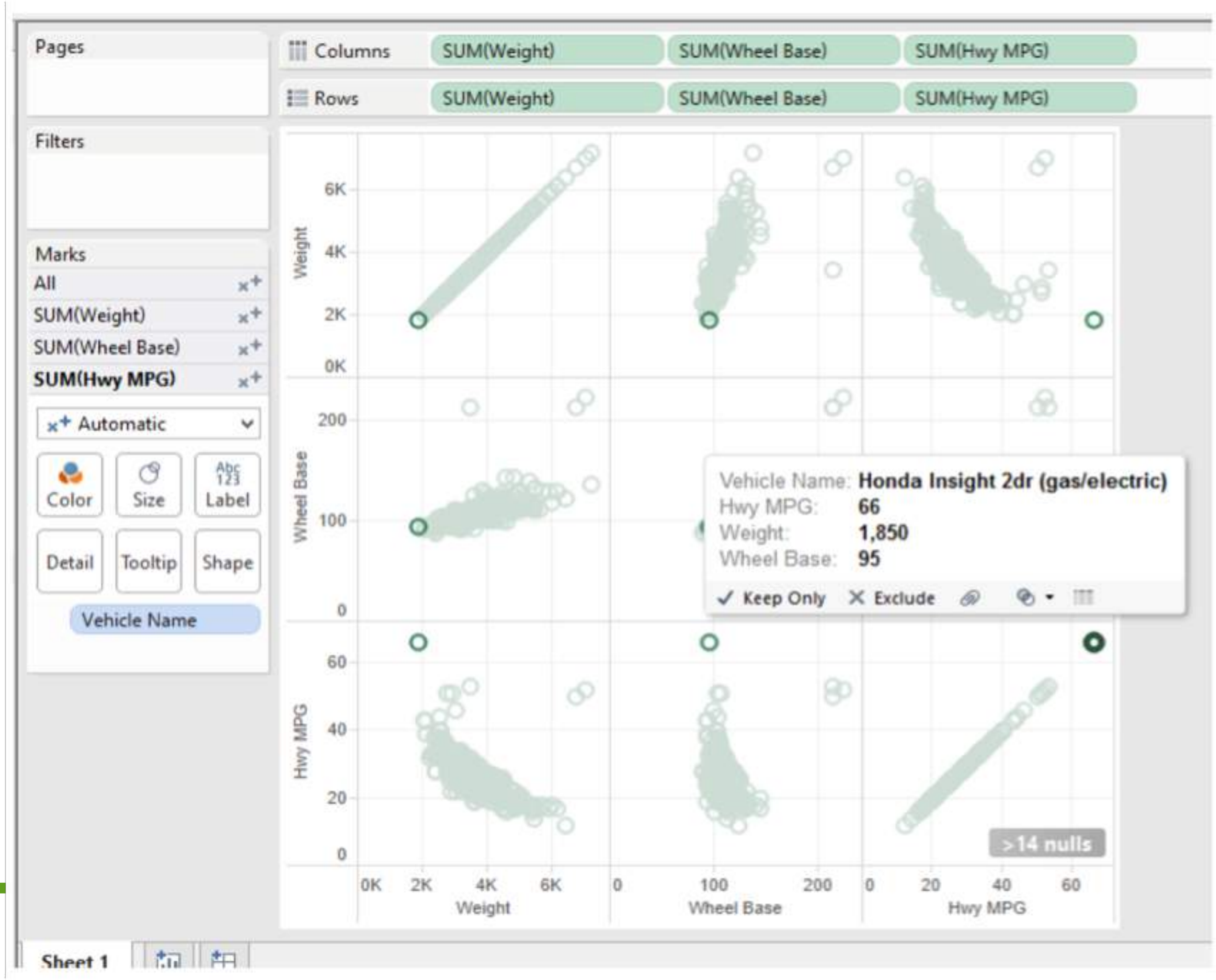
Scatterplot-Matrix

All variables versus all others (matrix)

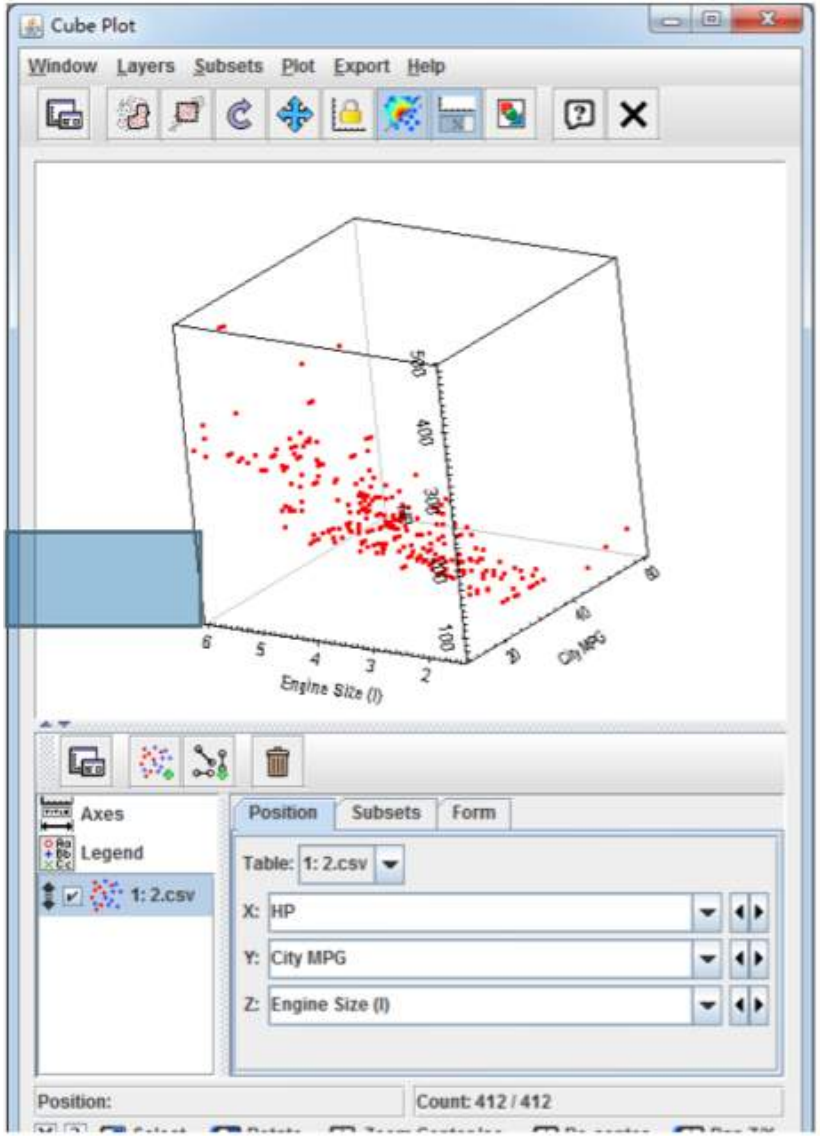


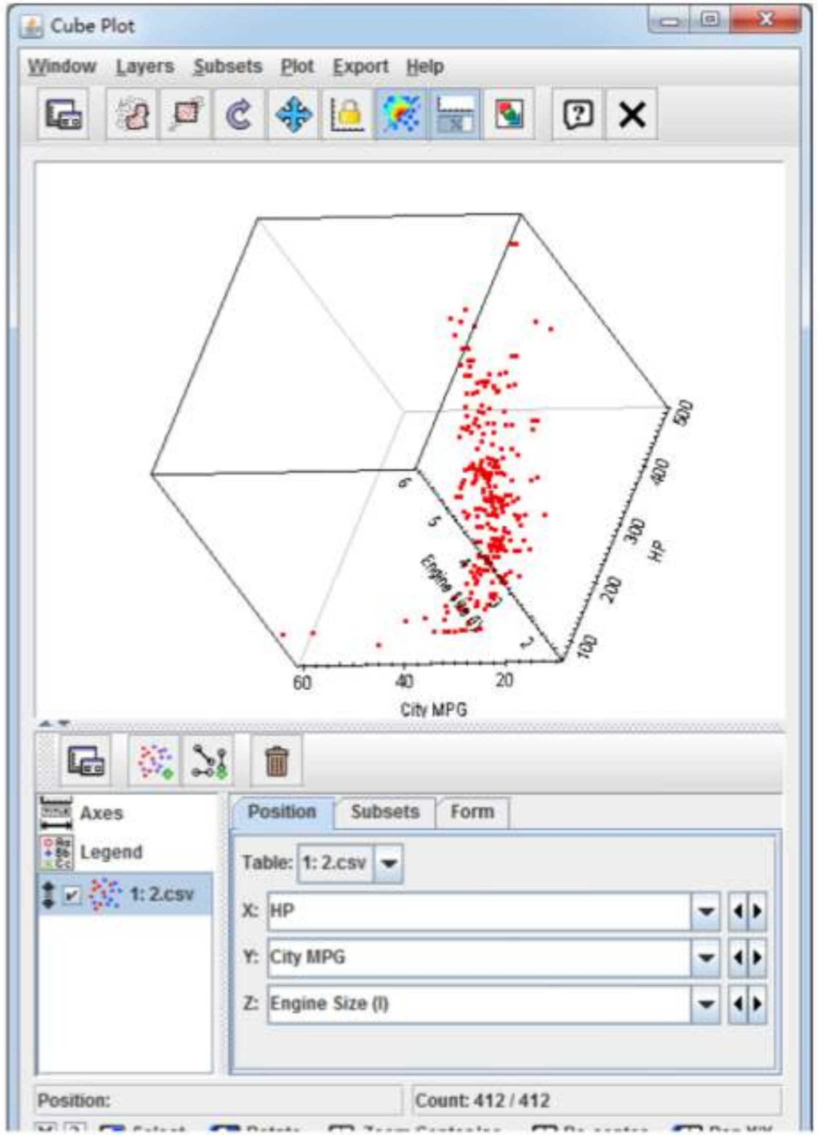
Scatterplot-Matrix

With details on demand

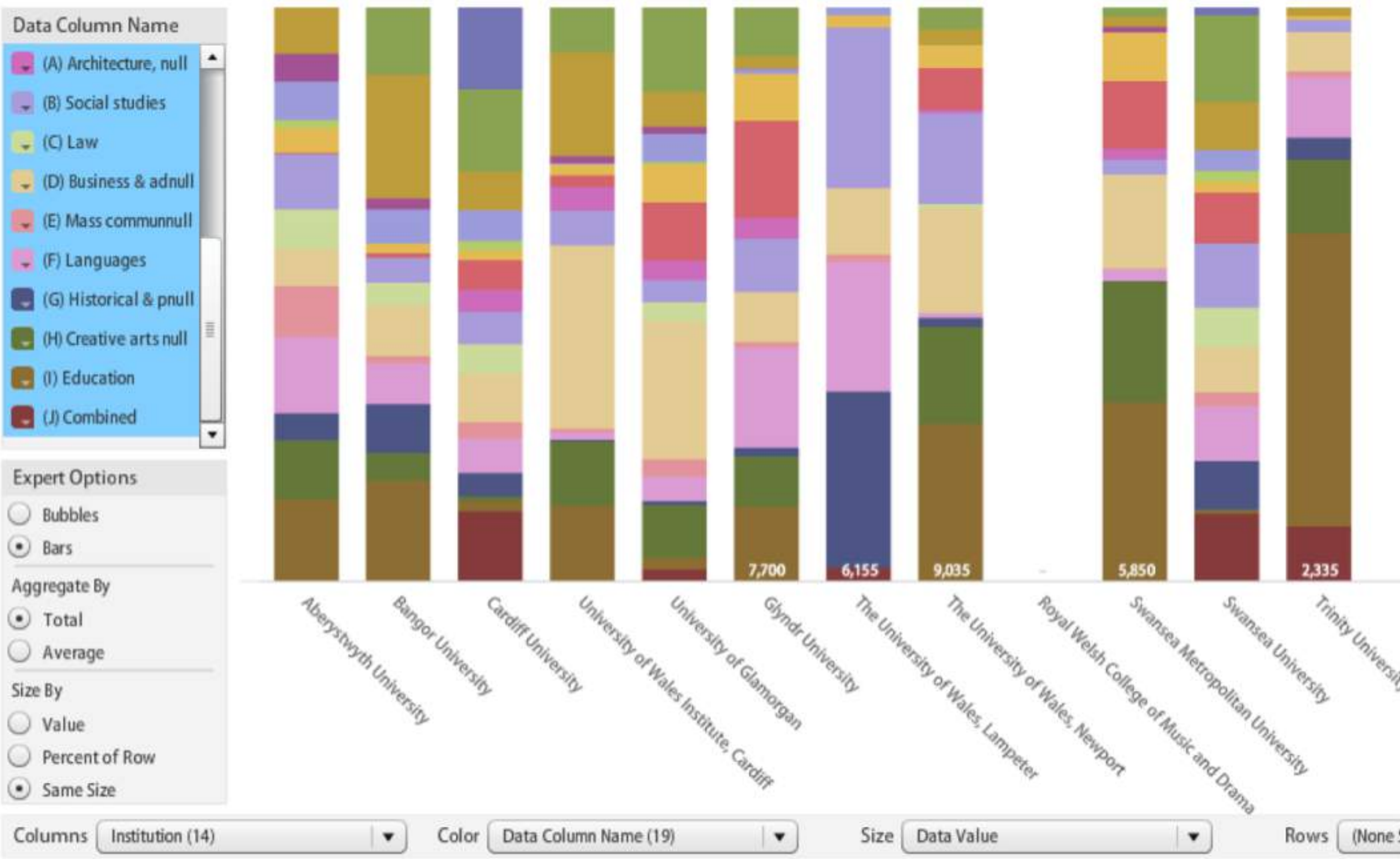


3D Scatterplot Example

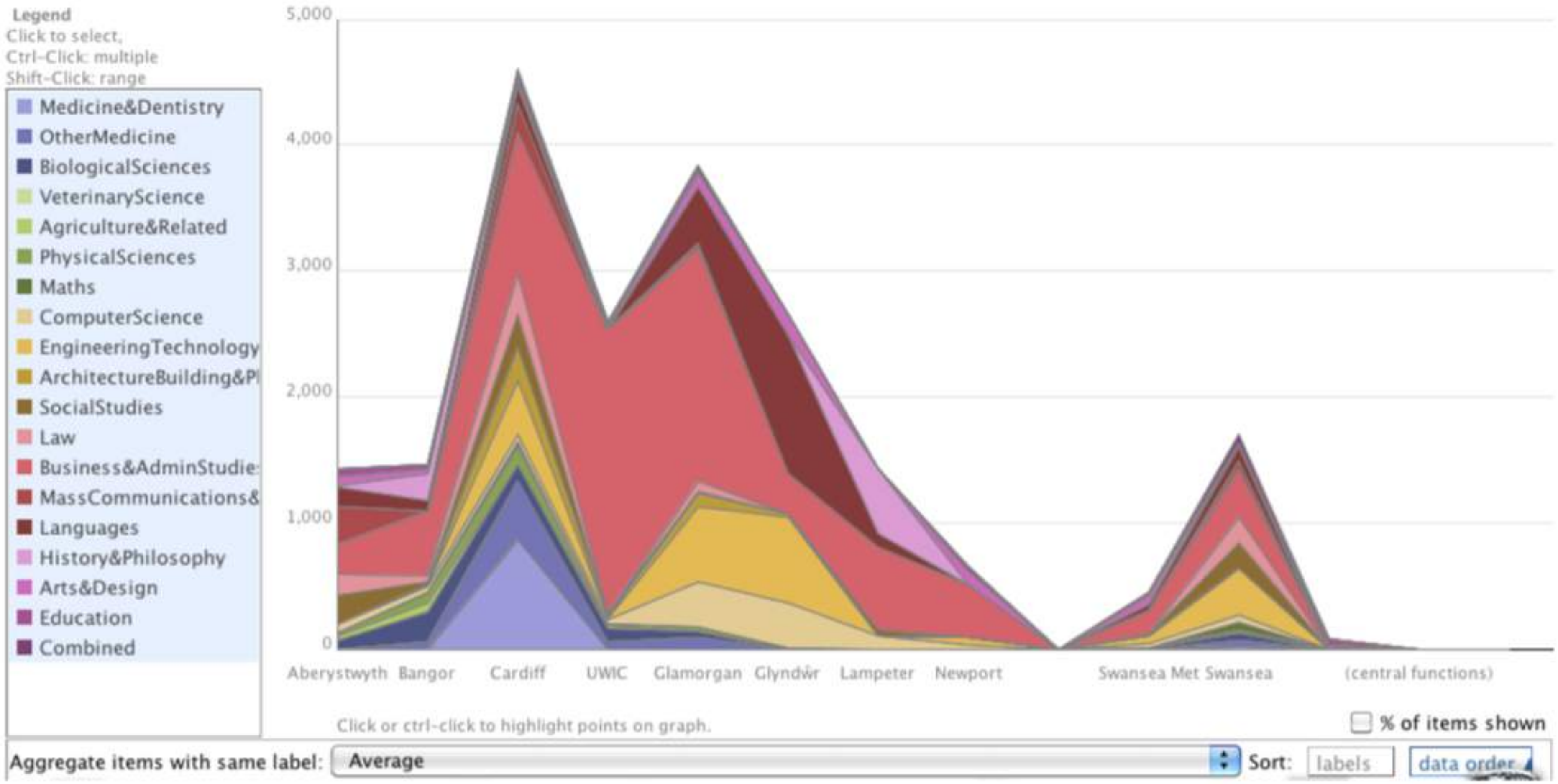




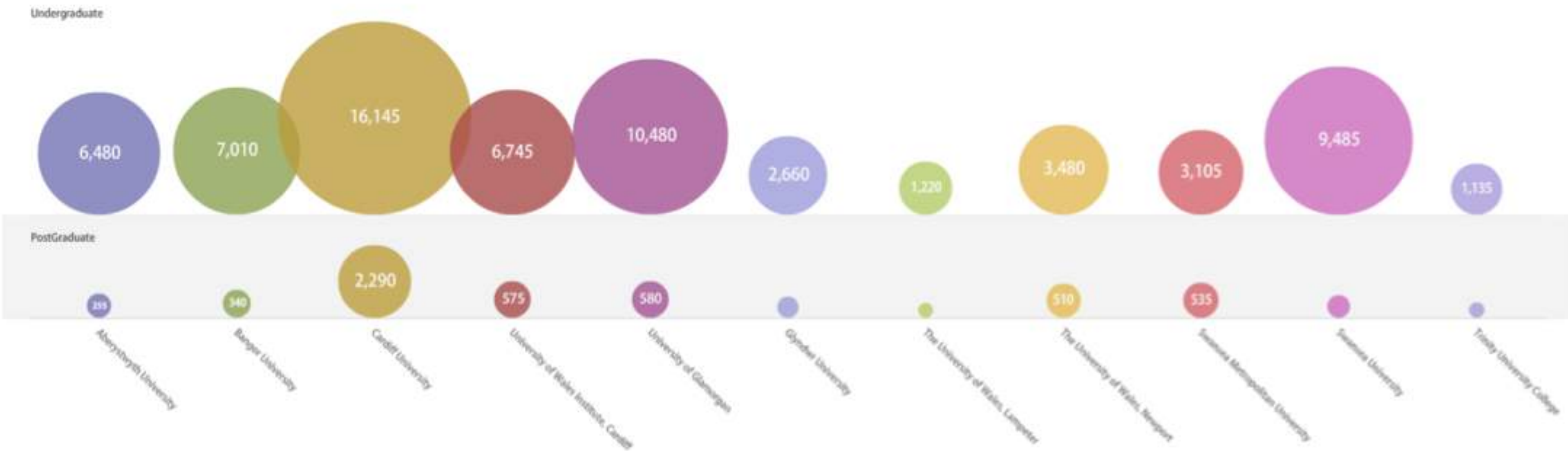
Stacked Bar Chart



Stacked Graph



Matrix Chart



Glyphs – Overview

Methods:

- Chernoff-Faces
- Star Glyphs
- Vector Glyphs
- Rugby Glyphs
- Sport Video Analysis

Chernoff-Faces

Data can be mapped to:

Head shape/curvature + size

Mouth length + curvature

Nose length + curvature

Eyes shape + size

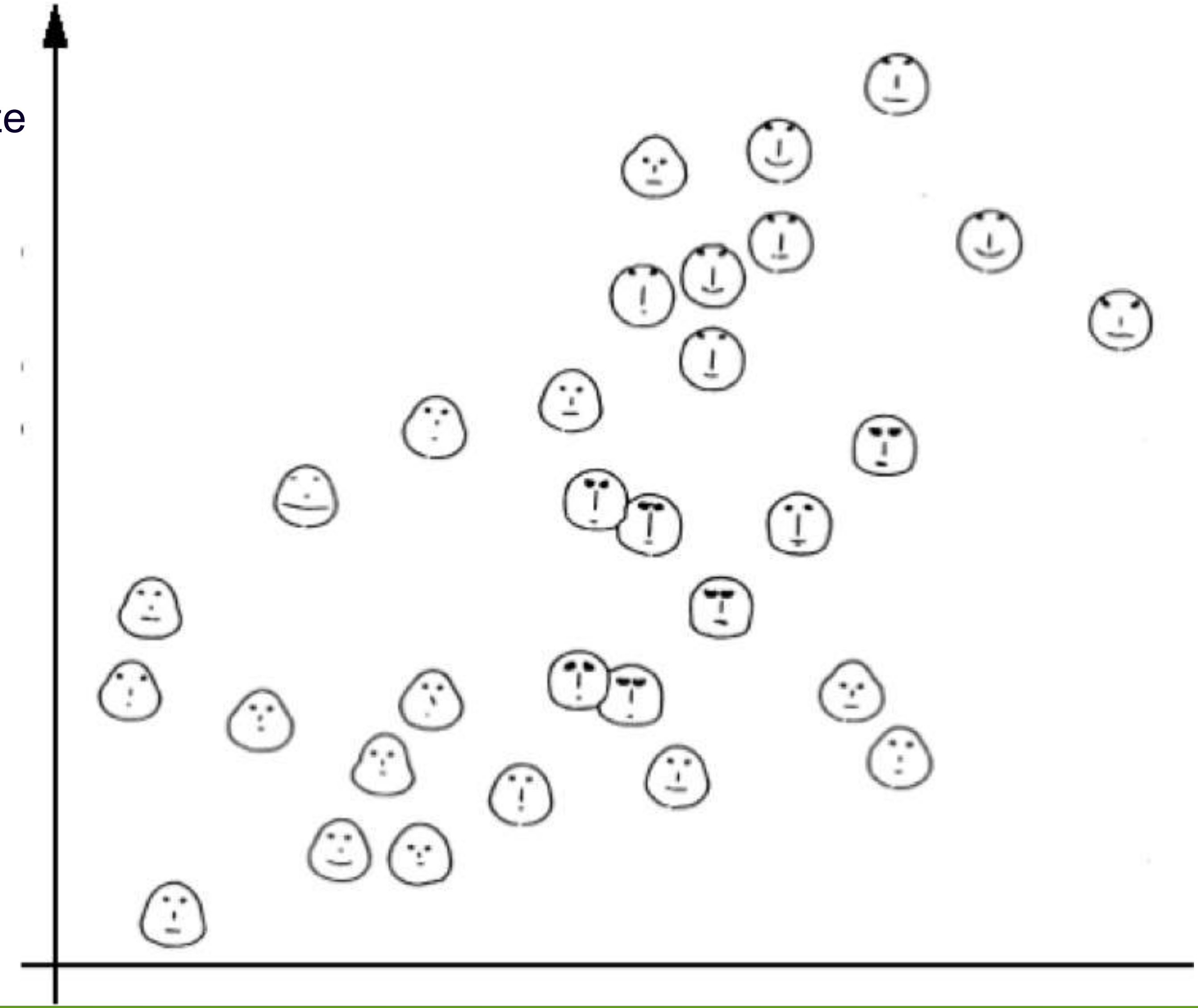
color

x,y position

Orginally used for 8-variate

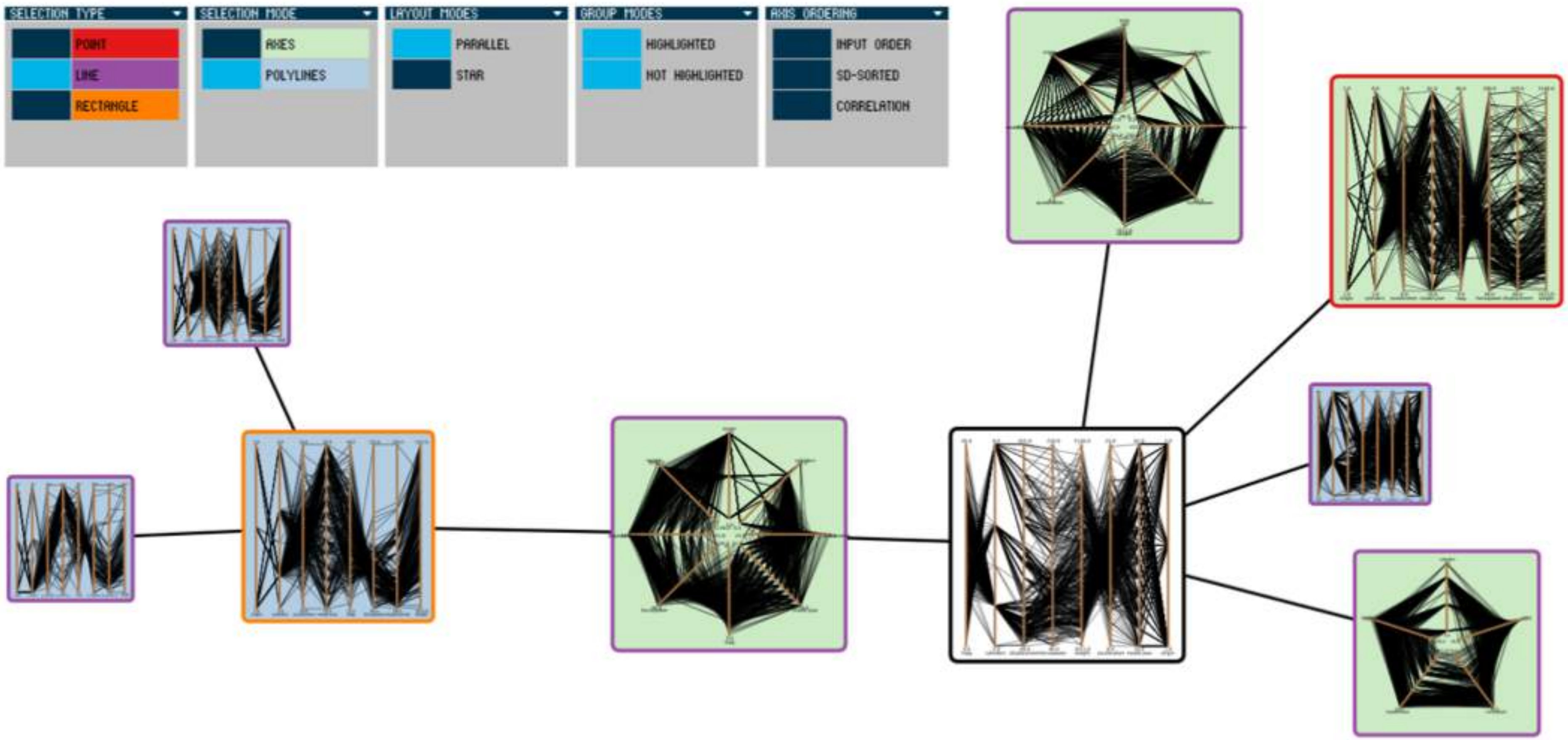
fossil specimens

(Chernoff, 1973)



Star Glyphs

EdiVis (Roberts et al., 2014)



Vector Glyphs

Vector glyphs for

visualization of fluid flow

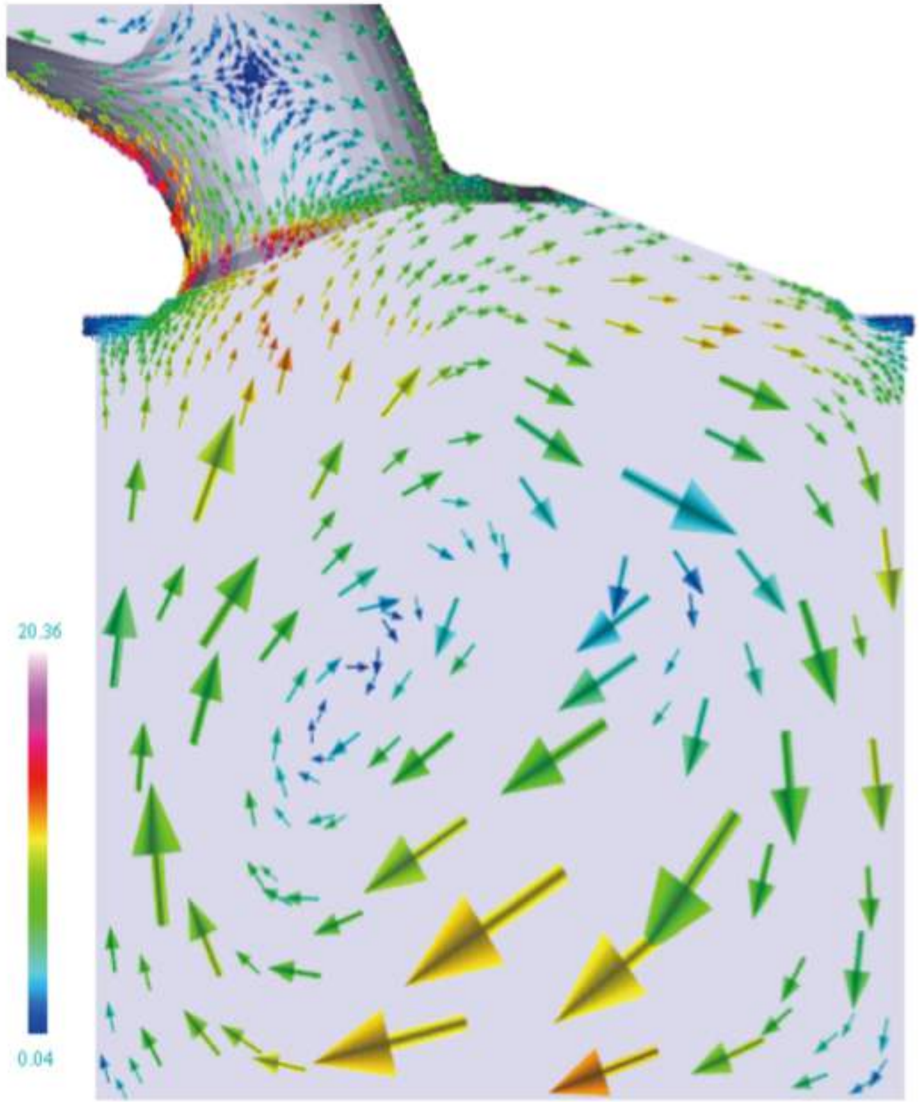
7 variates visualized:

Position (x,y,z)

velocity(x,y,z)

magnitude

(Peng et al., 2012)



Range Glyphs

Theta-Range glyphs for

visualization of fluid flow

8 variates visualized:

Position (x,y,z)

Velocity (x,y,z)

Magnitude,

Angle range

(Peng et al., 2012)

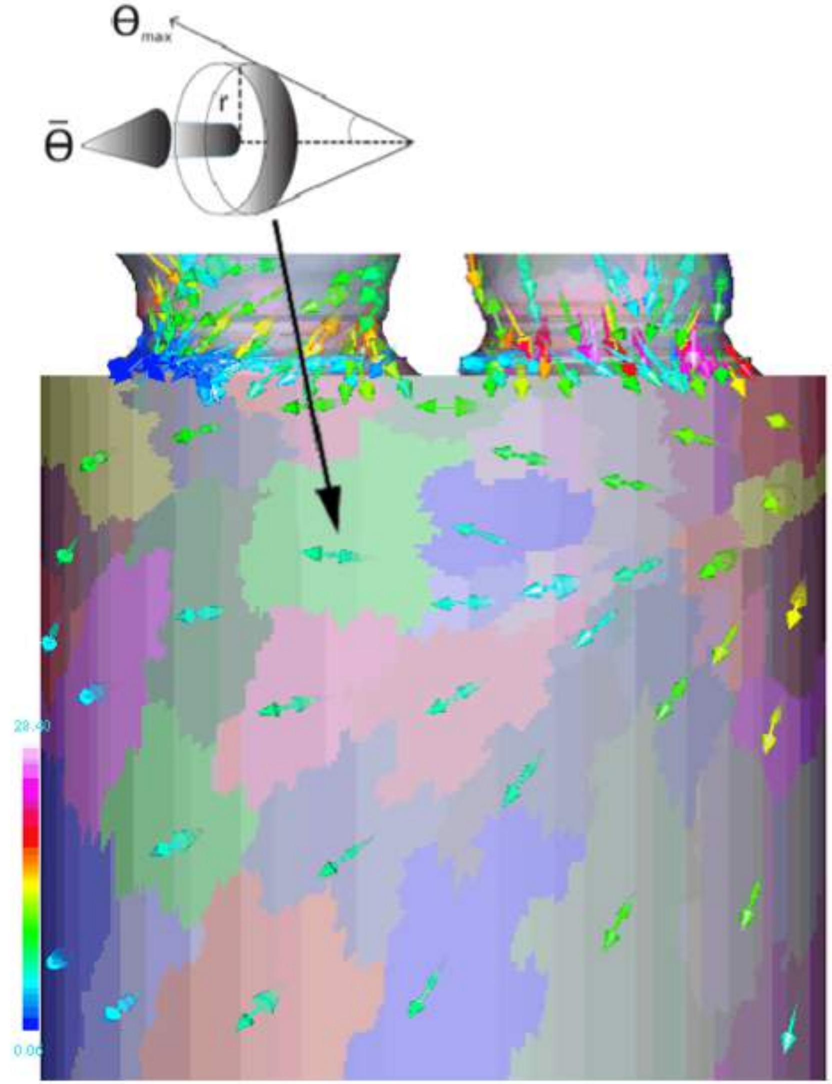


Fig. 10. (Top) the θ-range glyph whose radius represents the maximum range of vector field direction is illustrated. The result image with θ-range glyphs is shown in (Bottom). The result of the clustering is driven by an error measure ($\varepsilon = 18\%$). Glyph color is mapped to the velocity magnitude.

Range Glyphs

Vector glyphs for visualization of fluid flow

6 variates visualized:

Position (x,y,z)

Velocity range (min,max)

magnitude (Peng et al., 2012)

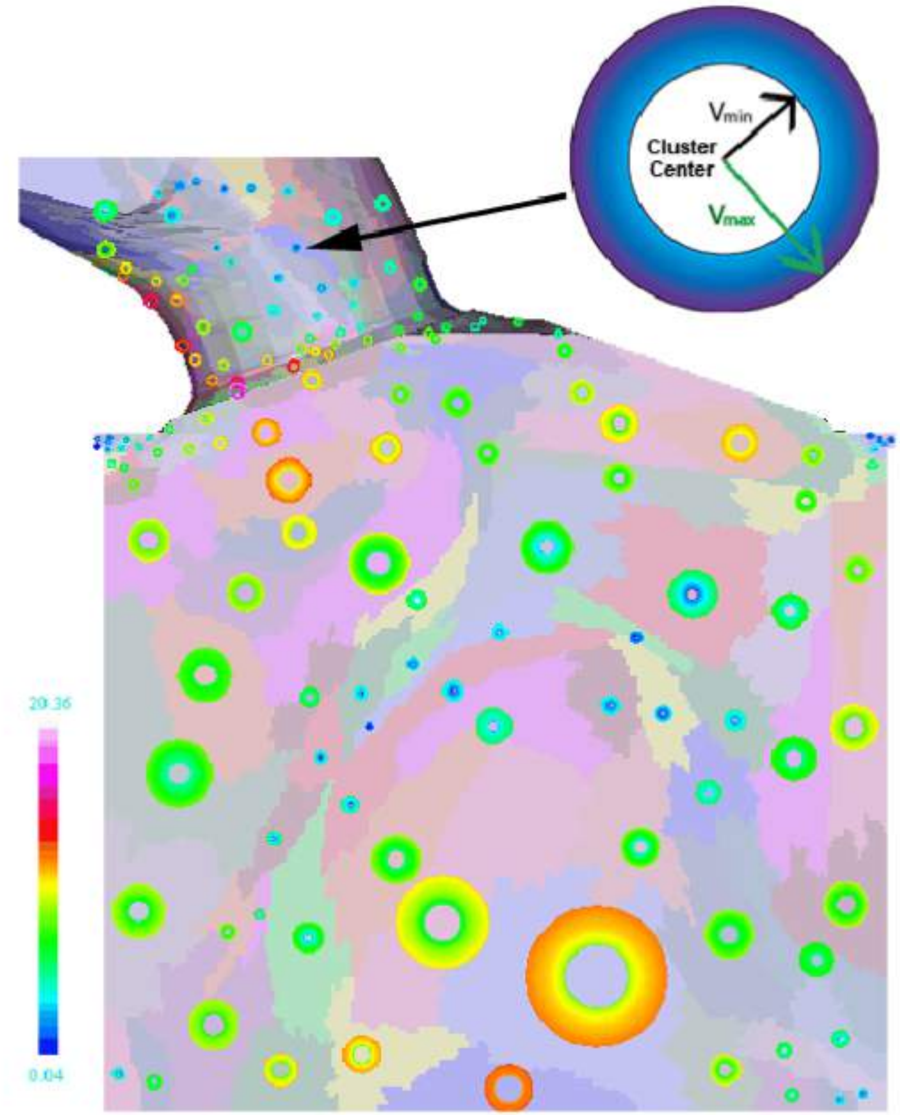


Fig. 9. |**v**|-range glyphs: (Top) a close-up look at a |**v**|-range glyph whose inner radius represents the minimum velocity magnitude while outer is mapped to the maximum. (Bottom) the result image with |**v**|-range glyphs applied to depict the variation in magnitude within each cluster .The result of the clustering driven by an error measure ($\varepsilon = 15\%$). Glyph color is mapped to velocity magnitude.

Hybrid Glyphs

Vector glyphs for visualization of fluid flow

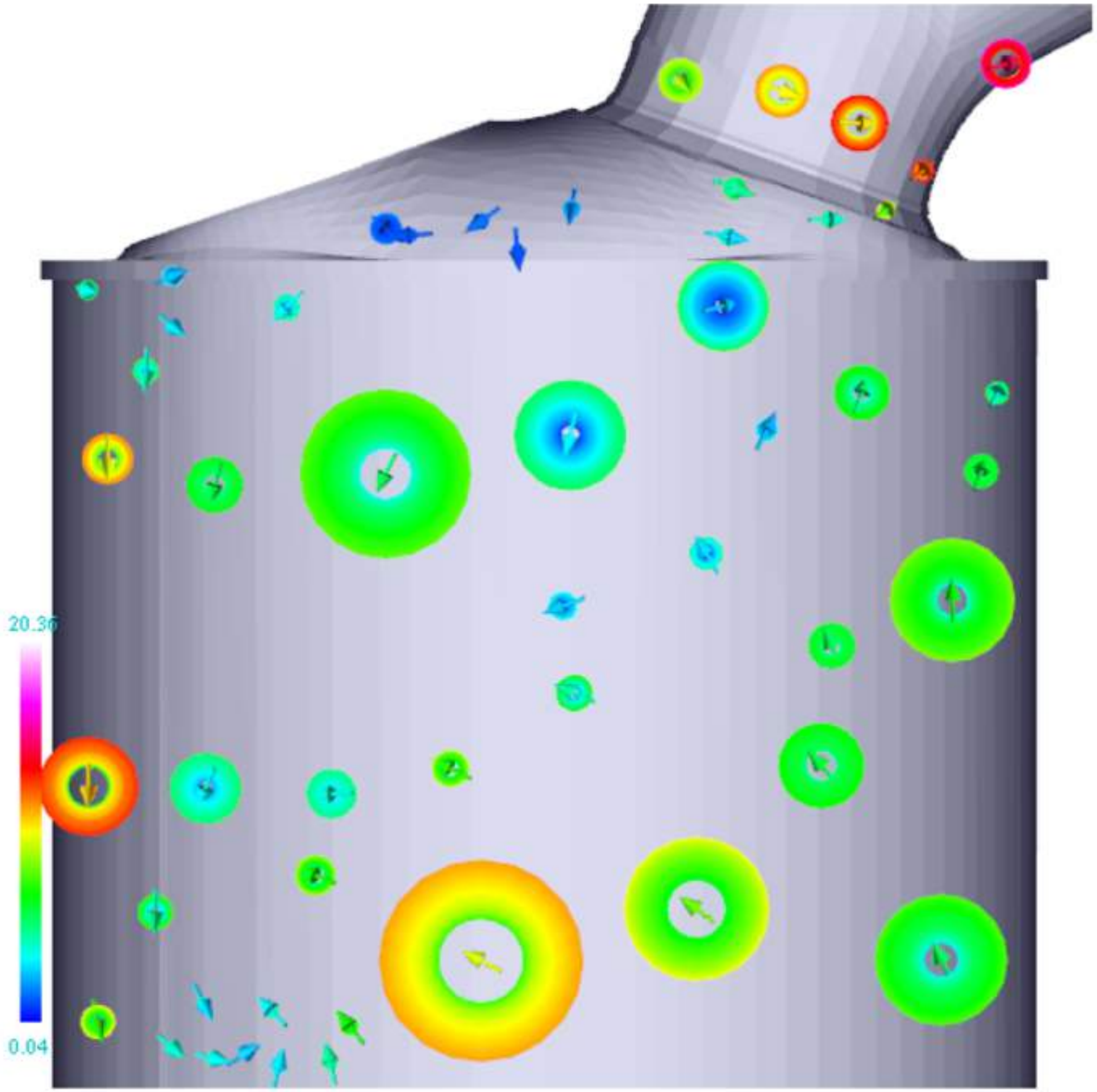
9 variates visualized:

Position (x,y,z)

Velocity (x,y,z)

Velocity range (min,max) magnitude

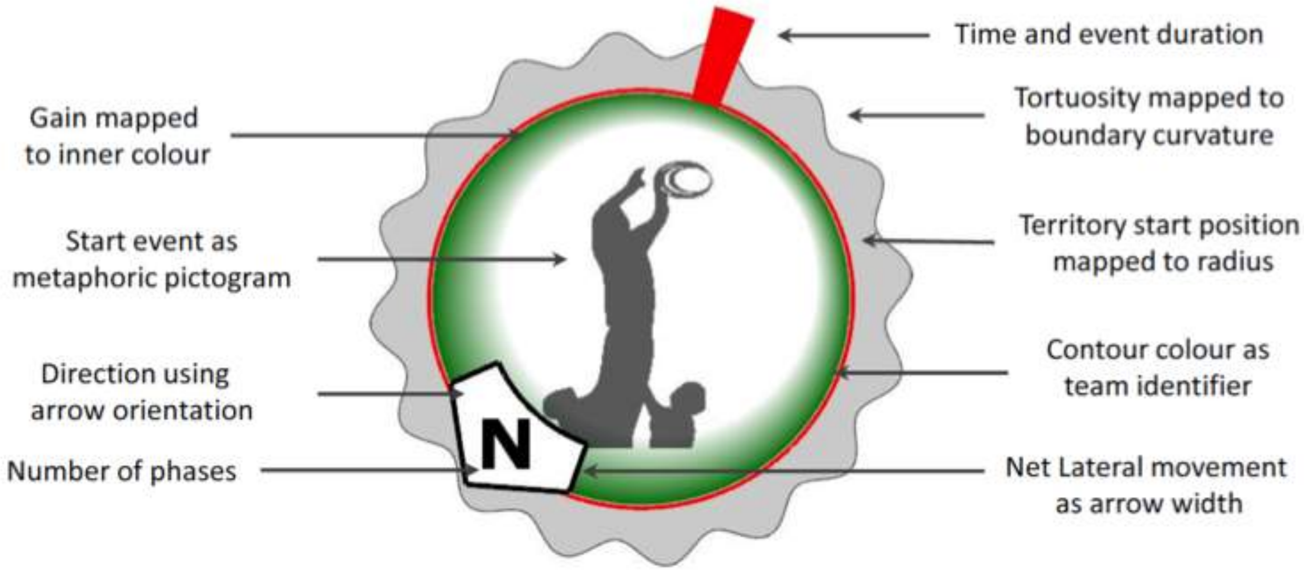
(Peng et al., 2012)



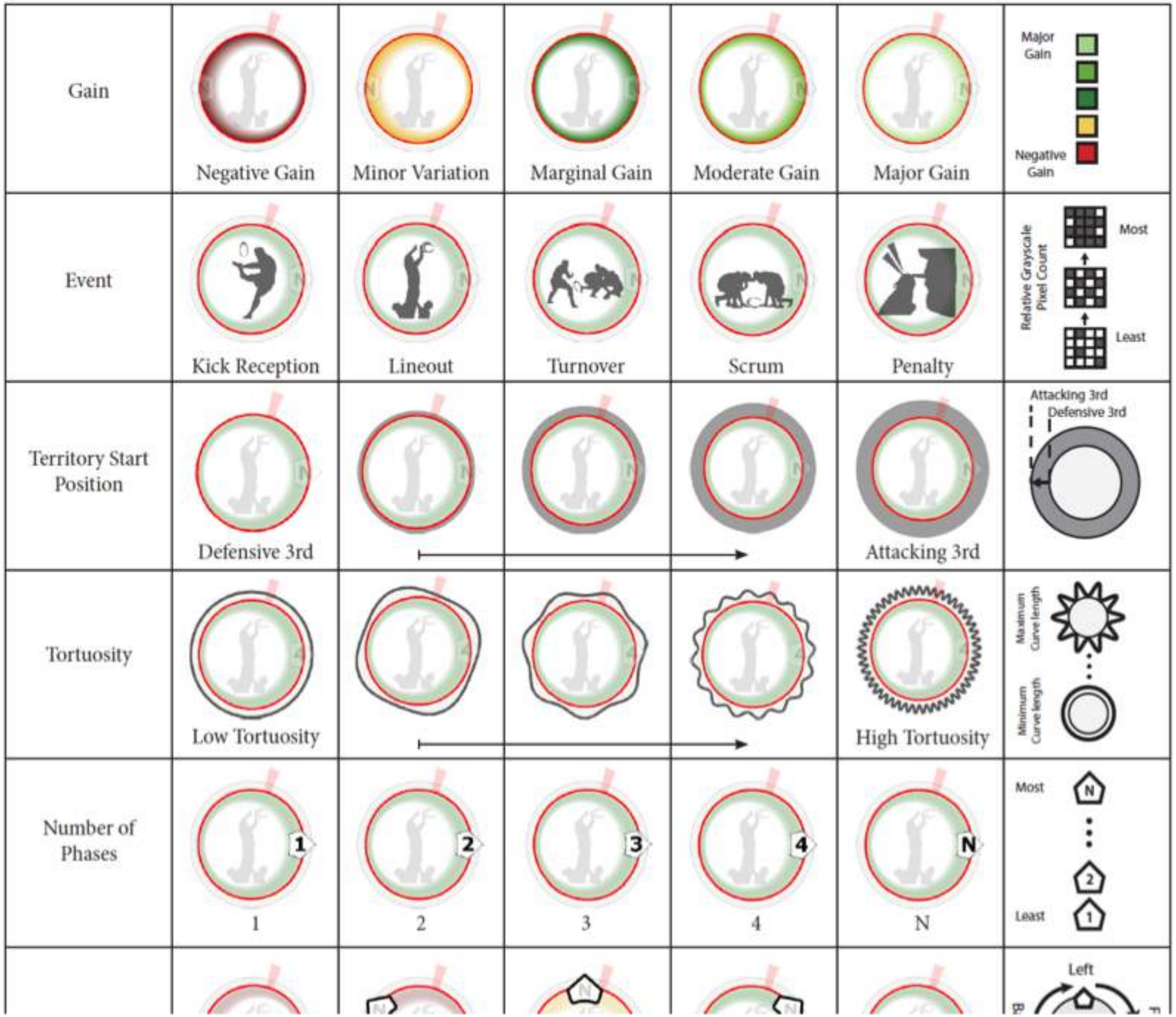
Sample High-Dimensional Glyph

10 dimensions

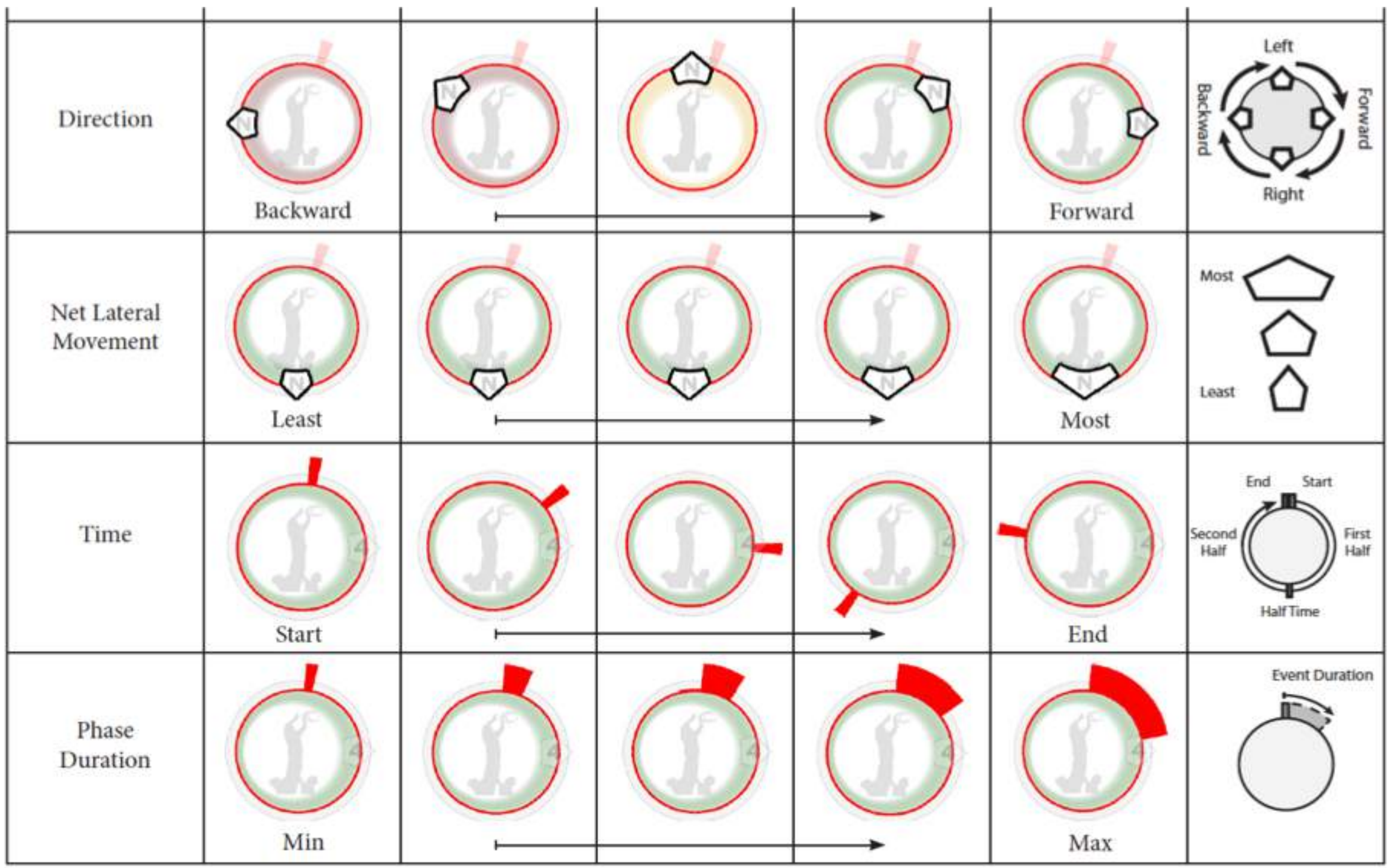
Sort Key	Typedness	Visual Channel
Gain	Ordinal	Colour
Event	Nominal	Pictogram
Territory Start Position	Interval	Size
Tortuosity	Ratio	Shape
Number of Phases	Ratio	Enumerate
Direction	Direction	Orientation
Net Lateral Movement	Ratio	Length
Time	Ratio	Location
Phase Duration	Ratio	Length
Team Identifier	Nominal	Colour



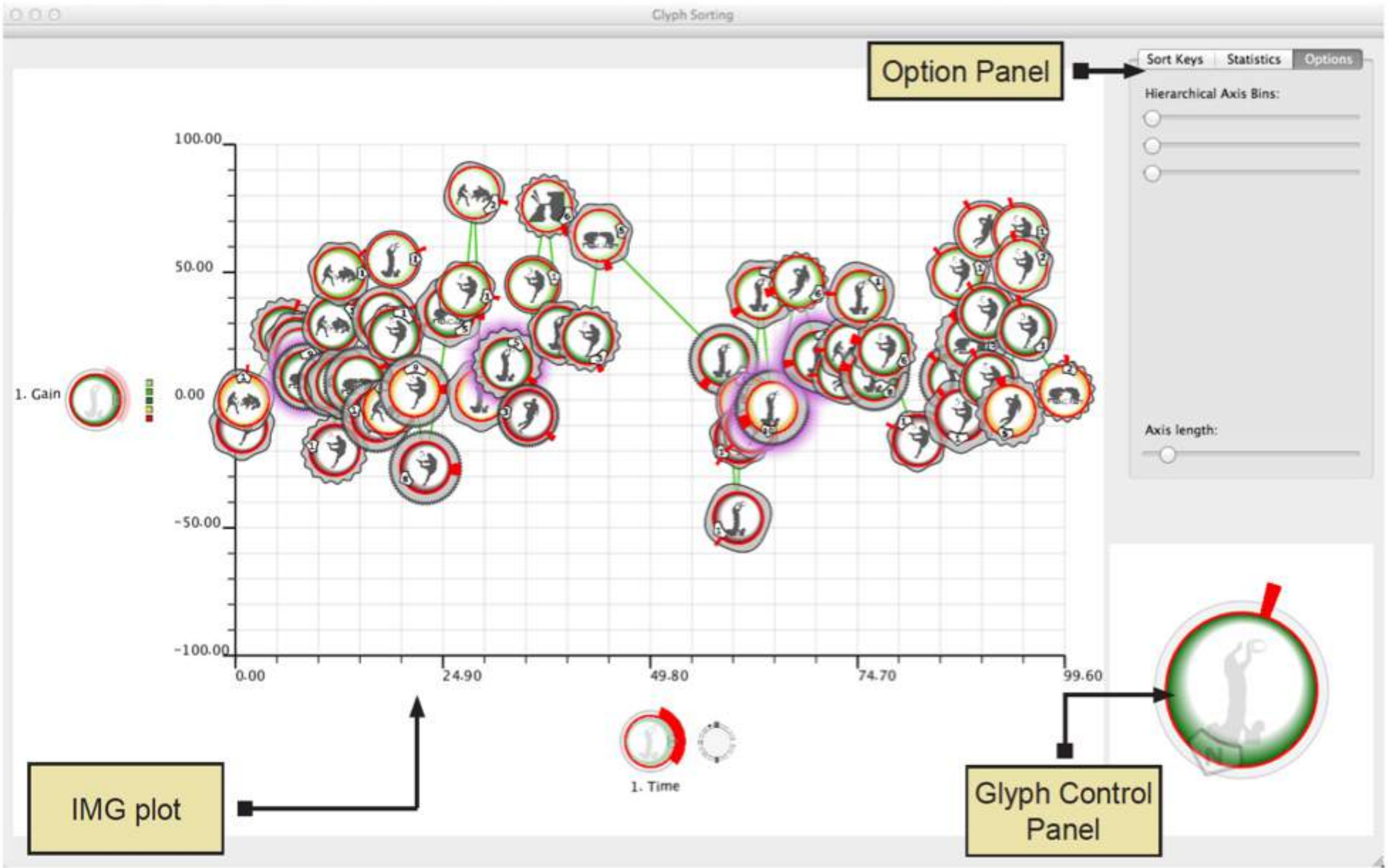
Glyphs for Visualizing Rugby Events



Glyphs for Visualizing Rugby Events



High-Dimensional Visualization System Interface



Comparing Two Matches and Demo

- 2 matches
- Gain (Y) vs Event (X) Plotted
- Events are:

Kick Reception,
Restart Reception,
Lineout,
Turnover,
Scrum,
Penalty

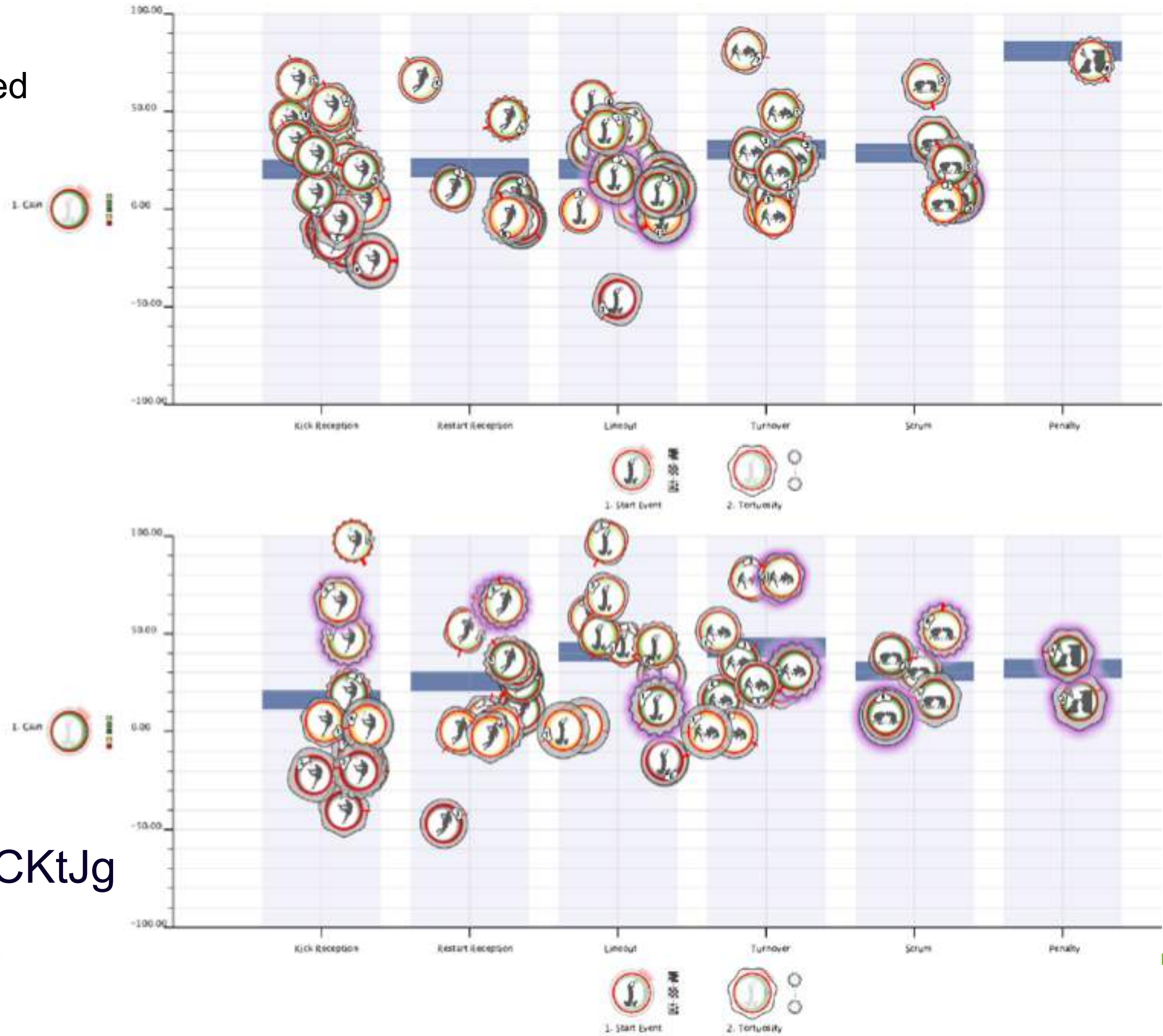
- Purple indicates

points scored

(Chung et al., 2015)

Full Tutorial video:

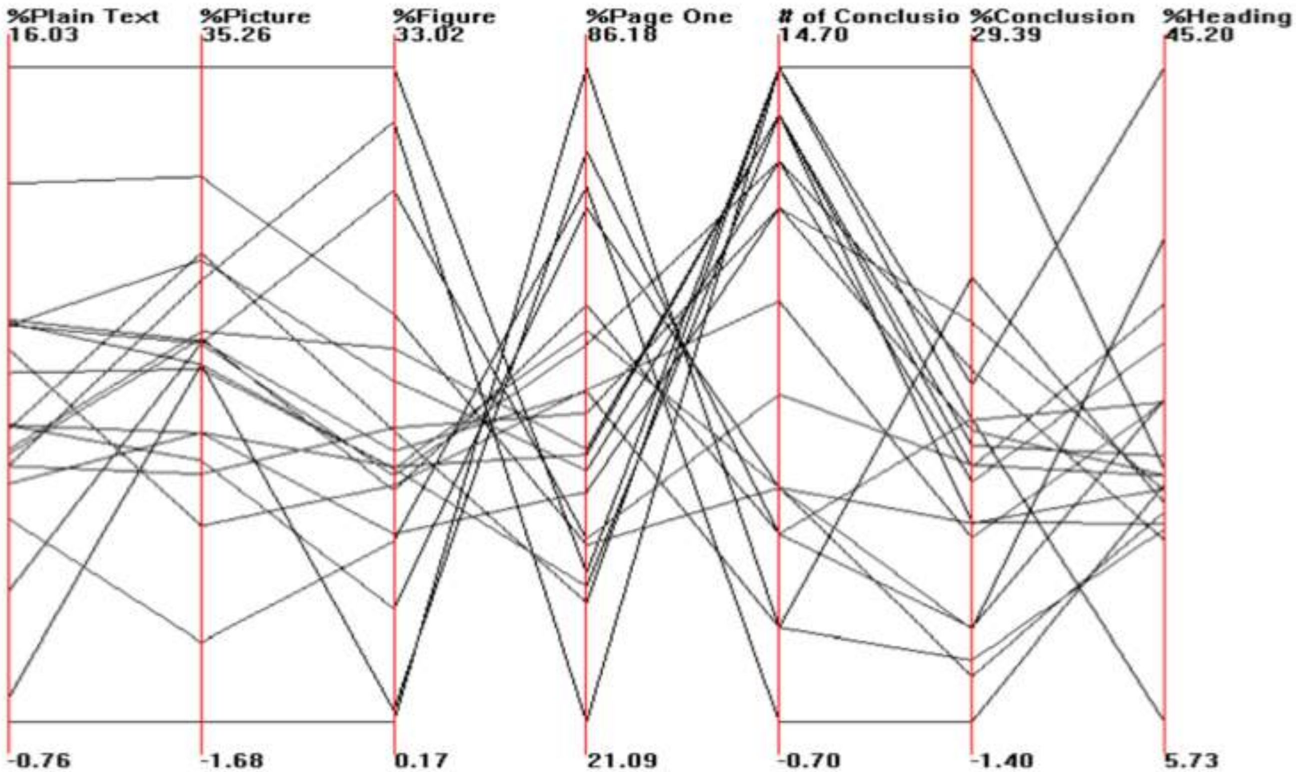
https://youtu.be/HNl67SCKtJg



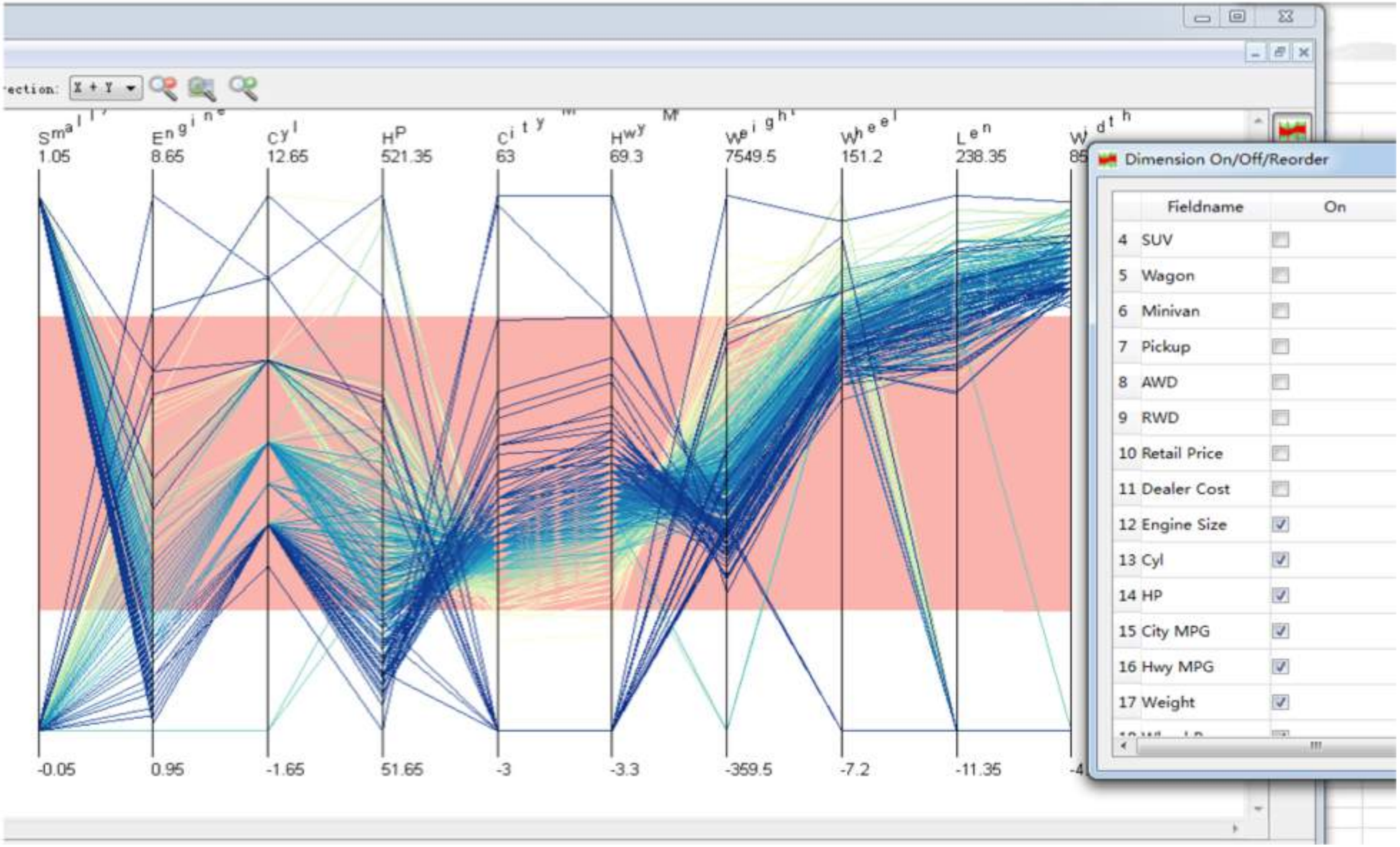
Introduction to Parallel Coordinates

Data: *n* dimensions (aka attributes)

- layout: *n* parallel axes
- one axis per data-attribute
- axes scaled to min/max-interval (Geng et al., 2011)

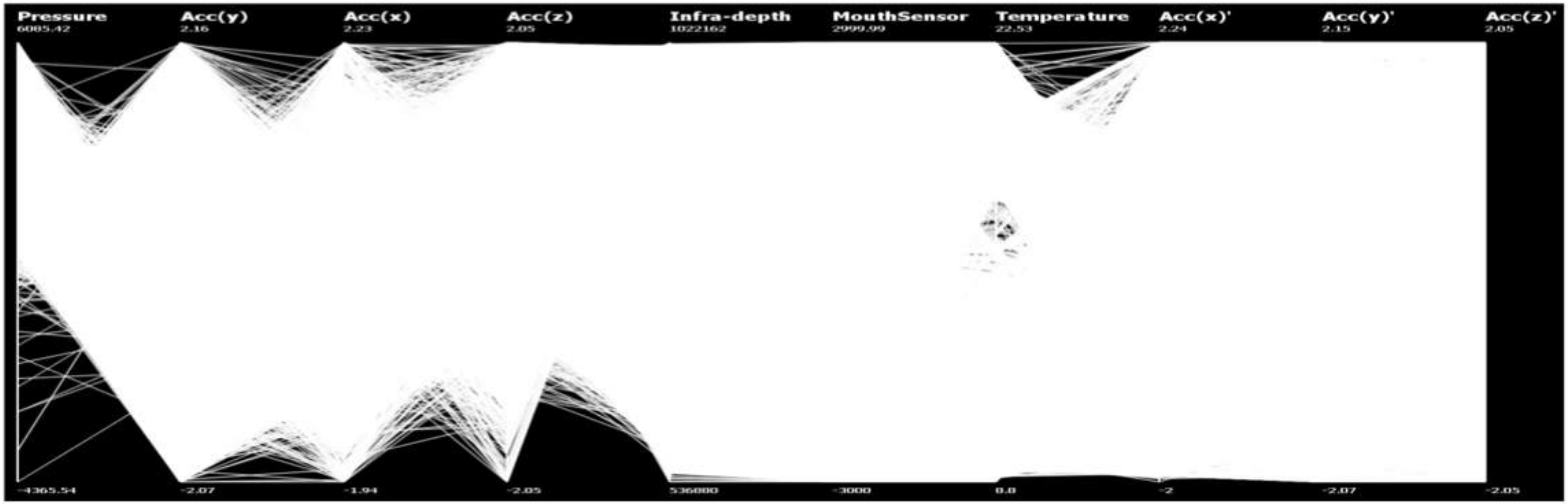


Visualization of Correlations (XMDV)



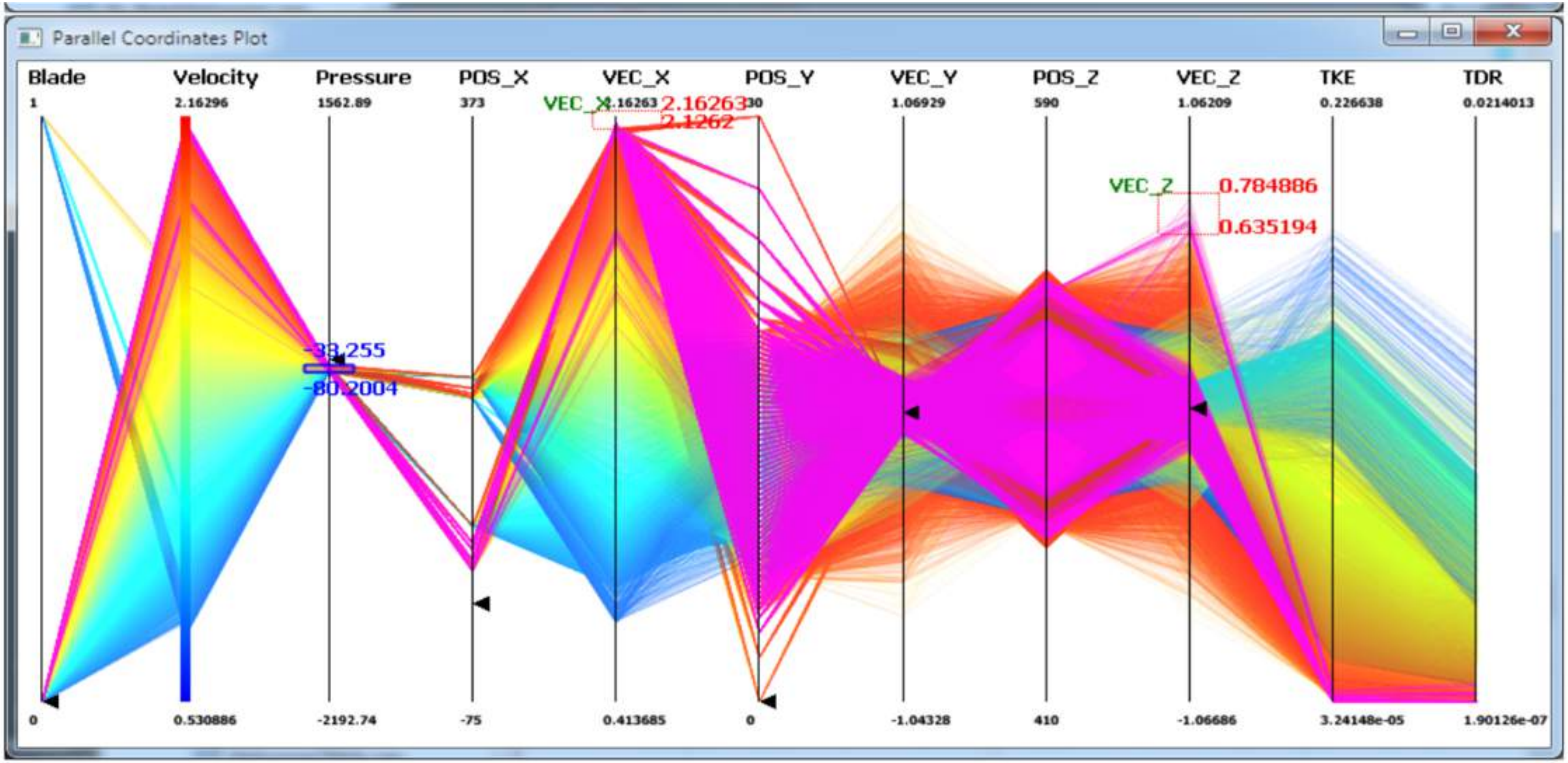
Parallel Coordinates: Challenge

- Overlapping Polylines (Geng et al 2011)

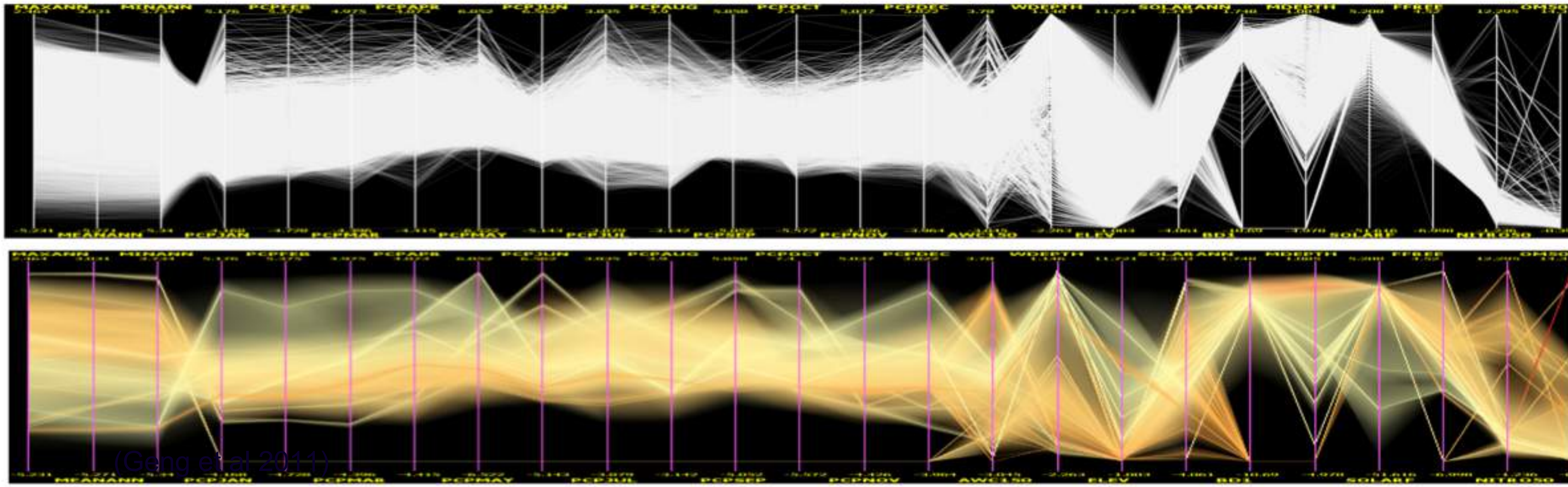


Parallel Coordinates: Color and Brushing

(Peng et al 2014)

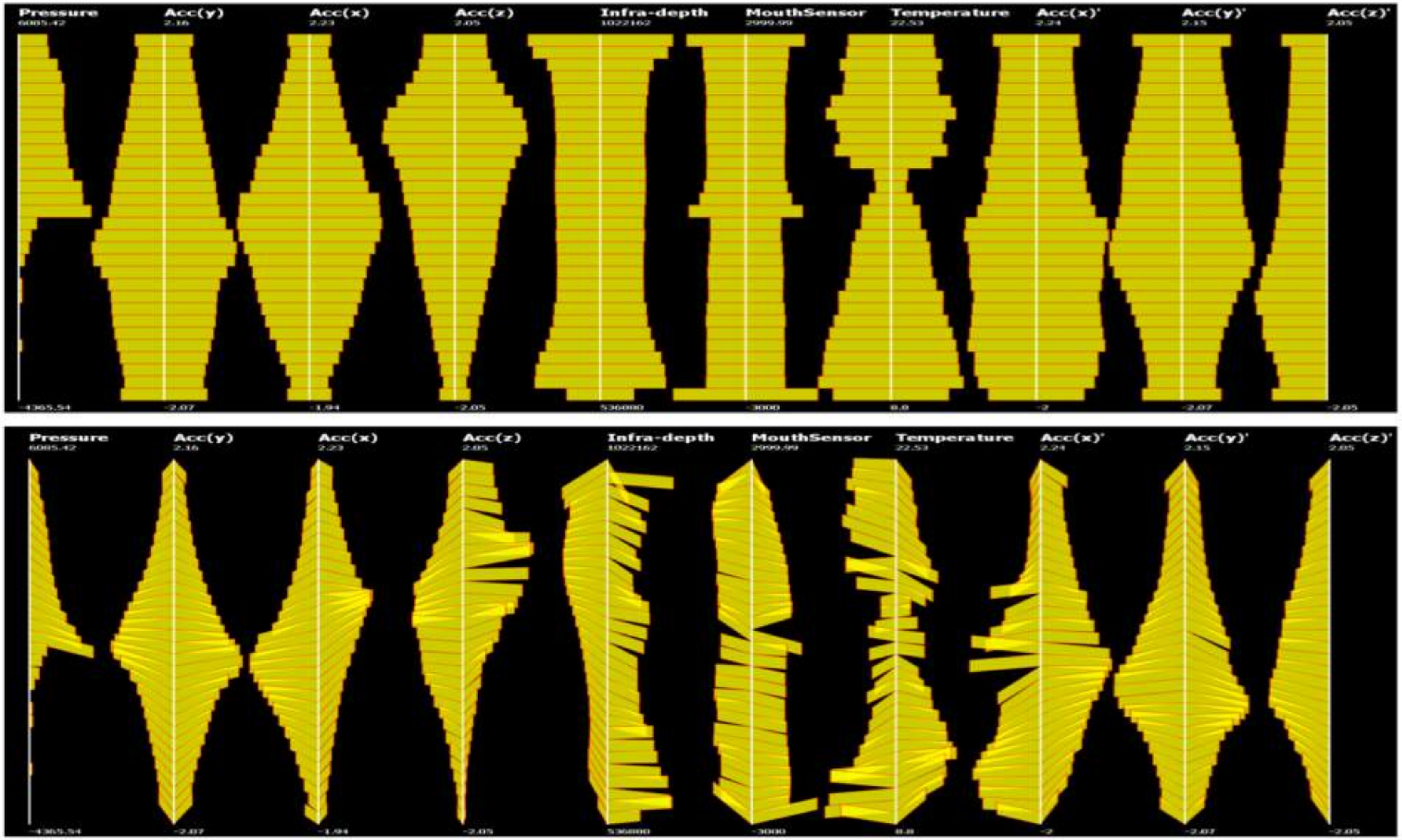


Hierarchical Parallel Coordinates



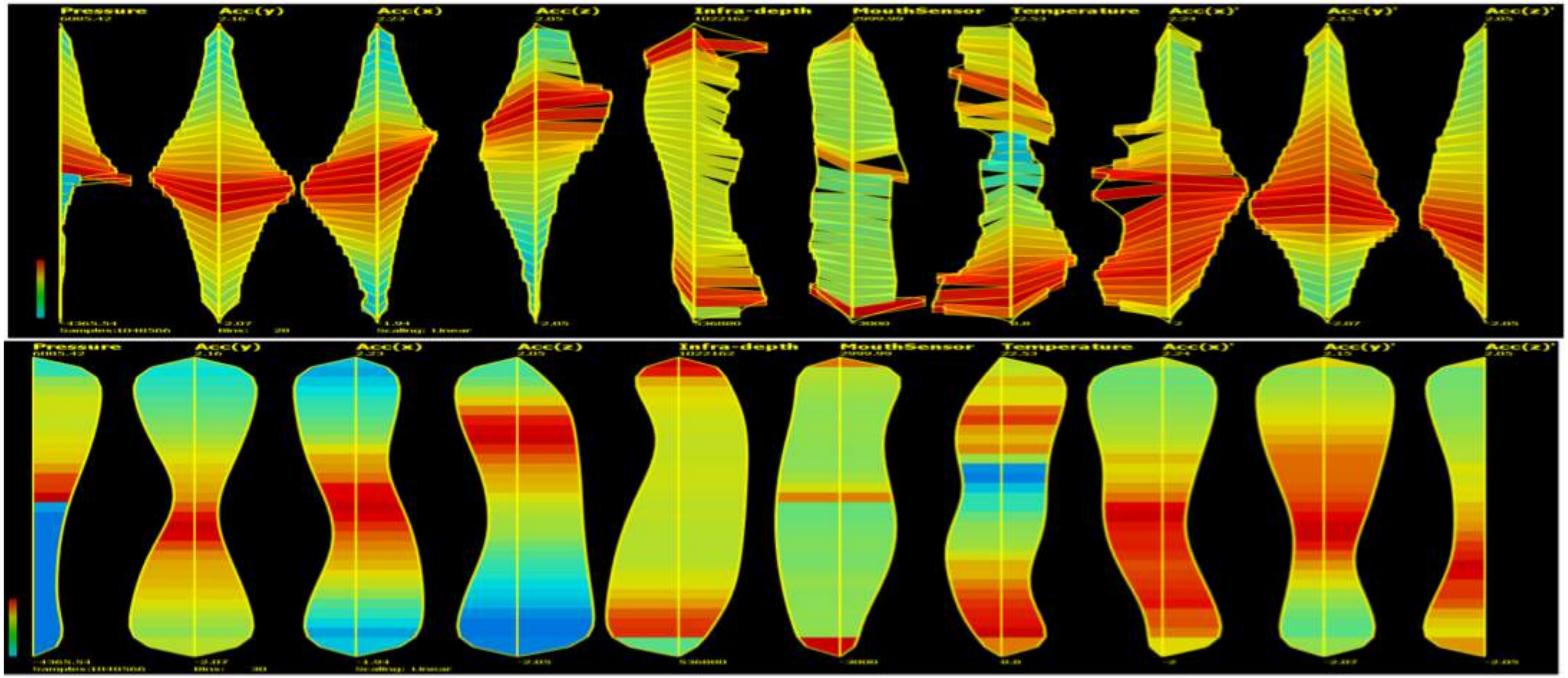
(Geng et al 2011)

Angular Histograms for Big Data



(Geng

Angular Histograms for Big Data



Demo Video: https://youtu.be/iDXdEyh1F5E (Geng et al 2011)

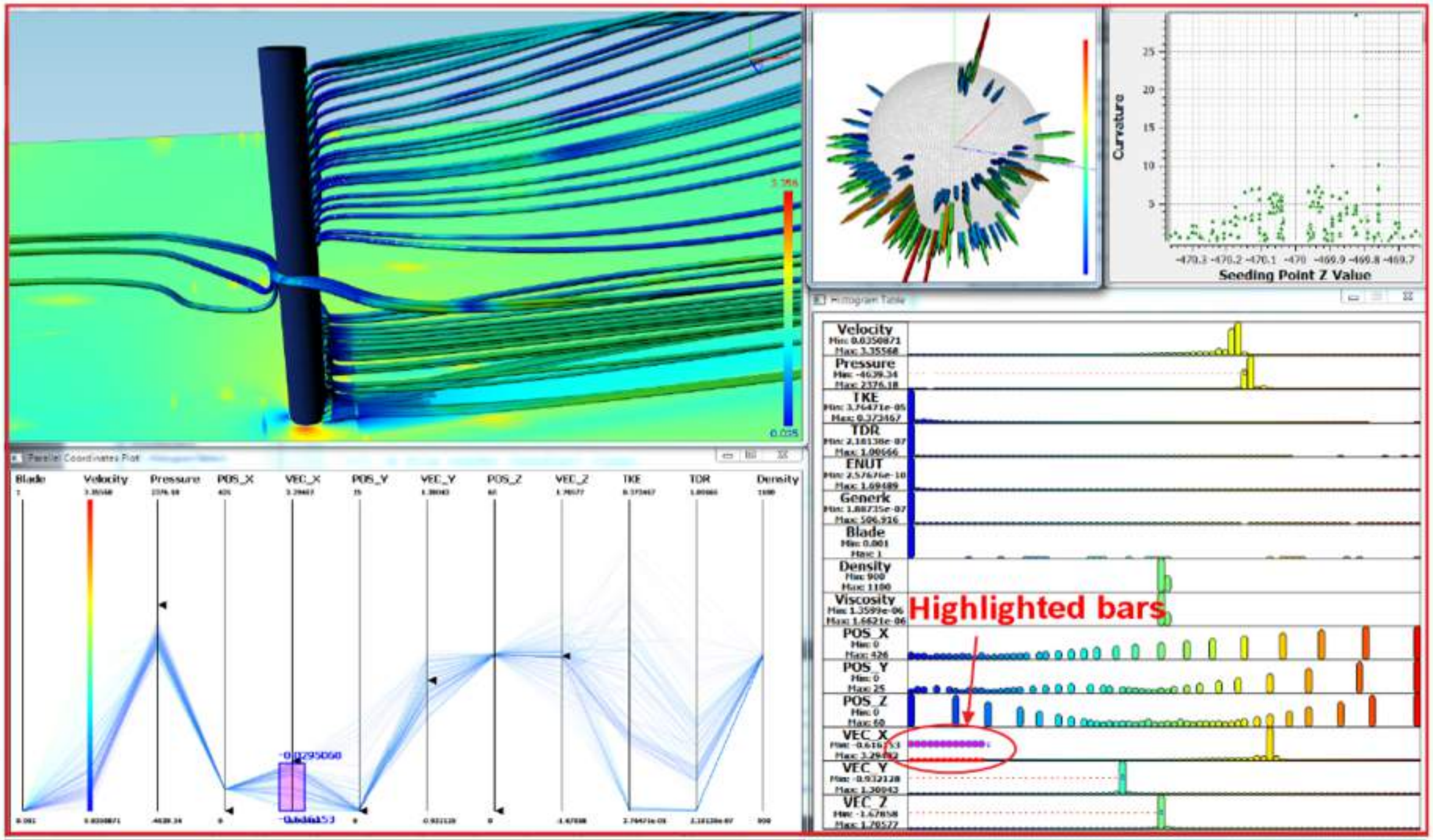
Interaction Techniques

Coordinate, Multiple Views (CMV)– Scientific and Information Visualization

Linking of Scientific and Information Visualization views

Linking and Brushing, Demo: https://youtu.be/51LOGGiSPMg

(Peng et al., 2014)



Further Reading

Zhao Geng, Zhenmin Peng, Robert S.Laramee, Rick Walker, and Jonathan C. Roberts, **Angular Histograms: Frequency-Based Visualizations for Large, High-Dimensional Data**, IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG), Vol. 17, No. 12, December 2011, pages 2572-2580

David H.S. Chung, Phillip A. Legg, Matthew L. Parry, Rhodri Bown, Iwan W Griffiths, Robert S. Laramee, and Min Chen, Glyph Sorting: Interactive Visualization for Multi-dimensional Data, Information Visualization, Vol 14, No 1, pages 76-90, January 2015

Zhenmin Peng, Zhao Geng, Michael Nicholas, Robert S. Larameee, Nick Croft, Rami Malki, Ian Masters, Chuck Hansen, Visualization of Flow Past a Marine Turbine: The Information-Assisted Search for Sustainable Energy, Computing and Visualization in Science, Vol 17, No 6, December 2014, pages 89-103

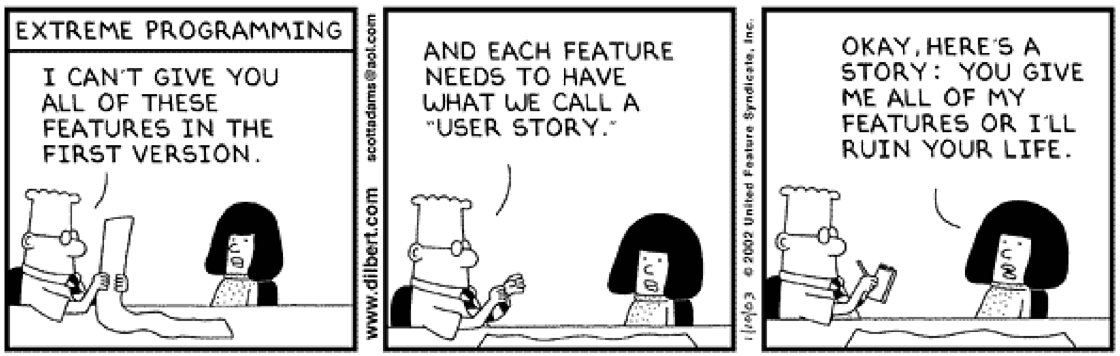
Acknowledgements

We thank the following:

Jorik Blaas, Charl Botha, Rhodri Bown, Guoning Chen, Min Chen, David Chung, T Nick Croft, Zhao Geng, Eduard Grundy, Iwan Griffiths, Charles Hansen, Helwig Hauser, Daniel Keim, Mark Jones, Phillip Legg, Dan Lipsa, Rami Malki, Ian Masters, Konstantin Mischaikow, Michael Nicholas, Matthew Parry, Frits H Post, Jonathan C Roberts, Rick Walker, Yunai Wang, Rory Wilson, Eugene Zhang

Visualization, Lecture #3

Volume Visualization, Part 1 (of 4)



Overview: Lecture #3

Contents of Volume Visualization, Part 1:

- Introduction
  - About Volume Data
  - Overview of Techniques
- The VolVis Conceptual Framework
- Simple Methods
  - slicing, cuberille

Volume Visualization

Introduction:

- VolVis = Visualization of Volume data
  - Picture/image 3D→2D
  - Projection (e.g., Maximum Intensity Projection-MIP), slicing, surface extractiontion, volume rendering, …
- Volume Data =
  - 3D× 1D Data
  - scalar data, 3D data space, space filling (dense-as opposed to sparse)
- User goals:
  - to gain insight into 3D Data
  - depends strongly on what user is interested in (focus + context)

Volume Data

Where does the data come from?

- Medical Applications
  - Computer Tomography (CT)
  - Magnet Resonance Imaging (MRI)
- Material testing/control
  - Industry-CT
- Simulation
  - Finite element methods (FEM)
  - Computational fluid dynamics (CFD)
- And other sources

3D Data Space

How is the volume data organized?

- Cartesian, i.e., regular grids:
    1. CT/MR: often dx=dy<dz, e.g. 135 sclices (z) by 512^2 values in x & y
    2. **Data enhancement**: iso-stack-computation = interpolation of additional slices, so that dx=dy=dz $\Rightarrow 512^3$ Voxel
    3. Data: **Cells** (Faces), Corners: **Voxel**
- Curvilinear grid i.e., unstructured:
    - Data organized as Tetrahedra or Hexahedra
    - Often: Conversion to Tetrahedra

Terminology

Tetrahedron. "A 3D primary cell that is a simplex with four triangular faces, six edges, and four vertices." –the Visualization Toolkit (VTK)

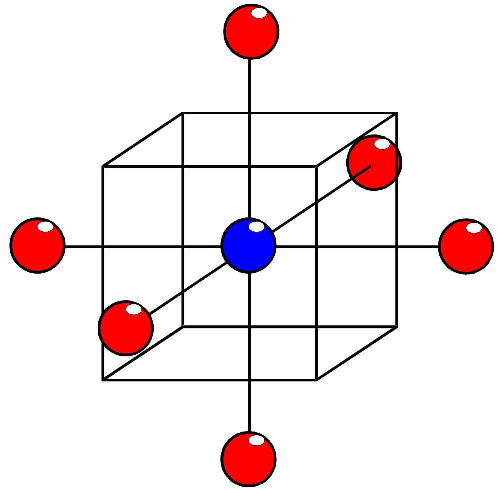
Volume Visualization (VolVis) – Challenges

Challenges:

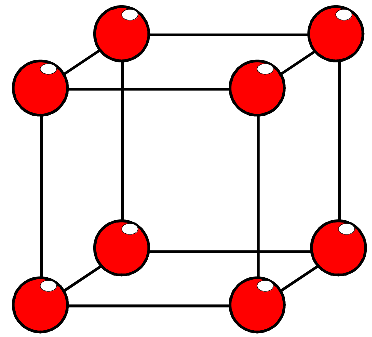
- rendering projection,
  so much information and so few pixels
- large data sets, e.g.,
  $512 \times 512 \times 1024$ voxels at 16 bit/voxel = 512 Mbytes
- Computational Speed,
  Interaction is very important, >10 frames per second (fps)

Voxels vs. Cells

Two ways to view the volume data:



- Data as a set of voxels
  - **voxel** = short for volume element (recall: pixel = "picture element")
  - voxel = point sample in 3D
  - not always interpolated
- Data as a set of cells
  - **cell** = Cubic Primitive (3D)
  - corners: 8 voxel
  - value(s) in cell: are always interpolated



Robert S. Laramee
r.s.laramee@swansea.ac.uk

Interpolation

(More) Terminology:

- **Interpolate**: ``Estimate a value of a function at a point, ***p***, given known function values and points that bracket ***p***."

f(1)*● ← *d* → (*p*)● ← (*D-d*) → ●*f(2)

Nearest neighbor: p = f(0)

Linear: p = f(1)(D-d)/D + f(2)d/D

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Interpolation

Example

f(1)● ← ***d*** → (***p***)● ← (***D-d***) → ●***f(2)***

T(1)● ← ***d*** → T(***x***)● ← (***D-d***) → ●***T(2)***

100C● ← ***2*** → ***T(?)●*** ← (***10-2***) → ●***200C***

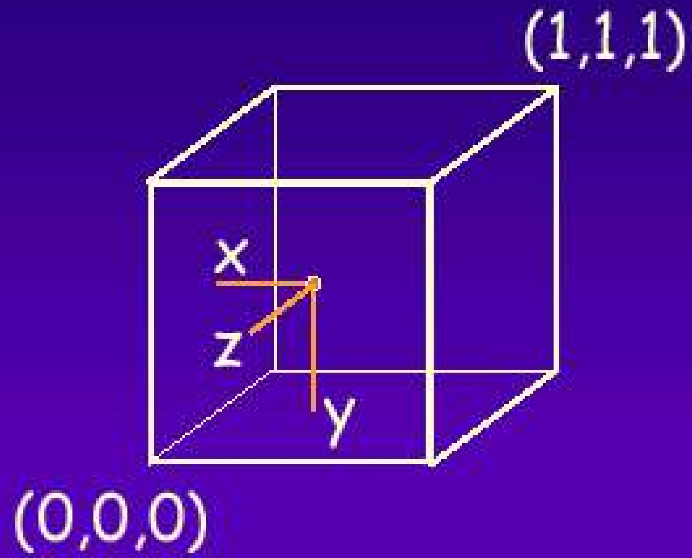
Nearest neighbor: T(?) = 100C

Linear:

T(?) = 100C(10-2)/10 + 200C(2/10)

120C = 100C(.8) + 200C(.2)

Interpolation



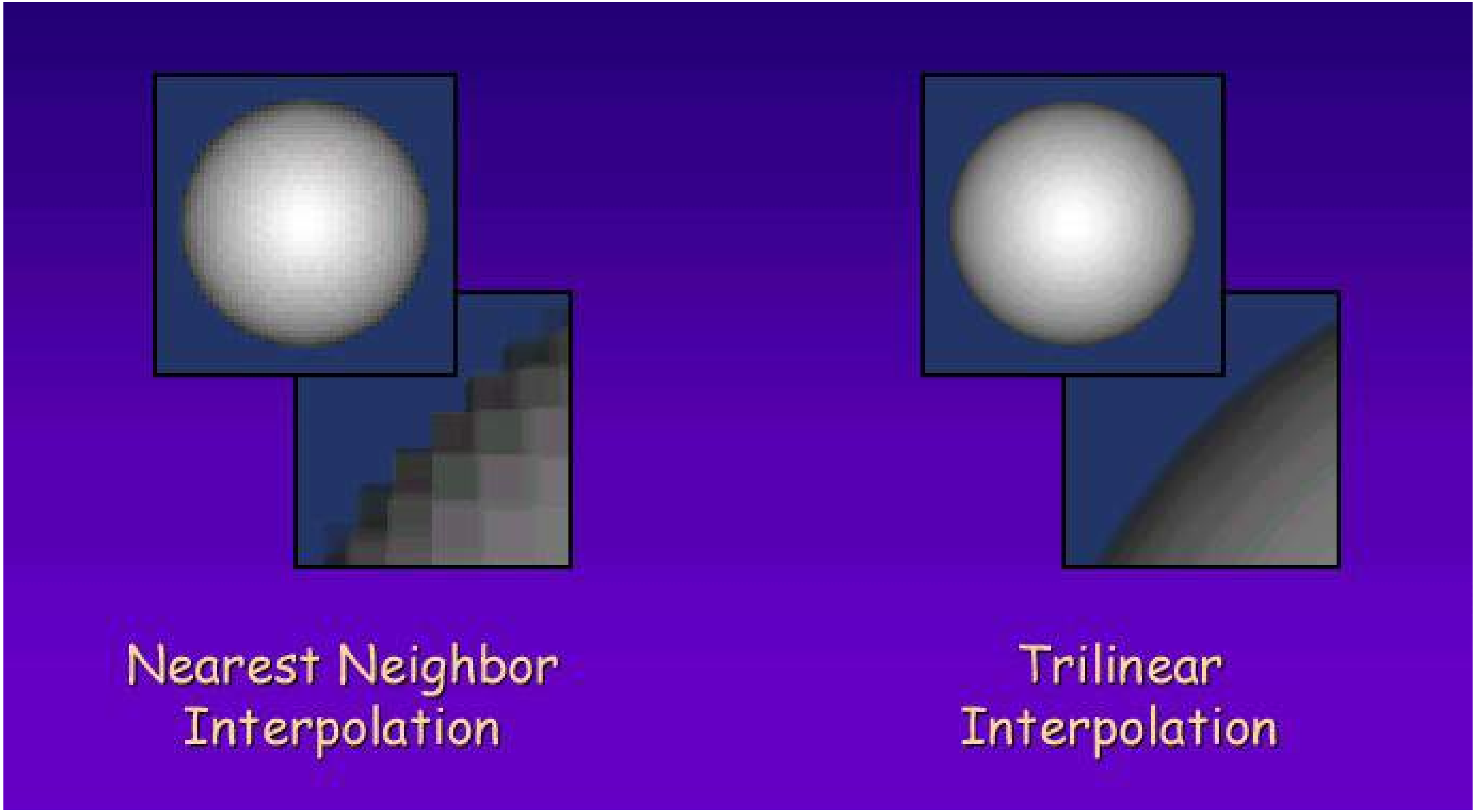
$$v = S(\mathrm{rnd}(x), \mathrm{rnd}(y), \mathrm{rnd}(z))$$

Nearest Neighbor

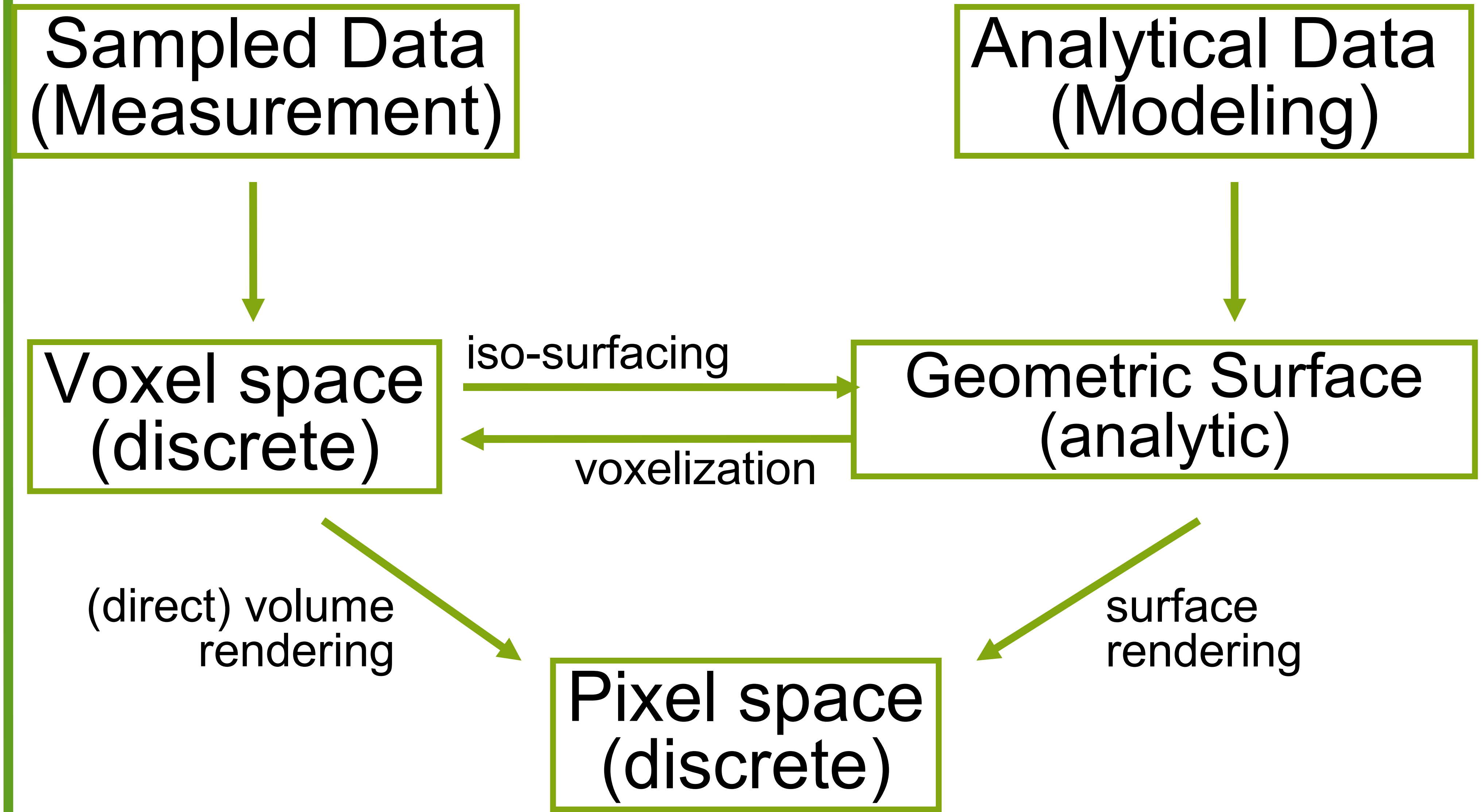
$$\begin{aligned} v = \; & (1-x)(1-y)(1-z)S(0,0,0) + \\ & (x)(1-y)(1-z)S(1,0,0) + \\ & (1-x)(y)(1-z)S(0,1,0) + \\ & (x)(y)(1-z)S(1,1,0) + \\ & (1-x)(1-y)(z)S(0,0,1) + \\ & (x)(1-y)(z)S(1,0,1) + \\ & (1-x)(y)(z)S(0,1,1) + \\ & (x)(y)(z)S(1,1,1) \end{aligned}$$

Trilinear

Interpolation – Influence



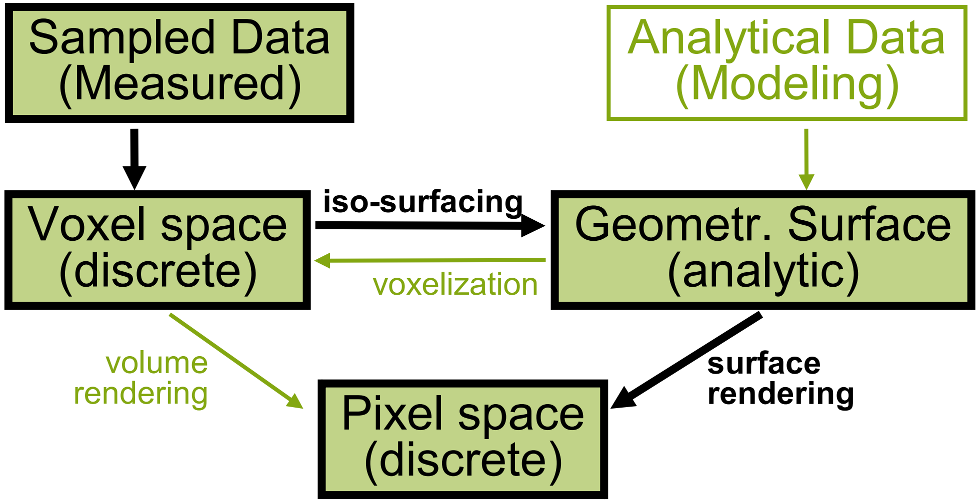
Conceptual VolVis Framework

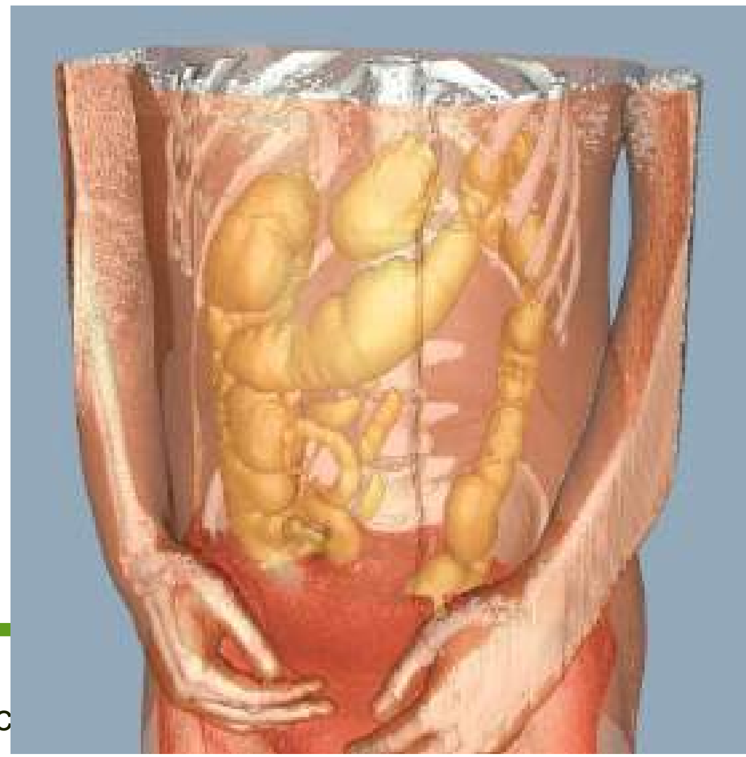


Conceptual VolVis Framework

Example 1:

- CT measurement
- iso-stack compuation
- iso-surface computation (marching cubes)
- Surface rendering (OpenGL)



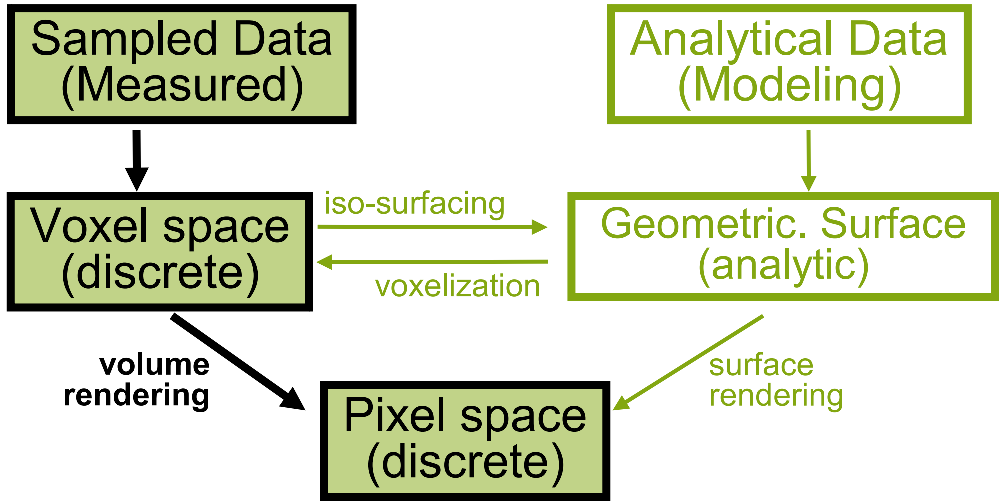


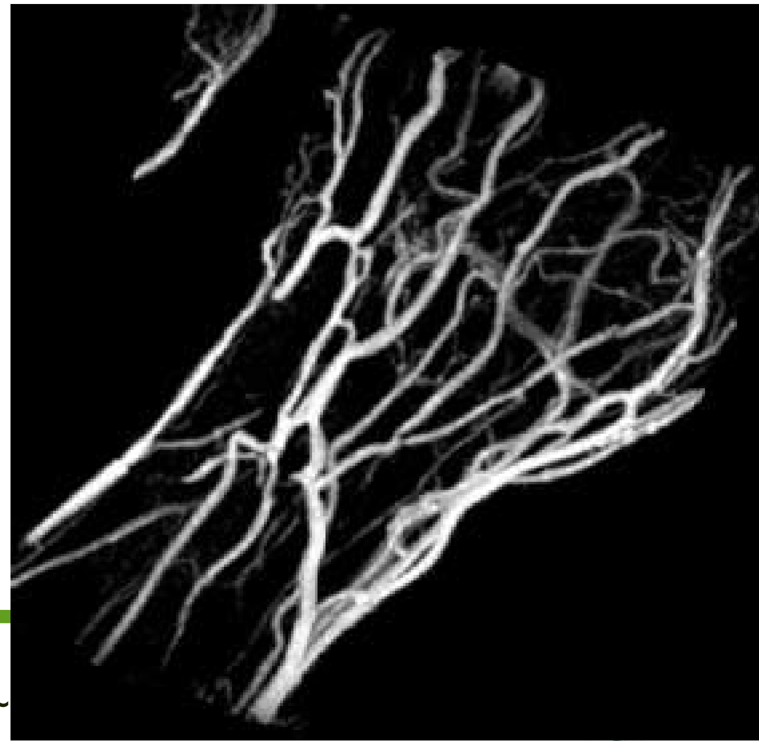
Robert S. Laramee
r.s.laramee@swansea.ac.uk http://cs.swan.ac.uk/~c

Conceptual VolVis Framework

Example 2:

- MR Measurement
- iso-stack computation
- MIP (maximum intensity projection)
- Image: Vessels in Hand

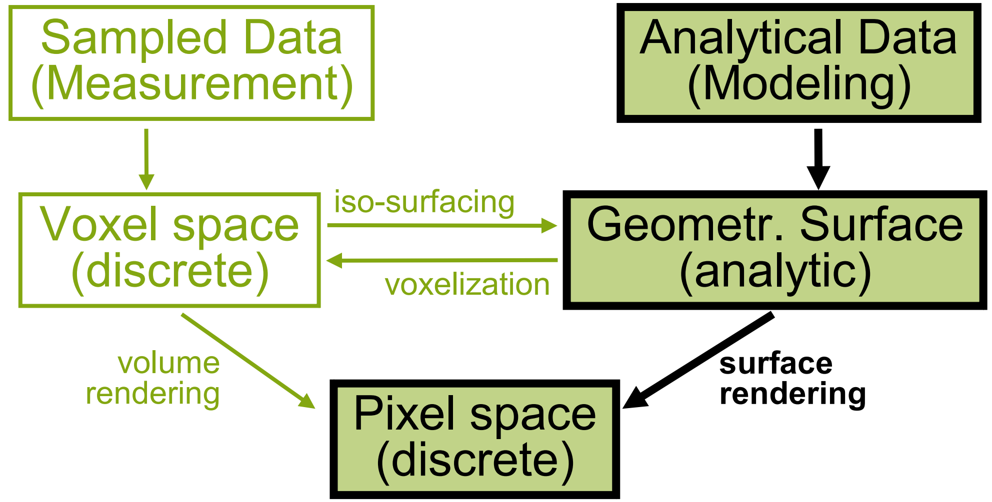


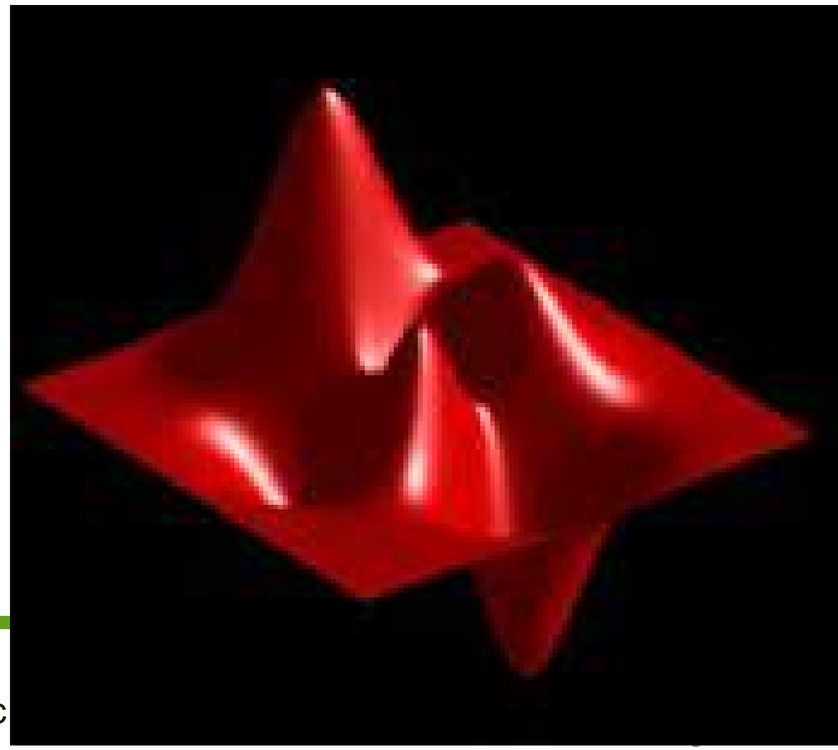


Conceptual VolVis Framework

Example 3:

- potential function $\rho(x,y,z)$
- iso-surface $\rho(x,y,z)=\rho_0$
- surface ray tracing





VolVis-Techniques – Overview

Simple Methods:

- (Indirect) slicing, MPR (multi-planar reconstruction)
- (Indirect) Surface-fitting methods:
  - marching cubes (marching tetrahedra)
- Direct volume visualization:
  - ray casting
  - shear-warp factorization
  - splatting
  - 3D texture mapping

Image-order vs. Object-order

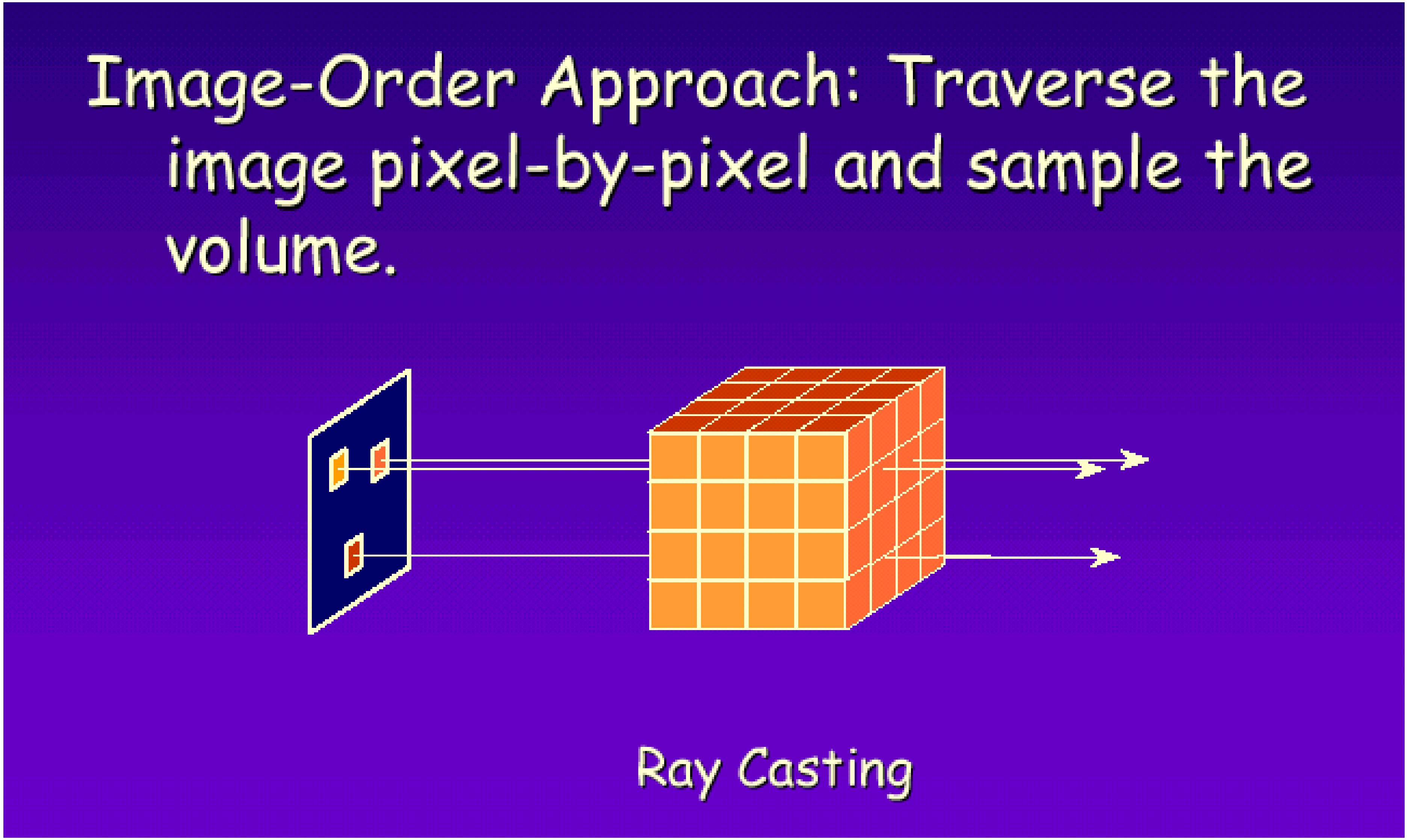
Image-order:

- FOR every pixel DO: …
- Cost is related to size of image (measured in pixels)
- Example: VolVis technique-ray casting

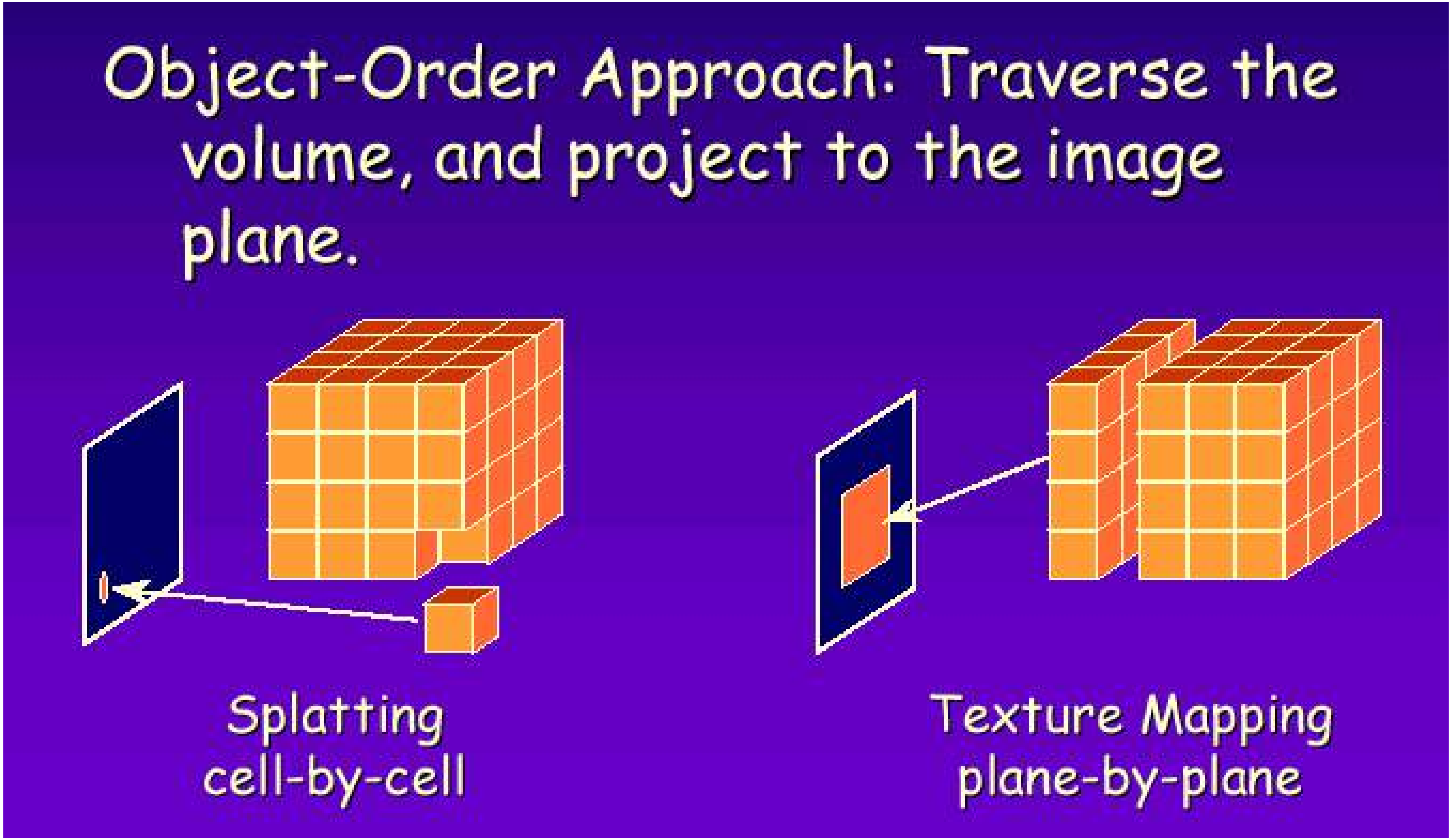
Object-order:

- FOR every object (voxel) DO: …
- Cost is related to the number of objects (number of voxels)
- Example: VolVis technique-splatting

Image-order Approach



Object-order Approach



Simple VolVis Methods

Slicing, cuberille

Slicing

Slicing:

- Axis-aligned slices
- regular grids: simple
- no transfer function
  no color
- windowing:
  contrast adjustment/enhancement

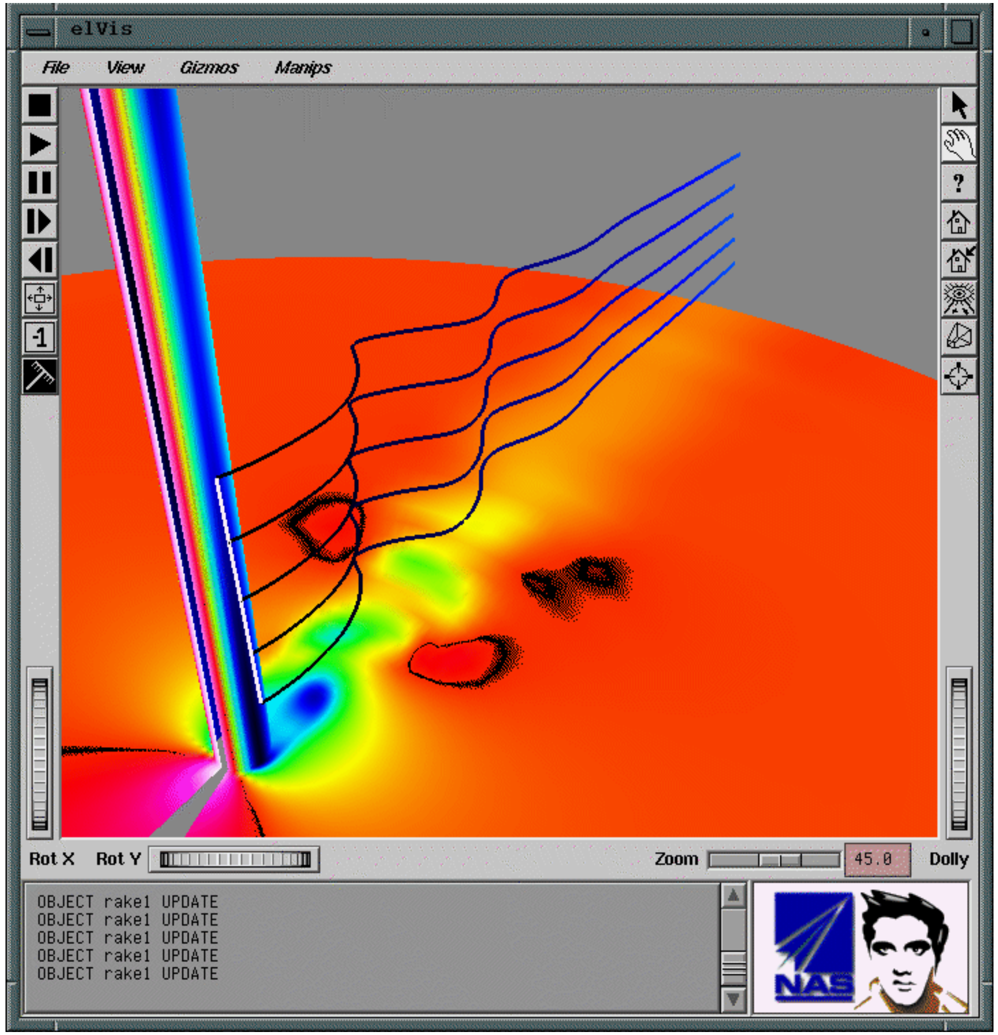
Slicing

Not so simple:

- slicing through all grids
- interpolation necessary

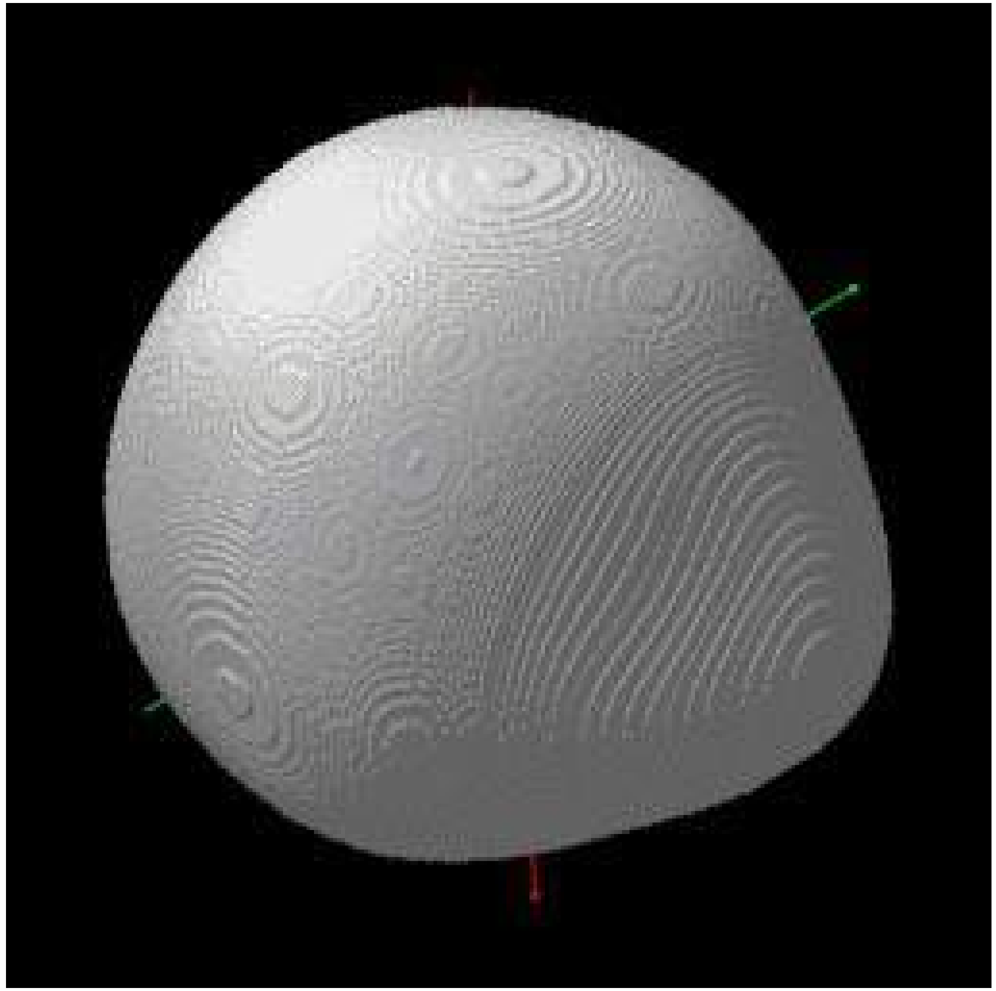
Slicing:

- Combines well with 3D-Vis.



Slicing

Example:
Analytical Data set



Cuberille

Cuberille [Herman & Liu 1979]:

- search for voxel containing iso-value
- Voxel $\Rightarrow$ Cubes
- render all 6 sides
- aliasing
- outdated

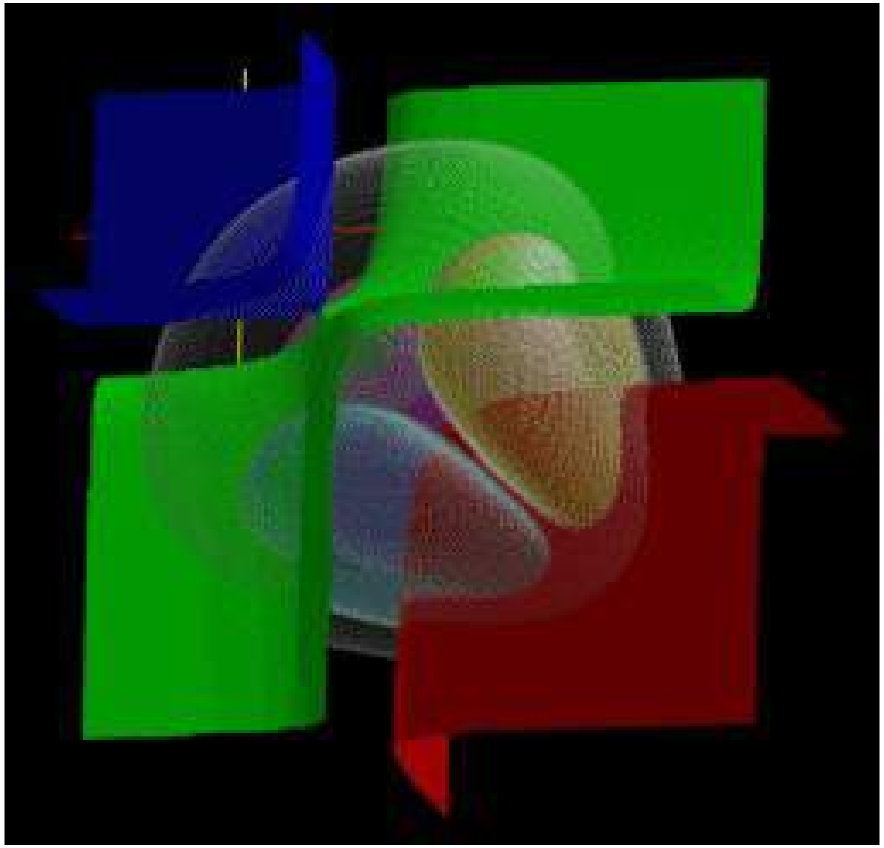
Terminology: Aliasing

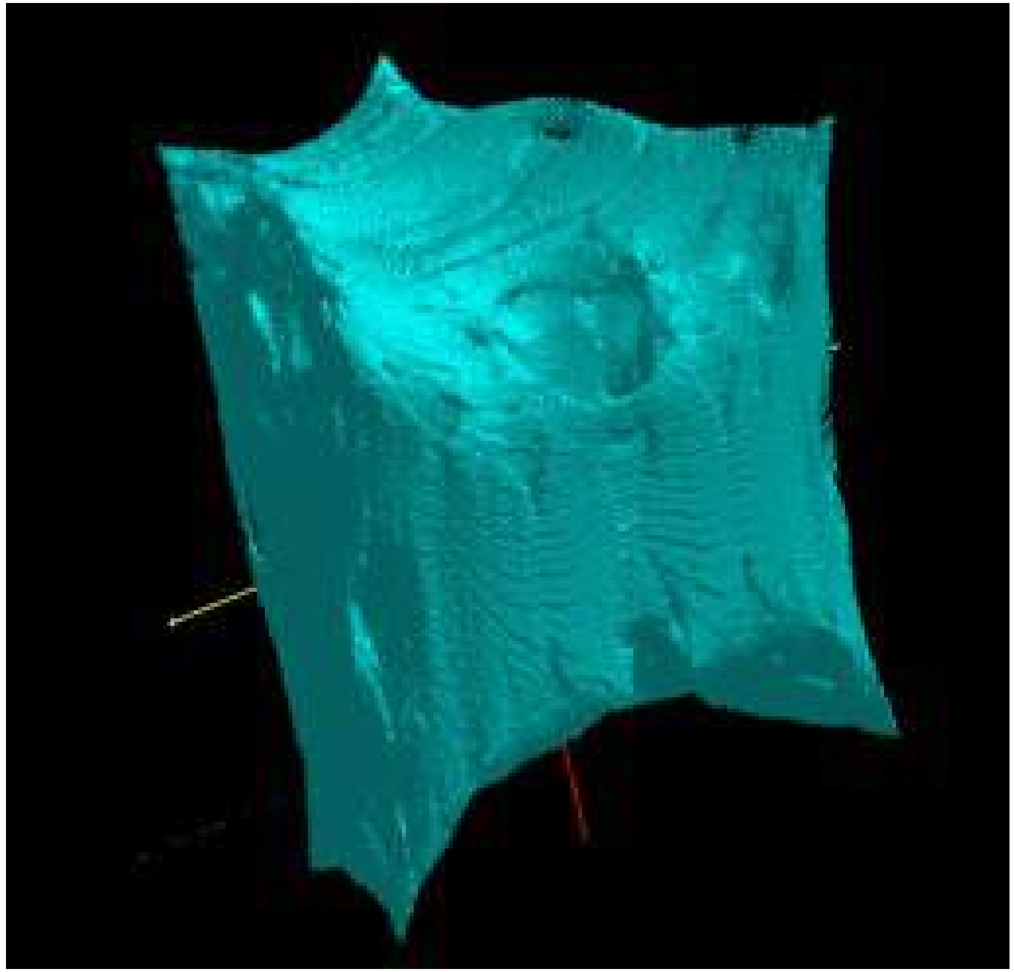
Aliasing: ``A rendering technique that assigns to pixels the color of the primitive being rendered, regardless of whether that primitive covers all or only a portion of the pixel's area. This results in jagged edges, or jaggies." -The OpenGL Programming Guide, 5th Edition (The "Red Book")

Anti-Aliasing: ``A rendering technique that assigns pixel colors based on the fraction of the pixel's area that's covered by the primitive being rendered. Anti-aliased rendering reduces or eliminates the jaggies that result from aliased rendering." -The OpenGL Programming Guide, 5th Edition (The "Red Book")

Cuberille

Examples:





Acknowledgements

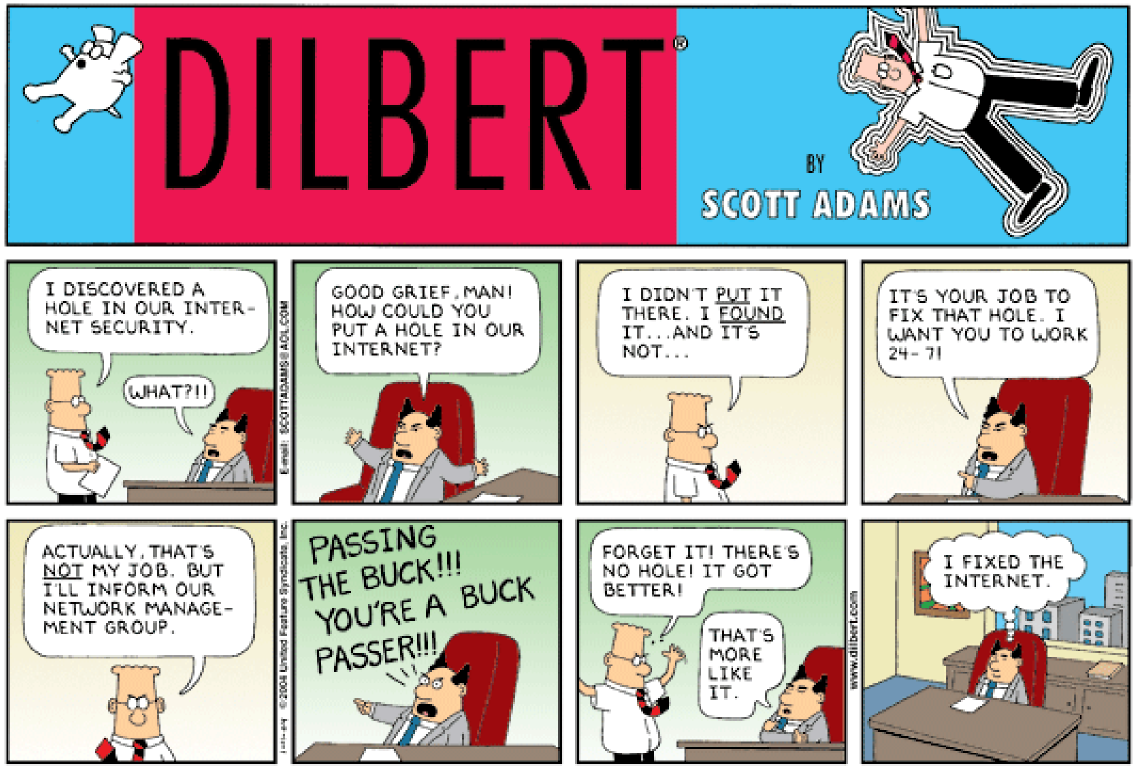
For more information, please see, **Data Visualization: Principles and Practice, Chapter 10 Volume Visualization** by A.C. Telea, AK Peters, 2008

We thank the following people for lecture material:

- Michael Meißner
- Roger Crawfis (Ohio State Univ.)
- Hanspeter Pfister
- Meister E. Gröller
- Torsten Möller
- Dirk Bartz
- Markus Hadwiger
- Helwig Hauser

Visualization, Lecture #4

Volume Visualization, Iso-Surface Extraction and Marching Cubes



Robert S. Laramee
r.s.laramee@swansea.ac.uk

Computer Science
Gwyddor Cyfrifiadur

Review: Lecture #3

Contents of previous lecture on volume Visualization:

- Introduction
  - About Volume Data
  - The Conceptual VolVis Framework
  - Overview of Techniques
  - Image vs Object order
- Simple Methods
  - slicing, cuberille

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Overview: Lecture #4

Contents of Volume Visualization, Part 1:

- Marching Cubes
  - Iso-contours
  - Iso-Surfaces
  - The Marching Cubes (MC) Algorithm

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes

Iso-Surfaces: Generation and Presentation

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Computer Science
Gwyddor Cyfrifiadur

Surface Rendering

Terminology

Isosurface: “A surface representing a constant valued scalar function” –The VTK

Isovalue: “The scalar value used to generate an isosurface.” The VTK

About Surface rendering:

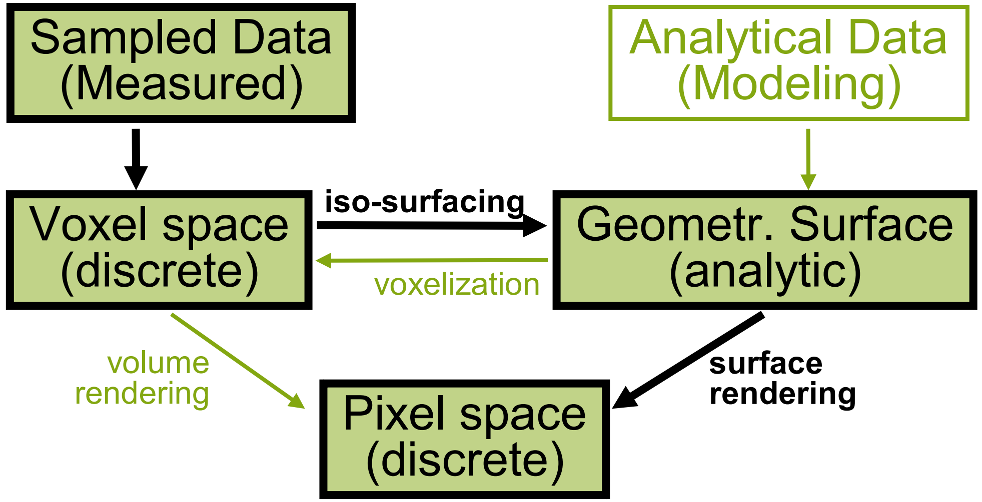
- indirect volume visualization technique
- isosurfaces are computed from 3D volume data

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Conceptual VolVis Framework

Example:

- CT measurement
- iso-stack compuation
- iso-surface computation (marching cubes)
- Surface rendering (OpenGL)



Robert S. Laramee
r.s.laramee@swansea.ac.uk

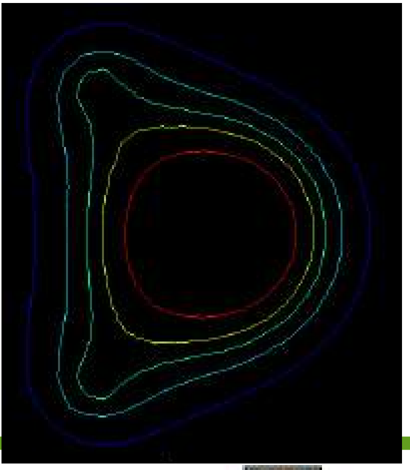
Iso-Surfaces

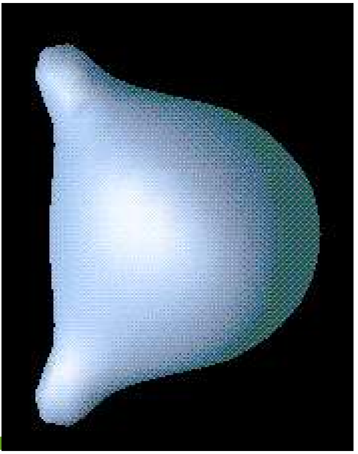
Indirect representation

Characteristics:

- Meaningful isovalue,
- Isovalue separates substances
- Interest in boundary regions
- Very selective (binary choice/filter)
- Uses traditional hardware
- Shading gives a 3D

impression



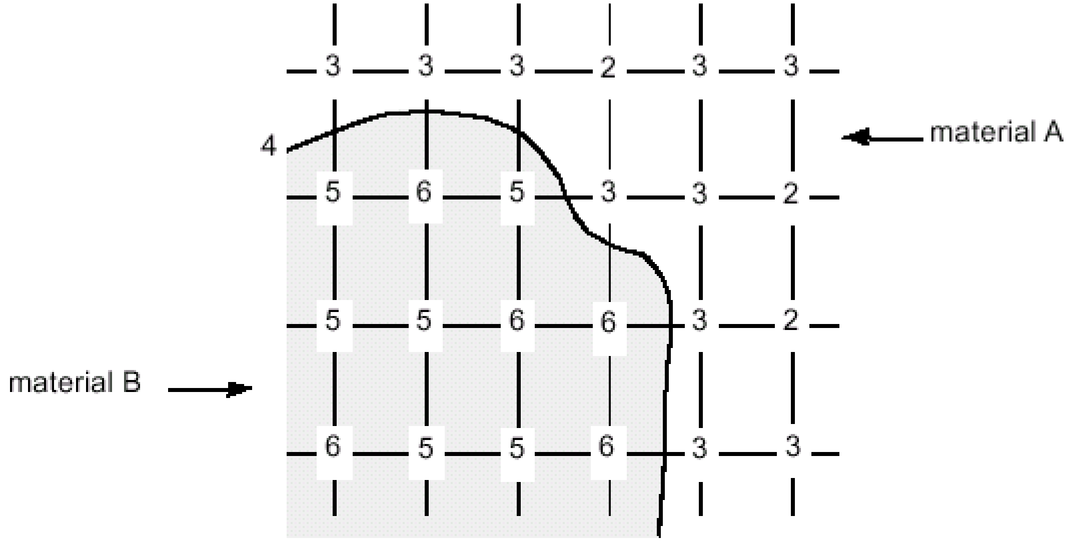


Robert S. Laramee
r.s.laramee@swansea.ac.uk

Planar Data and Iso-Contours

Iso-Contours:

- isovalue f_0
- Separate values > f_0 from values $\leq f_0$
- Can only be approximated from given samples
- Form/Position depends on reconstruction (interpolation)



Robert S. Laramee
r.s.laramee@swansea.ac.uk

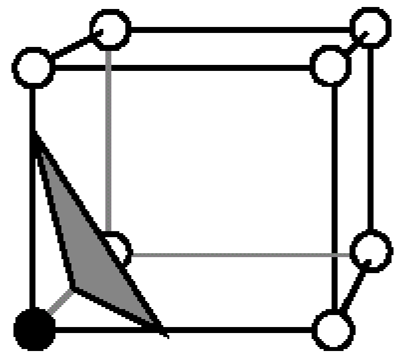
Approximating the Iso-Surface

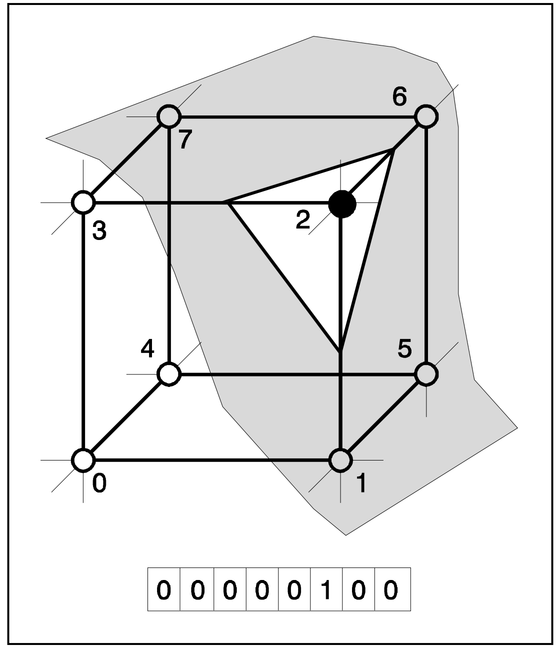
Approach:

- Iso-surface cuts volume = each cell is searched

Idea:

- Divide the Iso-Surface into a per-volume cell representation
- Use triangles





Marching Cubes - Overview

Overview of MC algorithm

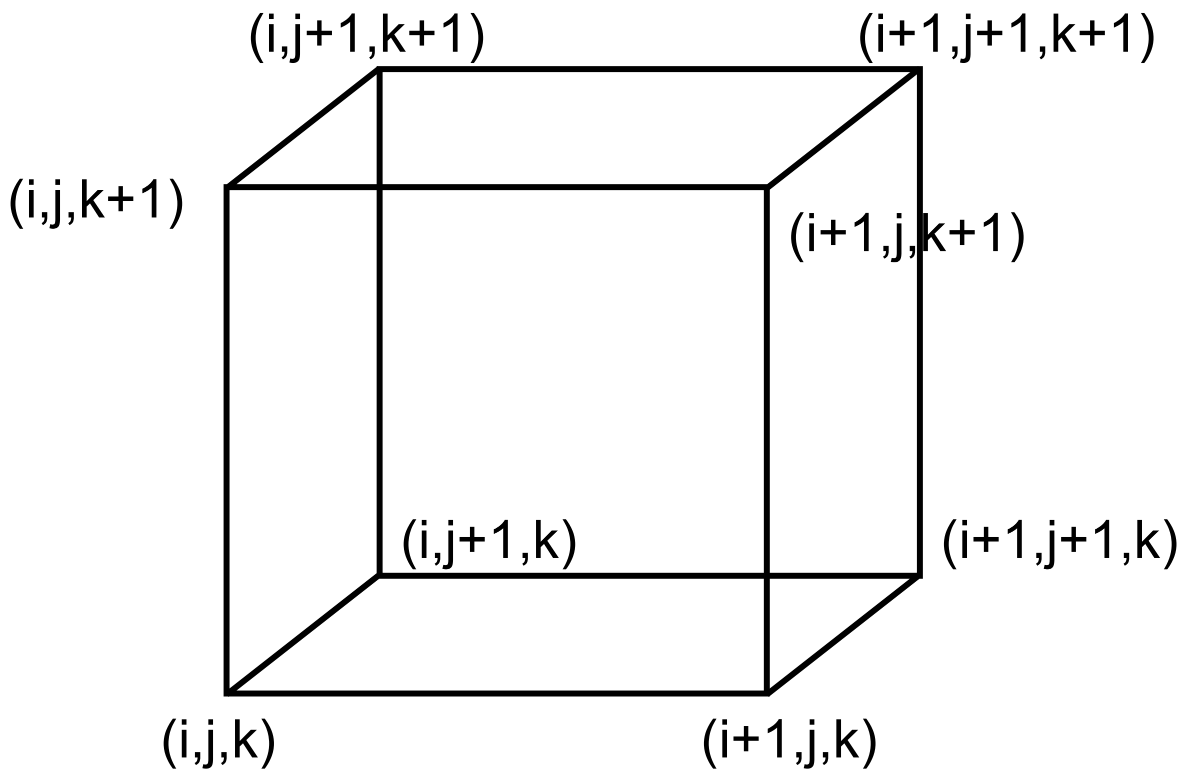
- Cell consists of 4(8) pixel (voxel) values:
  (i+[01], j+[01], k+[01])

1. Examine a cell
2. Classify each vertex as inside or outside
3. Build an index
4. Get edge list from table[index]
5. Interpolate the edge location
6. Compute gradients
7. Consider ambiguous cases
8. Go to next cell

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes – Step 1

Step 1: Consider a cell defined by eight data values

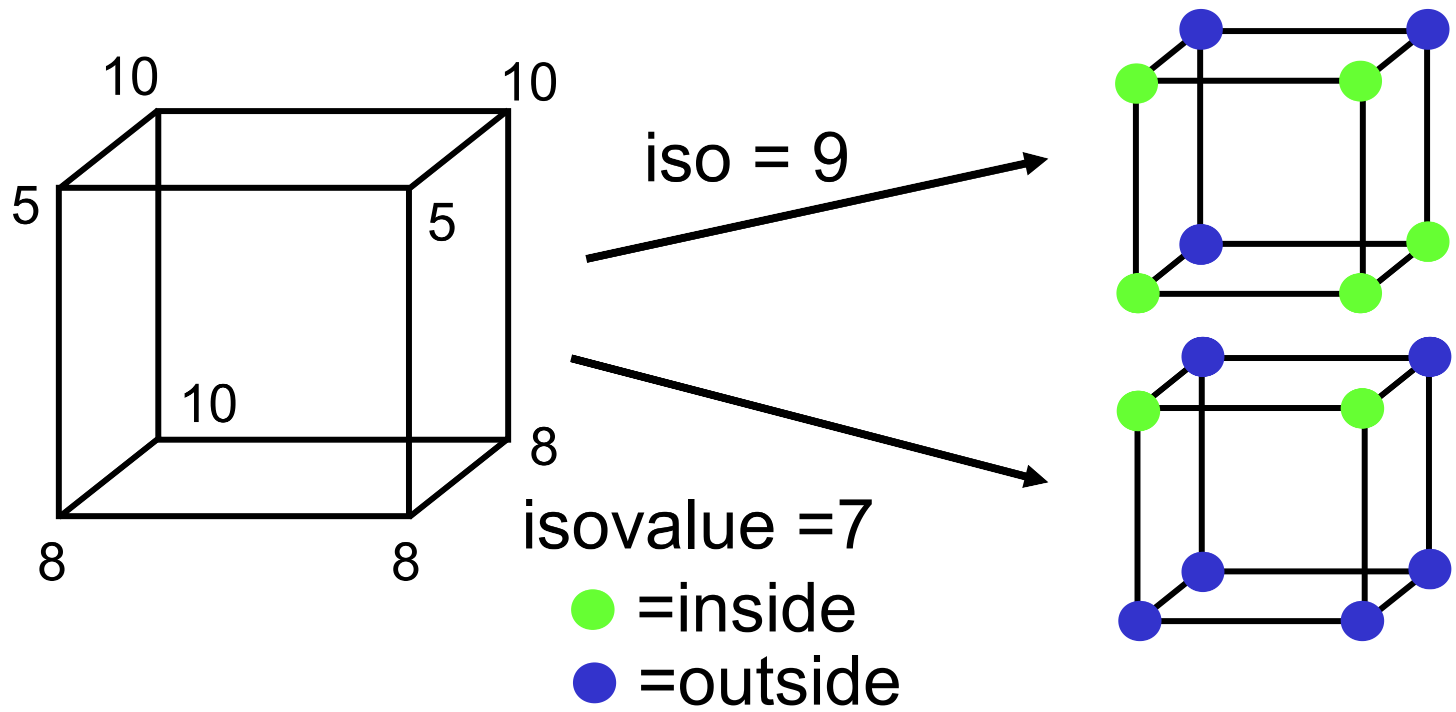


Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes – Step 2

Step 2: Classify each voxel according to whether it lies:

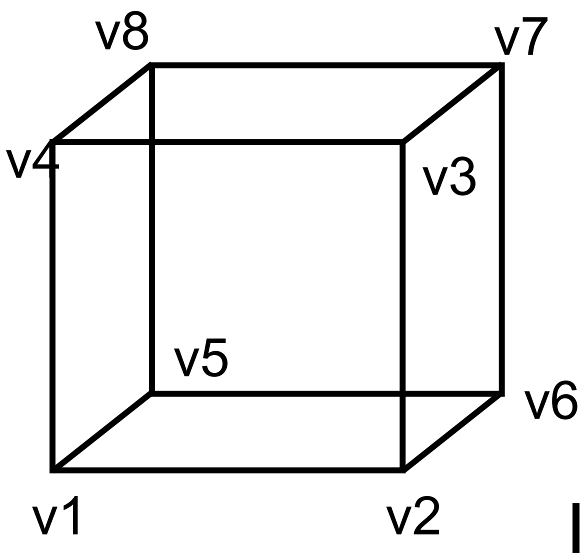
- Outside the surface (value > isosurface value)
- Inside the surface (value <= isosurface value)



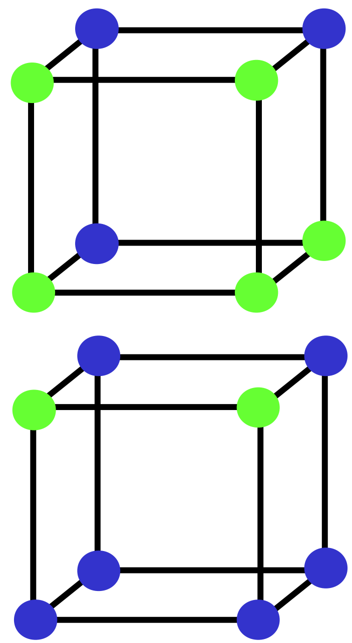
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes Step 3

Step 3: Use the binary labeling of each voxel vertex to create an index



inside =1
outside=0



11110100

00110000

Index:

v1	v2	v3	v4	v5	v6	v7	v8

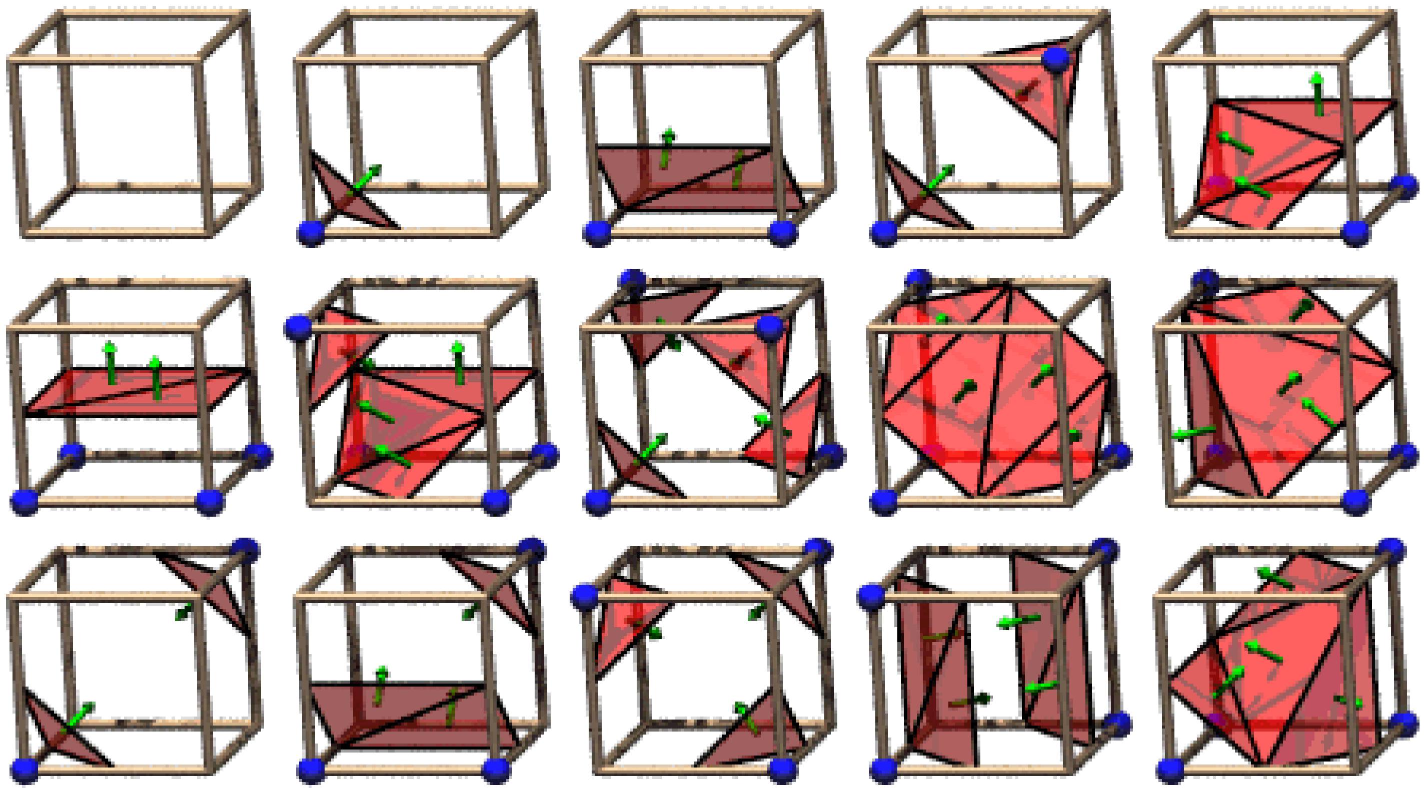
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Computer Science
Gwyddor Cyfrifiadur

Marching Cubes – Step 4

Step 4: For a given index, access an array storing a list of edges

All 256 cases can be derived from 1+14=15 base cases due to symmetries



The 15 Cube Combinations

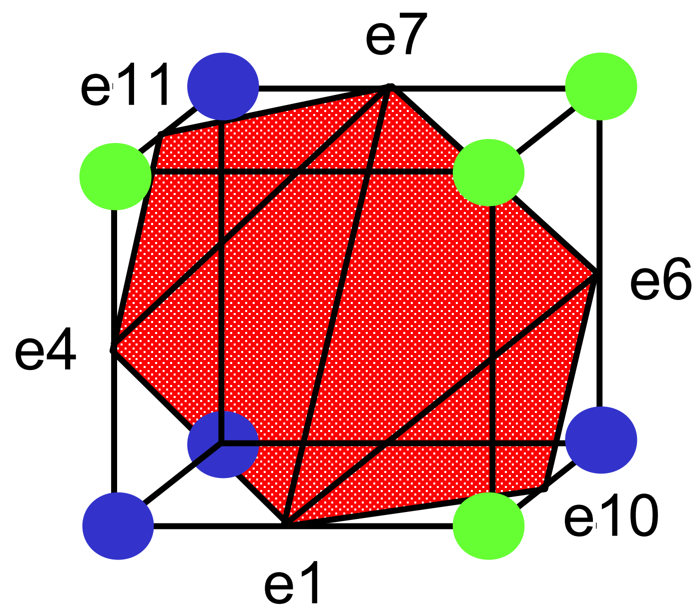
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes – Step 4 continued

Step 4 *cont.*: Get edge list from lookup (case) table

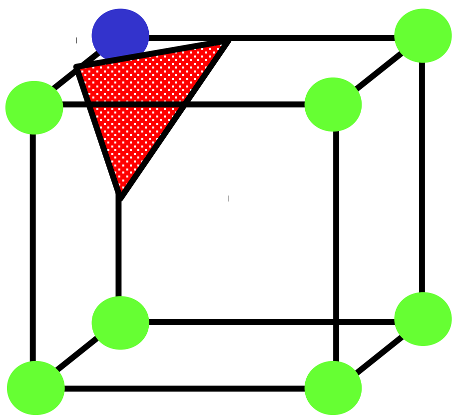
- Example for

  Index = 10110001
  triangle 1 = e4,e7,e11
  triangle 2 = e1, e7, e4
  triangle 3 = e1, e6, e7
  triangle 4 = e1, e10, e6



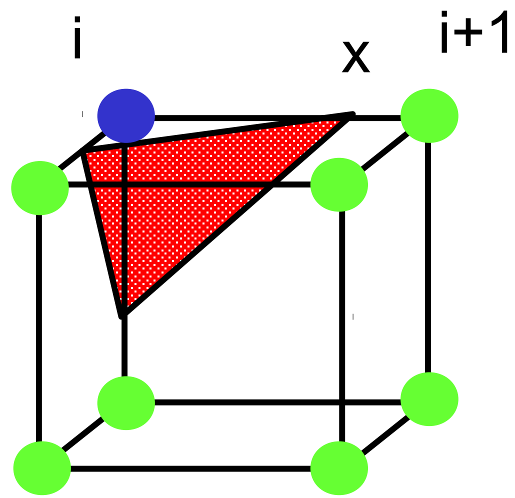
Marching Cubes – Step 5

Step 5: For each triangle edge, find the vertex location along the edge using linear interpolation of the voxel values



=10

=0



T=5

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

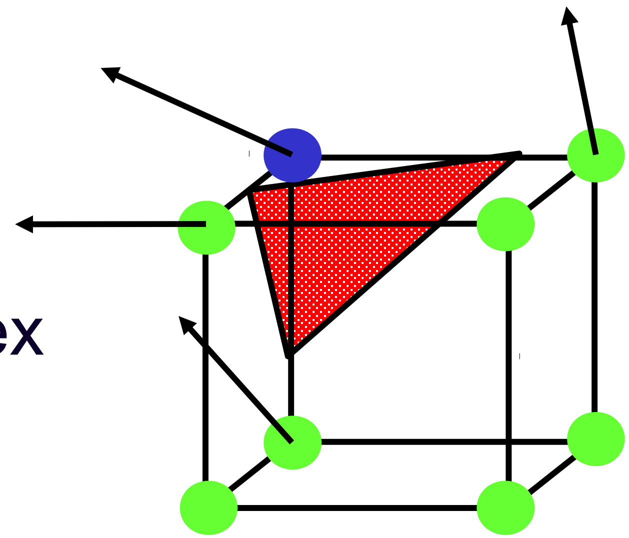
T=8

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes – Step 6

Step 6: Calculate the normal at each cube vertex (central differences)

- $G_x = V_{x+1,y,z} - V_{x-1,y,z}$
  $G_y = V_{x,y+1,z} - V_{x,y-1,z}$
  $G_z = V_{x,y,z+1} - V_{x,y,z-1}$
- Use linear interpolation to compute the polygon vertex normal (of the isosurface)

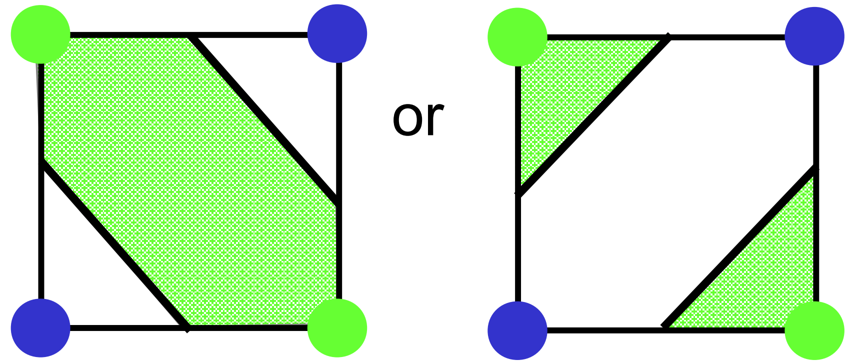


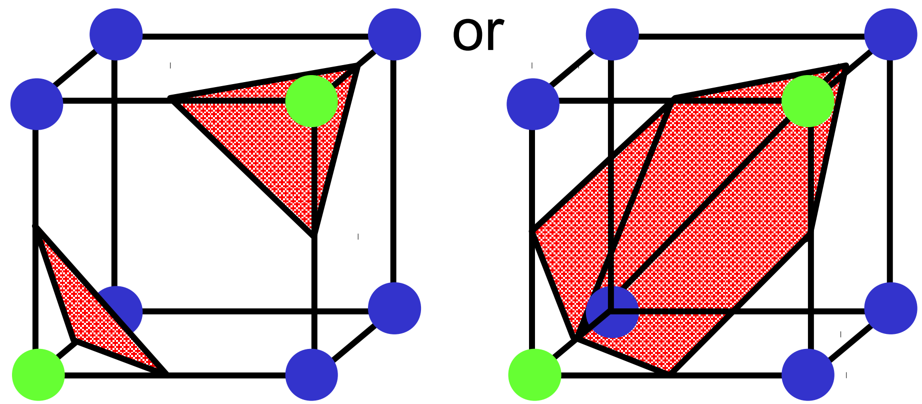
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes – Step 7

Step 7: Consider ambiguous cases

- Ambiguous cases: 3, 6, 7, 10, 12, 13
- Adjacent vertices: different states
- Diagonal vertices: same state
- Resolution: choose one case (the right one)

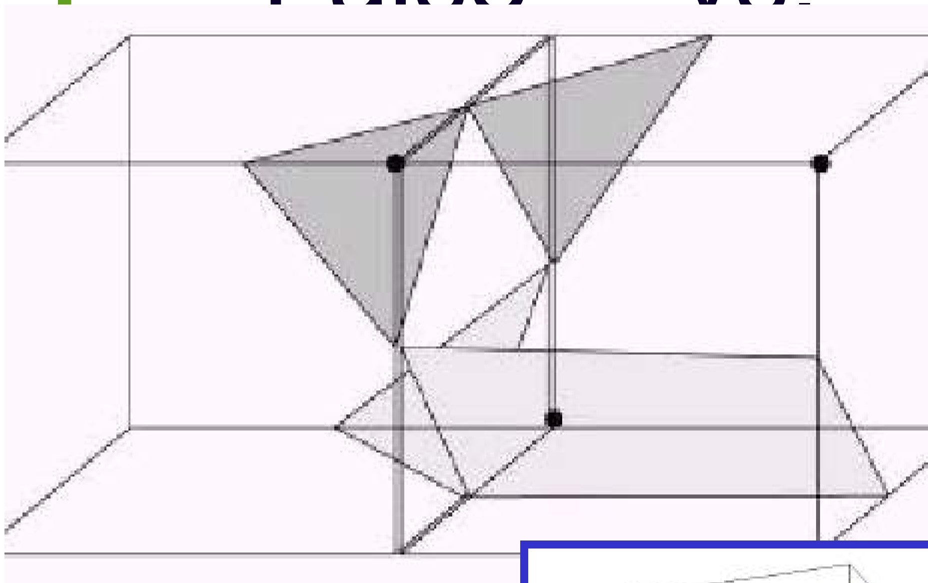


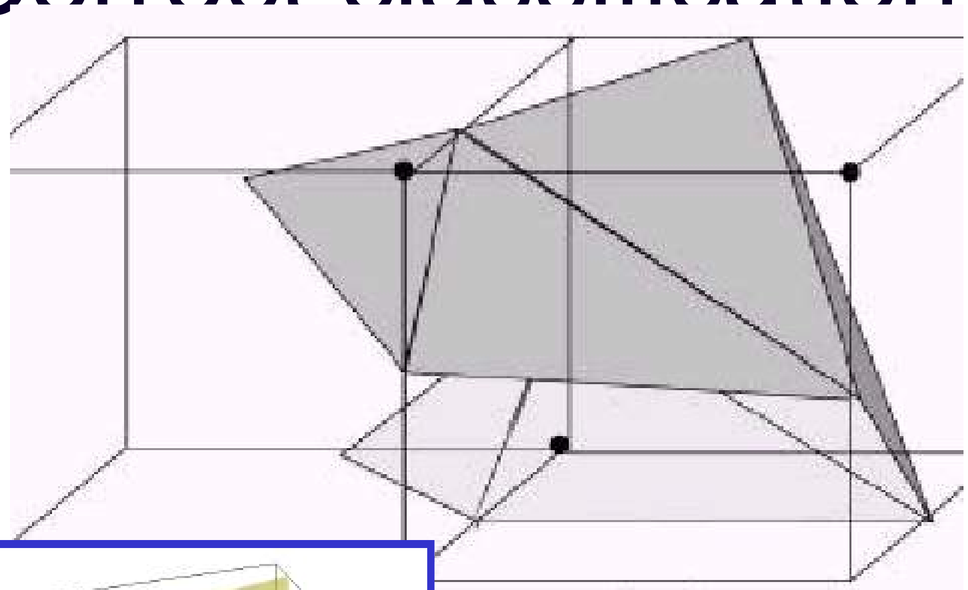


Robert S. Laramee
r.s.laramee@swansea.ac.uk

Warning: Surface Discontinuities

- False vs. Correct classification!





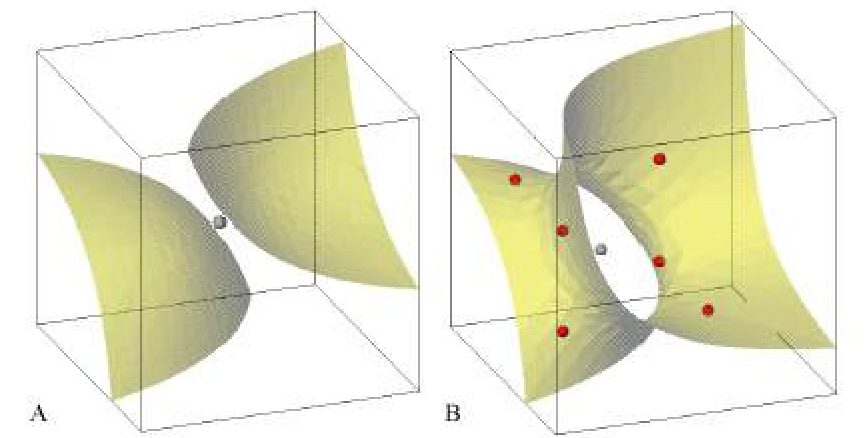


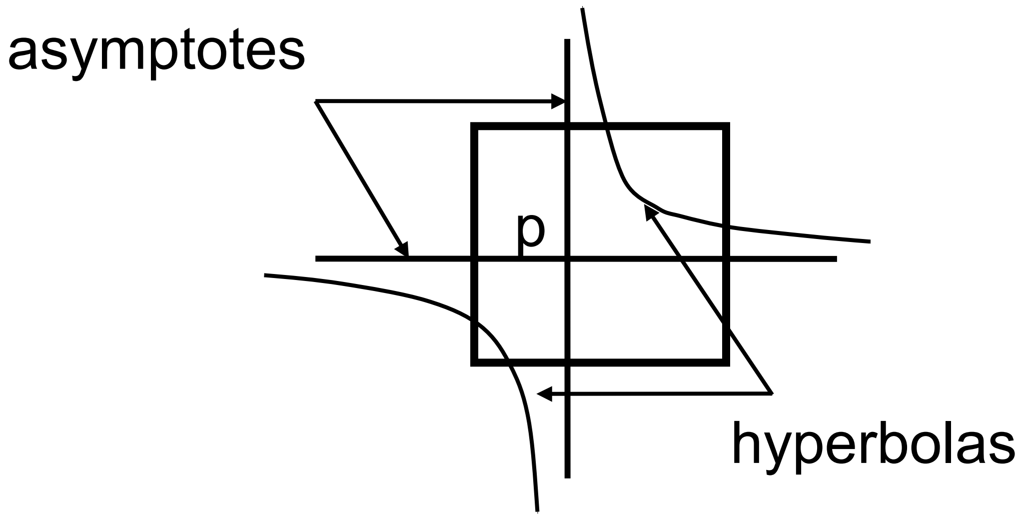
Figure 4: *Two internal configurations for the Marching Cubes configuration 5*

Marching Cubes – Step 7 Continued

Step 7 *cont*.: Consider ambiguous cases

Asymptotic Decider [Nielson, Hamann 1991]

- Assume bilinear interpolation within a face
- Hence isosurface is a hyperbola
- Compute the point p where the asymptotes meet
- Sign of S(p) decides the connectivity

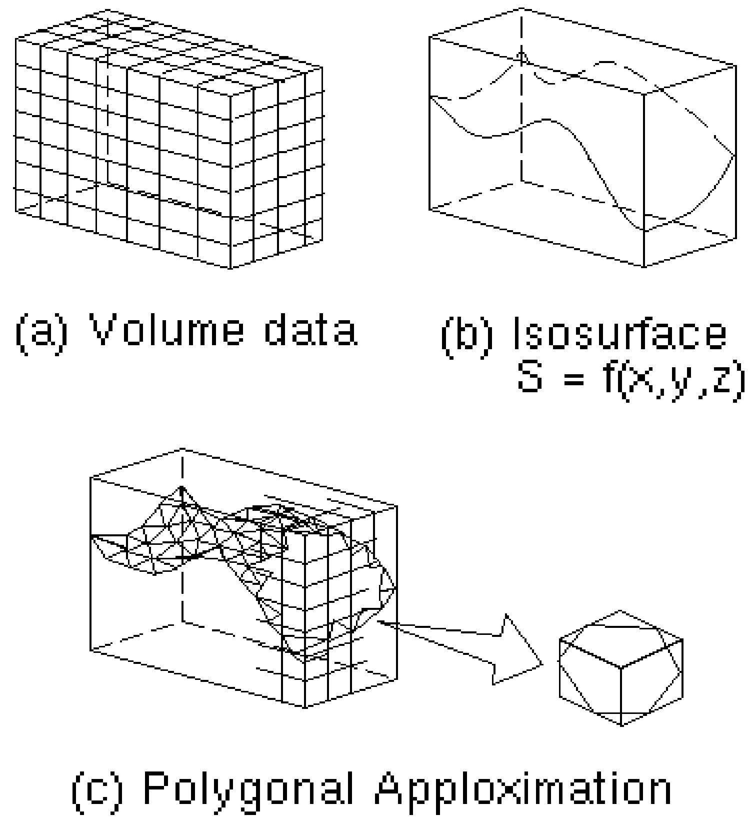


Robert S. Laramee
r.s.laramee@swansea.ac.uk

Marching Cubes - Summary

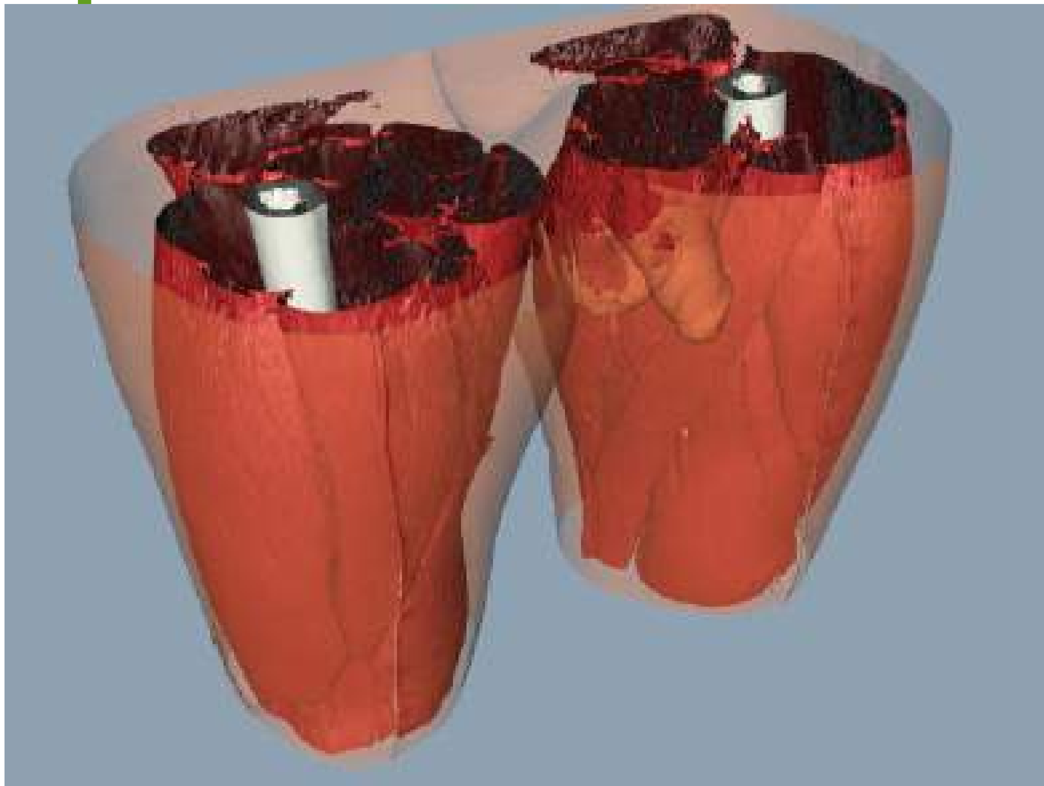
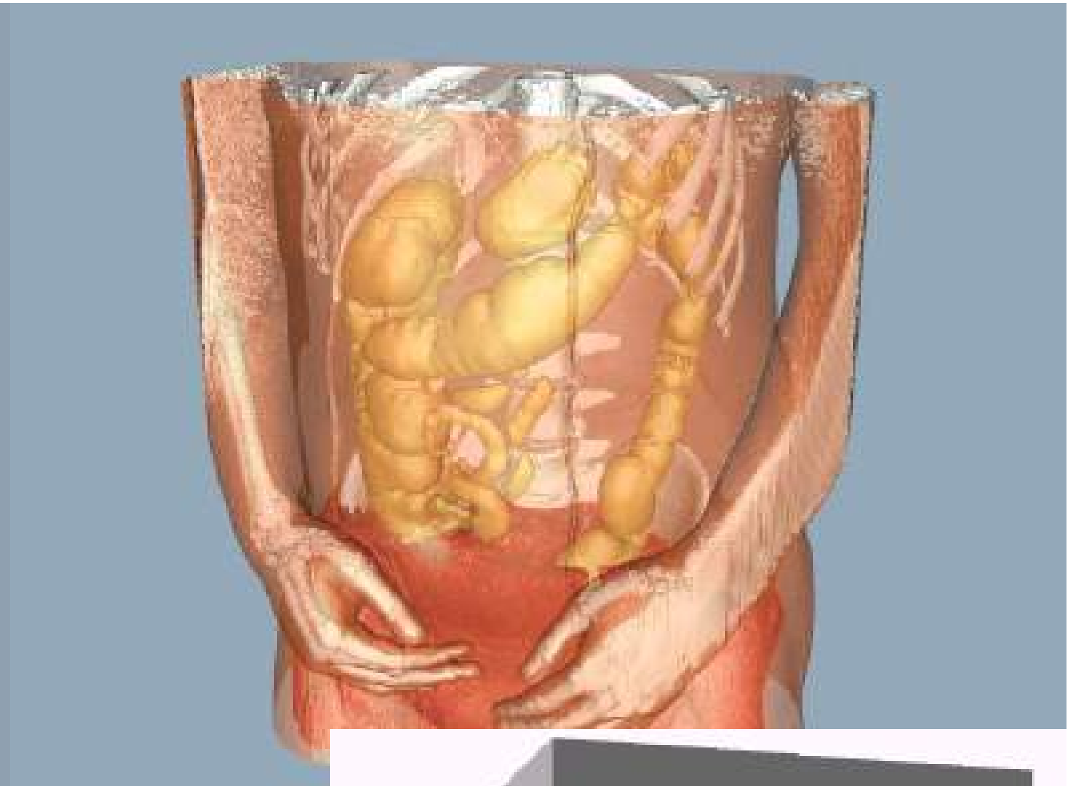
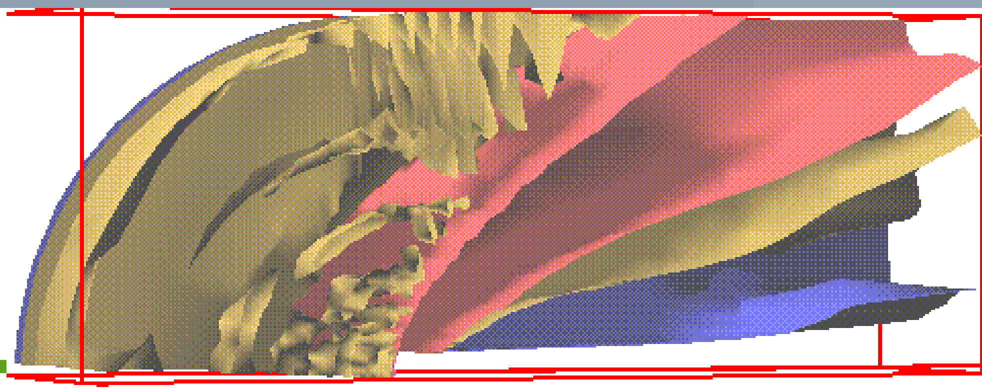
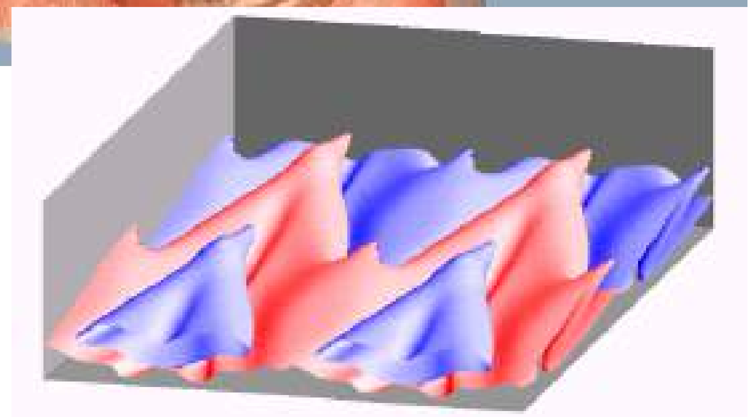
Summary

- 256 Cases
- Reduce to 15 cases by symmetry
- Ambiguity in cases 3, 6, 7, 10, 12, 13
- Causes holes if arbitrary choices are made
- Up to 5 triangles per cube



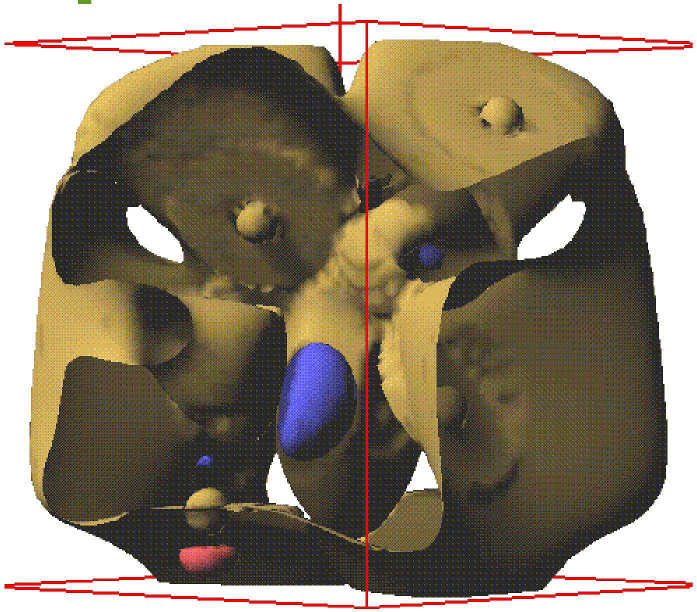
Robert S. Laramee
r.s.laramee@swansea.ac.uk

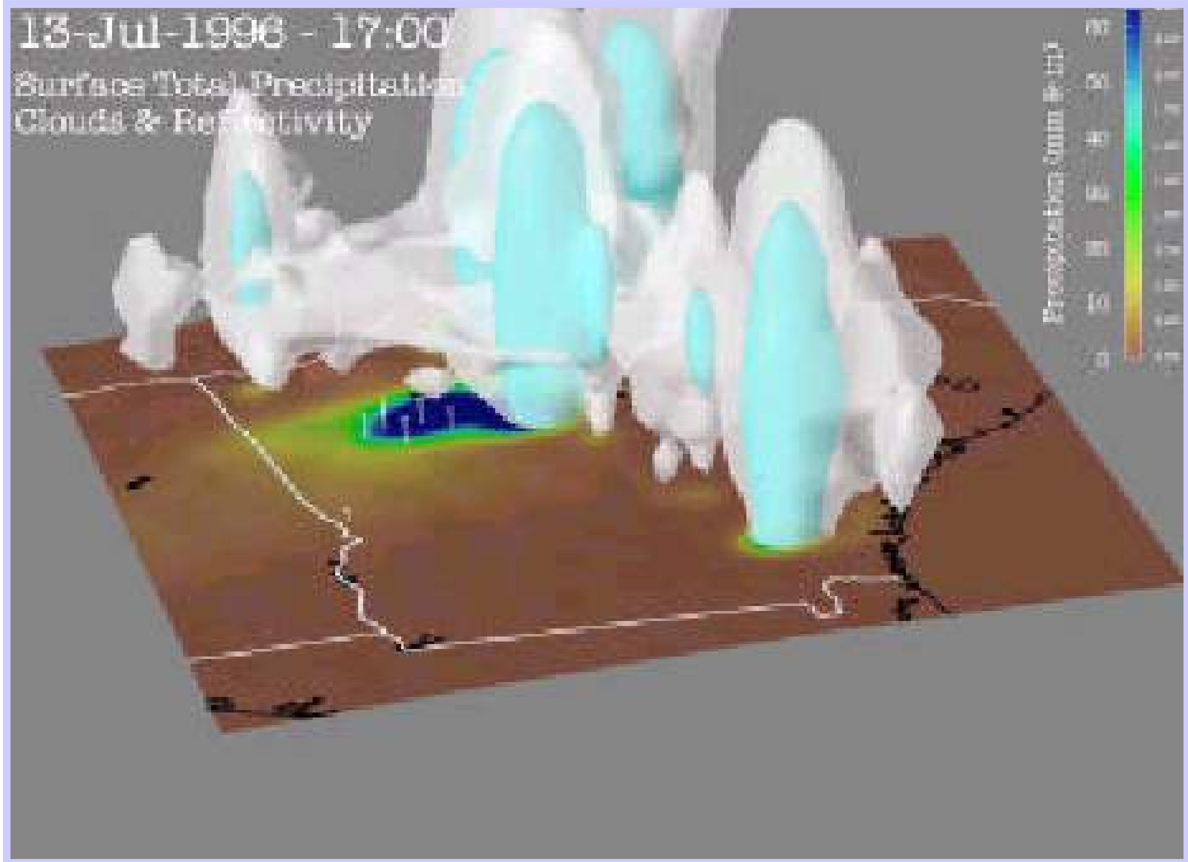
More Examples

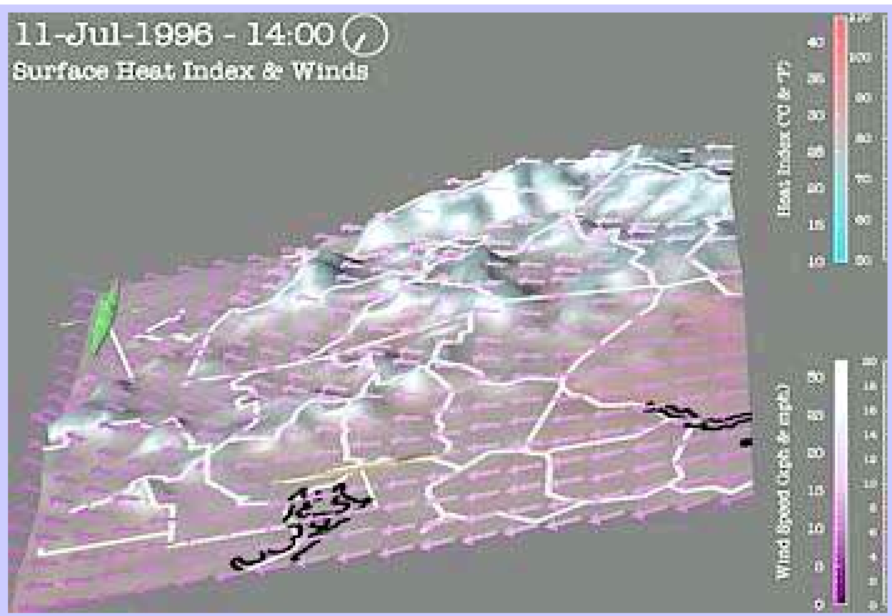





Robert S. Laramee
r.s.laramee@swansea.ac.uk

More Examples







Literature

Paper (more details):

- W. Lorensen & H. Cline: "**Marching Cubes: A High Resolution 3D Surface Construction Algorithm**" in *Proceedings of ACM SIGGRAPH '87 = Computer Graphics*, Vol. 21, No. 24, July 1987

Acknowledgments

For their contribution to the lecture material:

- Michael Meißner
- Roger Crawfis (Ohio State Univ.)
- Hanspeter Pfister
- Meister E. Gröller
- Torsten Möller
- Dirk Bartz
- Markus Hadwiger
- Helwig Hauser
- Daniel Weiskopf

Visualization, Lecture

Volume Visualization
Part 2 (of 4)
Direct Volume Visualization

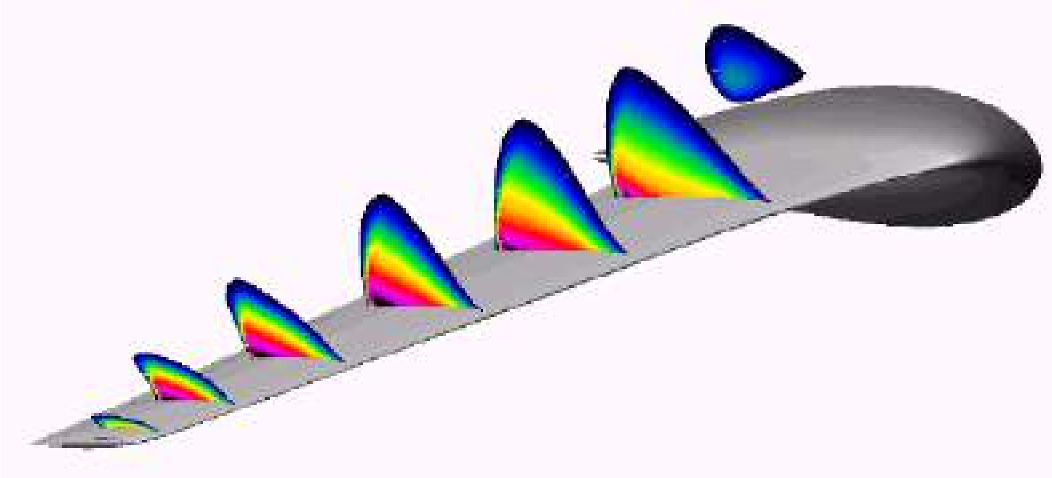
Robert S. Laramee
r.s.laramee@swansea.ac.uk

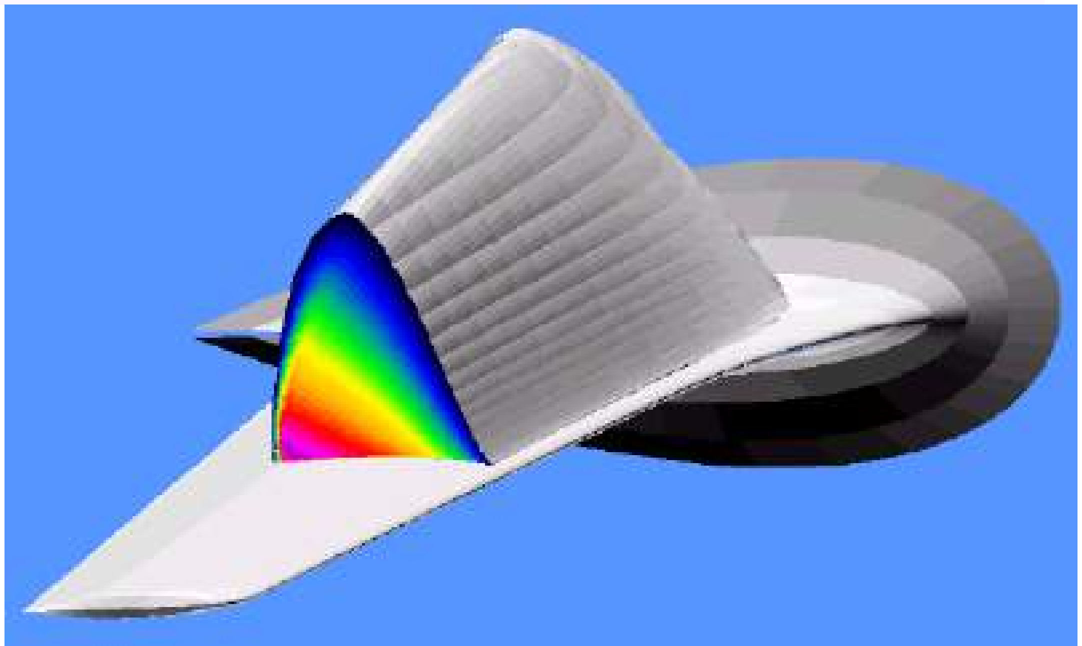
Recall: Lecture #4

Contents of Lecture #4:

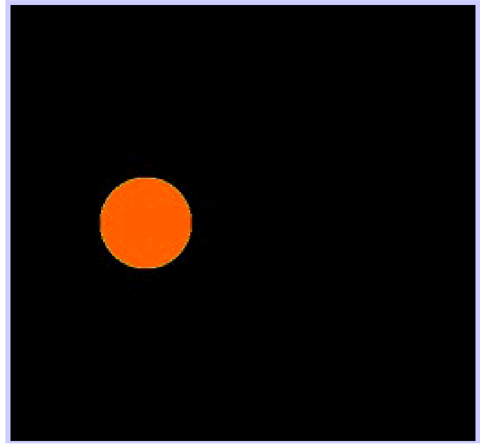
- isocontours
- isosurfaces
- the Marching Cubes (MC) algorithm

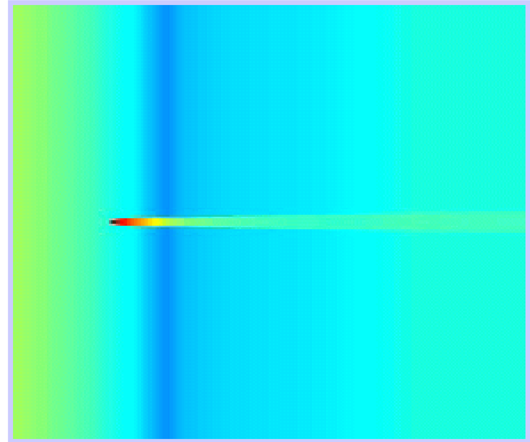
Review: Slices

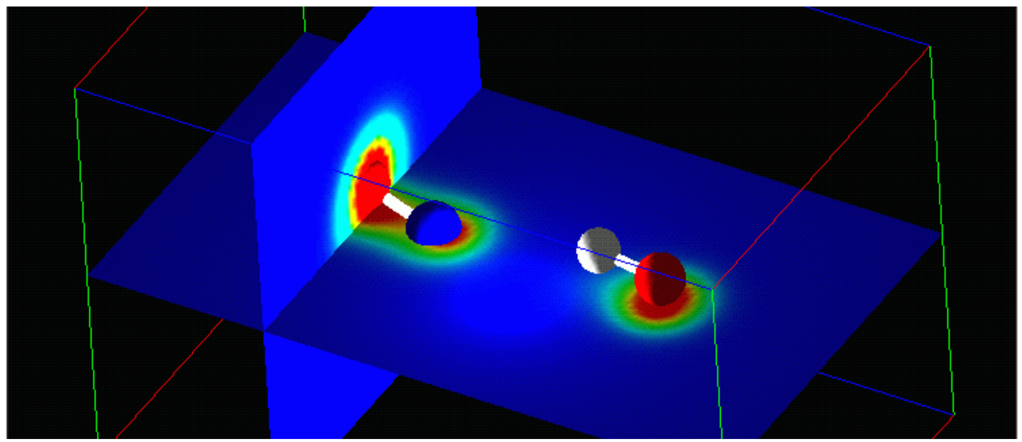




Scalar-clipping, combined with 3D







Overview: Lecture #5

Contents of Lecture #5:

- introduction to direct volume visualization
- ray functions
- classification
- transfer functions
- gradients
- ray casting
- compositing

Direct Volume Visualization

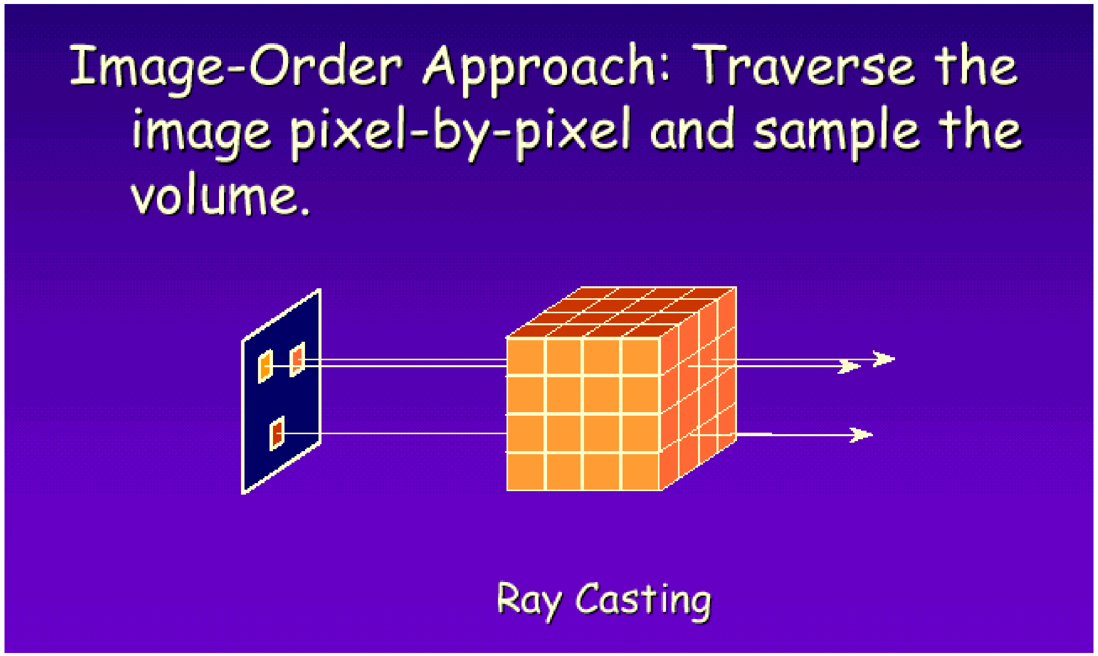
Overview:

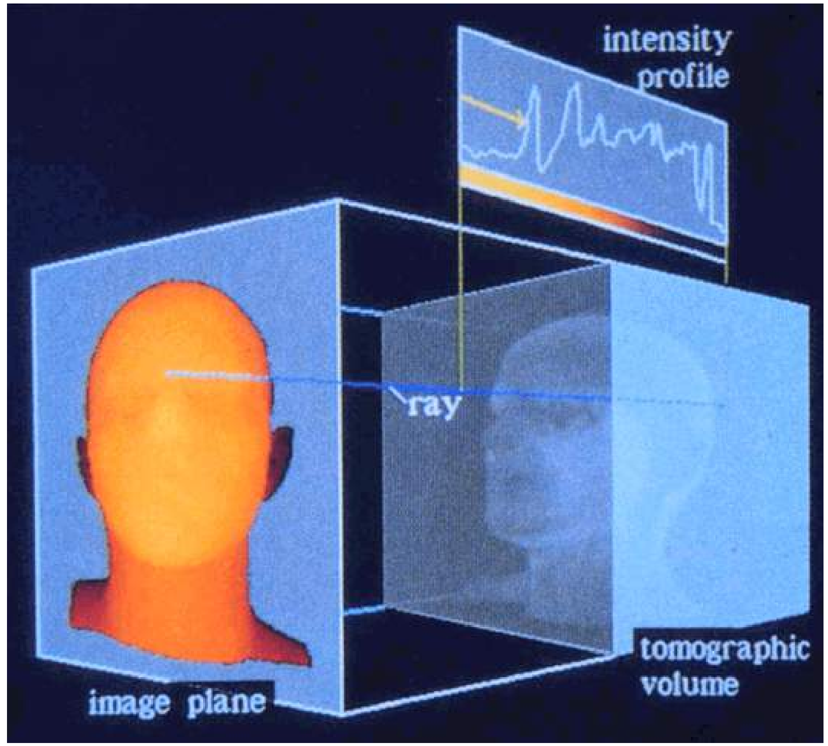
- No intermediate representation
- “Real 3D”
- Integration of so much information: difficult
- Object-order vs. image-order rendering
- Various techniques (ray casting, splatting, shear-warp, texture mapping, etc.)
- Various rendering techniques (compositing, MIP, first-hit, average, etc.)

Direct Volume Visualization

Terminology:

- **Ray Casting**: the value of each pixel in the image is determined by sending a ray through the pixel into the scene (image order volume rendering)





Robert S. Laramee
r.s.laramee@swansea.ac.uk

Computer Science
Gwyddor Cyfrifiadur

Rendering Techniques: Ray Functions

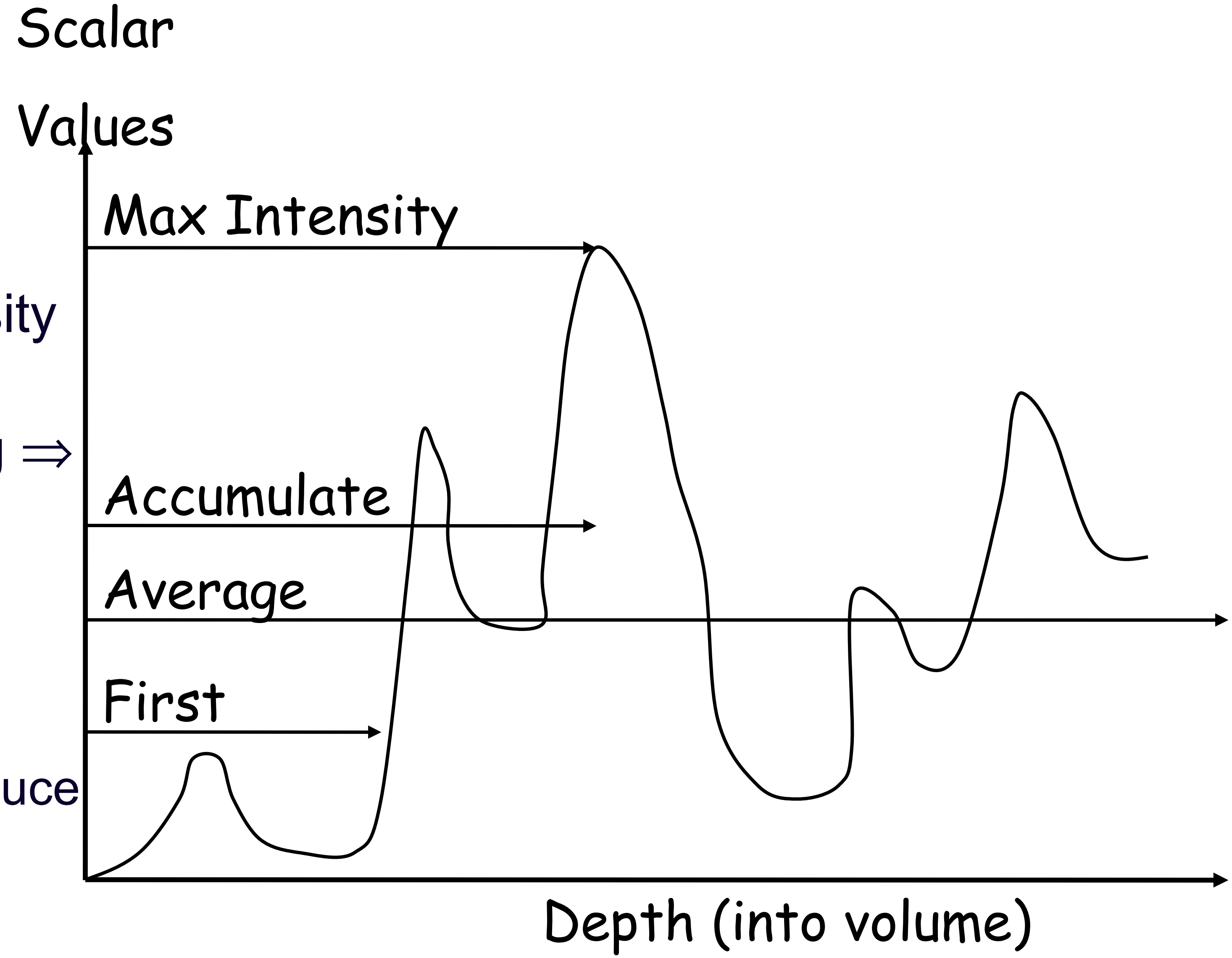
Ray Functions Overview:

- MIP ⇒

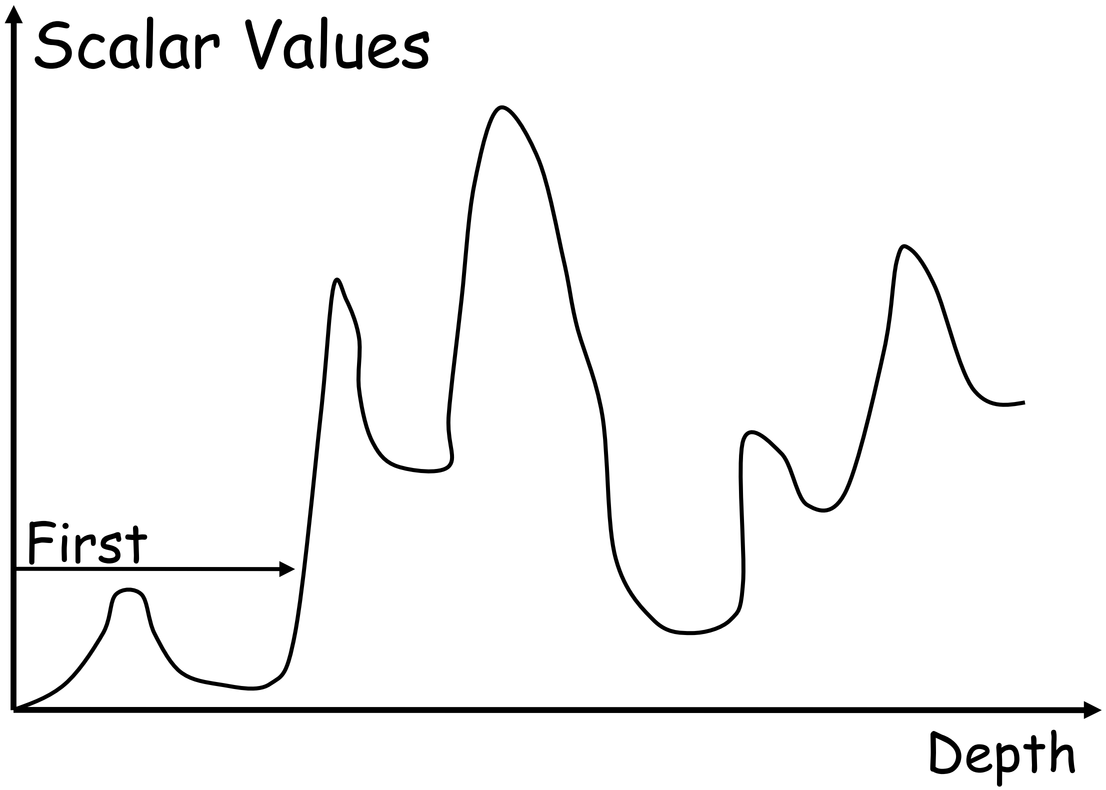
(**M**aximum **I**ntensity Projection)

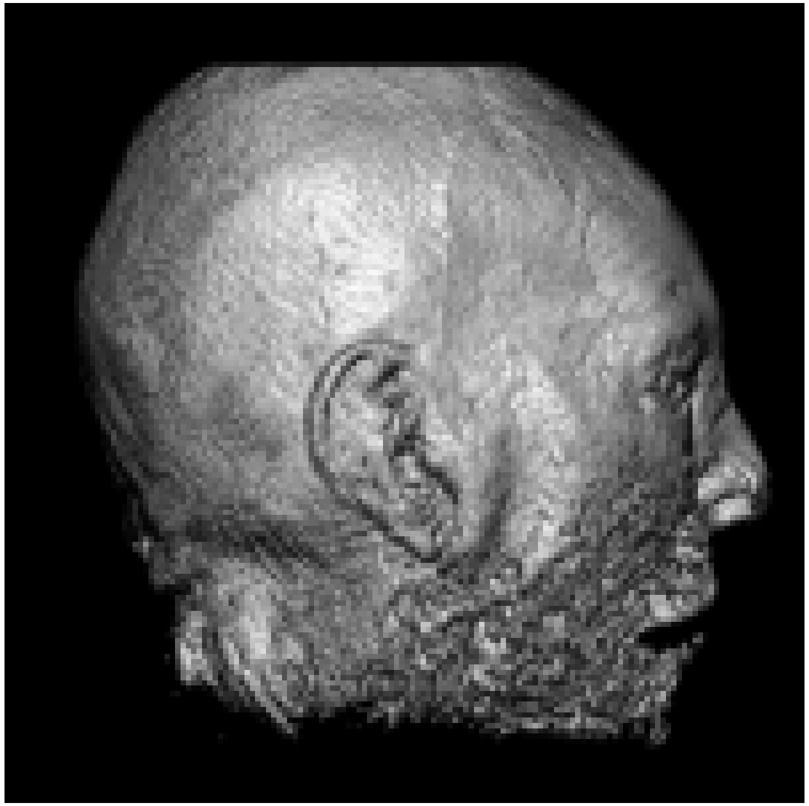
- Compositing ⇒
- X-Ray ⇒
- First Hit ⇒

Ray functions produce an ***intensity profile***.



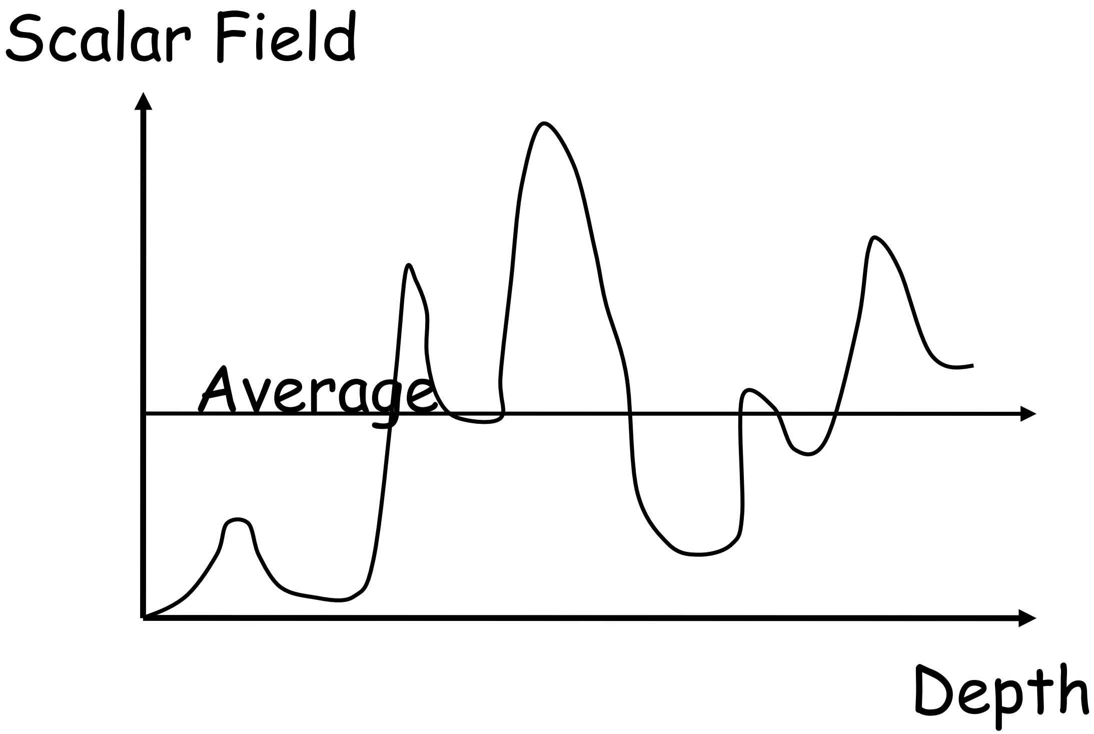
First Hit: Iso-surface Extraction





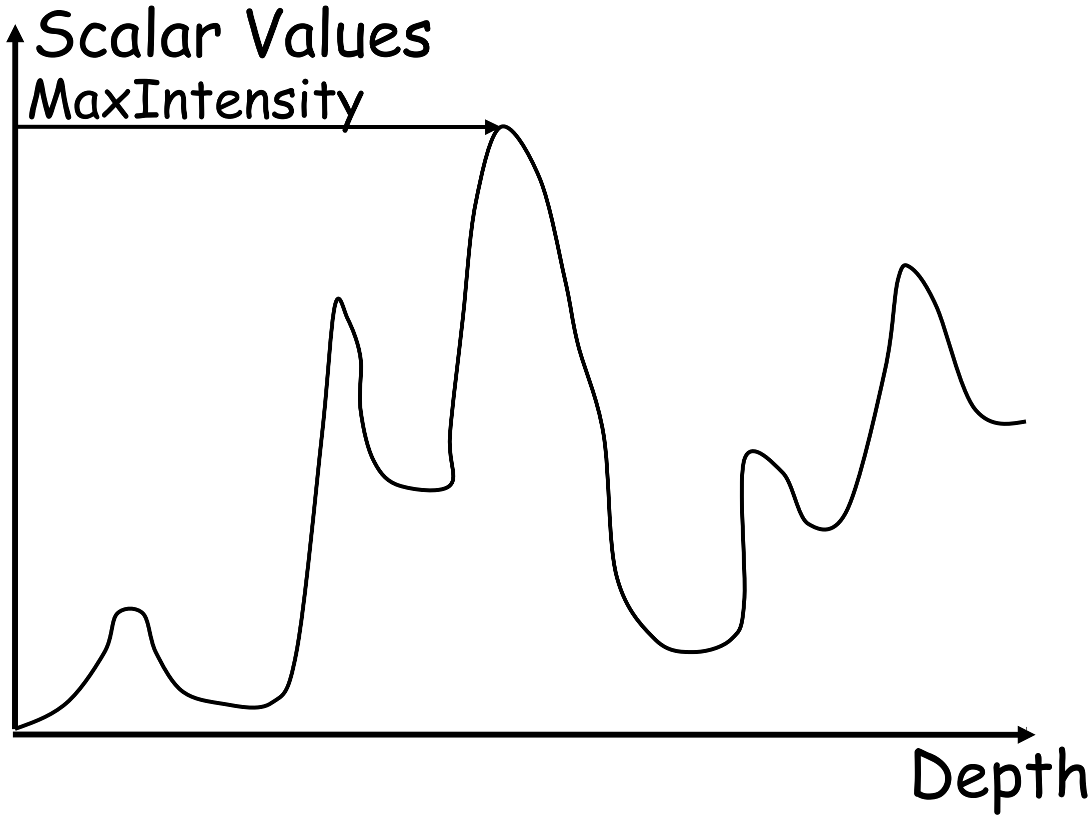
First Hit Ray Function: Extracts iso-surfaces (again), done by Tuy&Tuy '84

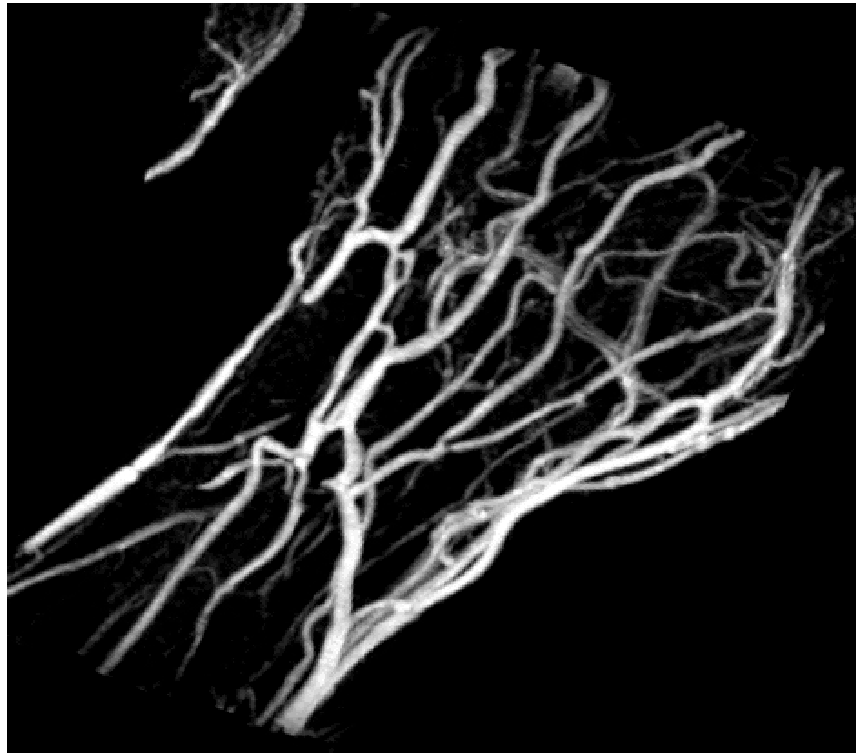
Average: Like X-Rays



Average Ray Function: Produces basically an X-ray picture

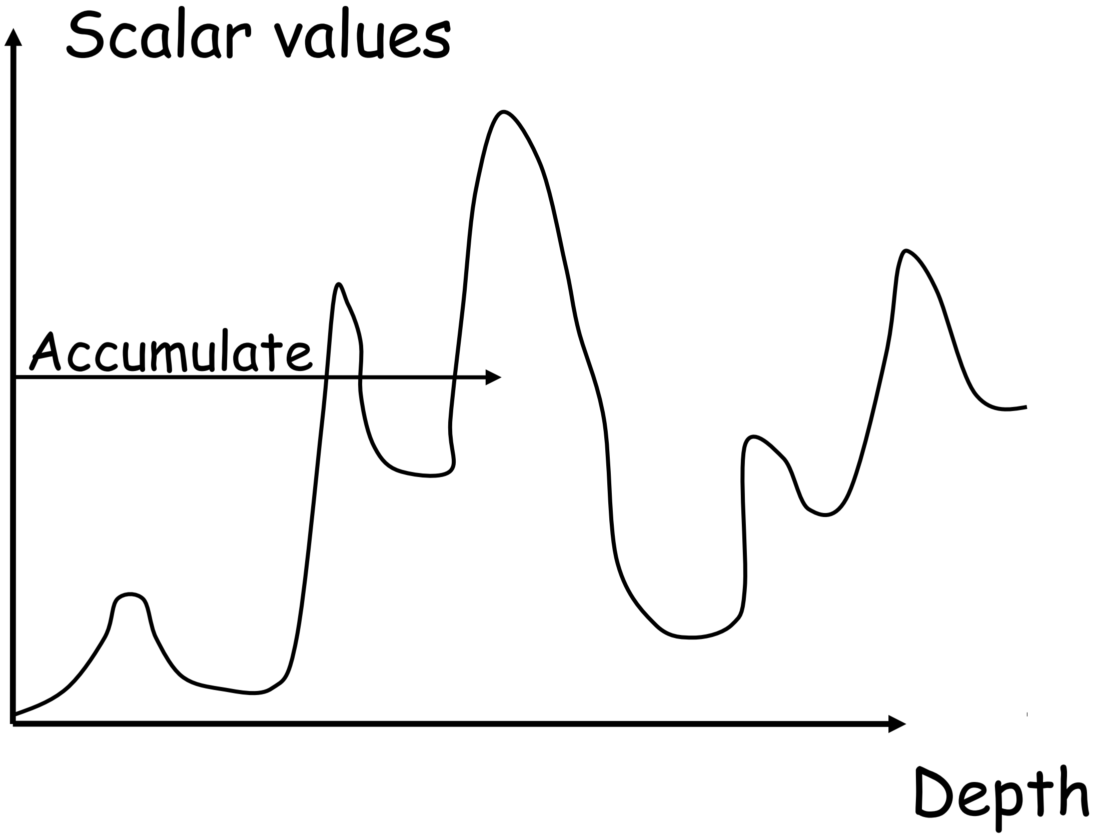
MIP: maximum-intensity projection

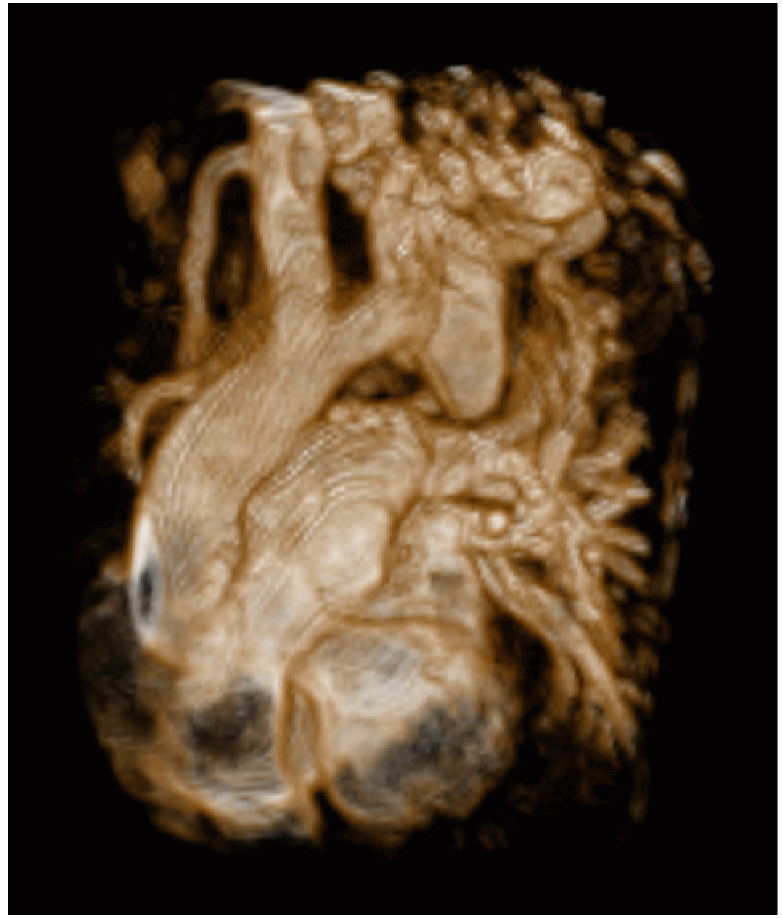




Max Ray Function: **M**aximum **I**ntensity **P**rojection (MIP) used for Magnetic Resonance Angiogram (MRA)

Compositing: Semi-transparency





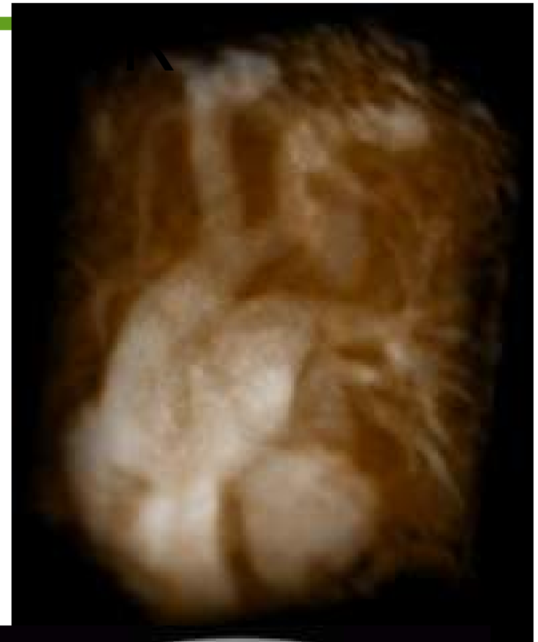
Accumulate: Make transparent layers visible.

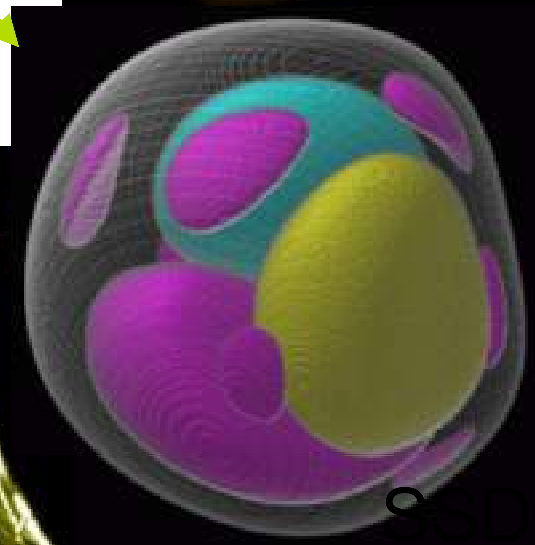
Levoy ‘88

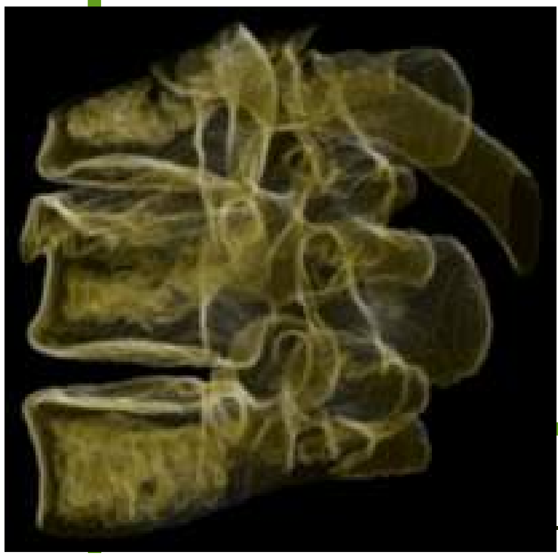
Combining Ray Functions

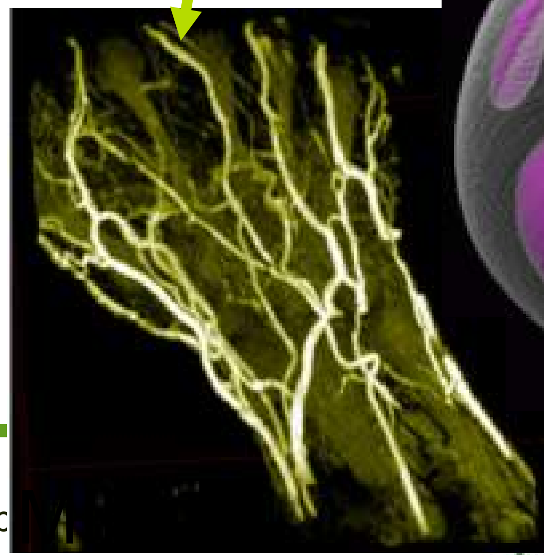
Some Possibilities:

- α-compositing
- shaded surface display
- maximum-intensity projection
- x-ray simulation
- contour rendering









Classification

Identifying the Data $\Rightarrow$ Semantic:

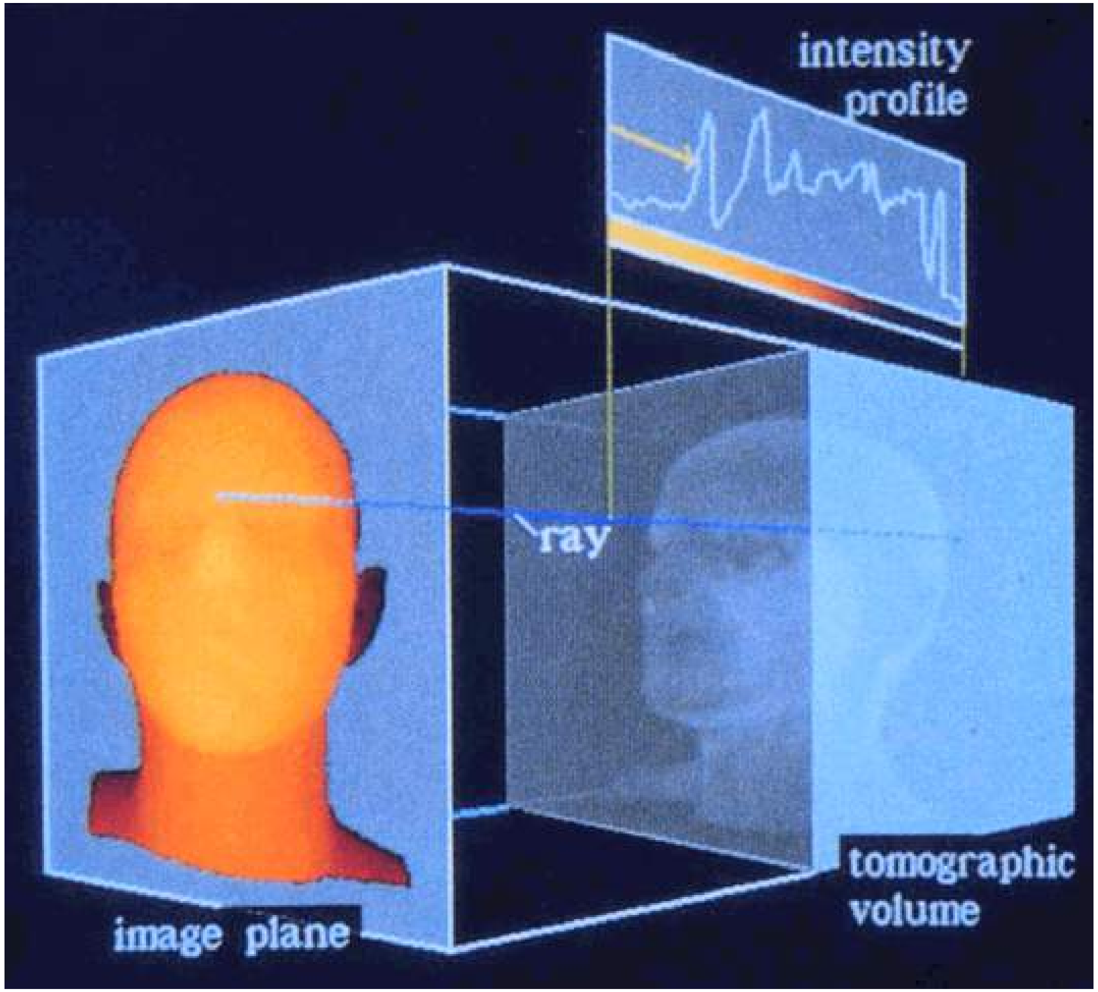
- ordering the objects, e.g.
  bones, skin, muscles, etc.
- the use of data values,
  gradient information
- goal: segmentation
- often: semi-automatic, i.e., manual
- automatic approximation:
  transfer functions
- Huge topic in image processing

Transfer Functions

Terminology

transfer function: an operation that maps scalar values in 3D to color, opacity and/or material values.

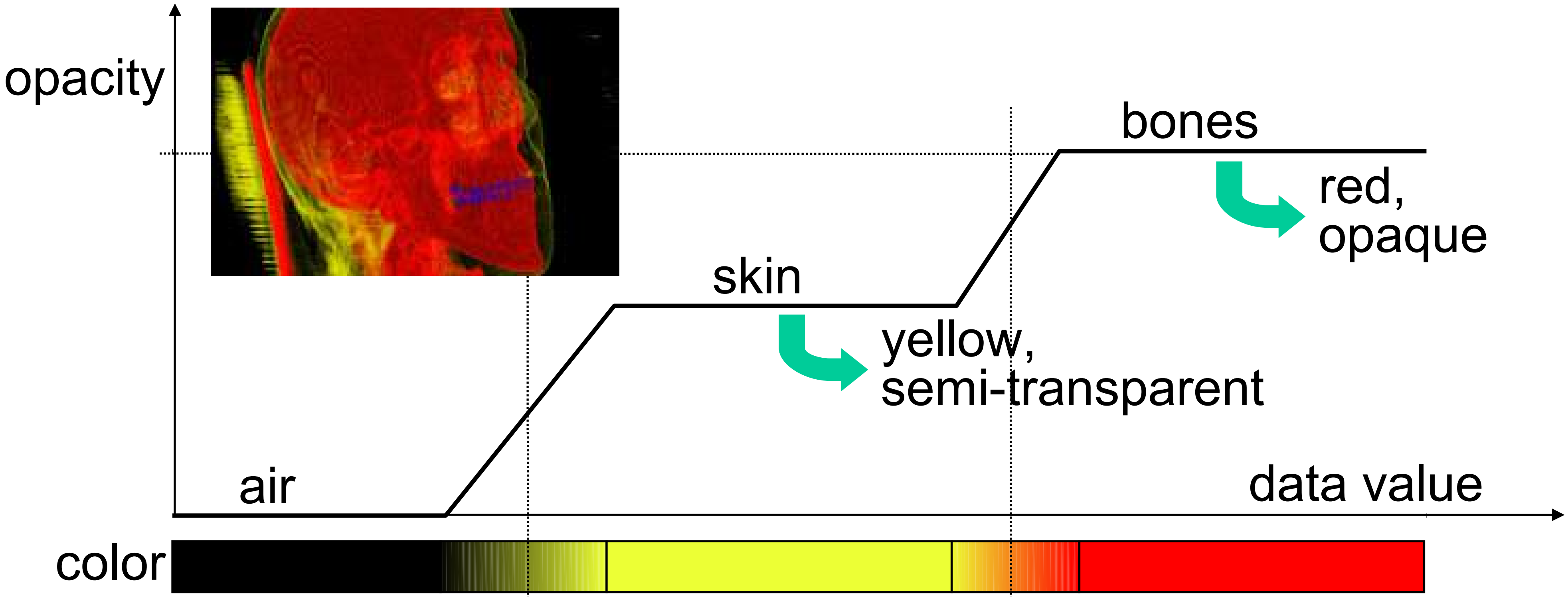
–bob's definition



Transfer Functions

Image Data→Presentable:

- 1.) data→color
- 2.) data→opacity (opposite of transparency)

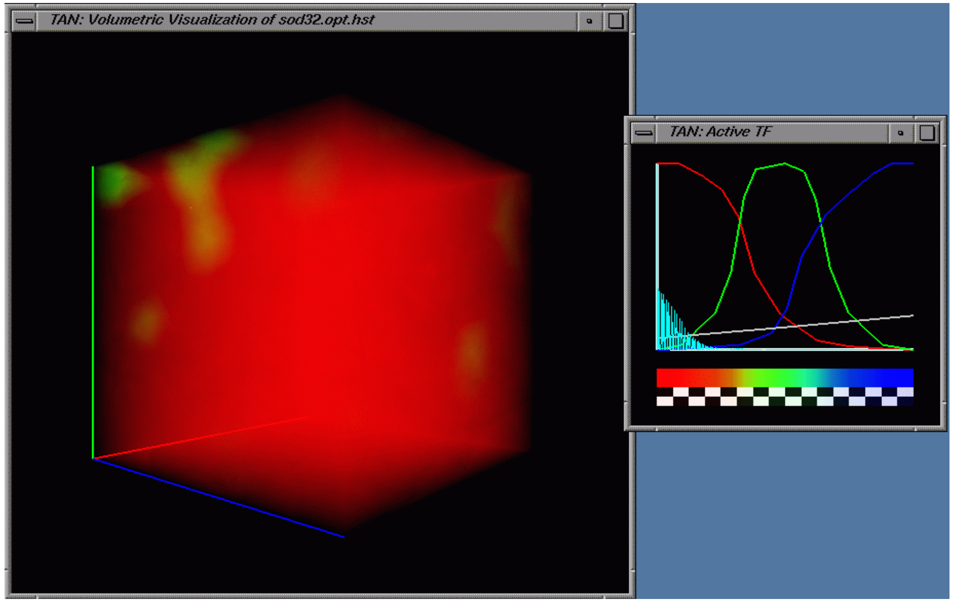


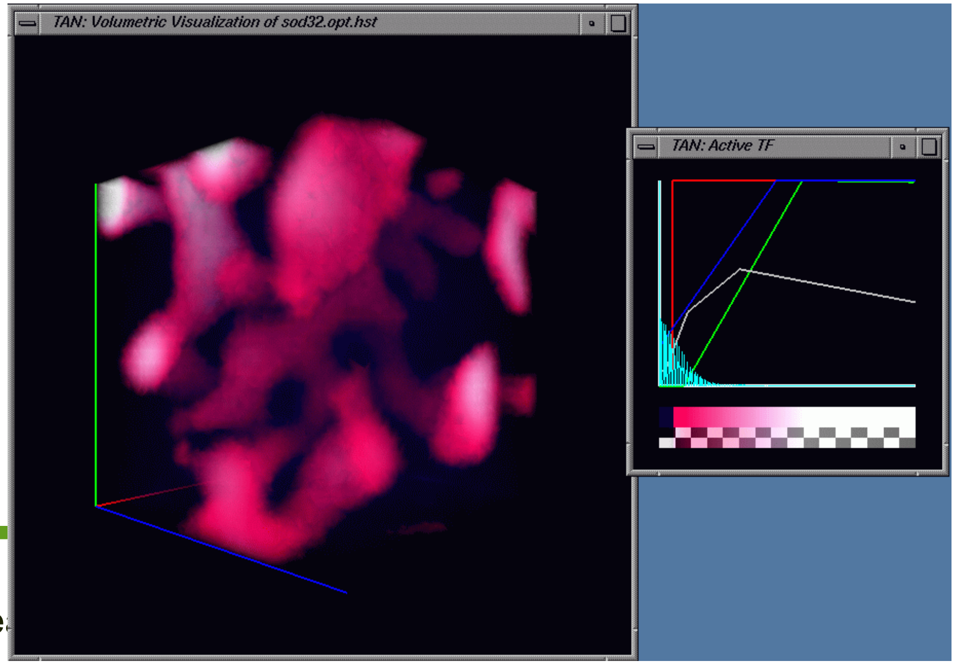
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Different Transfer Functions

Results:

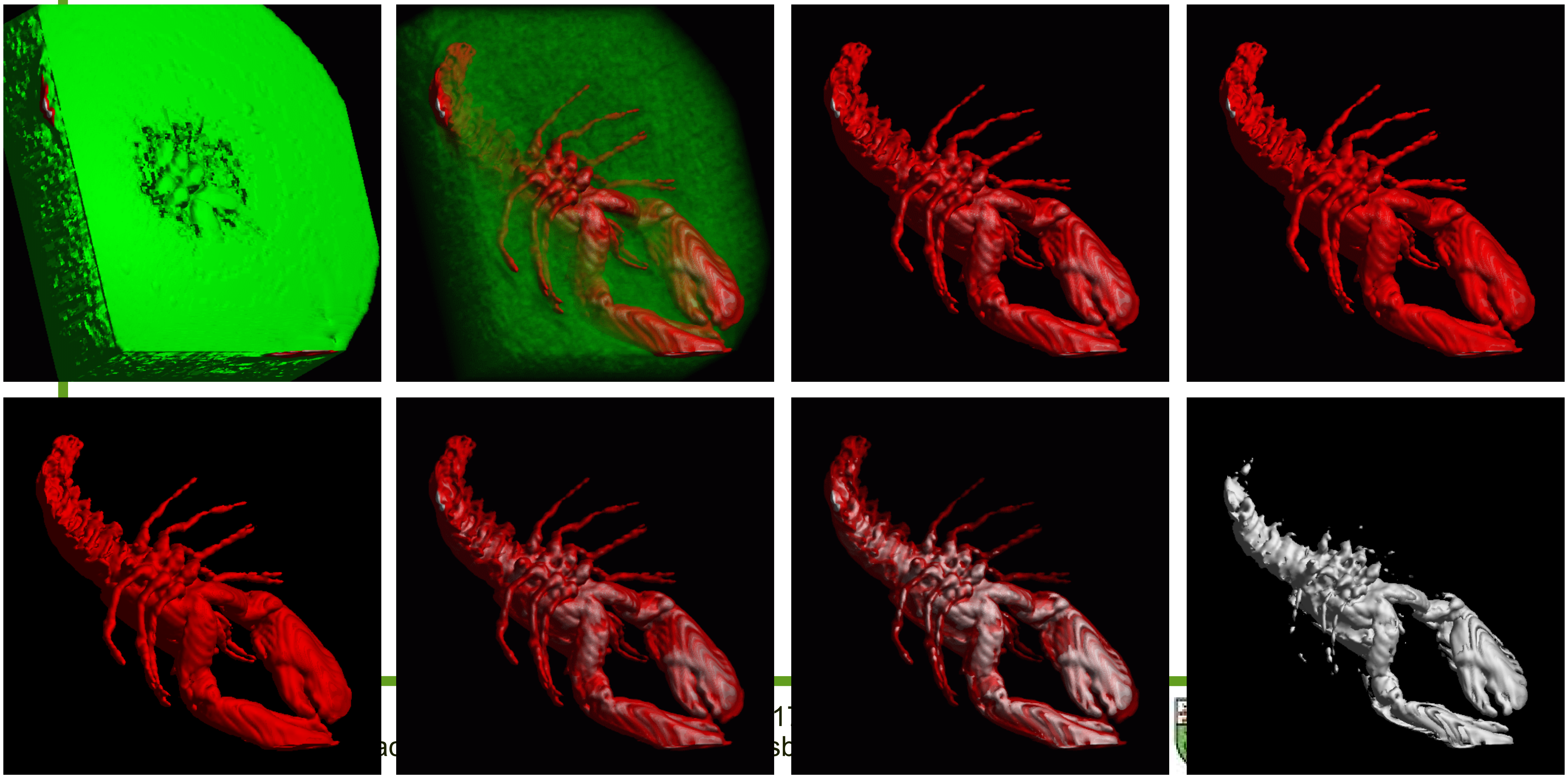
- depend strongly on the transfer function
- non-trivial setting/adjustment
- segmentation only conditionally necessary



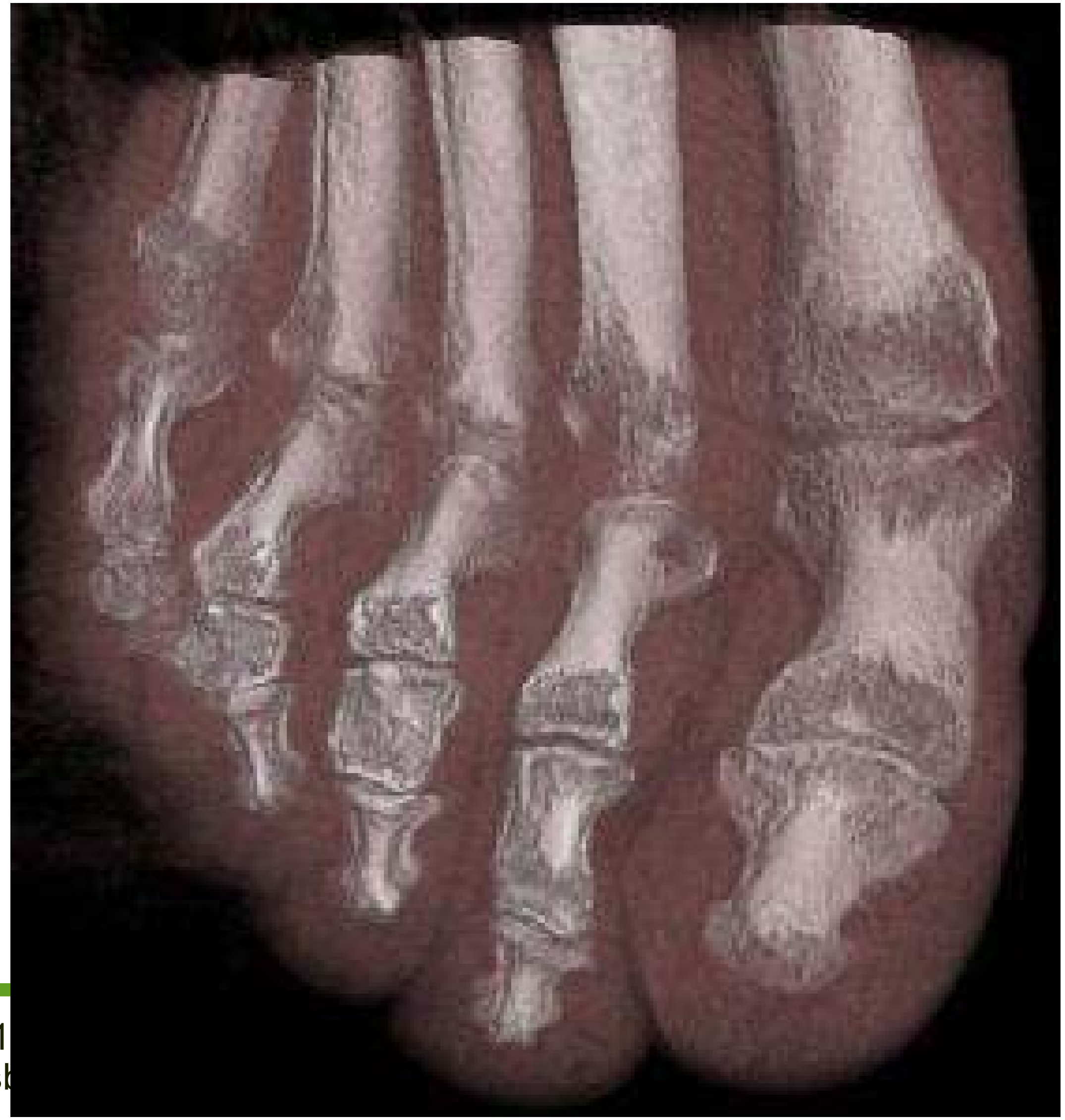


Lobster – Various Transfer Functions

three objects: surrounding, shell, meat



Foot data – 1 i.e. 2 Transfer Function-Components

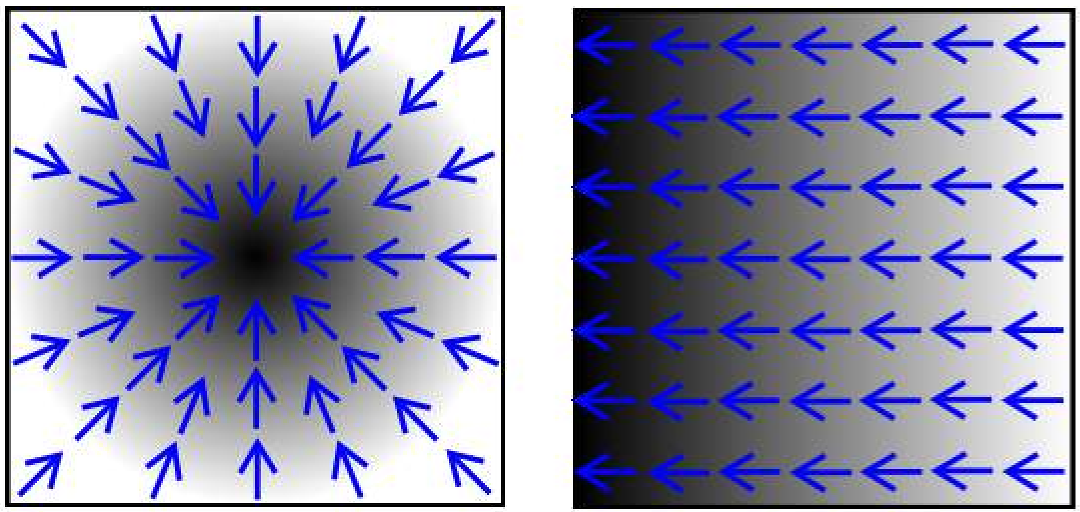


Gradients

Terminology

- **Gradient:** 1. the degree of inclination, or the rate of ascent or descent, in a highway, railroad, etc.
- 2. an inclined surface; grade; ramp.
- 3. *Physics*. a. the rate of change with respect to distance of a variable quantity, as temperature or pressure, in the direction of maximum change. b. a curve representing such a rate of change.
- 4. *Mathematics*. a differential operator that, operating upon a function of several variables, results in a vector the coordinates of which are the partial derivatives of the function.

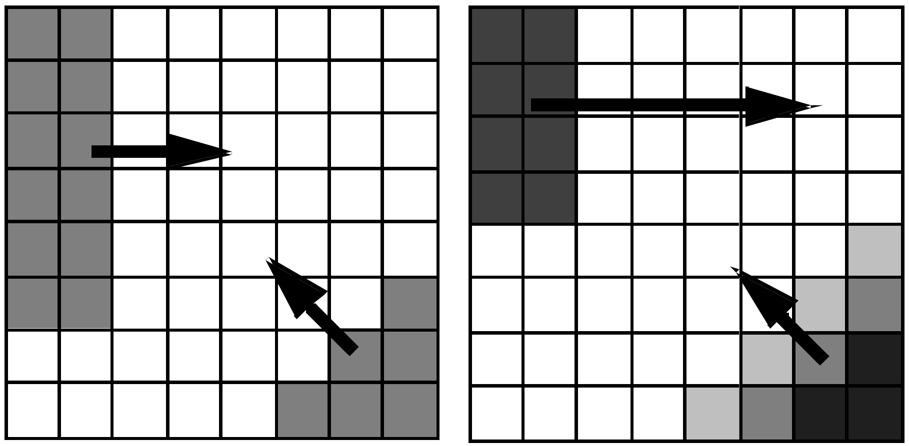
-www.dictionary.com



Including Gradients

Emphasizing transitions (or boundaries):

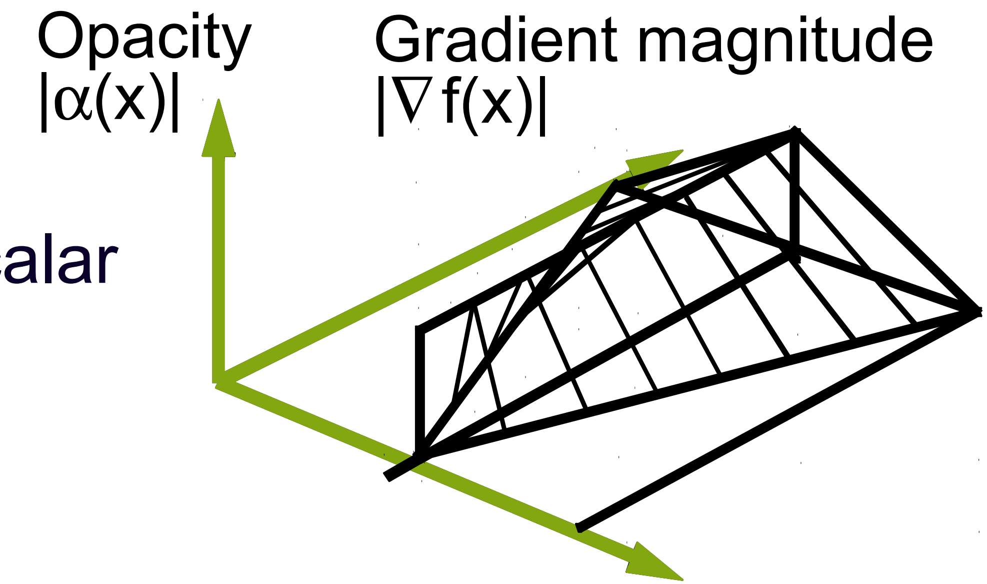
- special interest often in transition areas
- gradients: measure changes (like surface normals)
- larger gradient $\Rightarrow$ more opacity



Gradient Dependent Transfer Functions

2D-Transfer Function:

- Levoy '88
- Opacity is a function of scalar value *and* local gradient
- The greater the gradient magnitude, the higher the opacity
- highlights transitions (large gradients)
- dampens uniform areas (more transparency)



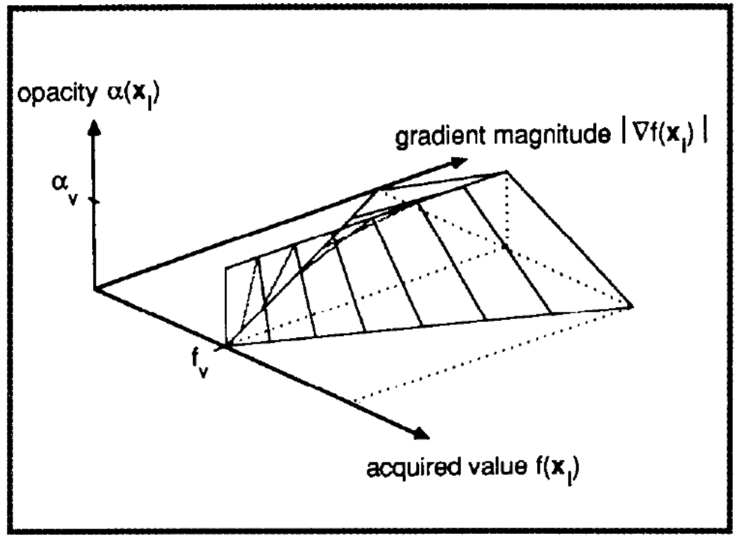


Figure 3. Calculation of opacities for isovalue contour surfaces.

Ray Casting / Compositing Detail

Classical
Image-Order Method

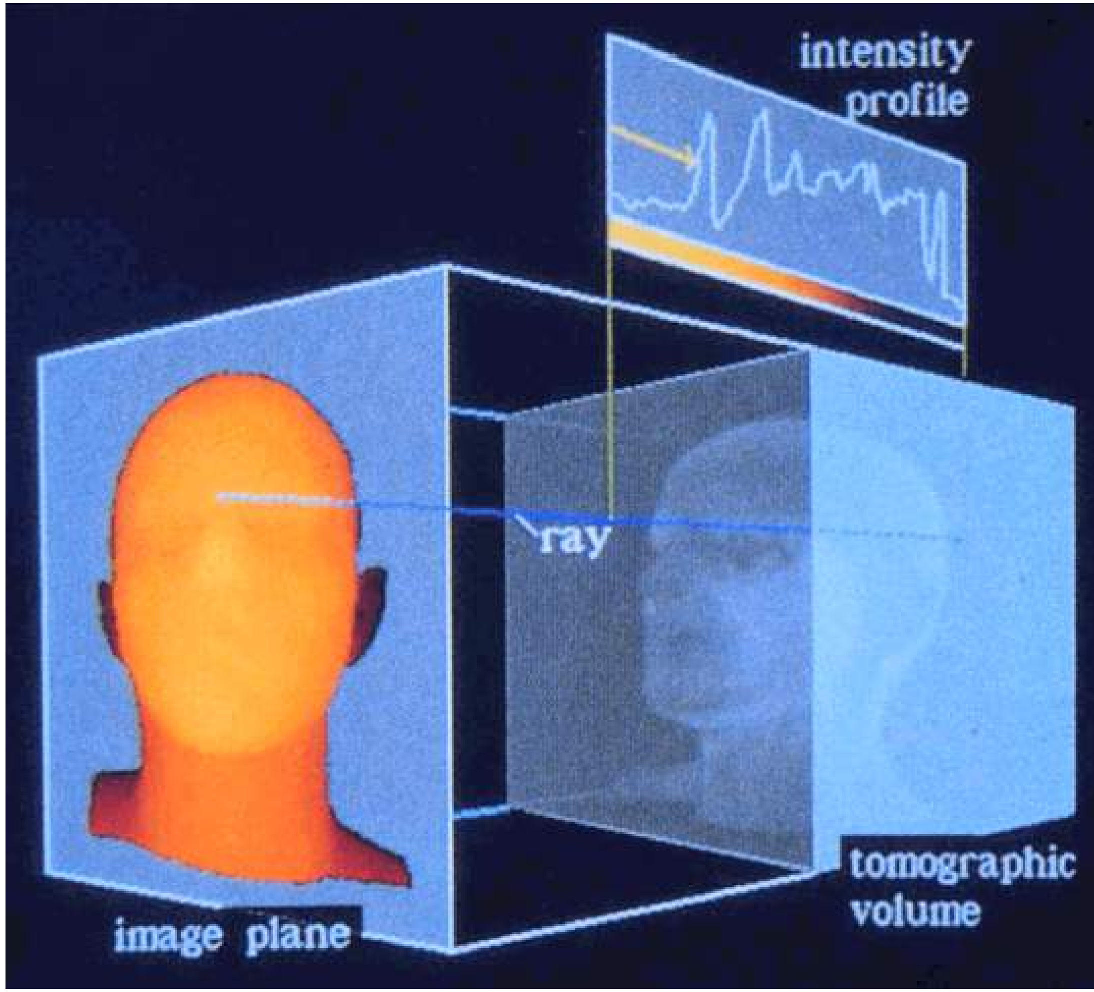
Ray Tracing vs. Ray Casting

- Ray Tracing: image generation method
- In volume rendering: only primary light scattering ⇒ thus Ray Casting
- Classical image-order method
- Ray tracing: radiate object surface or slices
  Ray casting: no objects, just scalar (e.g. density) values in 3D
- Theory: consider all density values
  Practice: traverse volume slice by slice
- Interpolation necessary per slice

Ray Through Volume Data

Overview:

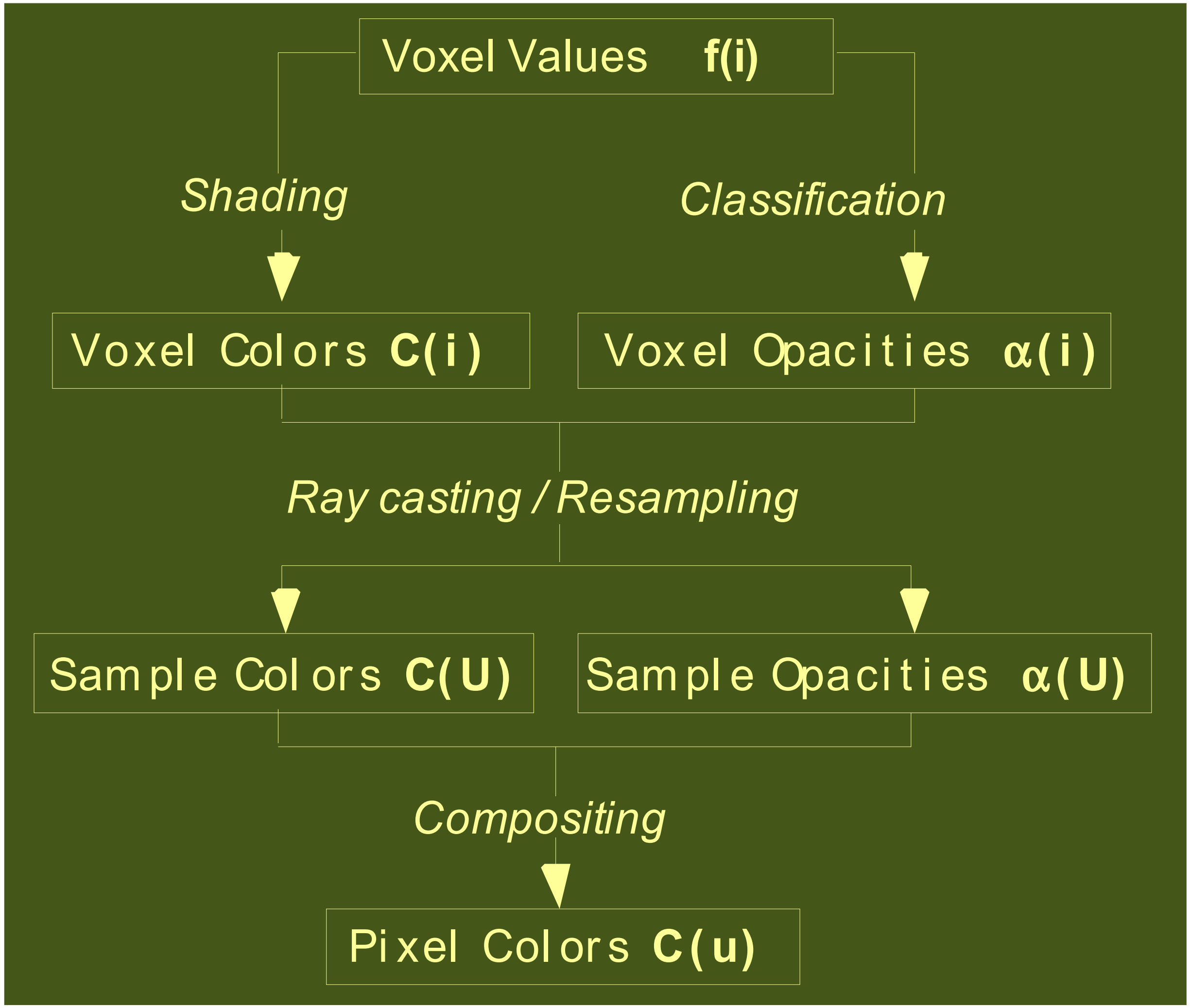
- Volume Data: 1D values defined in 3D $f(\mathbf{x}) \in R^1$, $\mathbf{x} \in R^3$
- Ray defined as half of a line: $\mathbf{r}(t) \in R^3$, $t \in R^1 > 0$
- Values along Ray: $f(\mathbf{r}(t)) \in R^1$, $t \in R^1 > 0$ (intensity profile)



Standard Ray Casting

Levoy '88:

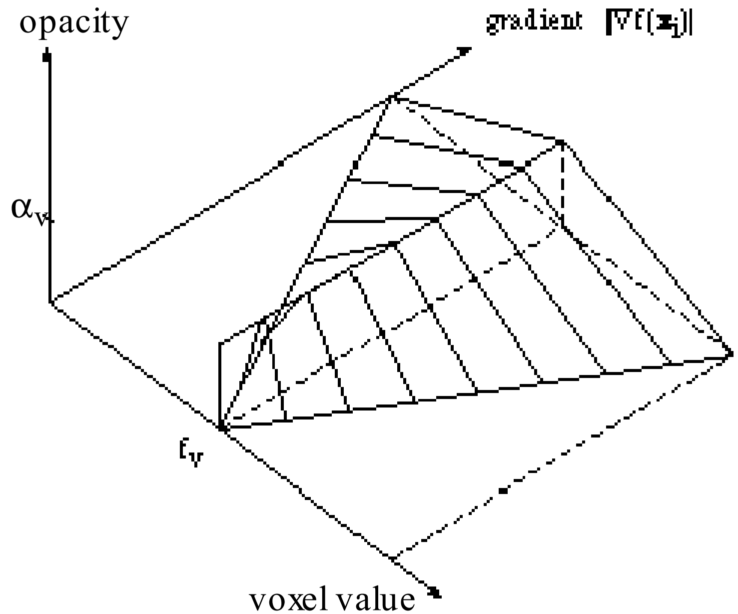
1. C(i), α(i) (using transfer function)
2. Ray casting, interpolation
3. Compositing



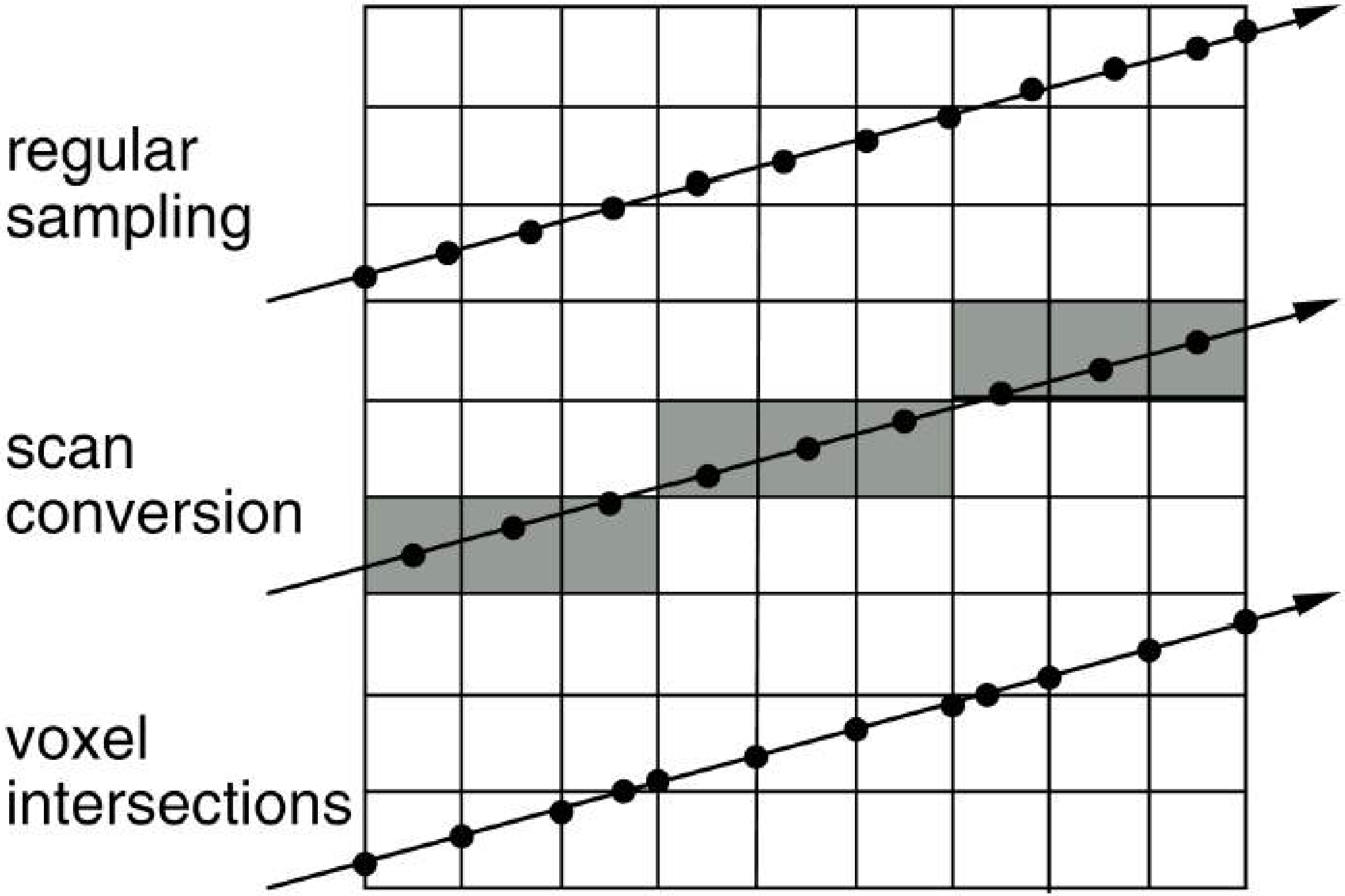
1. Shading and Classification of Voxel Values

Step 1.:

- voxel shading, $f(i) \rightarrow C(i)$:
  - according to transfer function
  - diffuse shading (Phong), gradient $\approx$ normals
- voxel classification, $f(i) \rightarrow \alpha(i)$:
  - Levoy '88, according to gradients
  - emphasizes transitions/boundaries



2. Ray traversal – Three Approaches



2. Ray Traversal, Interpolation

- Voxel-based vs. cell-based ray traversal
- Tri-linear (interpolotion within the cell) vs. Bi-linear (interpolation along cell edges)
- Tri-linear:
  - first 4* in z-Direction (Quadratic interpolation),
  - then 2* in y-Direction (Linear interpolation),
  - then 1* in x-Direction (value interpolation)

3. Compositing

Terminology

compositing: the sample-by-sample accumulation of color and opacity values along a ray as it traverses volume data. –bob

kernel: (a.k.a. convolution kernel a.k.a. convolution filter) “one or two dimensional images that are used for computing the weighted average of pixel images” –The OpenGL Programming Guide

3. Compositing: F2B vs. B2F

Back-to-Front (B2F):

- $c = c(1 - \alpha(i\Delta s)) + C(i\Delta s)$
- $\alpha = \alpha(1 - \alpha(i\Delta s)) + \alpha(i\Delta s)$

Front-to-Back (F2B):

- $c = C(i\Delta s)\ \alpha\ (i\Delta s)\ (1 - \alpha) + c$
- $\alpha = \alpha(i\Delta s)(1 - \alpha) + \alpha$

c = current color

α = opacity (inverse of transparency)

i = sample index

s = sample

Δs = distance between samples

C = color at sample

Front-to-back Compositing

Demo(s)

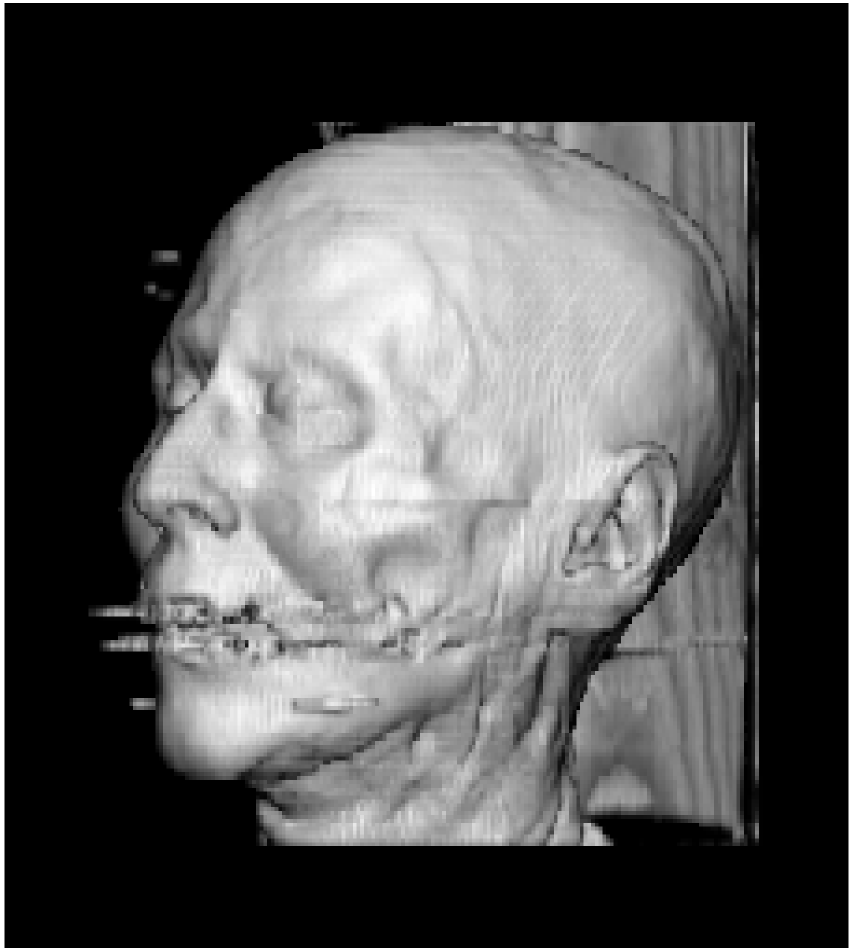
Compositing: F2B vs. B2F

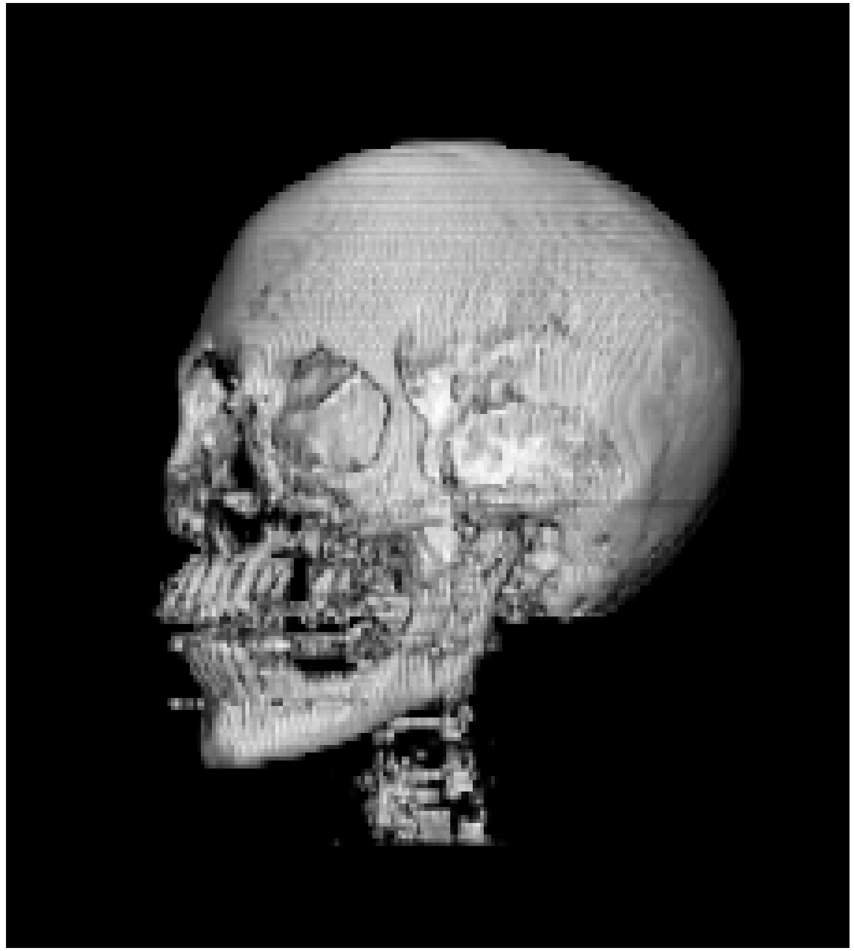
F2B: a ray can be stopped once opacity approaches 1.0 -early ray termination.

B2F: a generalization of the Painter's algorithm –less frequently used.

Ray Casting – Examples

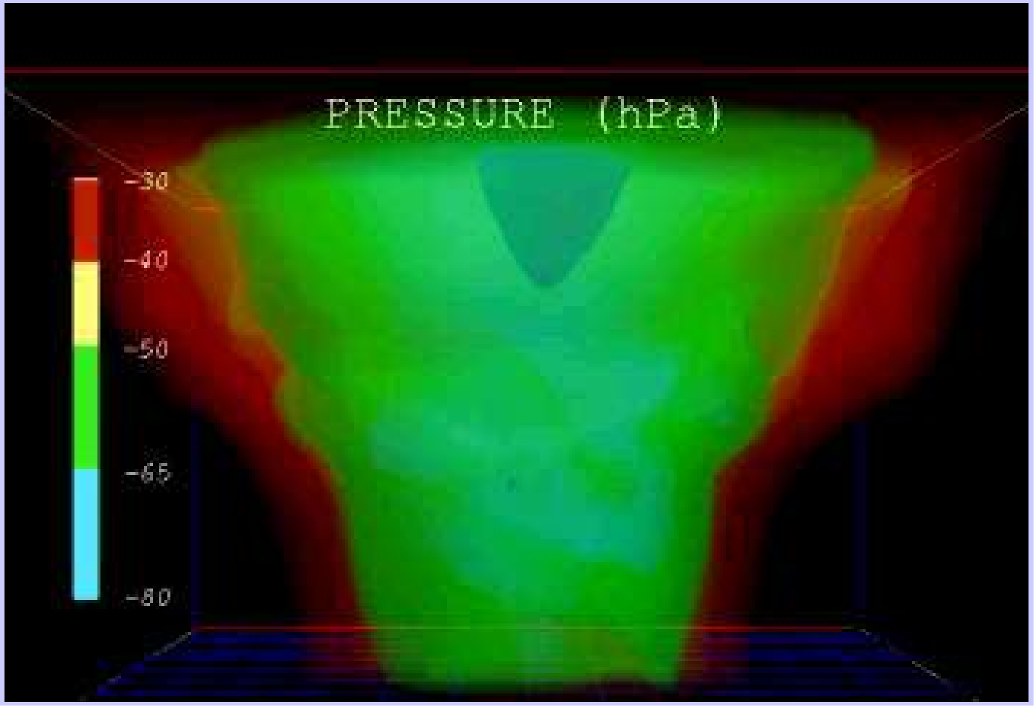
Different transfer functions (quasi-surface rend.), $256^2 \times 113$ CT data

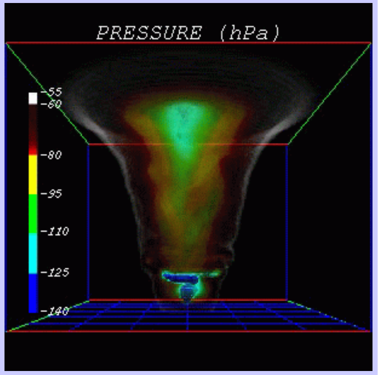




Ray Casting – More Examples

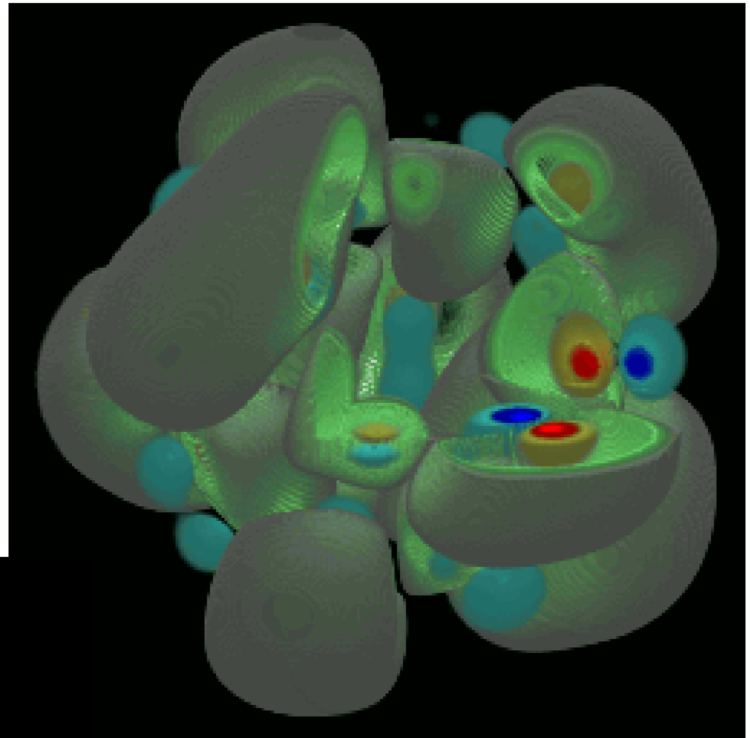
- Tornado Viz:
- Head data:

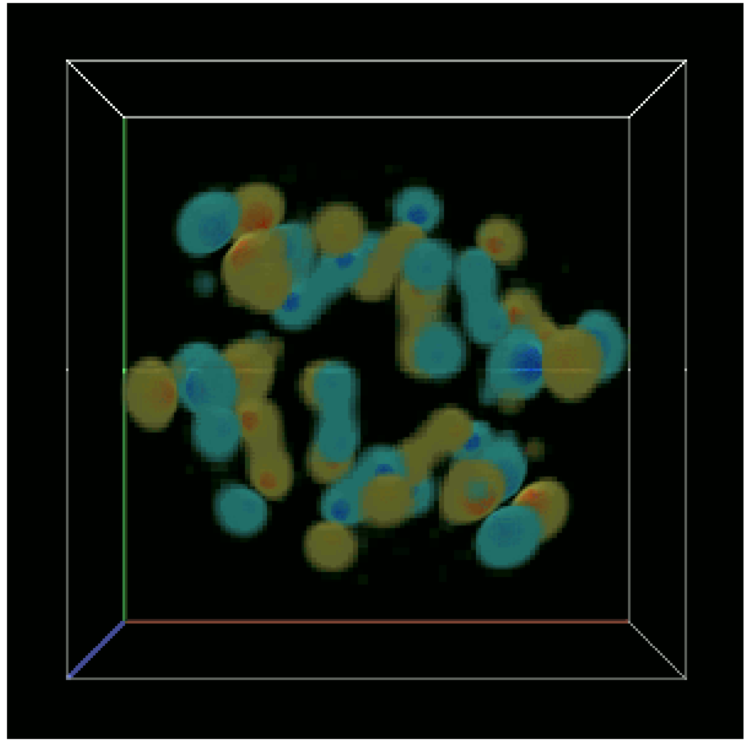


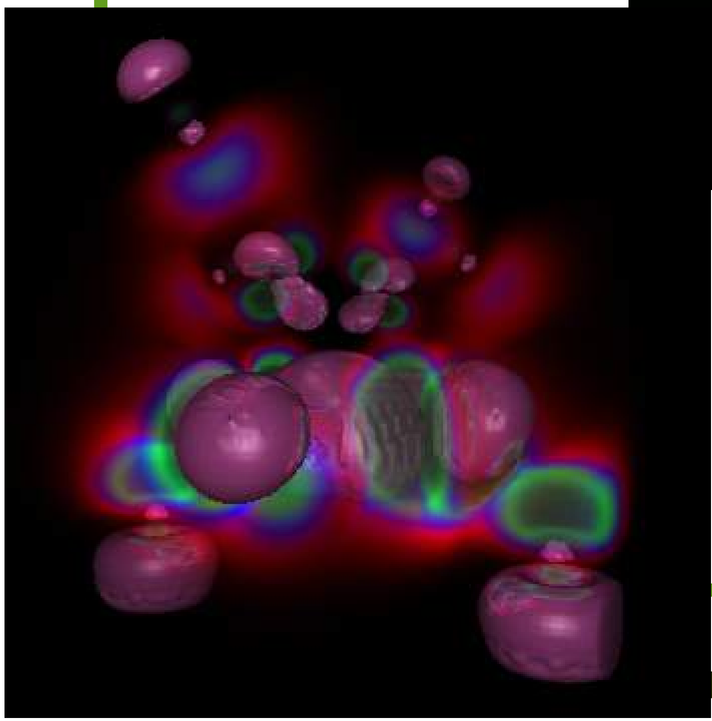


Ray Casting – more Examples

Molecular data:







Literature

Paper (more details):

- Marc Levoy: "**Display of Surfaces from Volume Data**" in *IEEE Computer Graphics & Applications*, Vol. 8, No. 3, June 1988

For more, see also

- **Data Visualization, Principles and Practice**, **Chapter 10 Volume Visualization**, by A. Telea, AK Peters, 2008

Acknowledgements

- We thank:
  - Nelson Max (LLNL), Marc Levoy (Stanford)
  - Hans-Georg Pagendarm (DLR, Göttingen)
  - Lloyd Treinish (IBM)
  - Roberto Scopigno,
Claudio Montani (CNR, Pisa)
  - Roger Crawfis (Ohio State Univ.)
  - Michael Meißner (GRIS, Tübingen)
  - M. Eduard Gröller
  - Torsten Möller
  - Roberto Scopigno,
Claudio Montani (CNR, Pisa)
  - Hans-Georg Pagendarm (DLR, Göttingen)
  - Michael Meißner (GRIS, Tübingen)
  - Torsten Möller
  - M. Eduard Gröller
  - Helwig Hauser

Visualization, Lecture

Volume Visualization, Shear-Warp Factorization

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Review: Previous Lecture

Contents of Previous Lecture = VolVis, Splatting:

- Filtering
  - review
  - interpolation, gradients
- Splatting
  - basics
  - implementation

Overview: Lecture

Contents of Lecture = VolVis, Shear-Warp:

- Shear-warp factorization
  - the original algorithm
  - optimizations
- Overview and Comparison of Volume Rendering Techniques
  - Ray Casting, Shear-Warp, Texture Mapping, Splatting

Shear-warp factorization

Fast, hybrid

image-object-order rendering

Shear-warp Factorization

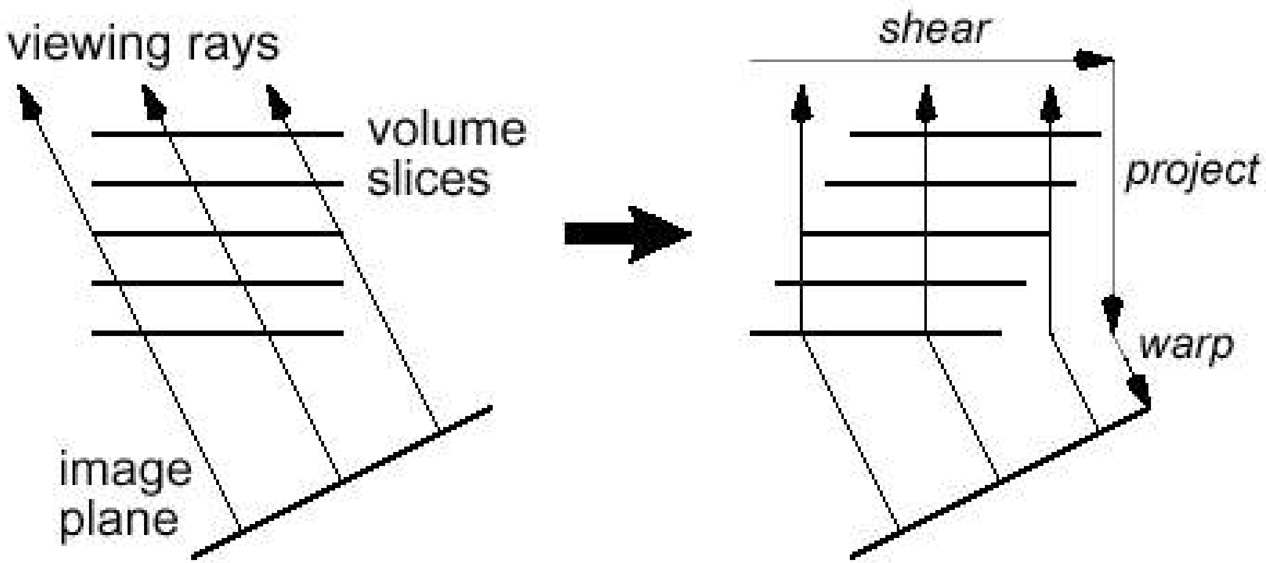
Factorizing the viewing transformation:

- $M_{view} = P \cdot S \cdot M_{warp}$

(**P** = permutation matrix for coordinate system, z axis principle viewing axis,

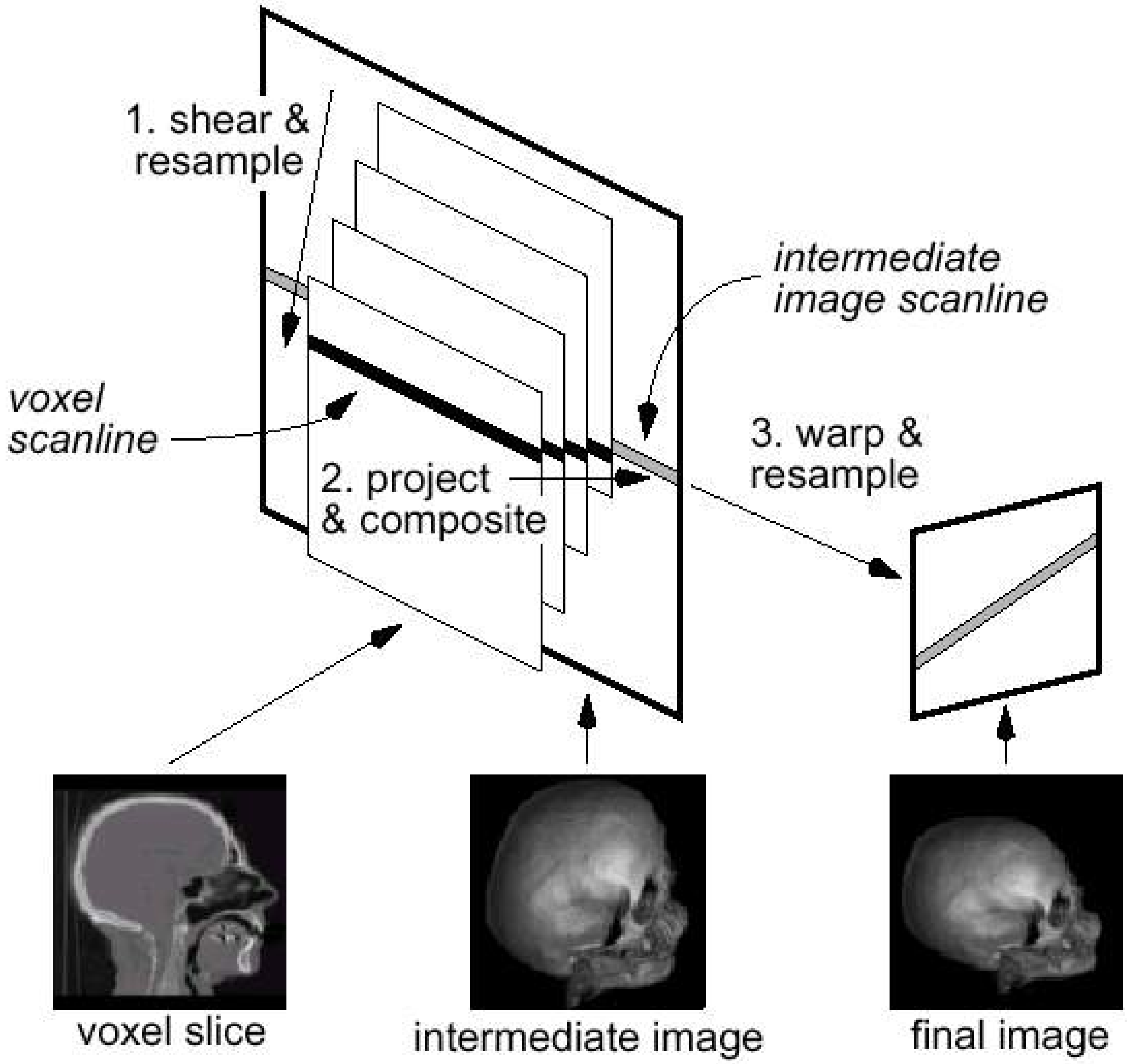
S = shears volume, **M** = transforms object to image coords)

Goal: parallel rays, voxel:pixel = 1:1, easy and fast compositing



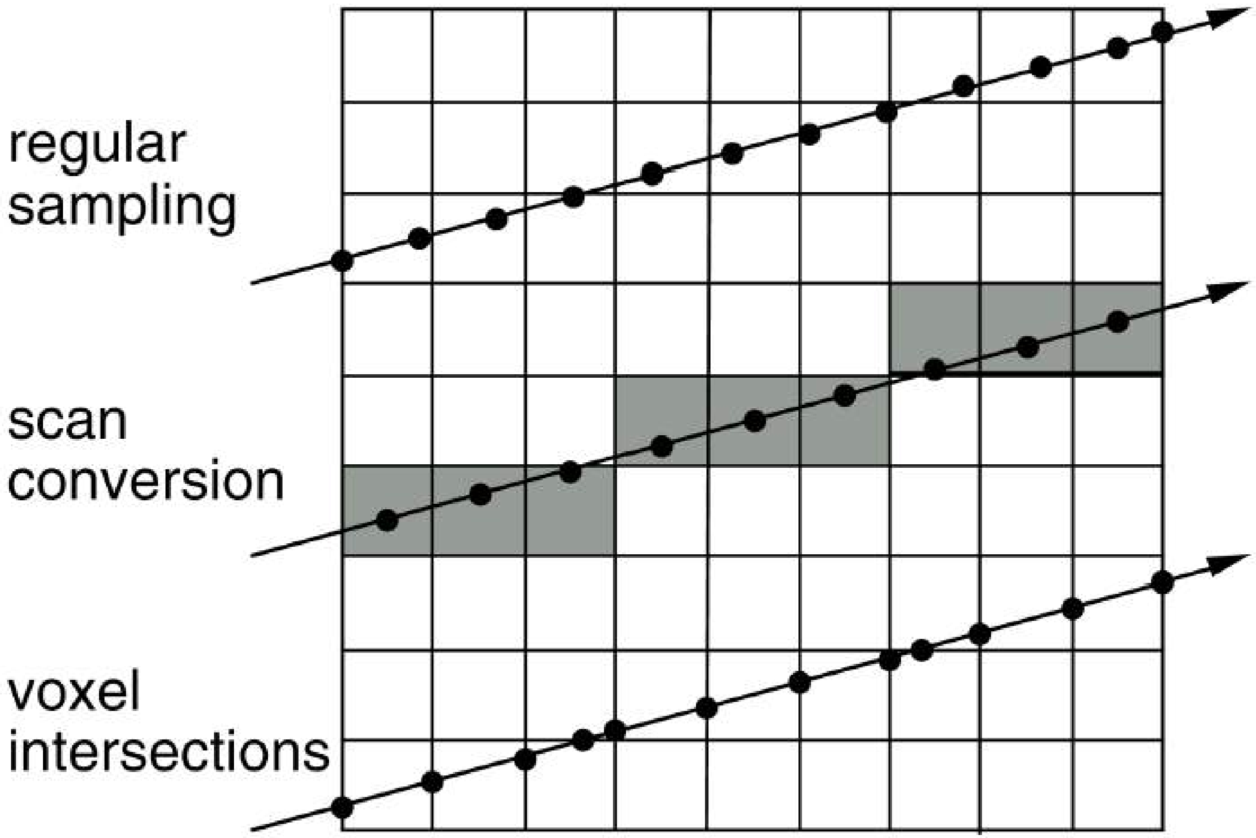
Shear, Project, Warp

1. shear-step
   - 2 shears (x and y)
   - bi-linear reconstruction
2. project-step
   - compositing using 1 voxel / pixel
   - result: intermediate image
3. warp-step
   - transform intermediate image to screen
   - affine transformation (rotation, translation, scale) in image space



Robert S. Laramee
r.s.laramee@swansea.ac.uk

Recall: Ray traversal – Three Previous Approaches



- With a shear, voxel-to-pixel ratio is1:1
- Rays are orthogonal to slices

Shear-warp Factorization

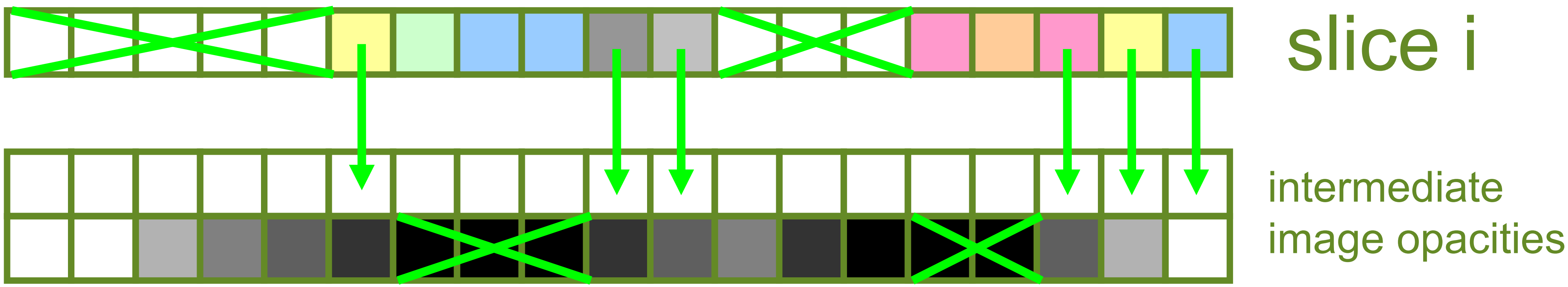
Implementation:

- Volume: 3* stacks of slices (for x, y, z)
- first shear in (x,y),
  then compositing and projection along z (x, y),
  finally a warp-operation for resulting image (including scaling)
- Acceleration:
  The use of run-length encoding for
  - volume data: runs of transparent voxels
  - image data (intermediate image): runs of opaque pixels
- simple and fast

Pseudo-Code

Shear-warp rendering (with RLE-coding):

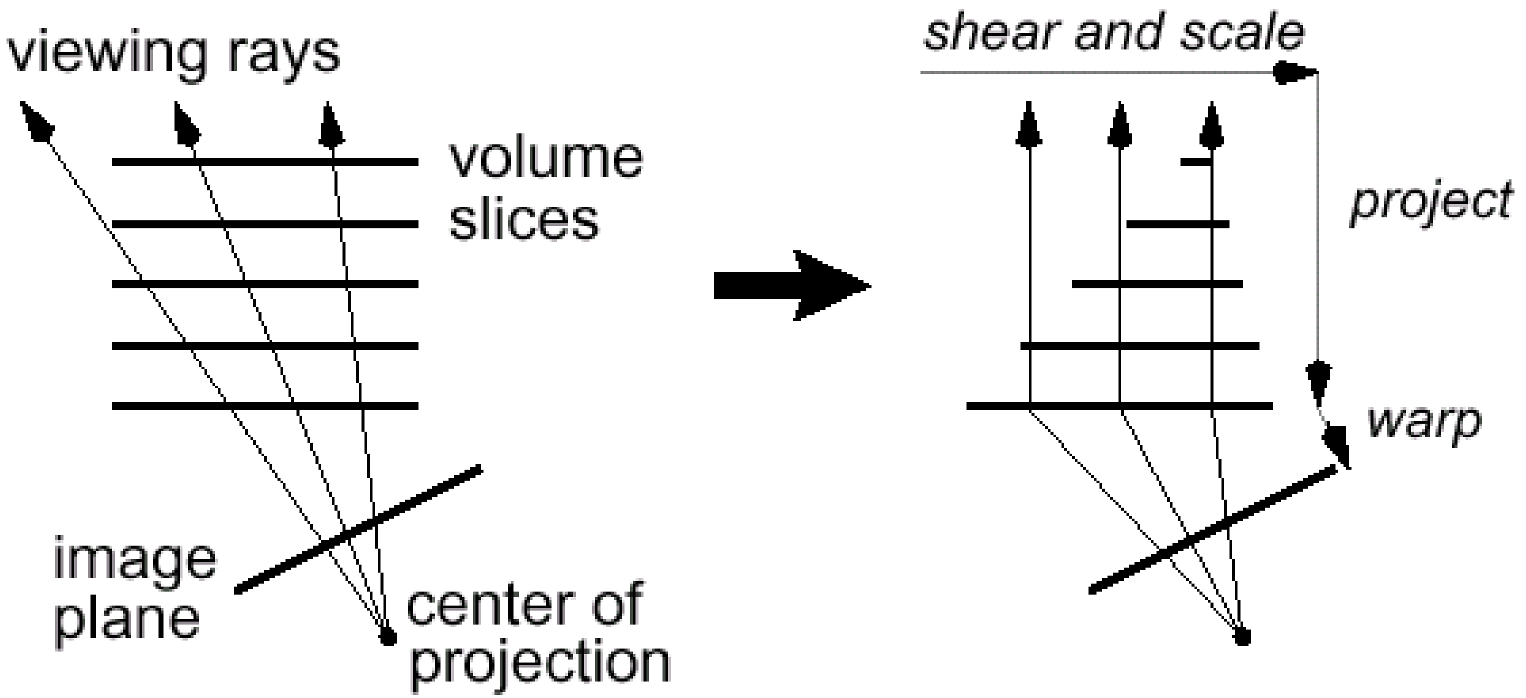
```
Select data-set (x, y, or z)
Apply shear
FOR all scanlines DO:
   FOR all slices, front-to-back DO:
      Skip transparent voxels (run)
      Skip opaque pixels (run)
      Do compositing
Warp intermediate image to screen
```



Perspective Projection

Instead of only shear:

- shear and scale: paying particular attention to reconstruction
- otherwise the same



Robert S. Laramee
r.s.laramee@swansea.ac.uk

Shear-warp – Conclusion

Advantages:

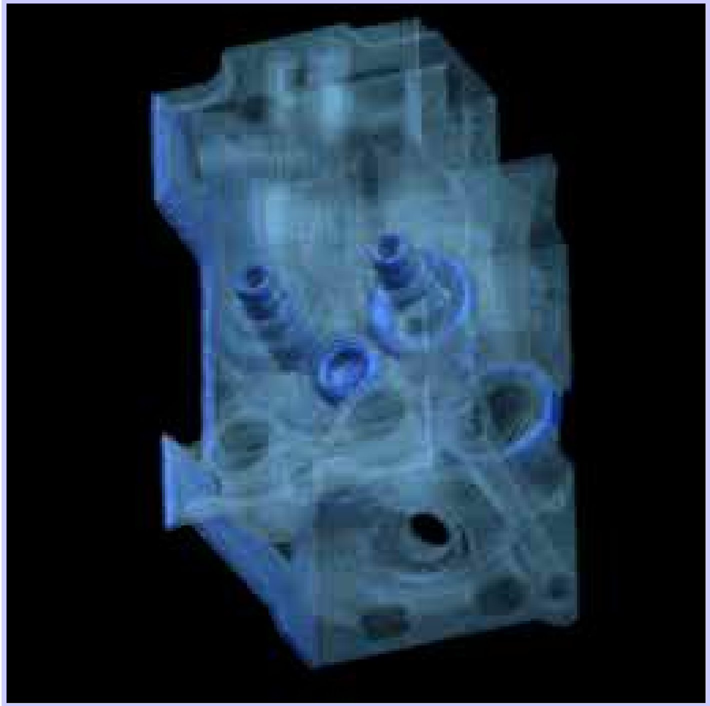
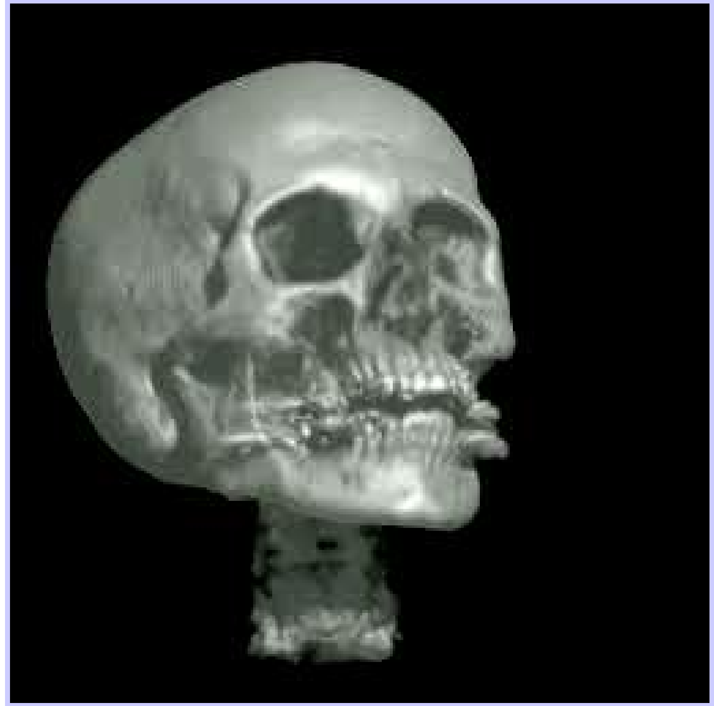
- fast
- easy
- perspective projection possible

Disadvantages:

- reconstruction only bi-linear (within slices)
- voxel/pixel (intermediate image) = 1 $\Rightarrow$ problem with zooming

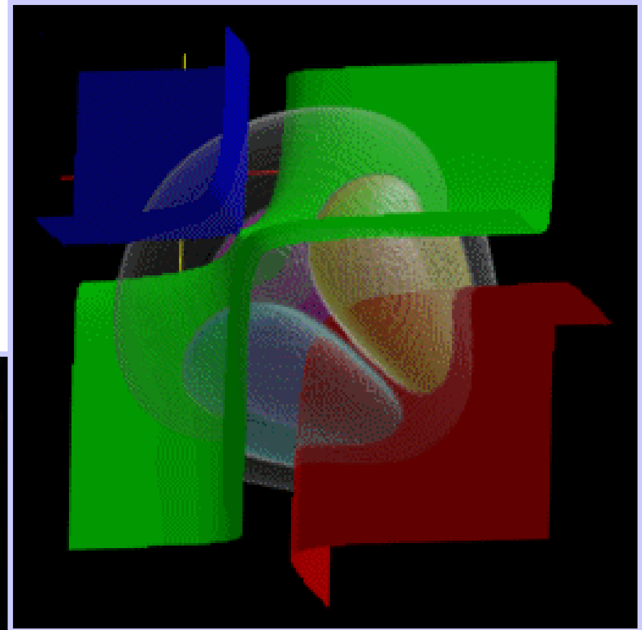
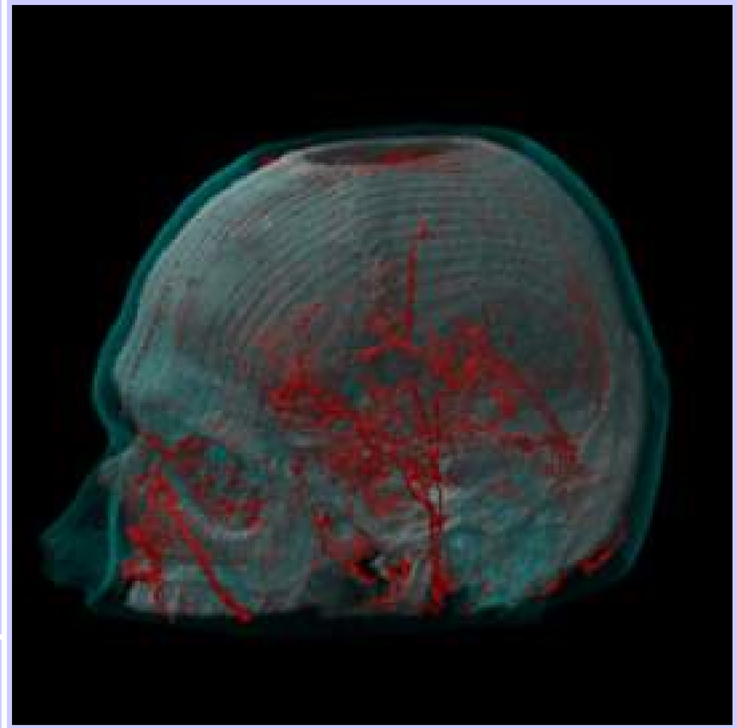
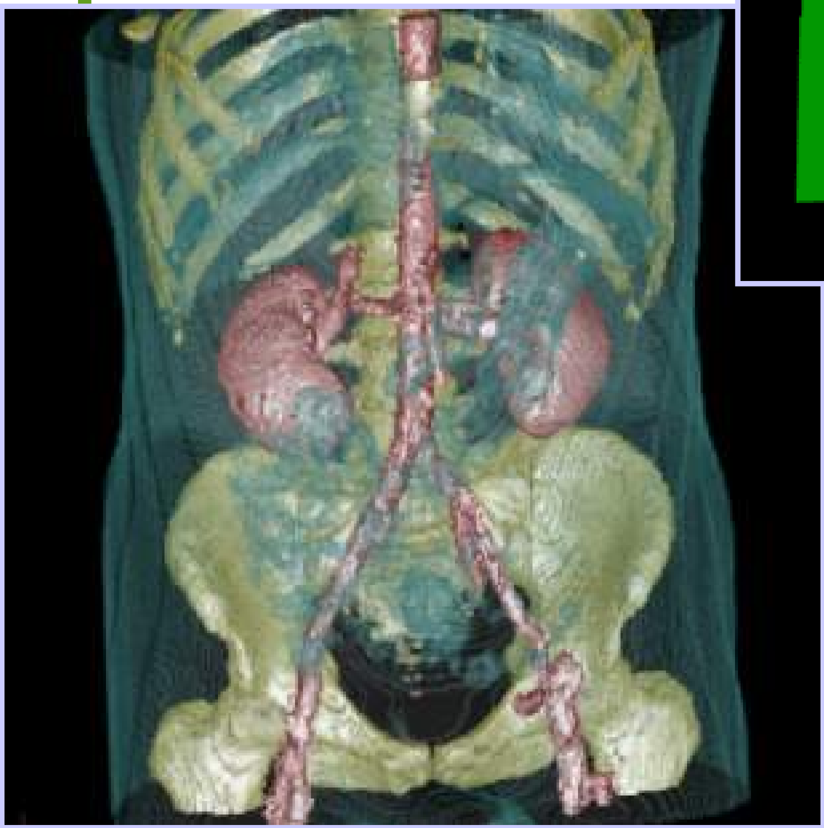
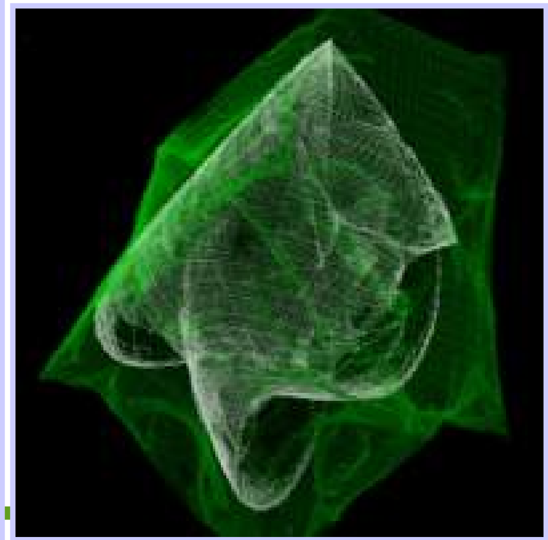
Shear-warp – Images

- Images from Ph. Lacroute:

Shear-warp – Images

- VRVis Images:

Literature

Paper (more details):

- Philippe Lacroute & Marc Levoy: “**Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation**” in *Proceedings of ACM SIGGRAPH ’94 = Computer Graphics*, July 1994

Acknowledgements

For this lecture material:

- Roberto Scopigno, Claudio Montani
- Peggy Li
- Roger Crawfis
- Hanspeter Pfister
- Philippe Lacroute
- Lukas Mroz
- Meister E. Gröller
- Torsten Möller
- Helwig Hauser
- Daniel Weiskopf

A Practical Evaluation of Popular Volume Rendering Algorithms

slides adapted from:

M. Meißner, J. Huang, D. Bartz,

K. Mueller, R.Crawfis

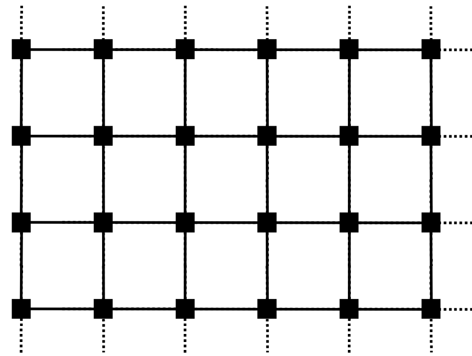
Overview

- Rendering Algorithms
- Framework
- Results
  - Image Quality
  - Rendering time
- Conclusions

Rendering Algorithms: Ray Casting

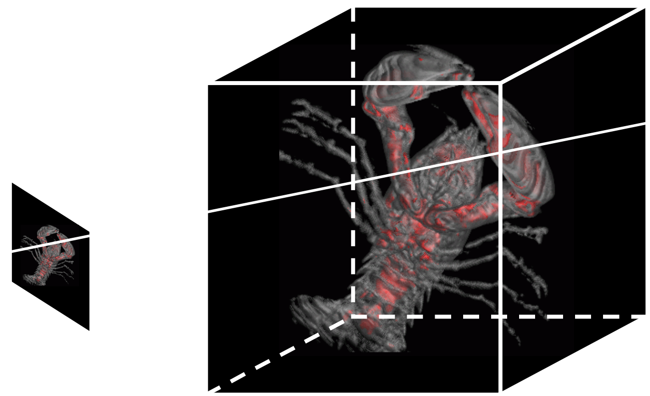
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

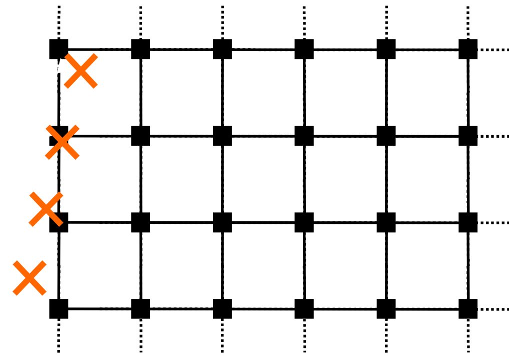
- Early ray termination



Rendering Algorithms: Ray Casting

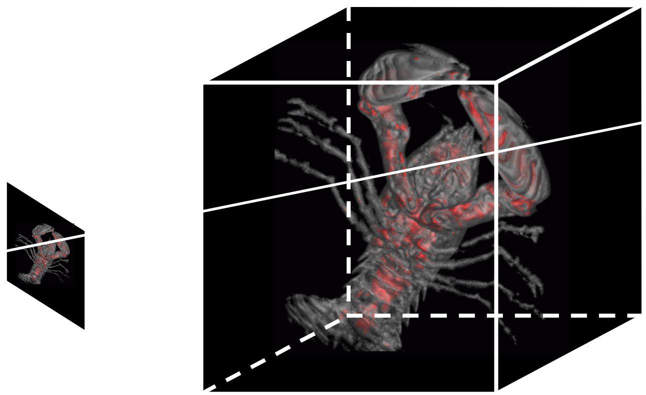
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

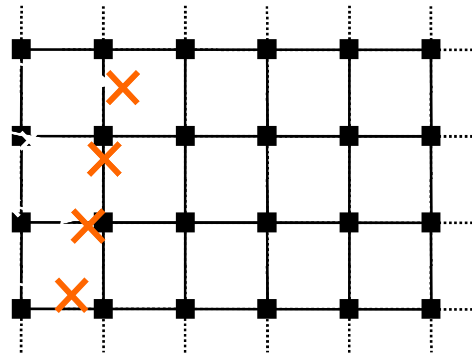
- Early ray termination



Rendering Algorithms: Ray Casting

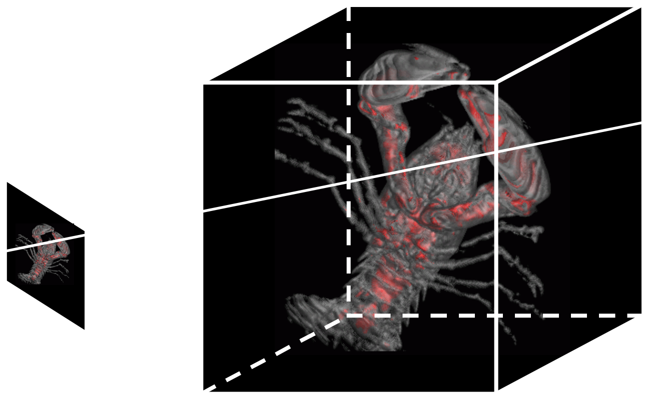
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

- Early ray termination

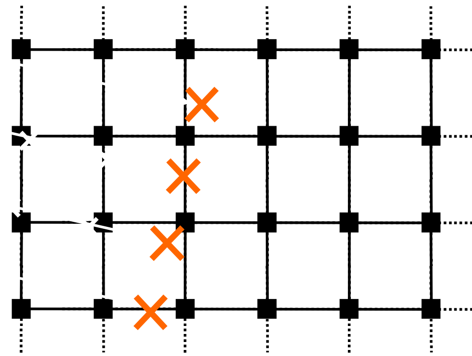


Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: Ray Casting

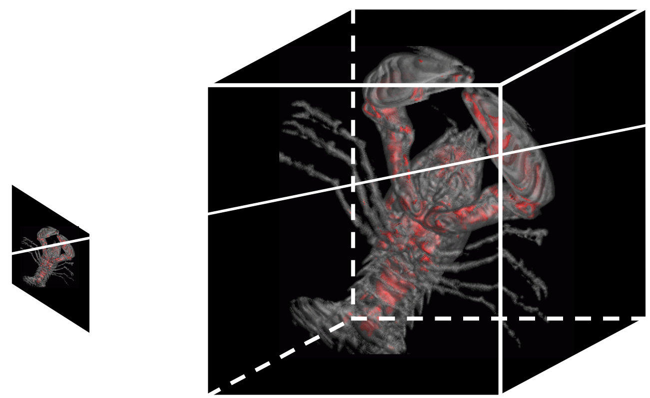
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

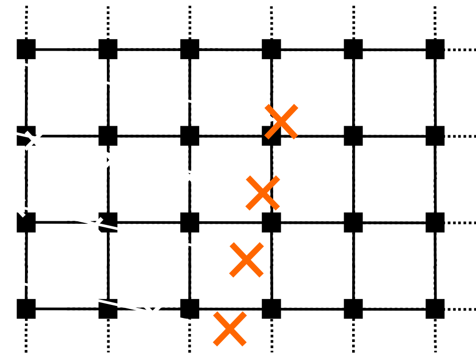
- Early ray termination



Rendering Algorithms: Ray Casting

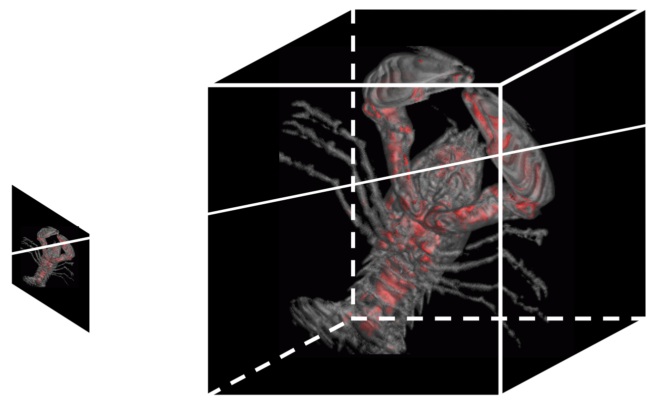
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

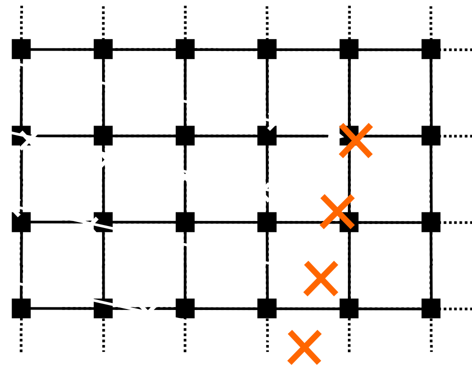
- Early ray termination



Rendering Algorithms: Ray Casting

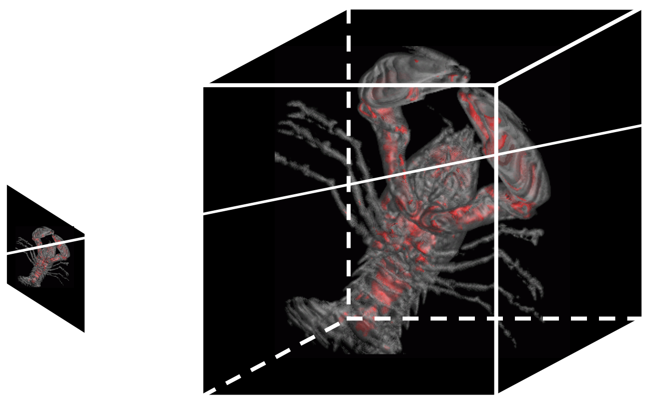
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

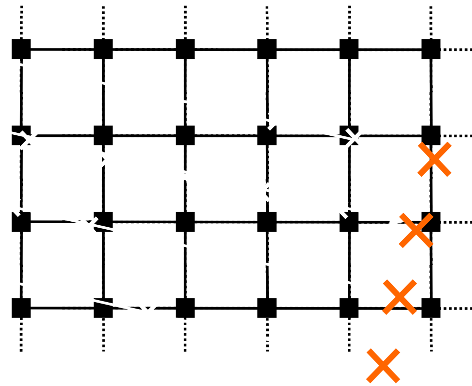
- Early ray termination



Rendering Algorithms: Ray Casting

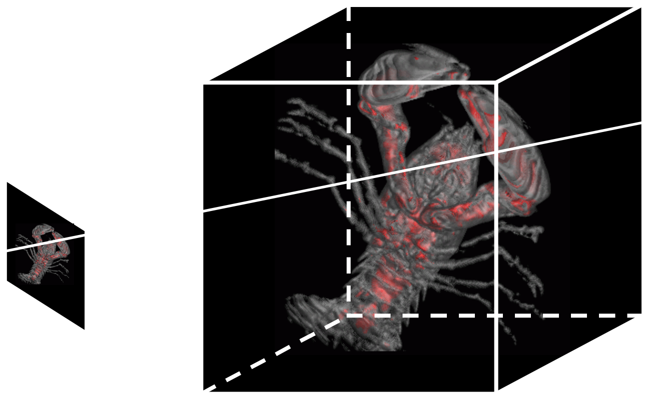
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

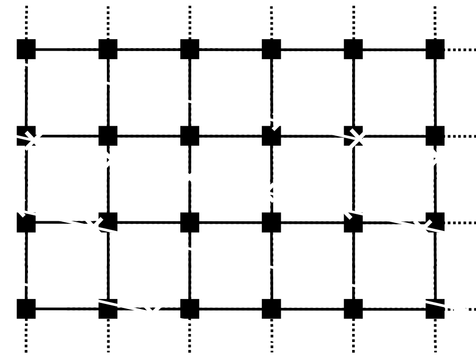
- Early ray termination



Rendering Algorithms: Ray Casting

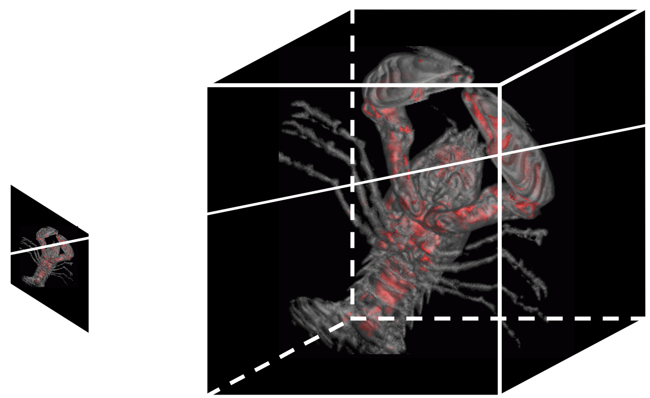
Rays are cast from view plane

- Trilinear interpolation
- Perspective & Parallel
- Different Sampling



Acceleration:

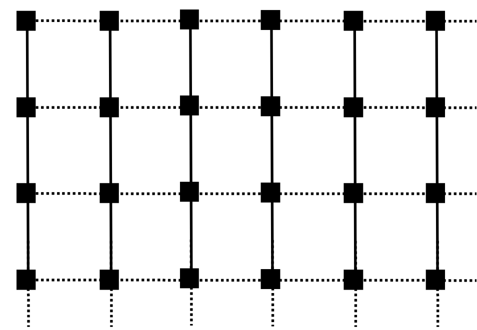
- Early ray termination



Rendering Algorithms: Shear Warp

Base plane aligned ray casting

- Bilinear interpolation
- Image quality depends on baseplane resolution



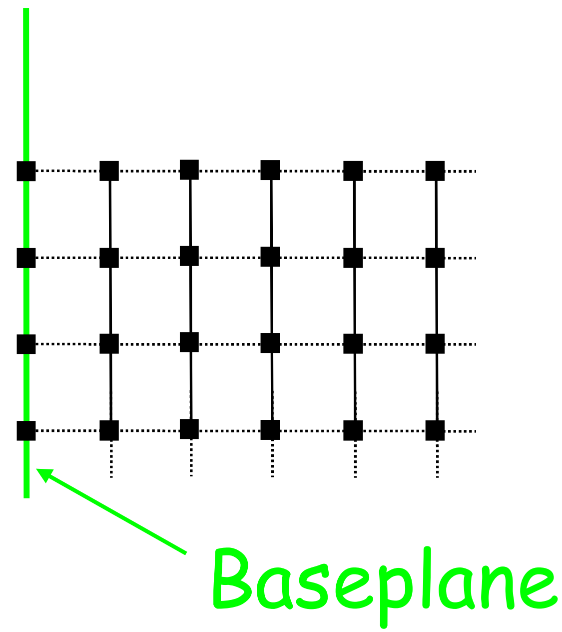
Acceleration:

- Regular memory access
- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



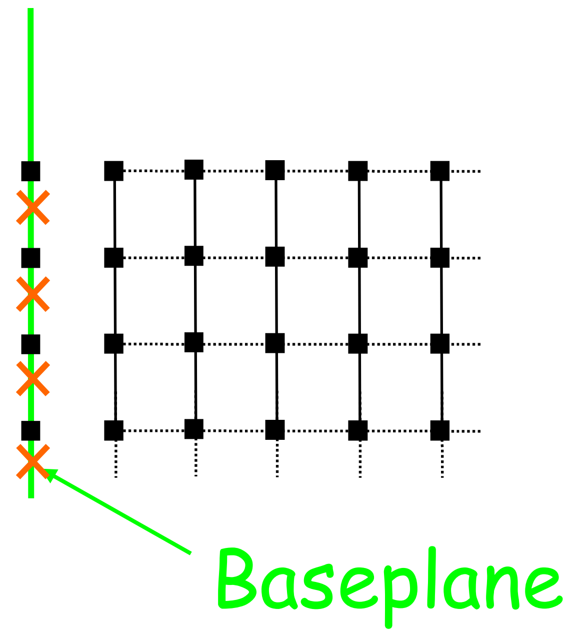
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

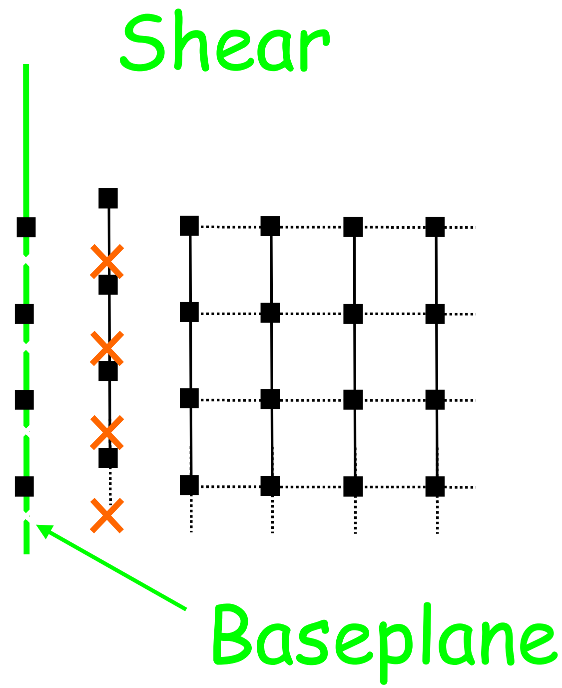
- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp



Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution

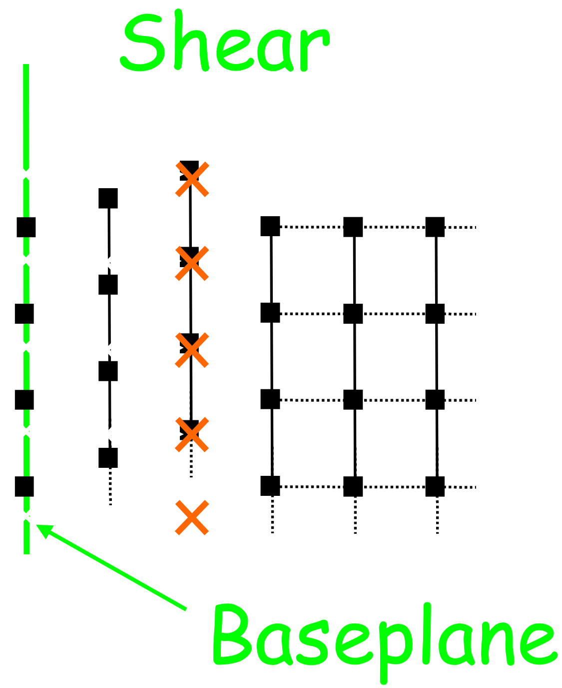
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

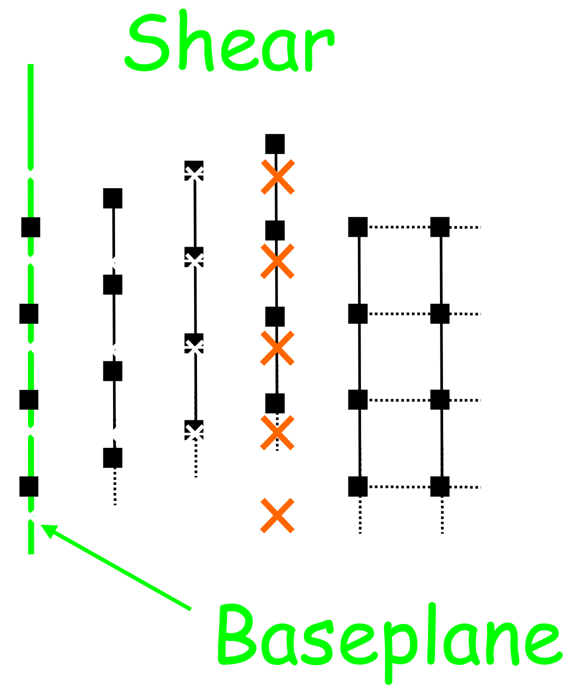
- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp



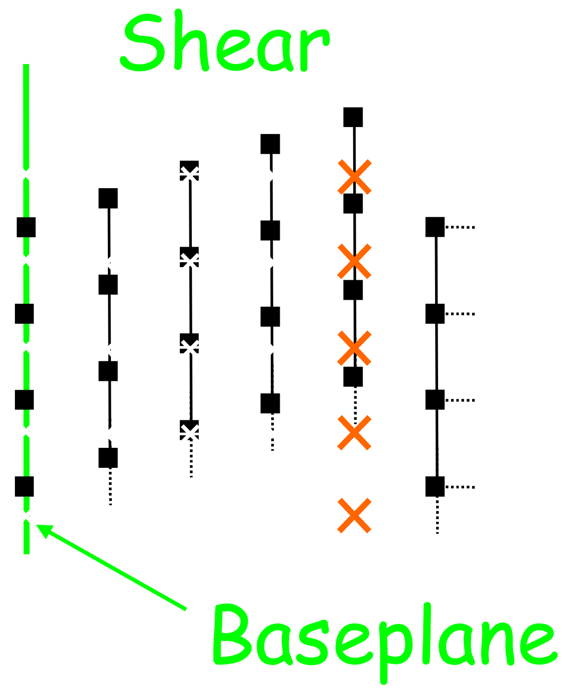
Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution

Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp



Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution

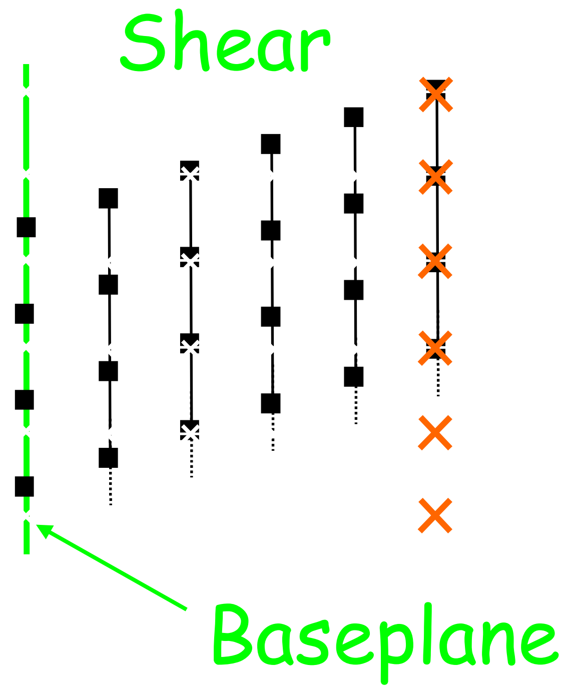
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



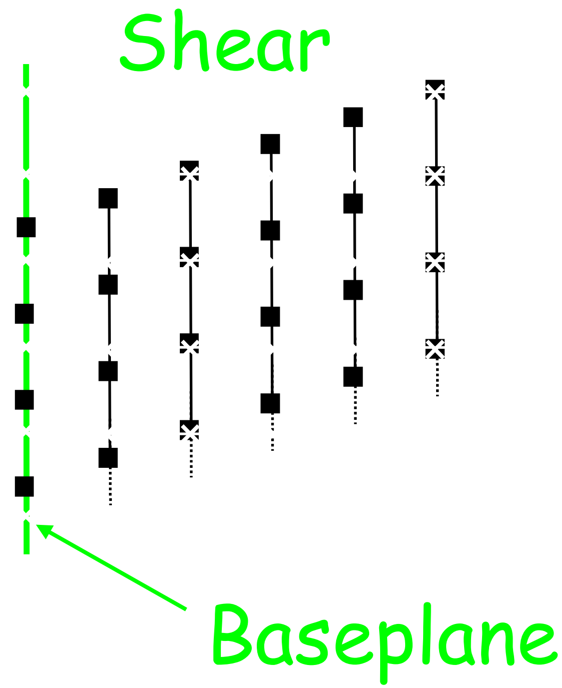
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



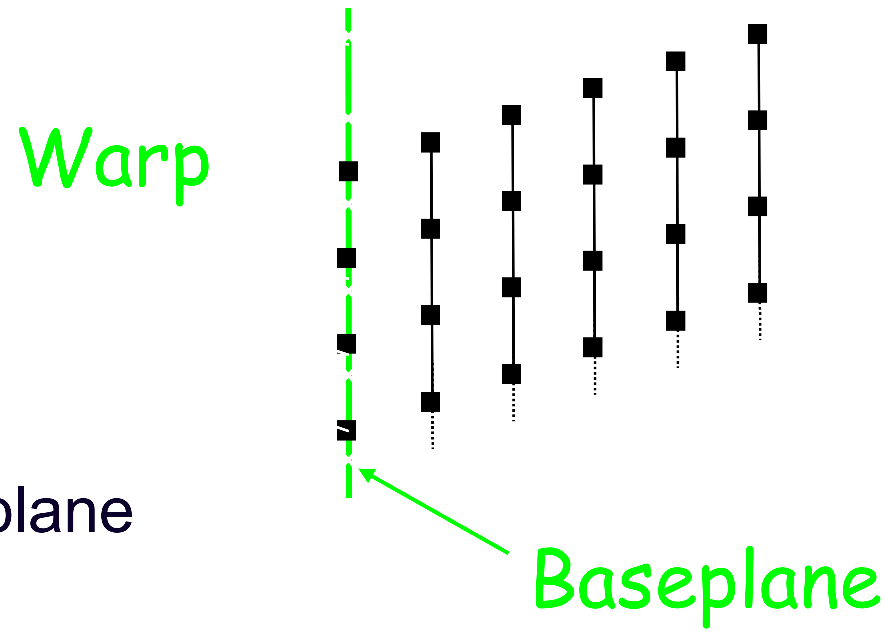
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: Shear Warp

Base plane aligned ray casting

- Bilinear interpolation
- Regular memory access
- Image quality depends on baseplane resolution



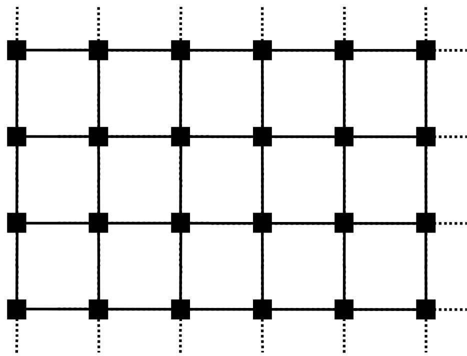
Acceleration:

- Run-length encoding
- Early ray termination

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



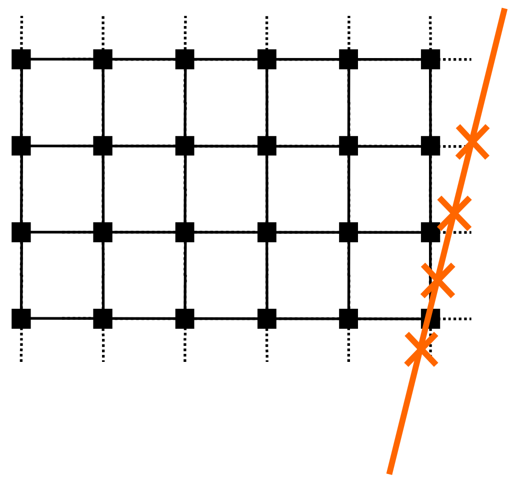
Acceleration:

- Hardware (brute force)

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



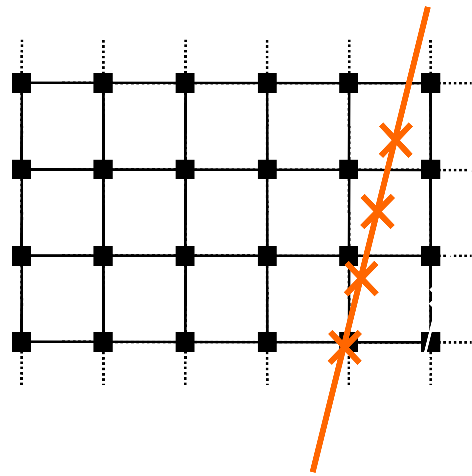
Acceleration:

- Hardware (brute force)

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



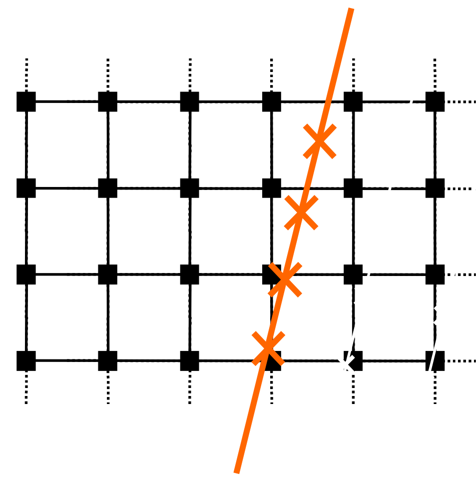
Acceleration:

- Hardware (brute force)

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



Acceleration:

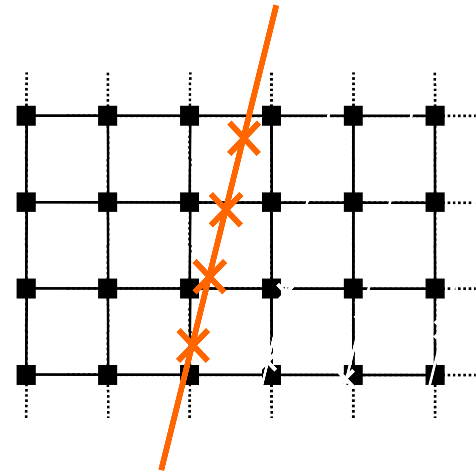
- Hardware (brute force)

Computer Science
Gwyddor Cyfrifiadur

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



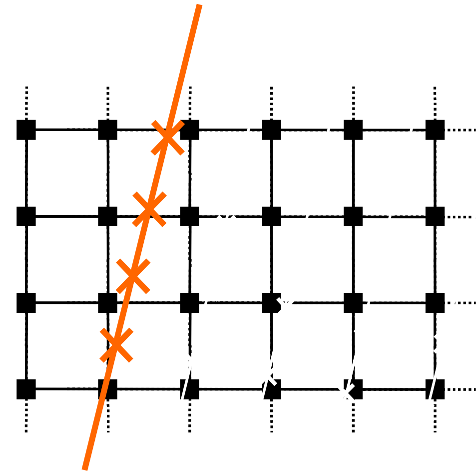
Acceleration:

- Hardware (brute force)

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



Acceleration:

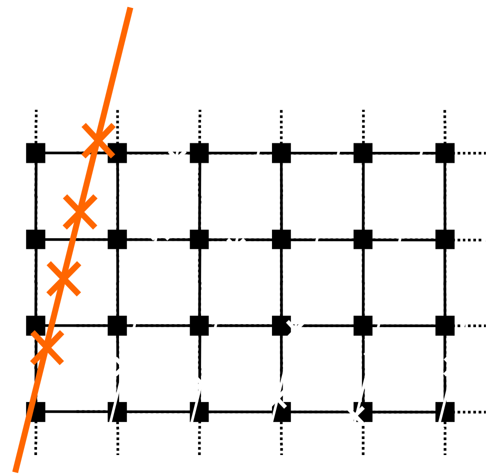
- Hardware (brute force)

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



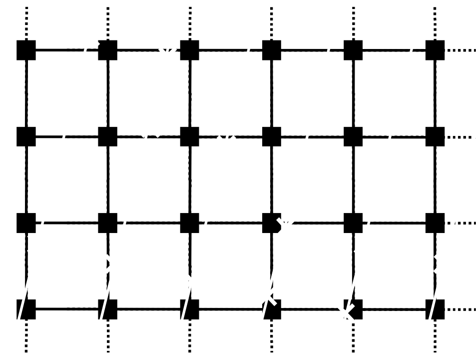
Acceleration:

- Hardware (brute force)

Rendering Algorithms: 3D Texture Mapping

Texture interpolation across triangles

- Trilinear interpolation
- No shading
- No opacity correction
- Limited precision



Acceleration:

- Hardware (brute force)

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: Splatting

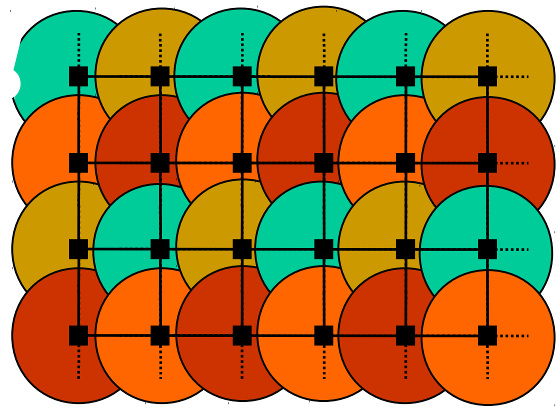


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

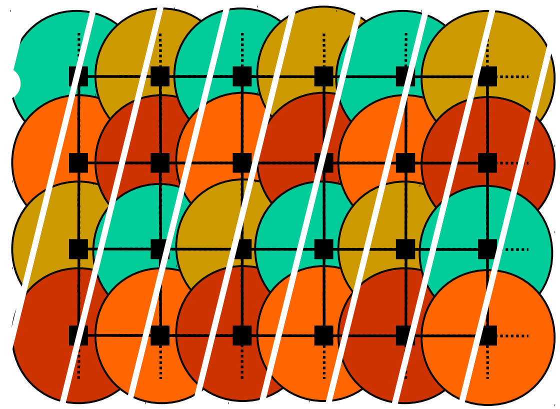


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

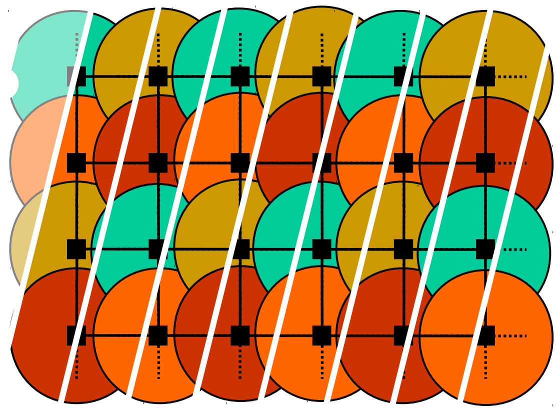


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

Acceleration:

- Early splat elimination

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: Splatting

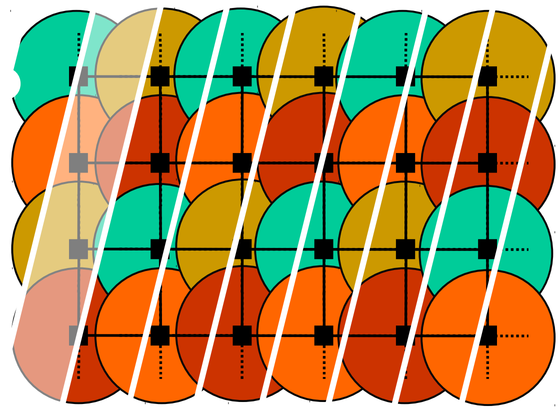


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

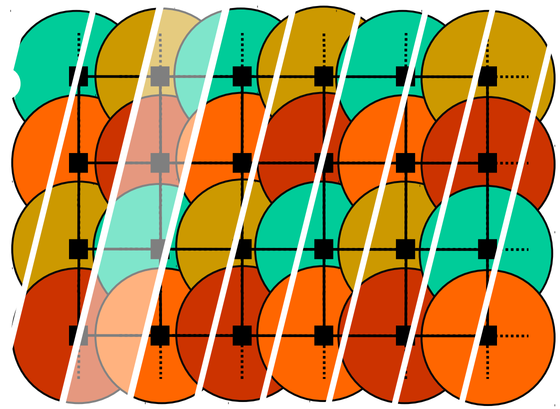
Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel



Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

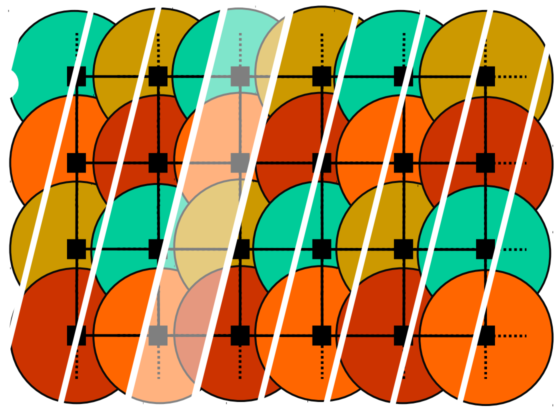


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

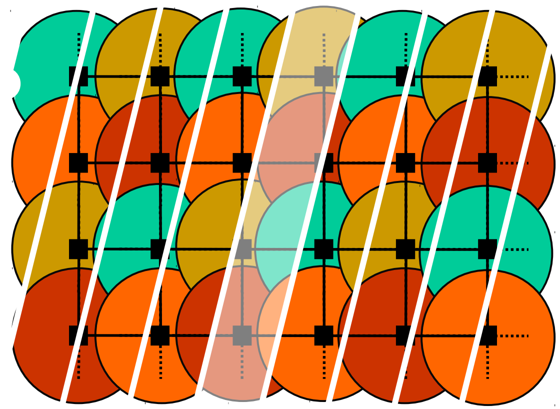
Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel



Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

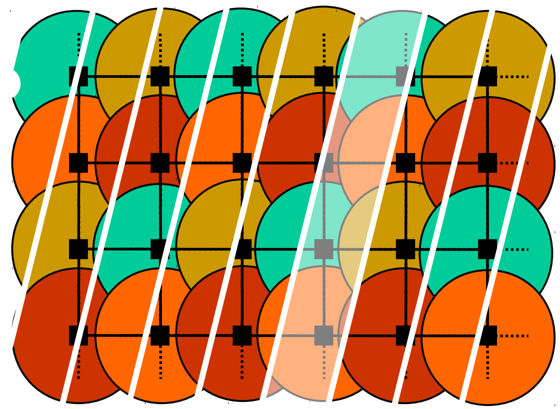


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

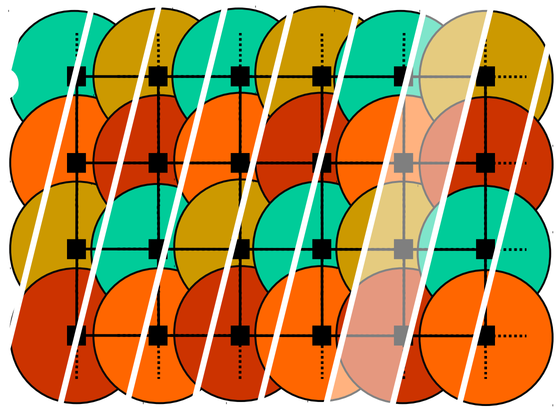


Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel

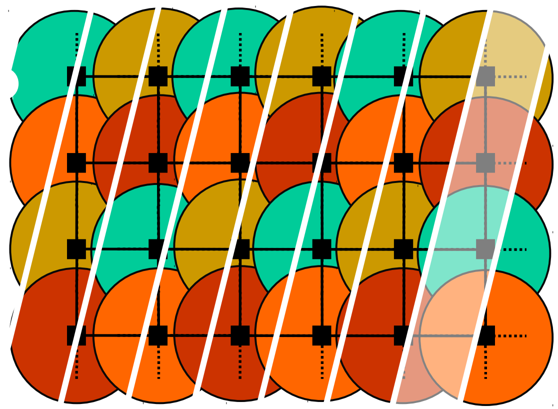
Acceleration:

- Early splat elimination

Rendering Algorithms: Splatting

Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel



Acceleration:

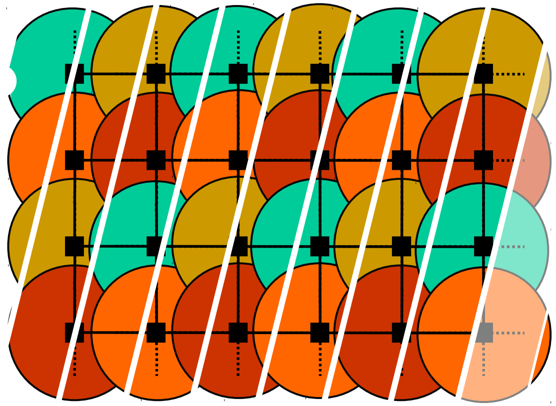
- Early splat elimination

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: Splatting

Image-aligned splatting

- Pre-integrated kernels and kernel sections
- Gaussian filter kernel



Acceleration:

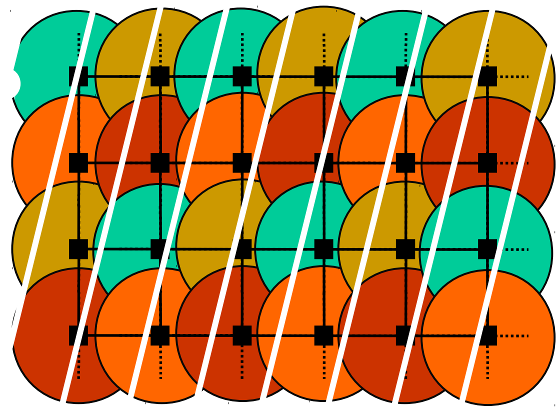
- Early splat elimination

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Rendering Algorithms: Splatting

Image-aligned splatting

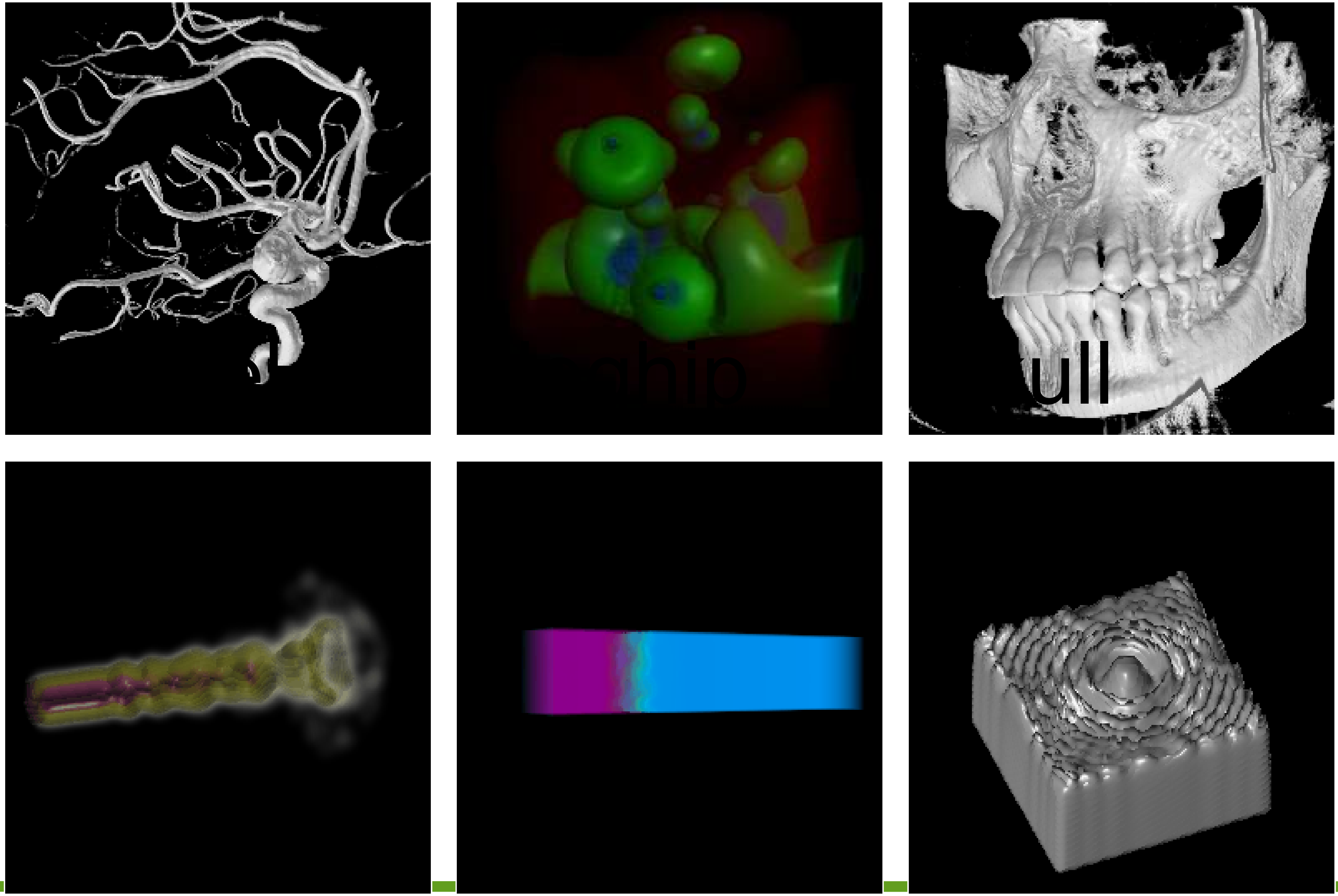
- Pre-integrated kernels and kernel sections
- Gaussian filter kernel



Acceleration:

- Early splat elimination

Datasets Used for Algorithm Comparison



Results

High Quality:

- Ray casting
- Splatting

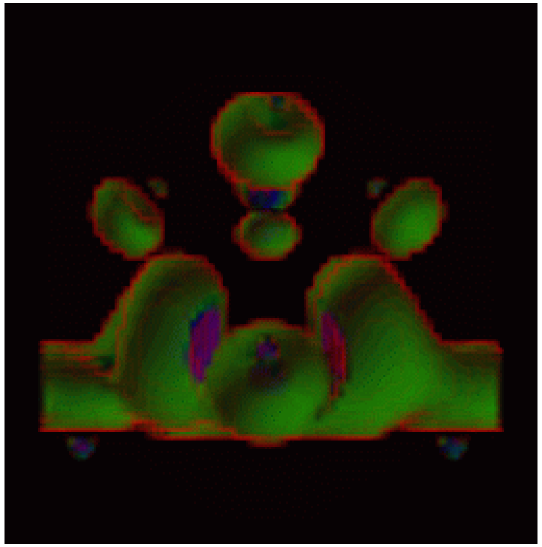
High speed

- 3D Texture Mapping
- Shear-warp

Results: Texture Mapping vs Ray Casting

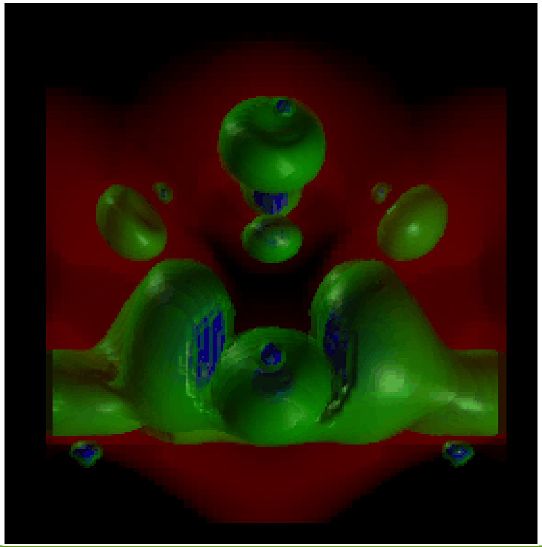
Texture

Mapping:



Ray

Casting:



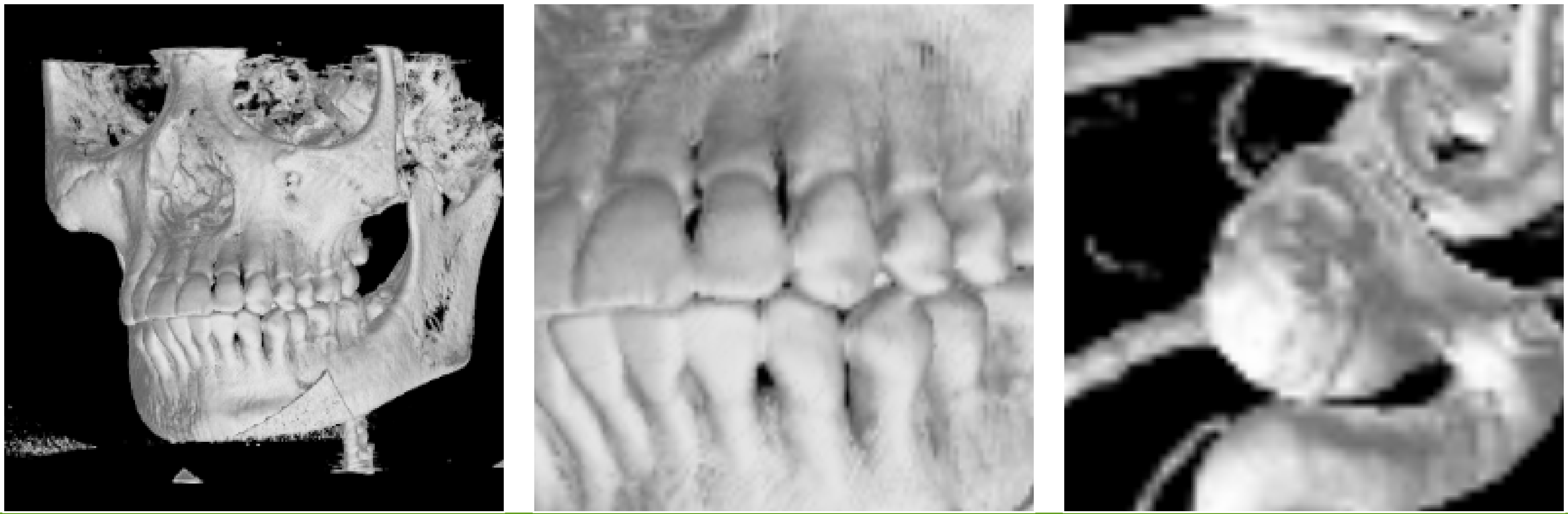
Results: Ray Casting vs. Shear Warp

Ray

Casting:

Shear

Warp:

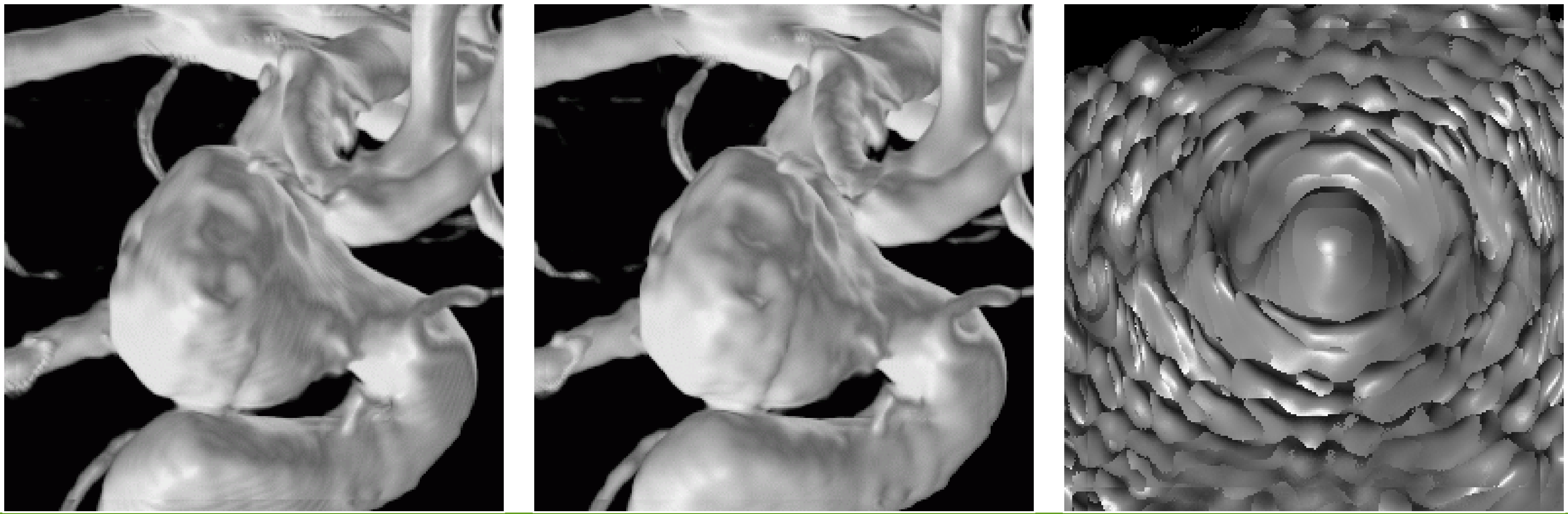


Results: Ray Casting vs. Splatting

Ray

Casting:

Splatting:



Robert S. Laramee
r.s.laramee@swansea.ac.uk

Results: Image Quality

- Shear warp almost resembles Ray Casting if image size and base plane size match
- Ray Casting and Splatting equal, Splatting introduces smoothing
- Texture Mapping lacks precision and functionality

Results: Rendering Time

All experiments were performed on:

- A SGI Octane with MXE graphics
- R10000 CPU
- 250 MByte of main memory
- 4 MByte of texture memory

Results: Rendering Time

3D Texture Mapping

- Subsecond render times for all presented datasets (viewport 256^2)
- Fast if volume fits into texture memory
- Larger volumes: Lower cache efficiency and bricking slow down

Shear Warp

- Subsecond render times for all presented datasets (viewport 256^2)
- Faster than Texture Mapping for low pixel content
- Run Length Encoding is preprocessing and classification dependent
- Cache efficient even for large volumes

Results: Rendering Time

Ray Casting

- Rendering in order of second(s)
- Scaling for larger viewports (images):
  - Better for high pixel content
  - Worse for low pixel content (empty space)
- Large volumes will require cubic addressing scheme (axis independent)

Splatting

- Rendering in order of second(s)
- Scaling for larger viewports (images):
  - Better for low pixel content
  - Worse for high pixel content

Conclusions

- Many input and output parameters to consider
- Empty regions are performance slow-downs for Ray Casting and Texture Mapping
- High pixel content is a performance slow-down for Splatting
- Shear Warp is a trade-off between quality and speed
- There seems to be a critical threshold for viewport size relative to dataset size where trilinear interpolation (Ray Casting) is better than footprint mapping (Splatting) and vice versa

More Information

- Datasets, classifications, views, and software available at:
- www.volren.org
- www.voreen.org

See also

Data Visualization, Principles and Practice, Chapter 10 Volume Visualization, by A Telea, AK Peters, 2008