Visualization

Introduction to Information Visualization

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Organisation

- Contact: bernd.froehlich@uni-weimar.de
- Office hours by appointment only
- Teaching assistants
  - M.Sc. Sebastian Thiele
  - Dipl. Medsys. Wiss. Patrick Riehmann / M.Sc. Henning Gründl
- Date and Location: Thursdays, 15:15h, BH 11, R015
- Additional lectures during lab class hours or Wednesday
- Lab classes
  - Will be announced later
- Web: www.uni-weimar.de/medien/vr
  - Lecture notes and videos of lectures (user: vr password: vr2015ss_buw)
  - Last year‘s lecture notes are still online (user: vr password: vr2014_ss)
  - Videos from 2012 course are still online on last year‘s webpage

Organisation

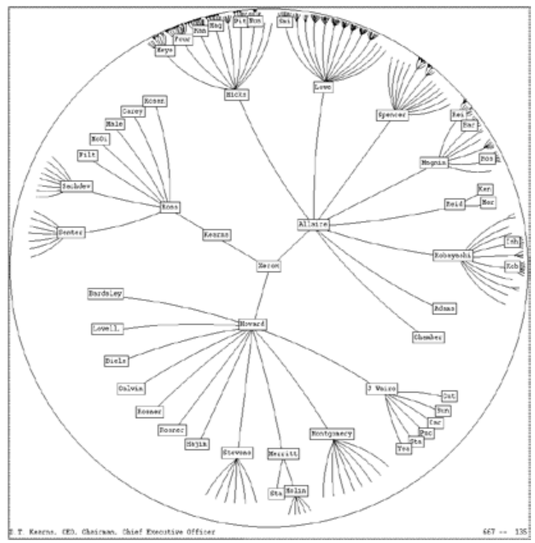
- Grading
  - Final exam (written) 30 minutes (50%)
  - Assignments (20%)
  - Final project (groups of two, workload about 1 week) (30%)
- Workload
  - 2 SWS lecture + 1 SWS lab class
  - 4.5 ECTS
- Modules
  - Grafische Informationssysteme (B.Sc. Medieninformatik)
  - Electives (M.Sc. Computer Science and Media)
  - Information Processing and Presentation (M.Sc. HCI)

Additional Lecture

- Next Tuesday, April 14 at 18:45-20:15h R015/B11
- Wed 29.4. at 15:15 (R015)

Visualization

- Scientific Visualization
  - Physical / simulated data
  - Data
    - Automotive
    - Weather
    - Medical
    - ...
- Information Visualization
  - Abstract, non-physical data
  - Data
    - Financial/business
    - Statistics
    - Software
    - Text
    - W W W
    - ...



Books (available at Library)

- Robert Spence: Information Visualization 2nd Ed
- Colin Ware: Information Visualization 2nd Ed.
- Riccardo Mazza: Introduction to Information Visualization, 2009
- Schumann, Müller: Visualisierung. Grundlagen und allgemeine Methoden
- Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman: Readings In Information Visualization: Using Vision to Think
- Jacques Bertin: Semiology of Graphics
- Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014. http://www.cs.ubc.ca/~tmm/vadbook/
- http://www.infovis-wiki.net

Beautiful Overview

- Heer, J., Bostock, M., and Ogievetsky, V. 2010. A tour through the visualization zoo. *Commun. ACM* 53, 6 (Jun. 2010), 59-67.
- http://portal.acm.org/citation.cfm?id=1743546.1743567

InfoVis Tools

- Mondrian
  - http://www.theusrus.de/Mondrian/
- Protovis
  - http://mbostock.github.com/protovis/
- d3.js: D3.js is a small, free JavaScript library for manipulating documents based on data
  - http://mbostock.github.com/d3/
  - http://techslides.com/over-1000-d3-js-examples-and-demos/
- Weave
  - http://www.oicweave.org/
- Tableau
  - http://www.tableausoftware.com/public/community

InfoVis Websites

- http://www.ncomva.se/guide/index.php
- IEEE Information Visualization Conference – the best international conference in information visualization
- Good InfoVis papers http://ivi.sagepub.com/

InfoVis Videos

- http://www.ted.com/talks/tags/visualizations
  - Hans Rosling‘s talks are entertaining
- https://itunes.apple.com/gb/podcast/uw-cse-colloquia-video-podcast/id325343789

Acknowledgements

- First two lectures are based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.
- Further based on the books and lectures by
  - Robert Spence: Information Visualization 2nd Ed
  - Colin Ware: Information Visualization 2nd Ed.
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Silvia Miksch, TU Wien
  - Jeffrey Heer, Stanford University

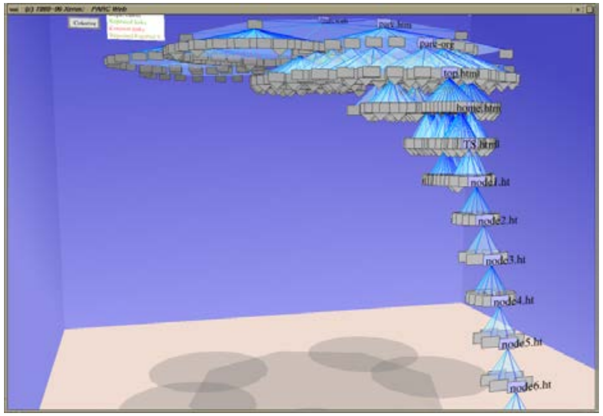
Big Data Challenge

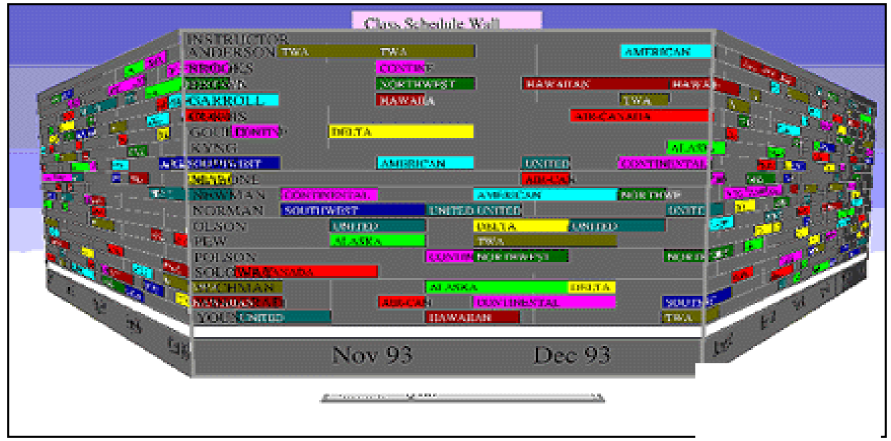
- Data production and dissemination
  - Faster creation and collection
  - Faster dissemination
- Fast growth of new digital information
  - 5 exabytes of new information in 2002 [Lyman 03]
  - 161 exabytes in 2006 [Gantz 07]
  - 1,200 exabytes in 2010 [Gantz 10]
- Necessitates better tools and algorithms for visually exploring and conveying information

The ability to take data—to be able to ***understand*** *it, to* ***process*** *it, to* ***extract value*** *from it, to* ***visualize*** *it, to* ***communicate*** *it—that's going to be a hugely important skill in the next decades, … because now we really do have* ***essentially free and ubiquitous data****. So the complimentary scarce factor is the ability to understand that data and extract value from it.*

Hal Varian, Google's Chief Economist
The McKinsey Quarterly, Jan 2009

Information Visualization





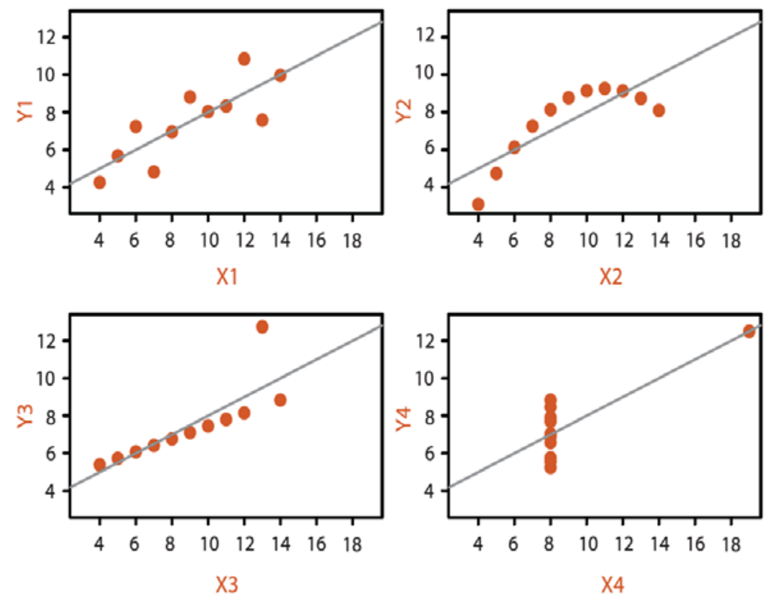
- In 1989 Xerox PARC Jock Mackinlay, Stu Card, and George Robertson coined the term “Information Visualization”
- What is “Information”?
  - Items, entities, things without a direct physical correspondence
  - Notion of abstractness of the entities is important
- What is “visualization”?
  - The use of computer-supported, interactive visual representations of data to amplify cognition. [Card, Mackinlay Shneiderman ‘98]
- Visuals help us think, they support discovery, decision making, explanation, …

Munzner's Definition of Visualization

- Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.
- Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.
- Many analysis problems ill-specified: don't know exactly what to ask in advance

Anscombe's Quartet

Identical statistics	
x mean	9
x variance	10
y mean	8
y variance	4
x/y correlation	~1



- Summaries lose information
  - Need to confirm expected and find unexpected patterns / assess validity of statistical model

Why Use Visual Representations?

- Human visual system is high-bandwidth channel to brain
  - Overview possible due to background processing
    - Subjective experience of seeing everything simultaneously
    - Significant processing occurs in parallel and pre-attentively
- Sound: lower bandwidth and different semantics
  - Overview not supported
    - Subjective experience of sequential stream
- Touch/haptics: impoverished record/replay capacity
  - Only very low-bandwidth communication thus far
- Taste, smell: no viable record/replay devices

Why focus on tasks and effectiveness?

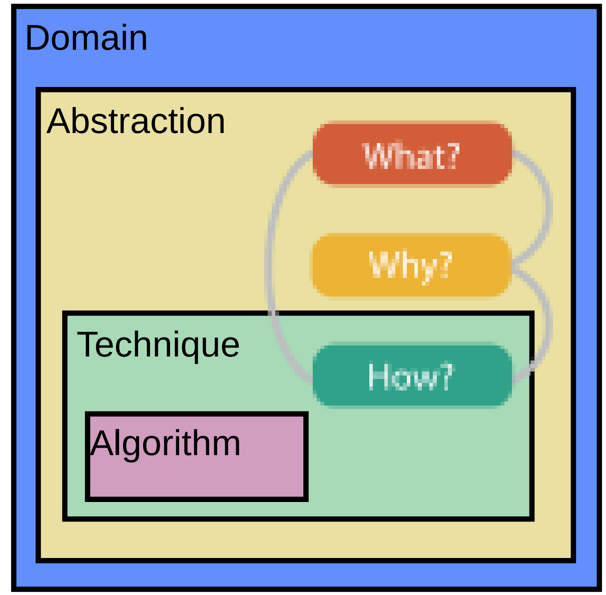
- Tasks serve as constraint on design (as does data)
  - Idioms do not serve all tasks equally!
  - Challenge: recast tasks from domain-specific vocabulary to abstract forms
- Most possibilities for visualization are ineffective
  - Validation is necessary, but tricky
  - Increases chance of finding good solutions if you understand full space of possibilities – ideal goal of this course
- What counts as effective?
  - Novel: enable entirely new kinds of analysis
  - Faster: speed up existing workflows

Resource Limitations

- Vis designers must take into account three very different kinds of resource limitations
  - Computational limits
    - Processing time
    - System memory
  - Human limits
    - Human attention and memory
  - Display limits
    - Pixels are precious resource, the most constrained resource
    - Information density: ratio of space used to encode info vs unused whitespace
      - Tradeoff between clutter and wasting space, find sweet spot between dense and sparse

Analysis Framework: Four Levels, Four Questions

- *Domain* situation
  - Who are the target users?
- *Abstraction*
  - Translate from specifics of domain to vocabulary of visualization
  - **What** is shown? **data abstraction**
  - **Why** is the user looking at it? **task abstraction**
- *Technique (Munzner: Idiom)*
  - **How** is it shown?
    - **Visual encoding technique**: how to draw
    - **Interaction technique**: how to manipulate
- *Algorithm*
  - Efficient computation

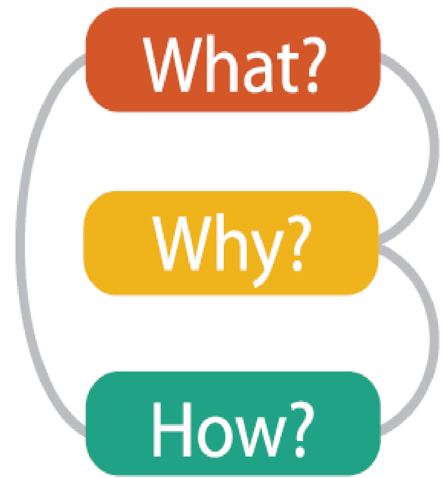


[A Nested Model of Visualization Design and Validation.
Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]

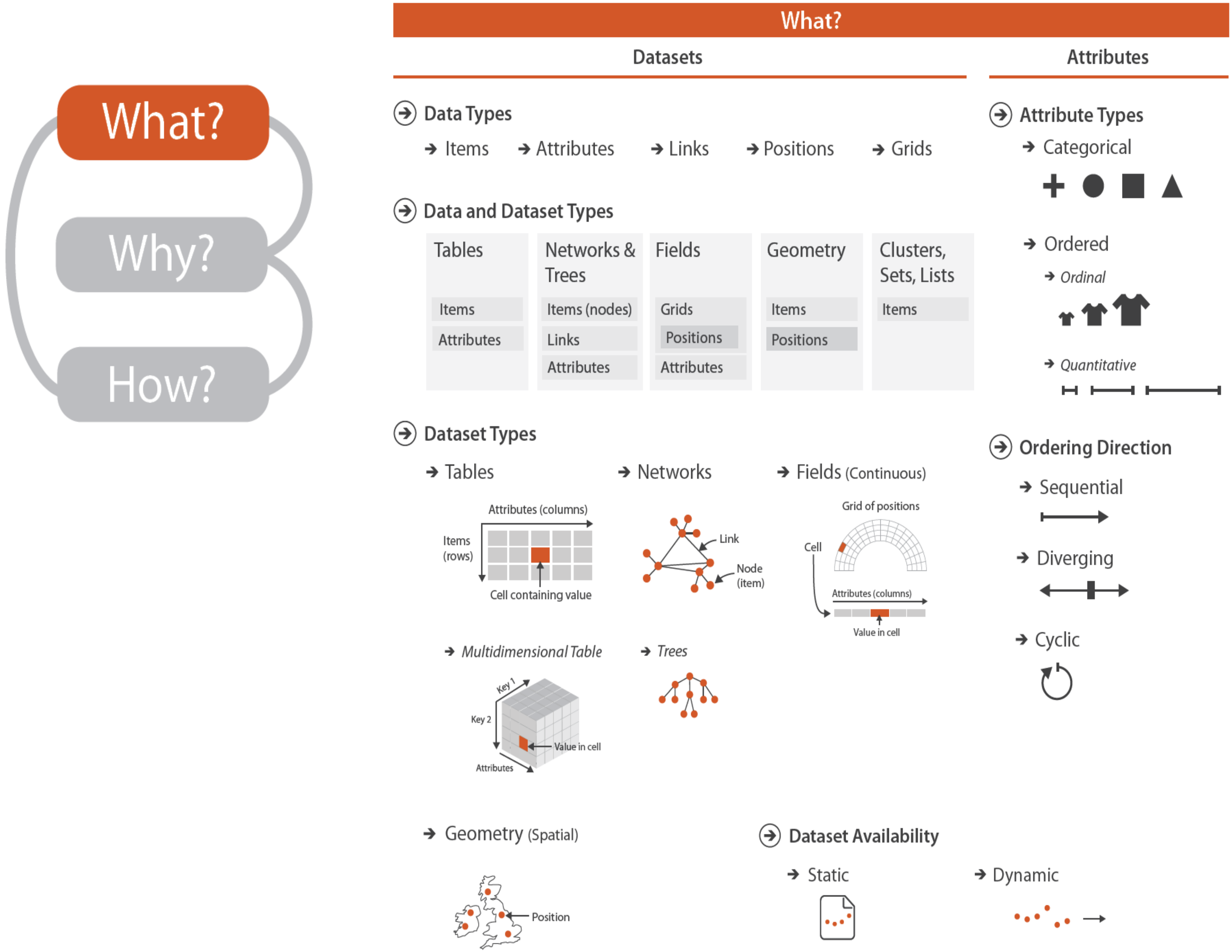
[A Multi-Level Typology of Abstract Visualization Tasks
Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

Analysis: What, Why, and How

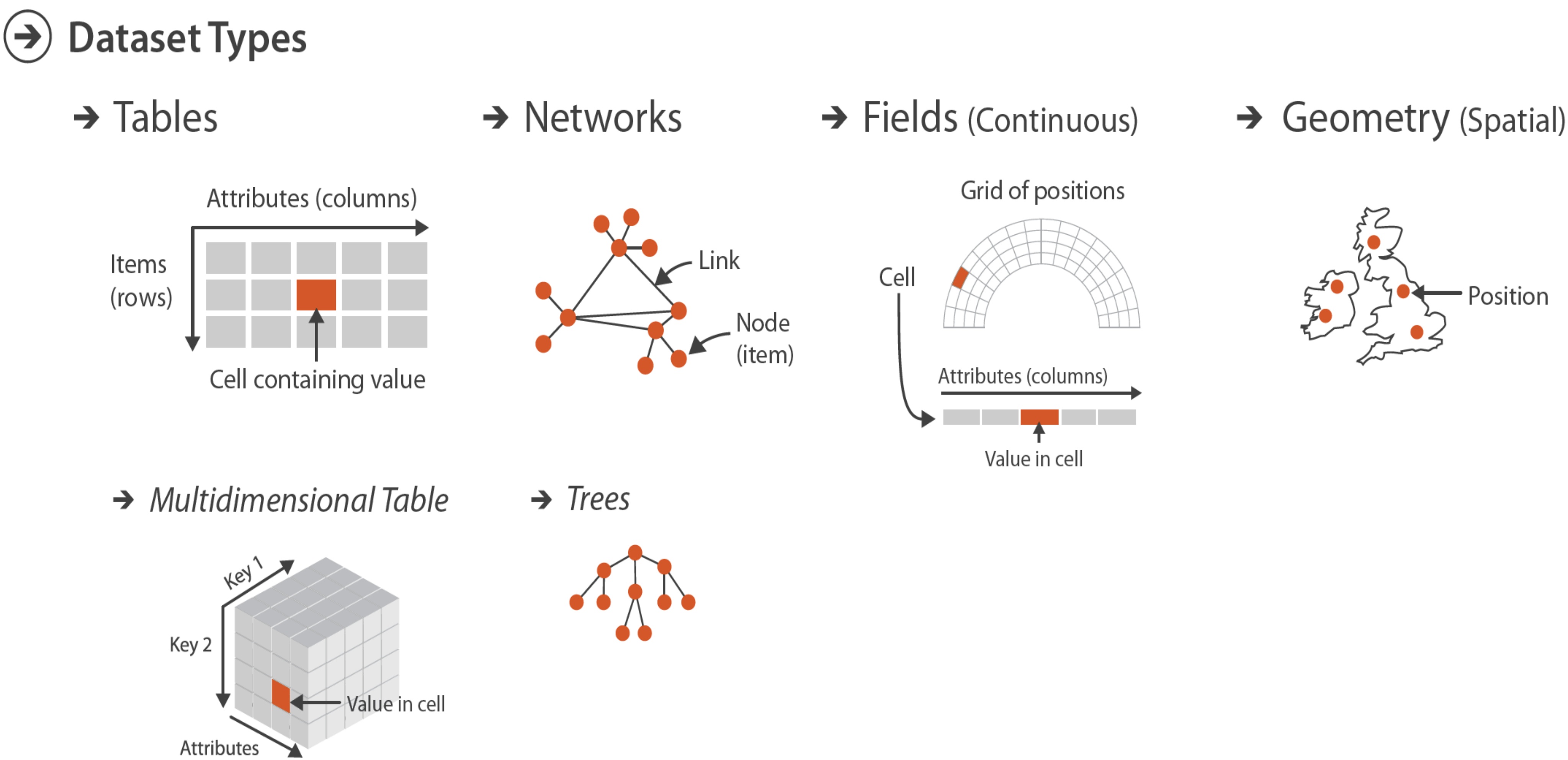
- What is shown?
  - **Data** abstraction
- Why is the user looking at it?
  - **Task** abstraction
- How is it shown?
  - **Technique**: visual encoding and interaction



- Abstract vocabulary avoids domain-specific terms
  - Translation process iterative and often complex
- What-why-how analysis framework helps to think systematically about design space


What?
Why?
How?
What?
Datasets
Attributes
Data Types
Items
Attributes
Links
Positions
Grids
Data and Dataset Types
Tables
Items
Attributes
Networks & Trees
Items (nodes)
Links
Attributes
Fields
Grids
Positions
Attributes
Geometry
Items
Positions
Clusters, Sets, Lists
Items
Dataset Types
Tables
Attributes (columns)
Items (rows)
Cell containing value
Networks
Link
Node (item)
Fields (Continuous)
Grid of positions
Cell
Attributes (columns)
Value in cell
Multidimensional Table
Key 1
Key 2
Value in cell
Attributes
Trees
Geometry (Spatial)
Position
Dataset Availability
Static
Dynamic
Attribute Types
Categorical
Ordered
Ordinal
Quantitative
Ordering Direction
Sequential
Diverging
Cyclic

Dataset Types



Dataset Types

Data and Dataset Types

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Availability

- Static

- Dynamic

Table

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	[illegible]	7/17/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	[illegible]	4-Not Specified	Small Pack	0.44	6/6/05
69	[illegible]	4-Not Specified	Wrap Bag	0.6	6/6/05
70	12/18/06	5-Low	Small Box	0.59	12/23/06
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06
96	4/17/05	2-High	Small Box	0.55	4/19/05
97	1/29/06	3-Medium	Small Box	0.38	1/30/06
129	11/19/08	5-Low	Small Box	0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

attribute

item

cell

Data Types

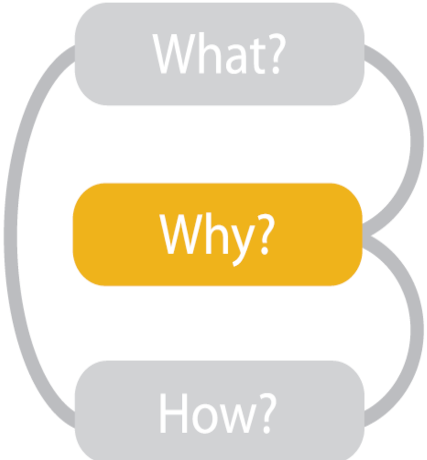
➔ Data Types

➔ Items ➔ Attributes ➔ Links ➔ Positions ➔ Grids

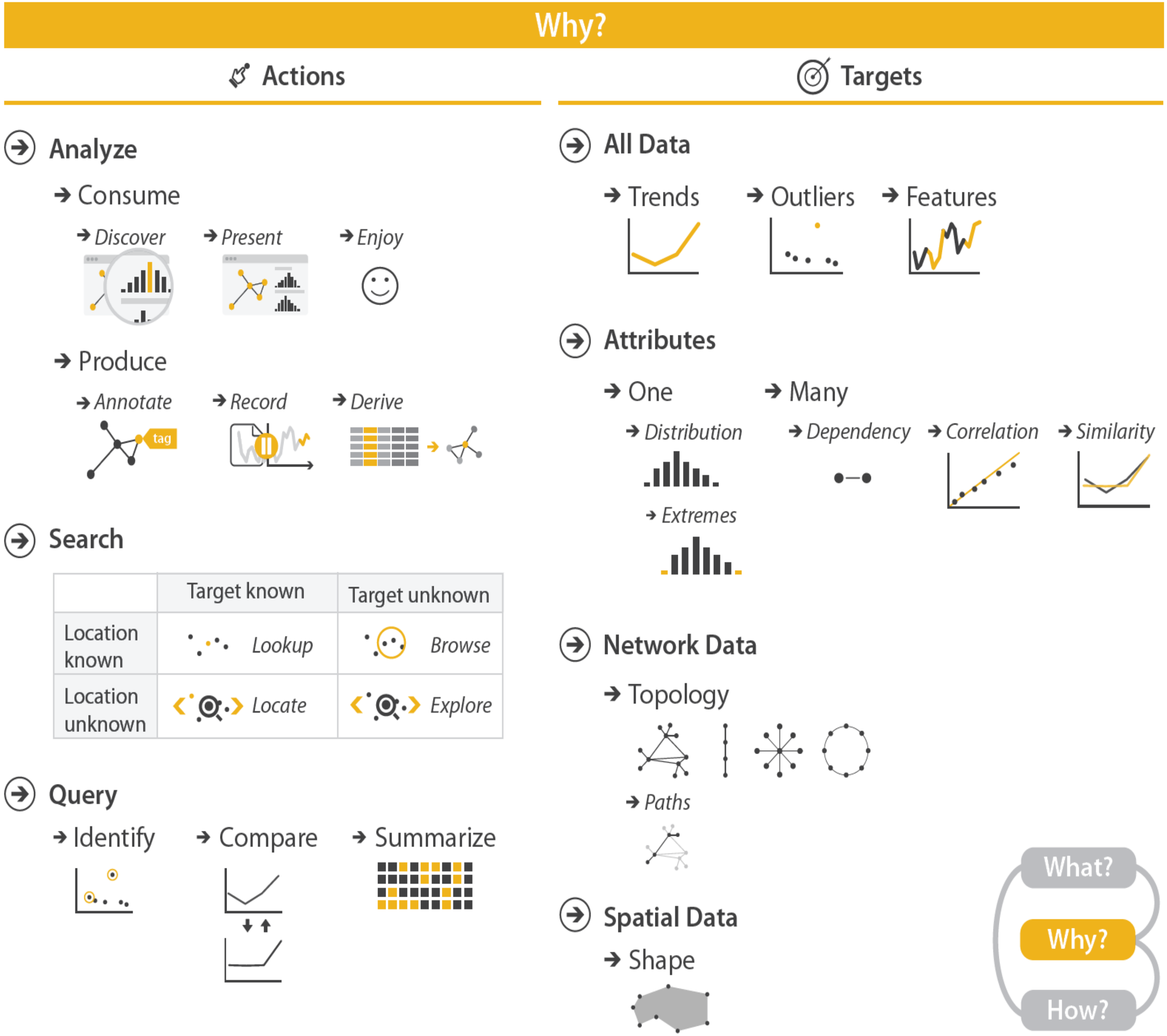
- **Item:** individual discrete entity
  - a row in a simple table or a node in a network.
  - ex: items may be people, stocks, coffeeshops, genes, or cities.
- **Attribute**: specific property that can be measured, observed, or logged
  - Synonyms for attribute are variable and data dimension)
  - ex: salary, price, number of sales, protein expression levels, or temperature.
- **Link**: specifies relationship between items, typically within a network.
- **Position:** spatial data, providing a location in two-dimensional (2D) or three-dimensional (3D) space.
  - Ex: a position might be a latitude–longitude pair describing a location on the Earth's surface or three numbers specifying a location within the region of space measured by a medical scanner.
- **Grid:** specifies the strategy for sampling continuous data in terms of both geometric and topological relationships between its cells.

Attribute Types

- Categorical data (nominal data)
  - No implicit ordering, often hierarchies of categories
  - Ex: fruit names, movie genres, file types, and city names
- Ordered data has well-defined ordering
  - **Ordinal** data has ordering but no arithmetic possible
  - **Quantitative** data allows computation
    - Integer and real numbers: measurements (temp,..), number of something, …

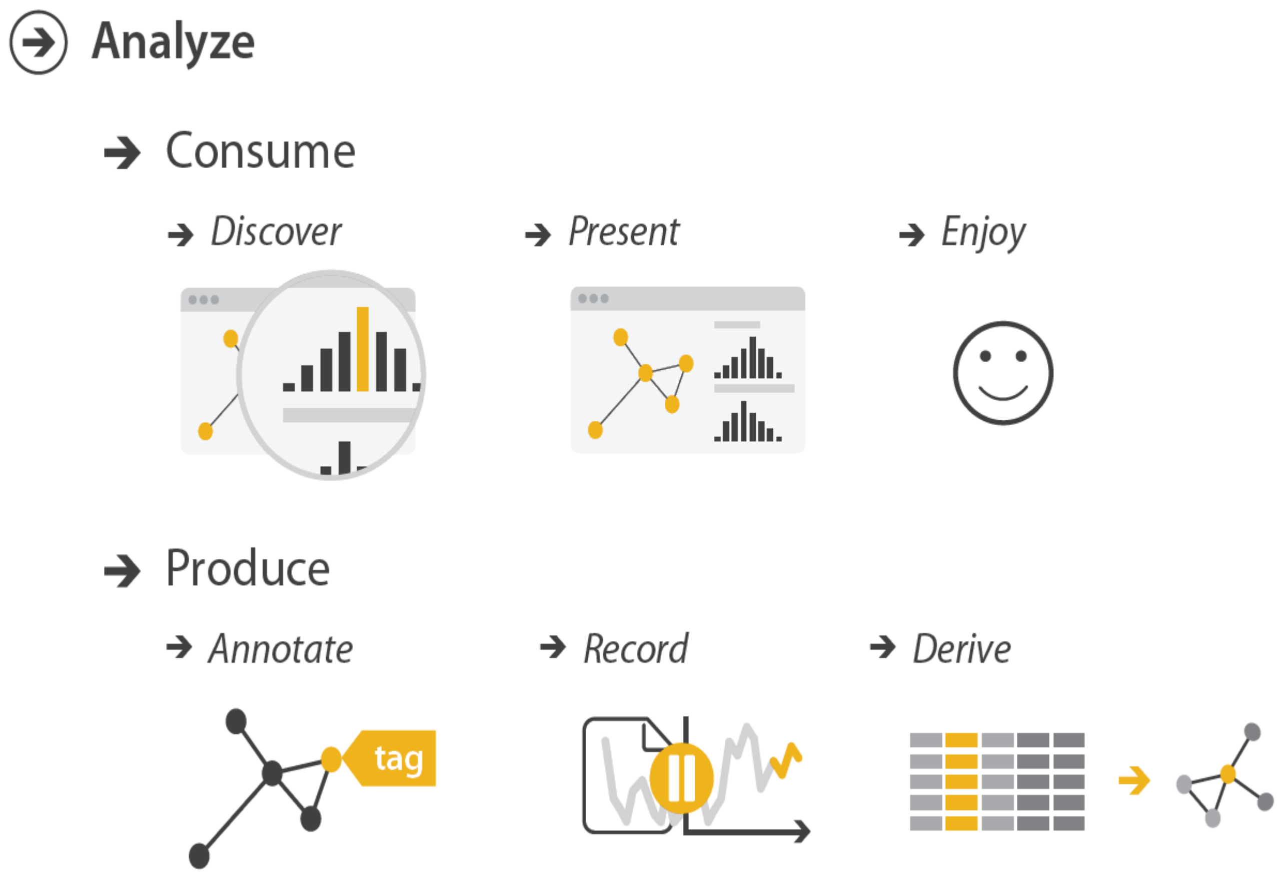


- {action, target} pairs
  - *Discover distribution*
  - *Compare trends*
  - *Locate outliers*
  - *Browse topology*



High-level Actions: Analyze

- Consume
  - Discover vs present
    - Classic split aka explore vs explain
  - Enjoy
    - Newcomer aka casual, social
- Produce
  - Annotate, record
  - Derive
    - Crucial design choice

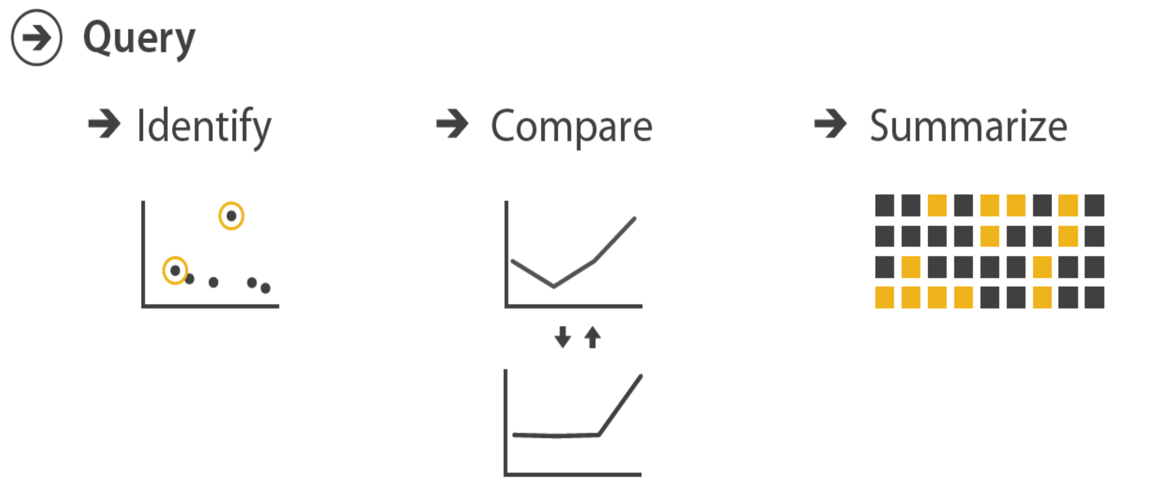


Actions: Mid-level Search, Low-level Query

- Search depends on what does user know?
  - Target?, location?
- Query follows search and further refines targets found
  - Identify a single target
  - Compare referes to multiple targest
  - Summarize refers to the full set of possible targets

Search

	Target known	Target unknown
Location known	*Lookup*	*Browse*
Location unknown	*Locate*	*Explore*



Why: Targets

➔ ALL DATA

➔ Trends

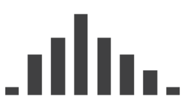
➔ Outliers

➔ Features

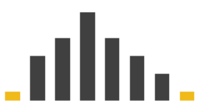
➔ ATTRIBUTES

➔ One

➔ *Distribution*



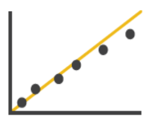
↓ *Extremes*



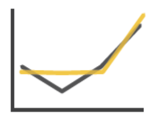
➔ Many

➔ *Dependency*

➔ *Correlation*

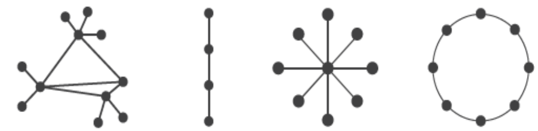


➔ *Similarity*



➔ NETWORK DATA

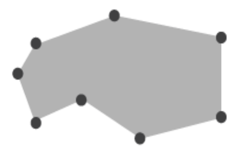
➔ Topology



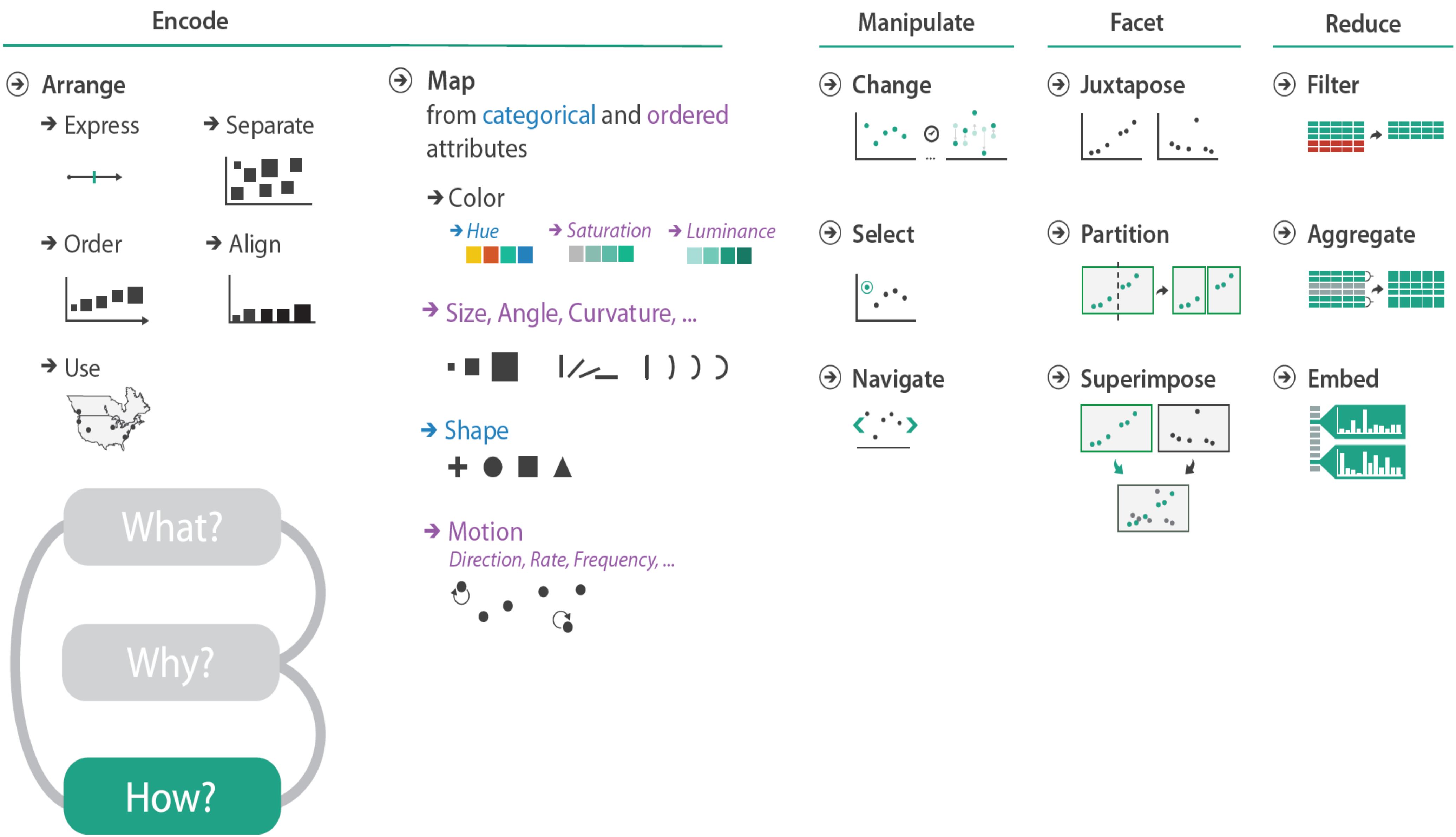
➔ *Paths*

➔ SPATIAL DATA

➔ Shape

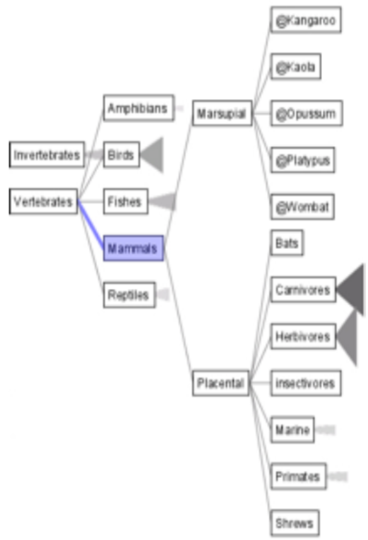


How?


Encode
Manipulate
Facet
Reduce
Arrange
Express
Separate
Order
Align
Use
Map
from categorical and ordered attributes
Color
Hue
Saturation
Luminance
Size, Angle, Curvature, ...
Shape
Motion
Direction, Rate, Frequency, ...
Change
Select
Navigate
Juxtapose
Partition
Superimpose
Filter
Aggregate
Embed
What?
Why?
How?

Analysis Example: Compare Techniques

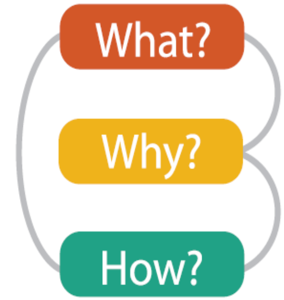
SpaceTree



TreeJuxtaposer

[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

[TreeJuxtaposer: Scalable Tree Comparison Using Focus +Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]



What?

→ **Tree**

Why?

→ **Actions**

→ Present → Locate → Identify



→ **Targets**

→ Path between two nodes

How?

→ **SpaceTree**

→ Encode → Navigate → Select → Filter → Aggregate





→ **TreeJuxtaposer**

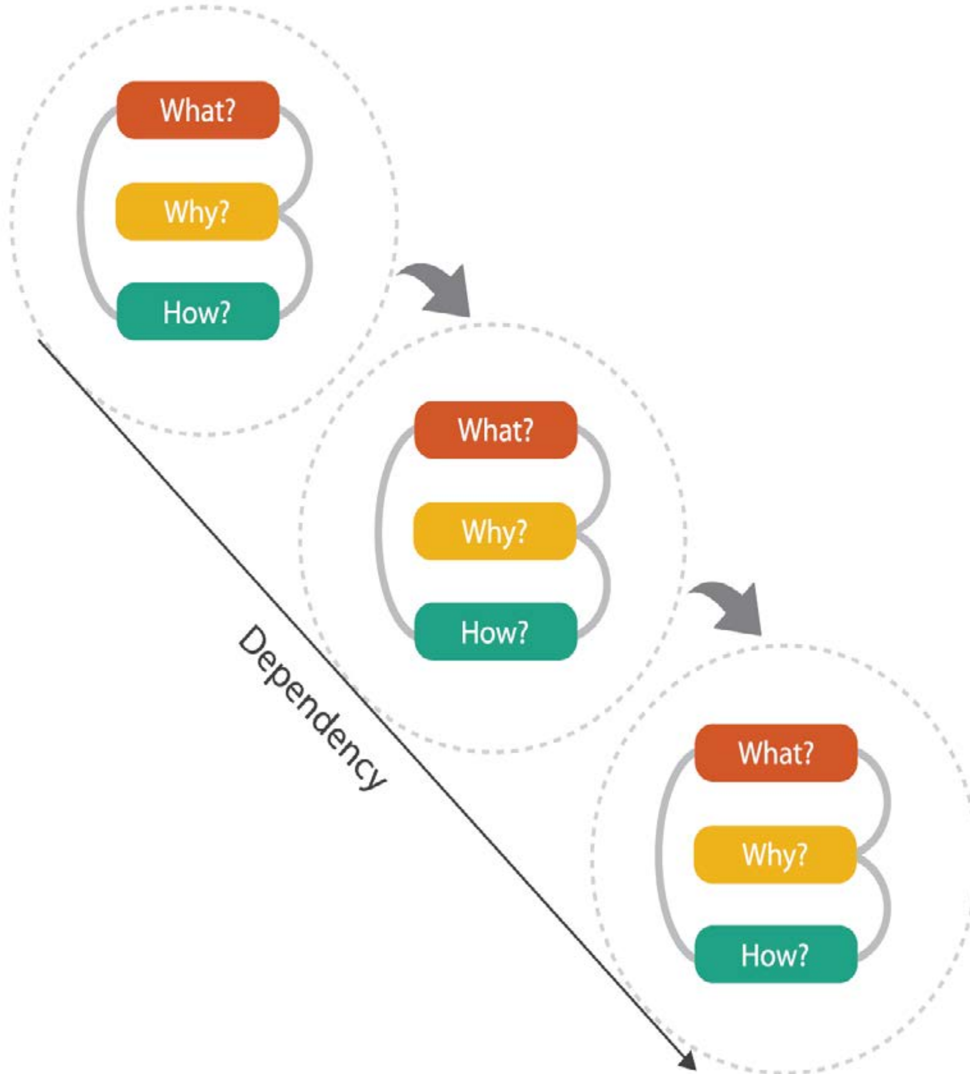
→ Encode → Navigate → Select → Arrange




Chained Sequences

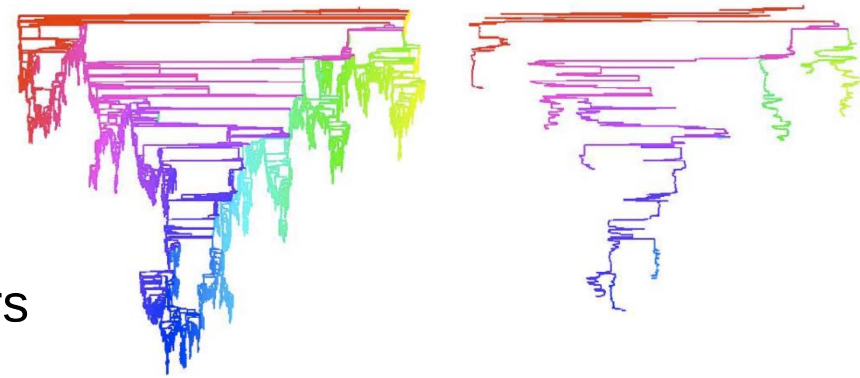
- Output of one is input to next
  - Express dependencies
  - Separate means from ends
    - A fundamental objective is an end that you are trying to achieve
    - A means objective is a way of achieving an end or fundamental objective
    - Focusing on ends rather than means helps find creative solutions to problems
    - Objectives only need to state the thing that matters, and what direction you'd like it to move

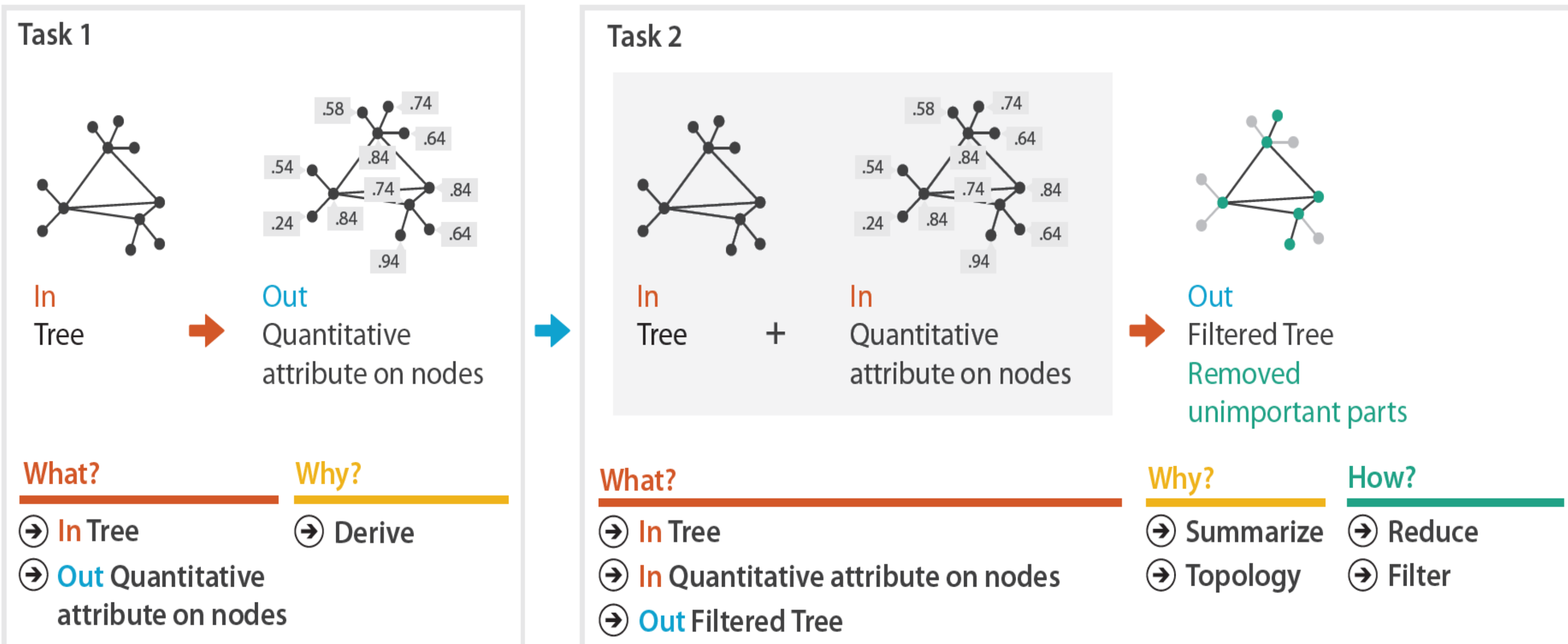
[http://www.structureddecisionmaking.org/steps/objectives/objectives2b/]



Analysis example: Derive one attribute

- Strahler number
  - Centrality metric for trees/networks
  - Derived quantitative attribute
  - Draw top 5K of 500K for good skeleton
- Fundamental objective: Topology …
- Means objective: compute Strahler numbers





[Using Strahler numbers for real time visual exploration of huge graphs. Auber. Proc. Intl. Conf. Computer Vision and Graphics, pp. 56–69, 2002.]

Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - Chap 2: What: Data Abstraction
  - Chap 3: Why: Task Abstraction
- A Multi-Level Typology of Abstract Visualization Tasks. Brehmer and Munzner. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis) 19:12 (2013), 2376–2385.
- Low-Level Components of Analytic Activity in Information Visualization. Amar, Eagan, and Stasko. Proc. IEEE InfoVis 2005, p 111–117.
- A taxonomy of tools that support the fluent and flexible use of visualizations. Heer and Shneiderman. Communications of the ACM 55:4 (2012), 45–54.
- Rethinking Visualization: A High-Level Taxonomy. Tory and Möller. Proc. IEEE InfoVis 2004, p 151–158.

End

Visualization

Introduction to Information Visualization II

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Additional Lecture

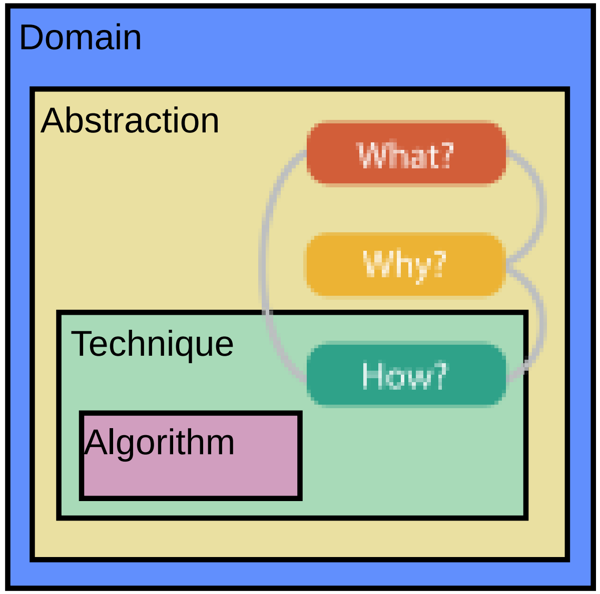
- Wednesday, 29.4. at 15:15 (R015/B11)

Acknowledgements

- First two lectures are based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.
- Further based on the books and lectures by
  - Robert Spence: Information Visualization 2nd Ed
  - Colin Ware: Information Visualization 2nd Ed.
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Silvia Miksch, TU Wien
  - Jeffrey Heer, Stanford University

Analysis Framework: Four Levels, Four Questions

- *Domain* situation
  - Who are the target users?
- *Abstraction*
  - Translate from specifics of domain to vocabulary of visualization
  - **What** is shown? **data abstraction**
  - **Why** is the user looking at it? **task abstraction**
- *Technique (Munzner: Idiom)*
  - **How** is it shown?
    - **Visual encoding technique**: how to draw
    - **Interaction technique**: how to manipulate
- *Algorithm*
  - Efficient computation

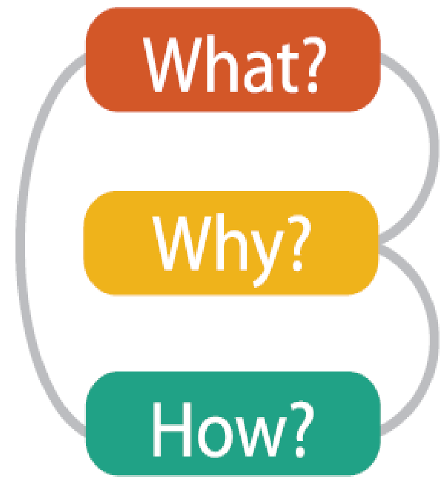


[A Nested Model of Visualization Design and Validation.
Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]

[A Multi-Level Typology of Abstract Visualization Tasks
Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

Analysis: What, Why, and How

- What is shown?
  - **Data** abstraction
- Why is the user looking at it?
  - **Task** abstraction
- How is it shown?
  - **Technique**: visual encoding and interaction



- Abstract vocabulary avoids domain-specific terms
  - Translation process iterative and often complex
- What-why-how analysis framework helps to think systematically about design space

What?

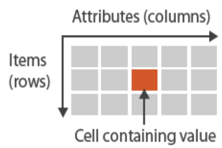
Datasets

Data Types
- Items
- Attributes
- Links
- Positions
- Grids

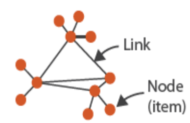
Data and Dataset Types

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

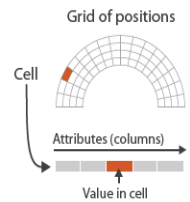
Dataset Types
- Tables



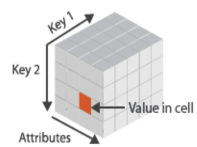
- Networks



- Fields (Continuous)



- *Multidimensional Table*



- *Trees*

- Geometry (Spatial)

Dataset Availability
- Static

- Dynamic

Attributes

Attribute Types
- Categorical

- Ordered
  - *Ordinal*

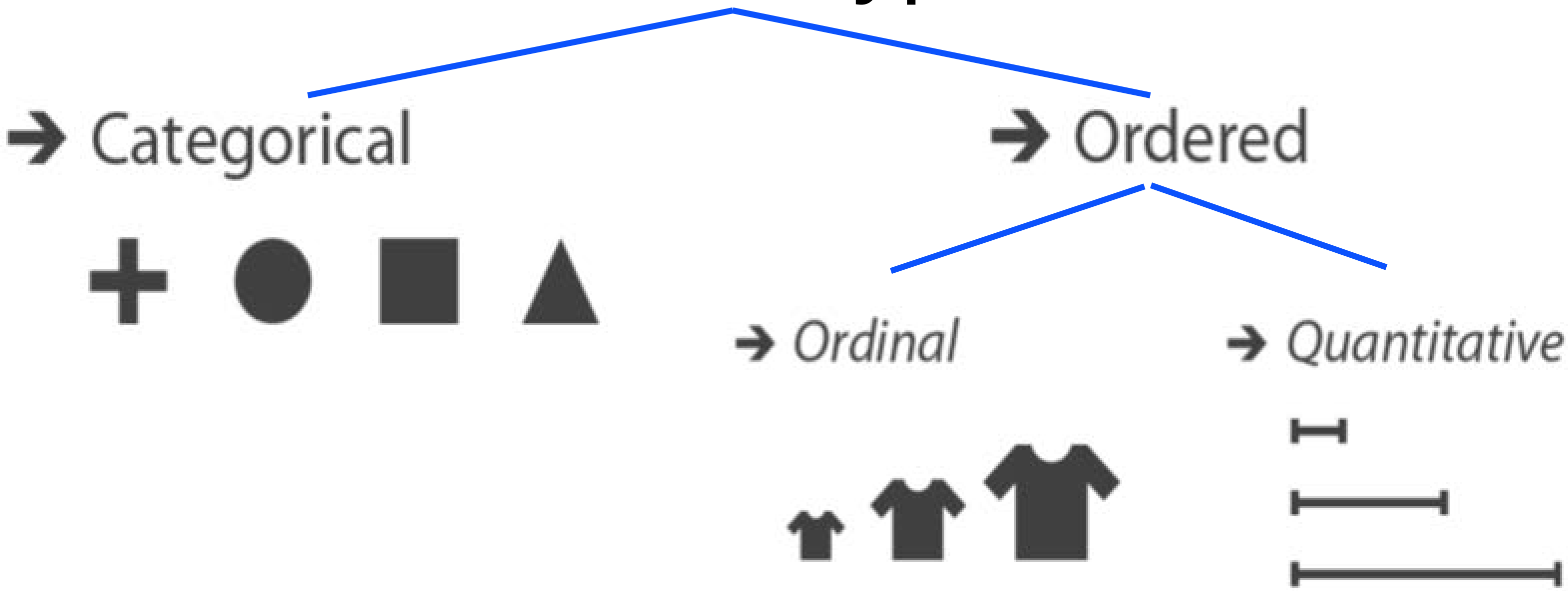
  - *Quantitative*

Ordering Direction
- Sequential

- Diverging

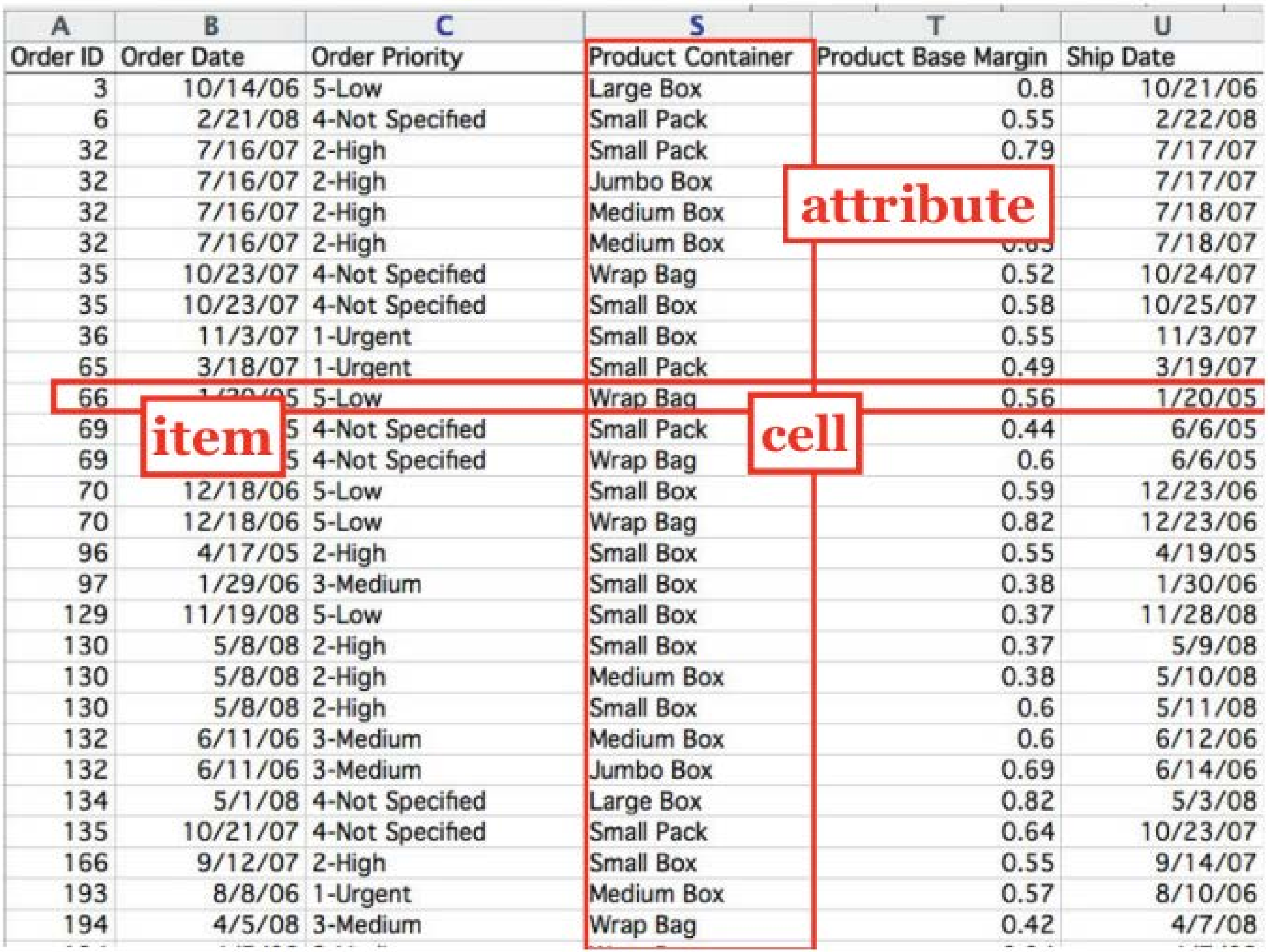
- Cyclic

Attribute Types



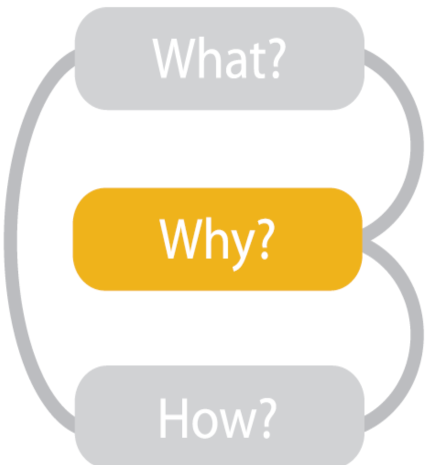
- Categorical data (nominal data)
  - No implicit ordering, often hierarchies of categories
  - Ex: fruit names, movie genres, file types, and city names
- Ordered data has well-defined ordering
  - **Ordinal** data has ordering but no arithmetic possible
  - **Quantitative** data allows computation
    - Integer and real numbers: measurements (temp,..), number of something, …

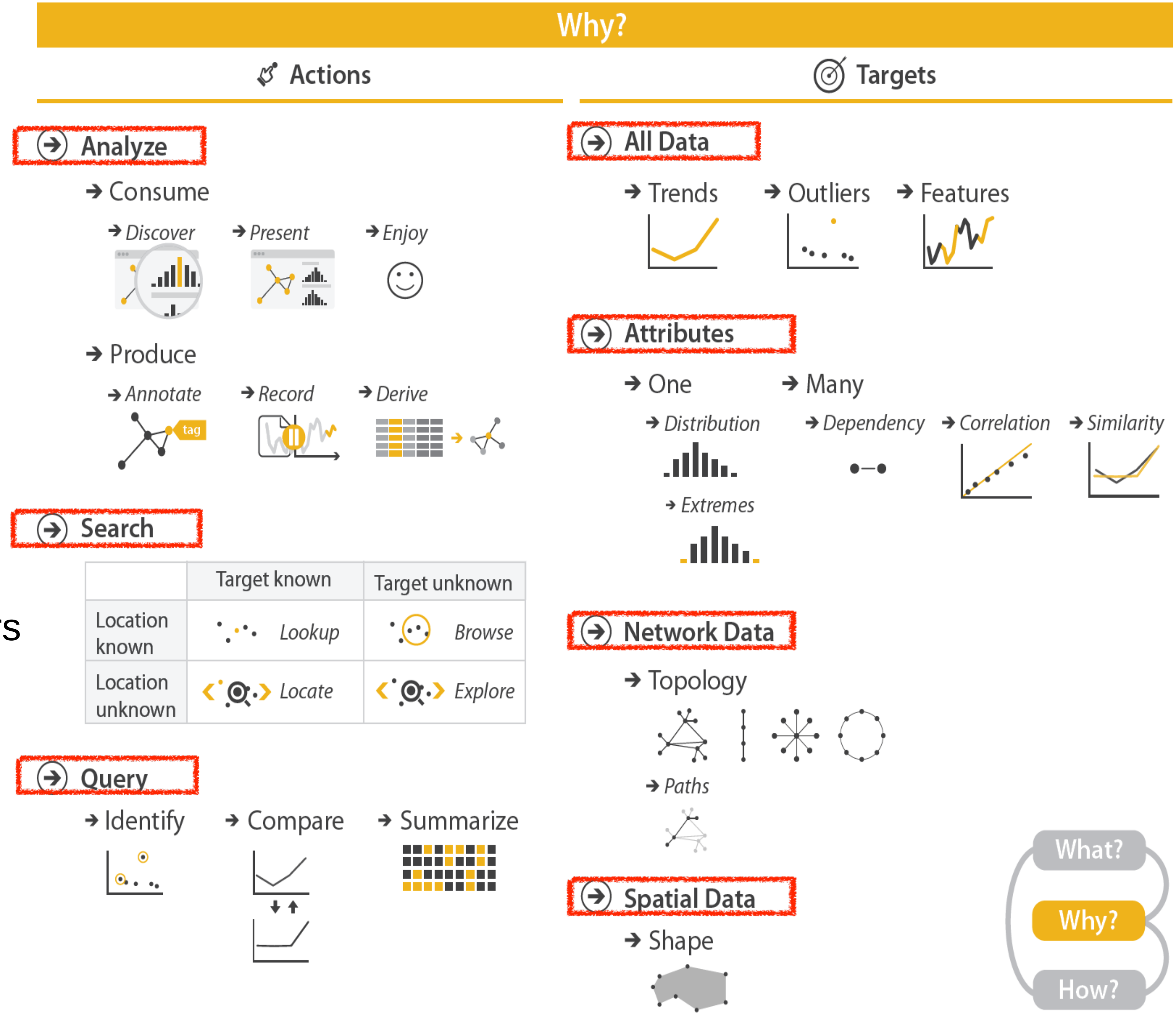
Table



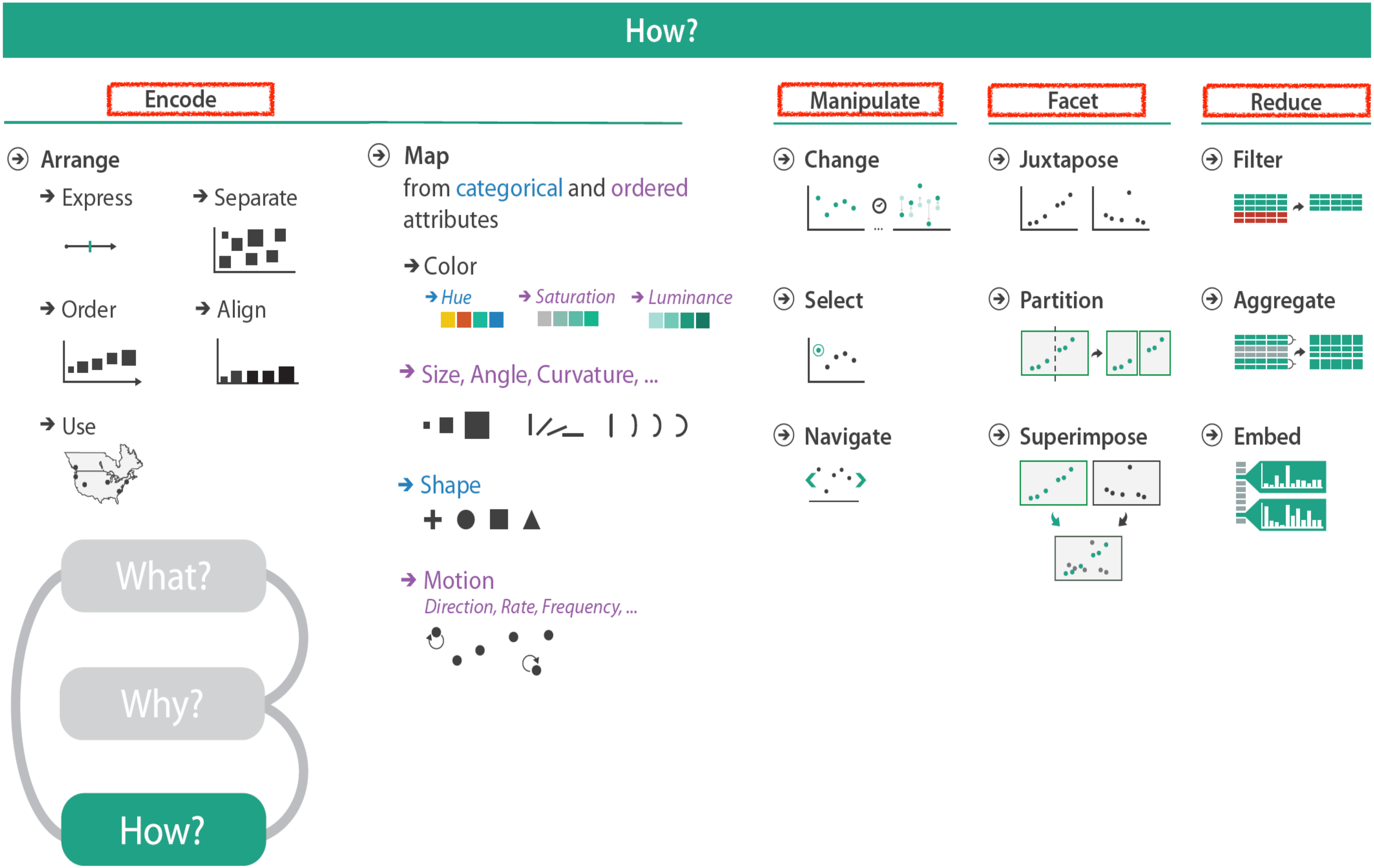
A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	[illegible]	7/17/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	[illegible]	5-Low	Wrap Bag	0.56	1/20/05
69	[illegible]	4-Not Specified	Small Pack	0.44	6/6/05
69	[illegible]	4-Not Specified	Wrap Bag	0.6	6/6/05
70	12/18/06	5-Low	Small Box	0.59	12/23/06
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06
96	4/17/05	2-High	Small Box	0.55	4/19/05
97	1/29/06	3-Medium	Small Box	0.38	1/30/06
129	11/19/08	5-Low	Small Box	0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

In each cell, an attribute value is stored

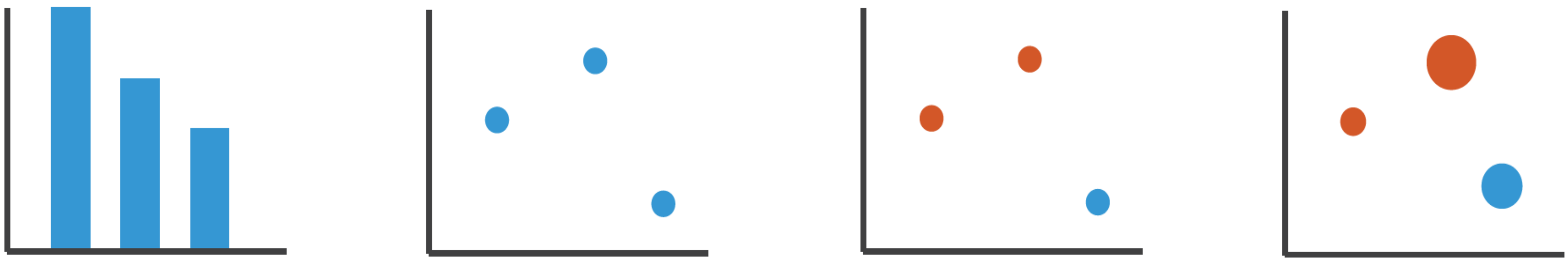




- {action, target} pairs
  - *Discover distribution*
  - *Compare trends*
  - *Locate outliers*
  - *Browse topology*


How?
Encode
Manipulate
Facet
Reduce
Arrange
Express
Separate
Order
Align
Use
Map
from categorical and ordered attributes
Color
Hue
Saturation
Luminance
Size, Angle, Curvature, ...
Shape
Motion
Direction, Rate, Frequency, ...
Change
Select
Navigate
Juxtapose
Partition
Superimpose
Filter
Aggregate
Embed
What?
Why?
How?

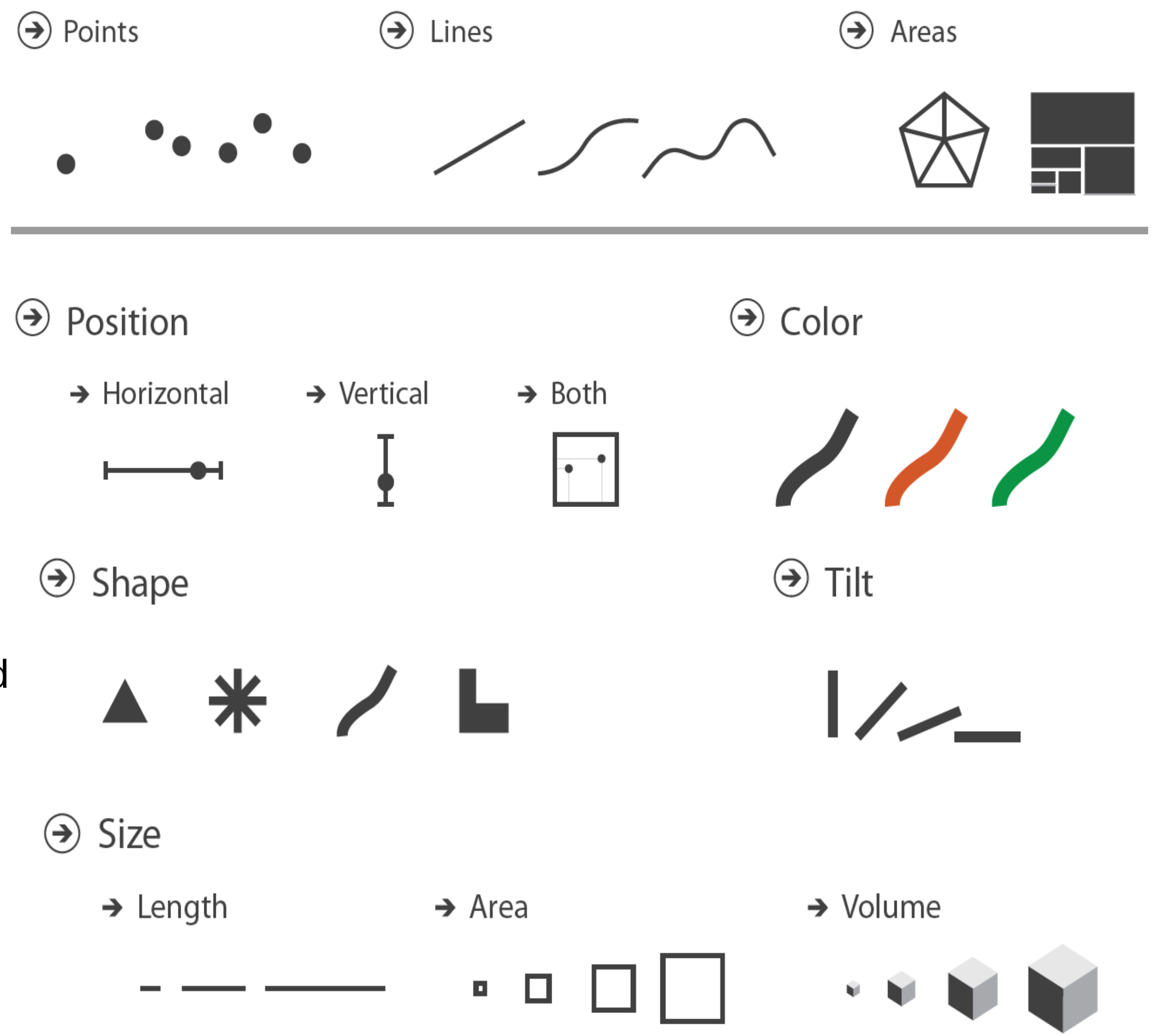
Visual Encoding



- Analyze visualization techniques and ask questions:
  - How did we decide to use those particular visual channels?
  - How many more visual channels are there?
  - What kinds of information and how much information can each channel encode?
  - Are some channels better than others?
  - Can all of the channels be used independently or do they interfere with each other?

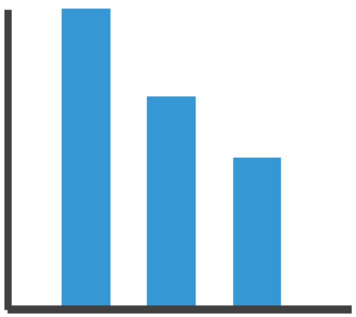
Definitions: Marks and Channels

- Marks
  - Geometric primitives
- Channels (visual attributes)
  - Control appearance of marks
  - Can redundantly code using multiple channels
- Some interactions
  - Point marks only convey position
    - Can be size and shape coded
  - Line marks convey position and length
    - Can only be size coded in 1D (width)
  - Area marks fully constrained
    - Cannot be size or shape coded



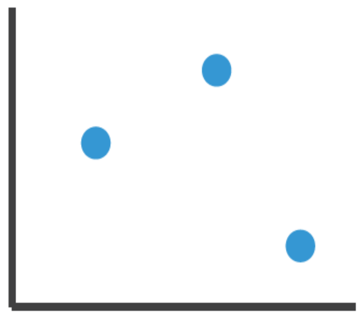
Visual encoding

- Analyze visualization technique structure as combination of marks and channels



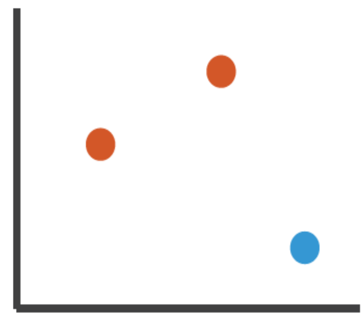
1:
Vertical position

Mark: line



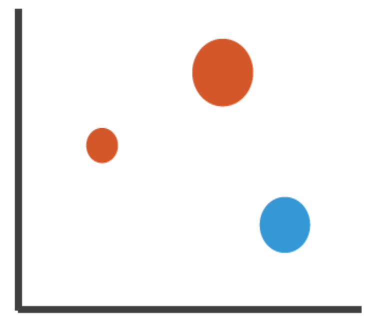
2:
Vertical position
Horizontal position

Mark: point



3:
Vertical position
Horizontal position
Color hue

Mark: point



4:
Vertical position
Horizontal position
Color hue
Size (area)

Mark: point

Channels: Effectiveness Rankings

Magnitude Channels: Ordered Attributes

Position on common scale

Position on unaligned scale

Length (1D size)

Tilt/angle

Area (2D size)

Depth (3D position)

Color luminance

Color saturation

Curvature

Volume (3D size)

Most — Effectiveness — Least

Same

Same

Identity Channels: Categorical Attributes

Spatial region

Color hue

Motion

Shape

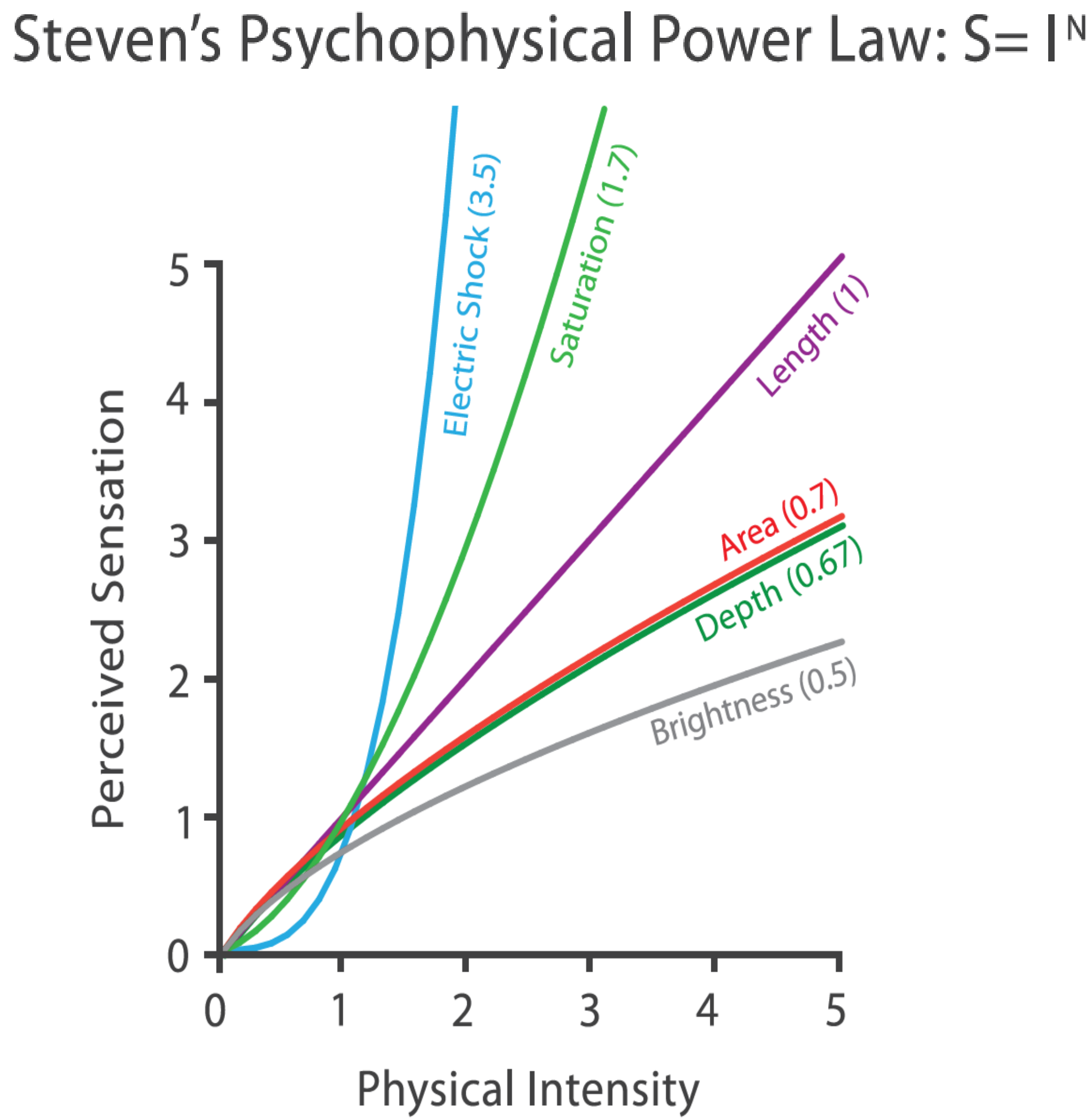
Effectiveness and Expressiveness Principles

- Effectiveness principle
  - Encode most important attributes with highest ranked channels
- Expressiveness principle
  - Match channel and data characteristics

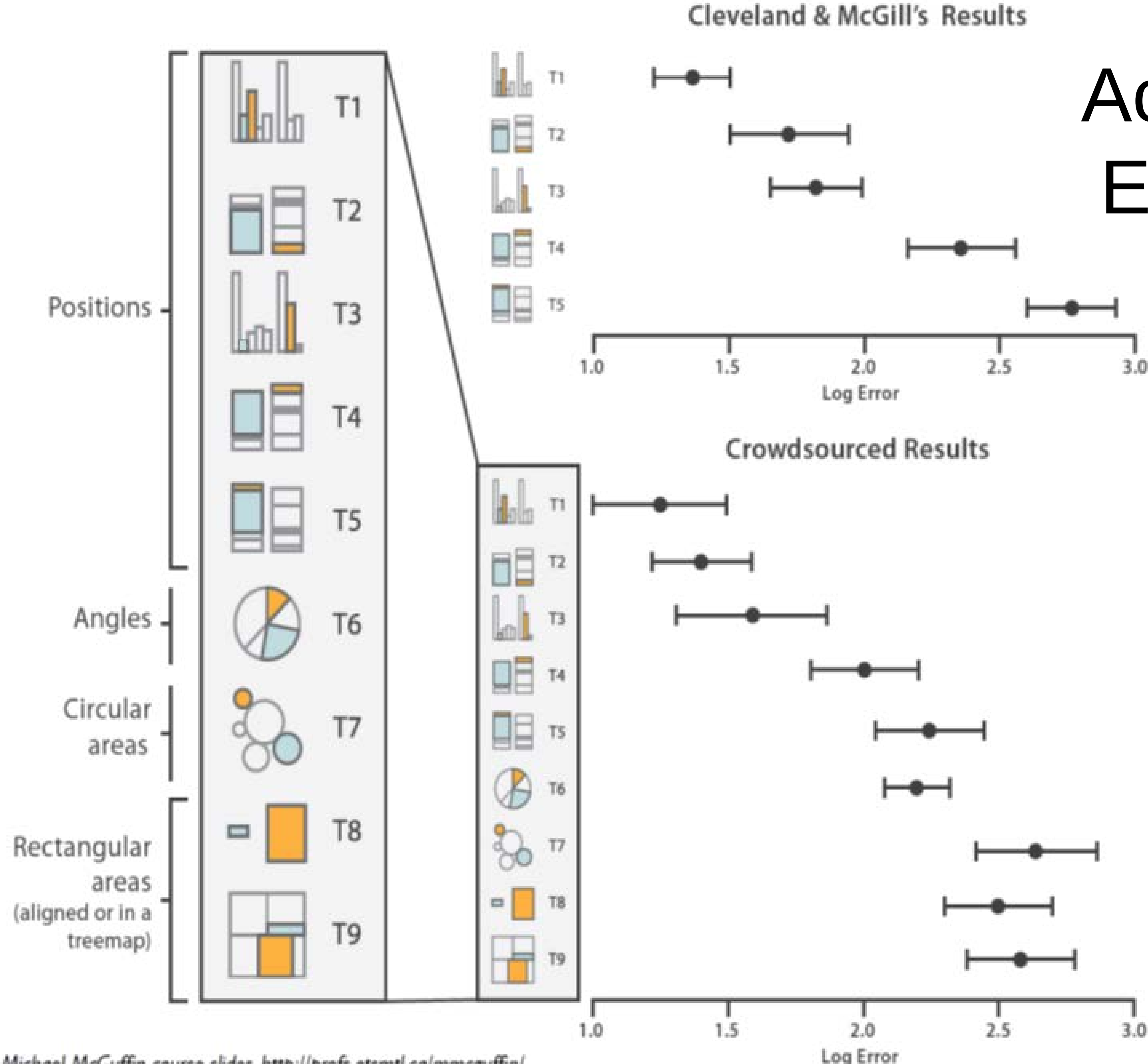
[*Automating the Design of Graphical Presentations of Relational Information.* Mackinlay. ACM Trans. on Graphics (TOG) 5:2 (1986), 110–141.]

- Rankings: where do they come from?
  - Accuracy
  - Discriminability
  - Separability
  - Popout

Accuracy: Fundamental Theory



Accuracy: Vis Experiments

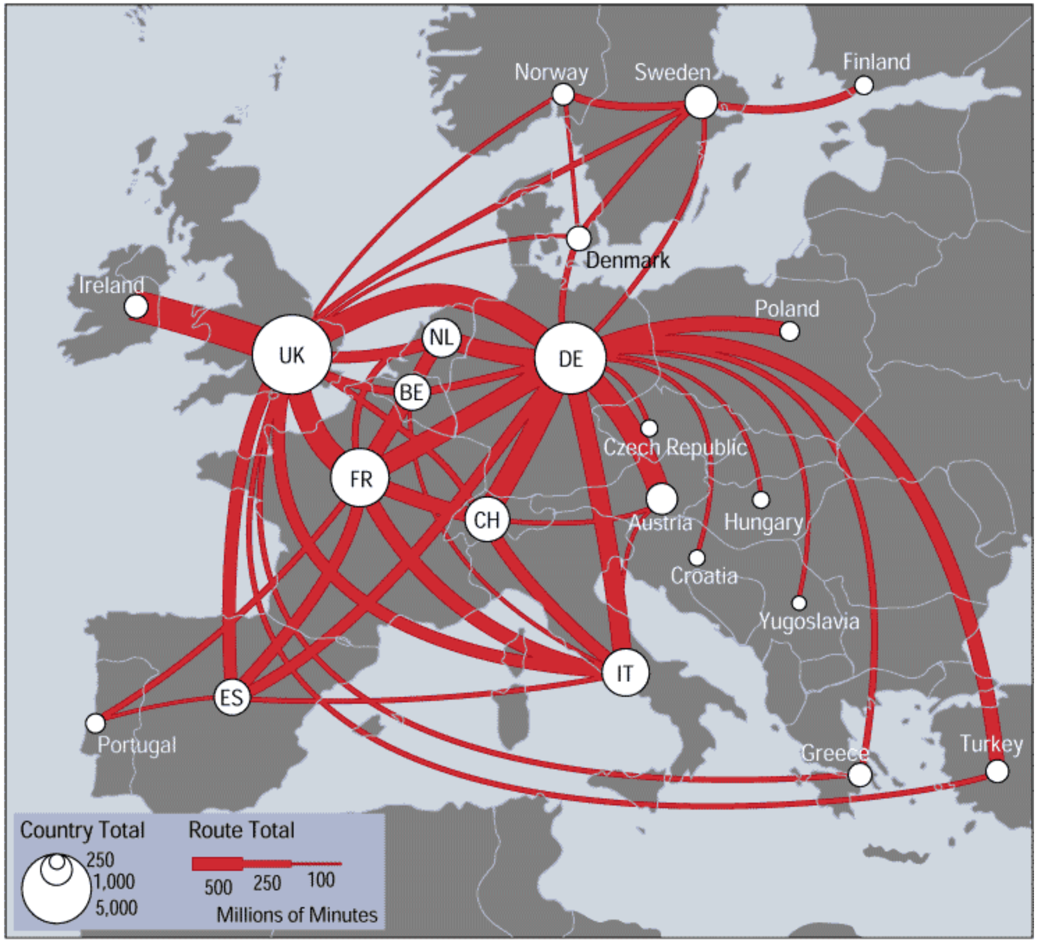


[Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. Heer and Bostock. Proc ACM Conf. Human Factors in Computing Systems (CHI) 2010, p. 203–212.]

after Michael McGuffin course slides, http://profs.etsmtl.ca/mmcguffin/

Discriminability: How many usable steps?

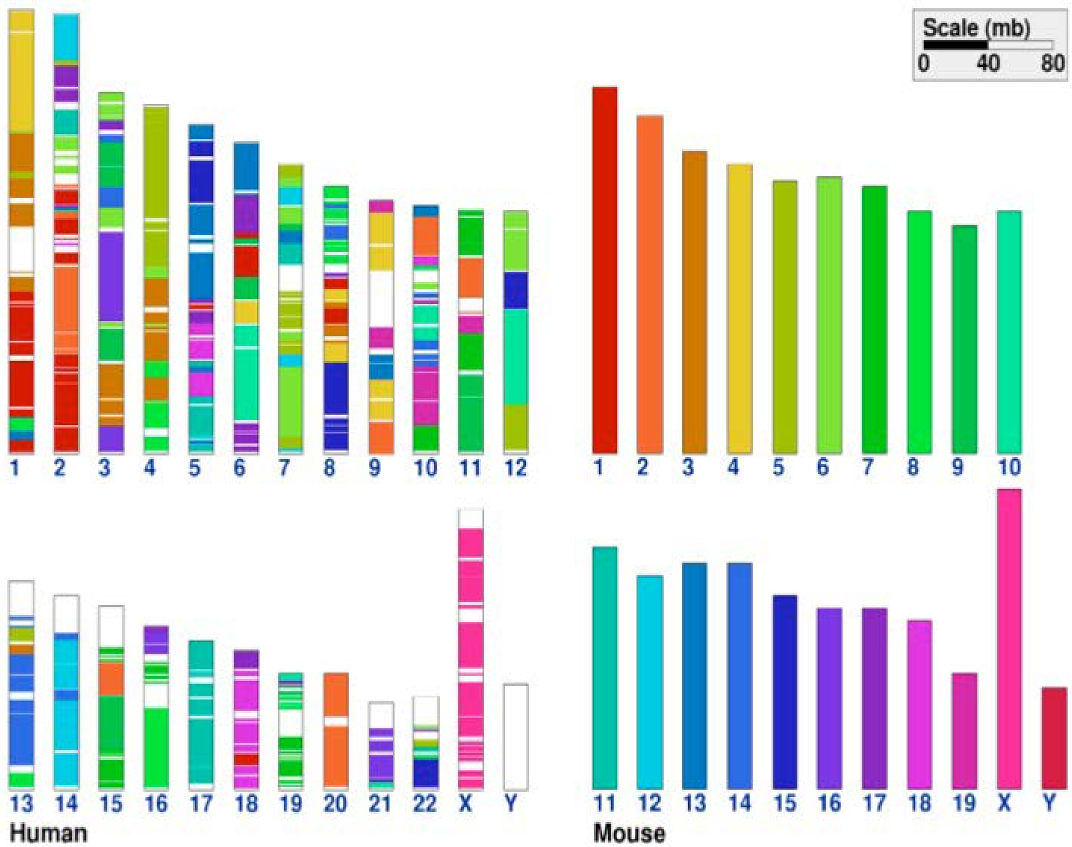
- Linewidth: only a few



[mappa.mundi.net/maps/maps 014/telegeography.html]

Discriminability of Colors for Categories

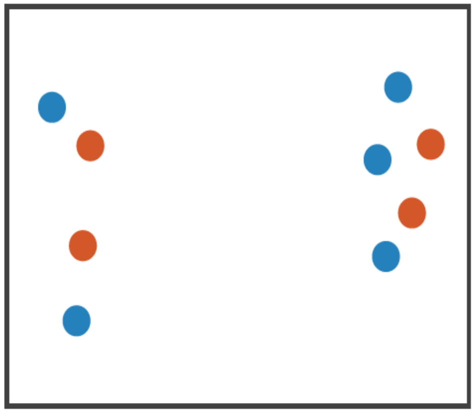
- Noncontiguous small regions of color: only 6-12 bins



Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. Bioinformatics 2007

Separability vs. Integrality

Position
+ Hue (Color)



Fully separable

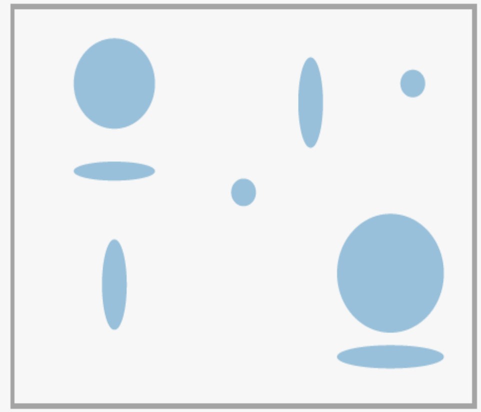
2 groups each

Size
+ Hue (Color)

Some interference

2 groups each

Width
+ Height



Some/significant interference

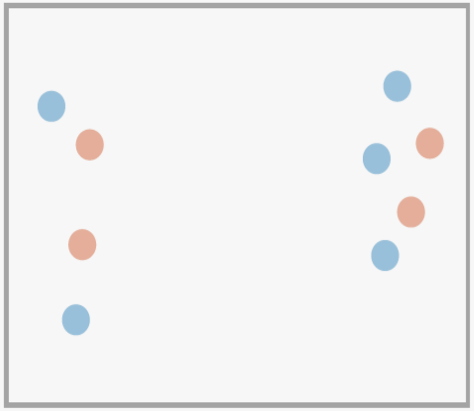
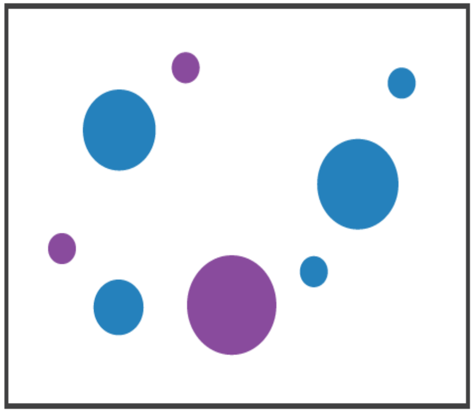
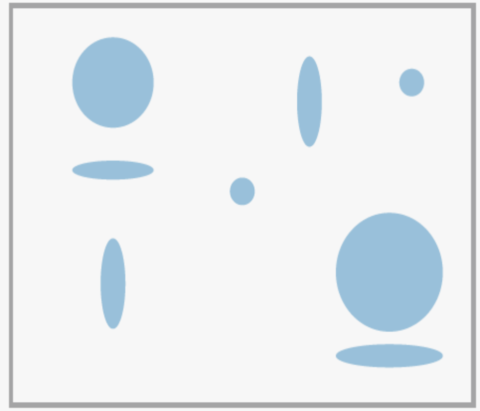
3 groups total: integral area

Red
+ Green

Major interference

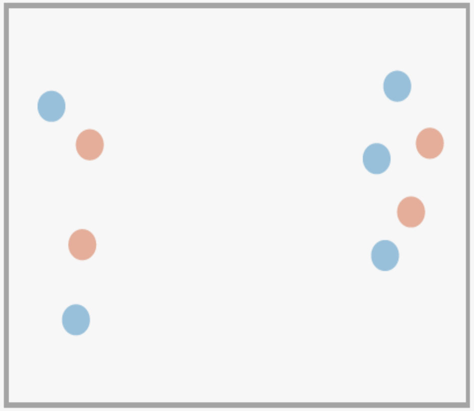
4 groups total: integral hue

Separability vs. Integrality

Position + Hue (Color)	Size + Hue (Color)	Width + Height	Red + Green
			
Fully separable	Some interference	Some/significant interference	Major interference
2 groups each	2 groups each	3 groups total: integral area	4 groups total: integral hue

Separability vs. Integrality

Position
+ Hue (Color)



Fully separable

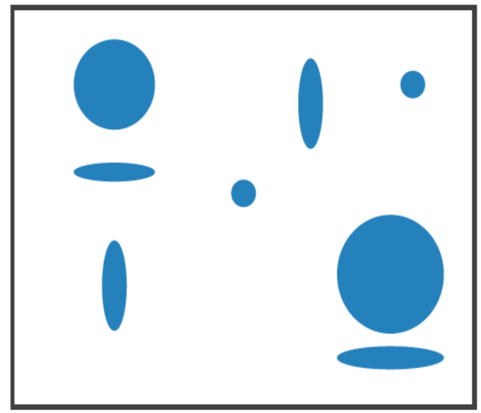
2 groups each

Size
+ Hue (Color)

Some interference

2 groups each

Width
+ Height



Some/significant interference

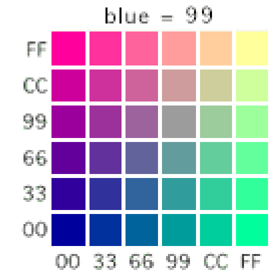
3 groups total:
integral area

Red
+ Green

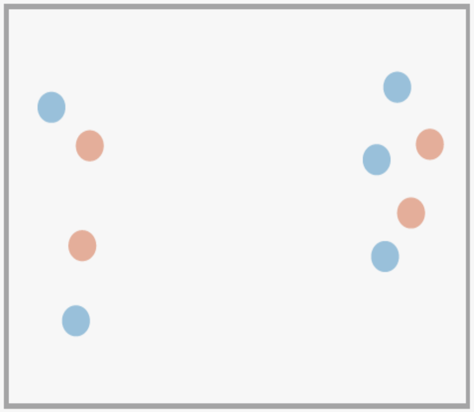
Major interference

4 groups total:
integral hue

Separability vs. Integrality



Position
+ Hue (Color)



Fully separable

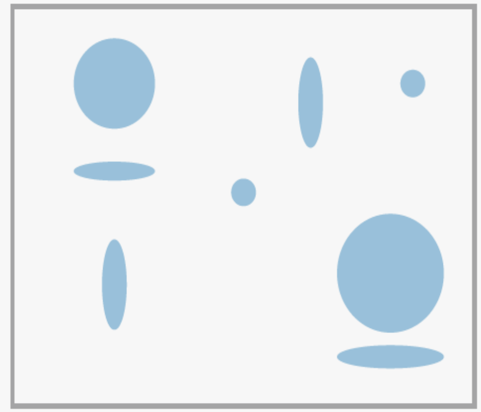
2 groups each

Size
+ Hue (Color)

Some interference

2 groups each

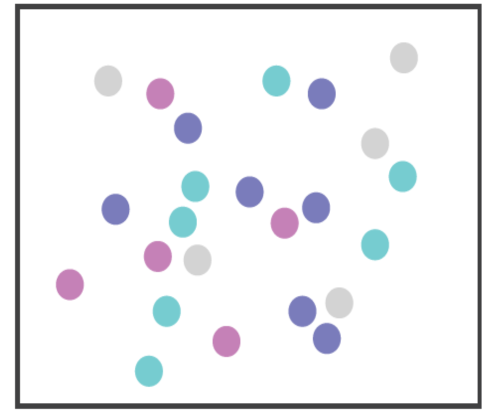
Width
+ Height



Some/significant interference

3 groups total: integral area

Red
+ Green

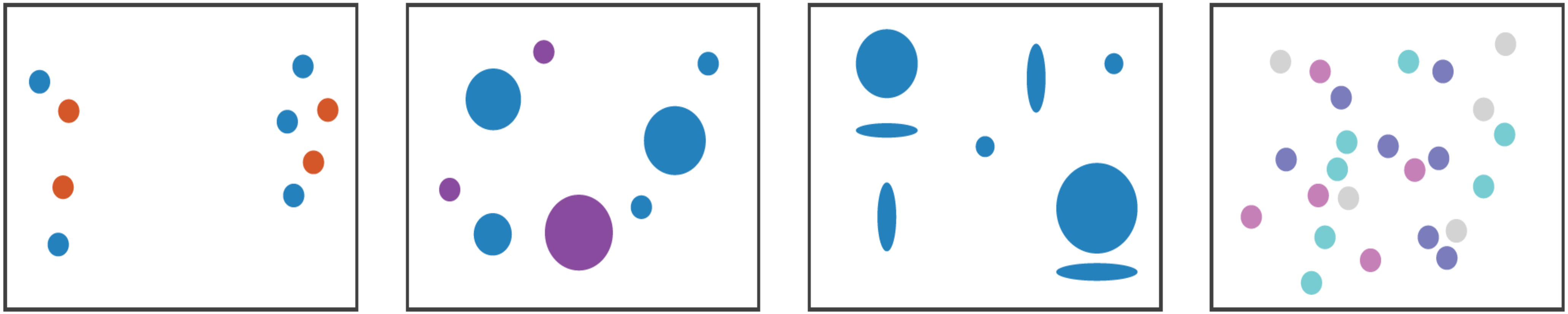


Major interference

4 groups total: integral hue (Blue is constant e.g. Blue=99)

Separability vs. Integrality

Position + Hue (Color) | Size + Hue (Color) | Width + Height | Red + Green



Fully separable | Some interference | Some/significant interference | Major interference

2 groups each

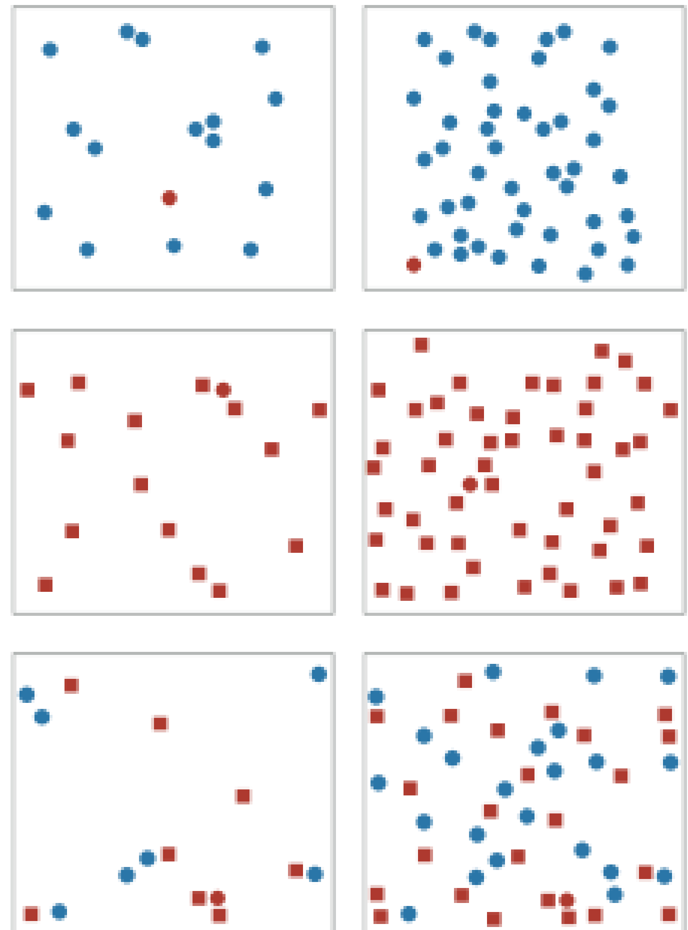
2 groups each

3 groups total: integral area

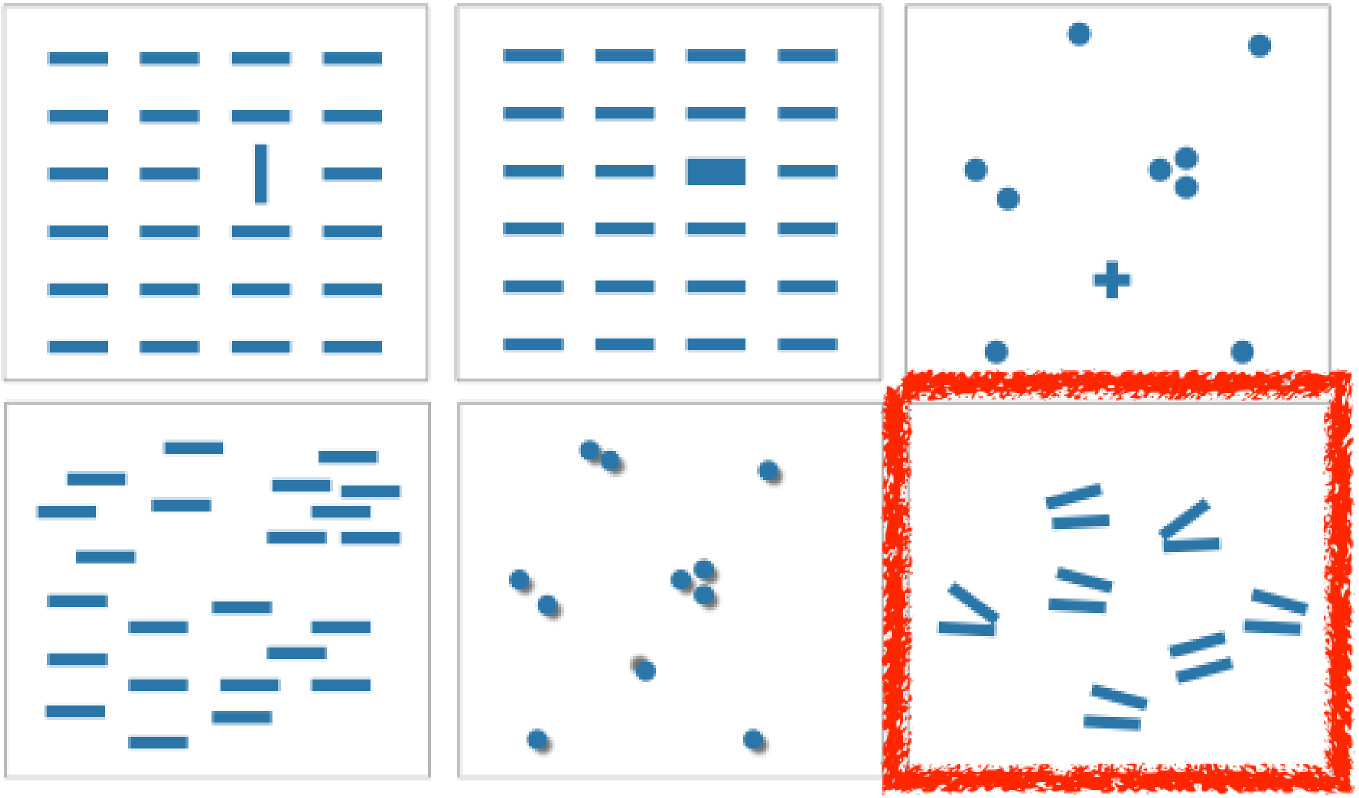
4 groups total: integral hue

Preattentive Processing (Popout)

- Find the red dot
  - How long does it take?
- Parallel processing on many individual channels
  - Speed independent of distractor count
  - Speed depends on channel and amount of difference from distractors
- Serial search for (almost all) combinations
  - Speed depends on number of distractors



Popout



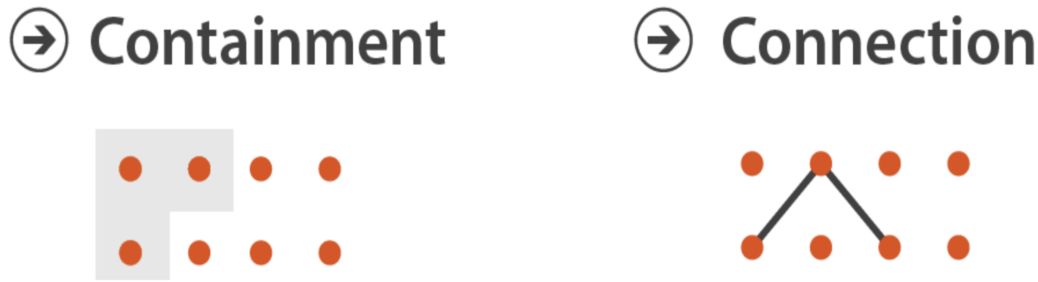
- Many channels: tilt, size, shape, proximity, shadow direction, ...
- But not all: parallel line pairs do not pop out from tilted pairs

Popout

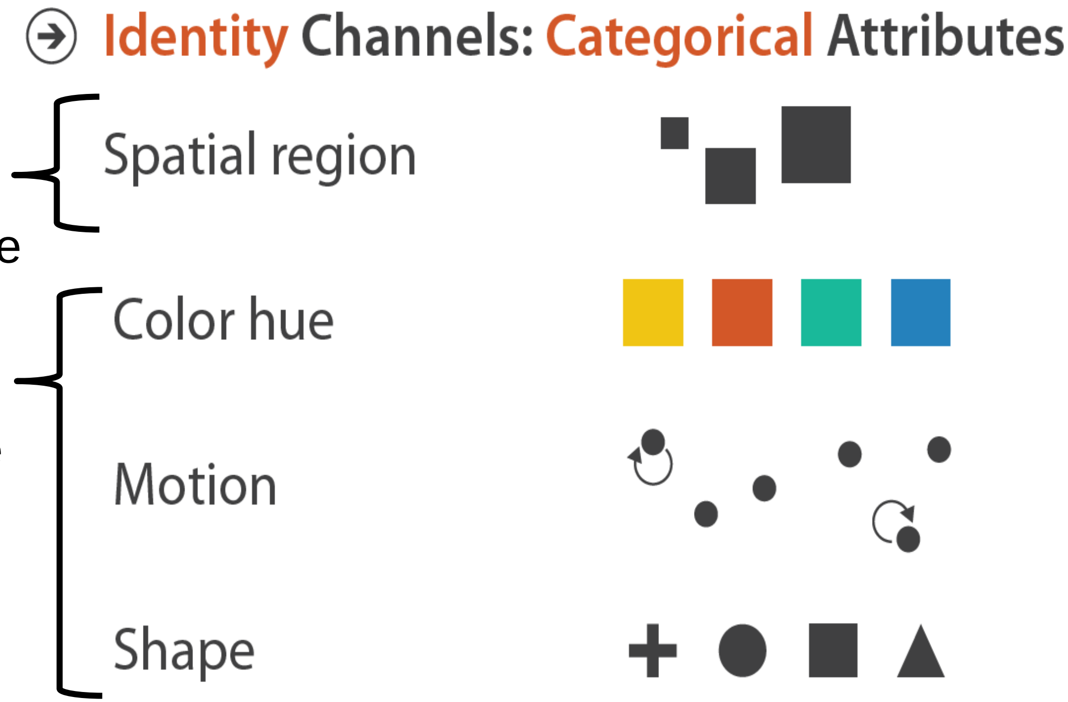
- Munzner: "*Most pairs of channels do not support popout, but a few pairs do: one example is space and color, and another is motion and shape. Popout is definitely not possible with three or more channels. As a general rule, vis designers should only count on using popout for a single channel at a time.*"

Grouping (Show Items that belong together)

- Containment
  - Area mark as link
- Connection
  - Line mark as link

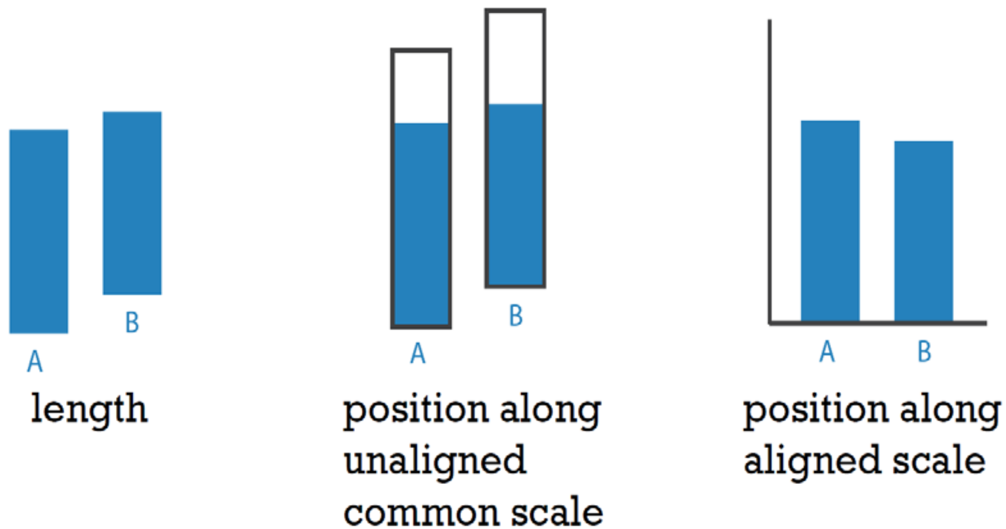


- Use group ID as a categorical attribute
  - Proximity
    - Put items in the same spatial region
  - Similarity
    - Color, motion, shape



Relative vs. Absolute Judgements

- Perceptual system mostly operates with relative judgements, not absolute
  - Accuracy increases with common frame/scale and alignment
  - Weber’s Law: ratio of increment to background is constant
    - Here: background is length of one bar
    - Filled rectangles differ in length by 15%, a difficult judgement
    - White rectangles differ in length by 30%, easier judgement



after [Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Cleveland and McGill. Journ. American Statistical Association 79:387 (1984), 531–554.]

Psychophysiological Measures

- JND: Just noticeable difference
  - Increment where humans detects change
- Weber's law
  - JND relative to a given cue is a constant fraction of cue
  - This is not generally true, but often a good approximation
  - Example: Intensity change relative to background intensity

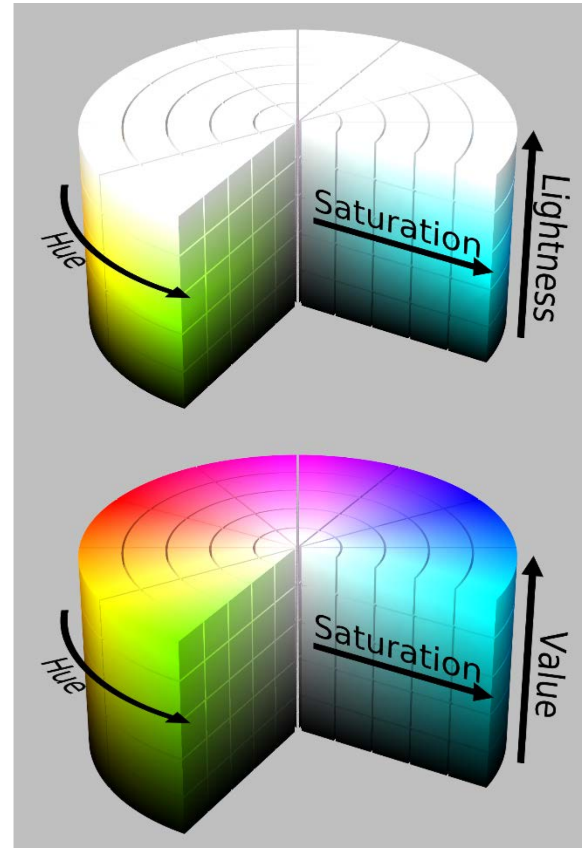
$$\frac{\Delta I}{I} = k$$

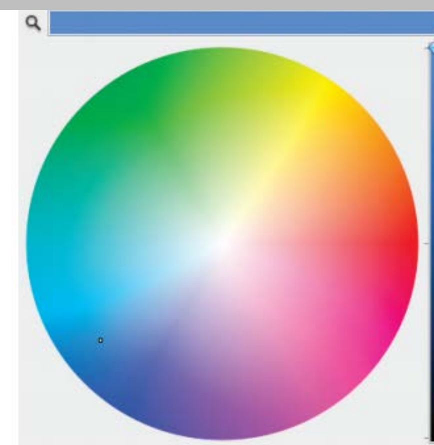
Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - Chap 5: Marks and Channels
- On the Theory of Scales of Measurement. Stevens. Science 103:2684 (1946), 677–680.
- Psychophysics: Introduction to its Perceptual, Neural, and Social Prospects. Stevens. Wiley, 1975.
- Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Cleveland and McGill. Journ. American Statistical Association 79:387 (1984), 531–554.
- Perception in Vision. Healey. http://www.csc.ncsu.edu/faculty/healey/PP
- Visual Thinking for Design. Ware. Morgan Kaufmann, 2008.
- Information Visualization: Perception for Design, 3rd edition. Ware. Morgan Kaufmann /Academic Press, 2004.

HSL Color Model

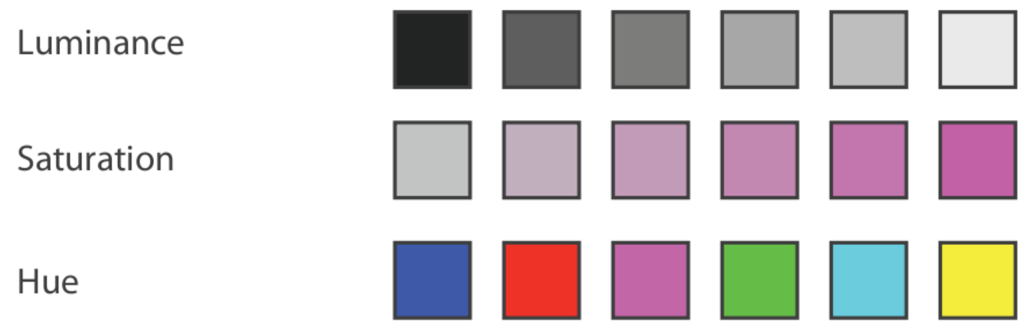
- Hue–Saturation–Lightness (HSL)
  - Intuitive color system
  - Heavily used by artists and designers
  - Similar to HSV – V (grayscale value) and L are linearly related
- Hue
  - Defines pure colors that are not mixed with white or black
- Saturation
  - Amount of white mixed with a pure color
- Lightness
  - Amount of black mixed with a color
- Color pickers
  - Disk with white at the center
  - Hue axis wrapped around the outside
  - Separate linear control for the amount of darkness versus lightness



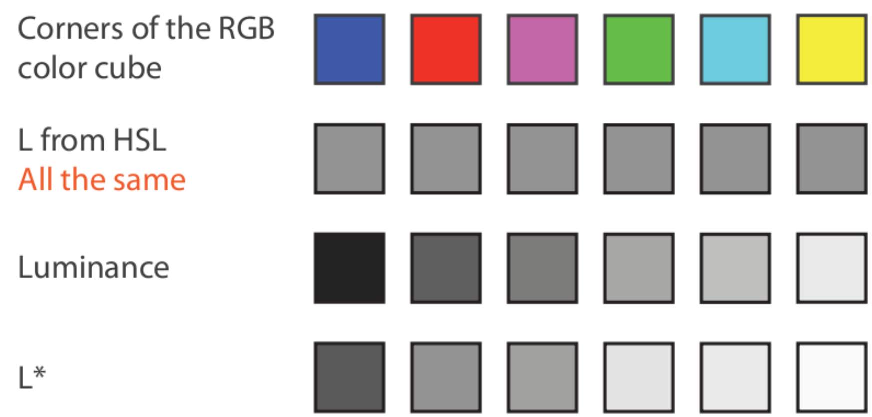


Color: Luminance, Saturation, Hue

- 3 channels
  - Ordered attributes (how much)
    - Luminance
    - Saturation
  - Categorical (what)
    - Hue



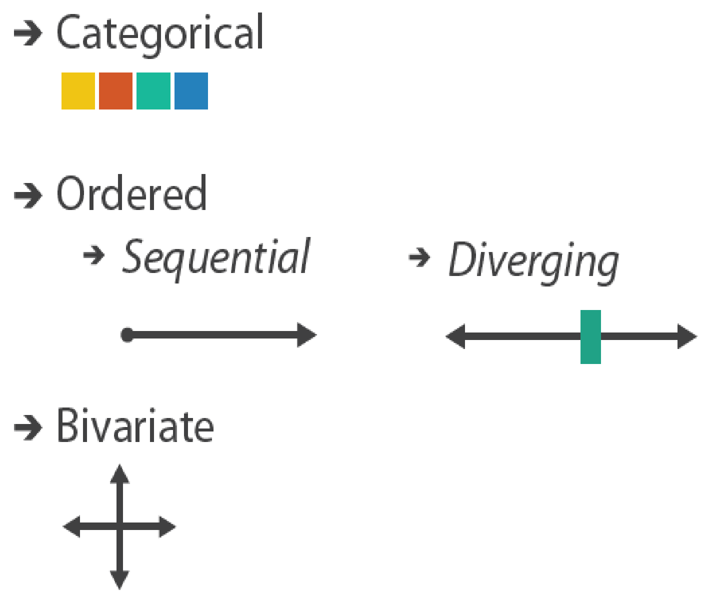
- Other common color spaces
  - RGB: poor choice for visual encoding
  - HSL: better, but be aware that
    - Lightness ≠ luminance
    - L* from L*a*b* perceptual color model is a better approximation of luminance



- Transparency
  - Useful for creating visual layers
    - Cannot be combined with luminance or saturation

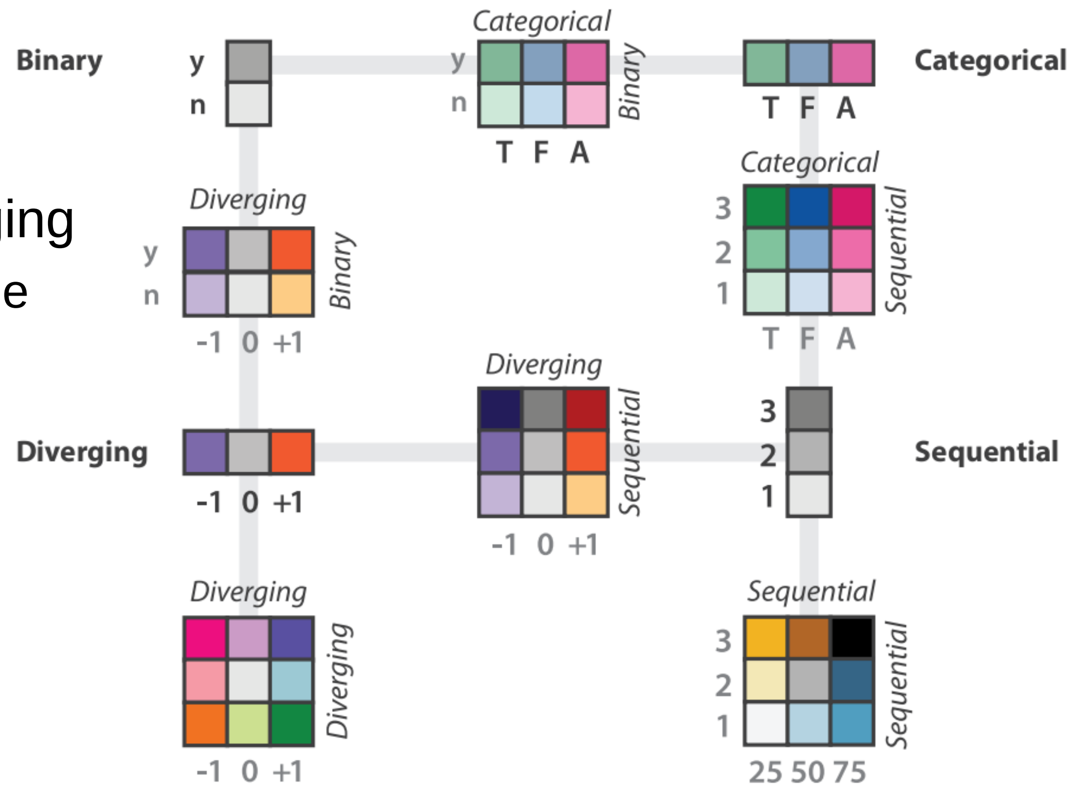
Colormaps

- Colormaps specify a mapping between data values and colors
- One dimensional colormaps
  - Categorical data
    - For noncontiguous use of color
    - 6-12 bins hue/color
    - 3-4 bins luminance, saturation
    - Only 6 to 12 bins in total
    - Size heavily affects salience
    - Use high saturation for small regions, low saturation for large
  - Ordered data use continuous color maps
    - Sequential range of data value
    - Diverging value ranges
- Bivariate / two-dimensional color maps
  - Encode two attributes simultaneously



Categorization of Colormaps

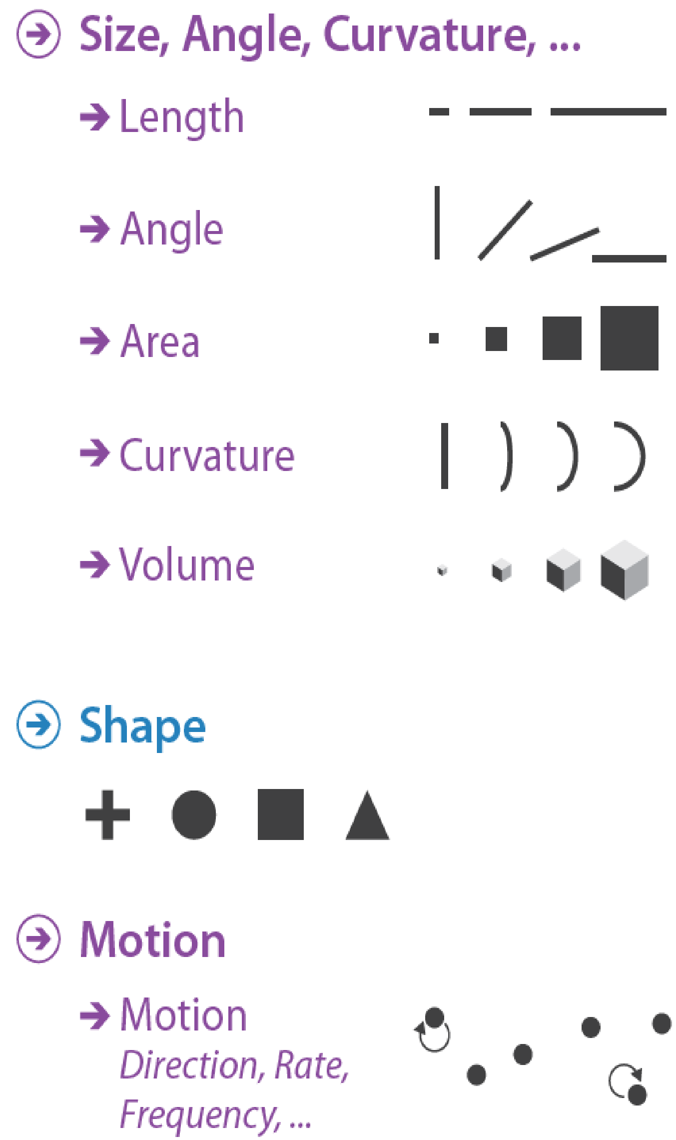
- Categorization
  - Categorical vs. ordered
  - Ordered: sequential vs. diverging
    - Diverging has zero point at the center
  - Here: binary encoding added
  - Bivariate colormaps work well for a few levels
- ColorBrewer
  - http://www.colorbrewer2.org
  - Creates colormaps based on perceptual guidelines



After [Color Use Guidelines for Mapping and Visualization. Brewer, 1994]
Examples: http://www.personal.psu.edu/faculty/c/a/cab38/ColorSch/Schemes.html]

Map Other Channels

- Size
  - Length accurate, 2D area ok, 3D volume poor
- Angle
  - Nonlinear accuracy
    - Horizontal, vertical, exact diagonal preferred
- Shape
  - Complex combination of lower-level primitives
  - Many bins
- Motion
  - Highly separable against static background
    - Binary: great for highlighting
  - Use with care – gets quickly annoying



Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - *Chap 10: Map Color and Other Channels*
- **ColorBrewer**, Brewer.
  - http://www.colorbrewer2.org
- *Color In Information Display*. Stone. IEEE Vis Course Notes, 2006.
  - http://www.stonesc.com/Vis06
- A Field Guide to Digital Color. Stone. AK Peters, 2003.
- *Rainbow Color Map (Still) Considered Harmful.* Borland and Taylor. IEEE Computer Graphics and Applications 27:2 (2007), 14–17.
- Visual Thinking for Design. Ware. Morgan Kaufmann, 2008.
- Information Visualization: Perception for Design, 3rd edition. Ware. Morgan Kaufmann /Academic Press, 2004.

End

Visualization

Introduction to Information Visualization III

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Big NEWS

- Grading
  - **No final exam this semester**
  - Assignments (40%)
  - Final project (individual projects or in groups of two, workload about 2 weeks) (60%)
    - For groups: each group member has to present her/his part of the project separately and has to make arguments that relate your work to Munzner’s what-why-how framework and the techniques introduced in this course
  - Final project be turned in at any time during the semester break but needs to be turned in at the latest by September 30!

Announcements

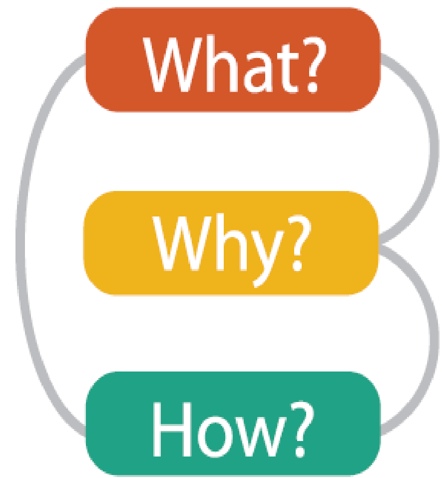
- Lab classes start next Tuesday, April 21
  - All Bachelor‘s degree students: 17:00-18:30h (in German)
  - All Master‘s degree students: 18:45-20:15h (in English)
  - Location: Lint Pool, Bauhausstr. 11
- Additional lecture: Wednesday, 29.4. at 15:15 (R015/B11)

Acknowledgements

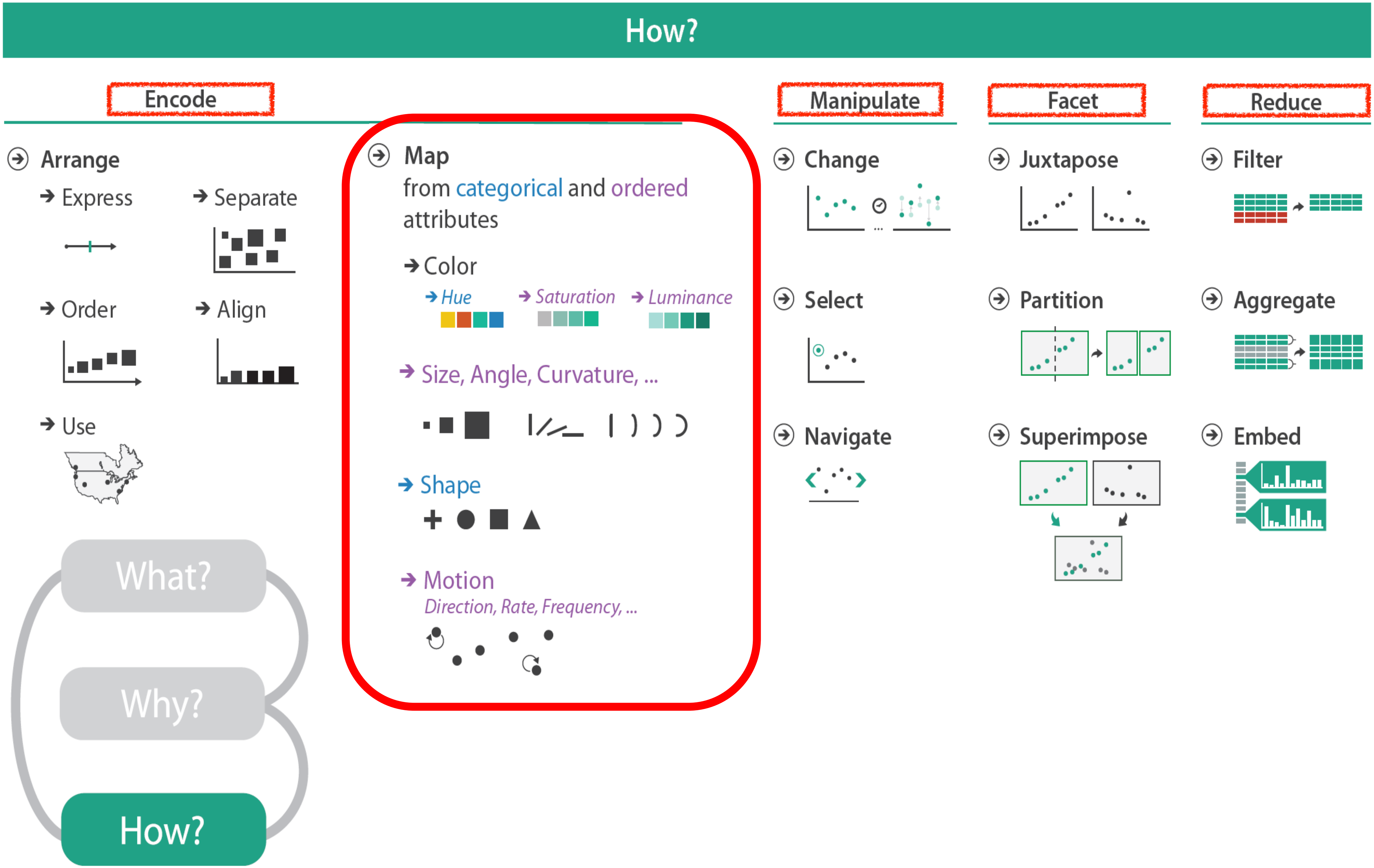
- First three lectures are based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.
- Further based on the books and lectures by
  - Robert Spence: Information Visualization 2nd Ed
  - Colin Ware: Information Visualization 2nd Ed.
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Silvia Miksch, TU Wien
  - Jeffrey Heer, Stanford University

Analysis: What, Why, and How

- What is shown?
  - **Data** abstraction
- Why is the user looking at it?
  - **Task** abstraction
- How is it shown?
  - **Technique**: visual encoding and interaction

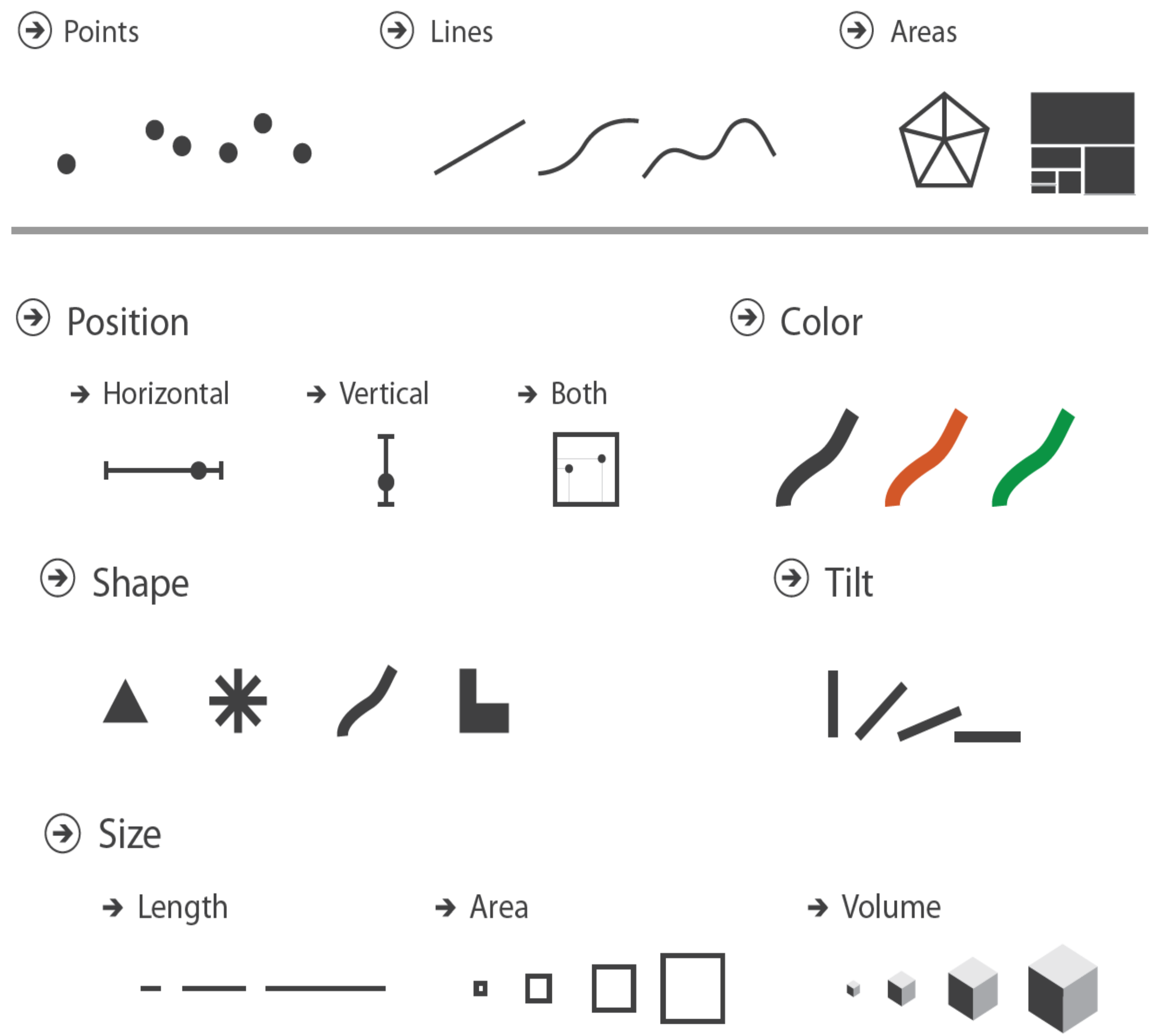


- Abstract vocabulary avoids domain-specific terms
  - Translation process iterative and often complex
- What-why-how analysis framework helps to think systematically about design space

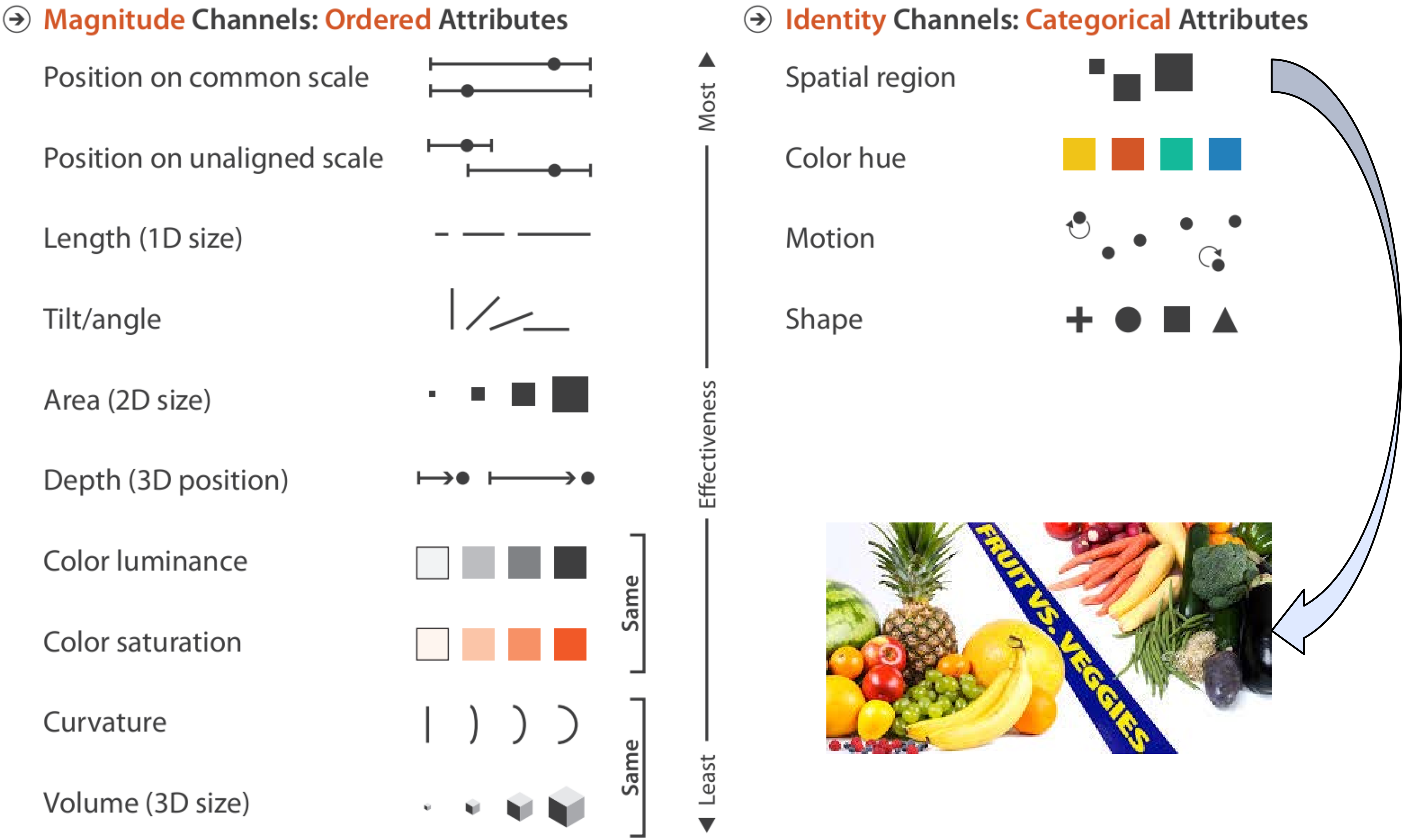

How?
Encode
Arrange
Express
Separate
Order
Align
Use
Map
from categorical and ordered attributes
Color
Hue
Saturation
Luminance
Size, Angle, Curvature, ...
Shape
Motion
Direction, Rate, Frequency, ...
Manipulate
Change
Select
Navigate
Facet
Juxtapose
Partition
Superimpose
Reduce
Filter
Aggregate
Embed
What?
Why?
How?

Definitions: Marks and Channels

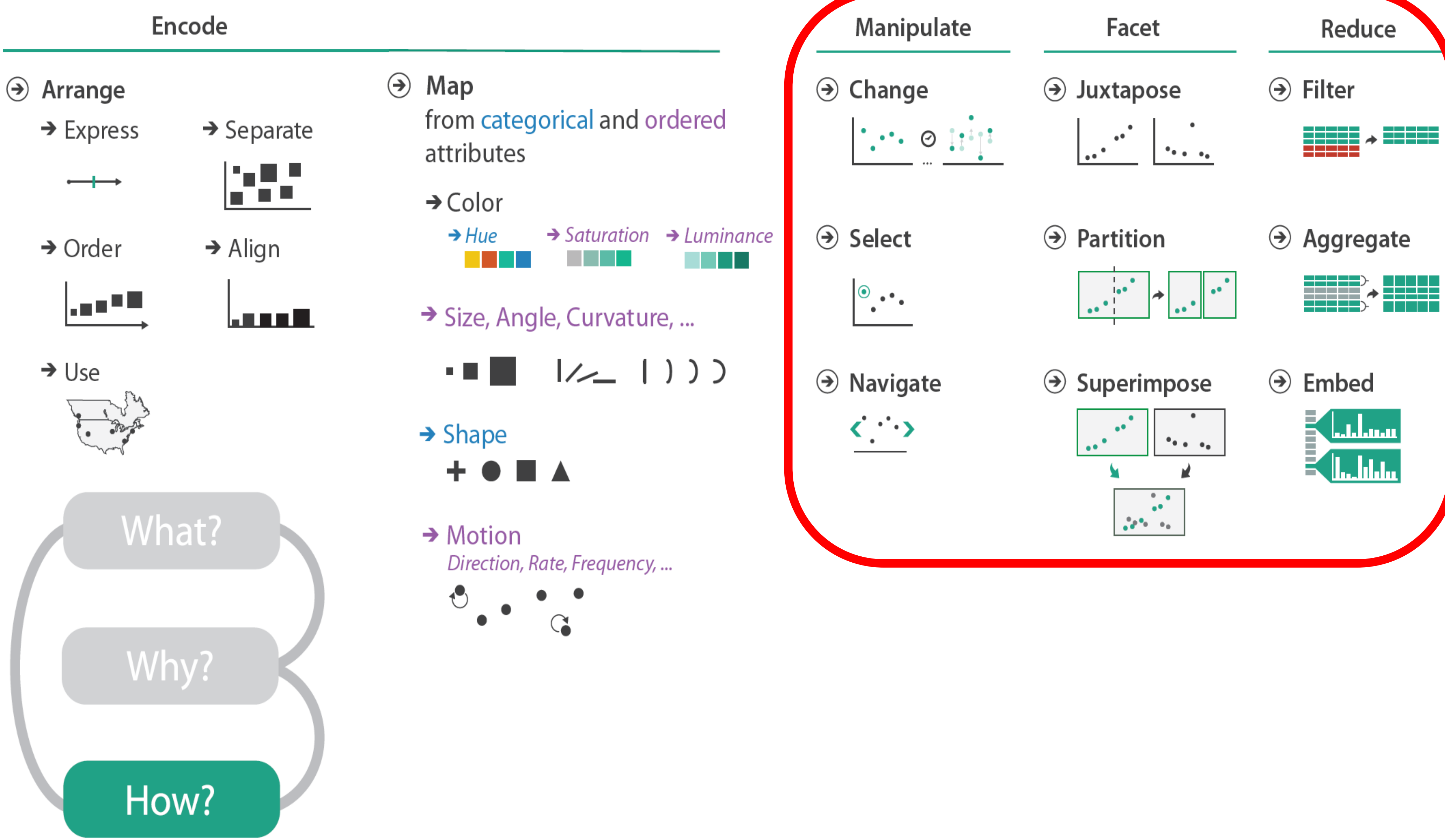
- Marks
  - Geometric primitives
- Channels (visual attributes)
  - Control appearance of marks
  - Can redundantly code using multiple channels



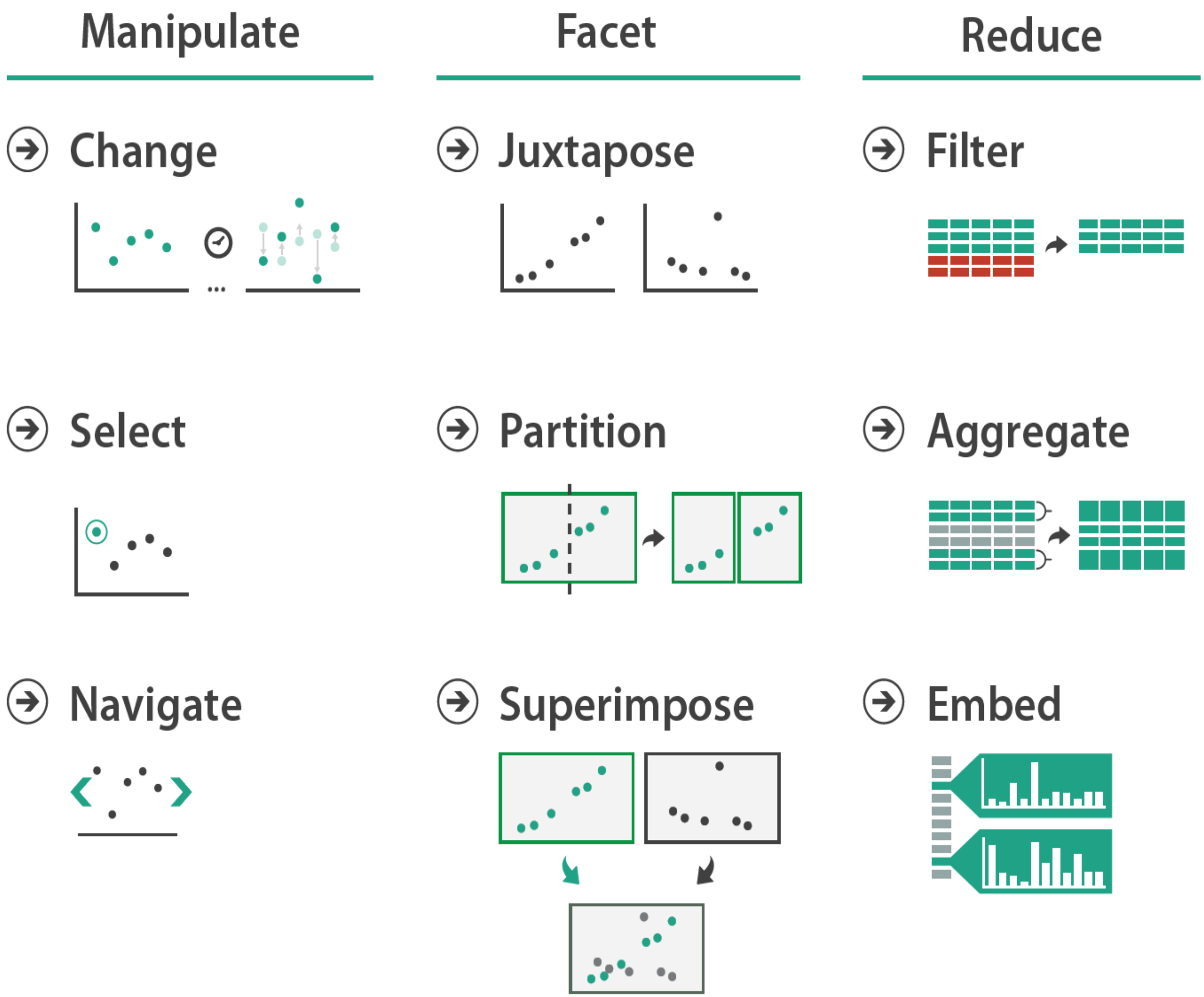
Channels: Effectiveness Rankings



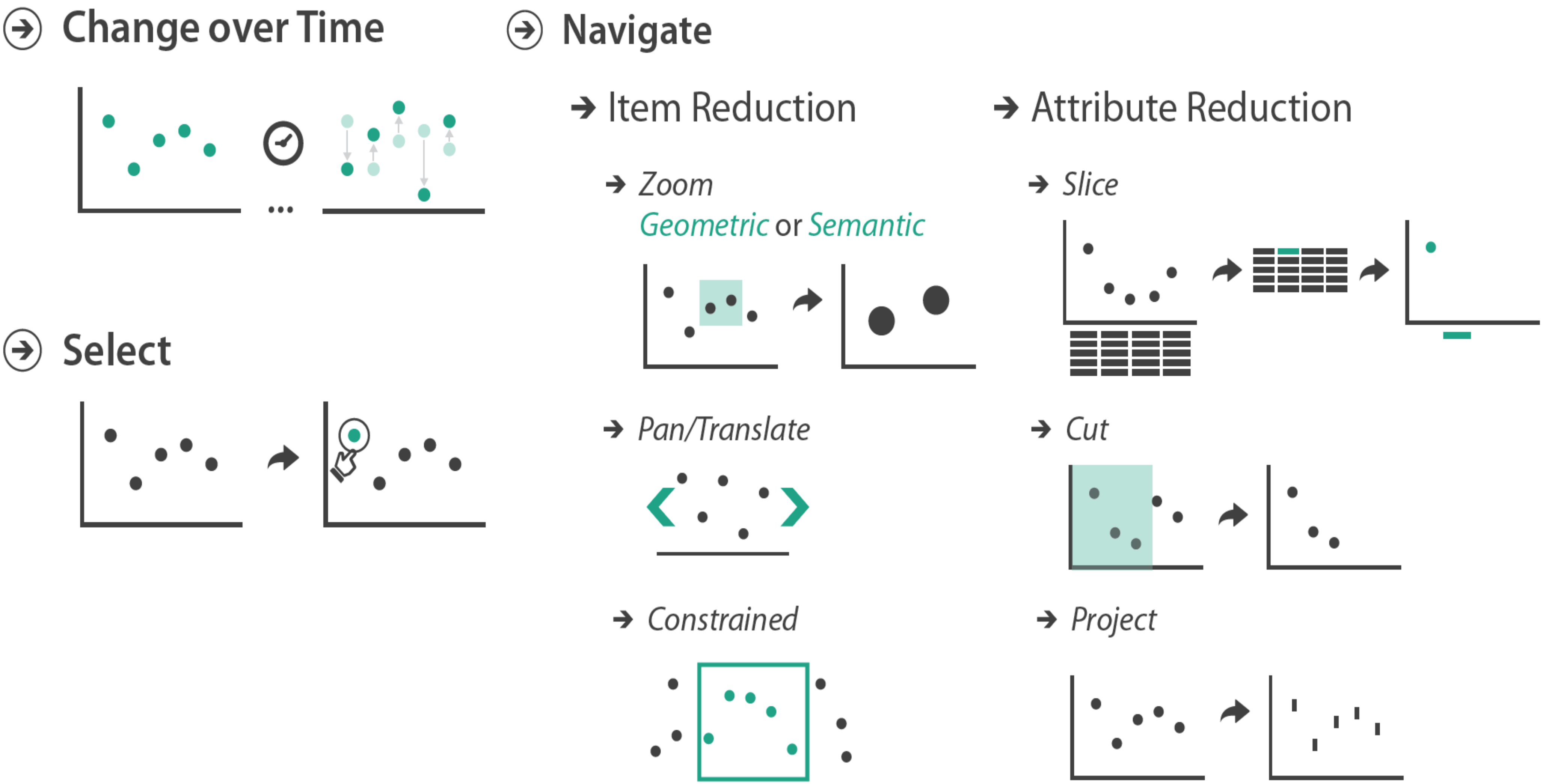
How?


Encode
Arrange
Express
Separate
Order
Align
Use
Map
from categorical and ordered attributes
Color
Hue
Saturation
Luminance
Size, Angle, Curvature, ...
Shape
Motion
Direction, Rate, Frequency, ...
Manipulate
Change
Select
Navigate
Facet
Juxtapose
Partition
Superimpose
Reduce
Filter
Aggregate
Embed
What?
Why?
How?

Interaction Technqiues Overview



Manipulate

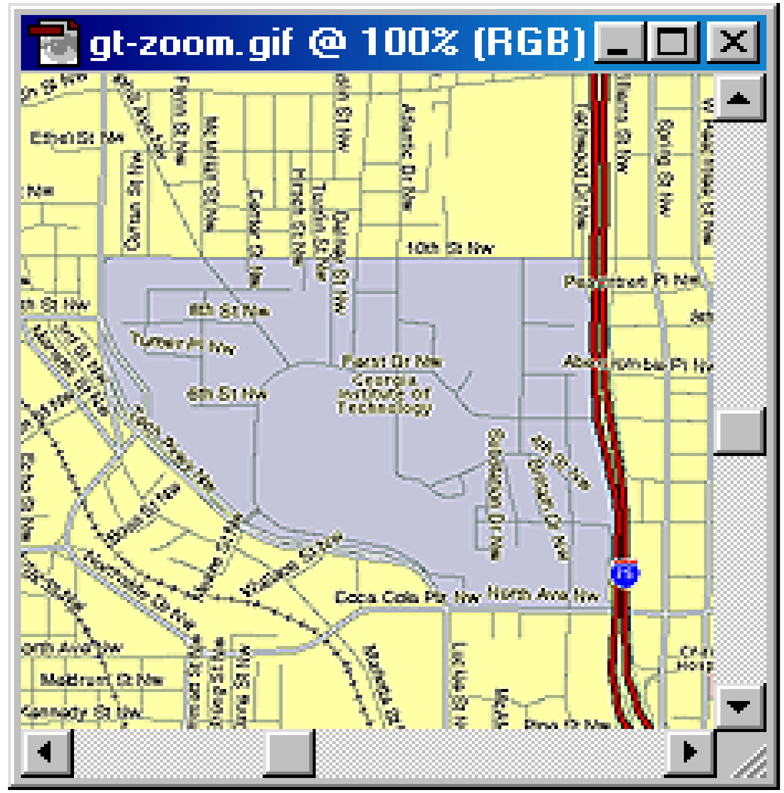


Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014. Chapter 11

Navigation Standard Solution: Scrolling

Detail-only window

- Single window with horizontal and vertical panning to access a larger virtual screen
- Issues:
  - Only single focus
  - Efficient scrolling requires cognitive map
    - No overview
    - Relationship between the current focus and the desired focus?
    - Must map relationship between current and next foci
      to the slider positions on the scroll bar
  - Works only when zoom factor is relatively small



Overview first, zoom and filter, details on demand

- Influential mantra from Ben Shneiderman

- What is needed
  - Overview (summary)
  - Focus view
  - Support for navigation

[The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Shneiderman. Proc. IEEE Visual Languages, pp. 336–343, 1996.]

[Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. van Ham and Perer. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 953–960.]

Overview + Detail

- Provide both overview and detail displays
- Two ways to combine:
  - Time - Alternate between overview and detail sequentially
  - Space - User different portions of the screen

detail

overview

Overview and detail (from *Civilization II* game)

Plaisant et al Article

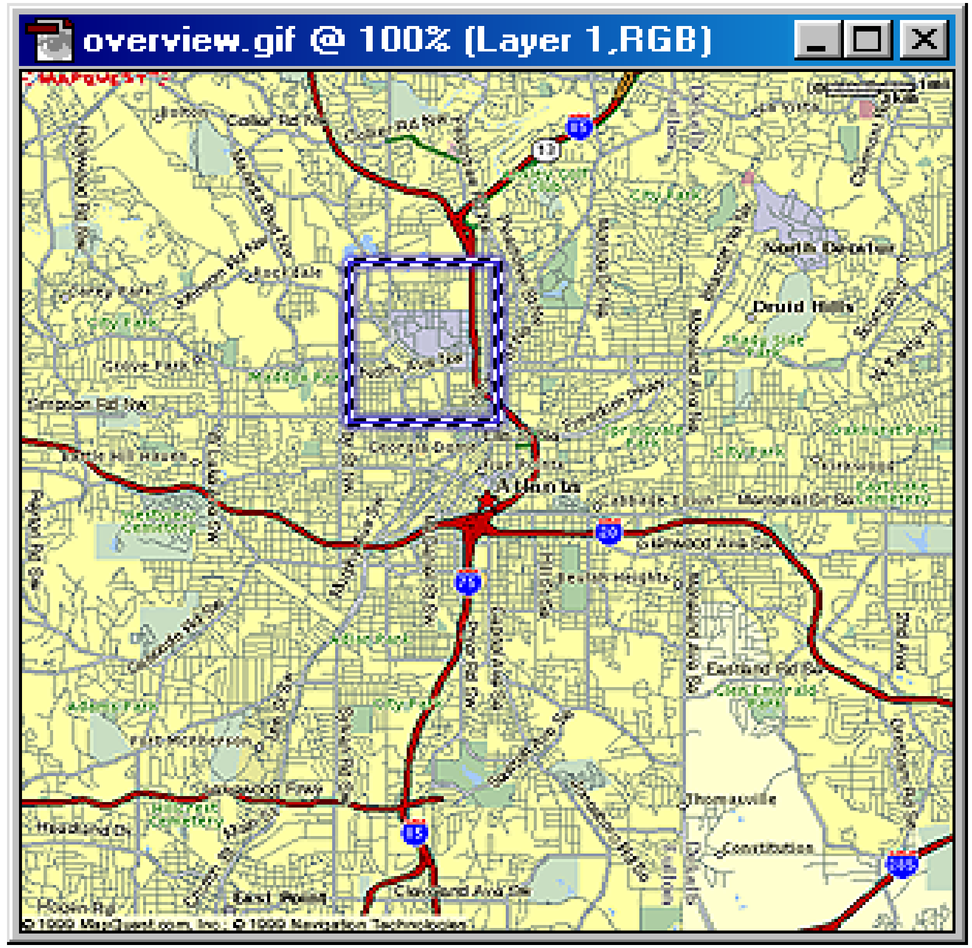
- Good introduction to overview/detail issue
- Application concern: viewing and editing large images
- Expanding the notion of the one dimensional scroll bar: zooming, diagonal panning, multiple detailed views
- List of visualization/interaction solutions

Image Browsers: Taxonomy and Design Guidelines. C. Plaisant, D. Carr, B. Shneiderman *IEEE Software* 12(2) March 1995 pp 21-32

One Window With Zoom & Replace

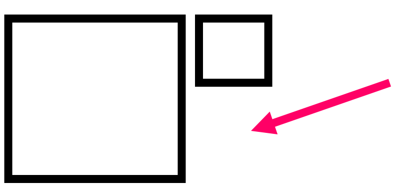
- Global view with selectable zoom area which then becomes entire view
- Variations can let users pan and adjust zoomed area and adjust levels of magnification
- Context switch can be disorienting

- Example: CAD/CAM



Single Coordinated Pair

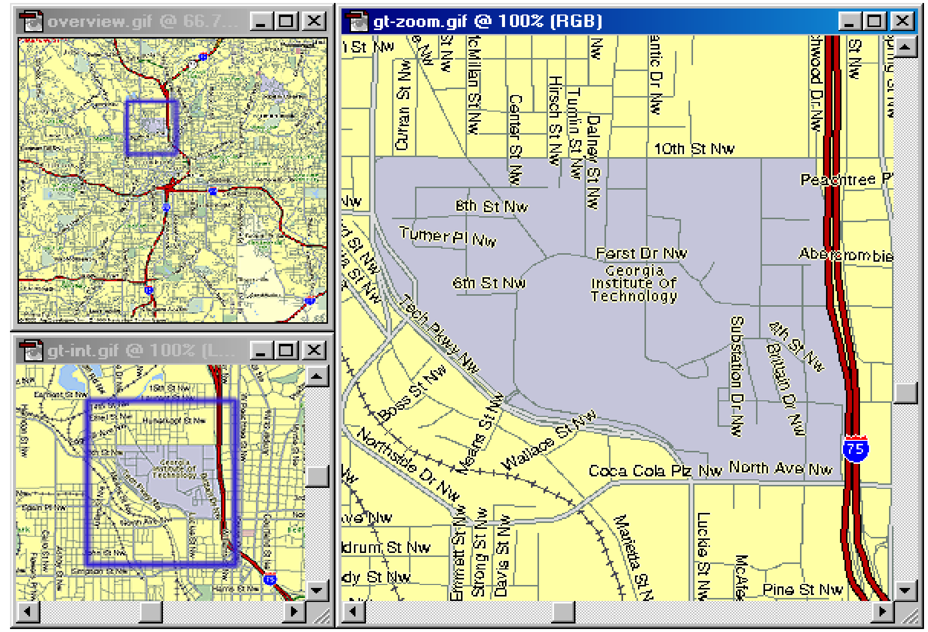
- Combined display of the overview and local magnified view (separate views)
- Some implementations reserve large space for overview; others for detail
- Issue: How big are different views and where do they go?



Wasted space

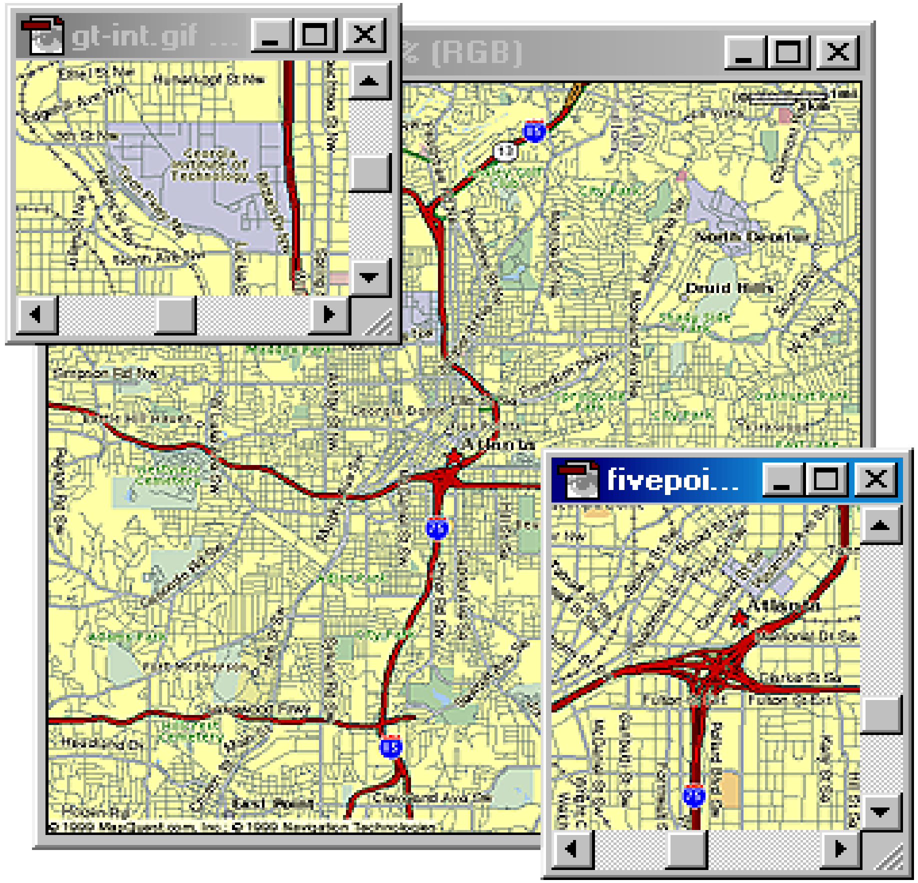
Tiled Multilevel Browser

- Combined global, intermediate, and detail views
- Views do not overlap
- Good implementations closely relate the views, allowing panning in one view to affect others



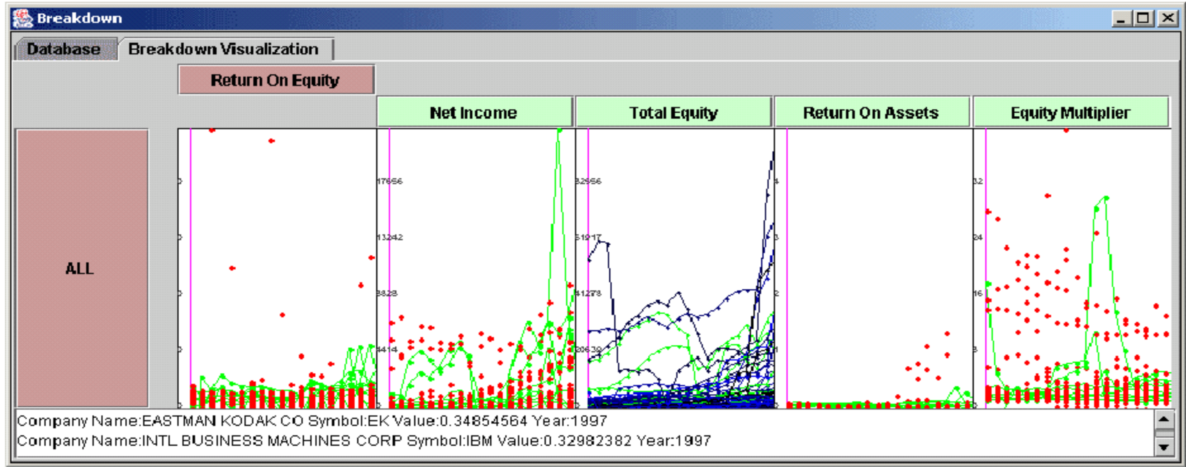
Free Zoom And Multiple Overlap

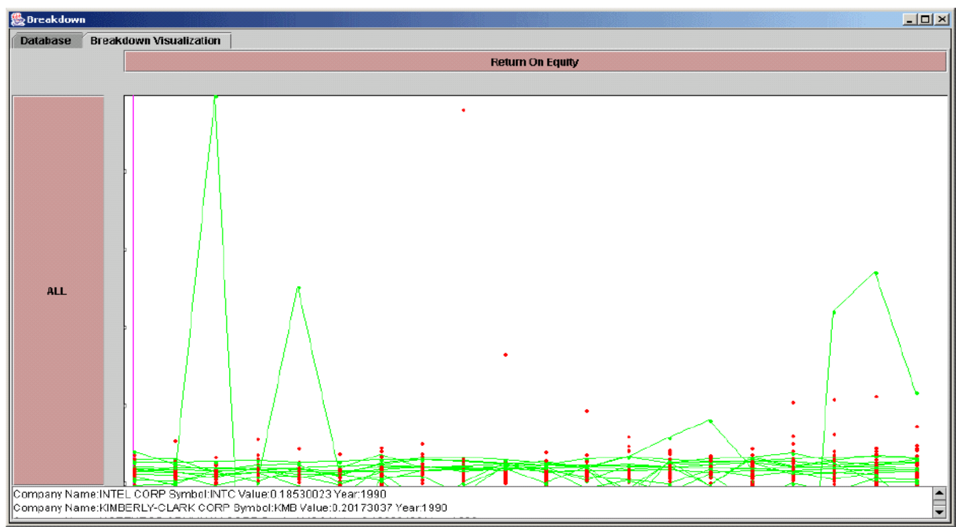
- Overview presented first; user selects area to zoom and area in which to create detailed view
- Flexible layout, but users must perform manual window management



Overview and Detail: Drill-down

- Select individual item or smaller set of items from a display for a more detailed view/analysis





Conklin et al. InfoVis2002

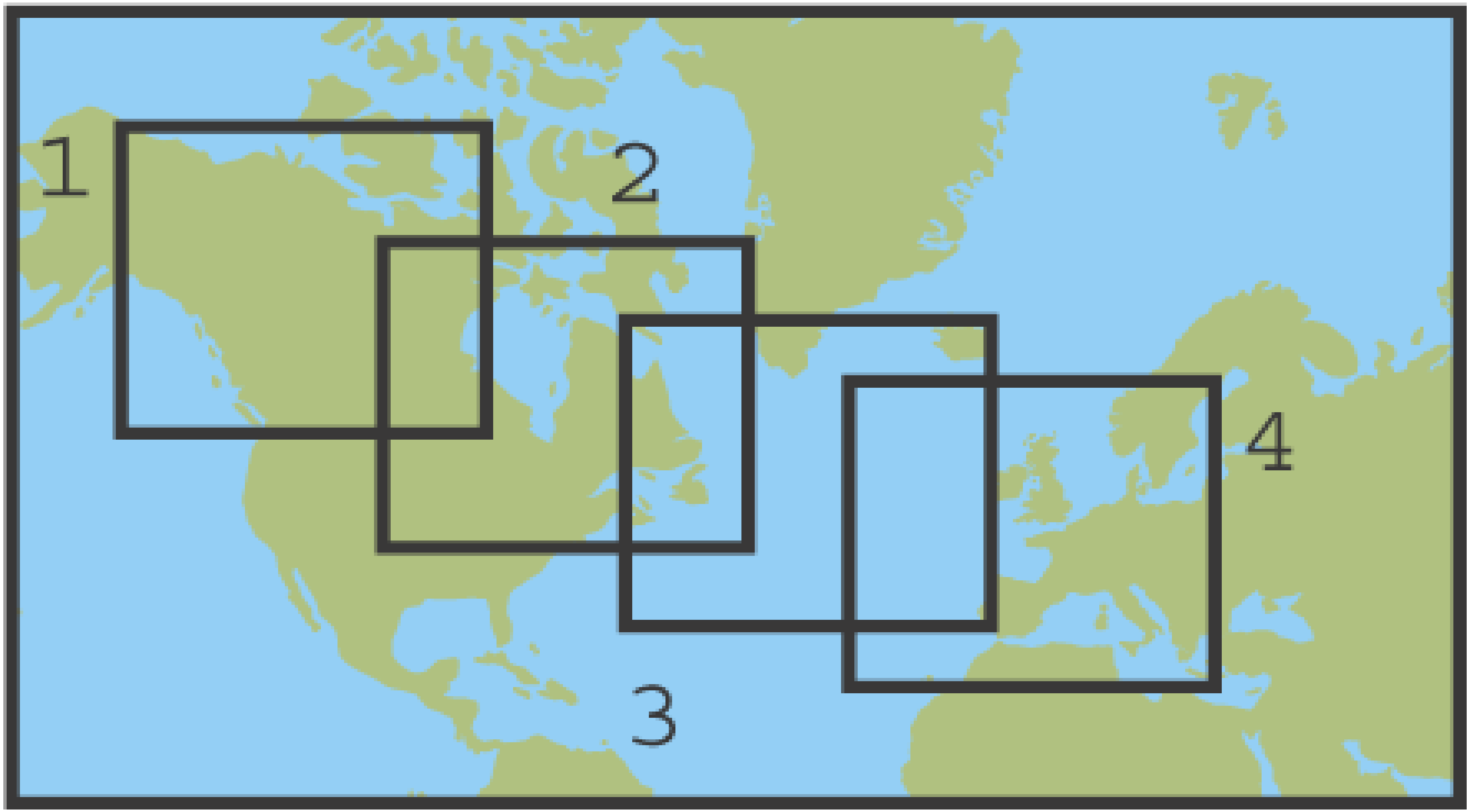
Panning and Zooming

- Panning
  - Smooth movement of camera across scene (or scene moves and camera stays still)
- Zooming
  - Increasing or decreasing the magnification of the objects in a scene

- Useful for changing focal point
- ZUI = Zoomable User Interface

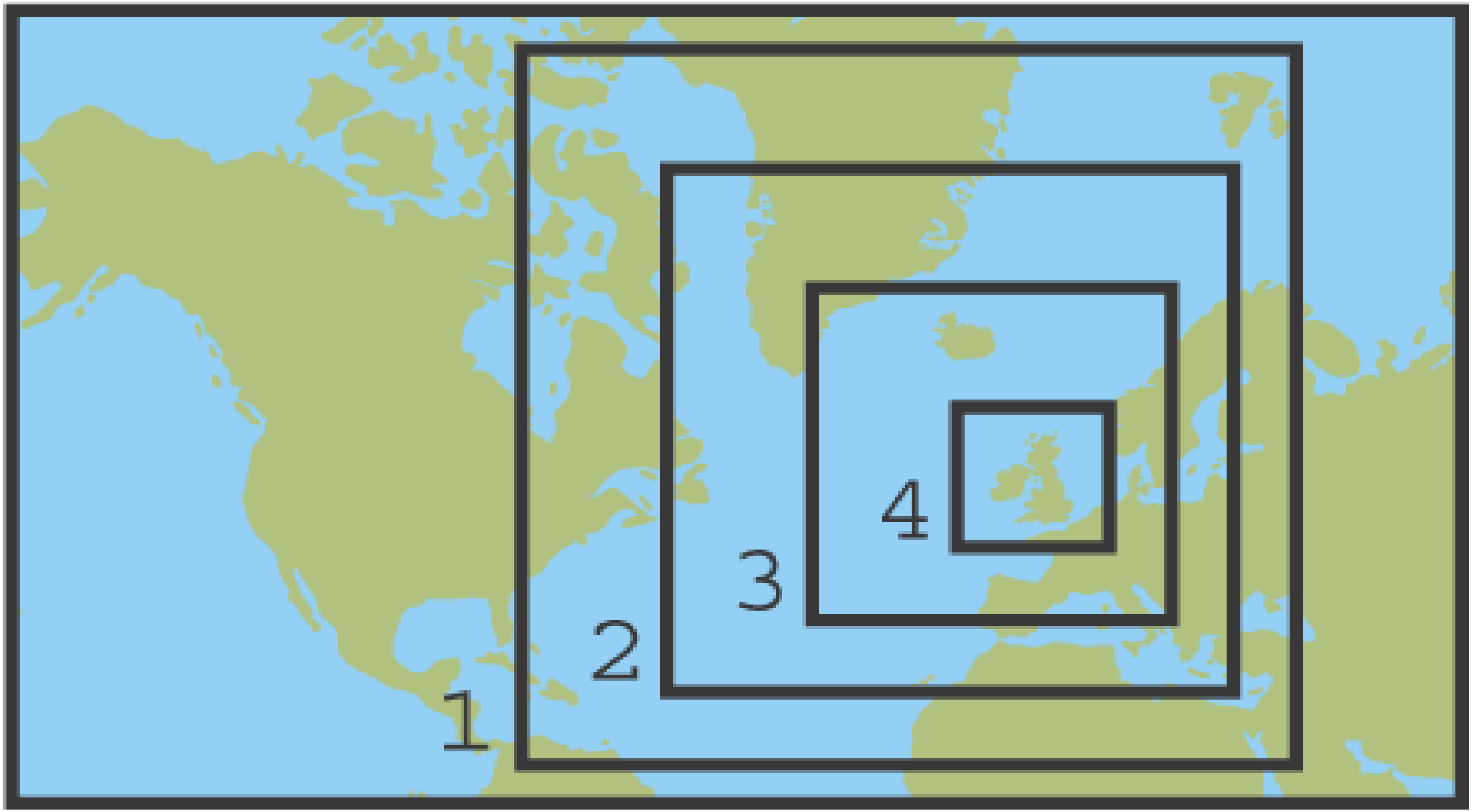
Panning

Smooth movement of a viewing window over a 2D image



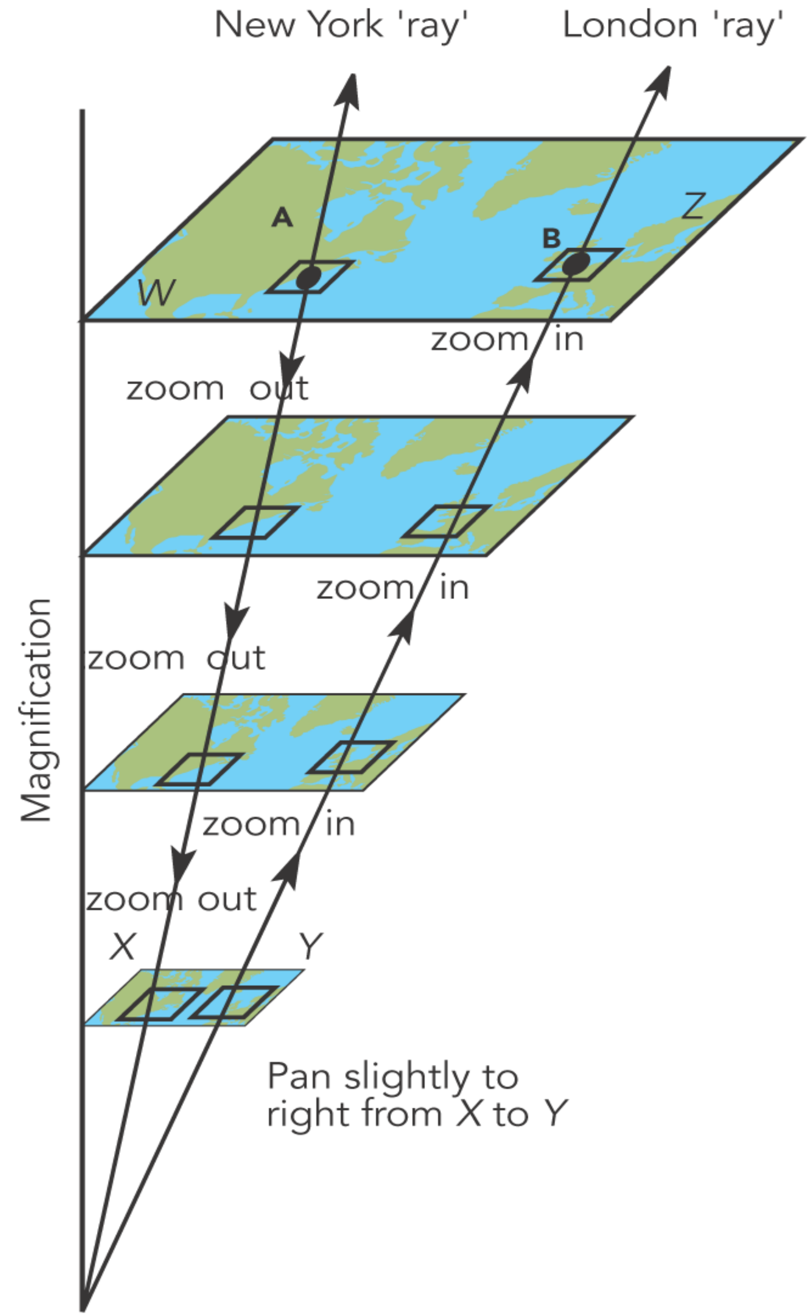
Zooming

Increasing/ or decreasing magnification



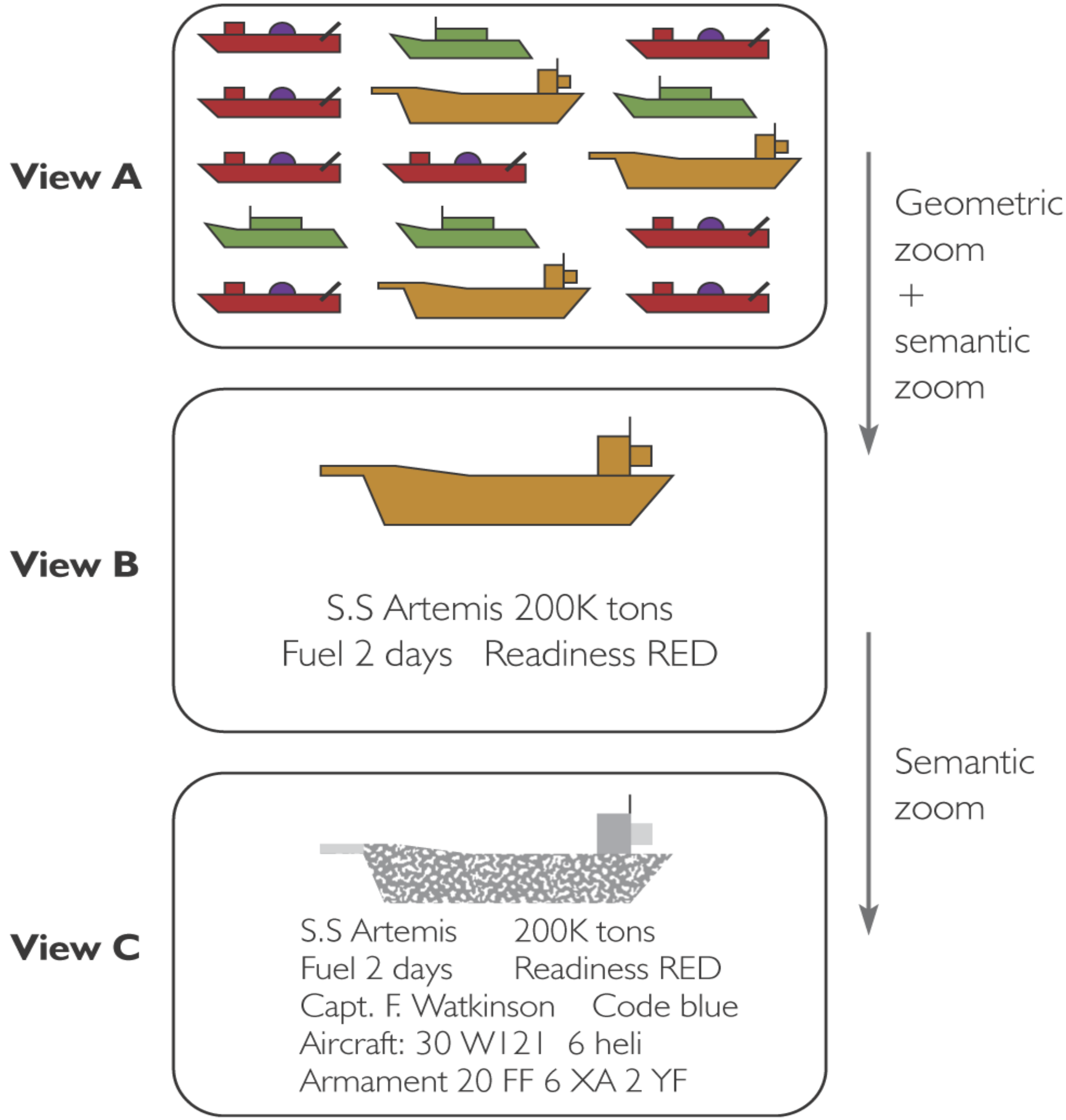
Combined Zooming and Panning

- Efficient task sequence
  - Zoom back
  - Pan
  - Zoom in



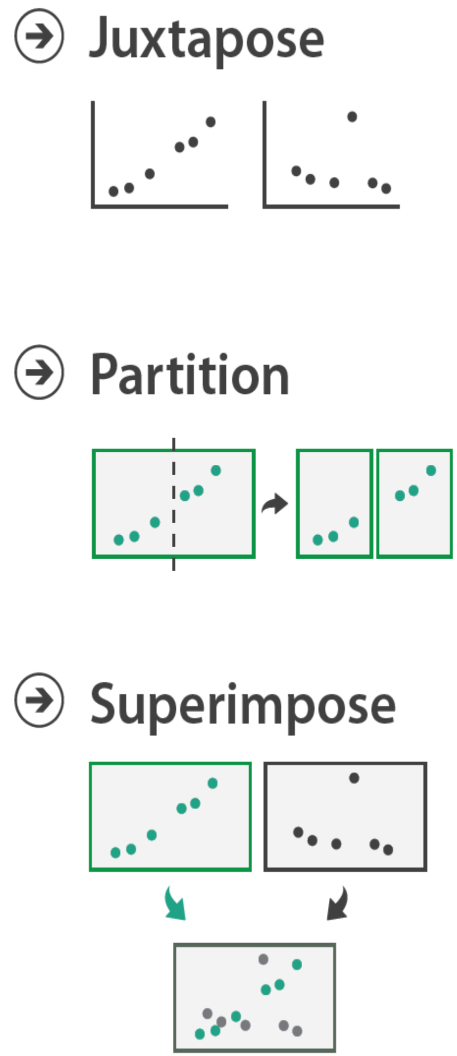
Geometric + Semantic Zoom

- Often just simple zooming is not all that useful
- Change representation based on zooming level!



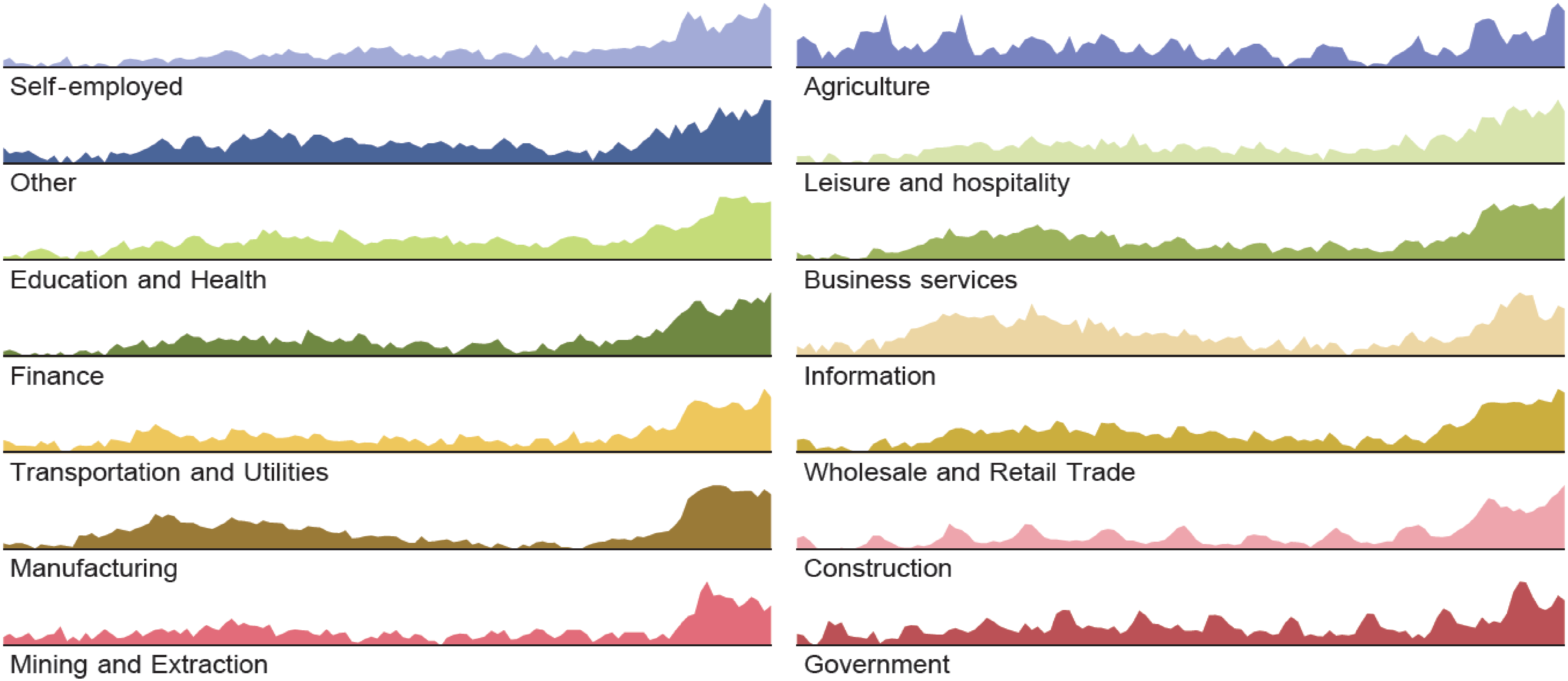
Facet (Split up)

- Handle visual complexity
  - Juxtaposing coordinated views side by side
    - Show different facets or views of the data
  - Superimposing layers in a single view
    - Requires splitting up in layers



Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014. Chapter 12

Juxtapose: Small Multiples



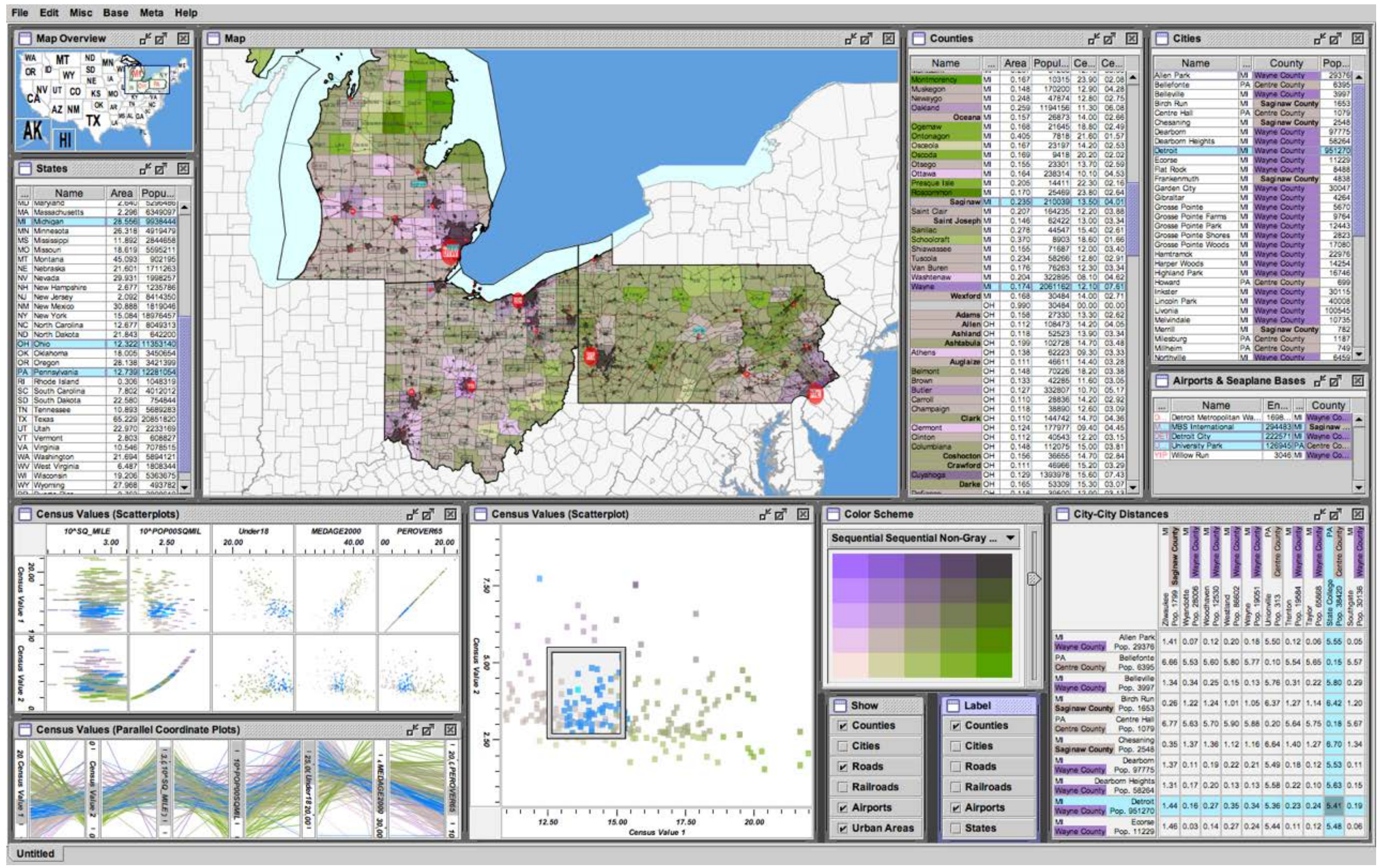
Unemployed U.S. workers by industry, 2000–2010
Data source: U.S. Bureau of Labor Statistics
http://hci.stanford.edu/jheer/files/zoo/ex/time/multiples.html

Small Multiples

- Can be constructed for plots, area graphs, bar charts, …
- Also interesting as linked views where coordinated interaction takes place across all individual visualizations
- Separate visualization can be more effective than overlaying all data in a single plot

Coordinated Multiple-View Display
Vis System: Improvise

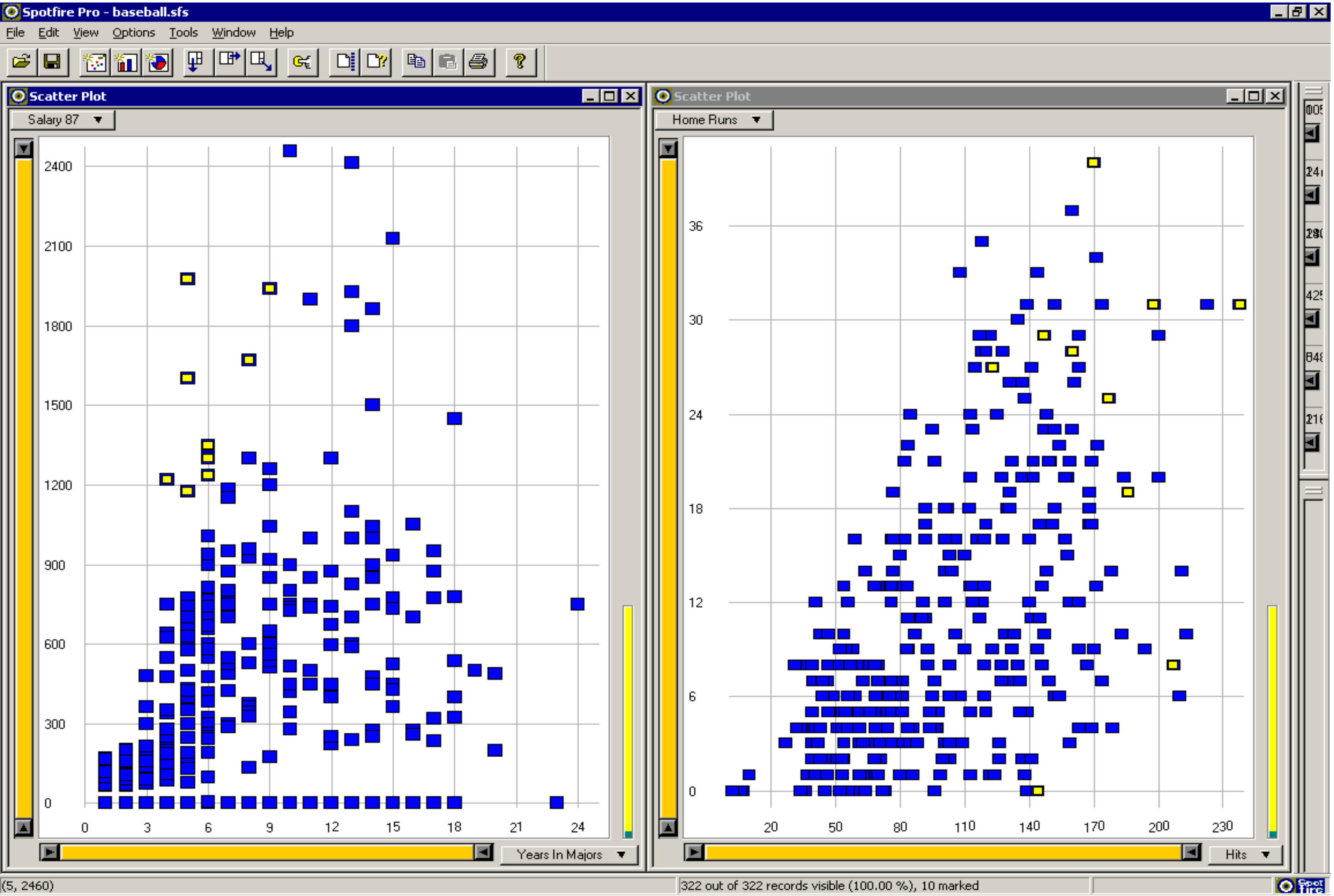
- Investigate power of multiple views
  - Pushing limits on view count, interaction complexity



[Building Highly-Coordinated Visualizations In Improvise. Weaver. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 159–166, 2004.]

Coordination Through Brushing and Linking

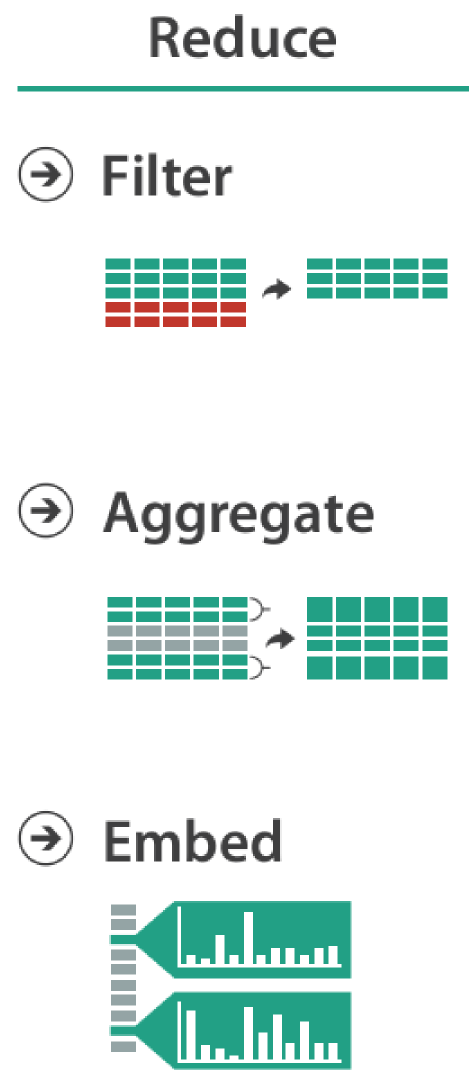
Select or designate/specify value, then see matching items elsewhere on the display



Example:
4D data
in linked
2D + 2D
display

Reduce Items and Attributes

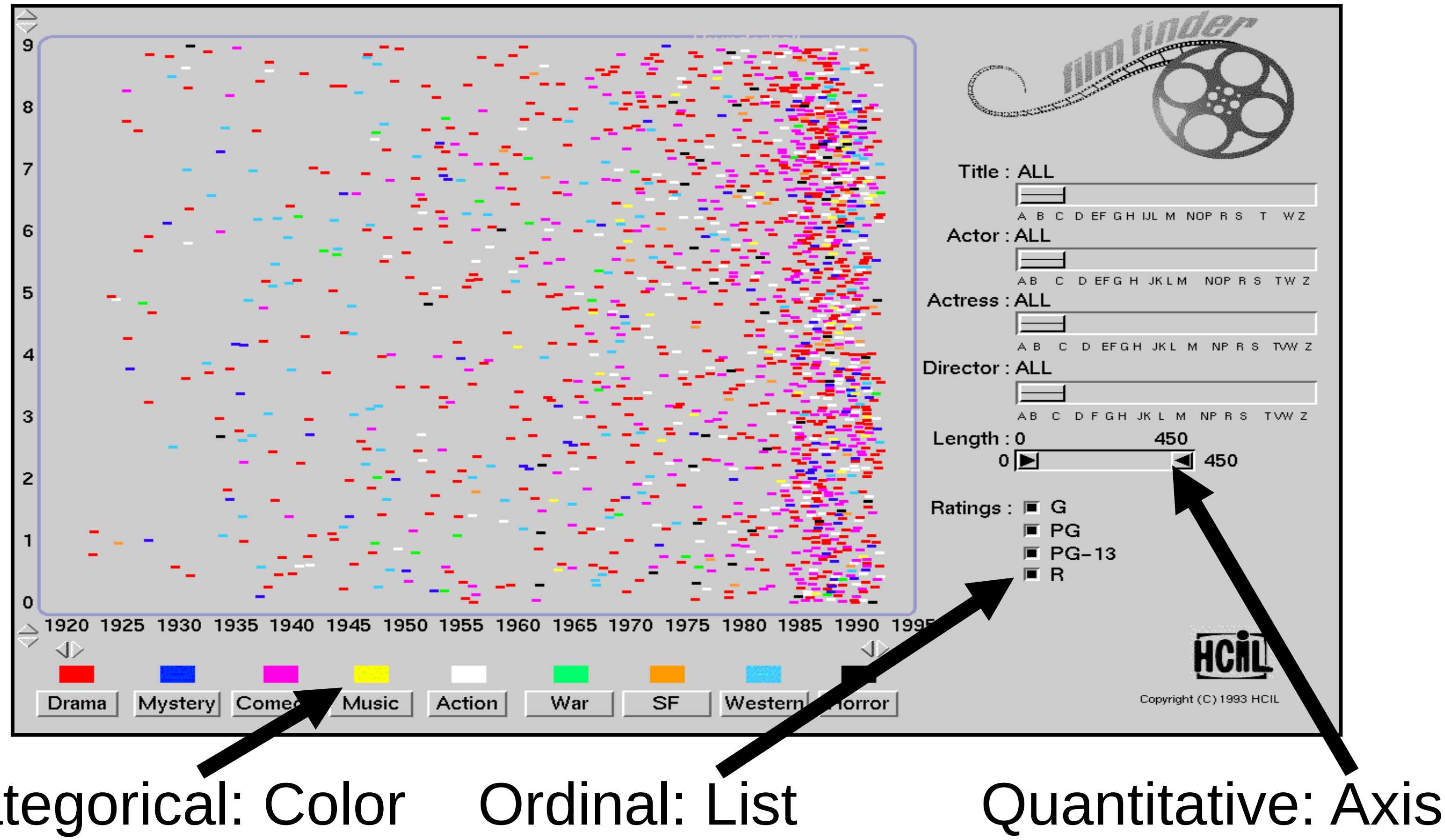
- Filter
  - Pro: straightforward and intuitive
    - to understand and compute
  - Con: out of sight, out of mind
- Aggregation
  - Pro: inform about whole set
  - Con: loses information
- Filter and Aggregation are not mutually exclusive
  - Combine filter, show aggregate of rest
  - Combine reduce, change, facet
- Embed
  - Show detail in context



Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014. Chapter 13 / 14

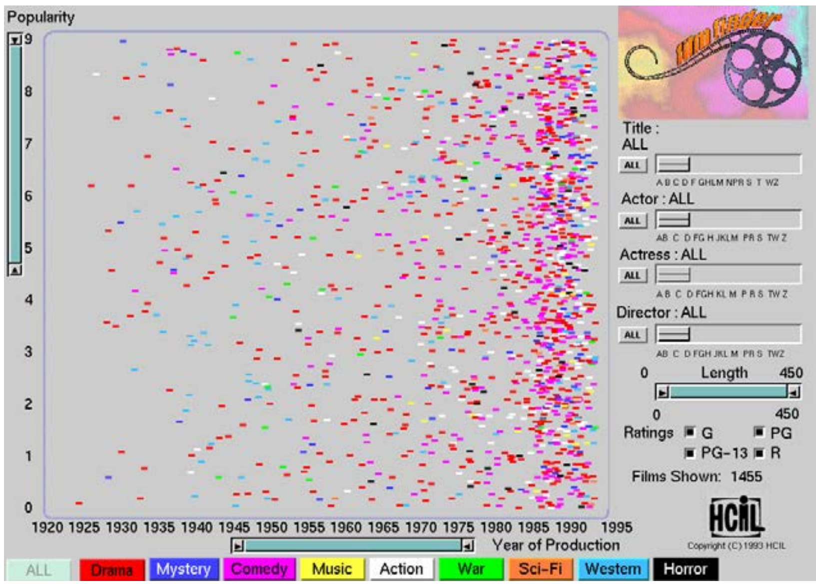
Example System Filmfinder

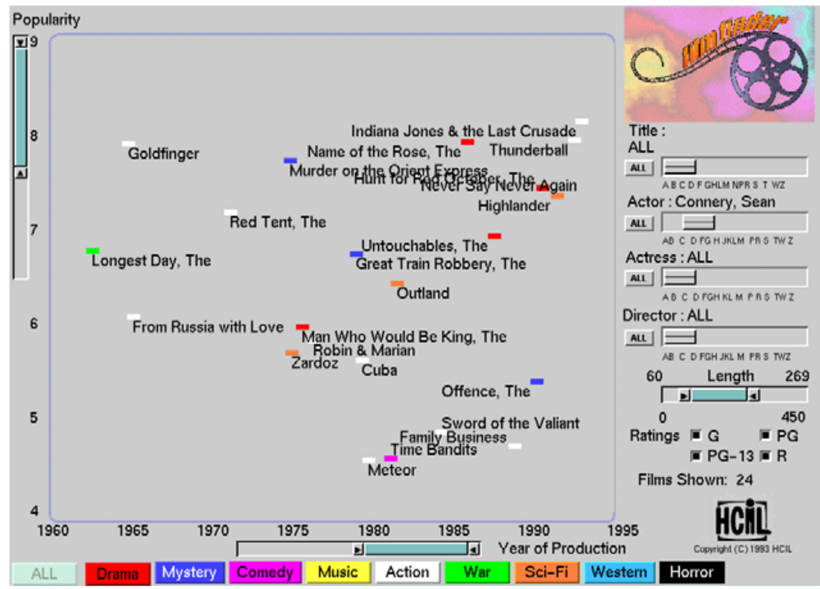
- Film Finder, University of Maryland



System Filmfinder Supports Dynamic Filtering

- Item filtering
- Browse through tightly coupled interaction
  - Alternative to queries that might return far too many or too few

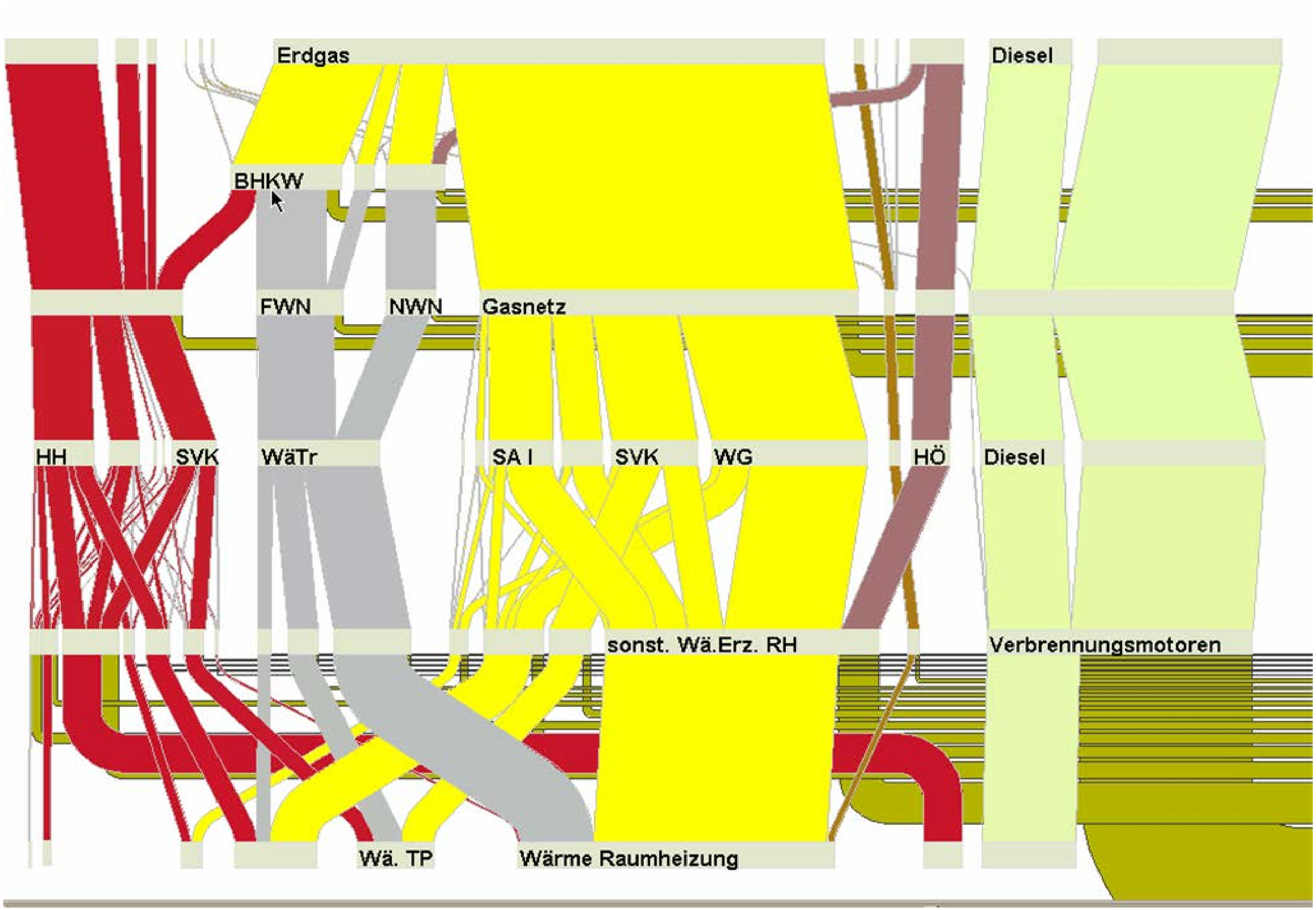




[Visual information seeking: Tight coupling of dynamic query filters with starfield displays. Ahlberg and Shneiderman. Proc. ACM Conf. on Human Factors in Computing Systems (CHI), pp. 313–317, 1994.]

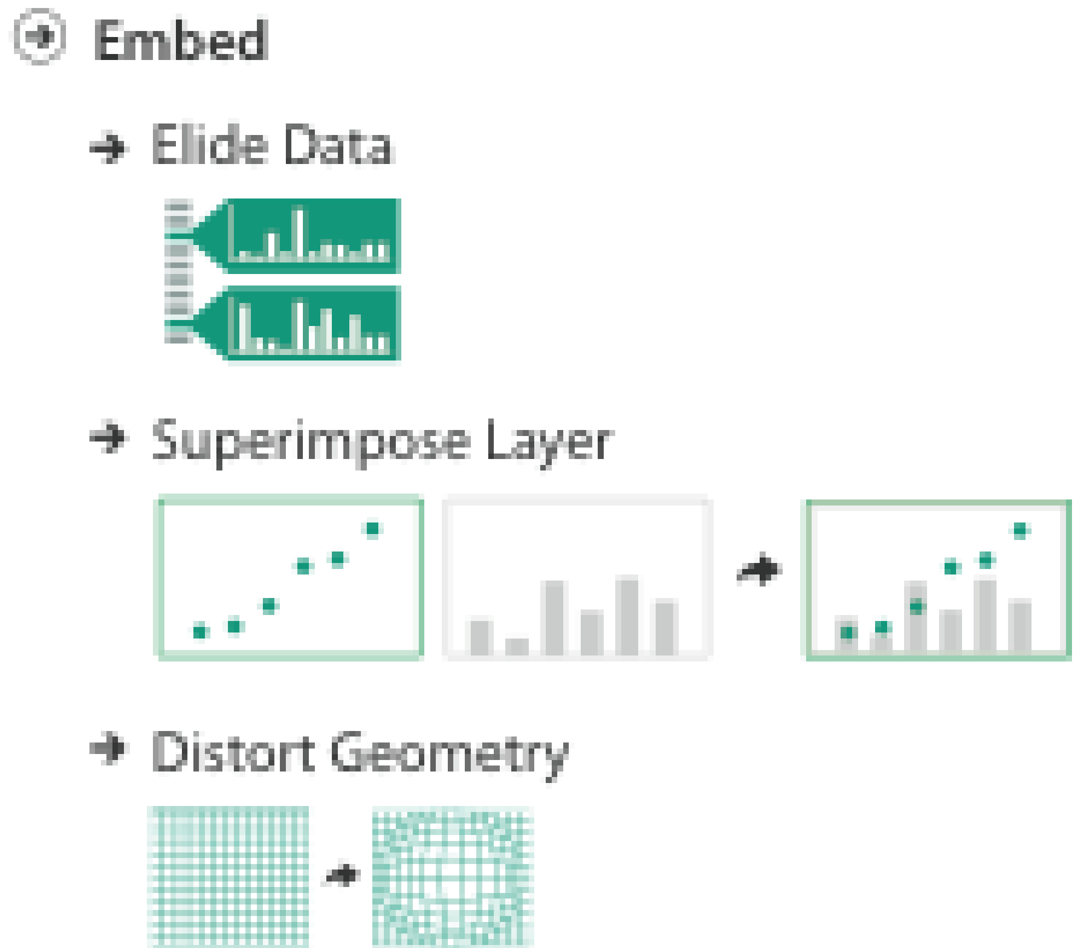
Aggregation Example

Aggregation Example



Embed: Focus + Context

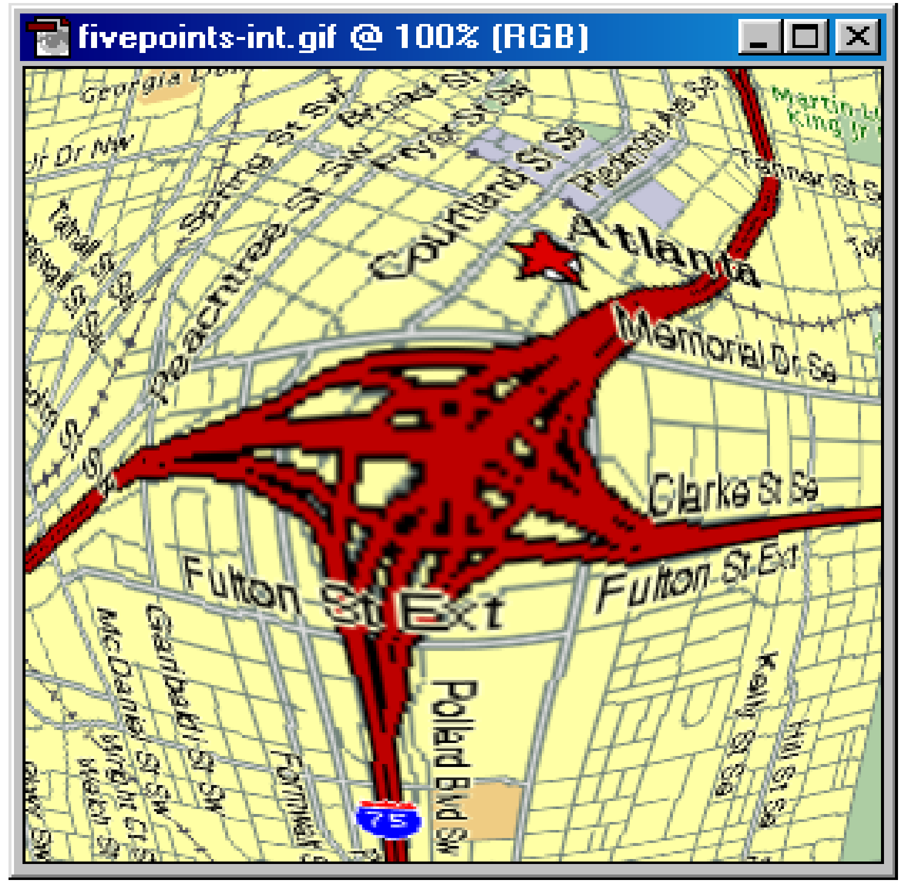
- Combine information within single view
  - Zooming hides the context
  - Two separate displays split attention
- Elide
  - Selectively filter and aggregate
- Superimpose layer
  - Local lens
- Distortion design choices
  - Region shape: radial, rectilinear, complex
  - How many focus areas: one, many
  - Region extent: local, global
  - Interaction metaphor



Distort Geometry: Fish-eye View

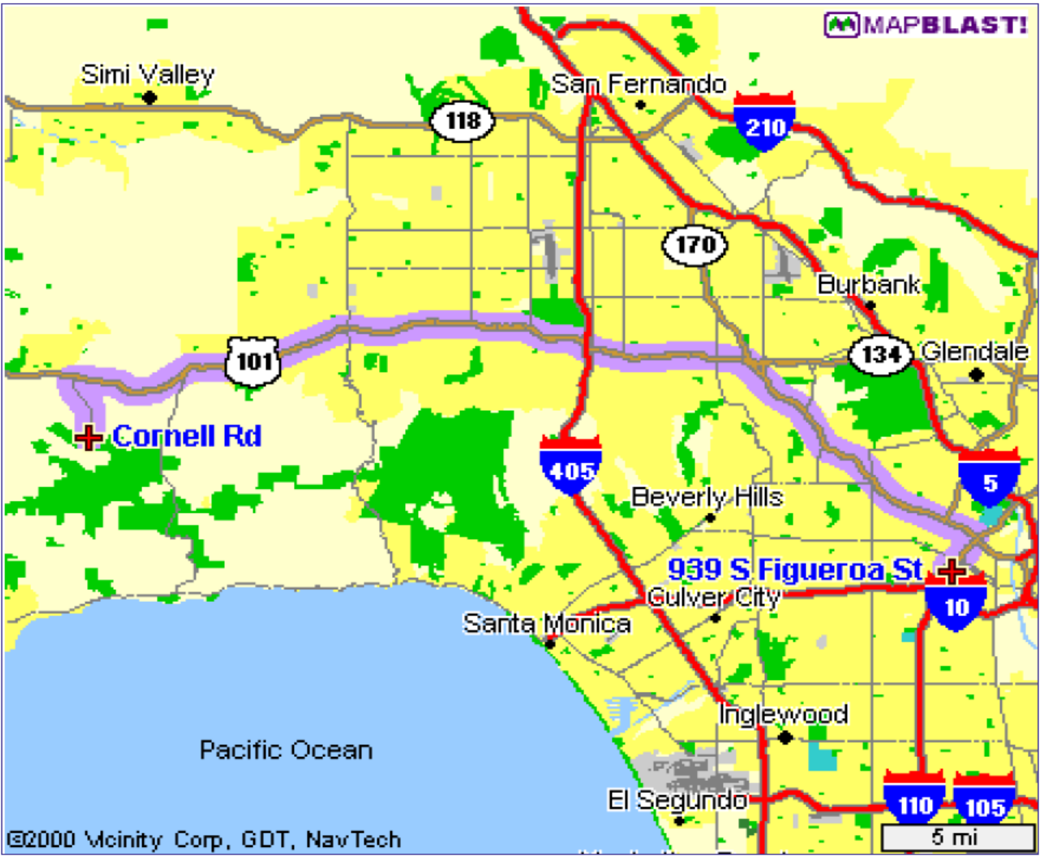
- Magnified image is distorted so that focus is at high magnification, periphery at low
- All in one view
- Distortion can be disorienting

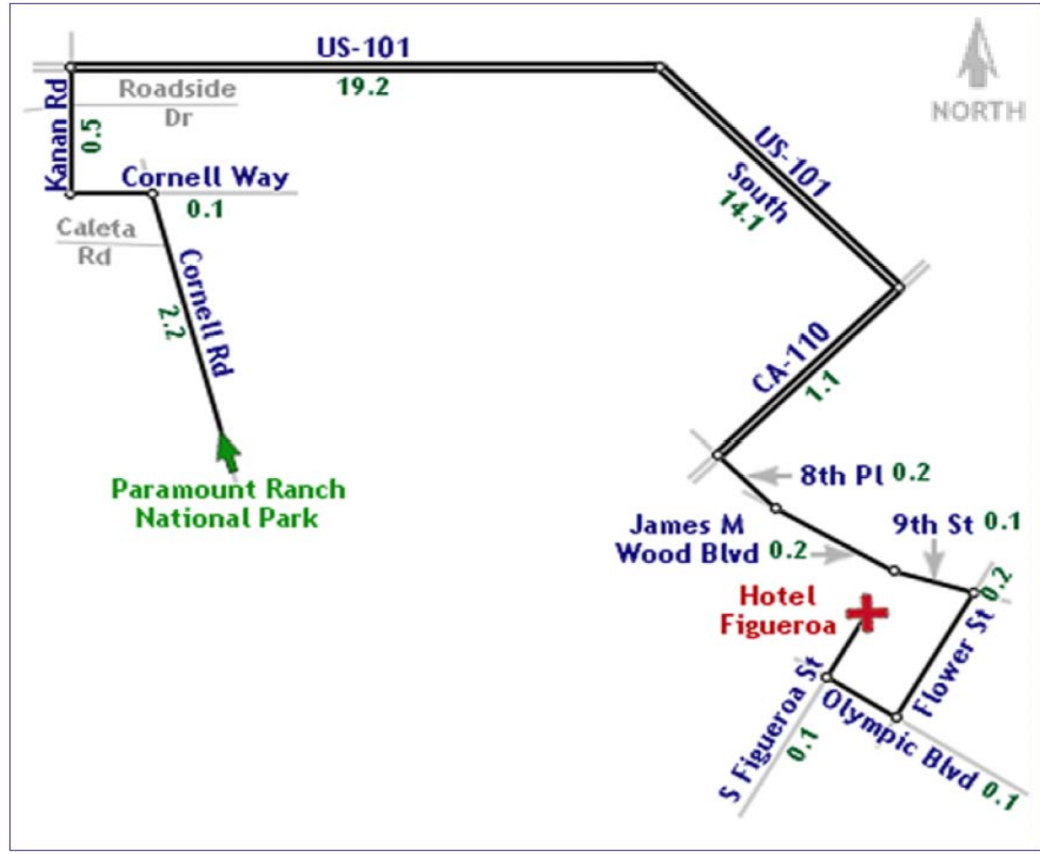
- This is focus + context
  It is **not** called
  OVERVIEW and DETAIL



A Really Useful Map!

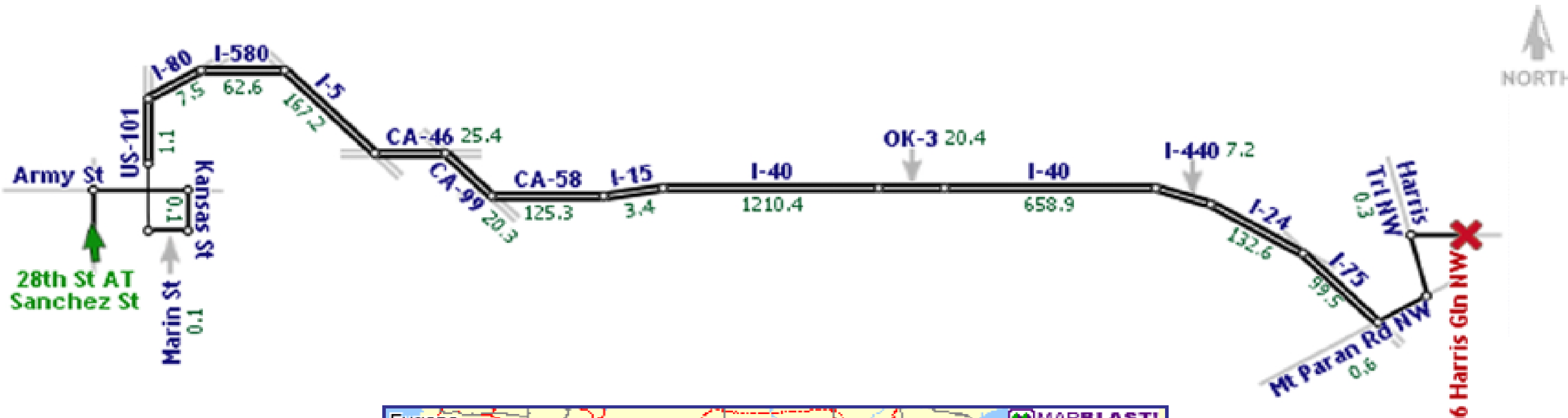
Rendering Effective Route Maps





Agrawala/Stolte, Siggraph 2001:
Rendering Effective Route Maps: Improving Usability Through Generalization

Another Route Map Example

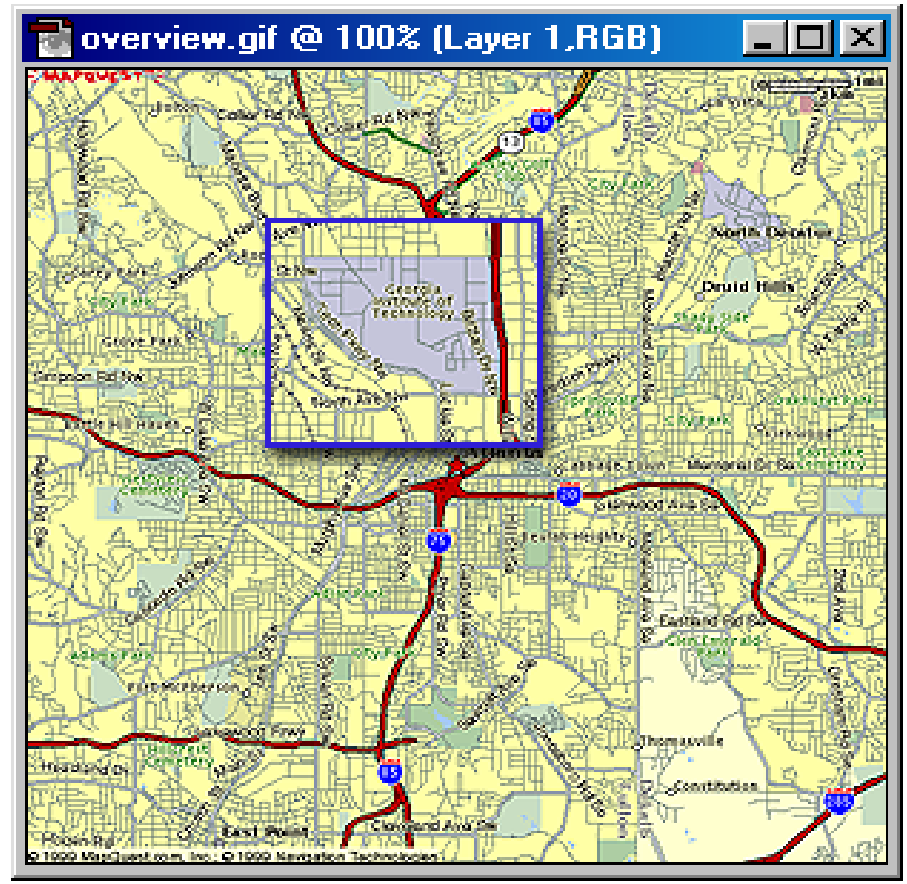


(a) San Francisco to Atlanta

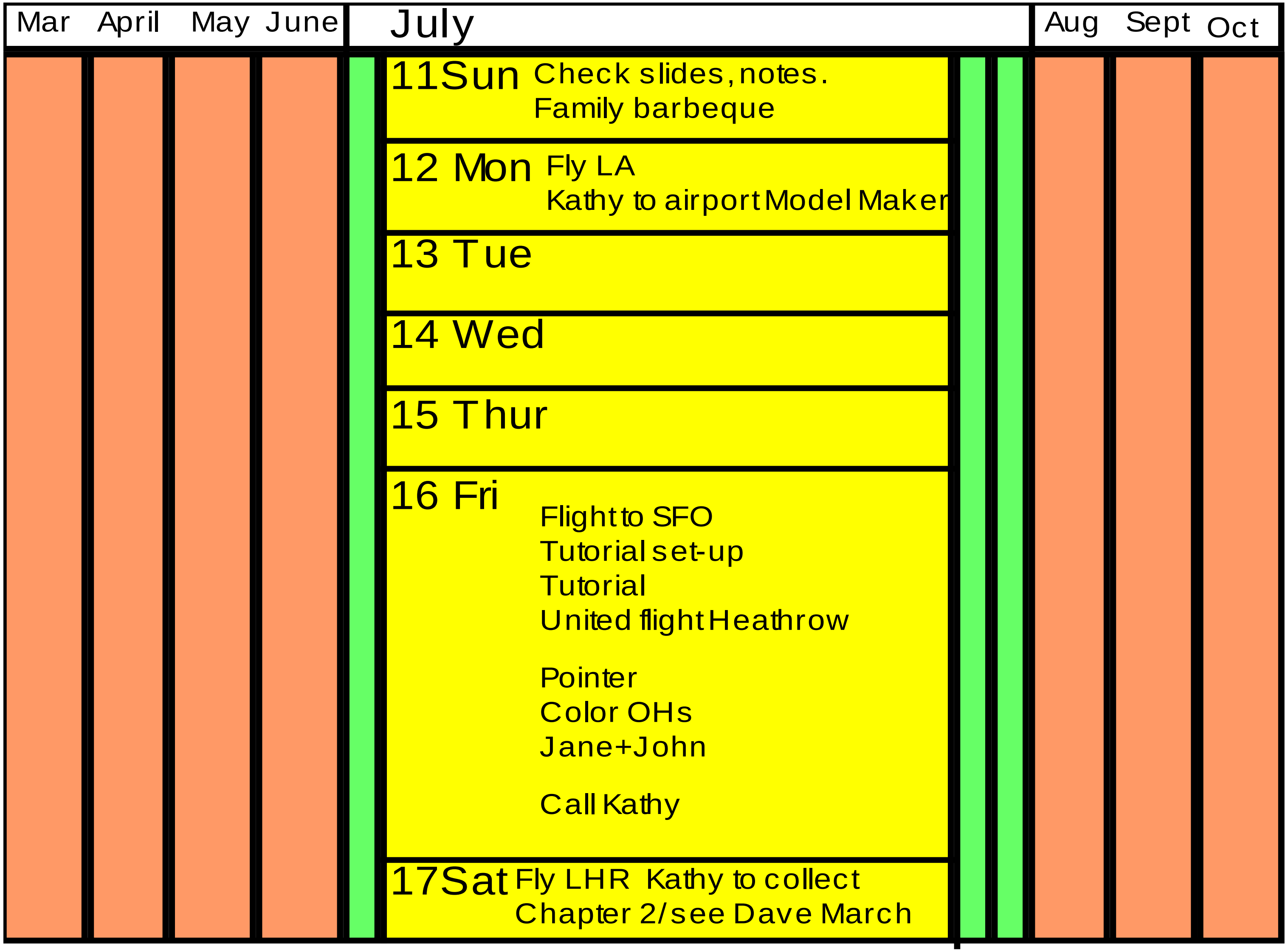
Superimpose Local Lens: Bifocal Magnified

- “Magnifying glass”
  zoomed image floats over overview image

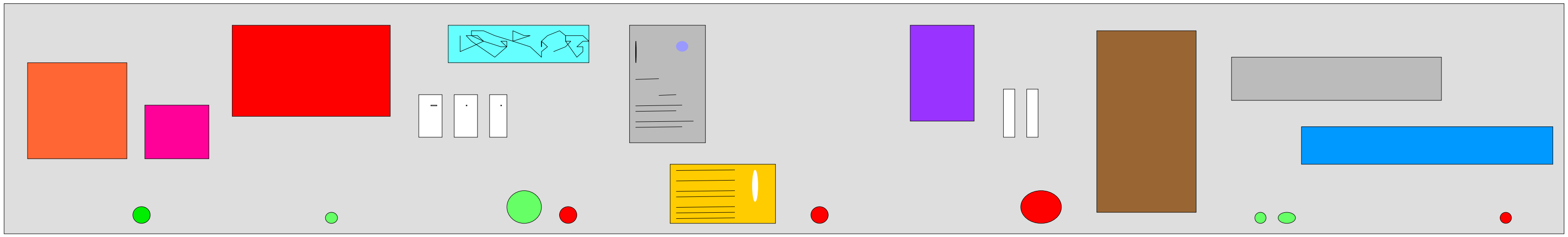
- Adjacent objects are obscured by the zoomed window



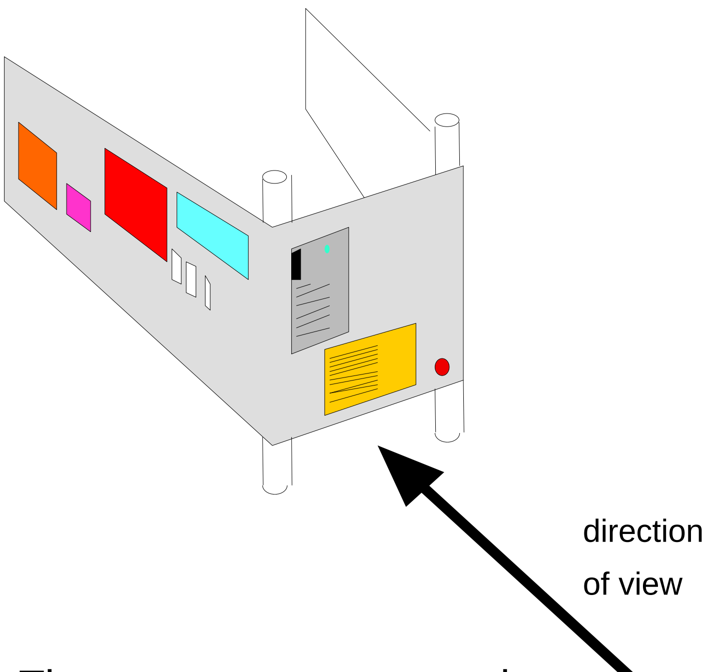
A First Bifocal Calendar (1980)



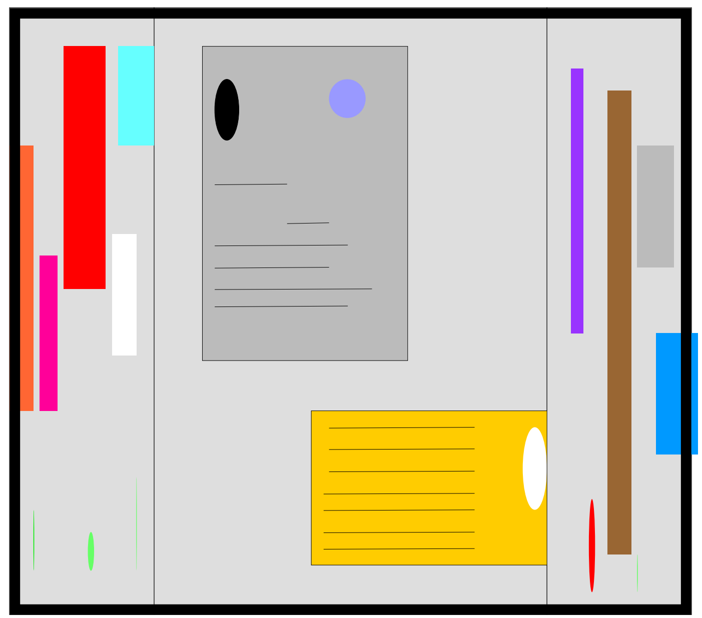
Bifocal Display (Spence & Apperly '82)



(a) An information space containing documents, emails, etc.

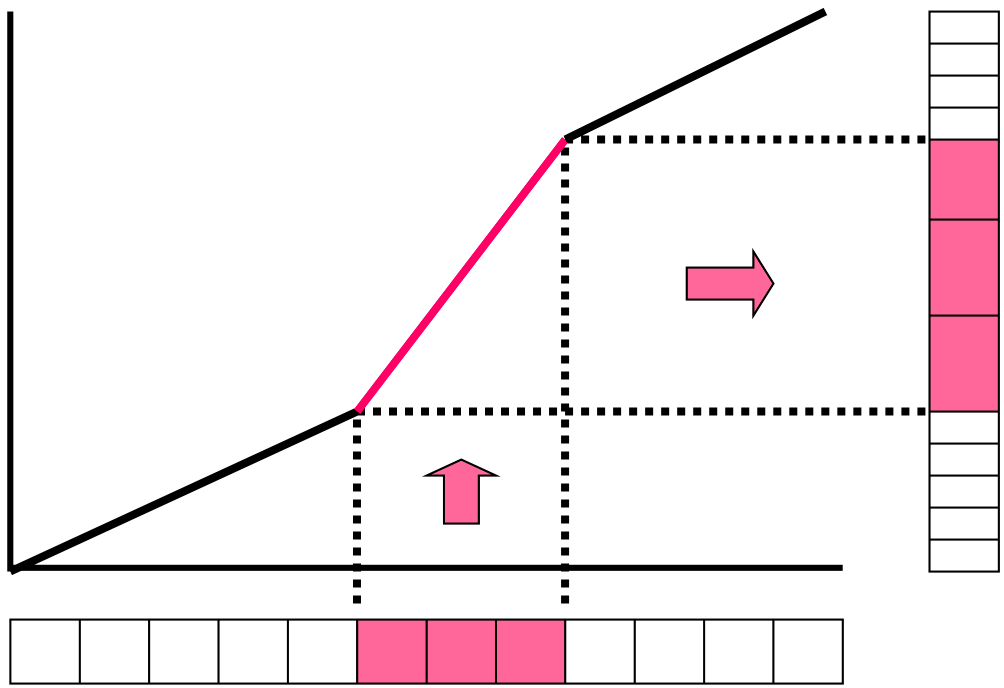


(b) The same space wrapped around two uprights.

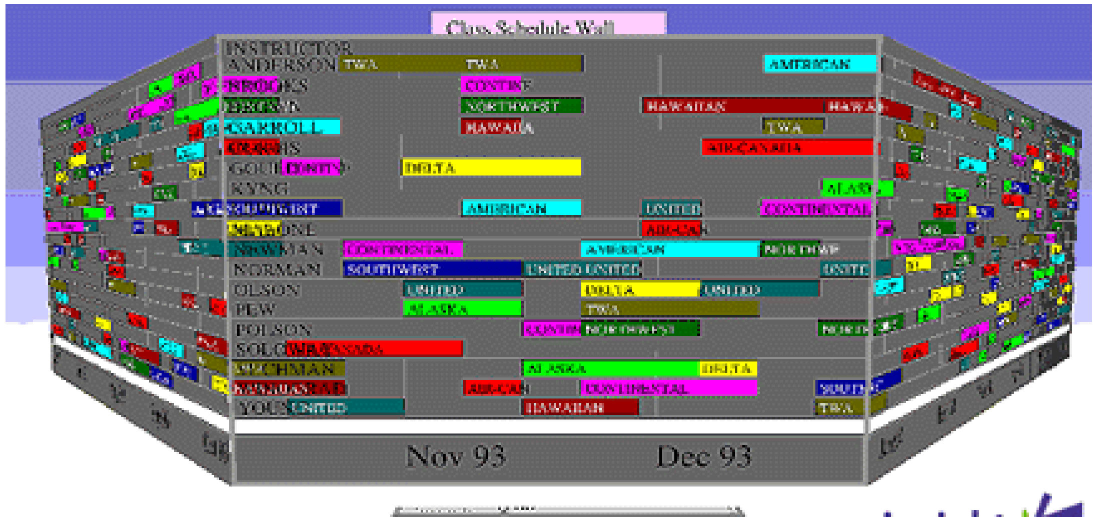


(c) Appearance of the information space when viewed from an appropriate direction

Visual Transfer Functions



Perspective Wall

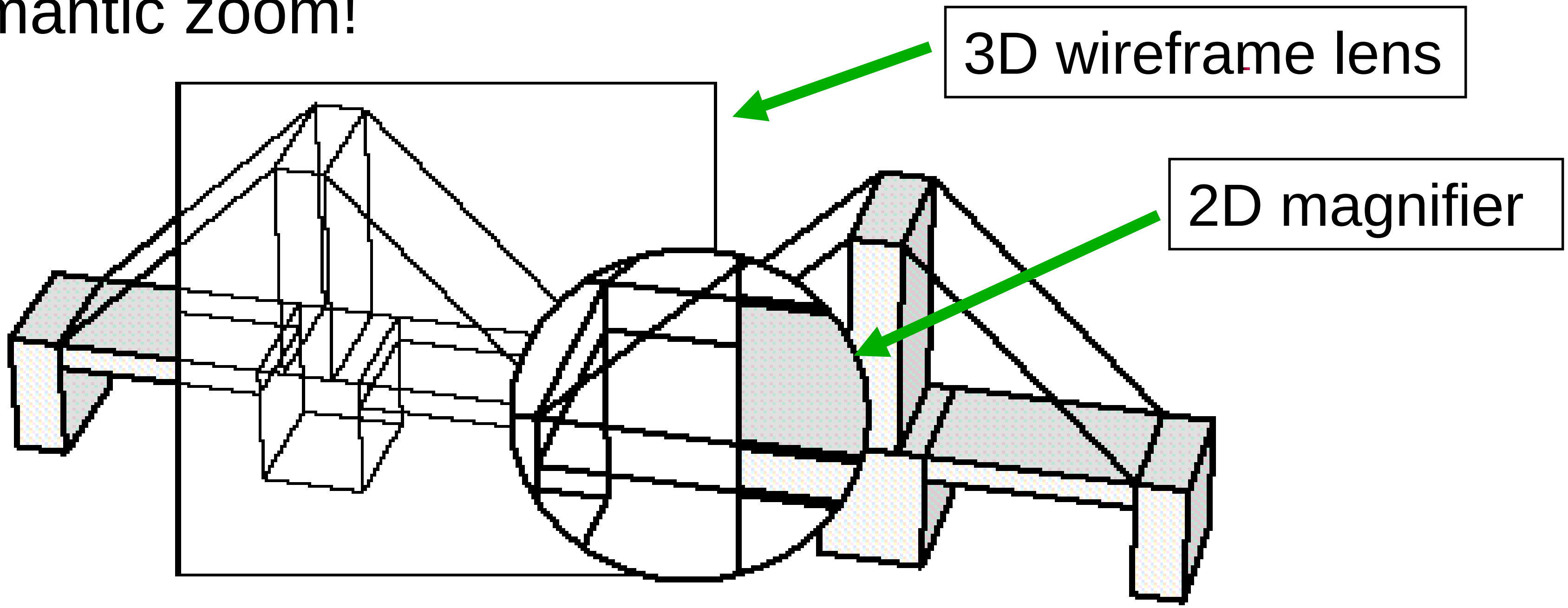


Mackinlay, Robertson, Card CHI'91

- 3D implementation of bifocal display
- Uses natural perspective to create focus and context
- Map work charts onto diagram, x-axis is time, y-axis is project

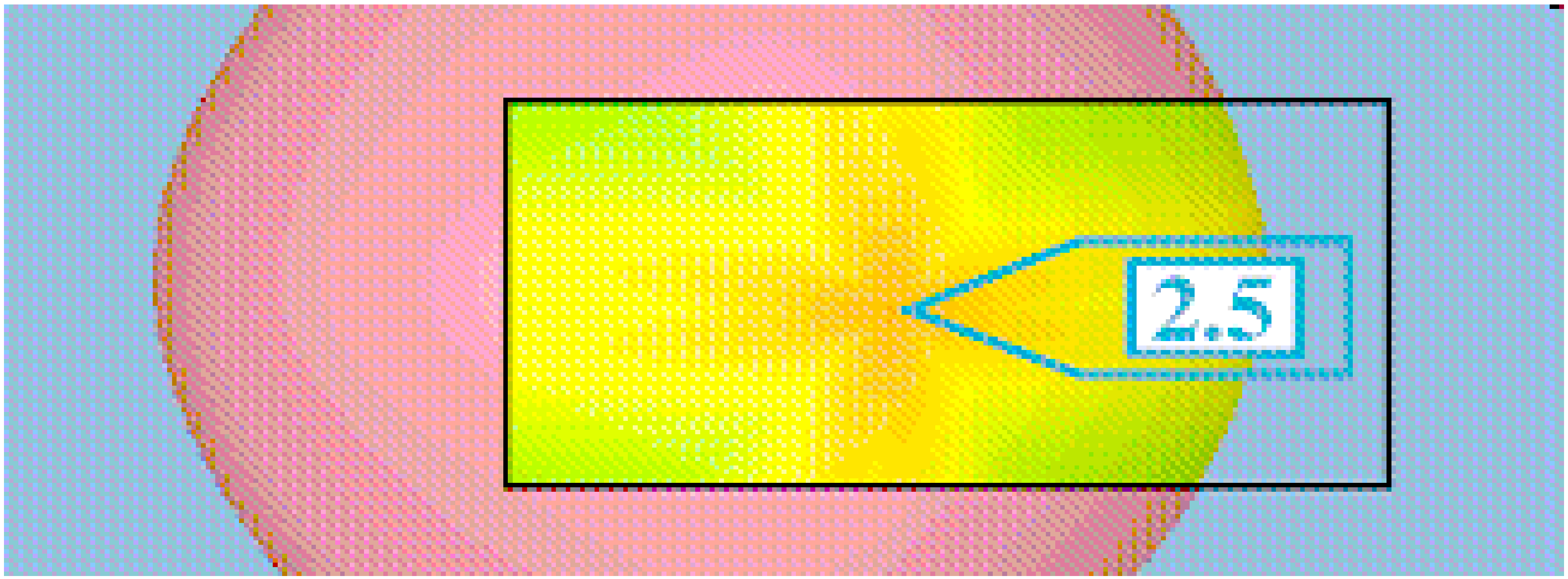
Superimpose Local Lens: Magic Lenses

- Regions/objects that can be moved around on display
- Semantically different content may be displayed in the lens e.g. semantic zoom!



Toolglass and Magic Lenses: The See-Through Interface, Siggraph 1993
Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, Tony D. DeRose
http://www2.parc.com/istl/projects/MagicLenses/93Siggraph.html

Superimpose Local Lens: Magic Lenses

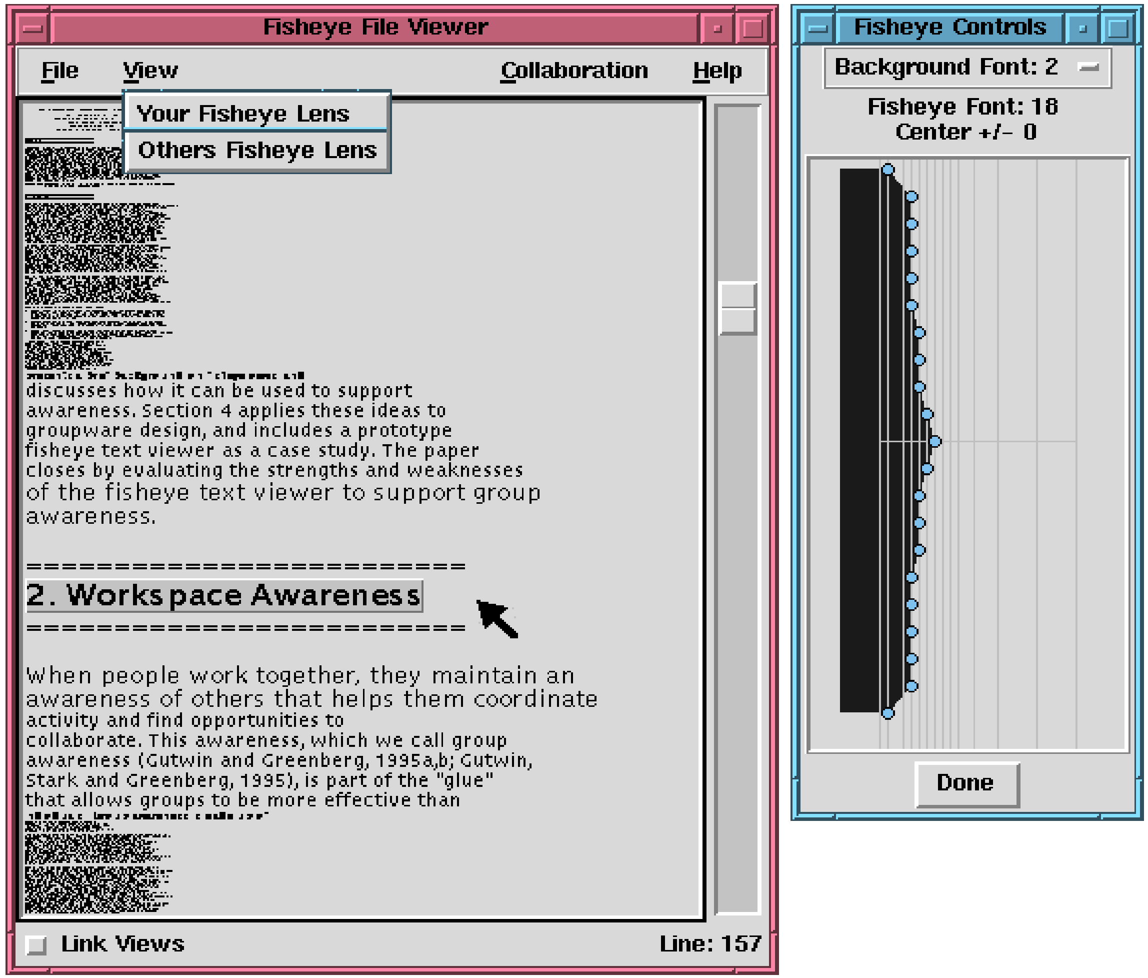


Gaussian curvature pseudo-color lens
Widget to read the value of the curvature

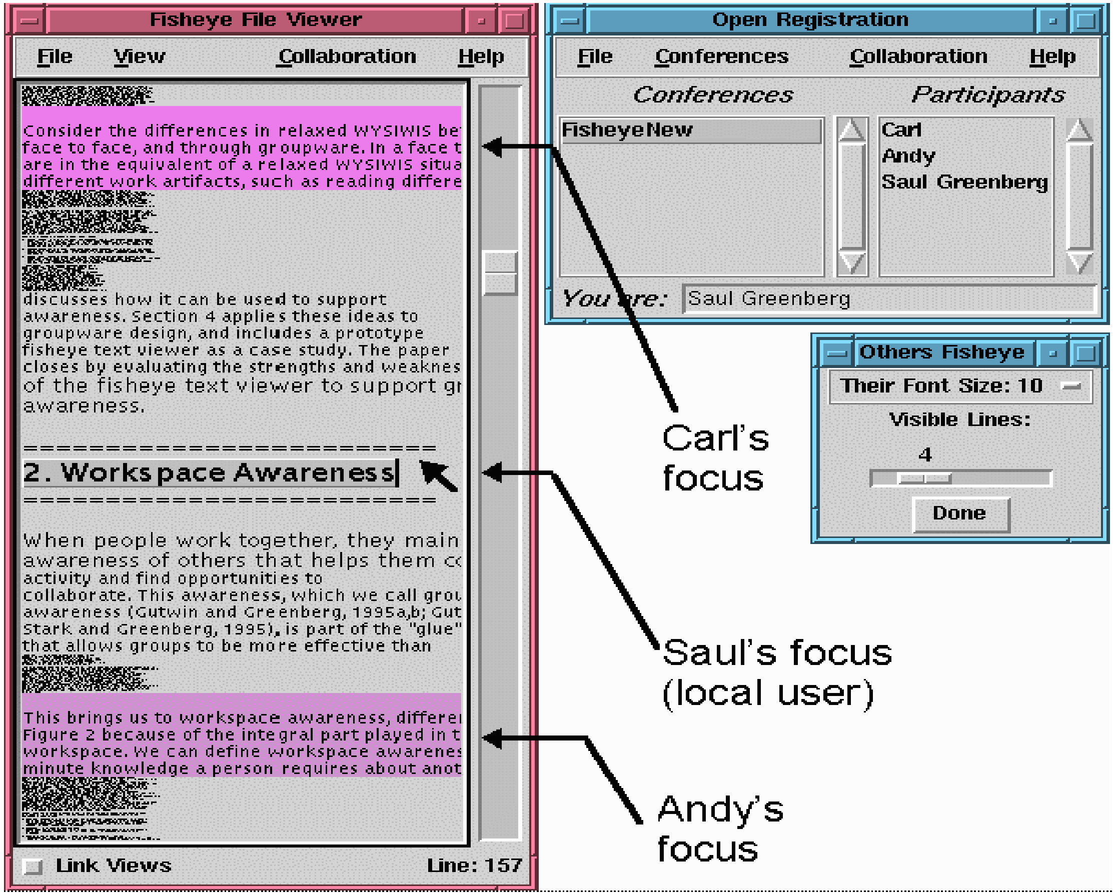
Example: Text/Program Viewing

Shown here are examples from Gutwin and Greenberg

Step function



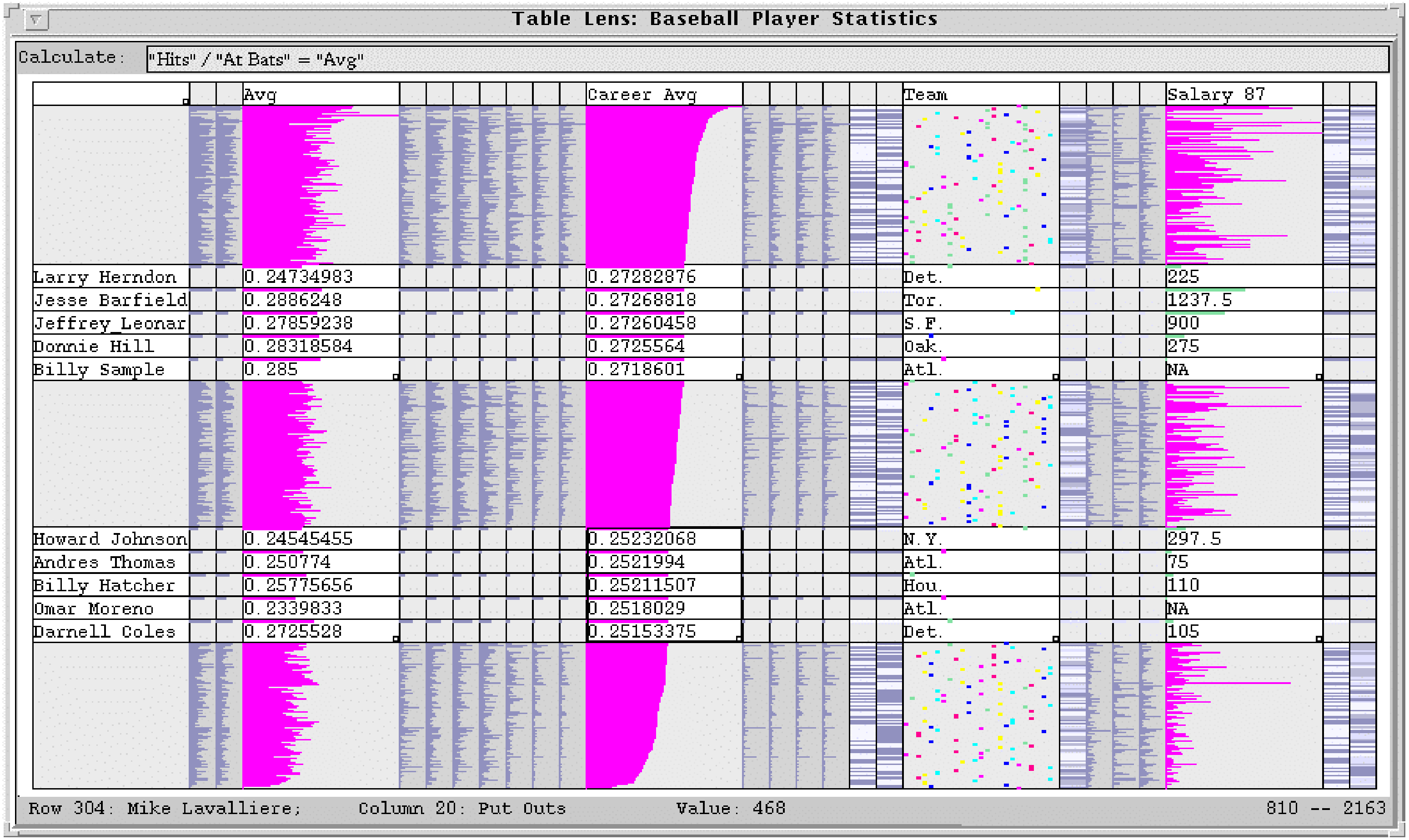
Applications



Shared text editor for CSCW

Gutwin and Greenberg, '96

Focus and Context Example: Eureka

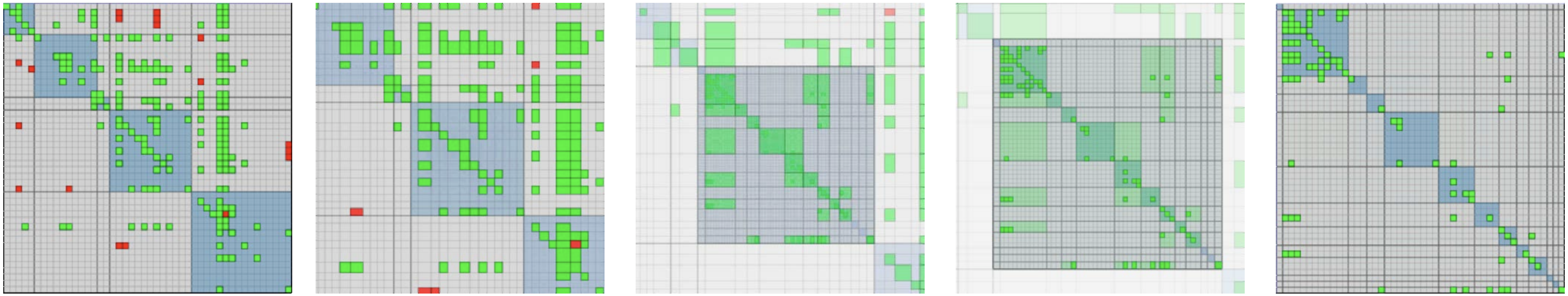


VIDEO

From Xerox PARC and Inxight
http://www.inxight.com/products/sdks/tl/

Important for Focus and Context: Animated transitions

- Smooth transition from one state to another
  - Alternative to jump cuts
  - Support for item tracking when amount of change is limited
  - Focus should open, shift and close in an animated way
- Animated transitions are useful in general
  - Example: multilevel matrix views
    - Scope of what is shown narrows down
      - Middle block stretches to fill space, additional structure appears within
      - Other blocks squish down to increasingly aggregated representations



[Using Multilevel Call Matrices in Large Software Projects. van Ham. Proc. IEEE Symp. Information Visualization (InfoVis), pp. 227–232, 2003.]

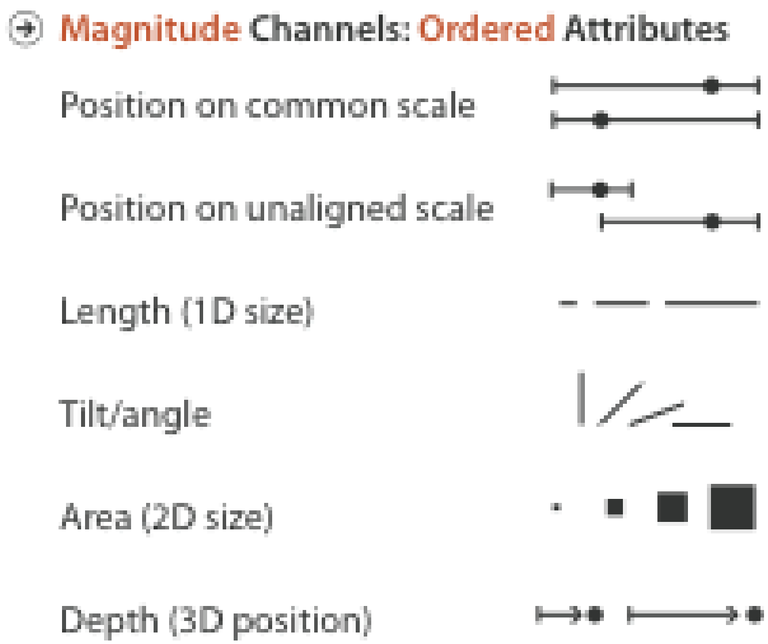
Munzner’s Rules of Thumb

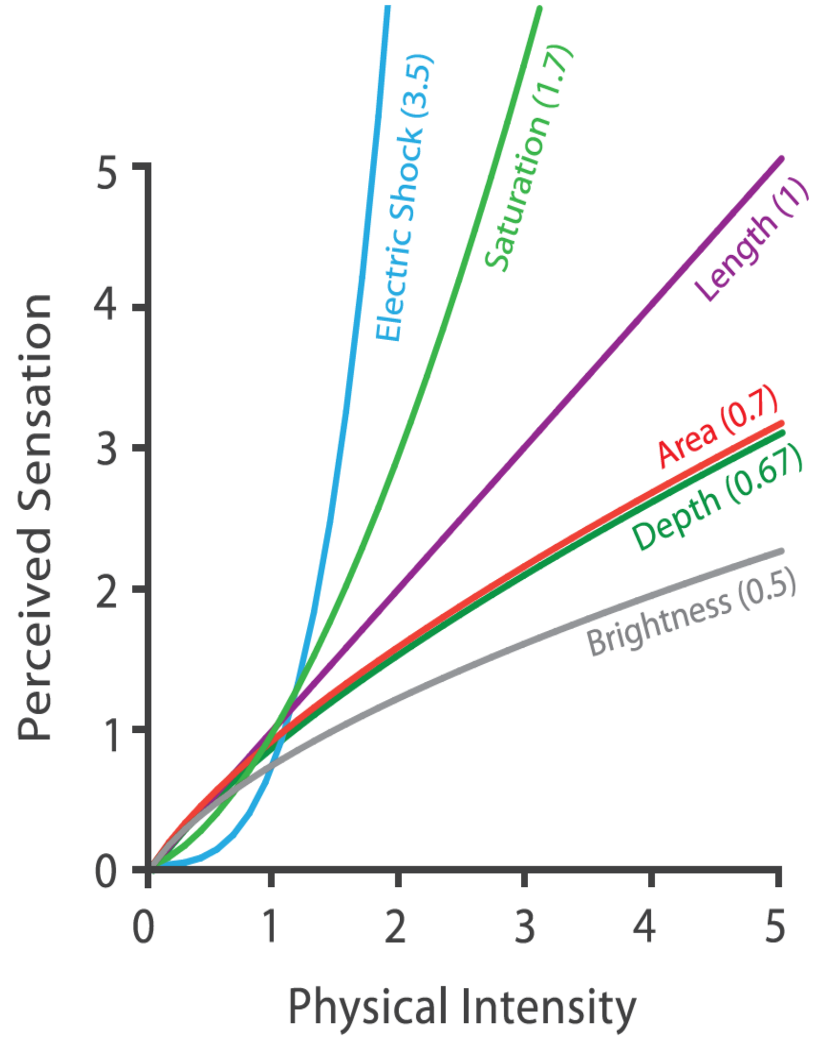
- No unjustified 3D
  - Power of the plane, dangers of depth
  - Occlusion hides information
  - Perspective distortion loses information
  - Tilted text isn’t legible
- No unjustified 2D
- Eyes beat memory
- Resolution over immersion
- Overview first, zoom and filter, details on demand
- Function first, form next

Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014. Chapter 6

No unjustified 3D: Power of the plane

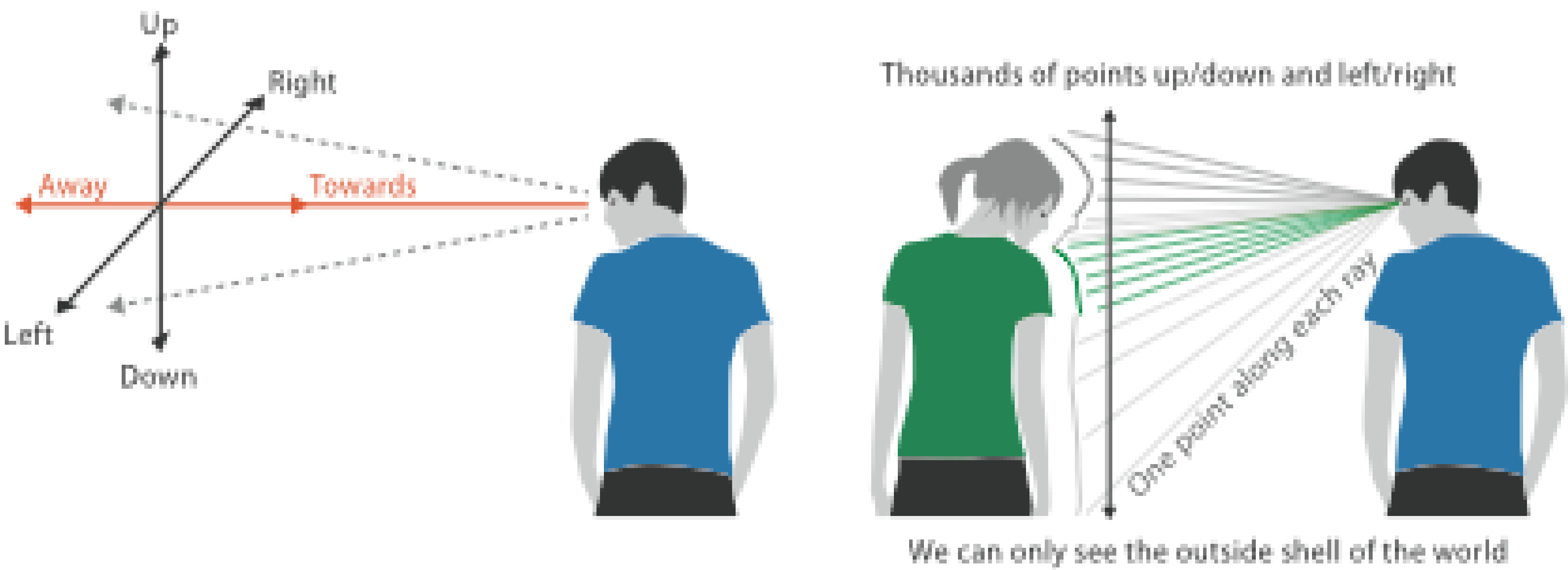
- High-ranked spatial position channels: planar spatial position
  - Not depth!





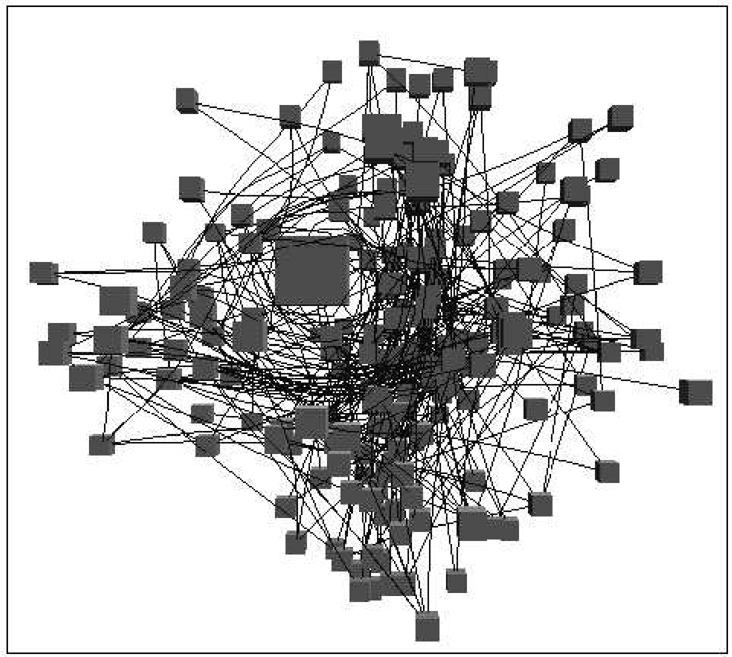
No unjustified 3D: Danger of depth

- Depth resolution is very limited
  - Acquire more info on image plane quickly from eye movements
  - Acquire more info for depth slower, from head/body motion



Occlusion hides information

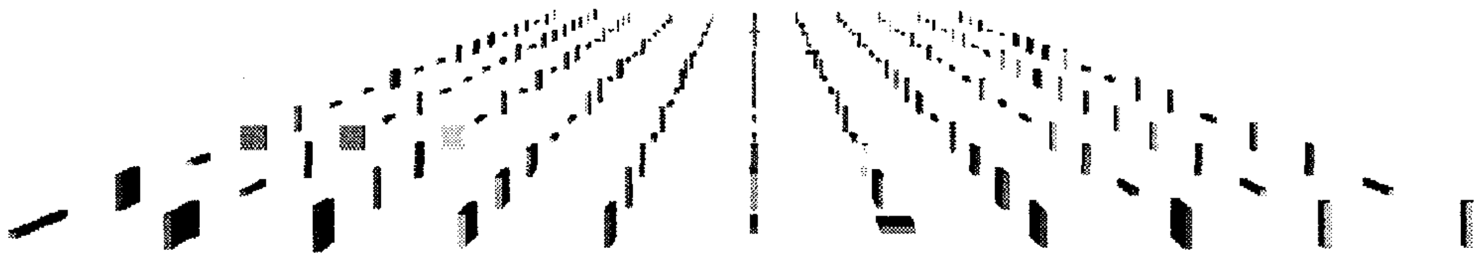
- Occlusion
- Interaction complexity: need to navigate, cut away, make transparent, ….



[Distortion Viewing Techniques for 3D Data. Carpendale et al. InfoVis1996.]

Perspective distortion loses information

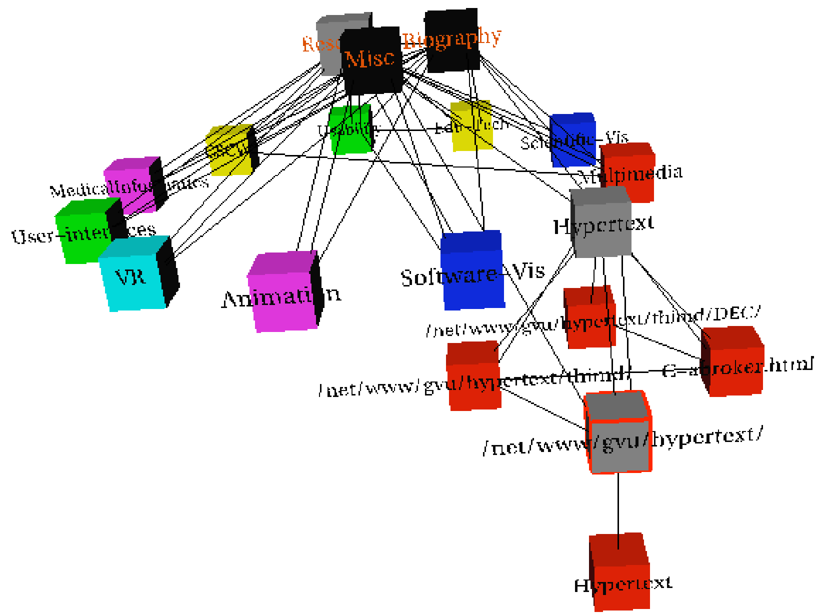
- Perspective distortion
  - Interferes with all size channel encodings
  - Power of the plane is lost!
  - But perspective is also a natural means for providing focus and context at the same time



[Visualizing the Results of Multimedia Web Search Engines. Mukherjea, Hirata, and Hara. InfoVis 96]

Tilted text isn't legible

- Text legibility
  - Far worse when tilted from image plane



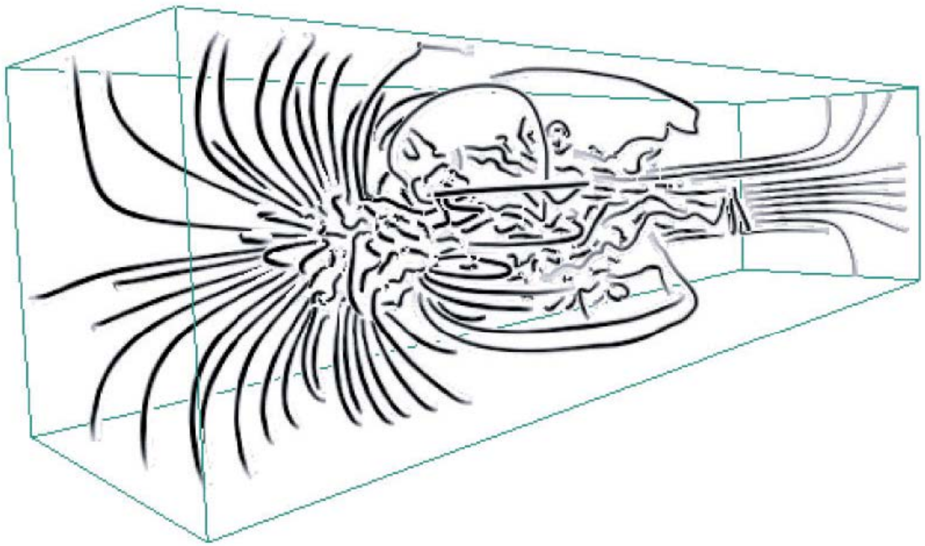
- Further reading

  [Exploring and Reducing the Effects of Orientation on Text Readability in Volumetric Displays. Grossman et al. CHI 2007]

[*Visualizing the World-Wide Web with the Navigational View Builder.* *Mukherjea and Foley. Computer Networks and ISDN Systems, 1995.]]*

Justified 3D: Shape Perception

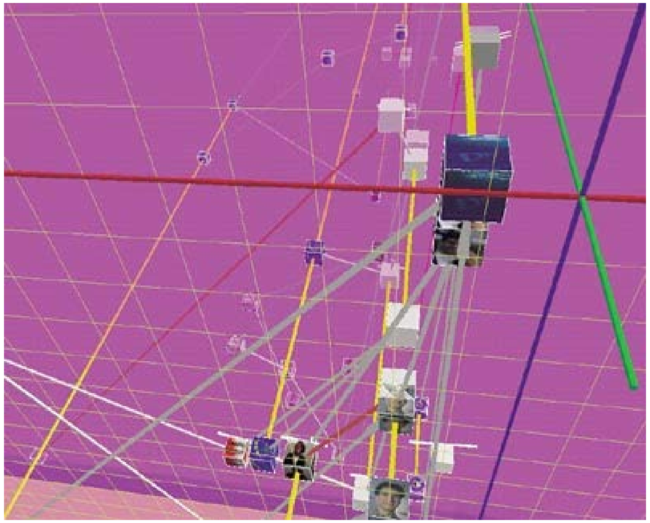
- Benefits outweigh costs when task is shape perception for 3D spatial data
  - Interactive navigation supports synthesis of a spatial mental model across many viewpoints



[Image-Based Streamline Generation and Rendering. Li and Shen. IEEE Trans. Visualization and Computer Graphics (TVCG) 13:3 (2007), 630–640.]

No Unjustified 3D

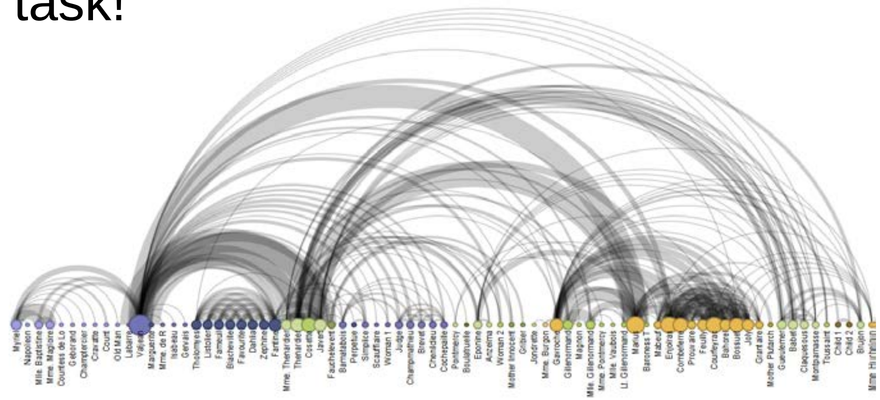
- 3D legitimate for true 3D spatial data
- 3D needs very careful justification for abstract data
  - Enthusiasm in 1990s, but now skepticism
  - Be especially careful with 3D for point clouds or networks

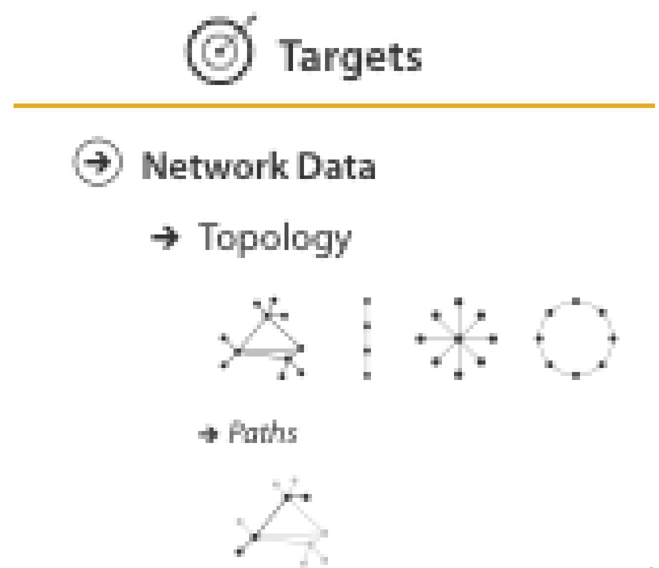


[WEBPATH-a three dimensional Web history. Frecon and Smith. Proc. InfoVis 1999]

No Unjustified 2D

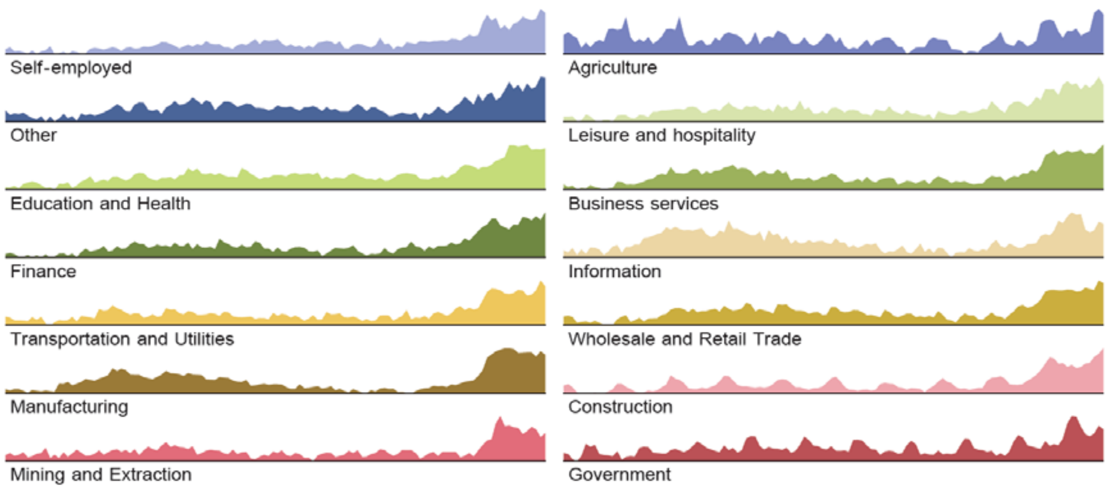
- Consider whether network data requires 2D spatial layout
  - Especially if reading text is central to task!
  - Arranging as network means lower information density and harder label lookup compared to text lists
  - 1D layout might also work well
- Benefits outweigh costs when topological structure/context important for task
  - Be especially careful for search results, document collections, ontologies
  - Lists can be fine …





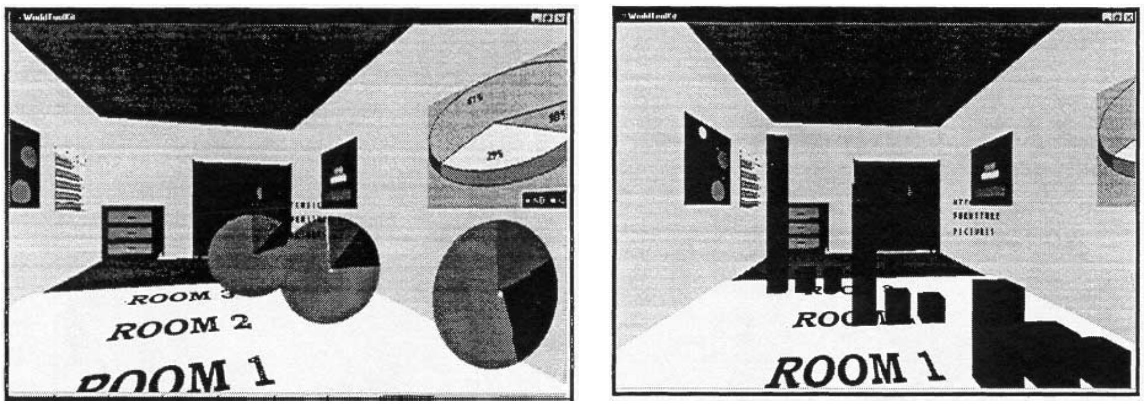
Eyes Beat Memory

- Principle: external cognition vs. internal memory
  - Easy to compare by moving eyes between side-by-side views
  - Harder to compare visible item to memory of what you saw
- Implications for animation
  - Great for choreographed storytelling
  - Great for transitions between two states
  - Poor for many states with changes everywhere
    - Consider small multiples instead



Resolution Beats Immersion

- Immersion typically not helpful for abstract data
  - Do not need sense of presence or stereoscopic 3D
- Resolution much more important
  - Pixels are the scarcest resource
  - Desktop also better for workflow integration
- Virtual reality for abstract data very difficult to justify
  - But of course for spatial data it can be quite suitable!



[Development of an information visualization tool using virtual reality. Kirner and Martins. Proc. Symp. Applied Computing 2000]

Form Follows Function

- Start with focus on functionality
  - Often aesthetics can be improved later on, as refinement
  - Good graphics design can significantly improve a visualization

- Don’t start with aesthetics
  - … but possibly with an idea for a visual design and layout of the data
  - However, functionality is core and is often hard to add later

Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - Chap 6: Rules of Thumb
- Visual Thinking for Design. Ware. Morgan Kaufmann, 2008.
- Information Visualization: Perception for Design, 3rd edition. Ware. Morgan Kaufmann /Academic Press, 2004.
- The use of 2-D and 3-D displays for shape understanding versus relative position tasks. St. John, Cowen, Smallman, and Oonk. Human Factors 43:1 (2001), 79–98.
- Evaluating Spatial Memory in Two and Three Dimensions. Cockburn and McKenzie. Intl. Journal of Human-Computer Studies 61:30 (2004), 359–373.
- Supporting and Exploiting Spatial Memory in User Interfaces. Scarr, Cockburn, and Gutwin. Foundations and Trends in Human Computer Interaction, 6. Now, 2013.
- Effectiveness of Animation in Trend Visualization. Robertson, Fernandez, Fisher, Lee, and Stasko. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis08) 14:6 (2008), 1325–1332.
- Animation: can it facilitate? Tversky, Morrison and Betrancourt. Intl Journ Human-Computer Studies, 57(4):247-262, 2002.
- Current approaches to change blindness. Simons. Visual Cognition 7:1/2/3 (2000), 1–15.
- The Non-Designer's Design Book, 3rd ed. Williams. Peachpit Press, 2008.

End

Visualization

Tabular Data / Multi-Attribute Data

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Big NEWS

- Grading
  - **No final exam this semester**
  - Assignments (40%)
  - Final project (individual projects or in groups of two, workload about 2 weeks) (60%)
    - For groups: each group member has to present his part of the project separately and has to make arguments that relate your work to Munzner’s what-why-how framework and the techniques introduced in this course
  - Final project be turned in at any time during the semester break but needs to be turned in at the latest by September 30!

Acknowledgements

- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Harald Reiterer, Universität Konstanz
  - Daniel A. Keim, Universität Konstanz
- Also based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.

Tabular Data / Multi-Attribute Data

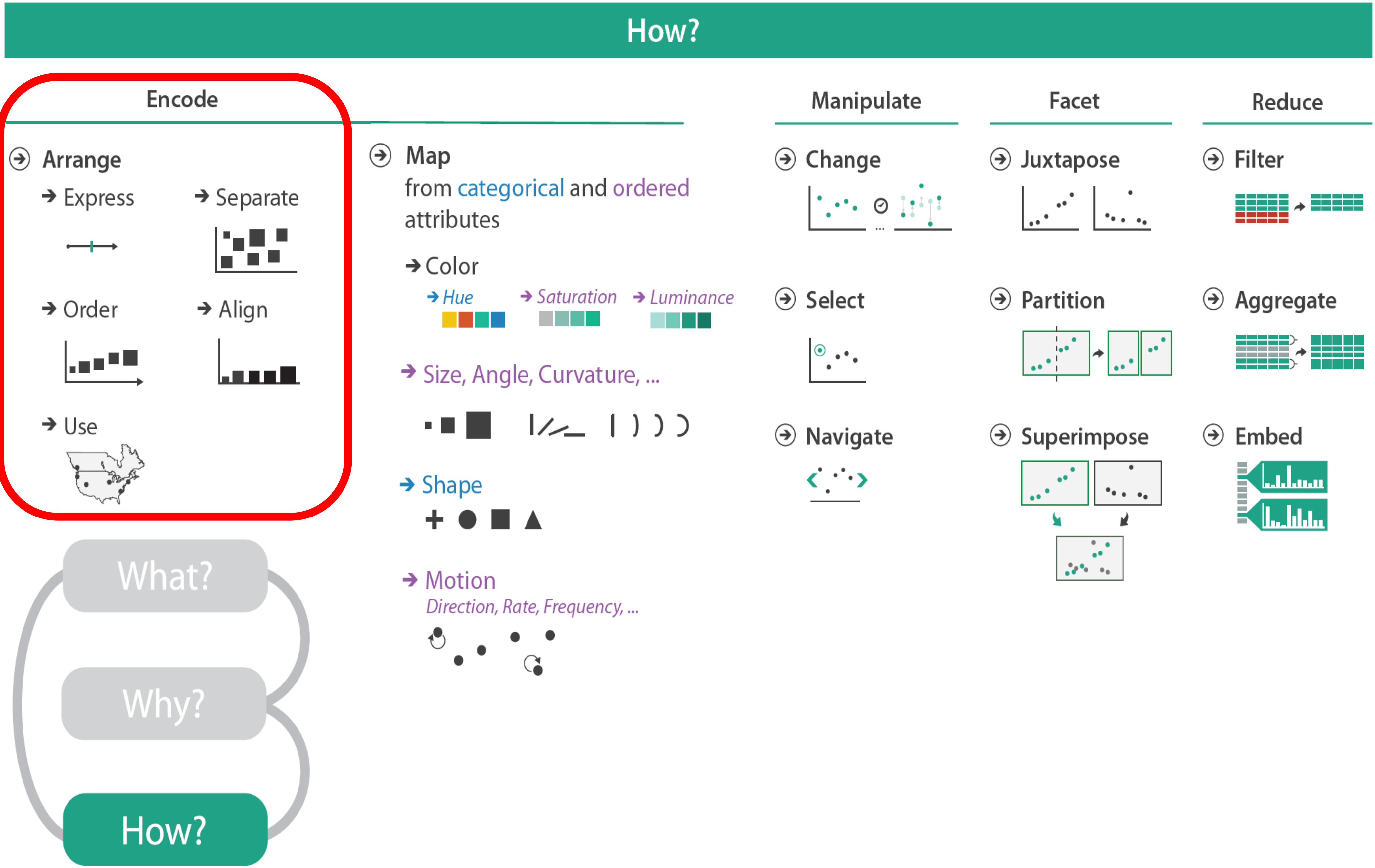
A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	[illegible]	7/17/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
32	7/16/07	2-High	Medium Box	[illegible]	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	[illegible]	5-Low	Wrap Bag	0.56	1/20/05
69	[illegible]	4-Not Specified	Small Pack	0.44	6/6/05
69	[illegible]	4-Not Specified	Wrap Bag	0.6	6/6/05
70	12/18/06	5-Low	Small Box	0.59	12/23/06
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06
96	4/17/05	2-High	Small Box	0.55	4/19/05
97	1/29/06	3-Medium	Small Box	0.38	1/30/06
129	11/19/08	5-Low	Small Box	0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

attribute

item

cell

In each cell, an attribute value is stored



How?
Encode
Arrange
Express
Separate
Order
Align
Use
Map
from categorical and ordered attributes
Color
Hue
Saturation
Luminance
Size, Angle, Curvature, ...
Shape
Motion
Direction, Rate, Frequency, ...
Manipulate
Change
Select
Navigate
Facet
Juxtapose
Partition
Superimpose
Reduce
Filter
Aggregate
Embed
What?
Why?
How?

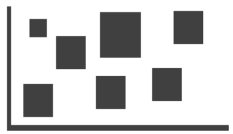
Arrange Space

Encode

Arrange

- Express

- Separate



- Order

- Align

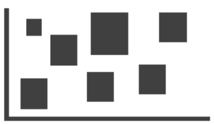
- Use

Arrange Tabular Data

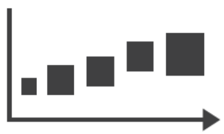
Express Values

Separate, Order, Align Regions

Separate



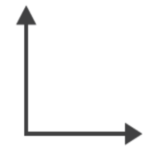
Order



Align

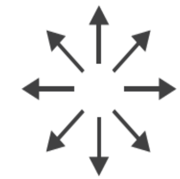
Axis Orientation

Rectilinear



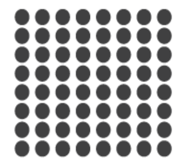
Parallel

Radial



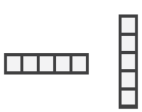
Layout Density

Dense

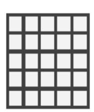


Space-Filling

1 Key
List



2 Keys
Matrix



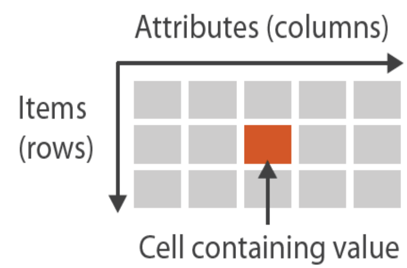
3 Keys
Volume

Many Keys
Recursive Subdivision

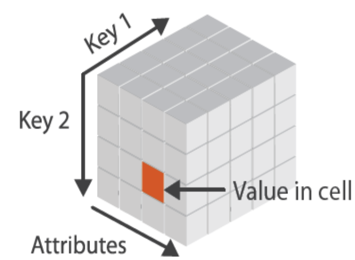
Keys and Values

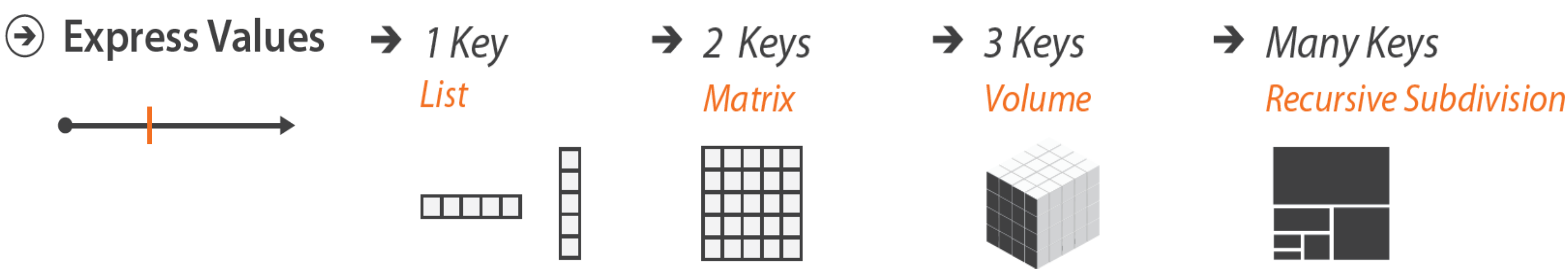
- Key
  - Independent attribute
  - Used as unique index to look up items
  - Simple tables: 1 key – often implicit
  - Multidimensional tables: multiple keys
- Value
  - Dependent attribute, value of cell
- Classify arrangements by key count
  - 0, 1, 2, many...

➔ Tables



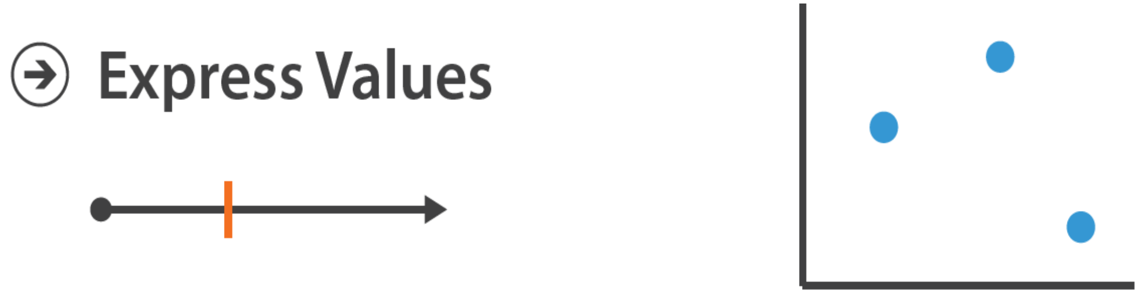
➔ *Multidimensional Table*

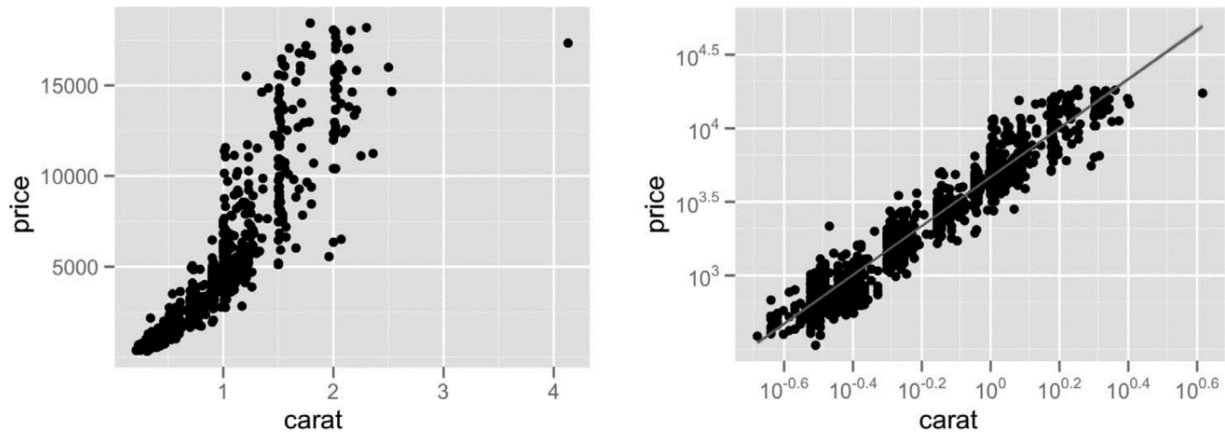




Technique: Scatterplot

- Express values
  - Quantitative attributes mapped to position
- Data
  - No keys, only values
  - 2 quant attributes
- Mark: points
- Channels
  - Horiz + vert position
- Tasks
    - Find trends, outliers, distribution, correlation, clusters
  - Scalability
    - Hundreds of items

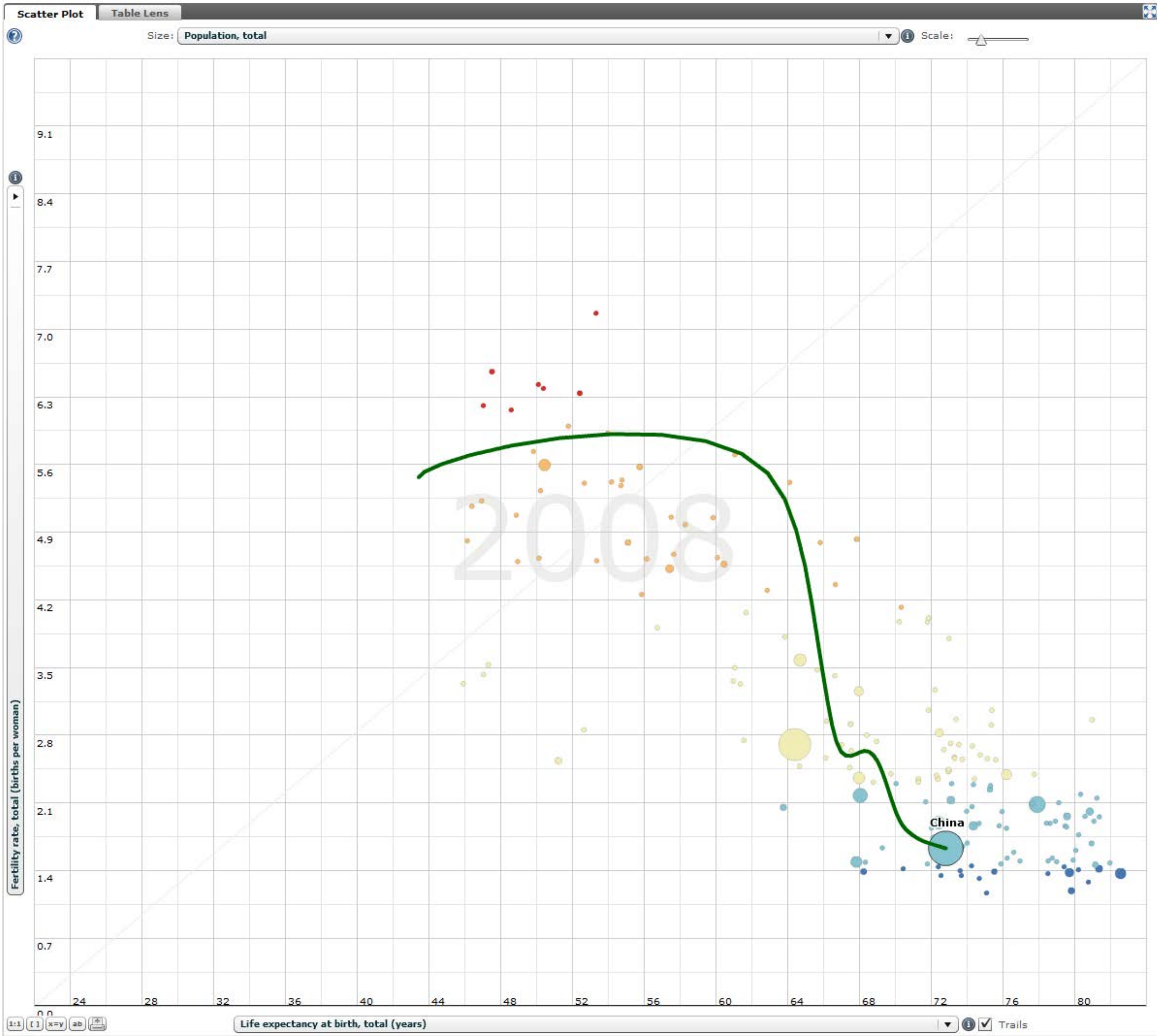




[A layered grammar of graphics. Wickham. Journ. Computational and Graphical Statistics 19:1 (2010), 3–28.]

Scatterplot (Bubbleplot) Example

- Data
  - Key: country (not encoded)
  - 3 quant attributes
- Mark: points
- Channels
  - Horiz + vert position
  - Size (population)
  - Color (redundant)
- The trail encodes development over time for a single data item

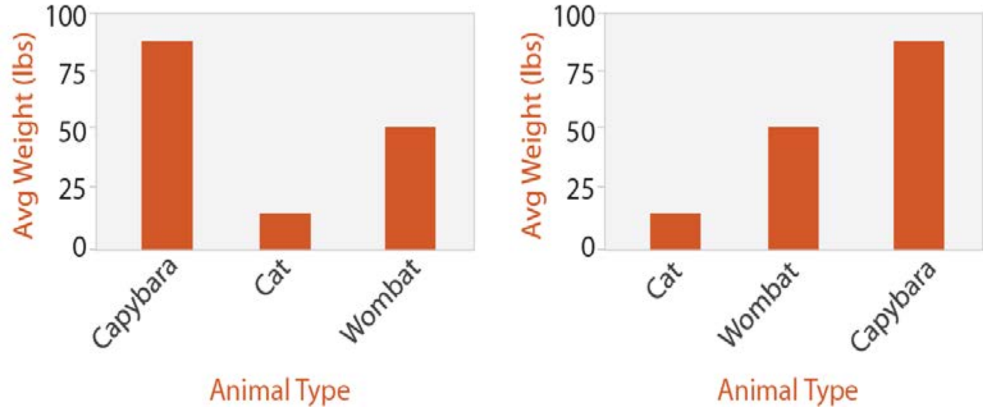


http://www.ncomva.se/apps/mdim/#story=5

Categorical Data and Keys

- Separate items into regions – especially for categorical attributes
  - Following expressiveness principle for categorical attributes
- Use ordered attribute to order and align regions

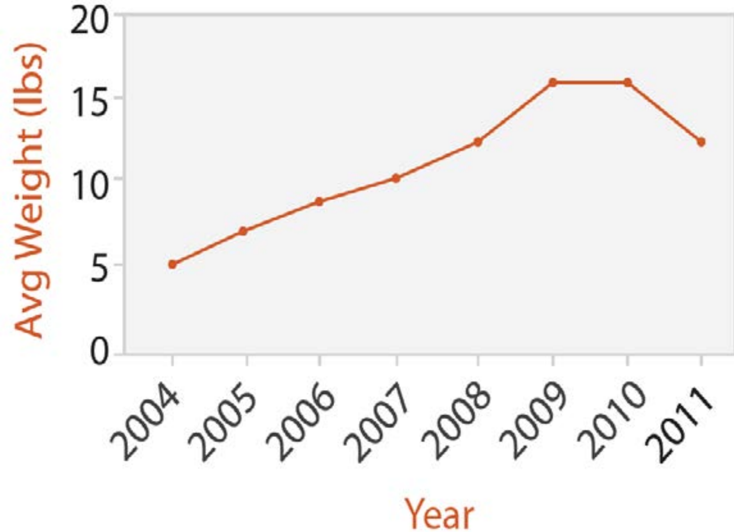
Technique: Bar Chart



- Data
  - One key, one value
    1 categ attrib, 1 quant attrib
- Mark: lines
- Channels
  - Length to express quantitative value
  - Spatial regions: one per mark (left, middle, right)
    - Separated horizontally, aligned vertically
    - Ordered by quantitative attribute
      - By label (alphabetical),
      - By length attribute (data-driven)
- Task
  - Compare, lookup values
- Scalability
  - Dozens to hundreds of values for key attributes

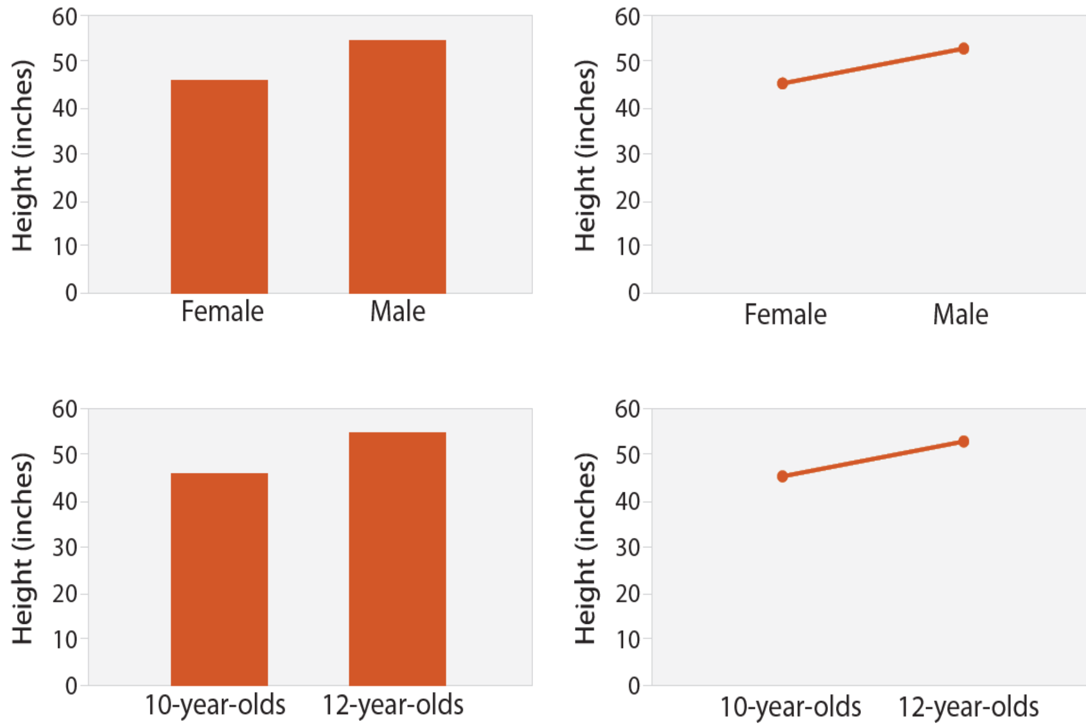
Technique: Line Chart

- Data
  - One key (year), one value (weight)
  - 2 quant attribs (weight as function of year)
- Mark: points
  - Line connection marks between them
- Channels
  - Aligned lengths to express quant value
  - Separated and ordered by key attribute into horizontal regions
- Task
  - Find trend
    - Connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next



Choosing Bar vs Line Charts

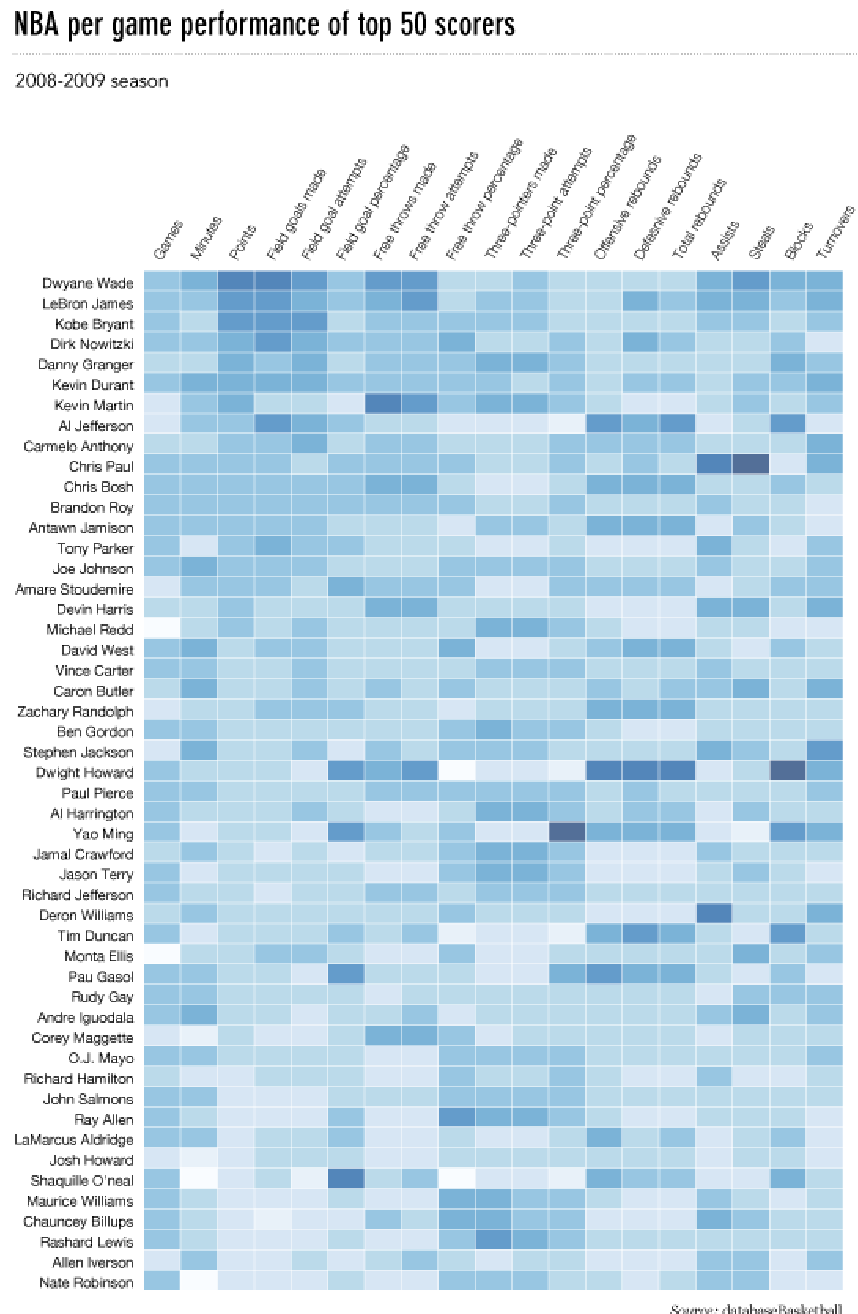
- Depends on type of key attribute
  - Bar charts if categorical
  - Line charts if ordered
- Do not use line charts for categorical key attributes
  - Violates expressiveness principle
    - Implication of trend so strong that it overrides semantics!
      - “The more male a person is, the taller he/she is”



after [Bars and Lines: A Study of Graphic Communication. Zacks and Tversky. Memory and Cognition 27:6 (1999), 1073–1079.]

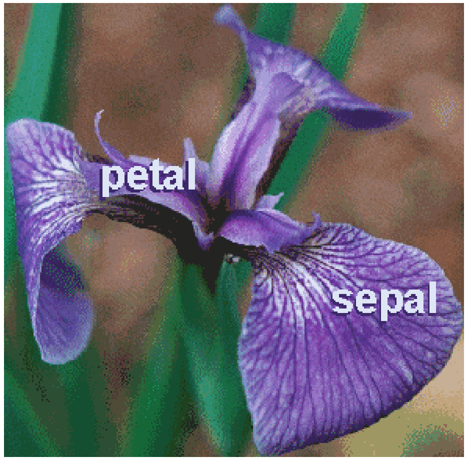
Technique: Heatmap

- Data: two keys, one value
  - 2 categ attribs (person, performance)
  - 1 quant attrib (expression levels)
- Marks: area
  - separate and align in 2D matrix
    - indexed by 2 categorical attributes
- Channels
  - Color by quant attrib
    - (ordered diverging colormap)
- Task
  - Find clusters, outliers
  - Ordering important – sorted by Points!
- scalability
  - 1M items,
    100s of different categorical values (levels),
    ~10 quant attribute levels



http://flowingdata.com/2010/01/21/how-to-make-a-heatmap-a-quick-and-easy-solution/

Multi-Attribute Data: Iris Dataset





Setosa Versicolor Virginica

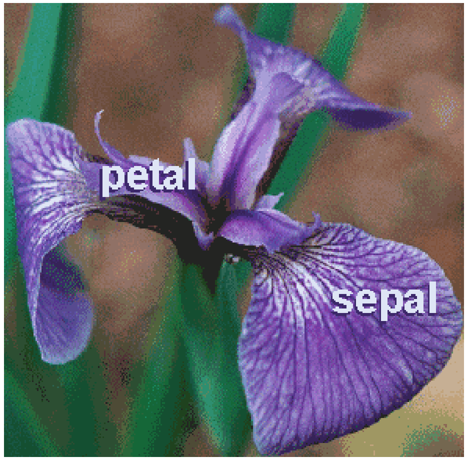
Anderson's data of iris flowers on the Gaspé Peninsula, Quebec, Canada

In German:
Schwertlilie
Sepal: Kelchblatt
Petal: Blütenblatt

- 4 features (attributes) with numerical values
  - sepal length in cm
  - sepal width in cm
  - petal length in cm
  - petal width in cm
- 1 categorical attribute: 3 different species: Iris Setosa, Iris Versicolour, Iris Virginica
- Data size: 150 items
- Data distribution: 50 entries for each flower species
- Table:

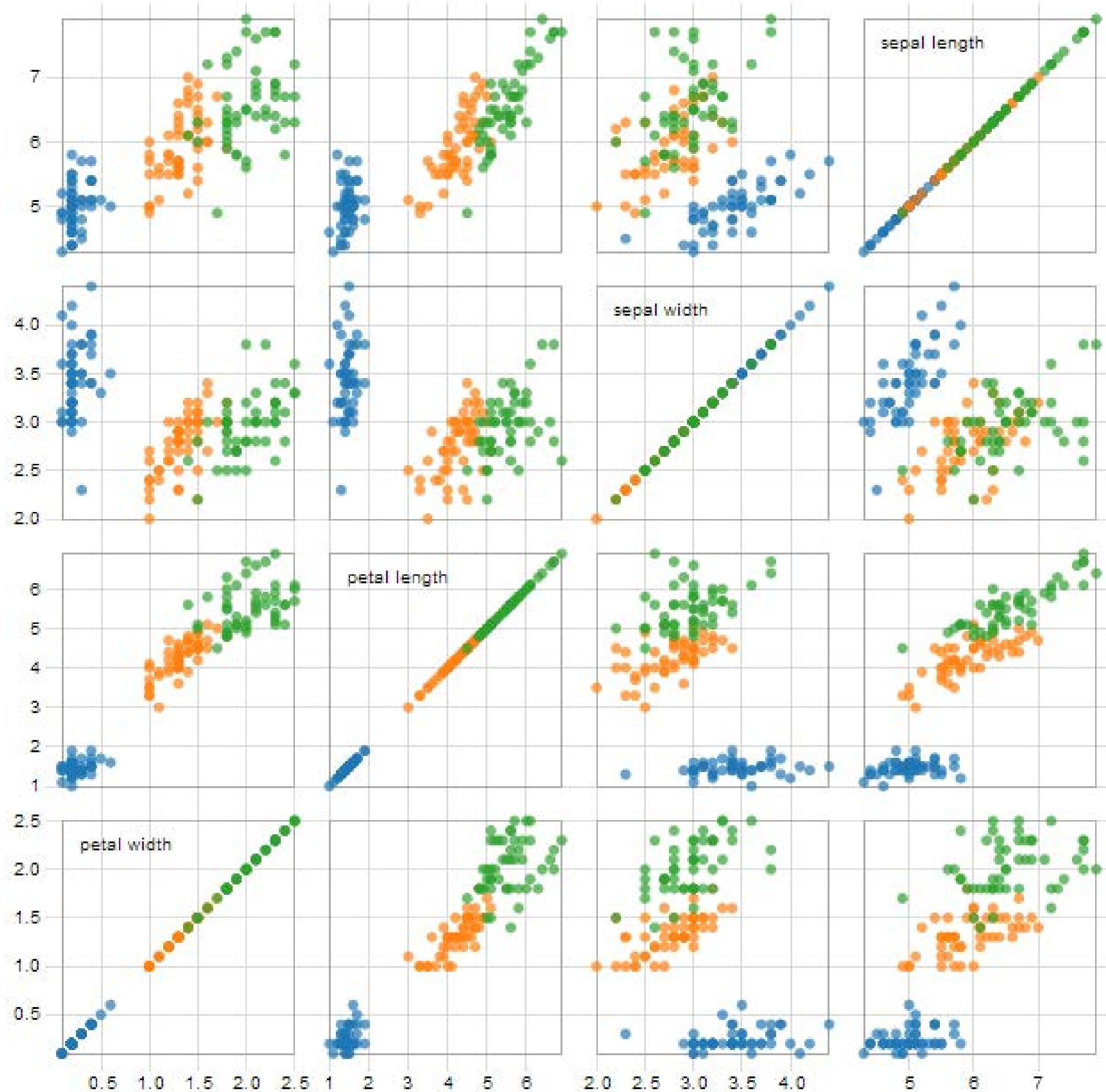
```
5.1,3.5,1.4,0.2,setosa
4.9,3.0,1.4,0.2,setosa
4.7,3.2,1.3,0.2,versicolour
4.6,3.1,1.5,0.2,virginica
...
```

Scatterplot Matrix (SPLOM)

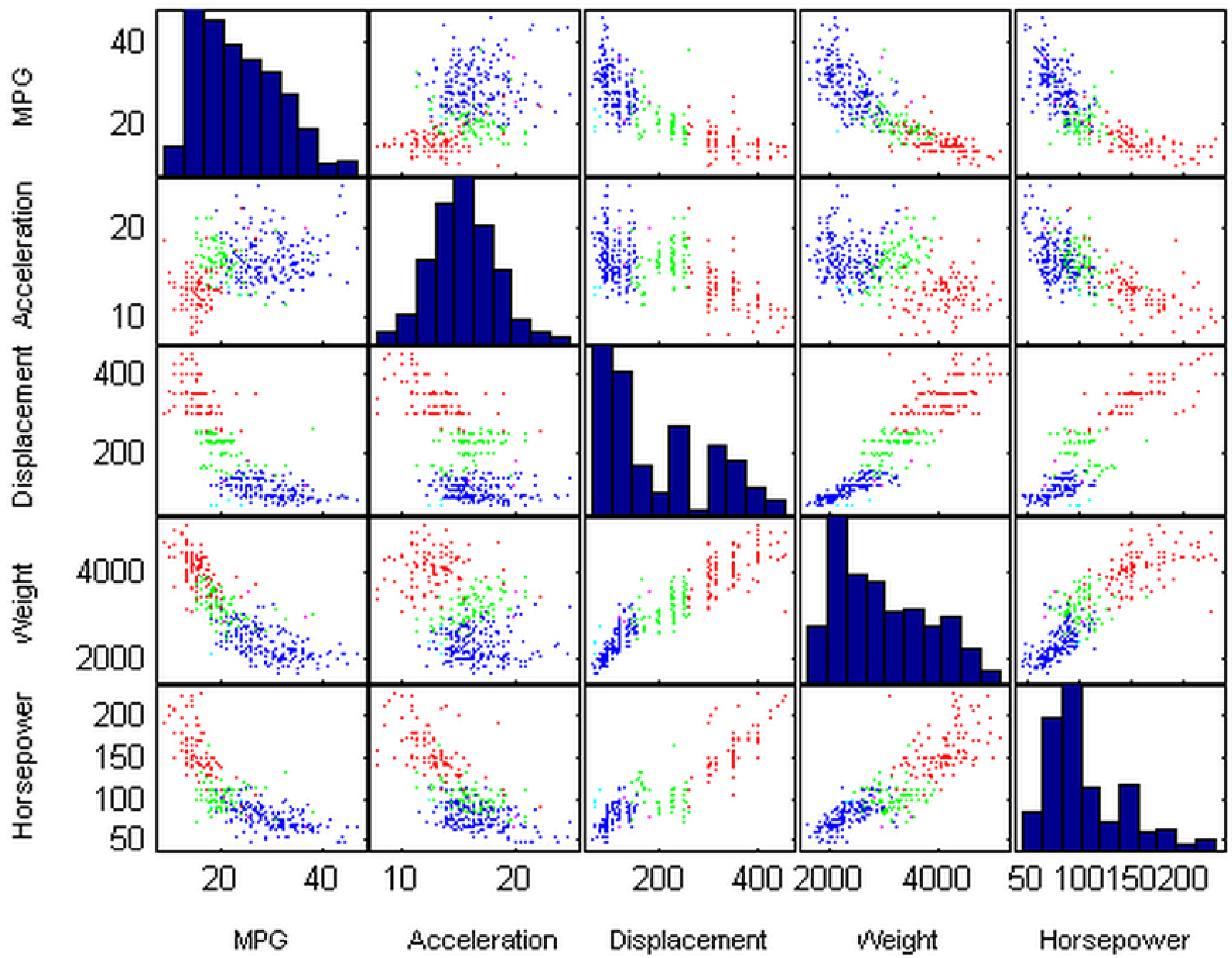


Setosa Versicolor Virginica

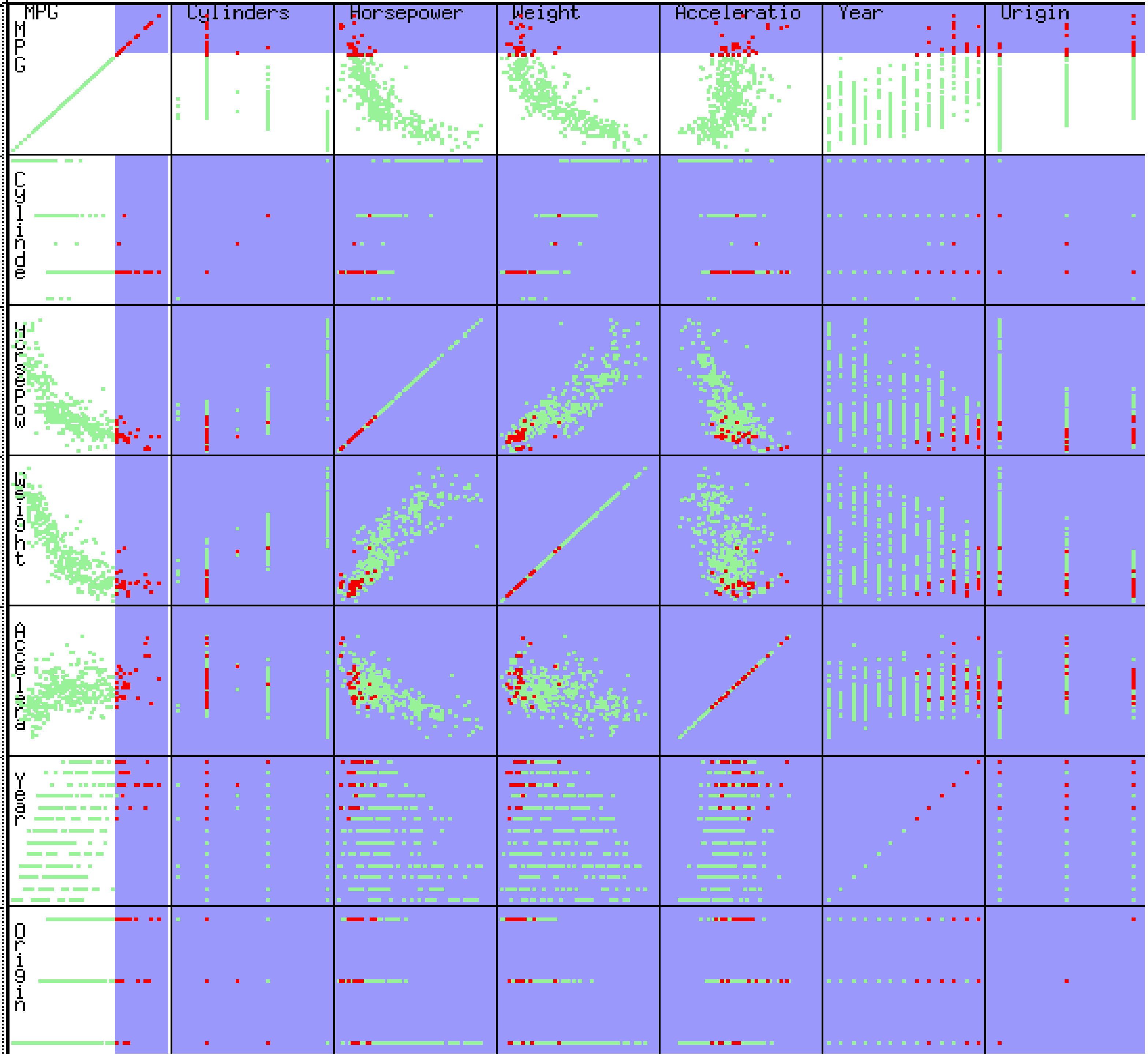
Anderson's data of iris flowers on the Gaspé Peninsula, Quebec, Canada



Scatterplot Matrix: Car Dataset



• 3 cylinders • 4 cylinders • 5 cylinders • 6 cylinders • 8 cylinders

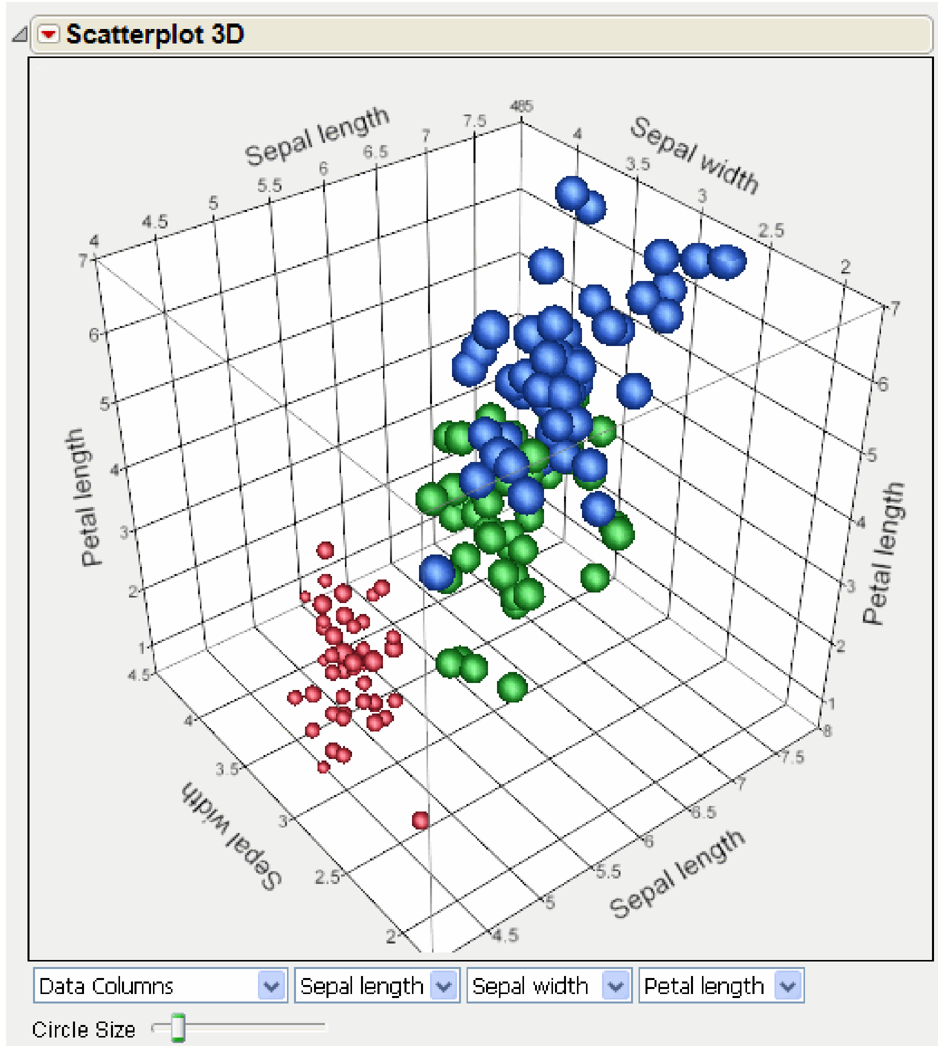


Brushing and Linking with Scatter Plot Matrix

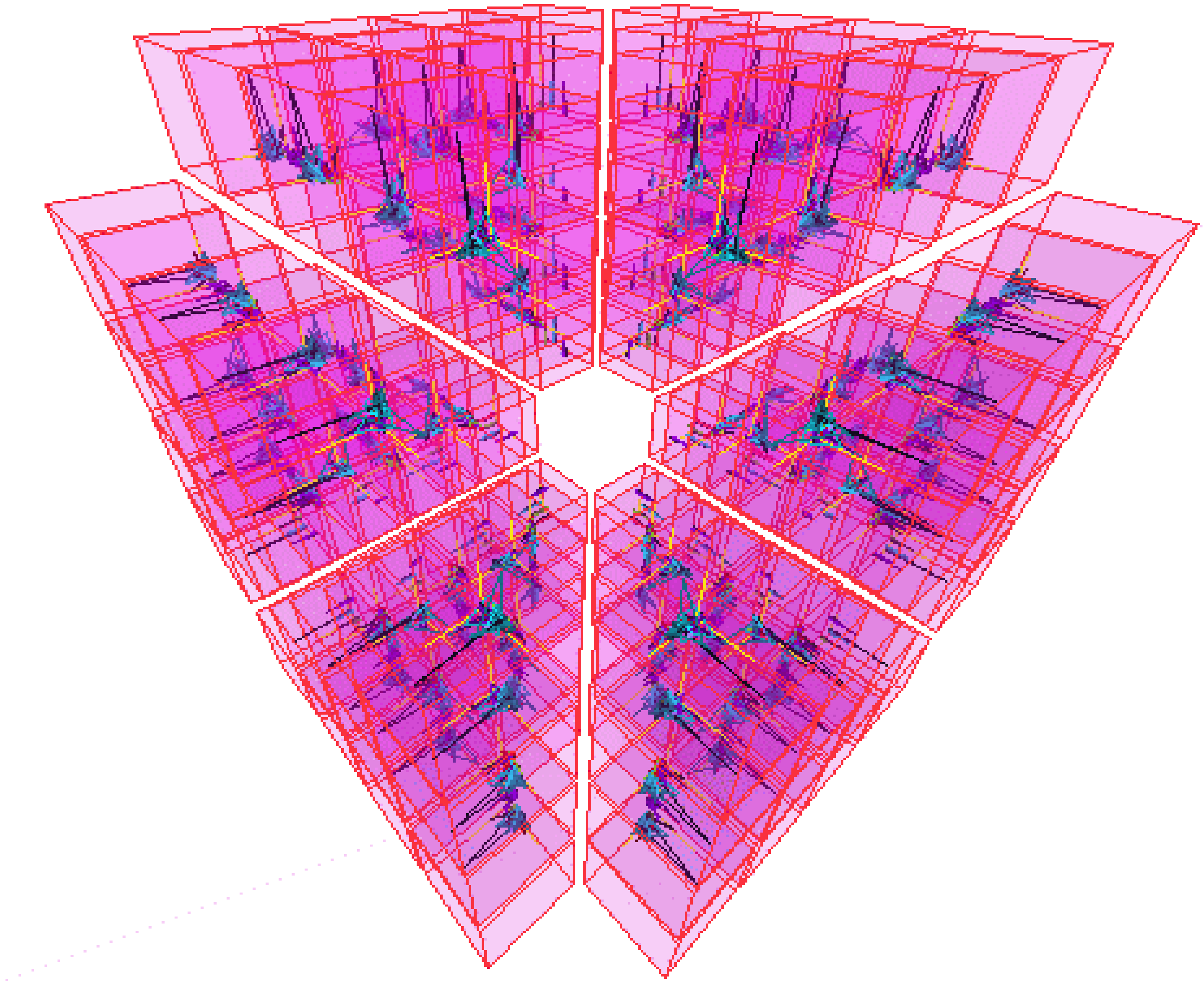
Use of a 'brushing' tool can highlight subsets of data

..now we can see what correlates with high MPG

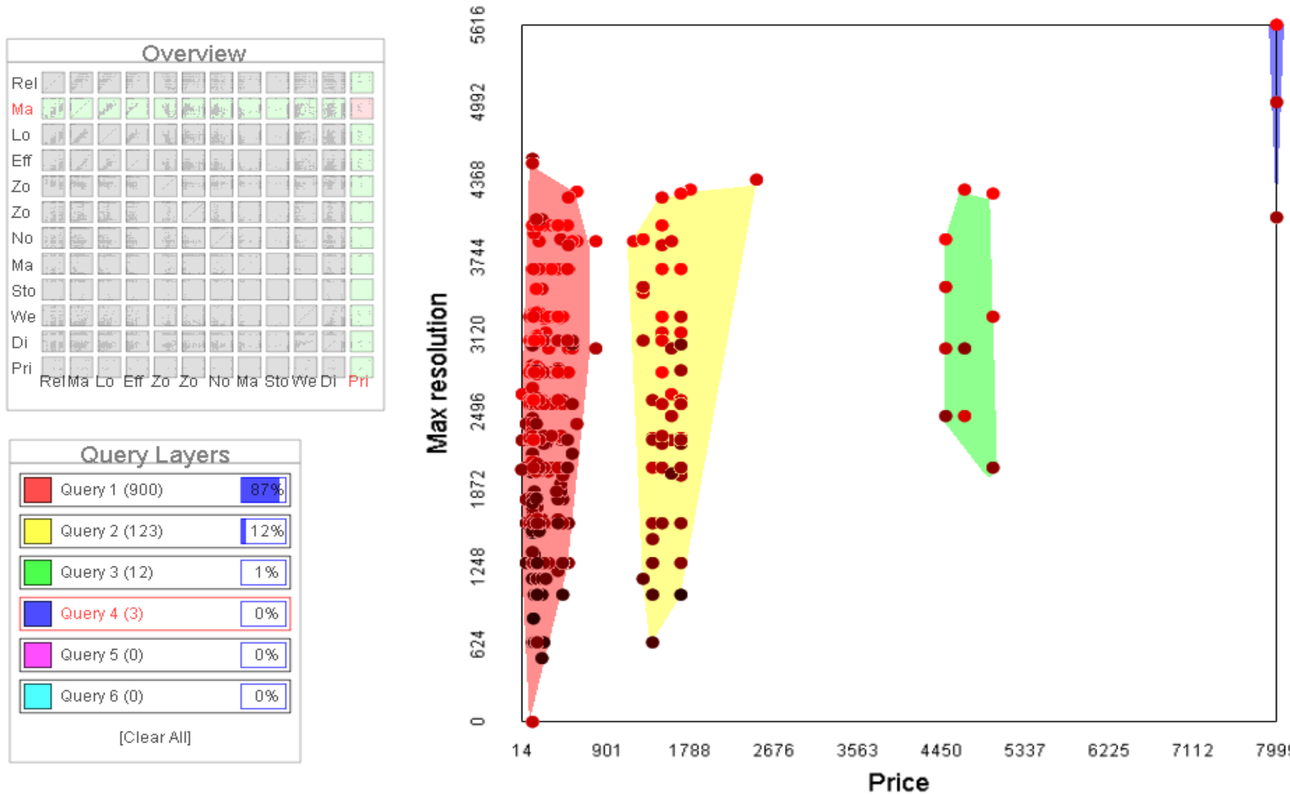
3D Scatterplot



3D-Scatterplot Matrix



Rolling the Dice



N. Elmqvist, P. Dragicevic, J.-D. Fekete. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. In *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2008)*, 14(6):1141-1148, 2008. (Best paper award)
https://engineering.purdue.edu/~elm/projects/scatterdice.html

VIDEO

Issues with Scatterplot Matrices

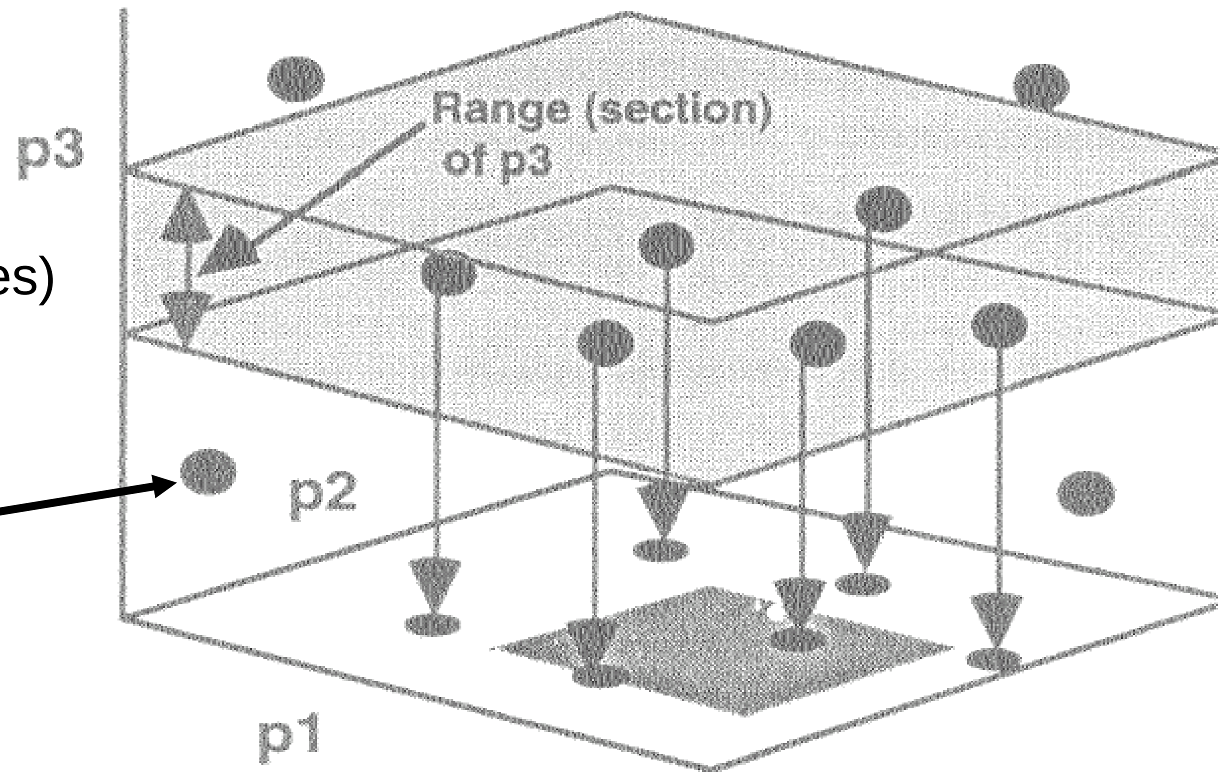
- Difficult to see higher dimensional data patterns that can be understood only when three or more data dimensions are simultaneously taken into account
- Requires interactions such as brushing and linking
- Scalability
  - One dozen attribs
  - Dozens to hundreds of items

Prosection Views [FB 94, STDS 95]

- A single projected section (prosection) of a scatterplot is called a prosection

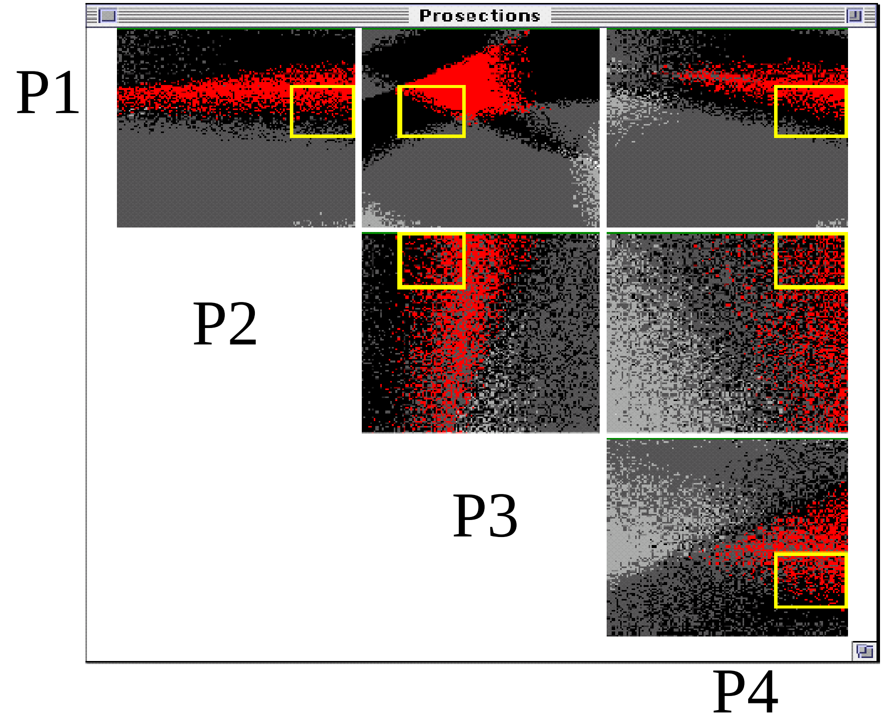
- Example: kettle
  - p1, p2, p3 are the parameters (attributes)
  - p1: radius
  - p2: thickness
  - p3: volume

not considered outside of the relevant range



Prosection Matrix [FB 94, STDS 95]

- Uses sensitivity color coding
  - Red: fullfil customer requirements
  - Black: fail with respect to one requirement
  - Dark gray: fail with respect to two requirements
  - Light gray: fail with respect to 3 requirements
  - White: fail with respect to more than 3 requirements



- Yellow squares mark tolerances for manufacturing
- Task: find max. yield is equivalent to placing yellow rectangle such that it contains mostly red points
- Example requirements for a light bulb design
  - Minimize stresses at various locations

VIDEO 1

VIDEO 2

Glyph-based Techniques

- Chernoff-Faces [Che 73, Tuf 83]
- FatFonts
- Stick Figures [Pic 70, PG88]
- Shape Coding [Bed 90]
- Color Icons [Lev 91, KK94]

Sign Classification [PB55]

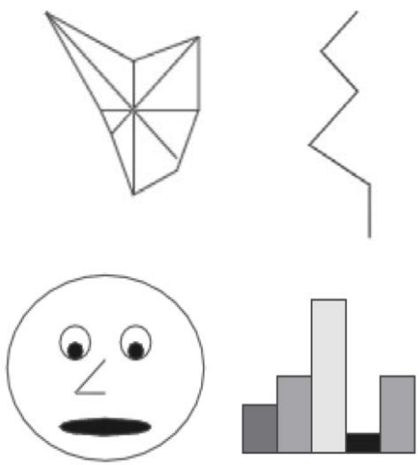
- **Icon** resembles the quality of the object it stands for
  - Items all share topological similarity with the object they are related to.
  - Examples: images, metaphors and diagrams.

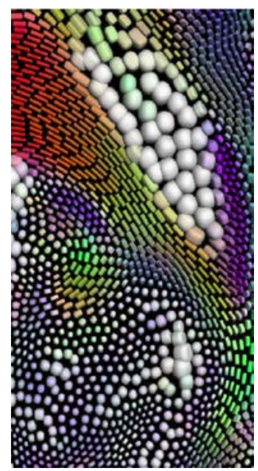
- **Index** demonstrates the influence of its object
  - Abstractions that rely on a physical cause/effect relation with the object to which they relate to.
  - Examples: clock, thermometer, fuel gauge.

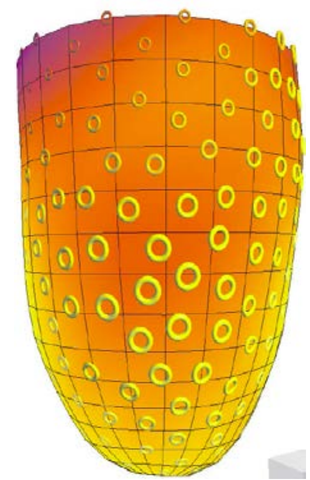
- **Symbol** is interpreted as a reference to its object
  - Abstractions which rely on a code conventionally used in order to determine meaning.
  - Examples: mathematical symbols, characters (?, !, …).

Glyphs

- A glyph is a small independent visual object that depicts attributes of a data record
- Glyphs are a visual channel for point marks
- Glyphs are a type of visual sign but differ from other types of signs such as icons, indices and symbols.





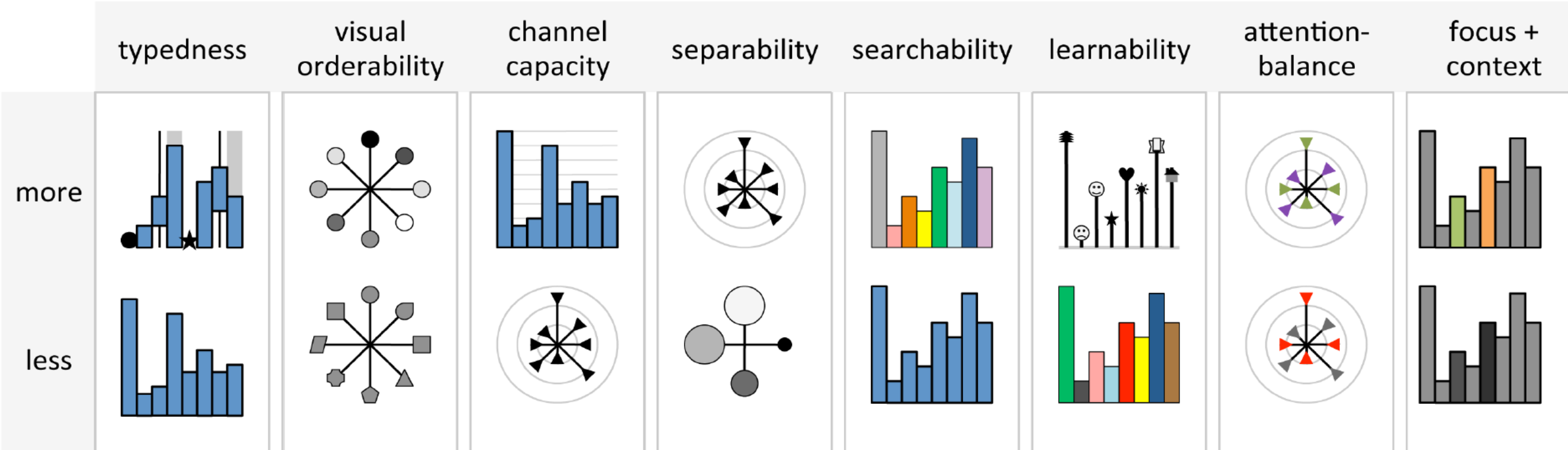


- **Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications**.
  R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward and M. Chen
  Eurographics State of the Art Report, pp. 39-63, 2013.
  http://diglib.eg.org/EG/DL/conf/EG2013/stars/039-063.pdf

Glyphs – Broader Interpretation

- A glyph is a small visual object that can be used independently and constructively to depict attributes of a data record or the composition of a set of data records;
- Each glyph can be placed independently from others, while in some cases, glyphs can be spatially connected to convey the topological relationships between data records or geometric continuity of the underlying data space; and
- Glyphs are a type of visual sign that can make use of visual features of other types of signs such as icons, indices and symbols.

Important Glyph Design Criteria [CLP13]



- Effectiveness principle
  - Encode most important attributes with highest ranked channels
  - Channel capacity, separability, searchability, learnability, attention balance, ability to highlight (here: focus and context)
- Expressiveness principle
  - Match channel and data characteristics
    - Typedness, visual orderability

Important Glyph Design Criteria [CLP13]

Typedness – This criterion refers to whether or not each visual channel in a glyph is appropriately selected to match with the data type of the variable to be encoded.

Such data types may include, but not limited to: nominal, ordinal, interval, ratio, and directional.

Visual Orderability – When a variable to be encoded is orderable, the corresponding visual channel should ideally be orderable visually
(e.g., size, greyscale intensity, but not an arbitrary set of shapes).

Channel Capacity – This refers to the number of values that may be encoded by a visual channel. Such a number is often affected by the size of a glyph and many perceptual factors
(e.g., just-noticeable-difference, interference from nearby visual objects).

Separability – When two or more visual channels are integrated into a compound channel, such as combining intensity, hue and saturation into a colour channel, the interference between different primitive channels should be minimised.

Searchability – This refers to the levels of ease when one needs to identify a specific visual channel within a glyph for a specific variable.

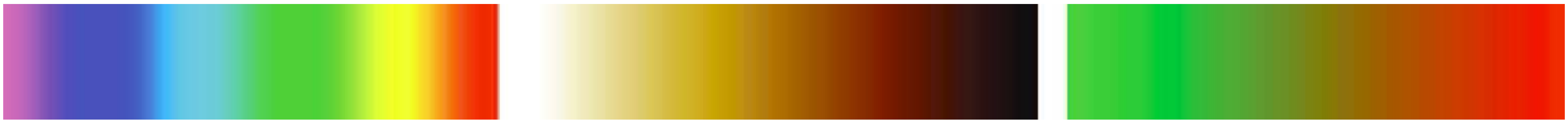
Learnability – This is often an important criterion in many applications. Ideally, a glyph design should be easy to learn, and easy to remember. There are many factors that may affect a visual design in this context, for instance, whether there are well-defined constructive rules, whether there are memorable metaphors, whether it is easy to guess, and so on.

Attention Balance – Different visual channels in a glyph will receive different levels of attention. Ideally, the levels of attention should correspond to the levels of importance of the variables. However, this is easier said than done as the relative importance of a variable is often undefined or may vary from tasks to tasks.

Focus and Context – This refers to the need to identify an individual visual channel under a certain interactive operation. For example, when a user select a certain variable as a sort key, it is desirable to highlight the corresponding visual channel so it stands out from other channels.

Perceptually Uniform Visual Encoding

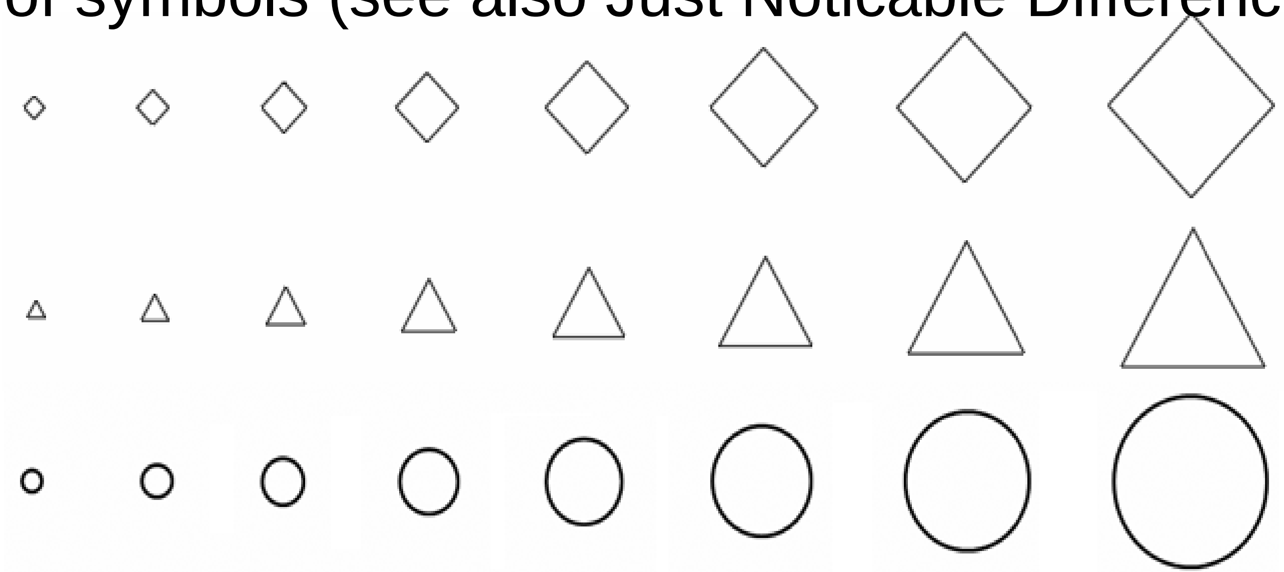
- Color

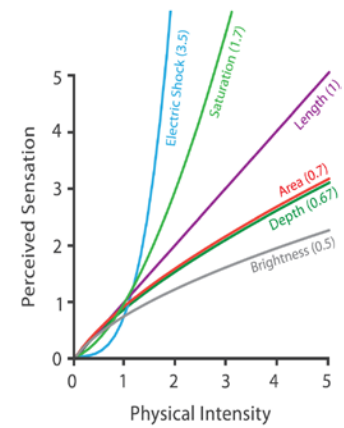


Rainbow black body radiation green-red isoluminant

[BT07]

- Size of symbols (see also Just Noticable Difference)

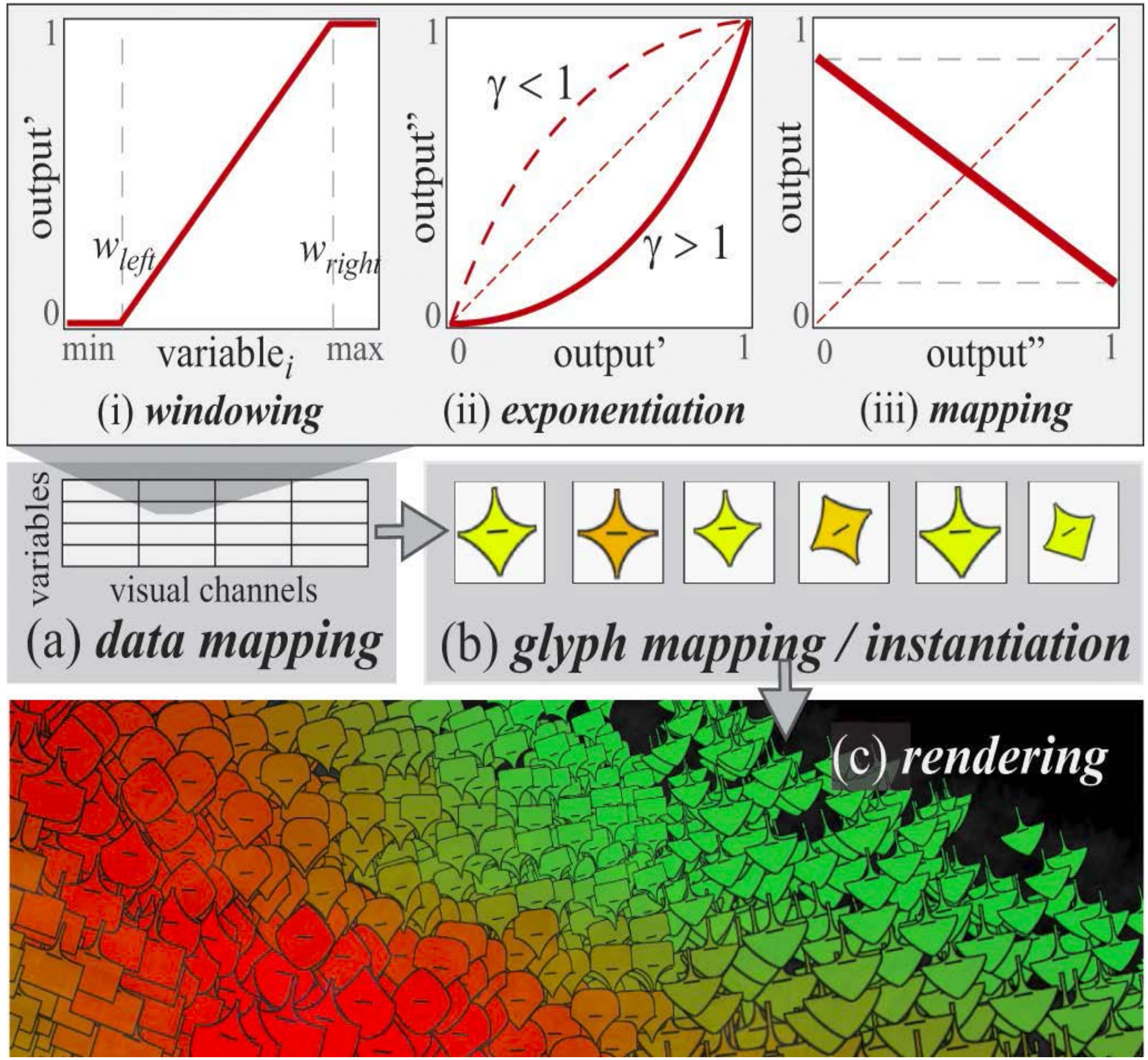




Steven’s power law transformation / Exponentiation [LMvW10]

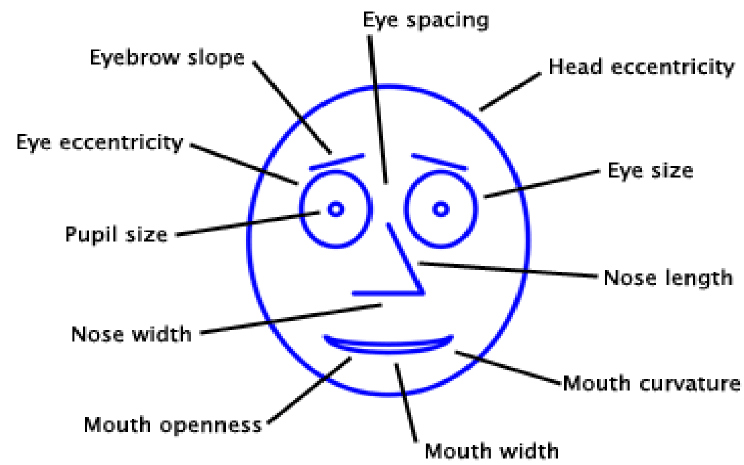
Glyph Generation Pipeline [LKH09]

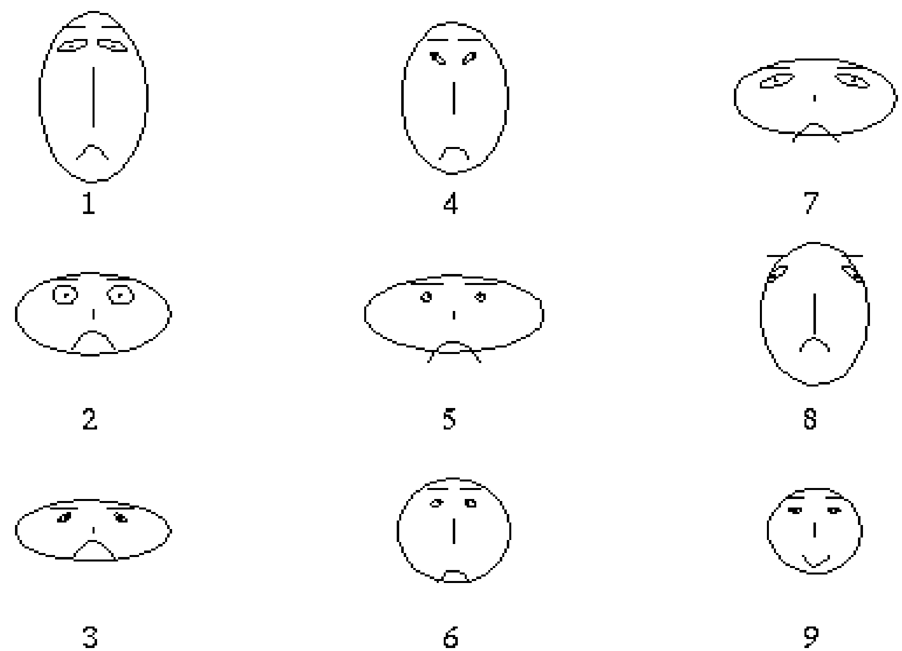
- Data Enhancement
  - Map data to [0, 1]
  - Exponentiation
  - Adapt mapping
    - e.g inverse
- Map data to glyph
- Render glyphs



Chernoff Faces [CHE73]

- 10 Parameters:
  - Head Eccentricity
  - Eye Eccentricity
  - Pupil Size
  - Eyebrow Slope
  - Nose Size
  - Mouth Vertical Offset
  - Eye Spacing
  - Eye Size
  - Mouth Width
  - Mouth Openness
- Problem
  - Assignment of variables
  - Also affects grouping (which faces are similar?)
- http://hesketh.com/schampeo/projects/Faces/chernoff.html





Chernoff Face Cartogram!

Slide from ‘Cartograms for Human Geography’, by D. Dorling, in Visualization in Geographical Information Systems, Wiley.

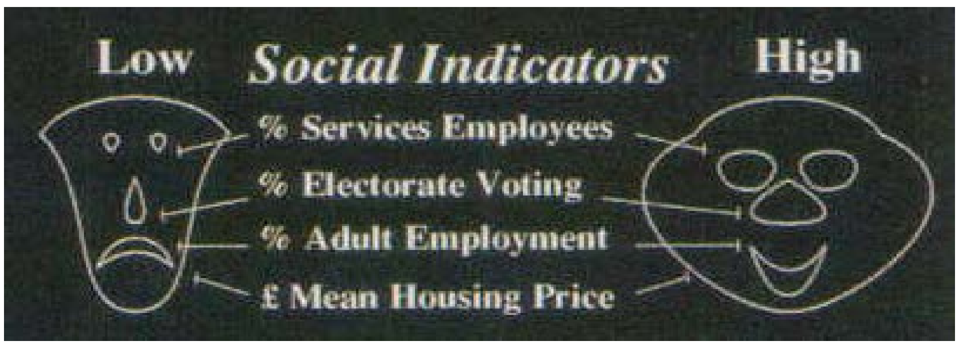
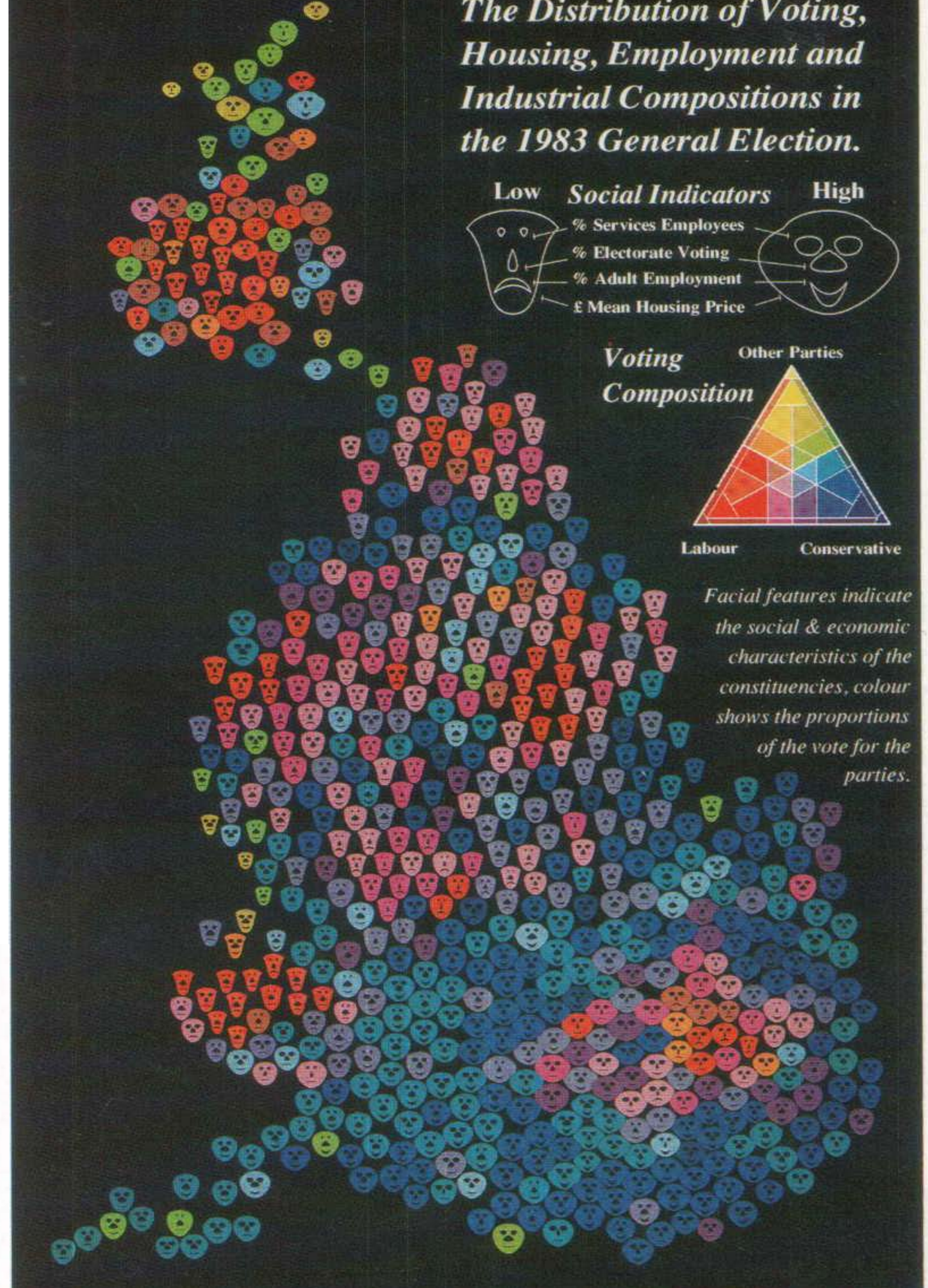
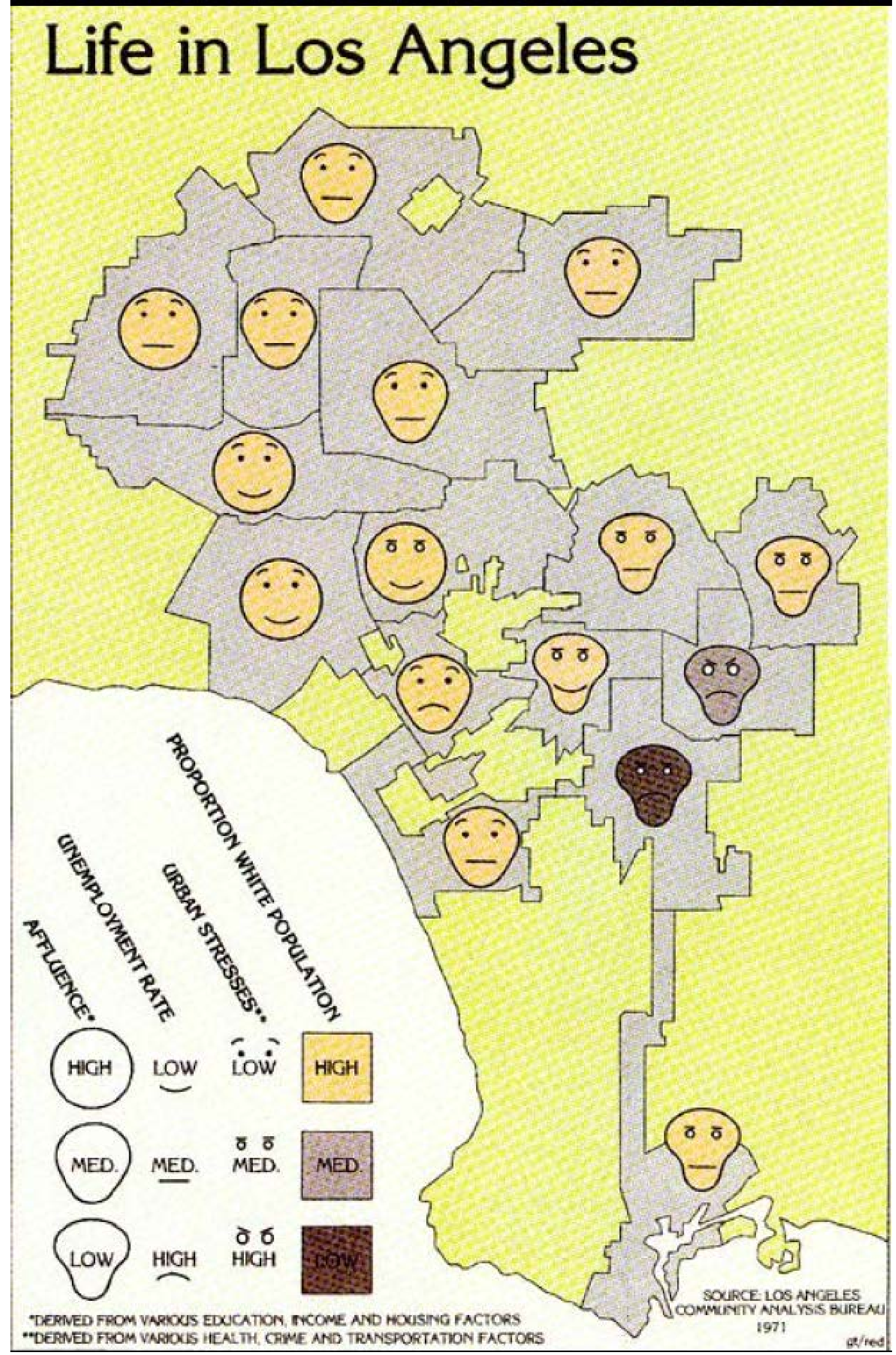


Plate 10 The distributions of voting, housing, employment and industry on a population cartogram. The 633 mainland parliamentary constituencies are each represented by a face whose features express the various variables, and which is coloured by the mix of voting, drawn on an equal electorate cartogram. The patterns in this picture are very interesting and could lead to endless discussion. The ‘deaths-heads’ inside Glasgow city are solidly red, while the happy-faces around the capital voted strongly for the government of the day. The Welsh may not have had much employment, or expensive housing, but they still turned out to vote in large numbers. This technique is particularly good for identifying exceptions, faces which do not fit in with the crowd



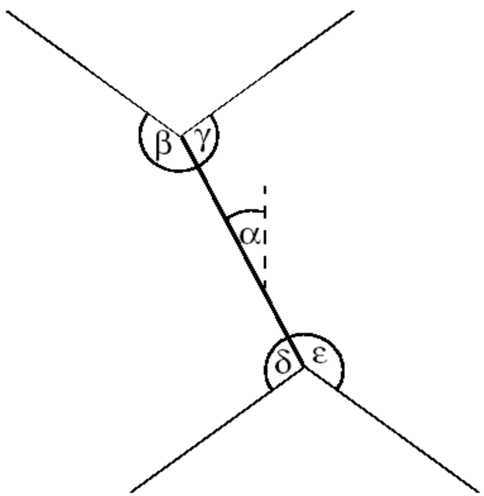
Chernoff Face Cartogram

Dr. Eugene Turner, 1979

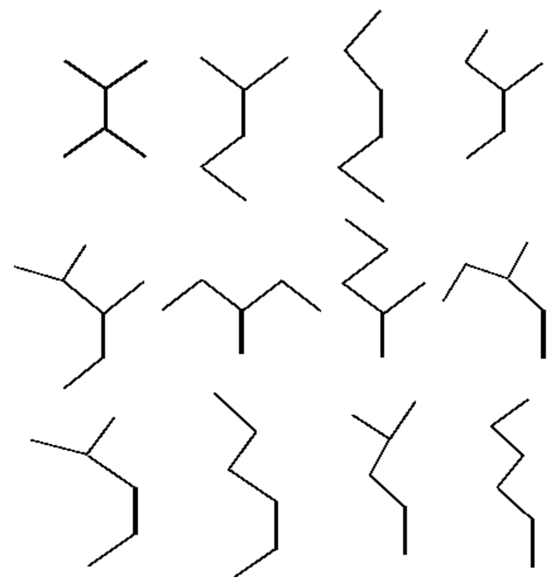


Stick Figures [PG 88]

- Visualization of the multidimensional data using stick figure icons
- Two attributes of the data are mapped to the display axes and the remaining attributes are mapped to the angle, color, and/or length of the limbs
- Texture patterns in the visualization show certain data characteristics

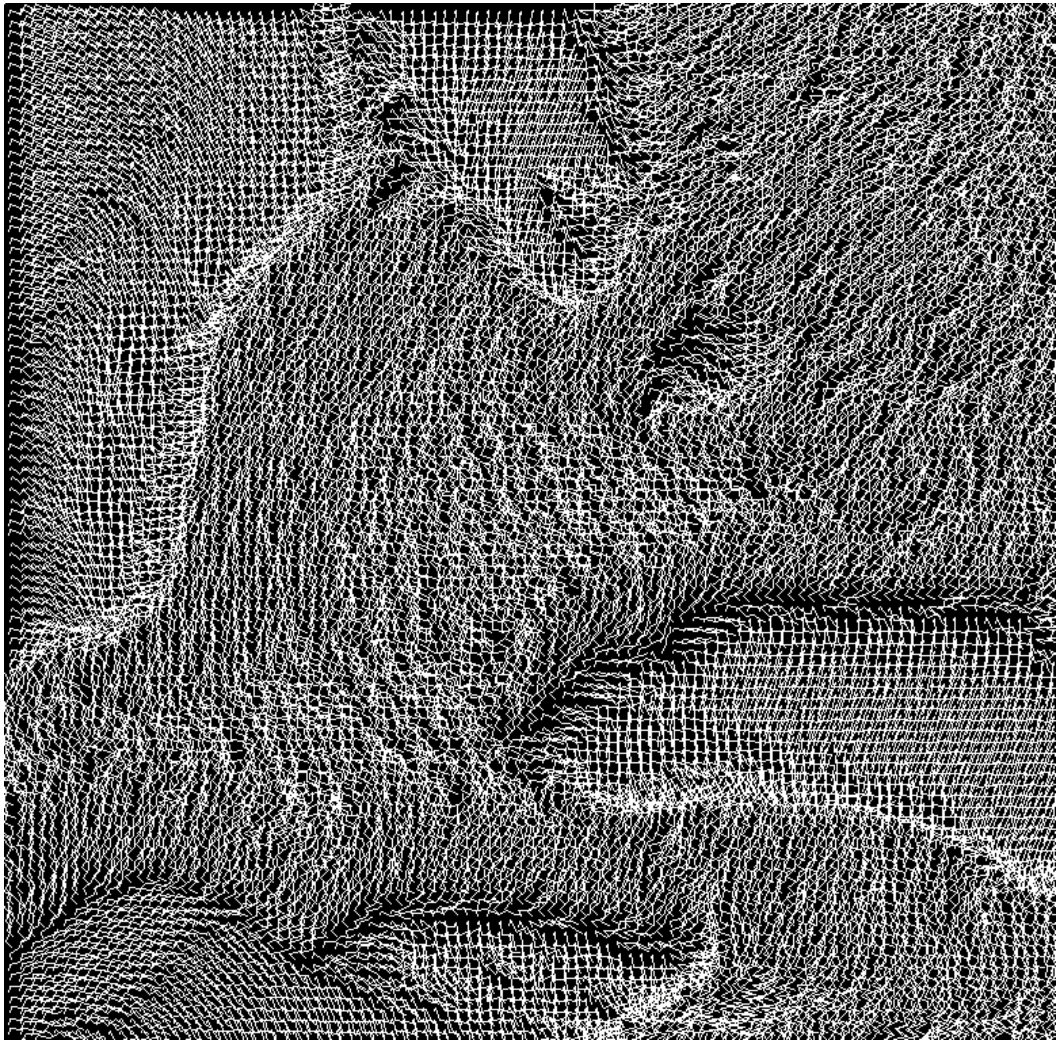


A stick figure with 5 parameters



Stick figure family with 5 segments in different configurations

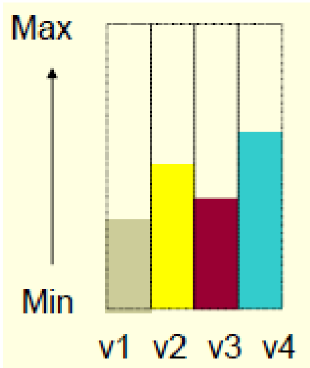
Stick Figures

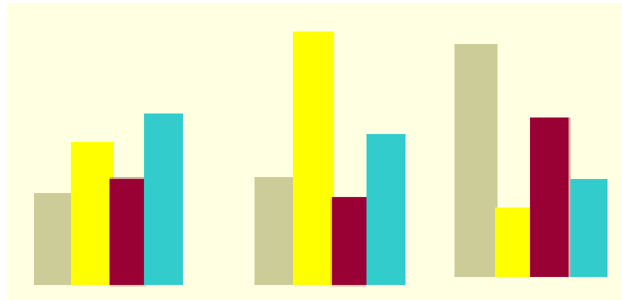


Parameters are taken from photographs with 5 different wavelength. Data from the great lakes region, USA.

http://ivpr.cs.uml.edu/gallery/
G. Grinstein, University of Massachusetts at Lowell

Profile Glyphs





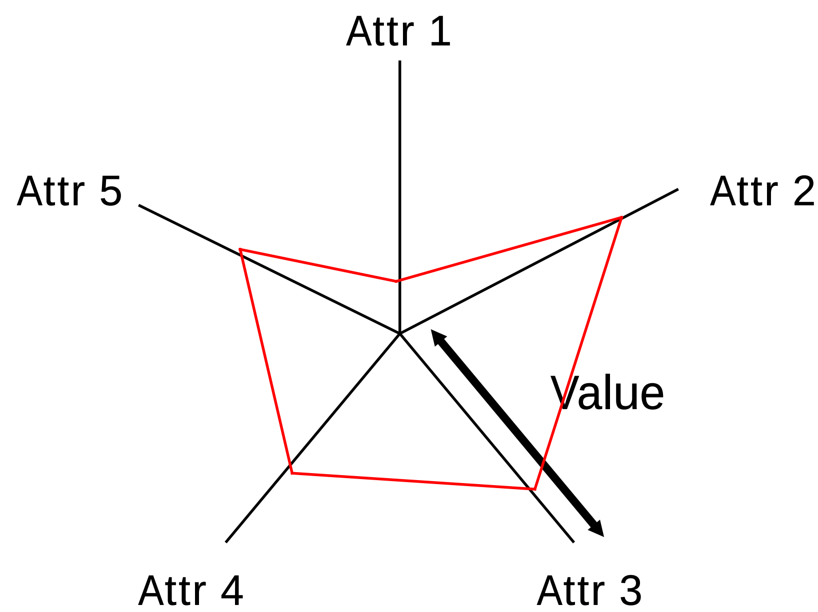
One data record with four attributes

Three data records

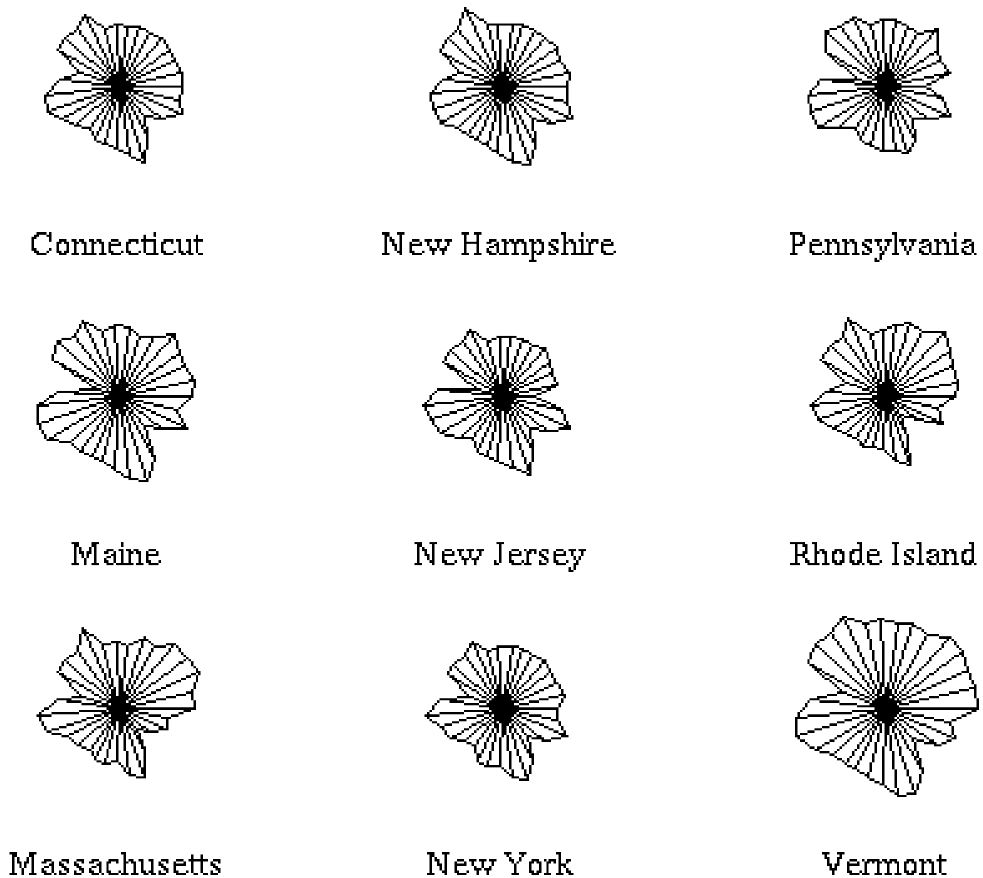
- Simple bar charts often used in a small-multiples layout

Star Glyphs (Star Plots)

- The attributes axes are drawn at equal angles around a circle
- Each “spoke” encodes an attribute’s value



Star Glyphs in a Small-Multiples Layout



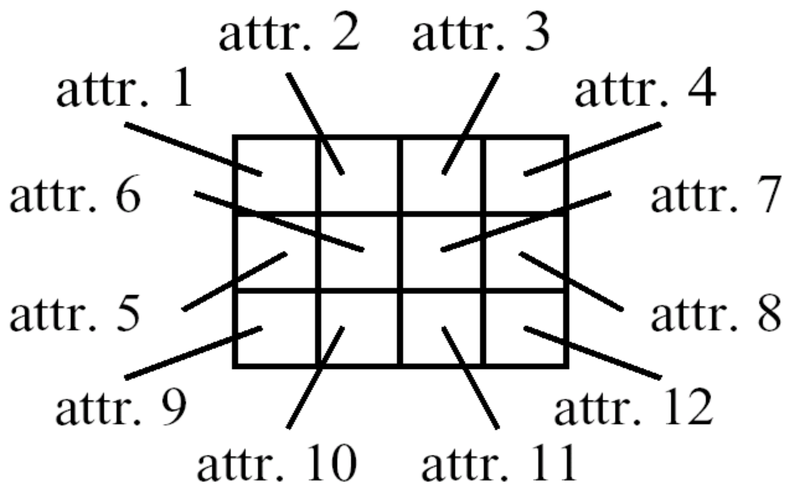
Star glyphs for 31 attributes in 9 states

Length of each "spoke" encodes attribute value

`http://seamonkey.ed.asu.edu/~behrens/asu/reports/compre/comp1.html`

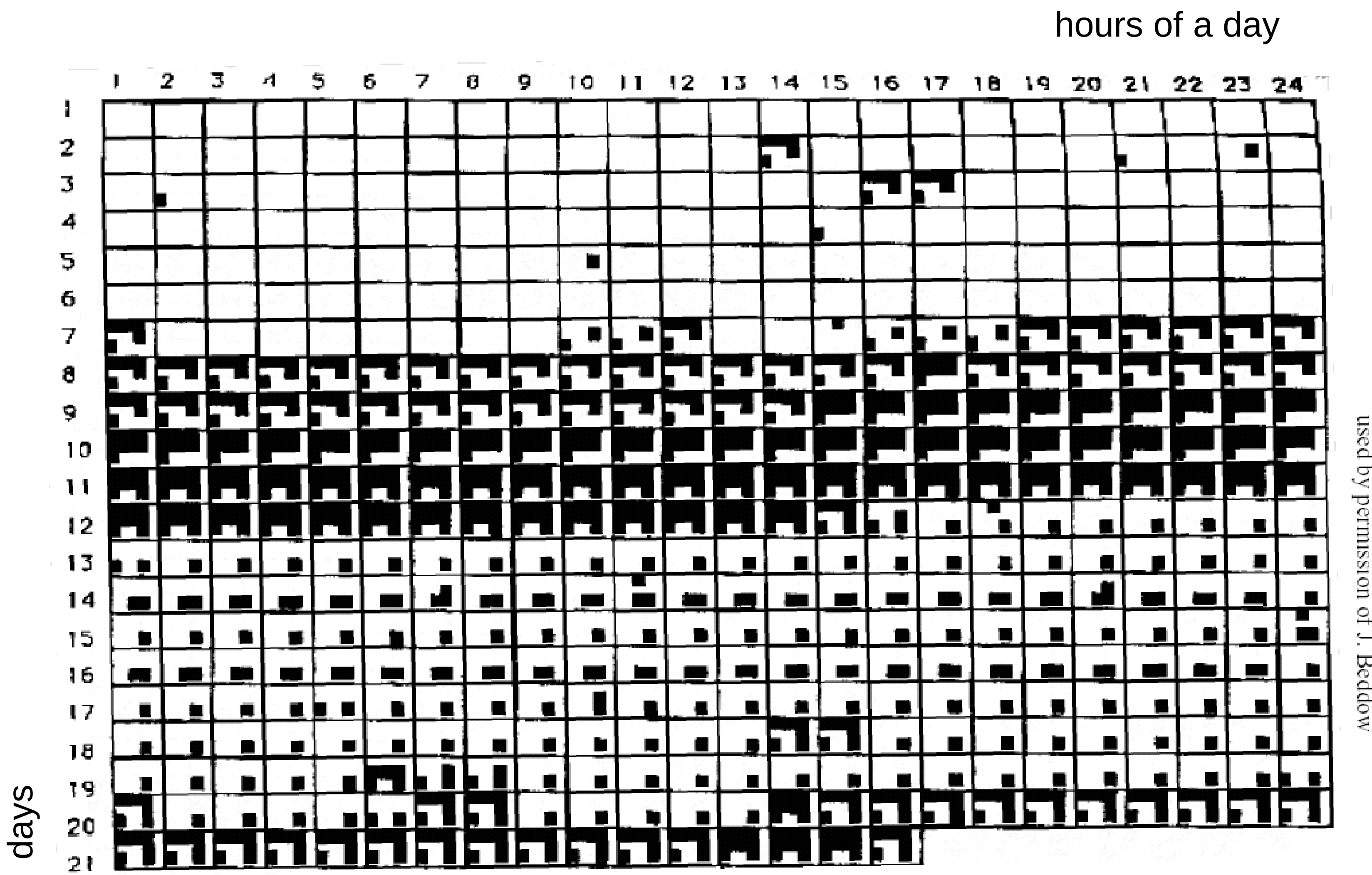
Shape Coding [Bed 90]

- The data is visualized using small arrays of fields
- Each field represents one attribute value
- Arrangement of attribute fields (e.g., 12-dimensional data):



- Arrays are arranged line-by-line according to a given sorting (e.g., the time attribute for time-series data)

Shape Coding

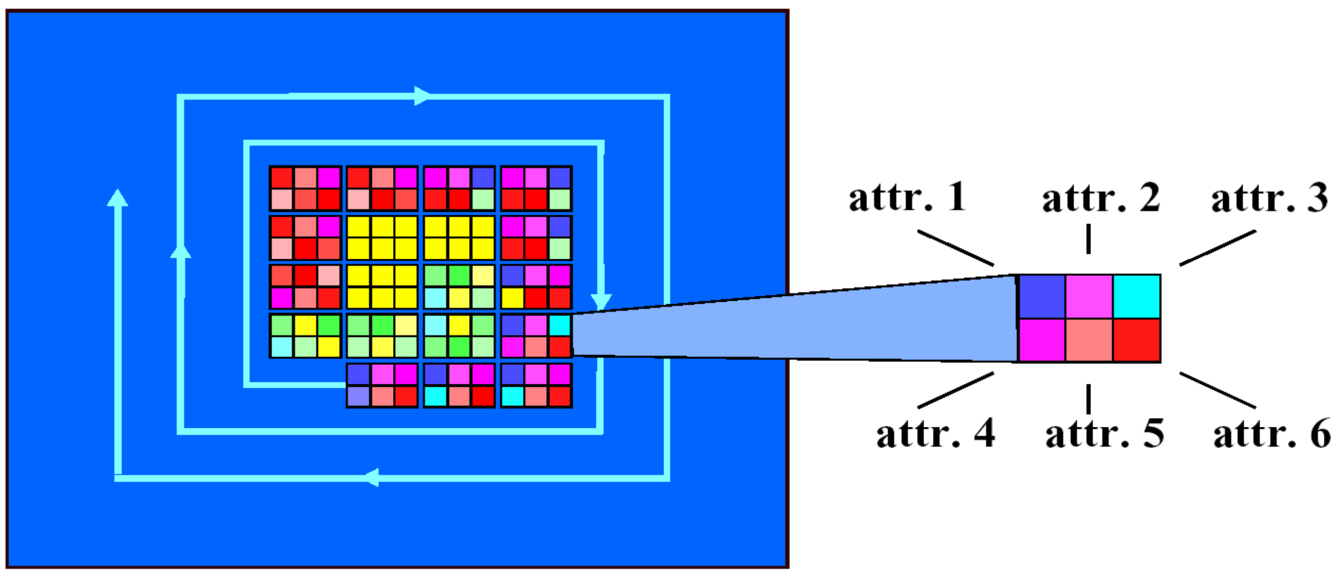


time series of NASA earth observation data

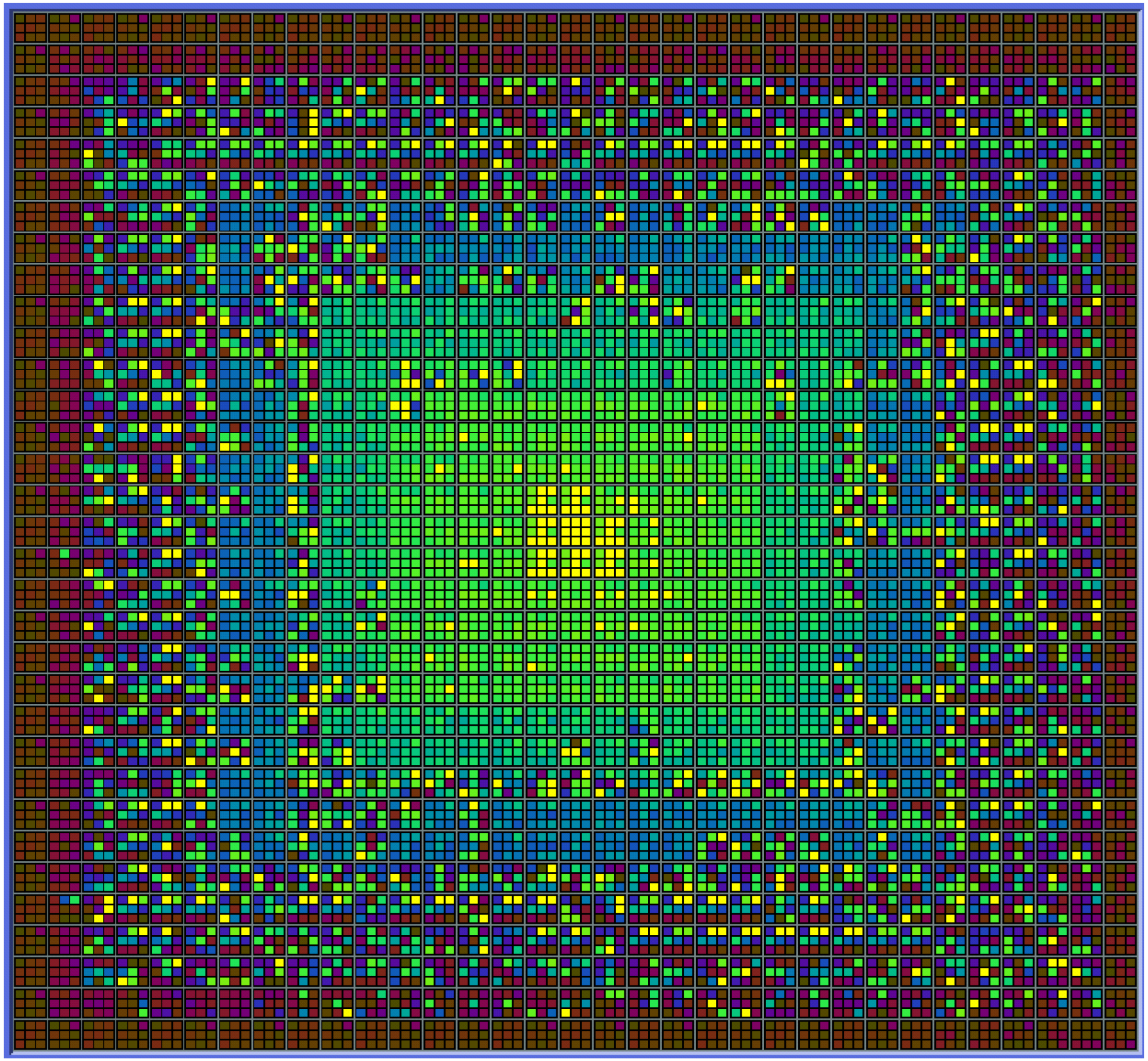
Color Icons [Lev91, KK94]

- Visualization of the data using color glyphs
- Color icons are arrays of color fields representing the attribute values based on some color map – e.g. high values map to yellow, low values to red
- Arrangement is query-dependent (e.g., spiral)

schematic representation of 6-dim. data



Color Icons

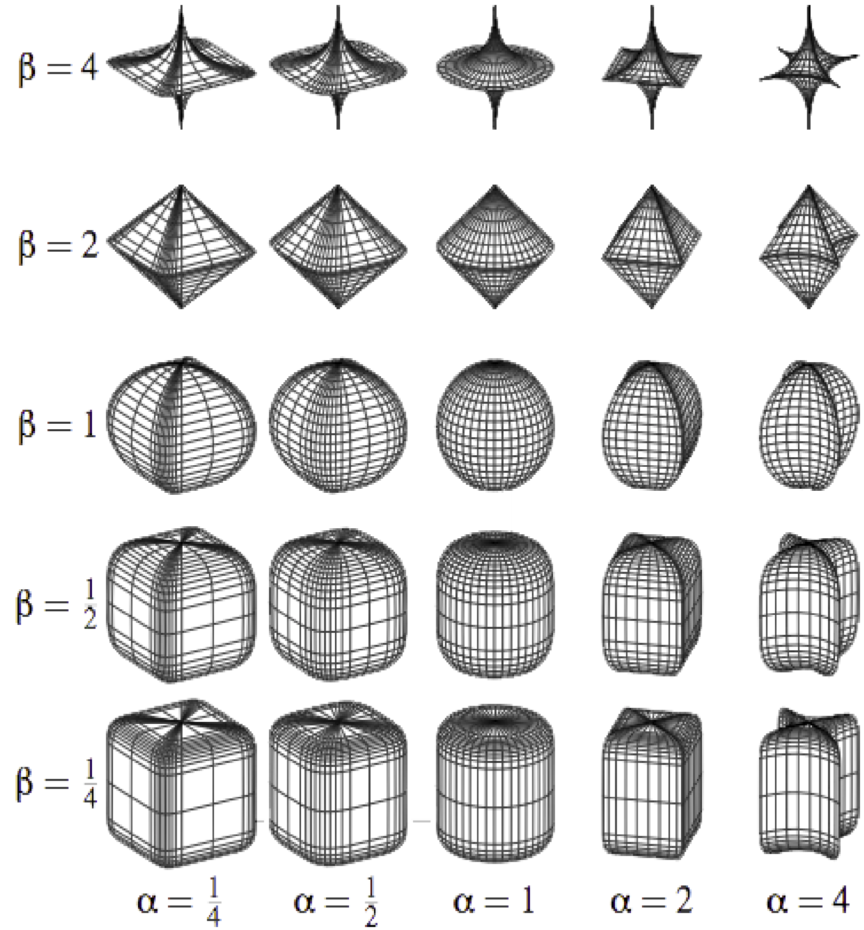


- Random data containing several clusters
- Spiral arrangement
- More than one pixel per attribute necessary
- Sorted by average

color map

Parametric Glyph Shape Design

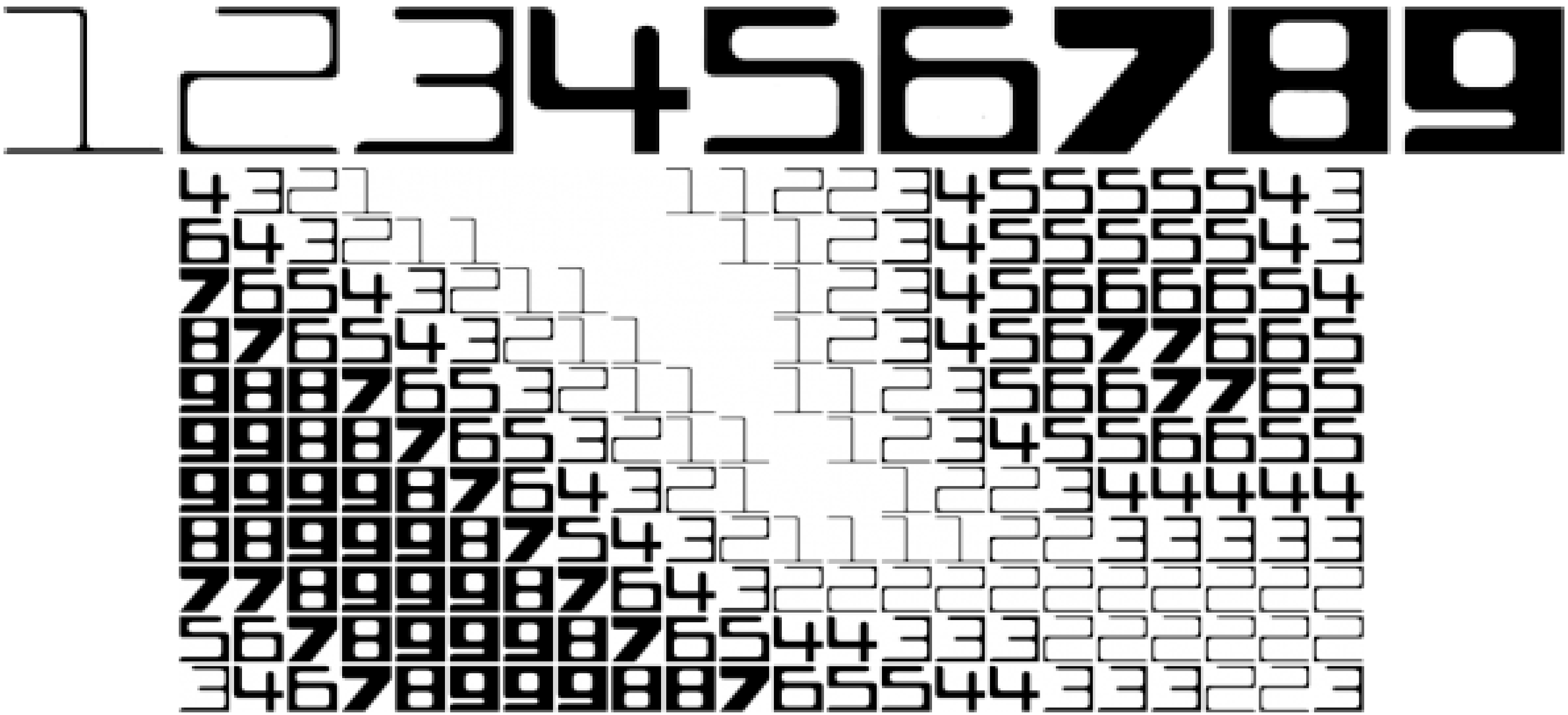
- Shape design is one of the most prominent visual channels.
- Adjusting the exponents **β** and **α** controls the superquadric shape.
- These are referred to as squareness parameters



[Kindlmann2004]

FatFonts (http://fatfonts.org)

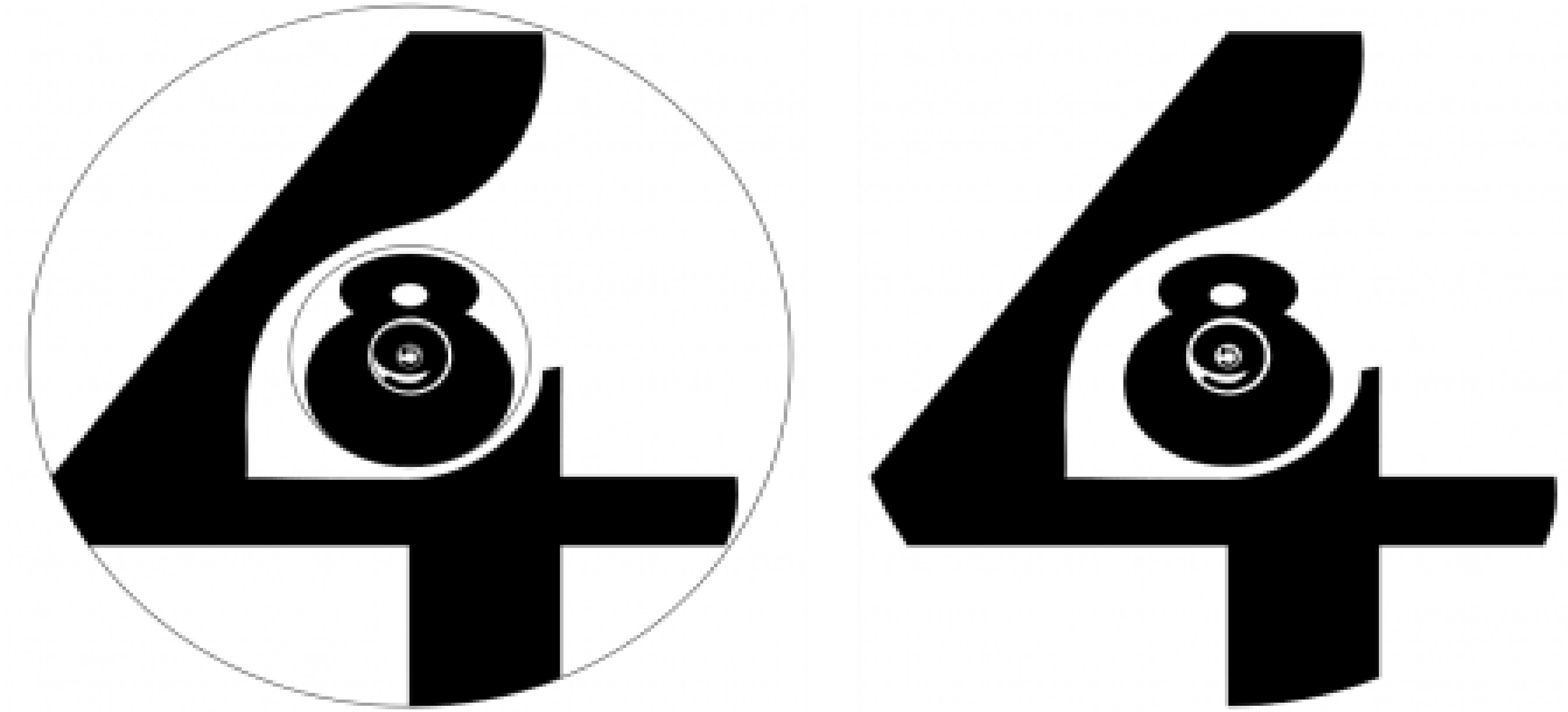
- Amount of dark pixels in a numeral character is proportional to the number it represents – zero represented as empty space



Miguel Nacenta, Uta Hinrichs and Sheelagh Carpendale. FatFonts: Combining the Symbolic and Visual Aspects of Numbers. In AVI '12: Proceedings of the International Working Conference on Advanced Visual Interfaces, 2012.

FatFonts

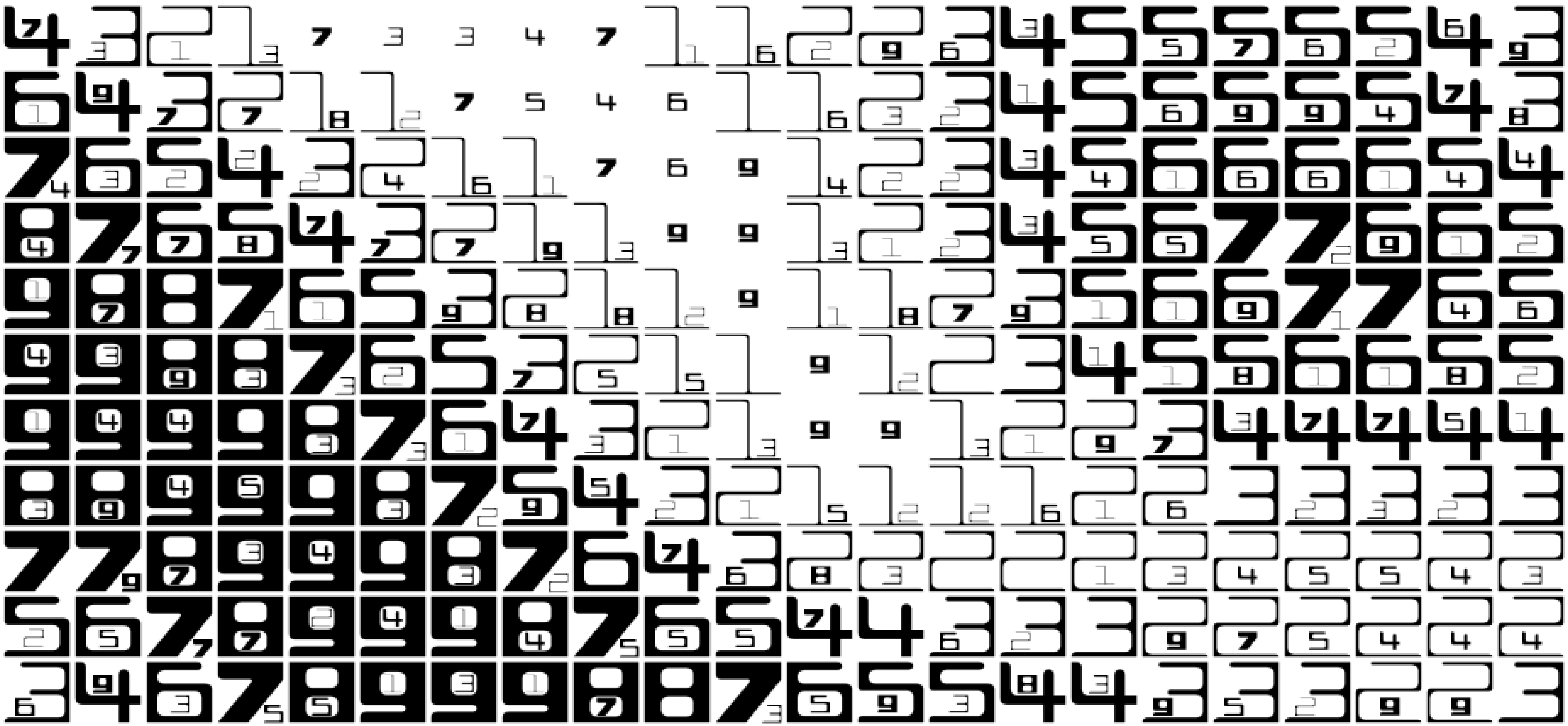
- Multi-level digits – nested representation



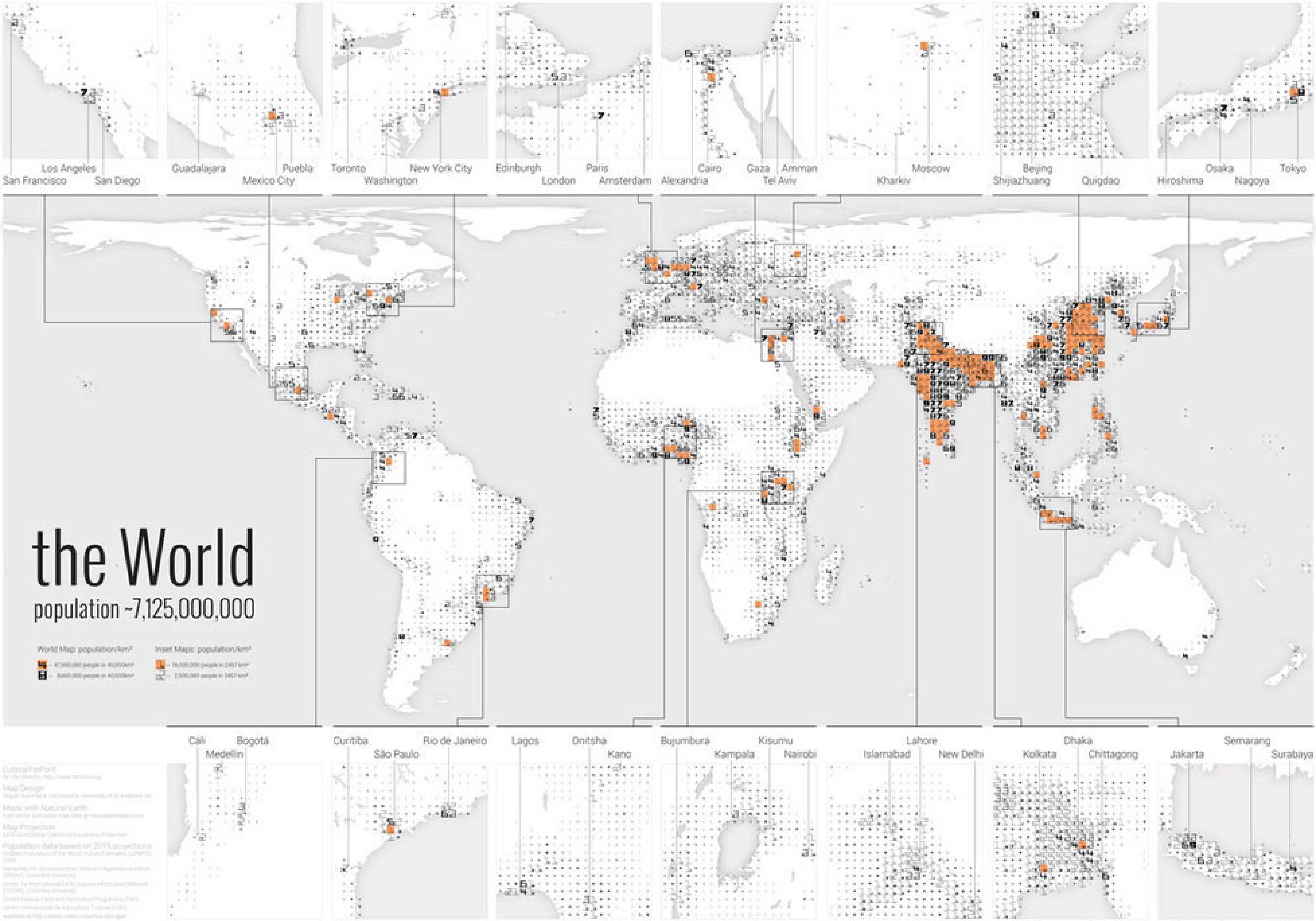
Represents the digit sequence 4895
Interpreted as 4,895 or 48,95 or …

FatFonts

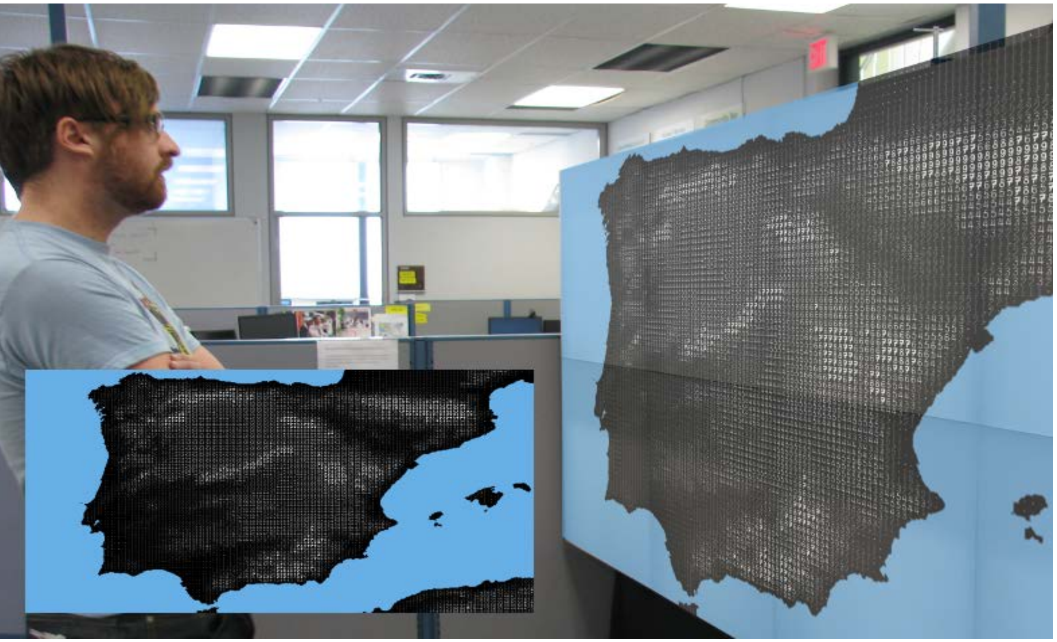
- Multi-level digits

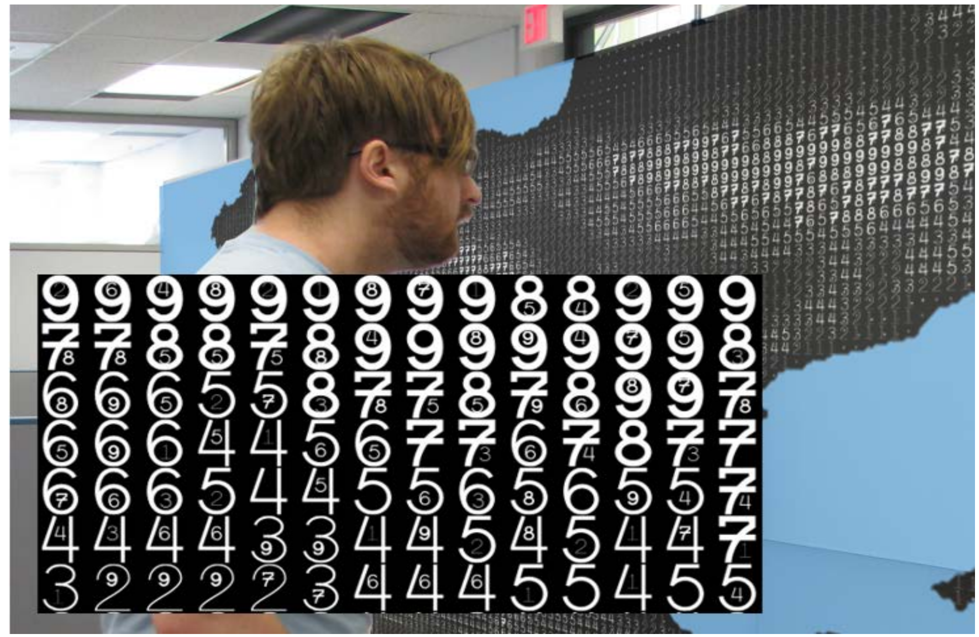


Two-level FatFonts used to represent a scalar field.


San Francisco
Los Angeles
San Diego
Guadalajara
Mexico City
Puebla
Toronto
Washington
New York City
Edinburgh
London
Paris
Amsterdam
Alexandria
Cairo
Gaza
Tel Aviv
Amman
Kharkiv
Moscow
Shijiazhuang
Beijing
Quigdao
Hiroshima
Osaka
Nagoya
Tokyo
the World
population ~7,125,000,000
World Map: population/km²
Inset Maps: population/km²
Cali
Medellín
Bogotá
Curitiba
São Paulo
Rio de Janeiro
Lagos
Onitsha
Kano
Bujumbura
Kampala
Kisumu
Nairobi
Islamabad
Lahore
New Delhi
Kolkata
Dhaka
Chittagong
Jakarta
Semarang
Surabaya
Map Design
Made with Natural Earth
Map Projection
Population data based on 2015 projections

FatFonts





FatFonts on large high resolution displays

Fat Fonts Video

- https://www.youtube.com/watch?v=B4bm2oe7uwM

Online Resources

- Many Eyes Visualization
  - http://hci.stanford.edu/seminar/abstracts/07-08/080201-viegas-wattenberg.html
- History of Data Visualization
  - http://hci.stanford.edu/seminar/abstracts/08-09/090306-heer.html
- www.tableausoftware.com – trial version
- InfoVis Toolkits
  - http://vis.stanford.edu/protovis/
  - http://d3js.org/
- Many more talks:
  - http://hci.stanford.edu/seminar/past/years.html

References

- [BED90] Beddow J.: Shape Coding of Multidimensional Data on a Microcomputer Display, Visualization ‘90, San Francisco, CA, 1990, pp. 238-246.
- [BT07] BORLAND D., TAYLOR R.: Rainbow color map (still) considered harmful. IEEE Computer Graphics and Applications
- 27, 2 (2007), 14–17.
- [CF12] CHEN M., FLORIDI L.: An analysis of information in visualisation. Springer Synthese, November 2013, Volume 190, Issue 16, pp 3421-3438 http://www.philosophyofinformation.net/publications/pdf/aaoiiv.pdf
- [CHE73] Chernoff H. Using faces to represent points in k-dimensional space graphically. J Am Stat Assoc 1973; 68: 361–368.
- [CLP2013] CHUNG D. H. S., LEGG P. A., PARRY M. L., BOWN R., GRIFFITHS I. W., LARAMEE R. S., CHEN M.: Glyph sorting: Interactive visualization for multi-dimensional data. Information Visualization November 29, 2013, doi: 10.1177/1473871613511959
- [FB94] Furnas G. W., Buja A.: ‘Prosections Views: Dimensional Inference through Sections and Projections’, Journal of Computational and Graphical Statistics, Vol. 3, No. 4, 1994, pp. 323-353.
- [Kindlmann2004] G. L. Kindlmann, Superquadric tensor glyphs., in: Joint Eurographics - IEEE TCVG Symposium on Visualization, 2004, pp. 147–154.
- [KK94] Keim D. A., Kriegel H.-P.: ‘VisDB: Database Exploration using Multidimensional Visualization’, Computer Graphics & Applications, Sept. 1994, pp. 40-49.
- [Lev91] Levkowitz H.: ‘Color icons: Merging color and texture perception for integrated visualization of multiple parameters’, In Visualization ‘91, San Diego, CA, October 22-25 1991.
- [LKH09] Lie AE, Kehrer J and Hauser H. Critical design and realization aspects of glyph-based 3D data visualization. In: Proceedings of the 2009 spring conference on computer graphics (SCCG ’09), Budmerice, Slovakia, April 2009 pp. 19–26. New York: ACM.
- [LMvW10] LI J., MARTENS J.-B., VAN WIJK J. J.: A model of symbol size discrimination in scatterplots. In Proc. Human Factors in Computing Systems (CHI’10) (2010), pp. 2553–2562.
- [Mac04] MACEACHREN A. M.: How Maps Work - Representation, Visualization, and Design. Guilford Press, 2004
- [PB55] PEIRCE C., BUCHLER J. E.: The Philosophical Writings of Peirce. New York: Dover, 1955.
- [PG88] Pickett R. M., Grinstein G. G.: ‘Iconographic Displays for Visualizing Multidimensional Data’, Proc. IEEE Conf. On Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ, 1988, pp. 514-519.
- [STDS95] Spence R., Tweedie L., Dawkes H., Su H.: ‘Visualization for Functional Design’, Proc. Int. Symp. on Information Visualization (InfoVis ‘95), Atlanta, GA, 1995, pp.4-10.

Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - *Chap 7: Arrange Tables*
- Visualizing Data. Cleveland. Hobart Press, 1993.
- *A Brief History of Data Visualization.* Friendly. 2008. http://www.datavis.ca/milestones

End

Visualization

Parallel Coordinates

SS2015

Bernd Fröhlich

Virtual Reality Systems
Faculty of Media
Bauhaus-Universität Weimar

Acknowledgements

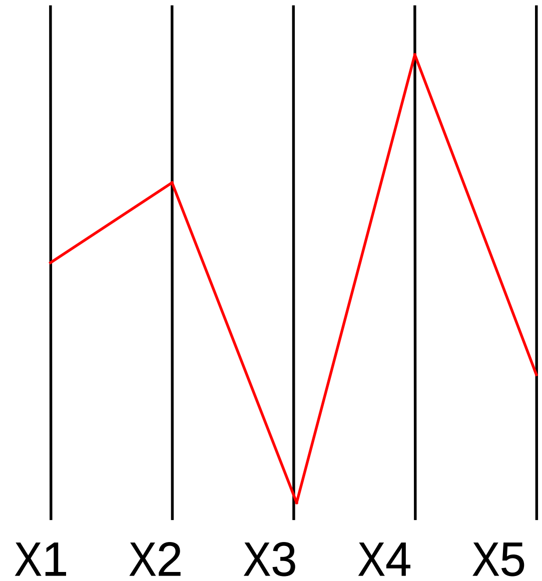
- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Harald Reiterer, Universität Konstanz
  - Daniel A. Keim, Universität Konstanz

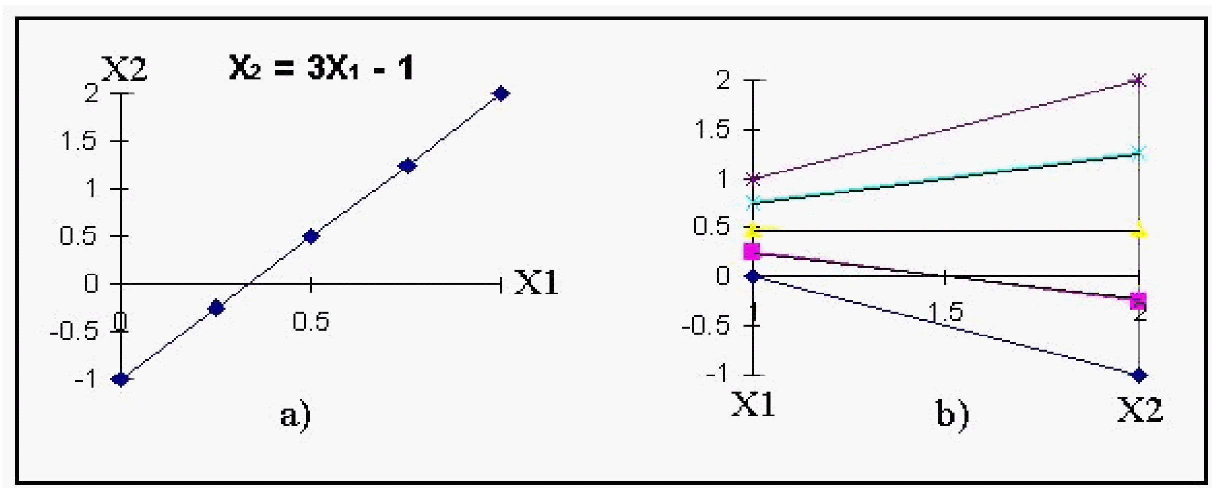
Multi-Attribute Data

- Scatterplot
- Prosection
- Glyph-based techniques
- Parallel coordinates

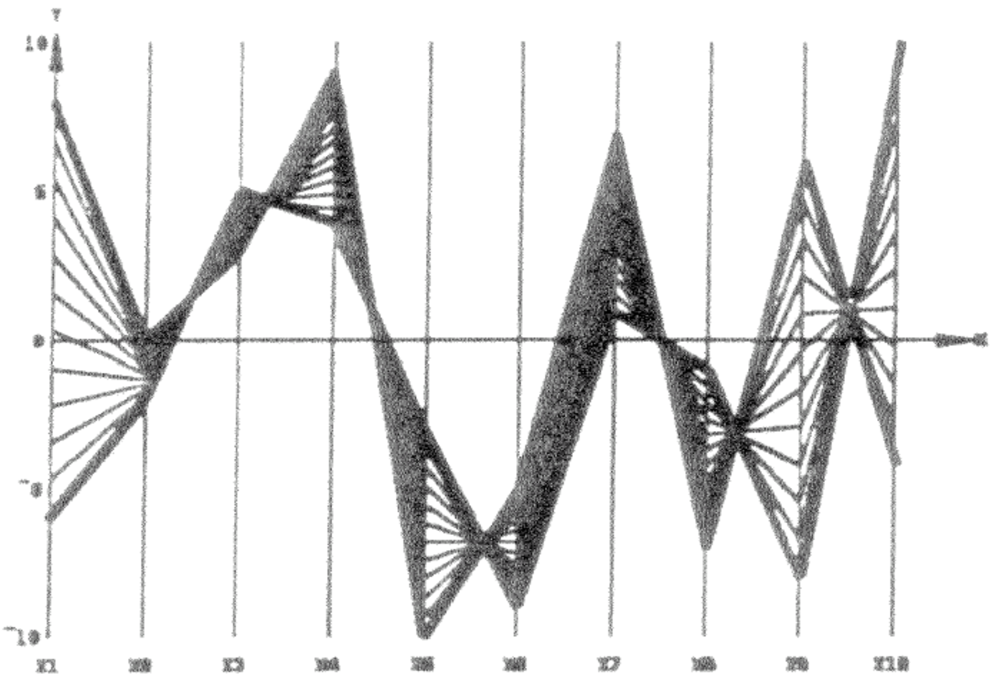
Parallel Coordinates

- Encode attributes as a set of vertical axes
- The axes are scaled to the [minimum, maximum] - range of the corresponding attribute
- A polyline represents one data record



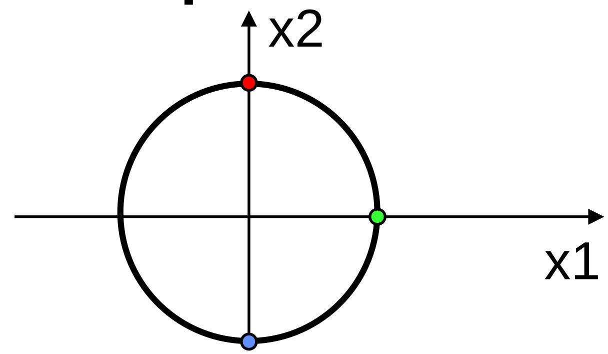


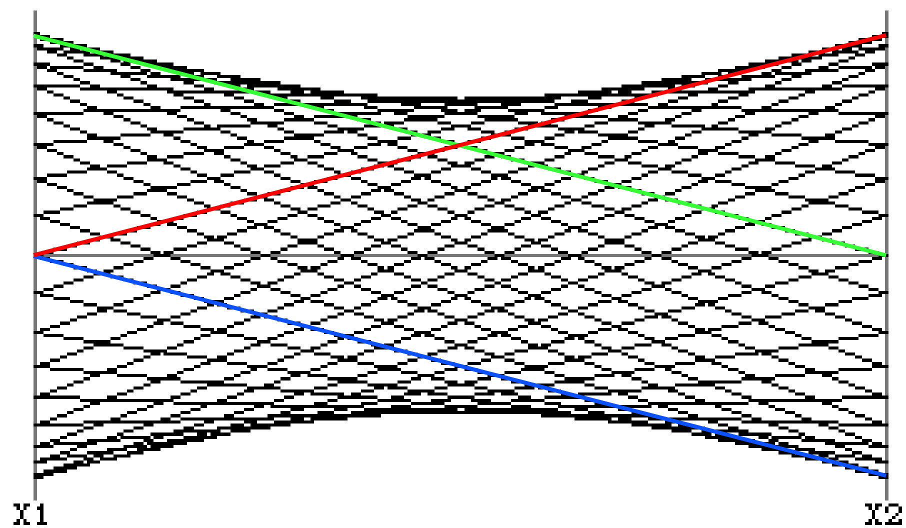
Parallel Coords Example



used by permission of A. Inselberg, Tel Aviv University, Israel

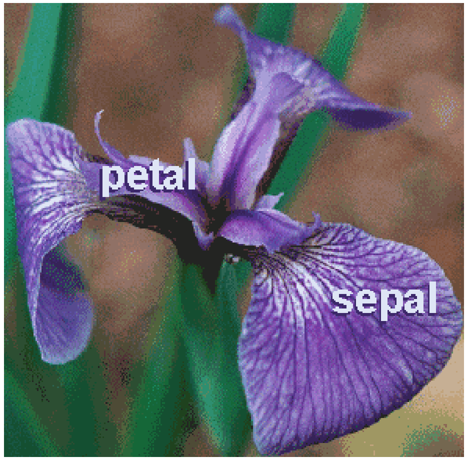
points on a line in 10-dim. space





points on a 2D circle

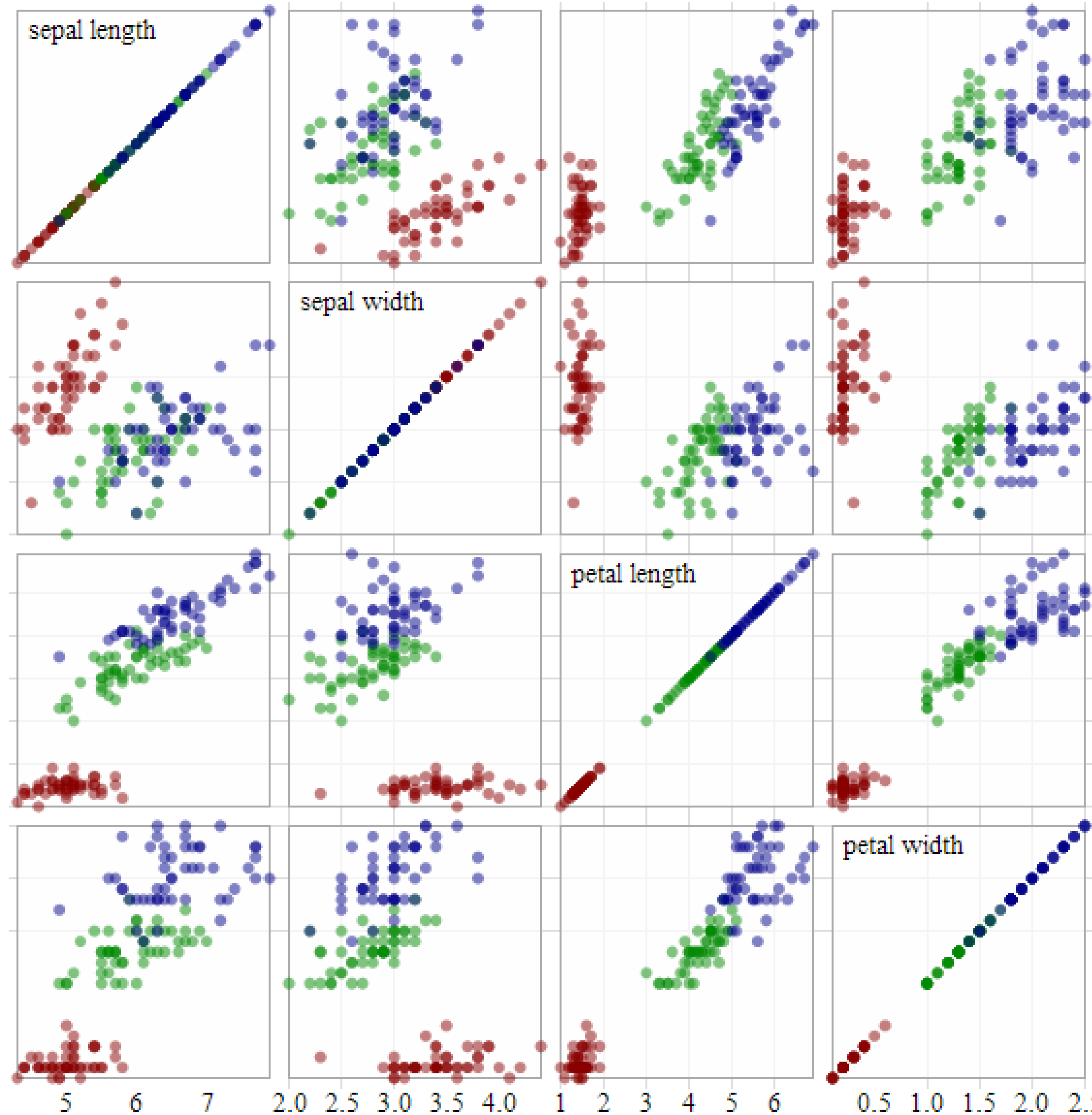
Multivariate Data: Iris Dataset



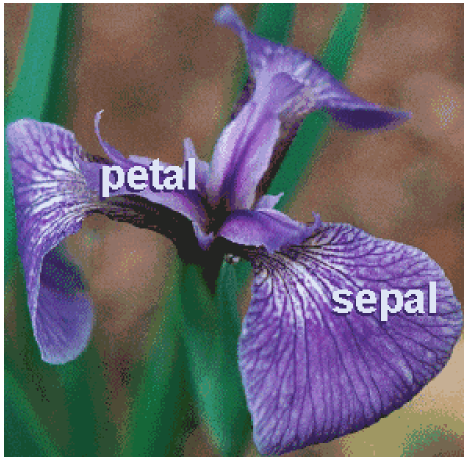
Setosa Versicolor Virginica

Anderson's data of iris flowers on the Gaspé Peninsula, Quebec, Canada

http://en.wikipedia.org/wiki/Iris_flower_data_set

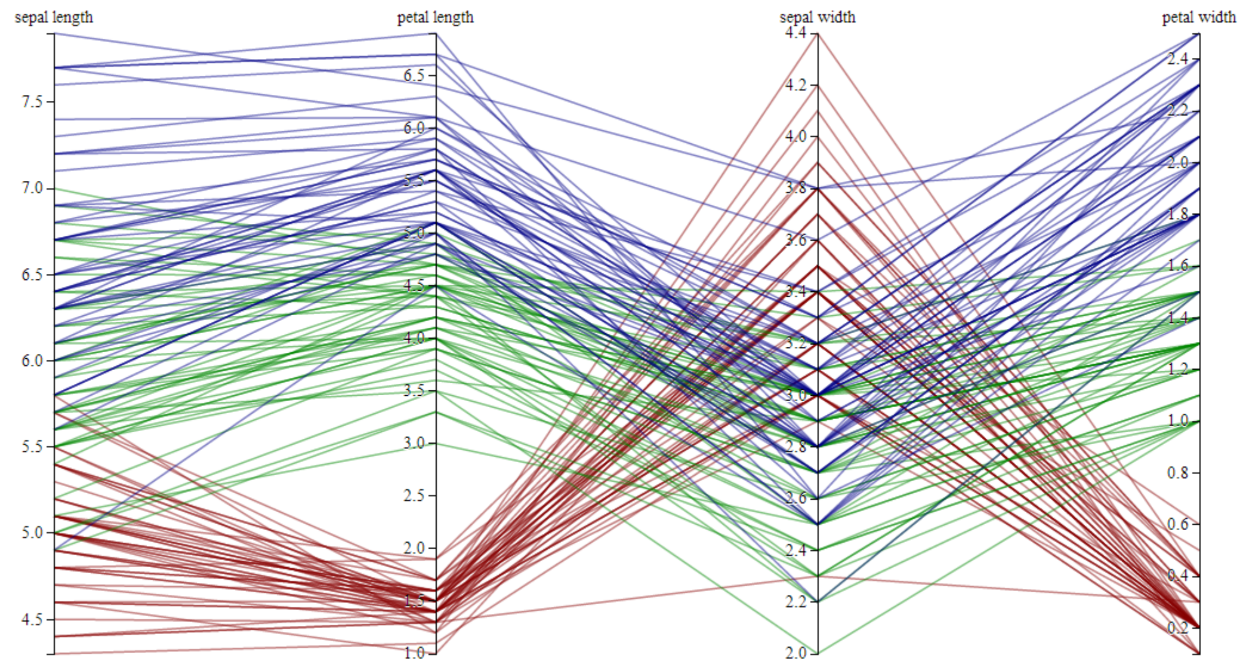


Iris Dataset in Parallel Coordinates

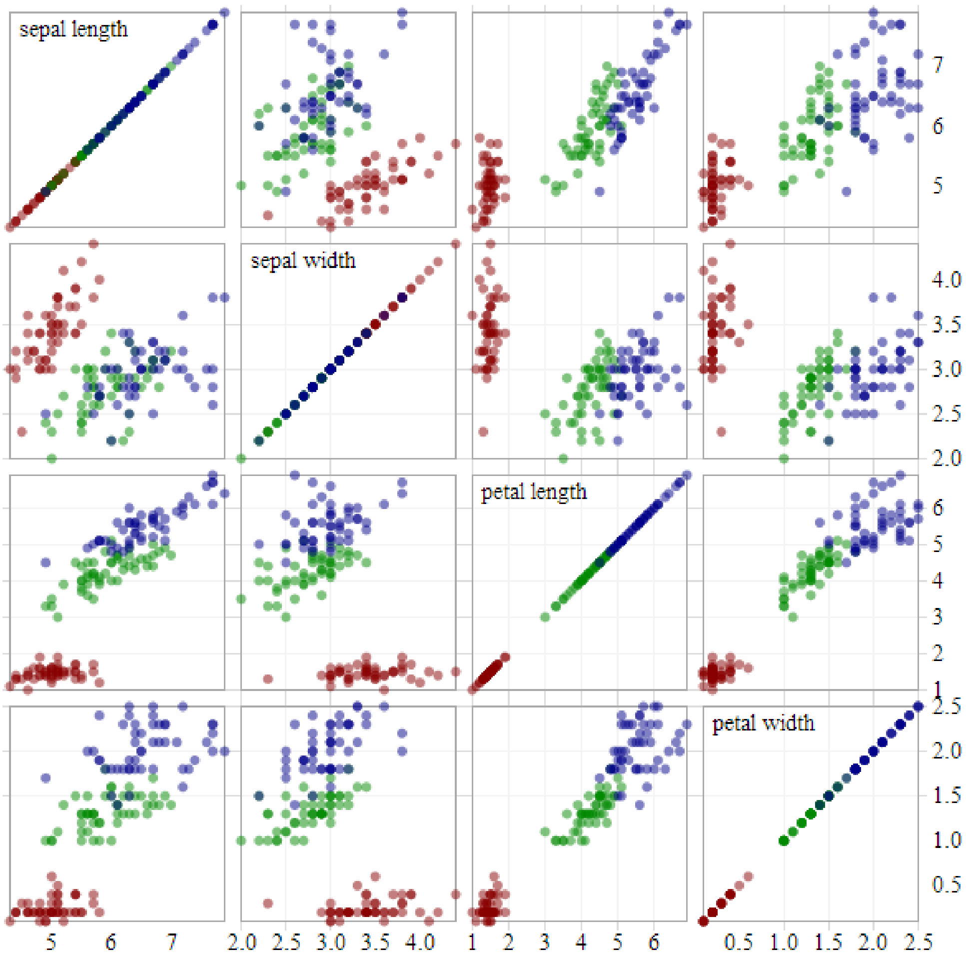


Setosa Versicolor Virginica

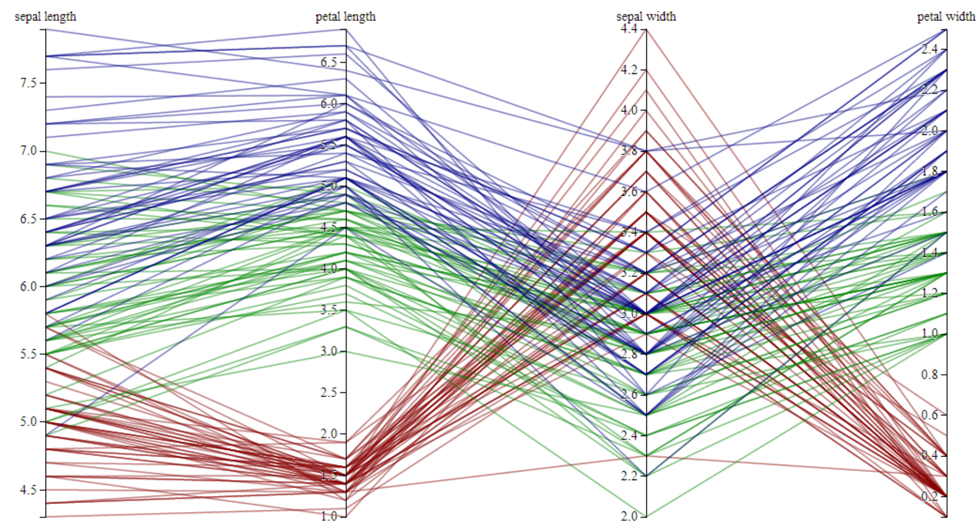
Anderson's data of iris flowers on the Gaspé Peninsula, Quebec, Canada



Scatterplot Matrix vs. Parallel Coordinates



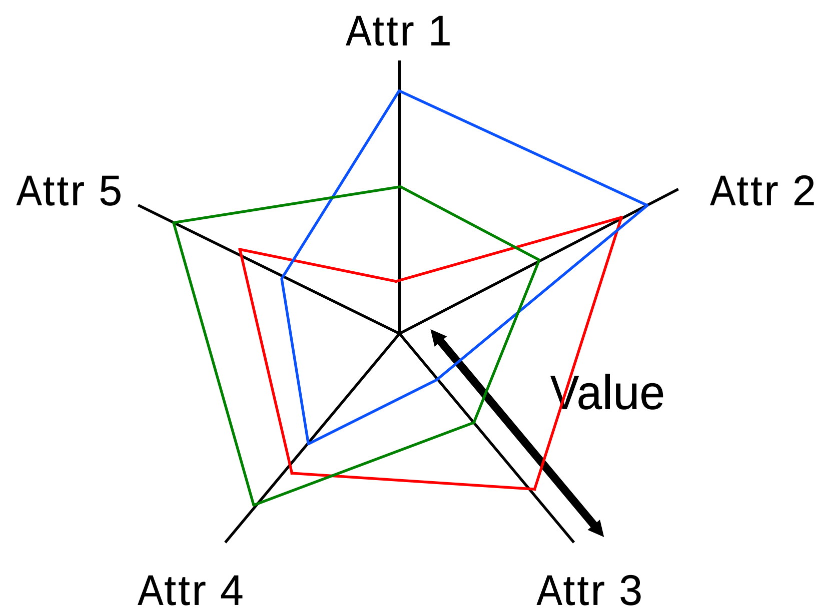
Setosa Versicolor Virginica



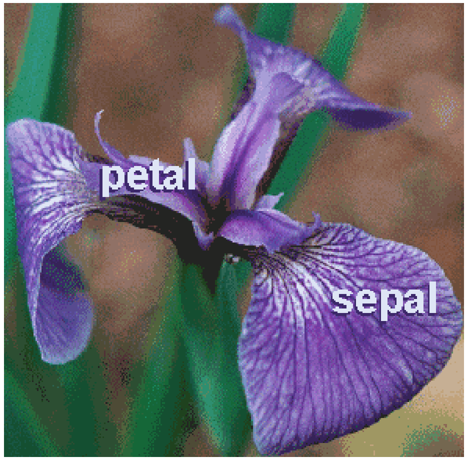
- Characteristics of relationship and correlation patterns between two attributes are easy to see
- Shows relationship across multiple attributes much better

Star Plots

- The attributes axes are equidistantly spaced in polar coordinates
- Each “spoke” encodes an attribute’s values
- Each data record is a polygon – here three data records in colors green, red and blue.
  - Strong interference of value with length of lines
  - Does not scale well with number of attributes
- Similar to star glyphs, which encode only a single data record and use a small multiples layout



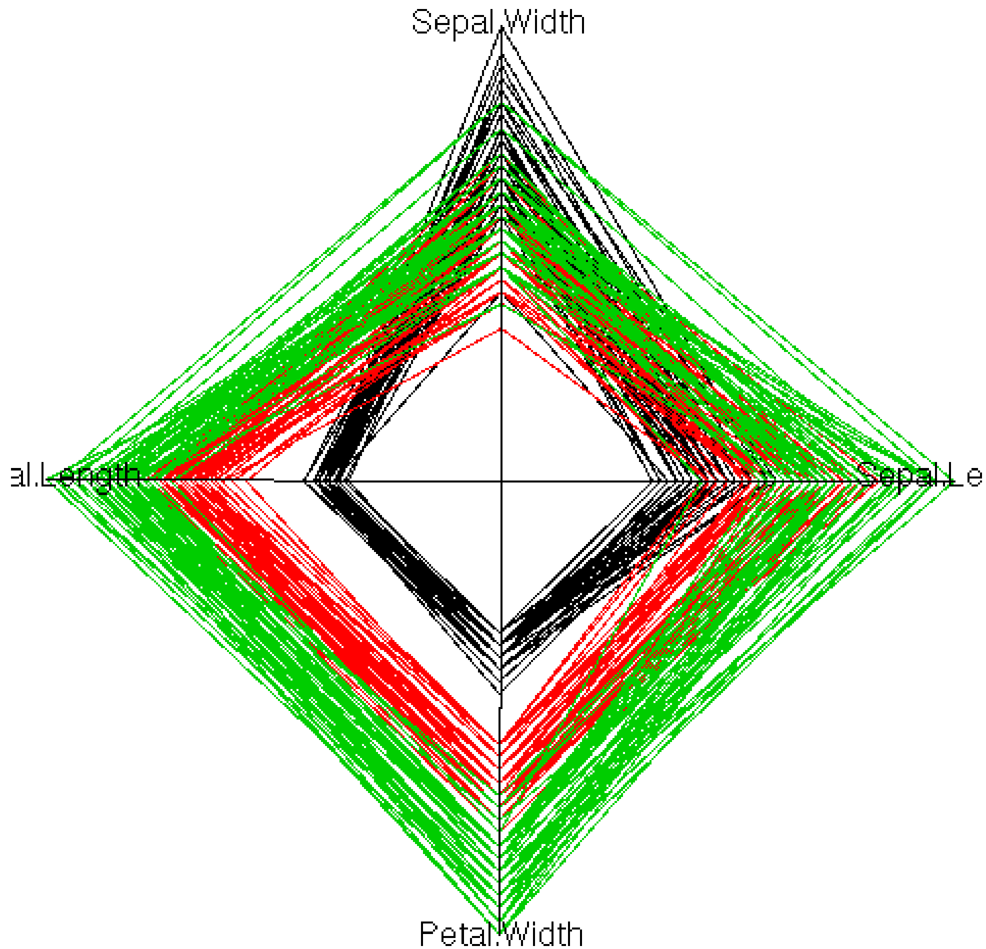
Iris Dataset as a Star Plot



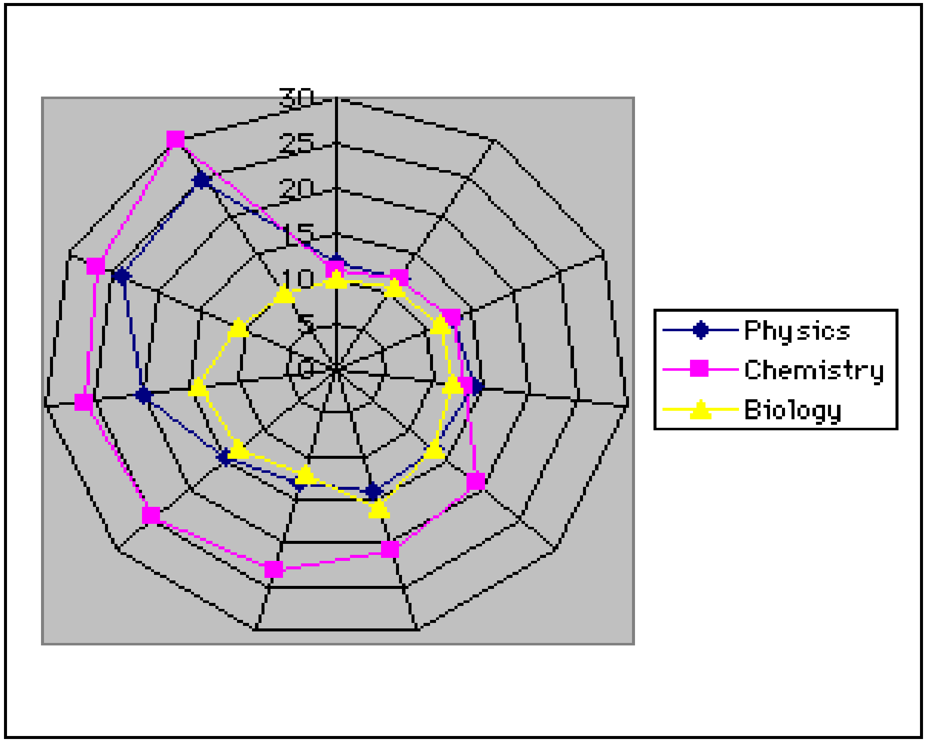


Setosa Versicolor Virginica

Anderson's data of iris flowers on the Gaspé Peninsula, Quebec, Canada



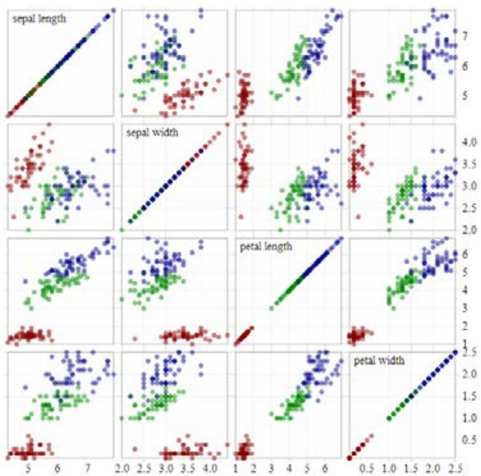
Radar Plot

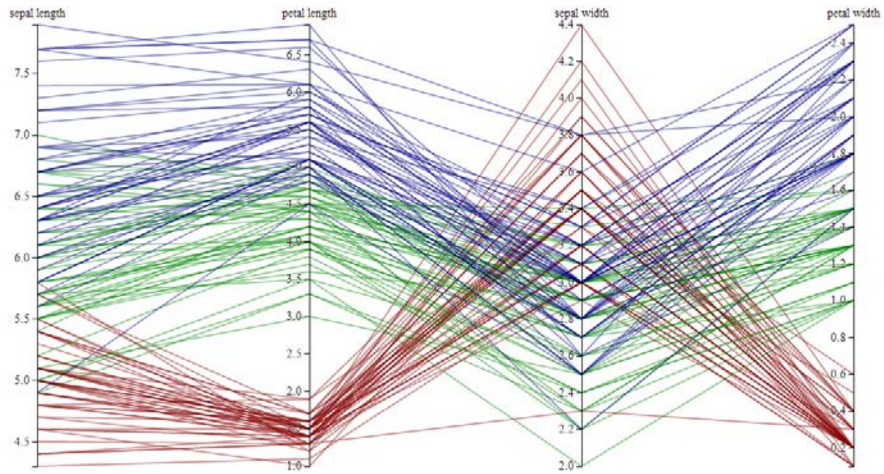


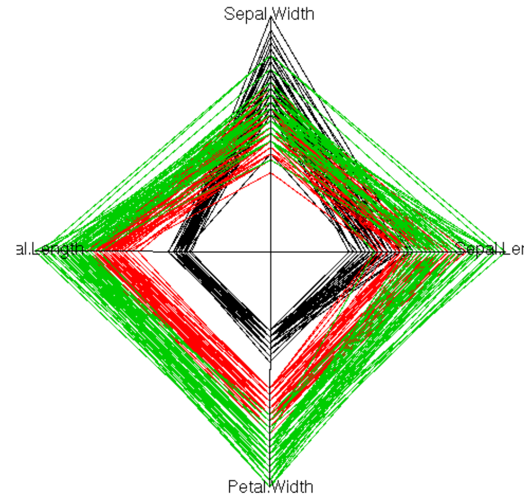
Radar plot for scores of 11 students in 3 different areas

- Each "spoke" encodes a data record instead of an attribute
- Points of different shapes and color are used to encode different attributes (scores in different areas)
- The circular polylines give an impression of the distribution of the attribute values
- Technique has limited scalability to a dozen data items and a few dozen attributes

Comparison of Techniques





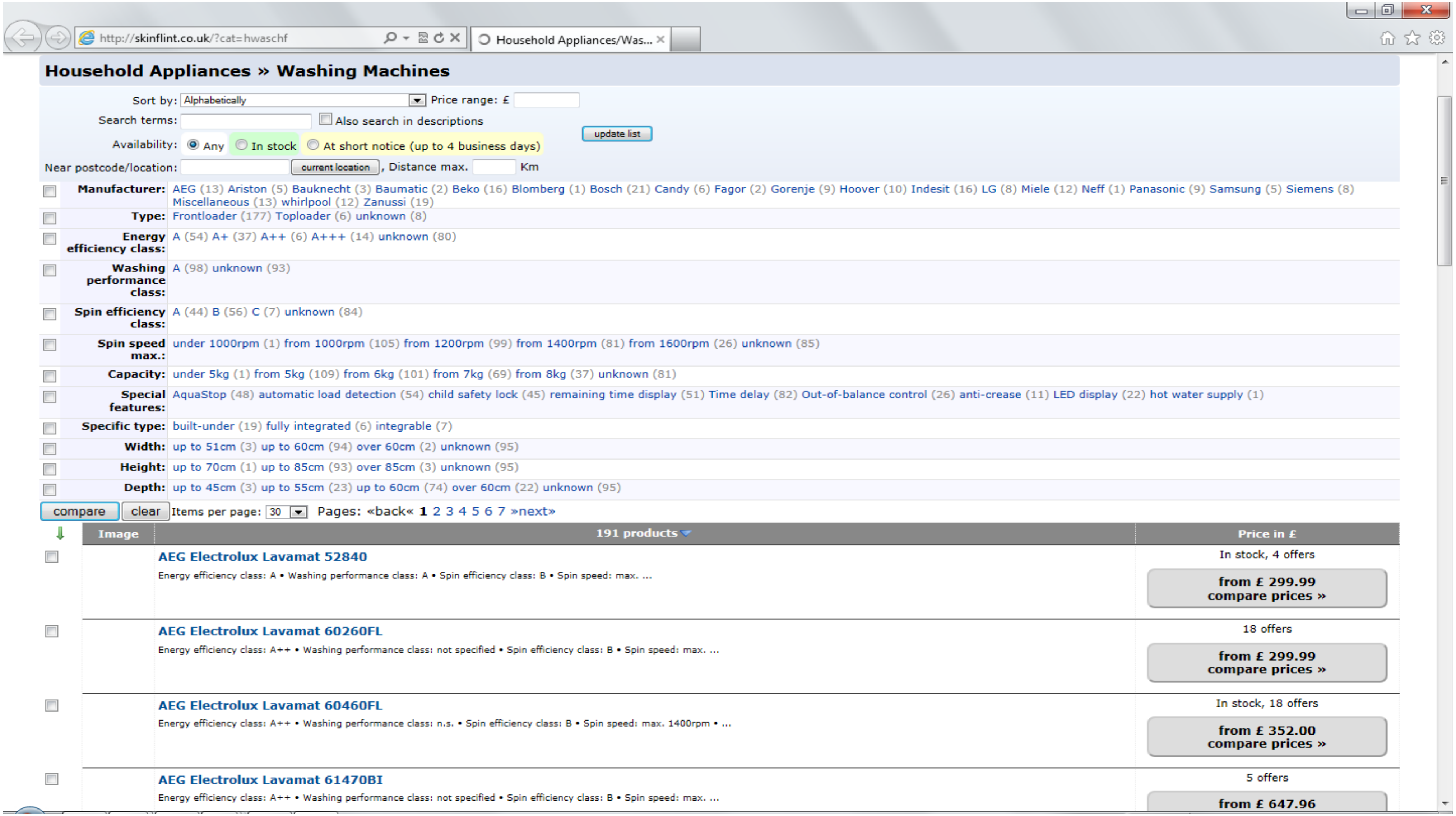


- Scatterplot matrix
  - Rectilinear axes
  - Point marks
  - Small multiples layout of all possible pairs of axes
  - Scalability
    - One dozen attributes
    - Hundreds of items

- Parallel coordinates
  - Parallel axes
    - Axis ordering is major challenge
  - Item as polyline
  - Scalability
    - Dozens of attributes
    - Hundreds of items

- Star plots
  - Radial axes
    - Axis ordering is major challenge
  - Item as polyline
  - Scalability
    - A dozen attributes
    - Hundreds of items

Parallel Coordinates for Multivariate Decision Making



Notebooks & Tablets » Notebooks

Sort by: Alphabetically Price range: £

Search terms: Also search in descriptions

Availability: Any In stock At short notice (up to 4 business days) update list

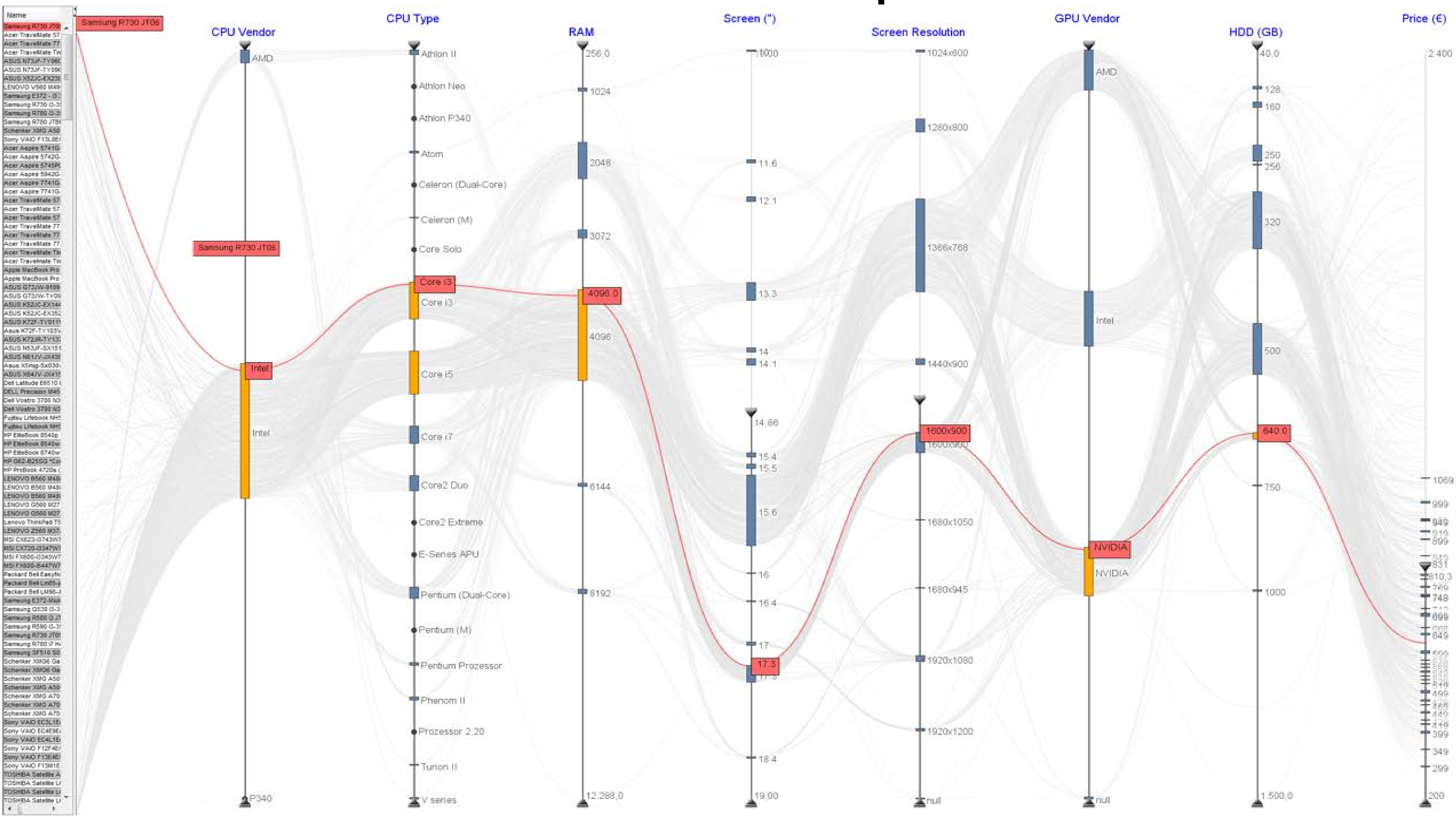
Near postcode/location: current location , Distance max. Km

Manufacturer:	ASUS (347) Acer (318) Apple (42) Dell (44) Fujitsu (81) HP Compaq (679) Lenovo (253) MSI (53) Packard Bell (42) Panasonic (50) Samsung (147) Miscellaneous (67) Sony (109) Toshiba (318)
Screen size:	5.6" (8) 7" (2) 8" (5) 8.9" (1) 10" (17) 10.1" (217) 10.2" (19) 10.4" (4) 11.1" (1) 11.6" (46) 12.1" (78) 12.5" (34) 13.1" (13) 13.3" (221) 13.4" (3) 14" (151) 14.1" (32) 14.5" (3) 15" (10) 15.4" (49) 15.5" (46) 15.6" (1262) 16" (5) 16.4" (7) 17" (19) 17.3" (253) 17.6" (1) 18.4" (21) unknown (22)
Screen size from:	from 5.6" (2528) from 7" (2520) from 8" (2518) from 10" (2512) from 11.1" (2255) from 12" (2208) from 13" (2096) from 14" (1859) from 15" (1673) from 15.4" (1663) from 16" (306) from 17" (294) from 18" (21) unknown (22)
Screen size to:	up to 9.9" (16) up to 10.9" (273) up to 11.9" (320) up to 12.9" (432) up to 13.9" (669) up to 14.9" (855) up to 15.9" (2222) up to 16.9" (2234) up to 17.9" (2507) up to 18.9" (2528) unknown (22)
Aspect ratio:	4:3 (37) 15:9 (231) 16:10 (175) 16:9 (2098) 21:10 (5) unknown (4)
Screen resolution:	800x600 (2) 1024x576 (2) 1024x600 (231) 1024x768 (35) 1280x720 (8) 1280x800 (121) 1366x768 (1609) 1440x900 (41) 1600x768 (5) 1600x900 (340) 1600x945 (3) 1680x945 (1) 1680x1050 (6) 1920x1080 (136) 1920x1200 (7) unknown (3)
Screen pixel pitch:	up to 0.180mm (83) up to 0.190mm (148) up to 0.200mm (306) up to 0.210mm (413) up to 0.220mm (924) up to 0.230mm (1067) up to 0.240mm (1281) up to 0.260mm (2524) up to 0.280mm (2526) over 0.280mm (1) unknown (23)
Screen type:	LCD glare (1366) LCD non-glare (1135) unknown (49)
CPU manufacturer:	AMD (312) ARM (2) Intel (2232) unknown (4)
Number of cores:	Single-Core (213) Dual-Core (2055) triple-core (15) Quad-Core (265) unknown (2)
CPU clock:	under 1GHz (2) from 1GHz (2543) from 1.5GHz (2433) from 2GHz (1867) from 2.5GHz (562) from 3GHz (1) unknown (5)
CPU TDP:	up to 5W (78) up to 10W (291) up to 15W (300) up to 20W (528) up to 25W (612) up to 30W (614) up to 35W (2281) up to 50W (2509) over 50W (1) unknown (40)
CPU type:	AMD A4-3 (23) AMD A6-3 (48) AMD A8-3 (4) AMD C- (17) AMD E- (87) AMD E2- (4) AMD V (7) Athlon 64 (7) Athlon II (37) Athlon Neo (6) Athlon X2 (4) Atom (256) Celeron (59) Celeron Dual-Core (45) Core 2 Duo (97) Core 2 Quad (1) Core 2 Solo (2) Core Duo (3) Core Solo (2) Core i3 (222) Core i3-2 (364) Core i5 (129) Core i5-2 (524) Core i5-3 (9) Core i7 (45) Core i7-2 (226) Core i7-3 (38) Cortex (2) Pentium B (89) Pentium Dual-Core (47) Pentium P (72) Pentium U (1) Pentium-M (1) Phenom II (29) Sempron (10) Turion 64 (3) Turion II (11) Turion II Ultra (4) Turion X2 (9) Turion X2 Ultra (2) unknown (4)
Memory:	under 512MB (2) from 512MB (2547) from 1GB (2539) from 2GB (2283) from 3GB (1882) from 4GB (1708) from 6GB (433) from 8GB (199) from 16GB (11) unknown (1)
Number of memory modules:	1x (1993) 2x (545) 4x (8) unknown (4)
Hard drive capacity:	under 8GB (5) from 8GB (2545) from 32GB (2539) from 80GB (2517) from 120GB (2505) from 250GB (2279) from 320GB (1992) from 500GB (1243) from 1000GB (88)
HDD type:	Flash (17) HDD (2355) HDD and Flash (12) HDD and SSD (20) SSD (146)
Optical drive:	Blu-ray (BD-R/RE) (25) Blu-ray (BD-R/RE/RW) (3) Blu-ray (BD-ROM) (155) DVD+/-RW (1834) DVD-ROM (2) DVD/CD-RW Combo (5) HD DVD (1) without (524) unknown (1)
Graphics (manufacturer/type):	AMD (dedicated) (331) AMD (IGP) (215) Intel (IGP) (1558) NVIDIA (dedicated) (399) NVIDIA (IGP) (28) SiS (IGP) (2) Miscellaneous (IGP) (17)
GPU series:	IGP (1761) Radeon Xpress (1) FirePro (5) Mobility Radeon X1 (5) Mobility Radeon HD 3 (16) Mobility Radeon HD 4 (33) Mobility Radeon HD 5 (68) Radeon HD 6 (313) Radeon HD 7 (48) Intel GMA9 (4) Intel GMA X3 (19) Intel GMA X4 (117) Intel HD Graphics (1114) Intel HD Graphics 2000 (6) Intel HD Graphics 3000 (609) GeForce 3 (25) GeForce 4 (3) GeForce 8 (10) GeForce 9 (9) GeForce GT (289) GeForce GT 1 (6) GeForce GT 2 (5) GeForce GT 3 (19) GeForce GT 4 (6) GeForce GT 5 (128) GeForce GT 6 (66) GeForce GTX 4 (19) GeForce GTX 5 (20) GeForce GTX 6 (16) GeForce G1 (7) GeForce G2 (3) GeForce Go (1) Quadro FX (15) Quadro NVS (24) Quadro (64)
Connectors:	USB 3.0 (692) USB 2.0 (2528) FireWire (351) Modem (345) LAN 10/100 (738) Gb LAN (1761) WLAN 802.11g (142) WLAN 802.11n (2403) Bluetooth (1616) eSATA (622) DVI (3) HDMI (1538) DisplayPort (387) PCMCIA (62) ExpressCard (678) Thunderbolt (29)
Number of USB ports:	under 2x (18) from 2x (2506) from 3x (2209) from 4x (834) from 5x (85) unknown (26)
number of USB ports 3.0:	from 1x (691) from 2x (326) from 3x (39) unknown (1859)
Card reader:	2in1 (481) 3in1 (212) 4in1 (504) 5in1 (568) 6in1 (77) 7in1 (55) 8in1 (44) 9in1 (12) SD-Card (181) unknown (416)
Webcam resolution:	without (373) 0.3 megapixels (594) 1.0 megapixels (67) 1.3 megapixels (587) 2.0 megapixels (339) 3.0 megapixels (4) 3 megapixels (7) unknown (579)
Operating System:	Windows 7 Starter (198) Windows 7 Home Premium (1324) Windows 7 Professional (791) Windows Vista Home Basic (11) Windows Vista Home Premium (68) Windows Vista Business (22) Windows XP Home (26) Windows XP Professional (49) Mac OS X (42) chrome OS (4) Android (28) Linux (12) Express Gate (1) without operating system (20)
Battery type:	Li-Ion (2433) Li-Polymer (86) unknown (31)
Battery cells:	from 3 cells (1510) from 4 cells (1469) from 6 cells (1423) from 8 cells (214) from 9 cells (55) unknown (1040)
Weight:	up to 1kg (20) up to 1.25kg (168) up to 1.5kg (430) up to 2kg (648) up to 2.5kg (1284) up to 3kg (2253) up to 4kg (2491) up to 5kg (2512) over 5kg (5) unknown (33)
Special features:	Netbook (256) LED (2251) 3D-Ready (20) Webcam (2177) UMTS (192) HSPA/HSDPA/HSUPA (137) GPS (23) keypad (425) illuminated keyboard (93) Fingerprint reader (393) TV-tuner (8) multi touch touchscreen (34) multi-touch trackpad (1082) touchscreen (78) Tablet (66) Convertible (37) UMPC (8) Wireless display (57) fanless (12) Crossfire/SLI (3) educational offer (4)
Warranty:	12 months (1475) 24 months (805) 24/12 months (4) 36 months (265) unknown (1)

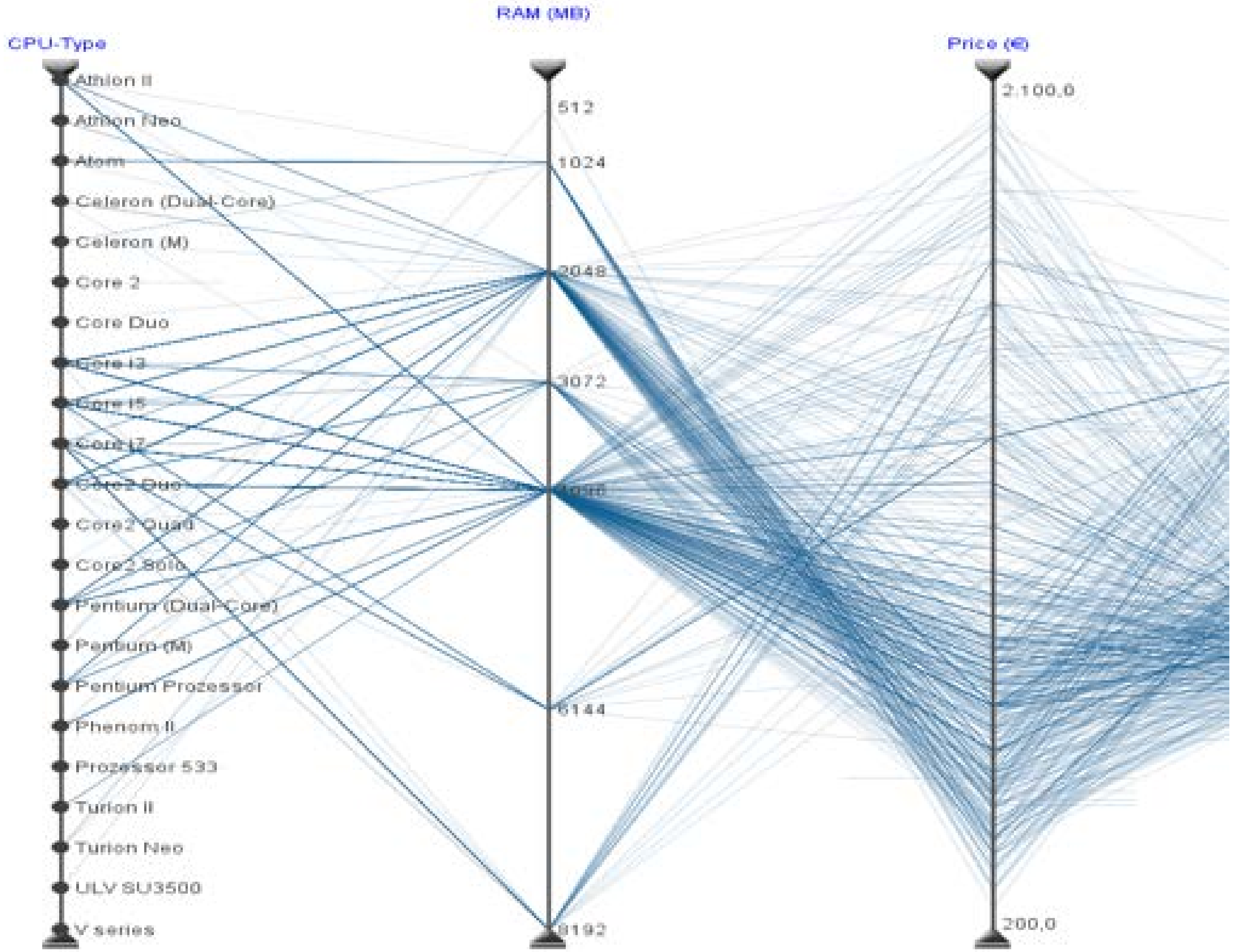
compare clear Items per page: 30 Pages: «back« 1 2 3 4 5 6 7 8 9 10 11 ... 83 84 85 »next»

Image	2550 products	Price in £
	**Acer Aspire 5250-3004G32Mnkk, red (NX.RJYEK.005)** AMD E-300 2x 1.30GHz • 4096MB (2x 2048MB) • 320GB • DVD+/-RW DL • AMD Radeon HD 6310 (IGP) shared memory • 3x USB ...	At short notice, 4 offers **from £ 299.99** **compare prices »**
	**Acer Aspire 5250-E304G50Mnkk, black (LX.RJY02.114)** AMD E-300 2x 1.30GHz • 4096MB (2x 2048MB) • 500GB • DVD+/-RW DL • AMD Radeon HD 6310 (IGP) shared memory • 3x USB ...	In stock, 20 offers **from £ 313.91** **compare prices »**
	**Acer Aspire 5253-C52G50Mnkk, black (LX.RD502.059)** AMD C-50 2x 1.00GHz • 2048MB • 500GB • DVD+/-RW DL • AMD Radeon HD 6310 (IGP) shared memory • 3x USB 2.0/LAN/WLAN ...	1 offer **£ 337.19** **compare prices »**

The Product Explorer



Product Data - Attribute Types

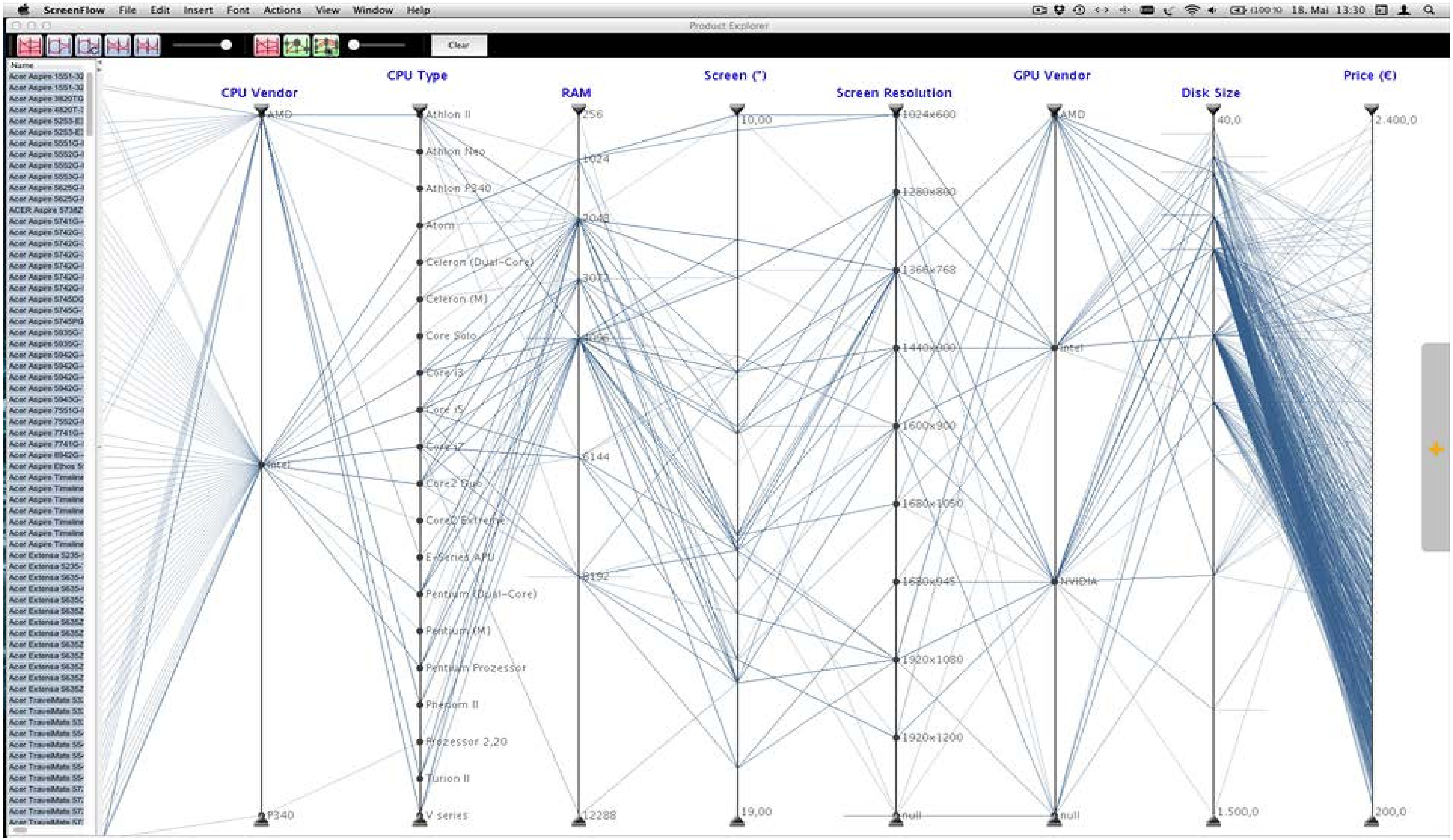


Categorial

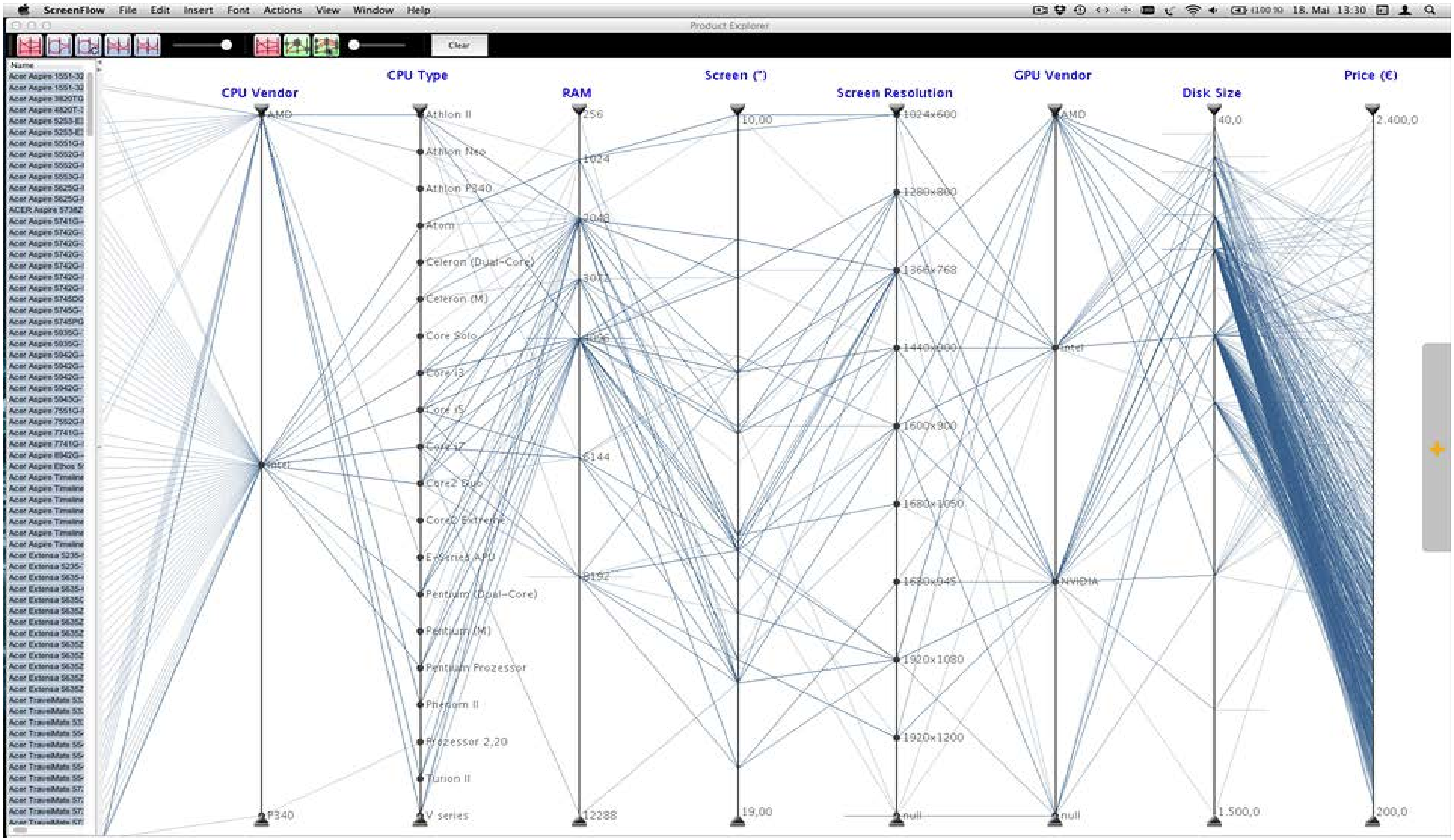
Ordinal
(pseudo-continuous)

Continuous

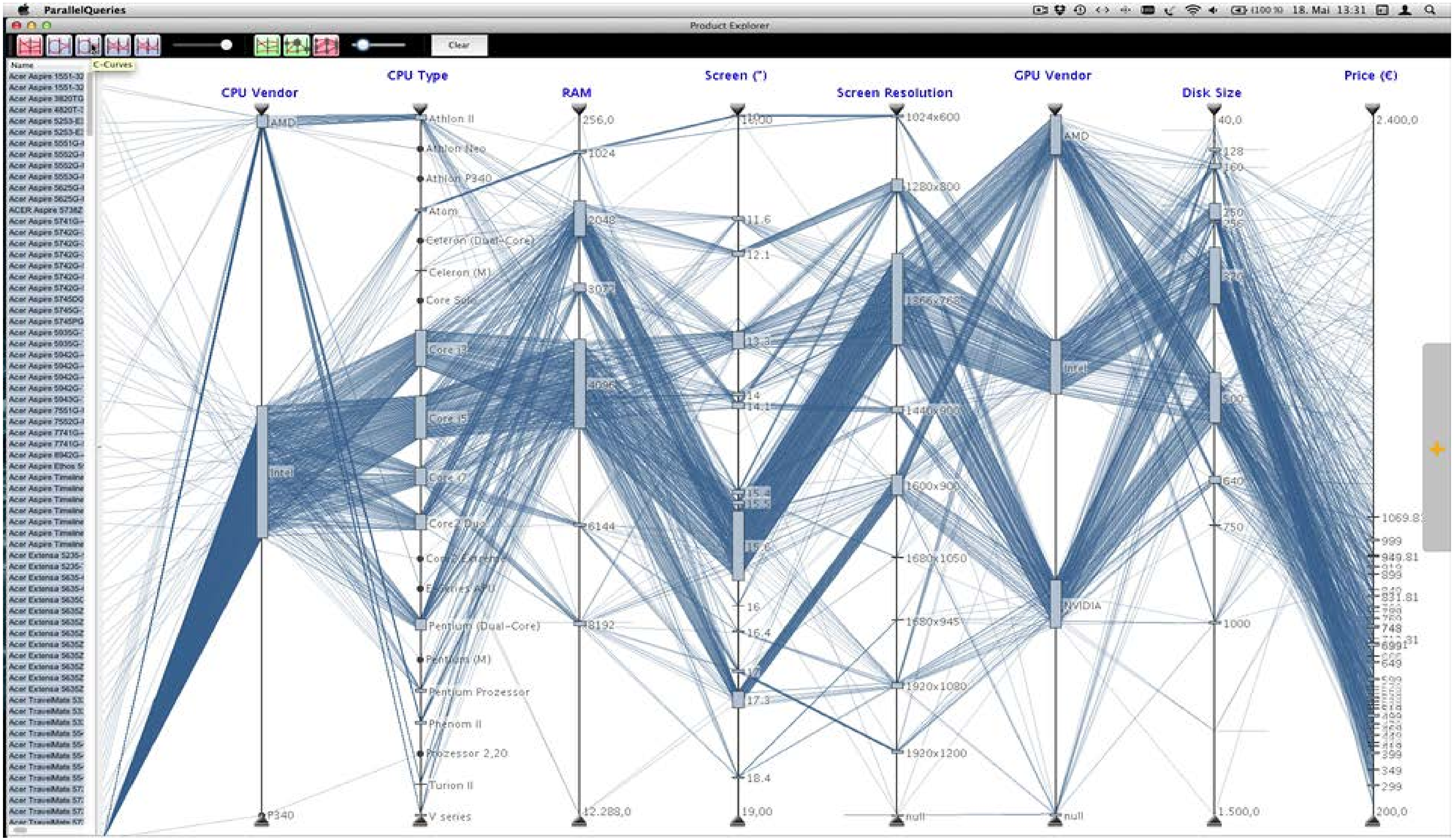
Extended Areas



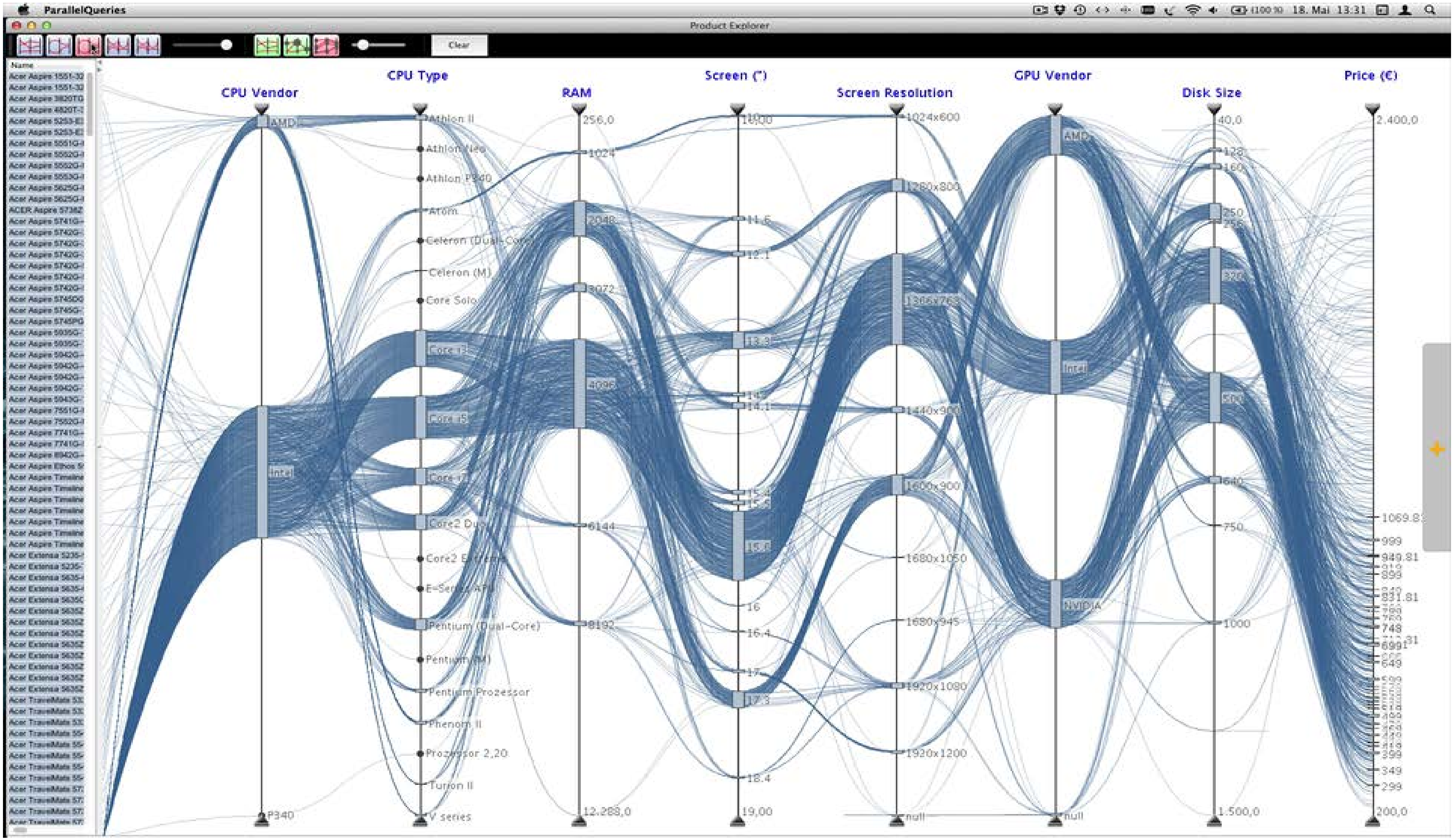
Extended Areas



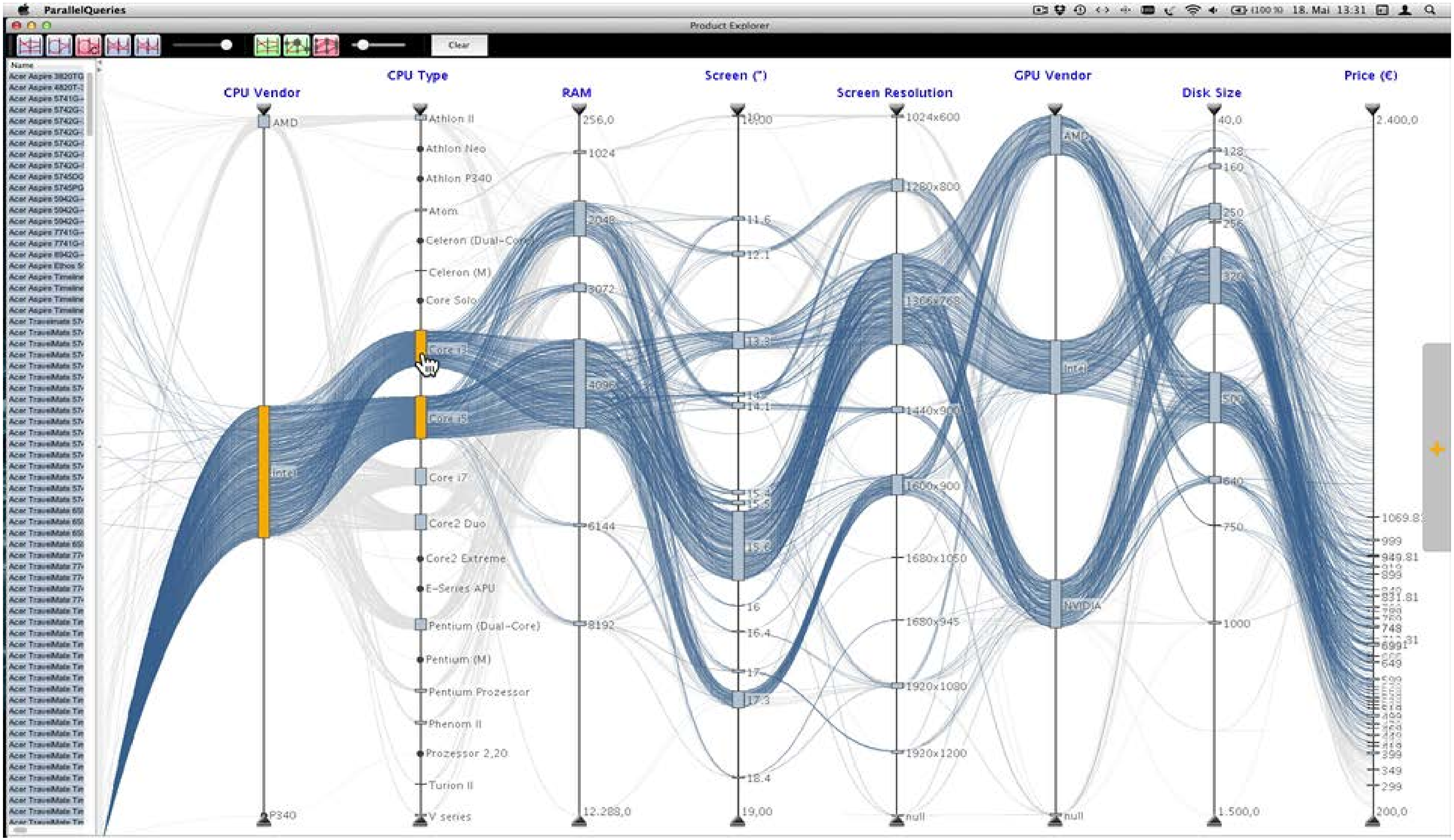
Cubic Curves



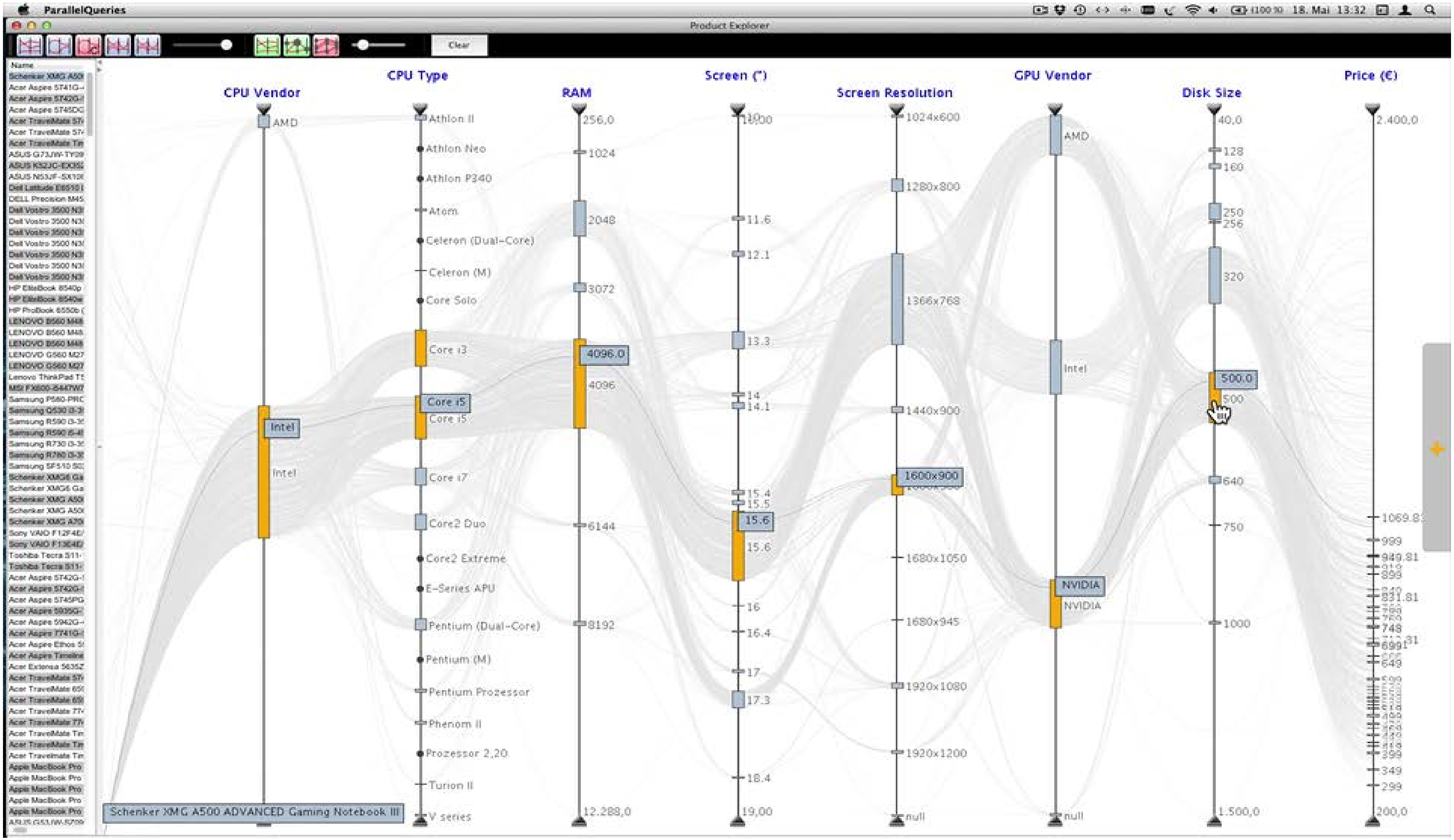
Visual Query Generation



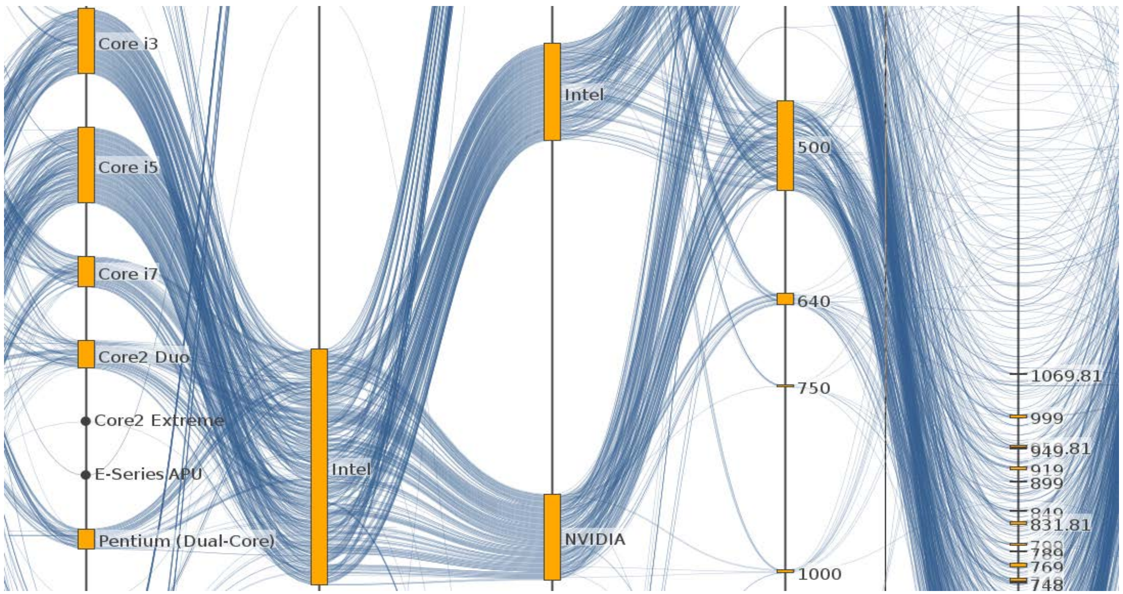
Visual Query Generation

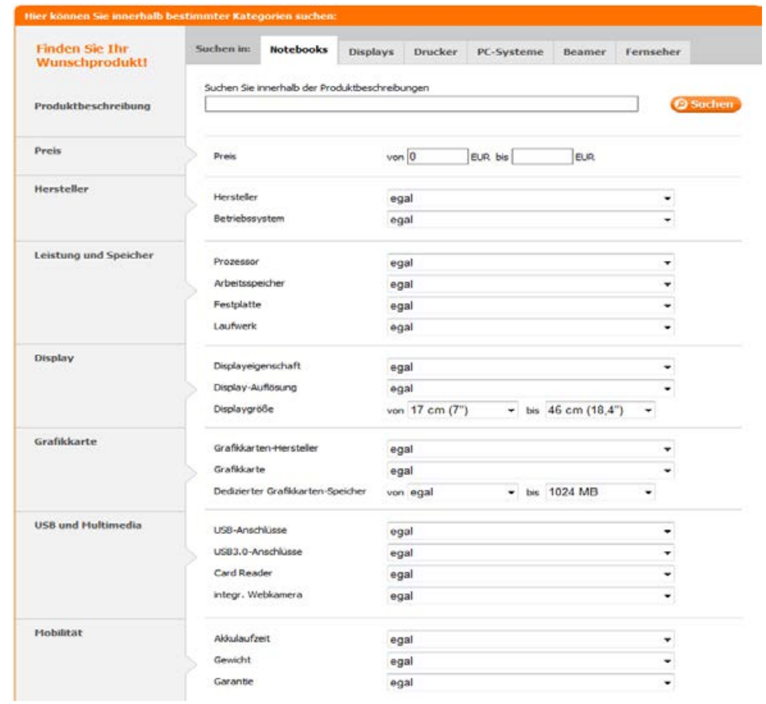


Visual Query Generation

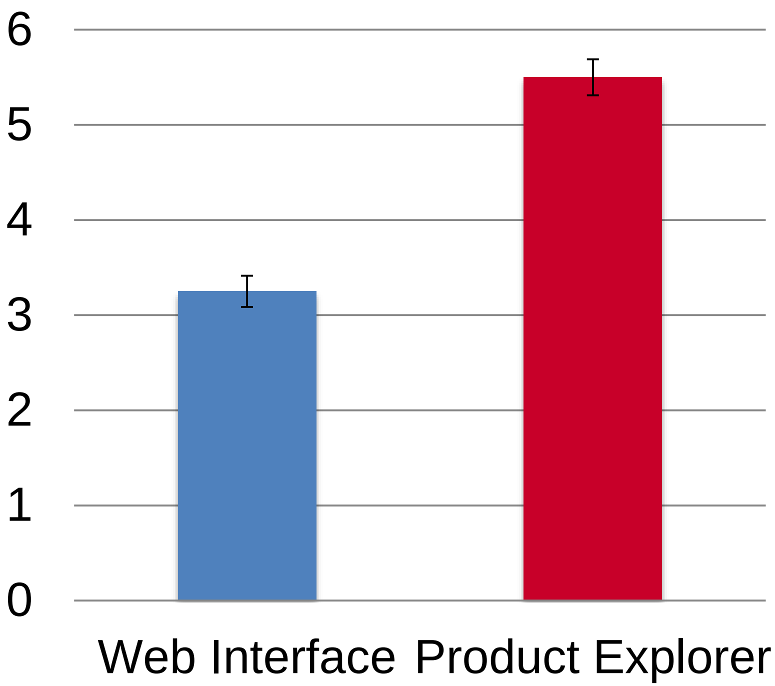


User Study: Product Explorer vs. Webshop

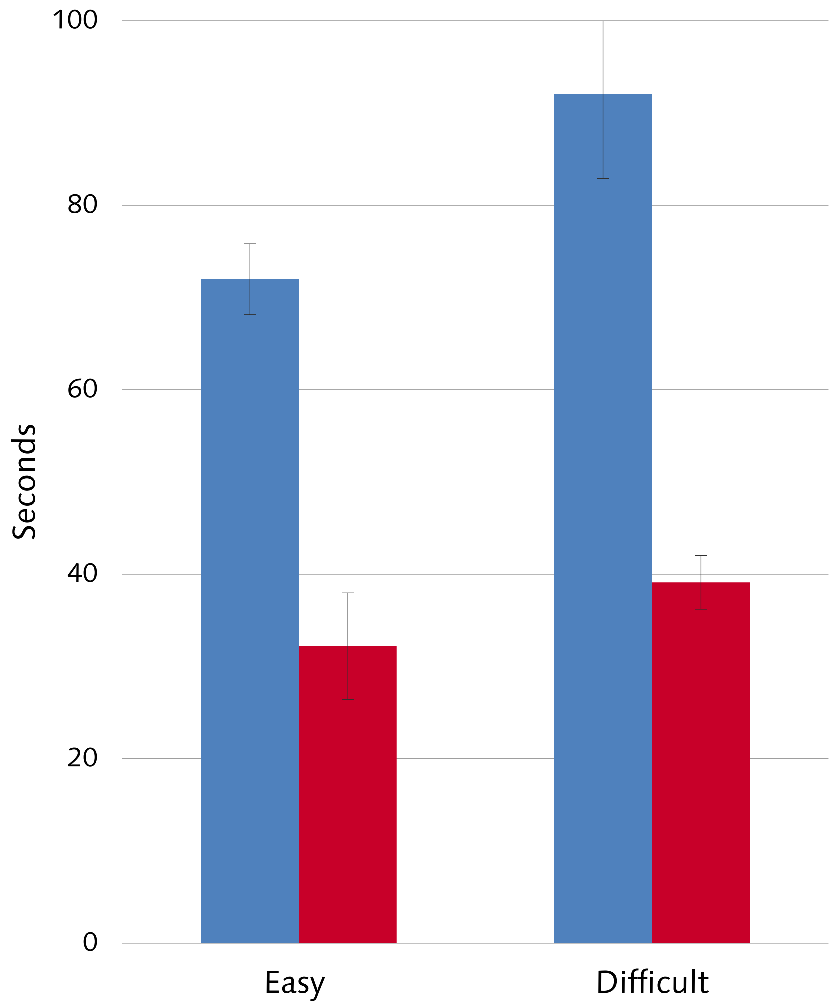




User Study



Preferences

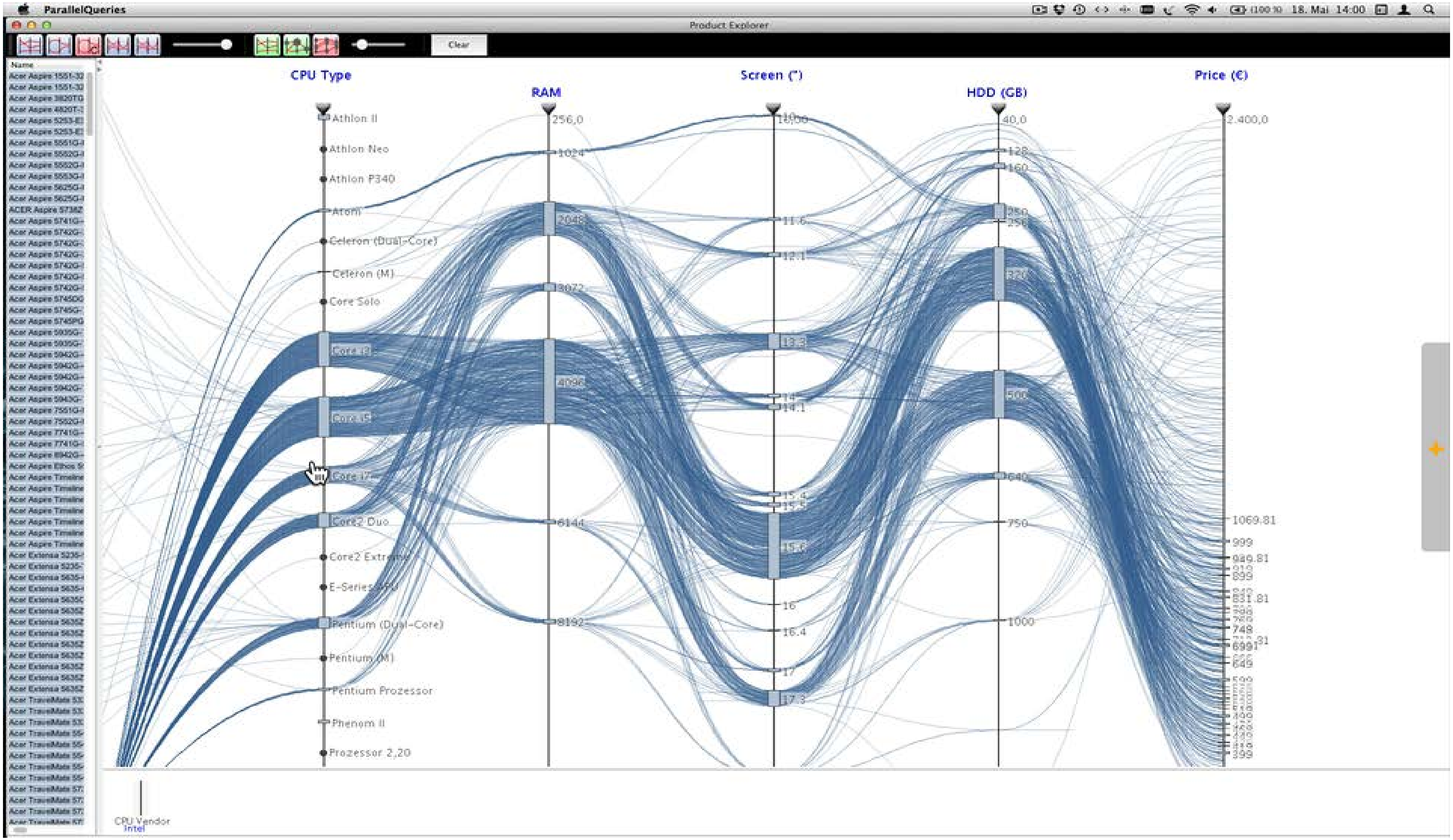


Task Completion Times

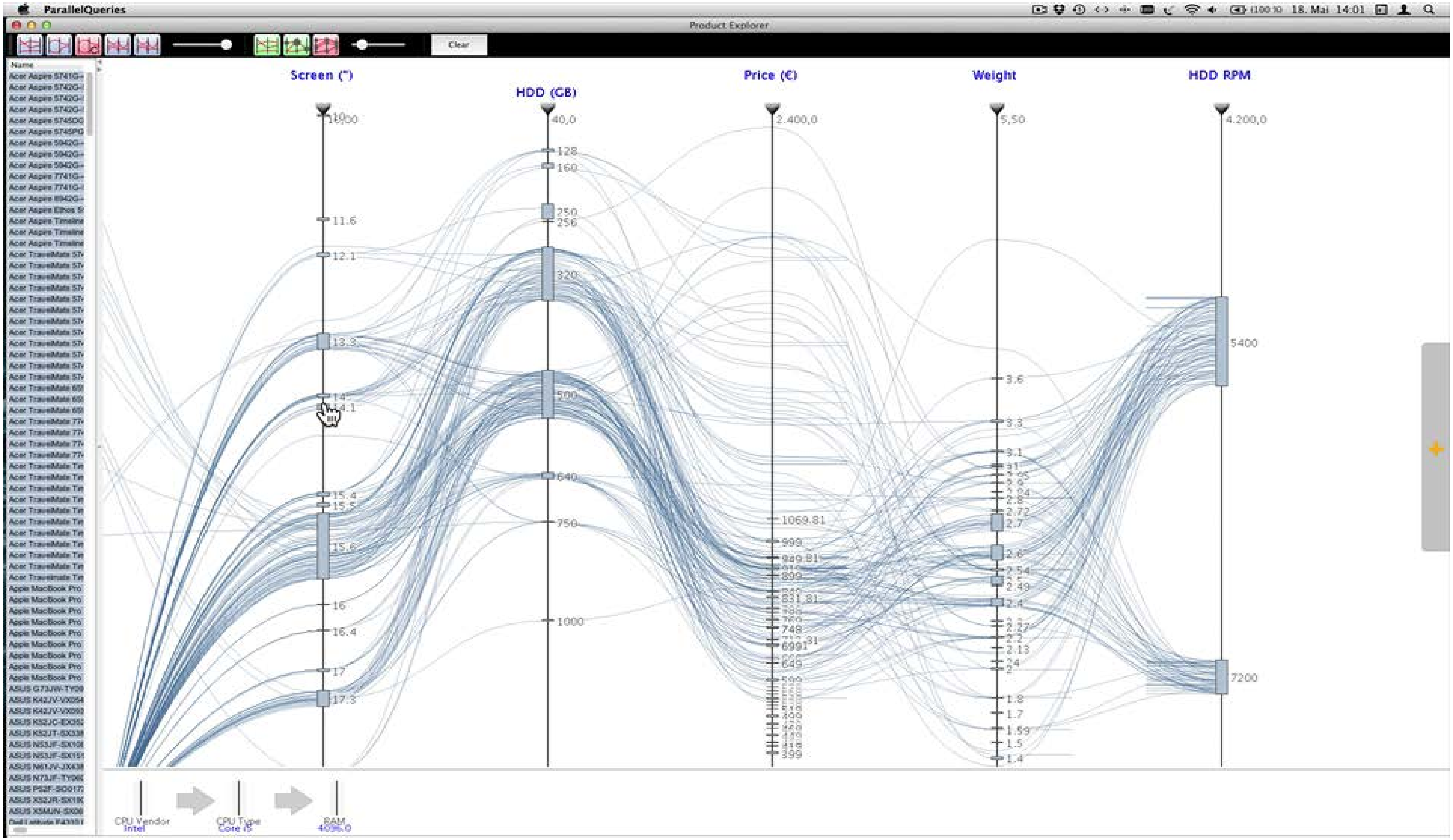
Reduce Clutter

- Exclusive decisions
- Axis repository

Exclusive Decisions and Axis Repository



Exclusive Decisions and Axis Repository



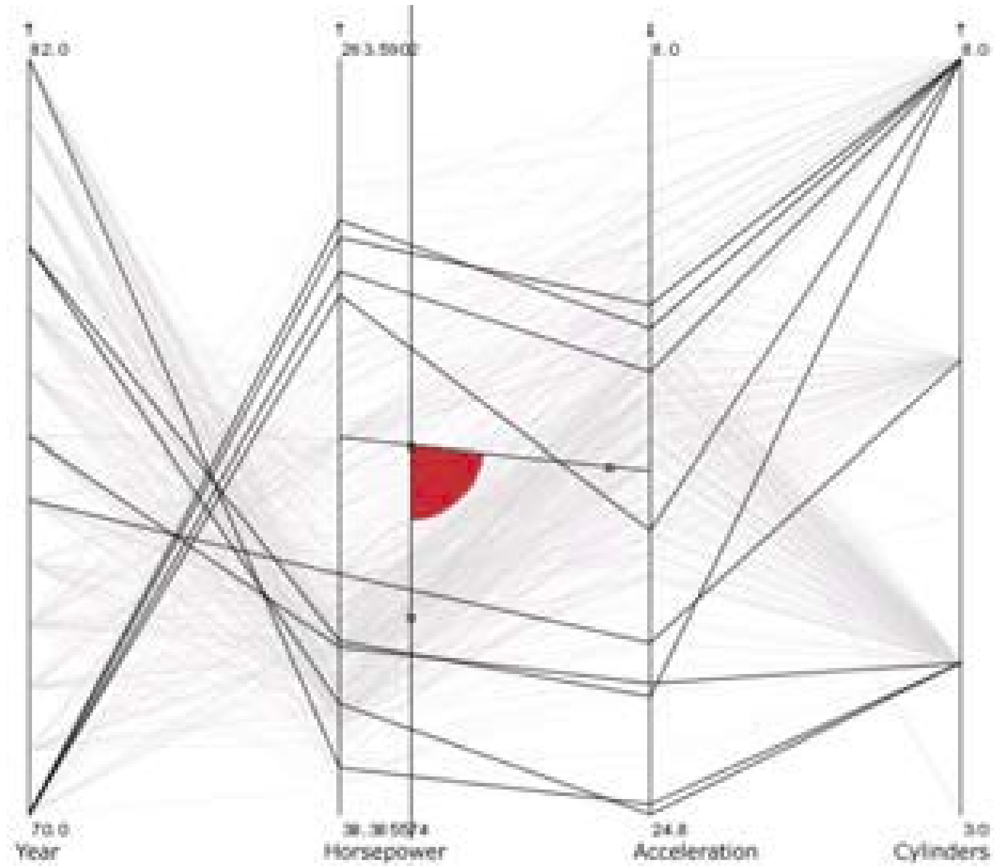
Product Explorer

- Parallel Coordinates for product data
  - Cubic curves and extended areas improve traceability of data records
  - Intuitive visual interface based on a set of simple rules
- User study confirms usability
- Complexity and scalability issues
  - Attribute repository
  - Decision bar

Riehmann, P., Opolka, J., Froehlich, B.
The Product Explorer: Decision Making with Ease
Advanced Visual Interfaces (AVI) 2012.

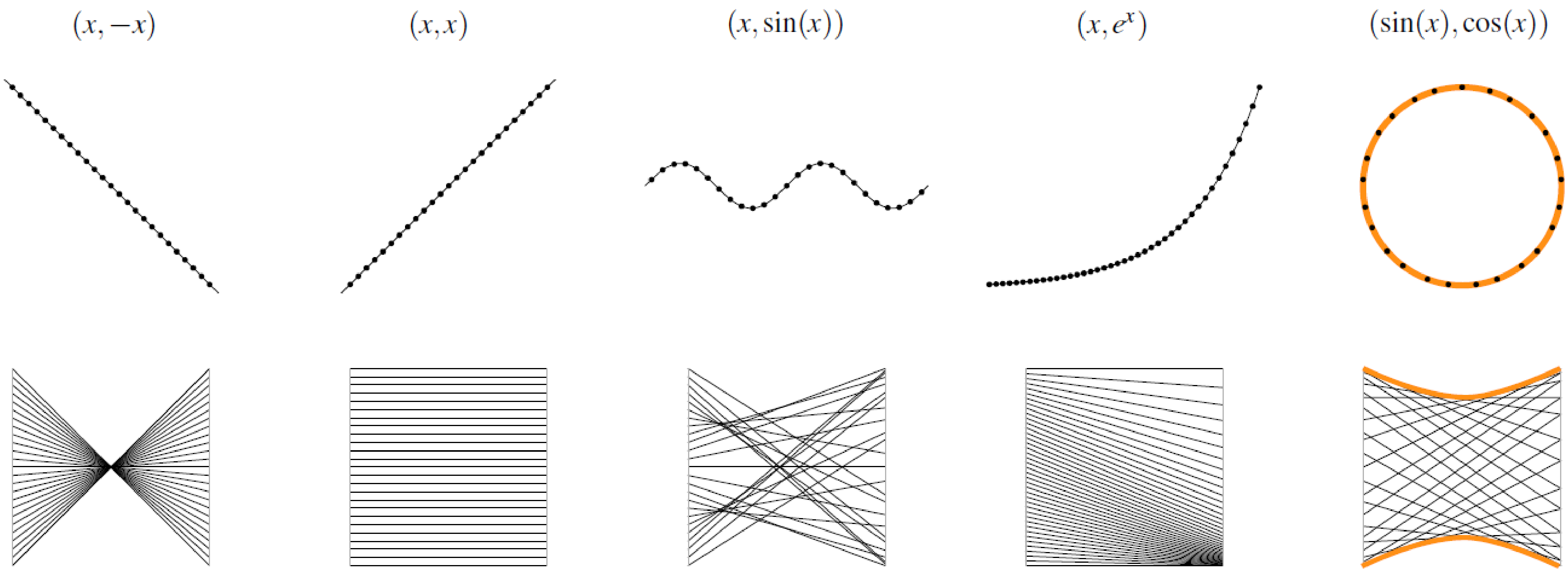
Interaction: Angular Brushing

http://old.vrvis.at/via/research/ang-brush/



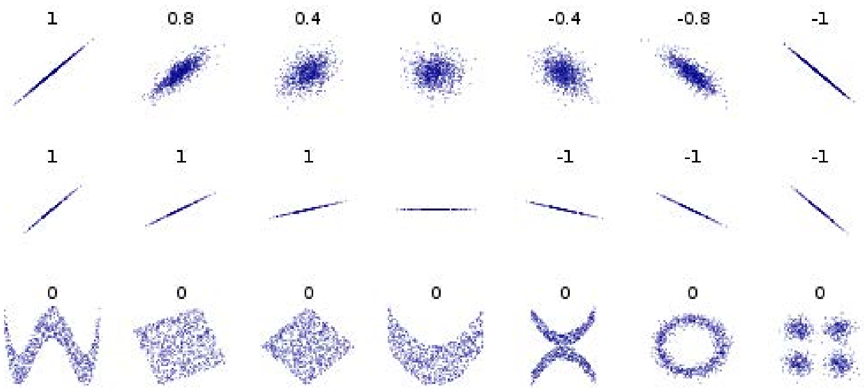
[Angular Brushing Video]()

Parallel Coordinates Patterns

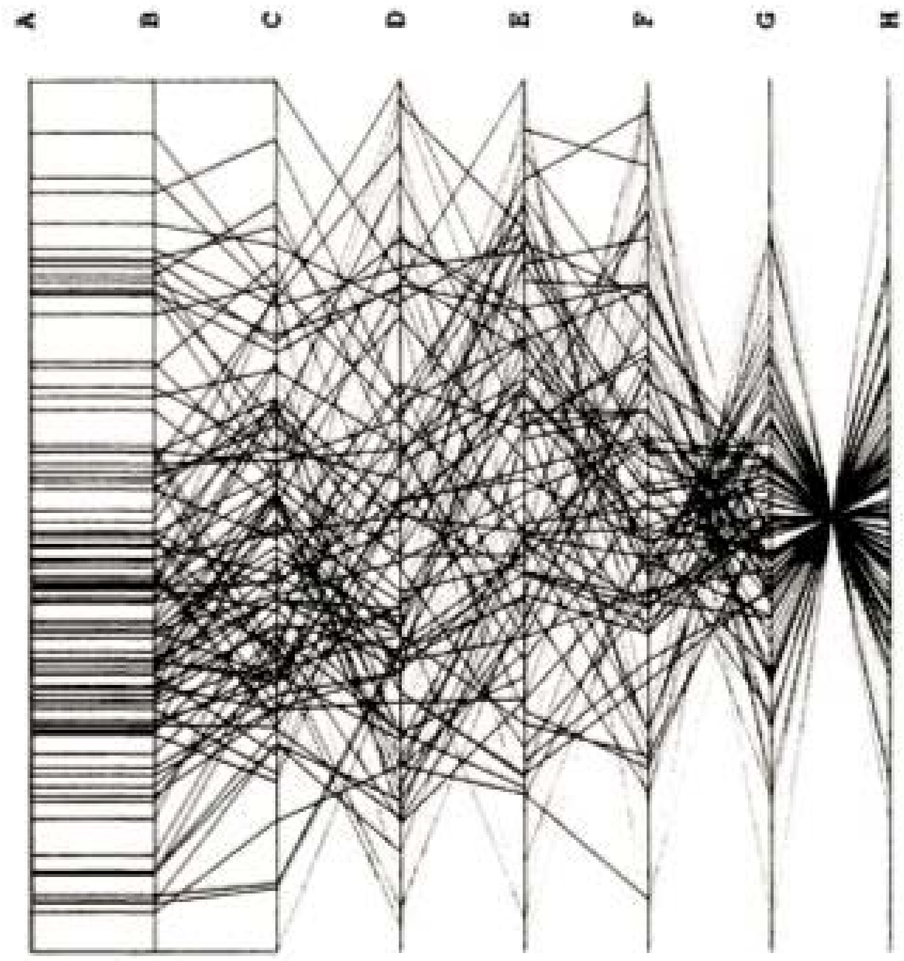


Top: Cartesian coordinates Bottom: Parallel Coordinates

Correlation Between Axis Pairs

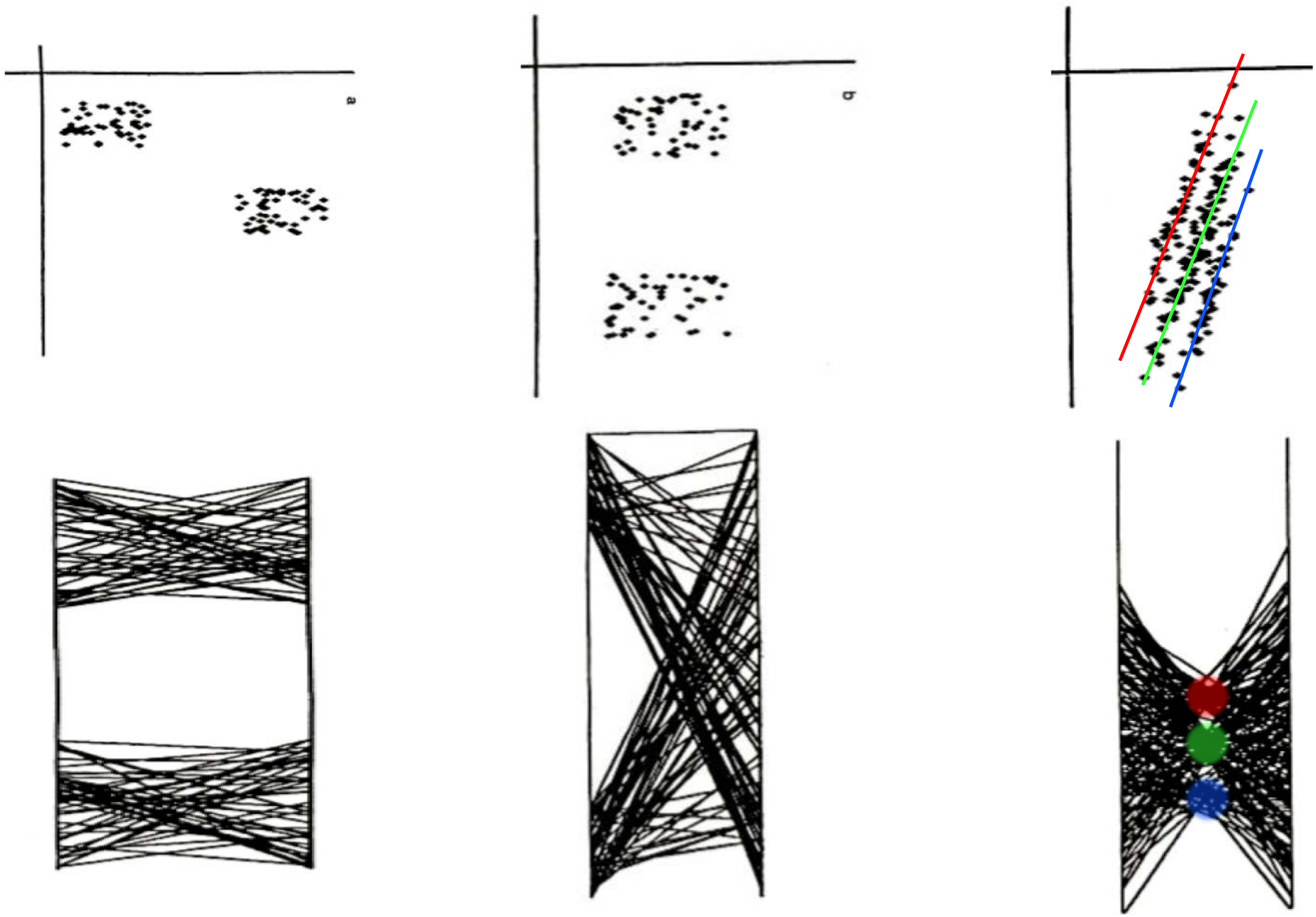


Correlations in Cartesian space



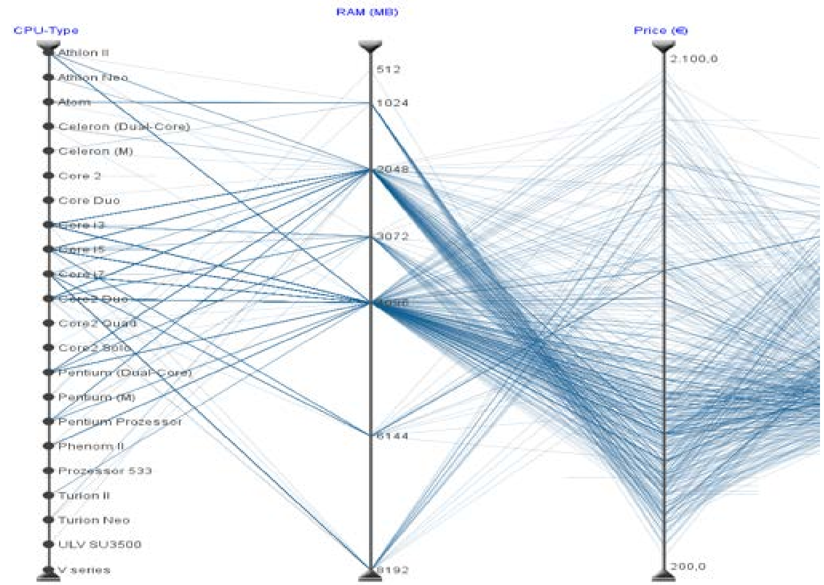
Correlation coefficient: 1 0.8 0.2 0 -0.2 -0.8 -1.0

Cluster Separation in Parallel Coordinates



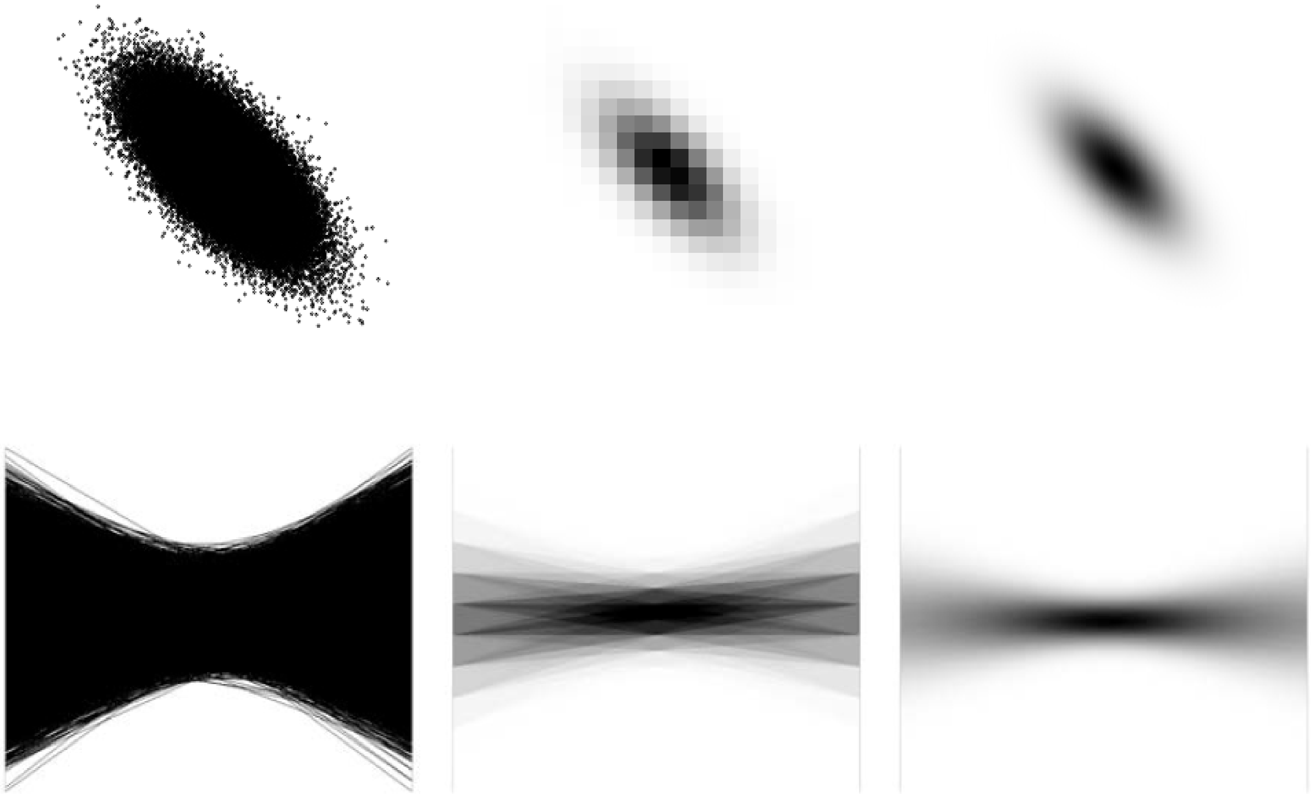
Parallel Coordinates: Issues

- Categorical and ordinal data have to be mapped to a metric scale before it can be visualized in parallel coordinates.
- Equally spaced parallel axes
  - Varying axes scale, although indicating relationships, may cause confusion
  - Connecting the data points can be misleading
  - Close axes as number of attributes increases
- The order of axes implicitly defines which patterns emerge between adjacent axes.
- The line-tracing problem occurs if two or more lines intersect an axis at the same position.
- Overplotting occurs in parallel coordinates which potentially occludes patterns in the data.



Overplotting

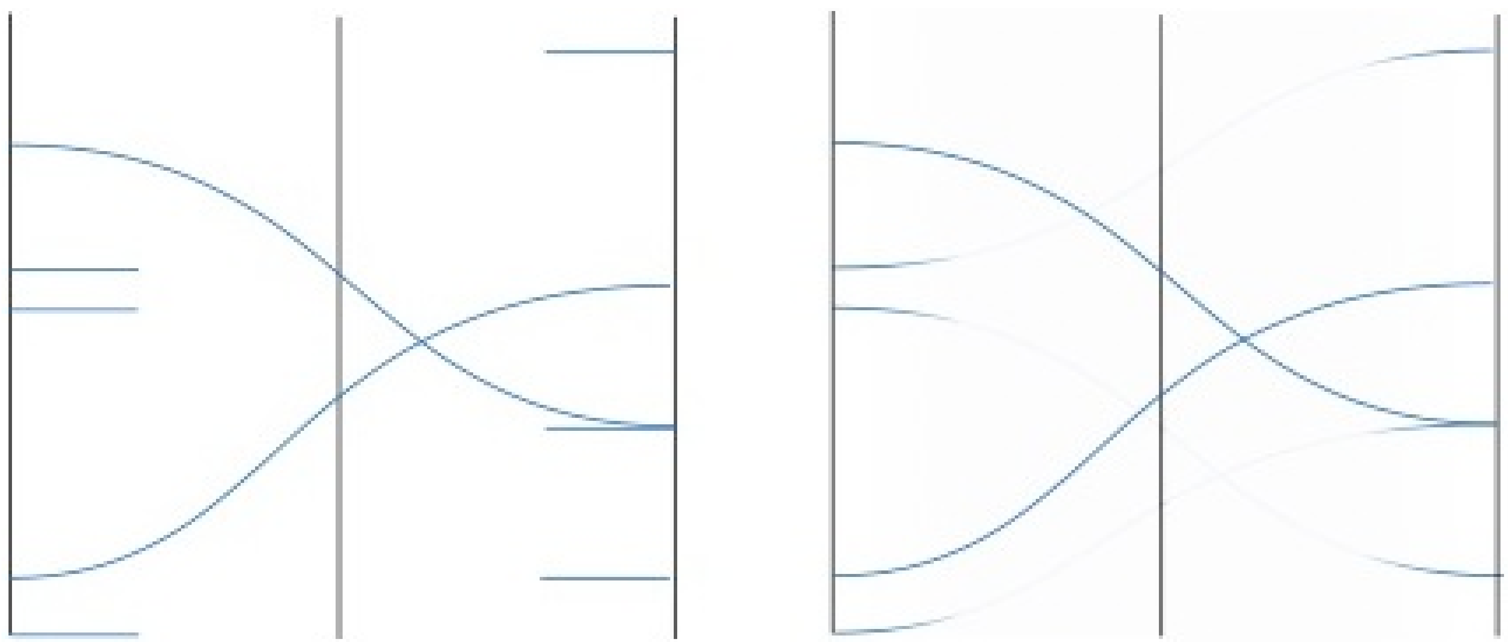
- Overplotting potentially occludes patterns in the data.



Traditional parallel coordinates Binned parallel coordinates Line-density model
100 000 observations drawn from a bivariate normal distribution (top, left)

Parallel Coordinates: Issues

- How to cope with missing attributes?
  - Often not all the attributes are specified
  - Draw stubs?
  - Draw lines that become transparent behind the unspecified attribute axis?



Parallel Coordinates: Summary

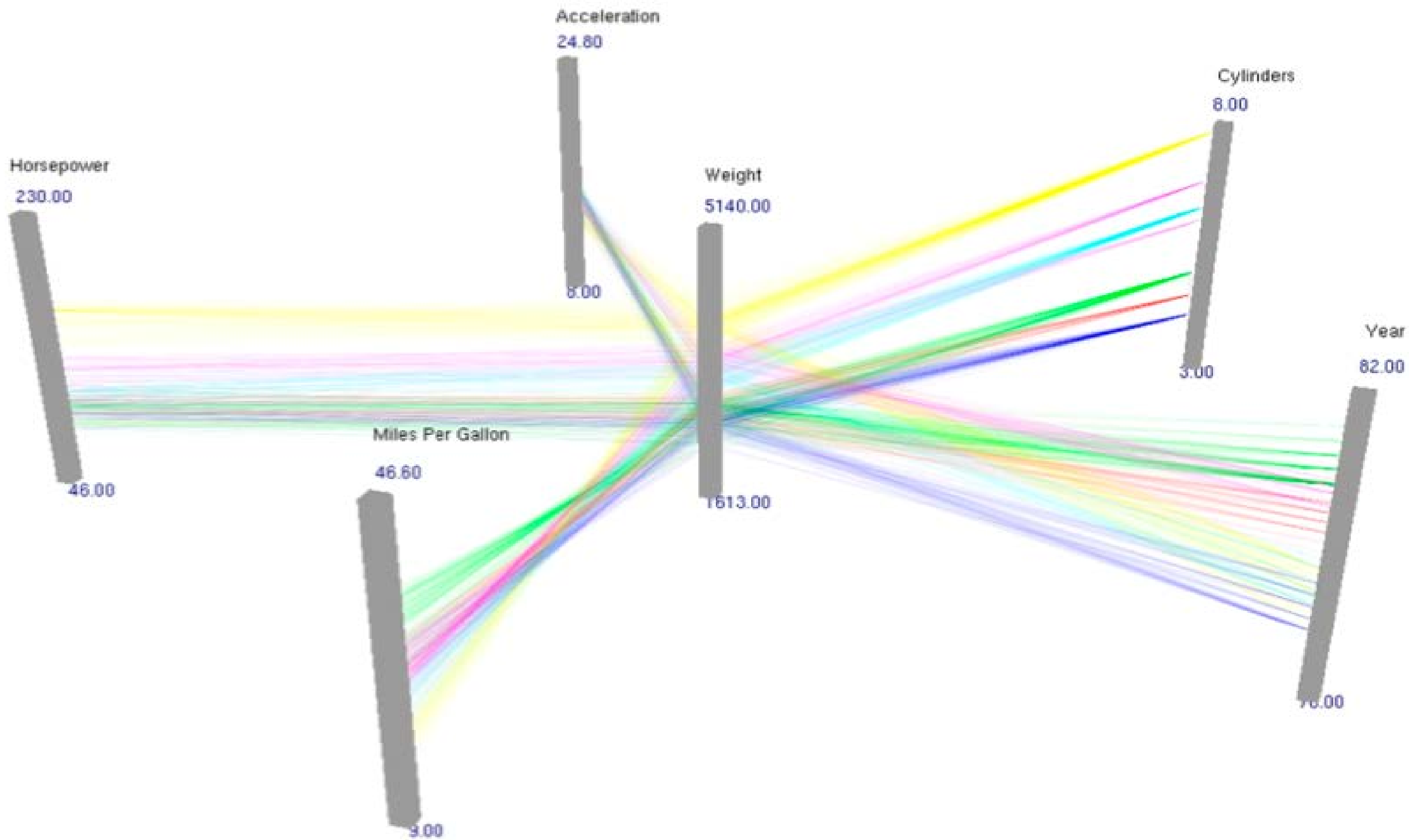
- Multi-dimensional data can be **visualized** in two dimensions with low complexity.
- Each variable is treated **uniformly**.
- Axes can be **swapped** by user.
- **Relations** within multi-dimensional data can be discovered ("visual data mining").
- May serve as a **preprocessor** to other methods.

Some more discussions:
http://eagereyes.org/techniques/parallel-coordinates

Some Extensions of Parallel Coordinates

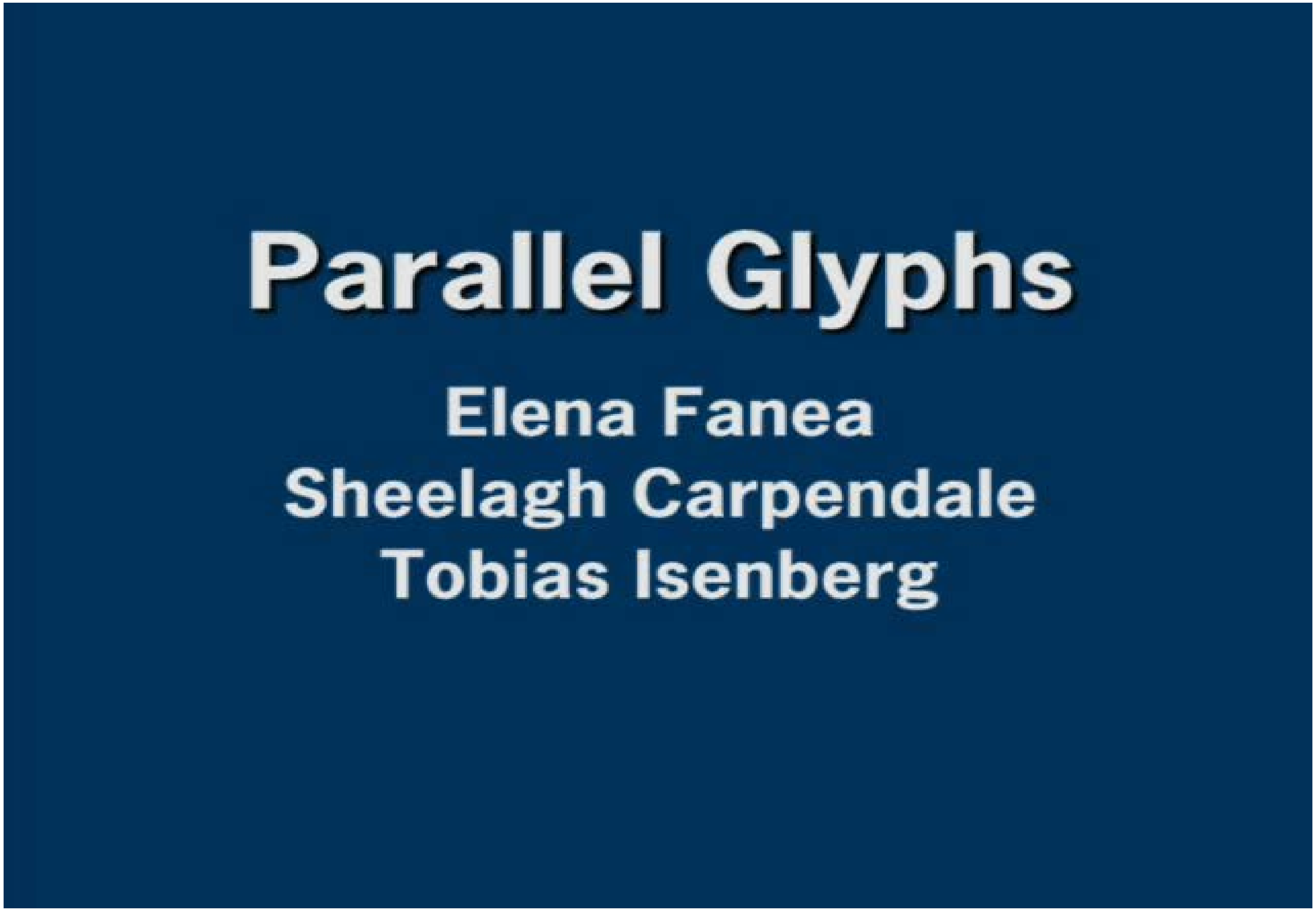
- 3-Dimensional Parallel Coordinates Display
- Parallel Glyphs
- Hierarchical Parallel Coordinates

3-Dimensional Parallel Coordinates Display



Johansson et al., 3-Dimensional Display for Clustered Multi-Relational Parallel Coordinates, IEEE 2005

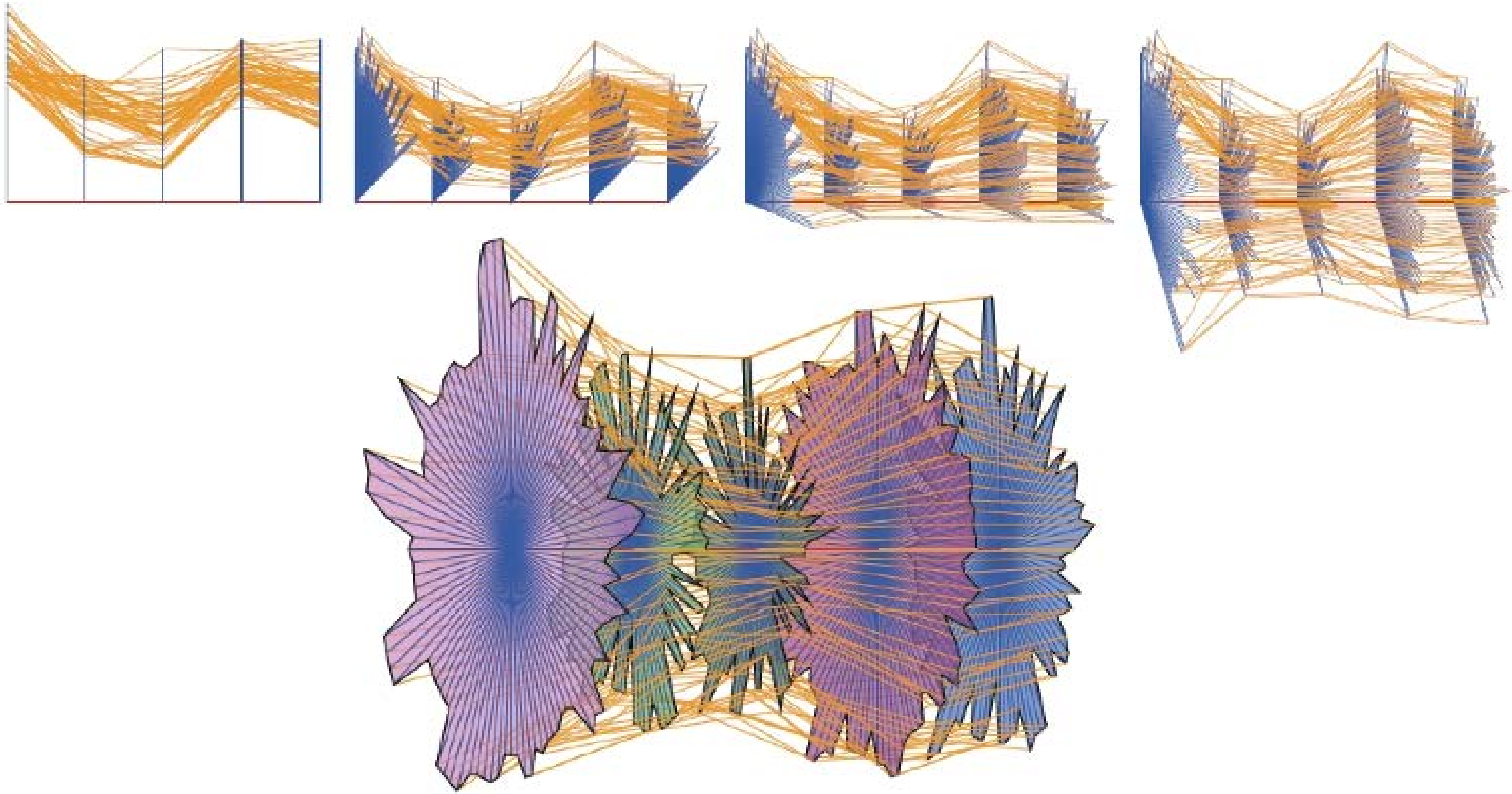
Parallel Glyphs



Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg (2005) An Interactive 3D Integration of Parallel Coordinates and Star Glyphs InfoVis 2005, October 23–25, 2005, Minneapolis, Minnesota, USA) Los Alamitos, CA. IEEE Computer Society, pages 149–156, 2005.

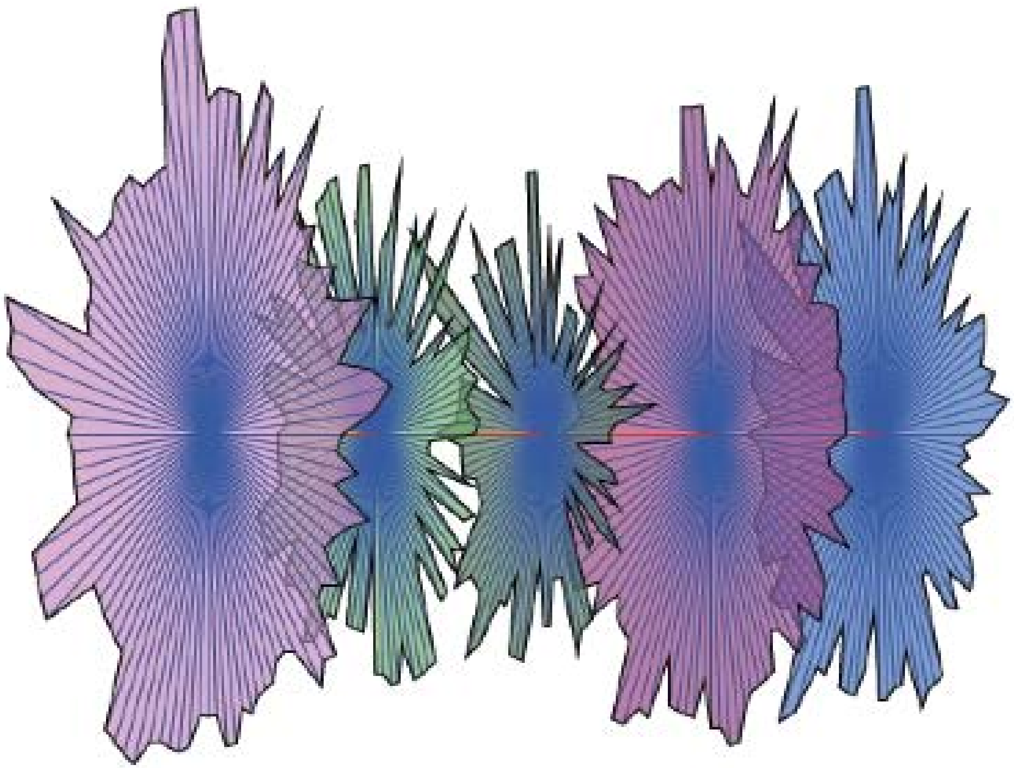
http://www.cs.rug.nl/~isenberg/personal/videos/Fanea_2005_I3I.wmv

Parallel Glyphs

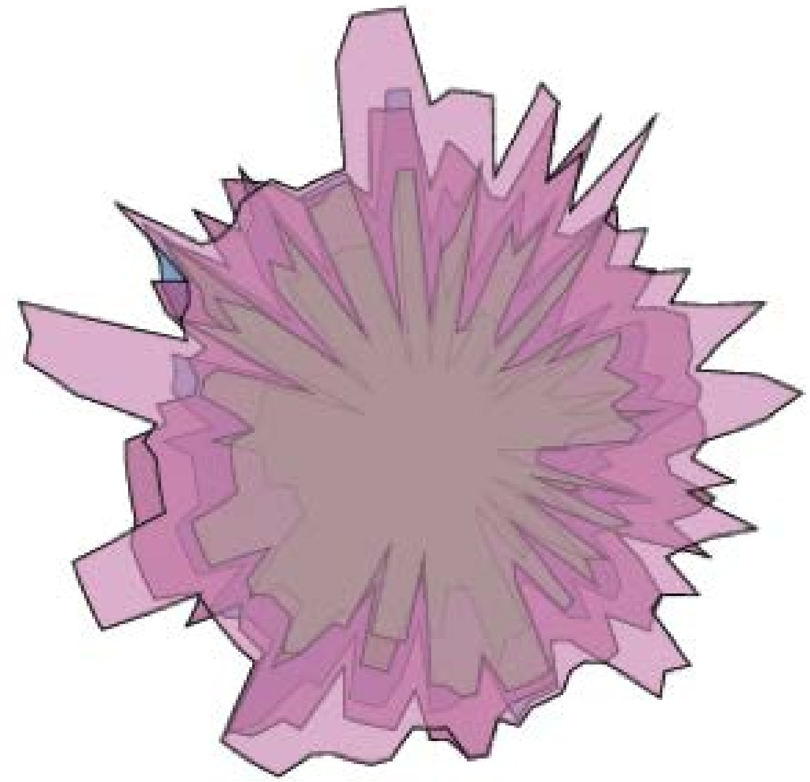


Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg (2005) An Interactive 3D Integration of Parallel Coordinates and Star Glyphs InfoVis 2005, October 23–25, 2005, Minneapolis, Minnesota, USA) Los Alamitos, CA. IEEE Computer Society, pages 149–156, 2005.

Parallel Glyphs



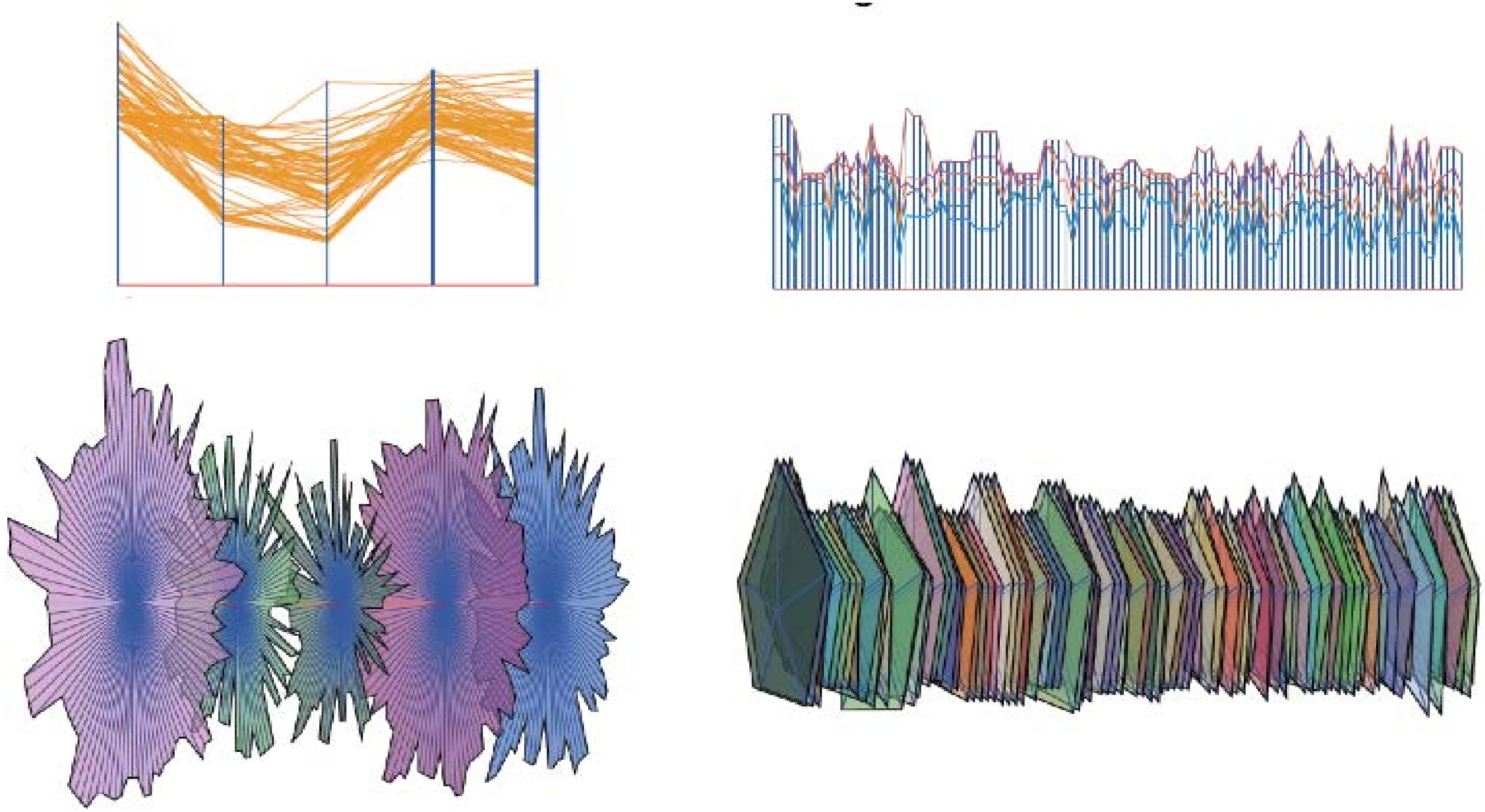
Compare data records
Visual patterns depends
highly on order of data items



Compare dimensions (attributes)
Similar to radar plot

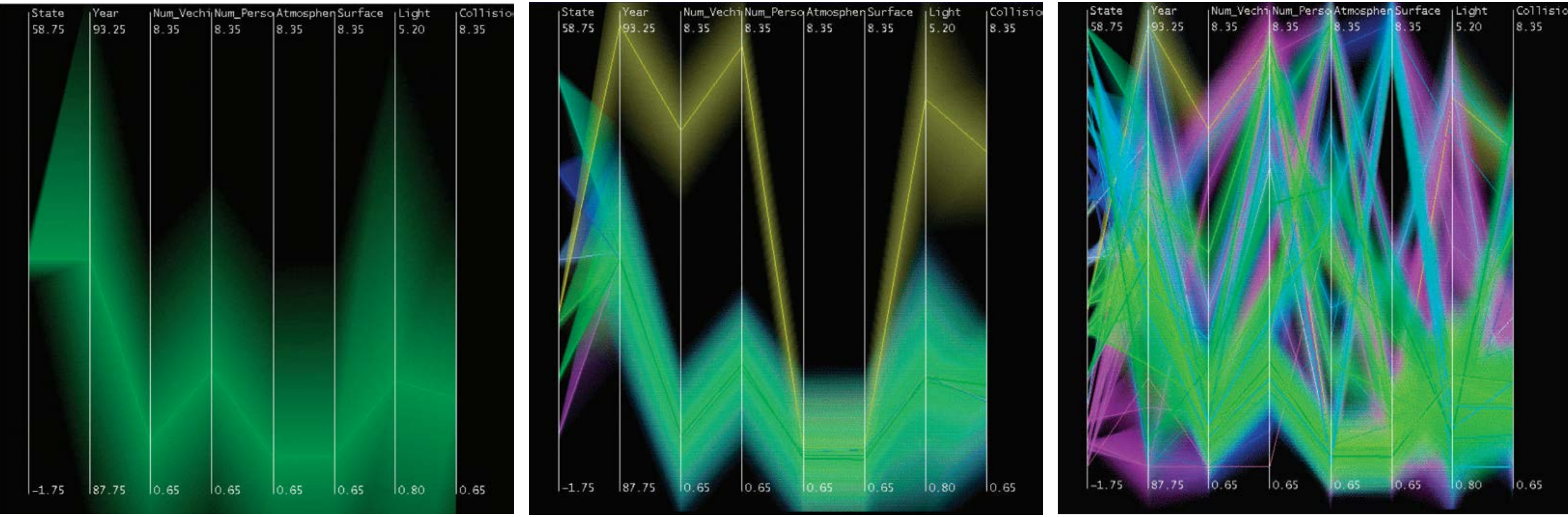
Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg (2005) An Interactive 3D Integration of Parallel Coordinates and Star Glyphs InfoVis 2005, October 23–25, 2005, Minneapolis, Minnesota, USA) Los Alamitos, CA. IEEE Computer Society, pages 149–156, 2005.

Parallel Glyphs



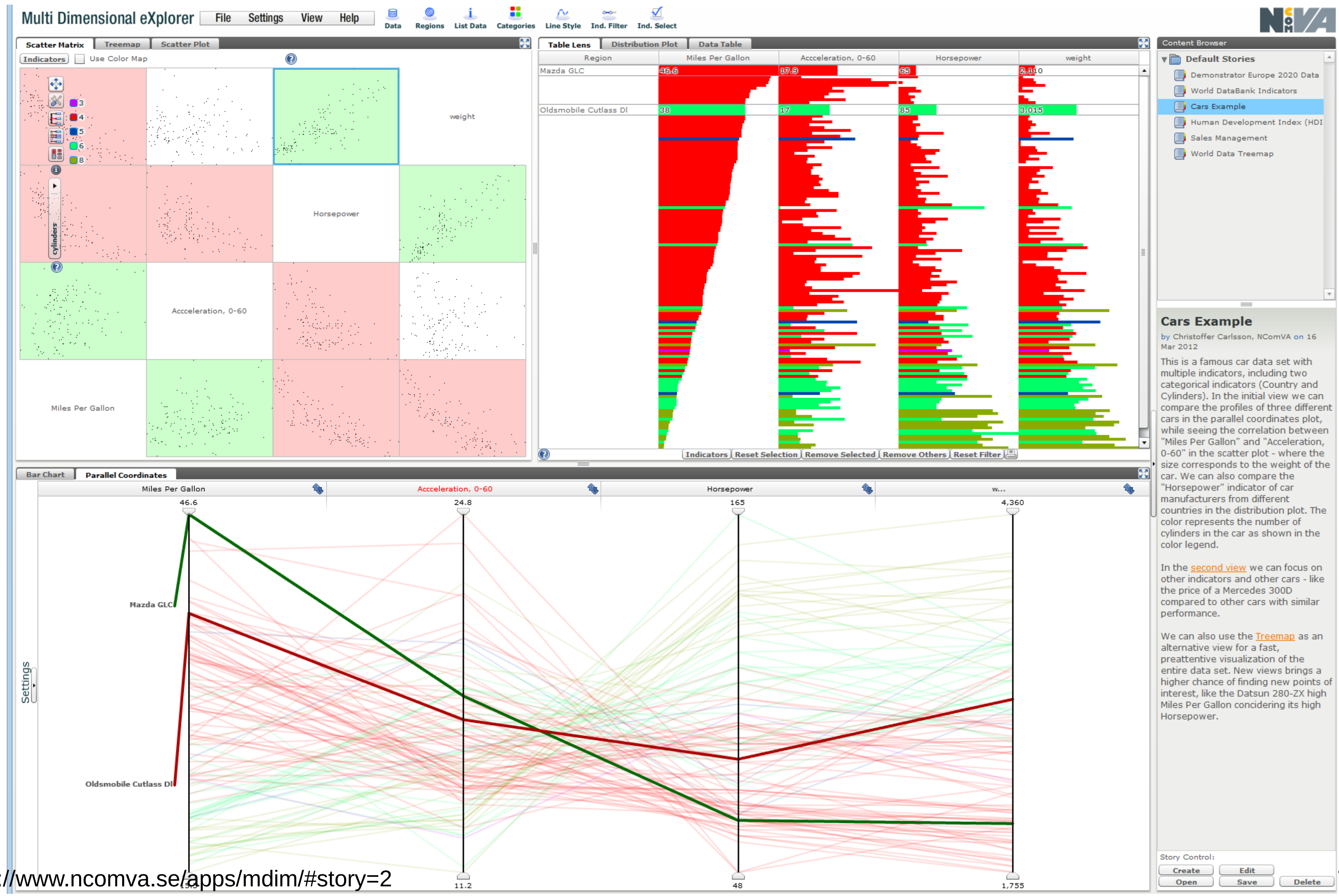
Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg (2005) An Interactive 3D Integration of Parallel Coordinates and Star Glyphs InfoVis 2005, October 23–25, 2005, Minneapolis, Minnesota, USA) Los Alamitos, CA. IEEE Computer Society, pages 149–156, 2005.

Hierarchical Parallel Coordinates



- Based on hierarchical clustering [Fua et al. 1999]
- Different cuts through the cluster hierarchy can be interactively chosen
- For each cluster a band defined by the minimum and maximum attribute value on each axis is shown as well as the center line

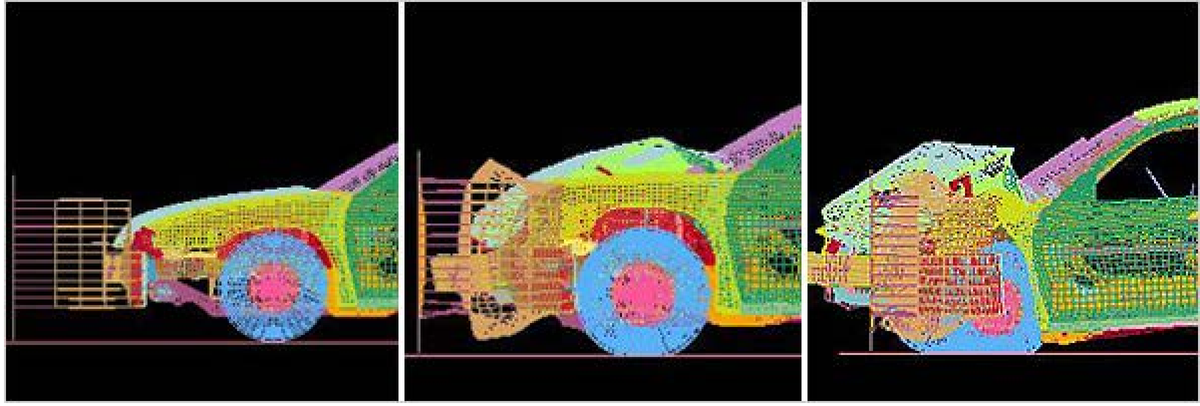
Coordinated Multi-View Display



http://www.ncomva.se/apps/mdim/#story=2

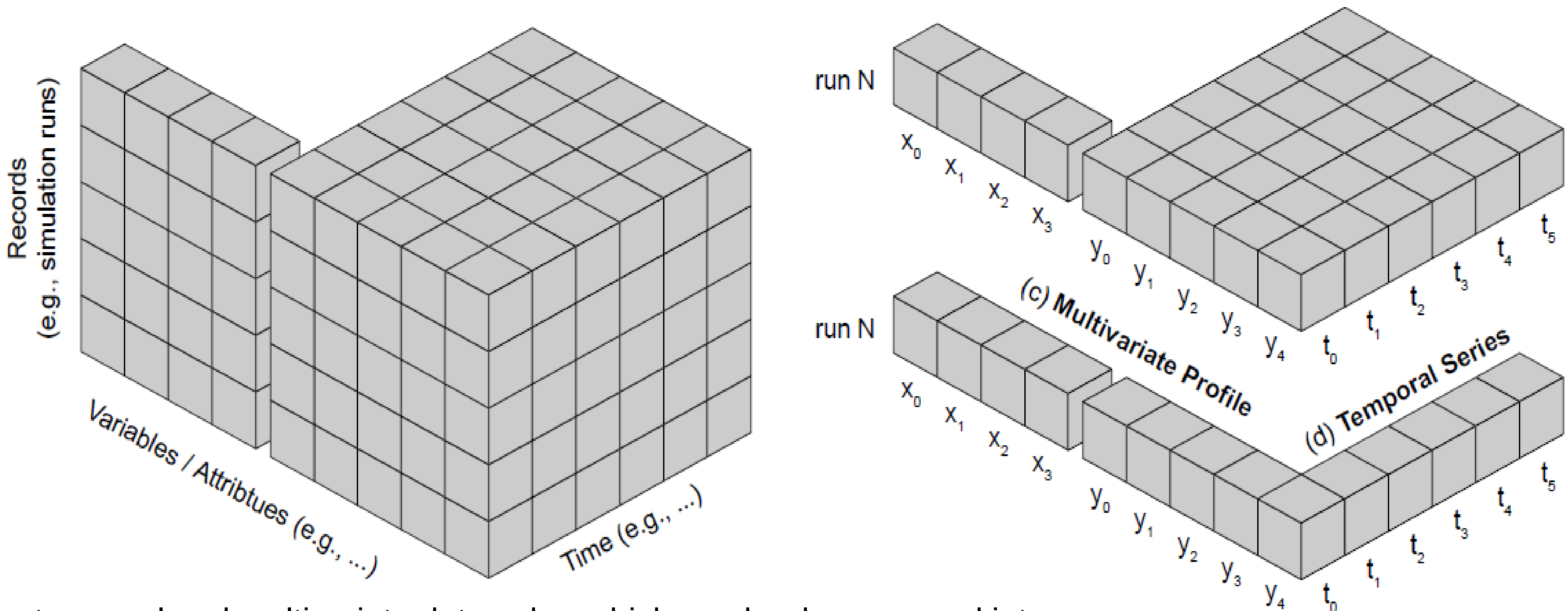
Simulation Data

- Two keys
  - Simulation run #
  - Time
- Many dependent attributes



- (a) **Space-Time-Attribute Cube**

(b) **Time-Attribute-Slice (e.g., for simulation run n)**



(a) The temporal and multivariate data cube, which can be decomposed into
(b) a set of time-attribute slices (one for each simulation run), or
(c) a set of multivariate profiles (one for each simulation run/time step combination), or
(d) a set of time series (one for each simulation run/attribute combination).

Simulation Explorer: Coordinated Multi-View Display

End

Visualization

Categorical Data

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

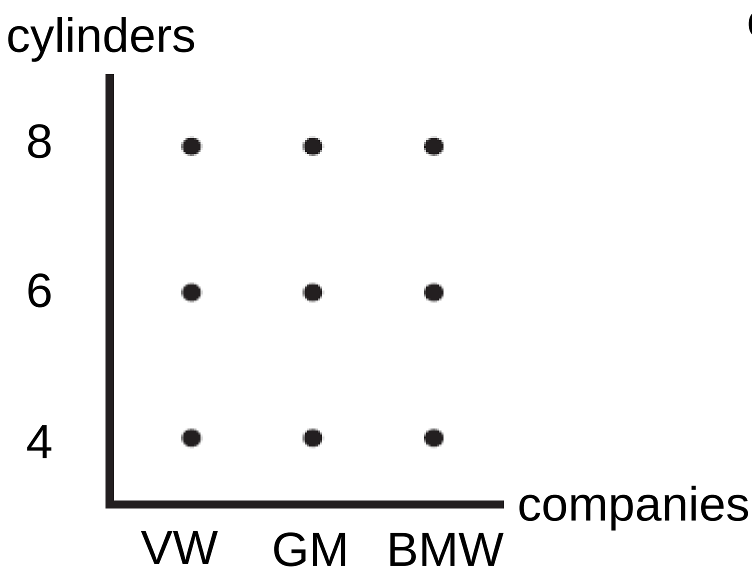
Acknowledgements

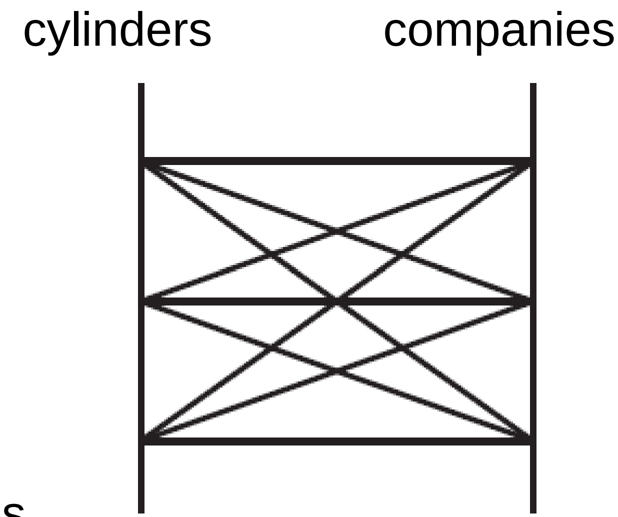
- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Daniel A. Keim, Universität Konstanz
- Heer, J., Bostock, M., and Ogievetsky, V. 2010. A tour through the visualization zoo. Commun. ACM 53, 6 (Jun. 2010), 59-67.
- http://portal.acm.org/citation.cfm?id=1743546.1743567

Categorical Data

- Treemaps
  - Shneiderman, Ben, and Martin Wattenberg. 2001. “Ordered treemap layouts.” In Proceedings of the IEEE Symposium on Information Visualization, 73–78.
- Extended parallel coordinates
  - Product Explorer ….
- Mosaic plots
  - Theus, Martin. 2002. “Interactive data visualization using Mondrian.” Journal of Statistical Software 7, no. 11: 1–9. http://www.theusrus.de/Mondrian/.
- Parallel sets
  - Bendix, Fabian, Robert Kosara, and Helwig Hauser. 2005. “Parallel Sets: Visual analysis of categorical data.” In Proceedings of the IEEE Symposium on Information Visualization,133–140.
  - Robert Korsara. 2010. Turning a Table into a Tree: Growing Parallel Sets into a Purposeful Project

Categorical Data

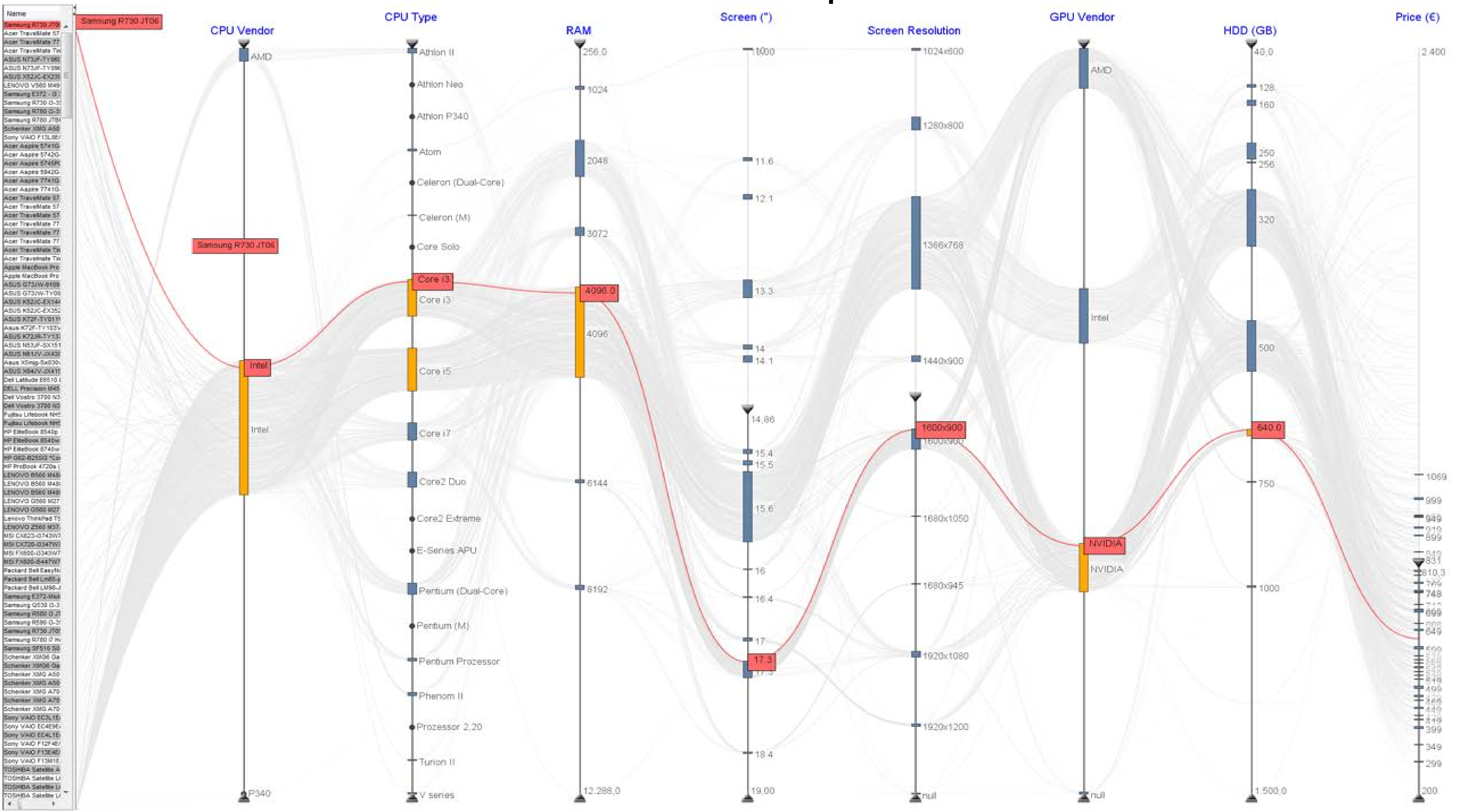




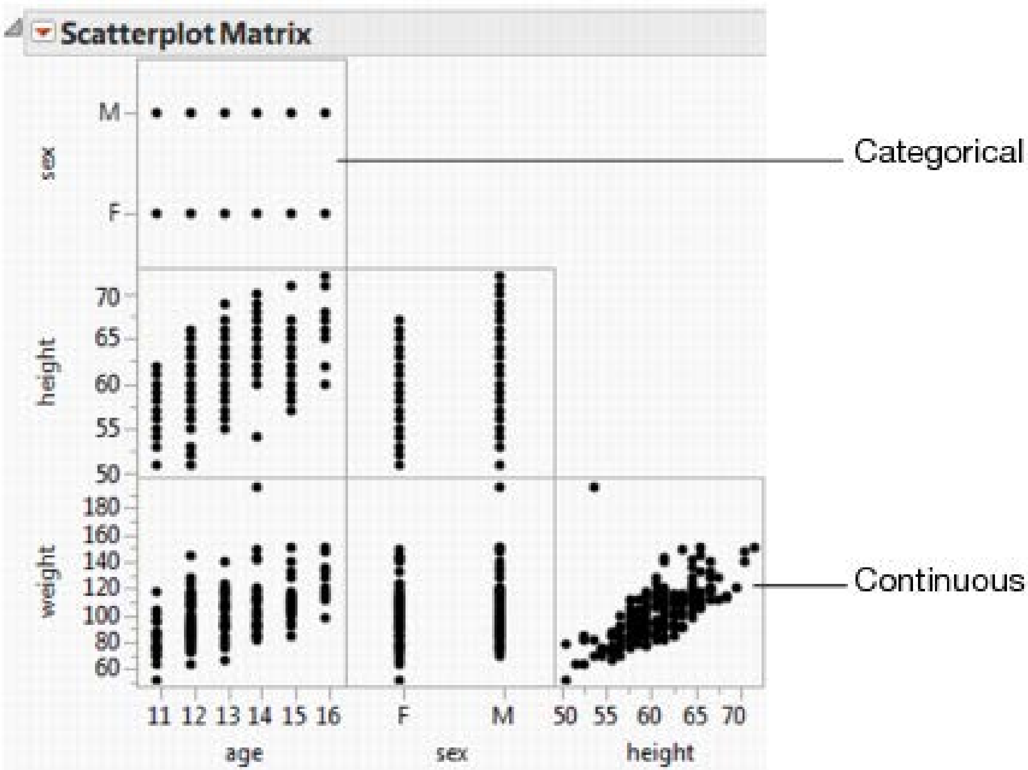
- Scatterplots and simple parallel coordinates don't work due to massive overplotting – here only 3 x 3 categories and 2 dimensions. The same is true for ordinal or even continuous data if only a few attribute values are assumed or data is quantized

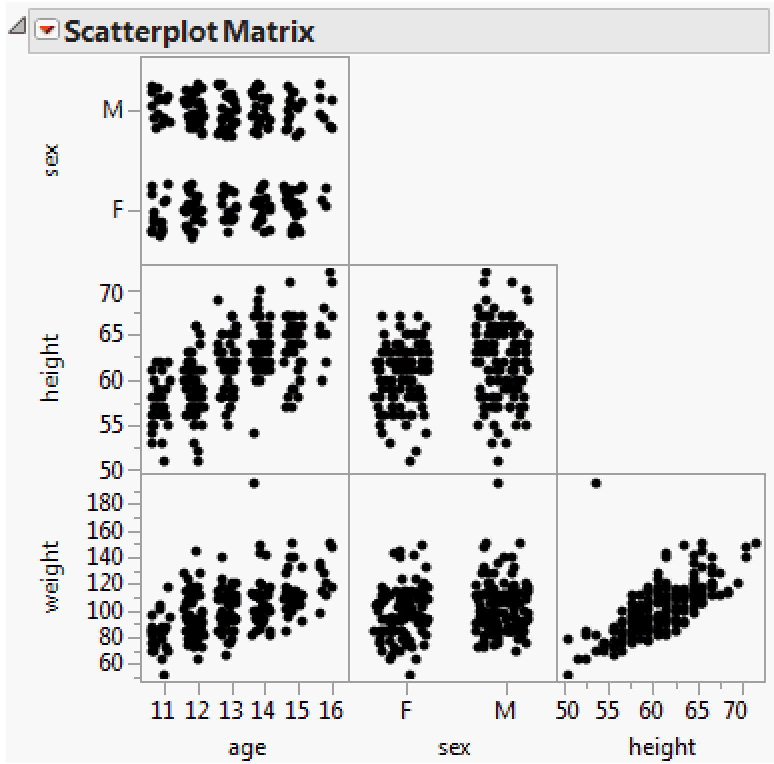
Parallel Coordinates for Categorical Data

The Product Explorer



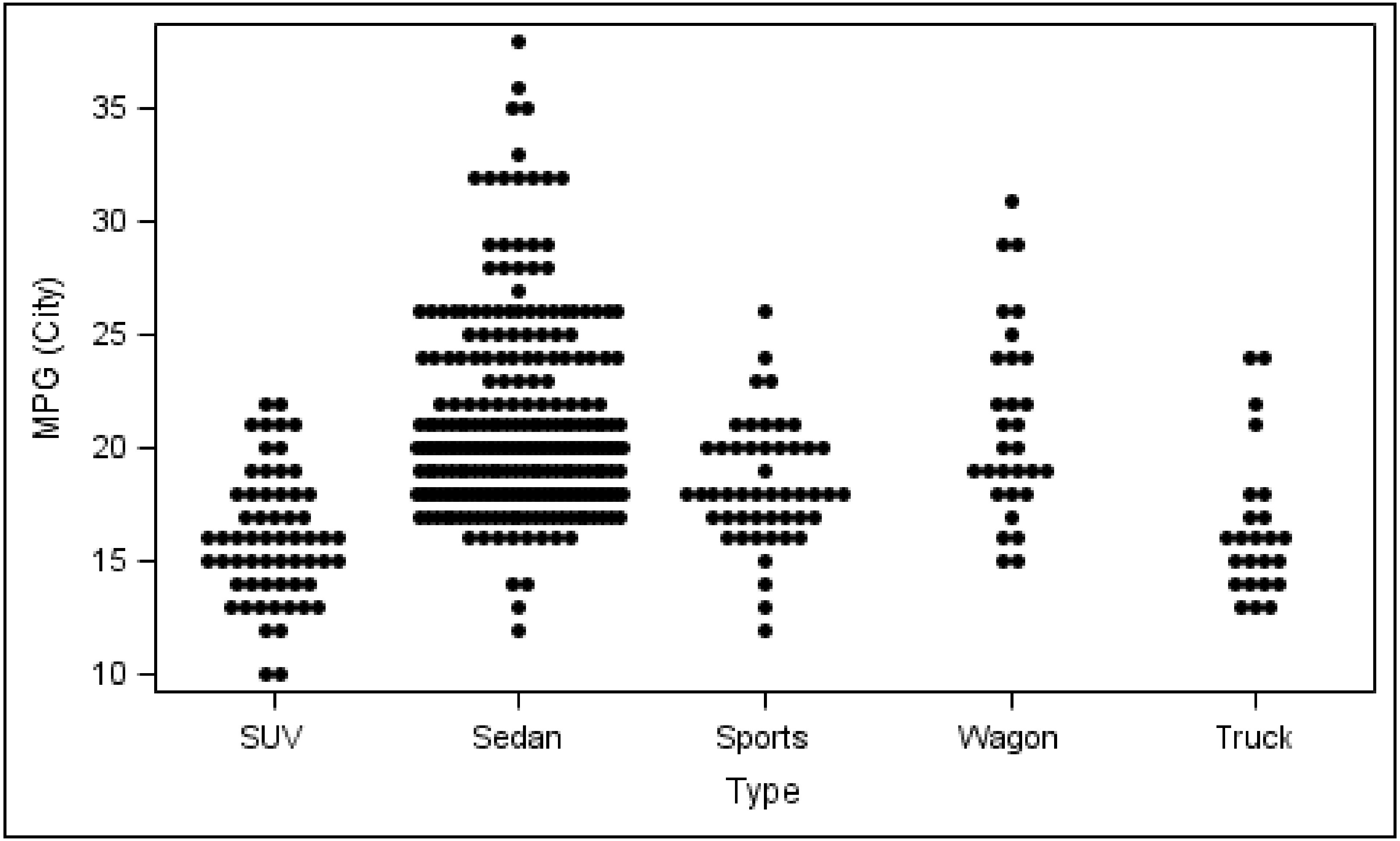
Jittered Scatter Plots





http://www.jmp.com/support/help/Example_of_a_Scatterplot_Matrix.shtml

Jittered Scatter Plots



Categorical Attribute Values as Keys

Attribute	Values
Class	Crew, First, Second, Third
Sex	Male, Female
Age	Adult, Child
Survived	Yes, No

Example: Titanic data set
(people on board of the Titanic)

Adults	*Survivors*		*Non-Survivors*	
	Male	*Female*	*Male*	*Female*
1st Class	57	140	118	4
2nd Class	14	80	154	13
3rd Class	75	76	387	89
Crew	192	20	670	3

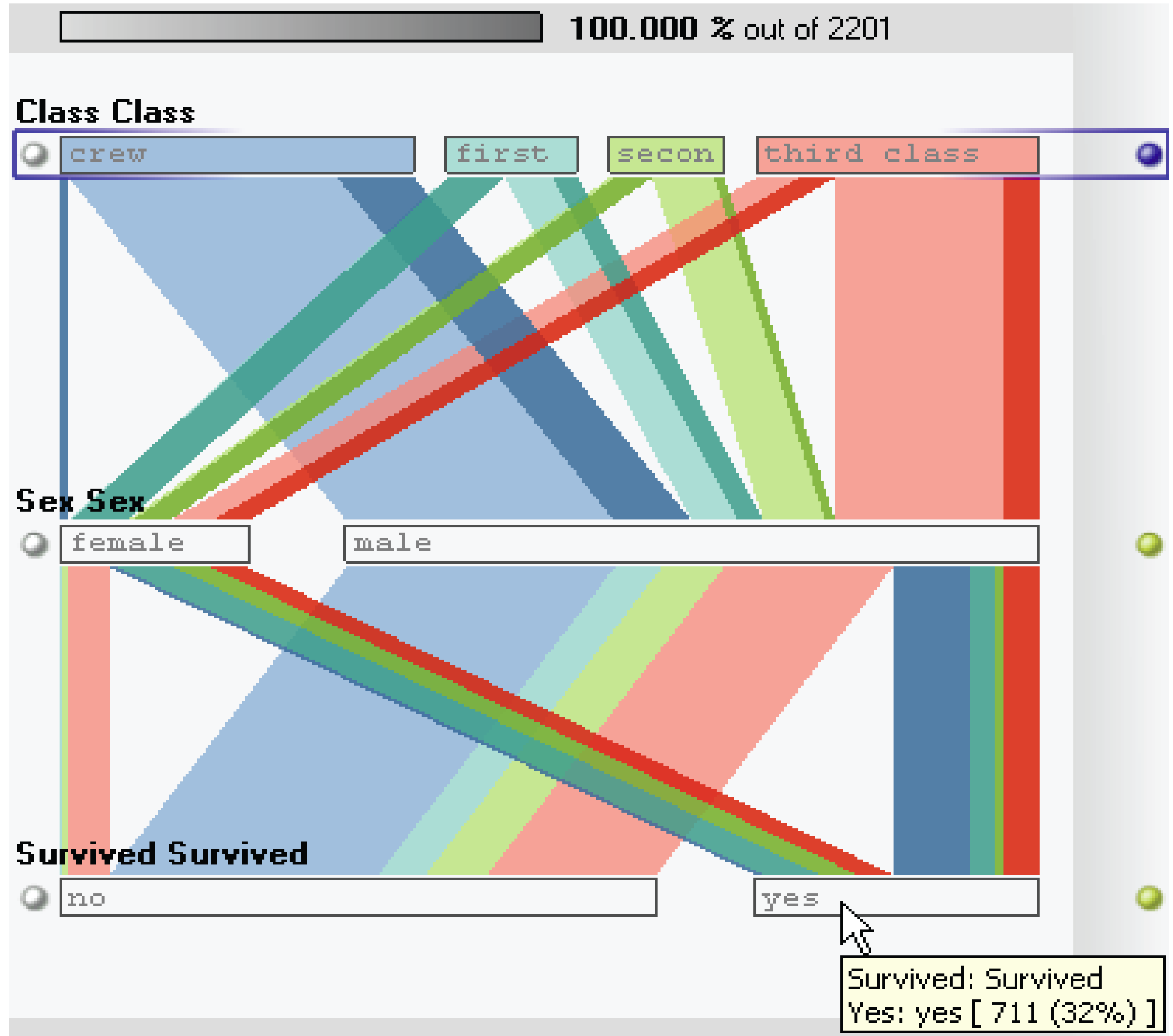
Children	*Survivors*		*Non-Survivors*	
	Male	*Female*	*Male*	*Female*
1st Class	5	1	0	0
2nd Class	11	13	0	0
3rd Class	13	14	35	17
Crew	0	0	0	0

- Often we have aggregated information
- A data item represents no longer e.g. an individual person
- Categorical attribute values serve as keys
  - [Class, Sex, Age, Survived] -> number of people

Parallel Sets

Original Design

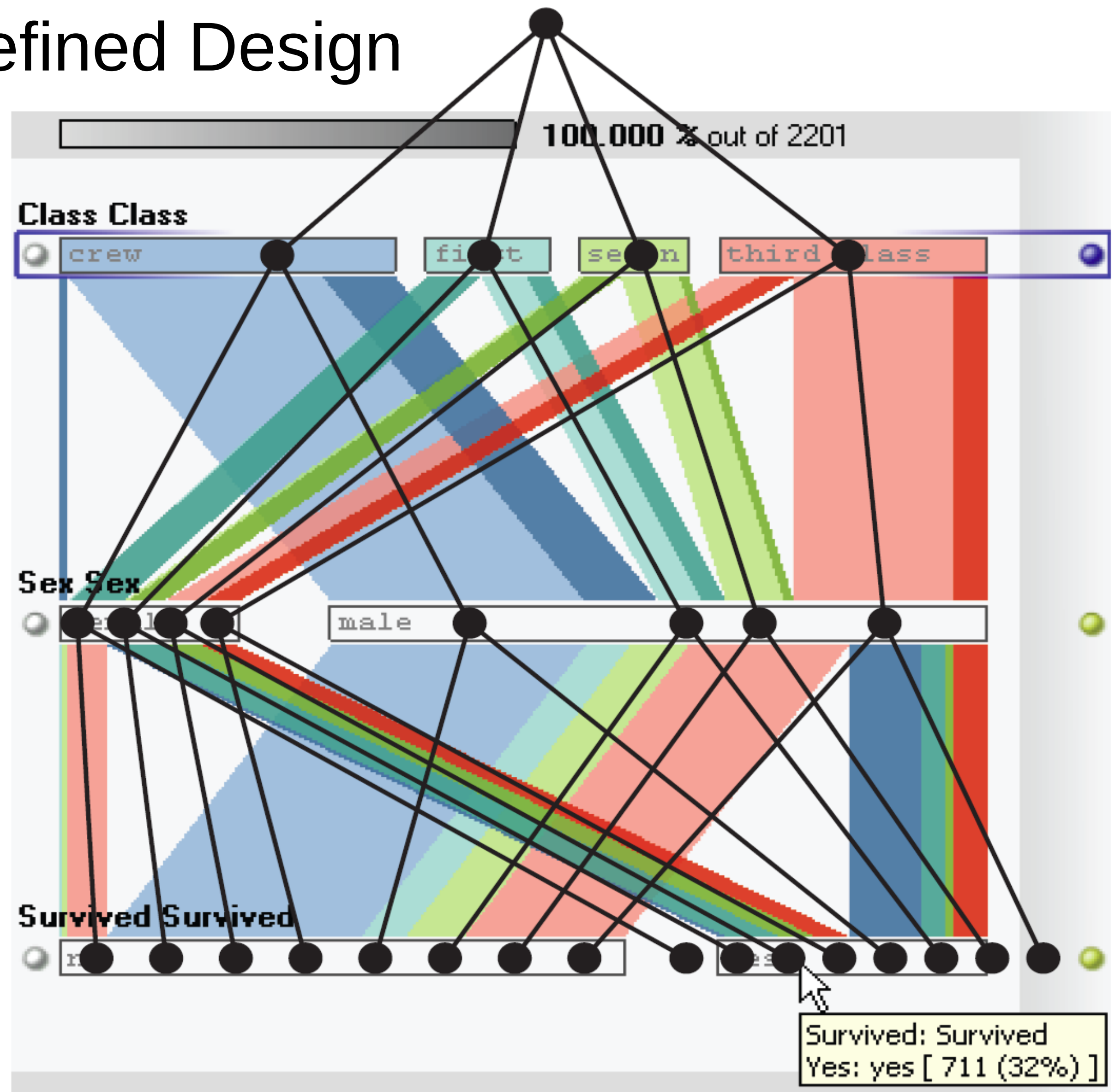
- Horizontal bars represent numerical values / fractions
- Ribbons connect categories that occur together
- Horizontal width of ribbons represent numerical values / fractions
- Colors encode cases
- Shows distributions on multiple axes simultaneously



Parallel Sets

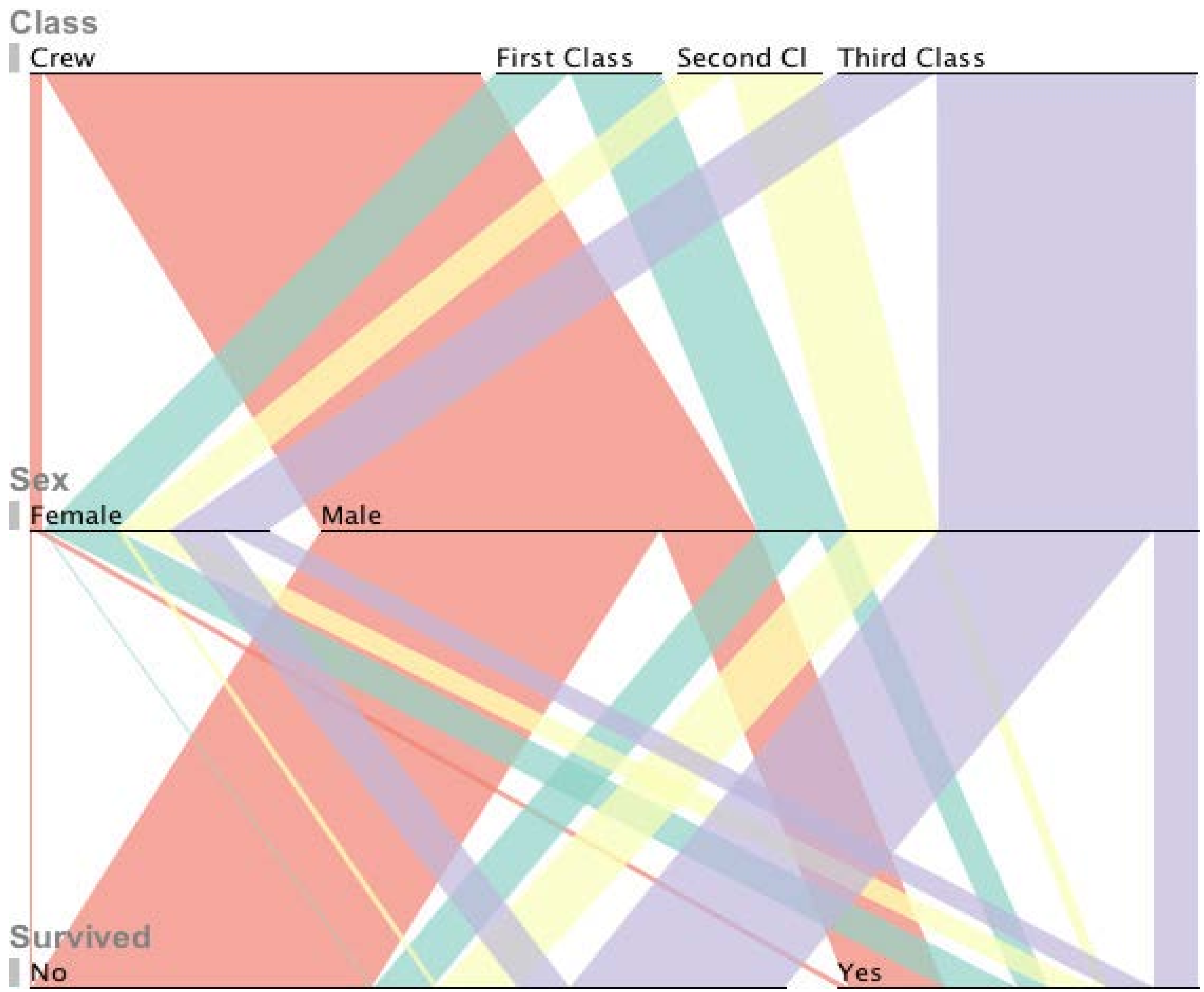
Refined Design

- Easier to follow ribbons
- Reveals the tree structure



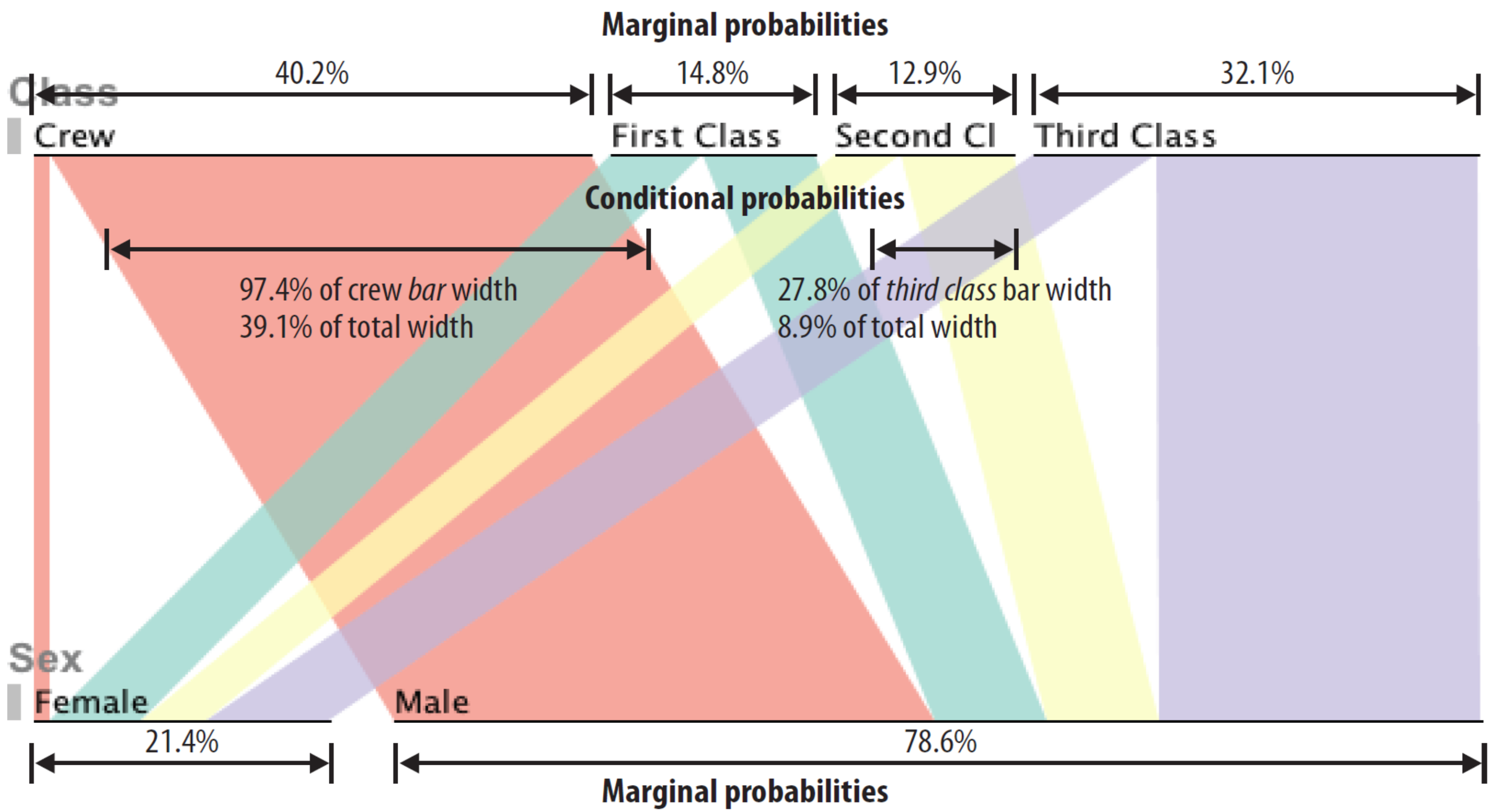
Parallel Sets
Refined Design

- Easier to follow ribbons
- Reveals the tree structure



Parallel Sets

Refined Design

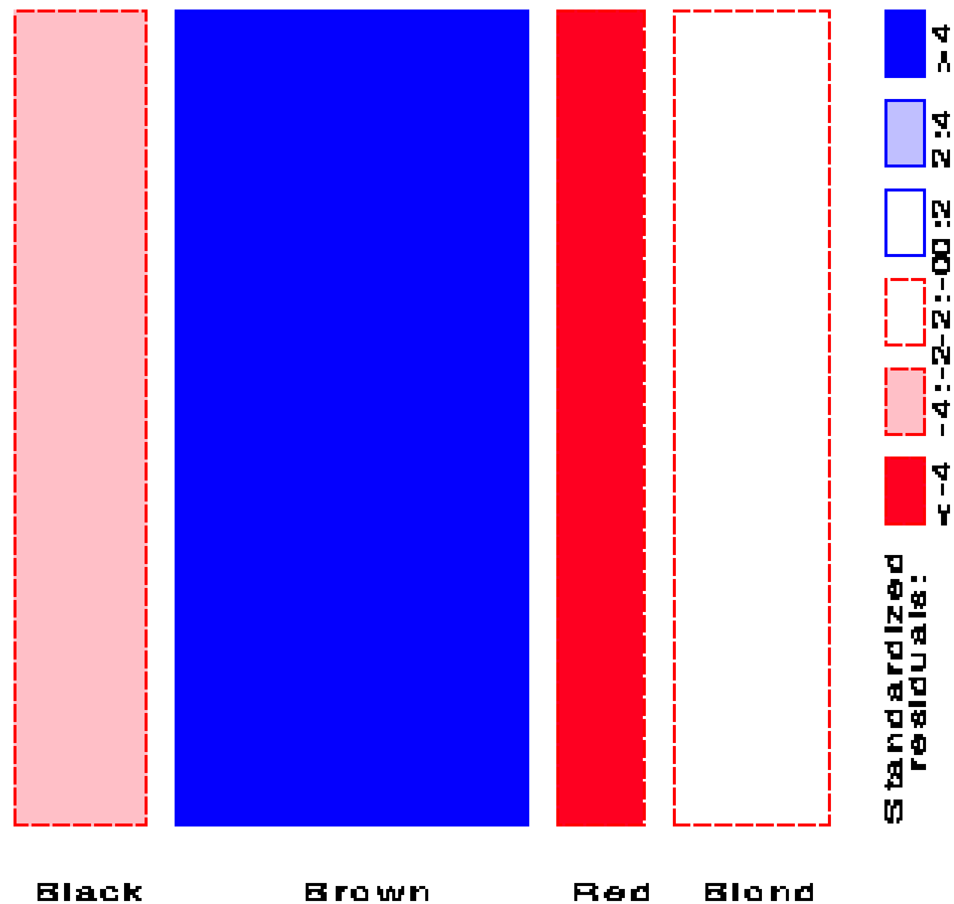


The width of each ribbon represents its marginal probability (proportional fraction) of the total data set, and also its conditional probability within each category

Mosaic Plots

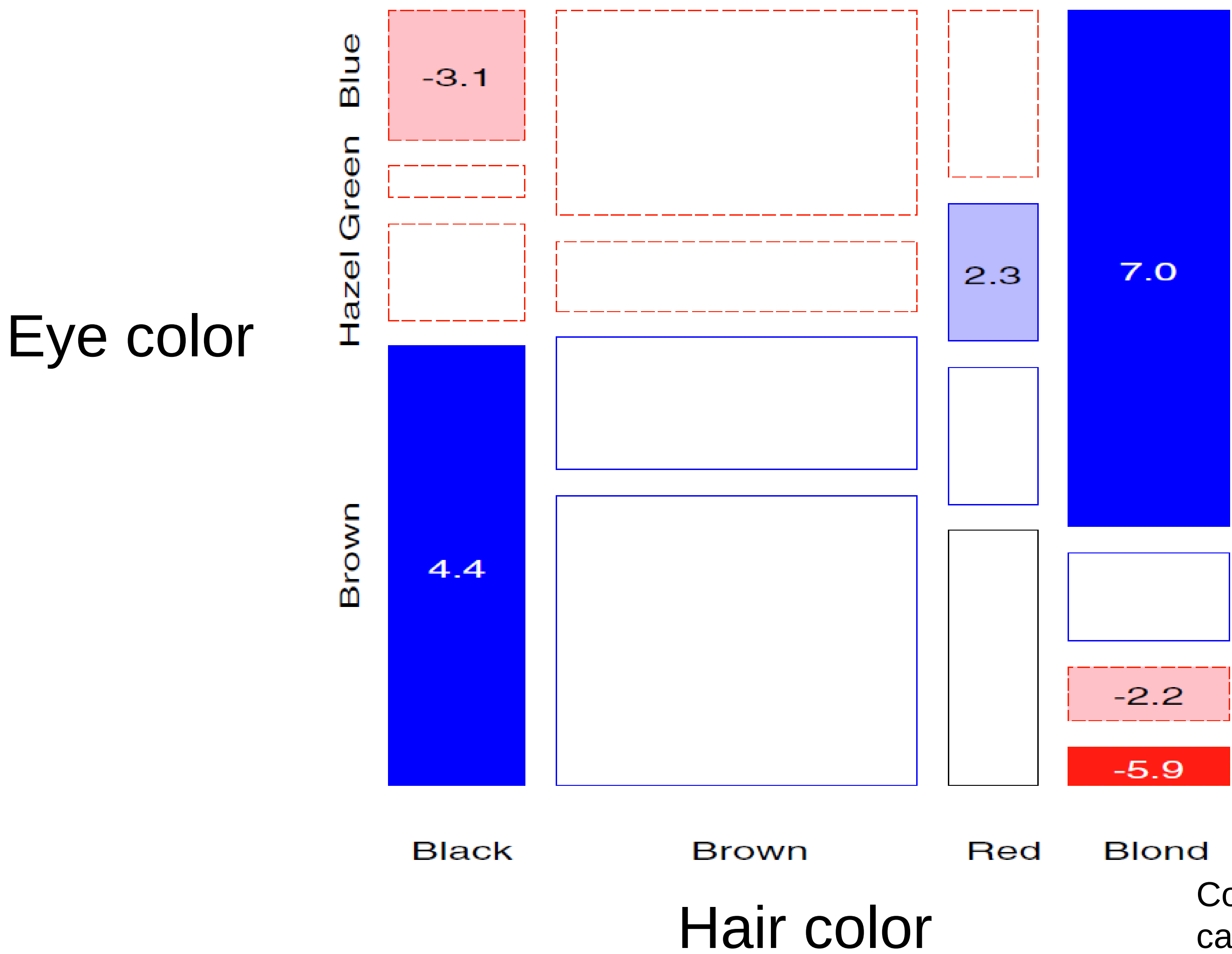
Hartigan and Kleiner (1981)

Eye Color	Hair Color: BLACK	BROWN	RED	BLOND	Total
Brown	68	119	26	7	220
Blue	20	84	17	94	215
Hazel	15	54	14	10	93
Green	5	29	14	16	64
Total	108	286	71	127	592



http://www.datavis.ca/papers/casm/casm.pdf

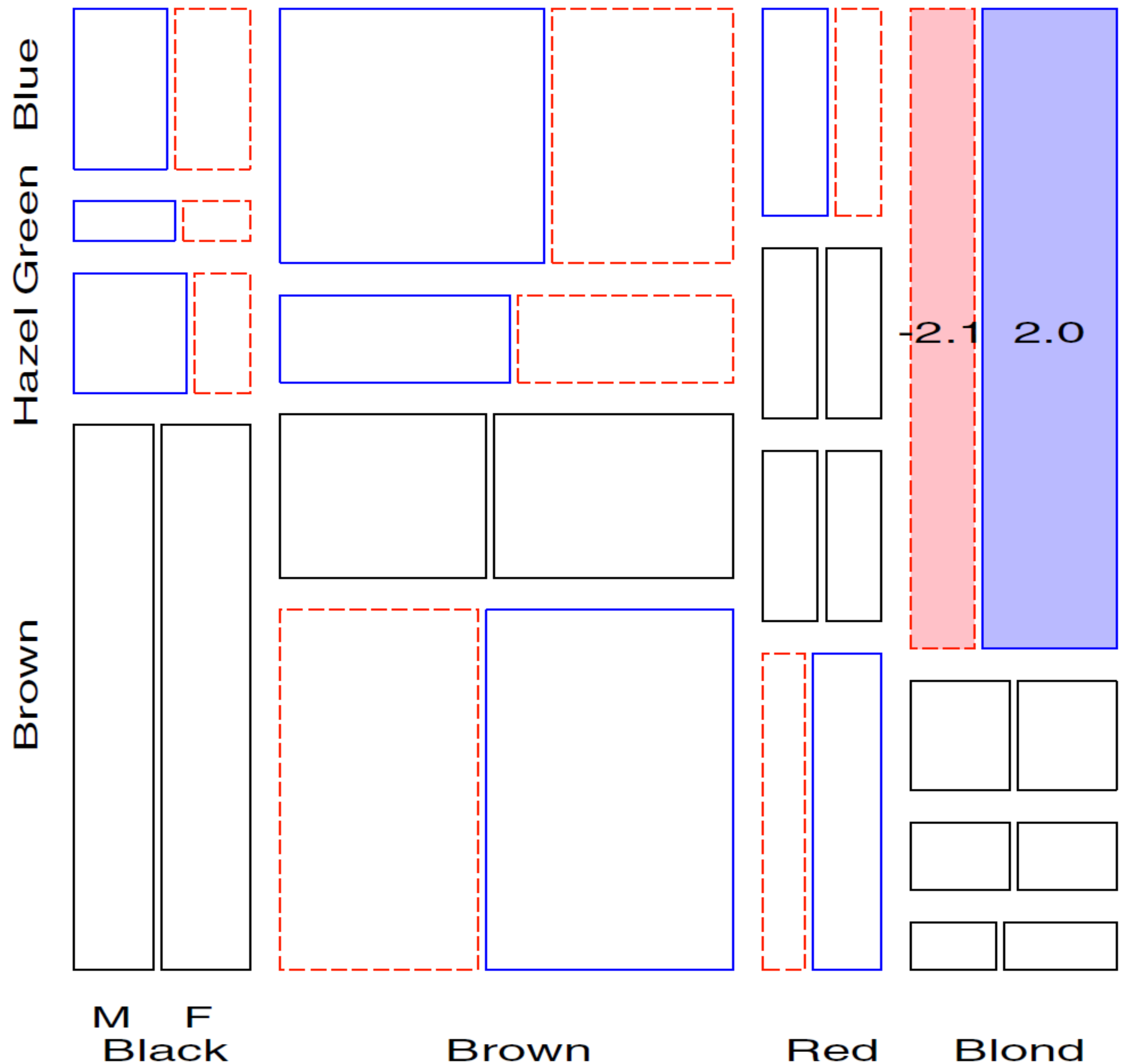
Mosaic Plots
Hartigan and Kleiner (1981)



Colors and numbers can show some other variable – here: residuals from the model of complete independence

Three-Way Mosaic Plot

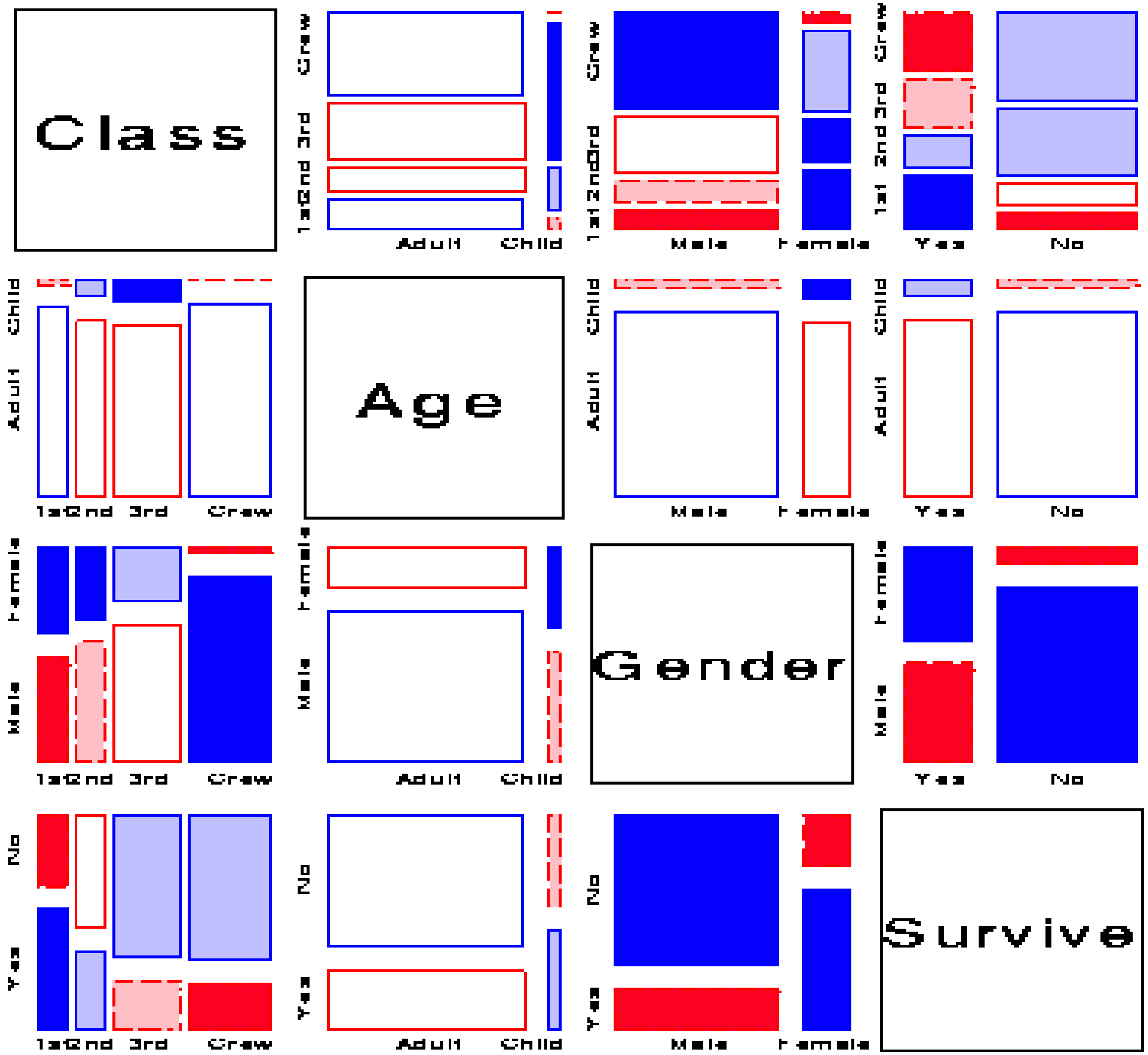
- Eye color vs. hair color vs. gender



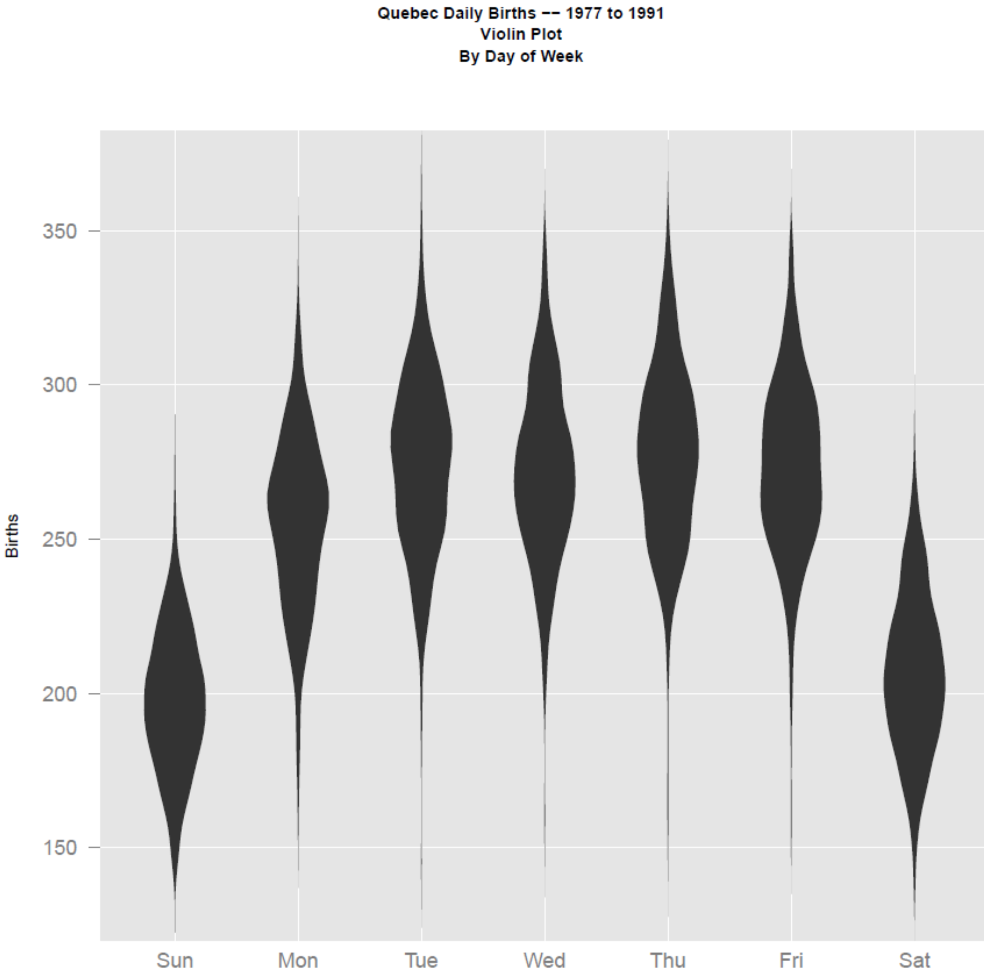
Colors and numbers show some other variable

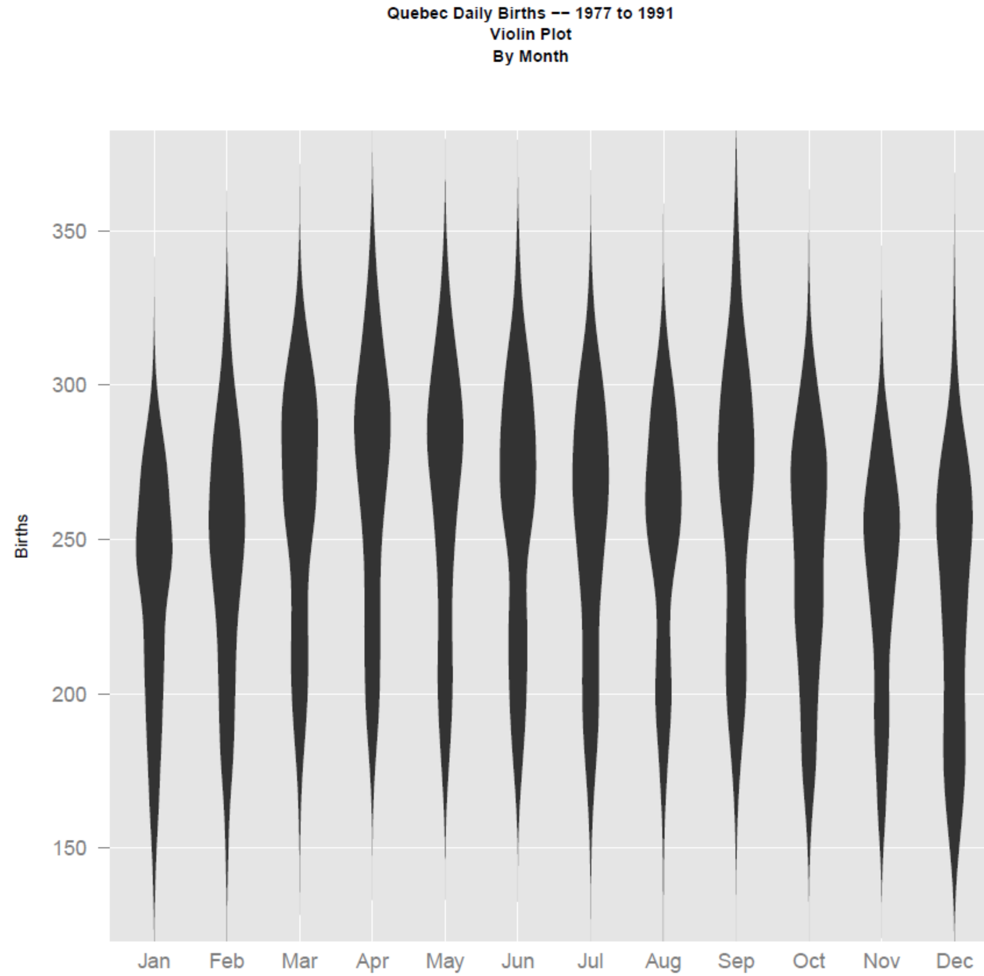
Mosaic Matrix Plots

Pairs of Attributes



Violin Plots
Categorical vs. Continuous Attribute



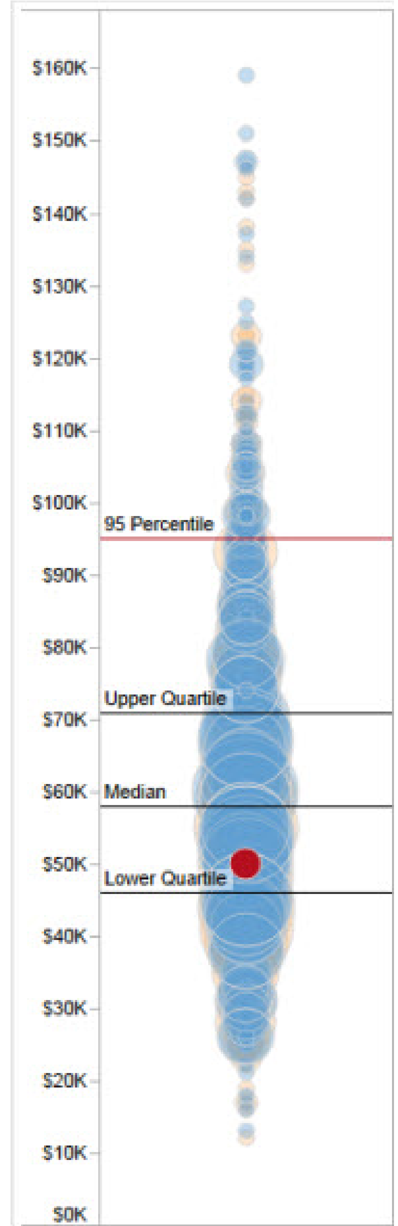


http://www.dashboardinsight.com/articles/digital-dashboards/building-dashboards/box-and-strip-plots-with-r.aspx

Categorical vs. Quantized Continuous Data

- Keys
  - Quantized salary value
  - Year (only one year available here)
- Value
  - Number of people with a particular salary

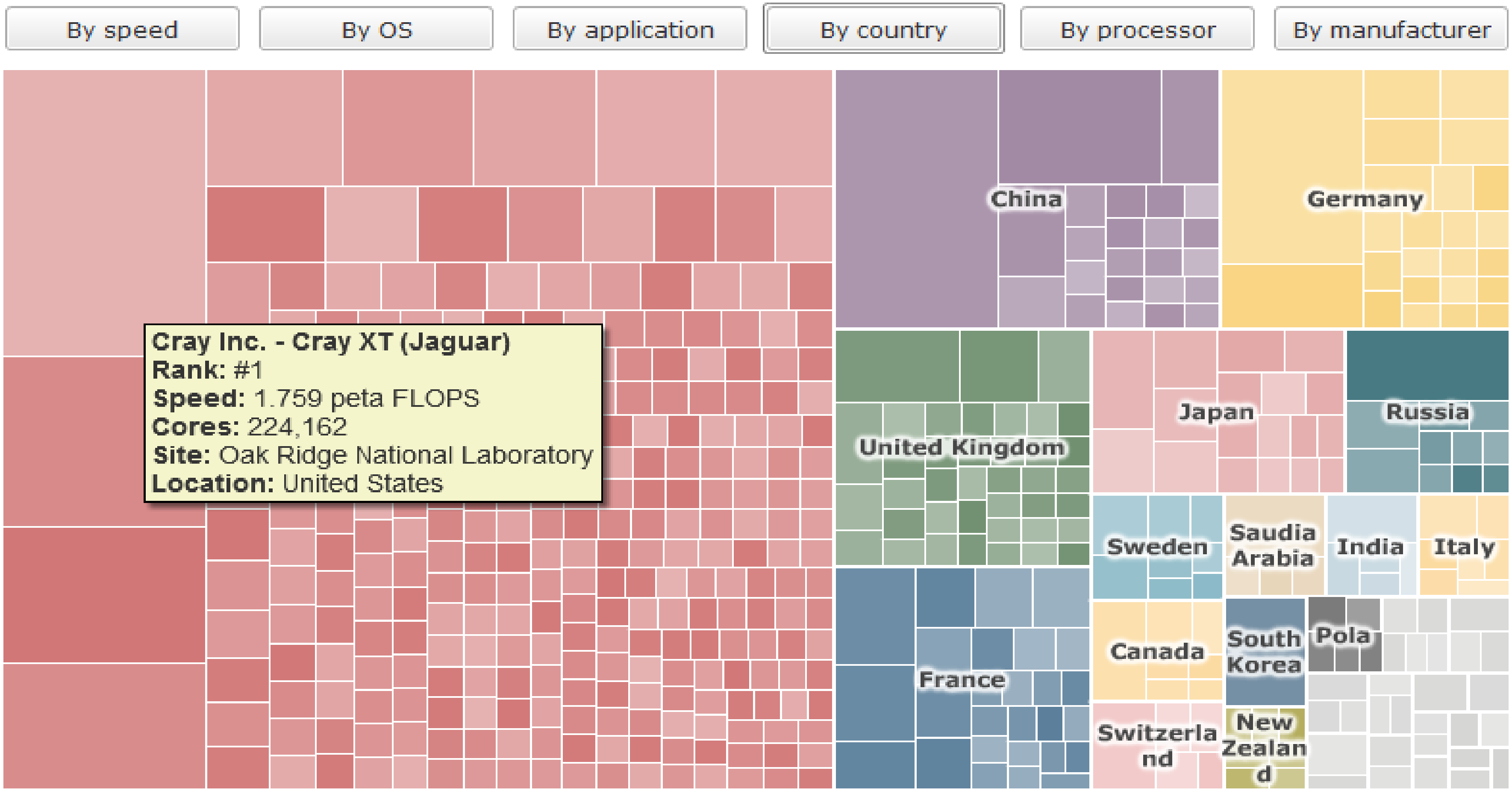
The size of the circle corresponds to number of respondents
In a survey reporting a salary close to the amount shown.
Circle colors represent gender.
The red dot might show your own salary
http://www.datarevelations.com/ive-got-the-jitters-and-i-like-it.html



2010

Treemaps for Categorical Data

Example: Super Computing Ranking



The biannual Top 500 supercomputer list has been released. Use this graphic to explore the world's fastest number crunchers or find out more about alternative supercomputer powers .

End

Visualization

Timeseries Data

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Acknowledgements

- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Daniel A. Keim, Universität Konstanz
- Heer, J., Bostock, M., and Ogievetsky, V. 2010. A tour through the visualization zoo. Commun. ACM 53, 6 (Jun. 2010), 59-67.
- http://portal.acm.org/citation.cfm?id=1743546.1743567

- http://survey.timeviz.net/
- http://www.cvast.tuwien.ac.at/TimeBench
- http://techslides.com/over-2000-d3-js-examples-and-demos/

Timeseries

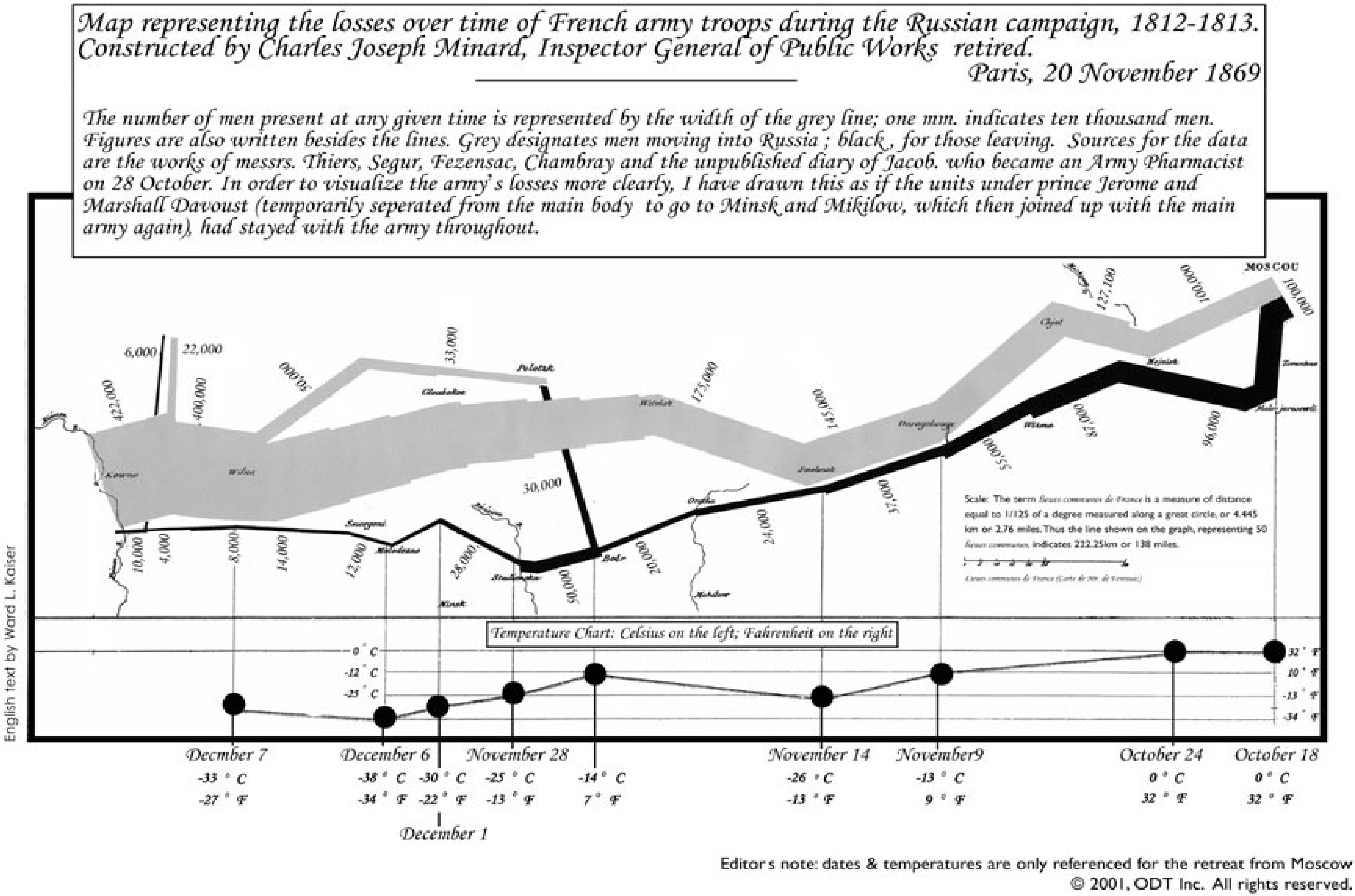
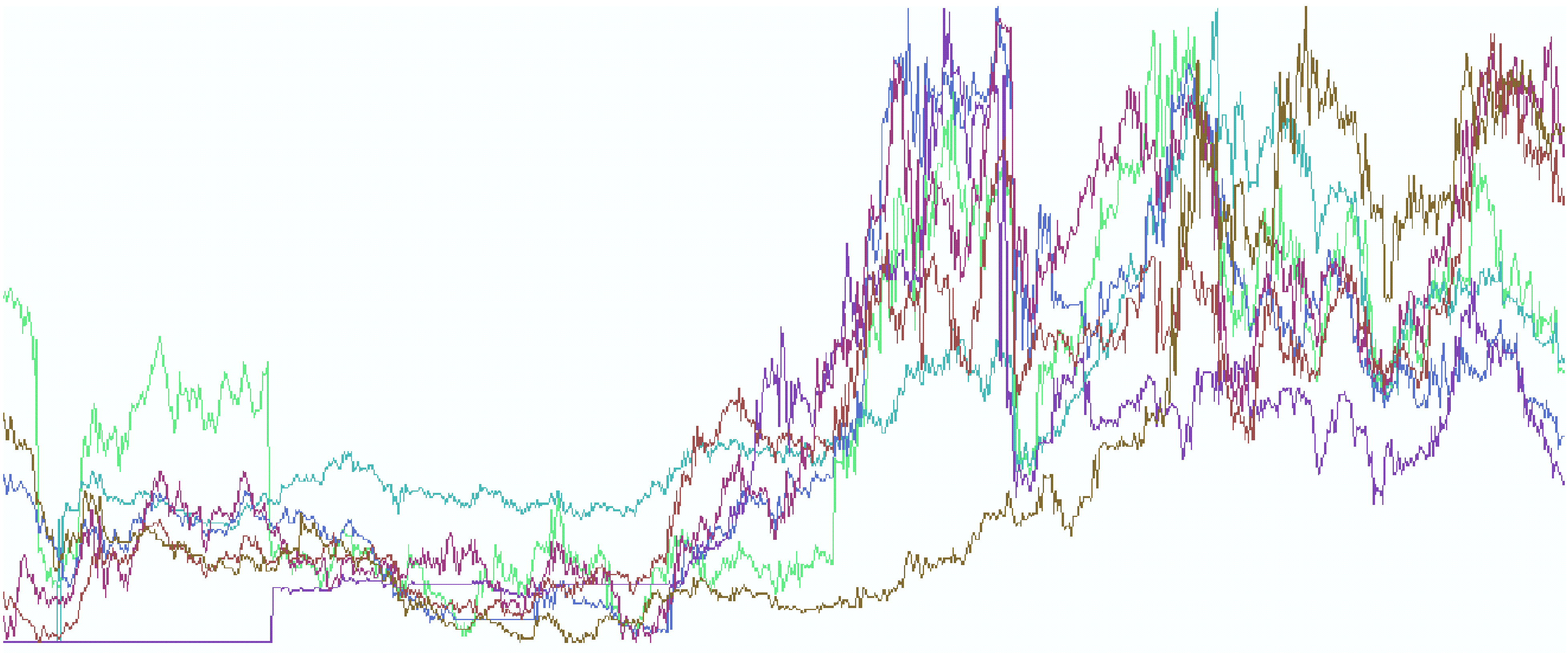


Figure 58. ***Minard s map of Napoleon s Russian campaign.***
This graphic has been translated from French to English and modified to most effectively display the temperature data.

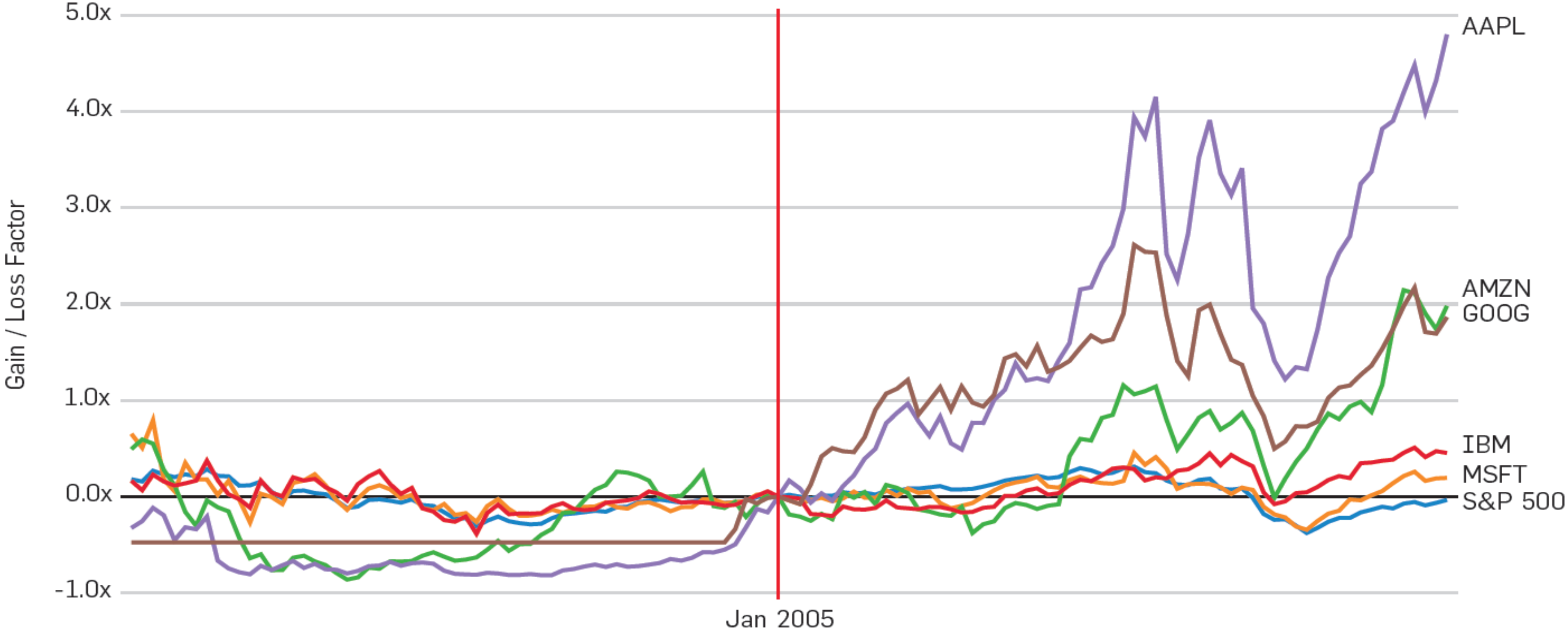
Time Series Data

- Sets of values changing over time very common
- Timevarying data examples
  - Finance (stock prices, exchange rates)
  - Science (temperatures, pollution levels, electric potentials)
  - Public policy (crime rates)
- How to compare and relate a large number of time series?

Standard Time Series Plot of 7 Stocks

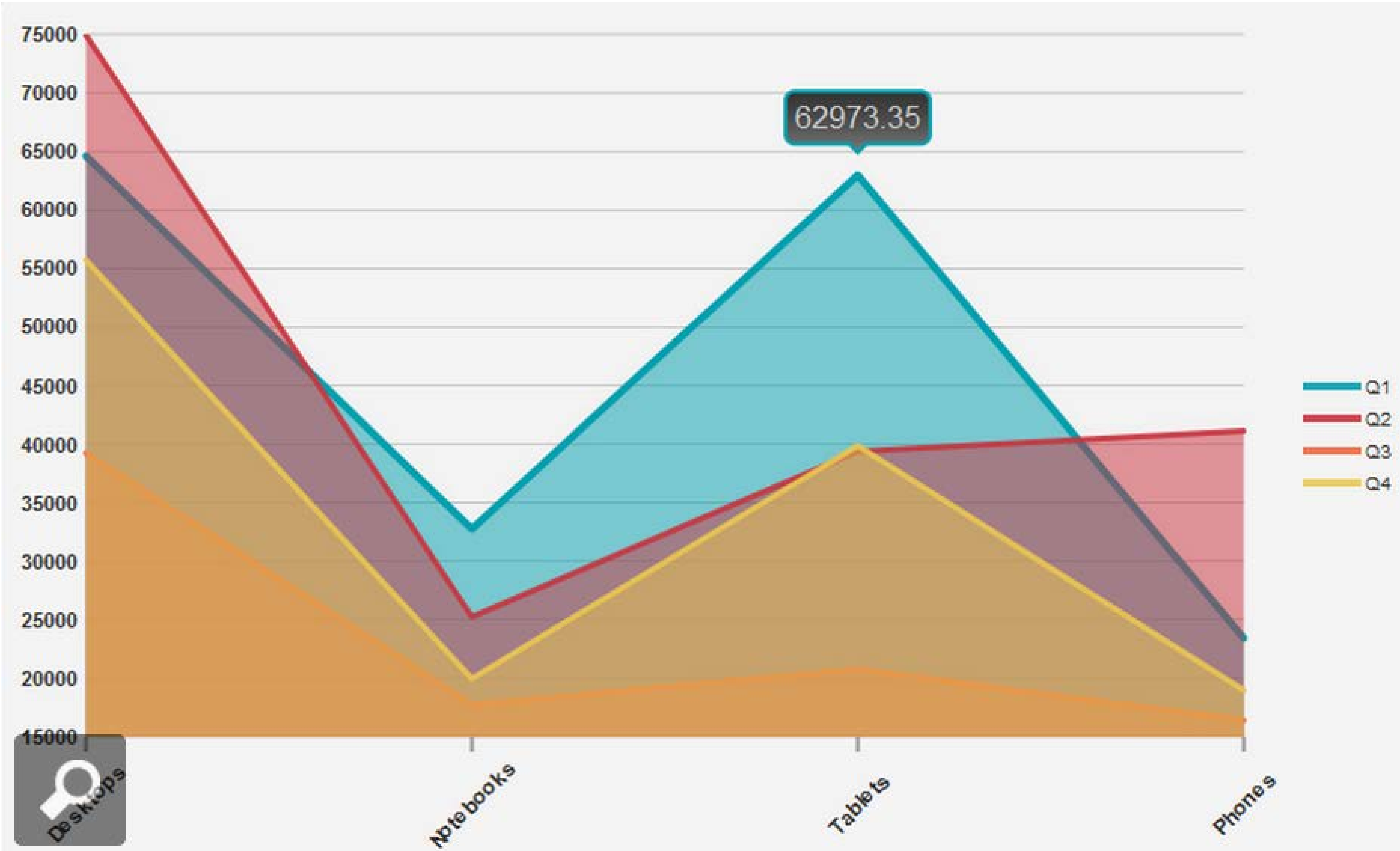


Index Charts
Show relative information



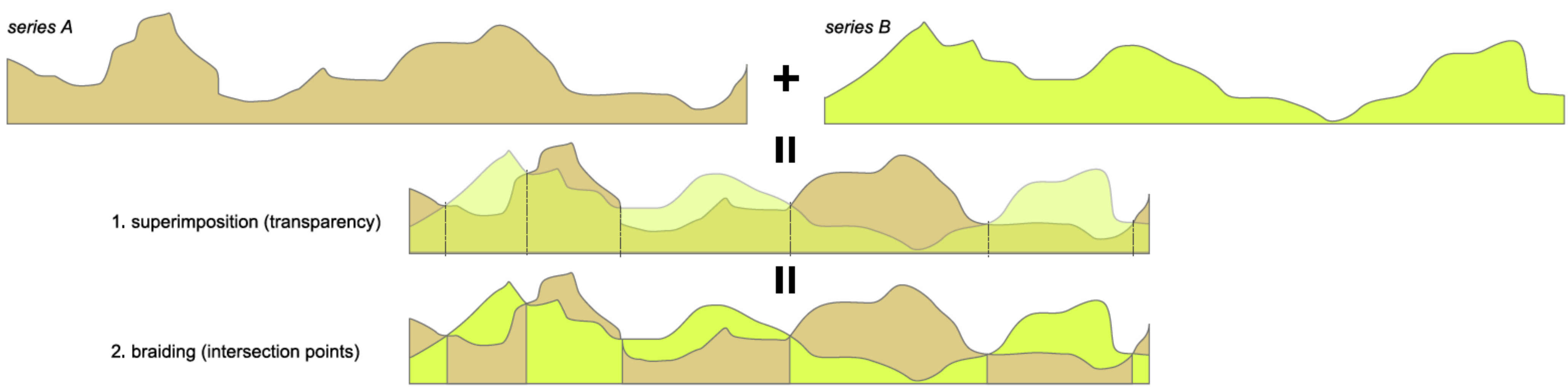
Index chart of selected technology stocks, 2000–2010.
Data source: Source: Yahoo! Finance;
http://hci.stanford.edu/jheer/files/zoo/ex/time/index-chart.html

Area Charts



62973.35
75000
70000
65000
60000
55000
50000
45000
40000
35000
30000
25000
20000
15000
Desktops
Notebooks
Tablets
Phones
Q1
Q2
Q3
Q4

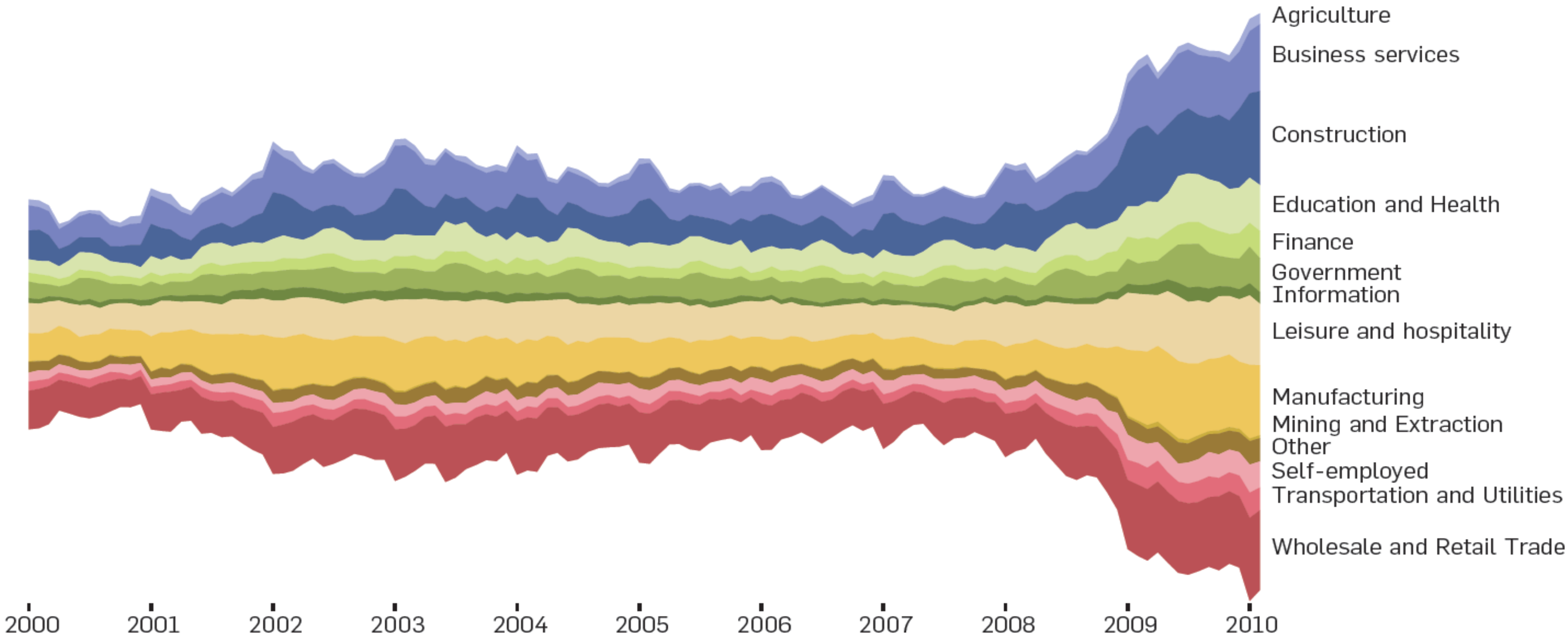
Braided Graph



- At each point in time, the graphs are sorted by value

Waqas Javed, Bryan McDonnel, Niklas Elmqvist, "Graphical Perception of Multiple Time Series," IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6, pp. 927-934, November/December, 2010

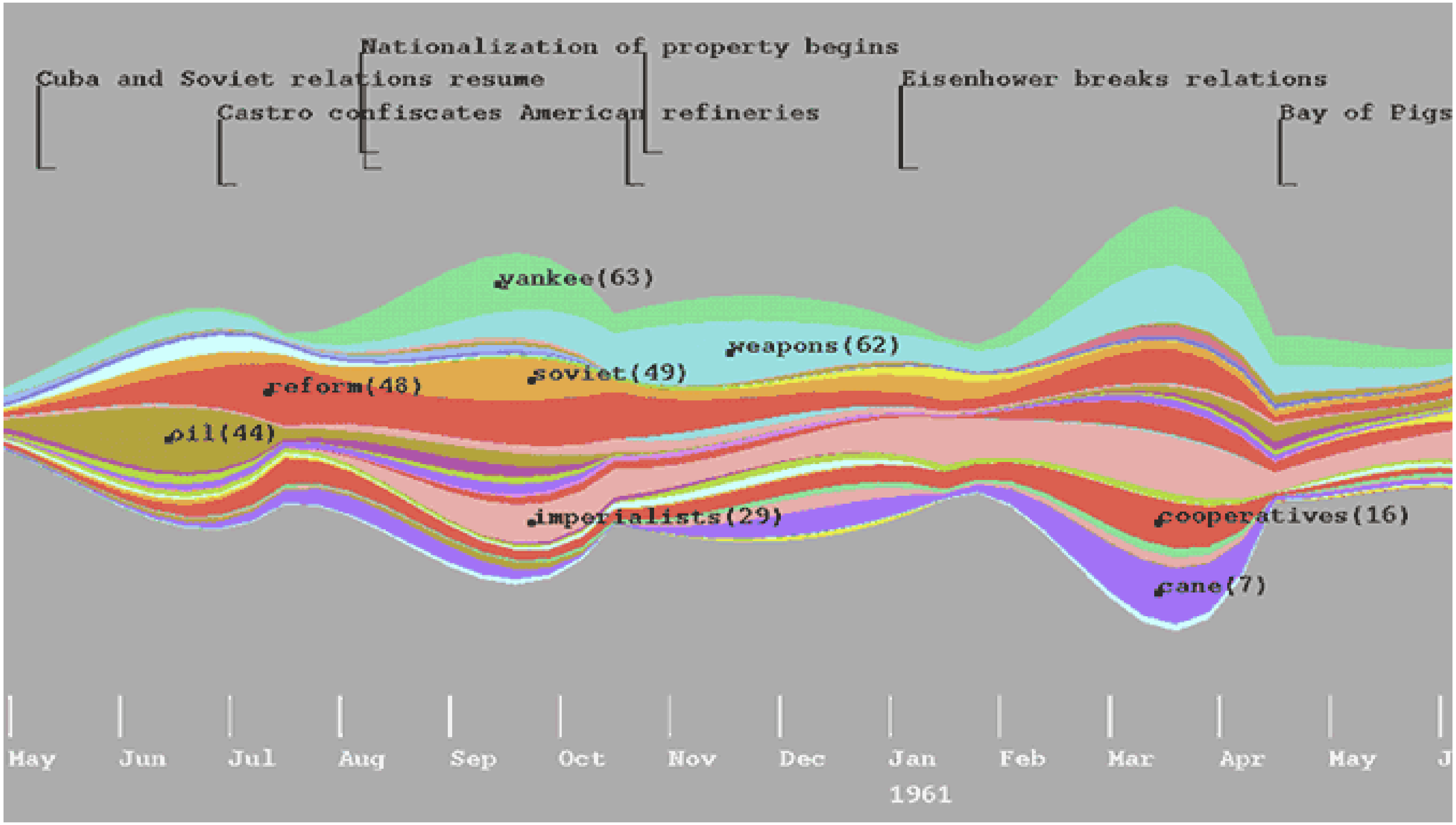
Stacked Graphs / Stream Graphs



Unemployed U.S. workers by industry, 2000–2010
Data source: U.S. Bureau of Labor Statistics
http://hci.stanford.edu/jheer/files/zoo/ex/time/stack.html

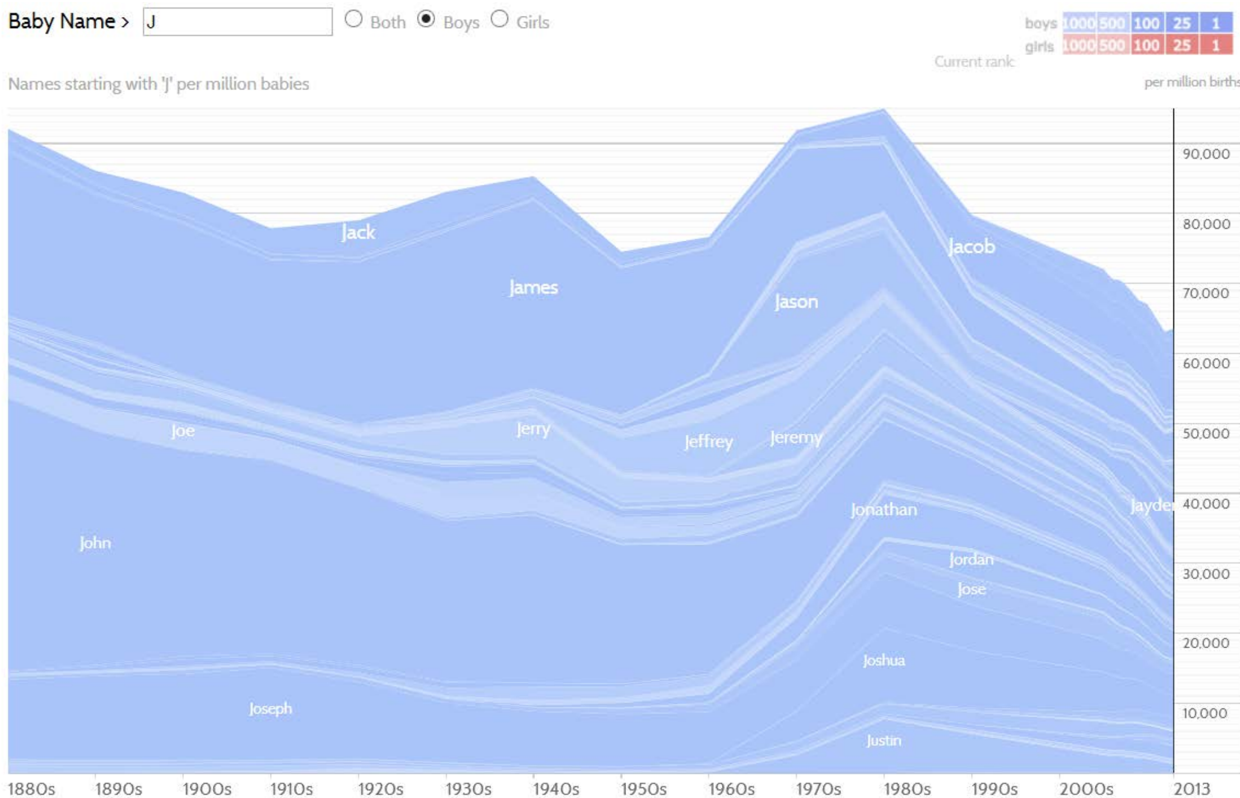
ThemeRiver

VIDEO



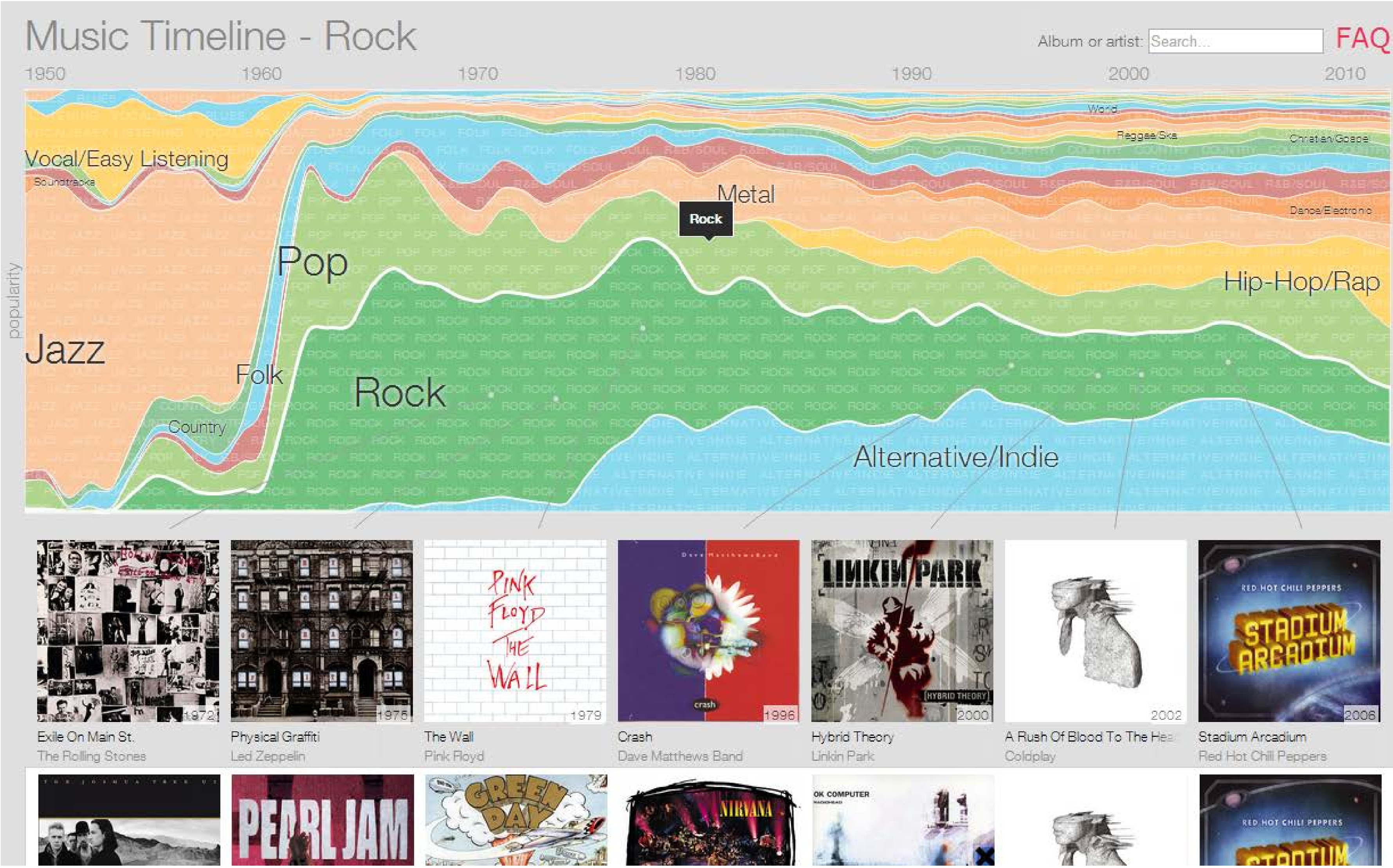
Video: http://vis.pnnl.gov/research_themeriver.stm

Baby Names Visualization



- Let users interactively explore name data, historical name popularity figures
- http://www.babynamewizard.com/voyager

Google Play Music



Popularity: how many Google Play Music users have an artist or album from a specific year in their music library

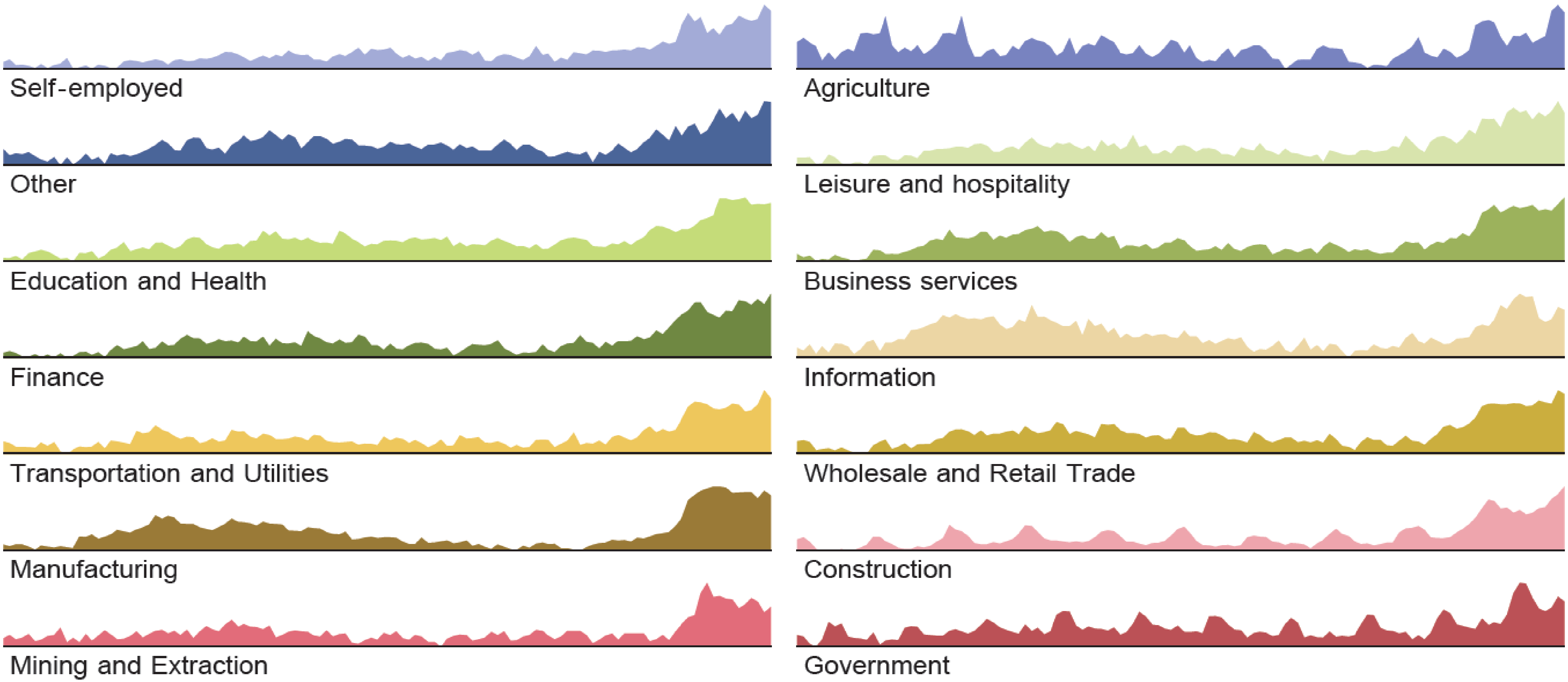
Stacked Graphs / Stream graphs

- Visual summation
- Shows aggregate patterns
- Interaction important
  - Drill-down into a subset of individual series
  - Interactive search and filtering
- Limitations
  - No negative numbers
  - Works only for data where summation makes sense (does not work for temperatures …)
  - Comparison of individual timeseries values is difficult across years
  - Scalability

Further Papers

- Paper Stacked Graphs – Geometry & Aesthetics by Lee Byron and Martin Wattenberg
- TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs (Best Paper Nominee)
  D. Baur, B. Lee, and S. Carpendale

Small Multiples

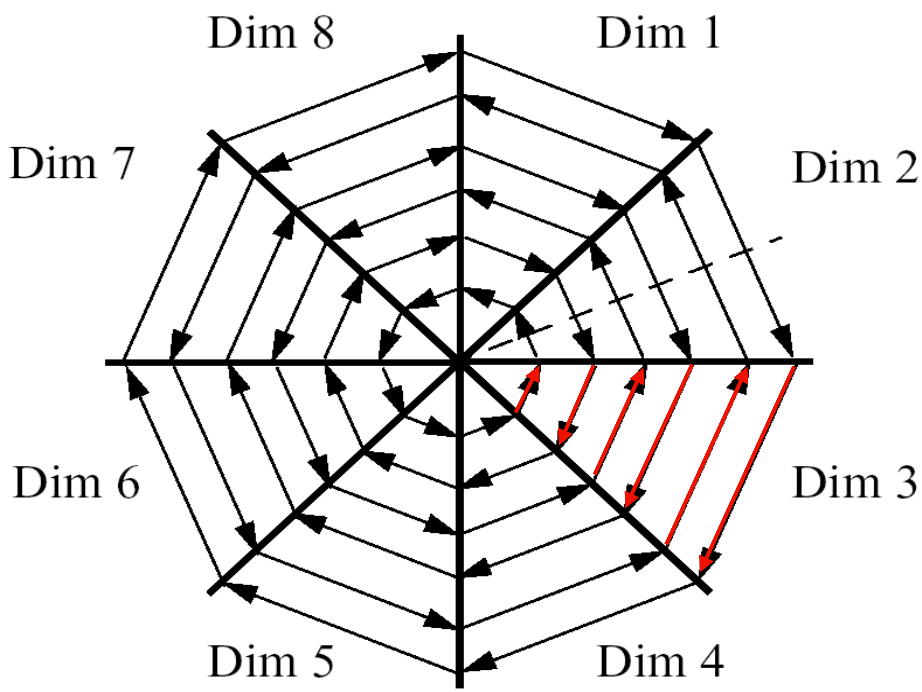


Unemployed U.S. workers by industry, 2000–2010
Data source: U.S. Bureau of Labor Statistics
http://hci.stanford.edu/jheer/files/zoo/ex/time/multiples.html

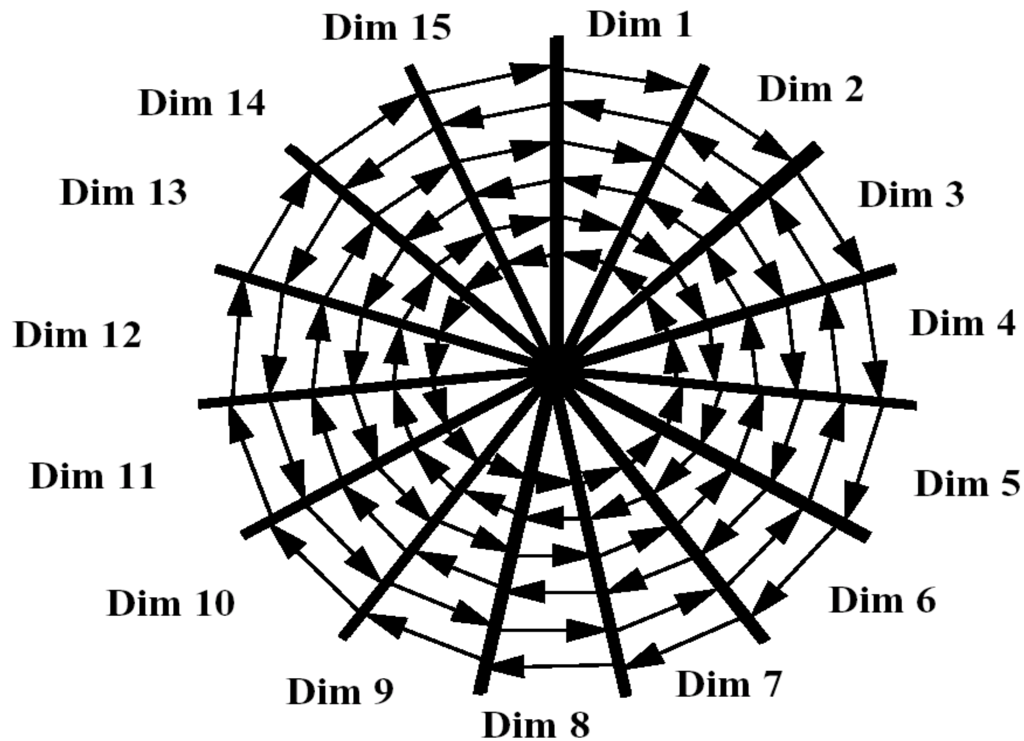
Small Multiples

- Can be constructed for plots, area graphs, bar charts, …
- Also interesting as linked views where coordinated interaction takes place across all individual visualizations
- Separate visualization can be more effective than overlaying all data in a single plot

Circle Segments Technique [AKK96]



Arrangement of 8-dim. Data

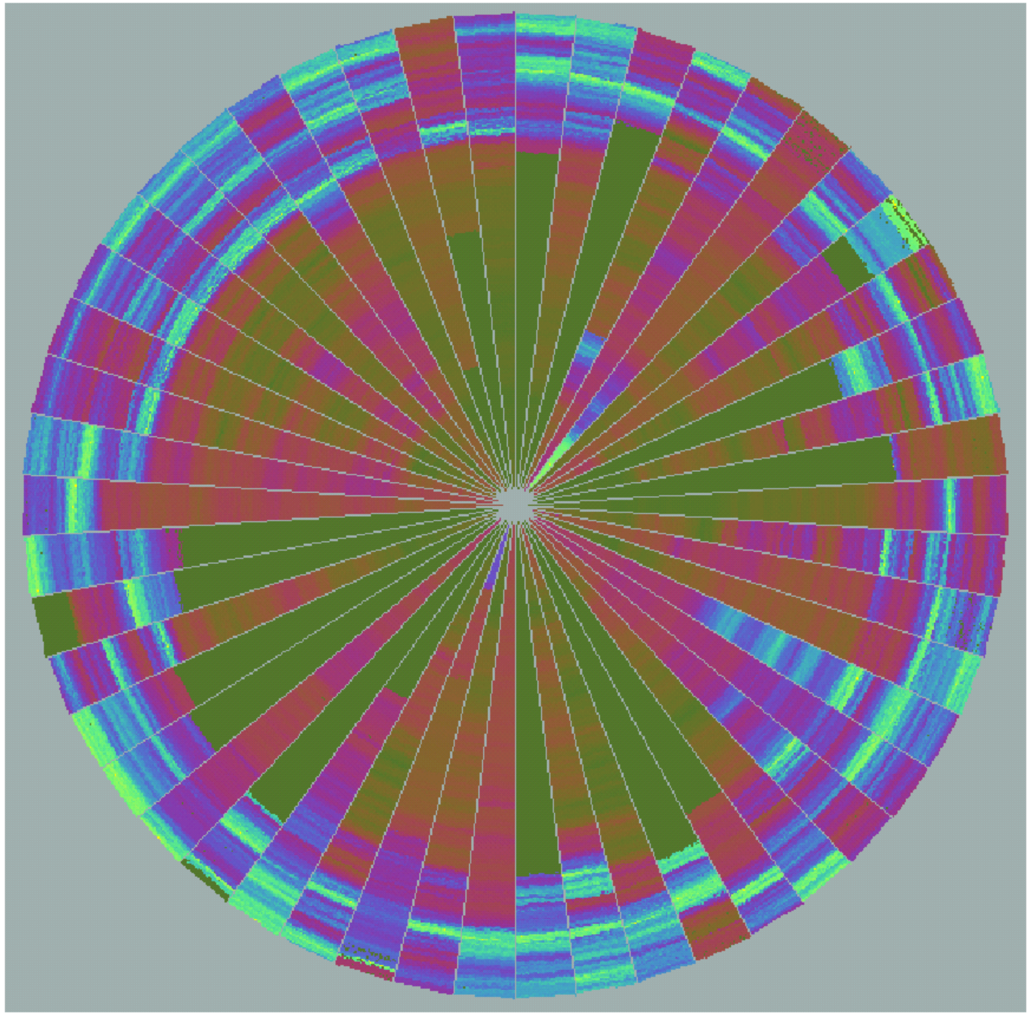


Arrangement of 15-dim. Data

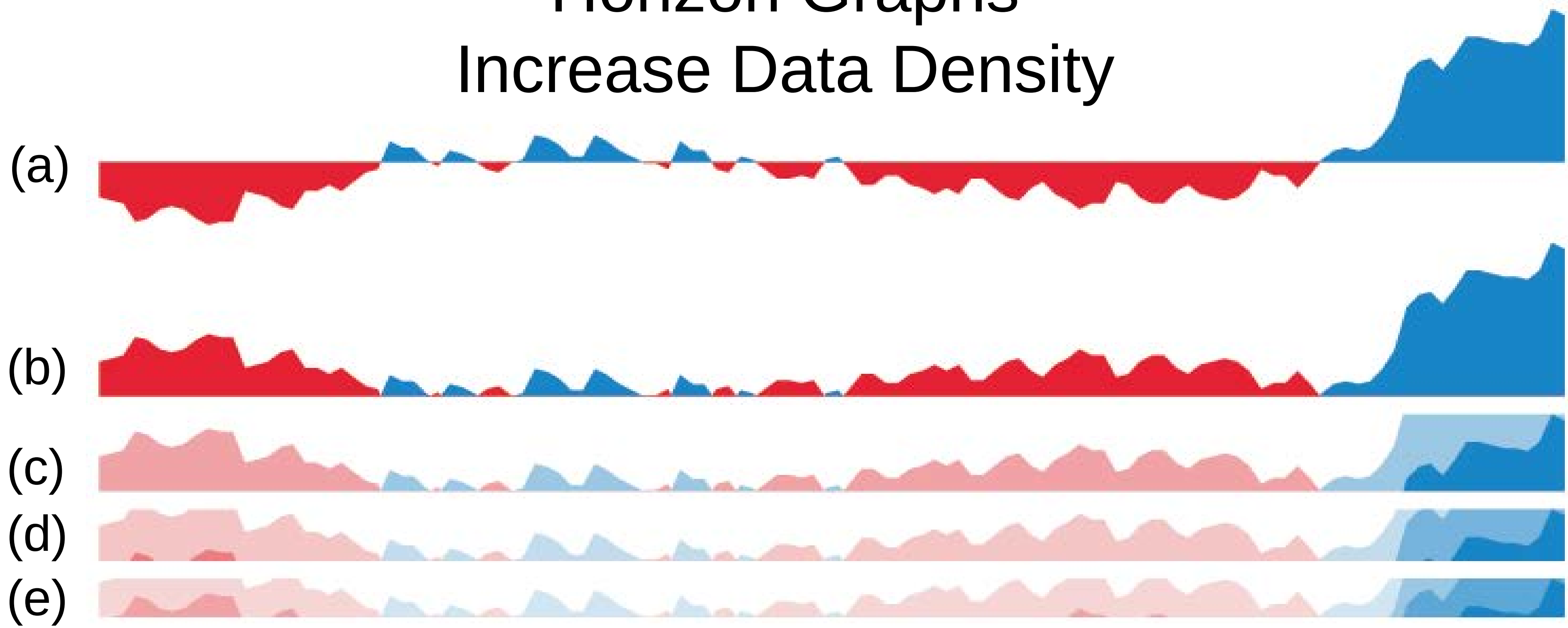
Each Segment is filled with data in a back to forth order and ***each pixel*** is used to represent data

Example

- Time series of 50 stocks (265000 data items) of the Frankfurt Stock Index
- Two keys
  - Stock
  - Time
- Only one value: stock price
  - Light color represents high stock price
  - Dark color represents low stock price
- Oldest data in the middle
- Small multiples in circular layout



Horizon Graphs
Increase Data Density



(a) Regular time series area graph

(b) Mirror negative values into positive

(c)-(e) Split data values at a threshold into multiple graphs stacked on top of each other

Unemployed U.S. workers by industry, 2000–2010

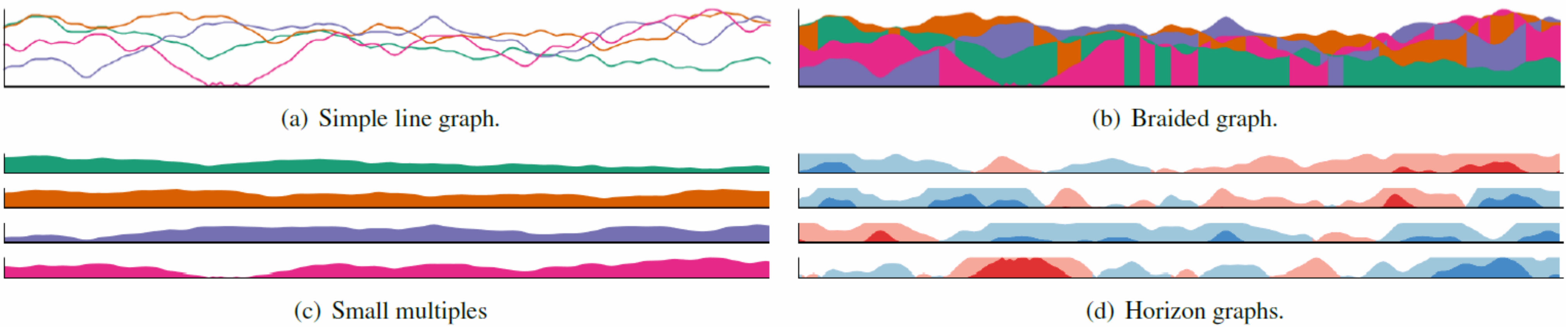
Data source: U.S. Bureau of Labor Statistics

http://hci.stanford.edu/jheer/files/zoo/ex/time/horizon.html (interactive demo)

Horizon Graphs

- Increases data density by dividing the graph into bands and layering them to create a nested form.
- The chart preserves data resolution but uses only a fraction of the space.
- Interpretation has to be learned
- Pretty slow at finding the maximum / minimum at a certain point in time due to cognitive effort to reconstruct value from multiple stacked layers
- Found to be more effective than the standard plot when the chart sizes get quite small.

Comparison of Graph Types

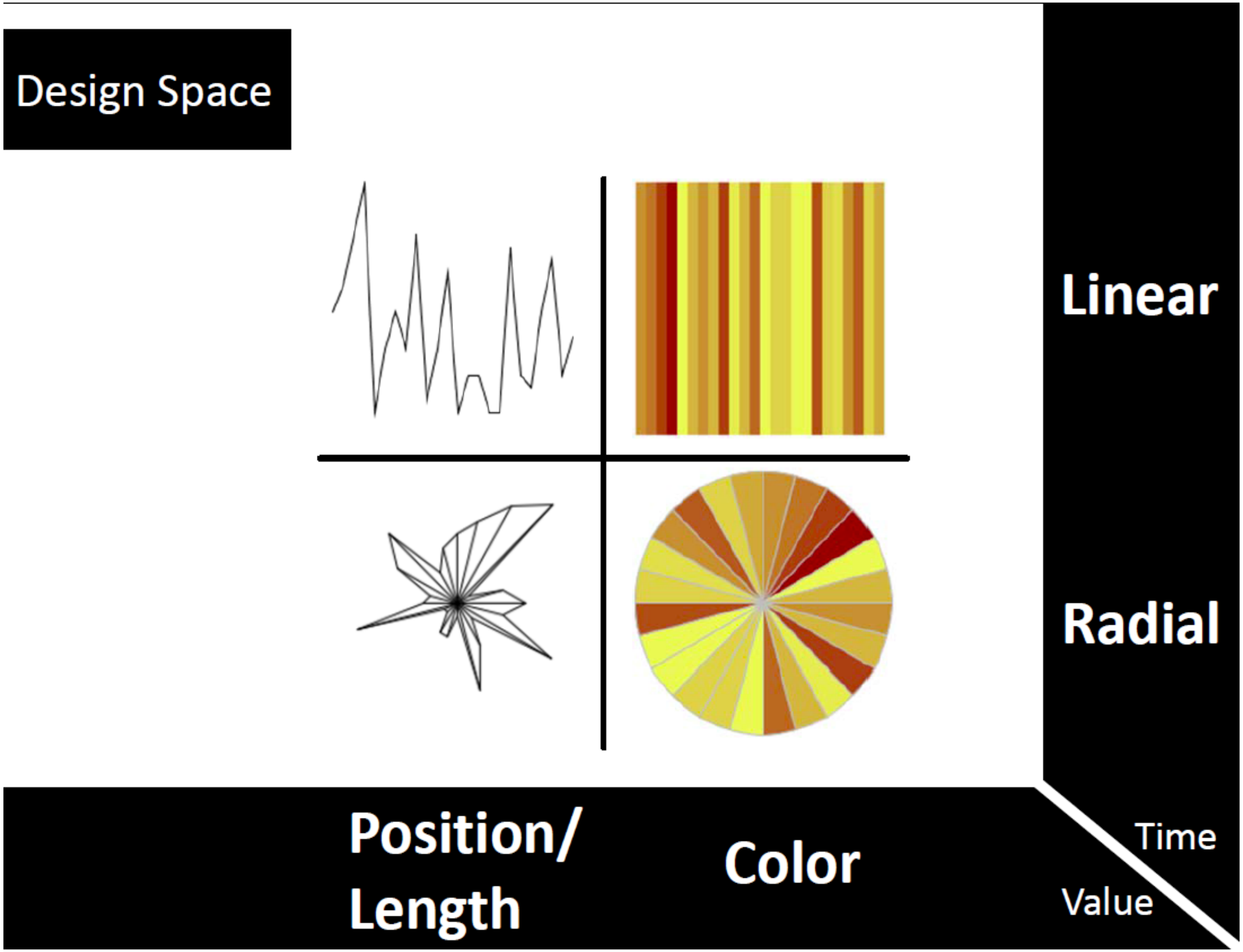


(a) Simple line graph. (b) Braided graph. (c) Small multiples (d) Horizon graphs.

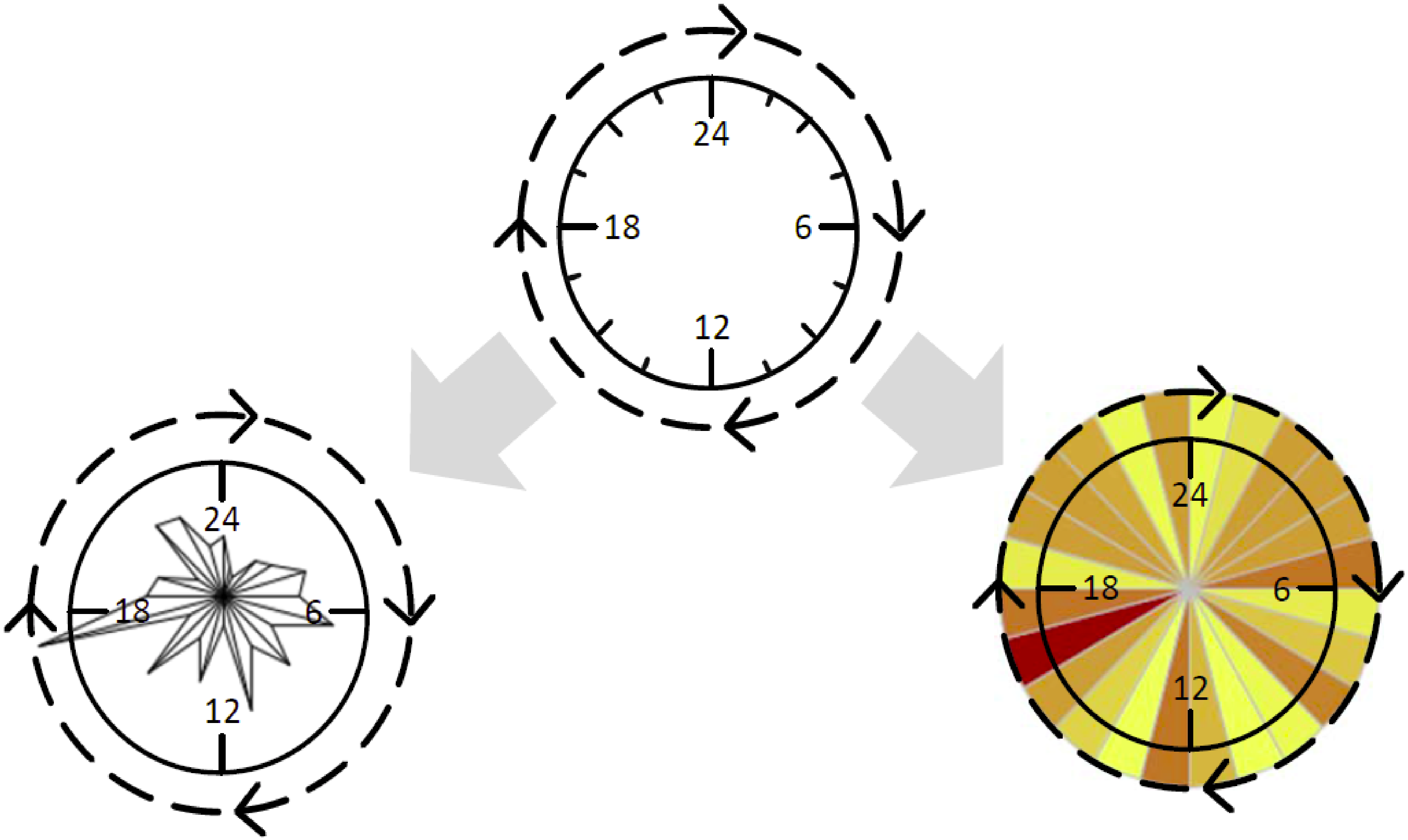
- Shared-space techniques--like standard line graphs--are typically more efficient for comparisons over smaller visual spans where the impact of overlap and clutter is reduced, e.g. finding the maximum at a certain point in time (focus on a single small visual area).
- Techniques that create separate charts for each time series--such as small multiples and horizon graphs--are generally more efficient for comparisons across time series with a large visual span, e.g. comparing the values at two different points in time.

Waqas Javed, Bryan McDonnel, Niklas Elmqvist, "Graphical Perception of Multiple Time Series,"
IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6, pp. 927-934, November/December, 2010

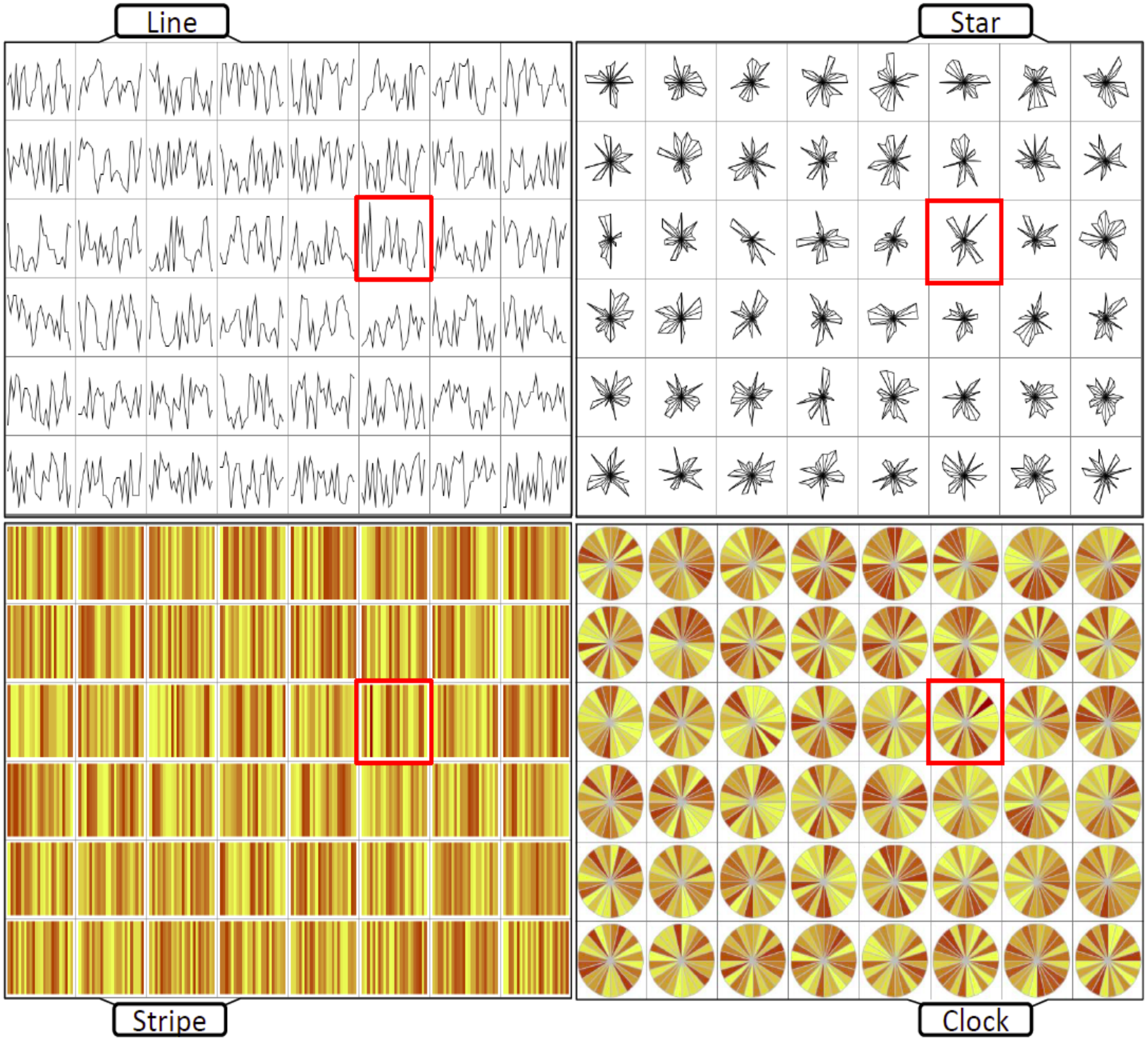
Time Glyphs


Design Space
Linear
Radial
Position/
Length
Color
Time
Value

Clock Metaphor

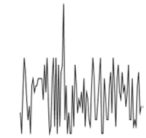


Small Multiples Layout

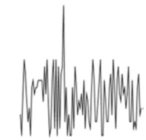


Recommendations for Time Glyphs

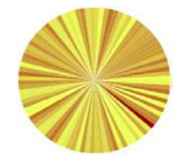
- LINEAR LAYOUT improves value comparison



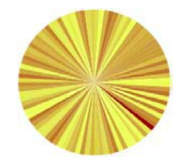
- POSITION/LENGTH encodings improve value comparison



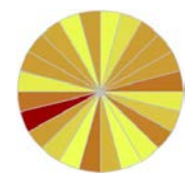
- Use TRIANGULAR SHAPES for color encoding



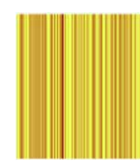
- Use CIRCULAR LAYOUTS for detecting temporal locations



- Use SUFFICIENT SPACE for time-dependent tasks

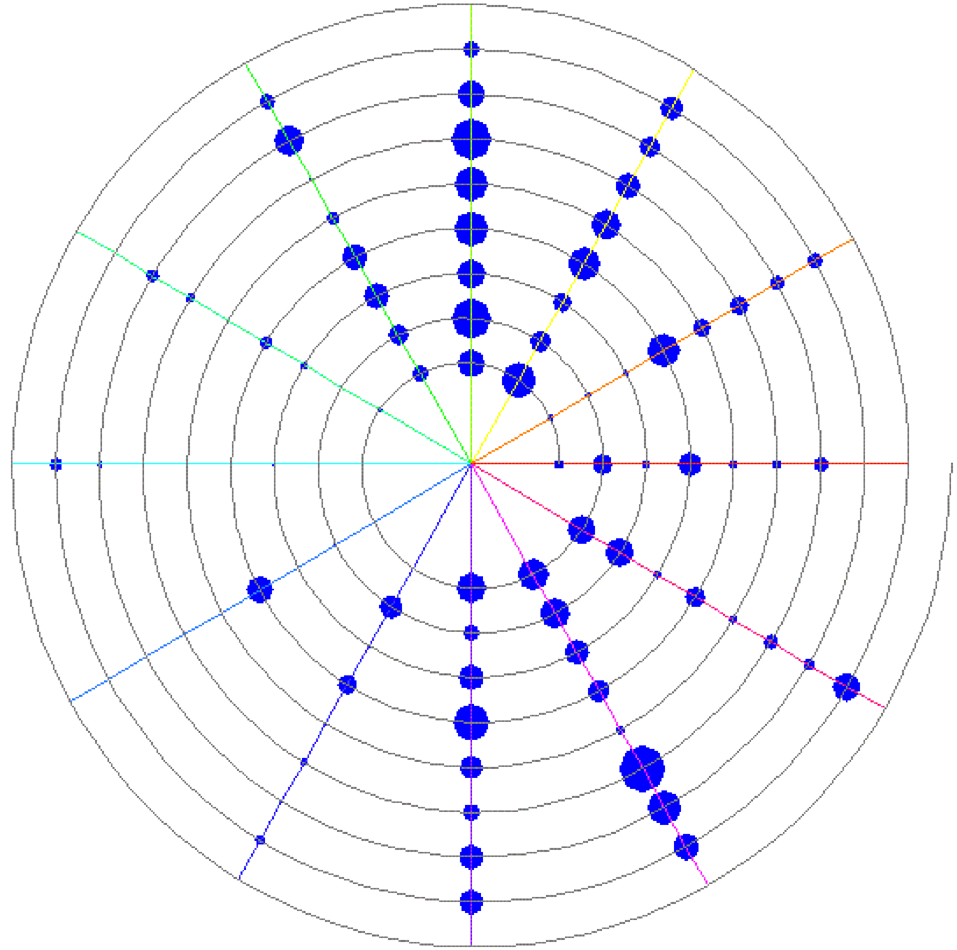


- Avoid COLOR ENCODINGS for higher data densities



Spiral Layout: Serial Periodic Data

- Radial visualization of time
  - Weeks, months, and years are periods that recur
- Encoding
  - One year per loop
  - Same month on radial bars
  - Quantity represented by size of blob



Spokes, showing monthly consumption percentages for Baphia Capparidifolia during the period 1980 – 1988.

Spiral Graphs

- Periodicity is important!

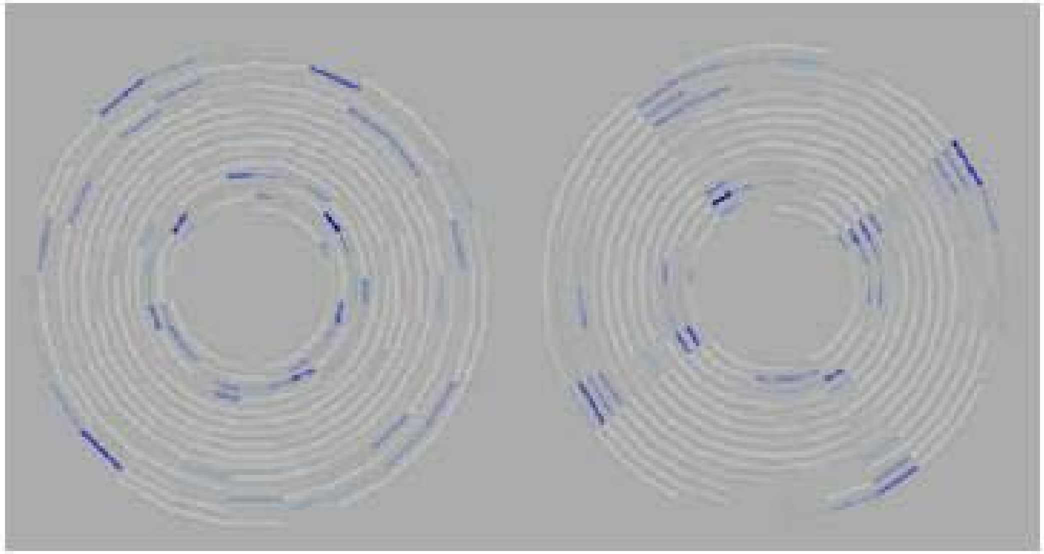
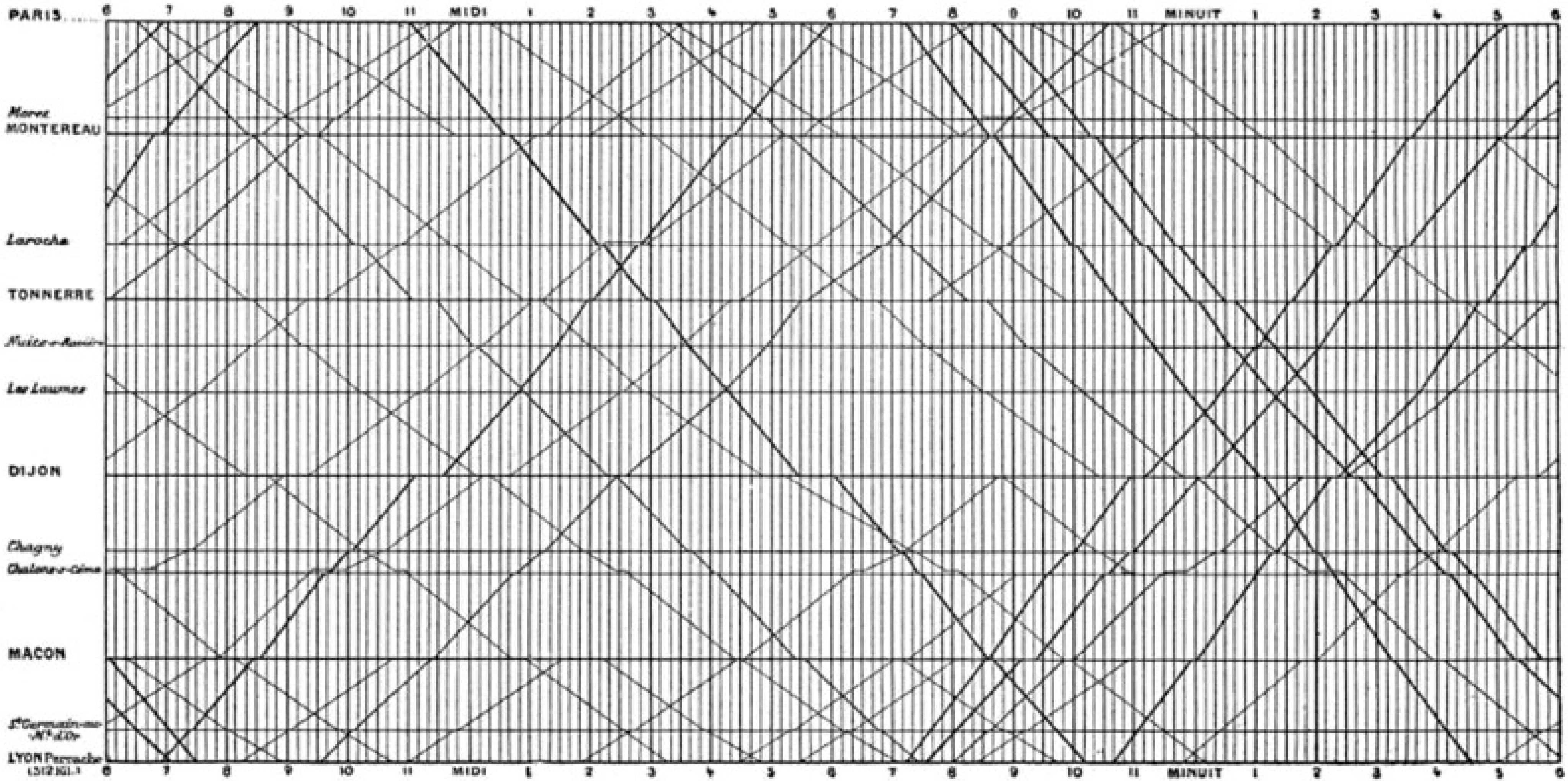


Figure 5: *Spiral Graph* Visualization of Monthly Notifications of Illness. Left: Assumed cycle of 27 days. Right: Valid cycle of 28 days.

John V. Carlis and Joseph A. Konstan. Interactive Visualization of Serial Periodic Data, UIST, 1998

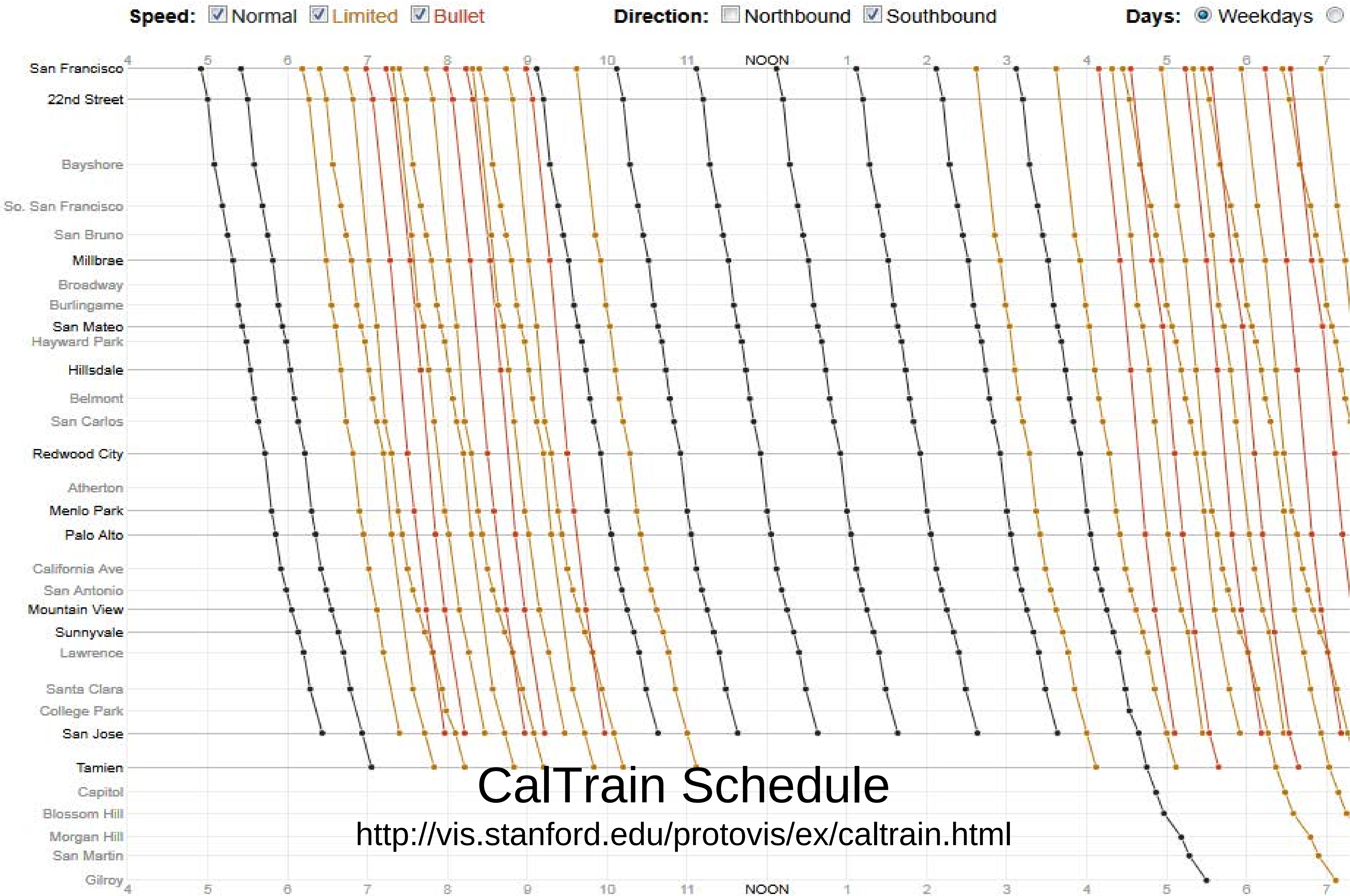
Marey's Train Schedule (1885)

Stops → Time



Marey’s Train Schedule (1885)

- Stations are separated vertically in proportion to distance
  - Slope of the line corresponds to train speed
    - the steeper the line, the faster the train
  - Horizontal line length: stop length
  - Intersection: time/place of crossing

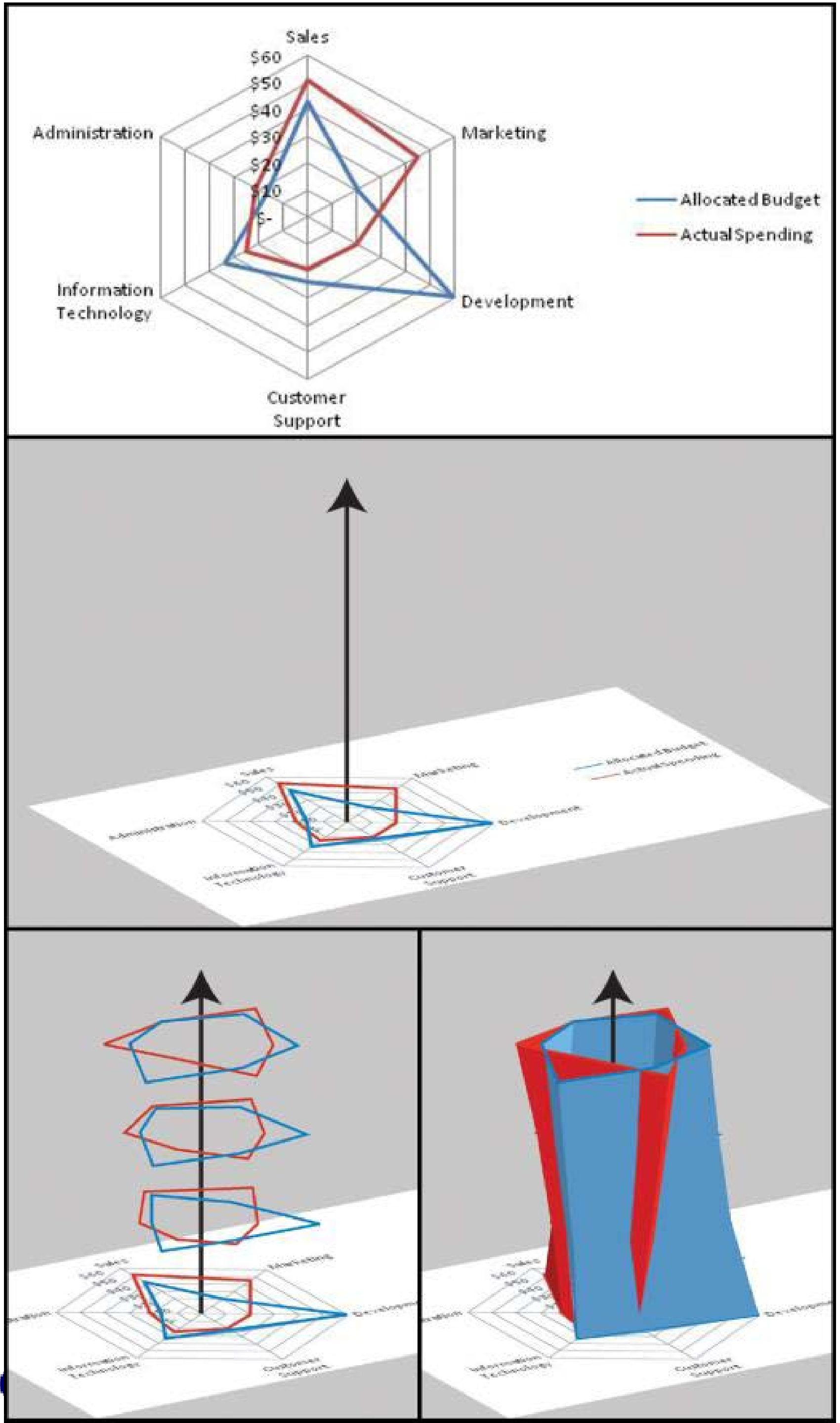


CalTrain Schedule

http://vis.stanford.edu/protovis/ex/caltrain.html

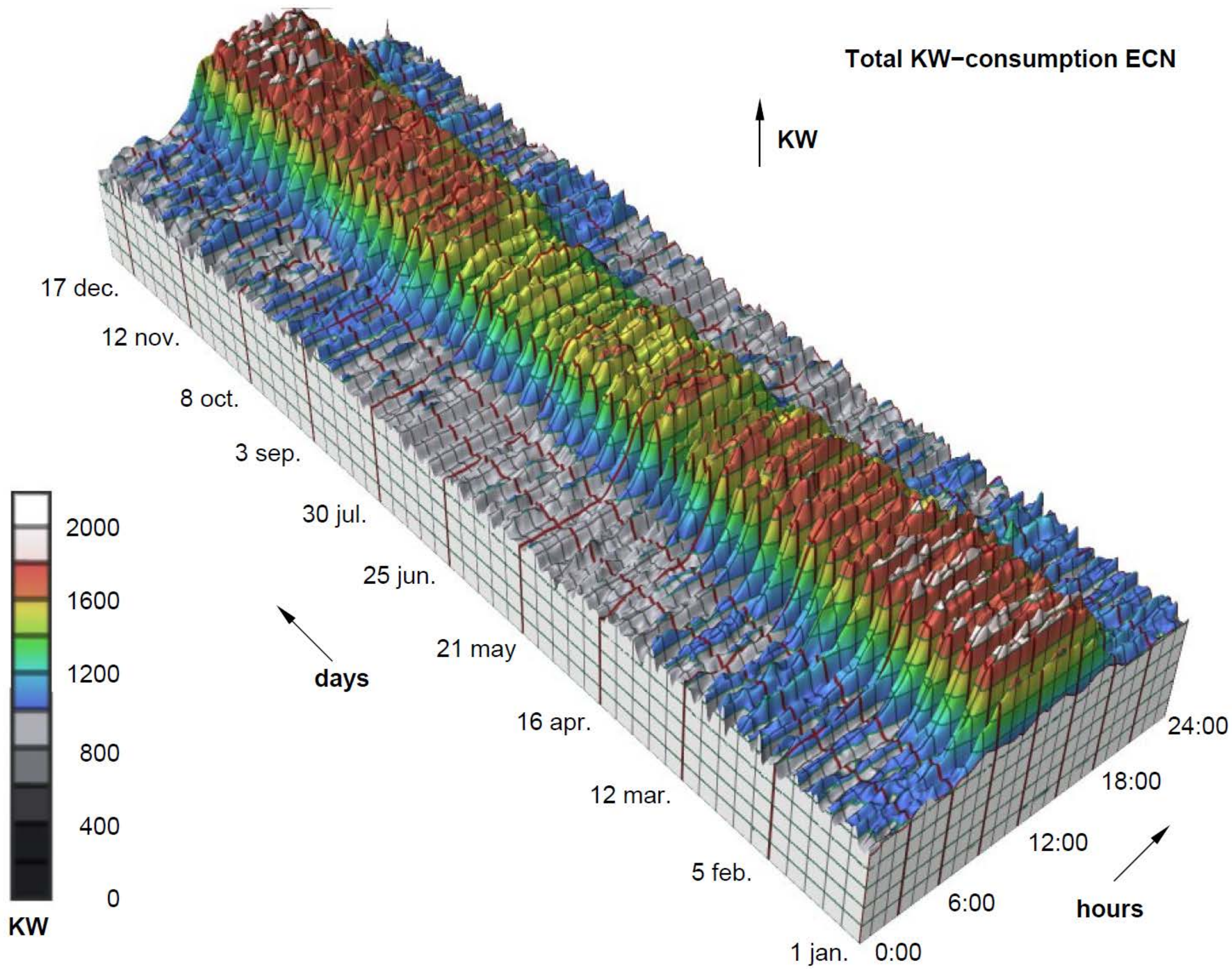
Wakame

- Sweeping star plots in 3D to visualize changes over time



C. Forlines, K. Wittenburg,
Wakame: Sense Making of Multi-Dimensional Spatial-Temporal Data,
ACM AVI, 2010

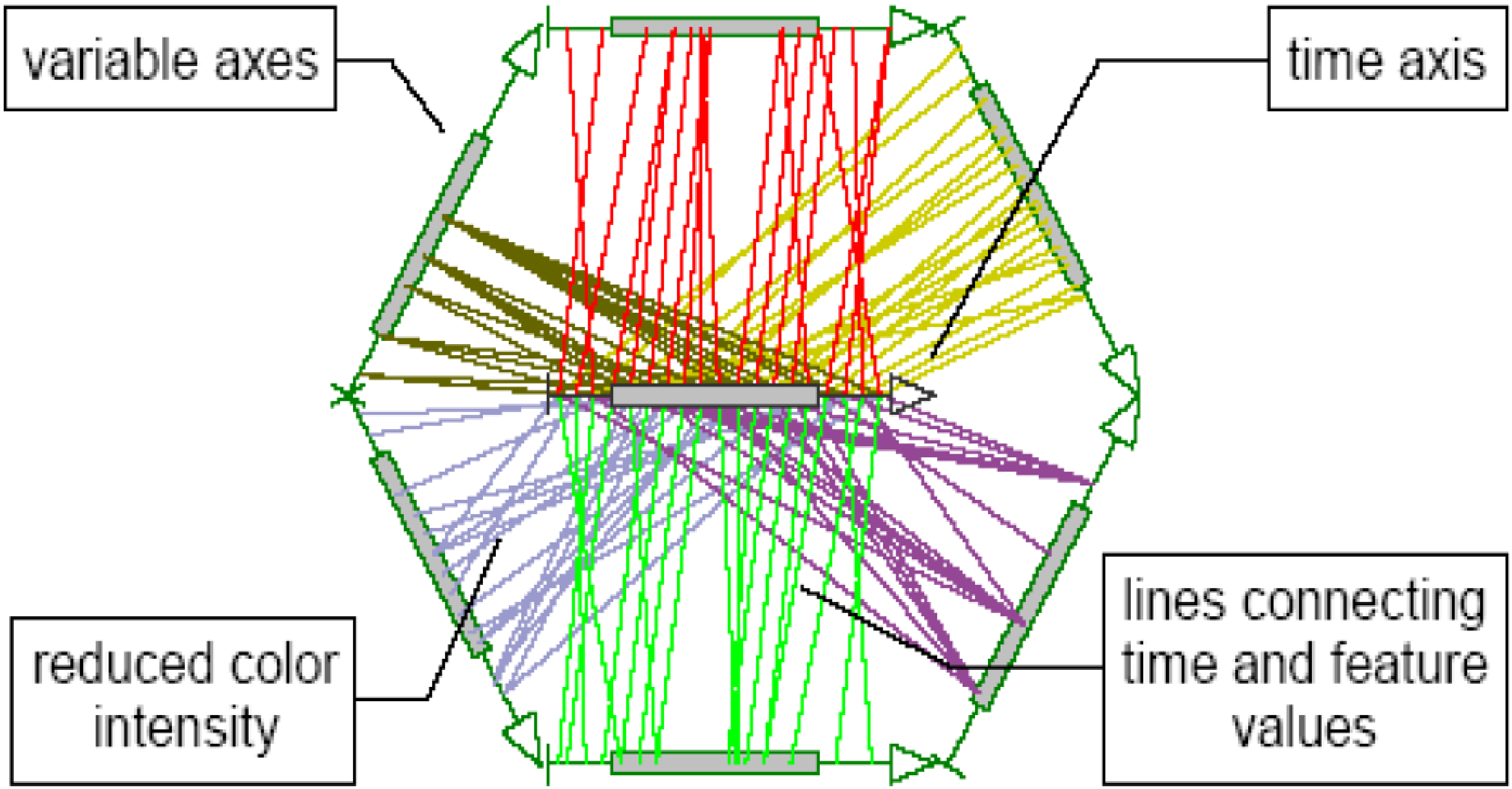
Perspective 3D-Visualization



Power demand by research facility, displayed as a function of hours and days

van Wijk and van Selow, Cluster and Calendar based Visualization of Time Series Data, IEEE InfoVis 1999

TimeWheel



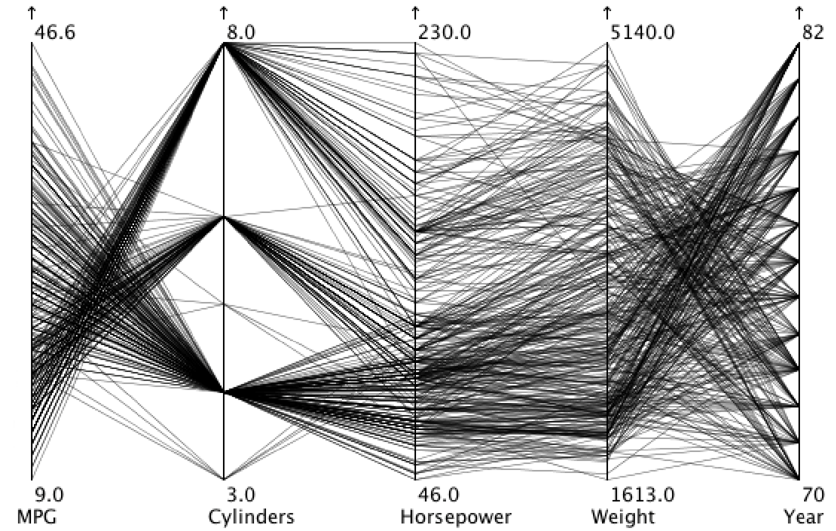
- Parallel coordinates arranged in a circular layout and the time axis in the center

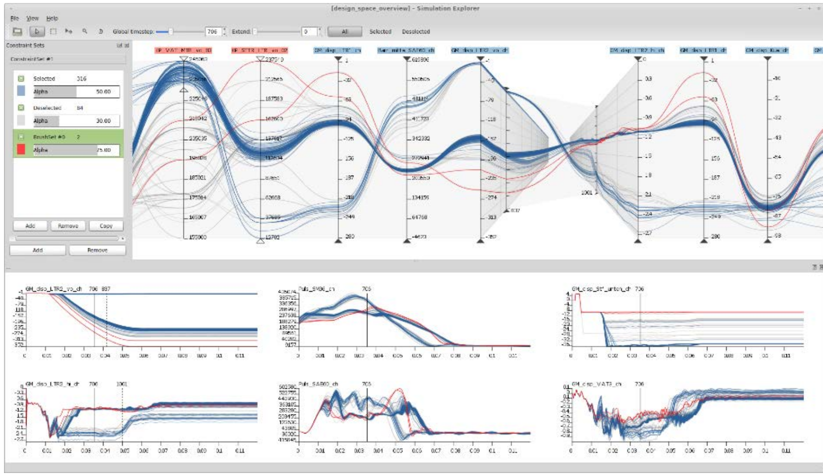
Parallel Coordinates + Time-dependent Data

Simulation Explorer: Coordinated Multi-View Display

Parallel Coordinates and Time-Dependent Data

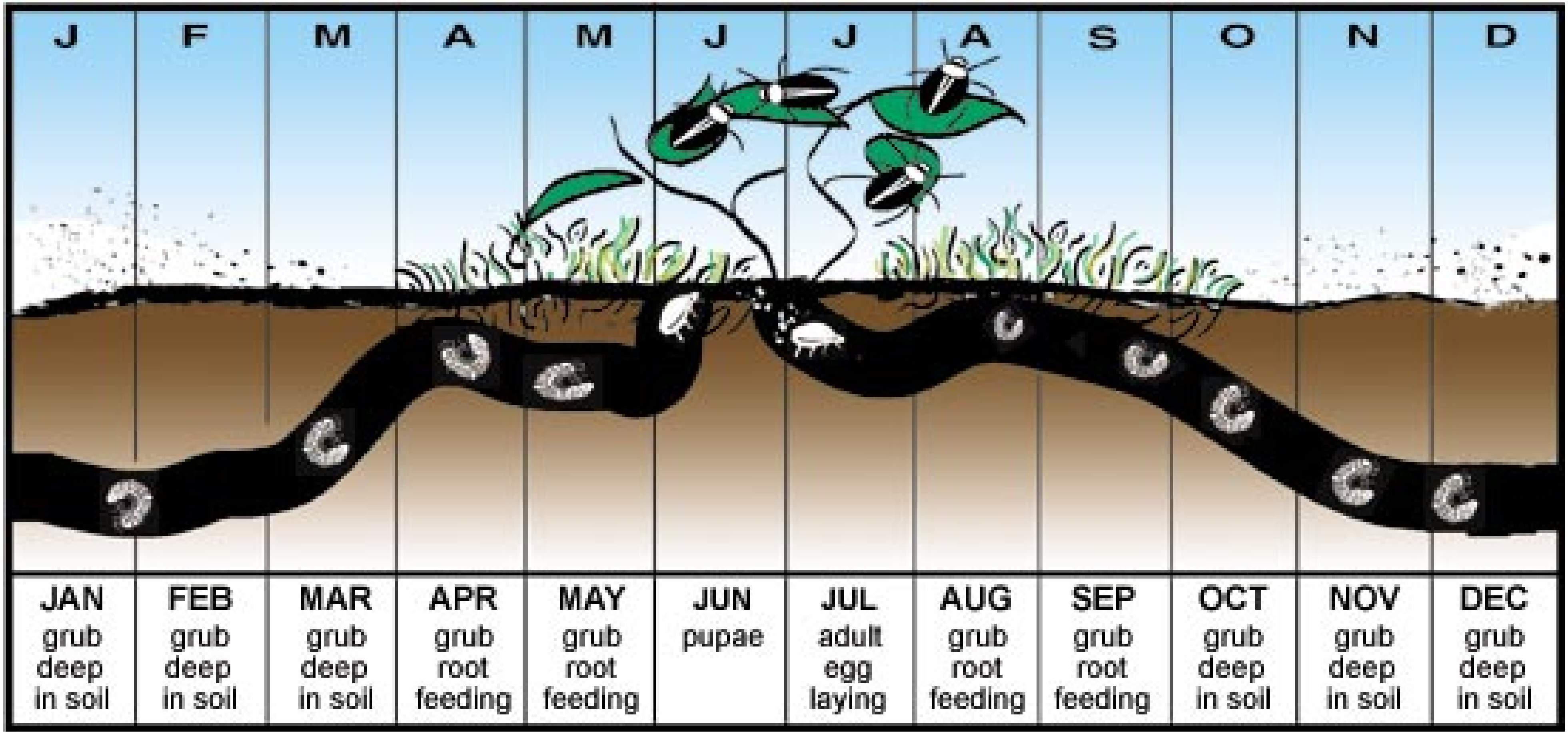
- Time is simply included as an additional axis in the parallel coordinates plot
- The attribute values for discrete time steps can be integrated as separate axes into the parallel-coordinates plot, which essentially turns it into a partial time-series plot. The order of the axes which represent a fixed point in time, needs to be always maintained – value of the car after one year, after two years, ….
- A coordinated multi-view approach is used to show the parallel-coordinates plot of a single time-step and a small multiples display of regular timeseries plots.
- Vanishing-point widget integrates time-series plots into the parallel-coordinates display.





Life circle of Japanese Beetles

Based on: L. Newman, Man and Insects, 1965



End

Visualization

Trees

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

No lecture next week

- 28.5.2015 – NO Lecture

Acknowledgements

- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Harald Reiterer, Universität Konstanz
  - Daniel A. Keim, Universität Konstanz
  - Jeff Heer, Stanford University
  - Also based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.

Data Space

- Statistical distributions: Multi-dimensional databases,…
- Time Series
- Trees/Hierarchies: directories,…
- Networks/Graphs: web, communications,…
- Maps
- Unstructured data, e.g. document collections: digital libraries,…

Trees/Hierarchical Data

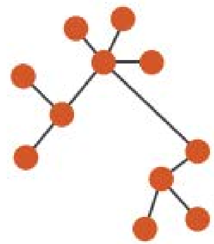
- Natural hierarchies
  - Spatial entities: counties, states, and countries
  - Command structures for businesses and governments
  - Software packages
  - Phylogenetic trees (evolutionary branching patterns)
  - …
- No apparent hierarchy
  - Use statistical methods (e.g. clustering)
  - …

Gallery: treevis.net - A Visual Bibliography of Tree Visualization by Hans-Jörg Schulz
Overview: http://staff.science.uva.nl/~marshall/publications/StarGraphVisuInInfoVis.pdf

Four Basic Visualization Techniques

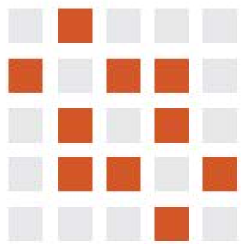
Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



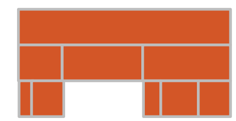
Adjacency Matrix
Derived Table

✔ NETWORKS ✔ TREES



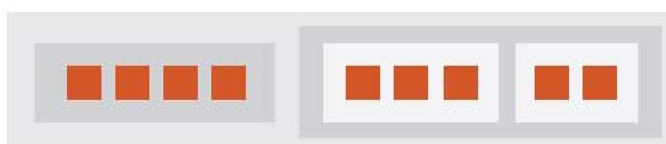
Adjaceny Diagrams
Area Marks

✖ NETWORKS ✔ TREES

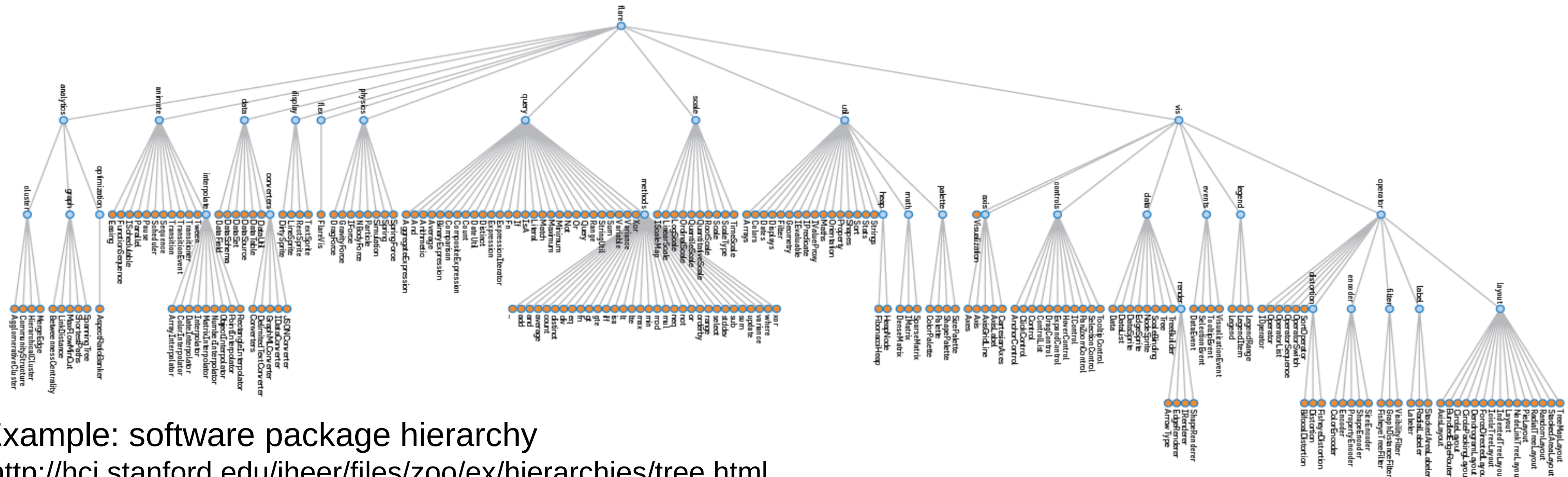


Enclosure
Containment Marks

✖ NETWORKS ✔ TREES



Cartesian Node-Link Diagrams



Example: software package hierarchy
http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/tree.html

- Encoding
  - Link: line marks
  - Node: point marks
- Cartesian axis orientation
  - Layer encodes depth in tree
- Tasks
  - Understanding topology
  - Following paths
- Scalability
  - 1K - 10K nodes

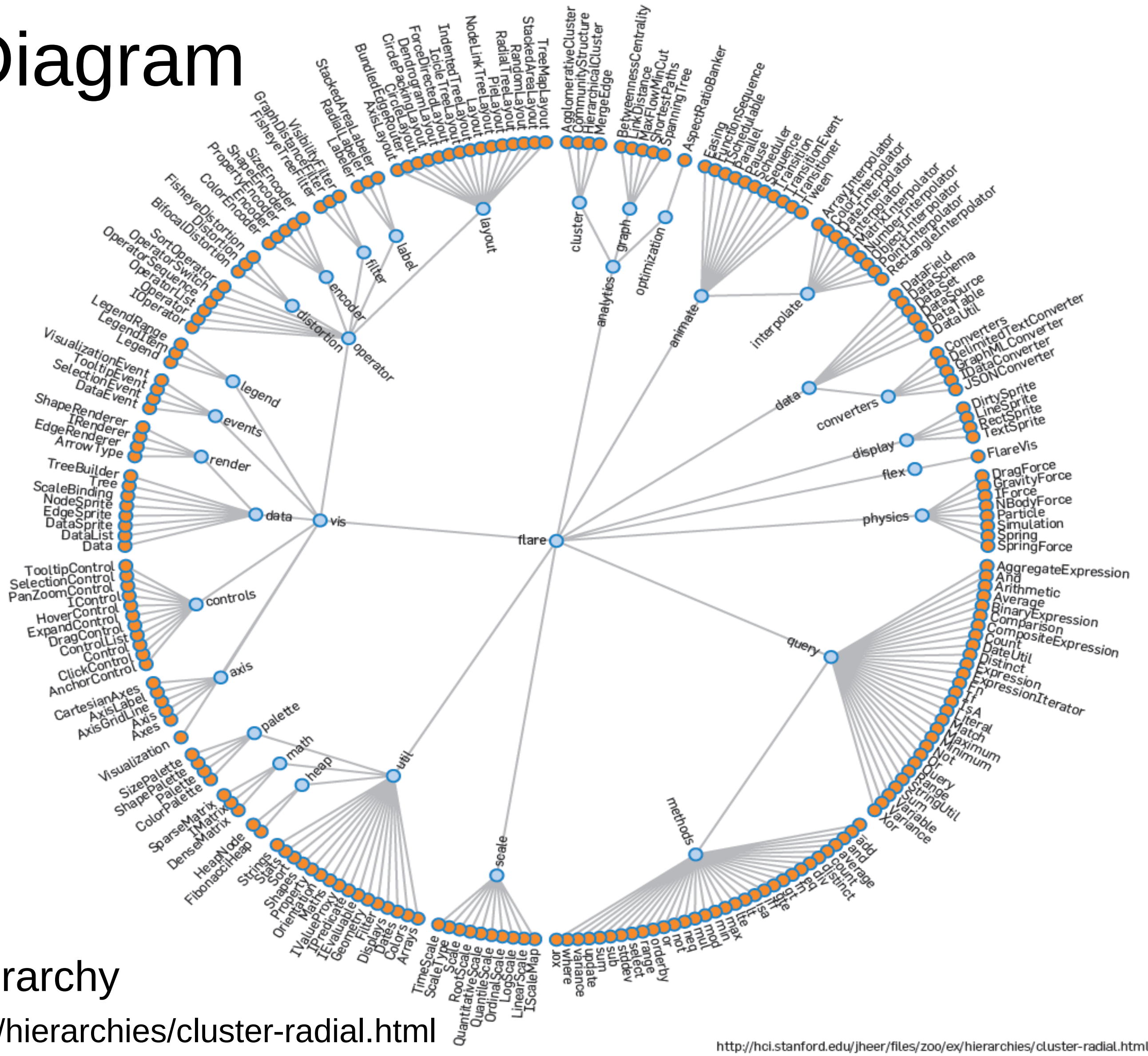
- Reingold-Tilford layout algorithm
  - Clearly encodes depth level
  - Edge crossings avoided
  - Isomorphic subtrees drawn identically
  - Ordering and symmetry preserved
  - *Compact layout (don’t waste space)*

Classic Tree Drawing Papers

- Wetherell, C. and A. Shannon (1979). "Tidy drawings of trees." IEEE Transactions on Software Engineering SE-5(5): 514--520.
- Reingold, E. M. and J. S. Tilford (1981). "Tidier Drawing of Trees." IEEE Transactions on Software Engineering 7(2): 223-228.
- Walker, J. Q. (1990). "A Node-positioning Algorithm for General Trees." Software - Practice and Experience 20(7): 685–705.
- Buchheim, C., M. Jünger and S. Leipert (2002). Improving Walker's Algorithm to Run in Linear Time. International Symposium on Graph Drawing, Lecture Notes in Computer Science 2528, Springer: 344-353.

Radial Node-Link Diagram Dendrogram

- Encoding
  - Link: line marks
  - Node: point marks
- Radial axis orientation
  - Angular proximity: siblings
  - Distance from center: depth in tree
  - Places leaf nodes on same level
- Tasks
  - Understanding topology
  - Following paths
- Scalability
  - 1K - 10K nodes

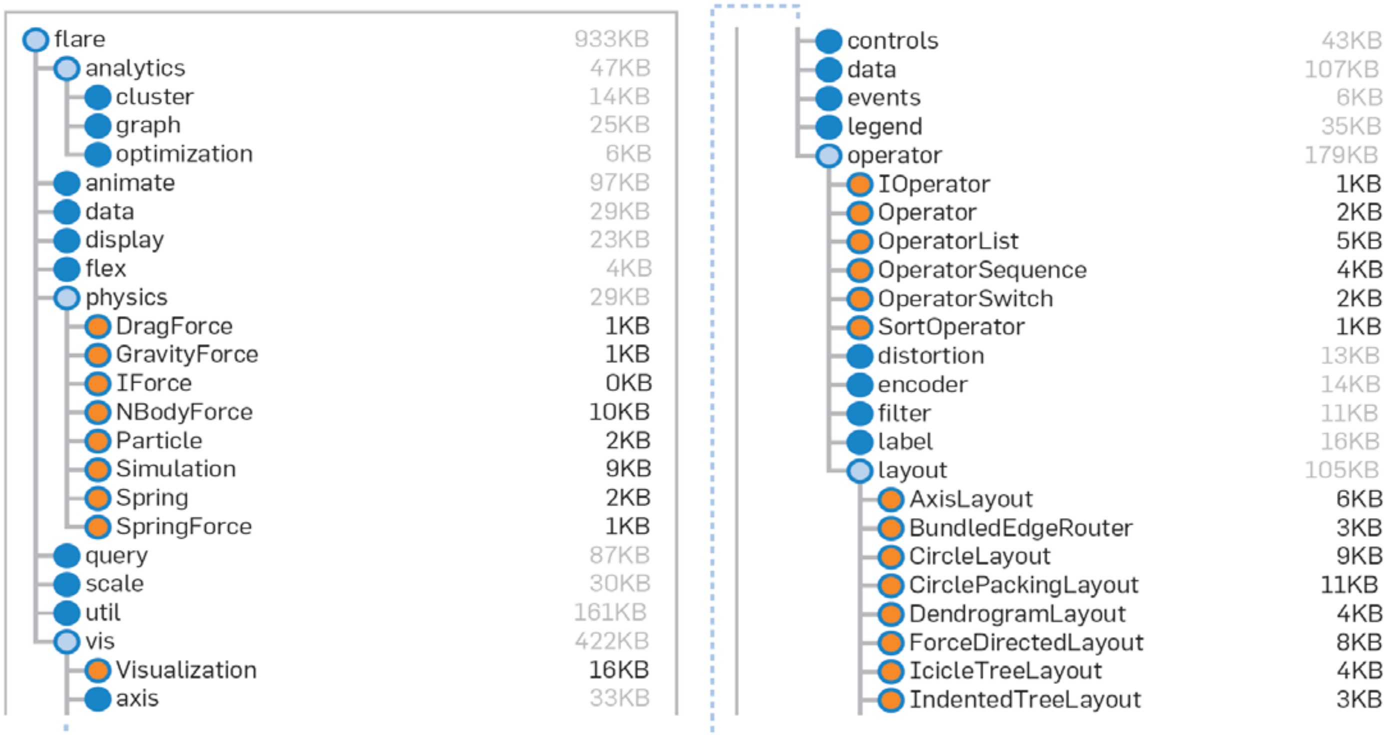


http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/cluster-radial.html

- Example: software package hierarchy

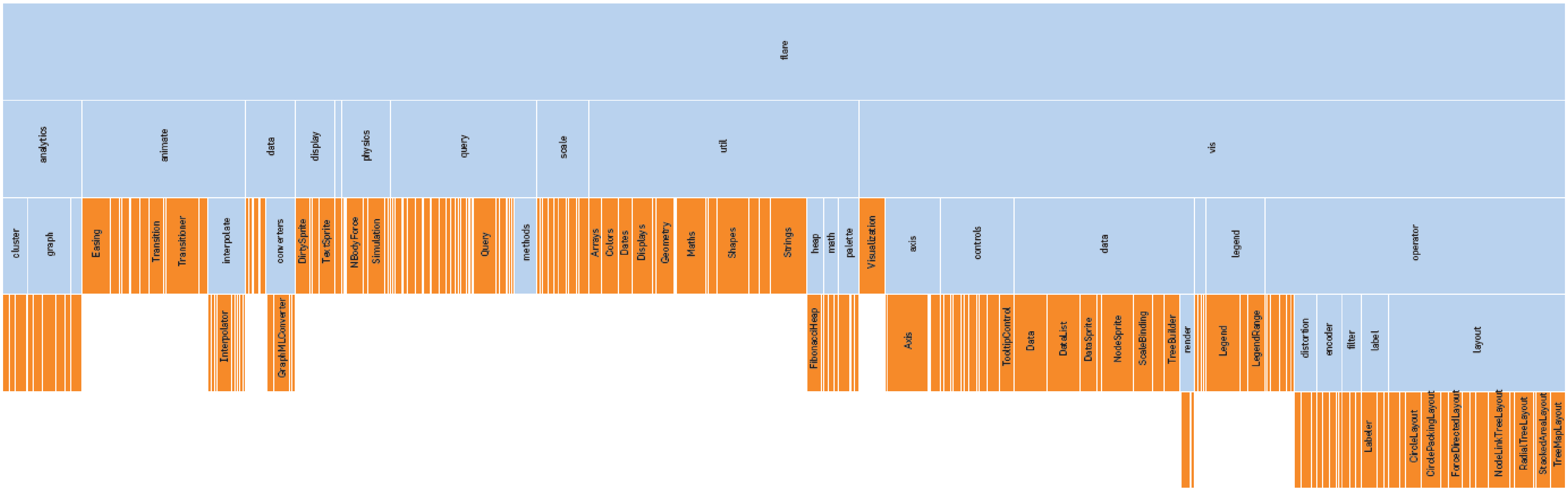
  http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/cluster-radial.html

Node-Link Diagram: Indented Tree



- Efficient *interactive* exploration of the tree to find a specific node.
- Rapid scanning of node labels and other node attributes that can be displayed adjacent to the hierarchy.
- Vertical space requirements!
- Difficult: multi-scale inferences
- http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/indent.html

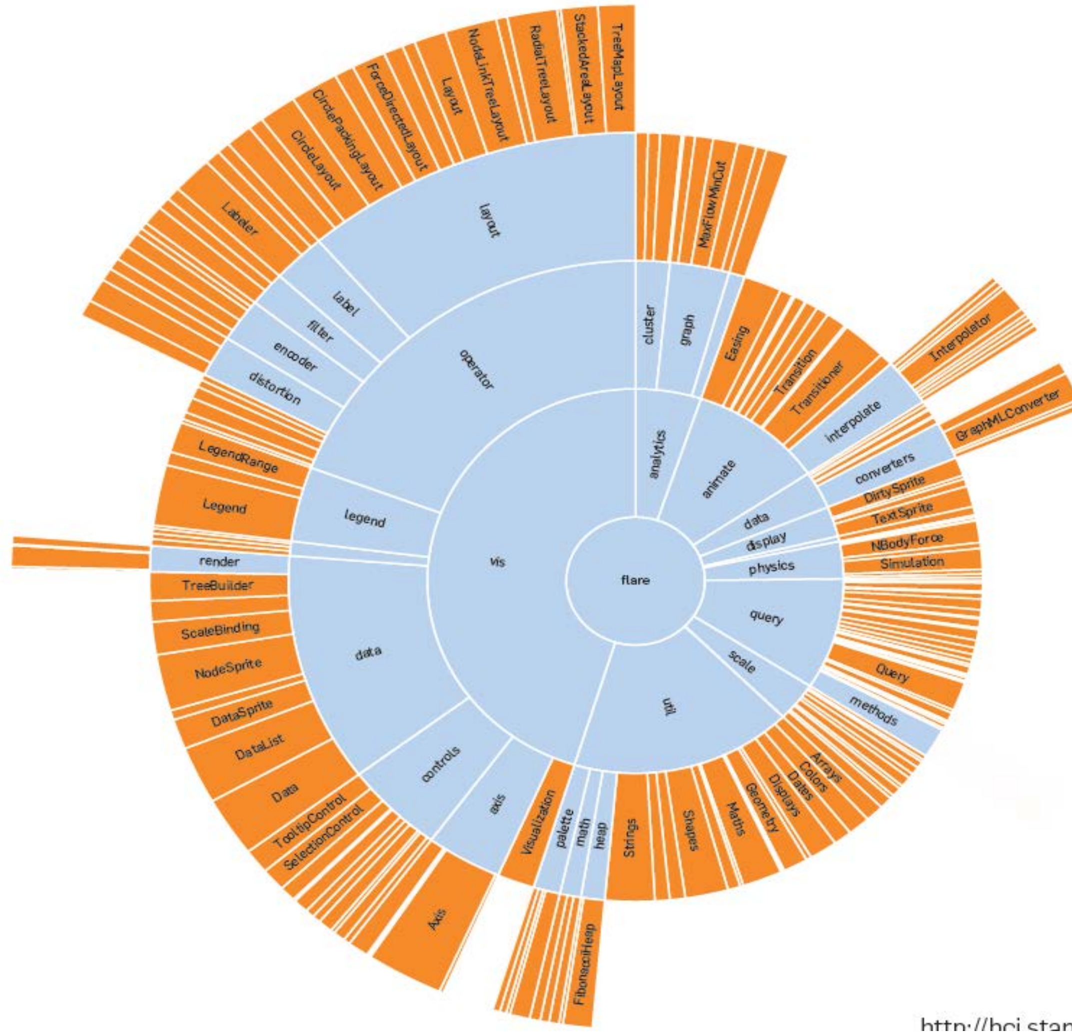
Adjacency Diagrams: Icicle Layout



- Space filling technique
- Nodes may show quantitative information, e.g. directory size
- Parent-child relationship by adjacency

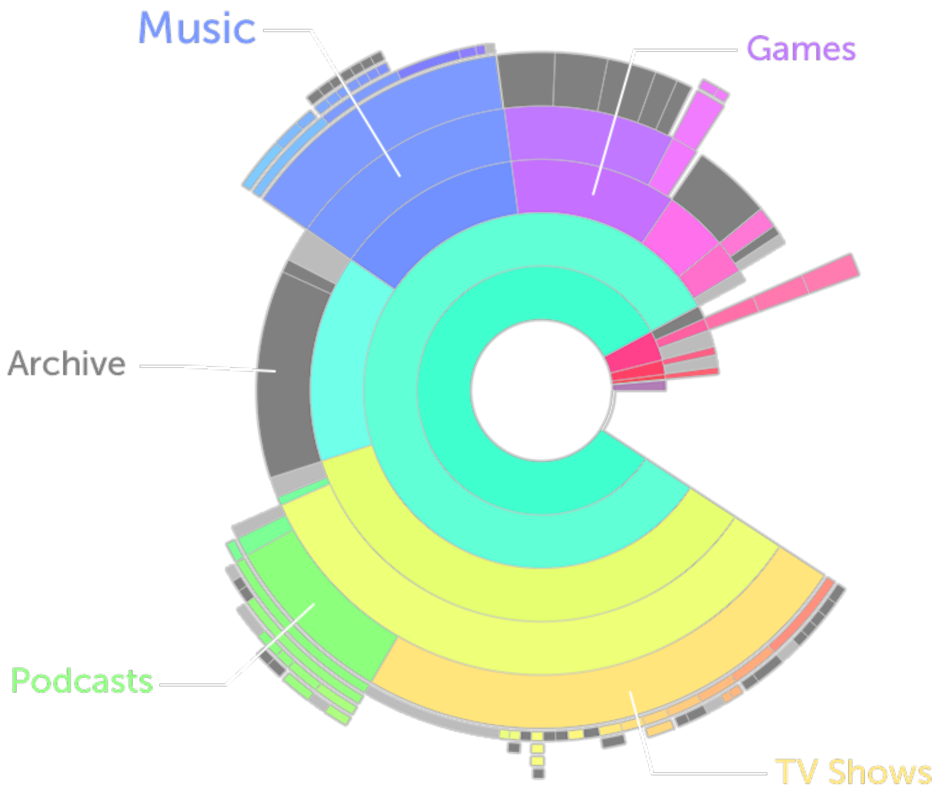
Adjacency Diagrams: Radial Layout

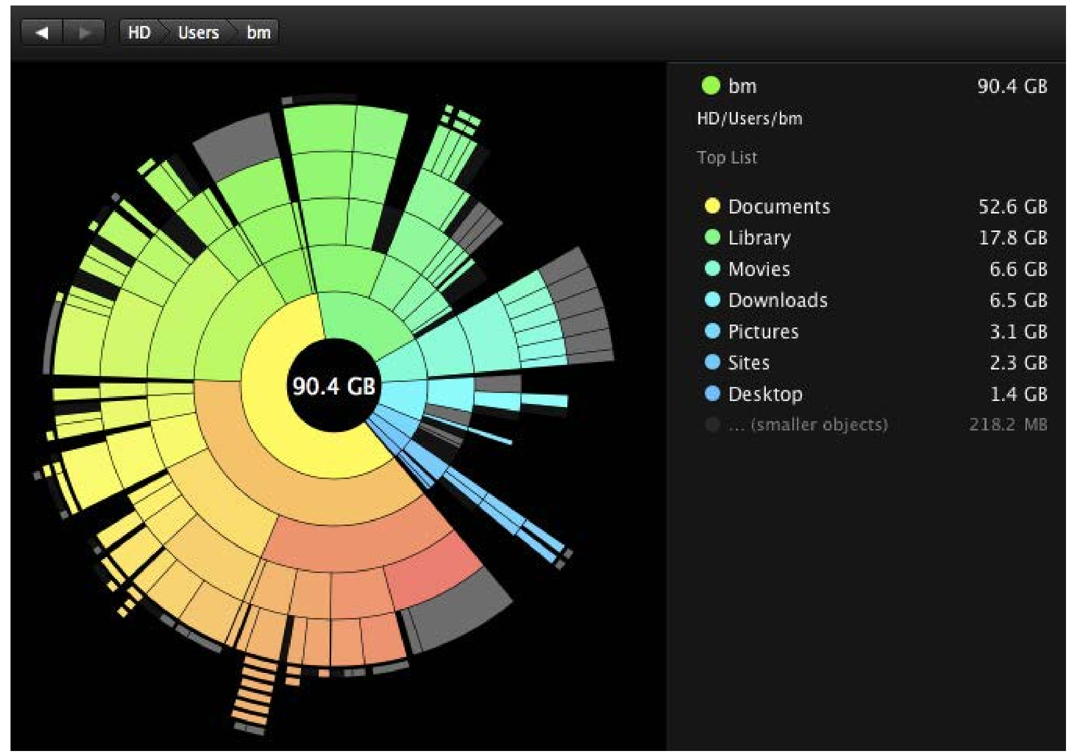
- Tree layout in polar coordinates
- Radial angle now corresponds to quantitative information rather than the area



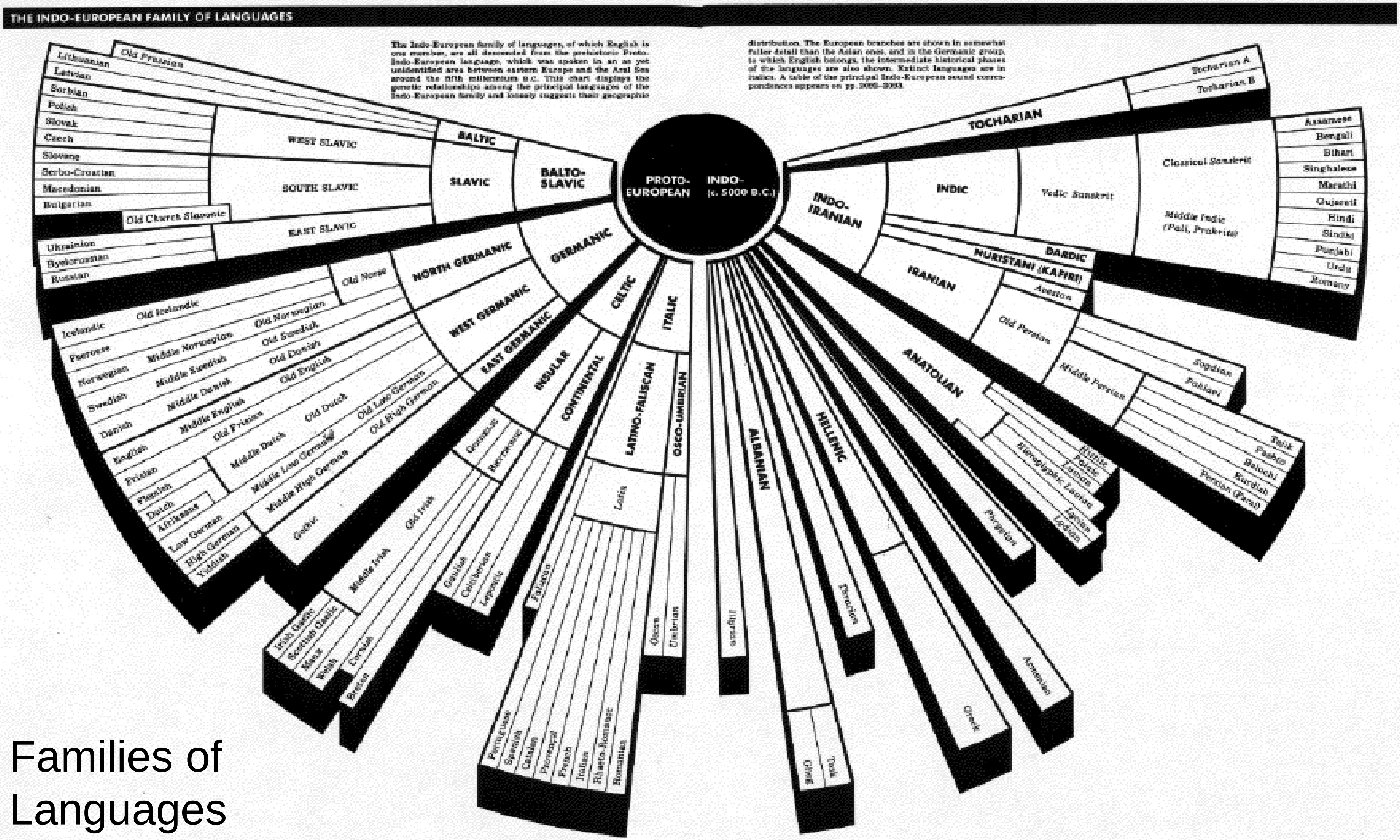
http://hci.stanfo

DaisyDisk





https://www.youtube.com/watch?v=9uOBTcXfON0



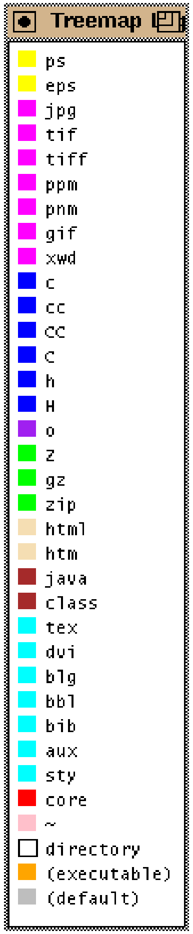
Families of Languages

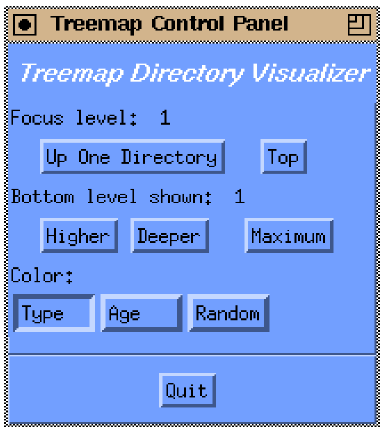
Appears in:

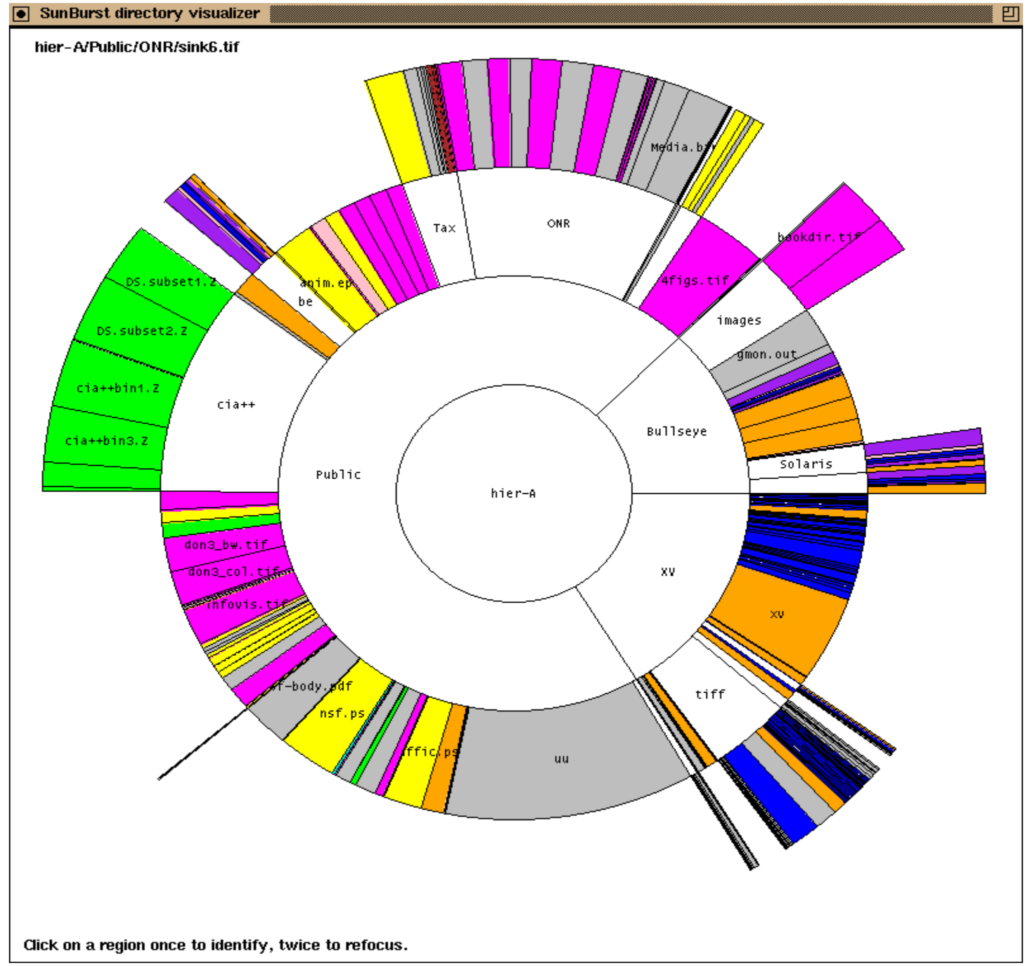
American Heritage Dictionary, 3rd Ed. Houghton Mifflin, 1992

SunBurst

VIDEO





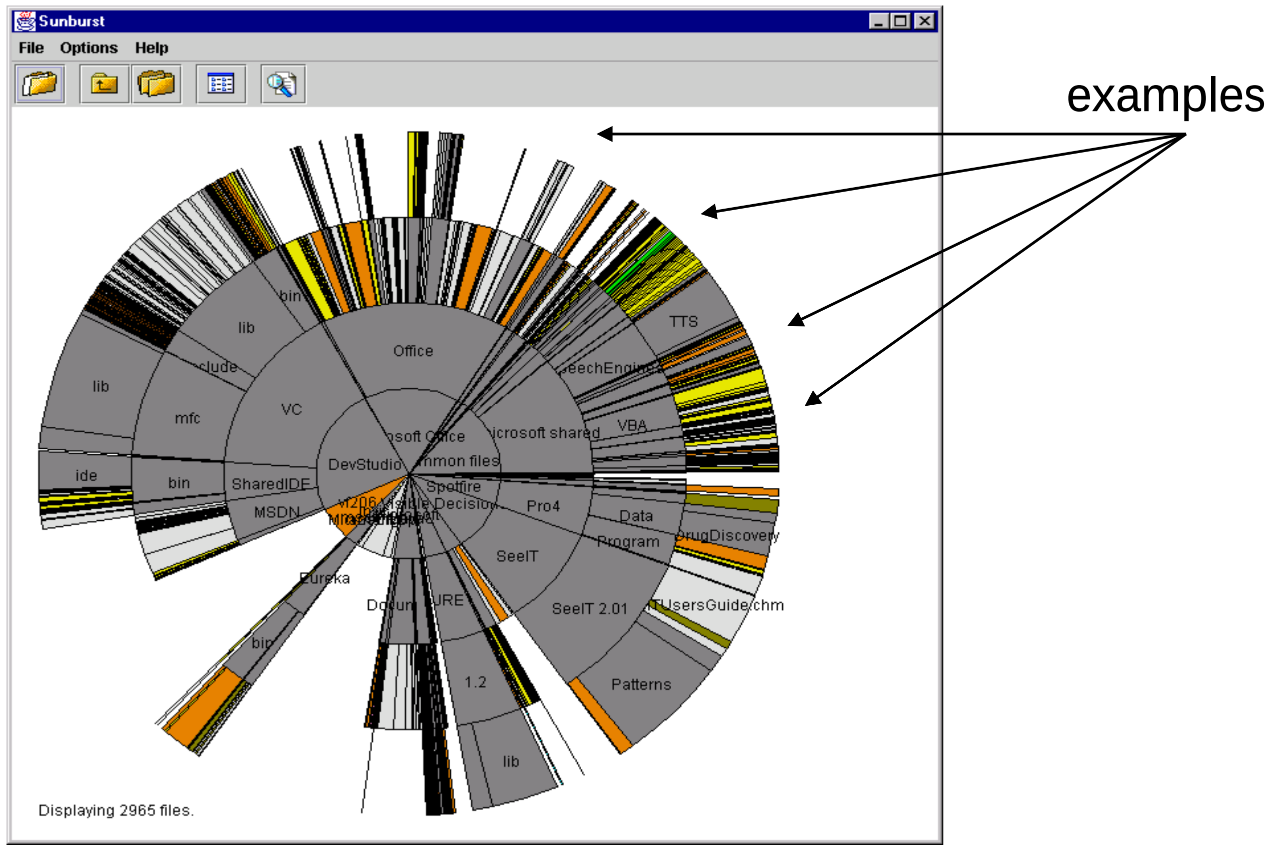


SunBurst

- Root directory at center, each successive level drawn farther out from center
- Sweep angle of item corresponds to size
- Color maps to file type or age
- Interactive controls for moving deeper in hierarchy, changing the root, etc.
- Double-click on directory makes it new root

SunBurst Negative

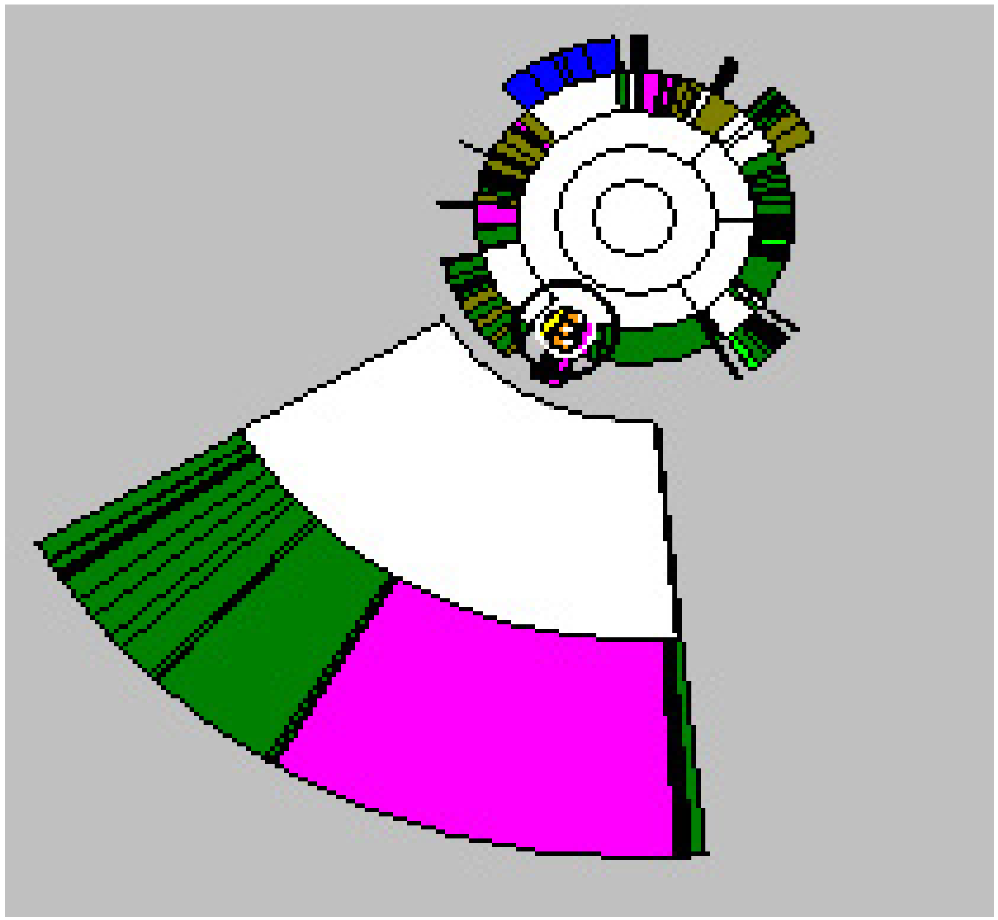
- In large hierarchies, files at the periphery are usually tiny and very difficult to distinguish



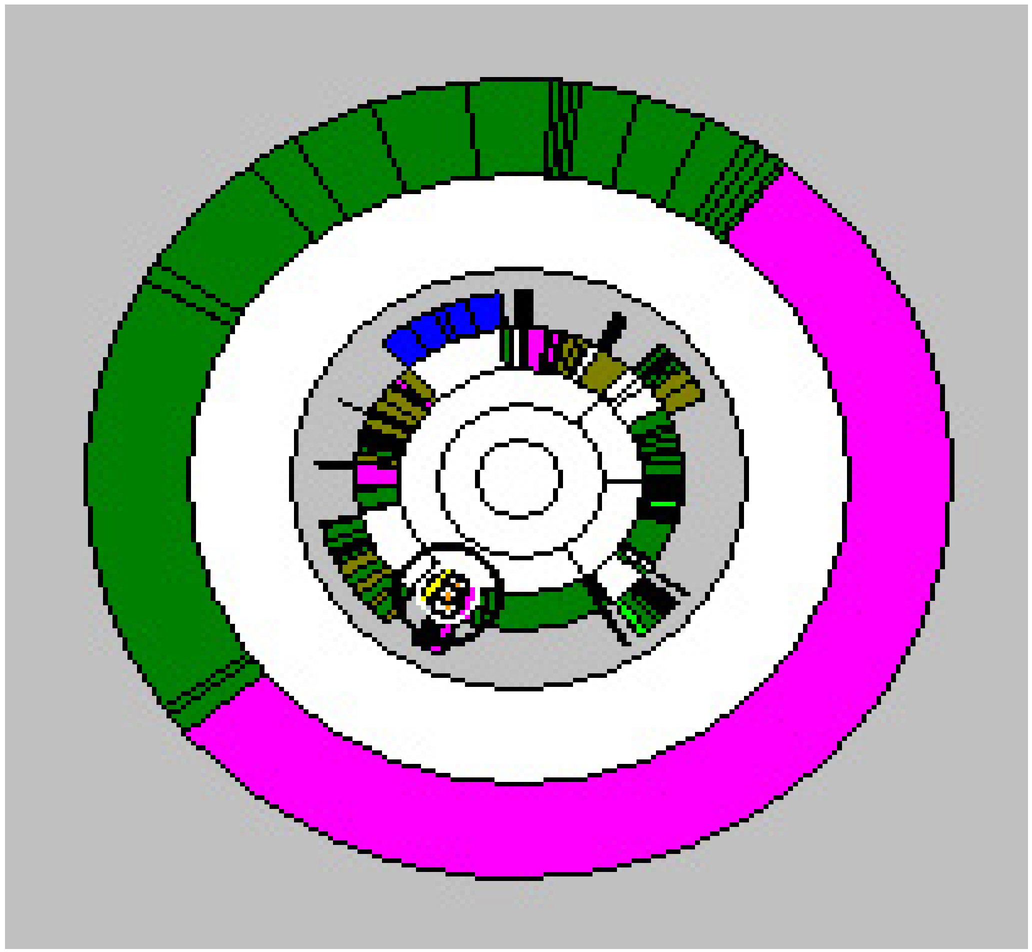
Sunburst: 3 Solutions

- Three visualization+navigation techniques developed to help remedy the shortcoming
  - Angular detail
  - Detail outside
  - Detail inside

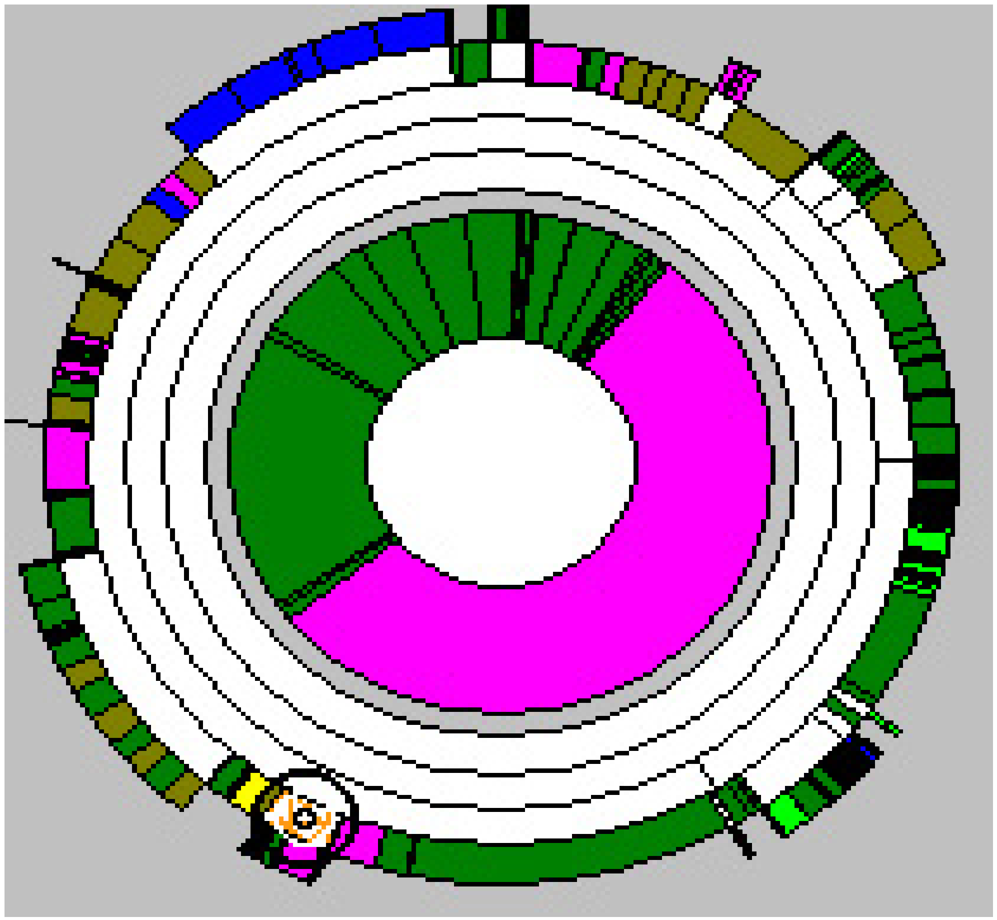
Design 1 - Angular Detail



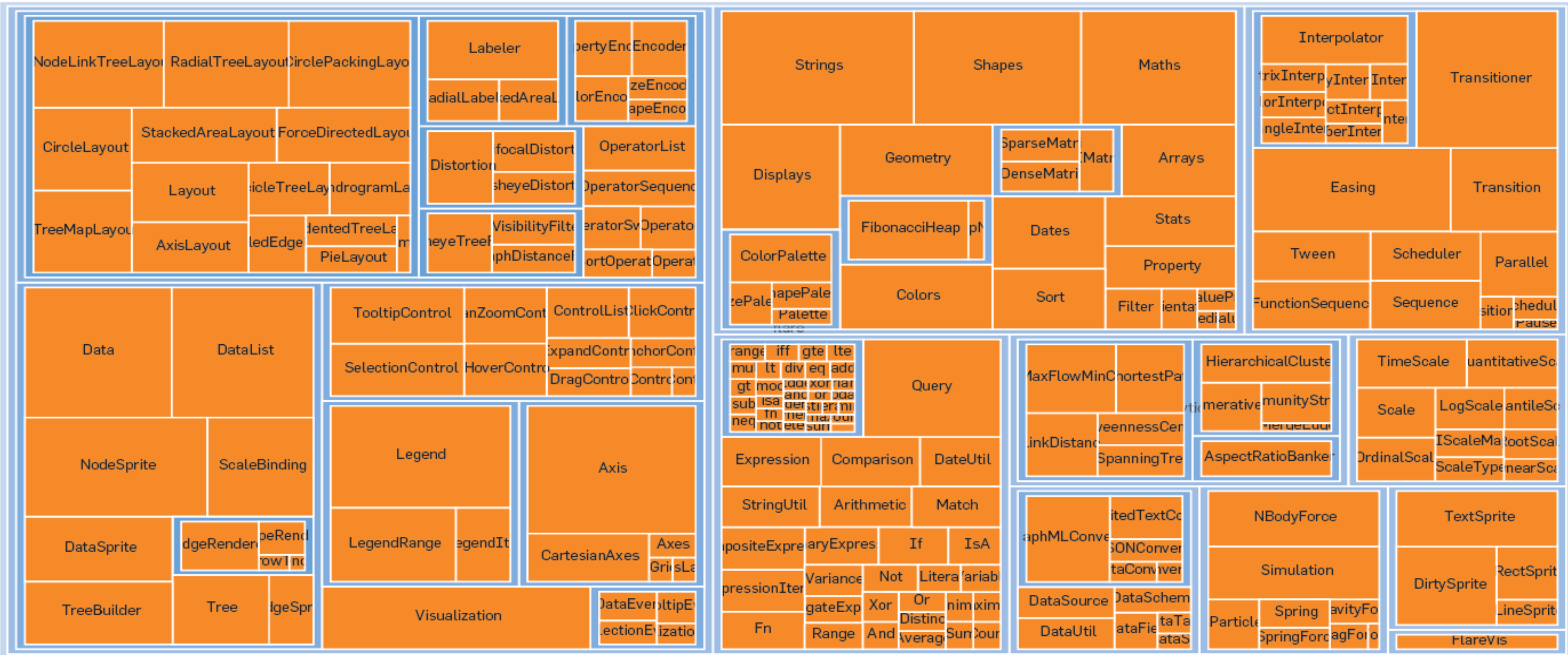
Design 2 - Detail Outside



Design 3 - Detail Inside

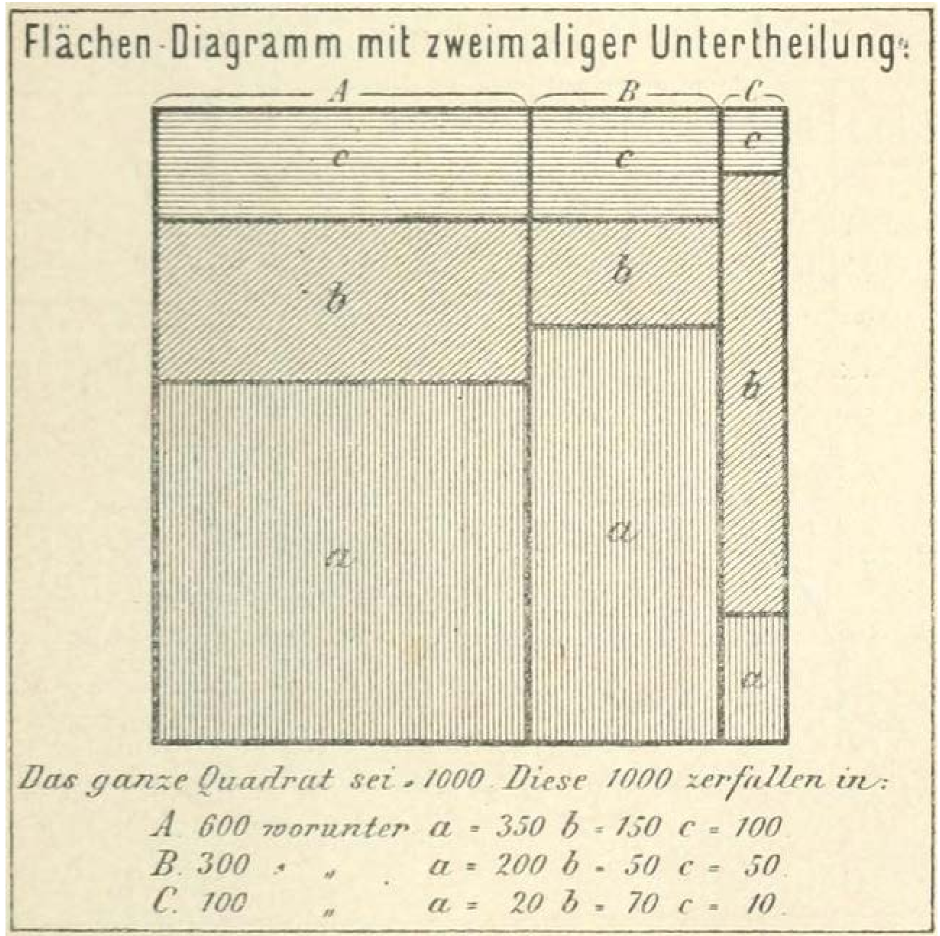


Enclosure Diagrams: Treemaps



http://hci.stanford.edu/jheer/files/zoo/ex/hierarchies/treemap.html
The Flare software package tree laid out into recursively subdivided rectangles. The area of each rectangle corresponds to the package's size.

First Treemaps

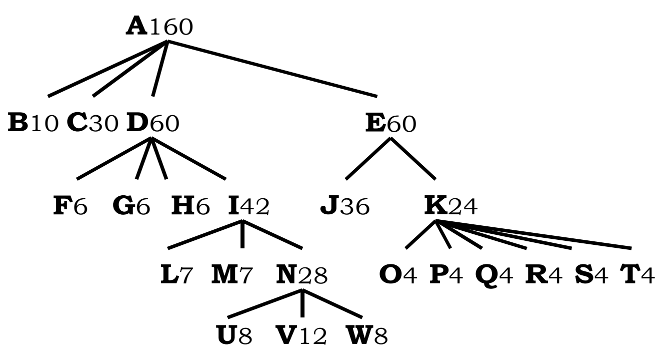


Die Gesetzmässigkeit im Gesellschaftsleben, statistische Studien
by Dr. Georg Mayr, Published 1877

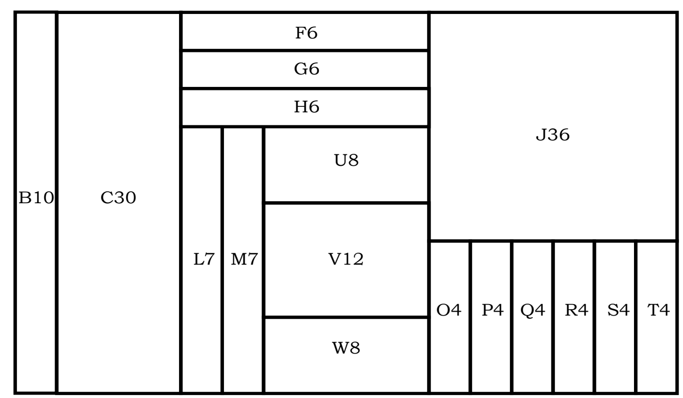
Treemaps

Reinvented by [JS 91, Shn 92, Joh 93]

- Space-filling representation
- Children are drawn inside their parent
- Classical slice-and-dice tiling approach
  - Alternate horizontal and vertical slicing at each successive level based on a data attribute such as file size, number of code lines, etc.
- Use area of partition to encode other variable of data items, e.g. use color for file age



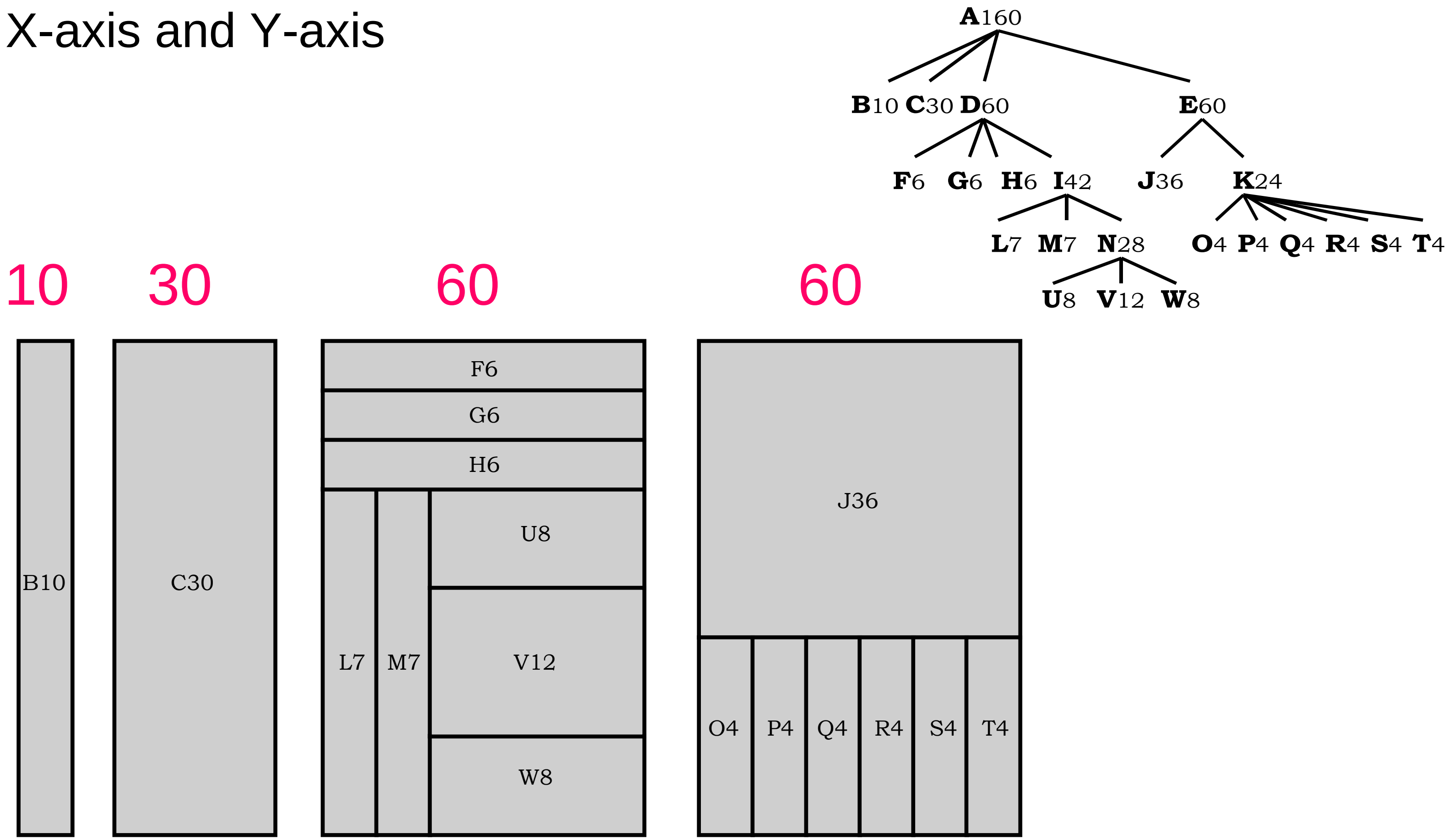
Node-Link Diagrams



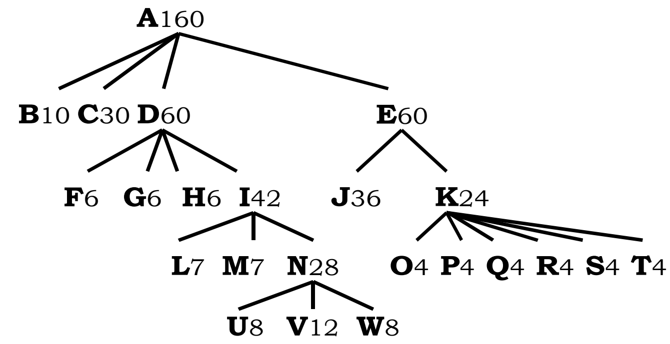
Containment / Enclosure

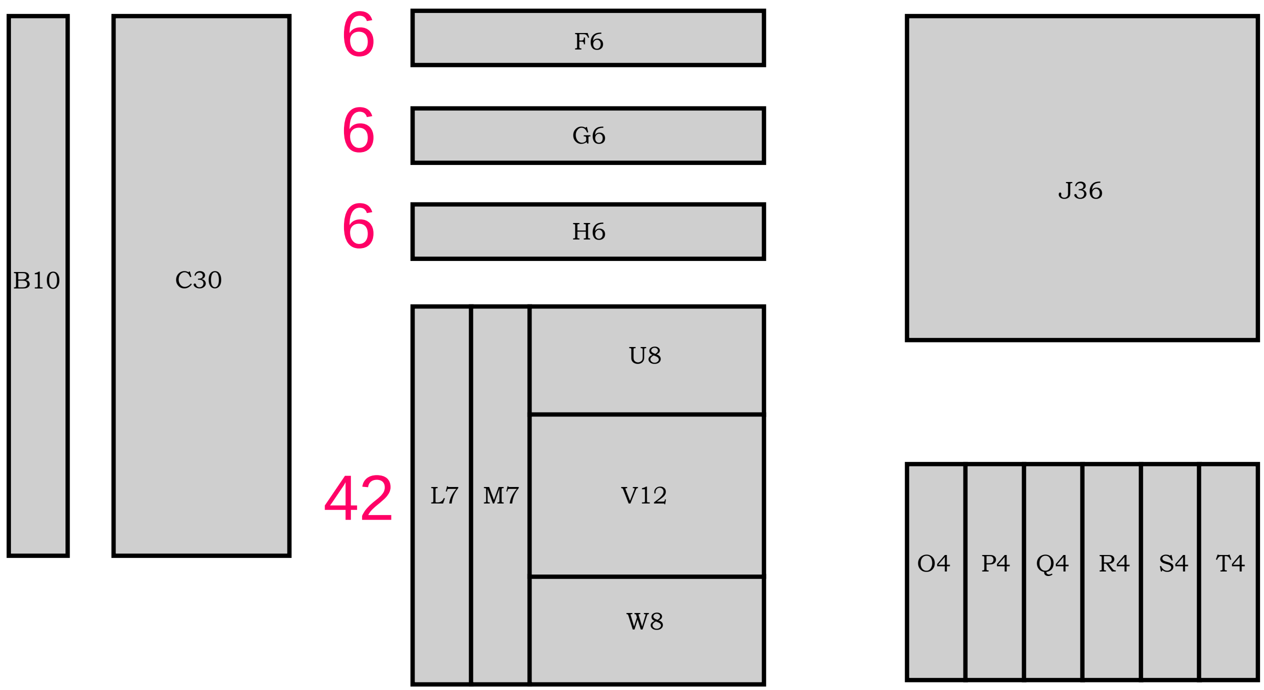
Classic Slice-and-Dice Tiling Algorithm

- Tree nodes include summed numeric value
- Here: Area shows numeric value
  - Alternate X-axis and Y-axis

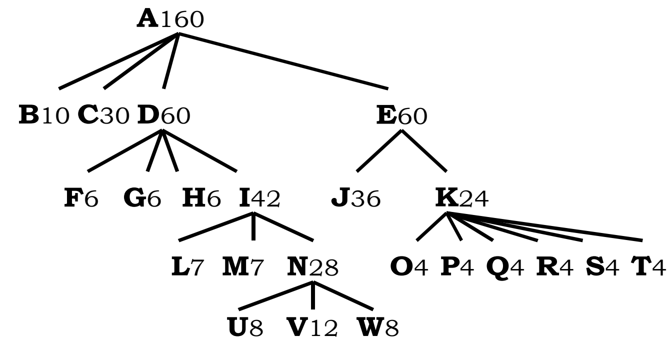


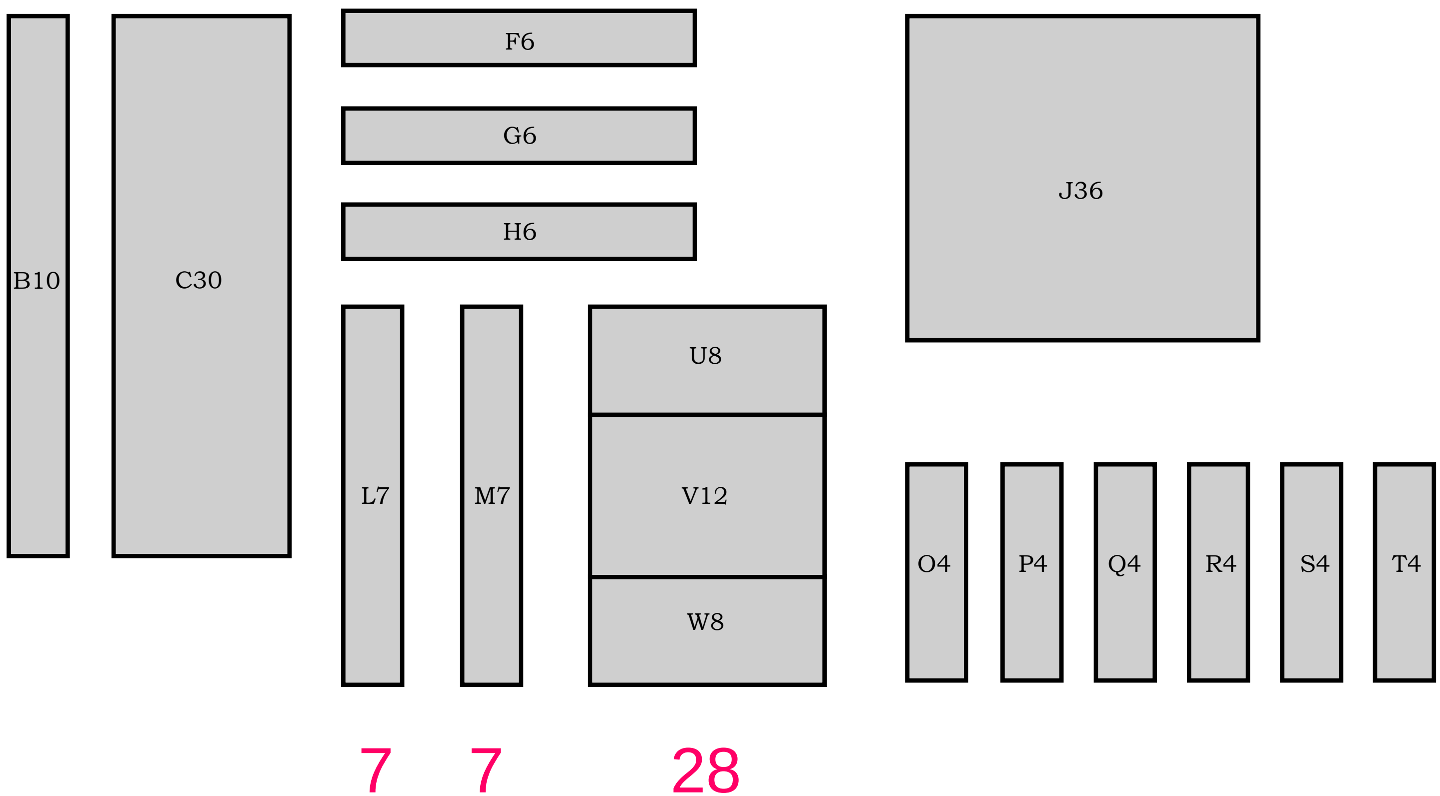
Y axis


A160
B10 C30 D60
E60
F6 G6 H6 I42
J36 K24
L7 M7 N28
O4 P4 Q4 R4 S4 T4
U8 V12 W8

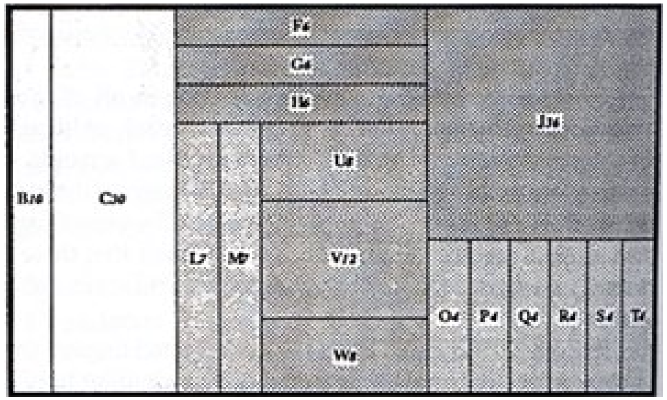

B10
C30
6
F6
6
G6
6
H6
42
L7
M7
U8
V12
W8
J36
O4
P4
Q4
R4
S4
T4

X axis

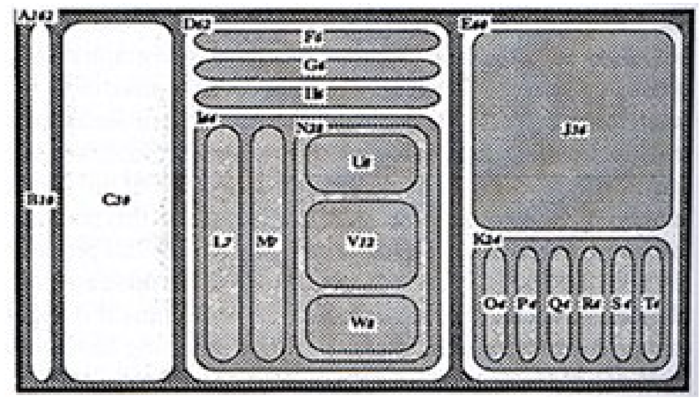

A160
B10 C30 D60
E60
F6 G6 H6 I42
J36 K24
L7 M7 N28
O4 P4 Q4 R4 S4 T4
U8 V12 W8


F6
G6
H6
J36
B10
C30
L7
M7
U8
V12
W8
O4
P4
Q4
R4
S4
T4
7
7
28

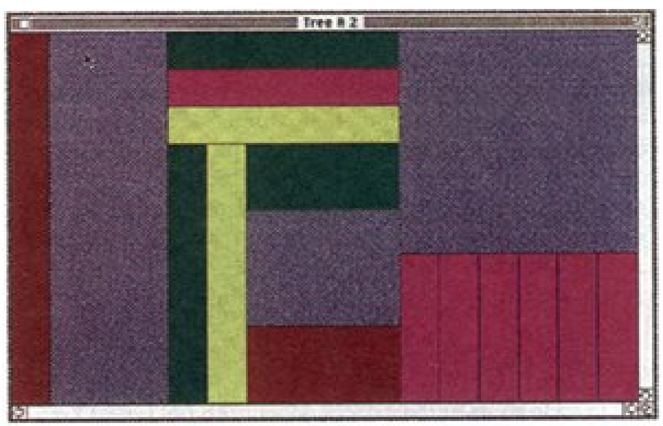
Nested vs. Non-nested

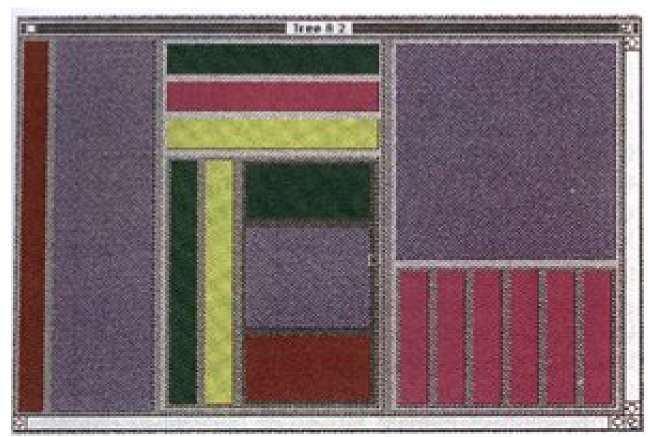


Non-nested Tree-Map

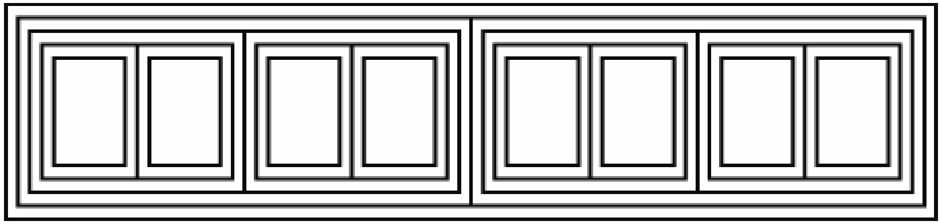


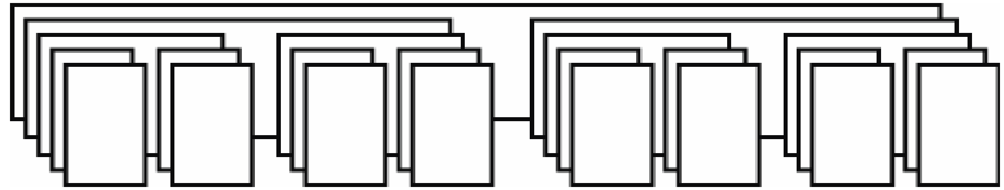
Nested Tree-Map





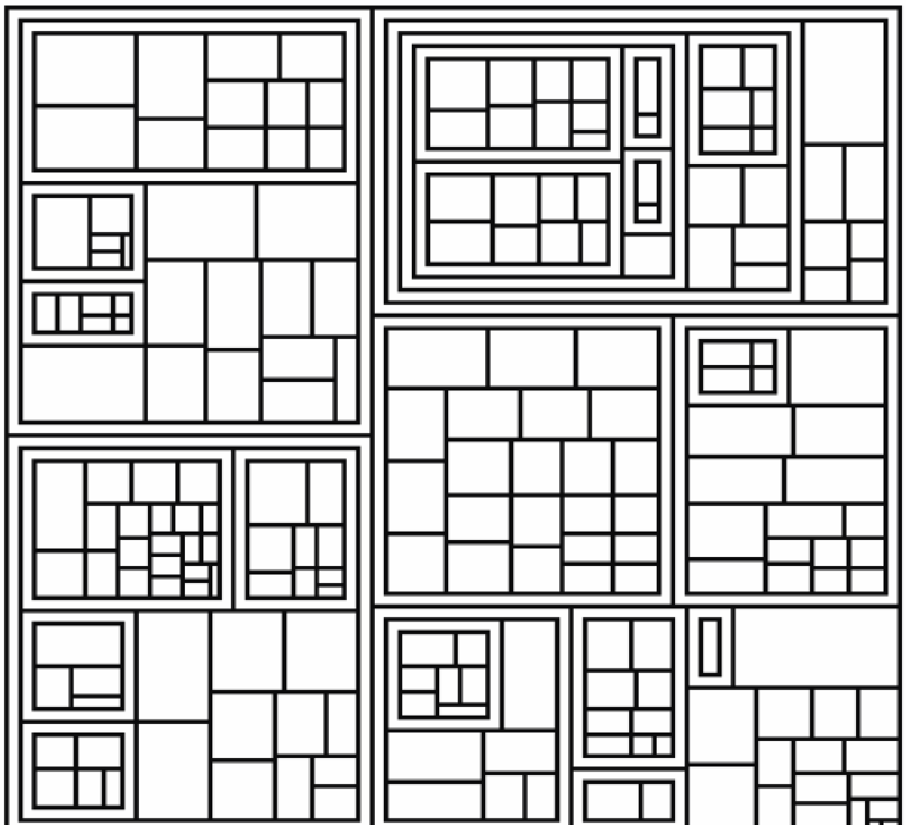
Cascaded Treemaps

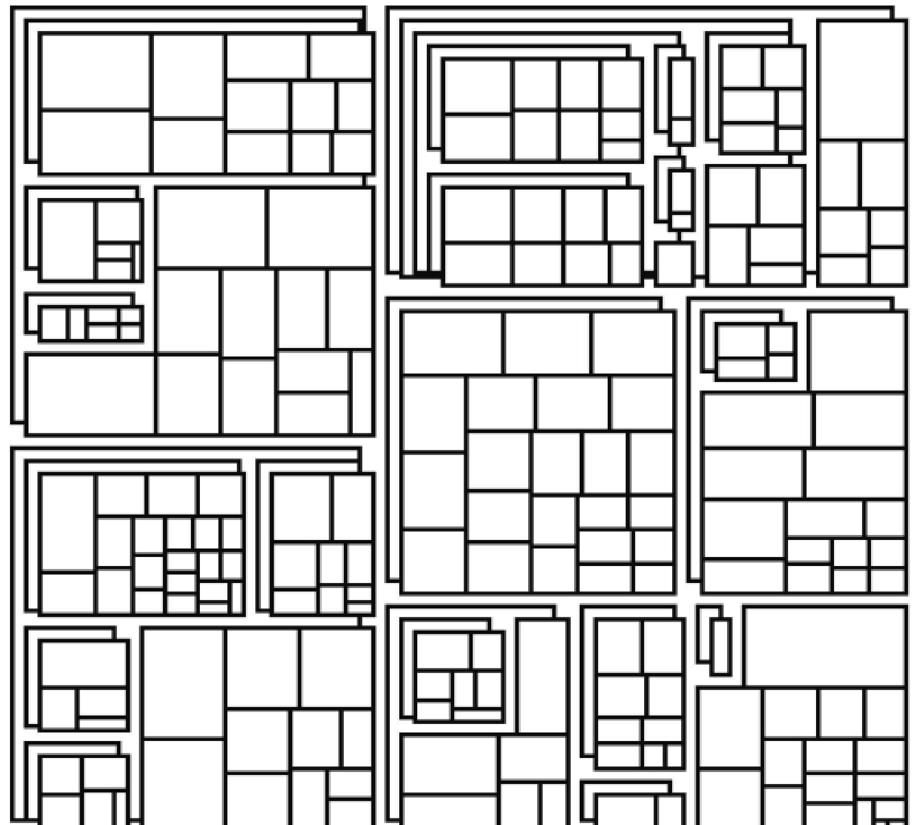




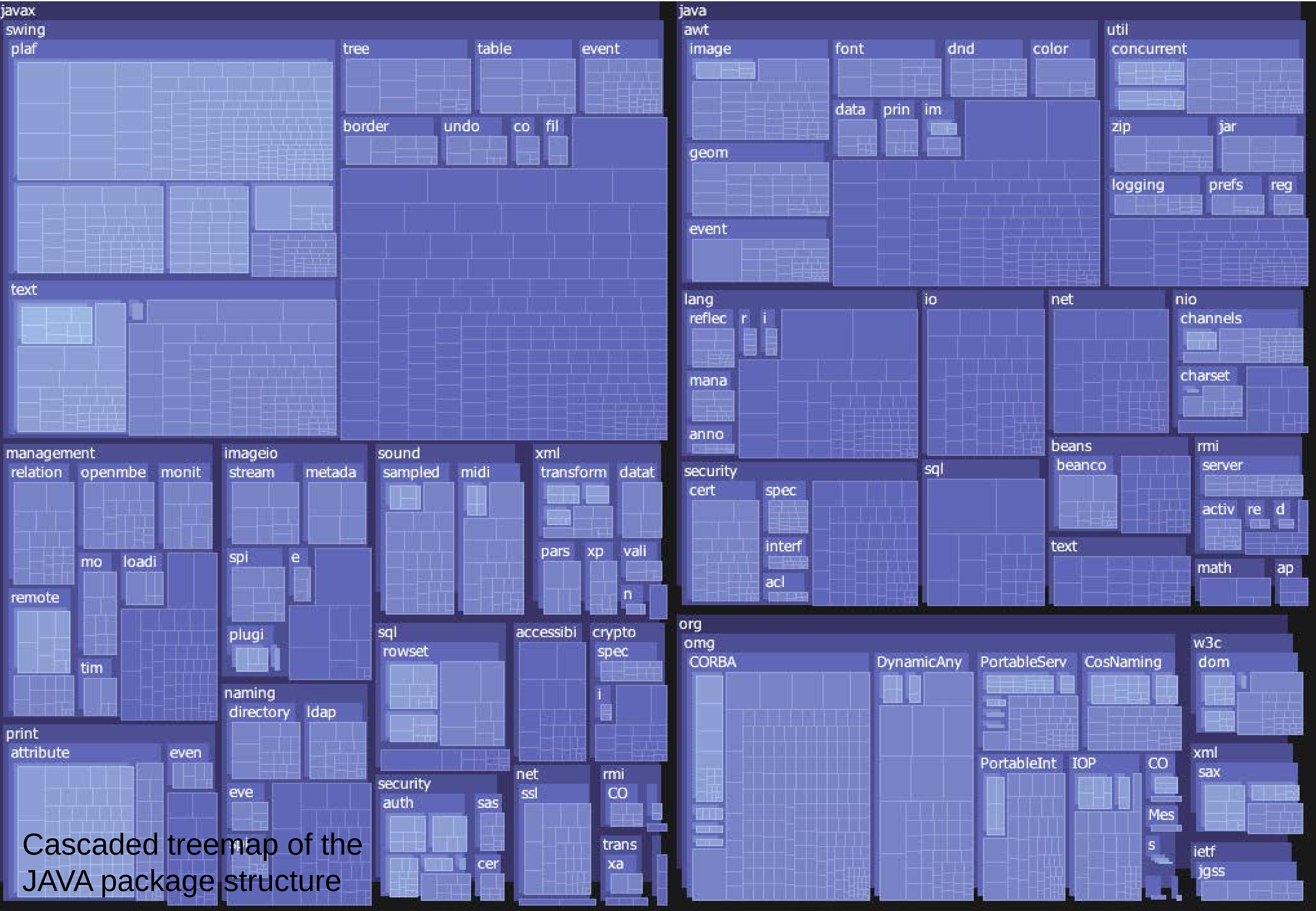
Nested layout

Cascaded layout



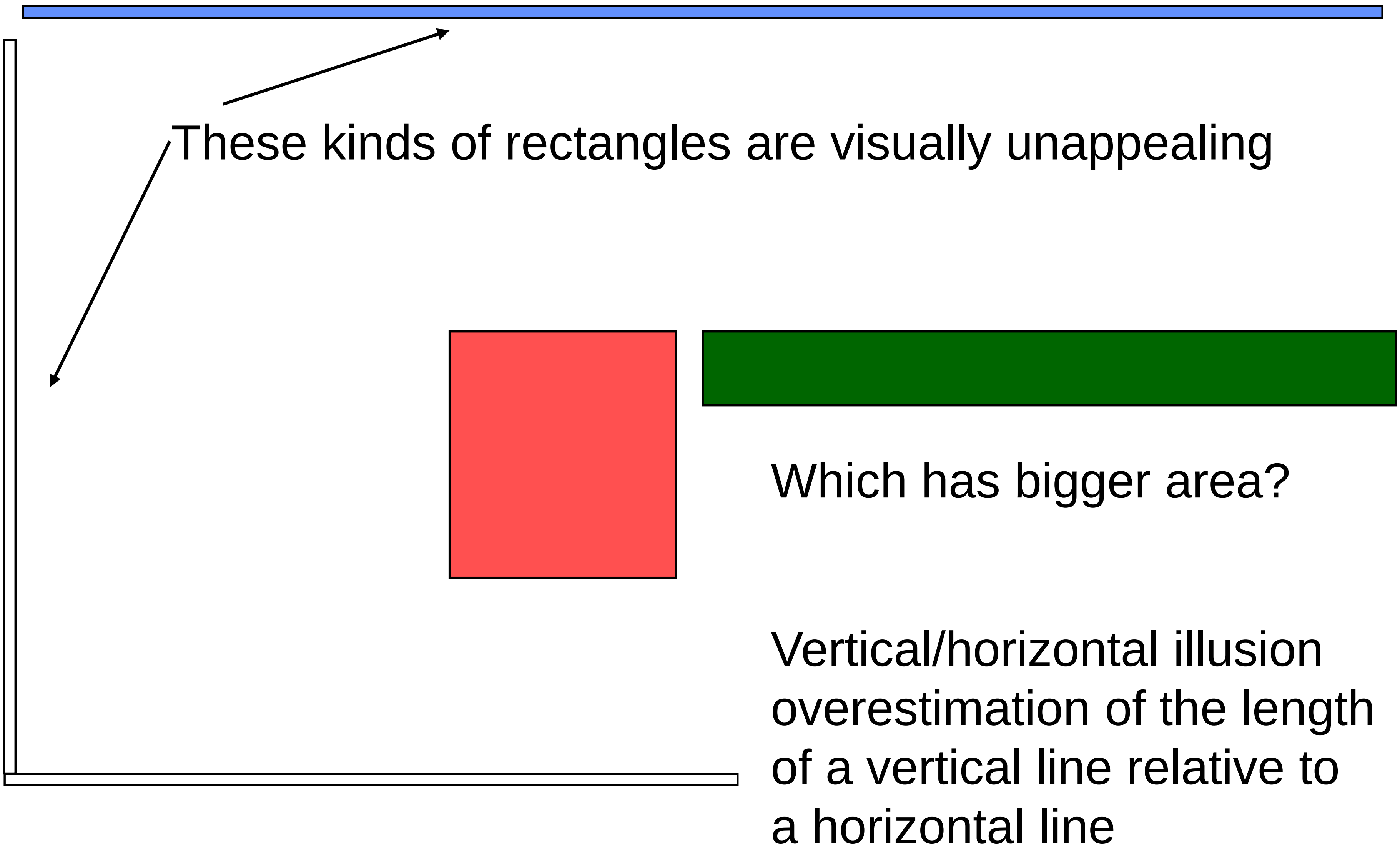


Hao Lü and James Fogarty. 2008. Cascaded treemaps: examining the visibility and stability of structure in treemaps. In *Proceedings of graphics interface 2008* (GI '08). 259-266.



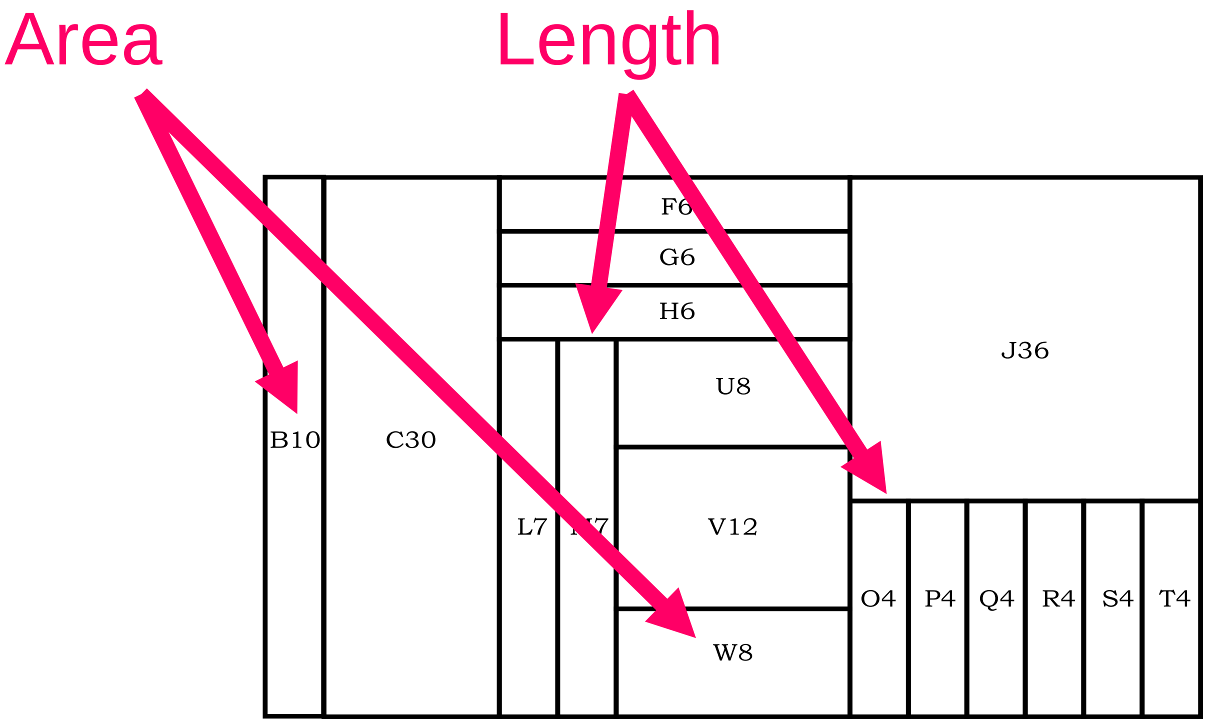
Cascaded treemap of the JAVA package structure

Aspect ratios



Comparing Across Levels

- Length or area?



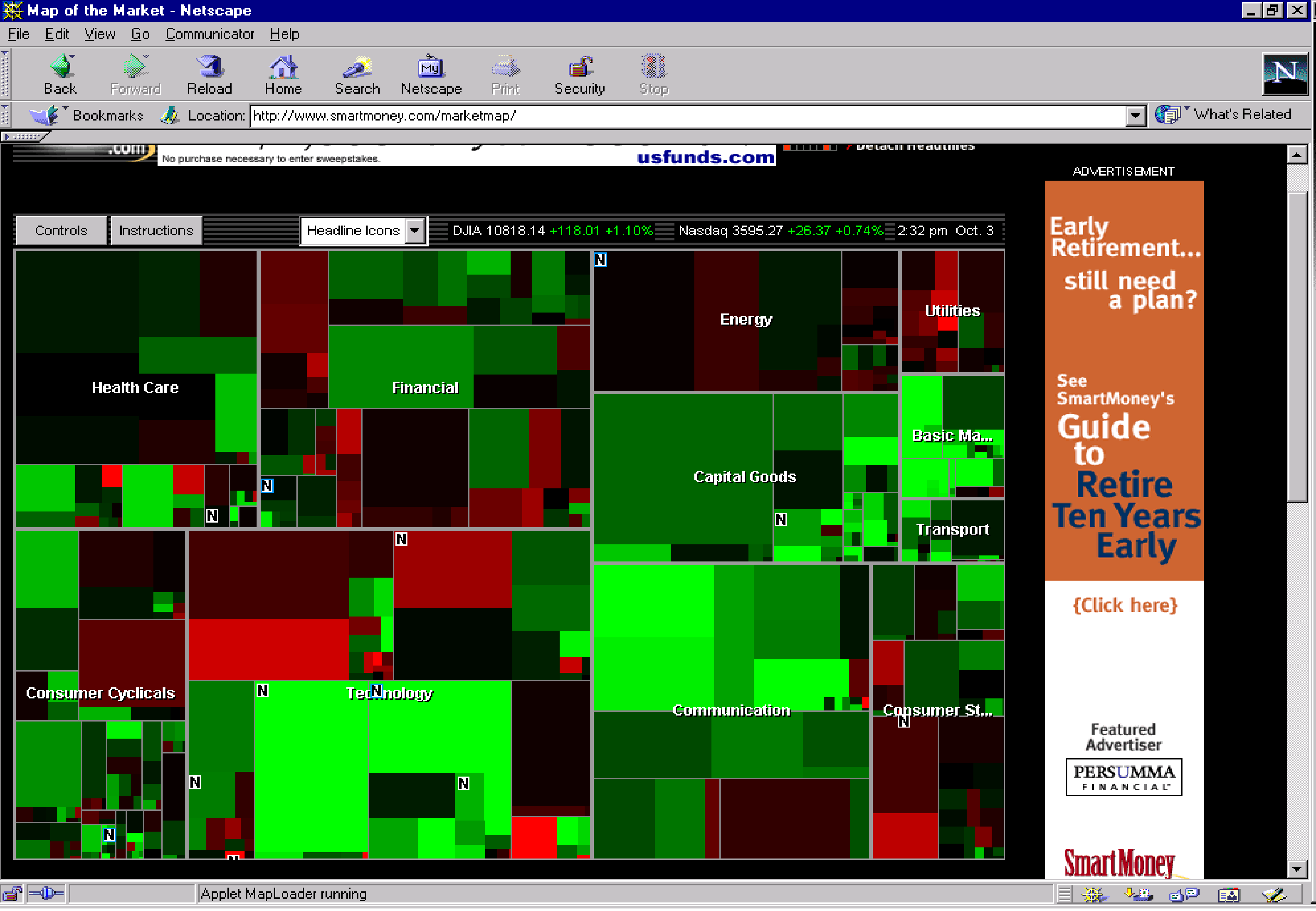
Treemap Affordances

- Good representation of two attributes beyond node-link: color and area
- Not as good at representing structure
  - What happens if it’s a perfectly balanced tree of items all the same size?
  - Also can get long-thin aspect ratios
  - Borders help on smaller trees, but take up too much area on large, deep ones

Nice summary of techniques at
http://www.cs.umd.edu/hcil/treemap-history/index.shtml

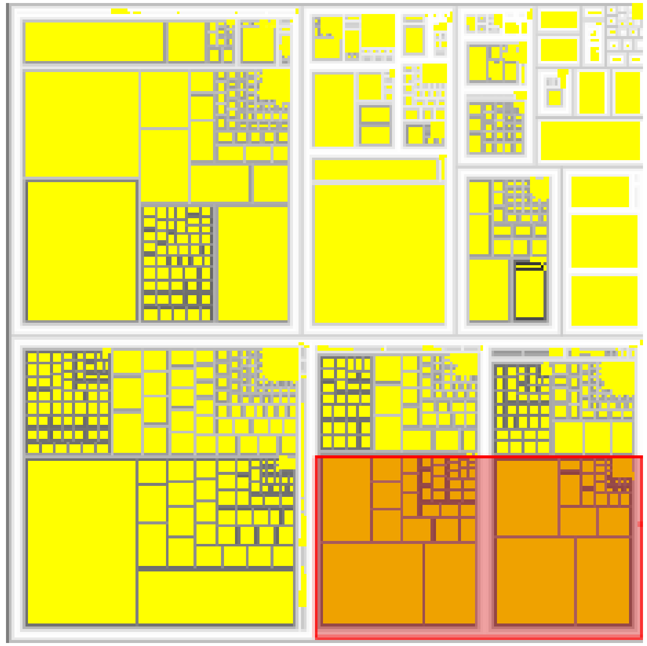
Example of a “Square” Treemap

- Map of the Market
  - Illustrates stock changes during a day
  - Basic squarifying algorithm with some hand tweaking
  - Takes advantage of shallow hierarchy
  - Hierarchy of categories (here market segments) – leaves are actual companies


Map of the Market - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.smartmoney.com/marketmap/
What's Related
No purchase necessary to enter sweepstakes.
usfunds.com
ADVERTISEMENT
Controls
Instructions
Headline Icons
DJIA 10818.14 +118.01 +1.10%
Nasdaq 3595.27 +26.37 +0.74%
2:32 pm Oct. 3
Health Care
Financial
Energy
Utilities
Basic Ma...
Capital Goods
Transport
Consumer Cyclicals
Technology
Communication
Consumer St...
Early Retirement... still need a plan?
See SmartMoney's Guide to Retire Ten Years Early
{Click here}
Featured Advertiser
PERSUMMA FINANCIAL
SmartMoney
Applet MapLoader running

Treemaps

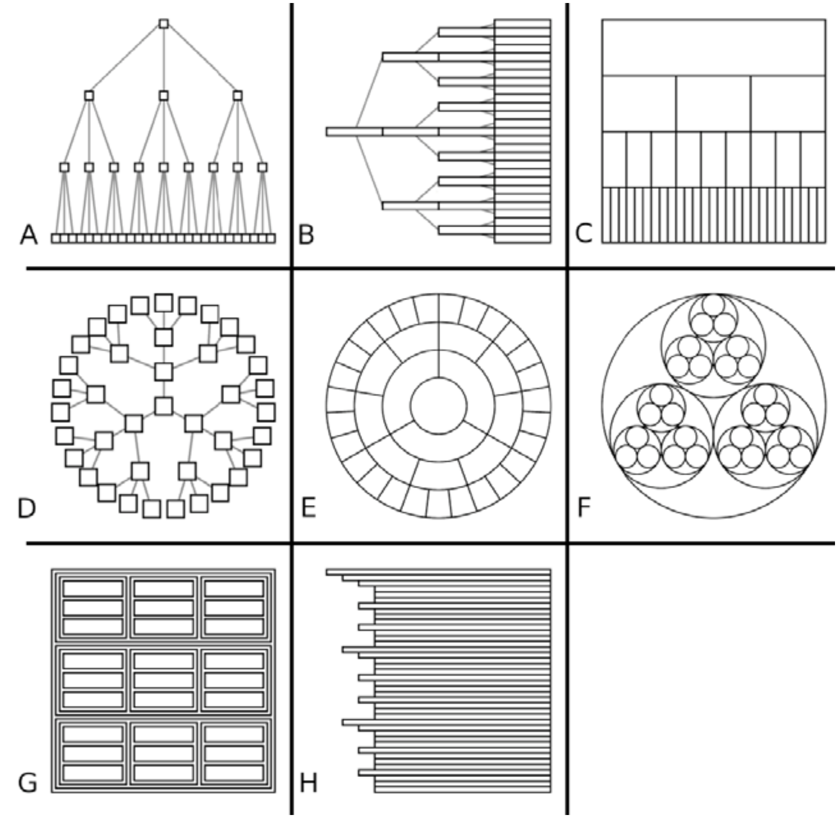
- Data
  - Hierachical structure with at least one quantative attribute at leaf nodes
- Encoding
  - Area containment marks for hierarchical structure
  - Rectilinear orientation
  - Size encodes quantative attribute
- Tasks
  - Query attribute at leaf nodes
- Scalability
  - 1M leaf nodes



http://tulip.labri.fr/Documentation/3_7/userHandbook/html/ch06.html

Comparison of Tree Drawing Techniques

- Data shown
  - Node relationships
  - Tree depth
  - Sibling order
- Design choices
  - Visualization of links by adjacency or explicit link drawing or containment
  - Rectilinear vs radial layout
- Considerations
  - Encoding of other node and link information
  - Information density?
    - Avoid wasting space



[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

End

Visualization

Graphs and Networks

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Acknowledgements

- Based on lectures by
  - Jock D. Mackinlay, Xerox PARC
  - Chris North, Virginia Tech
  - John Stasko, Georgia Tech, College of Computing
  - Harald Reiterer, Universität Konstanz
  - Daniel A. Keim, Universität Konstanz
  - Jeff Heer, Stanford University
  - Also based on slides and talks by T. Munzner and her book: Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.

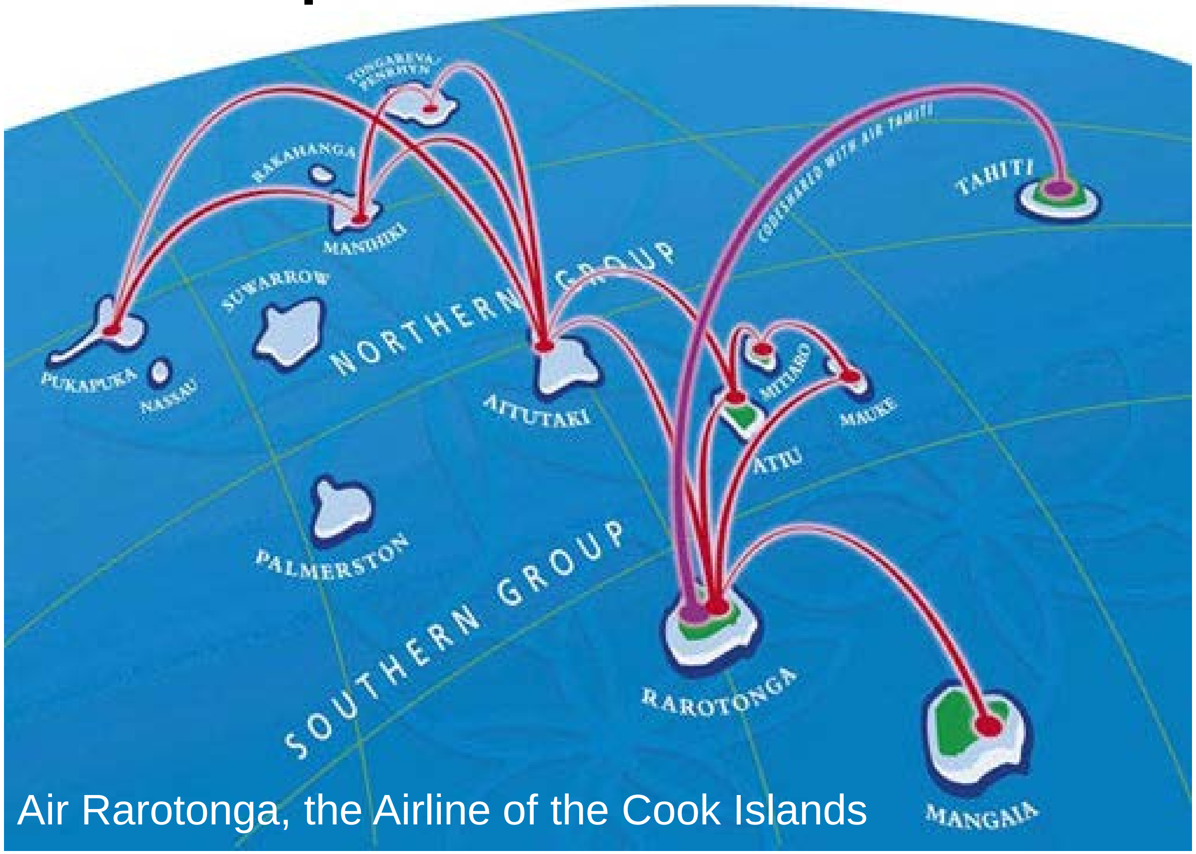
Data Space

- Tables: Multi-dimensional databases,…
- Trees/Hierarchies: directories,…
- Networks/Graphs: web, communications,…
- Time Series
- Maps
- Text: Unstructured data, e.g. document collections: digital libraries,…

More Infos

- Tamara Munzner's talk on node-link diagrams
  http://video.google.com/videoplay?docid=-6229232330597040086
- **Mike Bostock - D3**
  http://vimeo.com/29458354
- Interesting toolkit:
  http://blog.thejit.org/javascript-information-visualization-toolkit-jit/
- Heer, J., Bostock, M., and Ogievetsky, V. 2010. A tour through the visualization zoo. Commun. ACM 53, 6 (Jun. 2010), 59-67
  http://doi.acm.org/10.1145/1743546.1743567
- Draper, G. M., Livnat, Y., and Riesenfeld, R. F. 2009. A Survey of Radial Methods for Information Visualization. IEEE Transactions on Visualization and Computer Graphics 15, 5 (Sep. 2009), 759-776
  http://www.computer.org/portal/web/csdl/doi/10.1109/TVCG.2009.23
- Herman et al. Graph Visualization and Navigation in Information Visualization: A Survey. IEEE Transactions on Visualization and Computer Graphics 6, 1 (Jan. 2000), 24-43
  http://www.computer.org/portal/web/csdl

Graphs and Networks



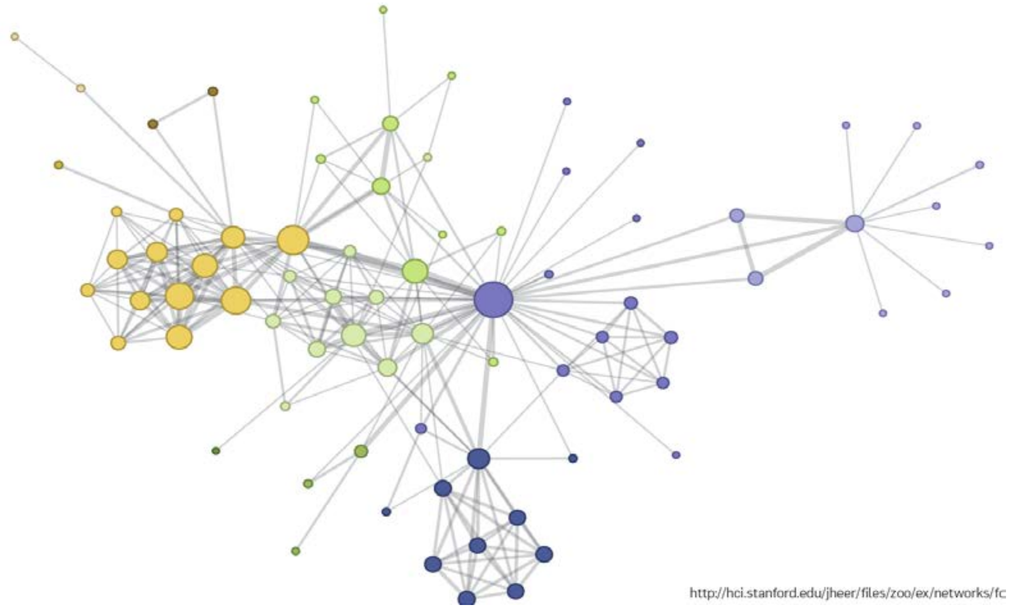
Air Rarotonga, the Airline of the Cook Islands

- Connections in the real world
  - Flight network
  - Professional colleagues
  - Electrical system
  - Road system …
- Modeling of a connected set as a Graph

Force-Directed Layout

- Graph as a physical system
- Nodes are charged particles
  >> repel each other
- Links are dampened springs
  >> pull related nodes together
- (Approximate) physical simulation for computing node positions.
- Interactivity allows to direct the layout and move nodes around to disambiguate links (multiple links on top of each other)

- Mike Stock: http://vimeo.com/29458354

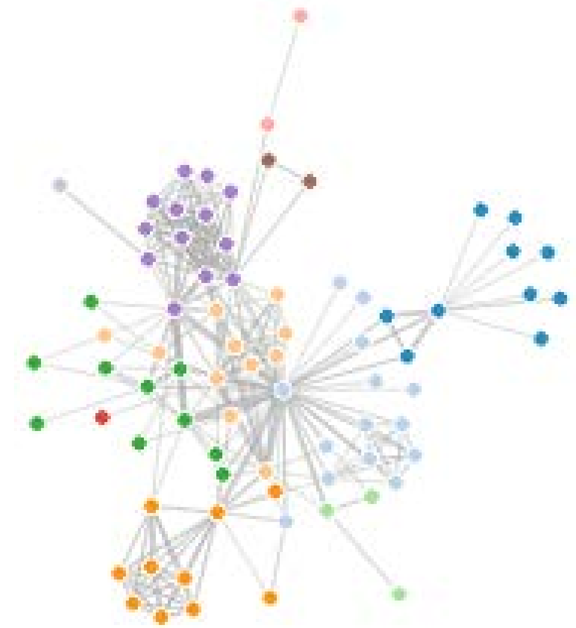


The network of character co-occurrence in the chapters of Victor Hugo's classic novel, *Les Misérables. Node colors depict* cluster memberships computed by a community-detection algorithm.
Interactive Demo
http://hci.stanford.edu/jheer/files/zoo/ex/networks/force.html

[Efficient and high quality force-directed graph drawing. Hu. The Mathematica Journal 10:37–71, 2005.]

Force-Directed Layout

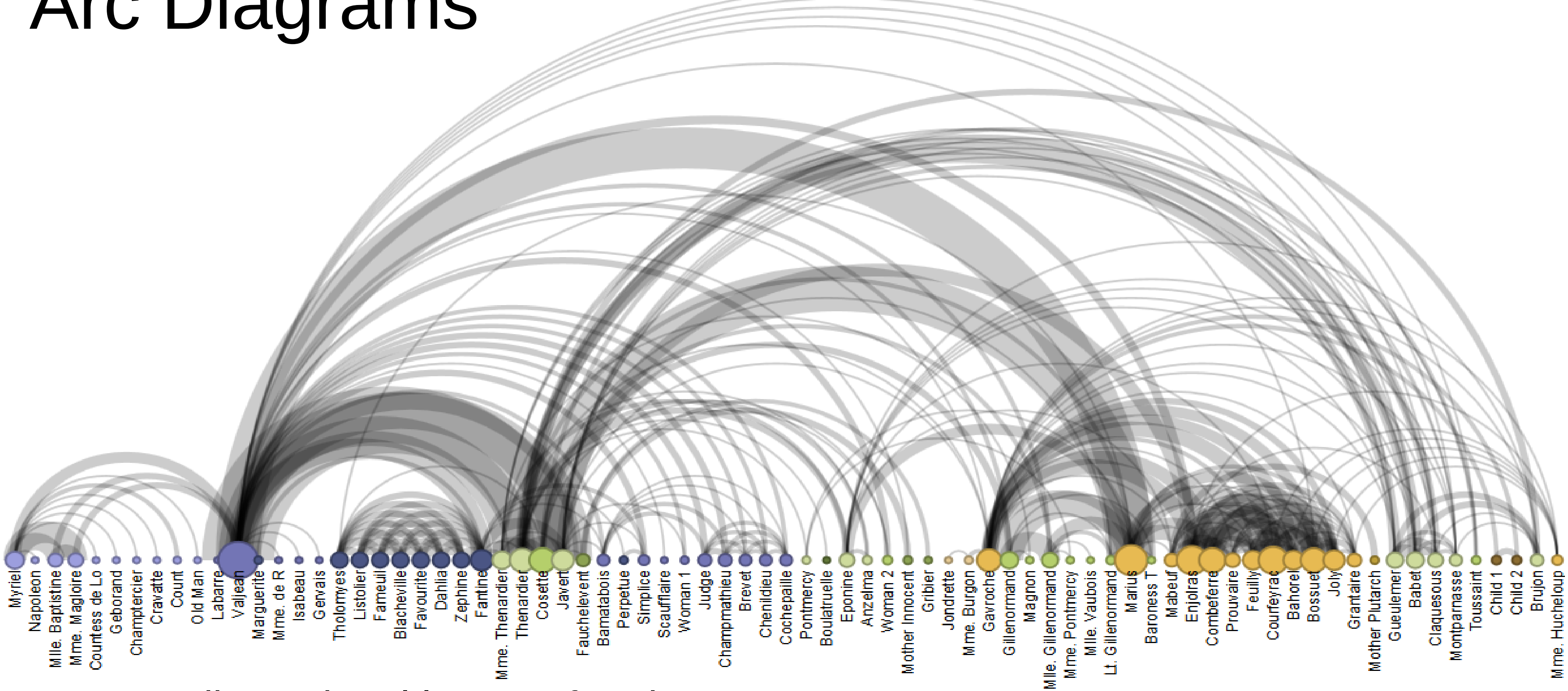
- Visual encoding
  - Link: line marks, node: point marks
- Considerations
  - Spatial position: no meaning directly encoded
    - Available to minimize crossings
  - Proximity semantics?
    - Sometimes meaningful
    - Sometimes arbitrary, artifact of layout algorithm
    - Perceptual conflict: long edges more visually salient than short
- Tasks
  - Explore topology; locate paths, clusters
- Scalability
  - Node/edge density E < 4N
  - Nodes, edges: 1K-10K
  - Hairball problem eventually hits



http://mbostock.github.com/d3/ex/force.html

Arc Diagrams

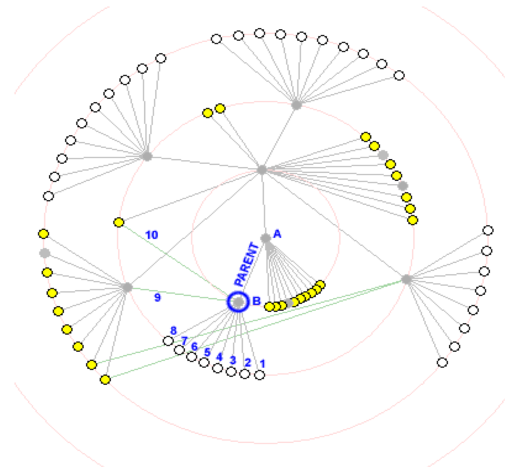
http://hci.stanford.edu/jheer/files/zoo/ex/networks/arc.html
Les Misérables character interaction.



- One-dimensional layout of nodes
- Circular arcs represent links
- Weighted graph: arc thickness related to number of co-occurences
- Seriation: sorting nodes to reveal cluster structure (http://innar.com/Liiv_Seriation.pdf)
  - Good ordering of nodes allows to identify cliques and bridges
- Node attributes displayed alongside nodes

Visualization of a peer-to-peer network: Gnutellavision

- Radial layout which focuses on a central node
  - Hosts (with color for status and size for number of files)
  - Nodes with closer network distance from focus on inner rings
  - Queries shown; can trace queries
- Gnutellavision as exploratory tool
  - Very few hosts share many files
  - Uneven propagation of queries
  - Qualitative assessment of queries (simple)



- Paper: Animated Exploration of Dynamic Graphs with Radial Layout, Yee, Fisher, Dhamija, Hearst, InfoVis 2001
- Video on our course web site

Video

Animation in Gnutellavision

- Goal of animation is to help maintain context of nodes and general orientation of user during refocus
- Transition paths
  - Linear interpolation of polar coordinates
  - Node moves in arc not straight line
  - Moves along circle if not changing levels (like great circles on earth)
  - Spirals in or out to next ring

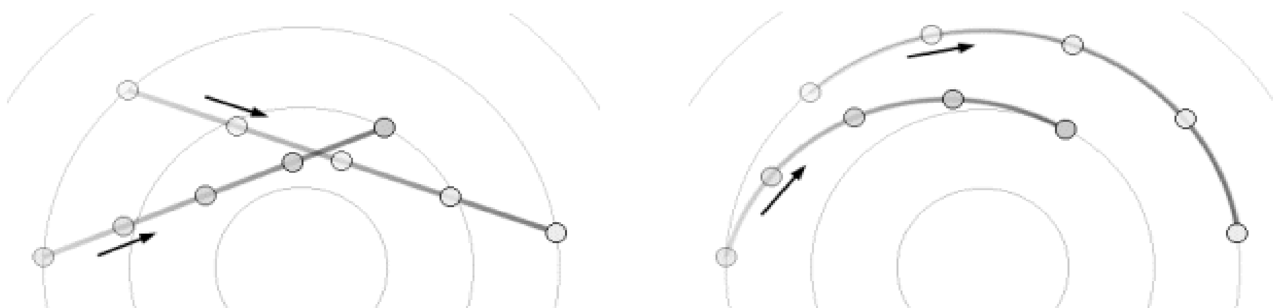
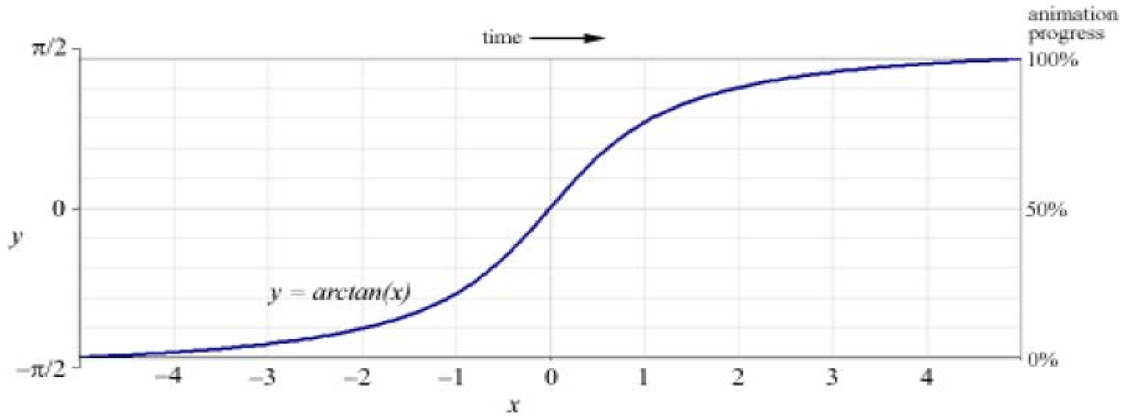


Figure 3. Interpolation in rectangular coordinates (left) can yield a confusing animation. Interpolation in polar coordinates (right) works better for radial layouts.

Animation (continued)

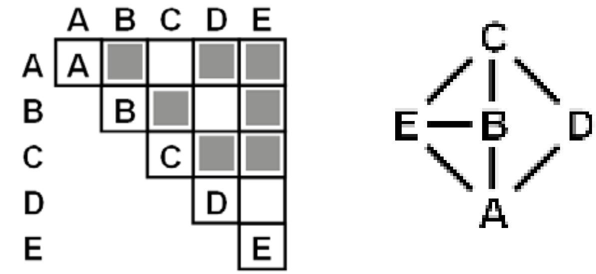
- Transition constraints
  - Orientation of transition to minimize rotational travel
    - Move former parent away from new focus in same orientation
  - Avoid cross-over of edges
    - to allow users to keep track of nodes

- Animation timing
  - Slow in Slow out timing
    - allows users to better track movement

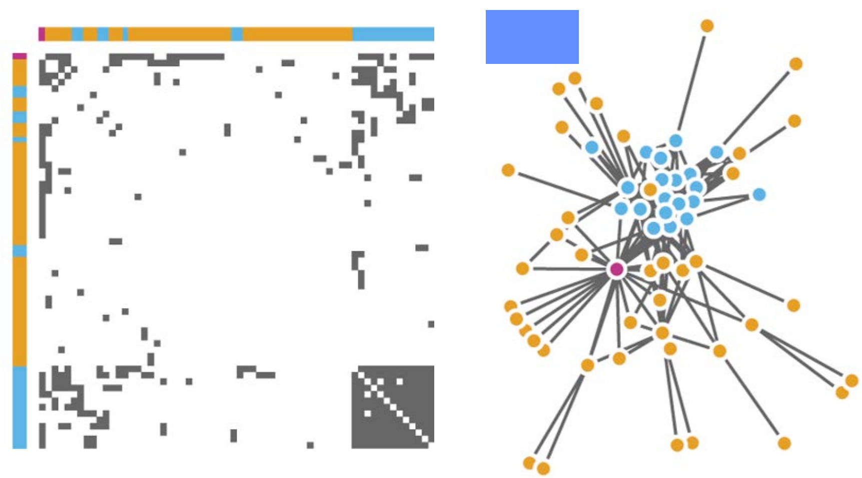


Adjacency Matrix

- Derived data: table from network
  - Keys: categorical attributes: all node pairs
  - Value: 1 quantitative attribute
    - Weighted edge between nodes
- Visual encoding
  - Cell shows presence/absence of edge
  - Color encode
- Scalability
  - 1K nodes, 1M edges



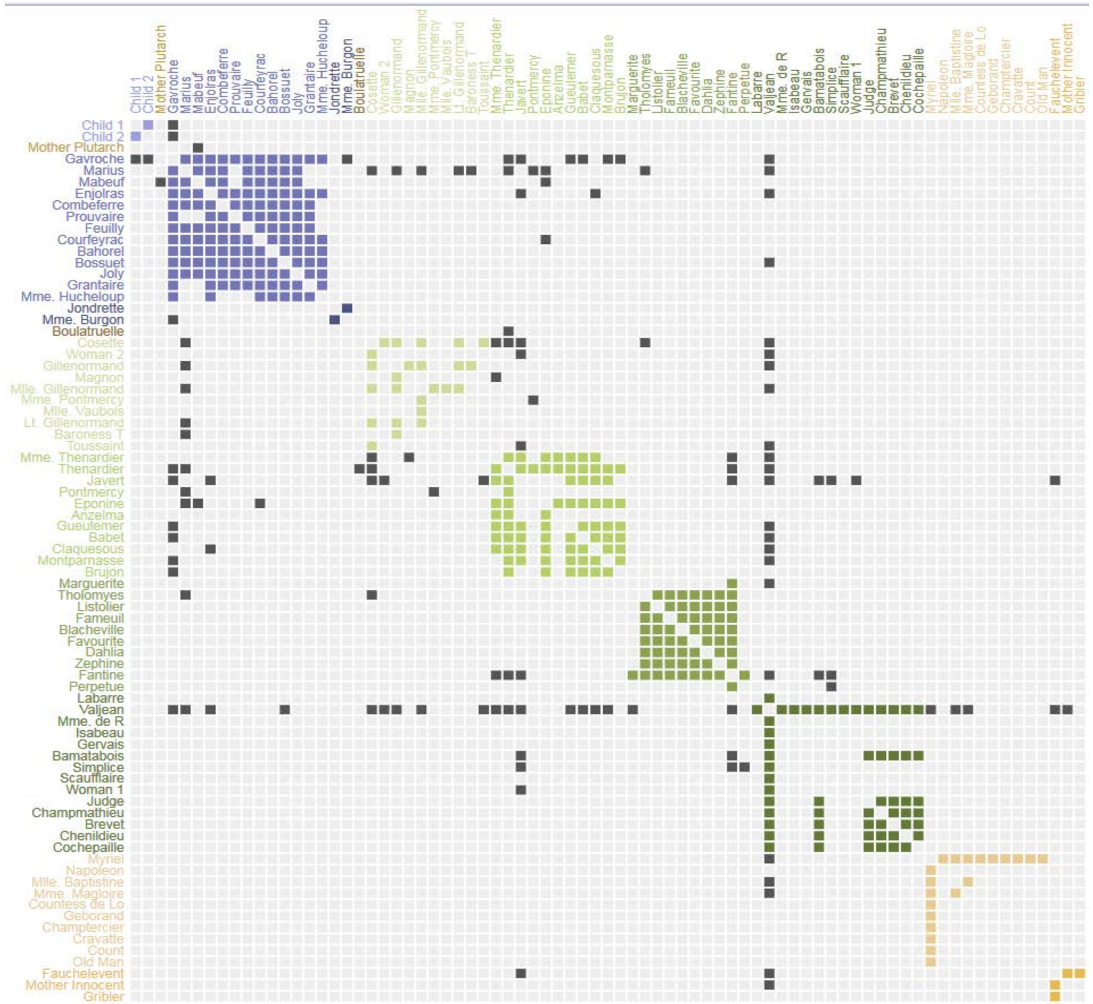
[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]



[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

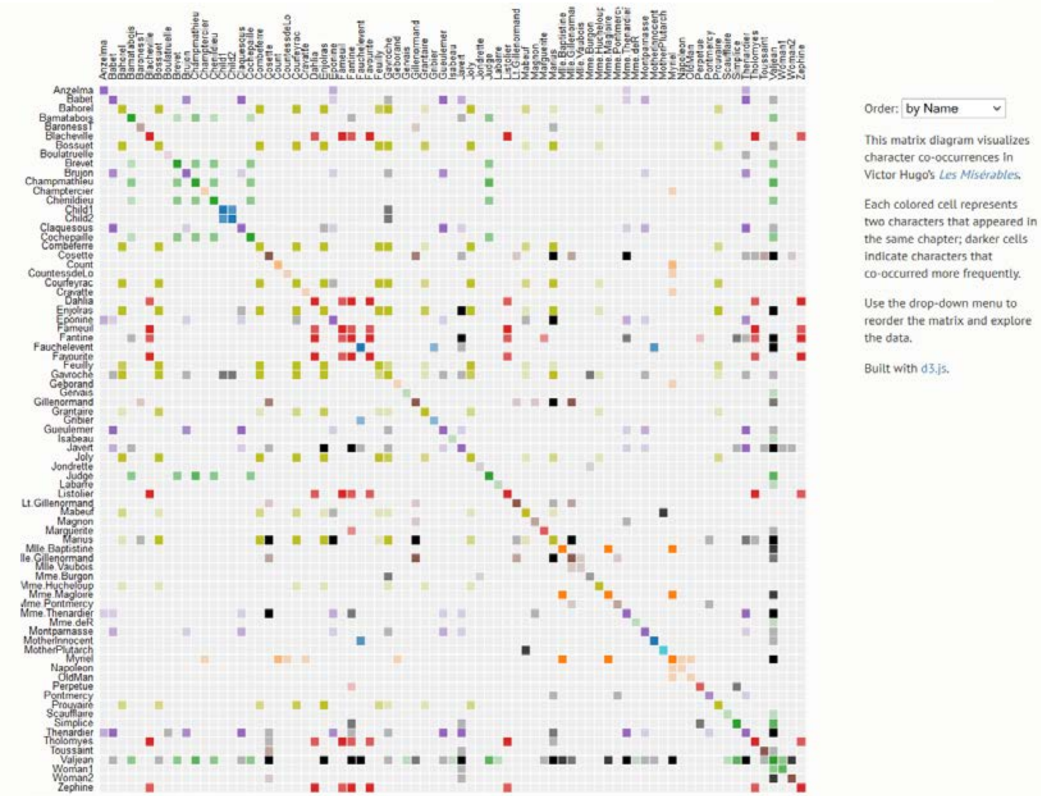
Adjacency Matrix

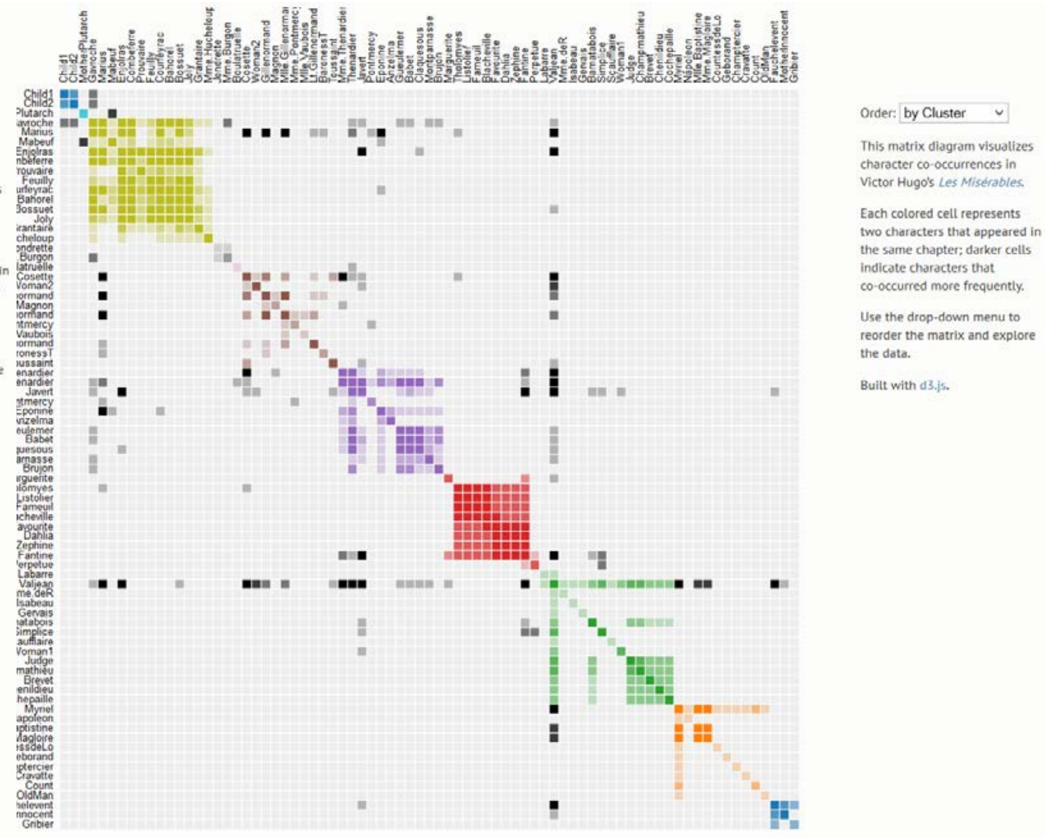
- Links shown by point marks / area marks
- Links inside a cluster have the same color
- Seriation: Ordering of nodes important
- Shows clusters and bridges
- Interactive grouping and reordering for exploration of network structure



http://bost.ocks.org/mike/miserables/

Sorting Matters!





Weighted Graphs

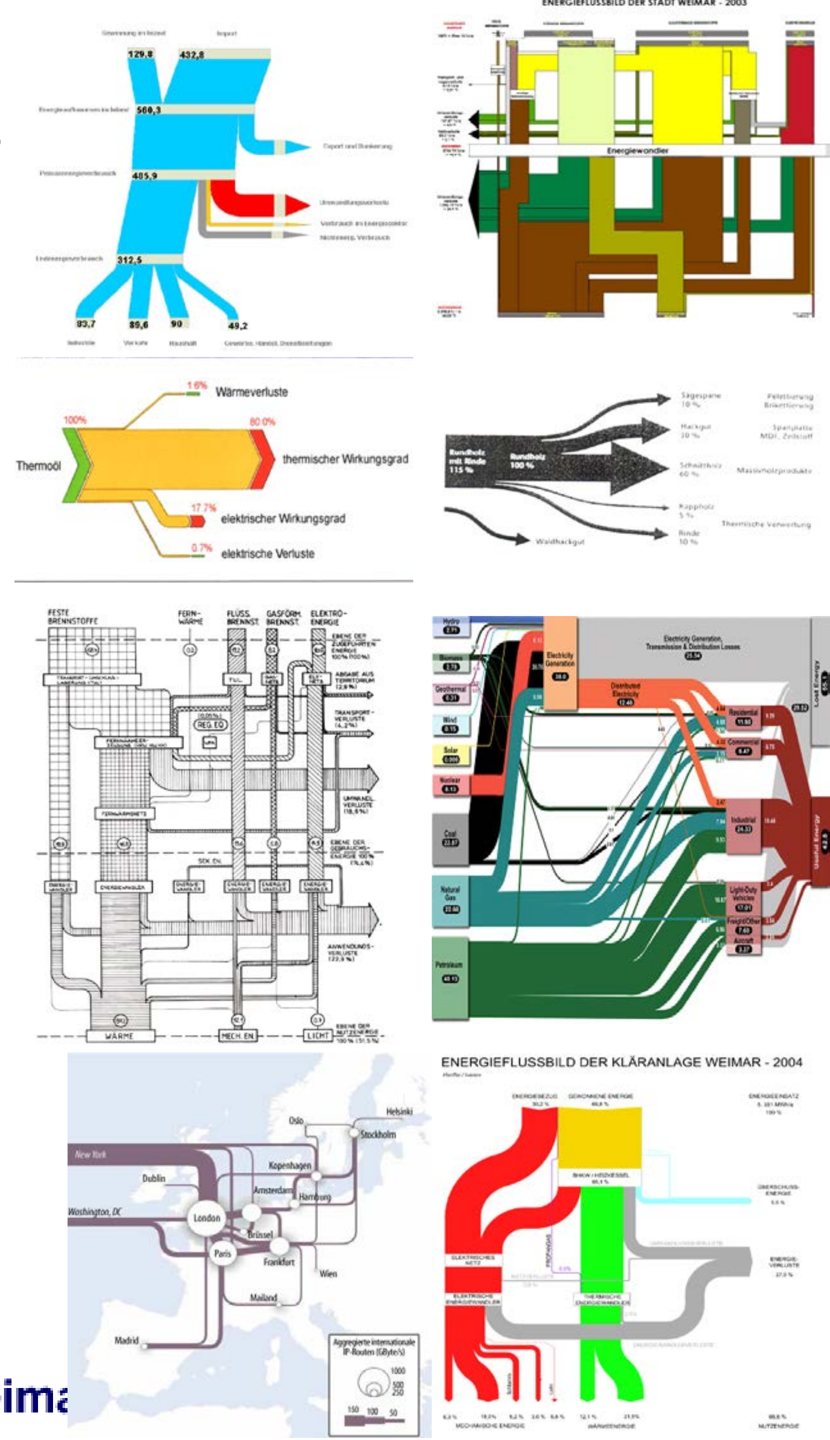
- Flow Diagrams and Sankey Diagrams

Visitor Flow on Web Pages

https://support.google.com/analytics/answer/1012040#funnels_for_destination_goals

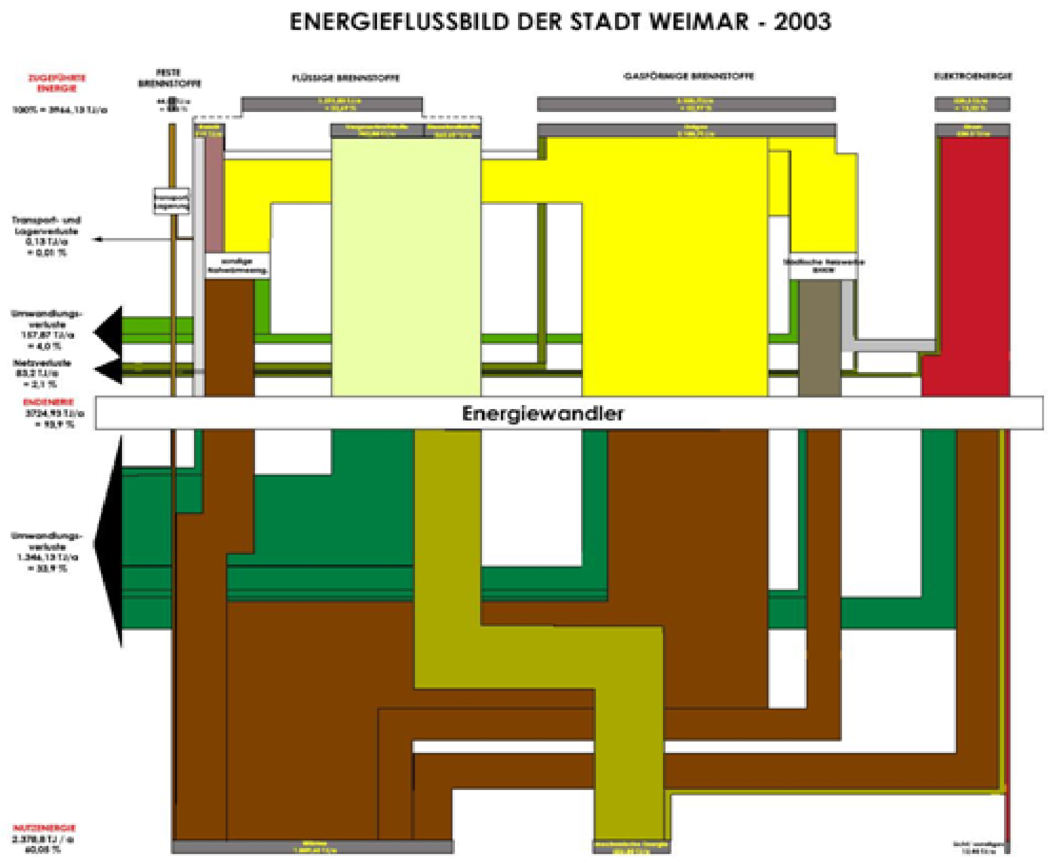
Aesthetics

- Various representations
  - Edge width is often only common feature
  - No standards for drawing diagrams
- Identified criteria and parameters
  - Flow direction and direction representation
  - Node order
  - Drawing of nodes and labeling
  - Drawing of edges
  - Width drawing
  - Loss representation
  - Edge drawing order
  - Colors



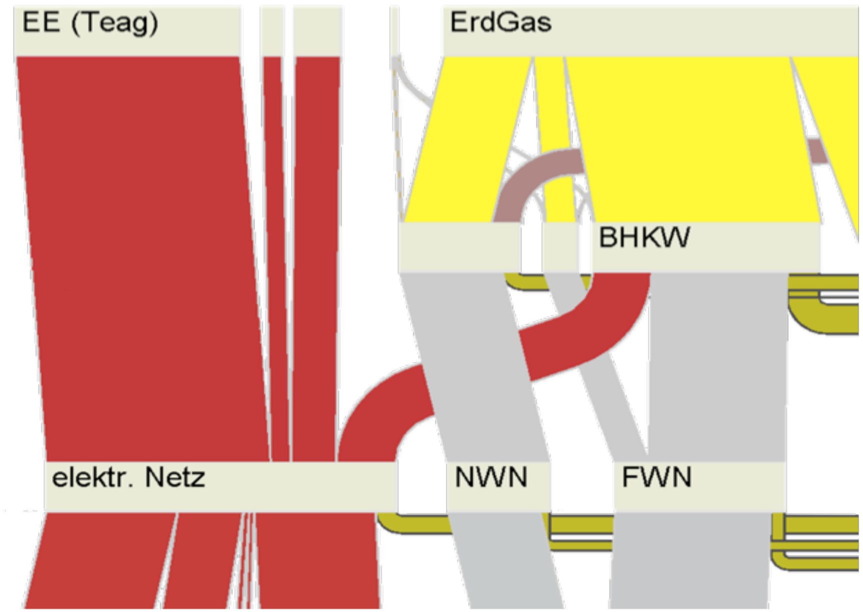
Energy Flow Structures In A City

- How is the energy distributed?
- What kind of energy is used for which purpose?
- How large are the amounts of used energy
- Where are the largest losses?
- How did data change over the years?
- How does the distribution network need to change?



Interactive Sankey Diagrams

- This sounds like the perfect opportunity to use infovis techniques
  - Flow tracing
  - Detail and overview
  - Level of detail
  - Detail on demand
  - Expert features
  - Simple animations



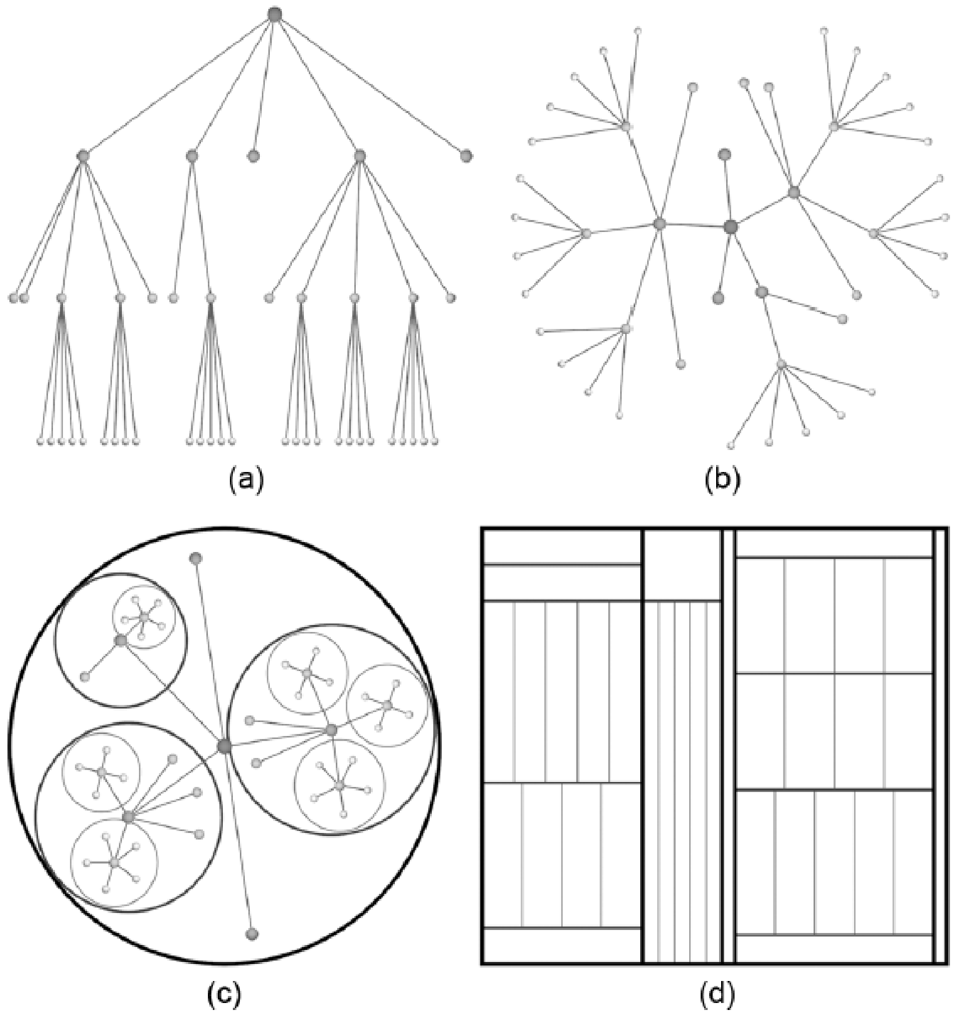
VIDEO

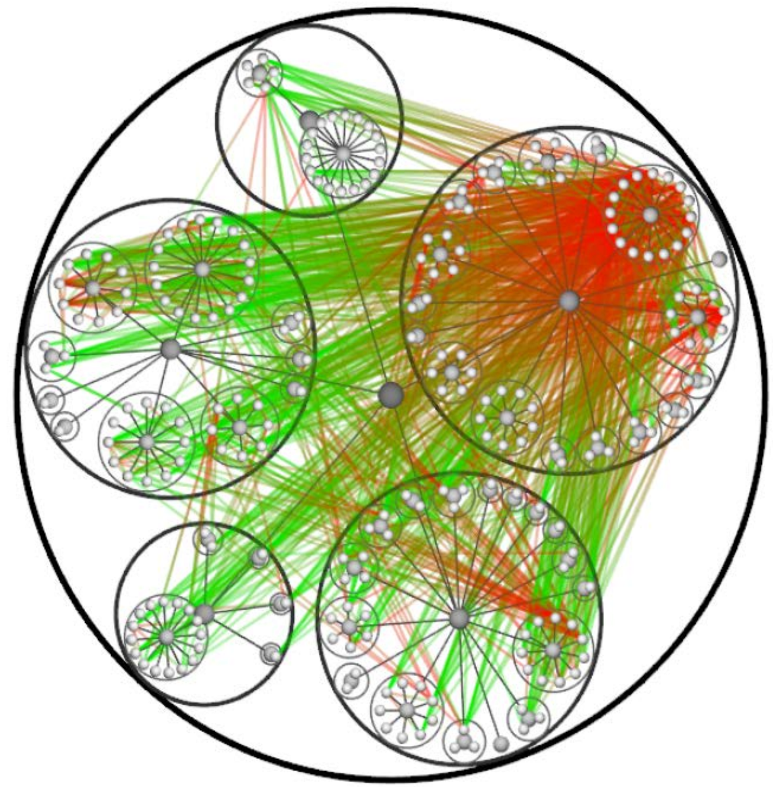
Compound Graphs

- Graphs consisting of two or more separate types of relationships, e.g. a hierarchical component and additional adjacency relations
  - Software package (hierarchy) – dependencies of classes among each other (adjacency relations)
  - Software package (hierarchy) – Call graph (adjacency relations)
  - Email communication in a hierachical organisation
  - Structure of a document (hierarchy) and forward/backward references and citations (adjacency relations)

- Approaches
  - Hierarchical edge bundles
  - The i-Disc

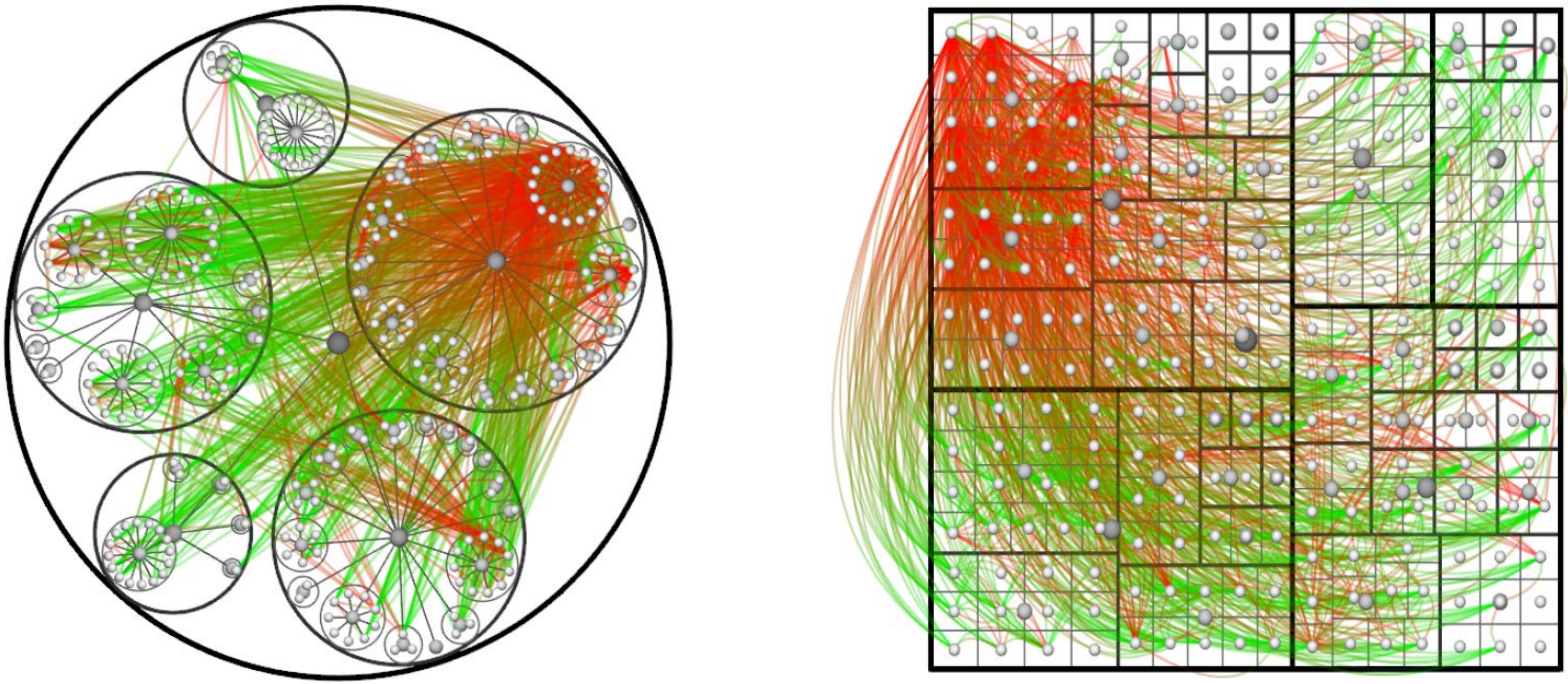
Hierarchy + Adjacency Edges





(a) Rooted tree, (b) radial tree
(c) balloon tree and (d) treemap

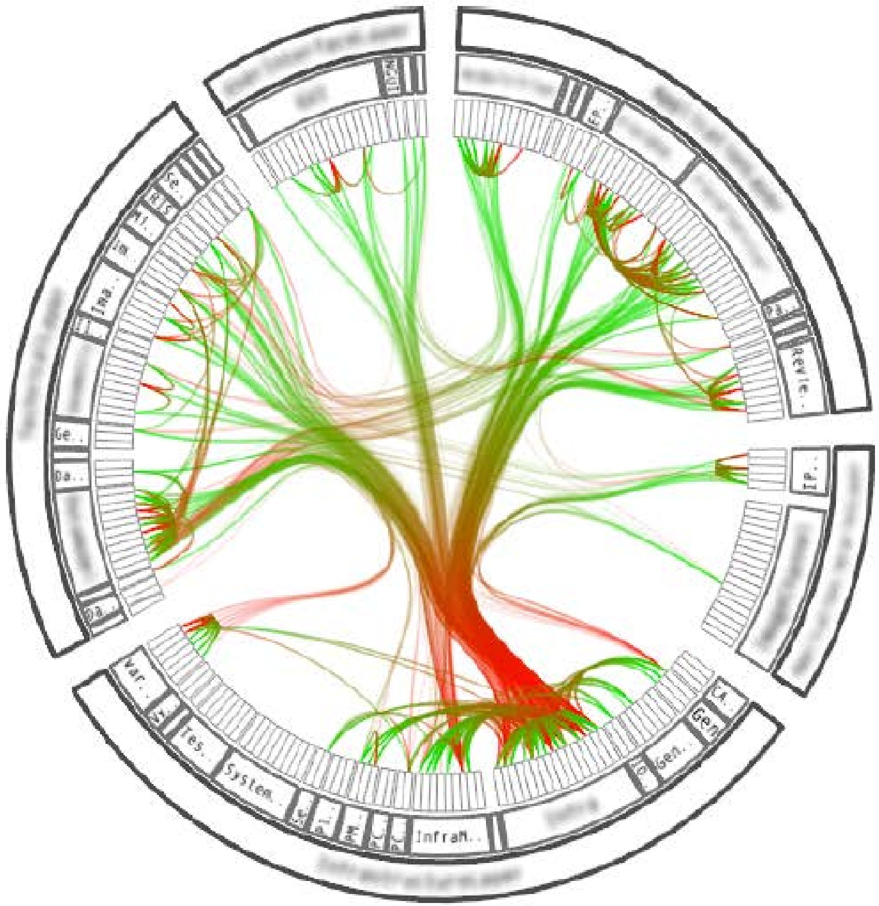
Hierarchy + Adjacency Edges



- A call graph visualized on top of the associated source code tree using (left) color-coded directed straight edges and (right) curved link edges (caller = green, callee = red).

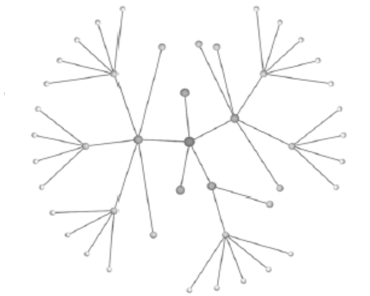
Hierarchical Edge Bundles

Visualization of Adjacency Relations in Hierarchical Data

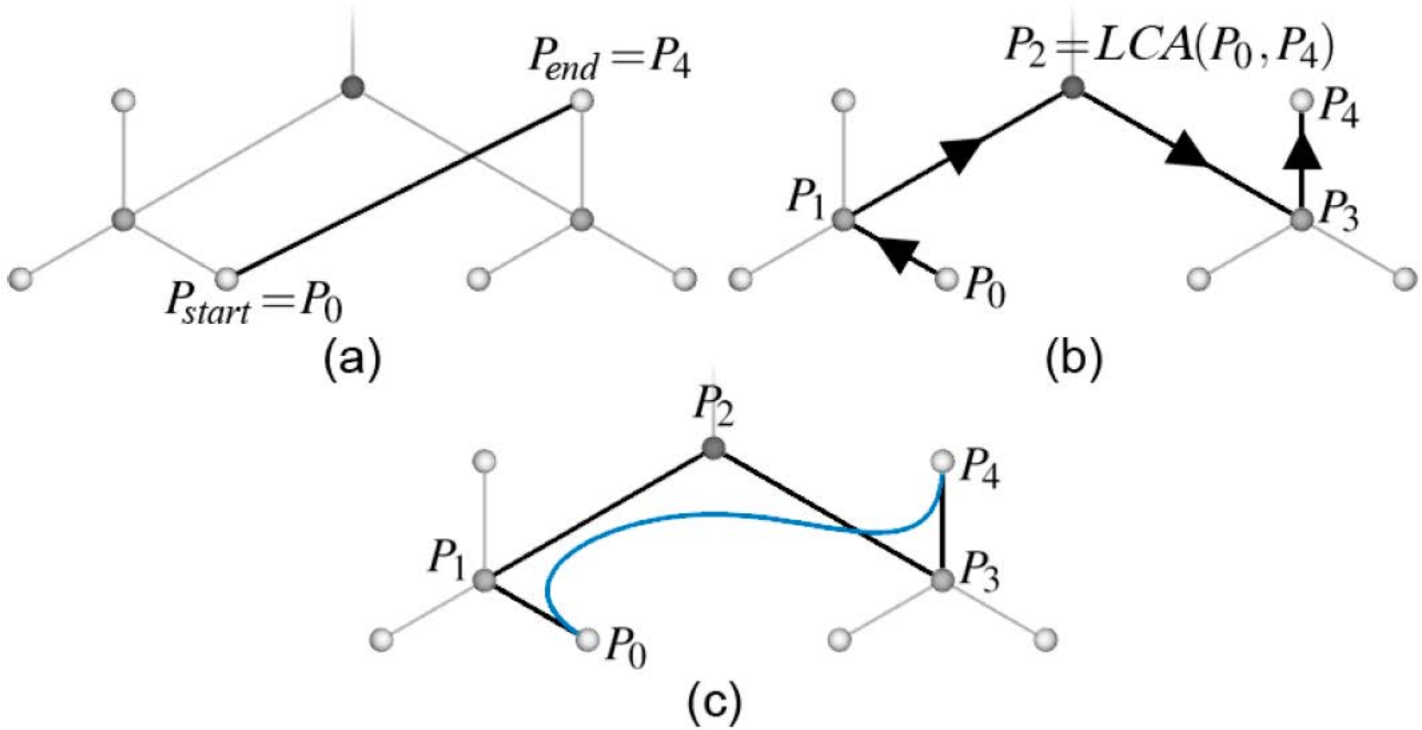


Holten, D., "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data," *Visualization and Computer Graphics, IEEE Transactions on* , vol.12, no.5, pp.741,748, Sept.-Oct. 2006
http://www.win.tue.nl/~dholten/

Generating Hierarchical Edge Bundles



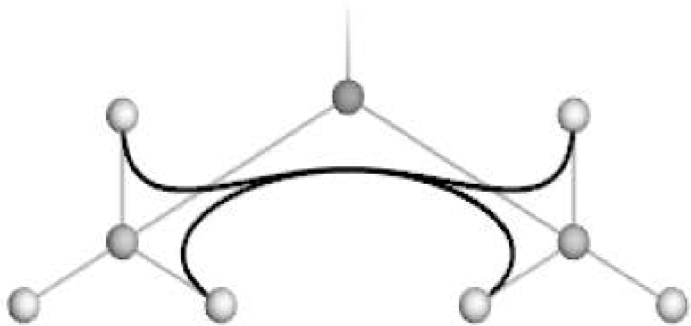
radial tree

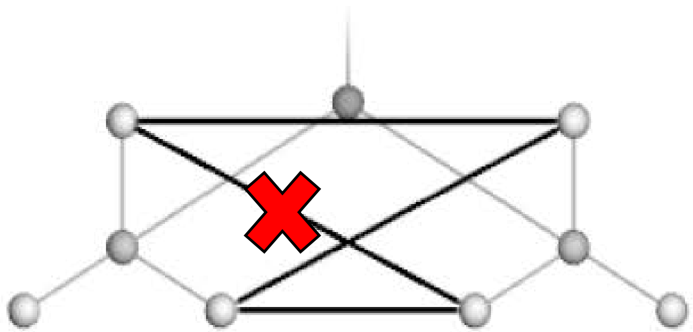


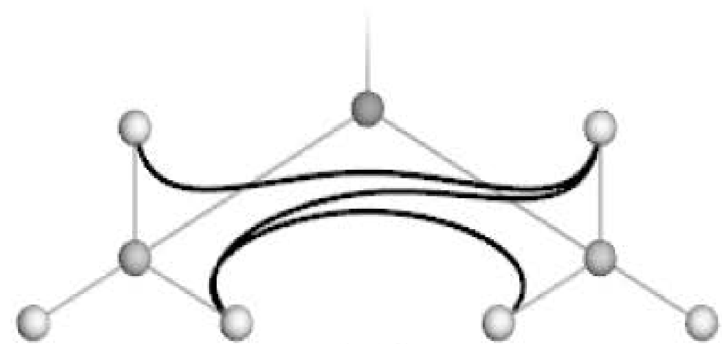
- Central idea: the path along the hierarchy between two nodes having an adjacency relation is used as the control polygon of a spline curve

Controlling the Shape of the Spline Curves

- Ambiguity: The bundle in (a) might contain each edge depicted in (b). The visualization in (e) clearly shows that which edges exist


(a)


(b)


(e)

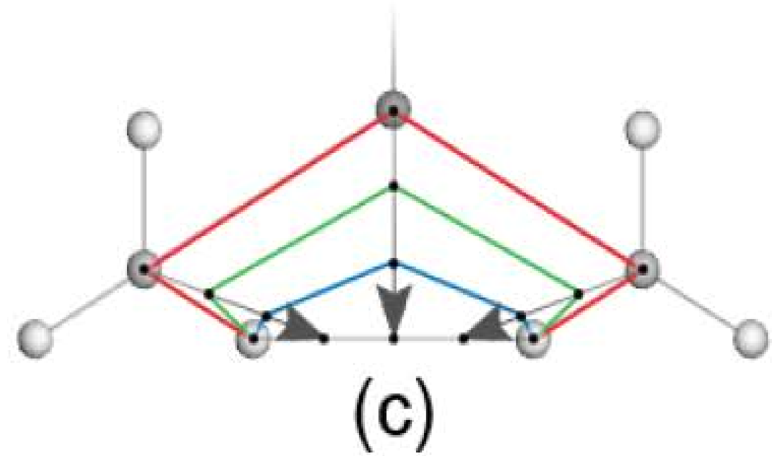
Controlling the Shape of the Spline Curves

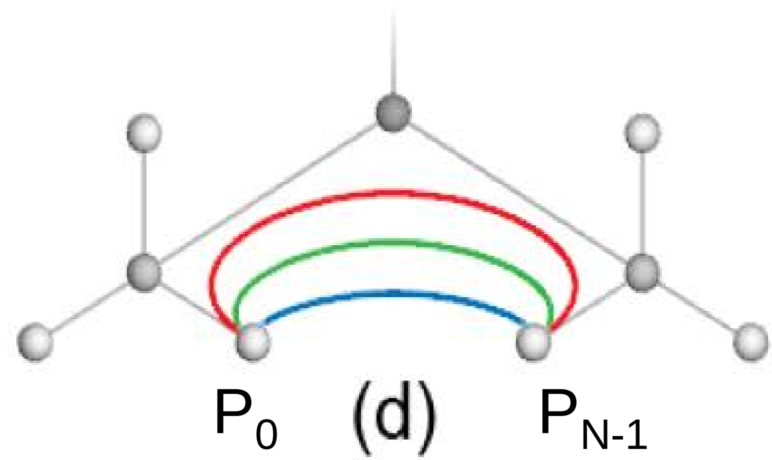
- (c) shows how different values of β (red = 1, green = 2/3 , and blue = 1/3) can be used to alter the shape of the control polygon

$$P_i' = \beta \cdot P_i + (1-\beta)(P_0 + \frac{i}{N-1}(P_{N-1} - P_0))$$

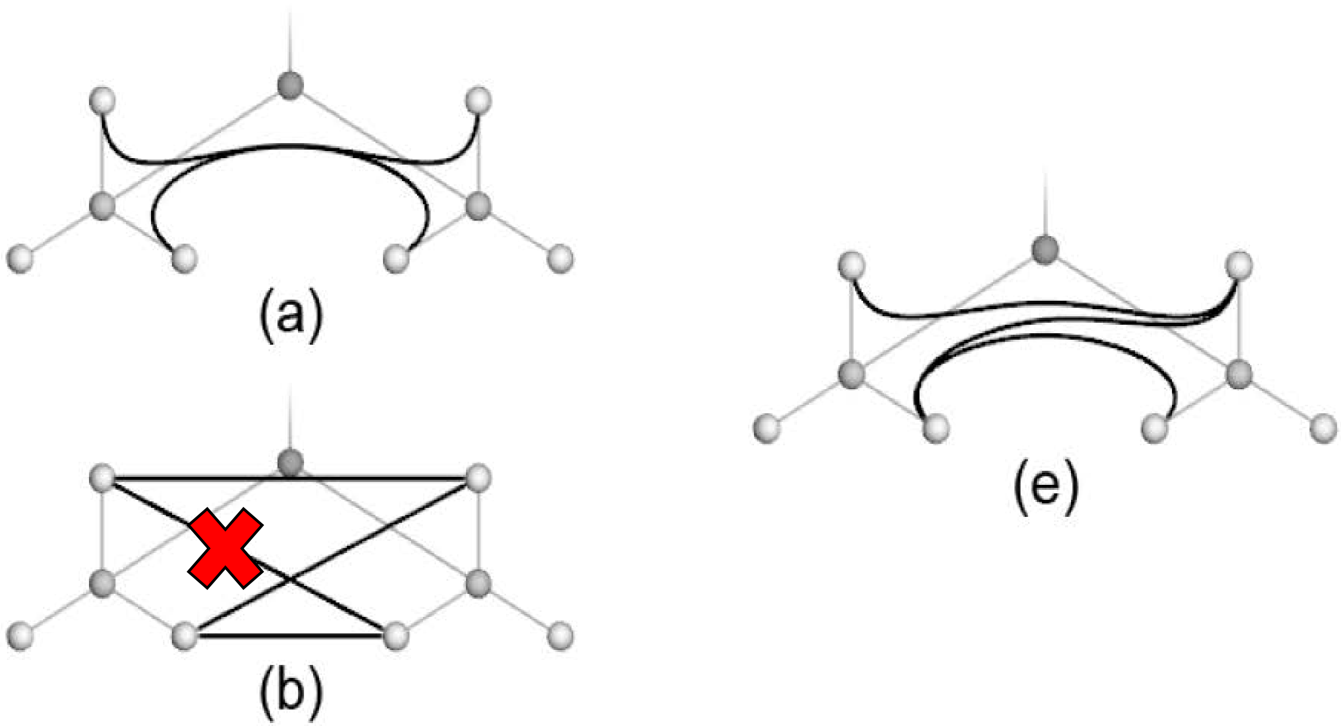
The value of (1-β) draws the control points towards a straight-line connection between the start-vertex P_0 and end vertex P_{N-1} of the adjacency relation

- (d) shows the change of the corresponding spline curves.



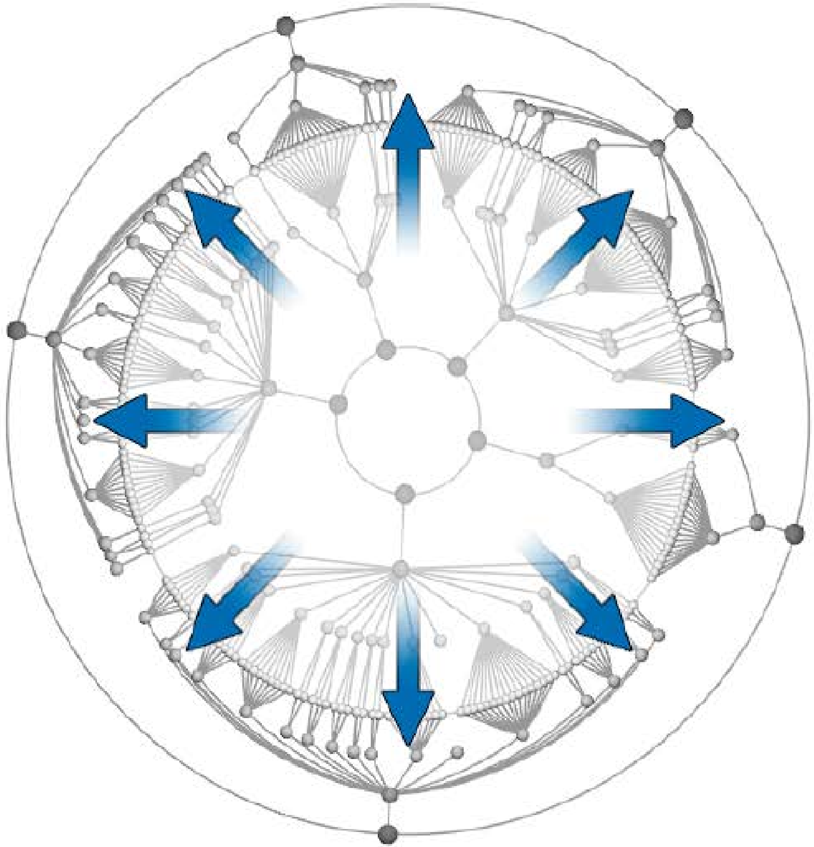


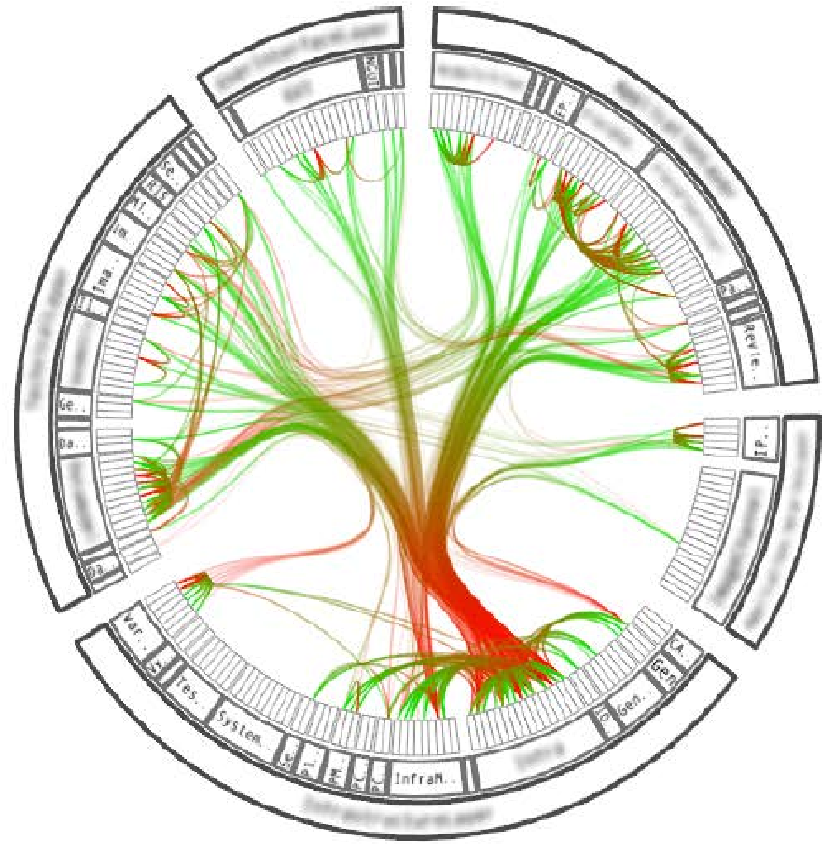
Controlling the Shape of the Spline Curves



- (e) a fairly high bundling strength (β = 0.8) can be chosen to to resolve the ambiguity

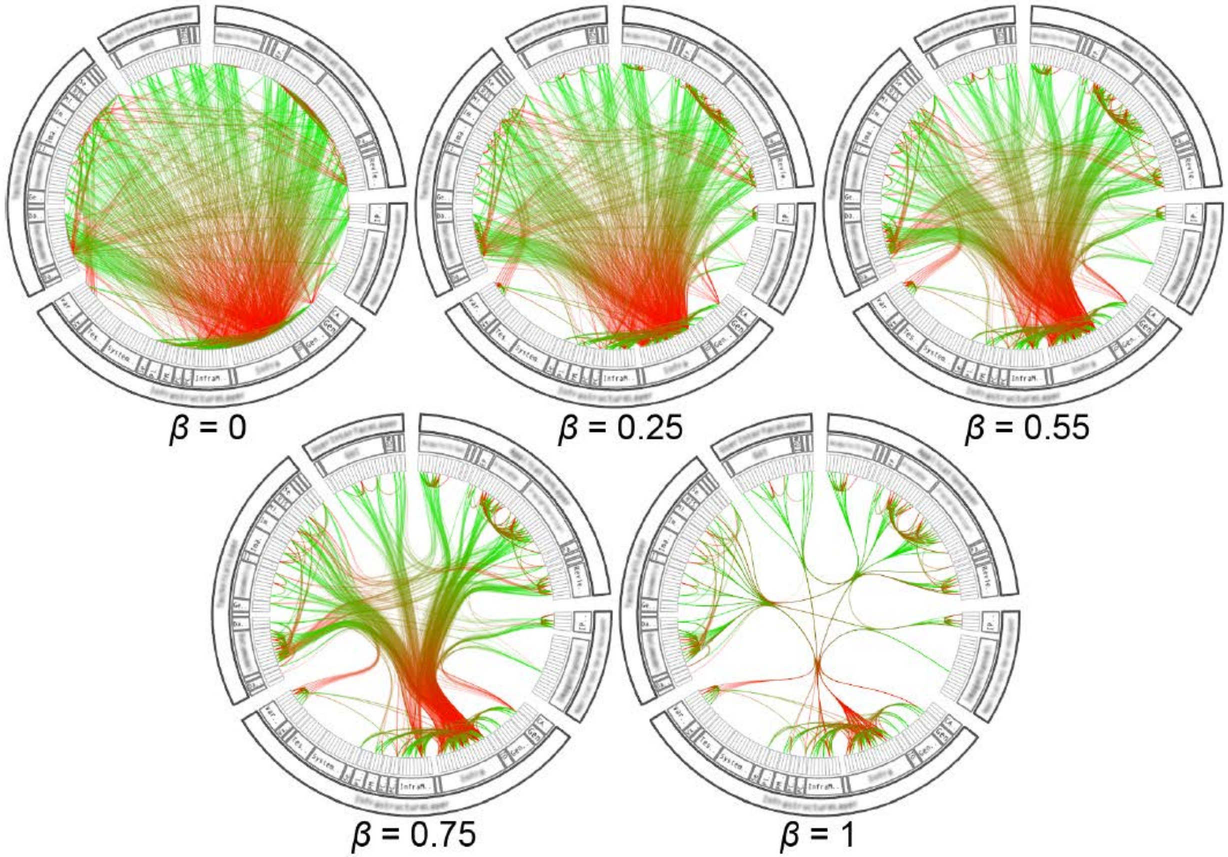
Constructing the Radial Layout





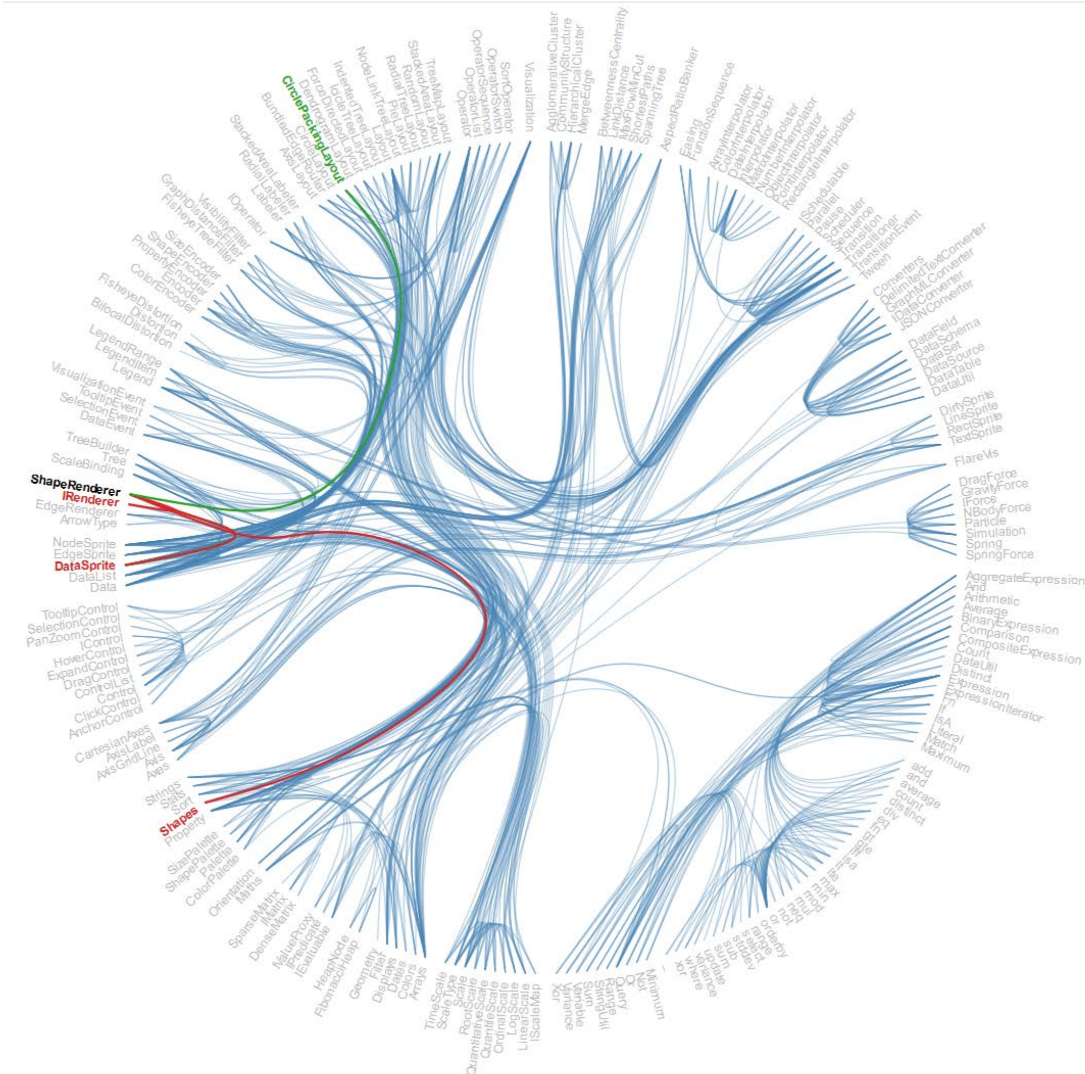
- (left) A radial tree layout is used for the inner circle and subsequently mirrored to the outside.
- (right) The inner layout is hidden and its structure is used to create the curved adjacency edges. An icicle plot based on the mirrored layout is used to show the hierarchy.

Bundling Strength



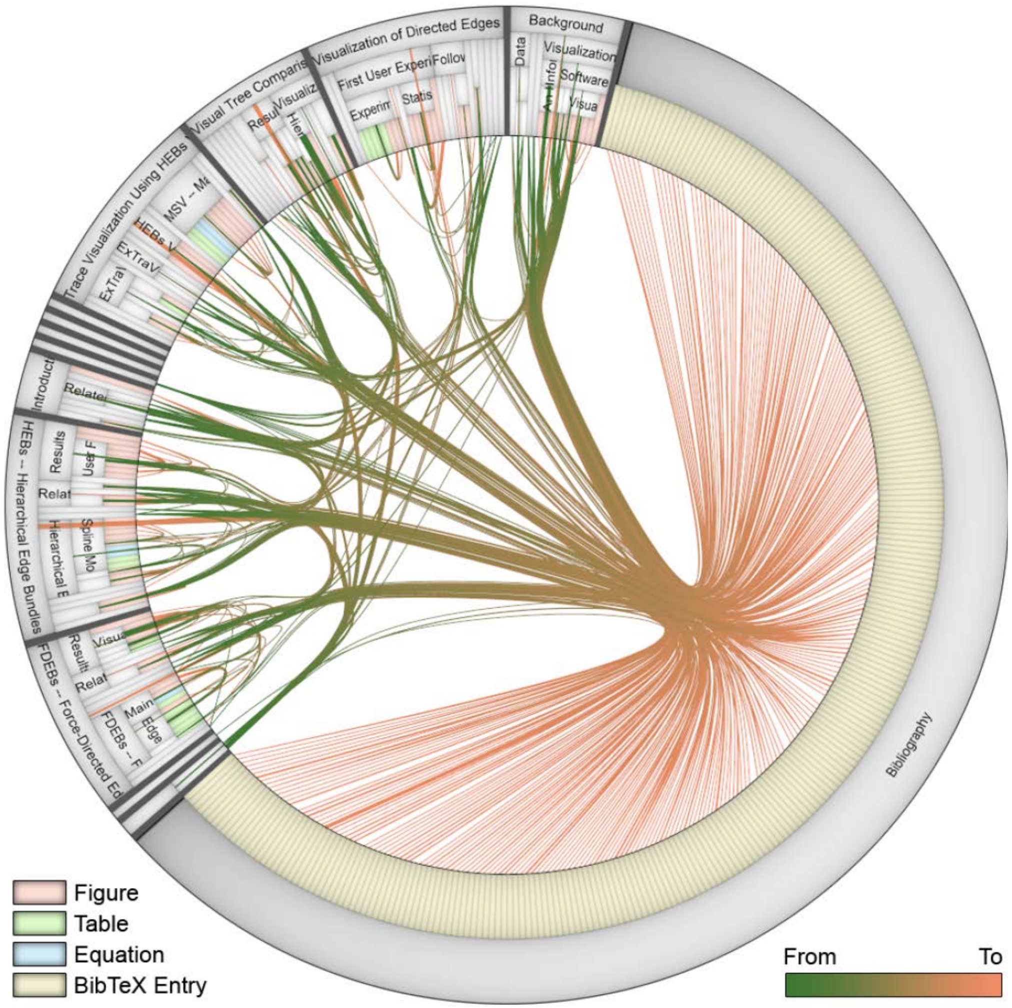
- Bundling strength provides a trade-off between low-level and high-level views of the adjacency relations.
  - low values mainly provide low-level, node-to-node connectivity information
  - high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Interaction: Trace Connections



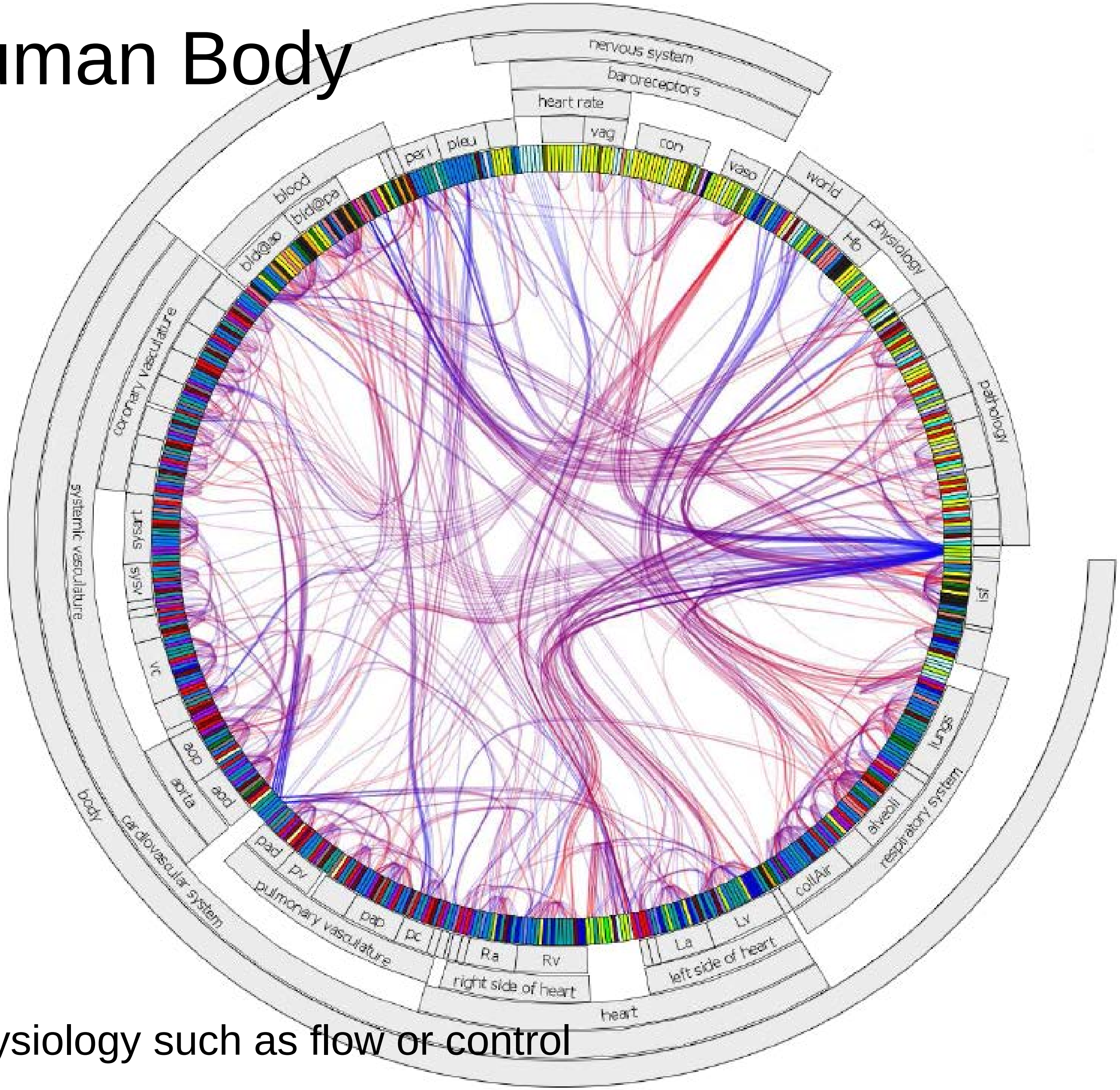
- Dependencies in a software package
  http://bl.ocks.org/mbostock/1044242 http://mbostock.github.com/d3/talk/20111116/bundle.html

Example: Holten‘s PhD Thesis



- The hierarchical organization of Holten’s thesis and the adjacency relations comprising ref and cite commands in LaTeX.

Example: Human Body

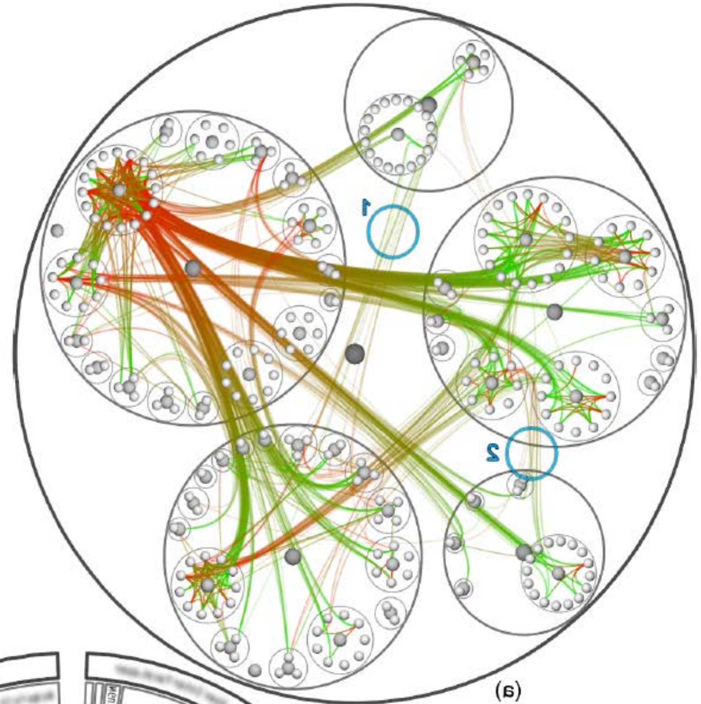
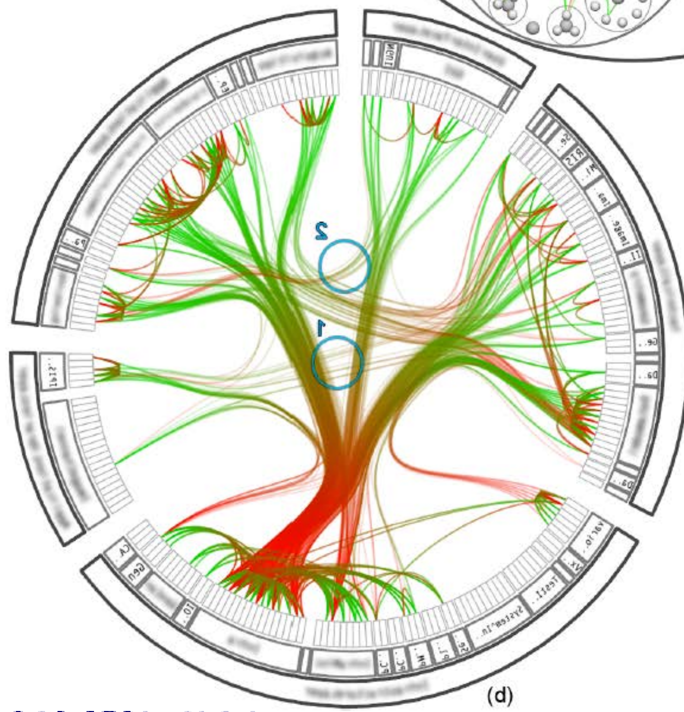


- Edges represent physiology such as flow or control

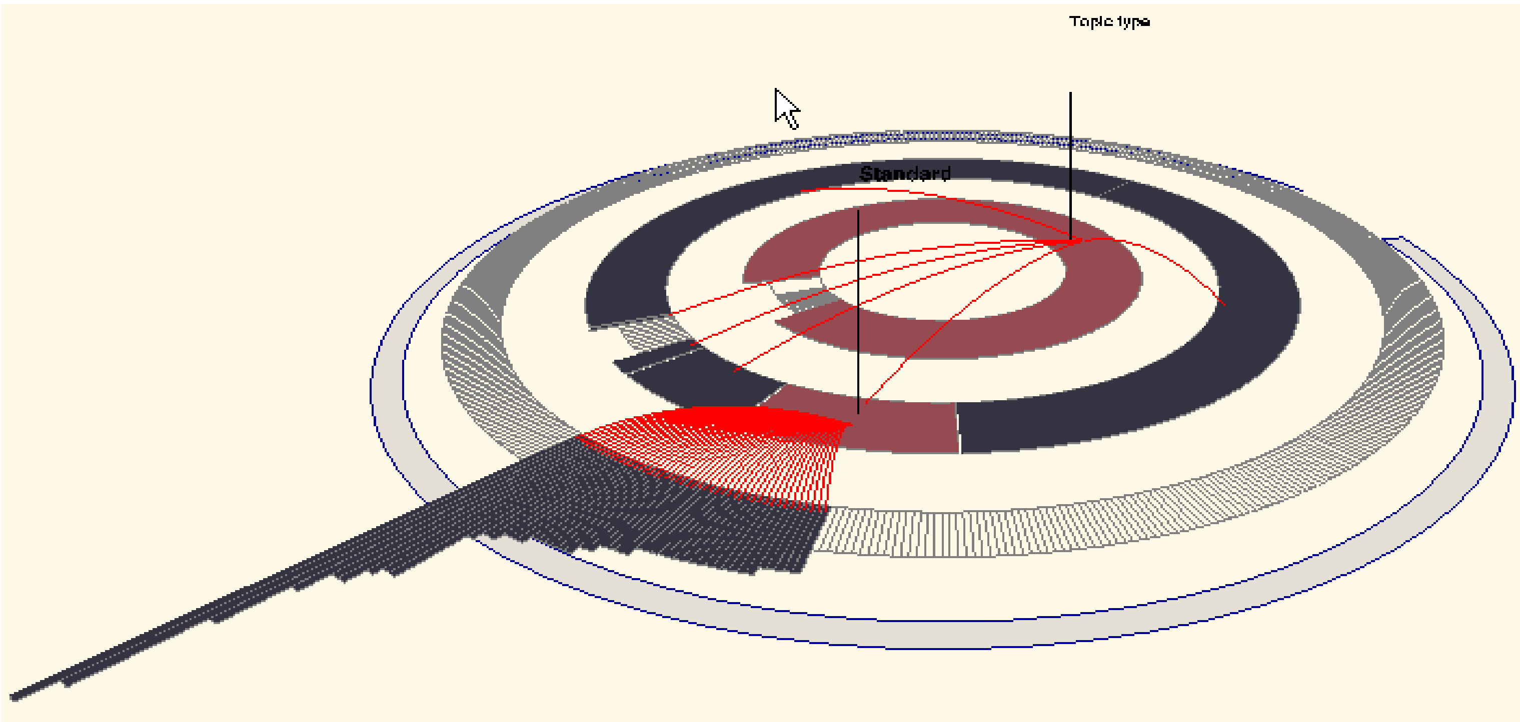
Observations

- Software system and its associated call graph.
- Bundling strength $\beta = 0.85$

- Bundling
  - reduces visual clutter,
  - makes it easier to perceive the actual connections
  - sparsely connected systems can be seen more clearly (encircled regions).
- Interaction necessary to reveal detailed relationships

The i-disc



Uses 3rd dimension to display adjacency relationships
Perspective: Natural focus and context display
http://www.uni-weimar.de/de/medien/professuren/vr/research/infovis/the-i-disc/

Further reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - Chap 9: Arrange Networks and Trees
- Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. von Landesberger et al. Computer Graphics Forum 30:6 (2011), 1719–1749.
- Simple Algorithms for Network Visualization: A Tutorial. McGuffin. Tsinghua Science and Technology (Special Issue on Visualization and Computer Graphics) 17:4 (2012), 383–398.
- Drawing on Physical Analogies. Brandes. In Drawing Graphs: Methods and Models, LNCS Tutorial, 2025, edited by M. Kaufmann and D. Wagner, LNCS Tutorial, 2025, pp. 71–86. Springer-Verlag, 2001.
- Treevis.net: A Tree Visualization Reference. Schulz. IEEE Computer Graphics and Applications 31:6 (2011), 11–15. http://www.treevis.net
- Perceptual Guidelines for Creating Rectangular Treemaps. Kong, Heer, and Agrawala. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis) 16:6 (2010), 990–998.

End

Visualization

Introduction to Scientific Visualization

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

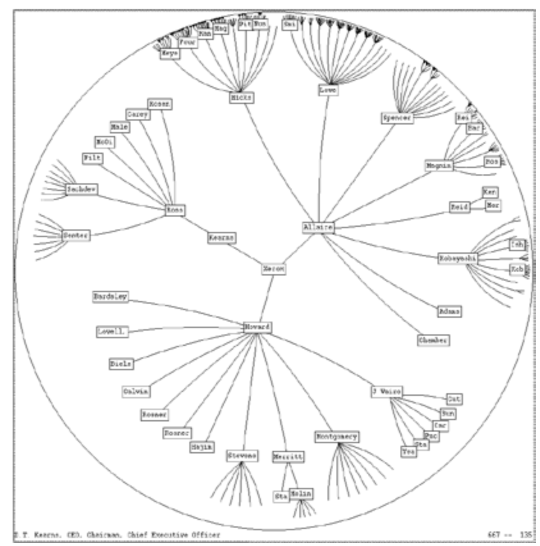
“The purpose of computing is insight not numbers“

Richard Hamming

Numerical Methods for Scientists and Engineers (McGraw Hill 1962)

Visualization

- Scientific Visualization
  - Physical / simulated data
  - Data
    - Automotive
    - Weather
    - Medical
    - ...
- Information Visualization
  - Abstract, non-physical data
  - Data
    - Financial/business
    - Statistics
    - Software
    - Text
    - W W W
    - ...



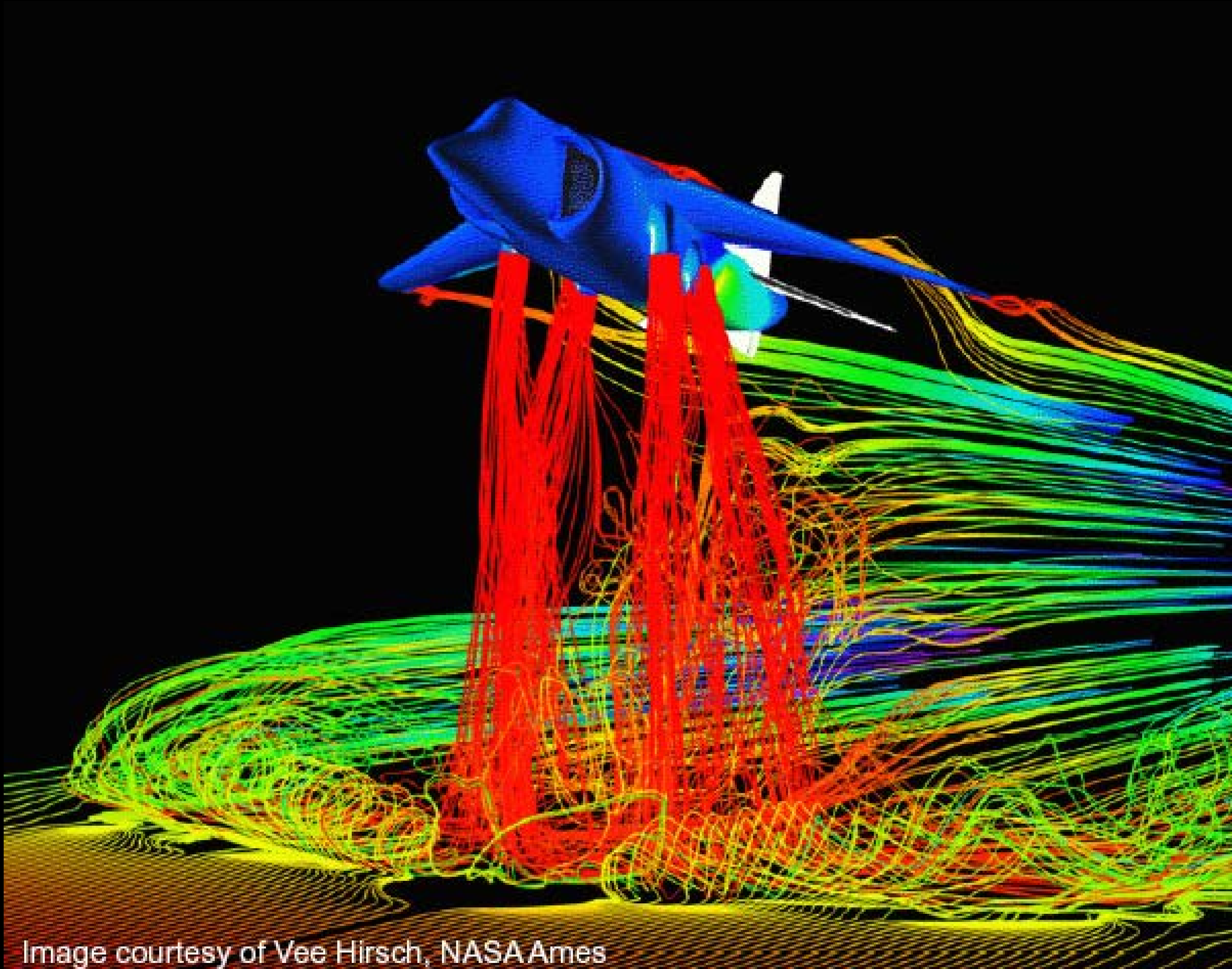


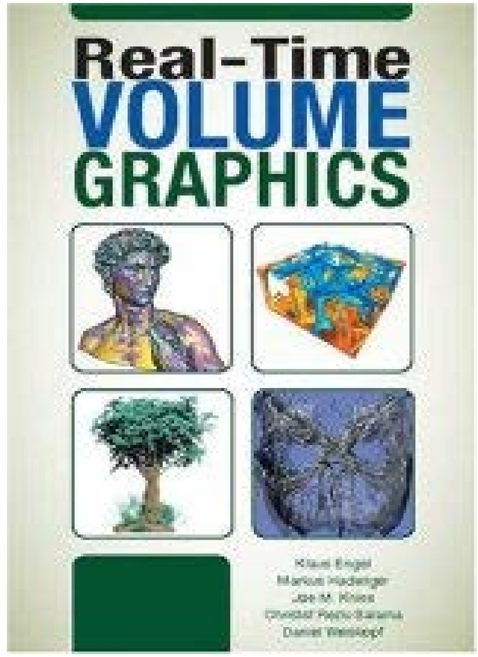
Image courtesy of Vee Hirsch, NASA Ames

Scientific Visualization Books

- Nielson, Hagen, Müller: Scientific Visualization
- Schumann, Müller: Visualisierung. Grundlagen und allgemeine Methoden
- Richard S. Gallagher: Computer Visualization : Graphics Techniques for Scientific and Engineering Analysis
- William Schroeder, Ken Martin, Bill Lorensen: The Visualization Toolkit : An Object-Oriented Approach to 3-D Graphics
- Charles Hansen, Christopher Johnson: The Visualization Handbook
- Video
  - http://www.ted.com/talks/anders_ynnerman_visualizing_the_medical_data_explosion.html

Volume Rendering Book

- Title: Real-Time Volume Graphics
- Christof Rezk-Salama, Klaus Engel, Markus Hadwiger, Joe M. Kniss, Daniel Weiskopf
- A K Peters (30. August 2006)
- ISBN-10: 1568812663
- ISBN-13: 978-1568812663



Classic Visualization Pipeline

- Filtering
  - Extract relevant data
  - Correct errors
  - Smooth data
- Mapping
  - Convert data into geometry with visual attributes
- Rendering
  - Display geometric data
  - View point, perspective, ...

Scientific Visualization Data

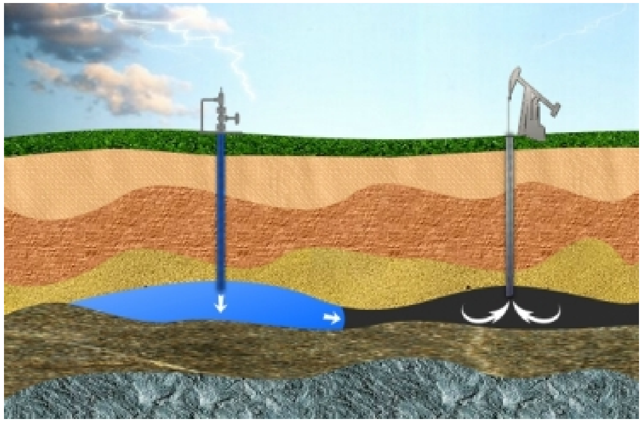
- Examples
- Classification
  - Source
  - Dimension
  - Topology

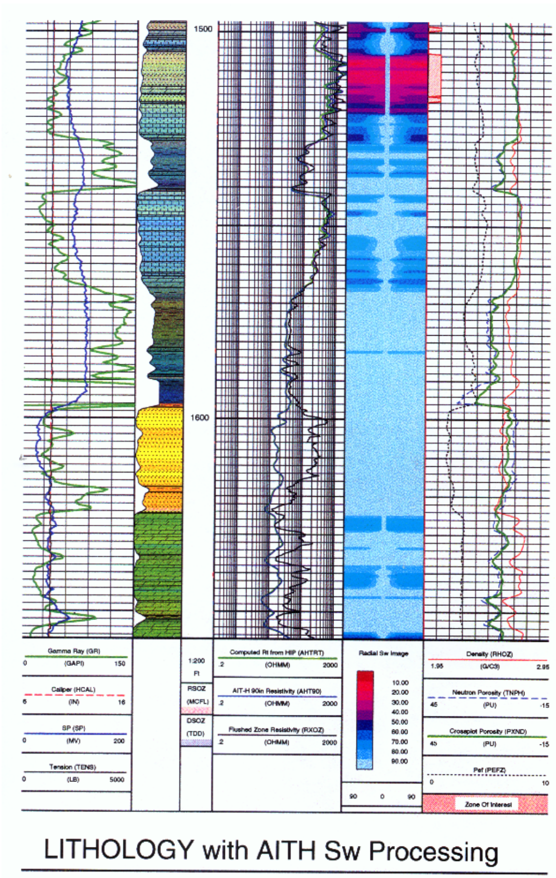
- Lecture based on lectures by
  - Greg Nielsen, Arizona State University
  - Roger Crawfis, Ion Barason, Bo Kågström, Eric Elmroth, Ohio State University and Umeå University

Examples

- 13 Data sets
  - Selected for variety
  - Images serve as overview of scientific visualization techniques

Well Logs



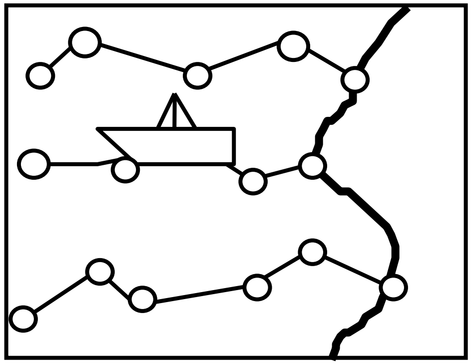


Source: Selley 1998

- Multiple parameters
  - Density
  - Natural radioactivity
  - Electrical resistance
  - ...
- (i, X_i, Y_i, Z_i, P_{i1}, P_{i2}, ...)

	X	Y	Z	P1	P2	...
1	6.3	7.5	9.0	10	0.7	...
2	6.4	7.5	9.3	11	0.6	...
3	6.4	7.7	9.5	8.7	0.2	...
...	...	...	...	...	...	...

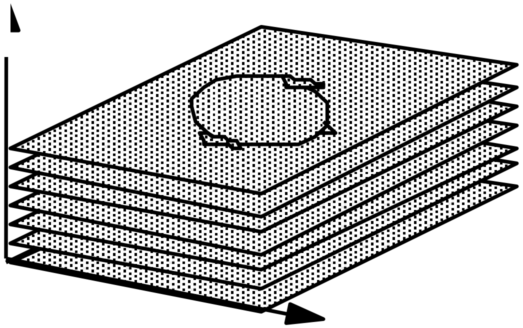
Big Sur Data



Location		Temperature
23.56	37.80	18.3
49.29	54.78	16.2
67.24	43.42	15.2
•	•	•
•	•	•
•	•	•

- Track Data (Courtesy R. Franke)
- $(x_{ij}, y_{ij}; T_{ij})$, $i = 1, \ldots, N$; $j = 1, \ldots, M$
- Irregular data
  - Needs scattered data interpolation

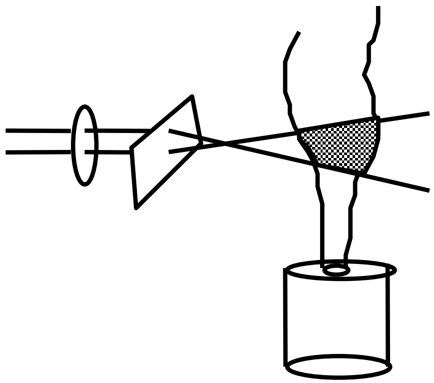
3D Scan Data



x_i	y_i	z_i	Density
0.00	0.00	0.00	243
0.00	0.00	0.12	175
•	•	•	•
0.00	0.00	1.00	186
0.00	0.12	0.00	187
•	•	•	•

- $F_{ijk} = F(X_i, Y_j, Z_k)$
- $i = 1,\ldots,N;\quad j = 1,\ldots,M;\ k = 1,\ldots,P$
- Medical Scanners, CT, MRI, etc.

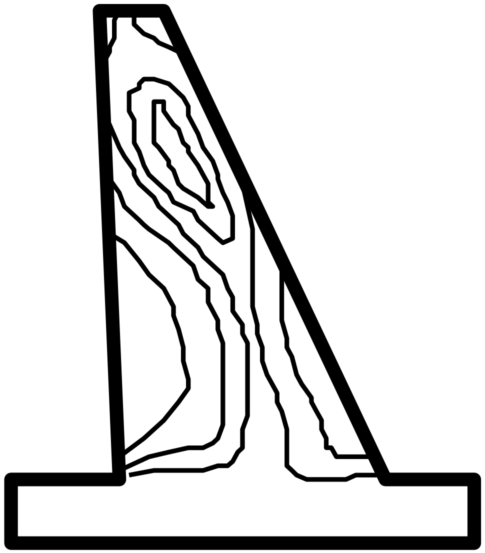
Flame Data



Location			Concentration
0.00	0.00	0.01	.001
0.00	0.00	0.02	.003
⋮	⋮	⋮	⋮

- Data courtesy M. Long, Yale University
- $(r_i\cos(\Theta_j), r_i\sin(\Theta_j), z_k; C_{ijk})$
- $r_i = r_{min} + i\Delta r$,
- $\Theta_j = \Theta_{min} + j\Delta\Theta$
- $z_k = z_{min} + k\Delta z$

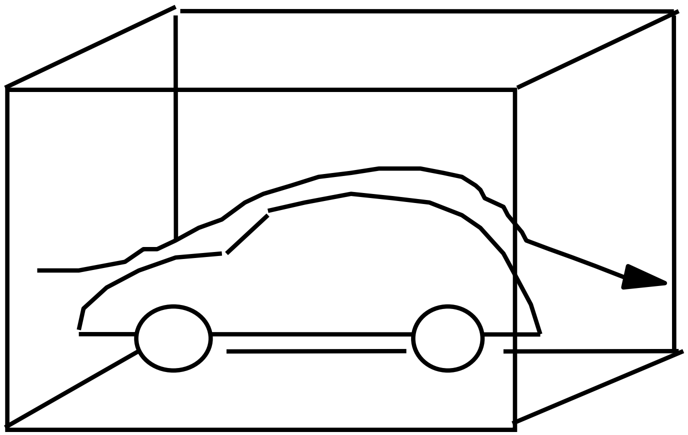
Wing Data



Location			Pressure
-132.1	38.5	6.1	0.164
-128.3	38.6	99.7	0.119
⋮	⋮	⋮	⋮

- Data courtesy NASA Ames
- $(X_{ij}, Y_{ij}, Z_{ij}, P_{ij})$
- $(X_{ij}, Y_{ij}, Z_{ij}) = W(u_i, v_j)$, $i = 1,\ldots,N_u$ $j = 1,\ldots,N_v$
- $W(u,v)$ is parametric surface of the wing

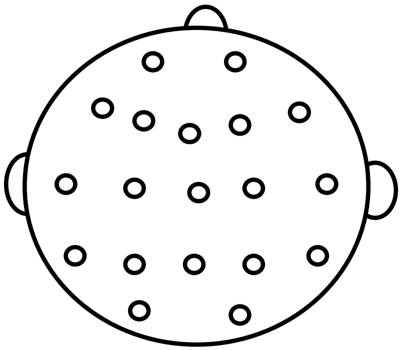
Nissan Wind Tunnel Data



Location			Velocity
7.77	9.45	3.85	(1.33, 2.34, 0.45)
4.14	-2.78	2.68	(1.86, 3.56, 1.25)
⋮	⋮	⋮	⋮

- Data courtesy Y. Nakajima, Nissan
- $(X_i, Y_j, Z_k; u_{ijk}, v_{ijk}, w_{ijk})$
- $i = 1,\ldots,N_x$, $j = 1,\ldots,N_y$, $k = 1,\ldots,N_z$

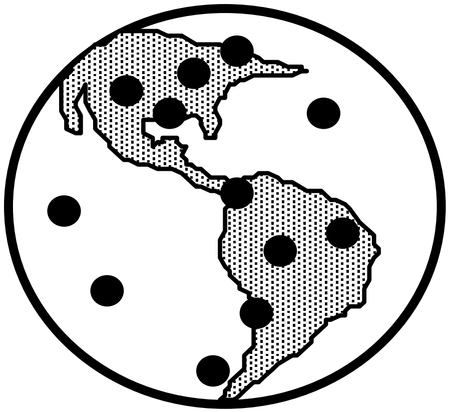
Brain Data



Location			Voltage $t_1, \bullet\bullet, \quad t_6$
6.54	4.56	5.64	0.33 • • 0.45
9.14	-3.14	1.38	1.86 • • 0.22
⋮	⋮	⋮	⋮

- Data Courtesy Brain Physics Group, Tulane University
- (X_i, Y_i, Z_i, P_{ij}), $i = 1,\ldots,N$, $j = 1,\ldots,6$
- $(X_i, Y_i, Z_i) \in$ Scalp
- Irregular data
  - Needs scattered data interpolation

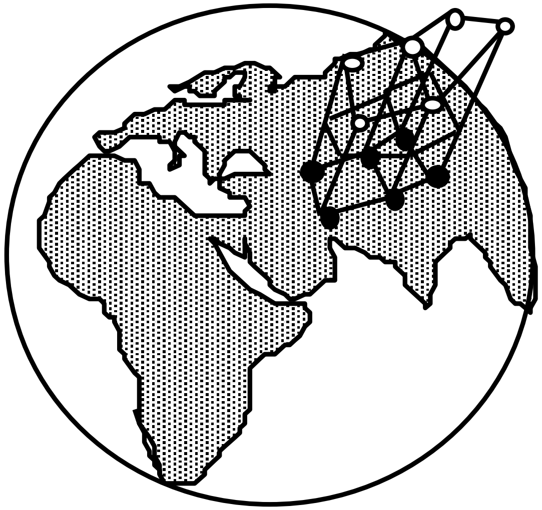
Weather Data



Longitude	Latitude	Height	Pressure
34 11' 30"	21 17' 13"	14.6m	1010
27 15' 55"	43 45' 21	23.6m	990
43 10' 37"	21 16' 23"	14.2m	1005
...	...	...	...

- Courtesy NPS and Univ. Delaware
- (X_i, Y_i, Z_i, P_i), $i = 1, \ldots, N$
- $(X_i, Y_i, Z_i) \in$ Surface of Earth
- Irregular data

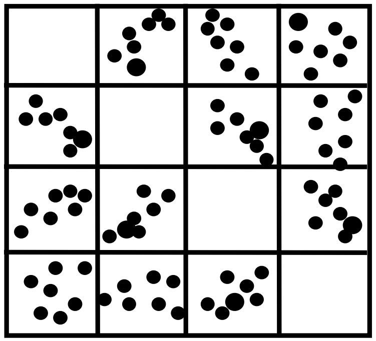
Climate Model Data



Longitude	Latitude	Level k	Velocity
43 19' 34"	23 36' 13"	1	(1.2, 2.5, 6.2)
43 19' 34"	23 45' 36"	1	(2.6, 2.9, 3.7)
44 20' 57"	23 36' 13"	2	(2.1, 5.2, 2.6)
...	...	...	...

- Data courtesy R. Crawfis and N. Max, LLNL
- $(\Theta_i, \Phi_j, \rho^k_{ij}; u_{ijk}, v_{ijk}, w_{ijk})$,
- $i = 1,\ldots,N$; $j = 1,\ldots,M$; $k = 1,\ldots,19$
- Spherical curvilinear grid

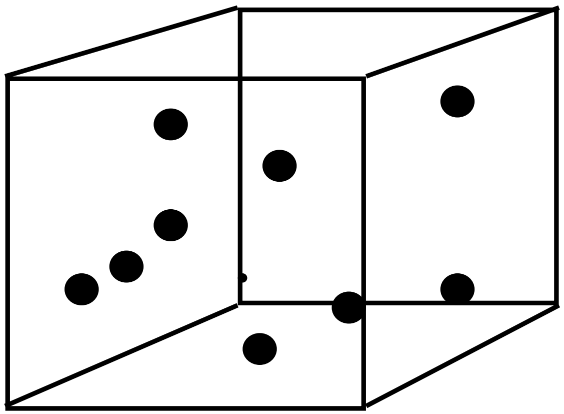
Stock Market Data



R0/4	EP	F2GRW	RTN
24.1	38.5	62.1	37.1
18.3	26.5	68.6	13.7
•	•	•	•
•	•	•	•
•	•	•	•

- Data courtesy E-K Koh
- R0/4 - July 1987 Performance
- EP - Earnings to Price Ratio
- F2GRW - Growth Potential
- RTN - October 1987 Performance
- (xi, yi, zi, wi), i = 1,...,N (i represents stock number i!)

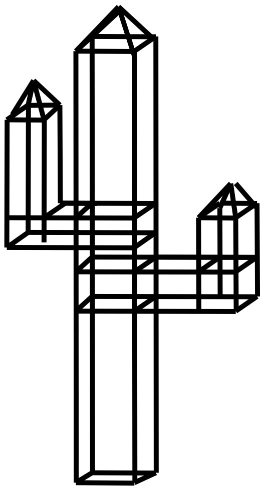
Sound Data



Location			Decibel
23.45	13.56	3.56	58.0
10.34	50.45	5.67	78.9
⋮	⋮	⋮	⋮

- (X_i, Y_i, Z_i, D_i), $i = 1,\ldots,N$
- Irregular data
  - Needs scattered data interpolation for the computation of a data value at an arbitrary location

Finite Element Data (FEM)

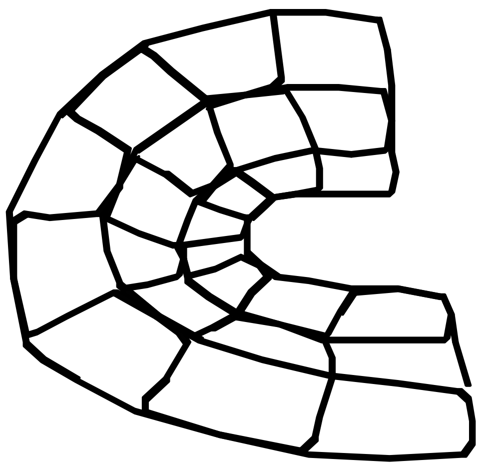


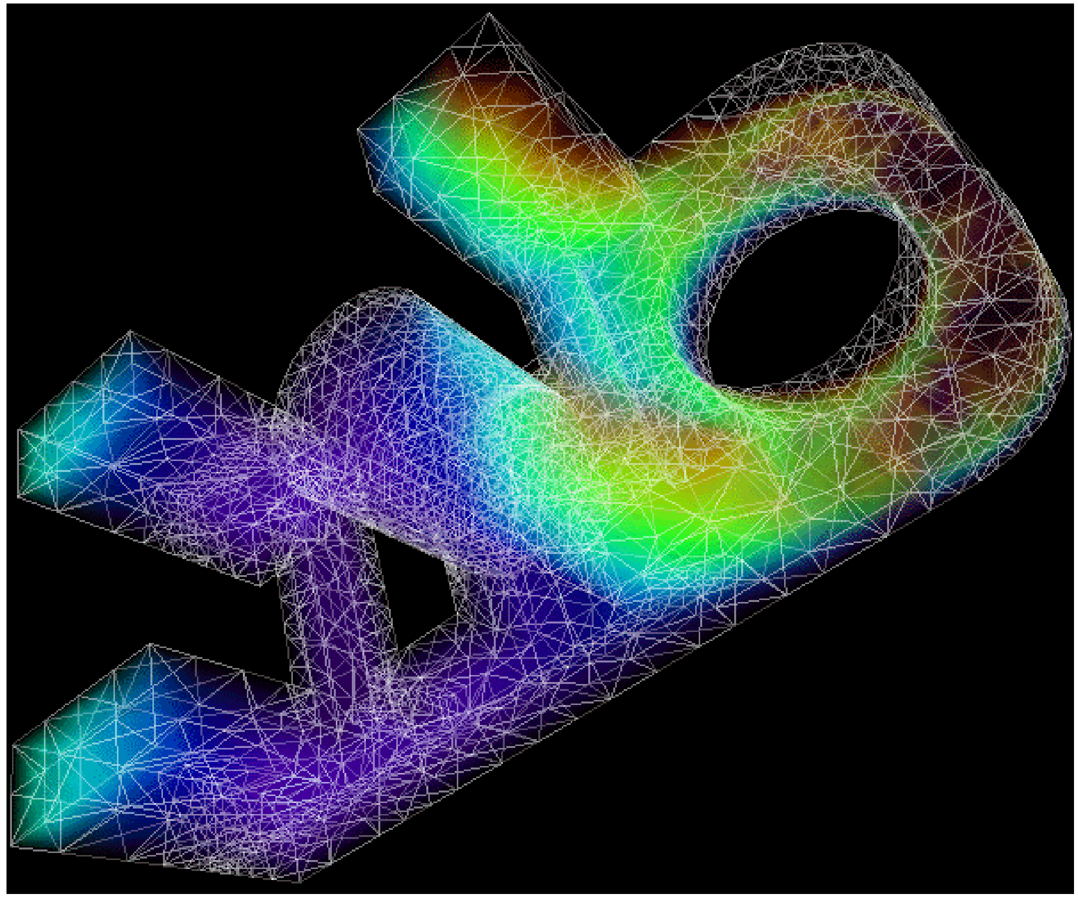
Location			Temp
21.9	69.2	23.1	88.7
95.3	11.9	99.7	78.9
⋮	⋮	⋮	⋮

- Set of cells
- Points: $P_1, P_2, \ldots, P_N$
- Faces: $F_k = P_{k1}, P_{k2}, \ldots, P_{kM}$
- Cells: $C_j = F_1, F_2, \ldots, F_c$

Finite Element Data (FEM)

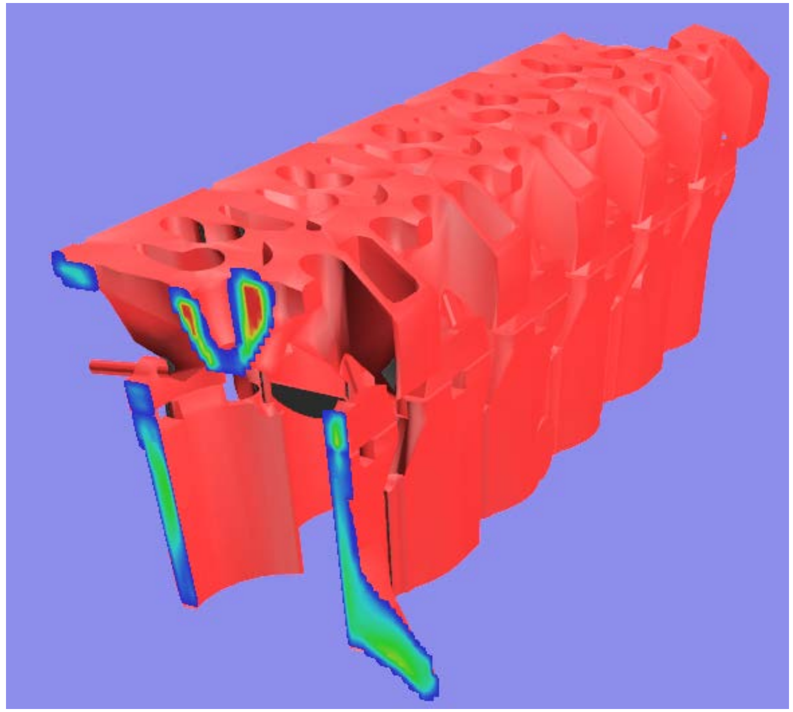
- Hexahedral grid
- Example:
  color represents
  temperature





Combustion Process Data

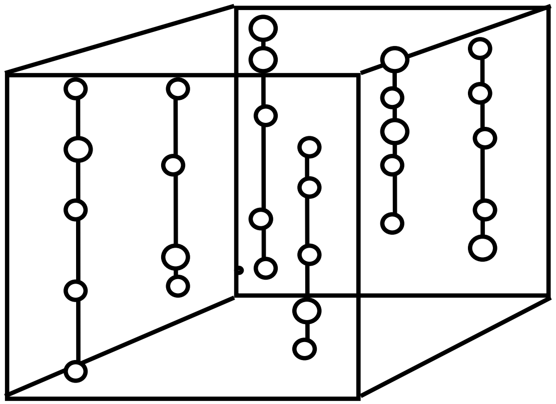
- Finite element cells
- Temperature computed for a number of time steps t_l
- $T(x_i, y_j, z_k, t_l)$



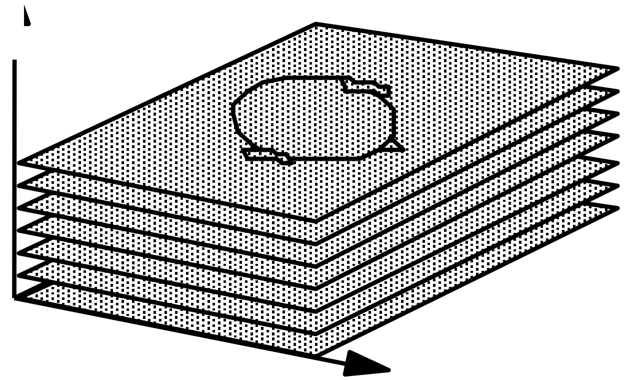
Classification

- ***Source***
  - ***Measured***
  - ***Simulated***
- Dimension
  - Domain (dt: Definitionsbereich)
  - Range (dt: Wertebereich)
- Topology
  - Grids
  - Structure

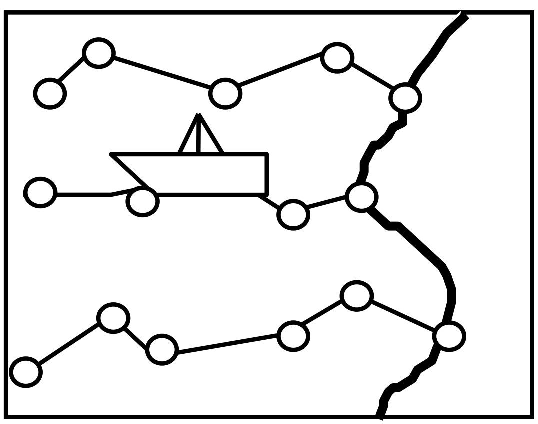
Measured Data



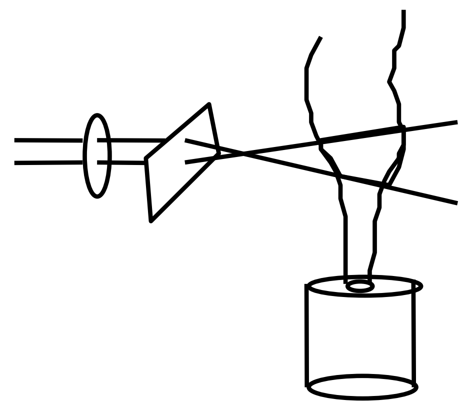
Well Log Data



Cat Scan Data

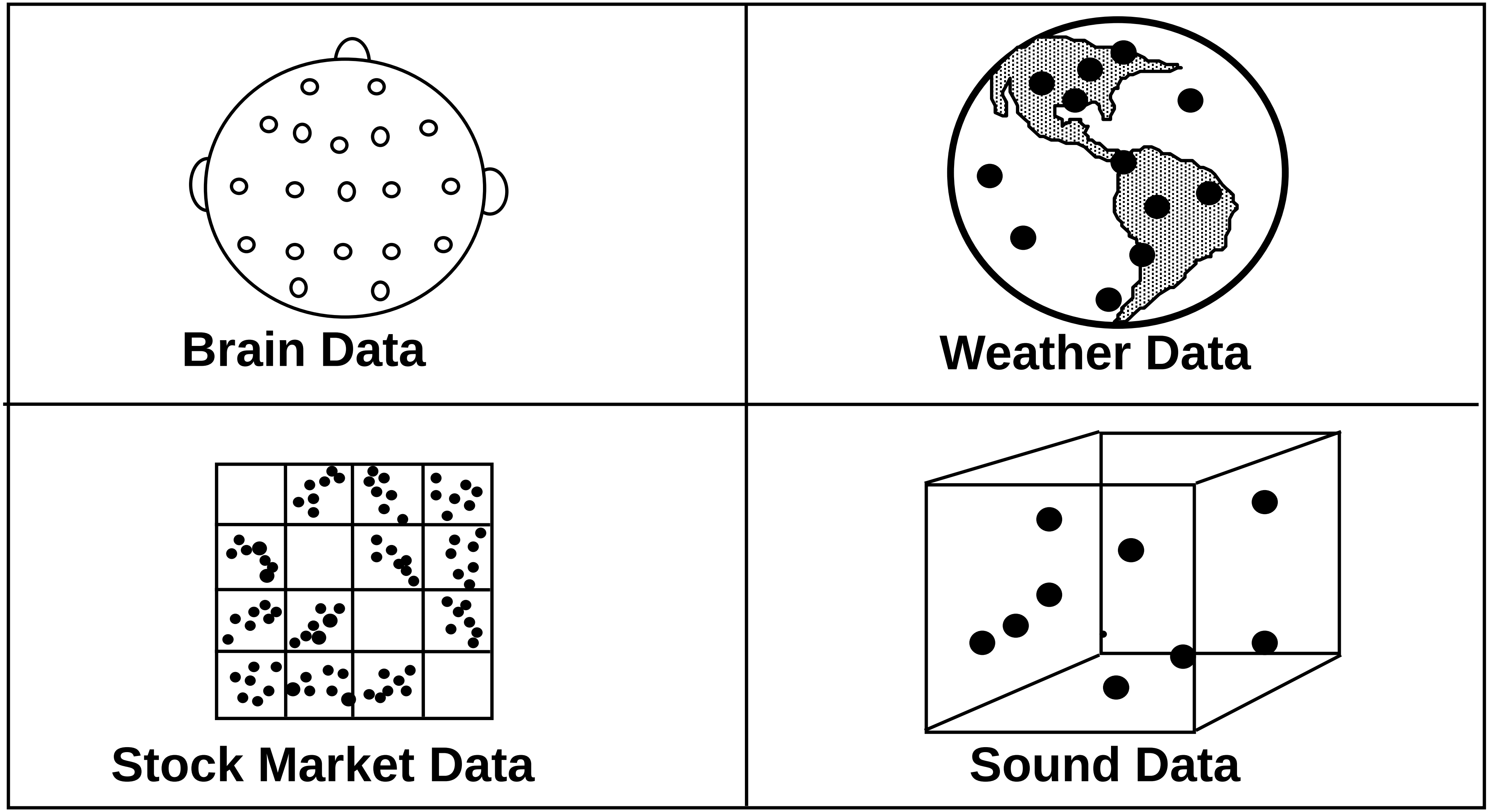


Big Sur Data

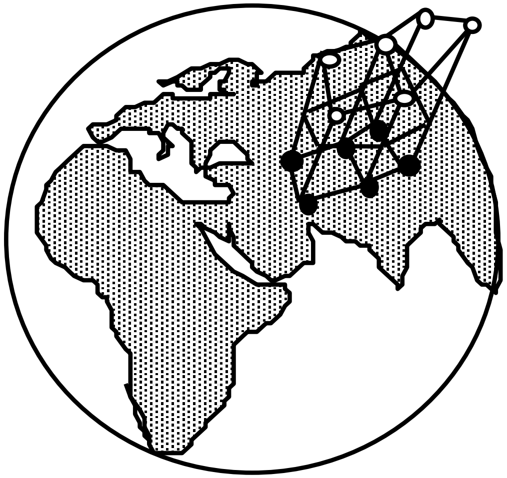


Flame Data

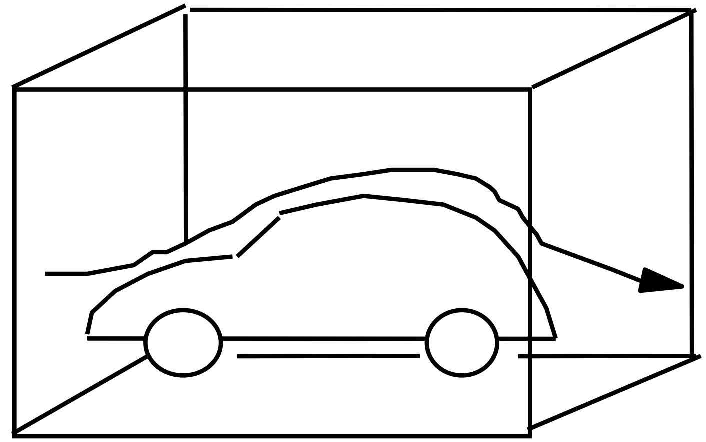
Measured Data



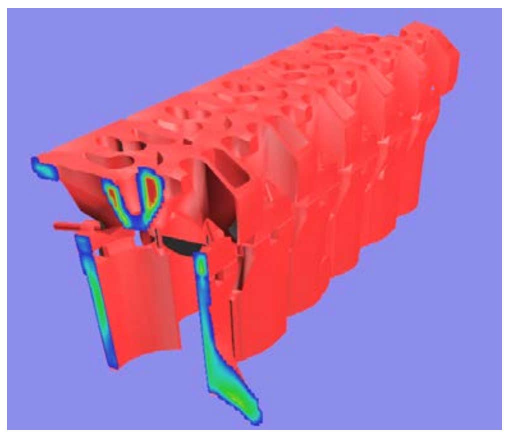
Simulated Data



Climate Model Data

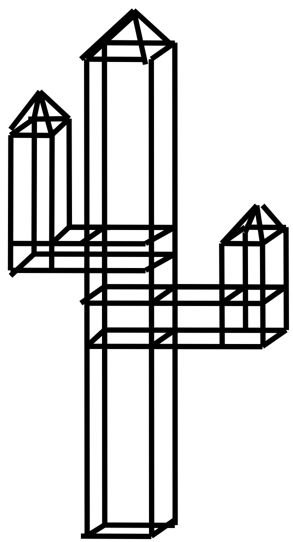


Nissan Car Data

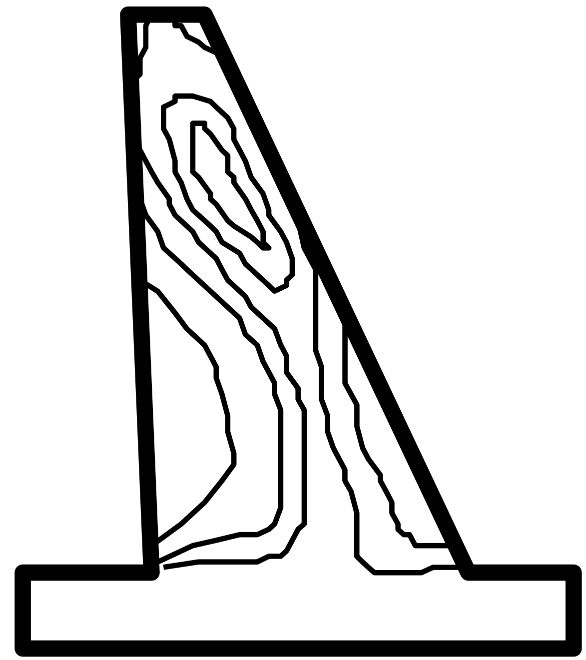


Combustion Process Data

Simulated Data



FEM Data



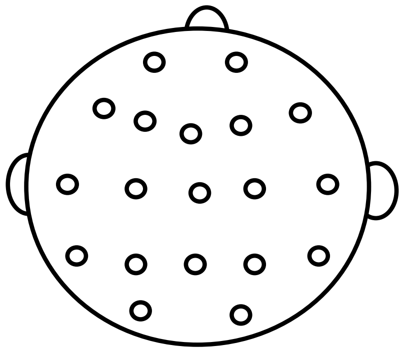
Wing Data

Classification

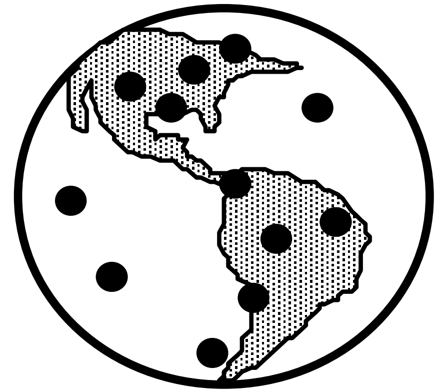
- Source
  - Measured
  - Simulated
- ***Dimension***
  - ***Domain (discrete, continuous)***
  - ***Range (discrete, continuous)***
- Topology
  - Grids
  - Structure

Domain - Definitionsbereich Range - Wertebereich

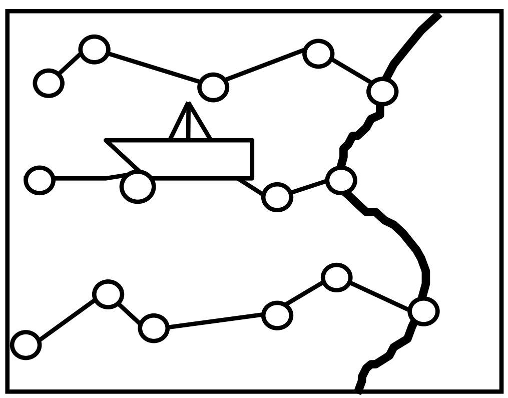
Domain: 2D



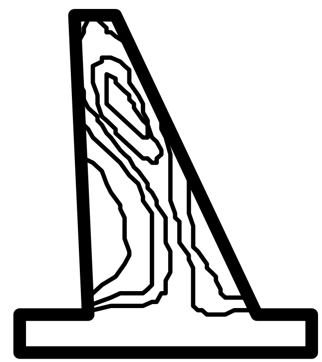
Brain Data



Weather Data

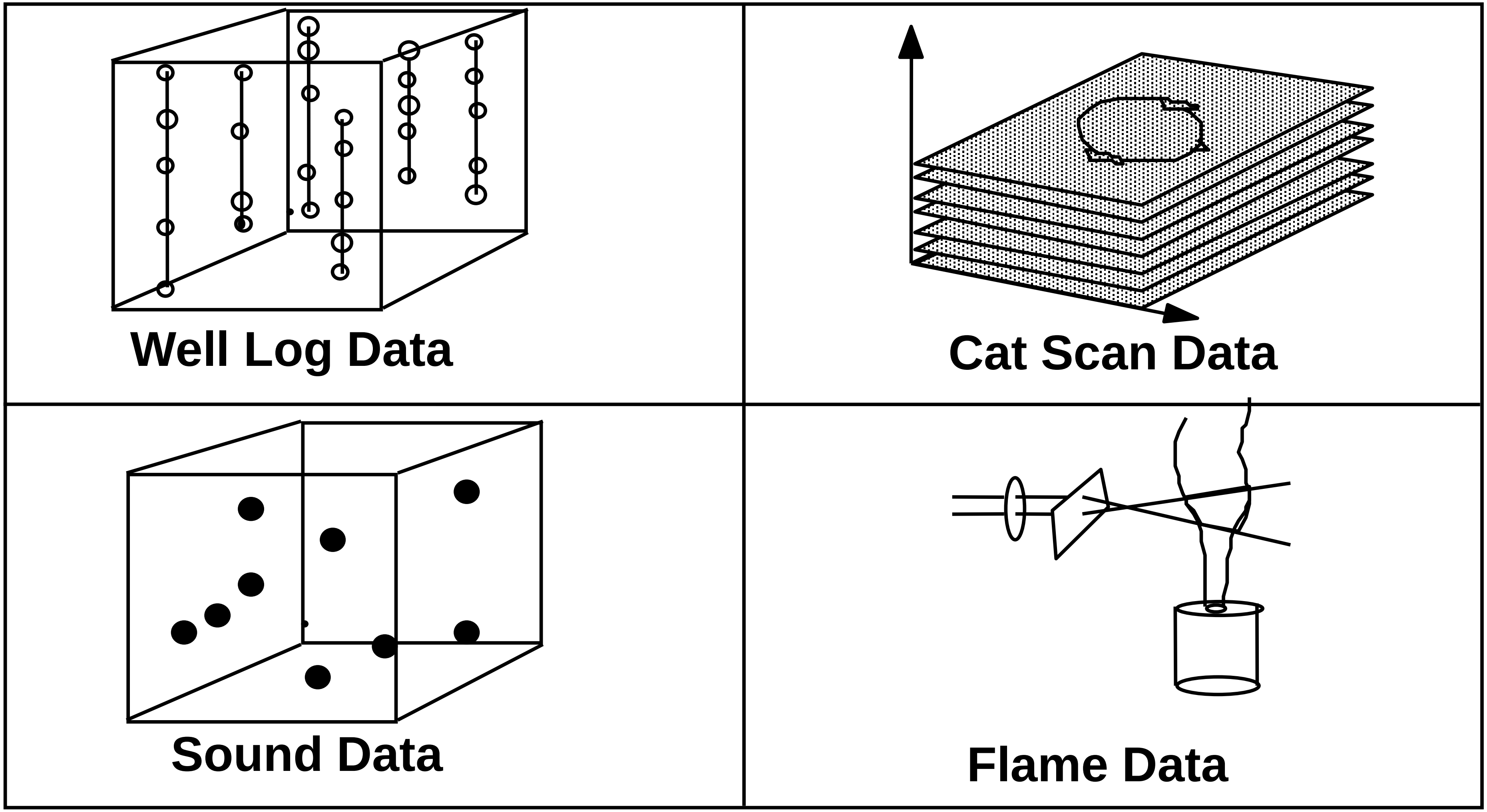


Big Sur Data

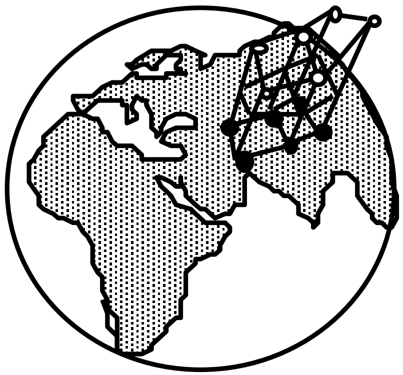


Wing Data

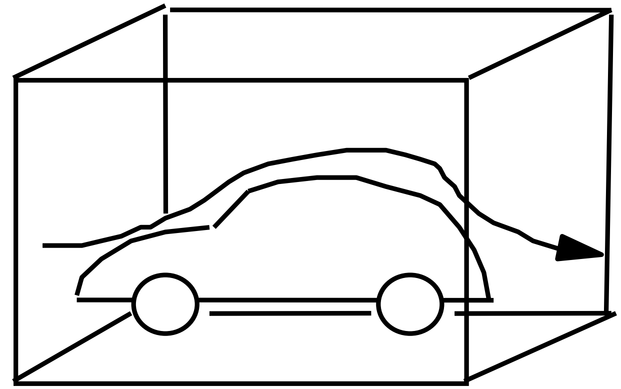
Domain: 3D



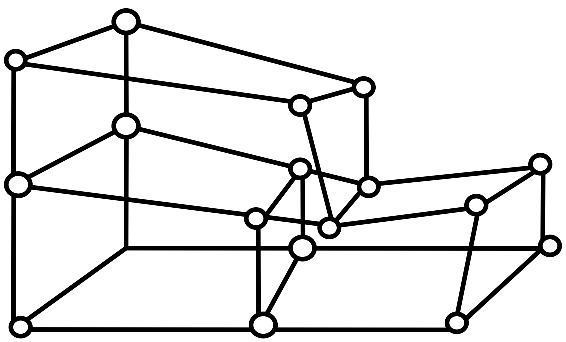
Domain: 3D



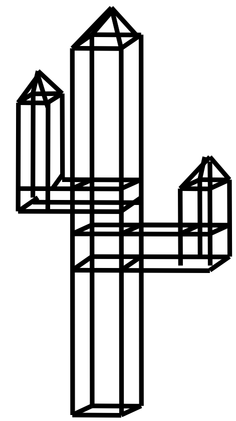
Climate Model Data



Nissan Car Data

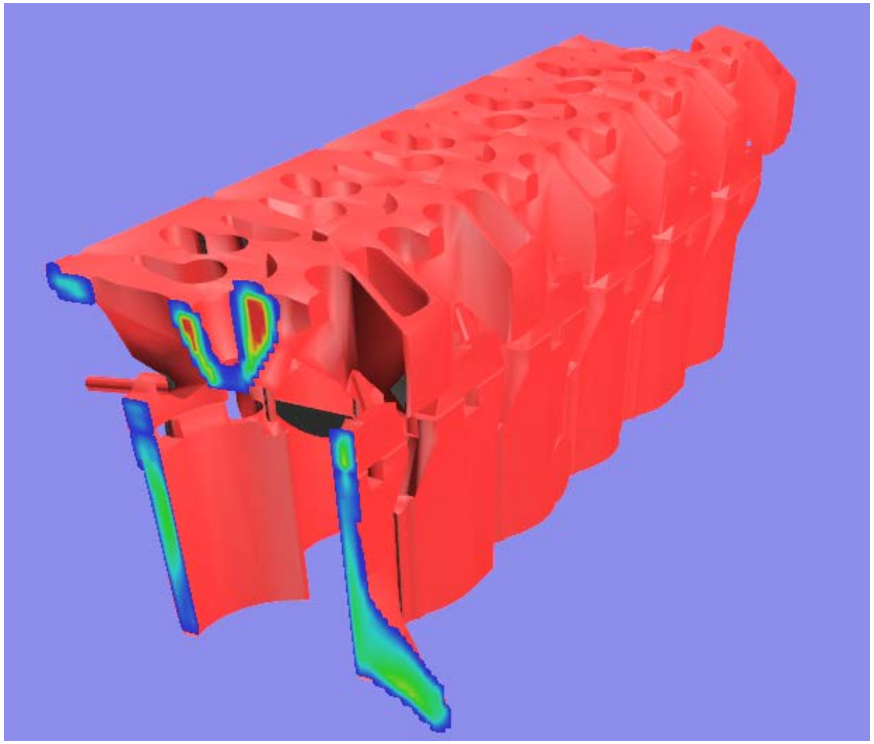


Reservoir Data



FEM Data

Domain: 4D … N-D



Combustion Process Data

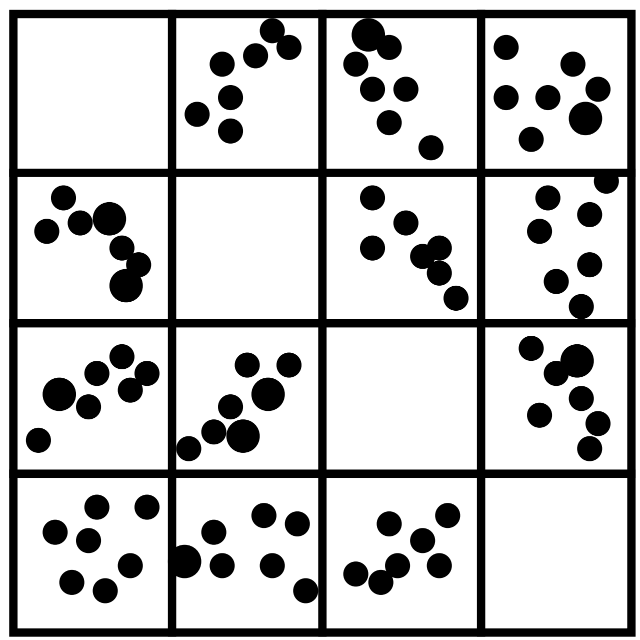
Range: Scalar

- Density
- Temperature
- Concentration
- Pressure
- Potential
- Height
- Decibel
- • • •

Range: Multi-Scalar / Multi-Attribute

No.	Stock	R0/4	EP	F2GRW	RTN
1	google	24.1	38.5	62.1	37.1
2	Ebay	18.3	26.5	68.6	13.7
⋮	⋮	⋮	⋮	⋮	⋮

Stock market data
(scatter plot)
For each stock (domain)
4D vector is provided



Range: Vector

- Velocity (u(x,y,z),v(x,y,z),w(x,y,z))
- Gradient of Scalar Field
  - F(x,y,z) $\nabla F = (\frac{dF}{dx}, \frac{dF}{dy}, \frac{dF}{dz})$
- Force, current, etc.

Range: Tensor

- Velocity gradient (n_{ijk}) =

$$\begin{pmatrix} \frac{du}{dx} & \frac{du}{dy} & \frac{du}{dz} \\ \frac{dv}{dx} & \frac{dv}{dy} & \frac{dv}{dz} \\ \frac{dw}{dx} & \frac{dw}{dy} & \frac{dw}{dz} \end{pmatrix}$$

- Stress
- Conductivity
- Moment of inertia
- ...

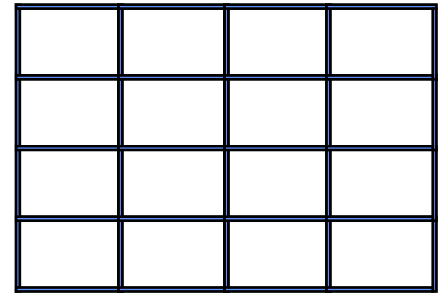
Classification

- Source
  - Measured
  - Simulated
- Dimension
  - Domain
  - Range
- ***Topology***
  - ***Grids***
  - ***Structure***

Terminology

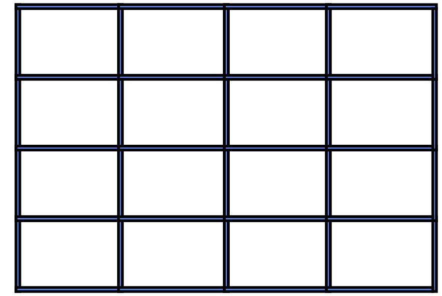
- Data
  - Regular
  - Irregular, scattered, unorganized
- Data structures
  - List, Array, Graph, ...
- Topology: Interconnection of points, edges, faces, etc into cellular decomposition of domain
- Grids & Meshes
  - Uniform, Rectangular, Curvilinear, Triangular, Rectilinear, Unstructured, Regular, Cartesian

Grids



- Processes (fluid simulation, deformation, …) or original domain data (temperature distribution, …) is often continuous
- Data has to be sampled and it is represented at discrete positions only – e.g. in grids
- The stored data values have to interpreted as coefficients of basis functions or reconstruction kernels (constant, linear, quadratic, …), which allow you to interpolate values in between data values.

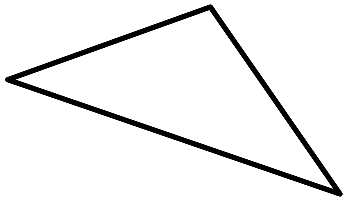
Grid Attributes



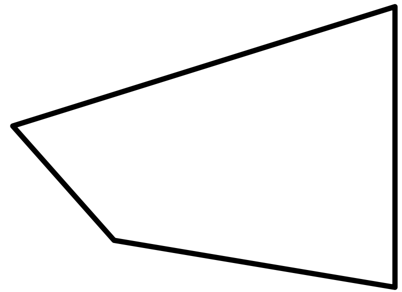
- Required
  - Dimension
  - List of vertices (explicit or implicit)
  - List of cells (explicit or implicit)
- Optional
  - List of edges
  - List of faces
  - Connectivity
    - Cell references vertices
    - Cell references edges
    - …

Cells

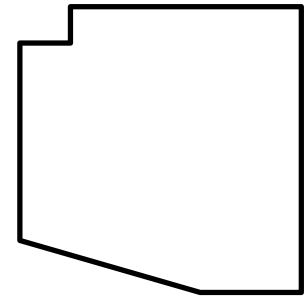
2D



Triangle

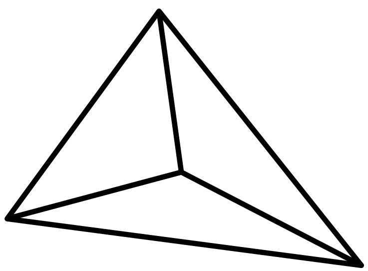


Quadrilateral

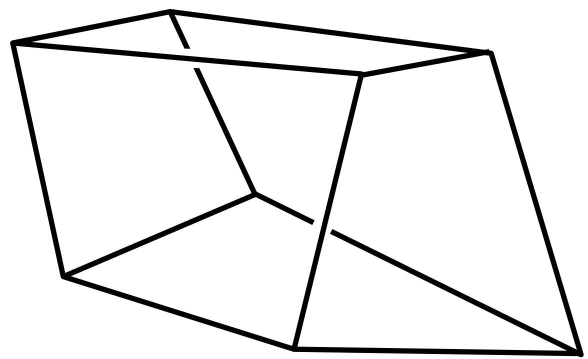


Polygon

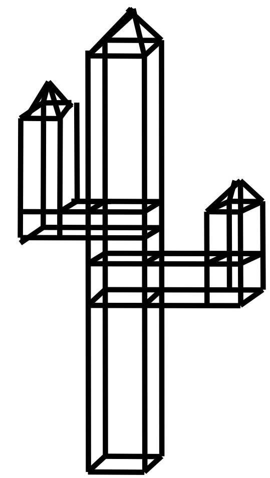
3D



Tetrahedron



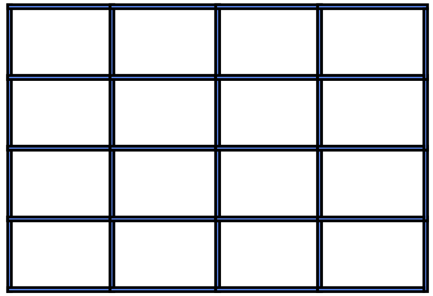
Hexahedron

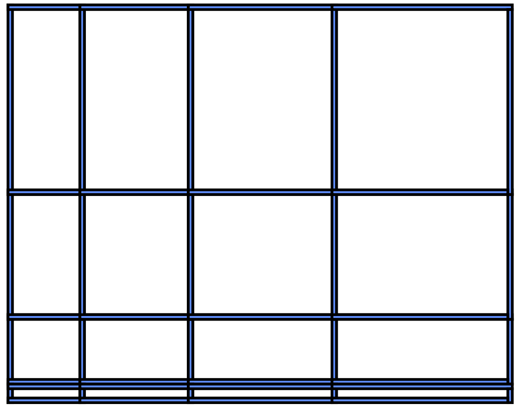


Polyhedron

Grids

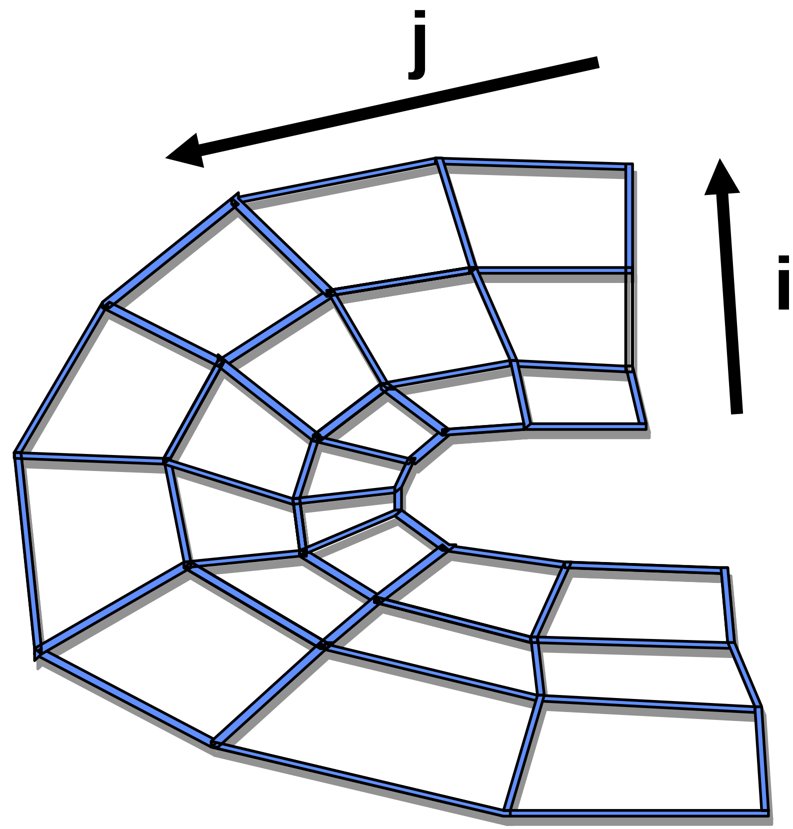
- Regular or Uniform
  - $x_{i+1} = x_i + \Delta x$
  - Required information
    - Resolution x, y, z
    - Δx, Δy, Δz
    - Data in 3D array
- Rectilinear or Perimeter
  - $x_{i+1} = x(i)$
  - Required information
    - Resolution x, y, z
    - Coordinates of grid lines
      - Cell localisation for a given point e.g. by binary search
    - Data in 3D array





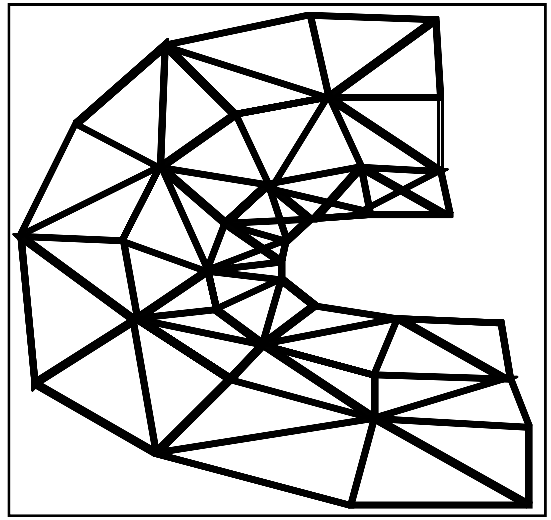
Curvilinear Grids

- Curvilinear
  - $x_i = x(i,j)$
  - $y_j = y(i,j)$
- Curves may not cross in i or j
- Deformed, potentially degenerated grid
- Information required
  - Resolution x, y, z
  - Coordinates of grid points
    - Cell localisation for a given point using an auxillary data structure
  - Data in 3D array



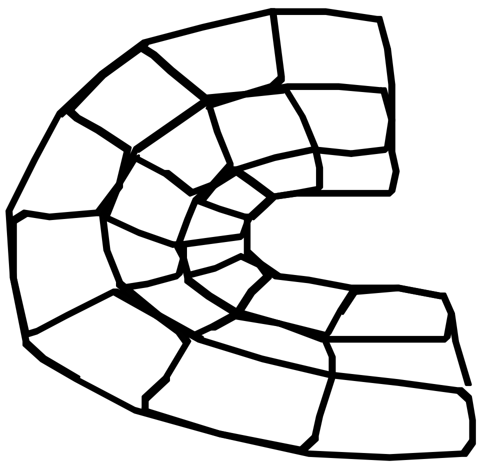
Unstructured Grids or Meshes

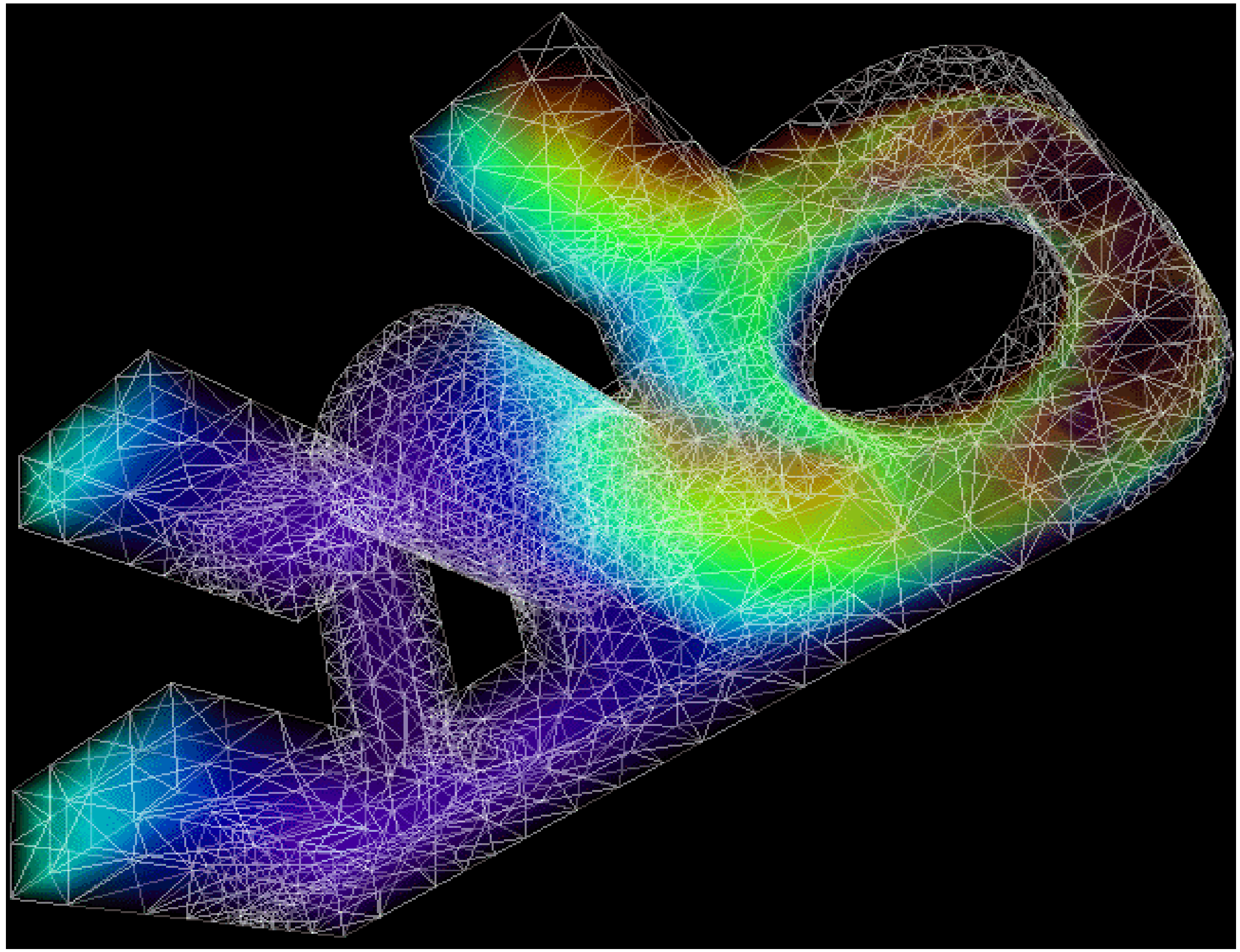
- Unstructured grids often used for finite-element data
  - Tetrahedral, hexahedral or mixed (zoo of cells)
- Required information
  - Number of vertices
  - Number of cells
  - List of vertices
  - List of cells
    - Cell localisation for a given point using an auxillary data structure



Unstructured Grids or Meshes

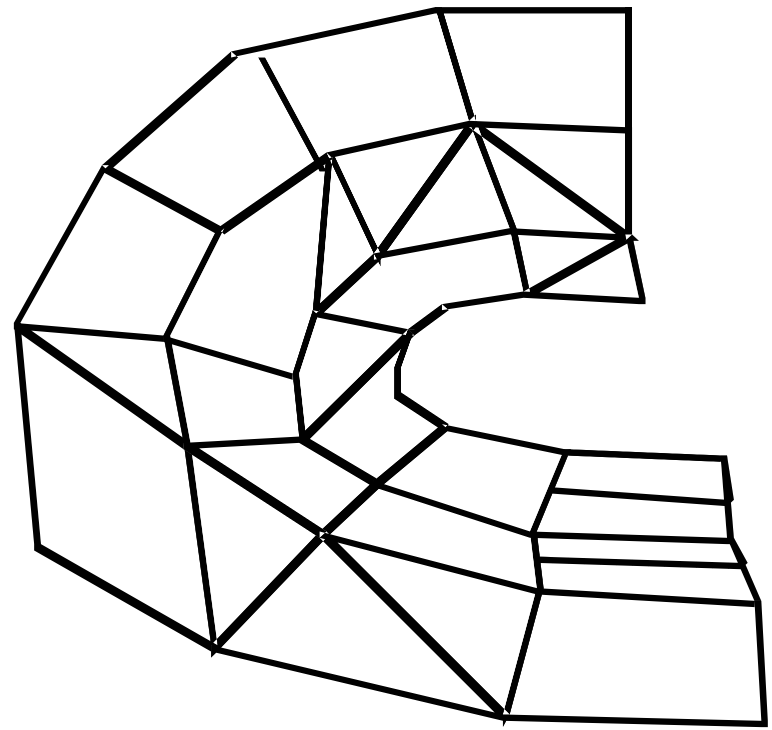
- Hexahedral





Unstructured Grids or Meshes

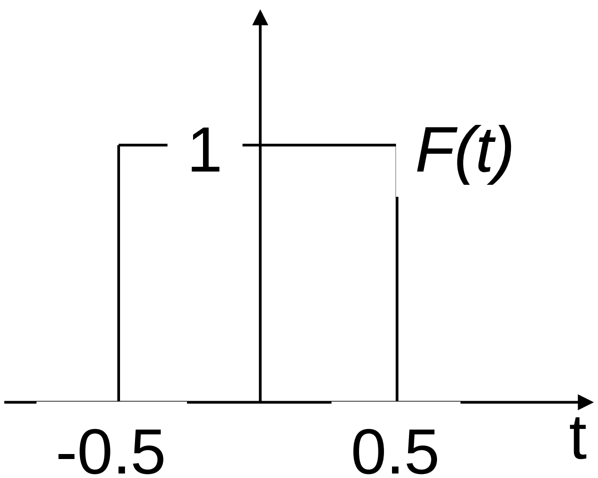
- Finite-element zoo

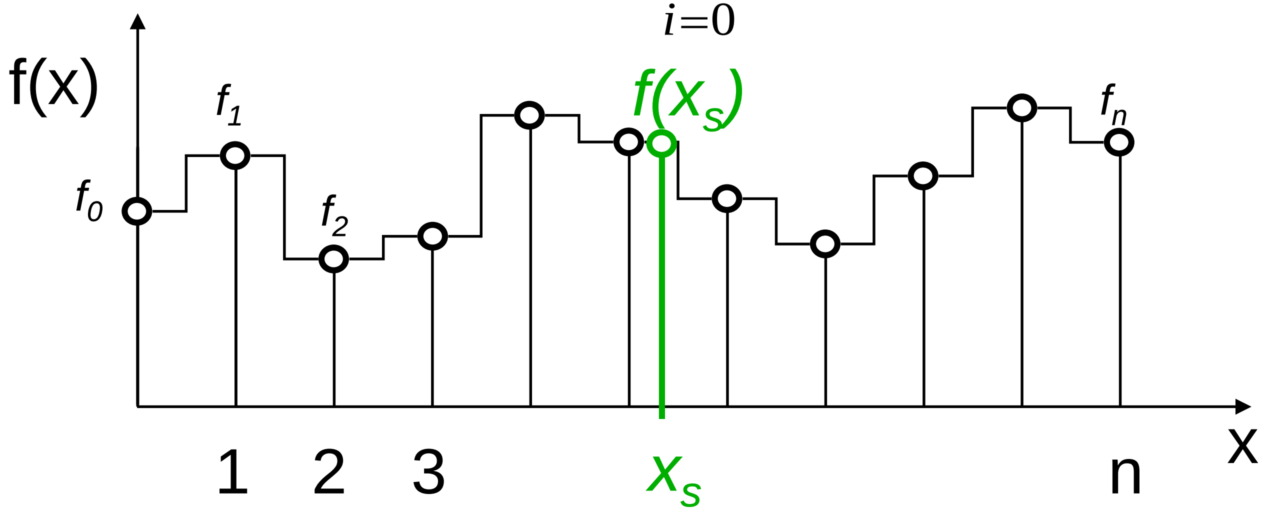


Piecewise Constant Reconstruction
Nearest Neighbor Interpolation

- Given a set of data values
  (x_0,f_0) … (x_i,f_i), (x_{i+1},f_{i+1}) … (x_n,f_n)
- A box-shaped reconstruction kernel $F(t)$
- In practice
  - Nearest neighbour search

$$f(x)=\sum_{i=0}^{n} f_i F(x-i)$$

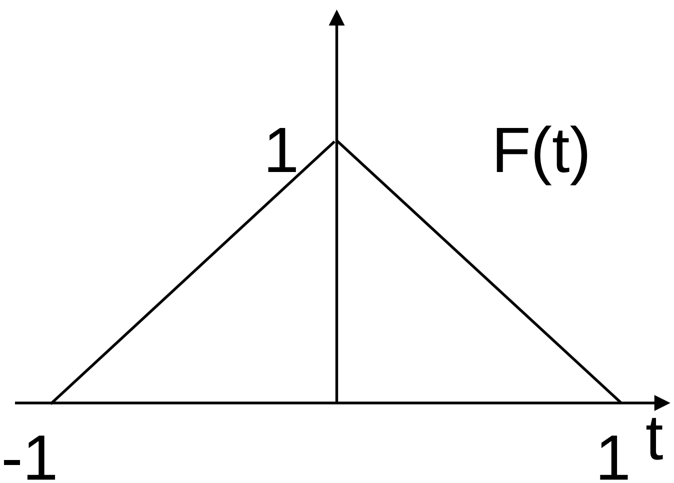


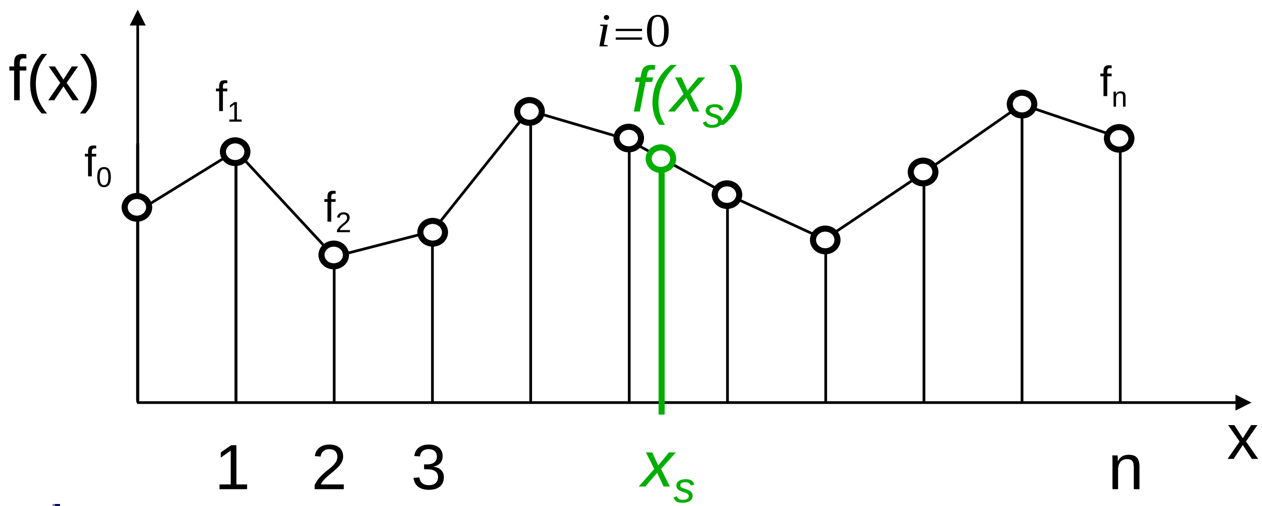


Linear Interpolation

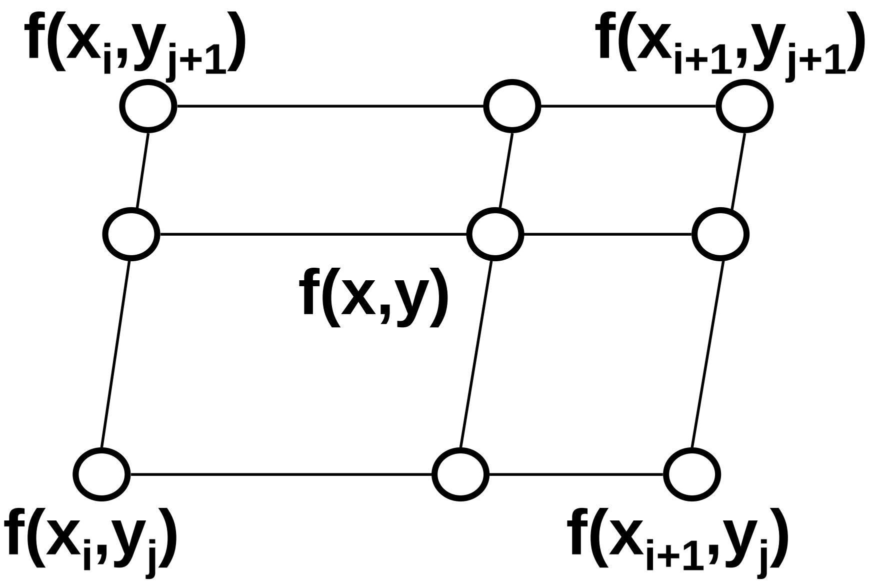
- Given a set of data values
  $(x_0,f_0) \ldots (x_i,f_i), (x_{i+1},f_{i+1}) \ldots (x_n,f_n)$
- A tent-shaped reconstruction kernel *F(t)*
- In practice for $x_{i+1} >= x > x_i$
  - $f(x) = (1-u)\, f_i + u\, f_{i+1}$
  - $u = (x-x_i) / (x_{i+1}-x_i)$

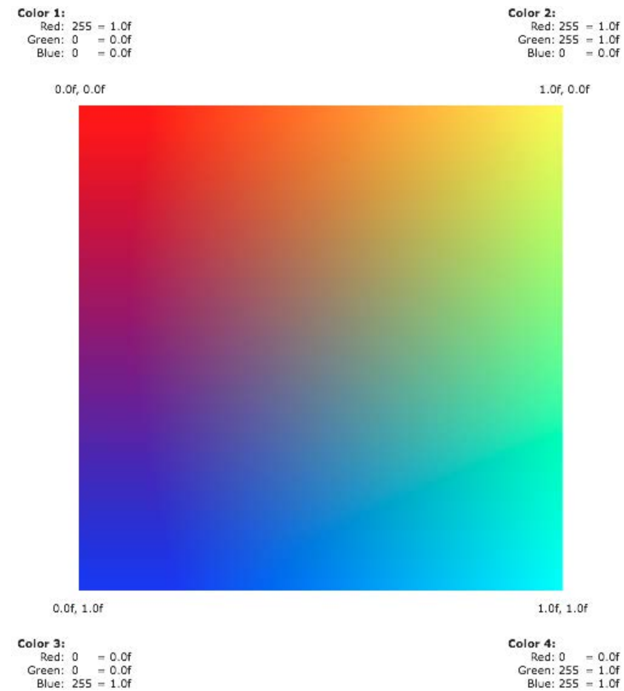
$$f(x) = \sum_{i=0}^{n} f_i F(x-i)$$





Bilinear Interpolation

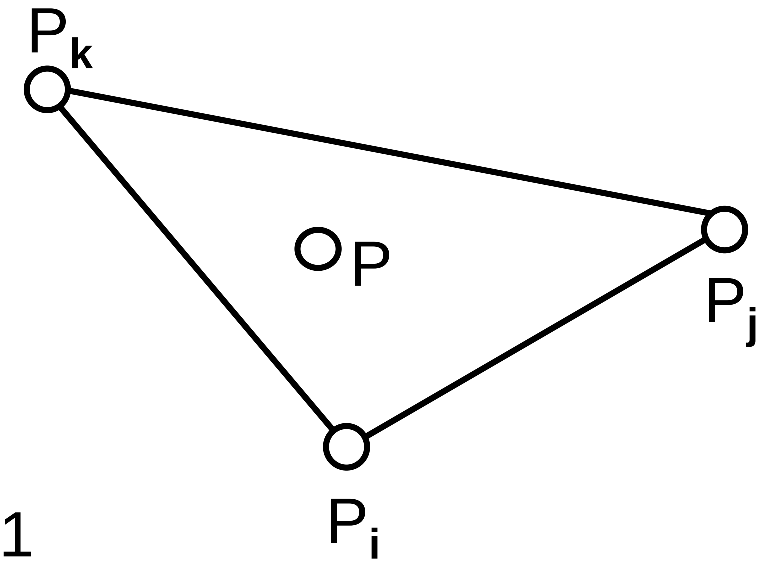




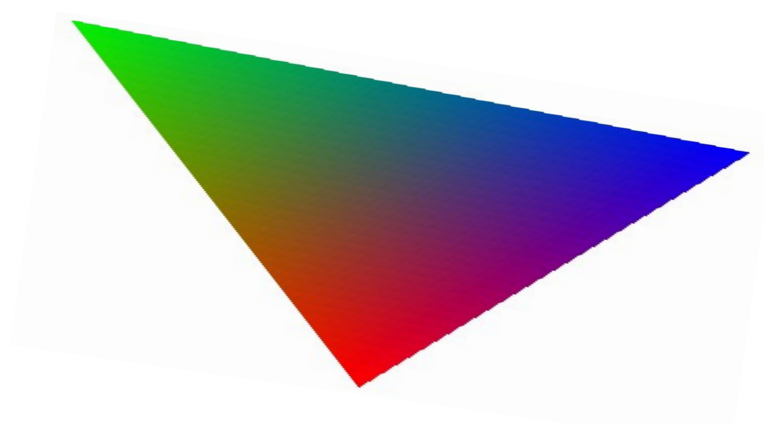
$$f(x,y) = (1-v)\,((1-u)\,f(\mathbf{x_i},\mathbf{y_j}) + u\,f(\mathbf{x_{i+1}},\mathbf{y_j})) + v\,((1-u)\,f(\mathbf{x_i},\mathbf{y_{j+1}}) + u\,f(\mathbf{x_{i+1}},\mathbf{y_{j+1}}))$$

Linear Interpolation for Triangles

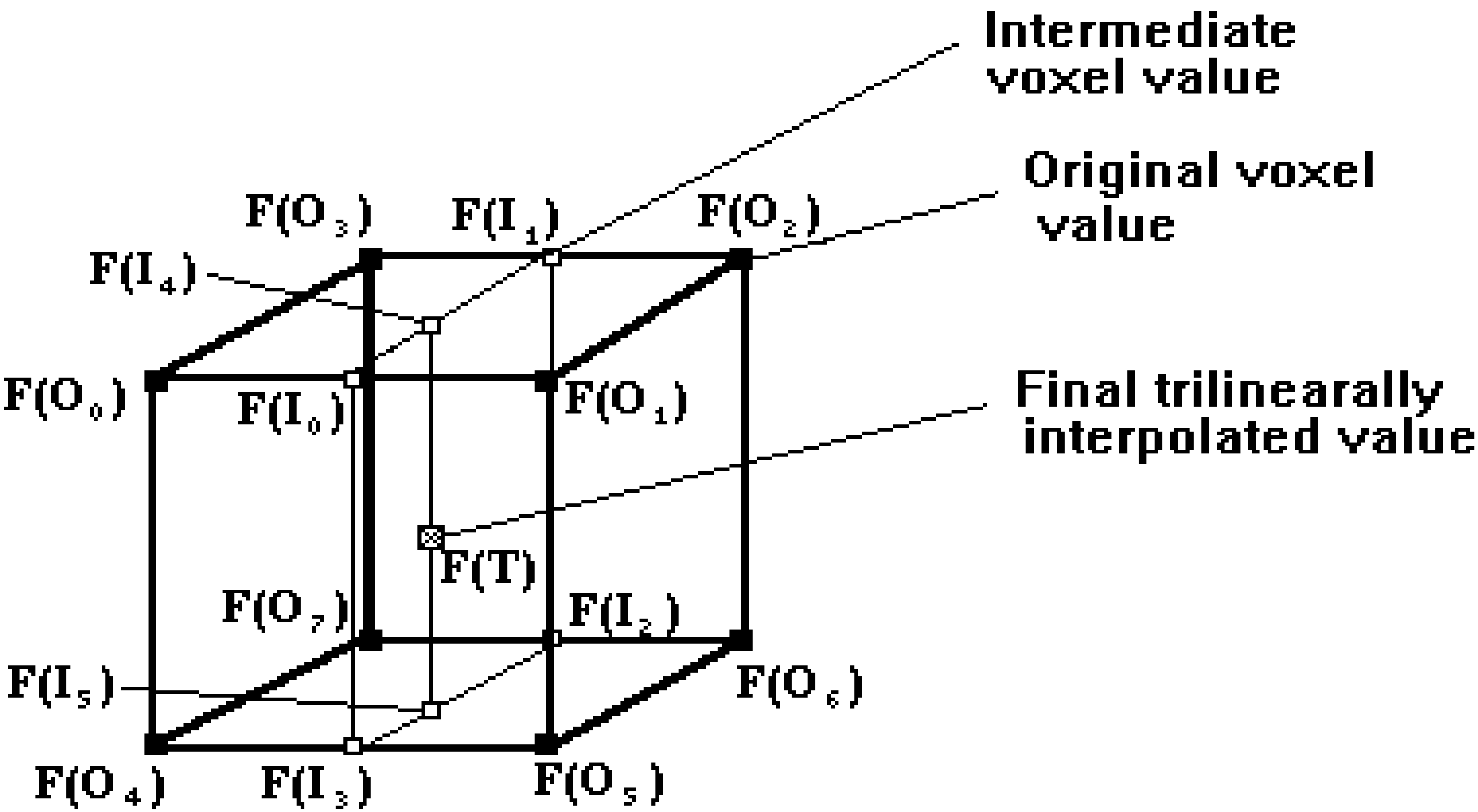
$$b_i = \frac{\begin{vmatrix} x - x_j & y - y_j \\ x - x_k & y - y_k \end{vmatrix}}{\begin{vmatrix} x_i - x_j & y_i - y_j \\ x_i - x_k & y_i - y_k \end{vmatrix}} \qquad b_j = \frac{\begin{vmatrix} x - x_i & y - y_i \\ x - x_k & y - y_k \end{vmatrix}}{\begin{vmatrix} x_j - x_i & y_j - y_i \\ x_j - x_k & y_j - y_k \end{vmatrix}} \qquad b_k = \frac{\begin{vmatrix} x - x_i & y - y_i \\ x - x_j & y - y_j \end{vmatrix}}{\begin{vmatrix} x_k - x_i & y_k - y_i \\ x_k - x_j & y_k - y_j \end{vmatrix}}$$



- Barycentric coordinates $b_i + b_j + b_k = 1$
- Linear interpolation: $P = b_i P_i + b_j P_j + b_k P_k$
- Data interpolation
  - $F(P) = b_i f_i + b_j f_j + b_k f_k$
- **Generalizes to Tetrahedron**



Trilinear Interpolation

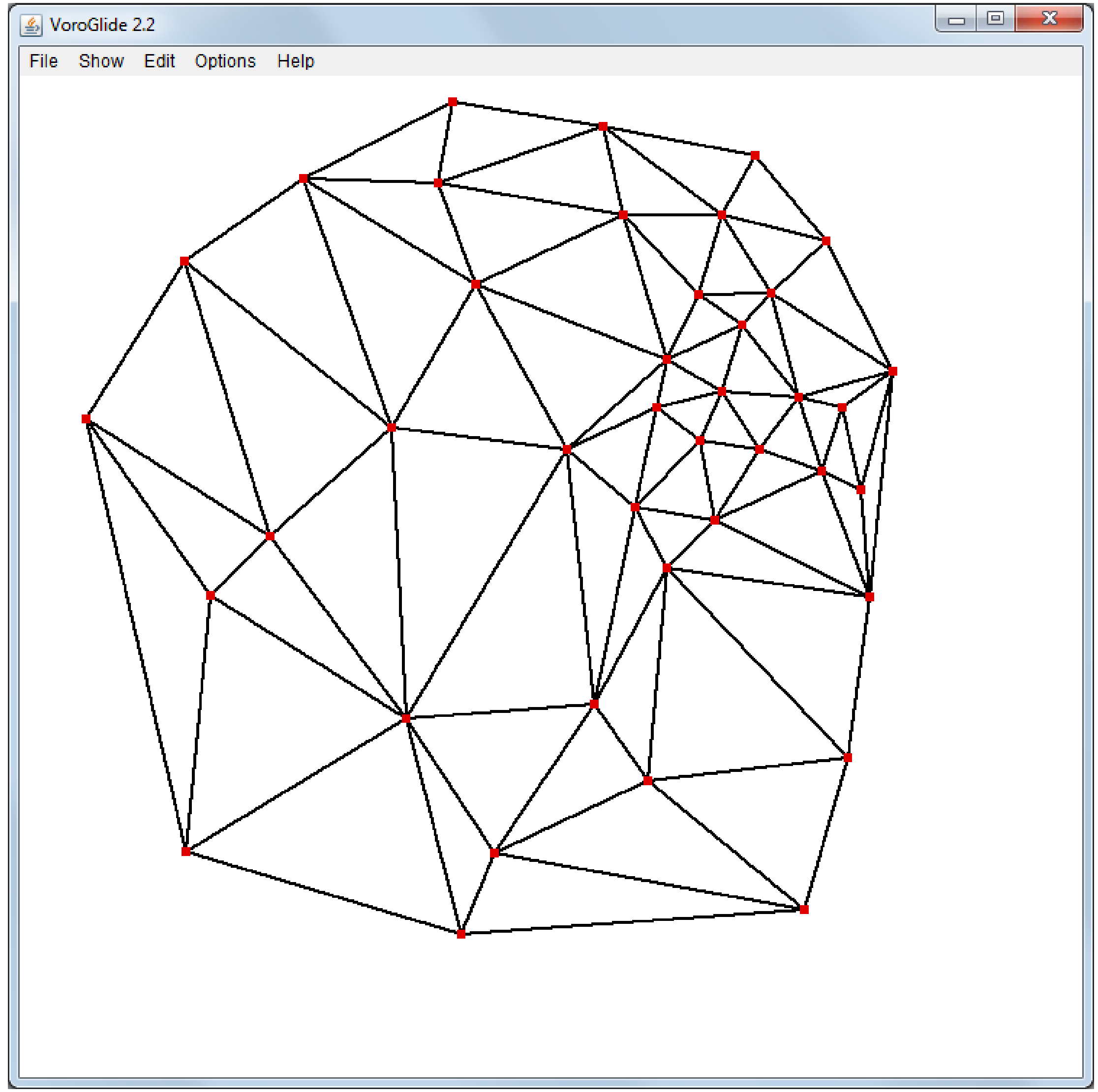


Higher Order Interpolation

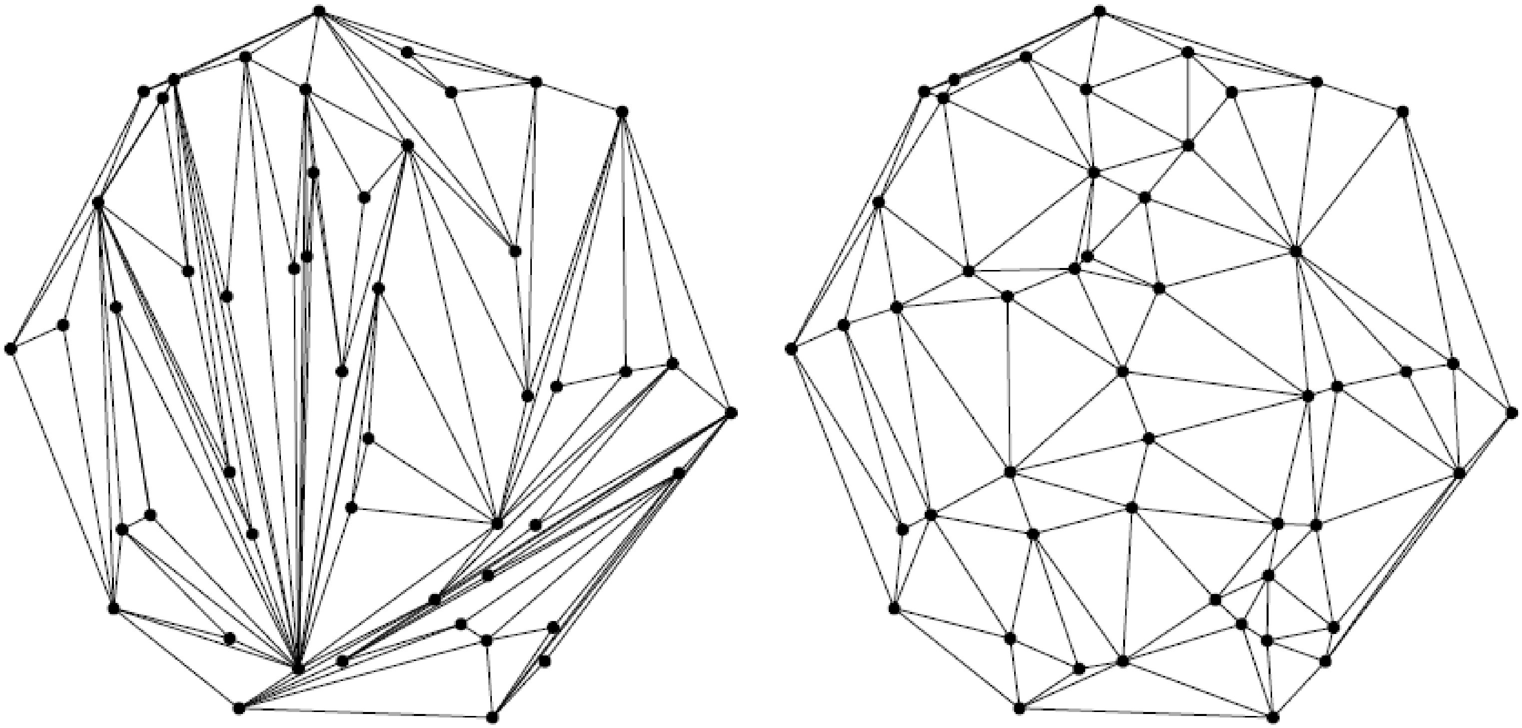
- Linear interpolation very common
- Higher order for (C1, C2, …) continuity across cell boundaries
  - Quadratic
  - Cubic
  - …

Delaunay Triangulation

- Named after the Russian mathematician Boris Nikolaevich Delone who invented the Delaunay triangulation in 1934
- G*lobally angle-optimal* triangulation (maximizes the smallest angle)



Delaunay Triangulation

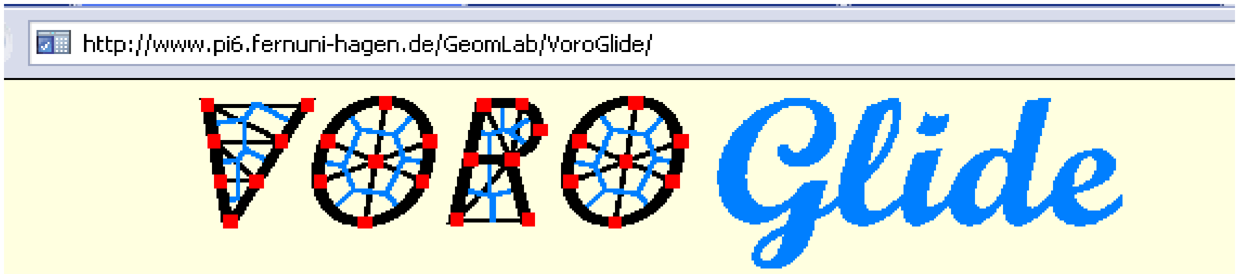


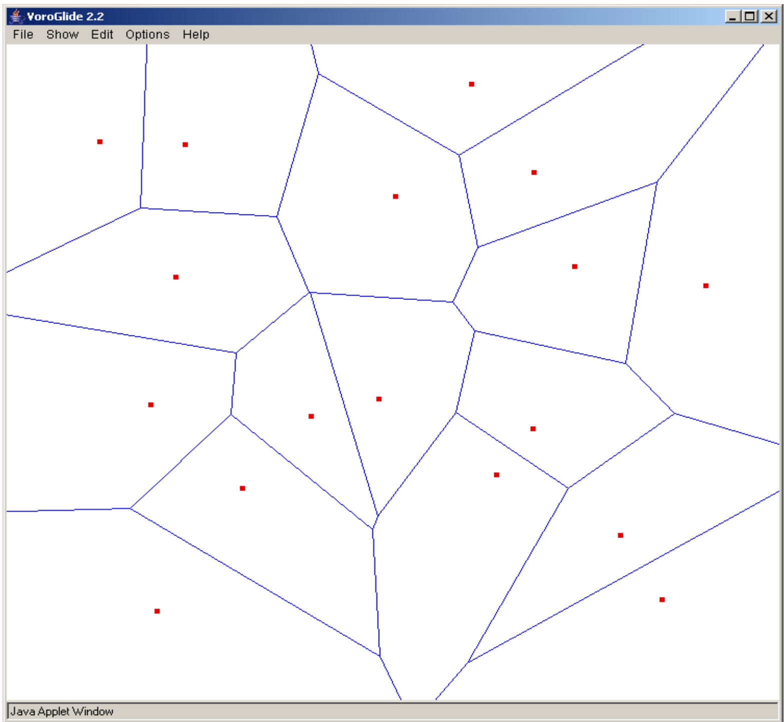
(a) Scan triangulation.

(b) Delaunay triangulation.

- In the plane, the Delaunay triangulation maximizes the minimum angle.
- The Delaunay triangulation does not necessarily minimize the maximum angle.
- A circle circumscribing any Delaunay triangle does not contain any other input points in its interior.

Voronoi Diagram and Delaunay Triangulation Animations

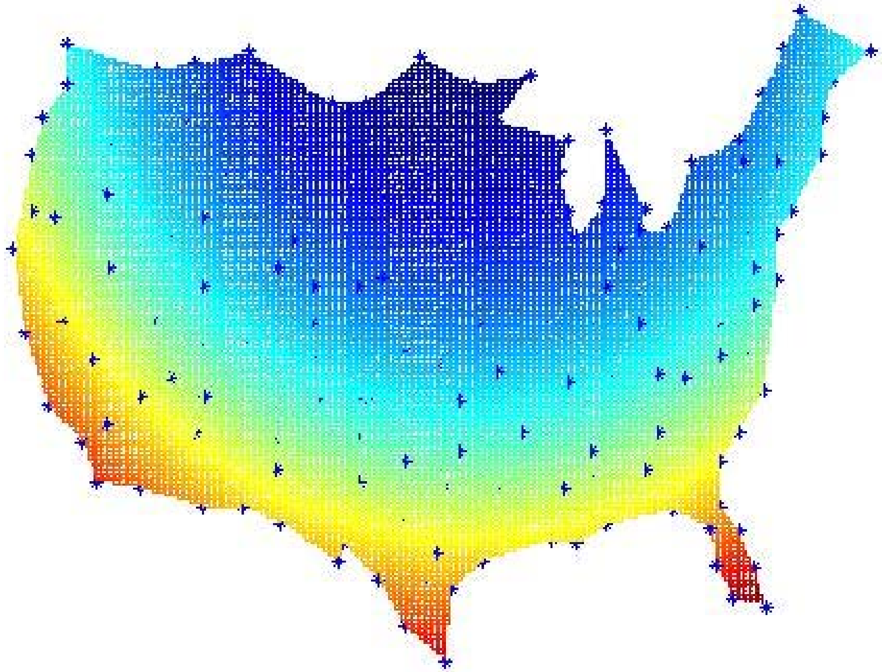




Java applet of Voronoi diagram construction by:
Christian Icking, Rolf Klein, Peter Köllner, Lihong Ma (Fernuni Hagen)

http://www.pi6.fernuni-hagen.de/GeomLab/VoroGlide/index.html.en

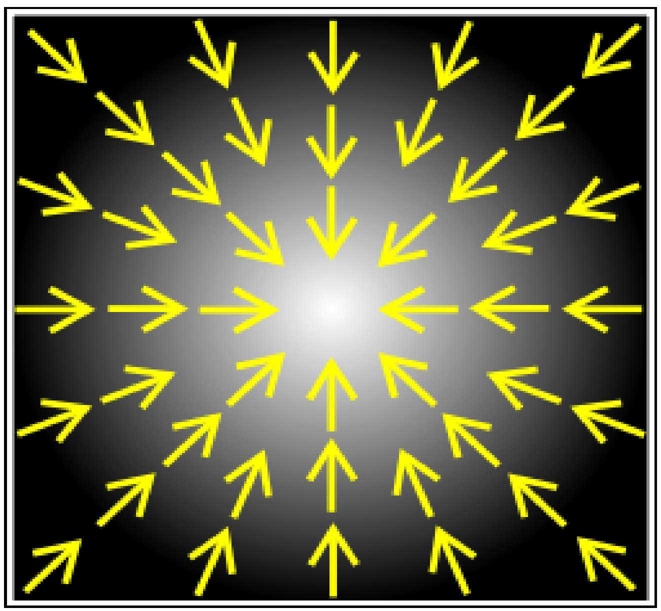
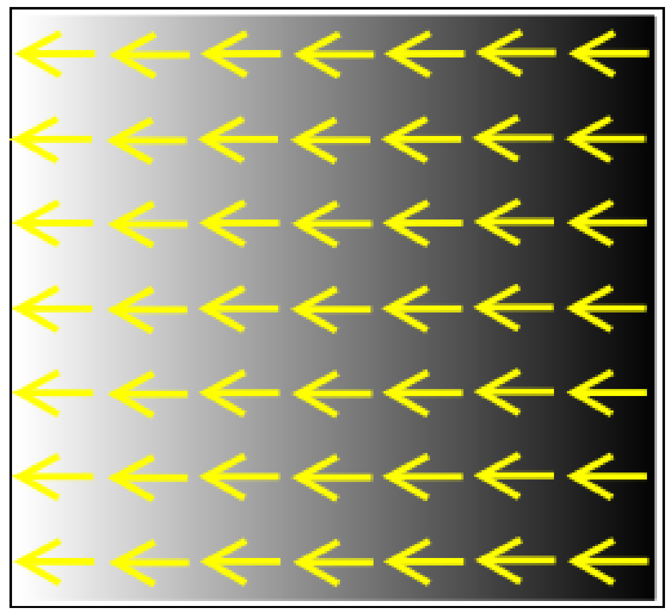
Scattered Data Interpolation



- A set of scattered temperature measurements is used to interpolate the temperature distribution across the US
- Scattered data interpolations are often based on triangulations (e.g. Delaunay triangulation) of the point set
- They may use linear or higher order interpolation over the triangles

Gradient Computation

- Gradients
  - Describe local changes in scalar fields
  - Magnitude and direction

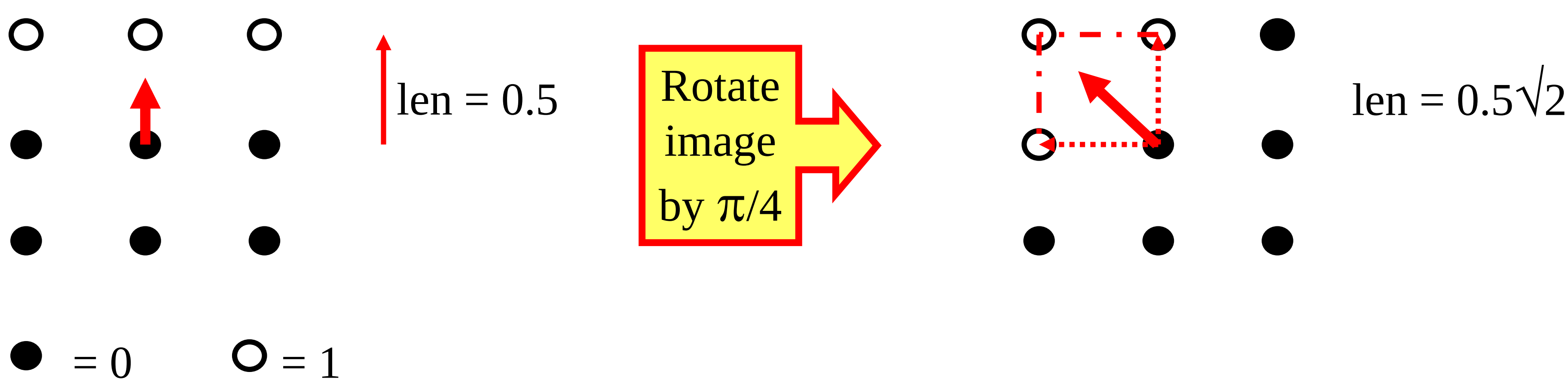
 

Gradient Computation

- Central difference

$$D_x = (f(x+1, y, z) - f(x-1, y, z)) / 2$$

  - Division by 2 is not necessary – scales all gradients by 0.5
  - Fast, easy to implement but may result in artefacts



Gradient Computation

- Sobel filter in 3D: 3x3x3 filter

slice x_{i-1}			*slice* x_i			*slice* x_{i+1}		
-1	-3	-1	0	0	0	1	3	1
-3	-6	-3	0	0	0	3	6	3
-1	–3	-1	0	0	0	1	3	1

- Apply to all 26 voxels around the voxel at which the gradient is required
- Rotate the above kernel for gradients in y and z direction
- Computationally quite expensive:
  - For each gradient vector
    54 multiplies (18 in each of 3 directions) and 51 additions (17 x 3)

End

Visualization

Color Maps, Contours and Isosurfaces

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media

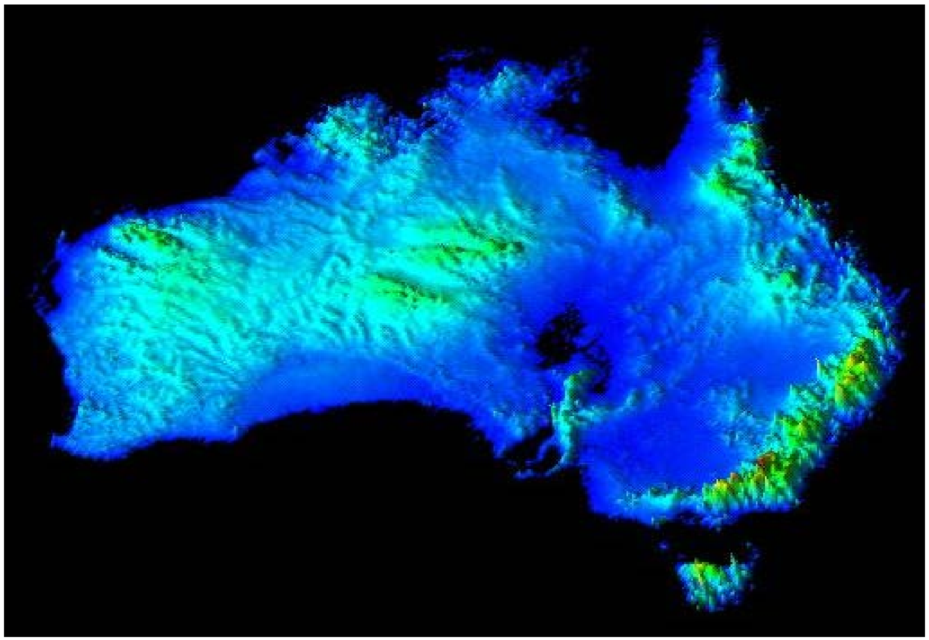
Bauhaus-Universität Weimar

Acknowledgements

- This lecture is based on
  - lectures by
    - John Morris, UWA
    - Crawfish, Barason et al, Umea University
    - Razk-Salama, Universität Siegen
  - Book Visualisierung by Schumann and Mueller

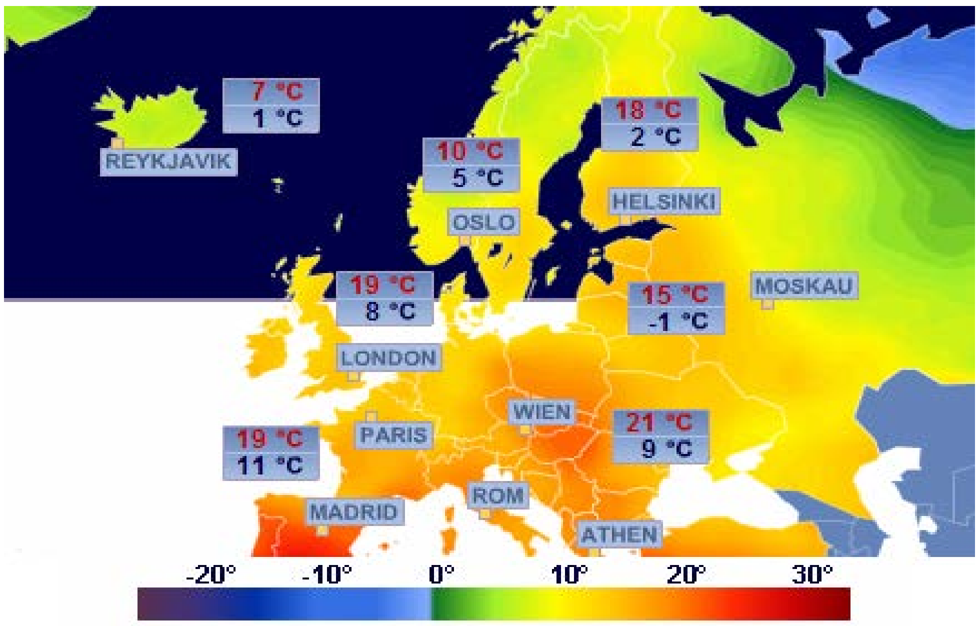
Colour Mapping

- Colour mapping is a mostly a one-dimensional technique
  - Mapping a scalar value to a single colour.
- However the display is not limited to one dimension. It can be 2-D or even 3-D



Colour Mapping

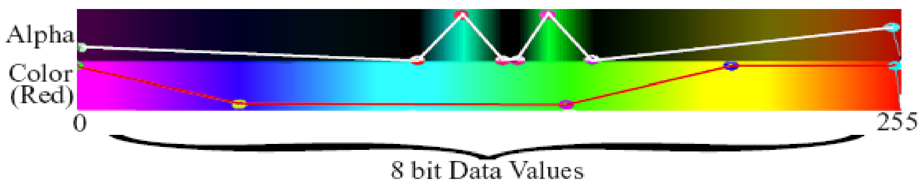
- Key to colour mapping
  - Choice of lookup table
- Table should
  - Accentuate the important features while minimising the less important differences.

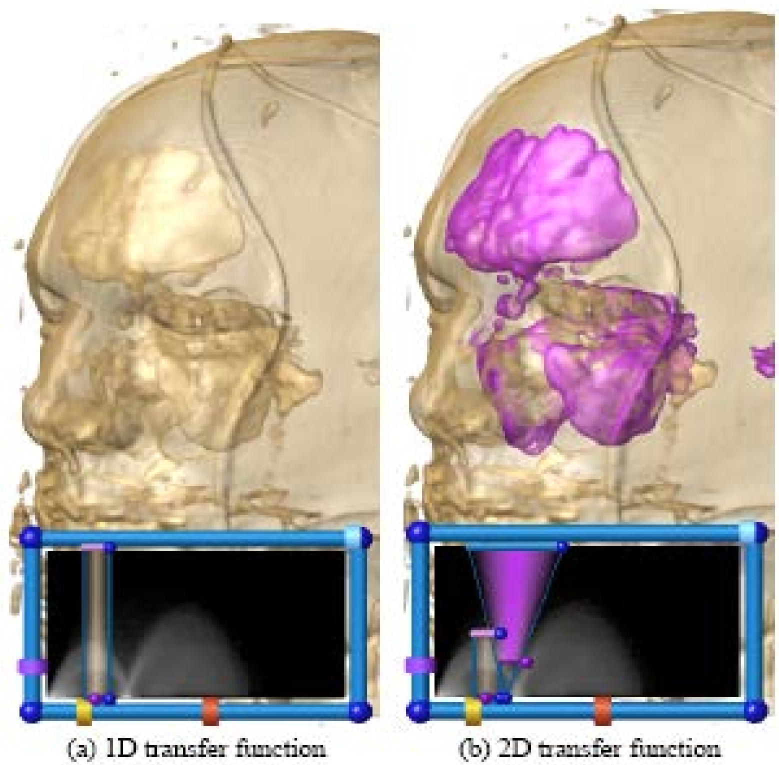


[wetter.de]

Colour Mapping

- Art or experience?
  - The design of a good mapping is as much an art as a science!



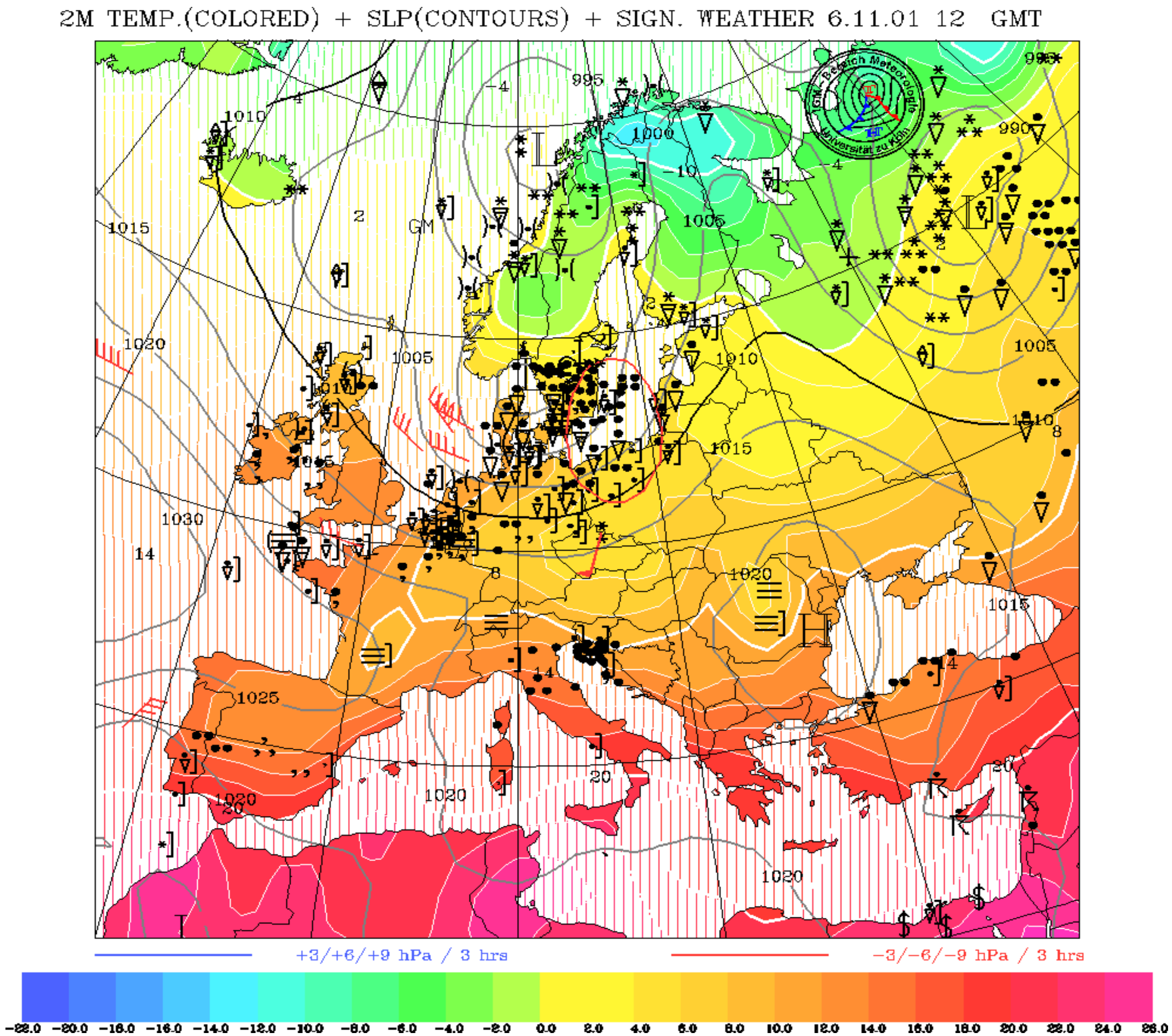


[www.cs.utah.edu/~jmk/research.html]

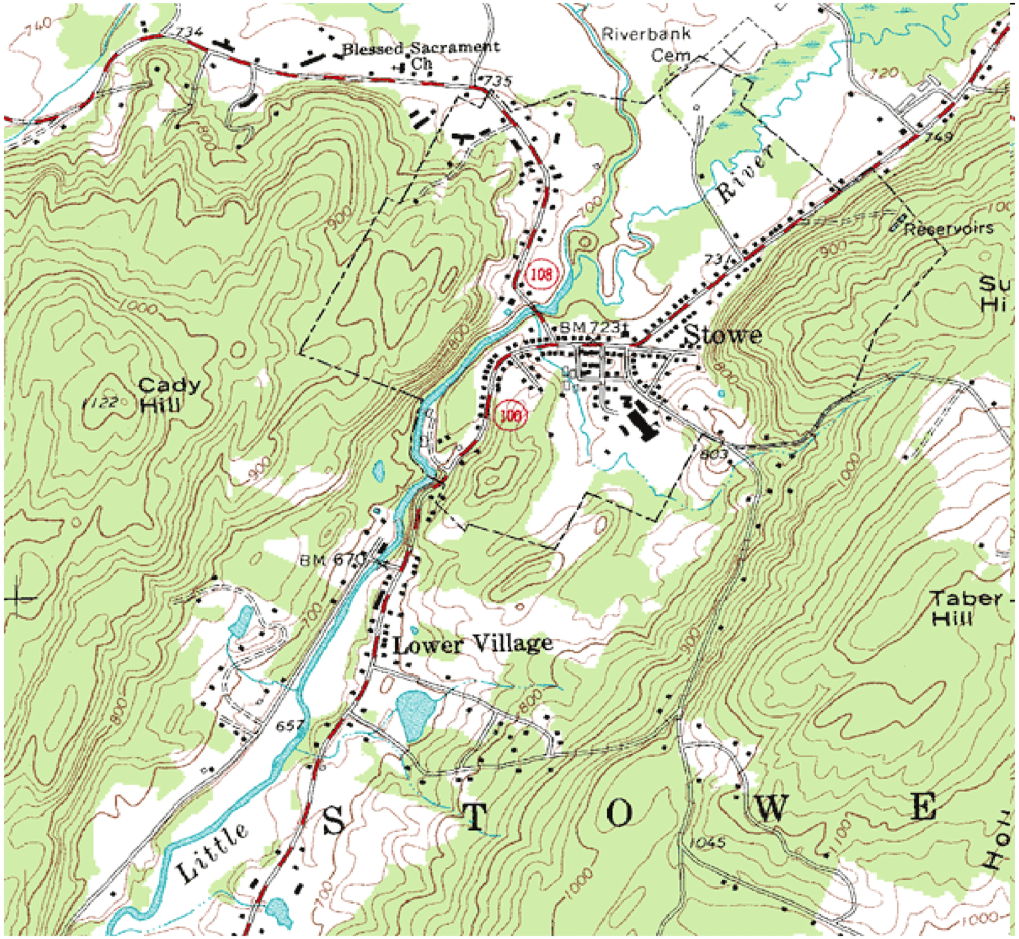
Contouring

- Construct boundaries between regions of different scalar values
- Represent a constant value (iso value):
  - 2D: contour lines / iso-line
  - 3D: iso-surface
- Natural extension to colour mapping.
- Weather maps frequently use contours to show lines of constant value
  - Isotherms for temperature
  - Isobars for pressure, etc.

Example: Weather Map



Example: A Terrain Map



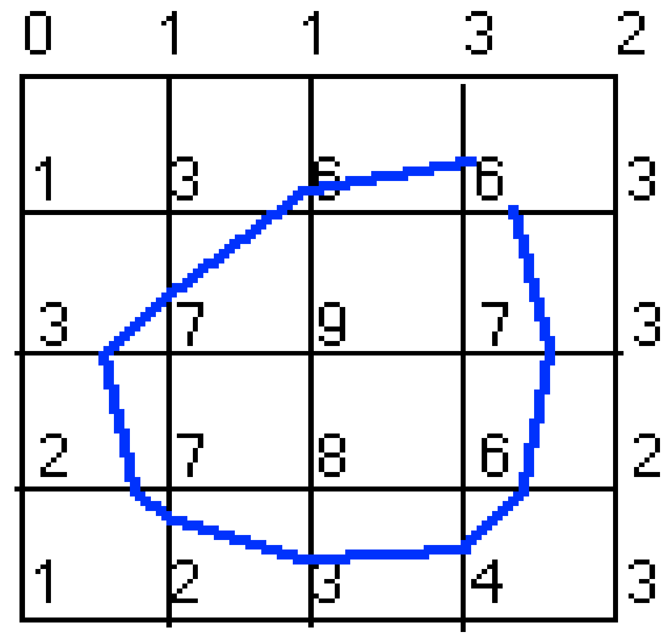
[Sample taken from the public domain USGS Digital Raster Graphic file o44072d6.tif for the Stowe, VT quadrangle]

Contouring - 2D Example

- A regular grid
  - Numbers in the cells represent values of some physical property, *f(x)*, at the sample point (the cell vertex nearest the number)
- We want to show where

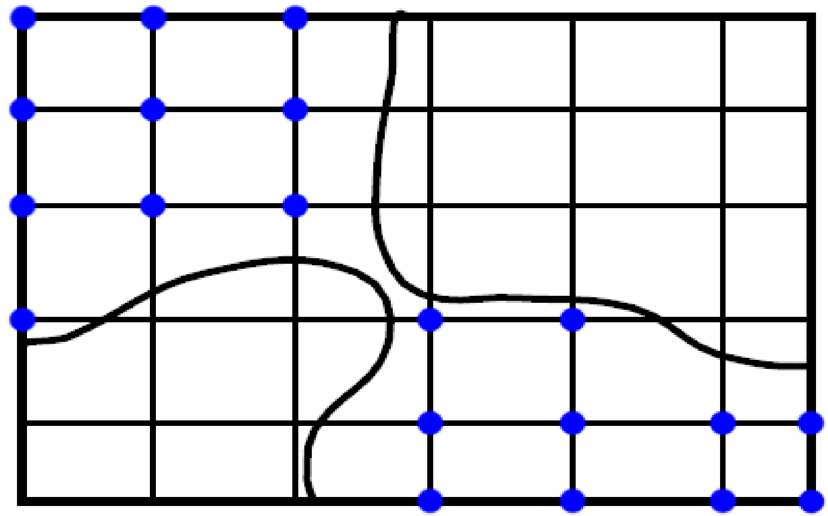
$$f(x) = 5$$

- Determine intersection points of the contour for each cell edge
  - Interpolate between two vertices
    - Use linear interpolation: simple, fast, often sufficient
- Connect pairs of intersection points of each cell if any
  - Assumes that there exists an even number of intersections with edges of a cell (2 or 4 intersections – or none)



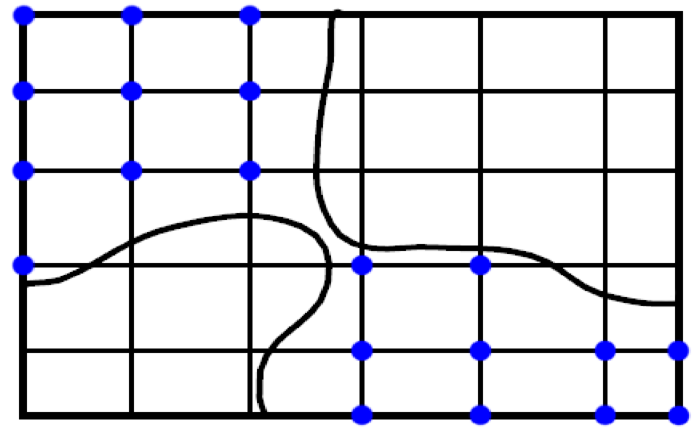
Contour Tracking

- Goal: find all contour lines for a given iso value in a discrete grid of data values
- Idea
  - Contour entering a cell must exit it – just follow it into the next cell
  - Contour entering from border exits through border



● Point *above* iso value

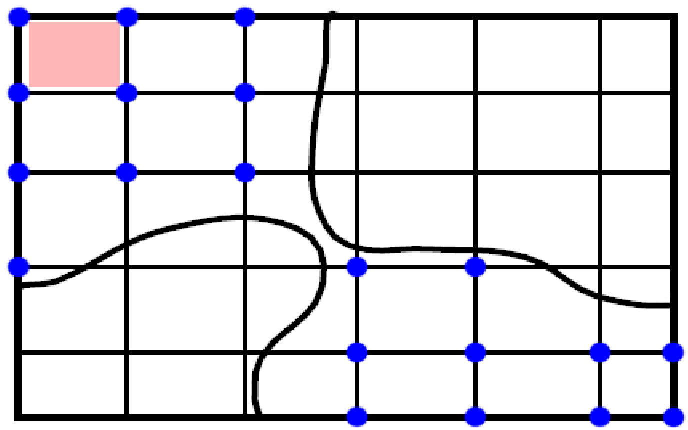
Contour Tracking Strategy



- Find contour beginning
  - begin trace with border intersections
  - then trace inner contour lines
- Track this contour as it crosses cell boundaries until it closes back on itself (inner trace) or …
- … it exits the dataset boundary (border trace)
- If there is known to be no contour left, stop otherwise check every edge for a contour beginning

Contouring - Marching Squares

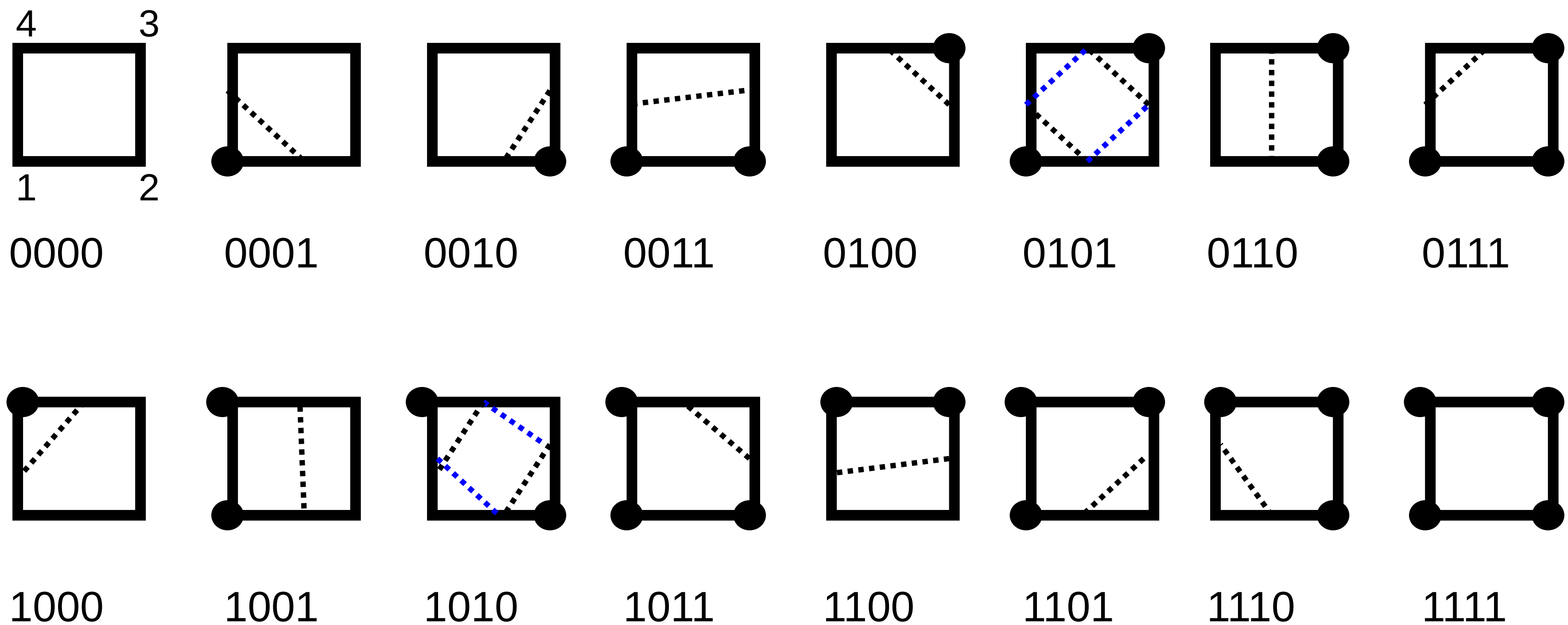
- Contour tracking evaluates every edge
- Marching Squares algorithm evaluates every grid cell



- Basic observation
  - There is only a finite number of ways a contour can pass through a cell
  - There is a limited number of topological states of a cell.
  - Enumerate all of these states
    - Place them in a table

Marching Squares

The possibilities:

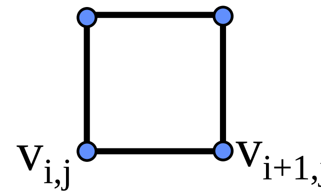


● Point *above* contour
(index bit = 1)

Marching Squares

- Topological state (i.e. how the contour passes through a cell) depends on
  - The number of vertices (4)
  - The relationships between the vertex values and the contour of interest.
- Count the possibilities
  - There are 4 vertices
  - Value at each vertex value can be
    - > (inside) or
    - =< (outside) the contour,
  - So there are $2^4 = 16$ possible ways a contour can pass through a cell
- Result: we know how a contour passes through the cell (the topology), but we don't know where the contour passes through the cell (the geometry)

Marching Squares



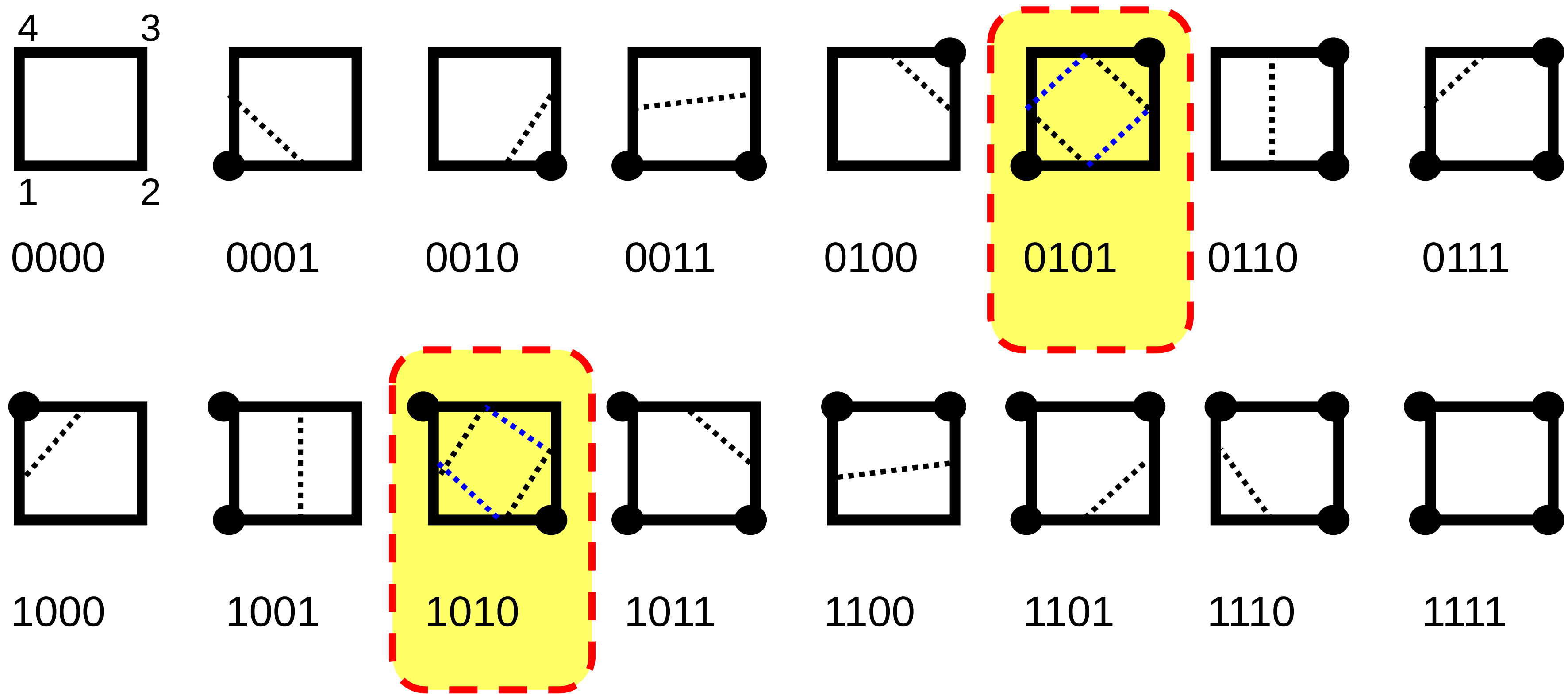
- Algorithm
  - Select a cell
  - Calculate inside/outside state of each vertex
  - Generate topological state index from 'inside' bits
  - Look up topological state using index
    - Results is a set of edges intersected by contour
  - Calculate contour intersection using interpolation for edges in the intersection set

$$r = (F - f(v_{i,j})) / (f(v_{i+1,j}) - f(v_{i,j})) \qquad \text{F: iso value}$$

- Issues
  - At the cell boundaries: duplicate edges & vertices may be created
    - Eliminate with coincident point-merging operation
  - Interpolate in the *same* direction for each edge of adjacent cells ⇨ otherwise round-off errors result in non-coincident points

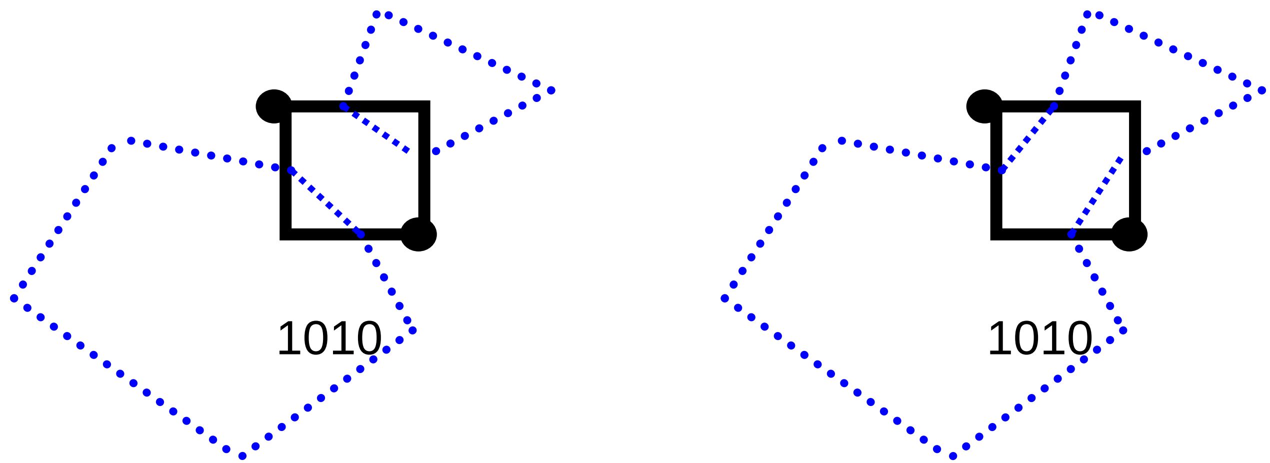
Marching Squares - Disambiguation

- 2D: two ambiguous topologies



Marching Squares - Ambiguities

- Two possible contours



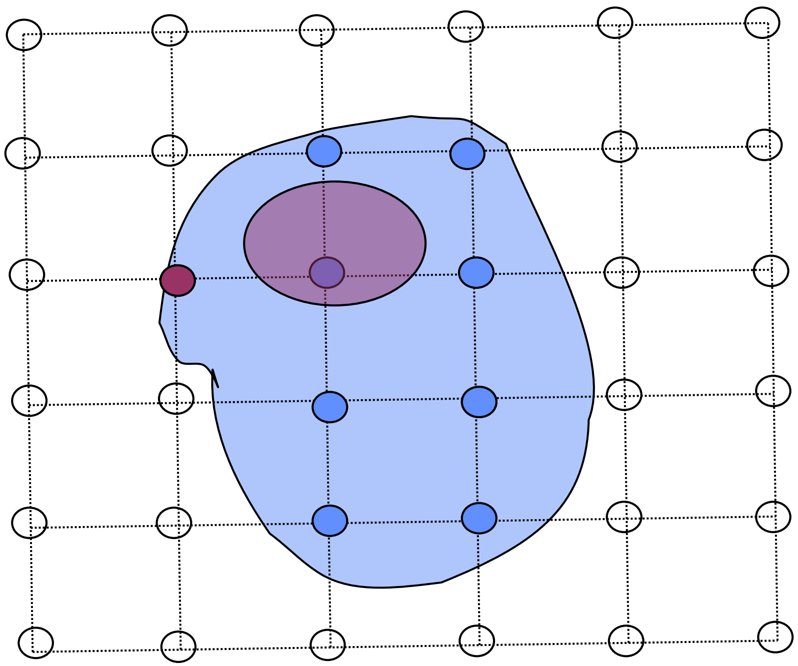
Break contour | Join contour

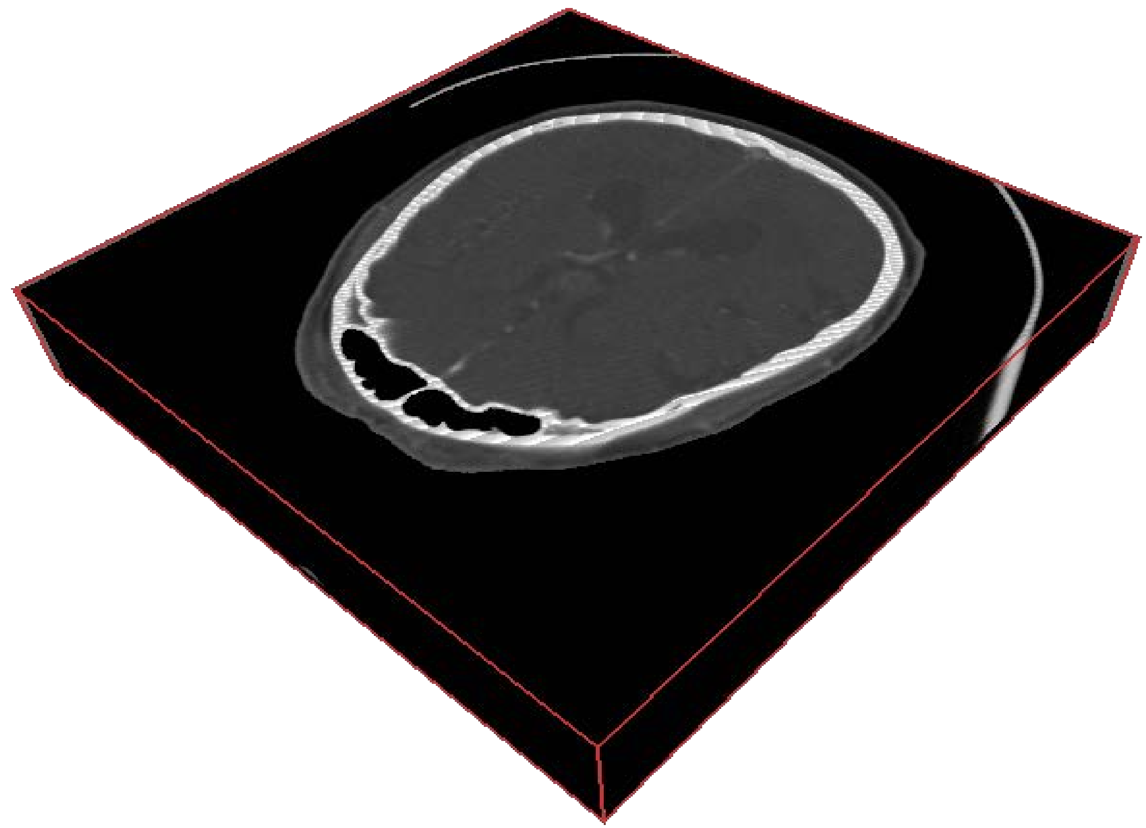
  - In 2D, choose either one
  - Either acceptable
  - We don't have enough information to make a decision
  - Or use asymptotic decider ...

Contour Tracking vs Marching Squares

- Marching Squares
  - Easy to implement
    - Important in equivalent 3D algorithm
  - Creates disconnected line segments and points
    - Merging operation to generate actual contour line needed
- Contour Tracking
  - Can generate a single polyline per contour line
    - Avoids the need to merge coincident points
  - Each edge must be checked if multiple contours exist

Volume Data on Grids



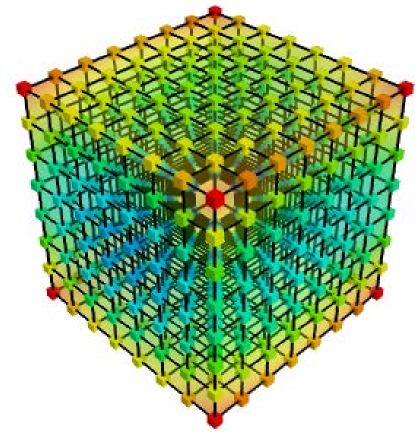


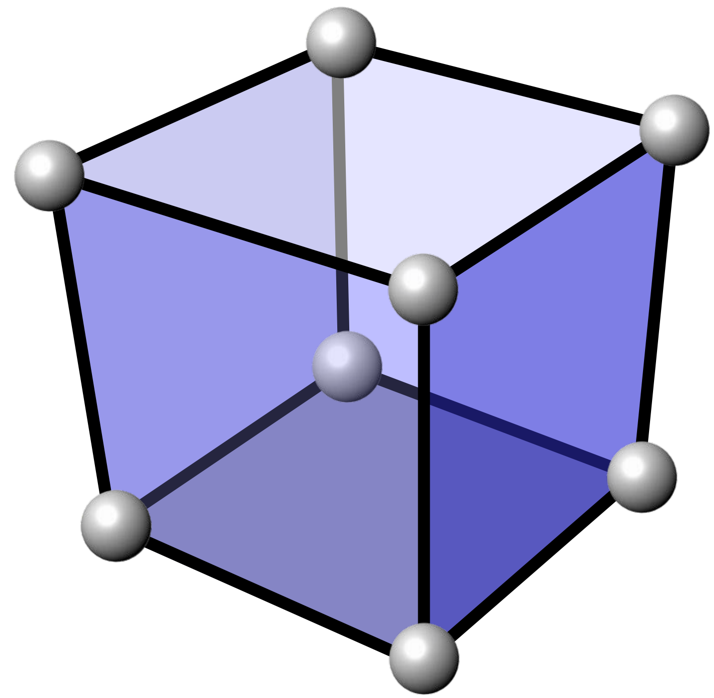
Given: A volume as a 3D array of scalar data values e.g. density

Goal: Compute image of the structures inside the volume

3D Contours - Iso-Surfaces

- Marching Cubes
  - Marching Squares extended into 3D
  - Goal: generate iso-surface (contour surface)
  - Assumption:
    - Fixed number of ways for a contour to pass through a cubic cell
  - Voxel values at vertices
- Contour stitching



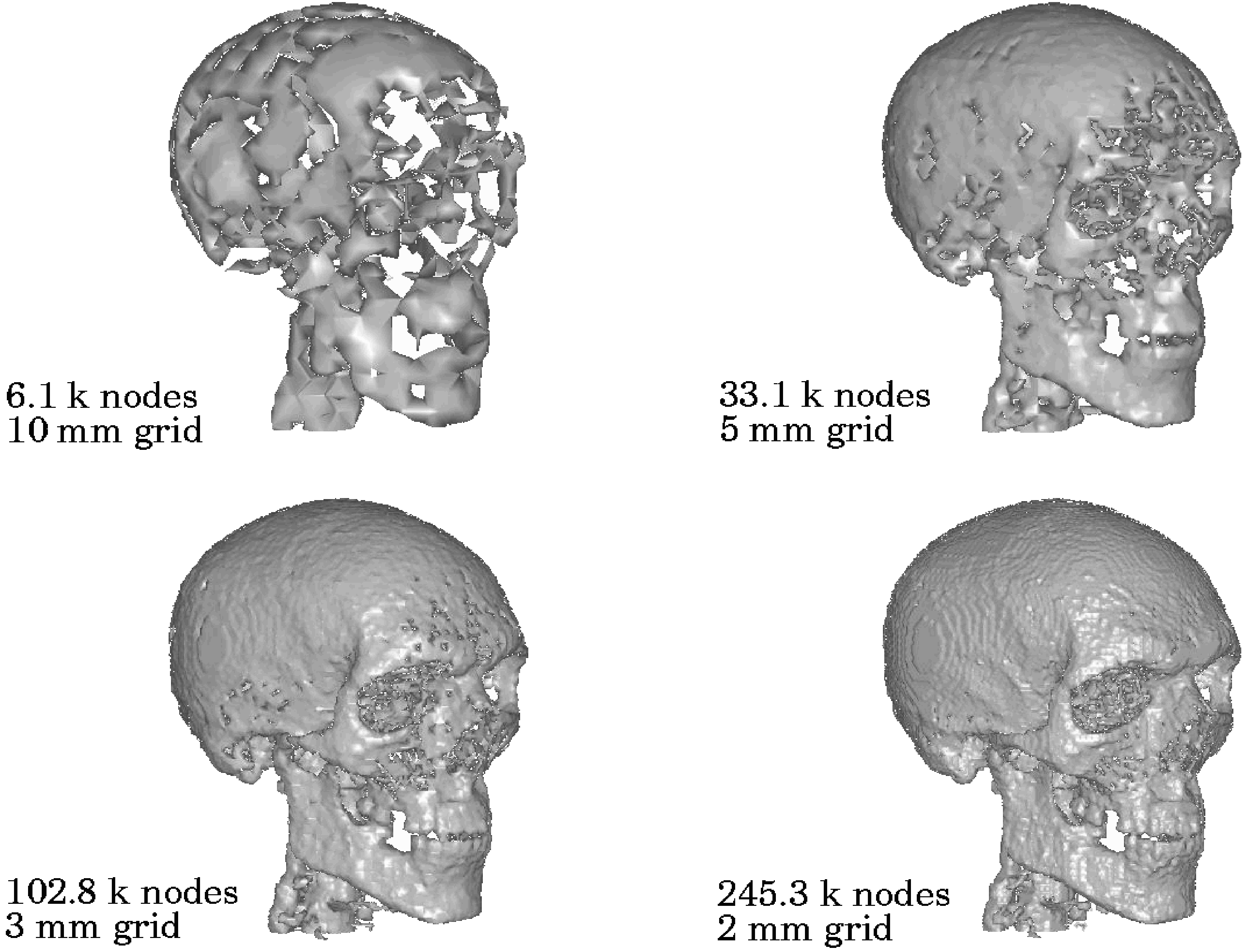


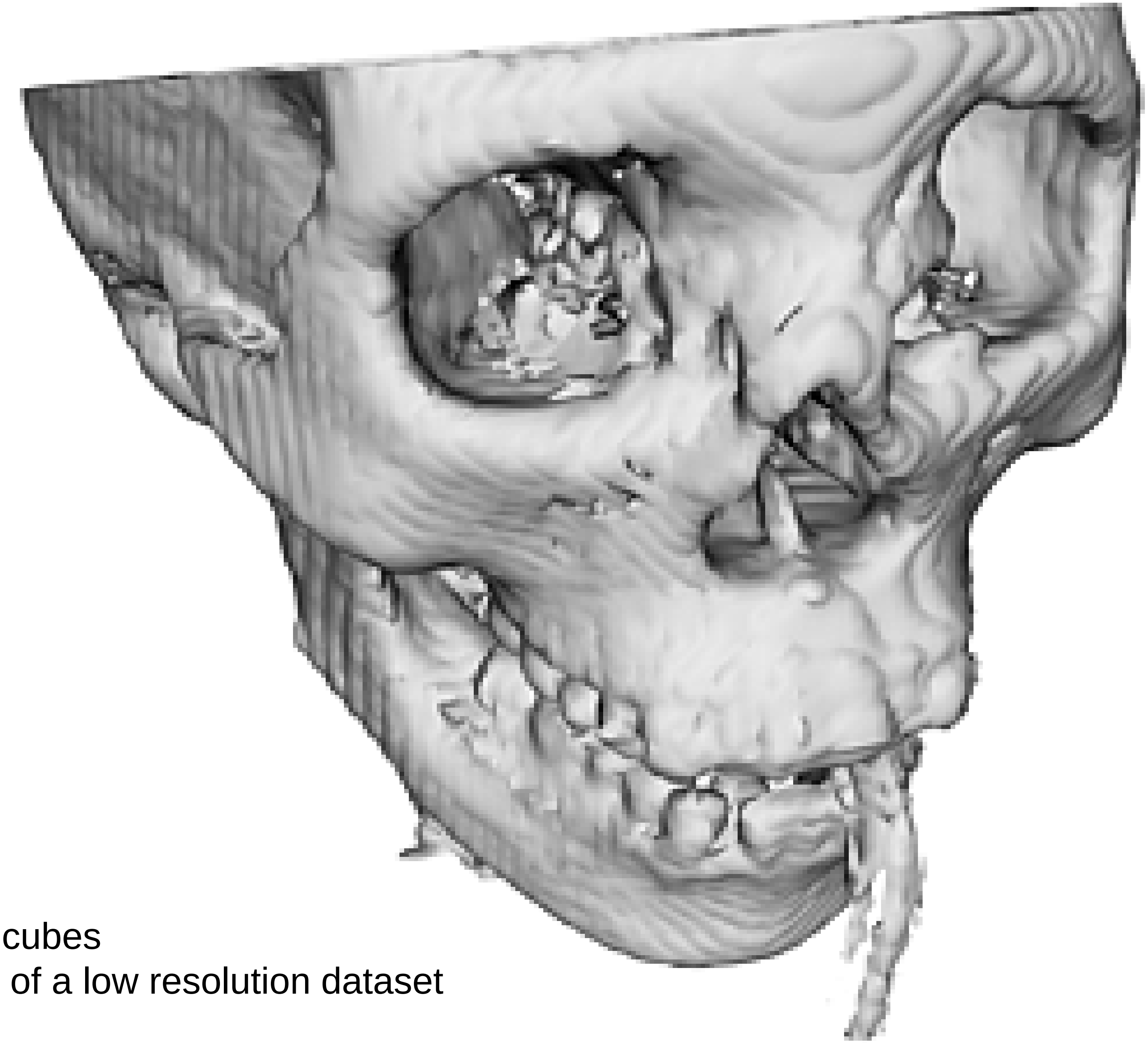
Example Dataset: CT of a Lobster

Resolution 120x120x34, 8Bit

z=3 z=10 z=17 z=23 z=30

rendering of isosurface at
iso value 40
© http://liris.cnrs.fr/dgtal/gallery/mc-lobster-40/

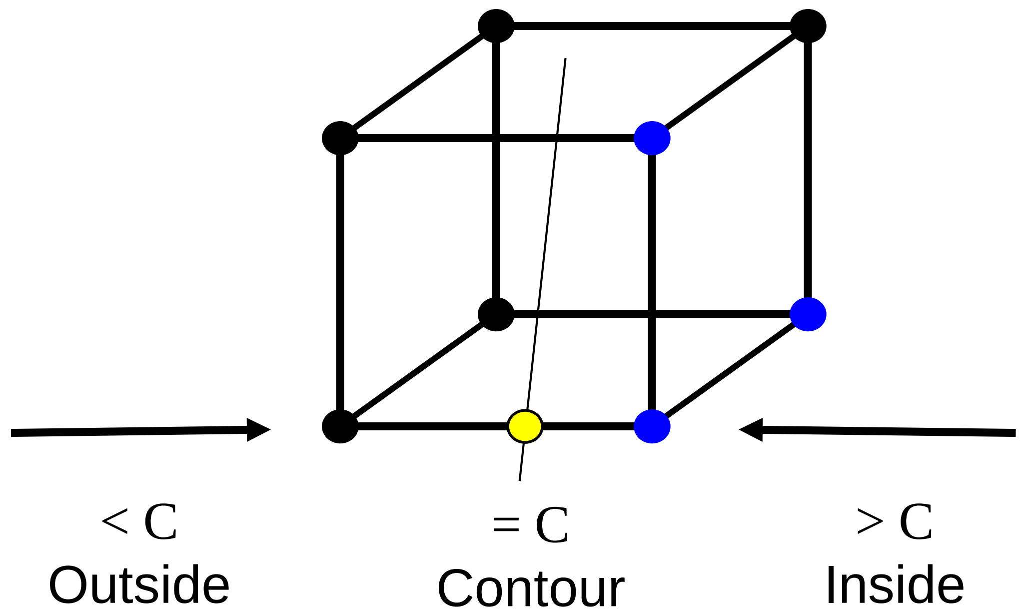

6.1 k nodes
10 mm grid
33.1 k nodes
5 mm grid
102.8 k nodes
3 mm grid
245.3 k nodes
2 mm grid



Typical image
of a marching cubes
reconstruction of a low resolution dataset

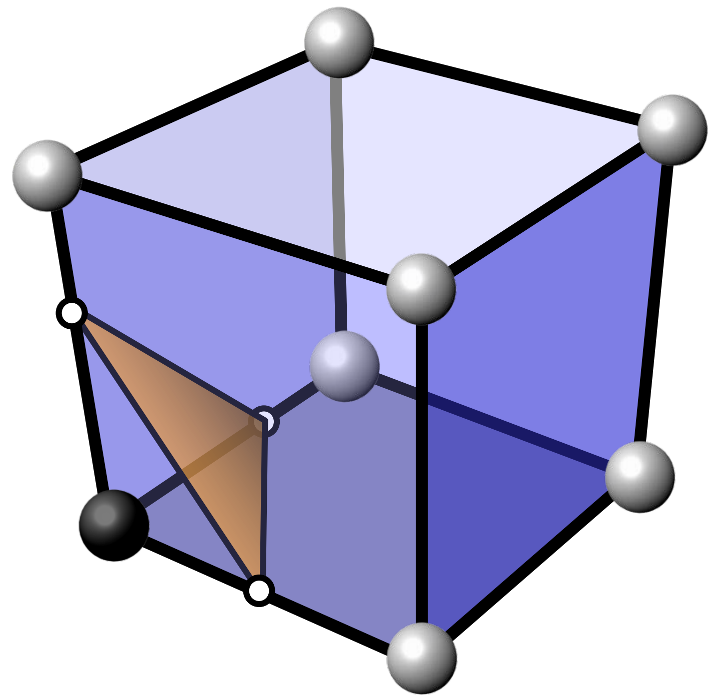
Marching Cubes

- Contour value C classifies each vertex as inside or outside
- Contour passes between an inside and outside vertex



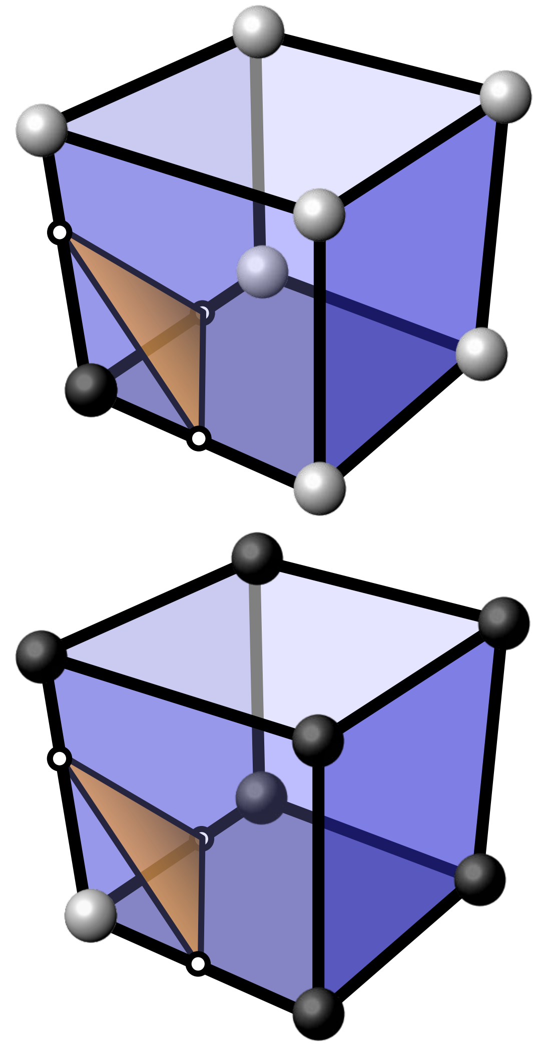
Marching Cubes

- Contour value classifies each vertex as inside or outside
- 8 vertices, so $2^8 = 256$ different topologies
- Cube has a high degree of symmetry
- Can reduce the number of distinct cases using symmetry operations on a basic set of cases
- What are the symmetry operations for a cube?



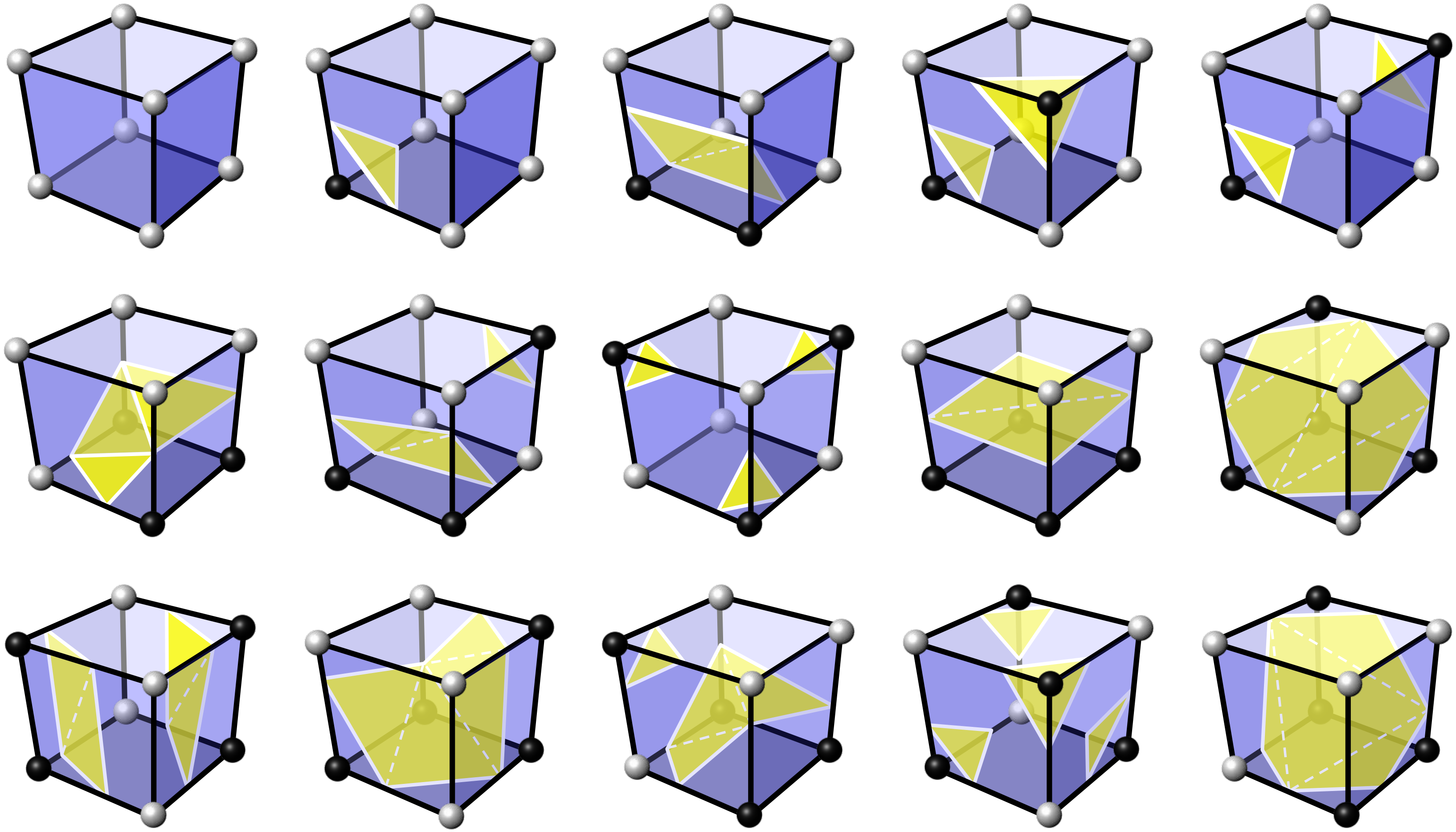
Marching Cubes – Distinct Topologies

- Rotation symmetry
- Complementary symmetry
  - Inside / outside choice is arbitrary
  - Half the number of cases can be generated by complementing inside / outside choice
  - Location of surface doesn't change, only the direction of the surface normal does
- So 256 cases reduce to ...



Marching Cubes - Distinct Topologies

- Only 15!



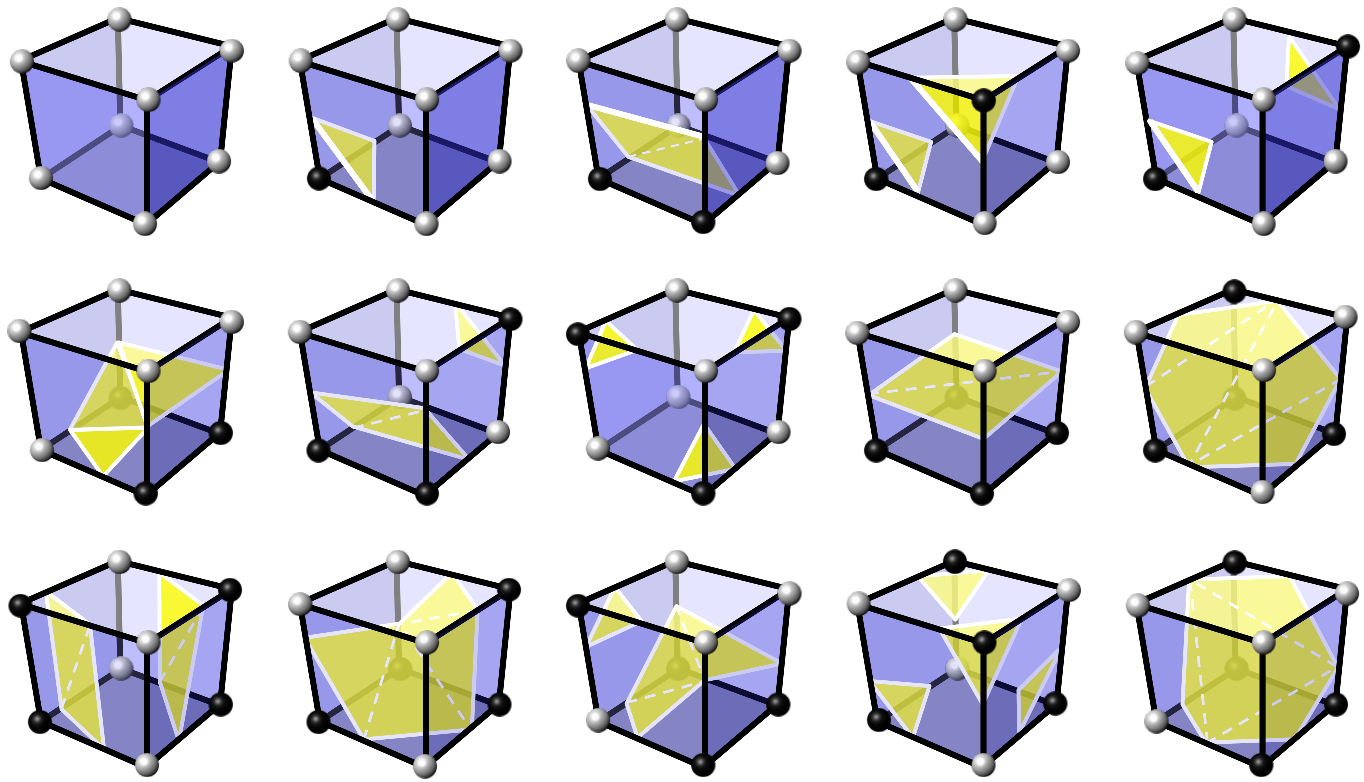
Marching Cubes - Cell Topology

- Algorithm looks like this:

```
/* C is contour value   */
/* p[k], k=0,7 contains vertex values */
index = 0;
for ( k=0; k<8; ++k ) {
   if ( p[k] > C )  index = index | (1<<k);
}
/* topology is a reference to a suitable
   structure which contains
   intersecting edge list */
topology = topology_table[index];
```

Marching Cubes – Surface Normals

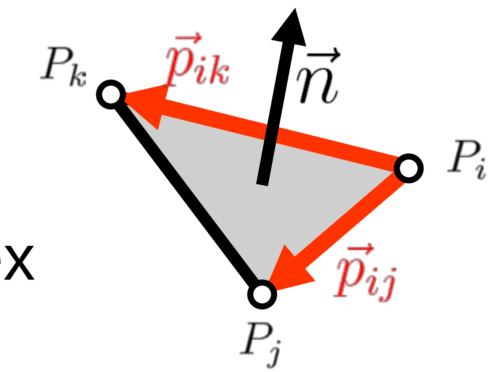
- Surfaces represented as sets of triangles



- Surface normals need to be determined for shading
- Requires consistent vertex ordering for correct normals

Marching Cubes – Surface Normals

- Use geometry of triangles to approximate surface normals
  - Flat shading – just use triangles' normals
  - Gouraud / Phong shading – interpolate vertex normals from adjacent triangles

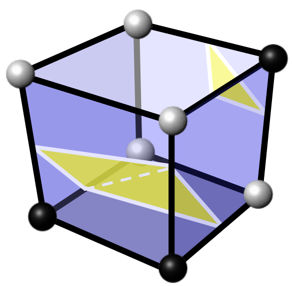


- Determine vertex normals at each triangle vertex
  - Use normals at voxel corners
    - Derived from gradient of the voxel values
  - Trilinear interpolation to find triangle vertex normals
  - More complex approach, but potentially more accurate

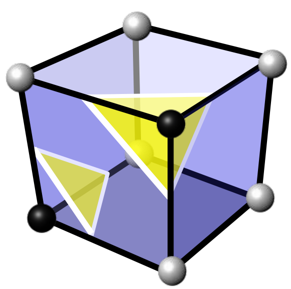
Marching Cubes Ambiguities

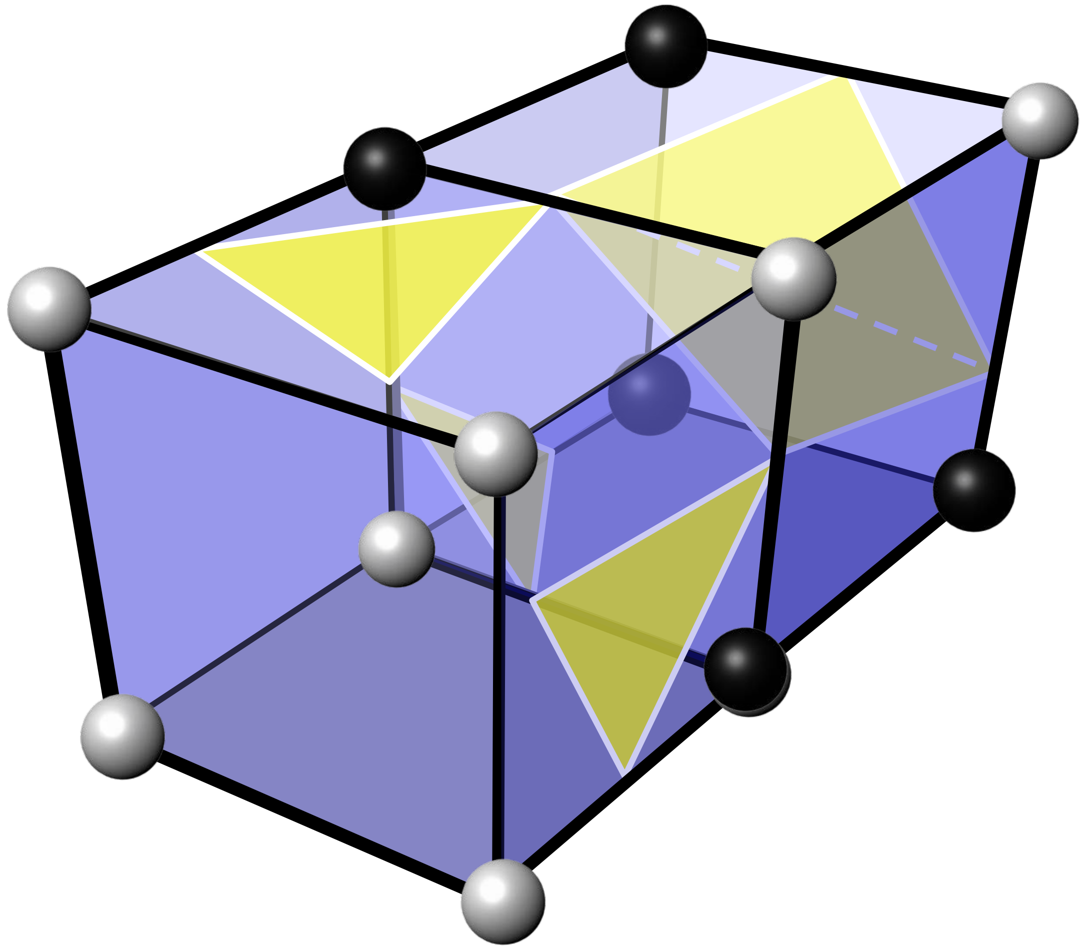
- Algorithm generates "holes“ – similar to the ambiguous 2D cases – diagonal points above/below iso value

Case 6:



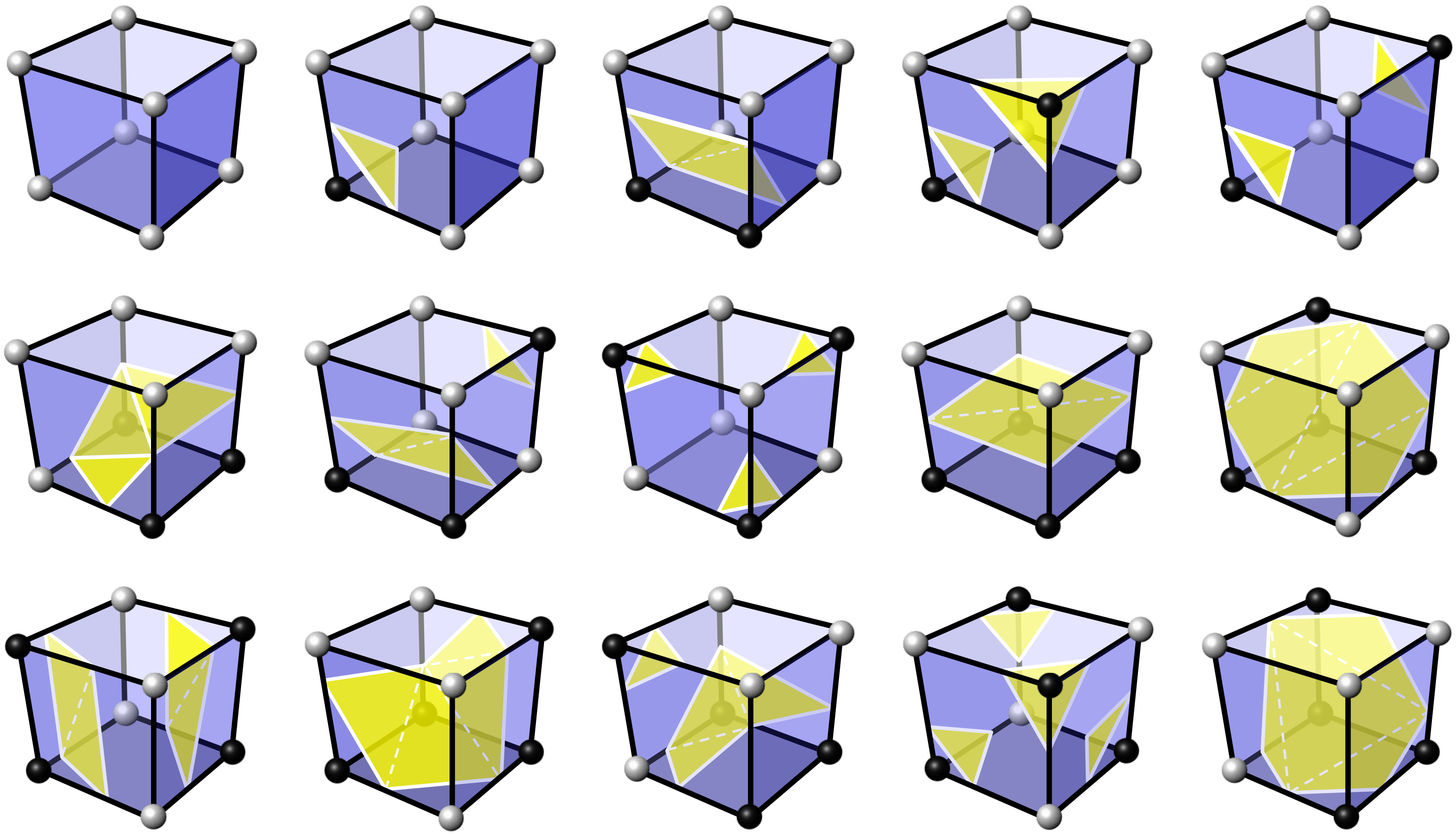
Case 3:





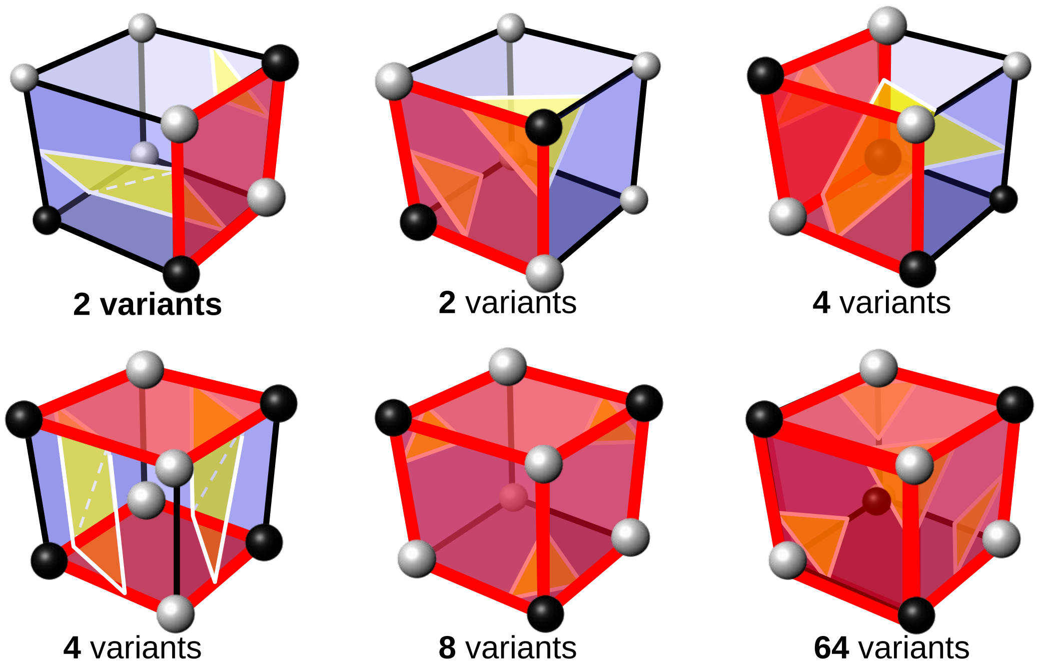
Marching Cubes – 'Hole-generating' Topologies

- 6 topologies can generate holes



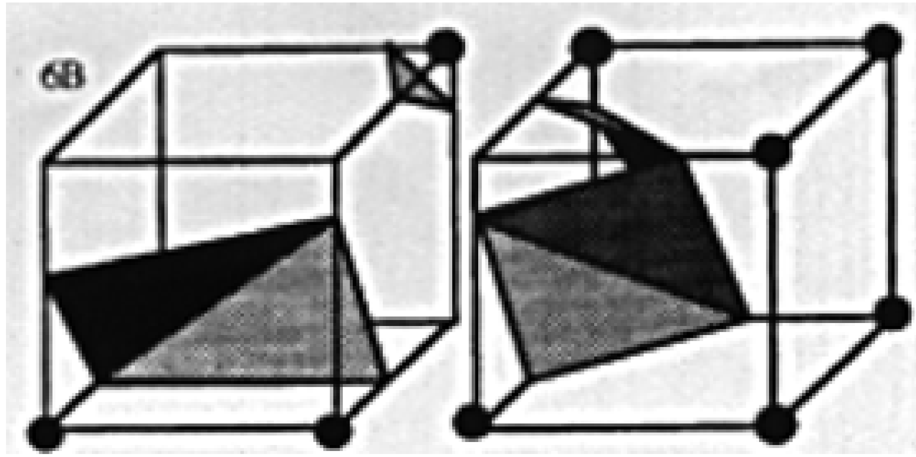
Marching Cubes – 'Hole-generating' Topologies

- 6 topologies can generate holes
  - At least on one face inside and outside marked vertices are arranged diagonally (all edges are cut)
  - Number of ambiguous faces i determines number of possible triangulations → 2^i



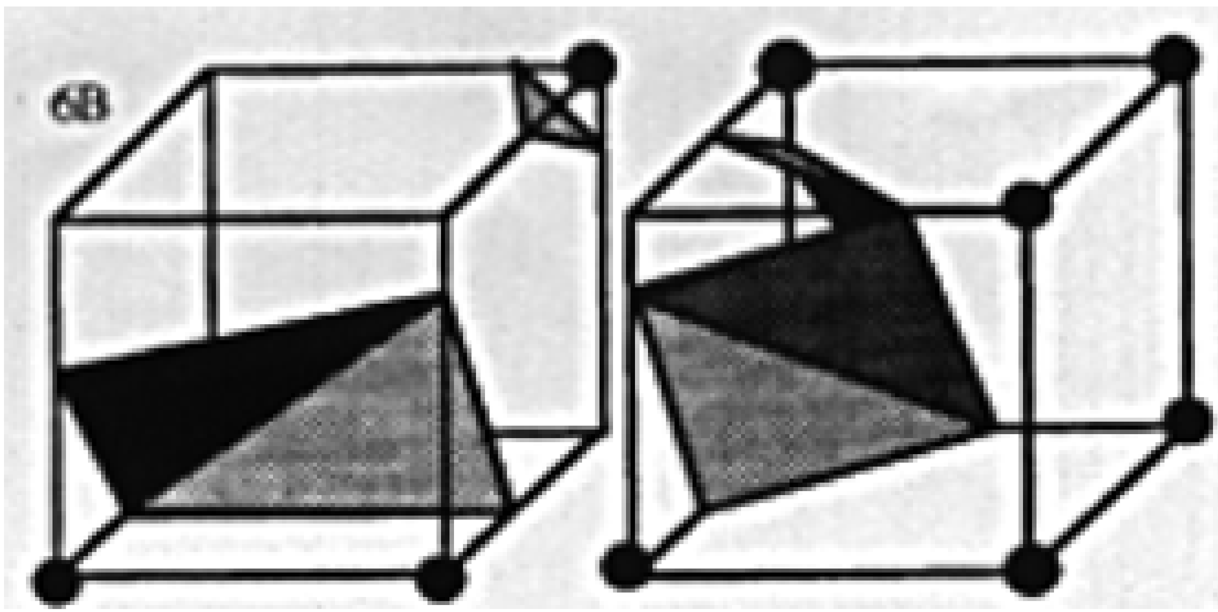
Resolving the Ambiguity

- Be consistent!
- Then resulting surface should contain no ambiguities
- Store the edge intercepts you chose
- In the next march step,
  - do not recalculate intercepts for the common face - they have been done.
- Use the triangles that would be consistent with the edge intercepts in the previous cube



Resolving the Ambiguity

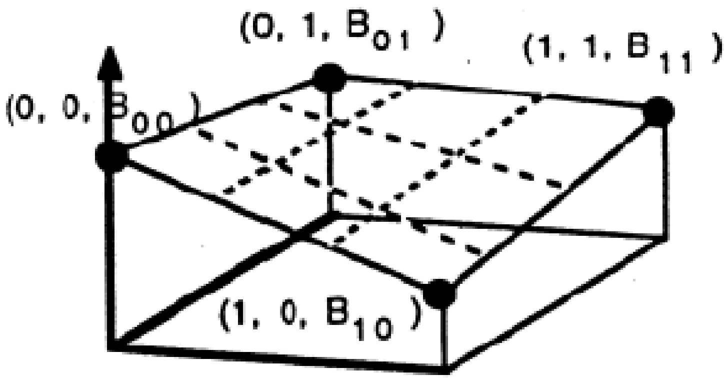
- The surface will be complete though not necessarily correct in a mathematical sense – according to the underlying tri-linear interpolation.
- The degree of incorrectness is likely to be small.
- These ambiguous cell faces are not common in medical visualisation applications.



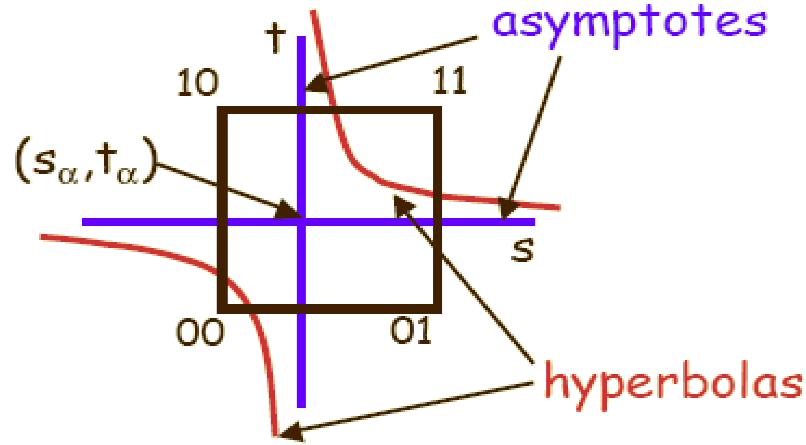
Resolving the Ambiguity

- Frequency of occurrence of the ambiguous cases
  - They are infrequent
  - Occur 2 orders of magnitude less often
    than the nine cases that do not have ambiguities.
  - Even when the ambiguous topologies do arise,
    results indicate that the triangulations originally defined by Lorensen et al. are the most frequent
- Nielson and Hamann: The asymptotic decider: Resolving the ambiguities in marching cubes
  (Proceedings of Visualization '91, pp 83-91)
- WE Lorensen, & HE Cline, Marching cubes: a high resolution 3d surface construction algorithm, ACM Computer Graphics, 21(4), 163-169, 1987.

The Asymptotic Decider



(B_{ij} are data values at grid points)

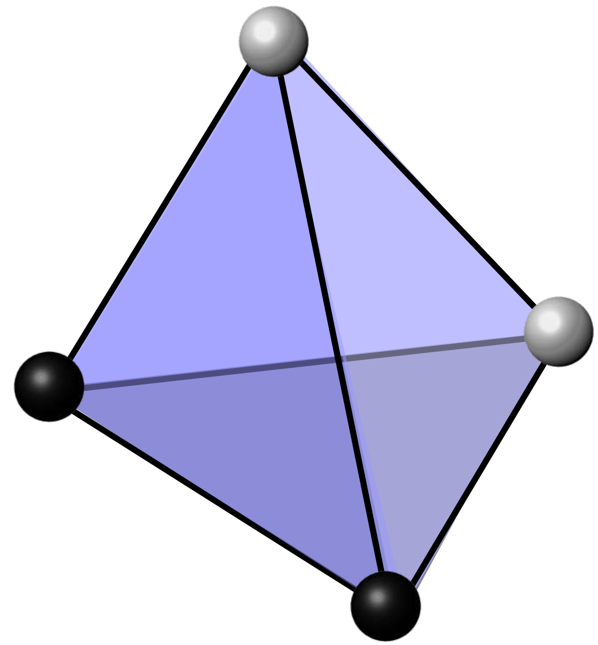


- Assume bilinear interpolation B(s,t) within a face
- C is iso-value
- Iso-lines are hyperbolas B(s,t) = C (quadratic in s and t)
- Asymptotes meet at point (s_α,t_α), which is the saddle point
- Compare B(s_α,t_α) to iso-value C
- Connect intersection points appropriately
- example above: If vertex 00 and 11 are above iso-value – and B(s_α,t_α) is below iso-value – then this would be the solution

Marching Tetrahedra

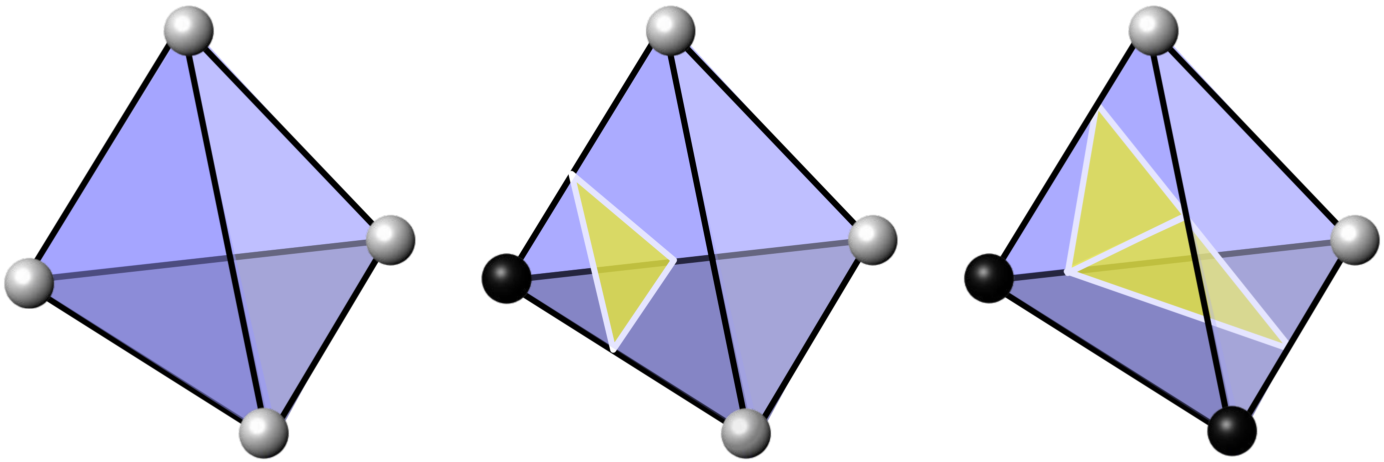
- Same idea as marching cubes using tetrahedra
  - Every cell configuration has a unique index
  - For every index there is a topology
  - 4 vertices → $2^4 = 16$ configurations

- Break up cubic cells into five tetrahedra



Marching Tetrahedra

- Three distinct topologies
  - Symmetry, rotation



- No ambiguous faces
- No ambiguous cells

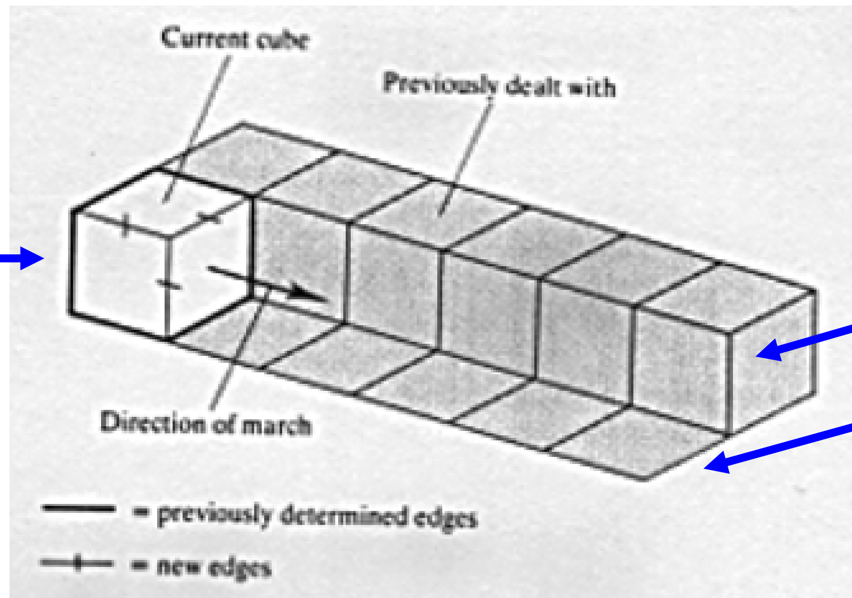
Marching Cubes - Efficiency

- Large amounts of data
  - Design must take into account the limitations of the computing hardware
  - Typically memory limitations - still a problem nowadays?
- Efficiency issues
  - Contour intersection with an edge
    - Relevant to four cells
      (ignoring cells on the boundary of the volume data)
    - Implementation should make use of coherence -
    - *No contour/edge intersection should be calculated twice!*
  - Large amount of empty cells
    - 30-70% of the computations on empty cells depending on the data
  - Minimize memory use

Marching Cubes - Algorithm

- Memory use is minimized
  - The marching cubes algorithm is based on relationships between nearest neighbours.
  - Hence at most only two slices need to be retained in memory at any one time.

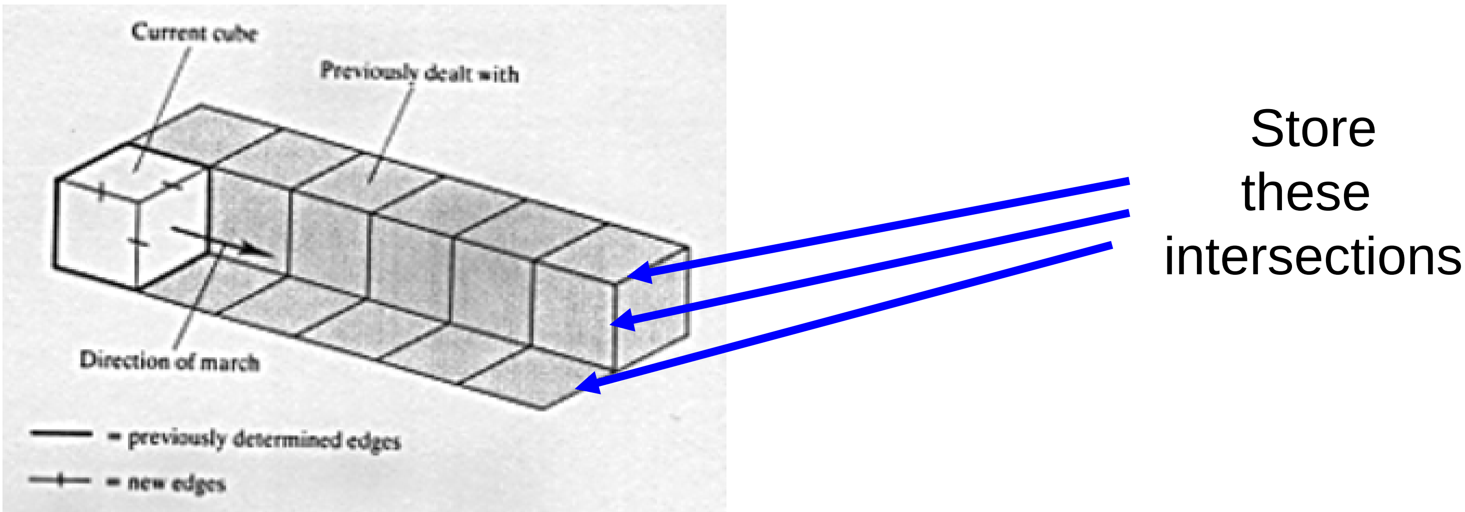
Current Cube



Need these two rows

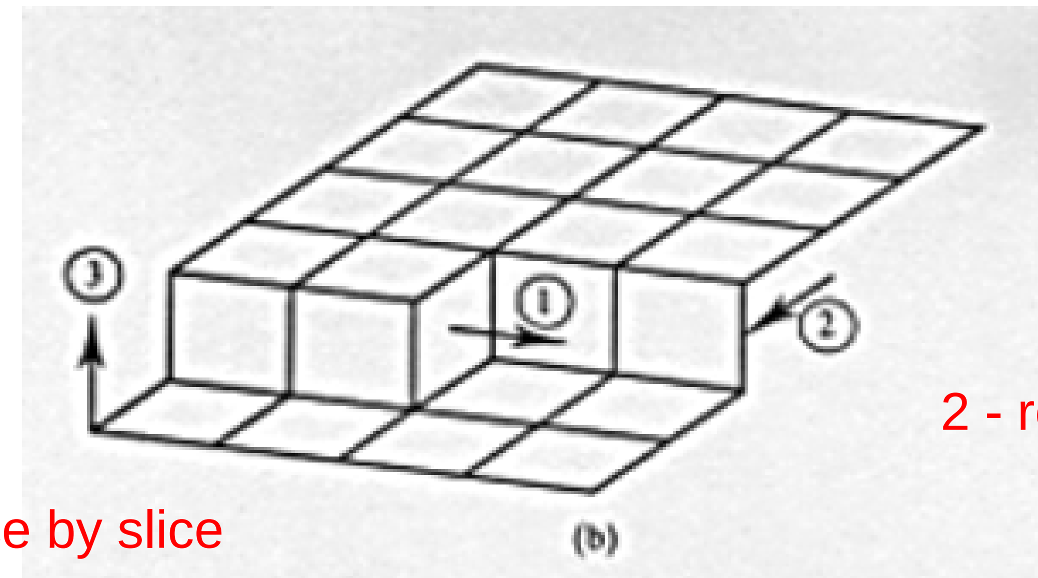
Marching Cubes - Algorithm

- Coherency is exploited by storing relevant edge intersections for later reuse.



Marching Cubes - Algorithm

- March ordering – cache coherent memory access
  - Assumption: index k is contiguous in memory
  - Slice by slice (incrementing i)
    - Row by row (incrementing j)
      - March voxel by voxel (incrementing k) from element (i,j,k) is to (i,j,k+1)



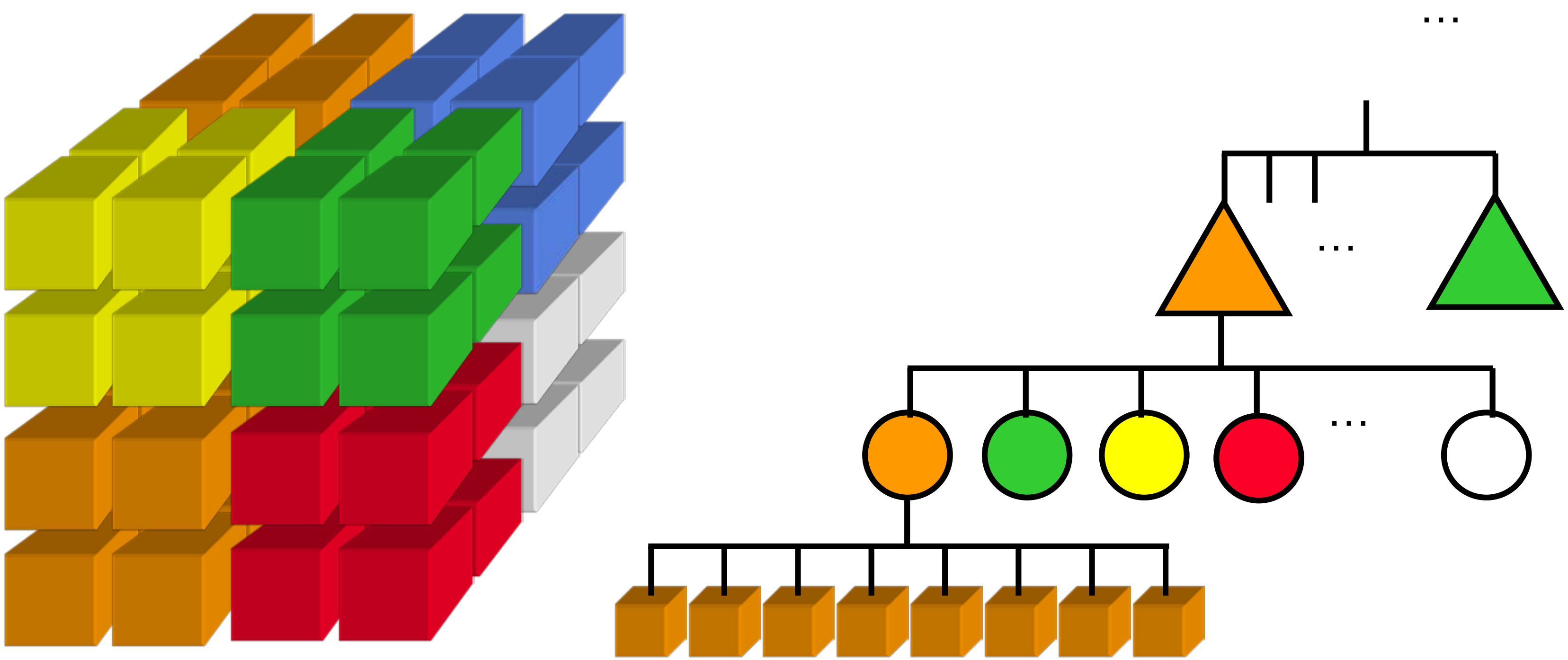
1 - along each row

2 - row by row

3 - slice by slice

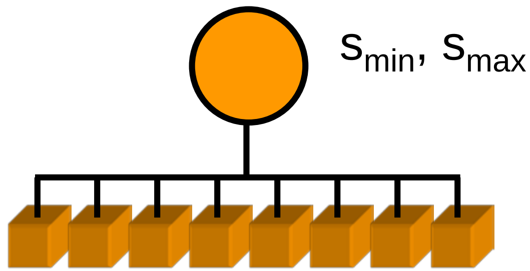
Marching Cubes – Acceleration

- Build min-max octree



Marching Cubes – Acceleration

- Use of min-max octree-hierarchy
  - Each node holds minimal and maximal data value of its children



  - Traverse tree top-down with iso value **C**
  - Discard nodes with $\mathbf{C < s_{min}}$ or $\mathbf{C > s_{max}}$

Marching Cubes – Acceleration

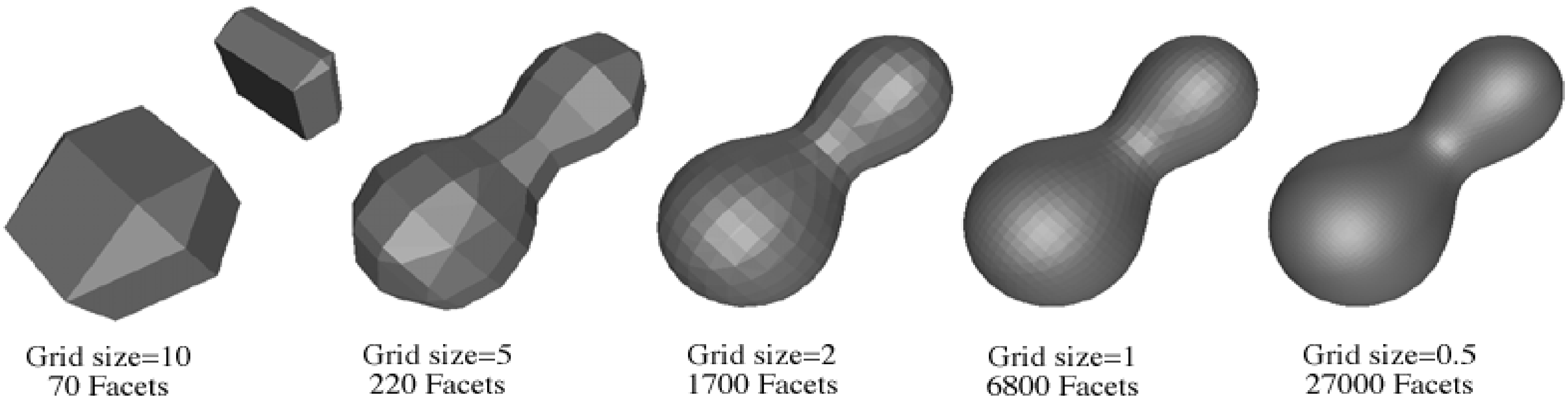
- Problem
  - No sequential processing of the cells
  - No reusing of calculated intersections
- Solution
  - Save all intersections
  - Querying of edge intersections
  - → Hash table holding intersections with edges as key element
    - → Edges originate at point (x, y, z) with indices (i,j,k) which defines an `index` (address of point in linear layout of 3D array)
      `index = i*y_size*z_size + j*z_size + k`
    - → Edges point along a positive axis direction, which defines a direction code (1 for x, 2 for y, 3 for z)

```
edge_hash_key = 4 * index + direction_code
```

    - → Fast lookups for saved edge intersections

Effect of Grid Size

- Grid cell size *vs* facets generated
  - N = number of voxels in each direction
  - Number of isosurface facets increases in general with order $O(N^2)$
  - Also interesting for size of hash table $O(N^2)$ edges!

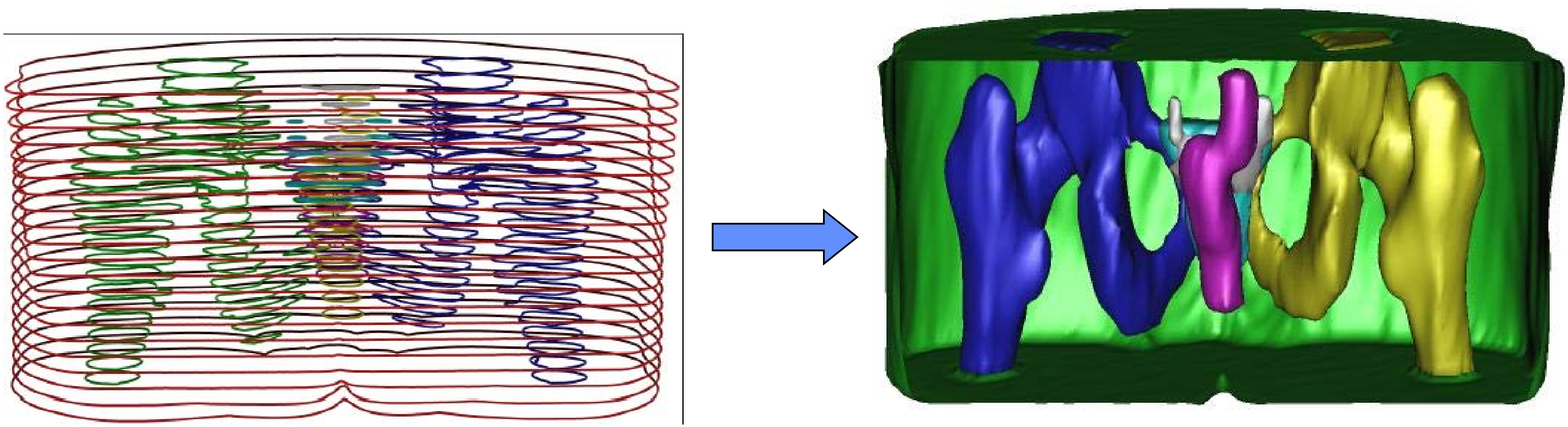


Literature

- William E. Lorensen and Harvey E. Cline. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics (Proceedings of SIGGRAPH 87), 21 (4), pp. 163-169 (July 1987, Anaheim, California).
- Gregory M. Nielson and Bernd Hamann. *The Asymptotic Decider: Removing the Ambiguity in Marching Cubes*, IEEE Visualization '91, pp. 83-91 (1991).
- T. Itoh and K. Koyamada. *Automatic isosurface propagation using an extrema graph and sorted boundary cell lists*, IEEE Transactions on Visualization and Computer Graphics, 1 (4), pp. 319-327 (December 1995).
- Yarden Livnat and Han-Wei Shen and Christopher R. Johnson. *A Near Optimal Isosurface Extraction Algorithm Using the Span Space,* IEEE Transactions on Visualization and Computer Graphics, 2(1), pp. 73-84 (March 1996, San Francisco, California).
- Han-Wei Shen and Charles D. Hansen and Yarden Livnat and Christopher R. Johnson. *Isosurfacing in Span Space with Utmost Efficiency (ISSUE)*, IEEE Visualization '96, pp. 287-294 (October 1996)
- Paolo Cignoni and Paola Marino and Claudio Montani and Enrico Puppo and Roberto Scopigno. *Speeding Up Isosurface Extraction Using Interval Trees,* IEEE Transactions on Visualization and Computer Graphics, 3(2), (April - June 1997). ISSN 1077-2626.
- Jane Wilhelms and Allen Van Gelder. *Octrees for faster isosurface generation,* ACM Transactions on Graphics, 11(3), pp. 201-227 (July 1992).

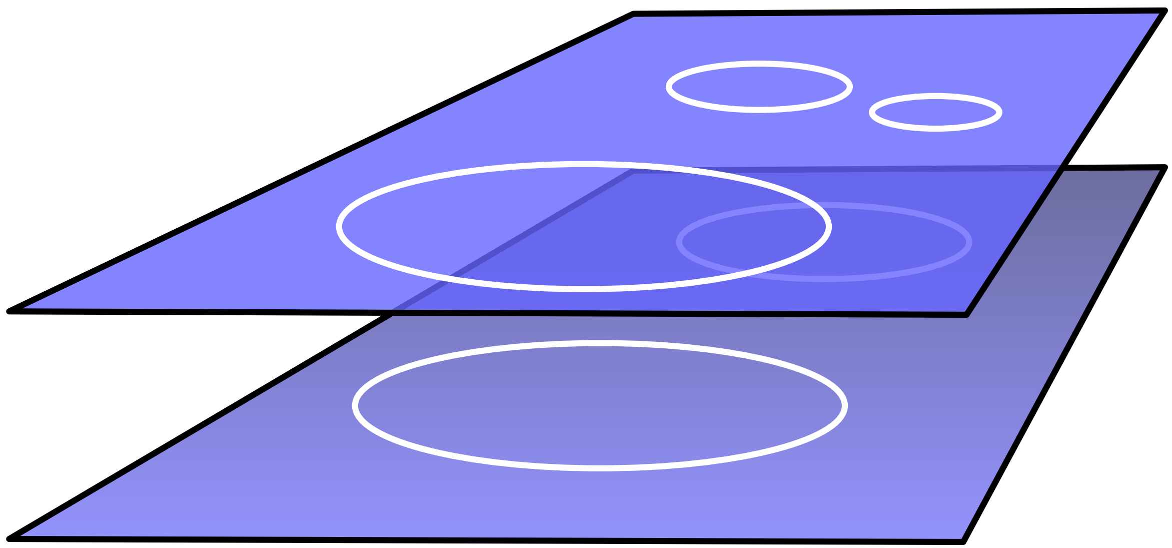
Contour Stitching

- Treat volume as a set of 2D slices
  - Apply 2D contouring algorithm on each slice to generate contours
  - Use a set of hand-drawn contours
- Stitch the slices together



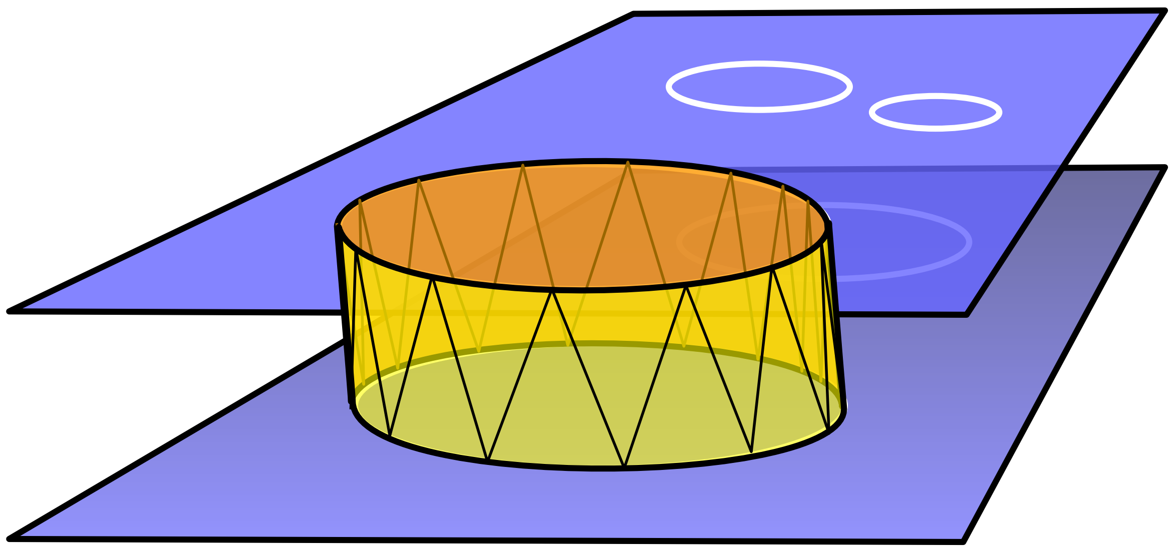
Contour Stitching

- Problems 1: Correspondence
  - Which contours correlate on different slices



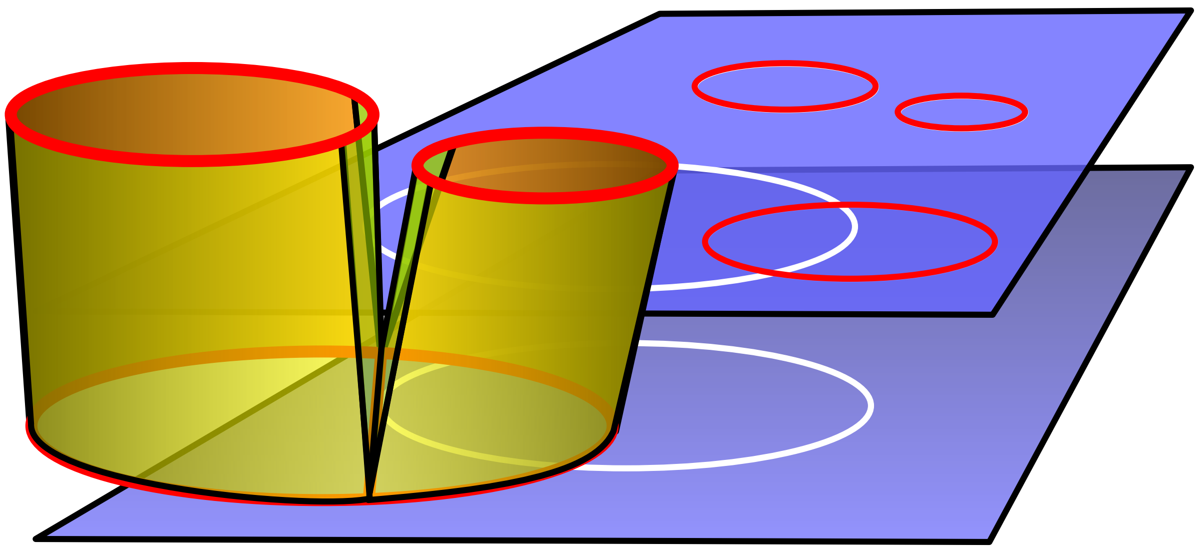
Contour Stitching

- Problems 2: Tiling
  - How to construct the surface



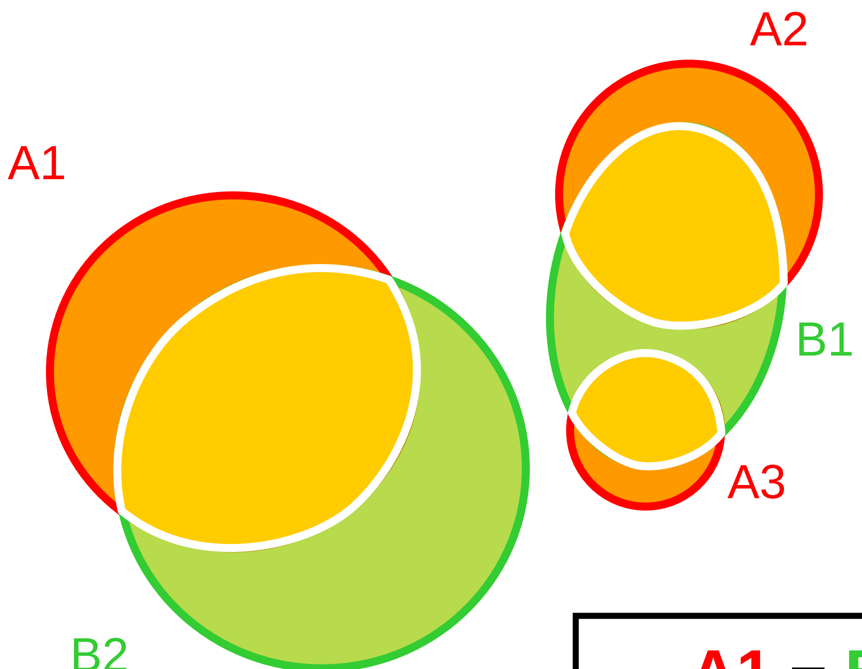
Contour Stitching

- Problems 3: Branching
  - How to construct surface in case of a branch



Contour Correspondence

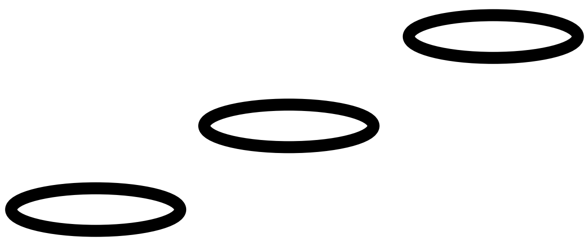
- Automatic solution is very complex
  - Trivial method: overlay contours and find overlaps



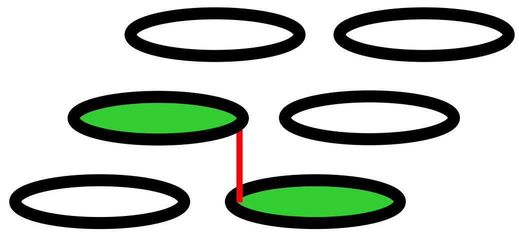
A1 – B2
A2, A3 – B1

Contour Correspondence

- Real world problems much more complex and dependent on data



no overlap

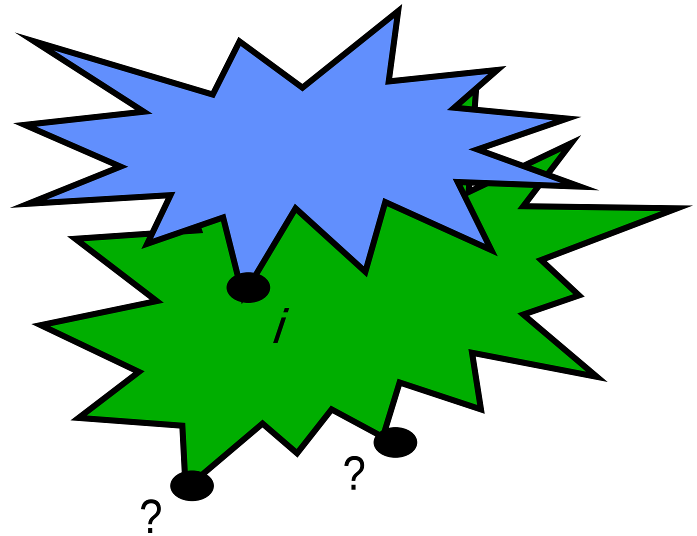


unwanted overlap

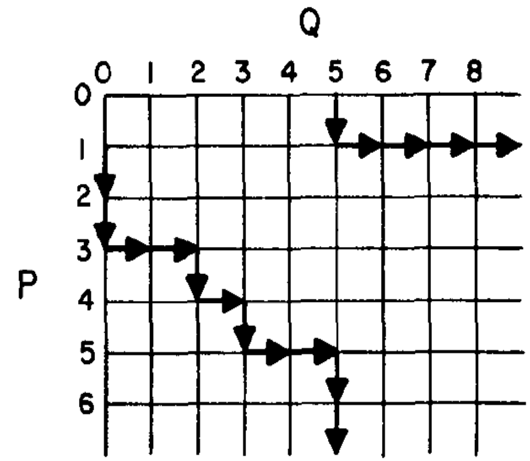
- Very often solved with user interaction

Contour Stitching

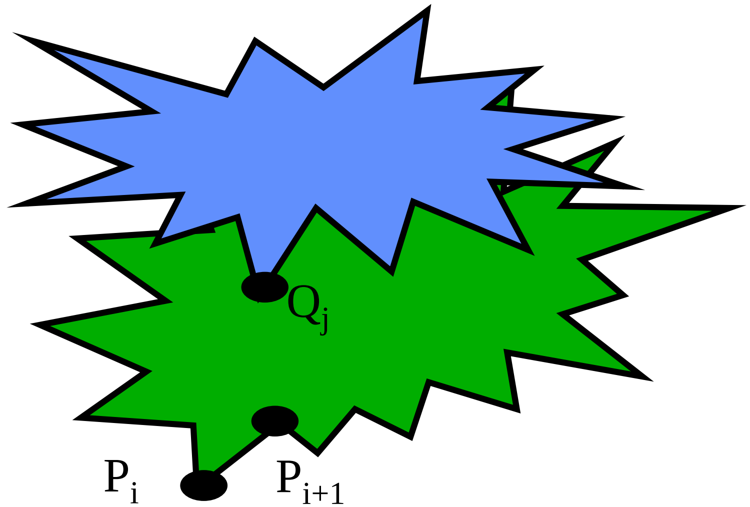
- Problem:
  - Given: 2 two-dimensional *closed* curves
    - Curve #1 has *m* points
    - Curve #2 has *n* points
  - Which point(s) does vertex *i* on curve one correspond to on curve two?

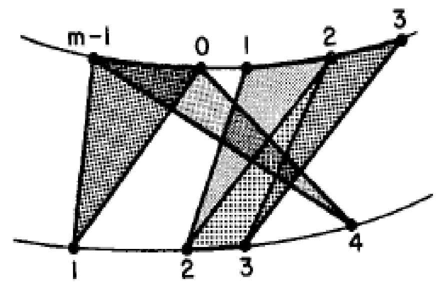


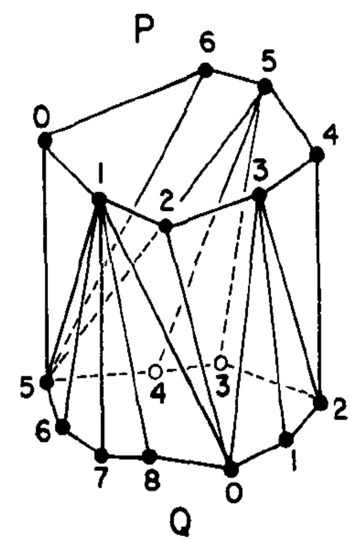
A Solution



- Fuchs, *et. al.*
  - Graph-based optimization problem
    - Find acceptable path with minimum cost (e.g. sum of length of edges)
  - 1 stitch consists of:
    - 2 spans/connecting edges between curves
    - 1 contour segment
  - Triangles of $\{P_i,Q_j,P_{i+1}\}$ or $\{Q_{j+1},P_i,Q_j\}$
    - Consistent normal directions







H. Fuchs, Z.M. Kedem, and S. Uselton. *Optimal surface reconstruction from planar contours*. Communications of the ACM, 20(10):693--702, October 1977

End

Visualization

Direct Volume Rendering

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Acknowledgements

- This lecture is based on
  - Lectures by
    - John Morris, UWA
    - W. Shen, Ohio State University
    - Torsten Moeller, Simon Fraser University, Vancouver
    - Chris Johnson, Scientific Computing and Imaging group at the University of Utah
    - Markus Hadwiger
  - Tutorial IEEE Visualization 2002 / 2003
    - High-Quality Volume Graphics on Consumer PC Hardware
  - Book Visualisierung by Schumann and Mueller
  - Book Real-Time Volume Graphics, Engel et al.

Further Reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - *Chap 8: Arrange Spatial Data*
- Overview of visualization. Schroeder and. Martin. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 3–39. Elsevier, 2005.
- Real-Time Volume Graphics. Engel, Hadwiger, Kniss, Reza-Salama, and Weiskopf. AK Peters, 2006.

"The purpose of computing is insight not numbers"

Richard Hamming

Numerical Methods for Scientists and Engineers (McGraw Hill 1962)

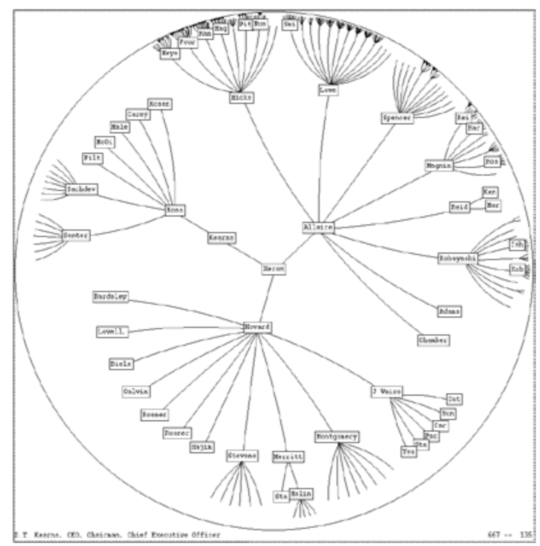
“The purpose of visualization is insight, not pictures“

Ben Shneiderman

1999

Visualization

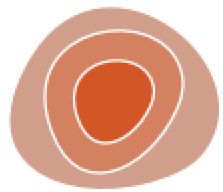
- Scientific Visualization
  - Physical / simulated data
    - Automotive
    - Weather
    - Medical
    - ...
- Information Visualization
  - Abstract, non-physical data
    - Financial/business
    - Statistics
    - Software
    - Text
    - W W W
    - ...



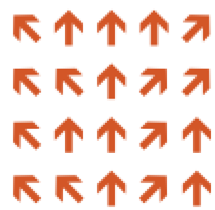
Arrange Spatial Data

- **Use Given**
  - Geometry
    - *Geographic*
    - *Other Derived*

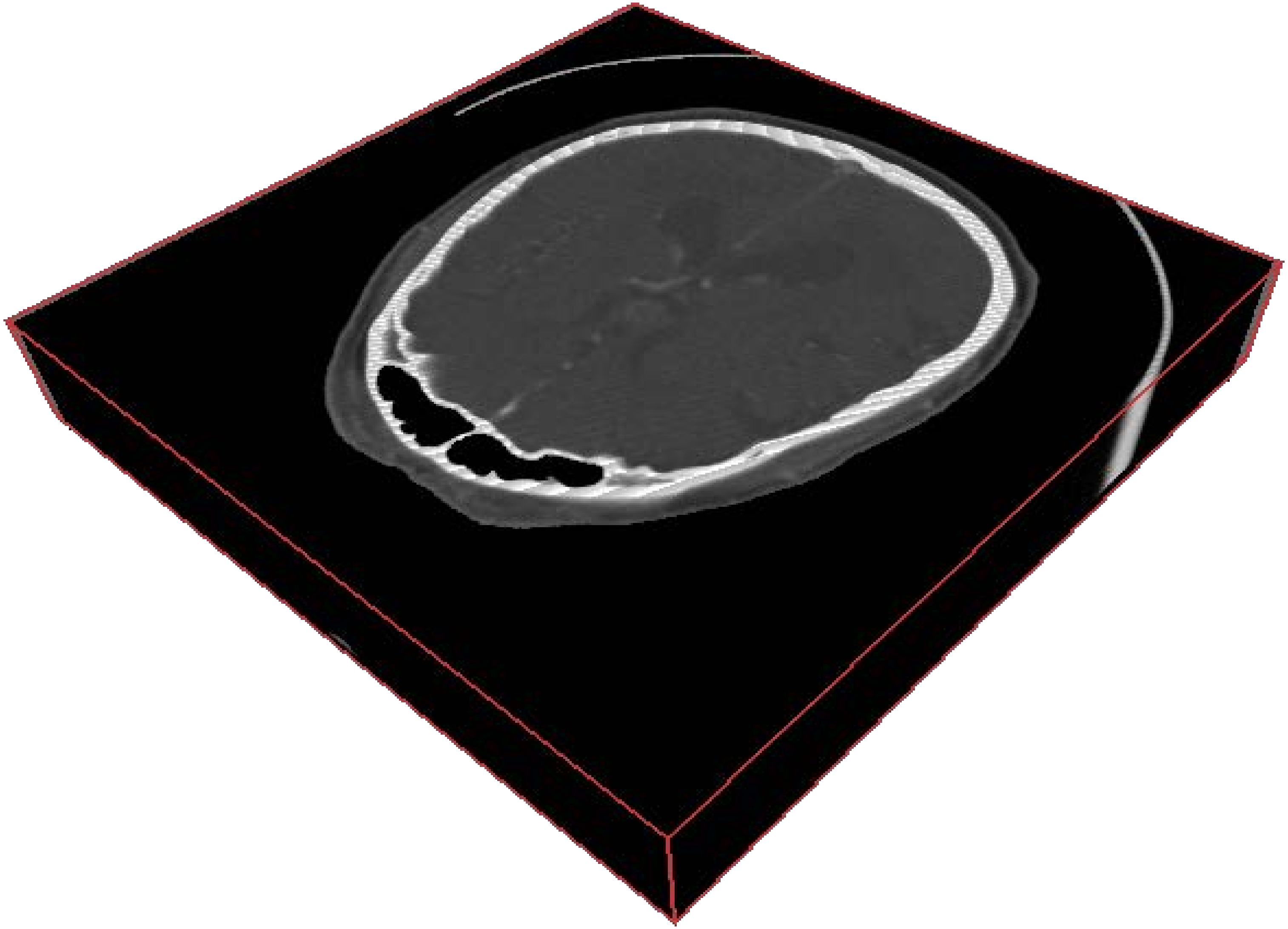
  - Spatial Fields
    - *Scalar Fields (one value per cell)*
      - *Isocontours* ***(see lecture 2)***
      - *Direct Volume Rendering* ***(this lecture)***



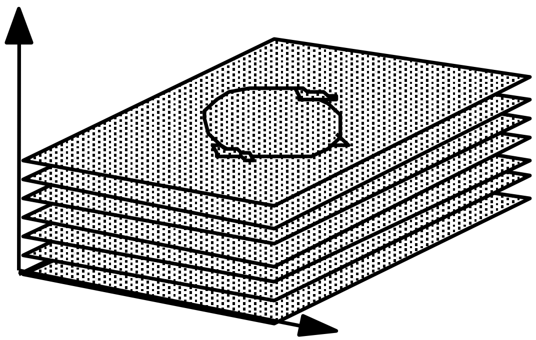
    - *Vector and Tensor Fields (many values per cell)*
      - *Flow Glyphs (local)*
      - *Geometric (sparse seeds)*
      - *Textures (dense seeds)*
      - *Features (globally derived)*



Typcial Volumetric Dataset



Scalar 3D Grids

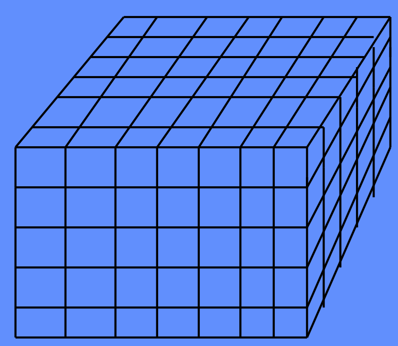


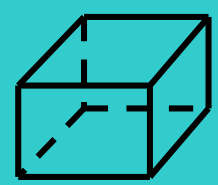
x_i	y_i	z_i	f
0.00	0.00	0.00	243
0.00	0.00	0.12	175
•	•	•	•
0.00	0.00	1.00	186
0.00	0.12	0.00	187
•	•	•	•

- $f_{\mathbf{ijk}} = f(X_{\mathbf{i}}, Y_{\mathbf{j}}, Z_{\mathbf{k}})$
  - i = 1,...,N; j = 1,...,M; k = 1,...,P
- Data sources
  - Medical Scanners, CT, MRI, etc.
  - Simulation of noise in a room …
- Data usually given on a regular grid, a 3D array of scalar values
- Source
  - Measured or simulated
- Dimension
  - Domain (german: Definitionsbereich)
    - 3D (volumetric)
  - Range (german: Wertebereich)
    - 1D (scalar)
- Topology
  - Regular grid

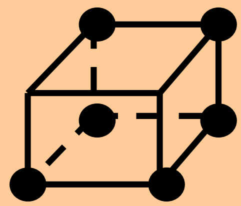
Voxels

Two definitions





A voxel is a cubic cell, which
has a single value that covers
the entire cubic region

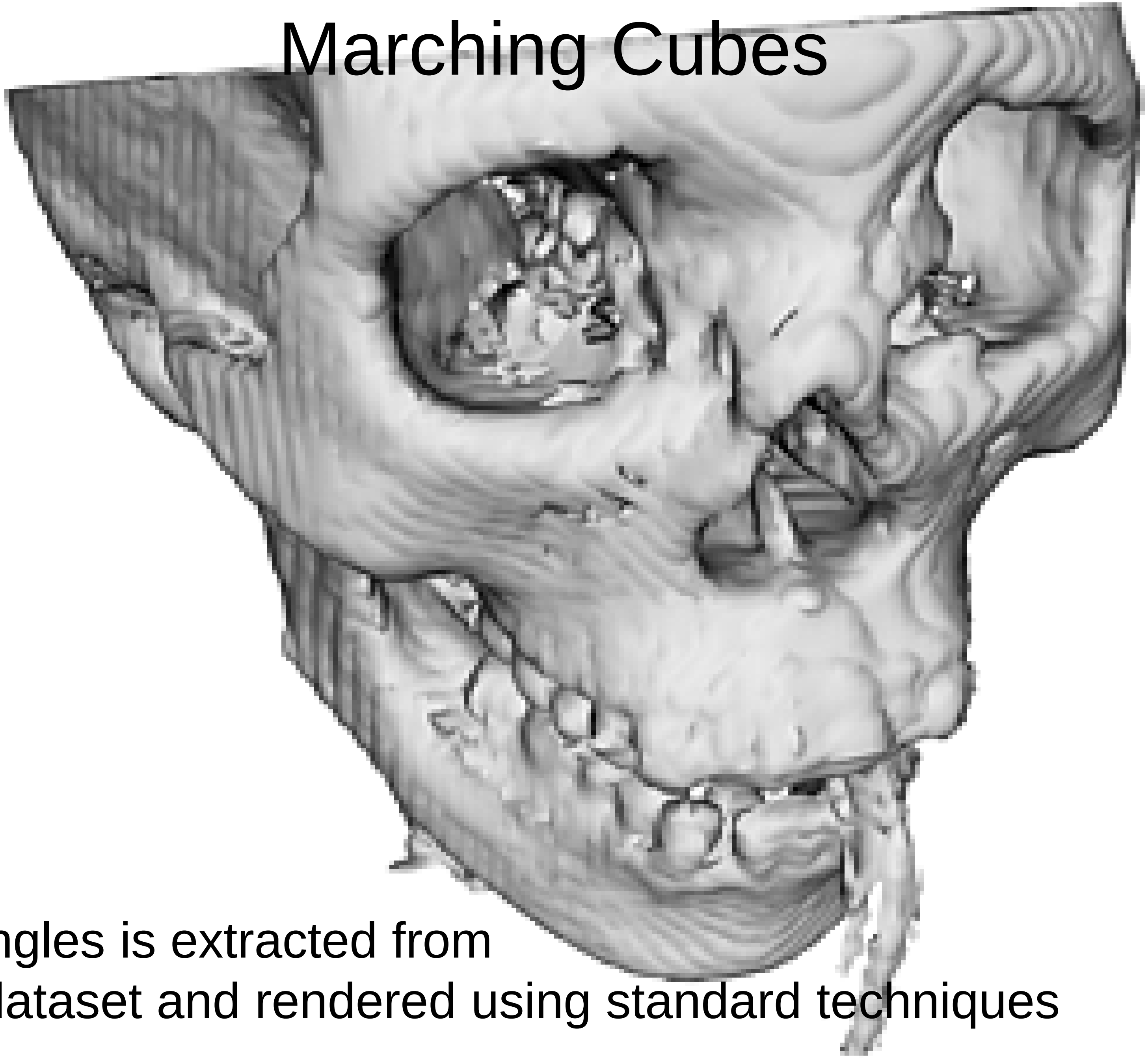


A voxel is a data point
at a corner of the cubic cell
The value of a point inside the
cell is determined by interpolation

Indirect Volume Rendering

- Marching cubes (iso surface extraction, lecture 2, online)
- Point plots
- 3D Glyphs

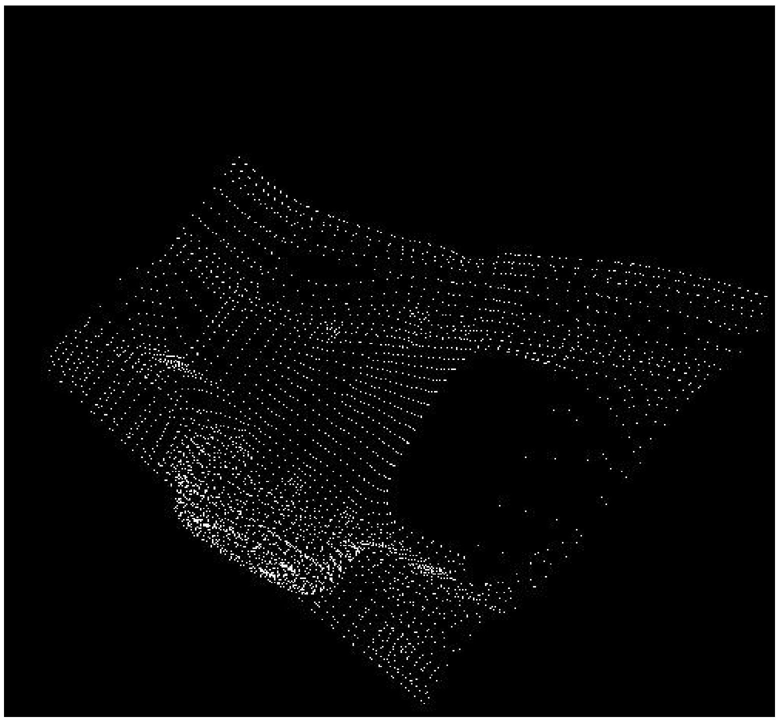
Marching Cubes

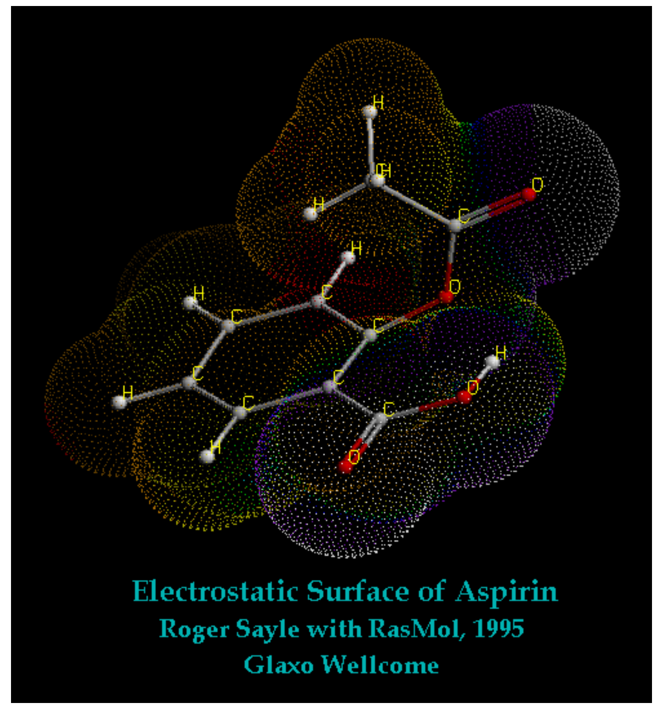


A set of triangles is extracted from
volumetric dataset and rendered using standard techniques

Point Plots

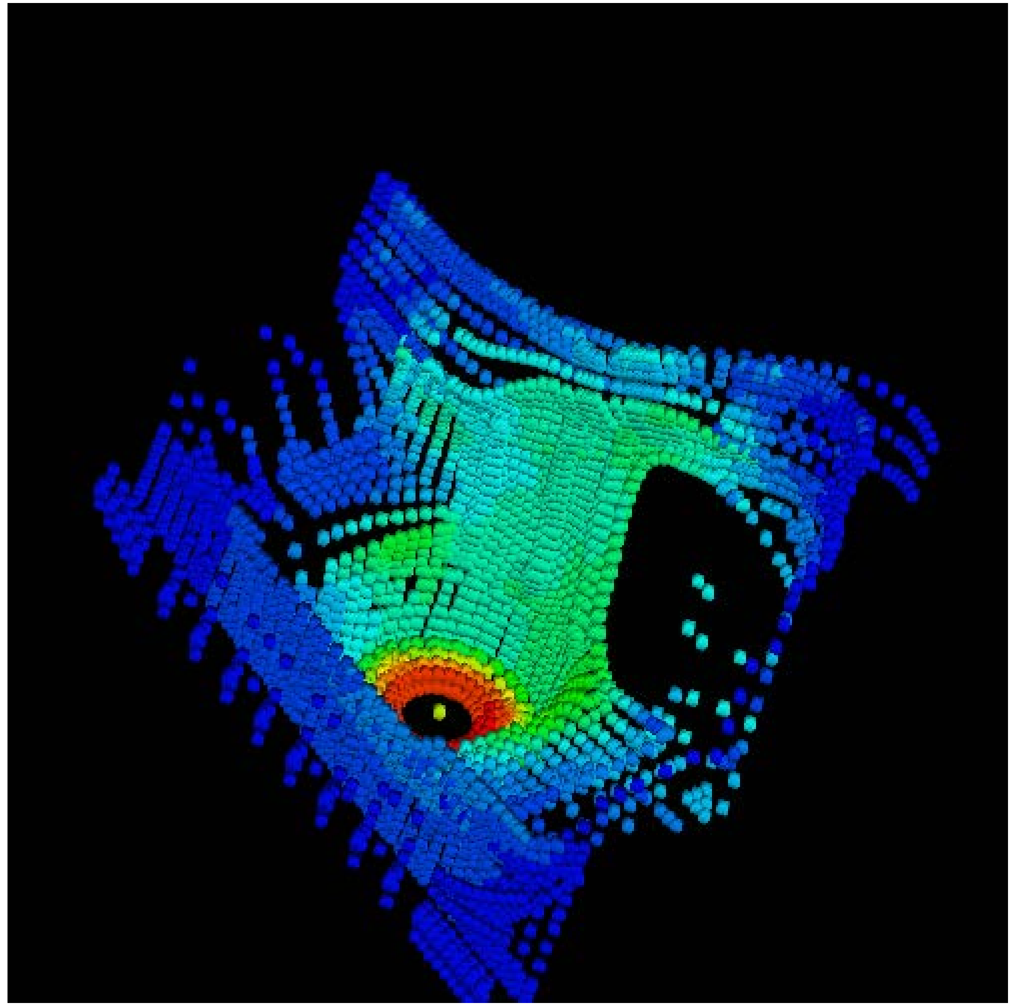
- Display points where volume data has a certain property e.g. data value, gradient, …
- Animation can bring out the surface or pattern
- Depth Cueing aids depth perception





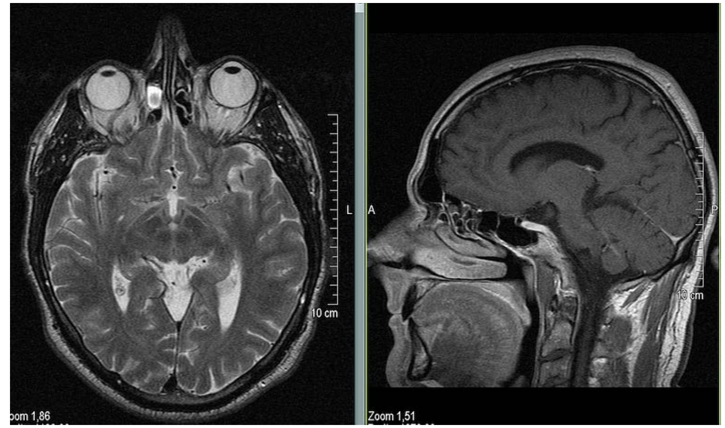
3D Glyphs

- Spheres or cubes dispersed throughout the volume
  - color-coded
  - optional shape-controlled
- Display spheres or cubes, where volume data has a certain property e.g. data value, gradient, …



Slicing and Similar Techniques

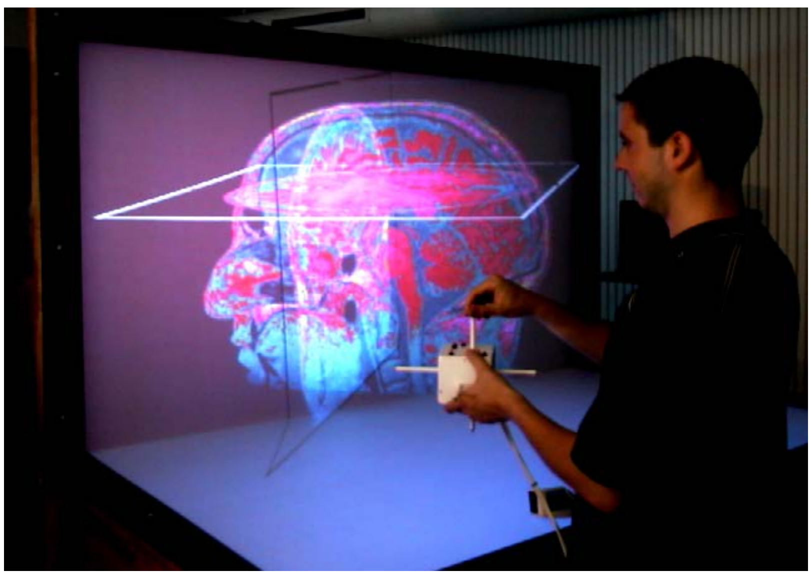
- Cross sections
- Fence diagrams
- Chair cuts

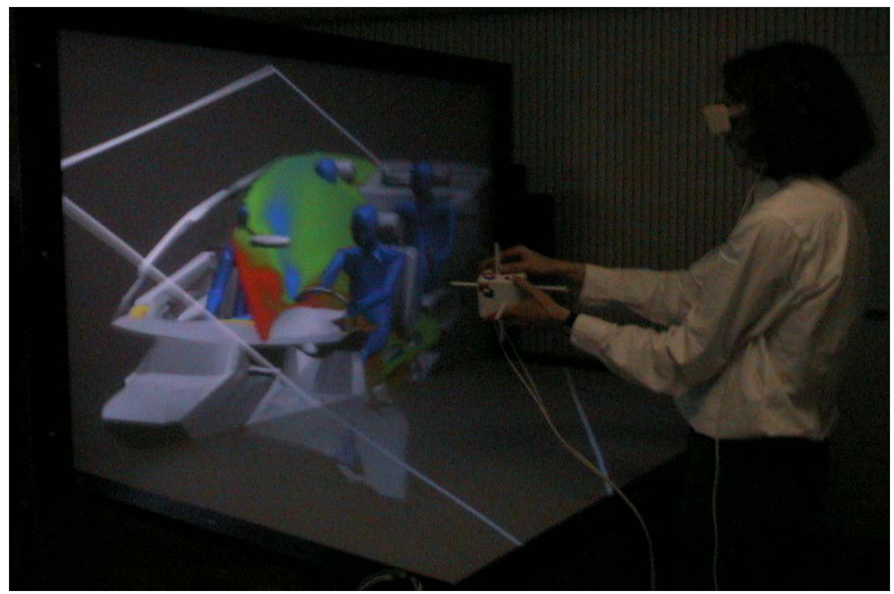


- 2D proxies for displaying the volume

Slicing

- 2D slicing planes in 3D
- Often: 3 planes orthogonal to a coordinate axis
- Arbitrary orientation
  - Specify the normal and a point

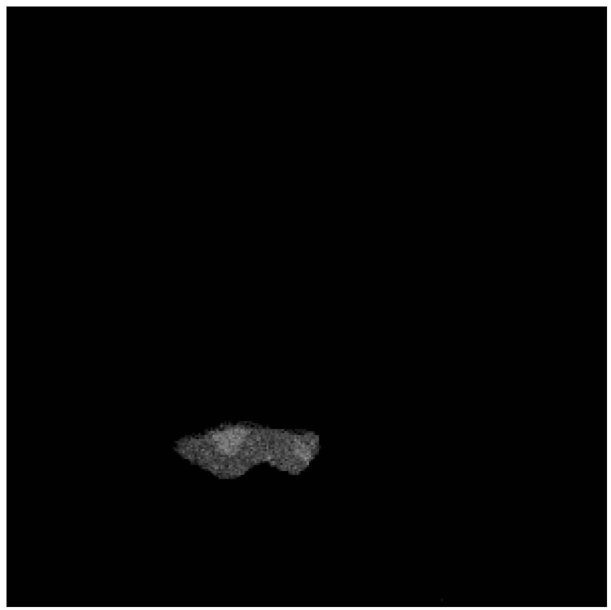




Cubic Mouse Video

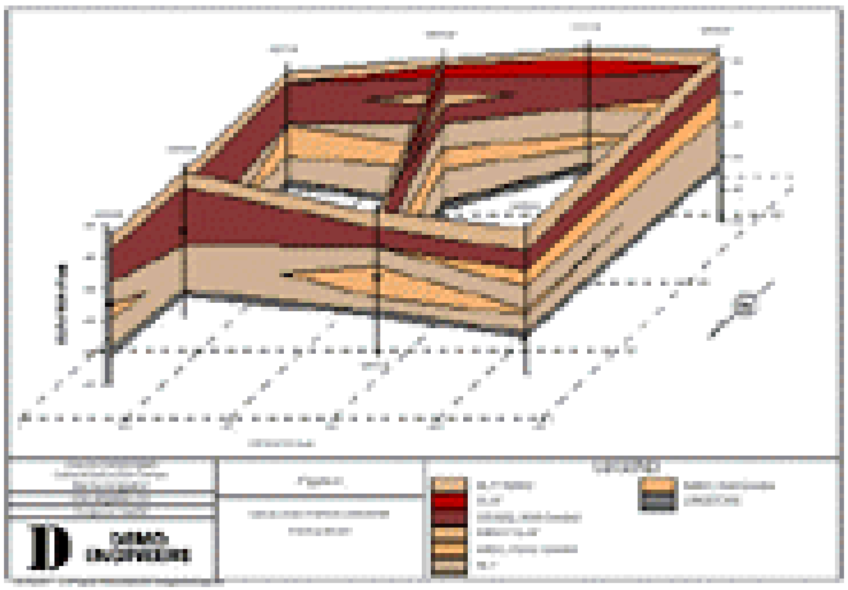
2D Flip Book of Slices

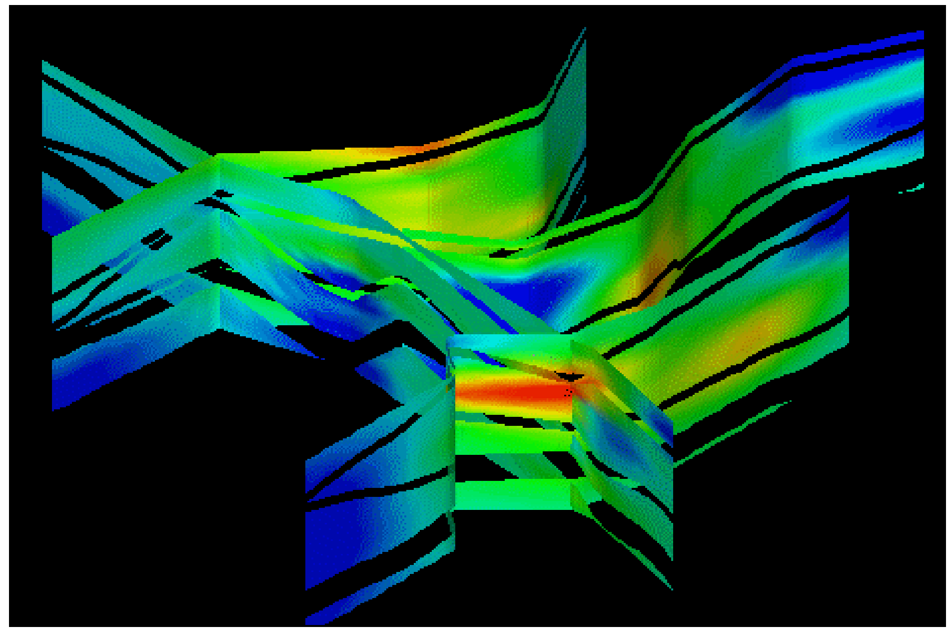
- Rather than view the 2D slice in 3D, rapidly play a sequence of parallel slice planes in a movie loop.
- Sometimes difficult to build a complete mental model.
- Example: diamondiferous eclogite http://www.ctlab.geo.utexas.edu/imagefolio/animations/tayflip.html



Fence Diagram

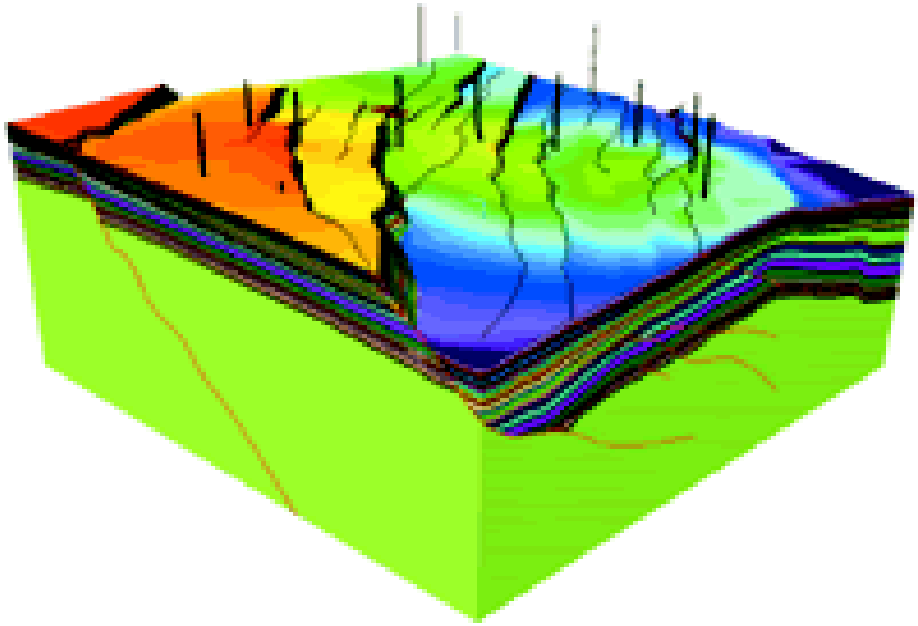
- A sequence of slicing planes connected together

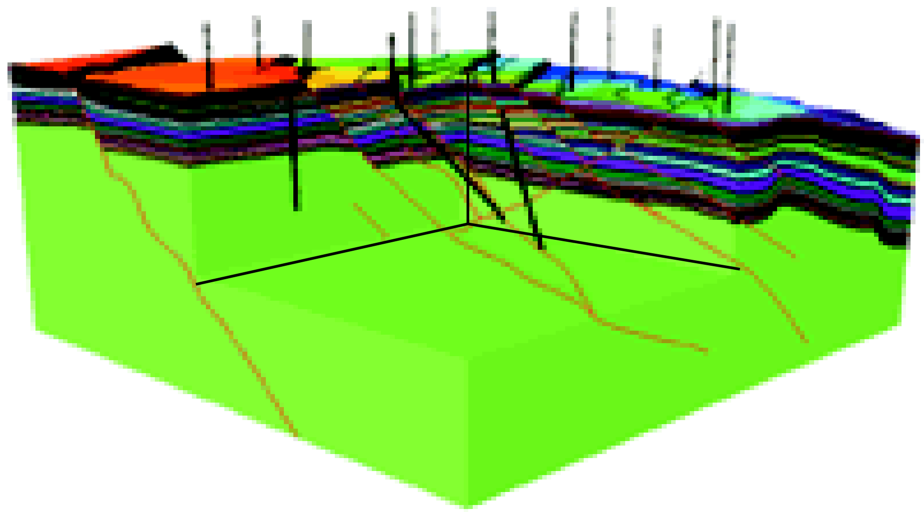




Chair-Cut

- One octant is cut away

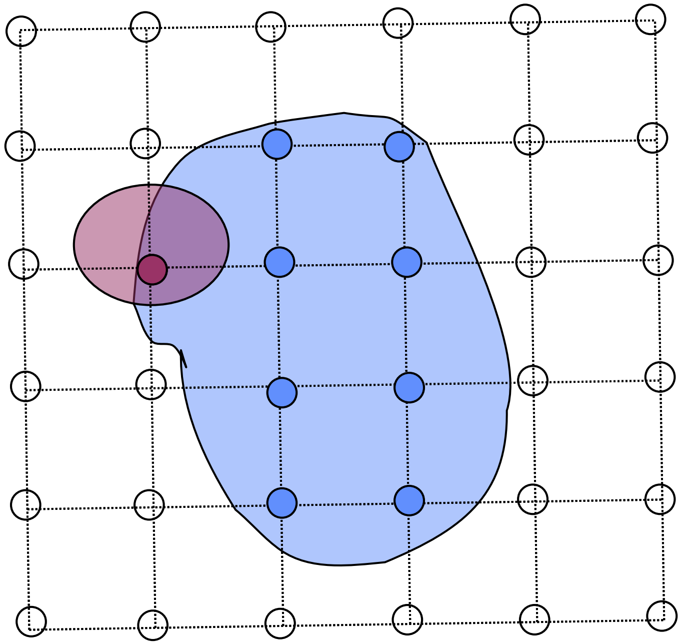


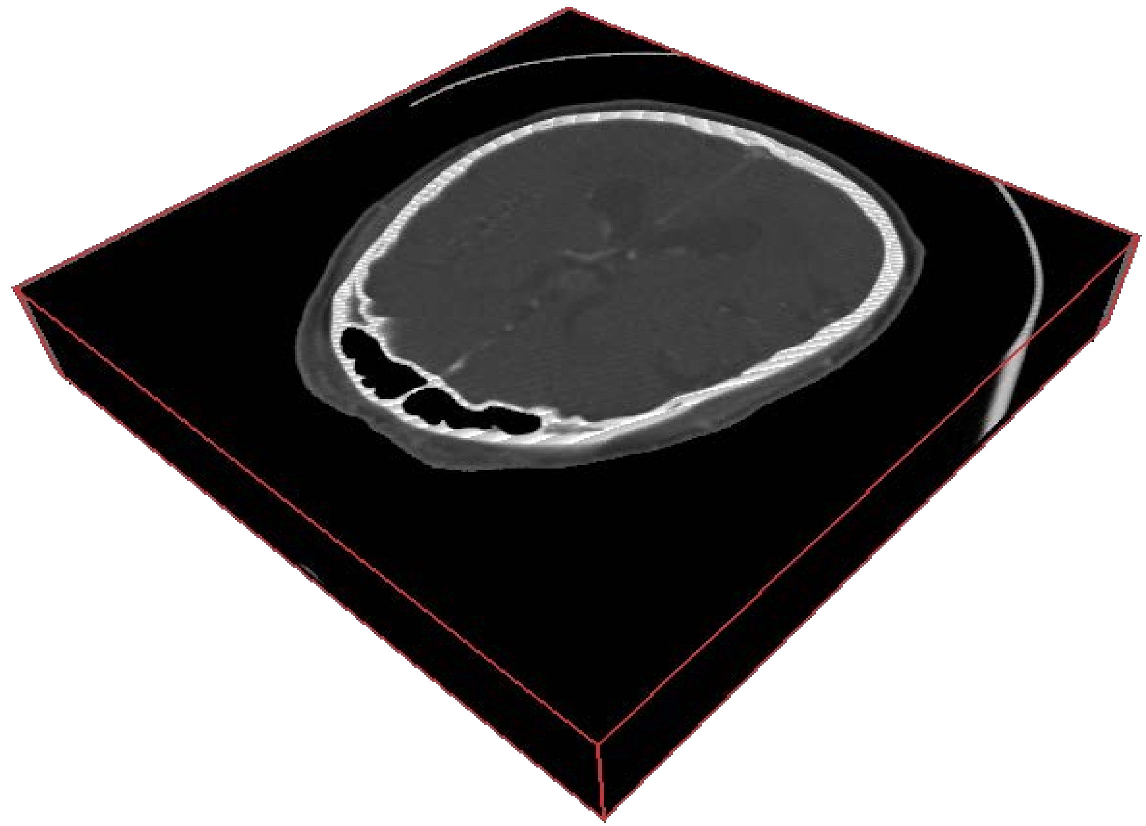


EarthVision image courtesy of BP Amoco

Direct Volume Rendering (DVR)

Direct Volume Rendering (DVR)



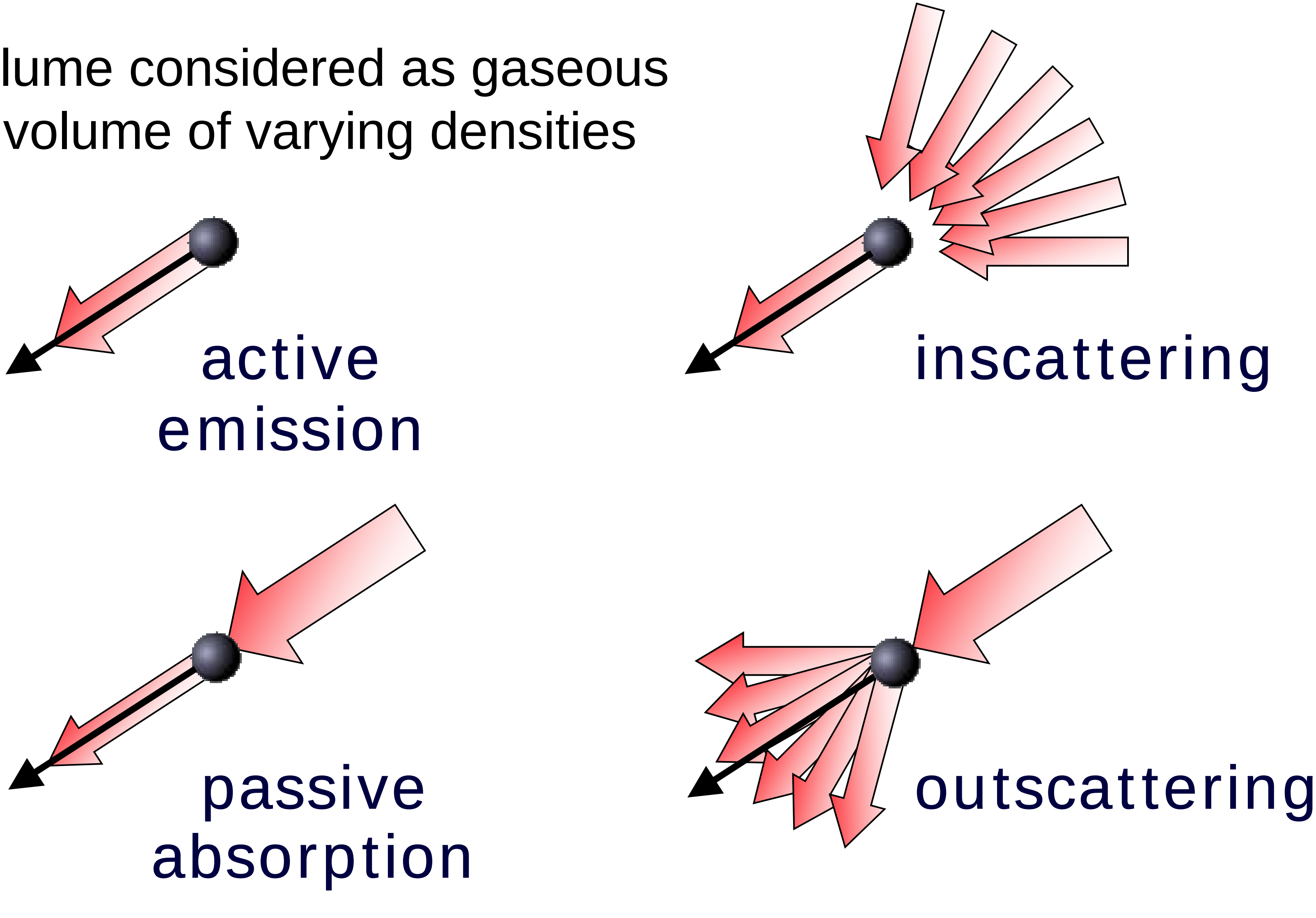


Given: A volume as a 3D array of scalar data values e.g. density

Goal: Compute image of the structures inside the volume from volumetric data without using an intermediate representation

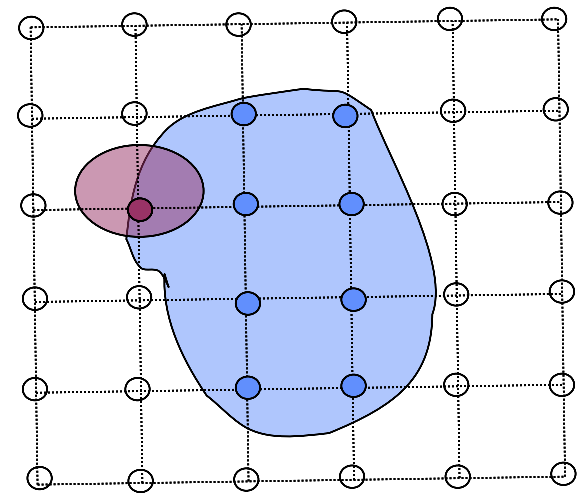
Optical Models for Gaseous Particle Volumes

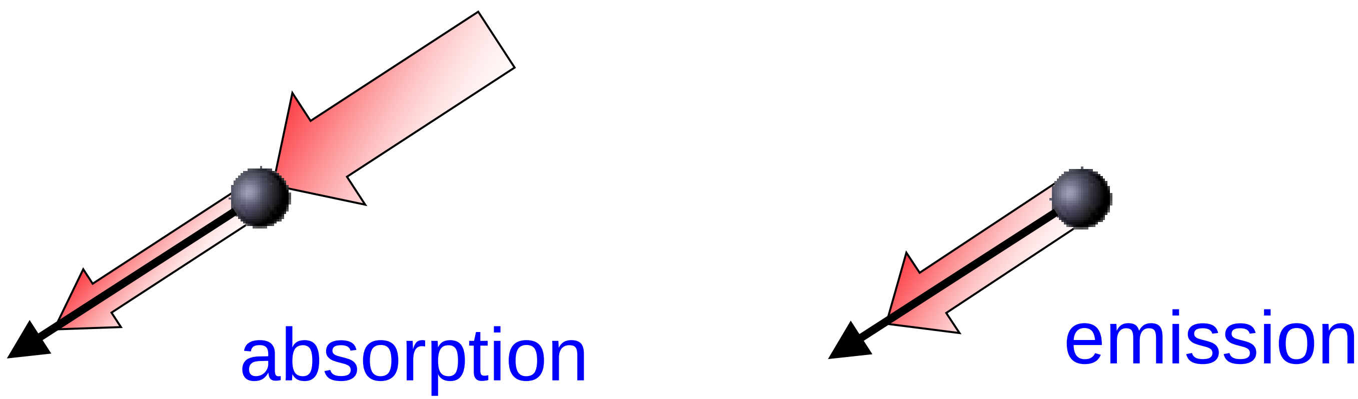
Data volume considered as gaseous particle volume of varying densities



3D Data Volume

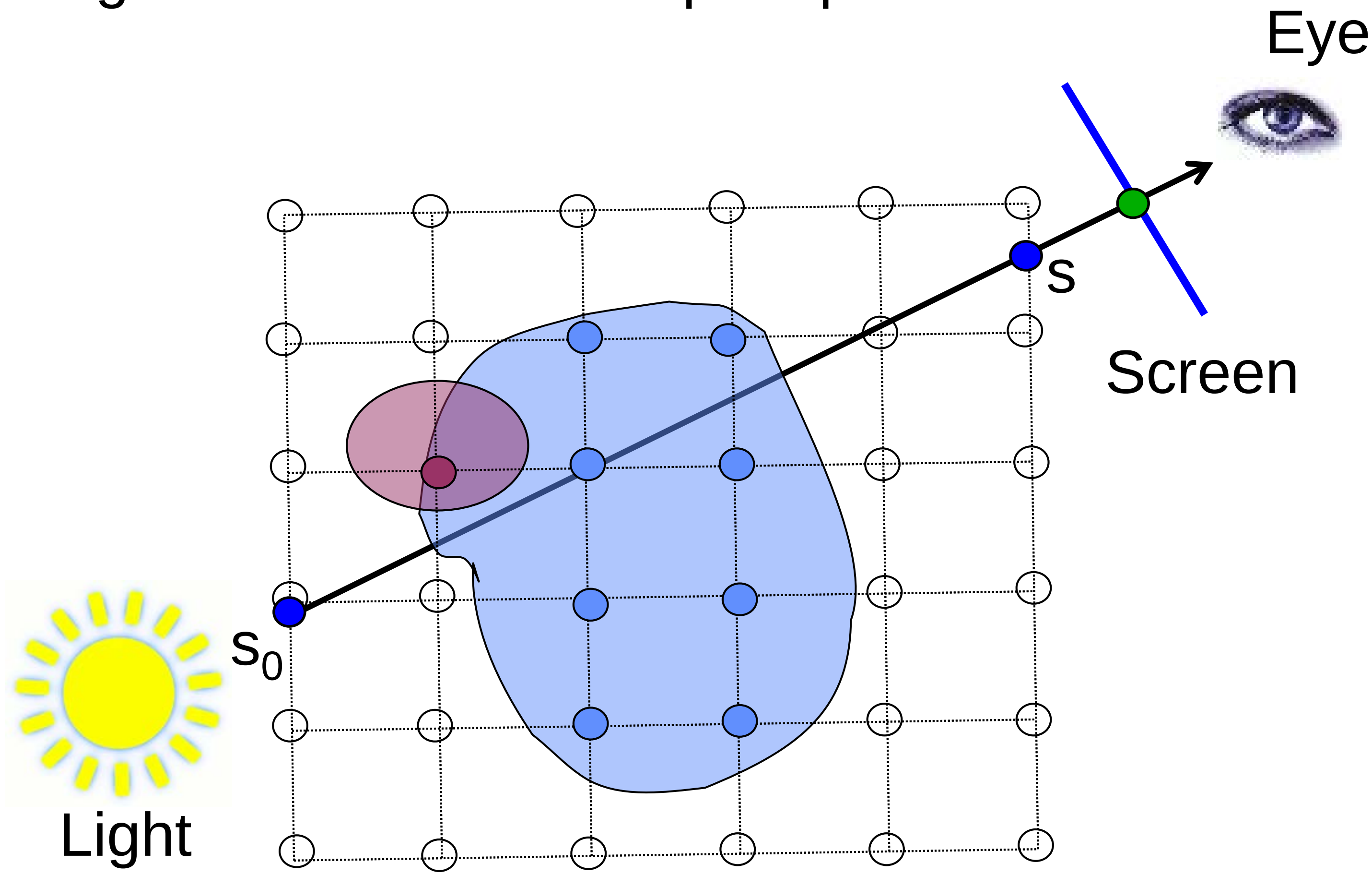
- Light interaction
  - Absorption only
  - Emission only
  - Absorption + emission (MOST OFTEN USED)
  - Scattering + shading/shadowing
  - Multiple scattering





Volume Rendering Integral

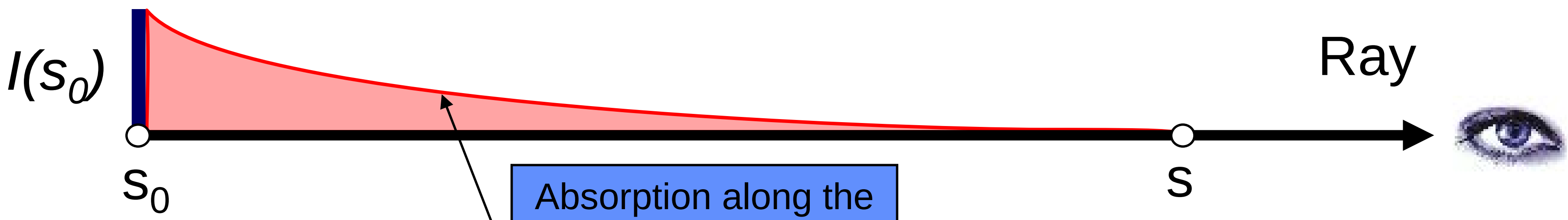
- Integrate light/color contributions along rays through the volume to compute pixel value



Volume Rendering Integral

What must be integrated?

physically correct: emission and absorption of light



Absorption along the ray from s_0 to s

Initial incoming intensity at s_0

$$\overline{I(s)} = \boxed{I(s_0)}\boxed{e^{-\tau(s_0,s)}}$$

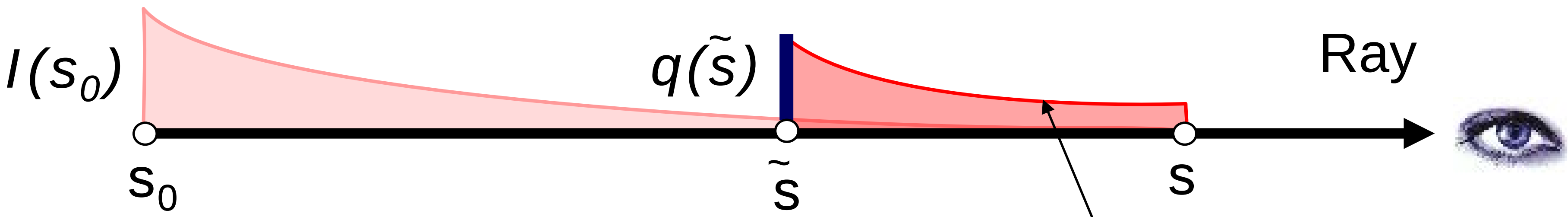
Optical depth τ
Absorption κ

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds.$$

The optical depth expresses the quantity of light removed from a beam by scattering or absorption during its path through a medium

Volume Rendering Integral

Also needs to consider light emission of particles along the ray path



Active Emission at $\tilde{s}$

Absorption along the distance $\tilde{s}$ - s

Volume Rendering Integral

$$\overline{I(s)} = I(s_0)e^{-\tau(s_0,s)} + \int_{s_0}^{s} \boxed{q(\tilde{s})}\boxed{e^{-\tau(\tilde{s},s)}}\, d\tilde{s}$$

- Interpretation of the volume render: Radiation towards the eye consists of
  - Sum of photons emitted from all points along the line segment, attenuated by the integrated absorptivity of the intervening medium, and
  - additional, the attenuated contribution from radiation entering the boundary surface

Volume Rendering Integral

$$\overline{I(s)} = I(s_0)e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})e^{-\int_{\tilde{s}}^{s}\kappa(t)dt} d\tilde{s}$$

$$\overline{I(s)} = I(s_0)e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})e^{-\tau(\tilde{s},s)} d\tilde{s}$$

With Transparency $$T(s_1, s_2) = e^{-\tau(s_1,s_2)}$$

$$\overline{I(s)} = I(s_0)T(s_0, s) + \int_{s_0}^{s} q(\tilde{s})T(\tilde{s}, s)d\tilde{s}$$

Transparency

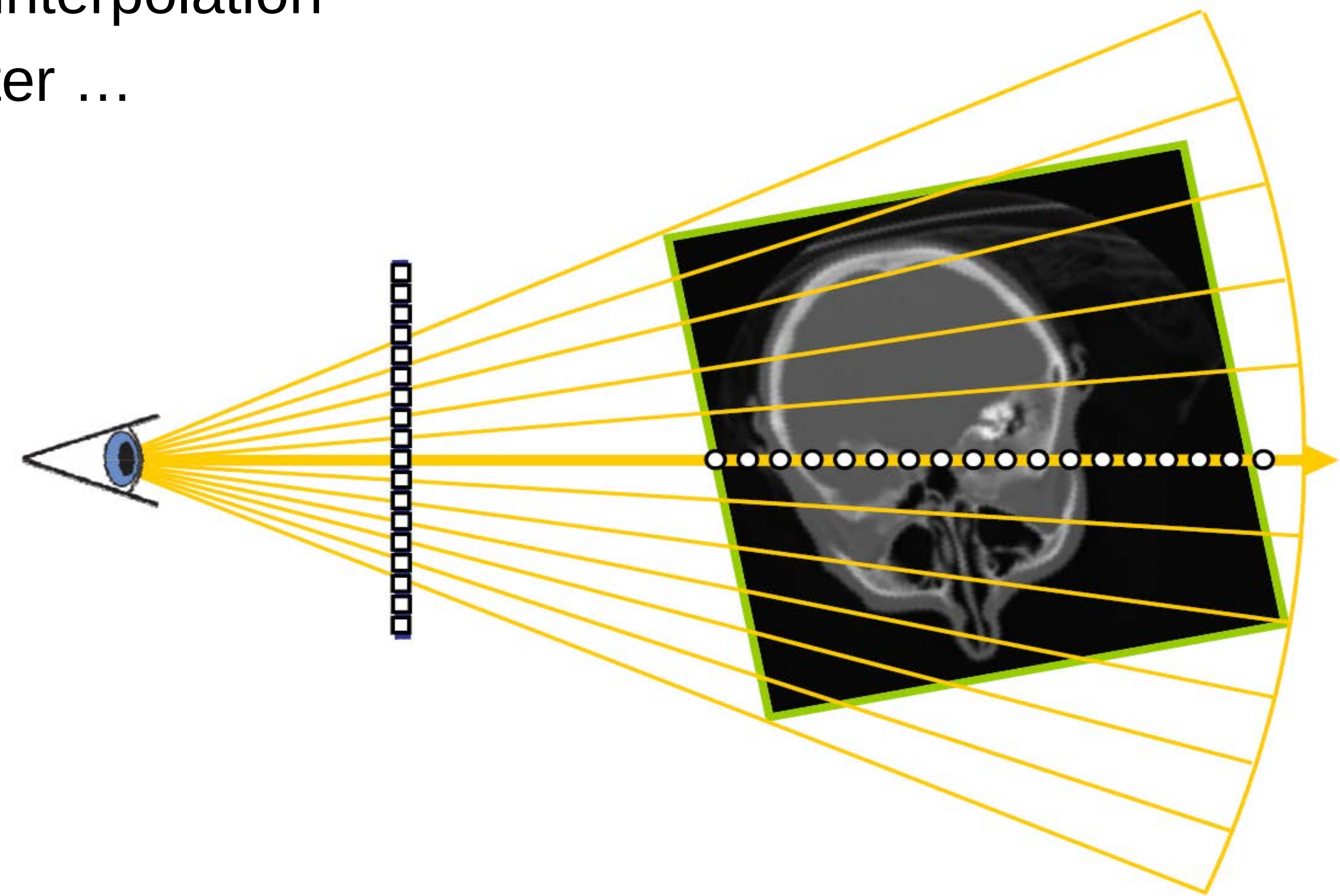
- Opacity
  - A material property preventing light from passing through the object (α = 1)
- Transparency
  - A material property that allows light to pass through the object (α = 0)
- Translucency, semi-transparency
  - Graded or blurred transp. (0 < α < 1)

- Alpha value is also called opacity

$$\alpha(s_1, s_2) = 1 - T(s_1, s_2)$$

Volume Ray Casting

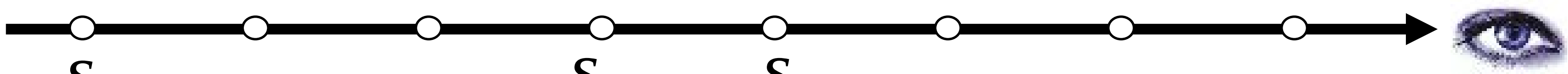
- Numerical approximation of volume rendering integral
- Resample volume at equi-spaced intervals along the ray
- Tri-linear interpolation
- Details later …



Discrete Solution

Resample the scalar field at discrete locations along the viewing ray using color C(s) and opacity α instead of emission q(s) and transparency T:

$\overline{I(s_i)}$ $\overline{I(s_{i+1})}$ Ray



s_0 s_i s_{i+1}

transfer function

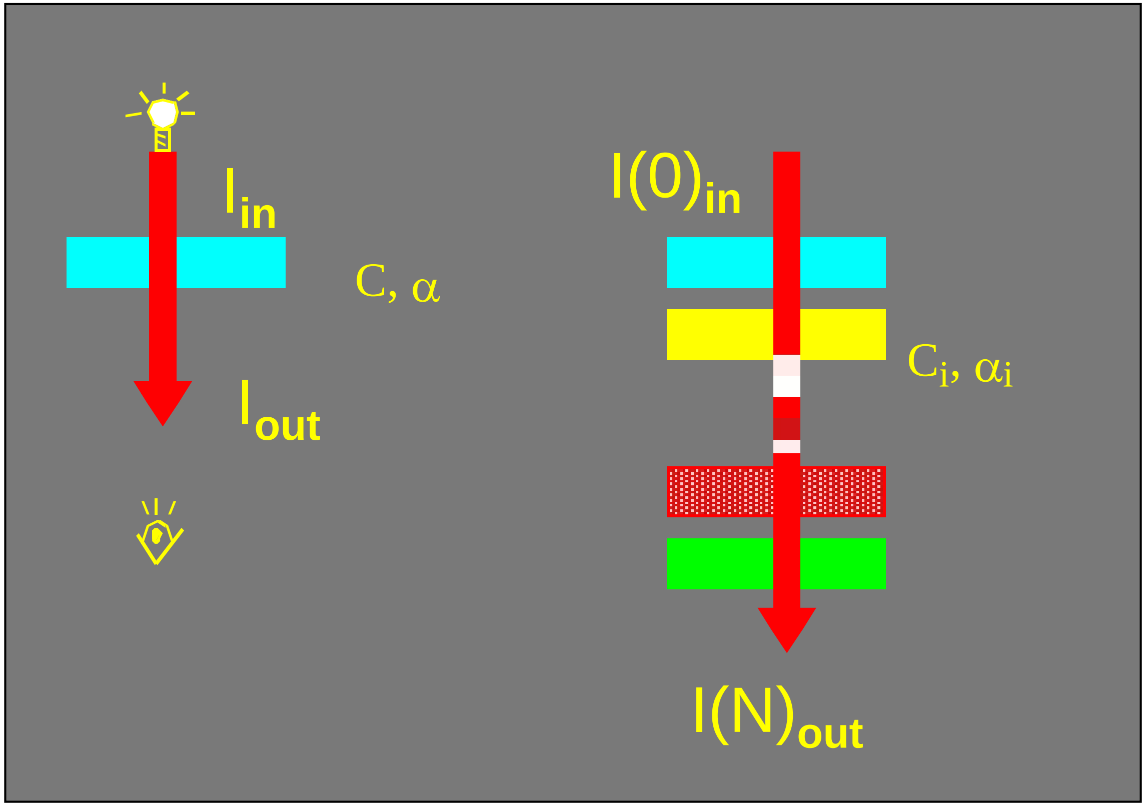
$I(s_0)$ | $C(s_i)\ \alpha(s_i)$ | $C(s_{i+1})\ \alpha(s_{i+1})$

Back-to-front Compositing / Blending with

$$\boxed{\overline{I(s_{i+1})}} = \alpha(s_{i+1})C(s_{i+1}) + (1-\alpha(s_{i+1}))\overline{I(s_i)}$$

accumulated intensity up to position s_{i+1}

Back-to-Front Compositing



$$I_{out} = \alpha\, C + (1-\alpha)\, I_{in} \qquad I(i)_{in} = I(i-1)_{out}$$

Classification through Transfer Function

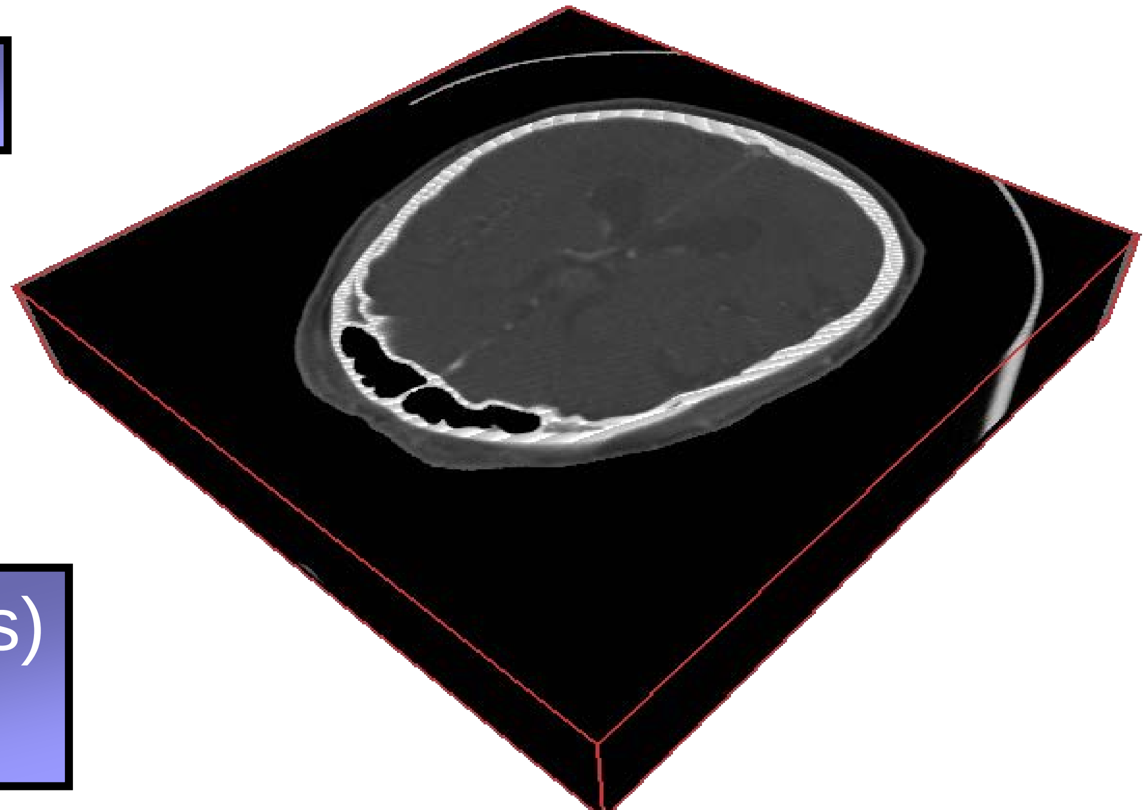
- How to obtain the emission/color values q(s) / C(s) and absorption values $\alpha(s)$ at location s?
- The **transfer function *T*** maps the raw data value *f(s)* at position *s* to color *C(f(s))* and opacity $\alpha(f(s))$!

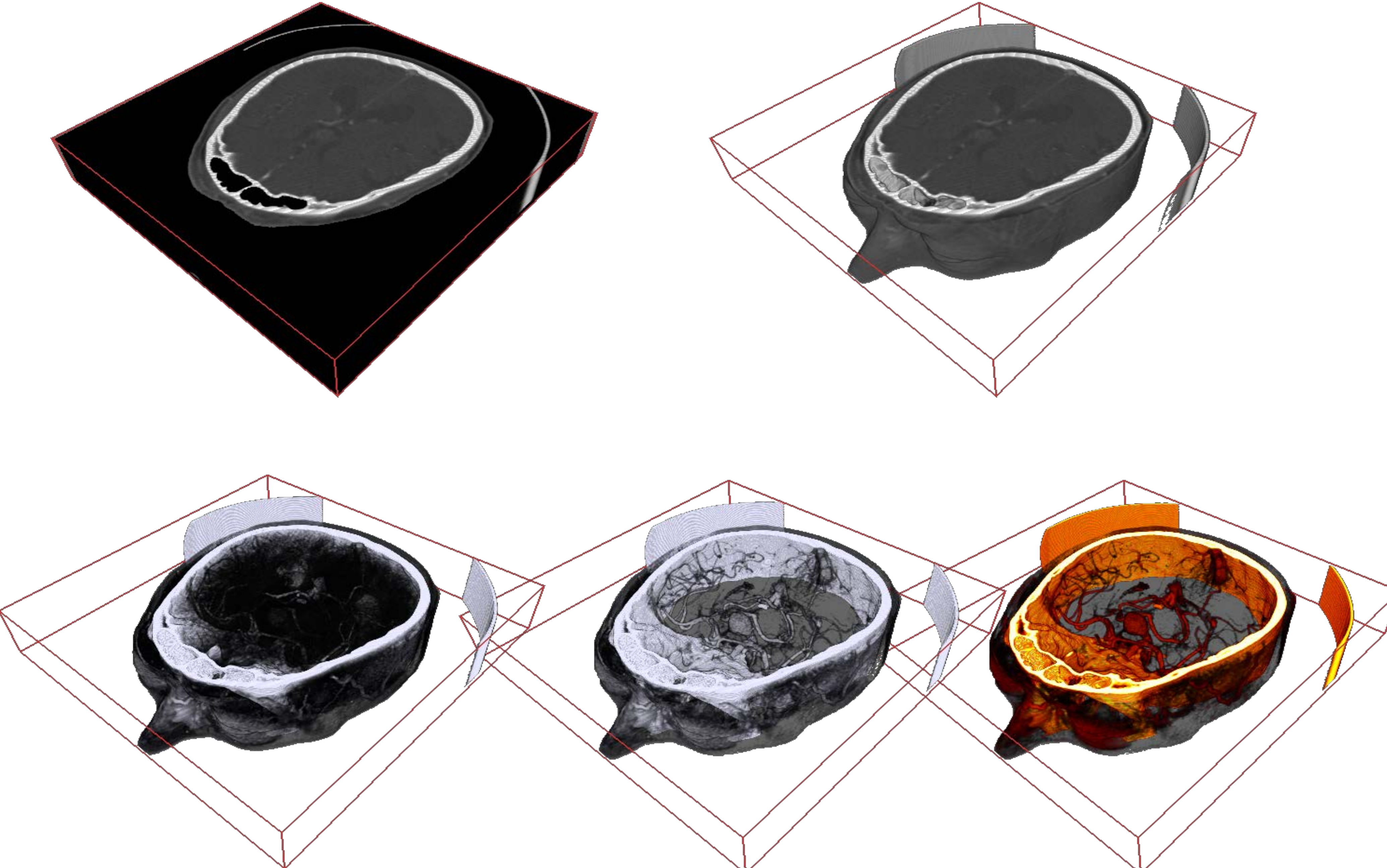
scalar value *f(s)*

T(f(s))

emission C (RGB-values)

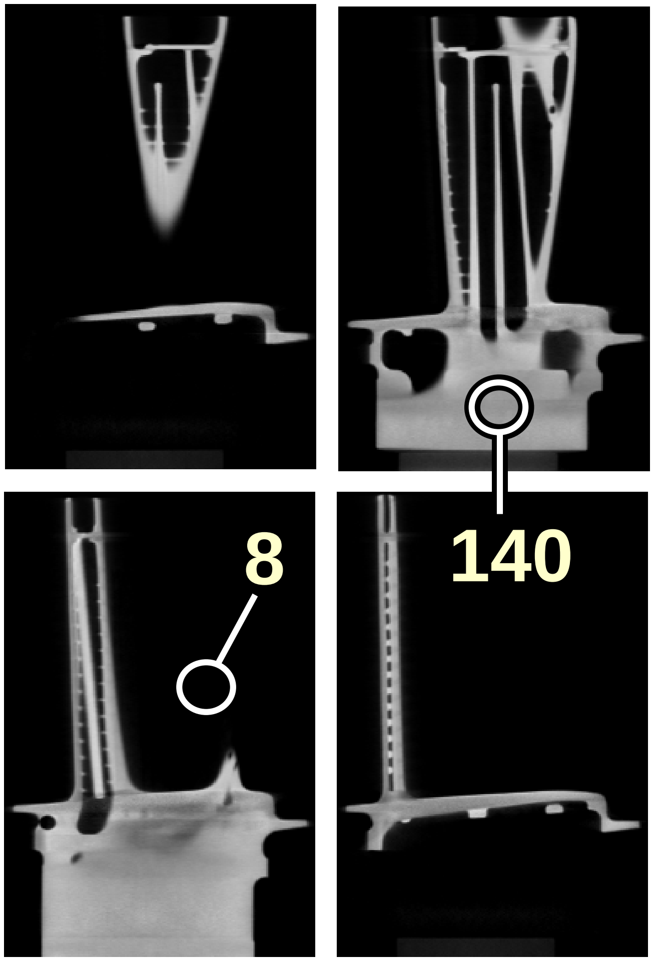
absorption α



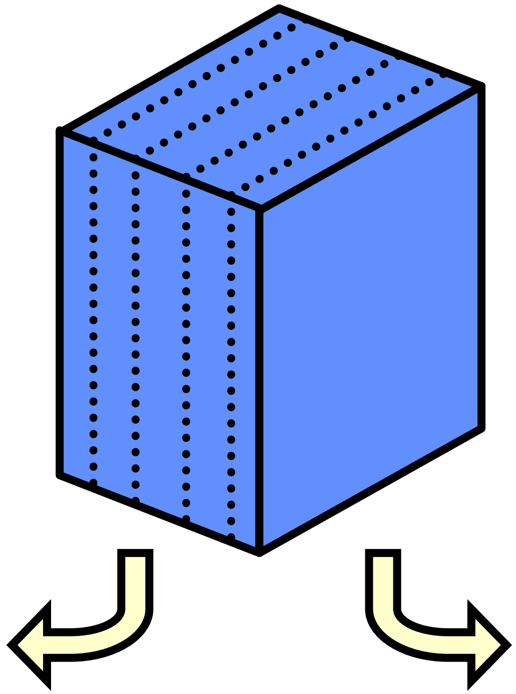


Transfer functions make volume data visible by mapping data values to optical properties

slices of the volume

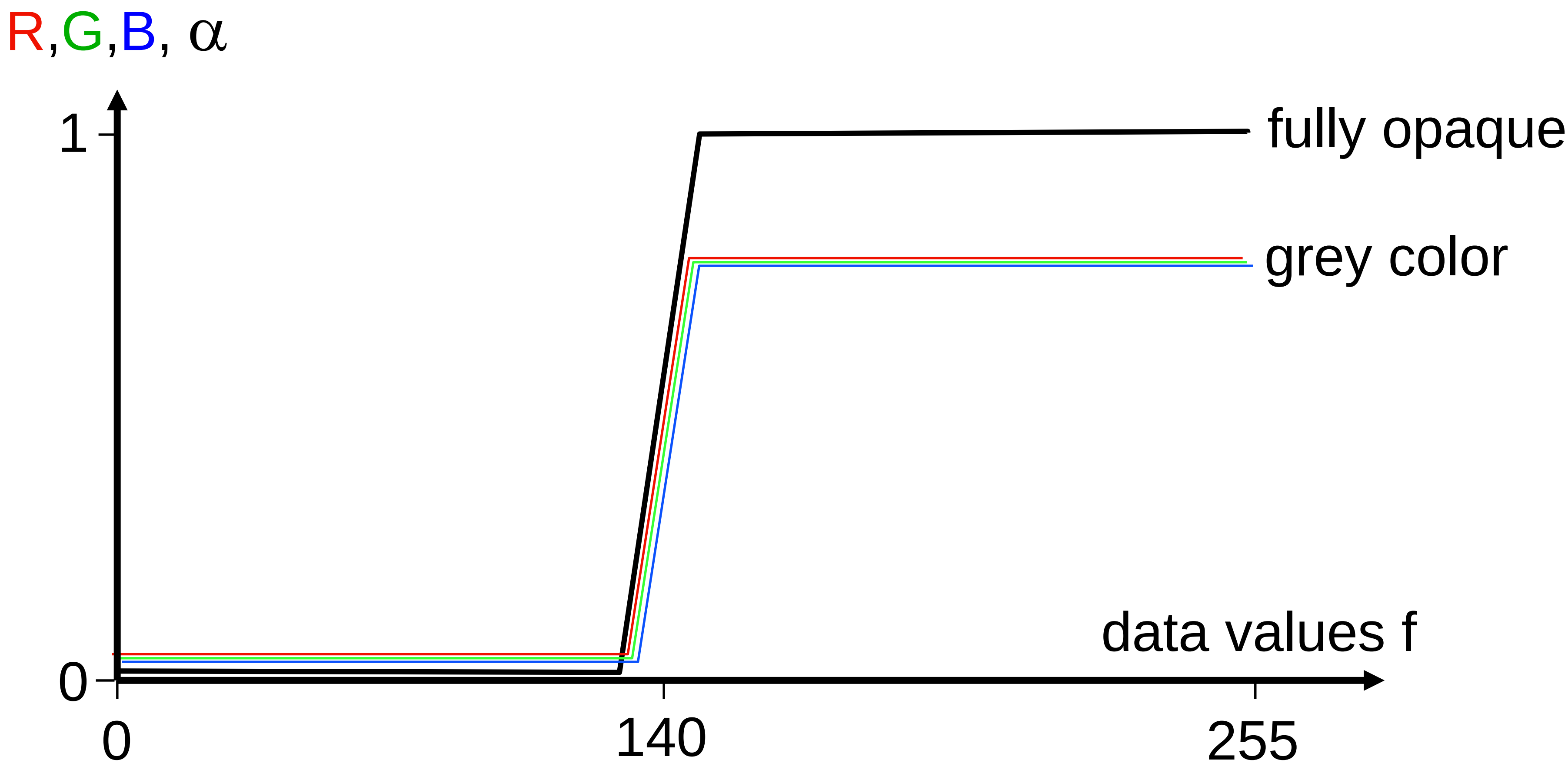


volume data

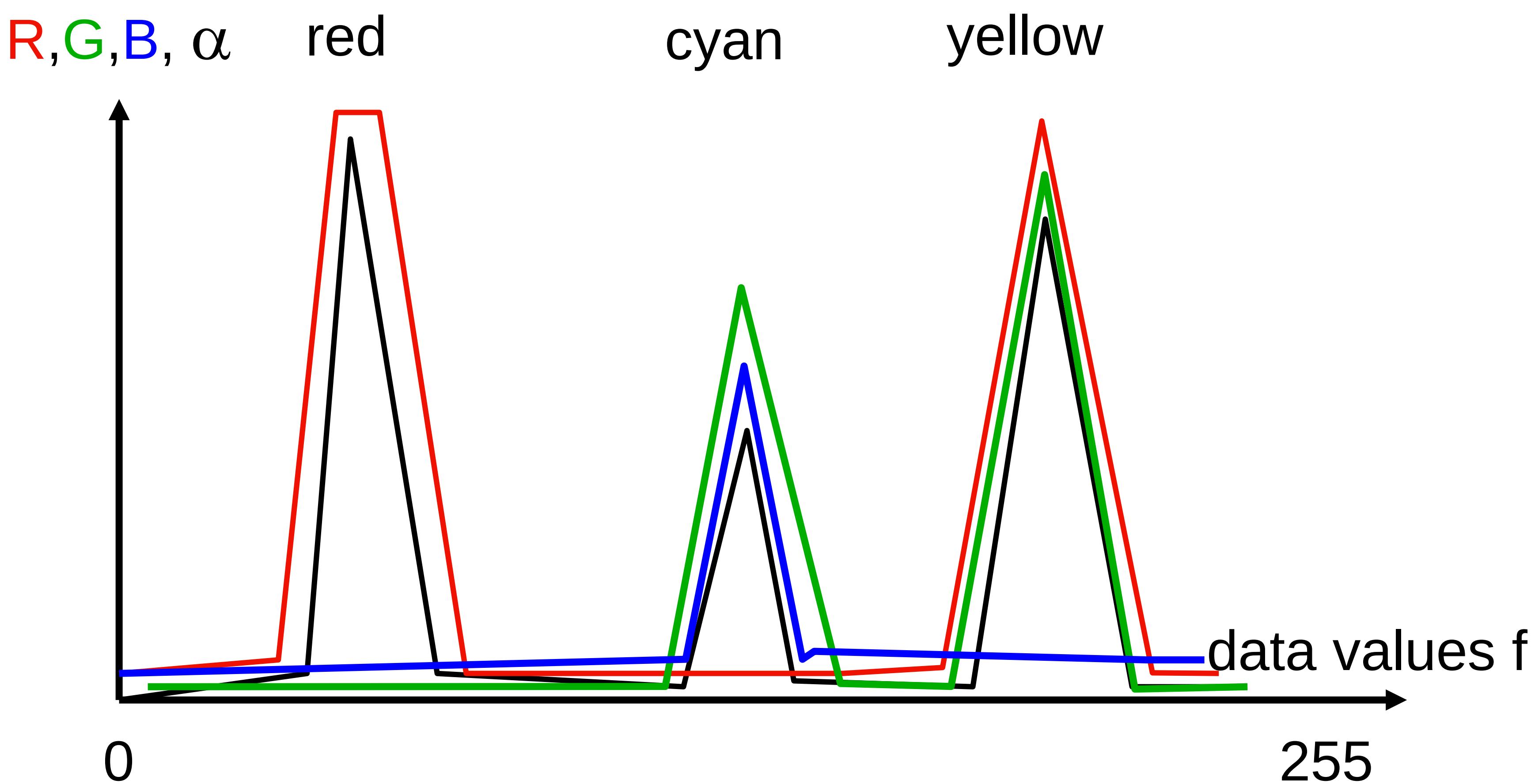


volume rendering

Basic Transfer Function for Data Set on Previous Slide

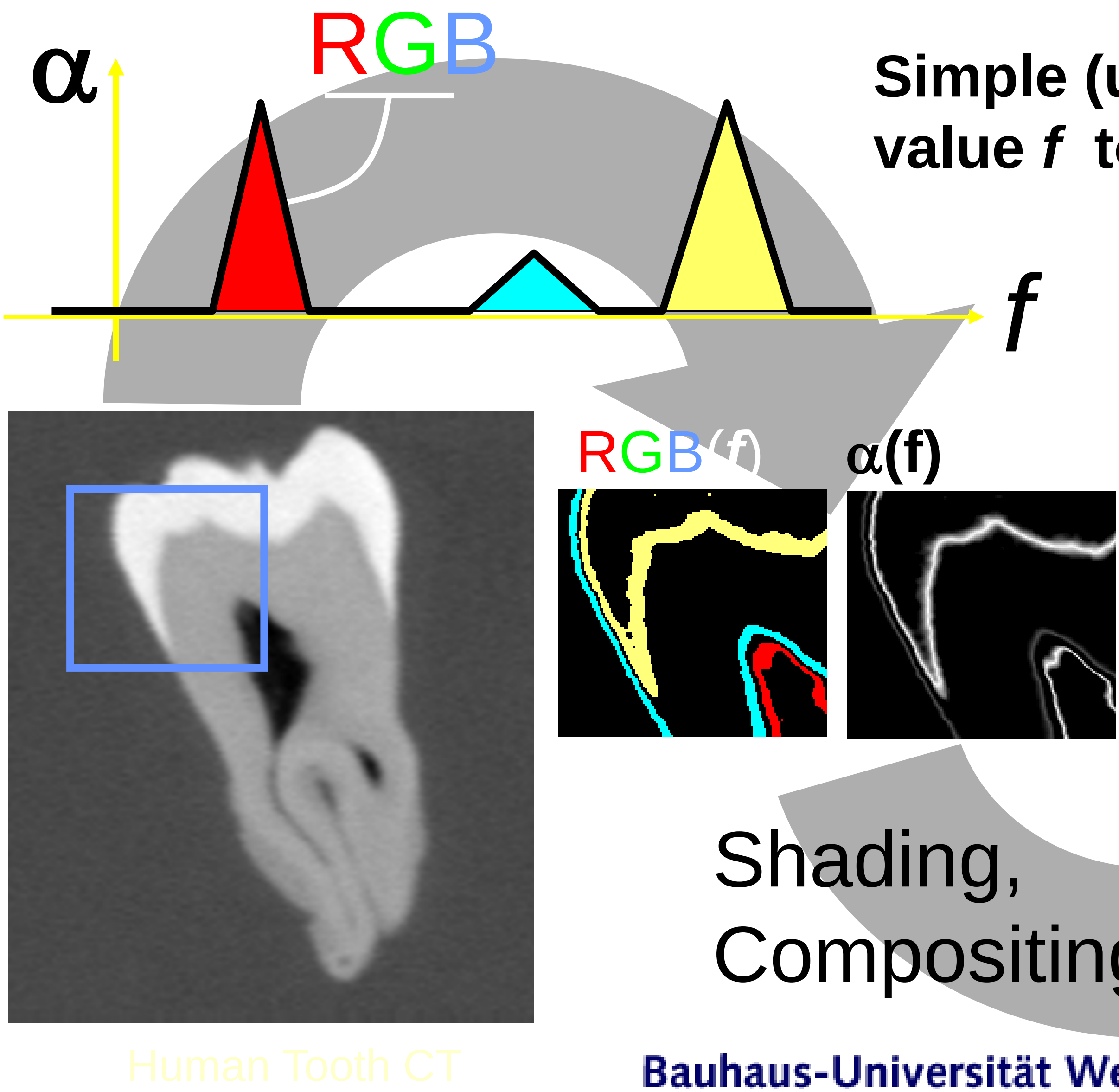


Common Basic Transfer Function

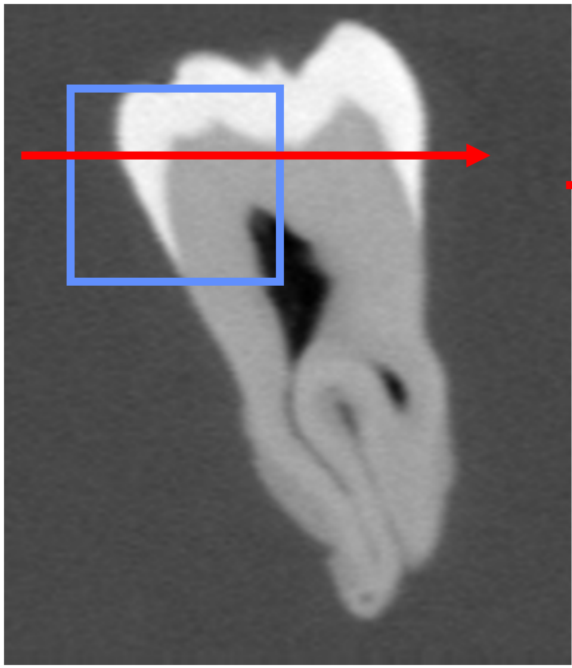


Transfer Functions (TFs)

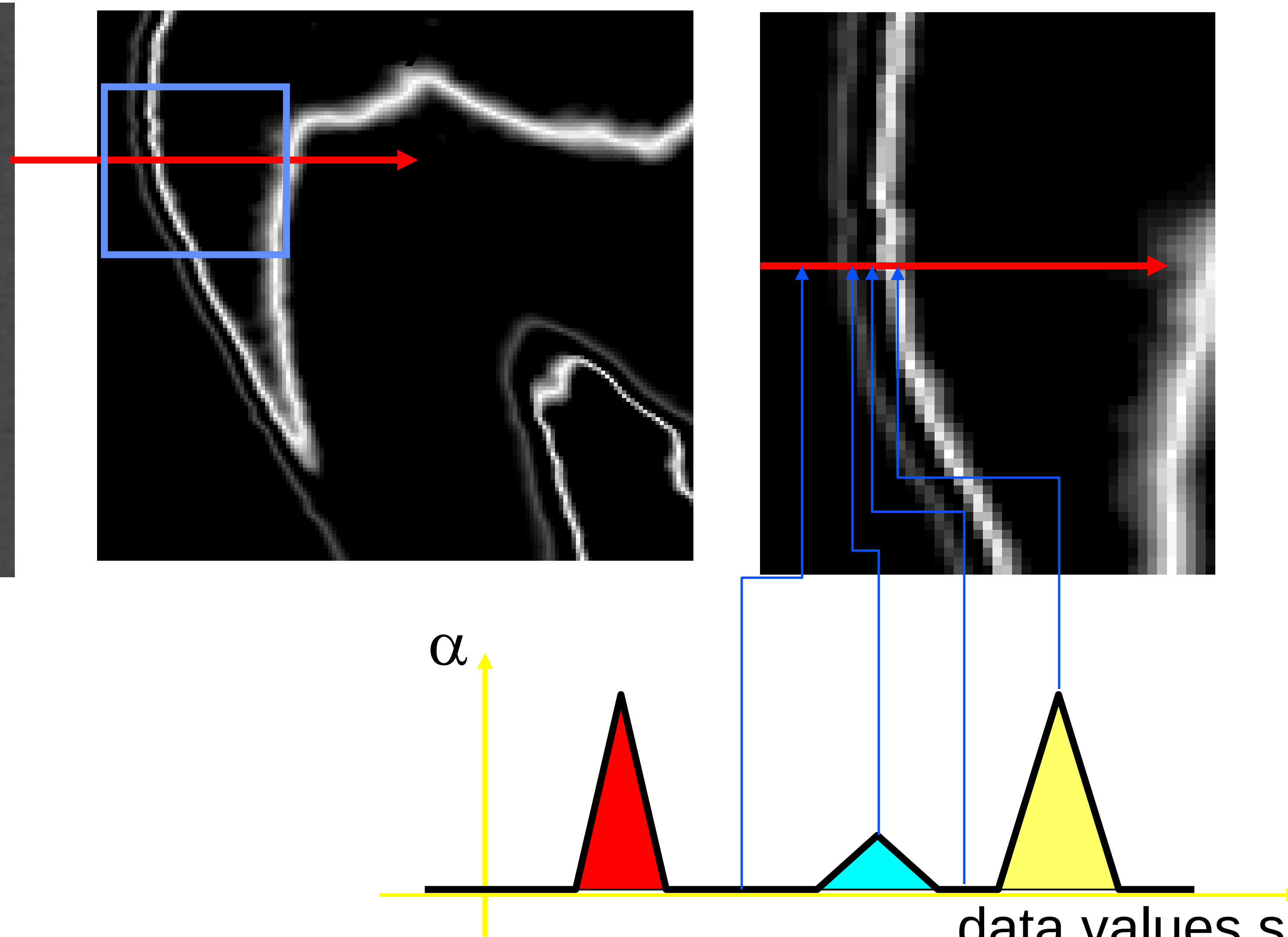
Simple (usual) case: Map data value *f* to color and opacity



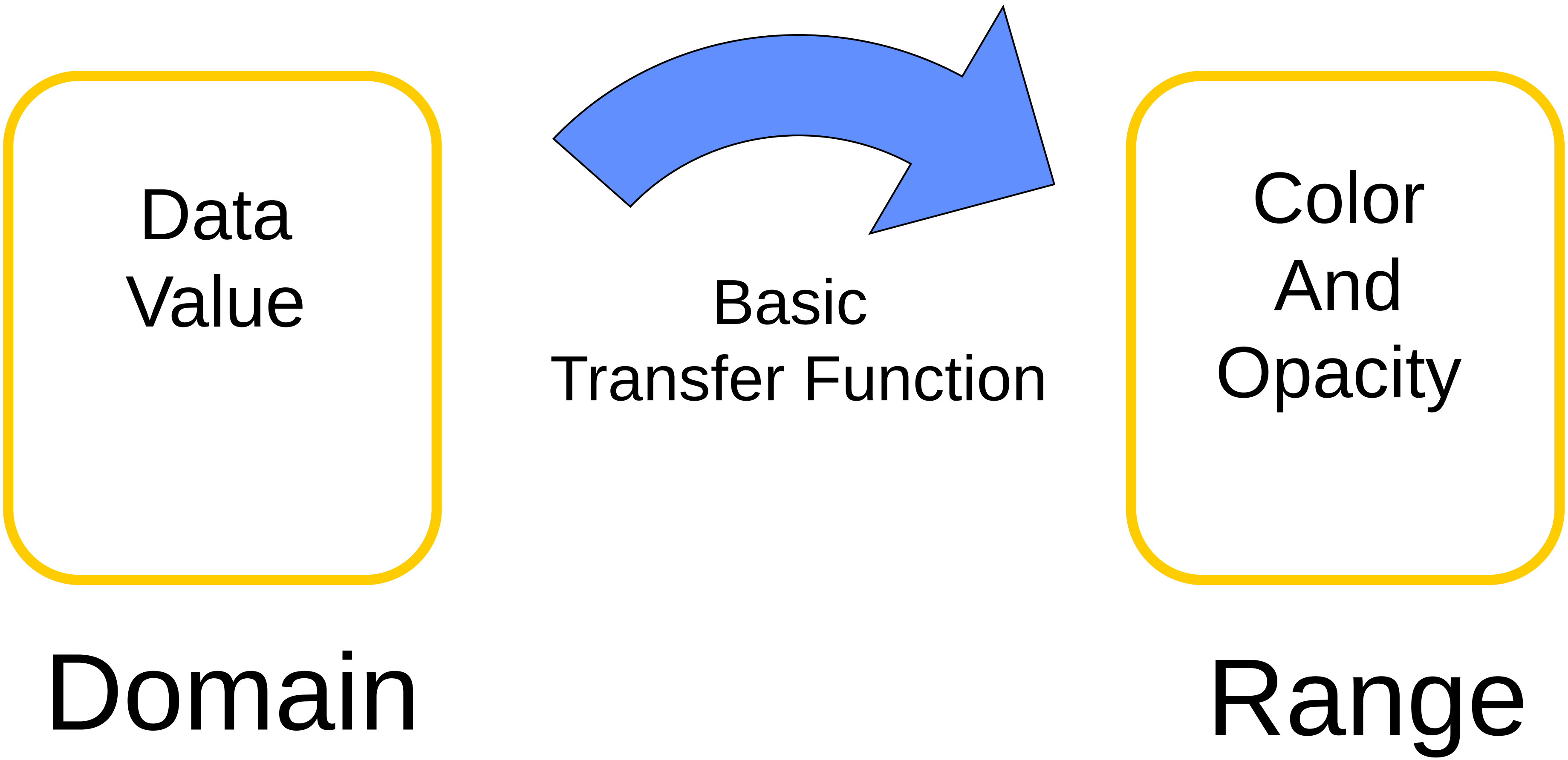
The Tooth: The Reason For the Double Surface



Human Tooth CT



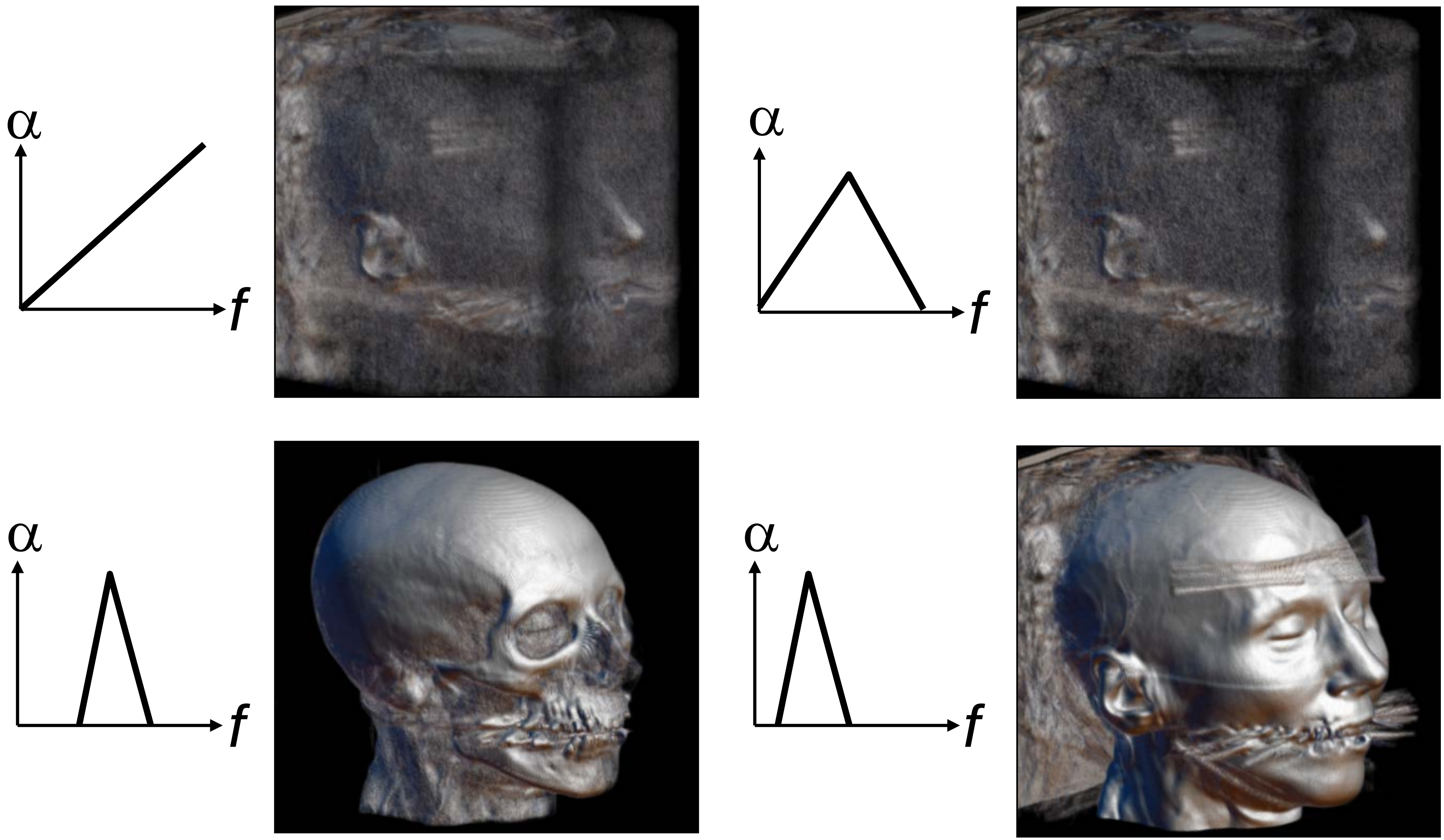
Terminology



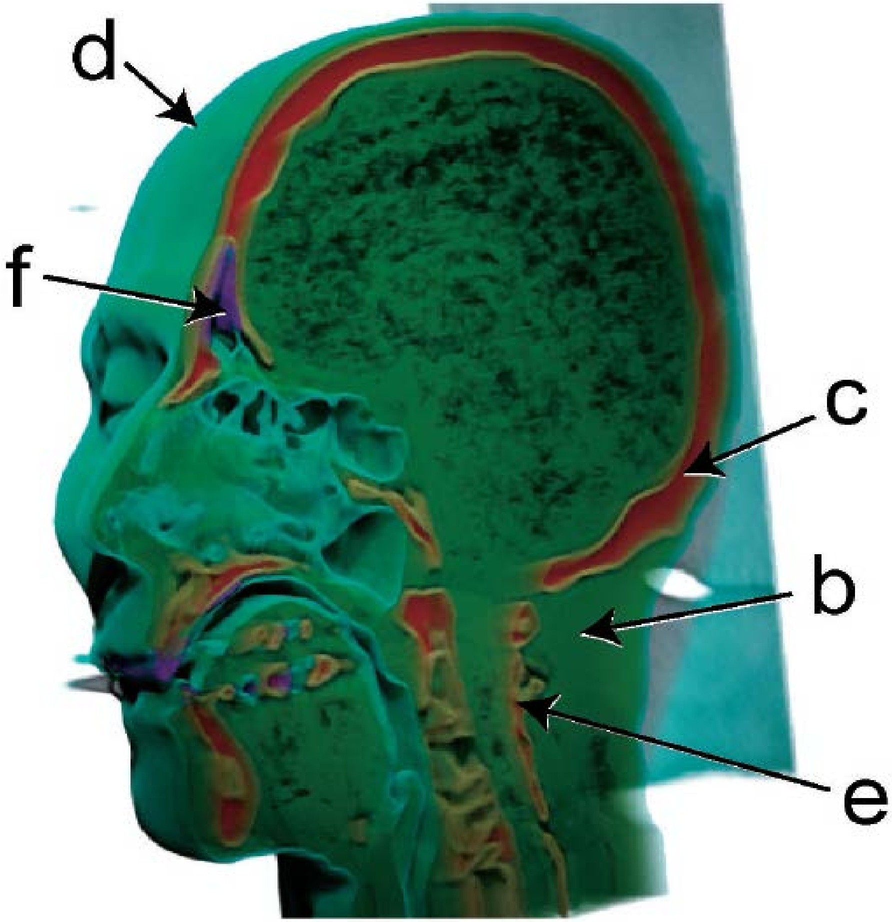
Transfer Functions

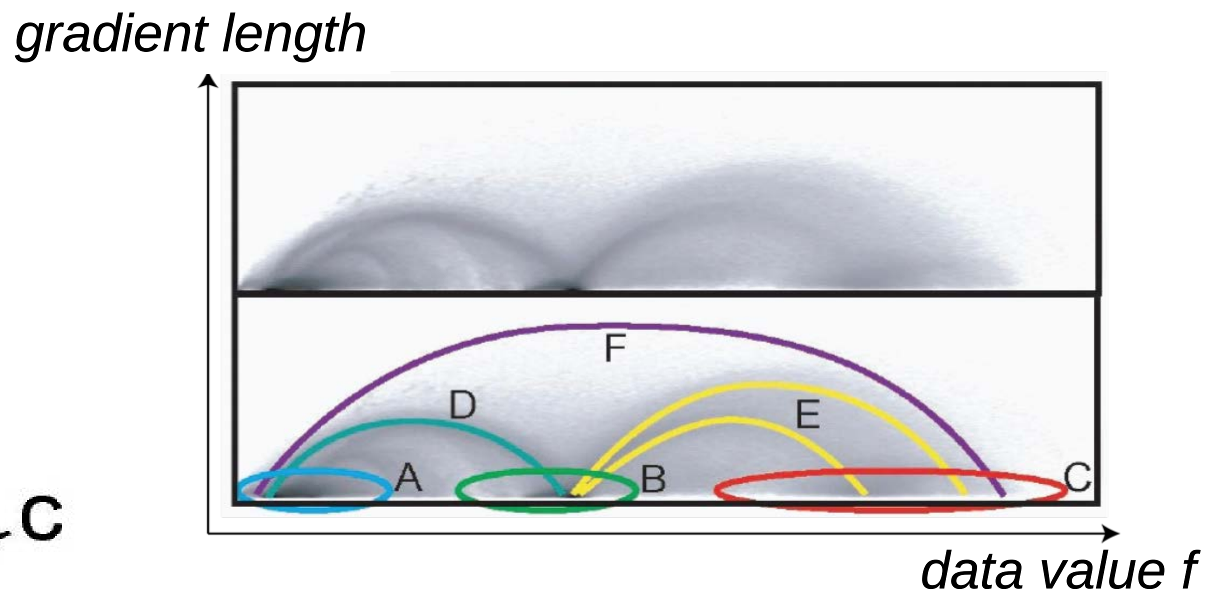
- Transfer function needs to map raw data values into presentable entities: color, intensity, opacity..
  - Raw-data $\Rightarrow$ material (R, G, B, a) or e.g. Phong parameters for lighting (k_a, k_d, k_s)
  - Material $\Rightarrow$ shaded material.
- Opacity functions are most important to look inside the volume and reveal structures you are interested in
- Color
  - Can help to distinguish features
- High gradient in the data values detects a surface and is used as a measure of its orientation respectively normal.

Some Examples: The same dataset!!



Two-Dimensional Transfer Functions

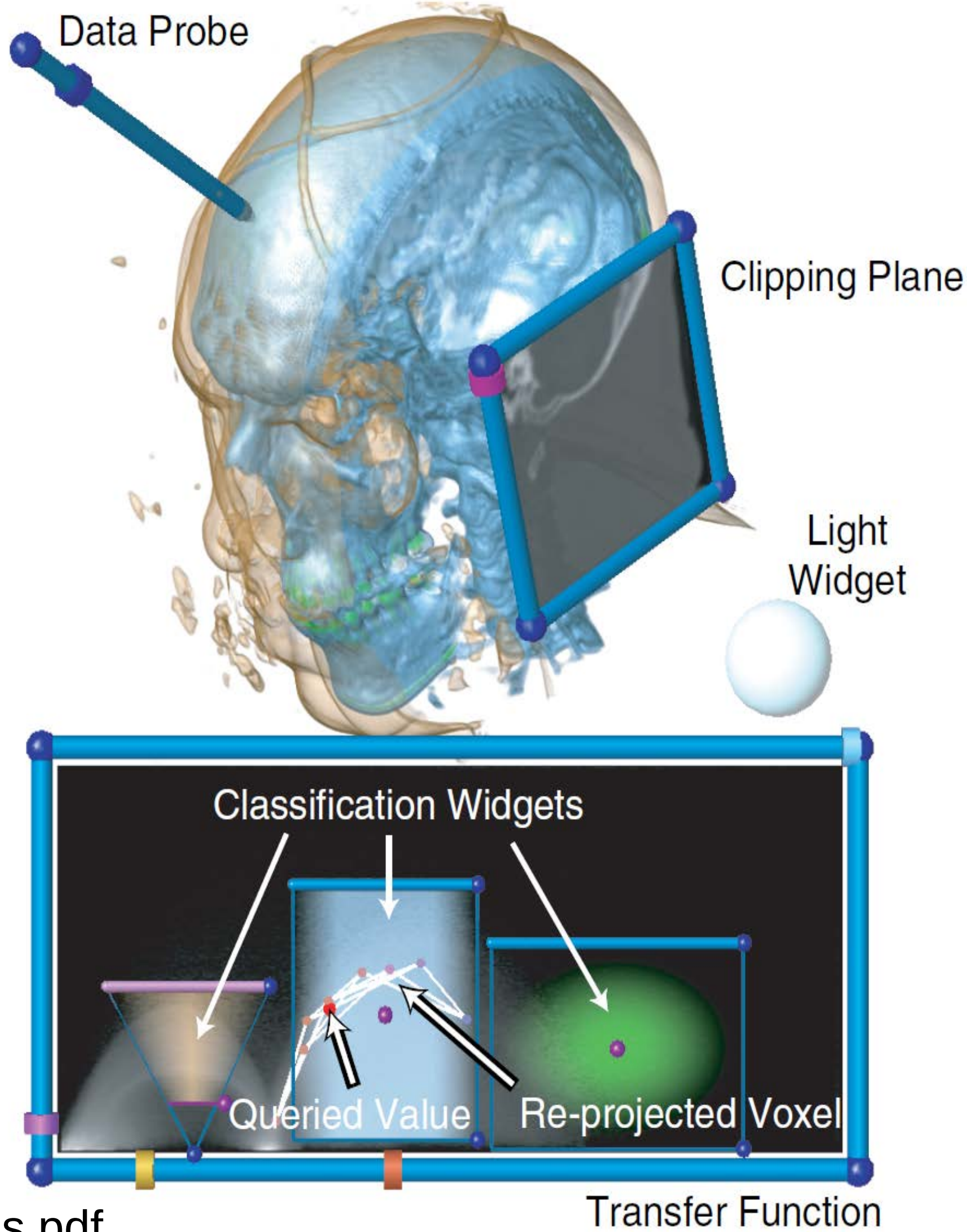




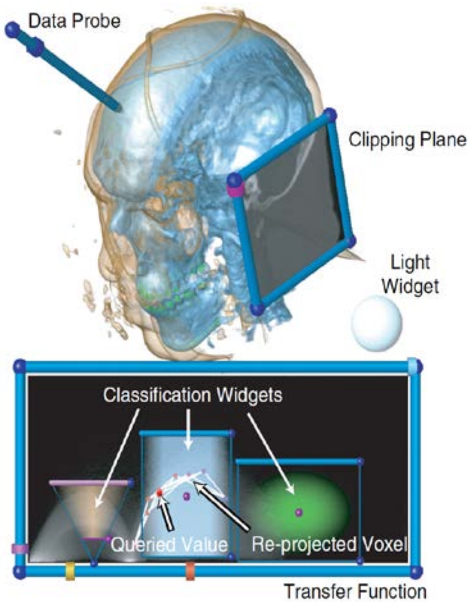
- Top: Histogram of all voxels of a volume represented as 2D points of data value and gradient length
- Bottom: Assign color and opacity to areas in the histogram

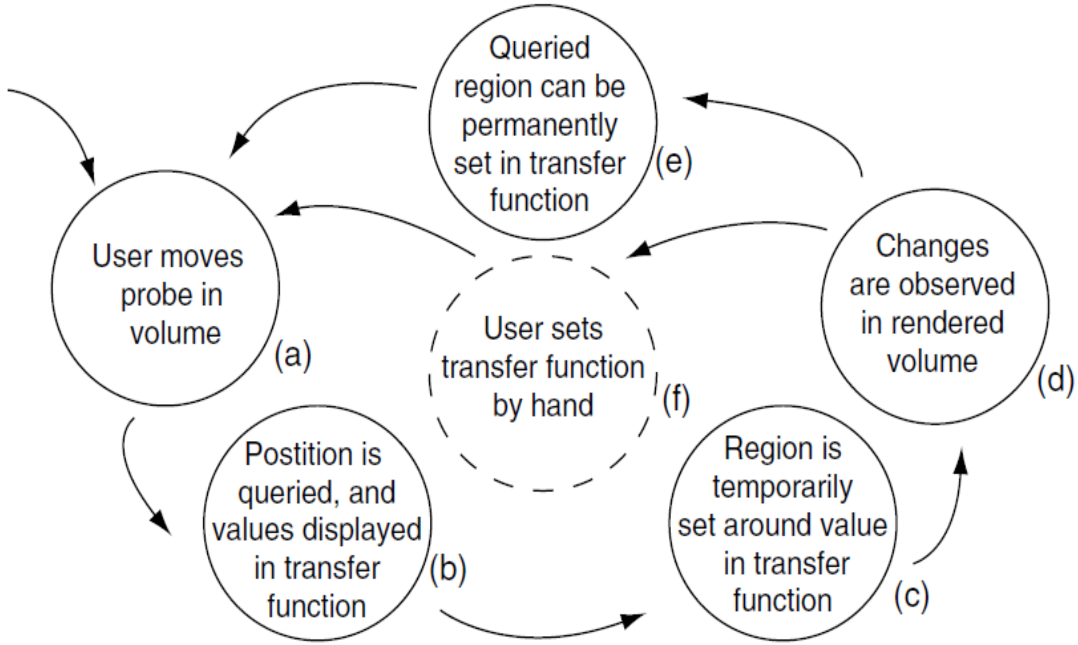
Dual Domain Interaction

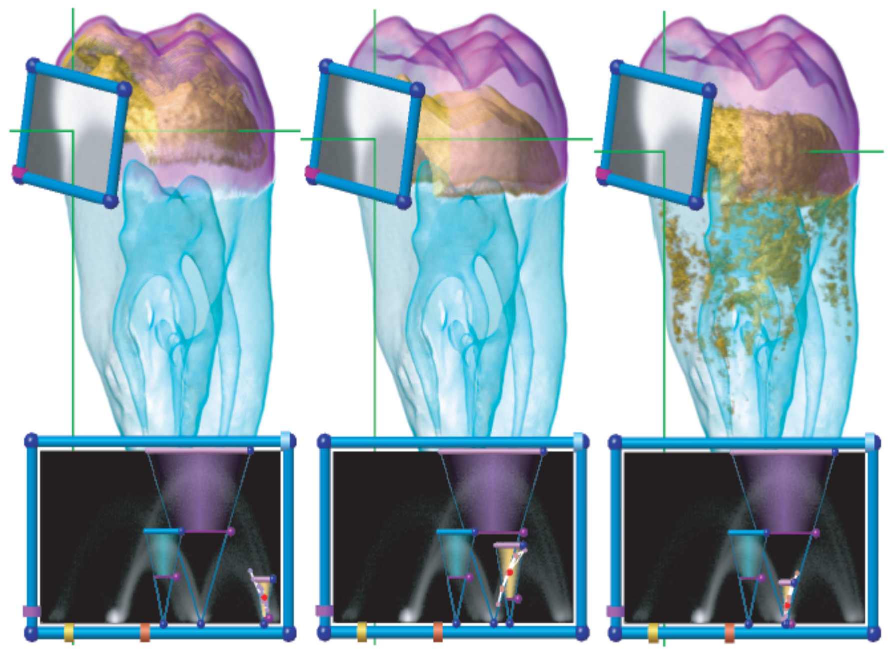
- Spatial domain
  - A single voxel corresponds to a single point in the transfer function domain
- Transfer function domain
  - A single point corresponds to one or many points in the spatial domain



http://www.cs.utah.edu/~jmk/papers/ch09_kniss.pdf

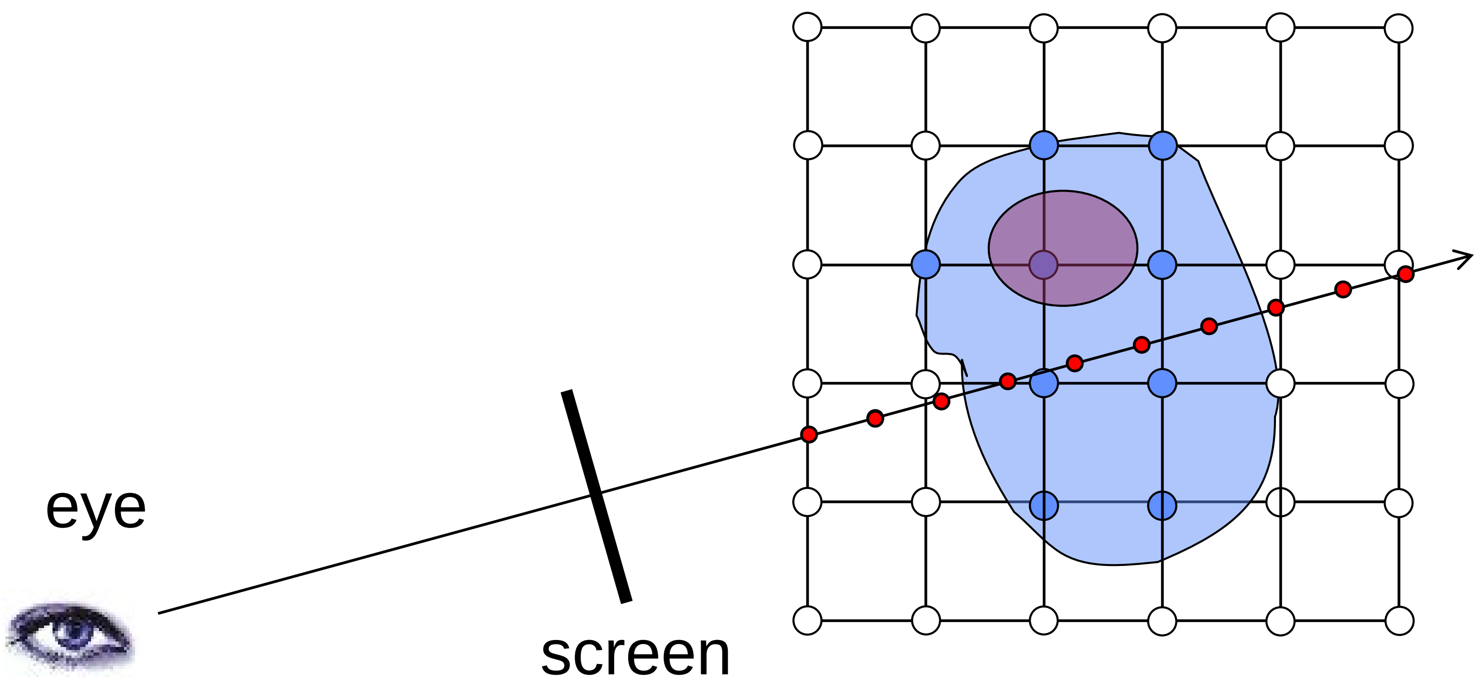

Data Probe
Clipping Plane
Light Widget
Classification Widgets
Queried Value
Re-projected Voxel
Transfer Function


User moves probe in volume
(a)
Postition is queried, and values displayed in transfer function
(b)
Region is temporarily set around value in transfer function
(c)
Changes are observed in rendered volume
(d)
Queried region can be permanently set in transfer function
(e)
User sets transfer function by hand
(f)

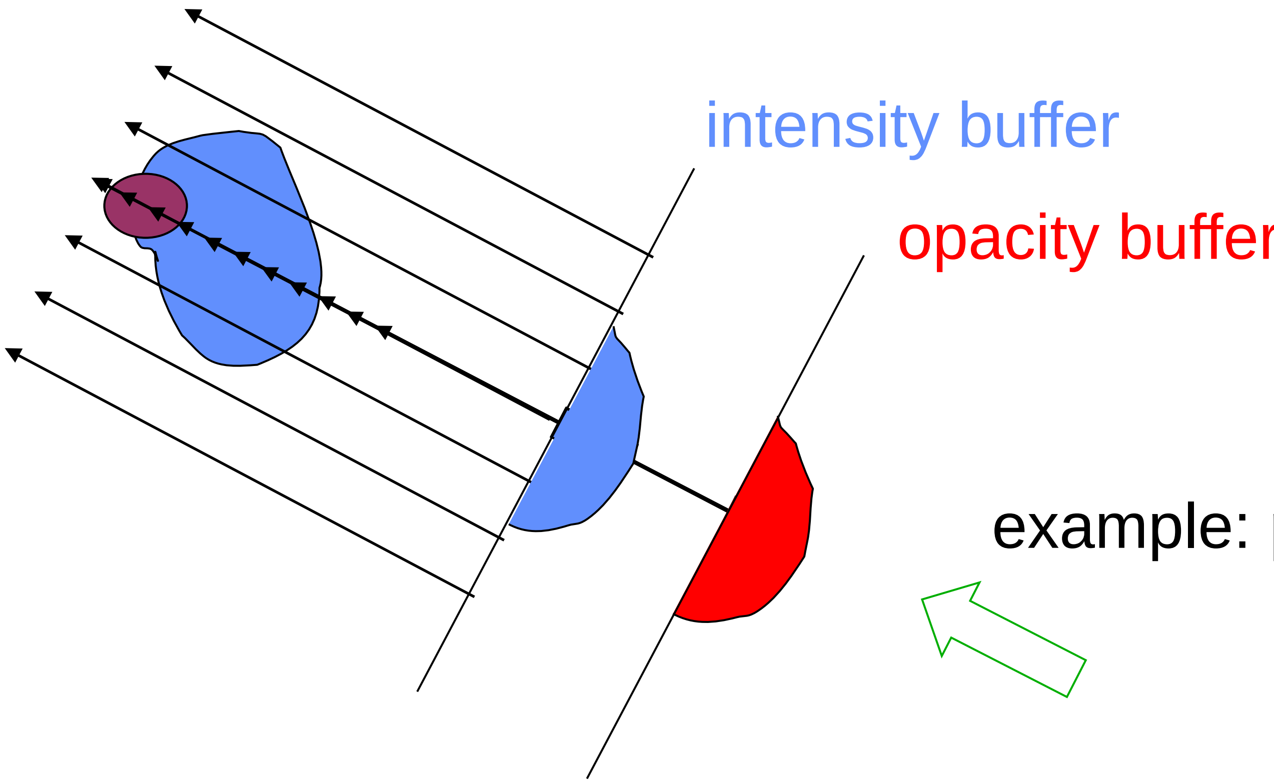


Volume Ray Casting

- Integrate light/color contributions along rays through the volume to compute pixel value



Volumetric Ray Integration



Compositing: Front-to-back Traversal

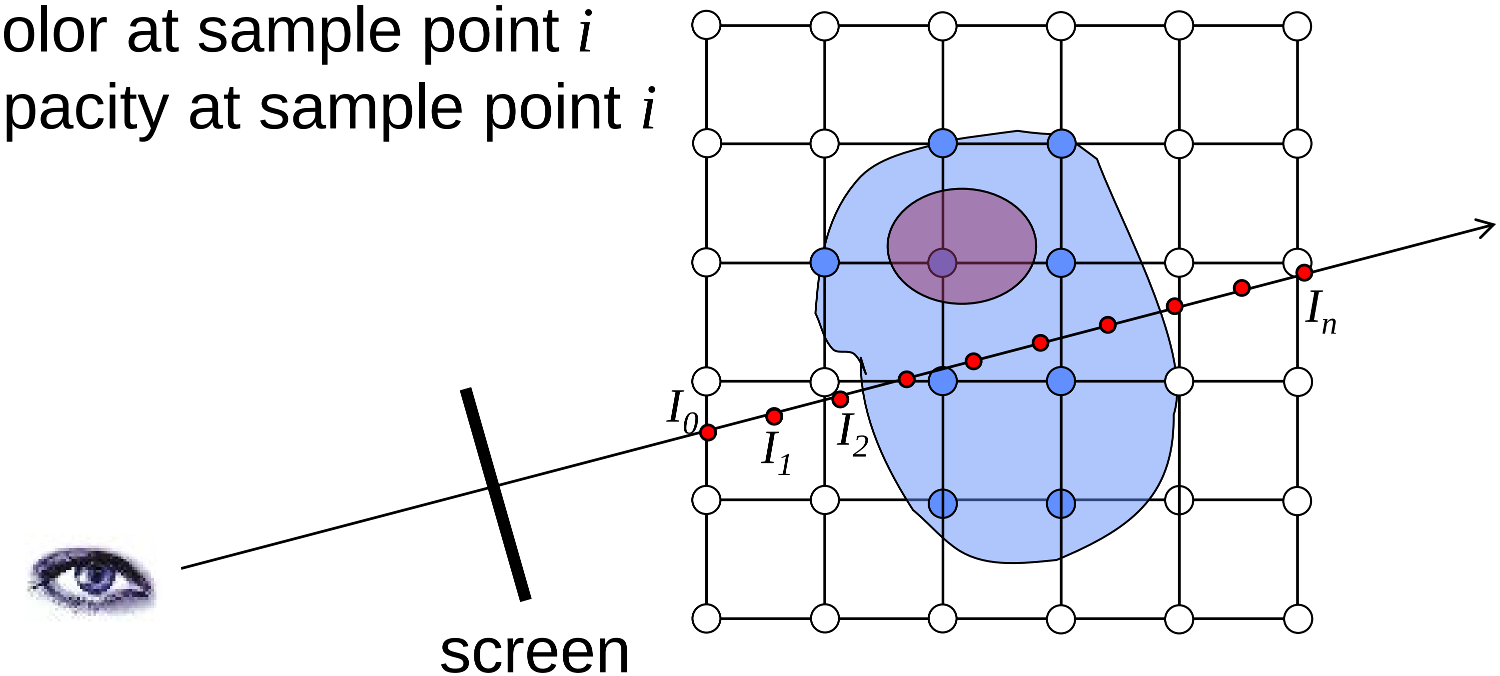
- Total intensity accumulated on a single ray

$$\sum_{i=0}^{n} I_i \prod_{j=0}^{i-1} (1-\alpha_j) = I_0 + I_1(1-\alpha_0) + I_2(1-\alpha_0)(1-\alpha_1) + \ldots$$

$I_i = C_i\ \alpha_i$ is intensity of sample point i

C_i is color at sample point i

α_i is opacity at sample point i



Compositing: Front-to-back Traversal

With transparency $T_i = (1 - \alpha_j)$

$$\sum_{i=0}^{n} I_i \prod_{j=0}^{i-1} T_j = I_0 + I_1 T_0 + I_2 T_0 T_1 + \ldots$$

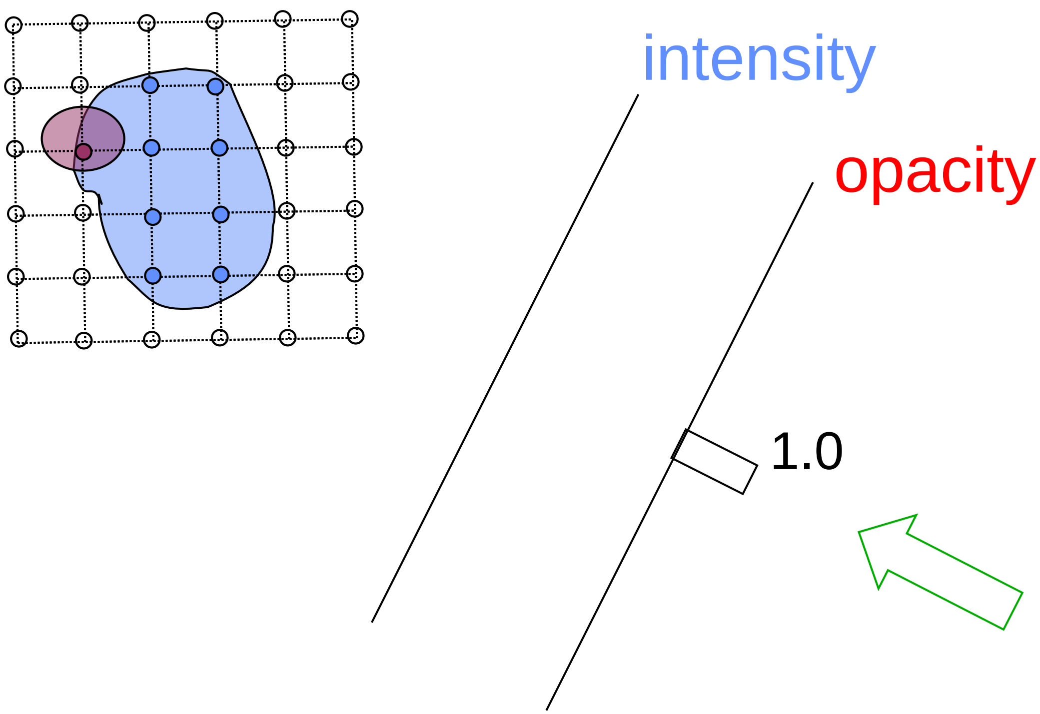
Intensity contribution of point *i* to corresponding pixel is I_i multiplied by product of all transparencies so far – which is the remaining transparency at location i

Front-to-back algorithm

```
trans = 1.0; inten = I[0];
for (i=1; i <= n; ++i) {
      trans = trans * T[i-1];
      inten = inten + trans * I[i];
}
```

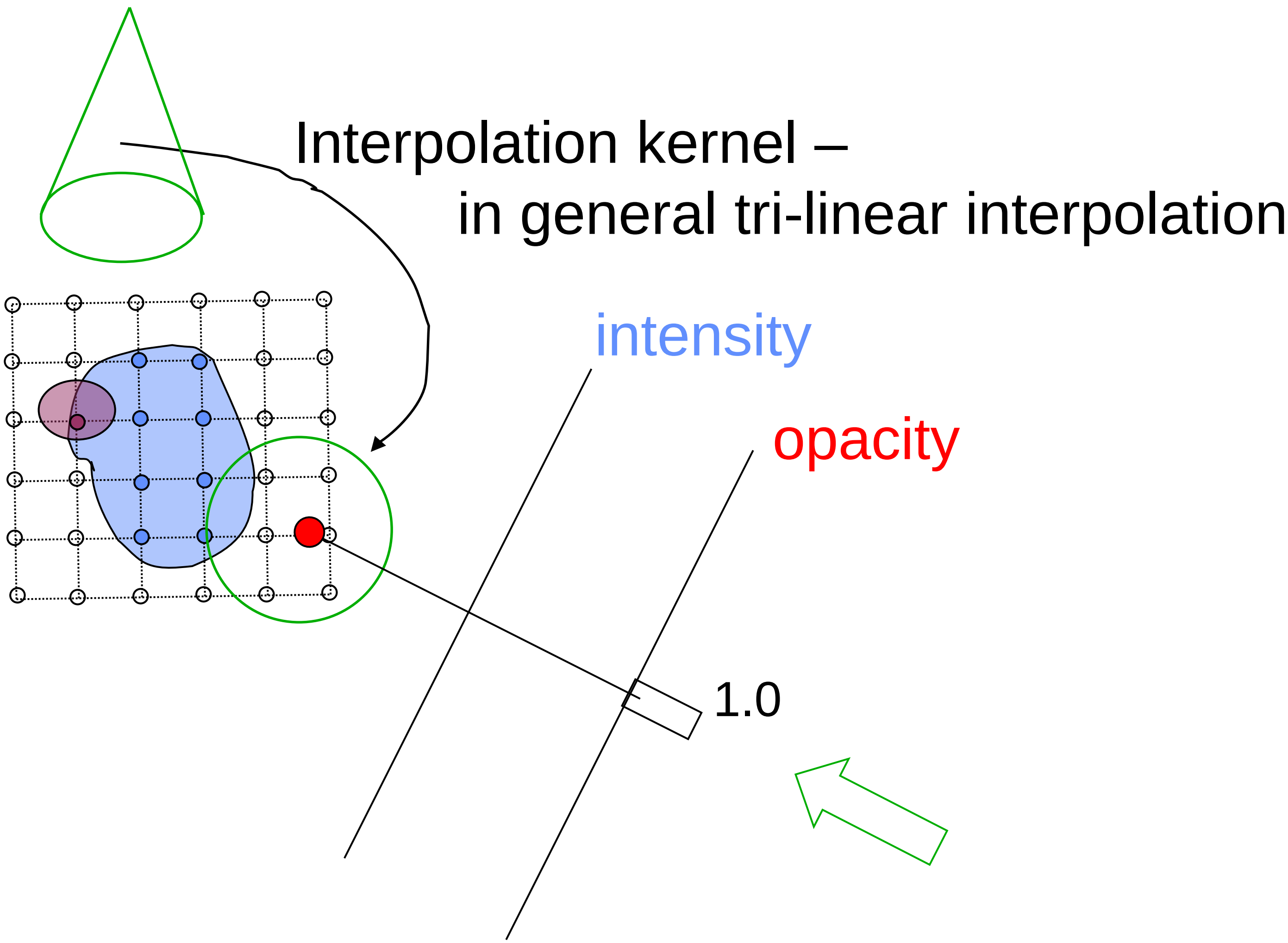
Early ray termination: terminate when `trans` approaches 0.0!

Ray Casting: Volumetric Compositing



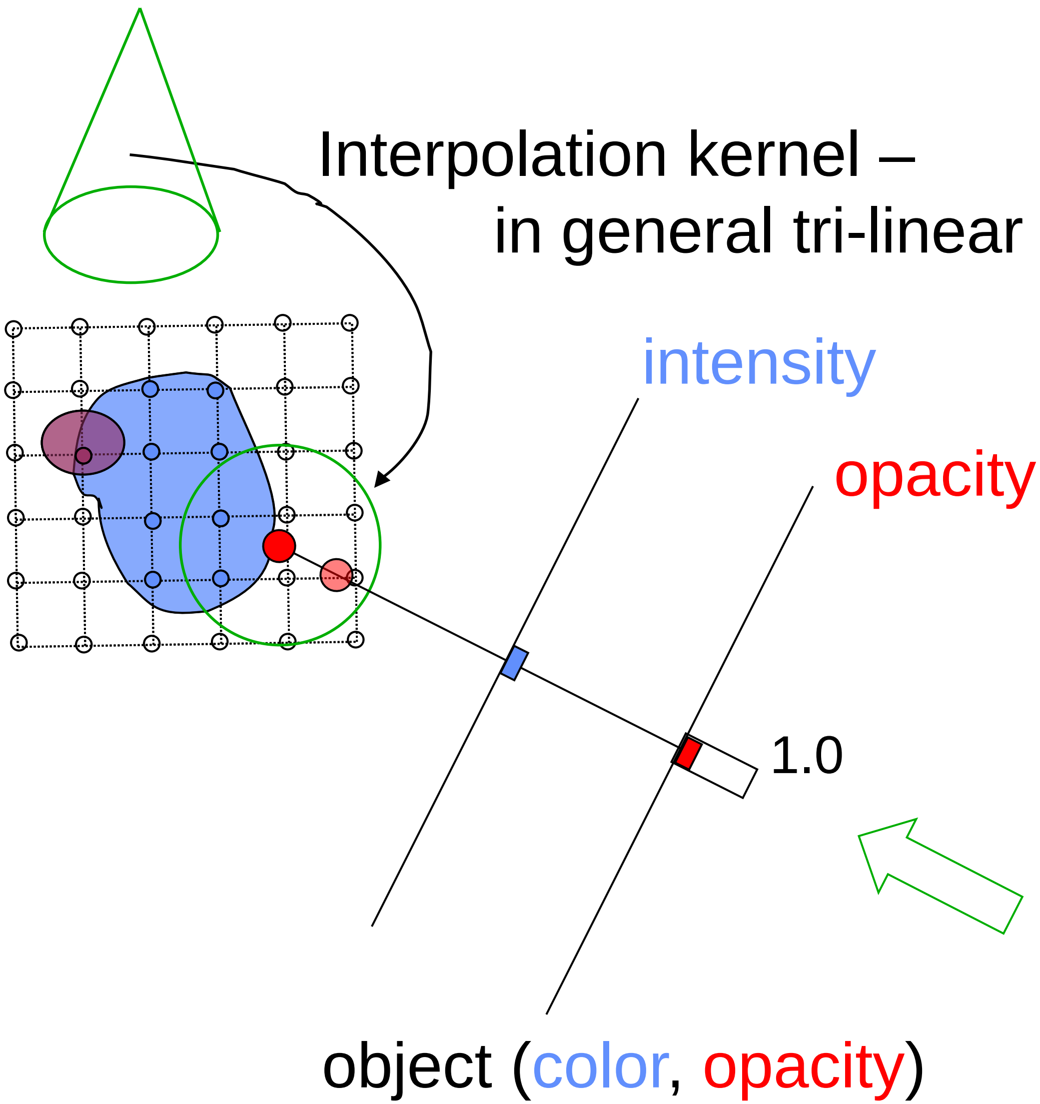
object (color, opacity)

Ray Casting: Volumetric Compositing

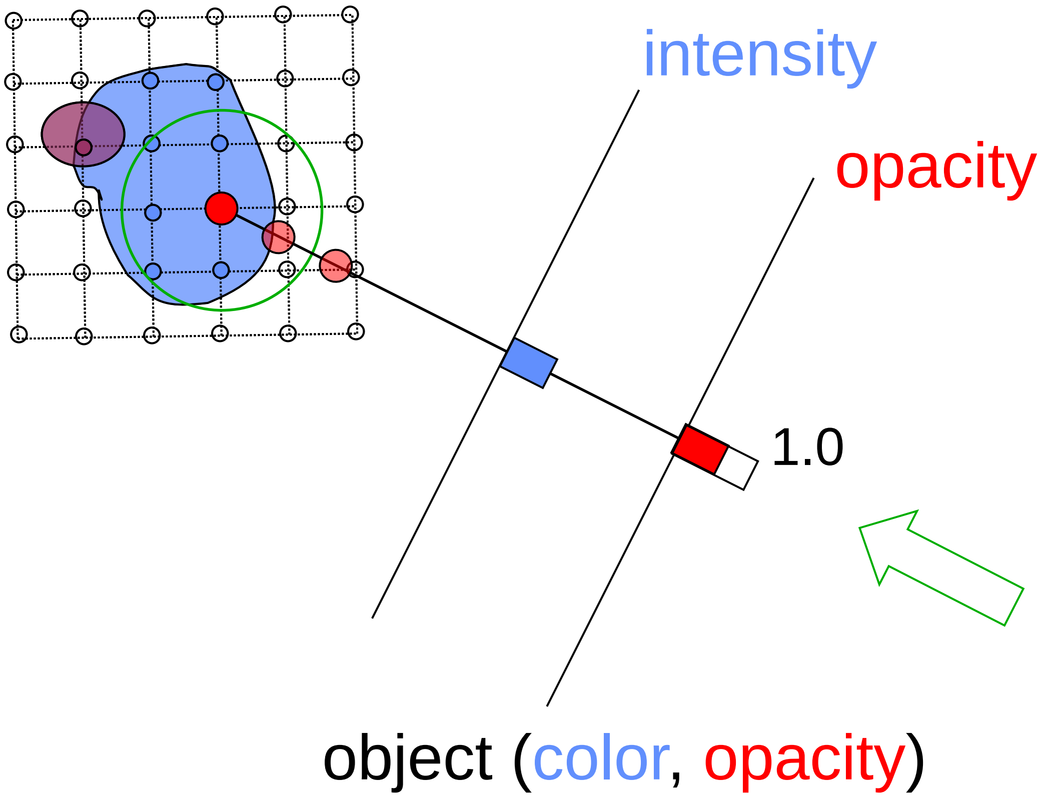


object (color, opacity)

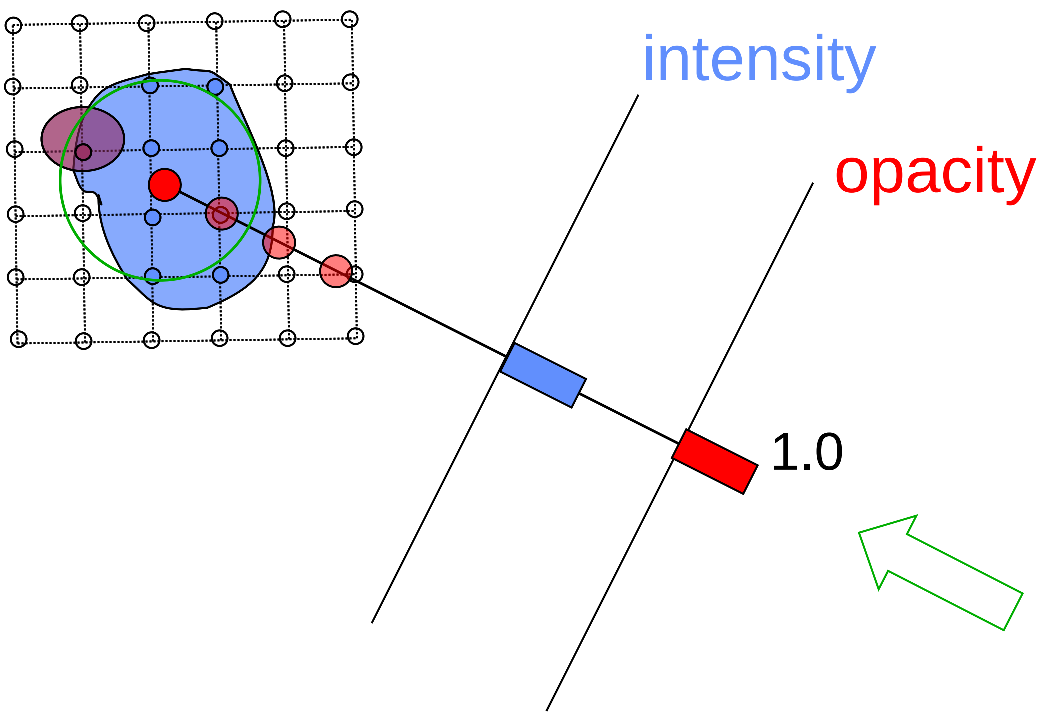
Ray Casting: Volumetric Compositing



Ray Casting: Volumetric Compositing

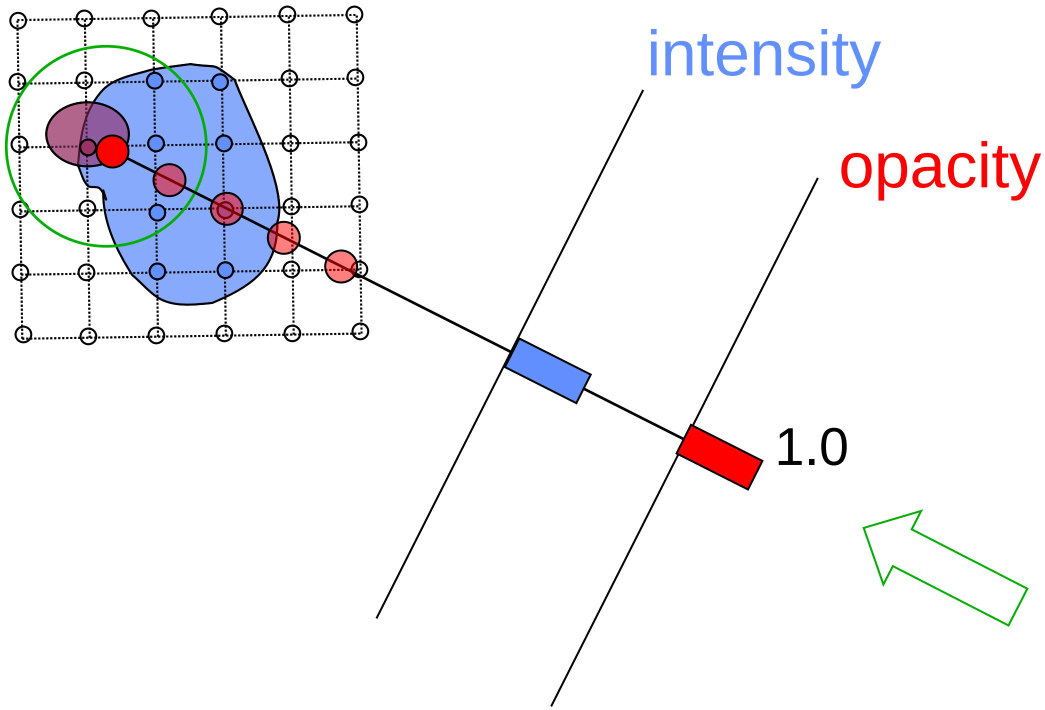


Ray Casting: Volumetric Compositing



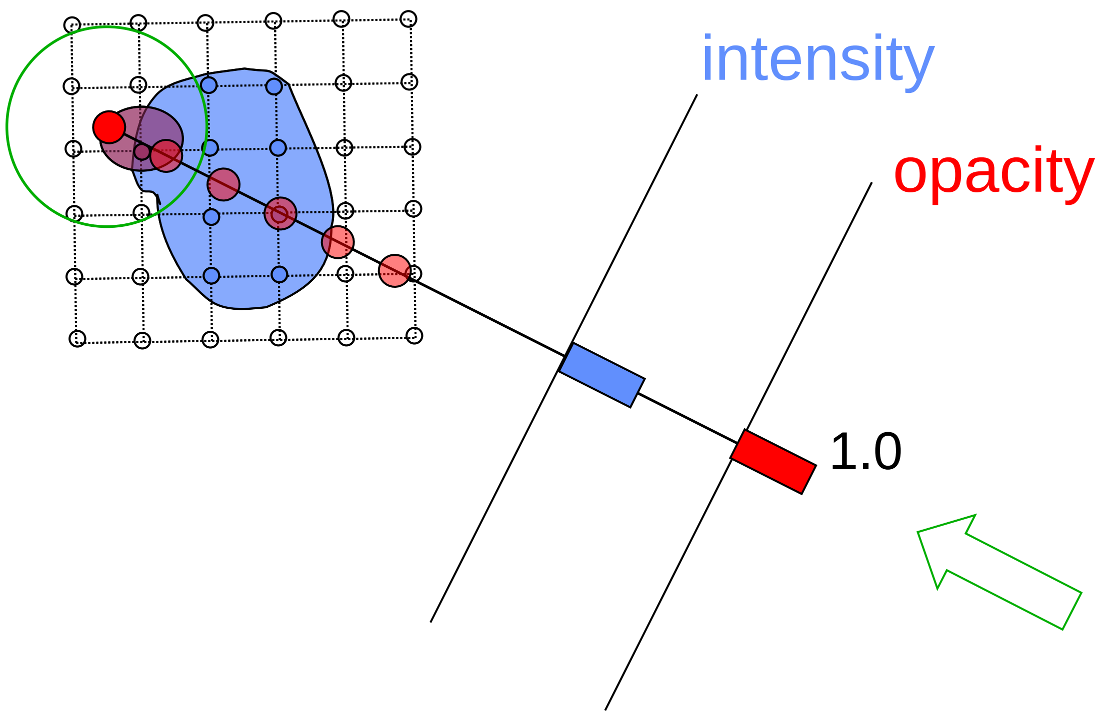
object (color, opacity)

Ray Casting: Volumetric Compositing



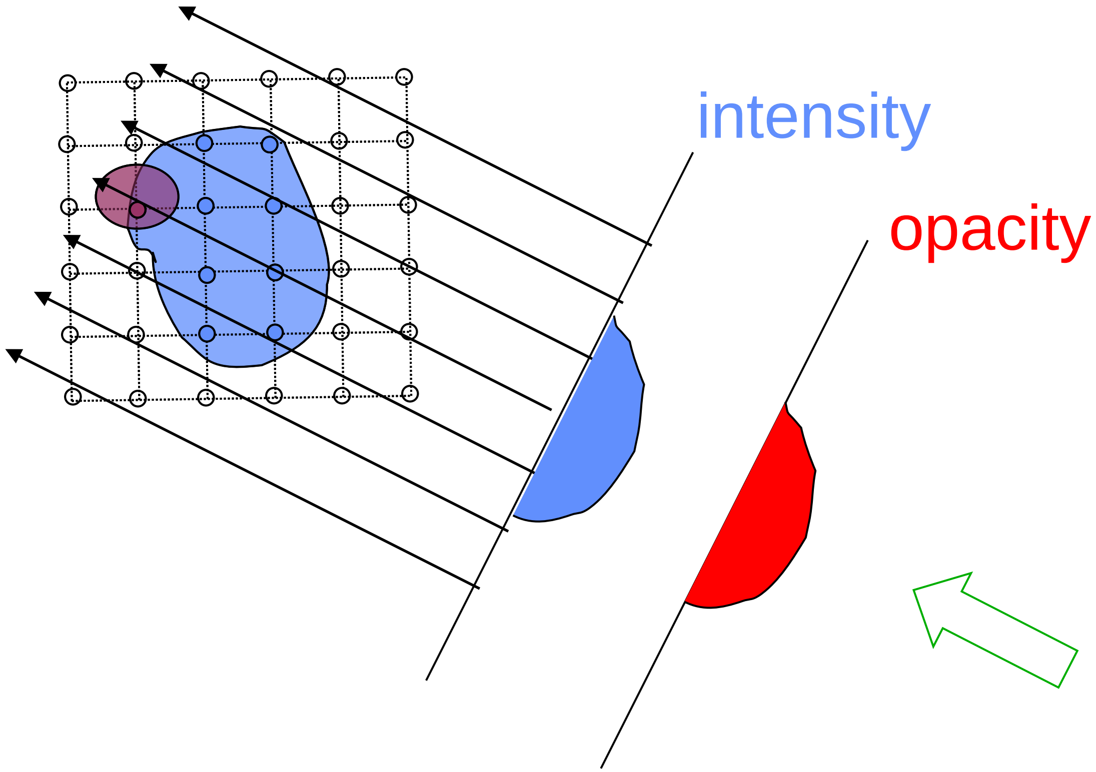
object (color, opacity)

Ray Casting: Volumetric Compositing



object (color, opacity)

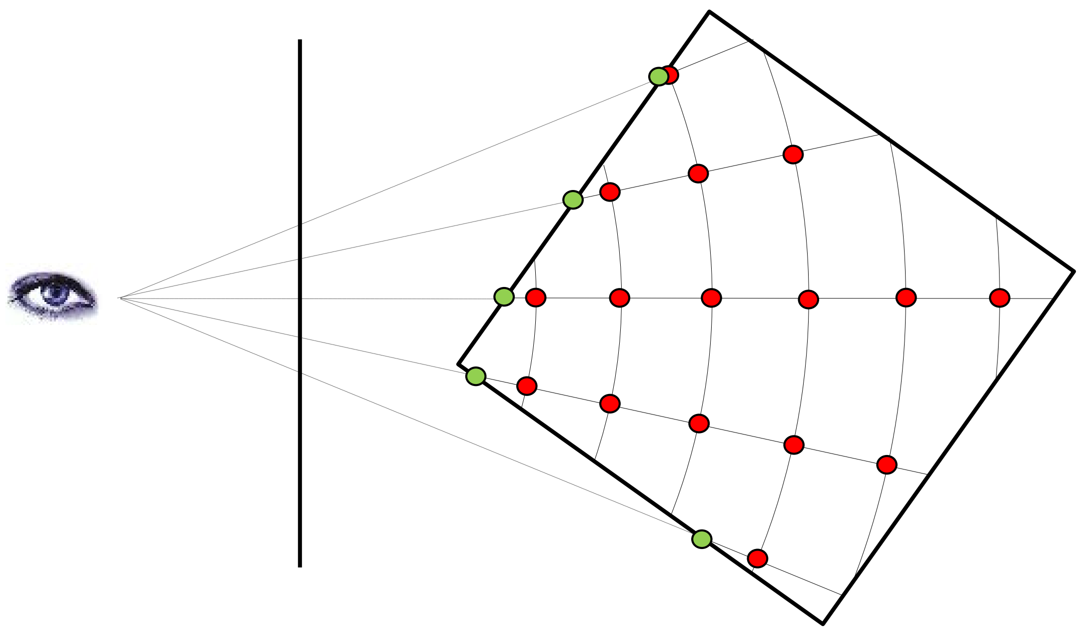
Ray Casting: Volumetric Compositing



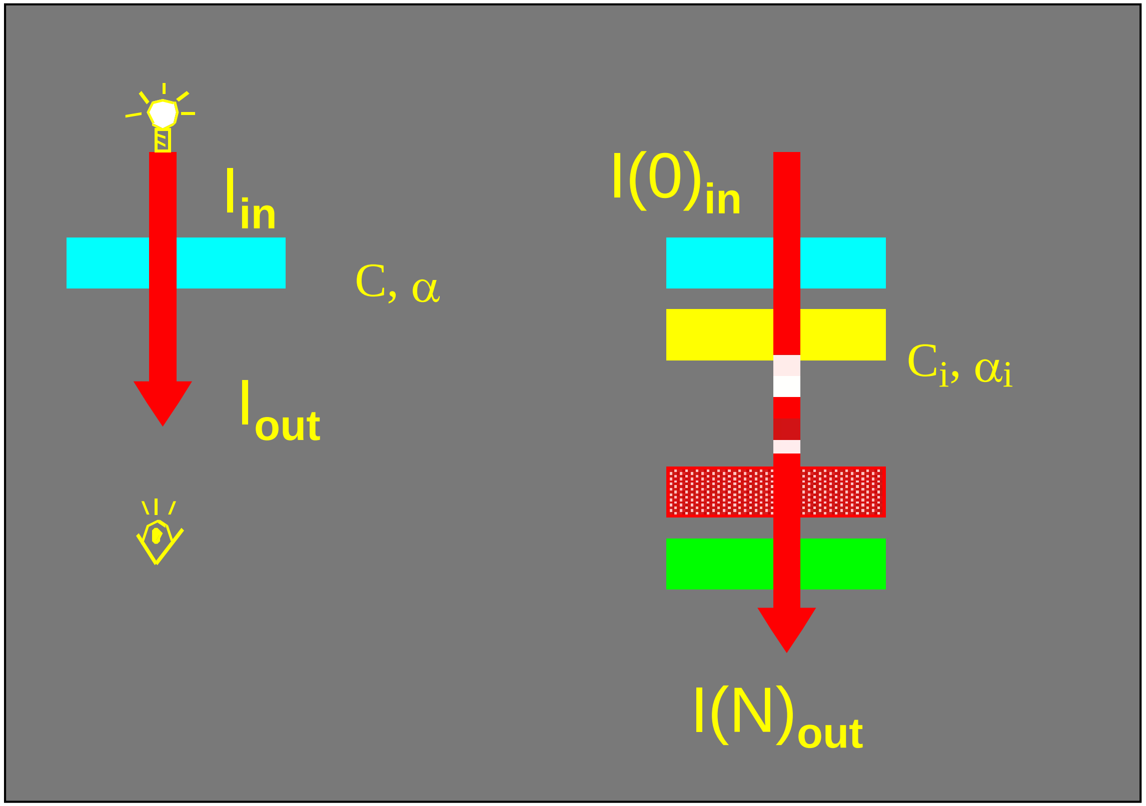
object (color, opacity)

Ray Casting: Sampling on Spherical Shells For Perspective Projection

- Creates a more consistent sampling pattern among adjacent rays and reduces aliasing



Back-to-Front Compositing



$$I_{out} = \alpha\, C + (1-\alpha)\, I_{in} \qquad I(i)_{in} = I(i-1)_{out}$$

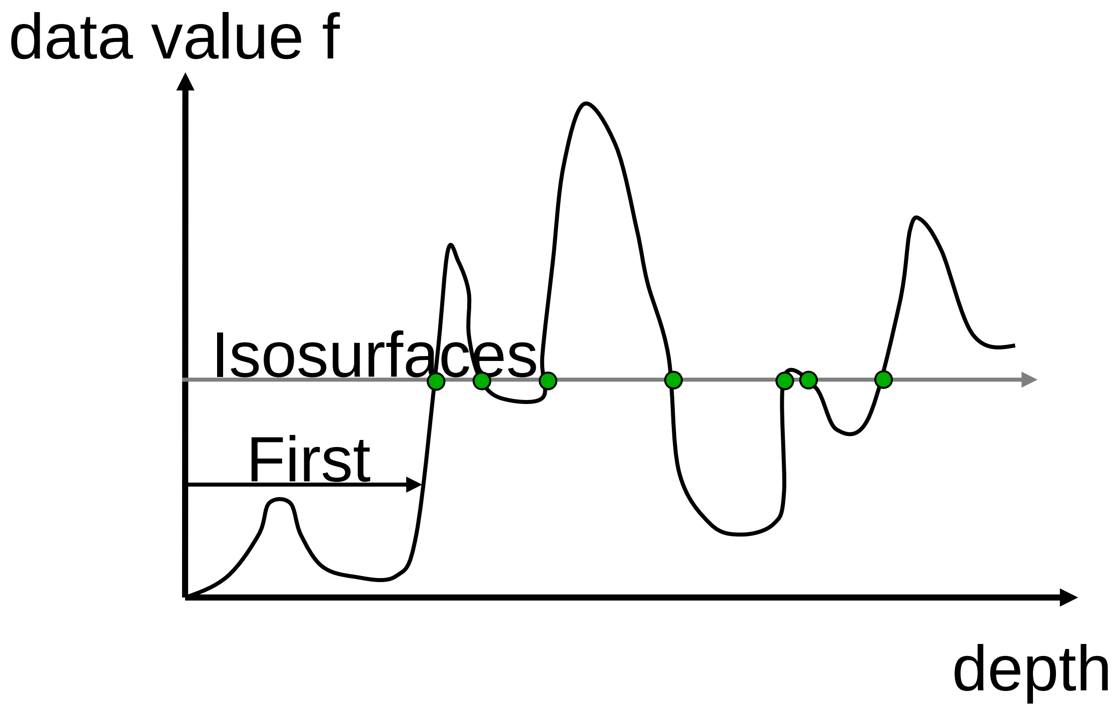
Back-to-front Traversal

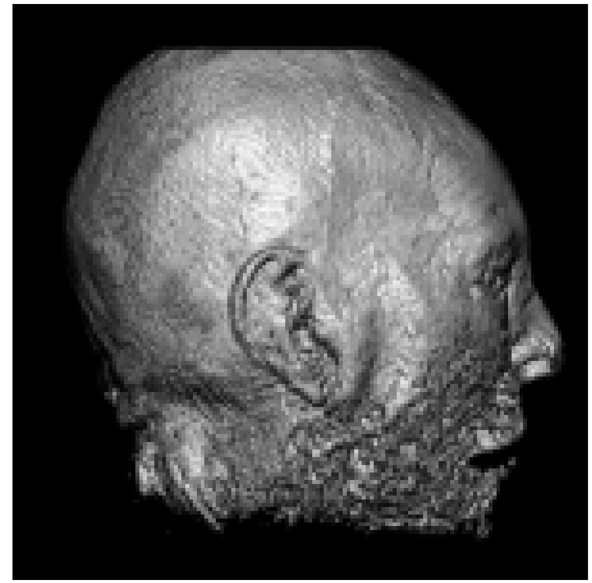
Back-to-front

```
inten = 0;
for (i=n; i >= 0; --i) {
      inten = C[i] * A[i] + inten * (1 - A[i]);
}
```

Uses opacity α_i instead of transparency $(1 - \alpha_i)$
Doesn't keep track of transparency
Doesn't allow early ray termination
Classical approach that was used for slice-based volume rendering

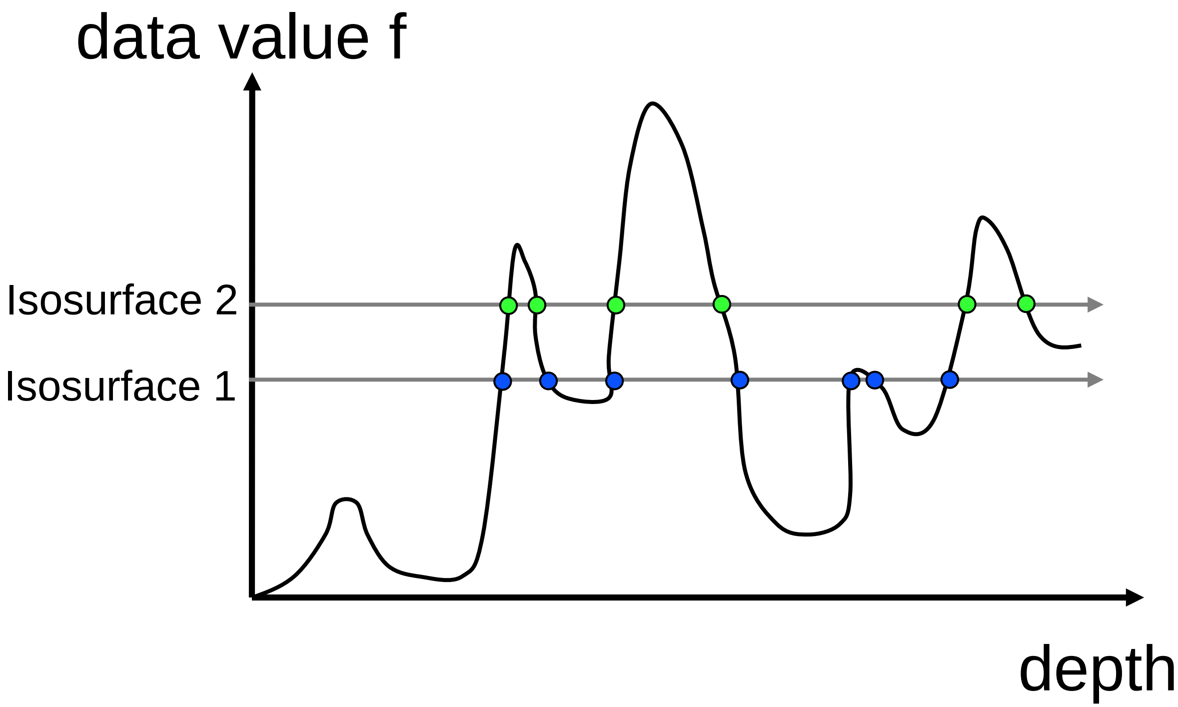
Ray Traversal
First Hit / Iso-Surfaces

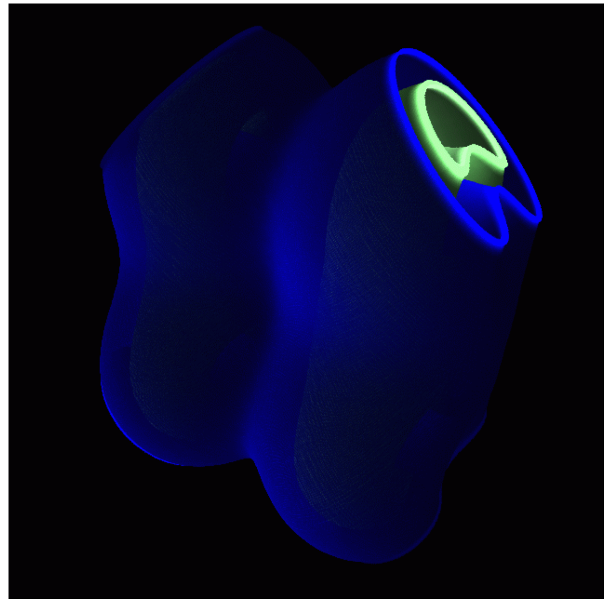




- Extracts iso-surfaces, first done by Tuy&Tuy ’84
- An iso surface is a surface which consists of points of constant data value

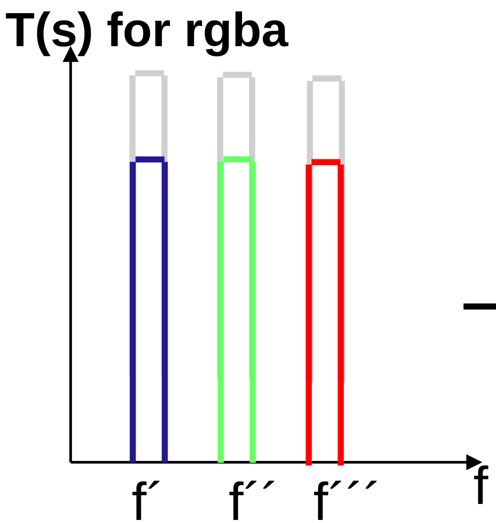
Ray Traversal
Iso-Surfaces

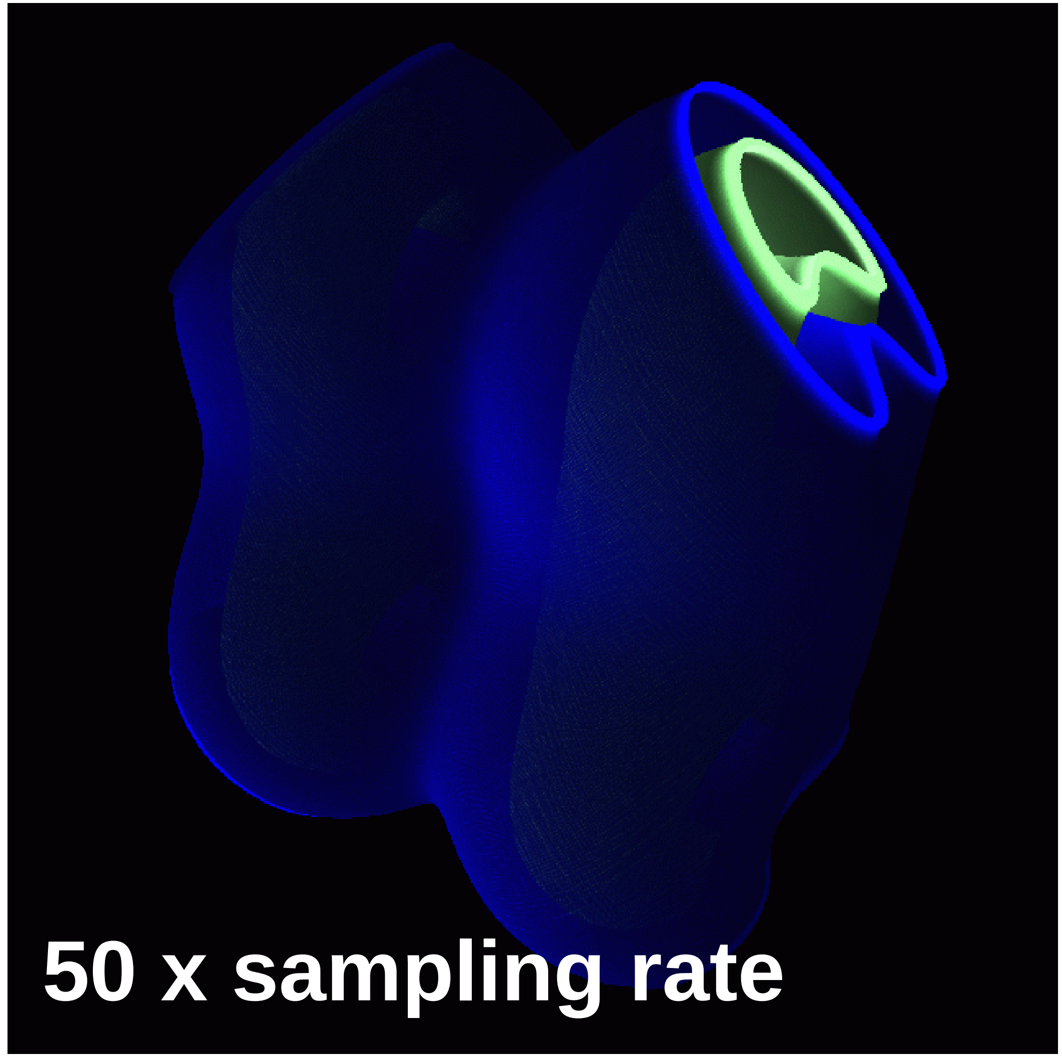




- Extracts all iso-surfaces for a single iso-value or multiple iso-values

Iso-Surface Ray Casting

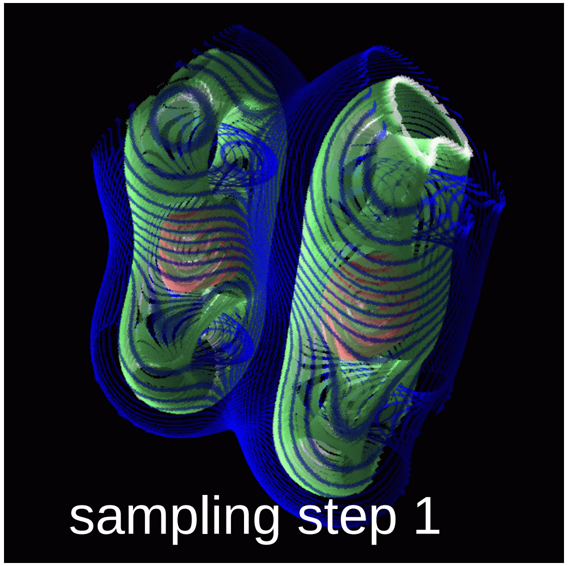


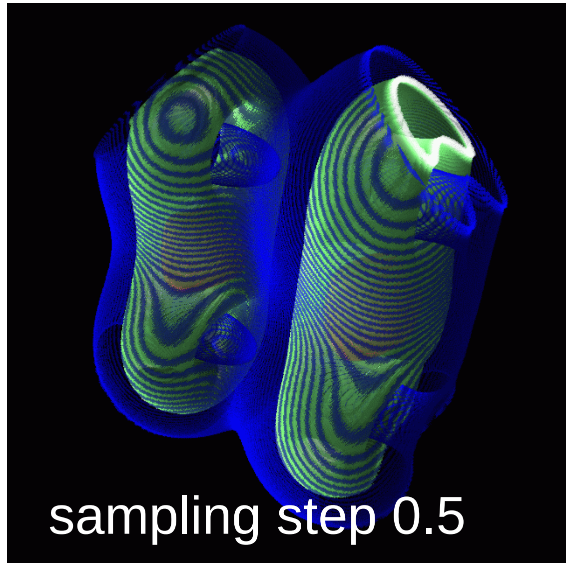


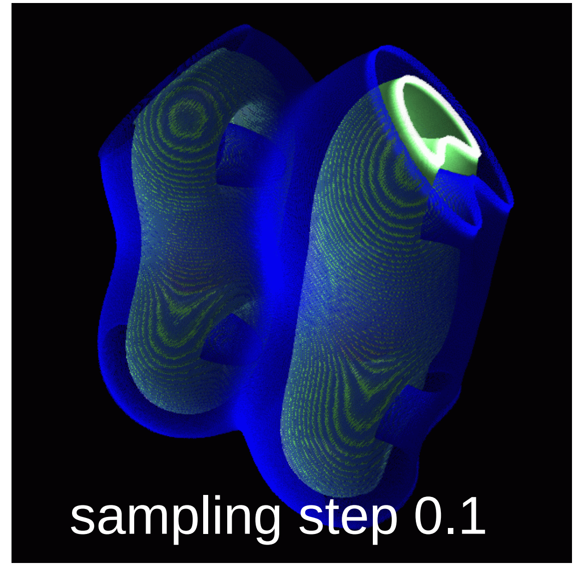
Transfer function T(f) creates iso-surfaces for f´, f´´, and f´´´ requires high sampling rate

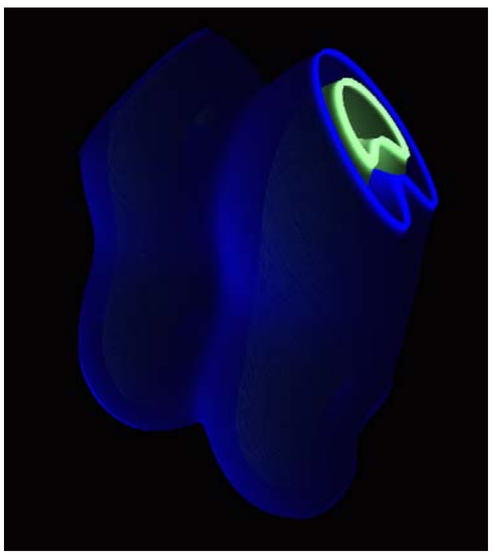
Iso-Surface Ray Casting

Influence of Sampling Rate



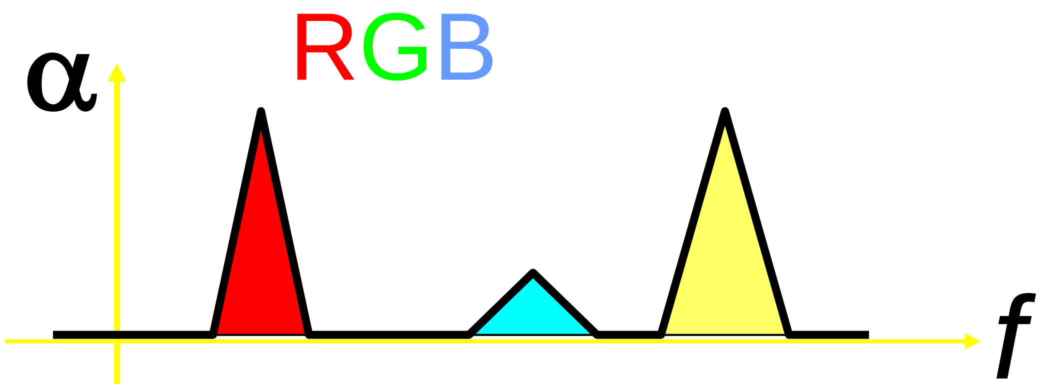






sampling step 0.02 – 50x

Iso-Surface Ray Casting

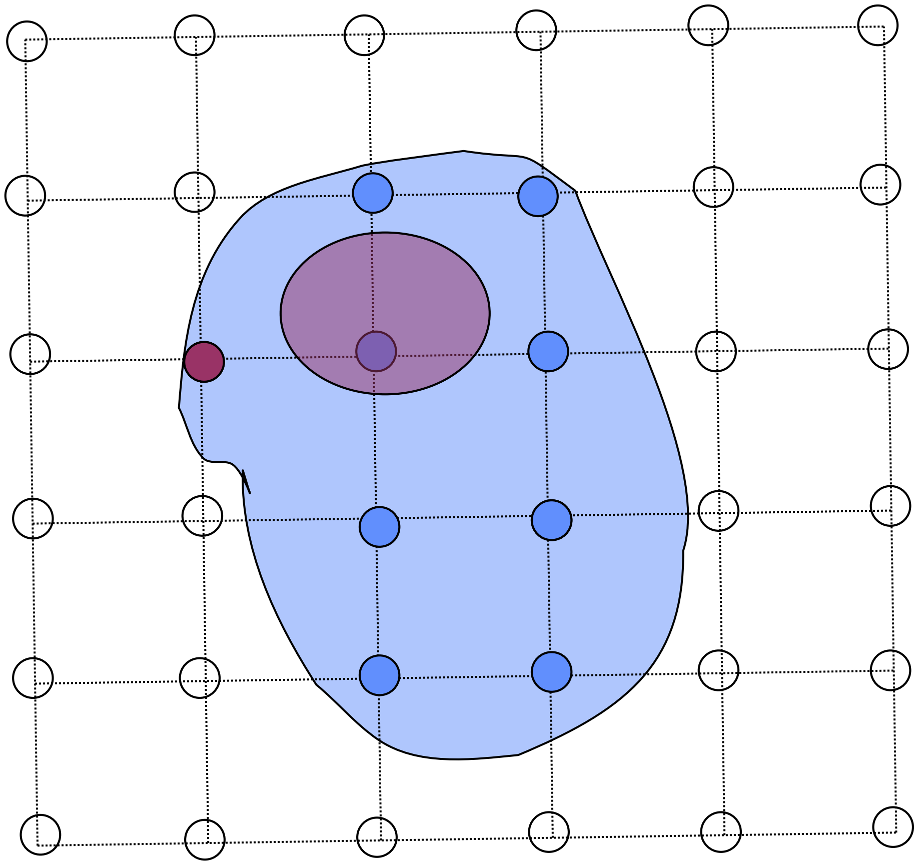


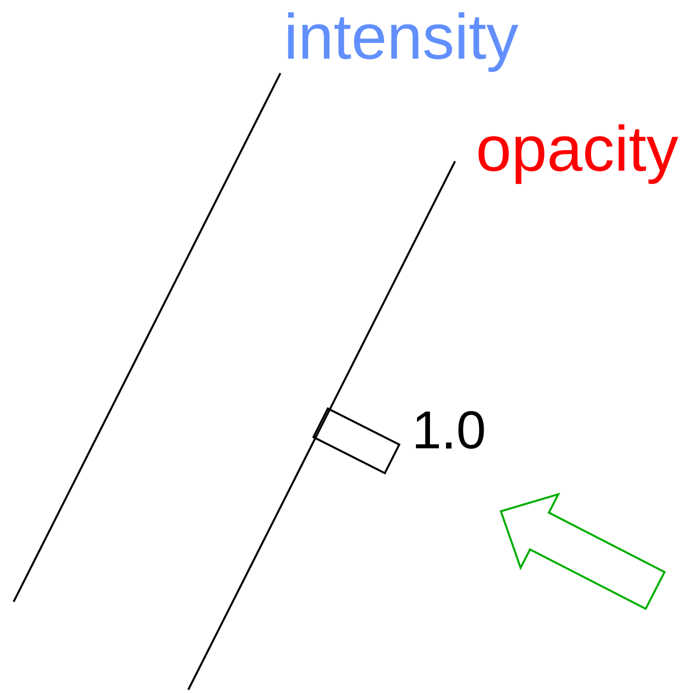
Translucent iso surfaces possible

Iso-Surface Ray Casting

- Alternative approach: search for iso surface value f´ along the ray (similar to first hit rendering) and composite the results
- Each iso-surface value (f´, f´´, f´´´, ….) may have a fixed color and opacity assigned. Alternatively shading parameters (ambient, diffuse and specular reflection parameters) and an opacity value may be assigned.

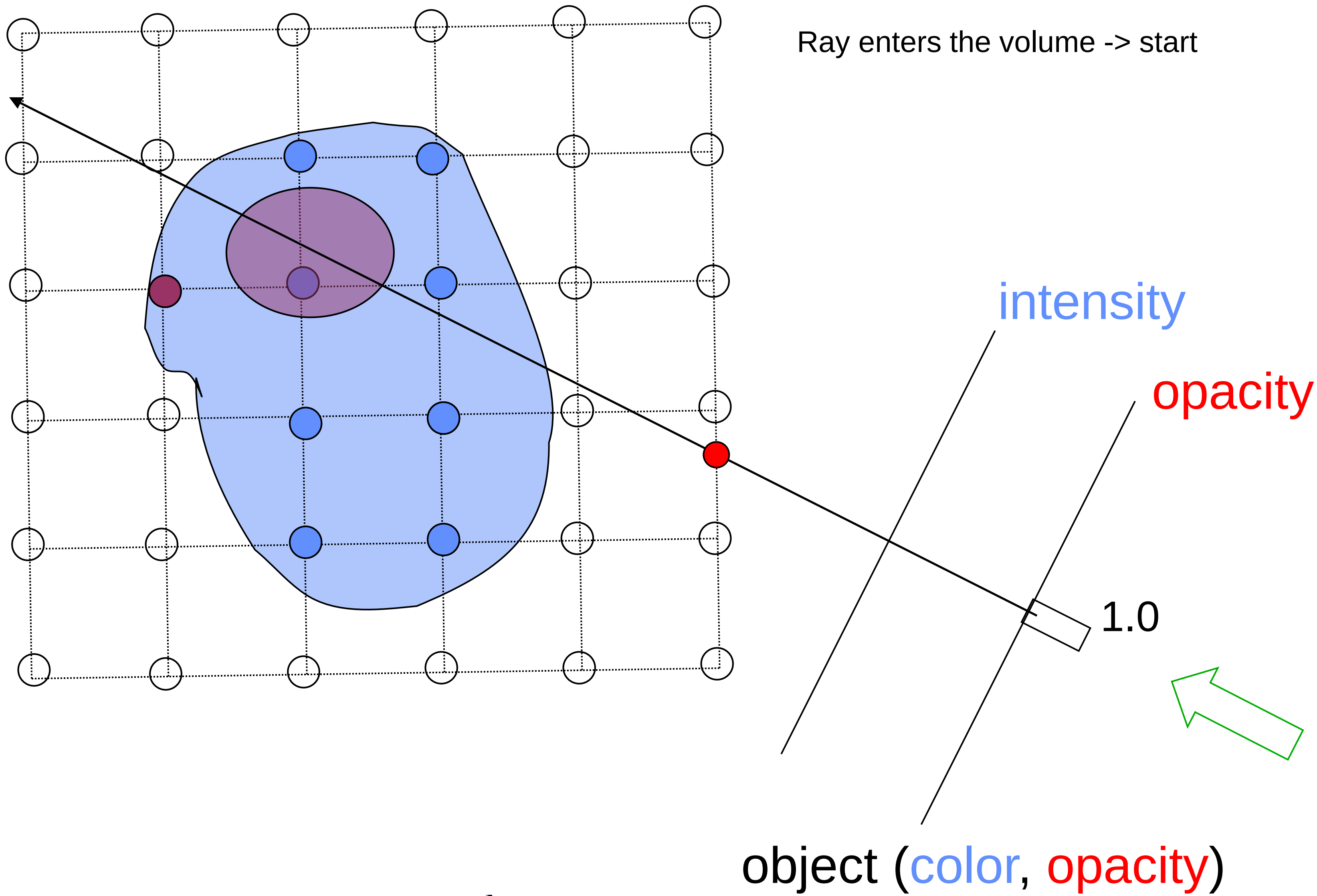
Iso Surface Ray Casting



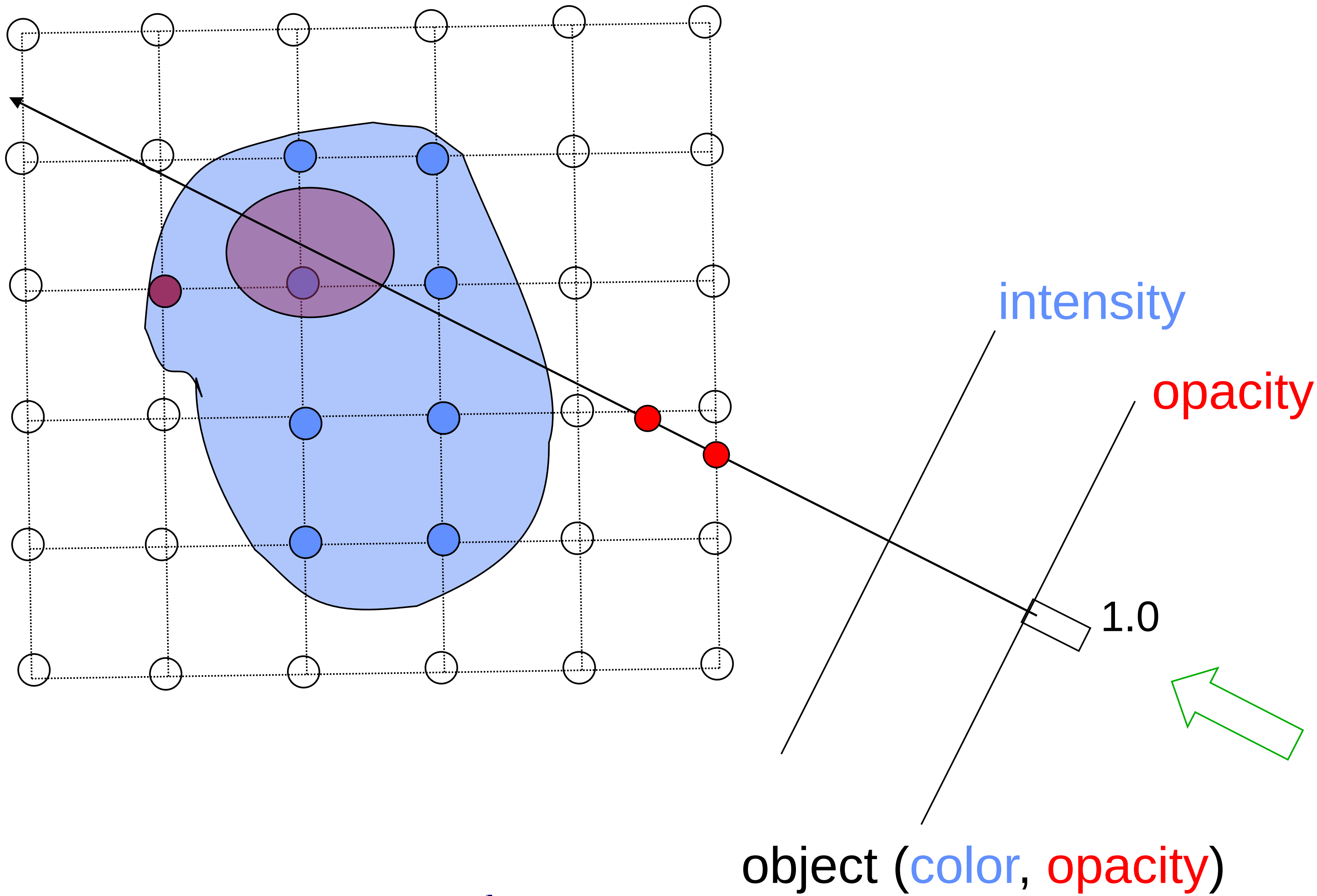


object (color, opacity)

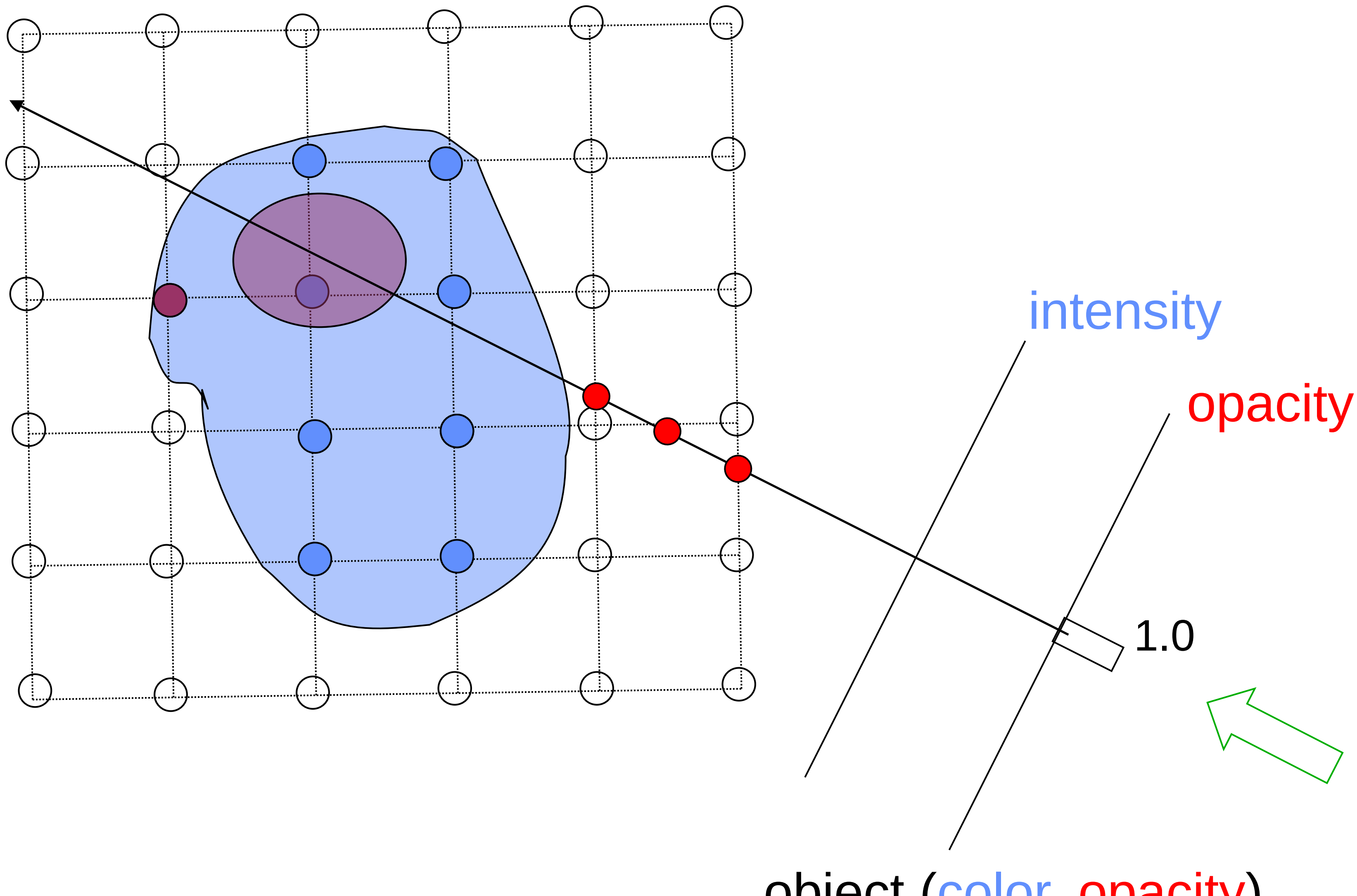
Iso Surface Ray Casting



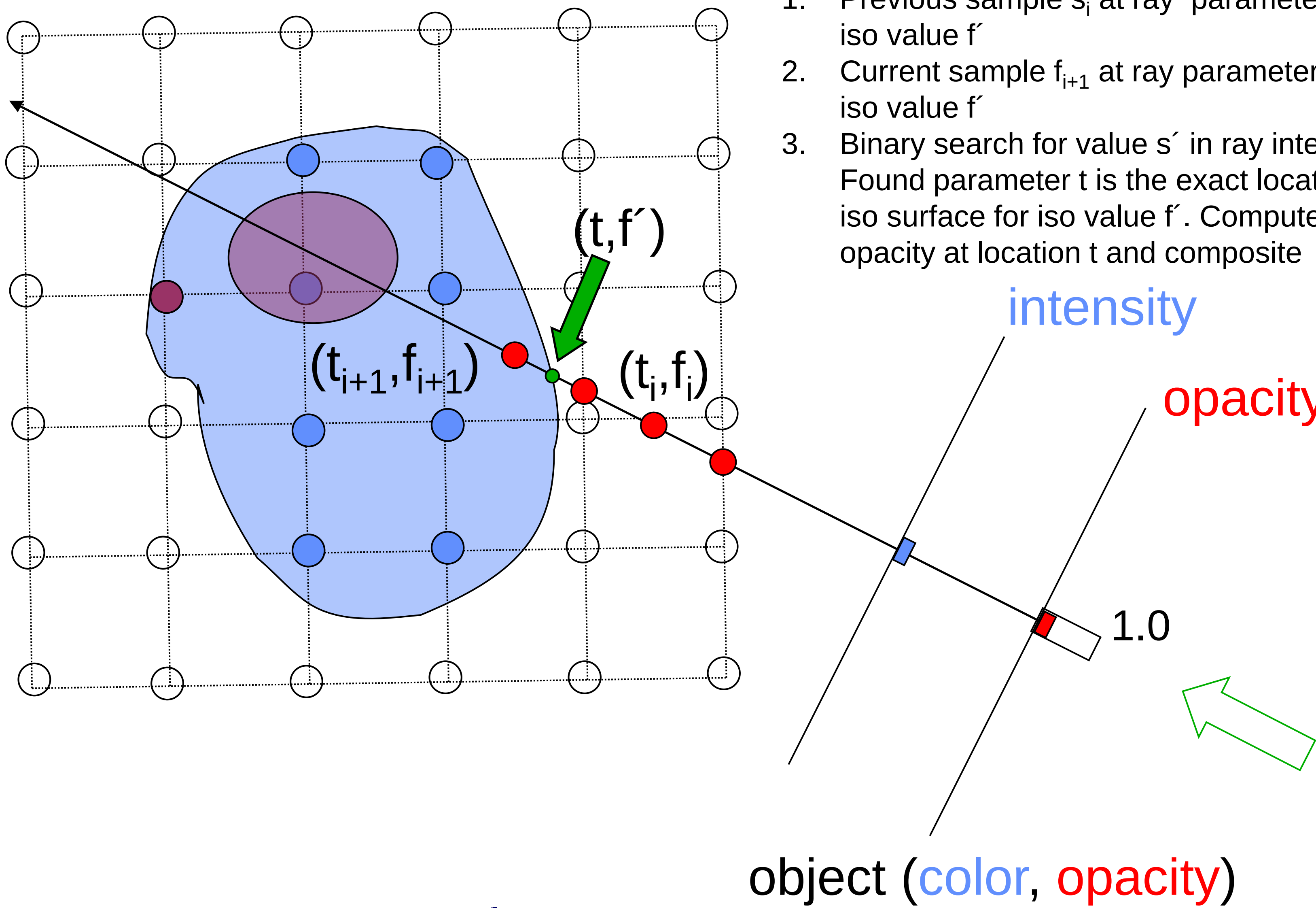
Iso Surface Ray Casting



Iso Surface Ray Casting

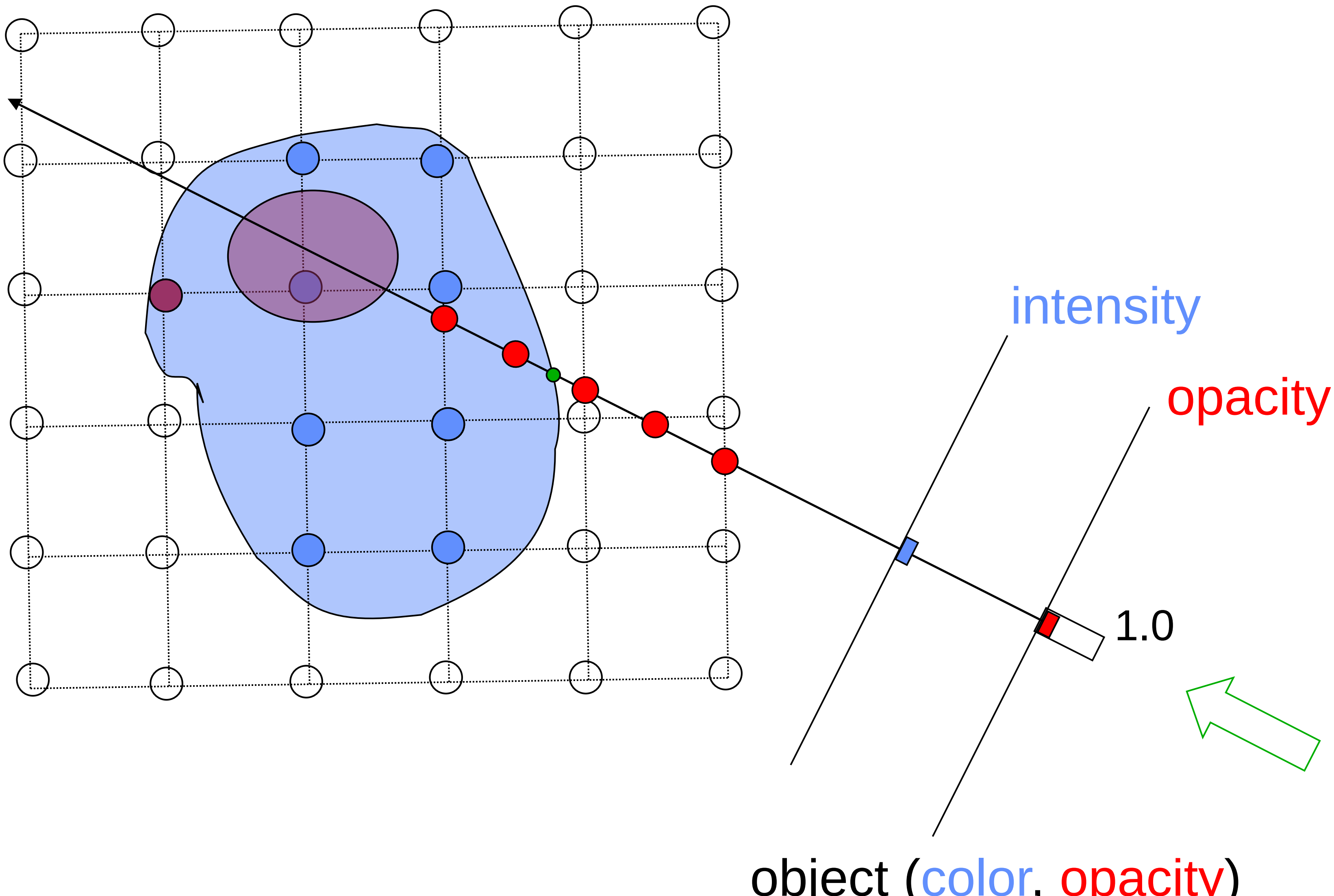


Iso Surface Ray Casting



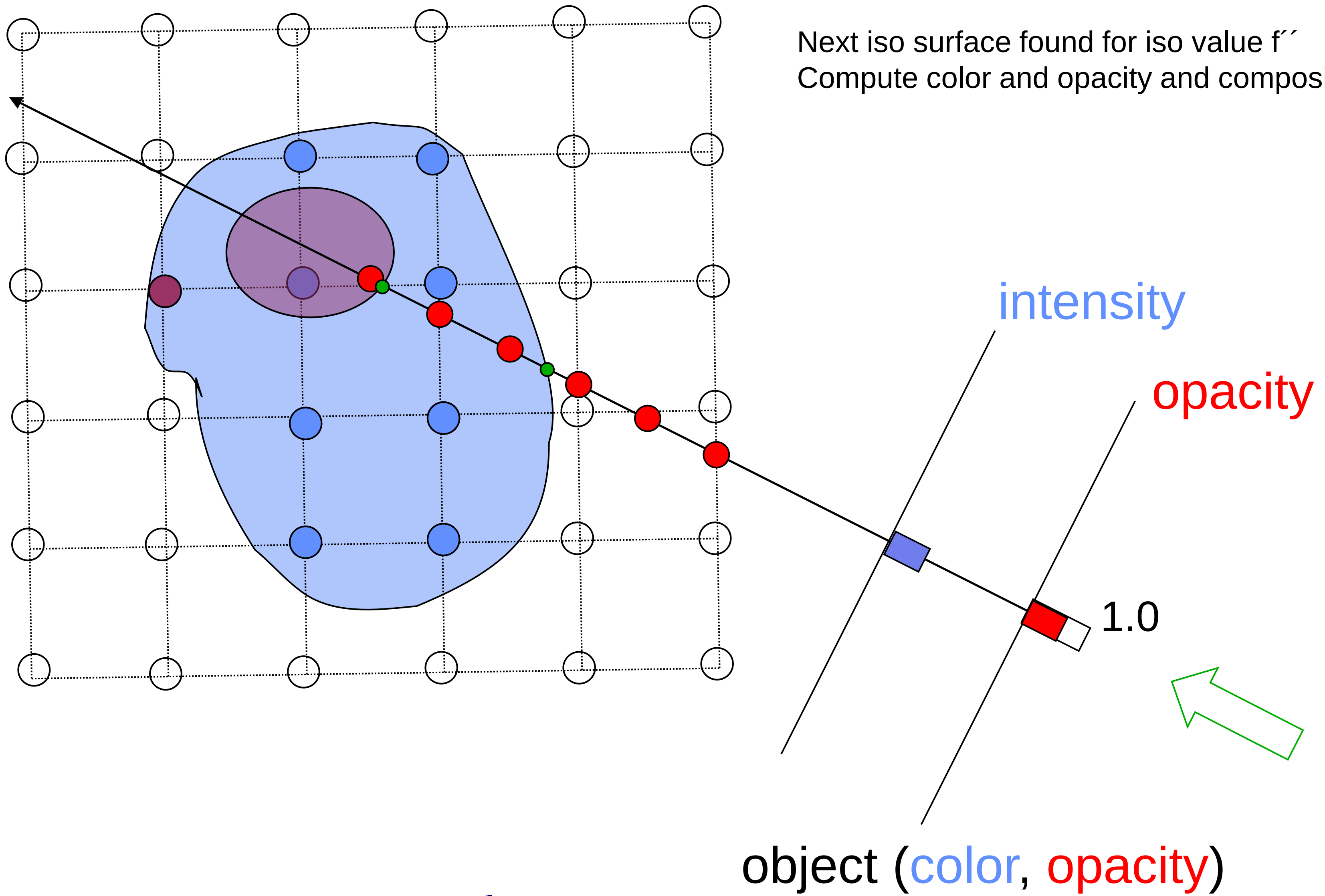
1. Previous sample s_i at ray parameter t_i below iso value f´
2. Current sample f_{i+1} at ray parameter t_{i+1} above iso value f´
3. Binary search for value s´ in ray interval $[t_i,t_{i+1}]$ Found parameter t is the exact location of the iso surface for iso value f´. Compute color and opacity at location t and composite

Iso Surface Ray Casting

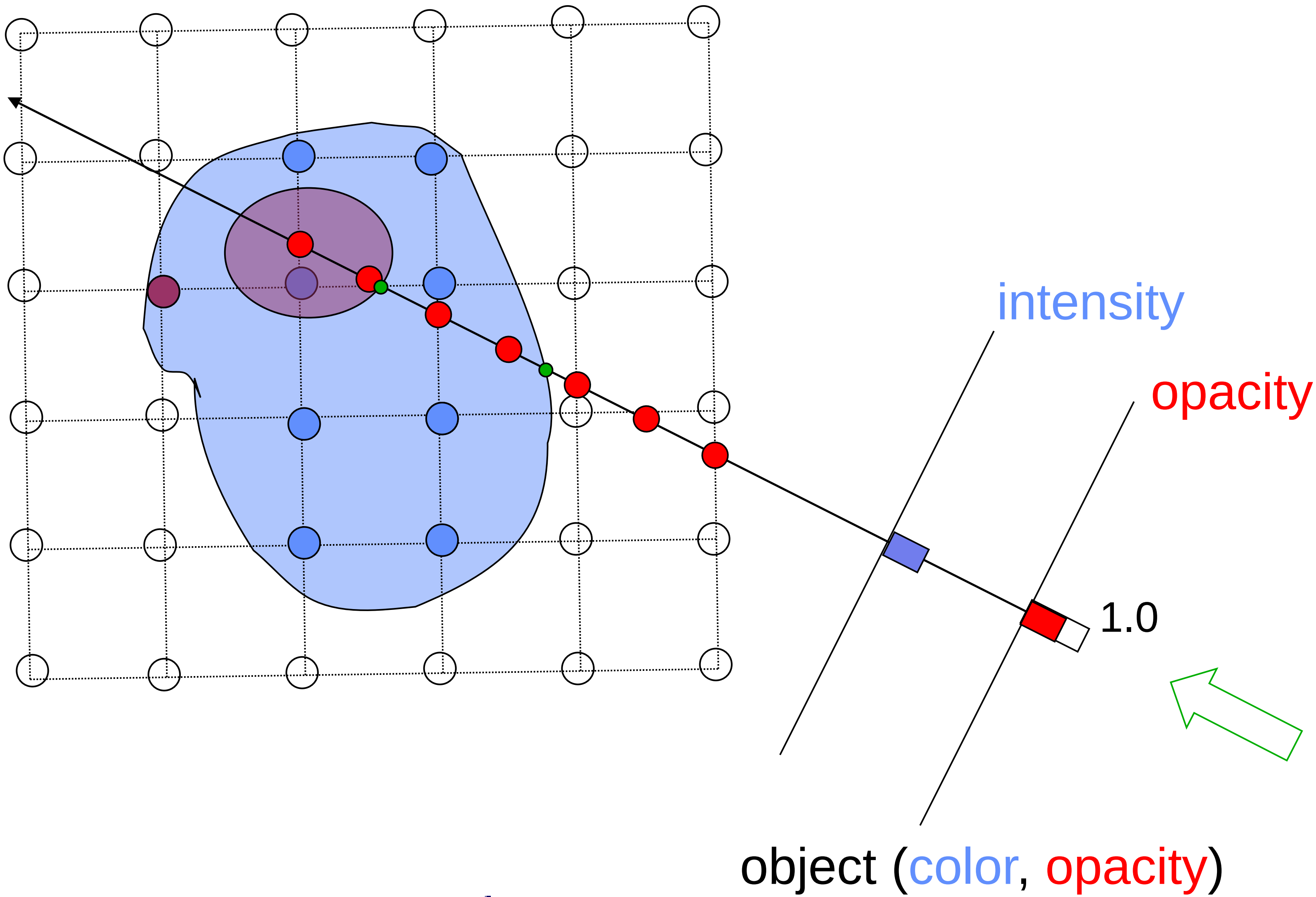


Iso Surface Ray Casting

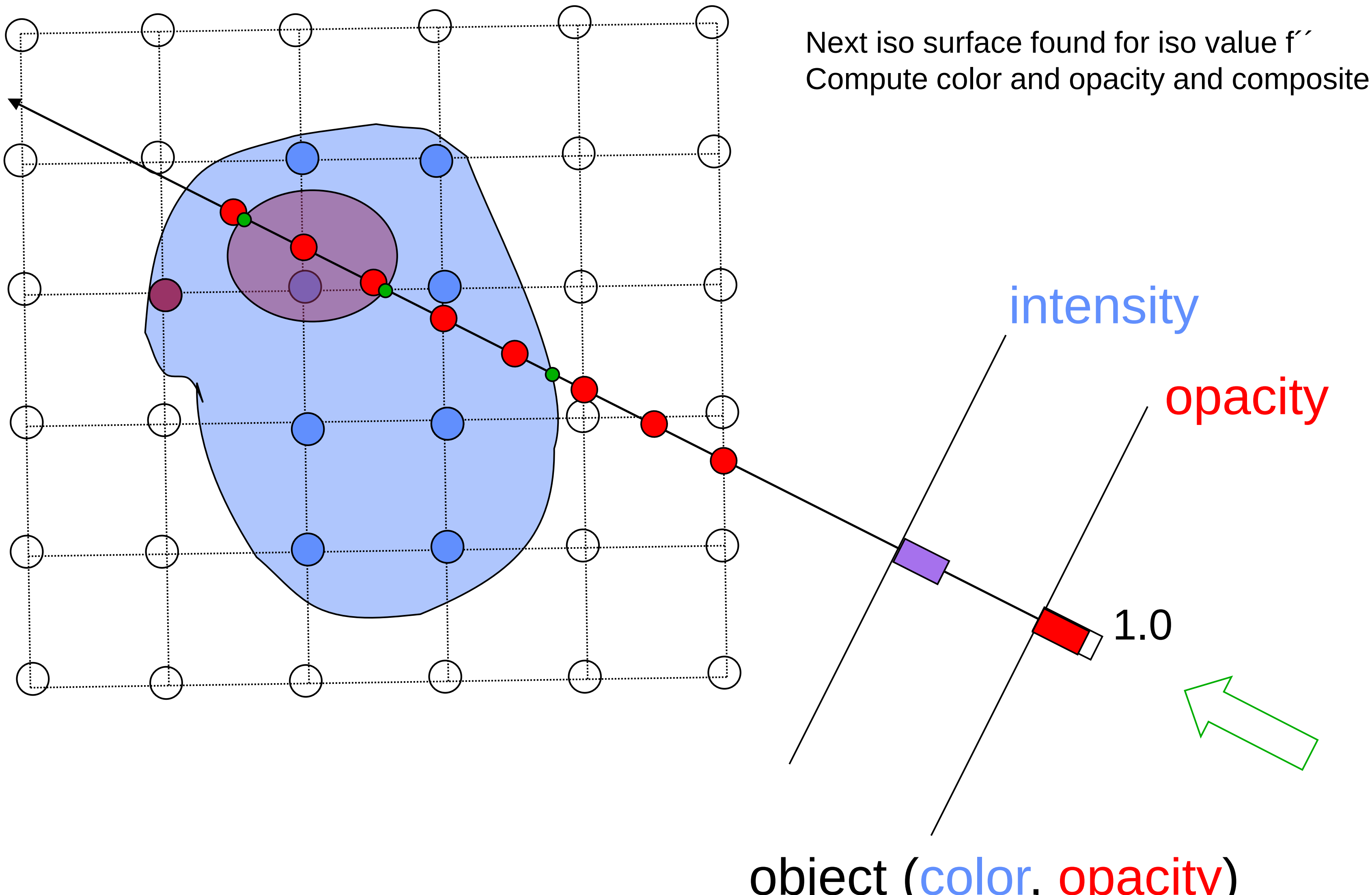
Next iso surface found for iso value f´´
Compute color and opacity and composite



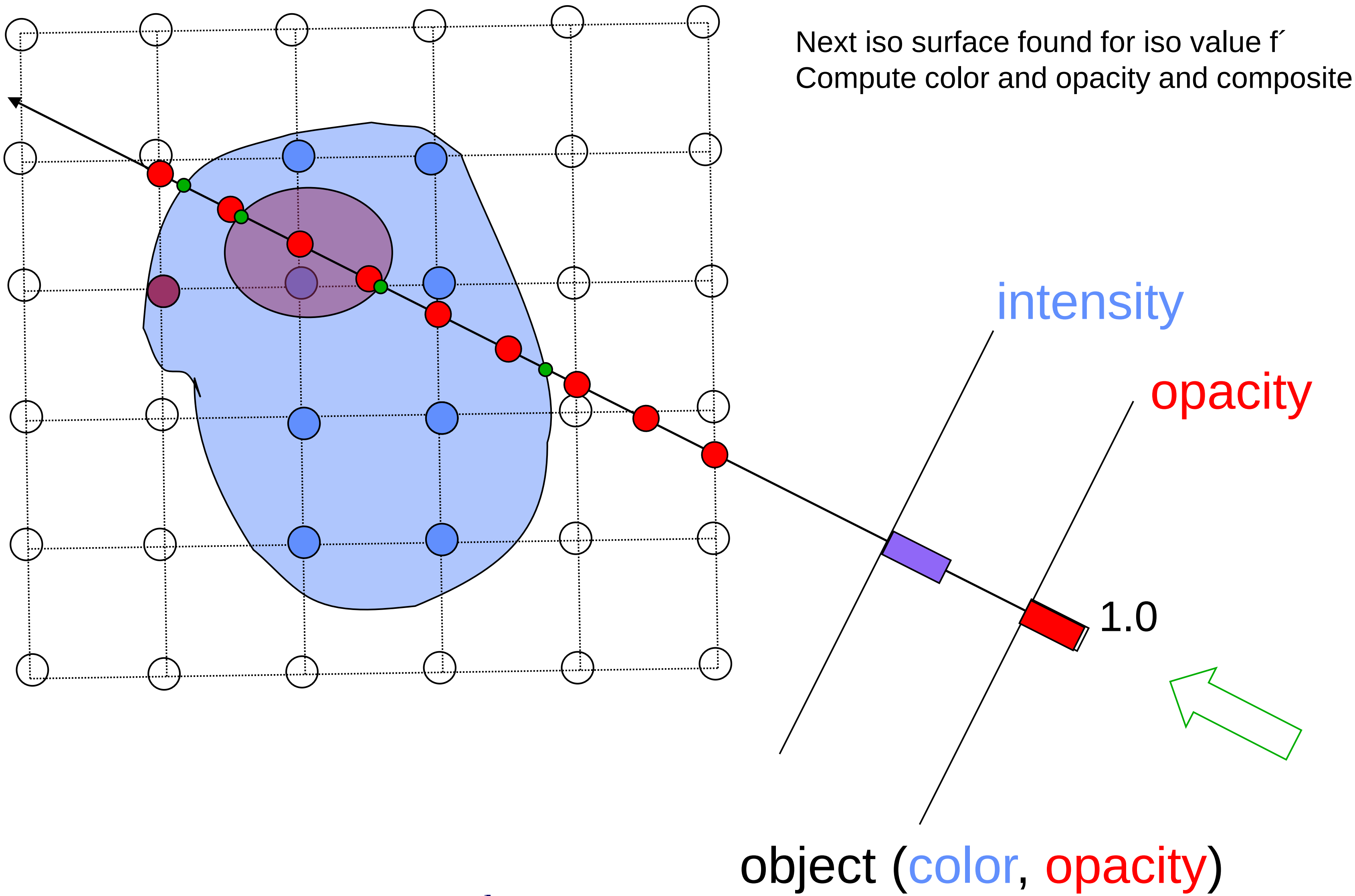
Iso Surface Ray Casting



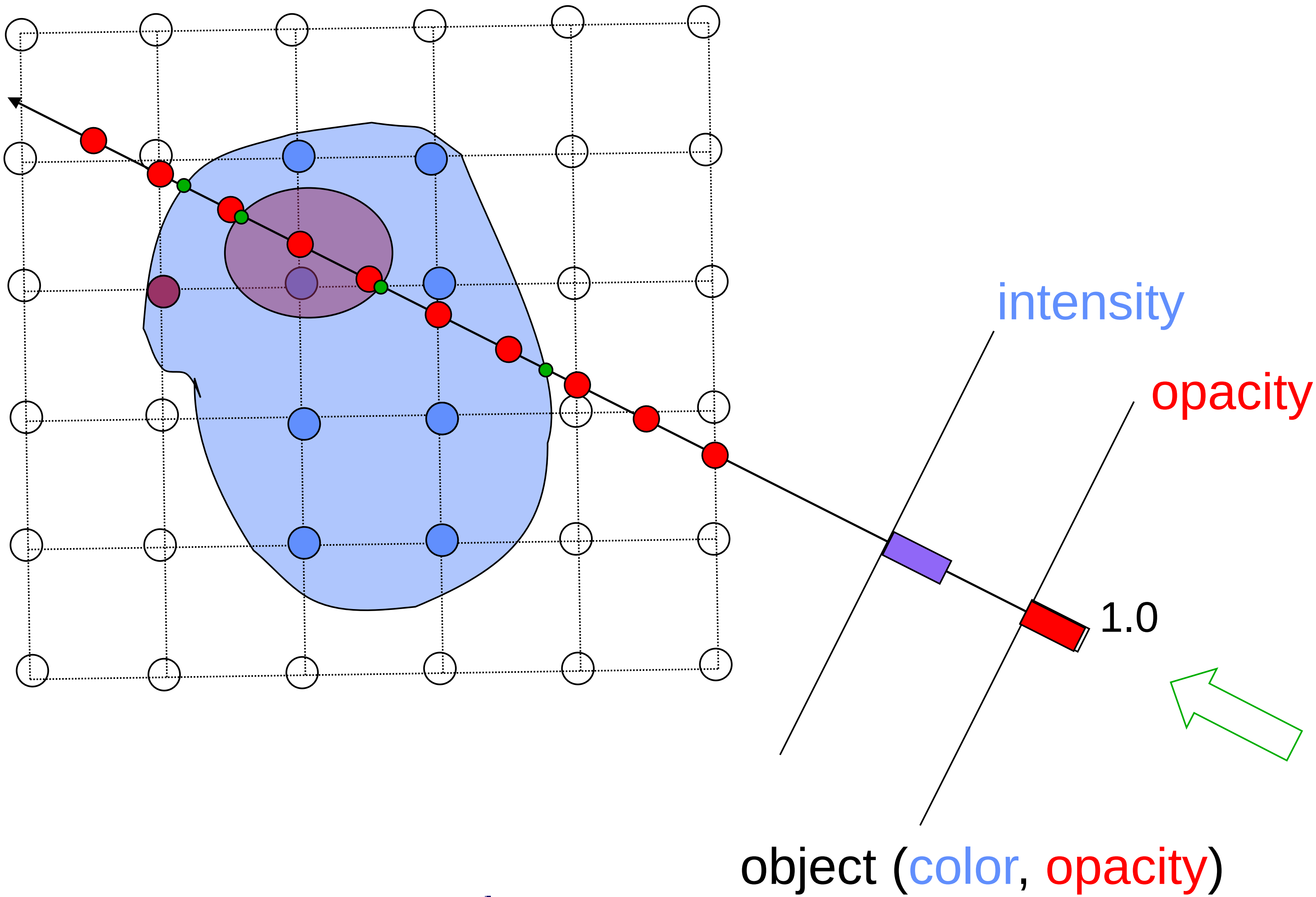
Iso Surface Ray Casting



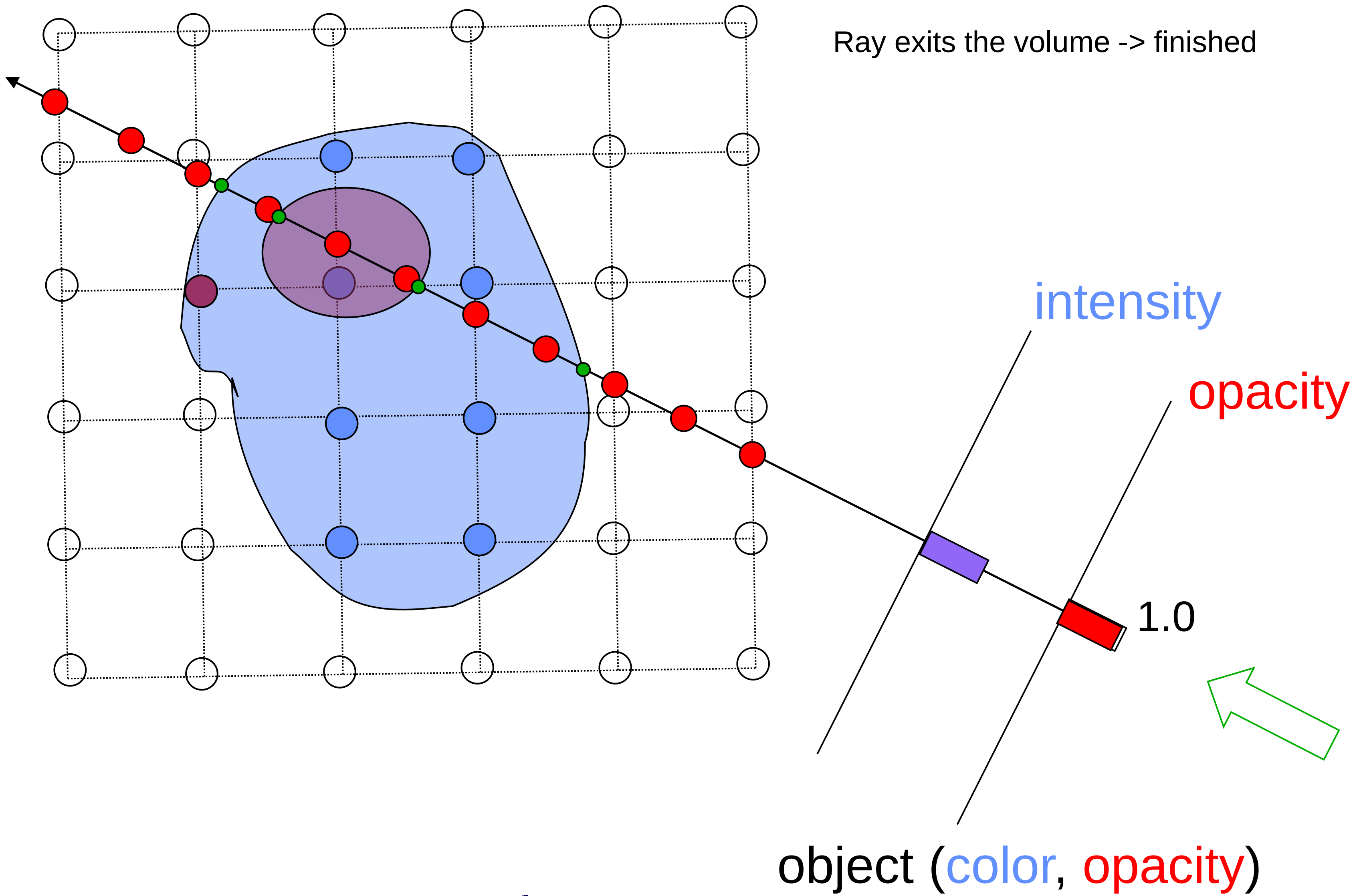
Iso Surface Ray Casting



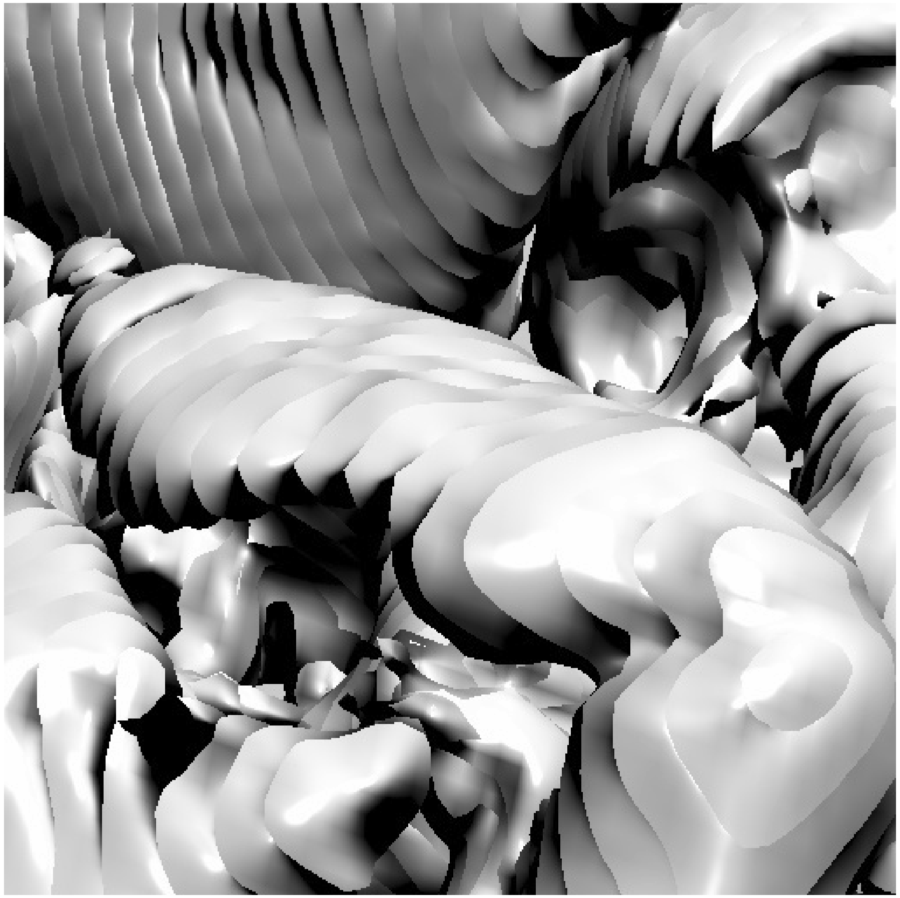
Iso Surface Ray Casting



Iso Surface Ray Casting



Results Iso-Surface Ray Casting

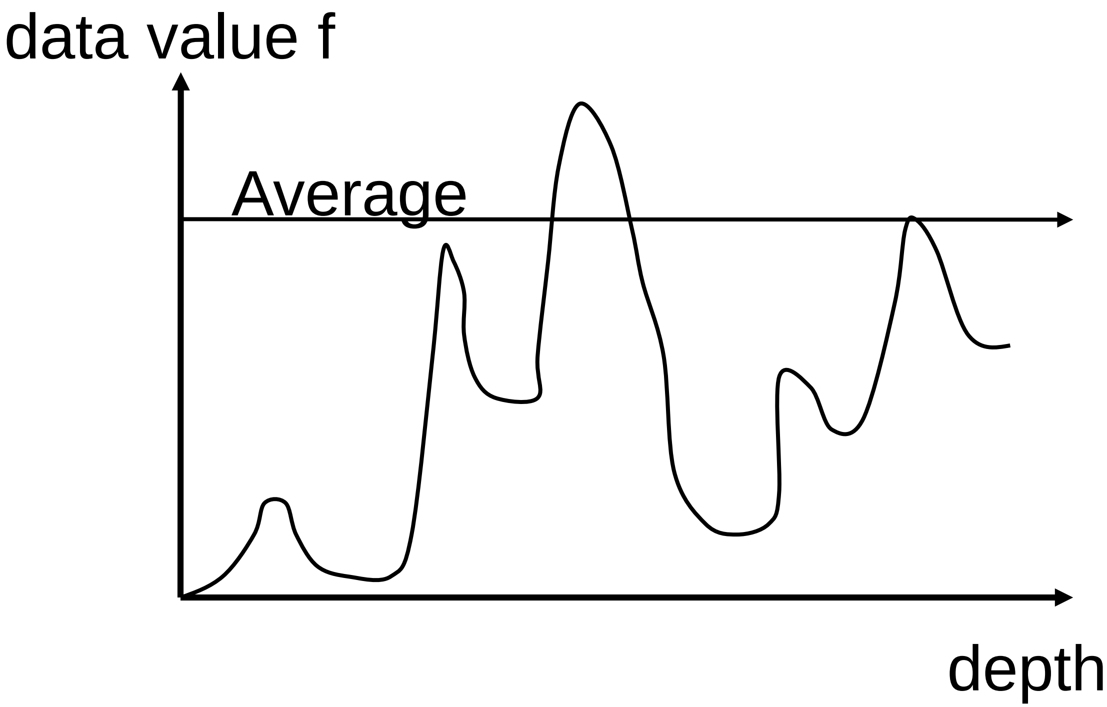


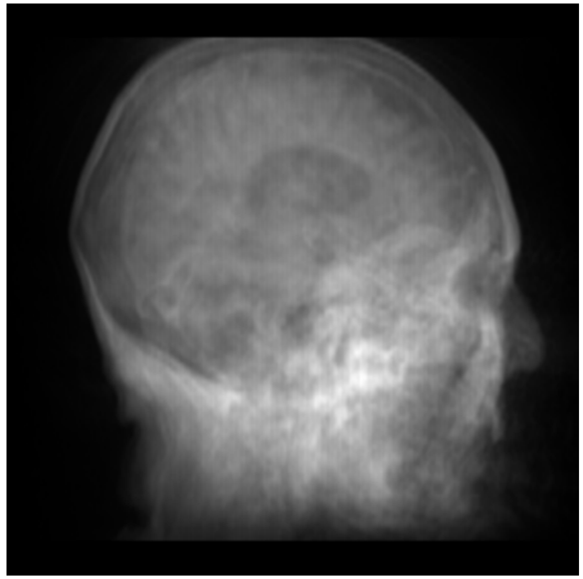
without binary search

with binary search

sample distance: 5 voxels

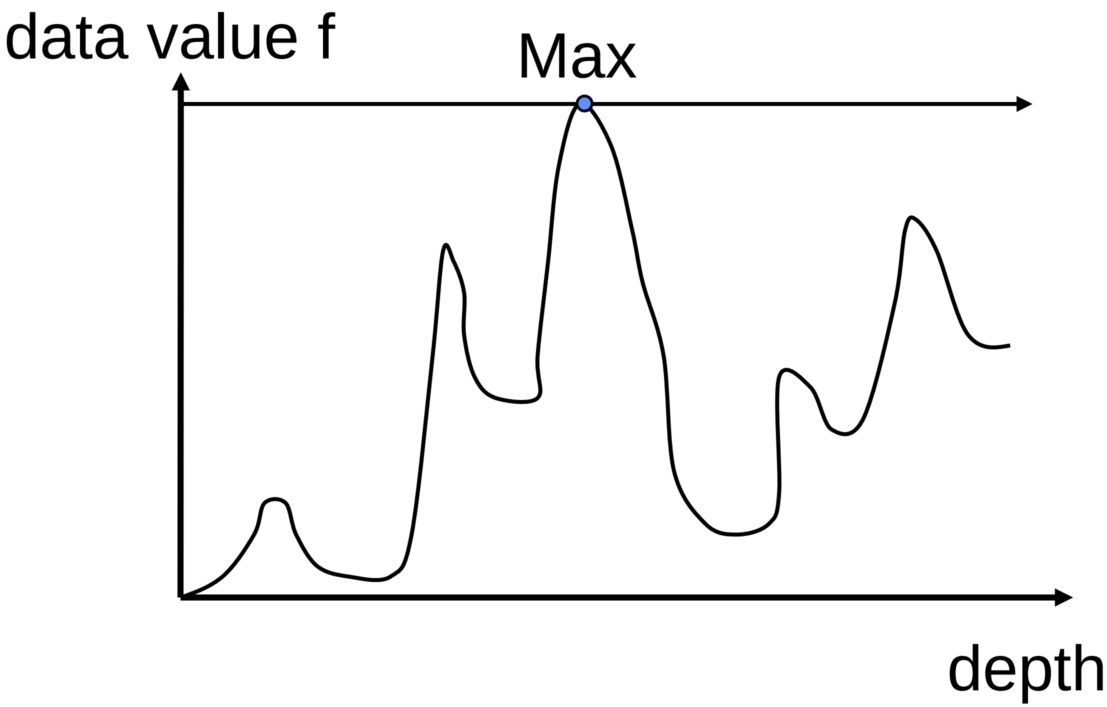
Ray Traversal - Average

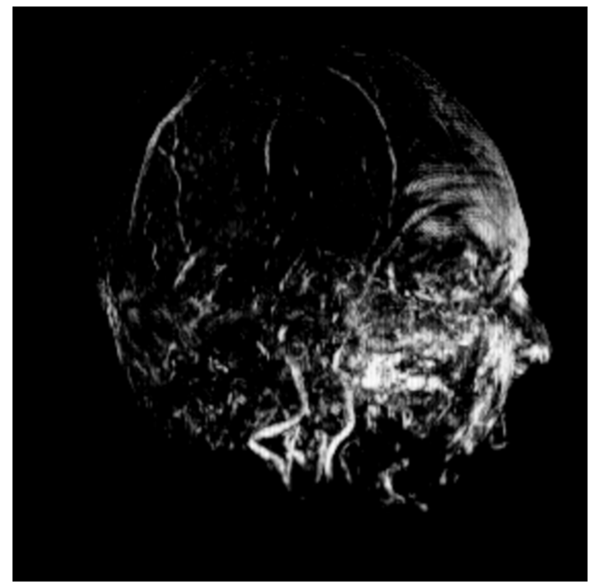




- Produces basically an X-ray picture

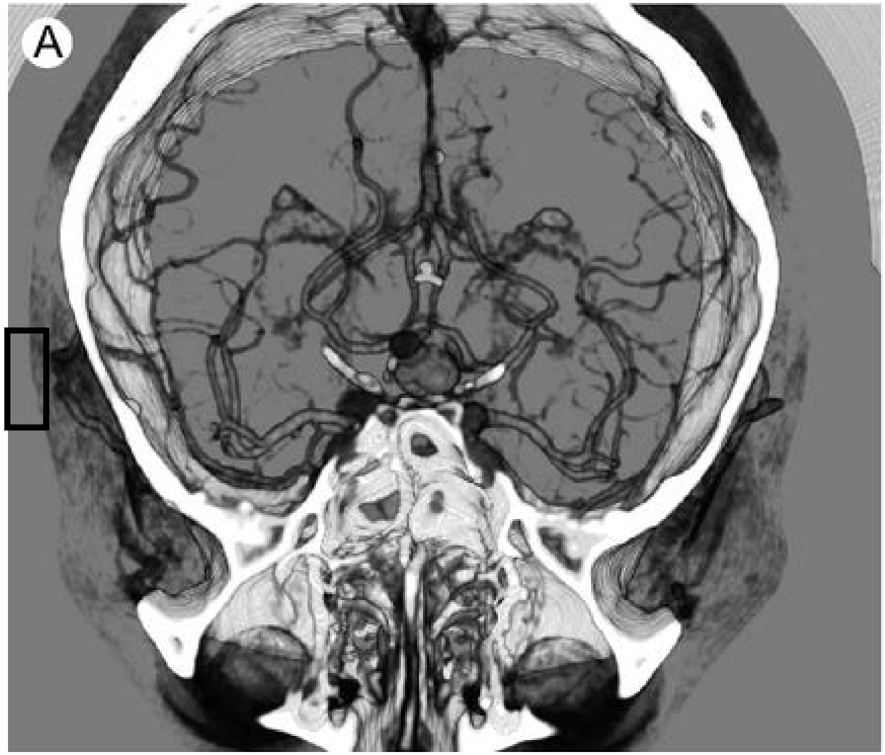
Ray Traversal - MIP

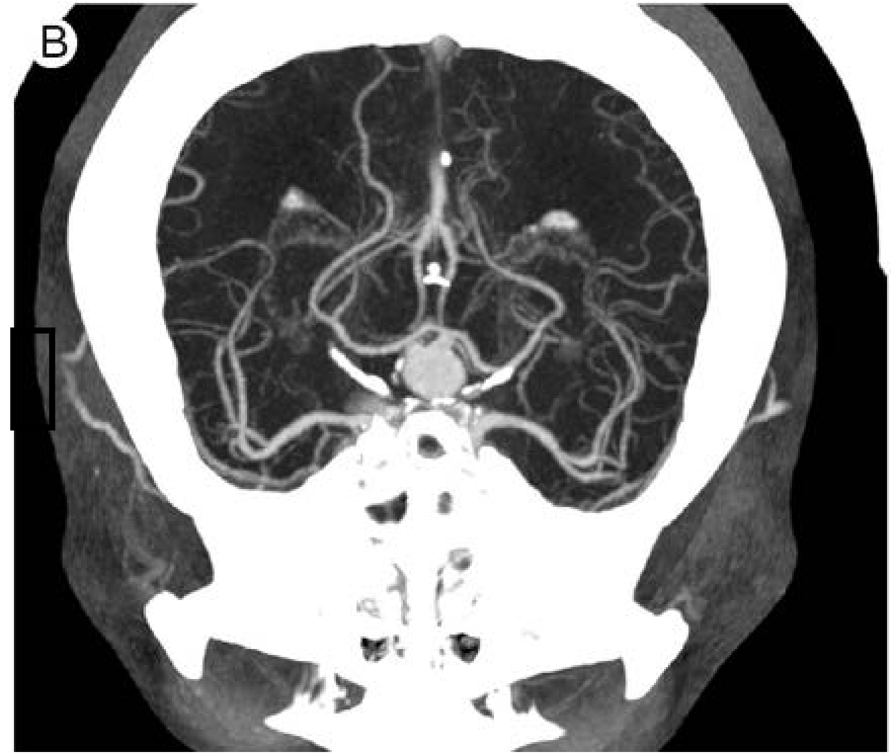




- Maximum Intensity Projection (MIP) used for Magnetic Resonance Angiogram

Maximum Intensity Projection





Emission/Absorption

Maximum Intensity Proj.

Check out talk by Anders Ynnerman

http://www.ted.com/talks/anders_ynnerman_visualizing_the_medical_data_explosion.html

End

Visualization

Advanced Volume Rendering Techniques

SS 2015

Bernd Fröhlich

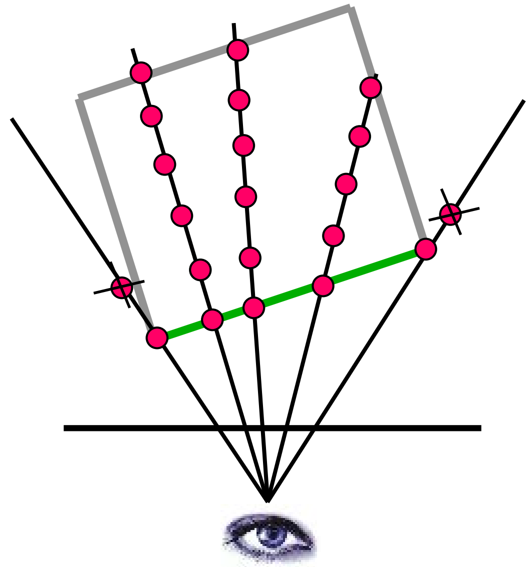
Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Acknowledgements

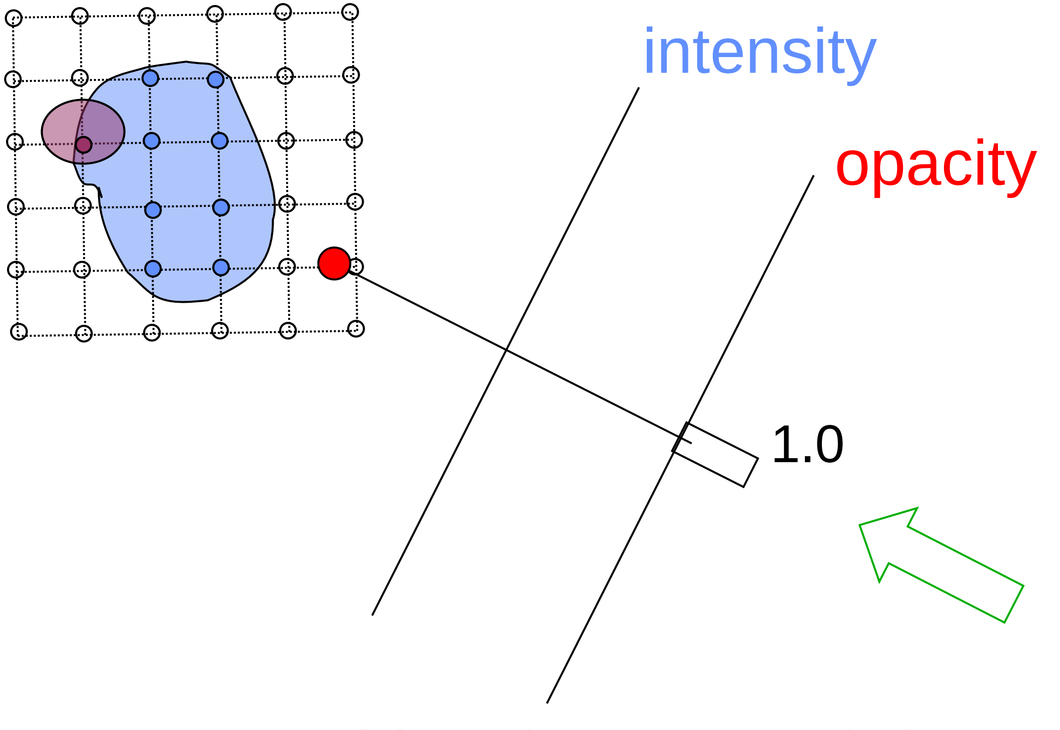
- This lecture is based on
  - Lectures by
    - John Morris, UWA
    - W. Shen, Ohio State University
    - Torsten Moeller, Simon Fraser University, Vancouver
    - Chris Johnson, Scientific Computing and Imaging group at the University of Utah
    - Markus Hadwiger
  - Tutorial IEEE Visualization 2002 / 2003
    - High-Quality Volume Graphics on Consumer PC Hardware
    - http://www.cs.utah.edu/~jmk/sigg_crs_02/courses_0067.html
  - Book Visualisierung by Schumann and Mueller
  - Book Real-Time Volume Graphics, Engel et al.

GPU Ray Casting

- Approach
  - Store volume in 3D texture
  - Render bounding box of volume data
  - For each generated fragment (Pixel)
    - Reconstruct ray
    - March through volume
      in fragment shader
      - Composite front-to-back
  - Early ray termination
    possible



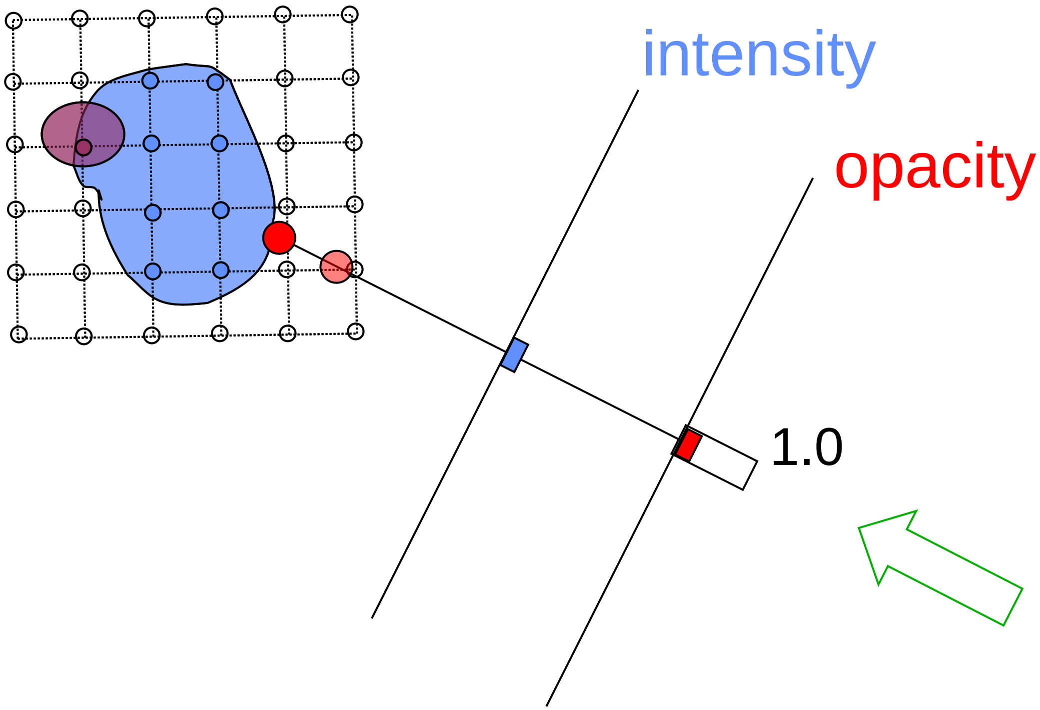
Optimizations: Early Ray Termination



object (color, opacity)

Optimizations: Early Ray Termination

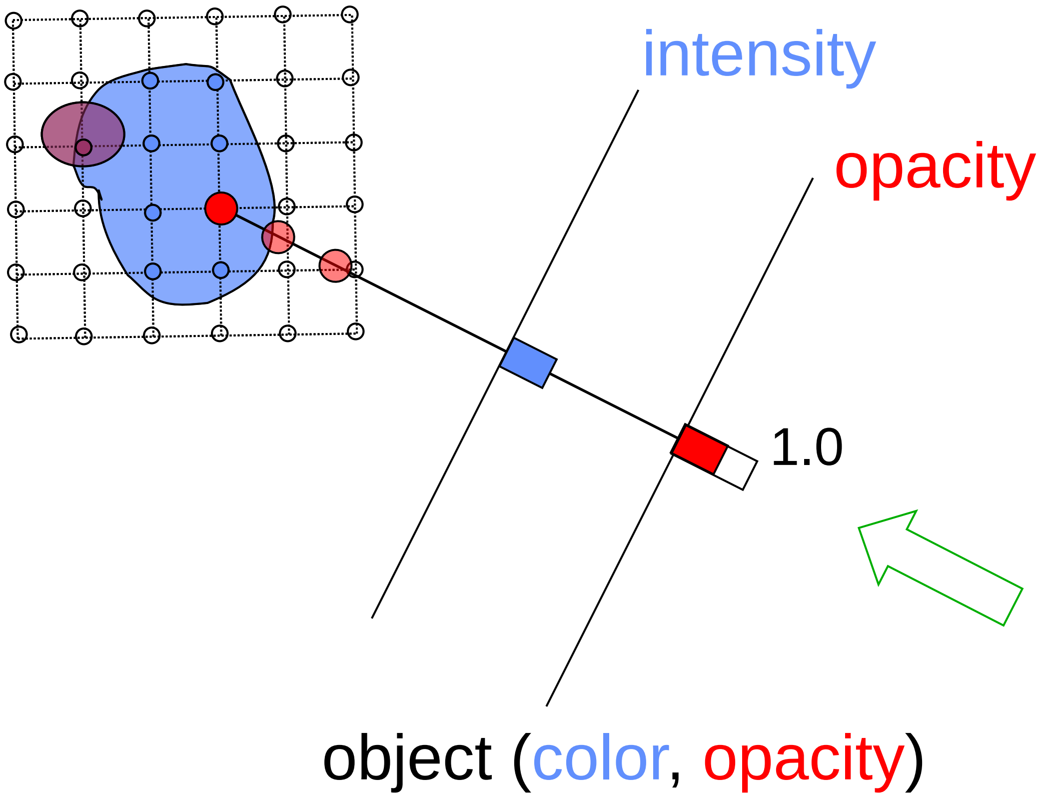
volumetric compositing



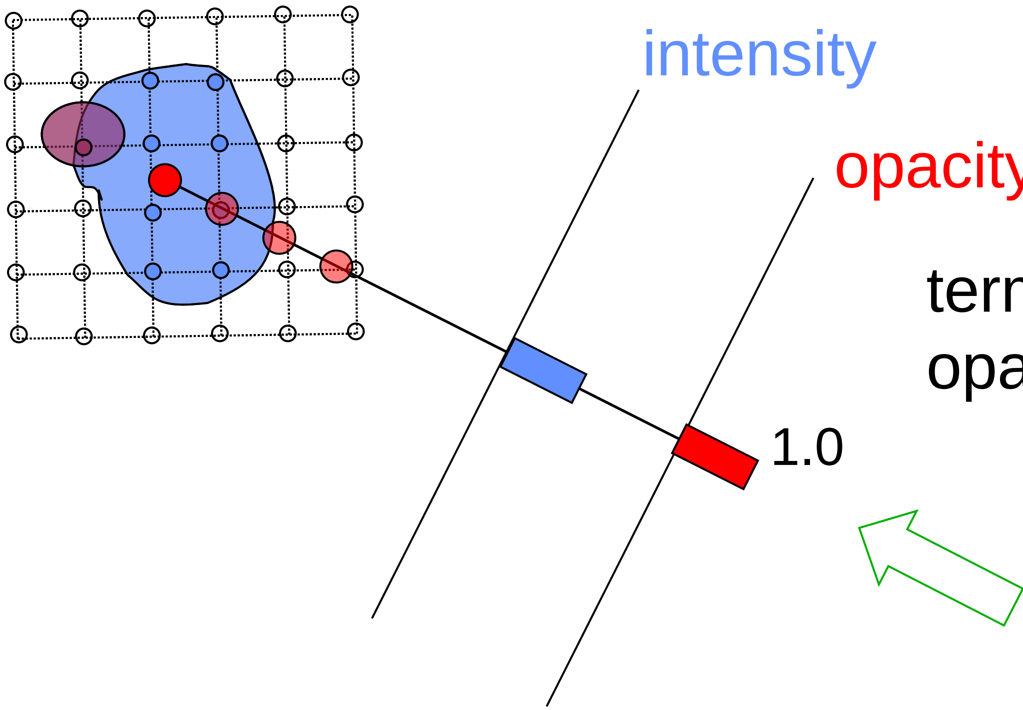
object (color, opacity)

Optimizations: Early Ray Termination

volumetric compositing



Optimizations: Early Ray Termination



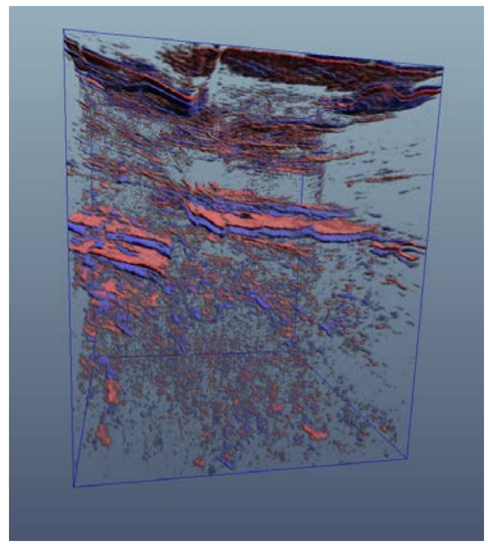
terminate ray:
opacity 1.0 reached

object (color, opacity)

Empty Space Skipping: Motivation

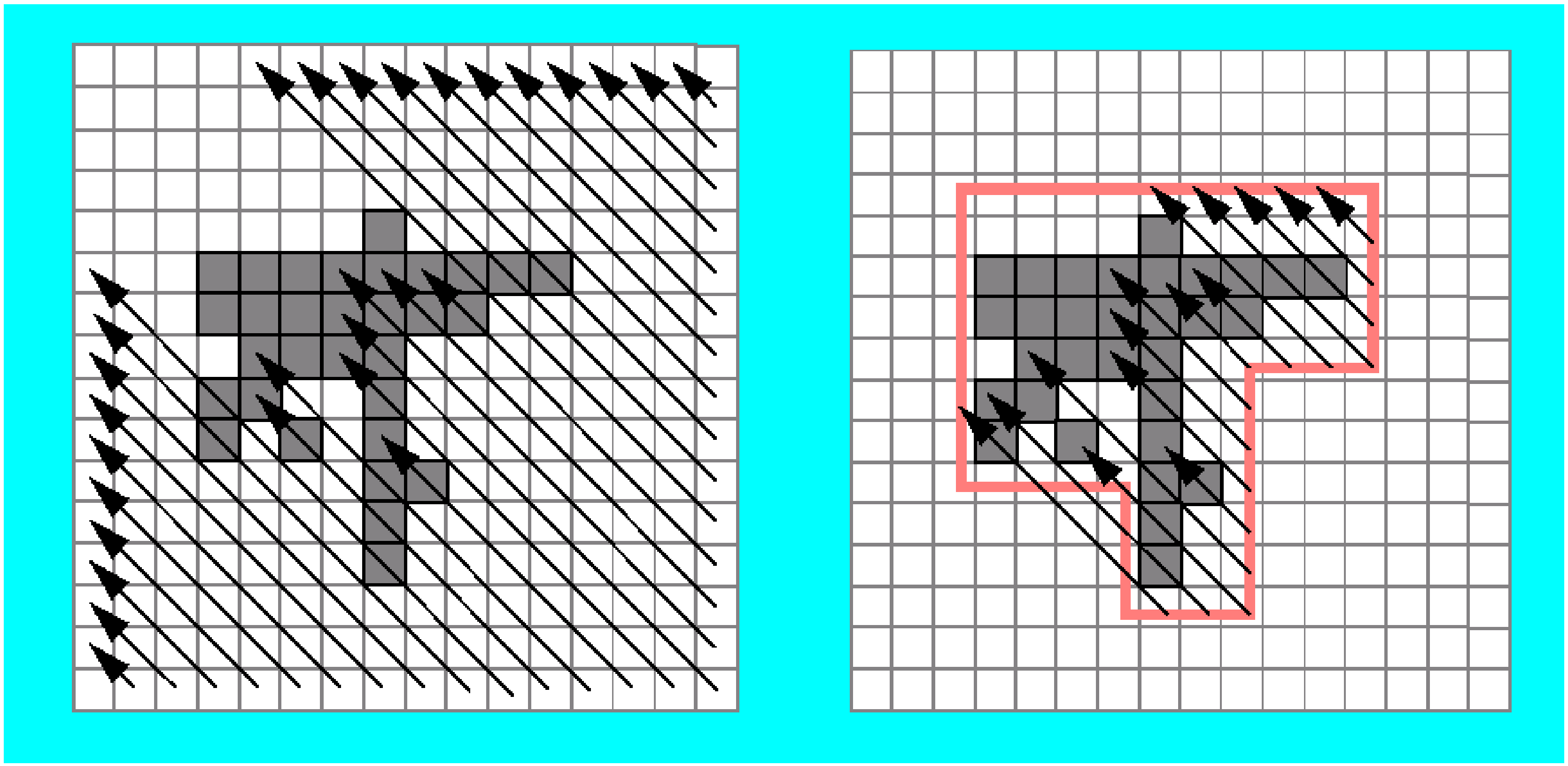
- Large empty volume regions
  - Depending on opacity transfer function
  - Do not contribute to final picture
  - Treated the same as filled regions
  - Create unnecessary computations

⇨ Skipping over empty regions improves efficiency



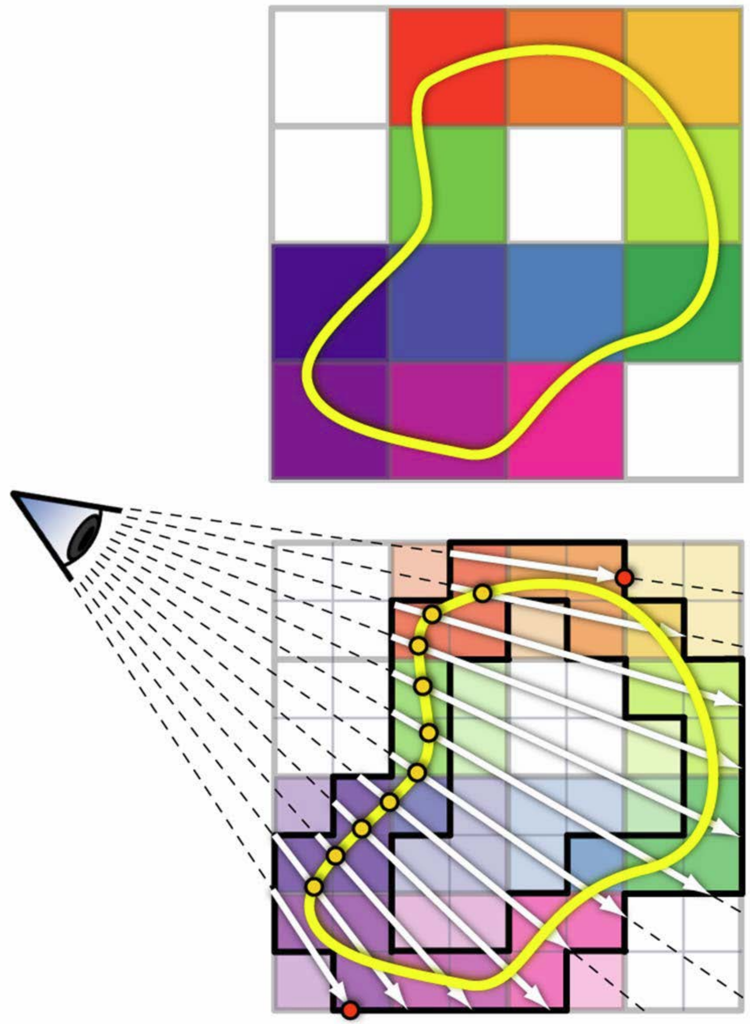
Empty Space Skipping: Data Structure

- Tight bounding volume of the data instead of a bounding box



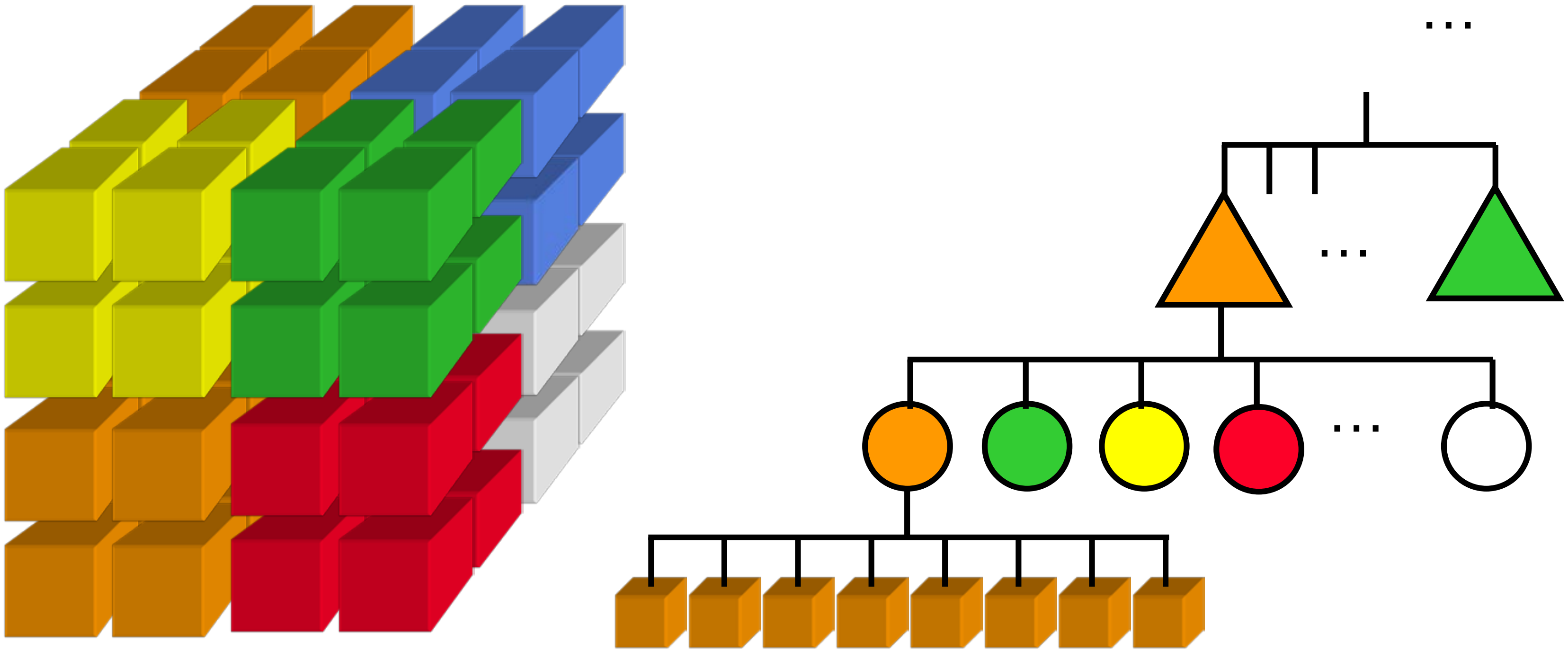
Empty Space Skipping: Data Structure

- Coarse single-level grid or multiple grid levels or octree
  - Store min/max data values in grid cells
  - Skip over coarse grid cells if no relevant data is contained



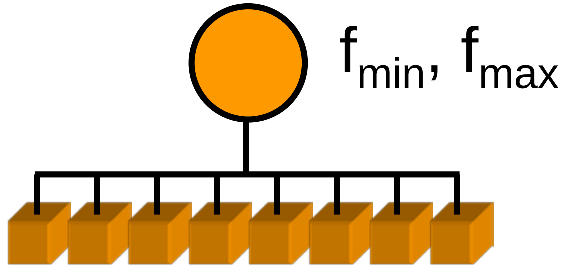
Min-Max Octree

- Build min-max octree



Use of Min-Max Octree Hierarchy

- Each node holds minimal and maximal data value of its children

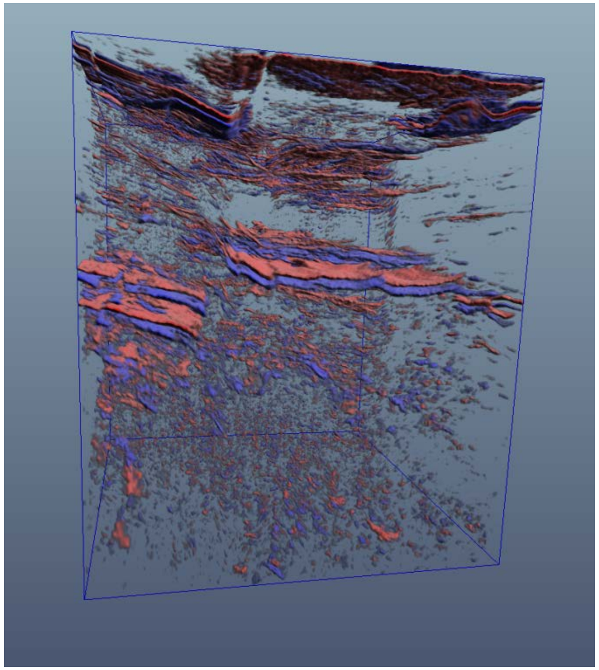


- For iso-surface ray casting
  - Recursively traverse octree with ray and intersect only those children which contain iso-value **C** and discard nodes with $\mathbf{C < f_{min}}$ or $\mathbf{C > f_{max}}$

Optimizations: Empty Space Skipping

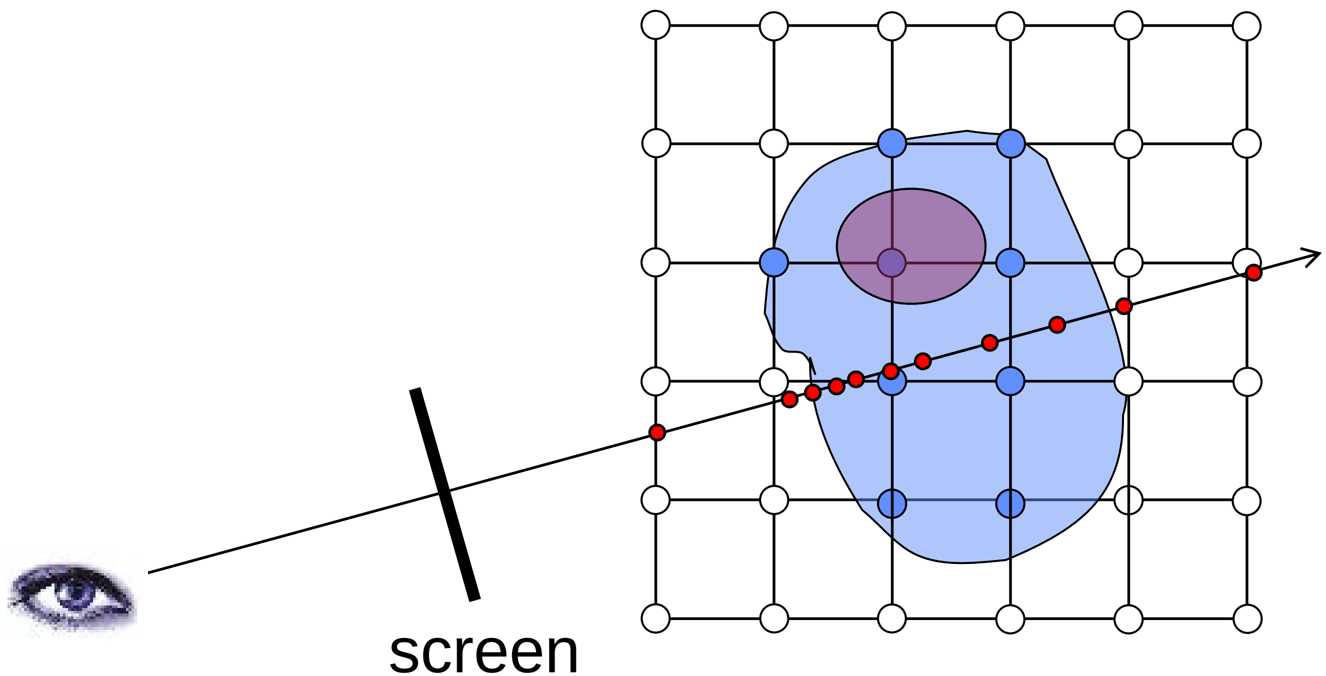
- Skip over empty regions during ray traversal
  - Identify areas for skipping (overhead)
  - Reduces volume texture lookups
  - Reduces sample interpolations
  - Reduces compositing calculations

⇨ Improves efficiency



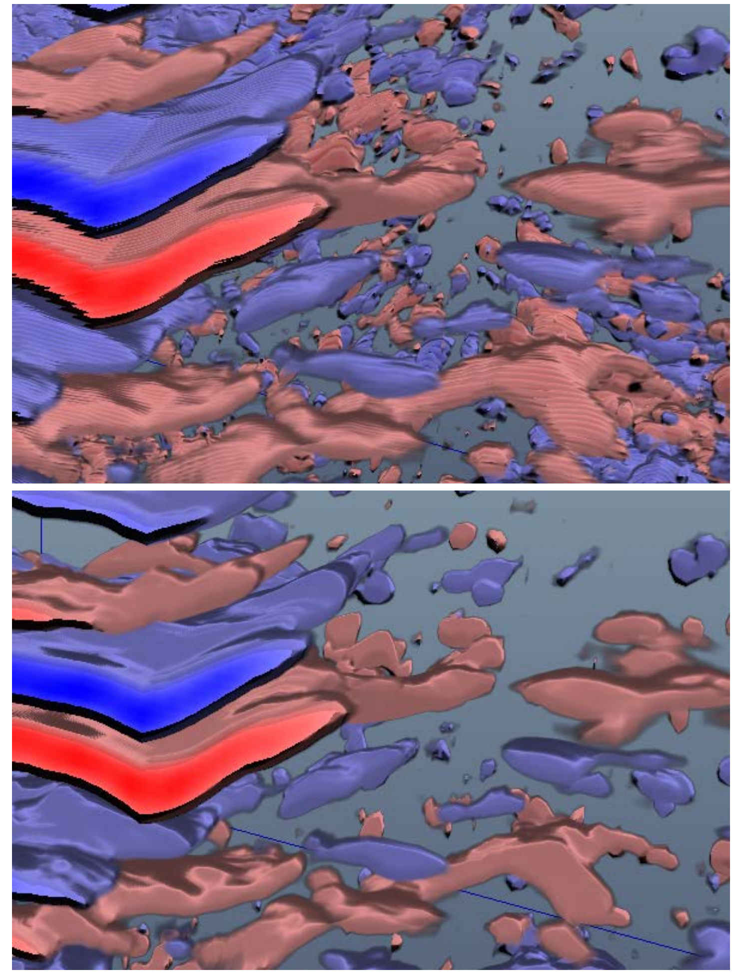
Optimizations: Adaptive Ray Sampling (Hanrahan 92)

- Sampling distance is adjusted to the significance of the data
  - E.g. depending on the gradient magnitude or accumulated opacity



Optimizations: Adaptive Sampling

- Adapt sampling density along ray
  - Minimize in homogeneous volume regions
  - Concentrate samples in high detail regions
  - Depending on
    - Viewer distance
    - Gradient magnitude
    - Accumulated opacity along ray
- Obvious extension of empty space skipping
- ⇨ Improve quality and efficiency
- Needs opacity correction!



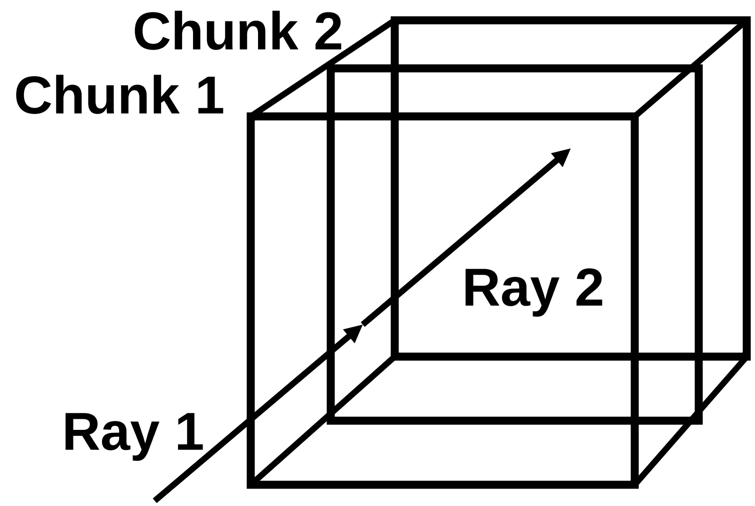
Opacity Correction

- Approximation of the volume rendering integral through sampling assumes constant sampling distance d
  ⇨ need to adjust alpha and intensity based on actual sampling distance $\tilde{d}$
- Transparency correction $$\tilde{T} = T^{\frac{\tilde{d}}{d}}$$
- Opacity correction $$\tilde{\alpha} = 1 - (1 - \alpha)^{\frac{\tilde{d}}{d}}$$
- Intensity computation $$\tilde{I} = C\tilde{\alpha}$$

α	Regular opacity	d	Regular sampling distance
$\tilde{\alpha}$	Adapted opacity	$\tilde{d}$	Adapted sampling distance

Partial Ray Compositing

- Both rendering methods (front-to-back / back-to-front) allow partial ray *compositing*, split a ray in segments and treat each segment as a separate ray

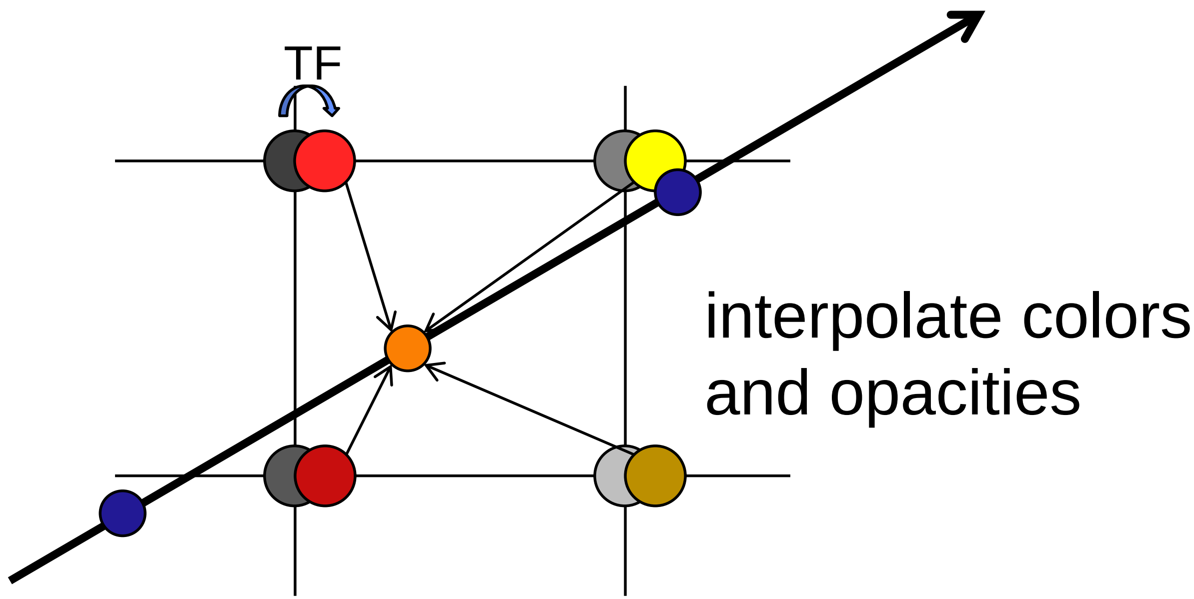


- Ray 1 and Ray 2 can be calculated independently and *in parallel*
- Result is composited sum from both rays
  - no early ray termination
  - may still be useful if the data is distributed to different graphics cards (data partitioning) and volume is mostly translucent

- NOTE: Parallelization typically based on screen partitioning

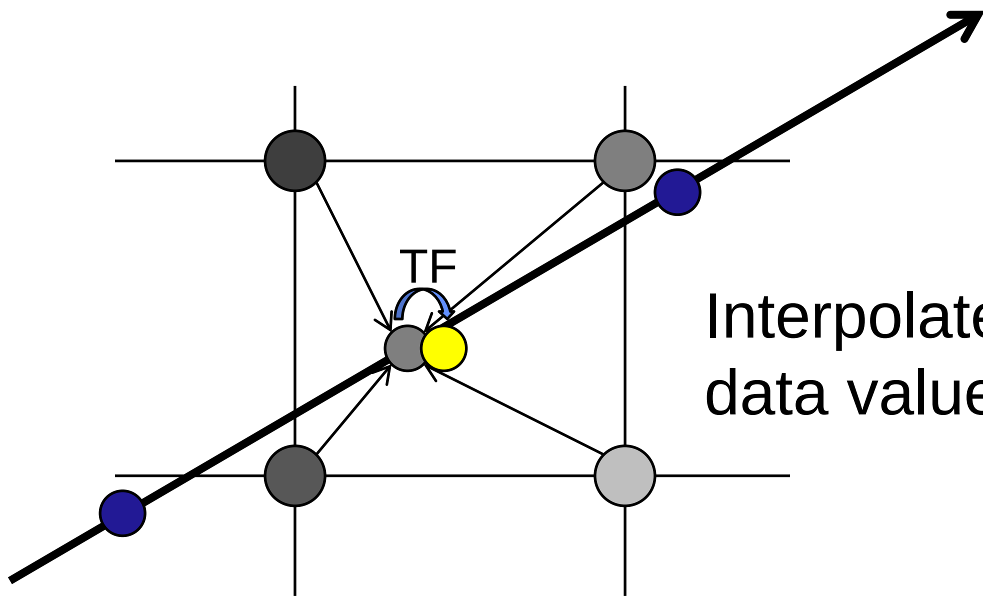
Pre-Classification

- For each sample along the ray
  - Use transfer function (TF) for each data value of the voxel surrounding the sample point
  - Tri-linear interpolation of colors and opacities
  - Composite resulting color and opacity



Post-Classification

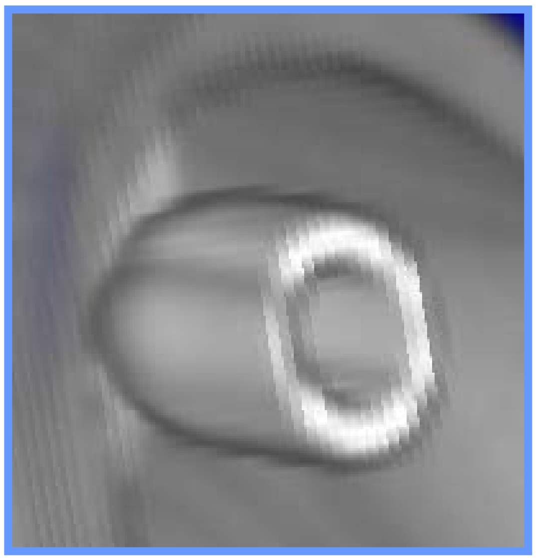
- For each sample along the ray
  - Tri-linear interpolation of data values at the voxel vertices surrounding the sample point
  - Use transfer function TF for resulting interpolated data value to obtain color and opacity at sample point
  - Composite resulting color and opacity

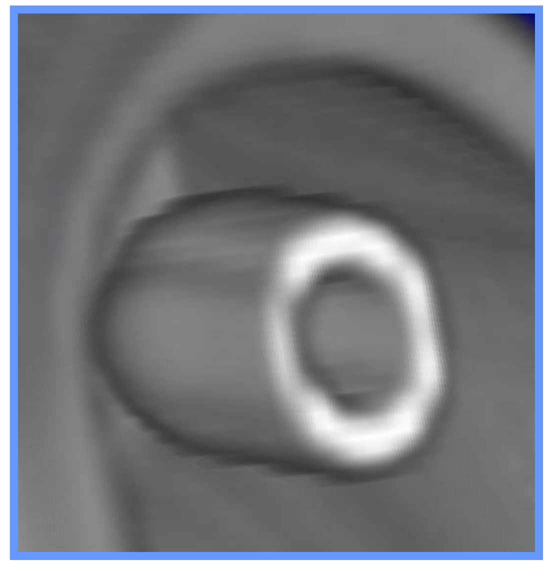


Interpolate
data values

Ray Integration

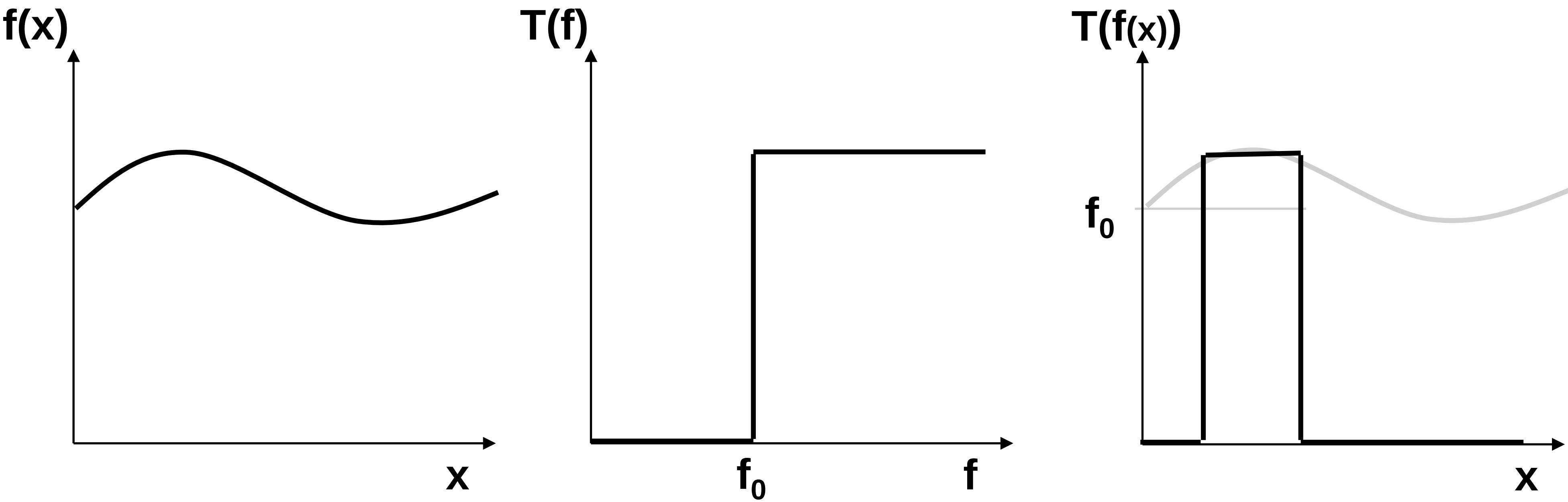
- Discrete approximation of volume rendering integral will converge against correct result for sample distance d → 0. Low sampling rate will result in sampling artifacts.





High-Frequency TFs

- High frequencies in the transfer function T increase required sampling rate



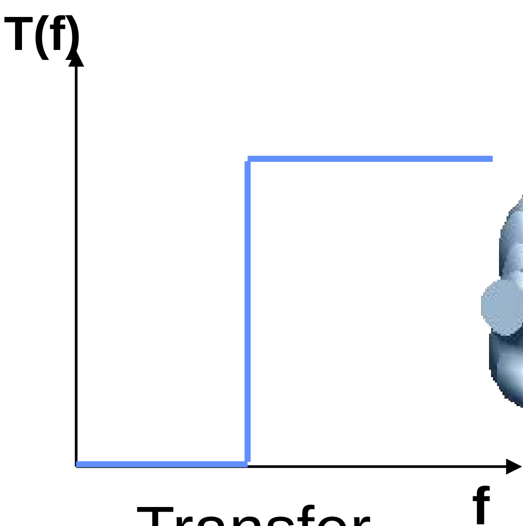
raw data along a ray

Transfer function

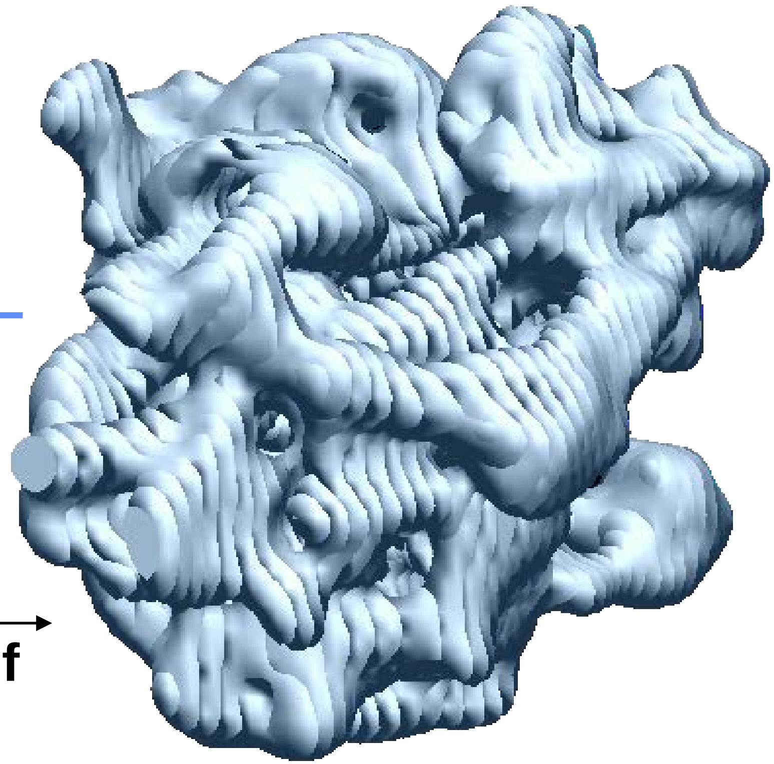
Transfer function applied to data

High-Frequency TFs

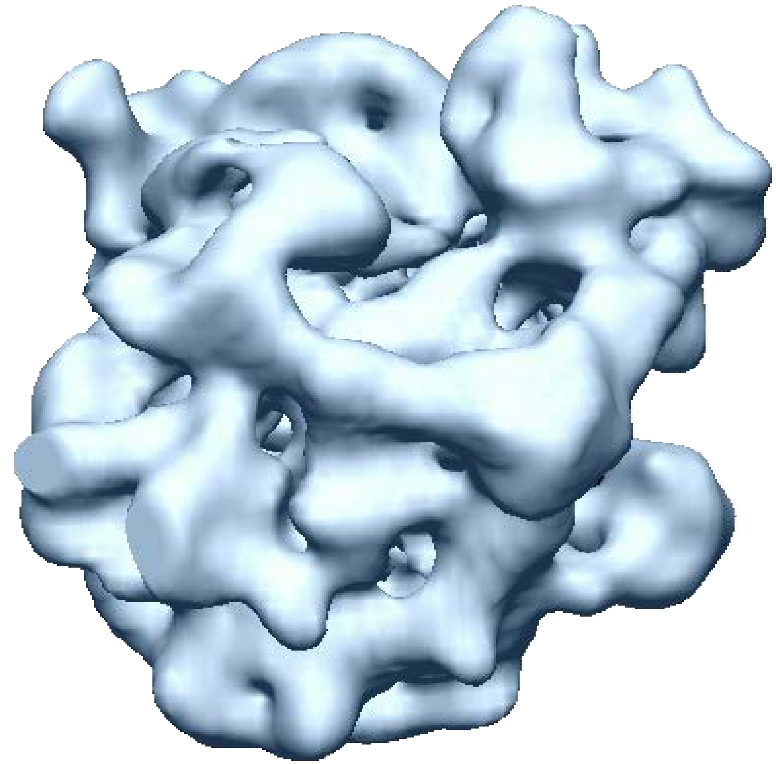
Cryoelectron-microscopic volume
Isosurface of Escherichia Coli Ribosome at 18 Ångström



Transfer function

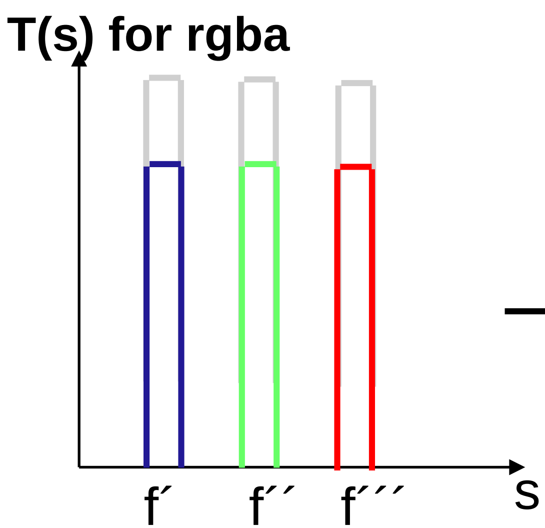


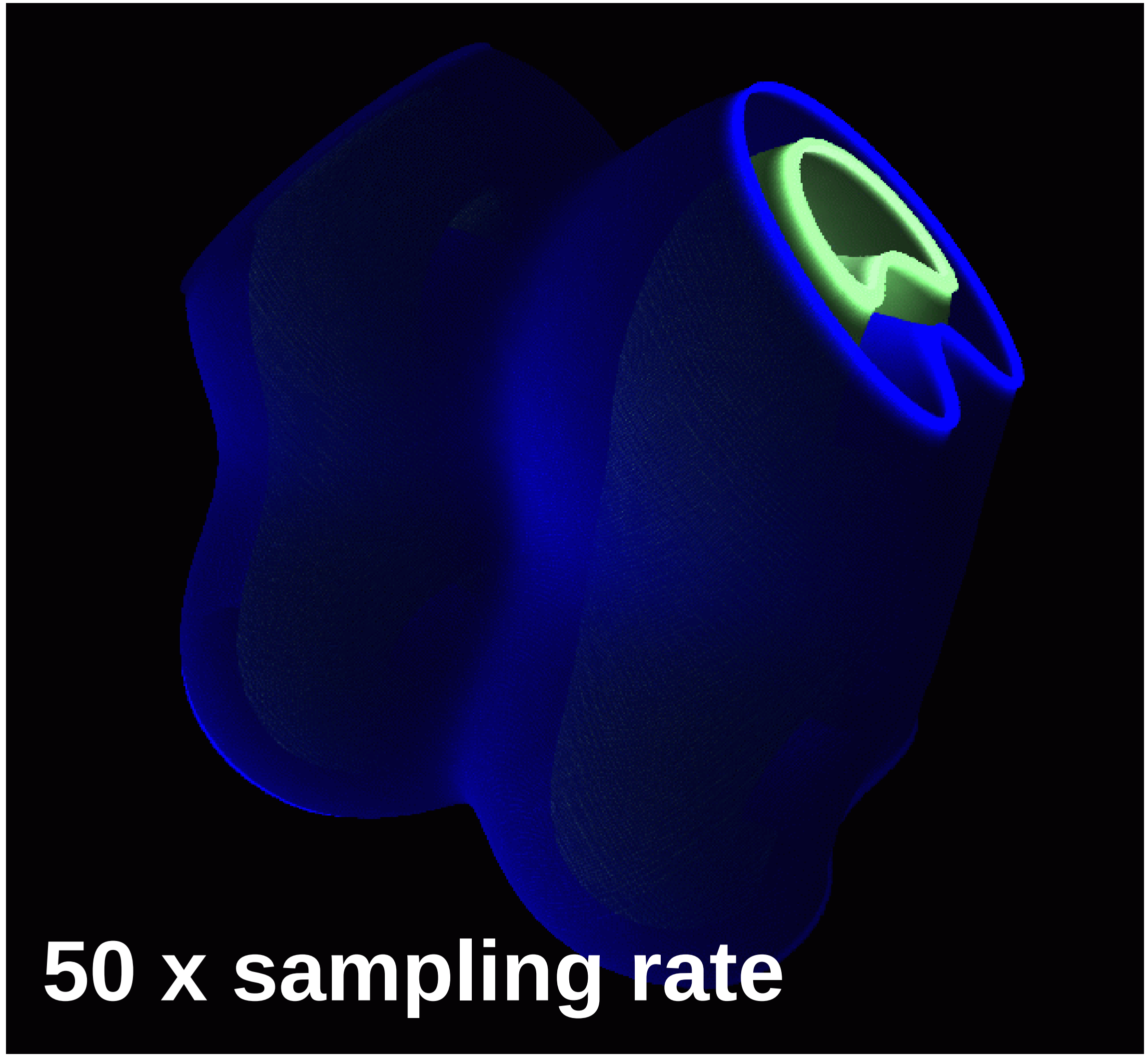
64 data slices



10 times more slices

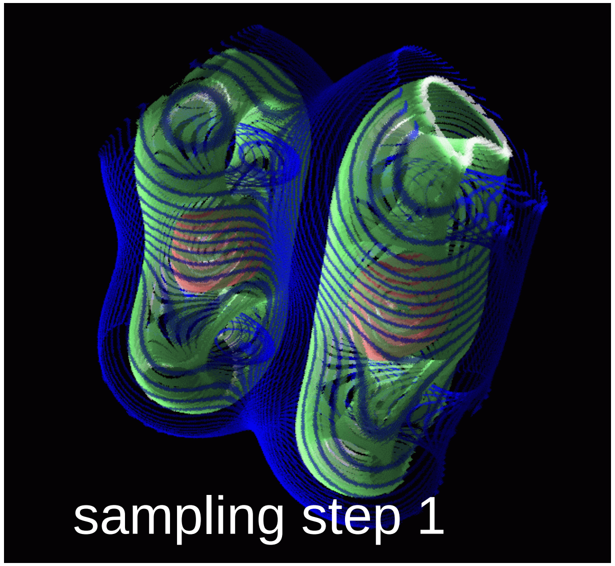
Iso-Surface Ray Casting

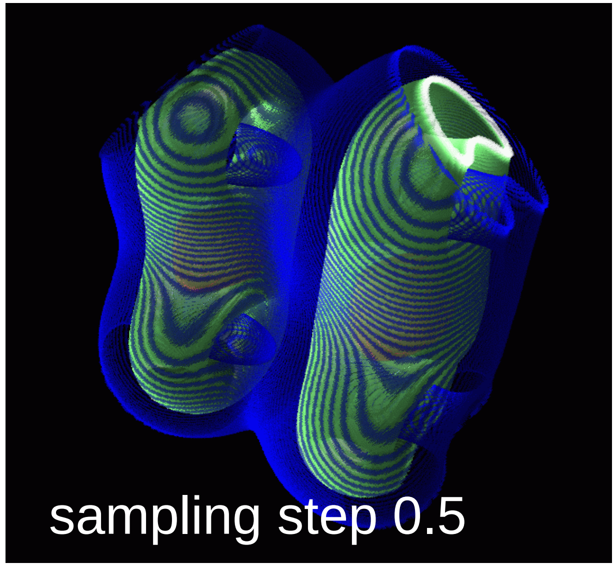


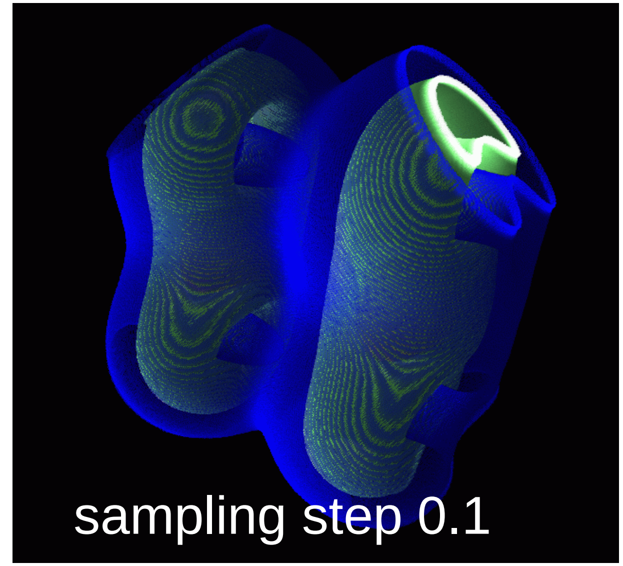


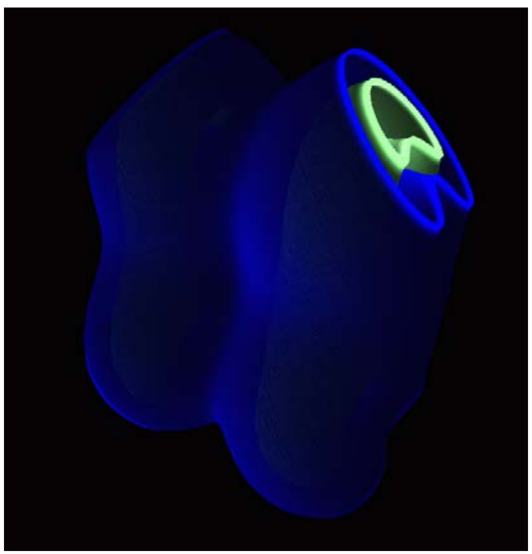
Transfer function T(f) creates iso-surfaces for f´, f´´, and f´´´ requires high sampling rate

Iso-Surface Ray Casting
Influence of Sampling Rate





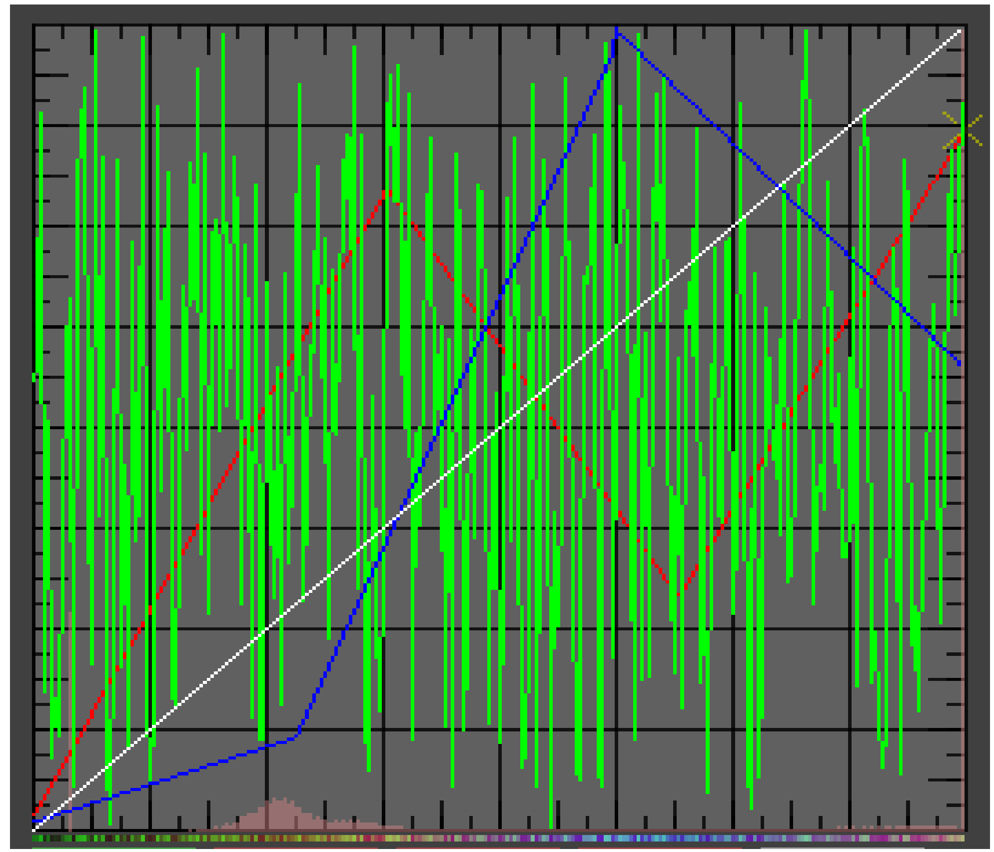




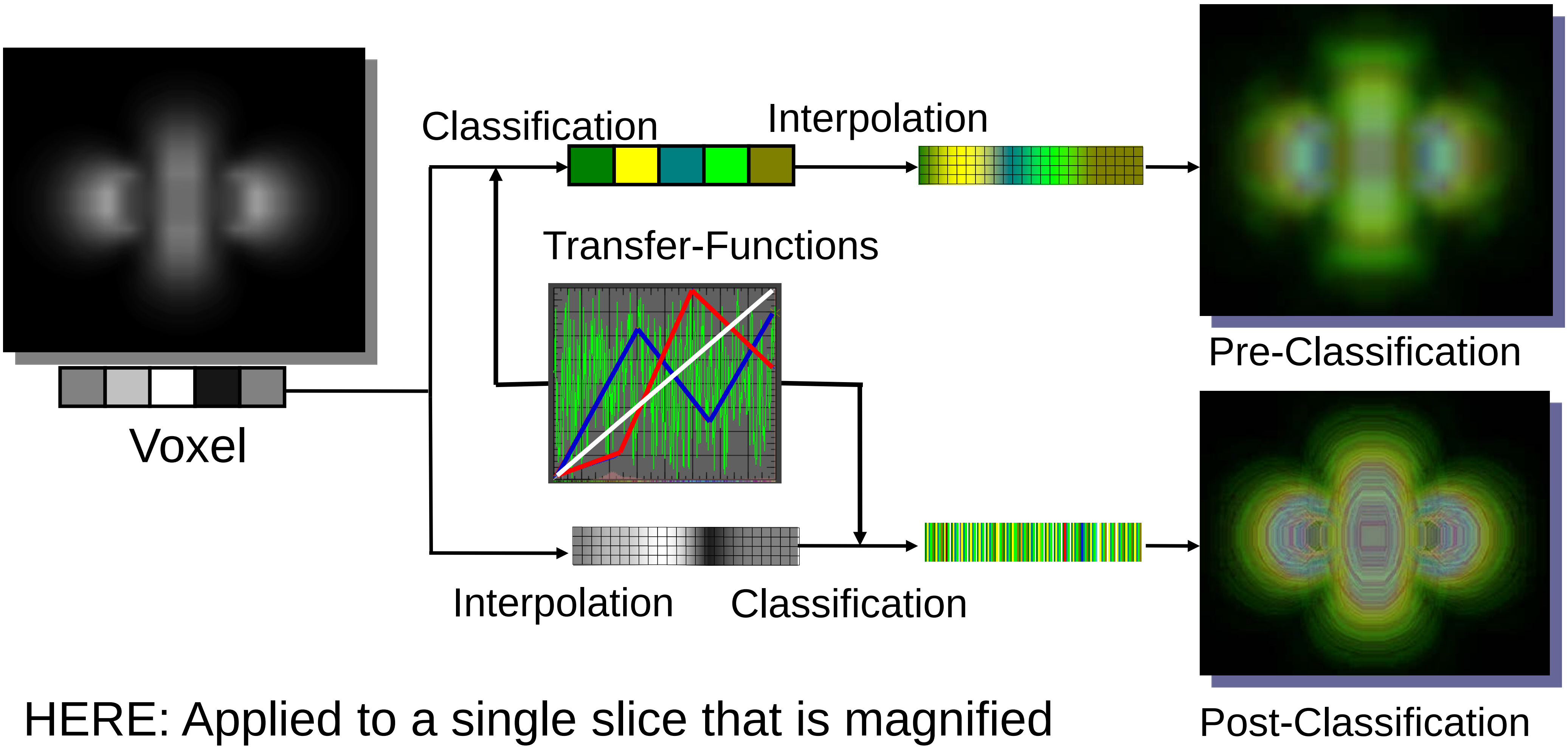
sampling step 0.02 – 50x

High-Frequency TFs

- **Red: piecewise linear**
- **Green: random**
- **Blue: piecewise linear**
- **Alpha: identity**

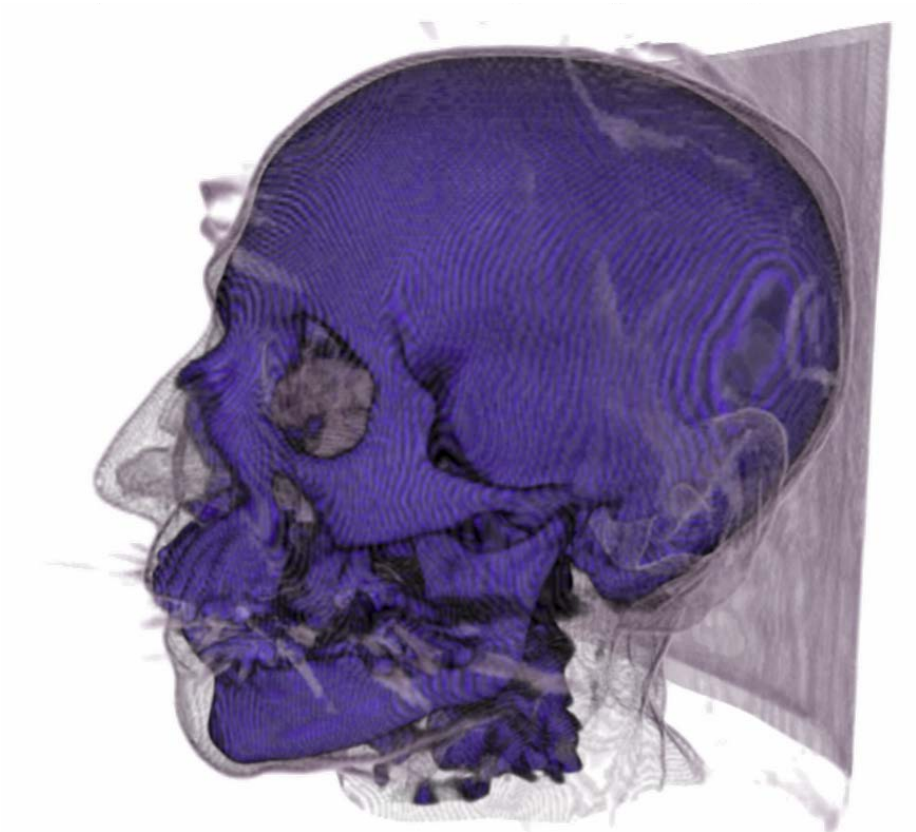


Pre- and Post-Classification

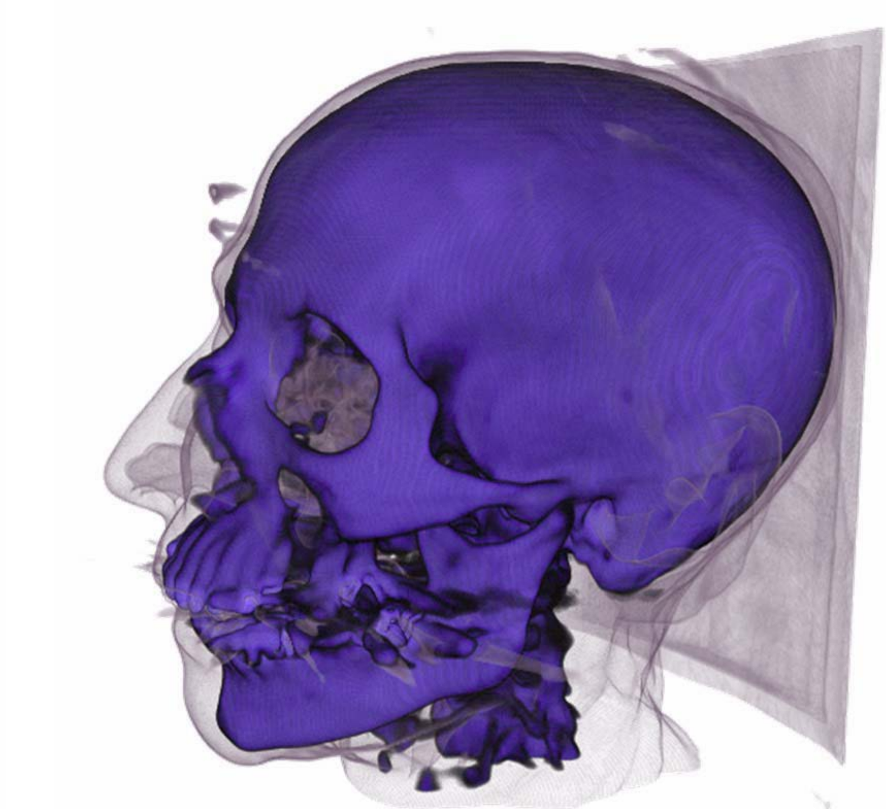


HERE: Applied to a single slice that is magnified
such that one voxel covers multiple pixels

Quality: Pre-Classification vs. Post-Classification



pre-classification

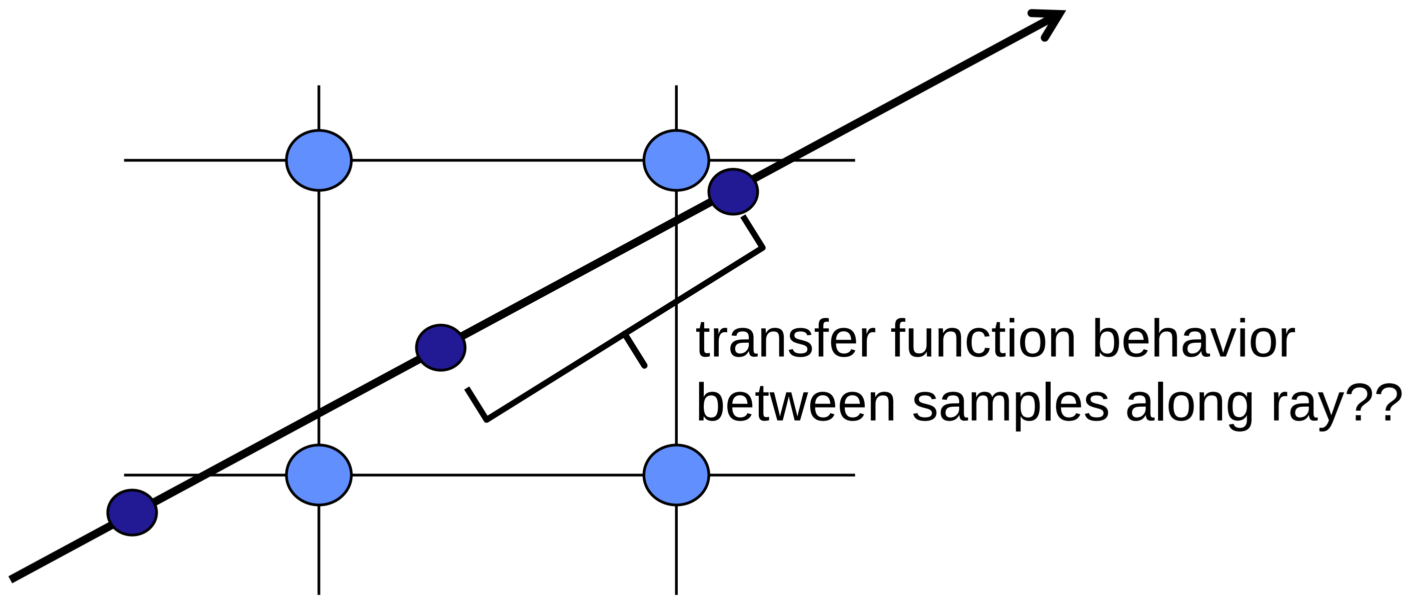


post-classification

Same transfer function, same resolution, same sampling rate

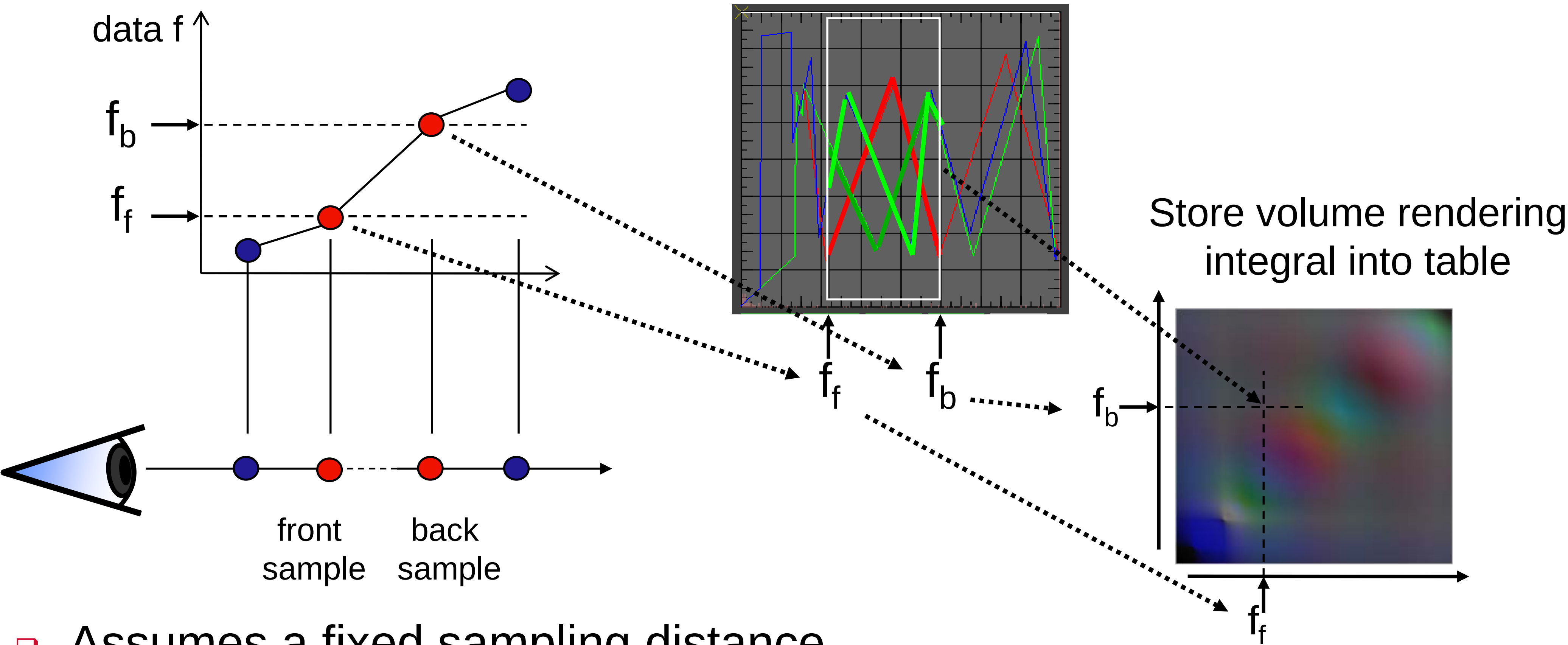
High-Frequency TFs

- Pre-Classification
  - No high-frequencies from TF
- Post-Classification
  - Reproduces high-frequencies of the transfer function at sample points
  - To reproduce high frequencies between sample points => increase sampling rate



Pre-Integrated Classification

Precompute the volume rendering integrals of all possible combinations of data values considering the transfer function and assuming a linear change of data values between samples f_f and f_b



- Assumes a fixed sampling distance

Computation of Pre-Integration Tables

- Need to compute entries of 2D pre-integration table for colors and opacities

$$\alpha(f_f, f_b) = 1 - \exp(-d\int_0^1 \kappa((1-\omega) f_f + \omega f_b) d\omega)$$

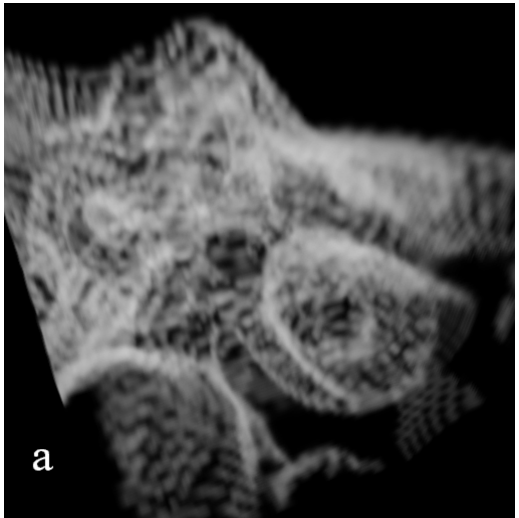
$$I(f_f, f_b) = \int_0^1 C((1-\omega) f_f + \omega f_b)\kappa((1-\omega) f_f + \omega f_b)$$
$$\exp(-\omega d\int_0^{\omega} \kappa((1-\varpi) f_f + \varpi f_b) d\varpi) d\omega$$

- C(f) and κ(f) are the transfer functions to obtain a color and absorption value corresponding to the data value f
- d is the actual sample distance between f_f and f_b
- Use numerical integration or simply compositing based on very small steps – needs opacity correction!
- Assumes linear change of the data value between f_f and f_b

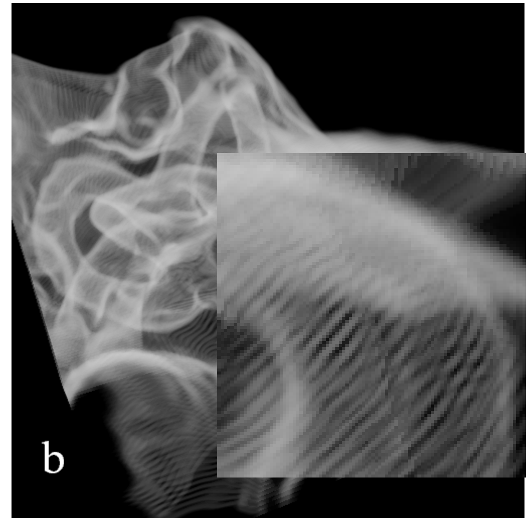
Pre-Integration is Expensive

- Pre-integration for each transfer function may take some seconds => no interactive change of TF
  - Use hardware acceleration
- Pre-integration table large
  - 12bit x 12bit x RGBA8 = 4096x4096x32bit = 64MB
- Alternative solution: use integral functions
  - Less precise
- Engel K., Kraus M., Ertl T.: High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In Proceedings of Eurographics/SIGGRAPH Workshop on Graphics Hardware (2001), pp. 9–16. http://portal.acm.org/citation.cfm?id=383515
- High-Quality Lighting and Efficient Pre-Integration for Volume Rendering Eric Lum, Brett Wilson, and Kwan-Liu Ma. In Proceedings of Joint Eurographics-IEEE TVCG Symposium on Visualization May, 2004, pp. 25-34 http://maxradi.us/documents/preintegration/

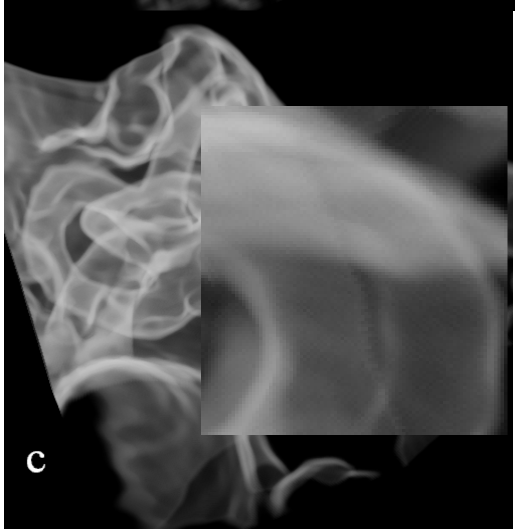
Pre-Integrated Volume Rendering



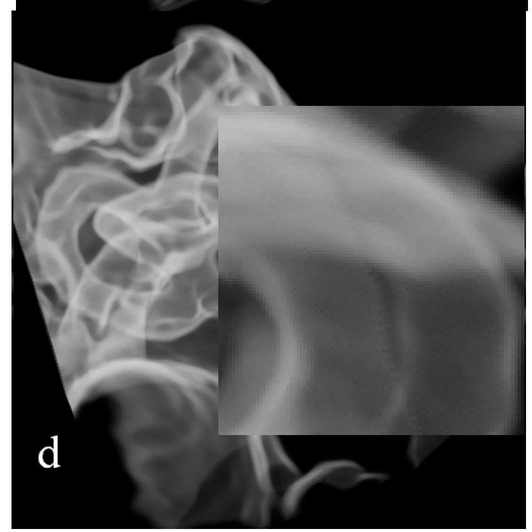
128 slices
pre-
classification



128 slices
post-
classification



284 slices
post-
classification



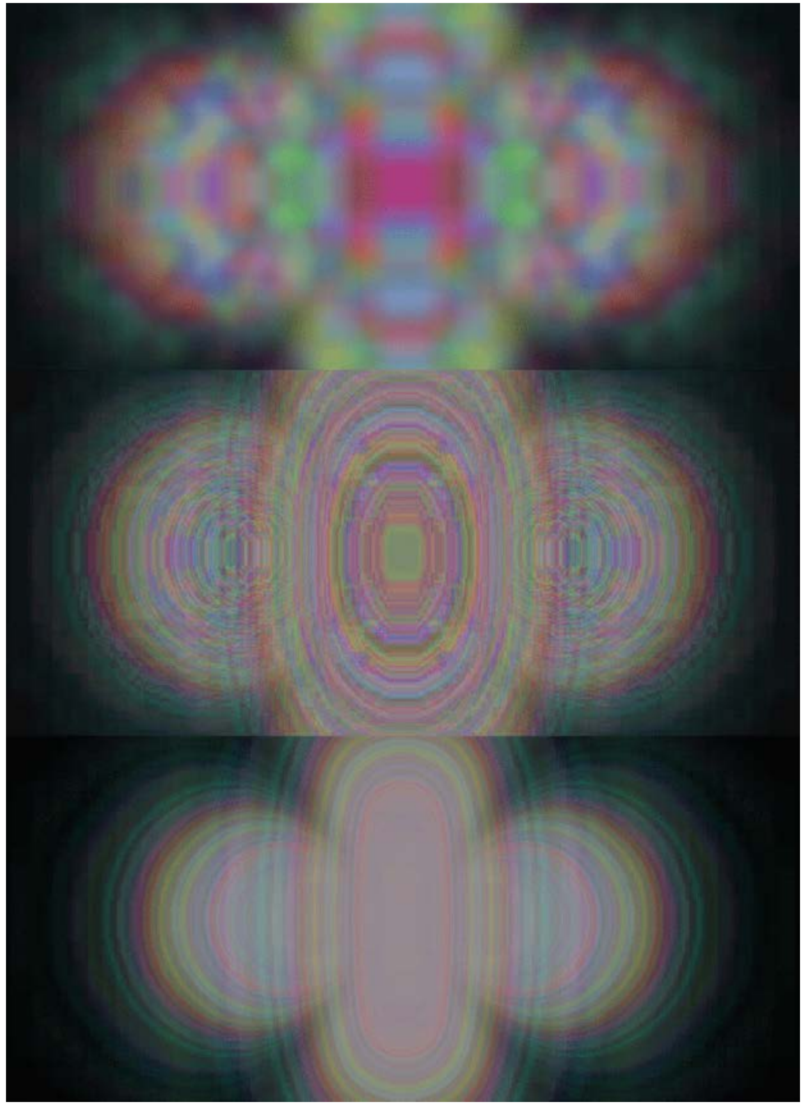
128 slices
pre-integrated

Pre-Integrated Volume Rendering

Pre-Classification

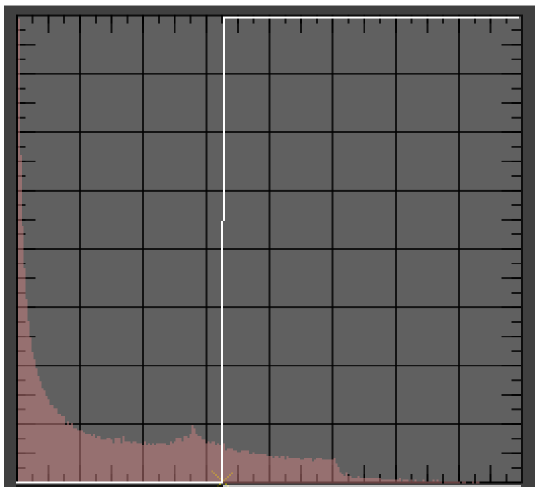
Post-Classification

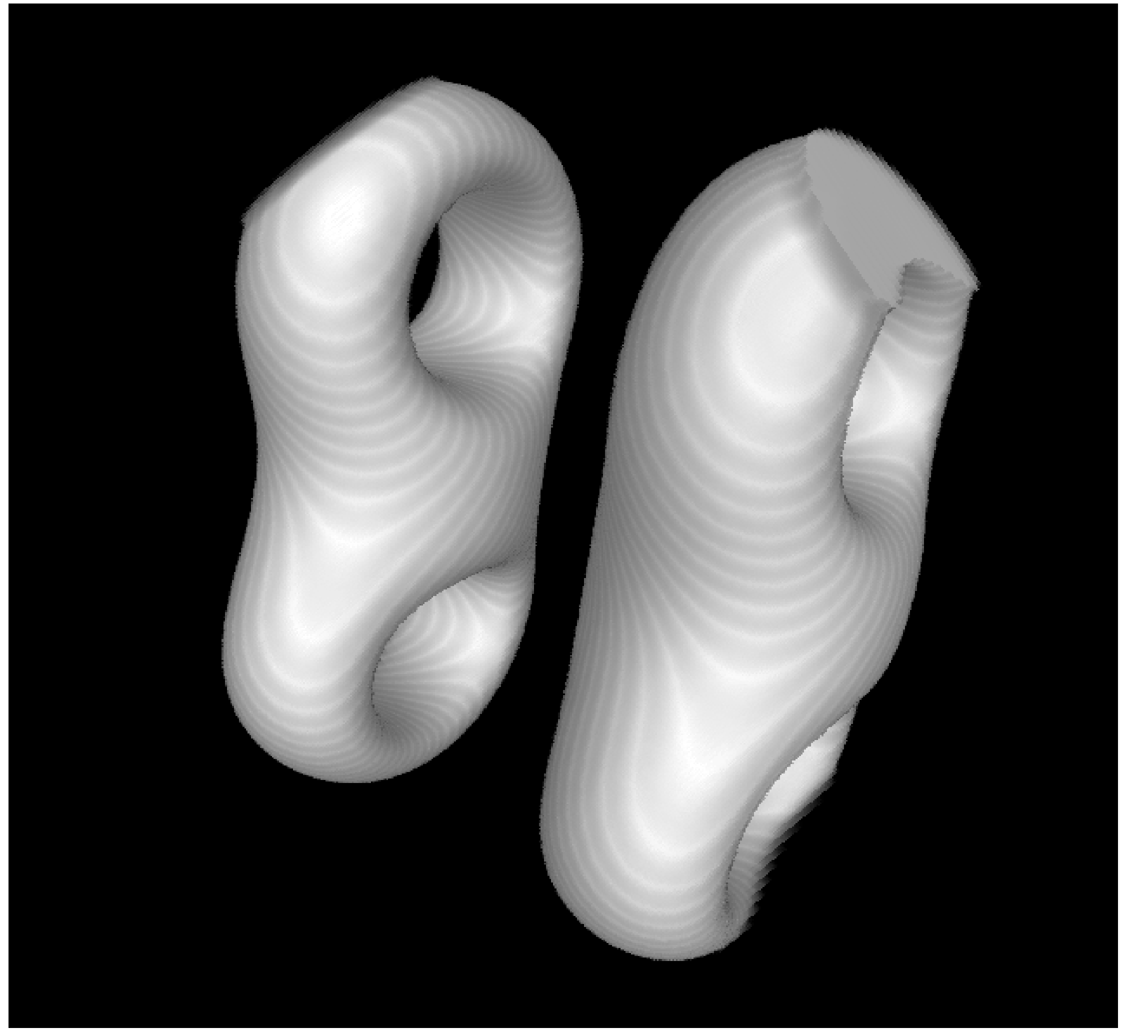
Pre-Integrated-
Classification



Pre-Integrated Volume Rendering

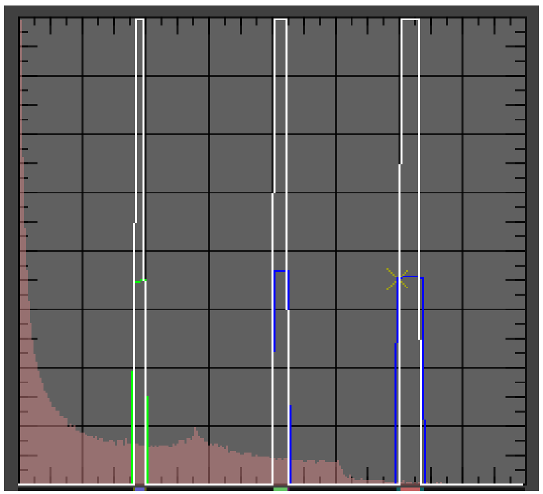
single peak

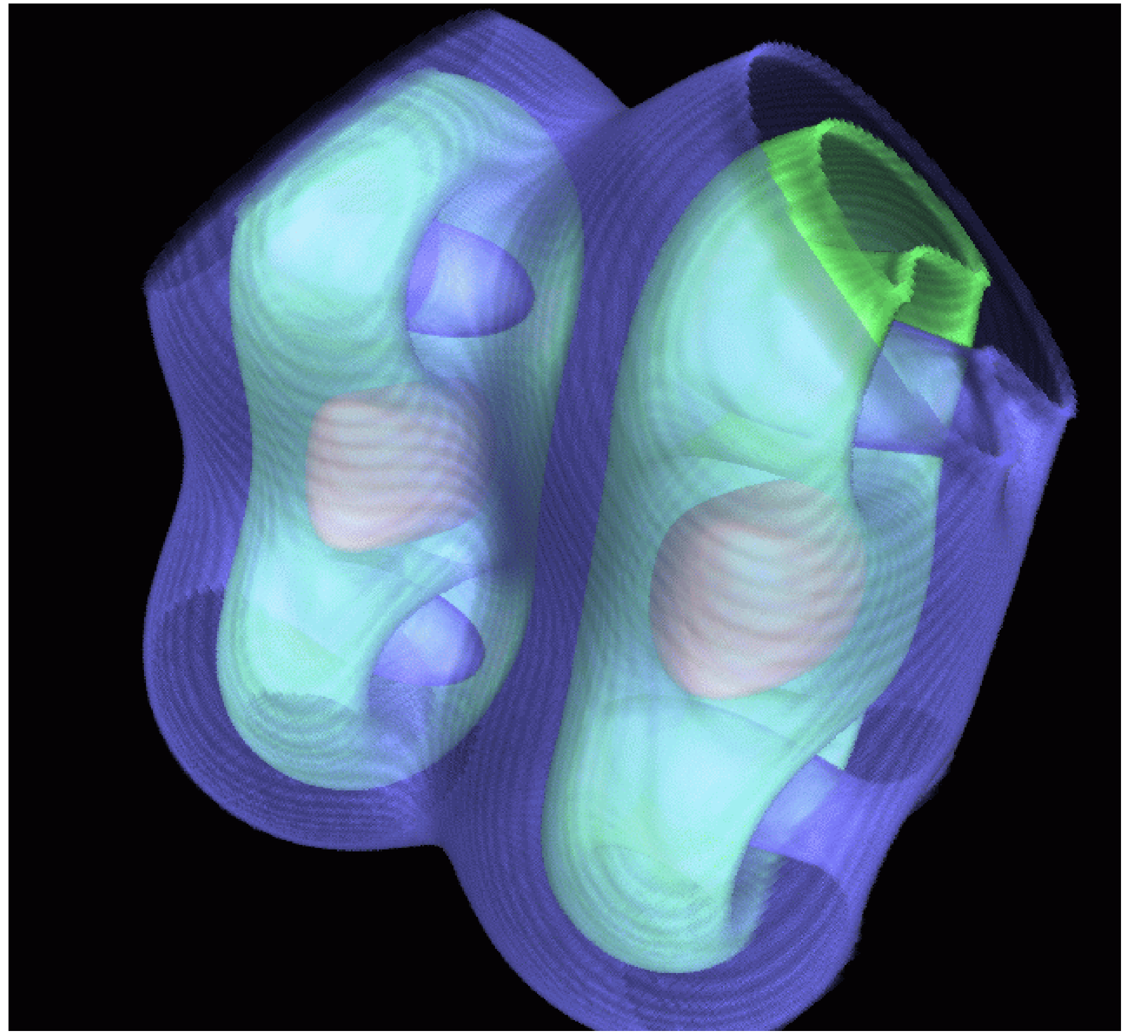




Pre-Integrated Volume Rendering

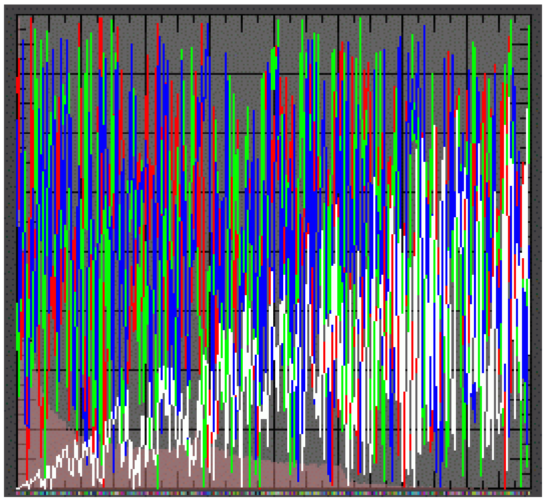
multiple peaks

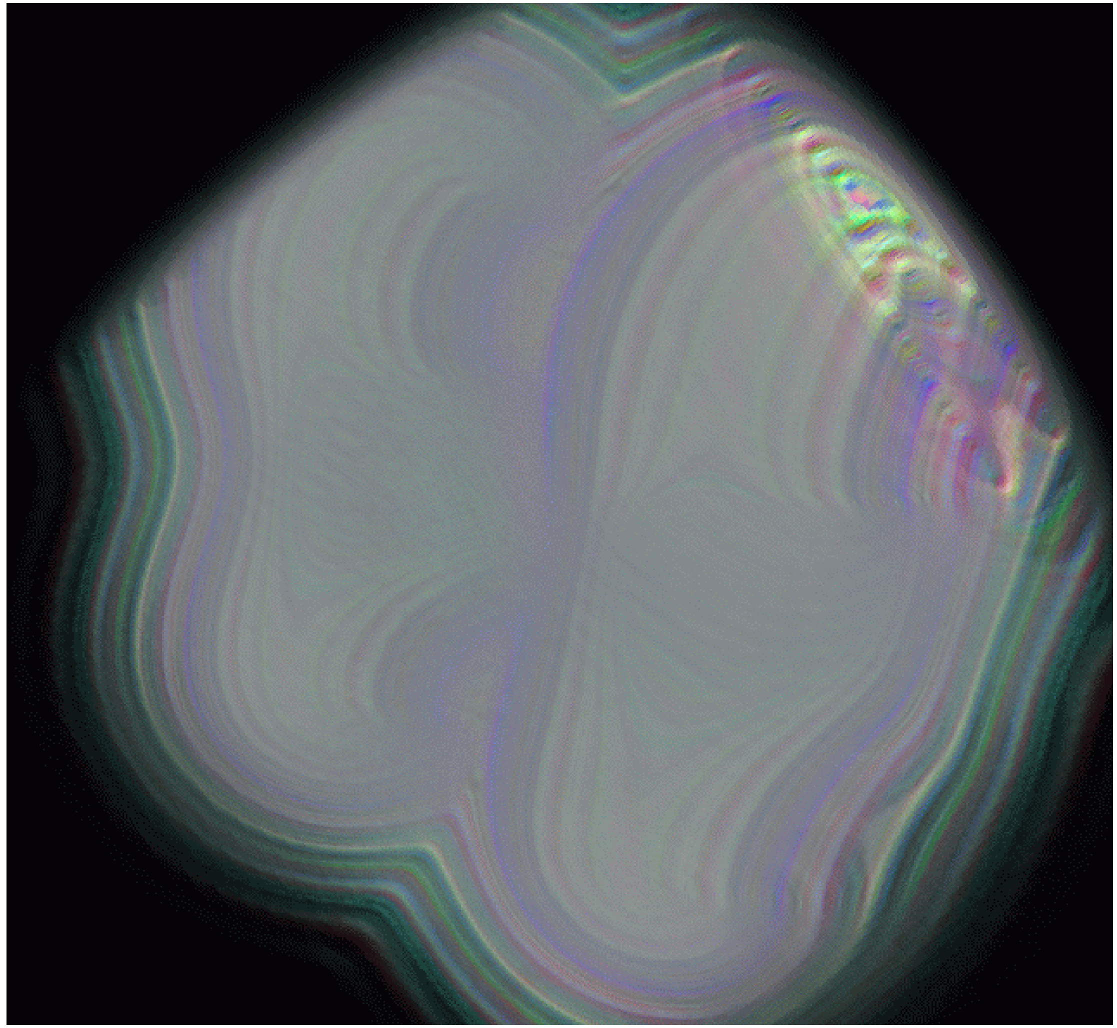




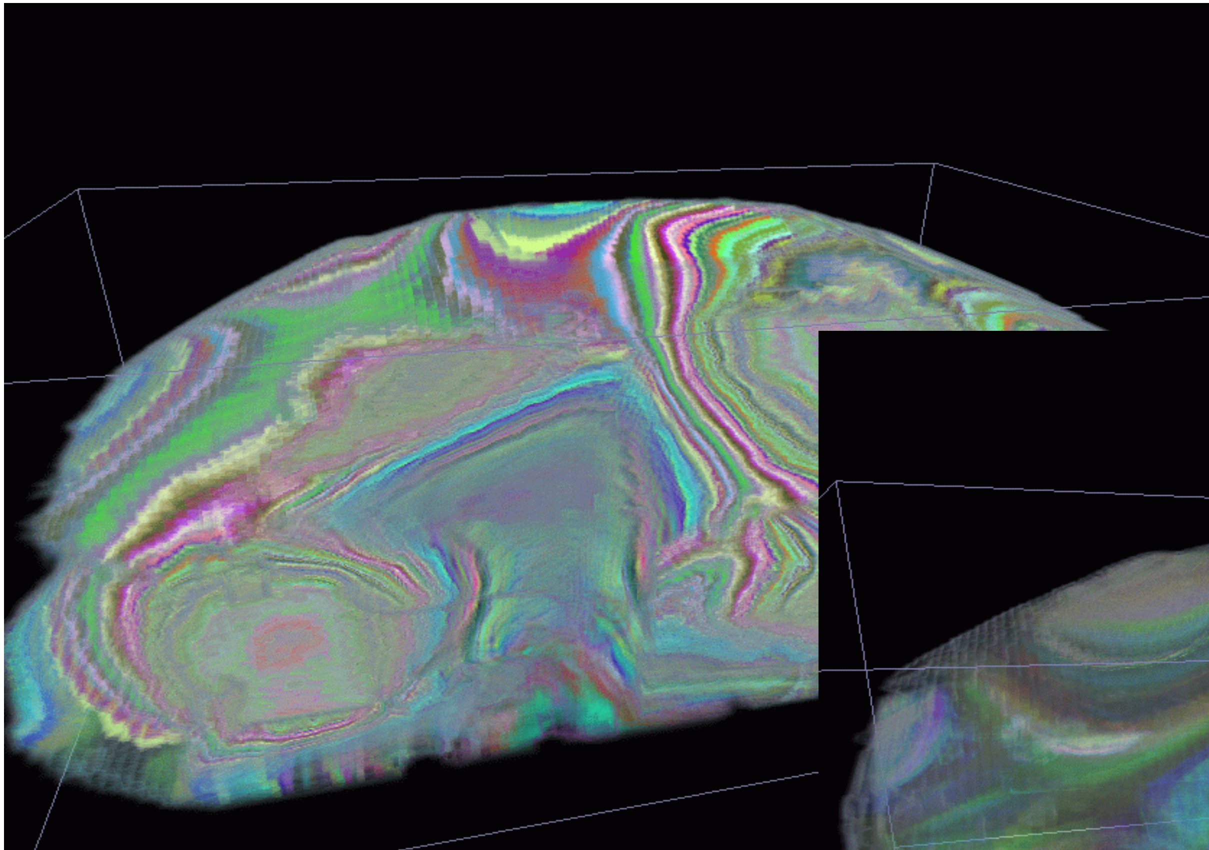
Pre-Integrated Volume Rendering

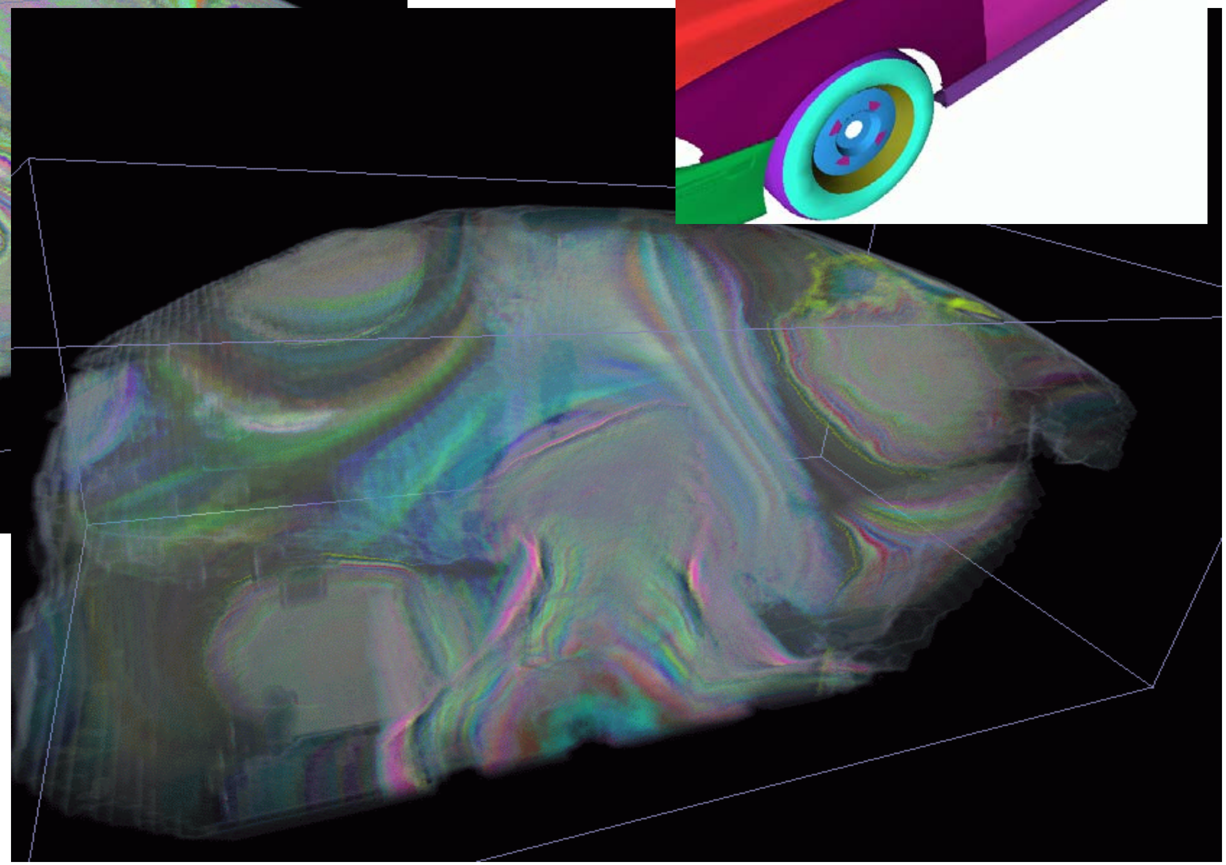
many peaks

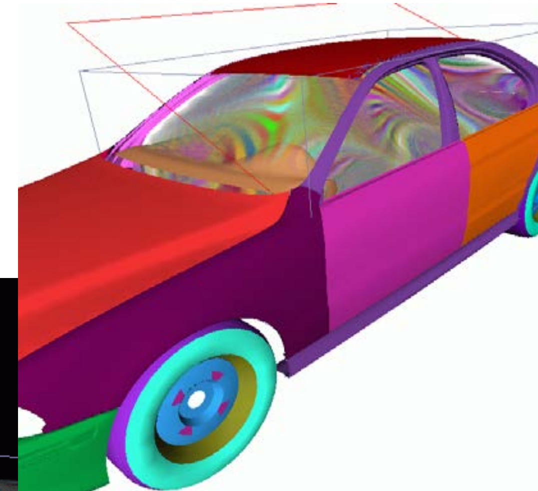




Pre-Integrated Volume Rendering

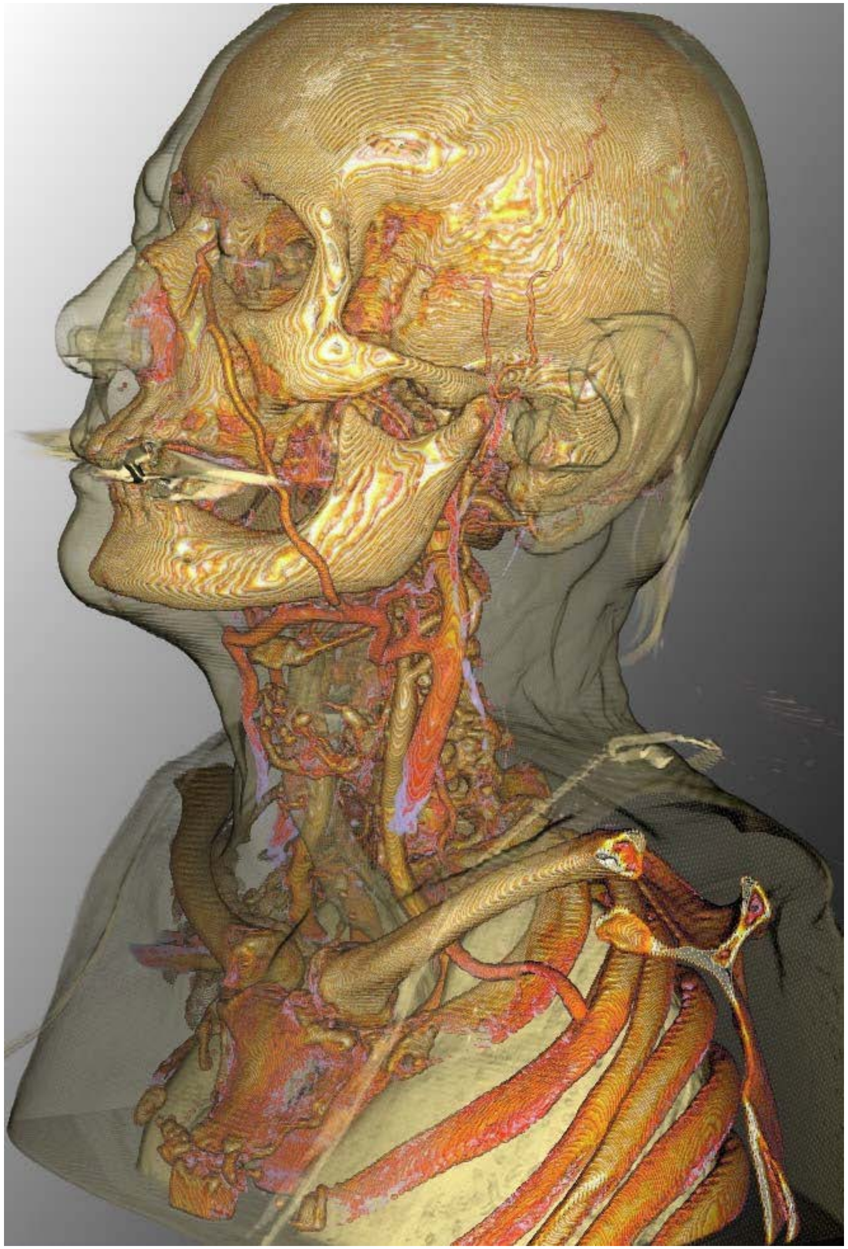


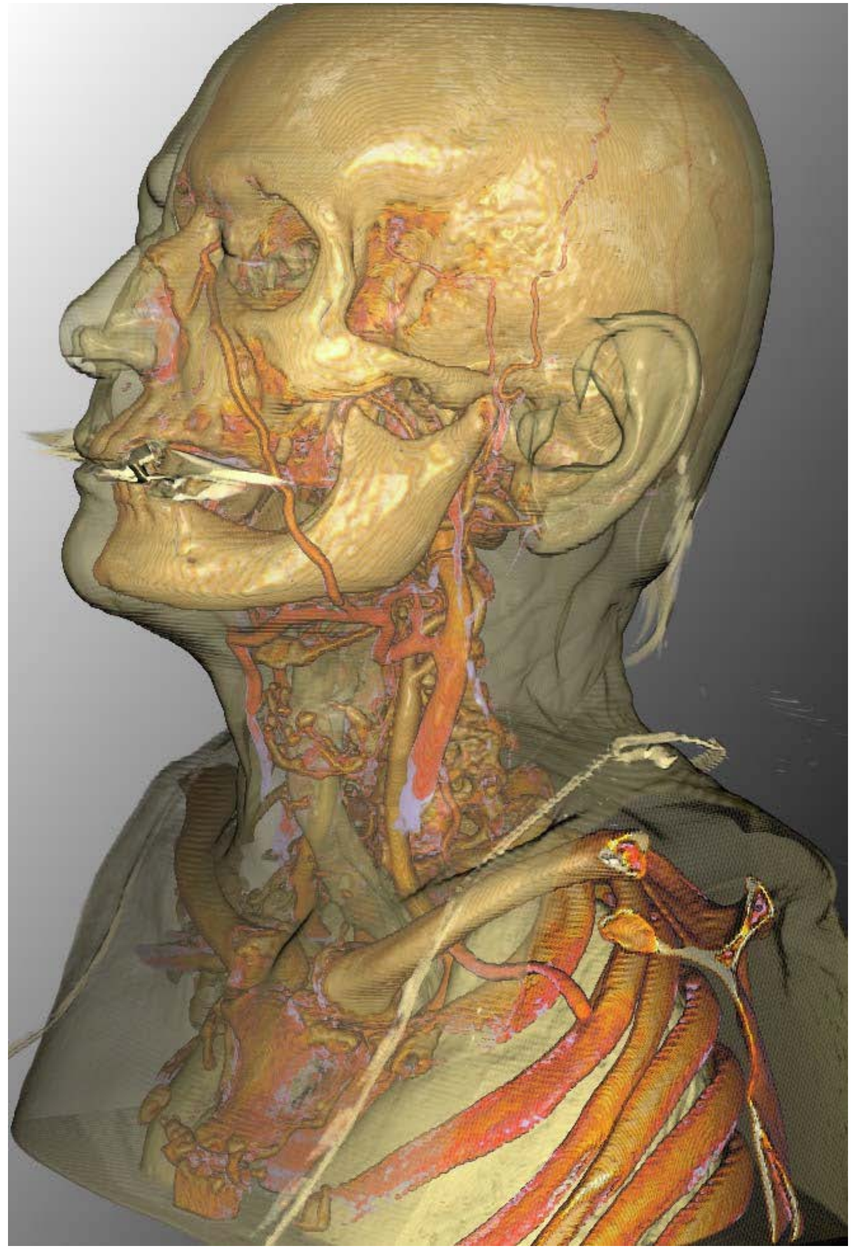




noise volume inside car

each layer may represent one db change





High-Frequency Transfer Functions

- Pre-classification
  - No high-frequencies from TF
- Post-classification
  - Reproduces high-frequencies of the transfer function at the sample points
  - To reproduce high frequencies between sample points => increase sampling rate
- Pre-integration
  - Reproduces high-frequencies of the transfer function between sample points under the assumption that the data changes linearly between sample points
  - To reproduce high frequencies in the data between sample points => increase sampling rate

End

Visualization

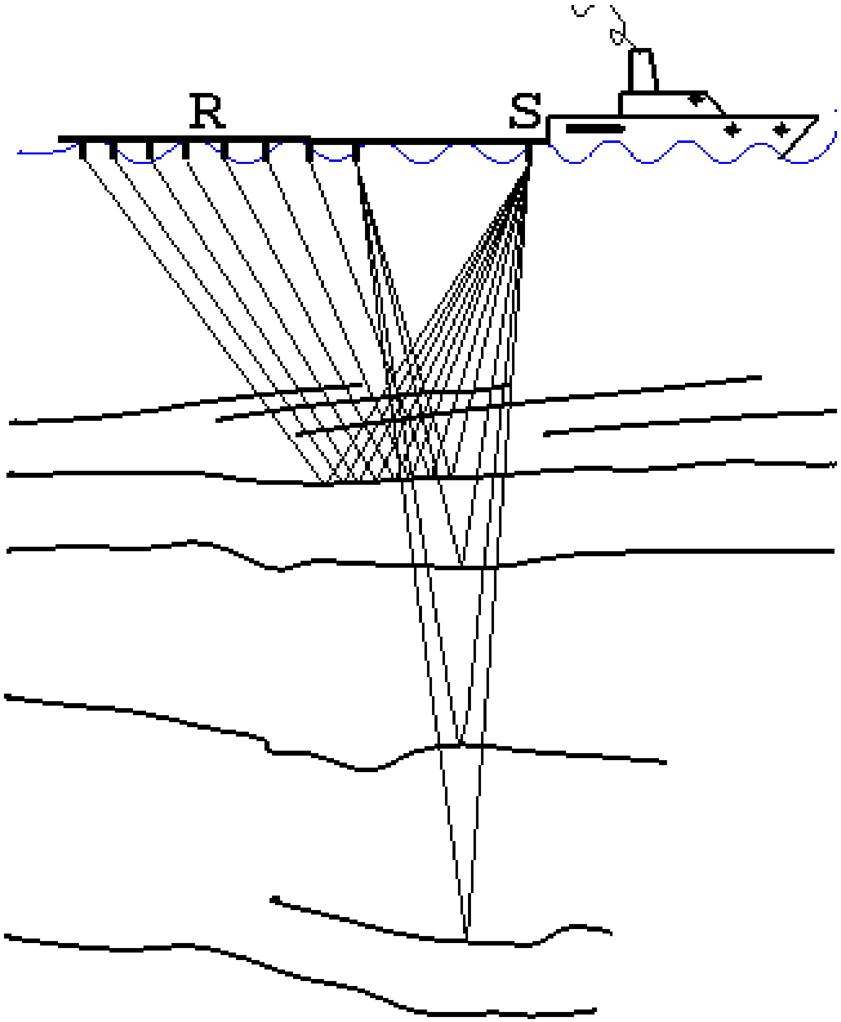
Multi-Resolution Volume Rendering

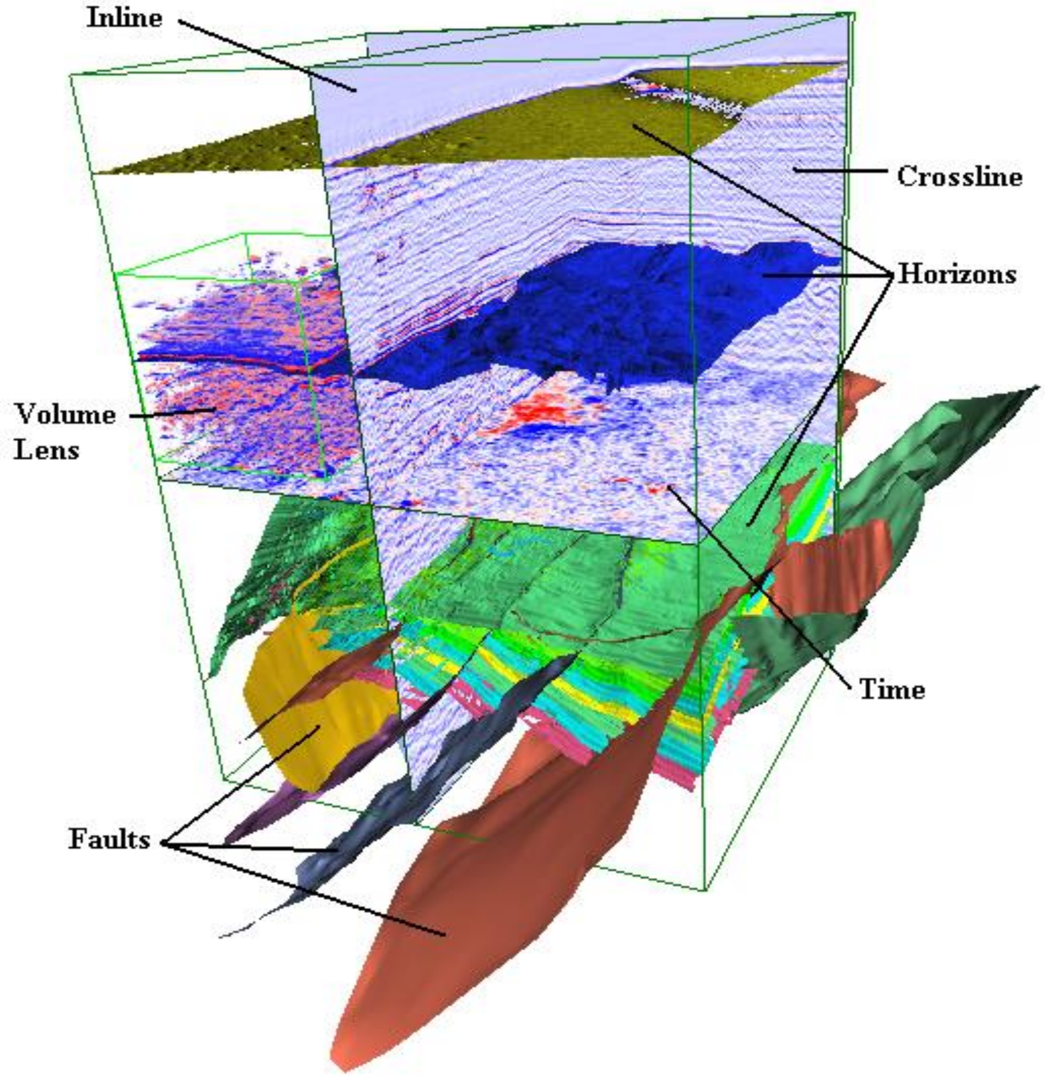
SS2015

Bernd Fröhlich

Virtual Reality Systems
Faculty of Media
Bauhaus-Universität Weimar

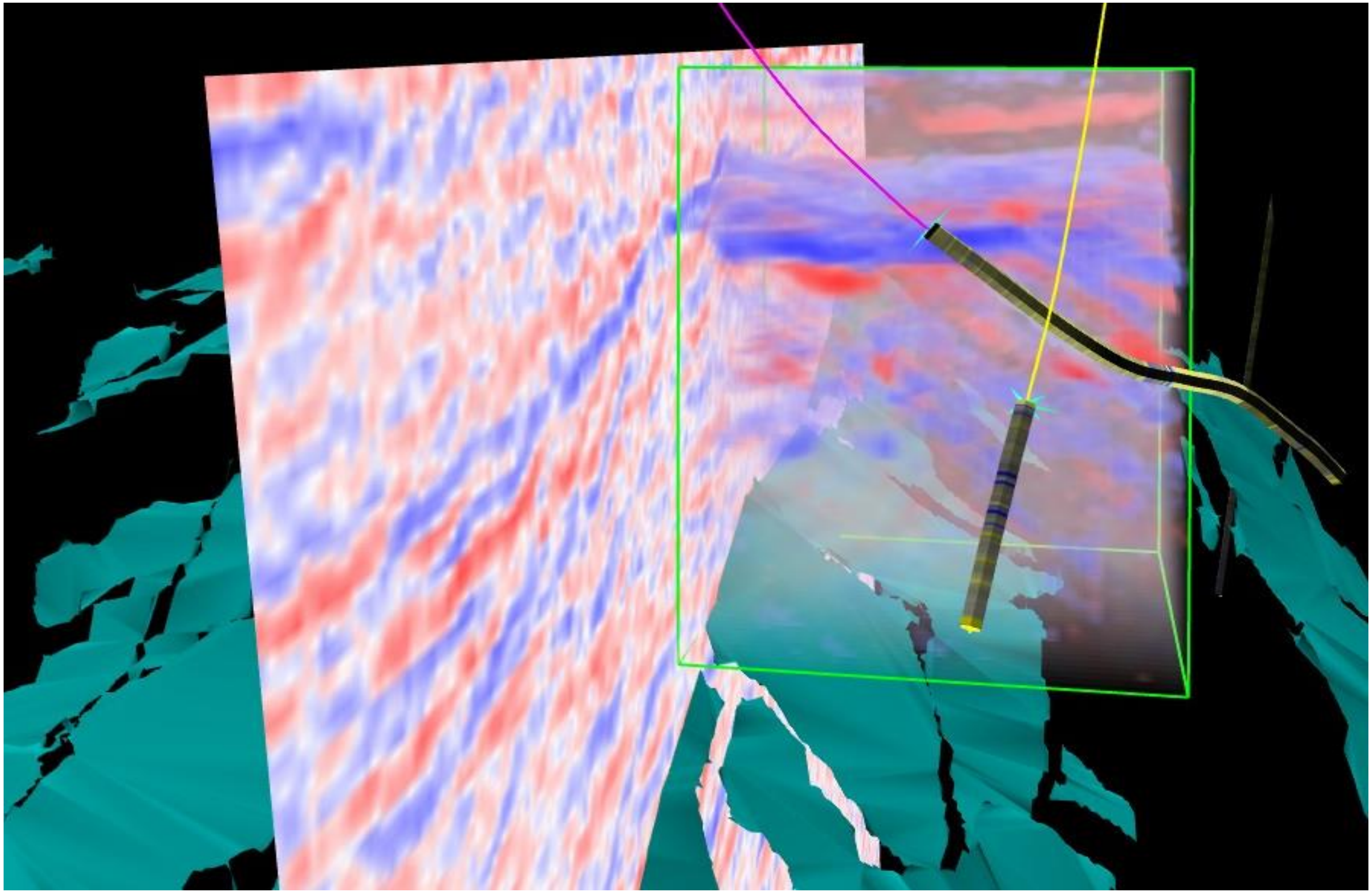
Large Volumes





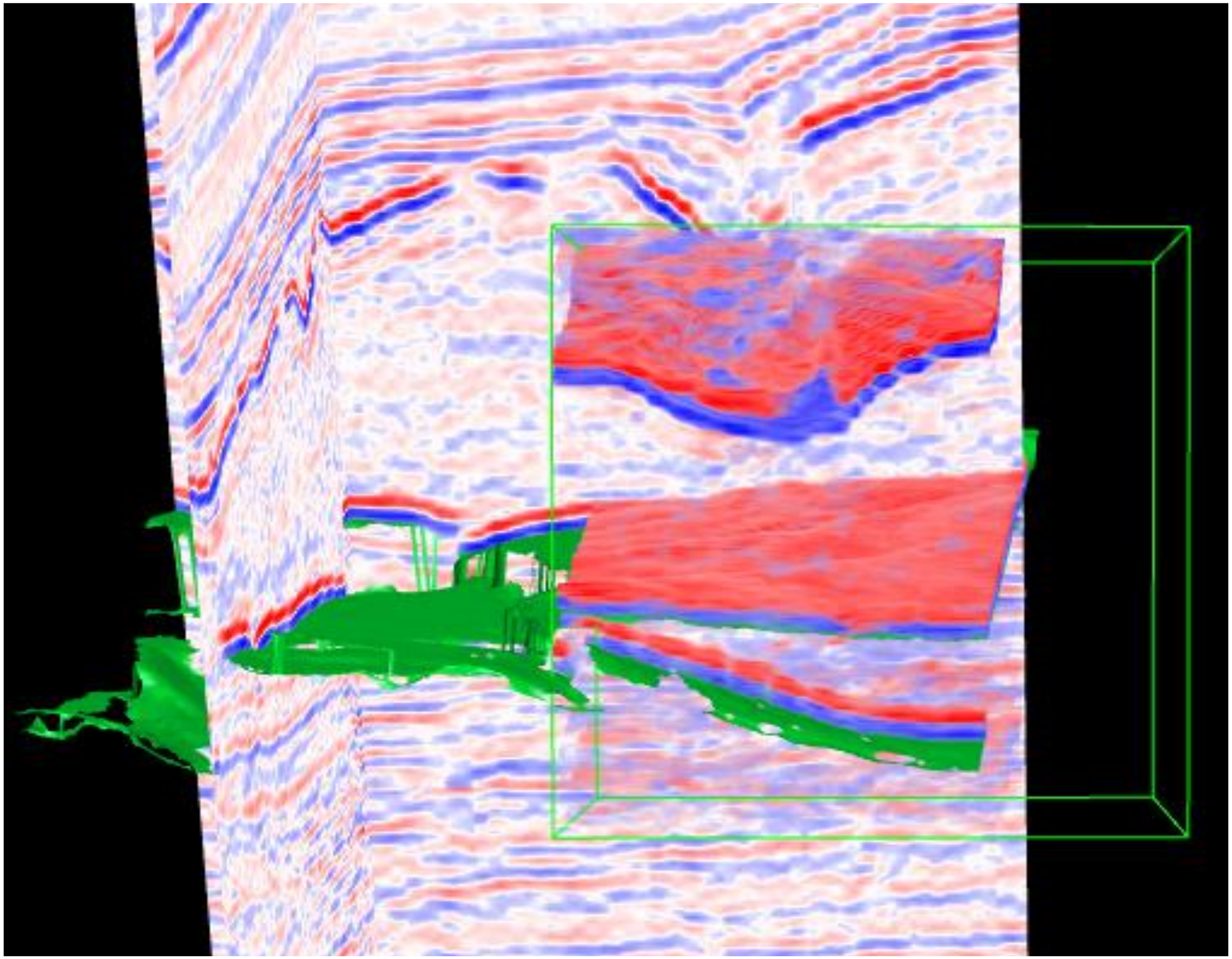
VRGeo consortium: vrgeo.com

Roaming Through Large Volumes

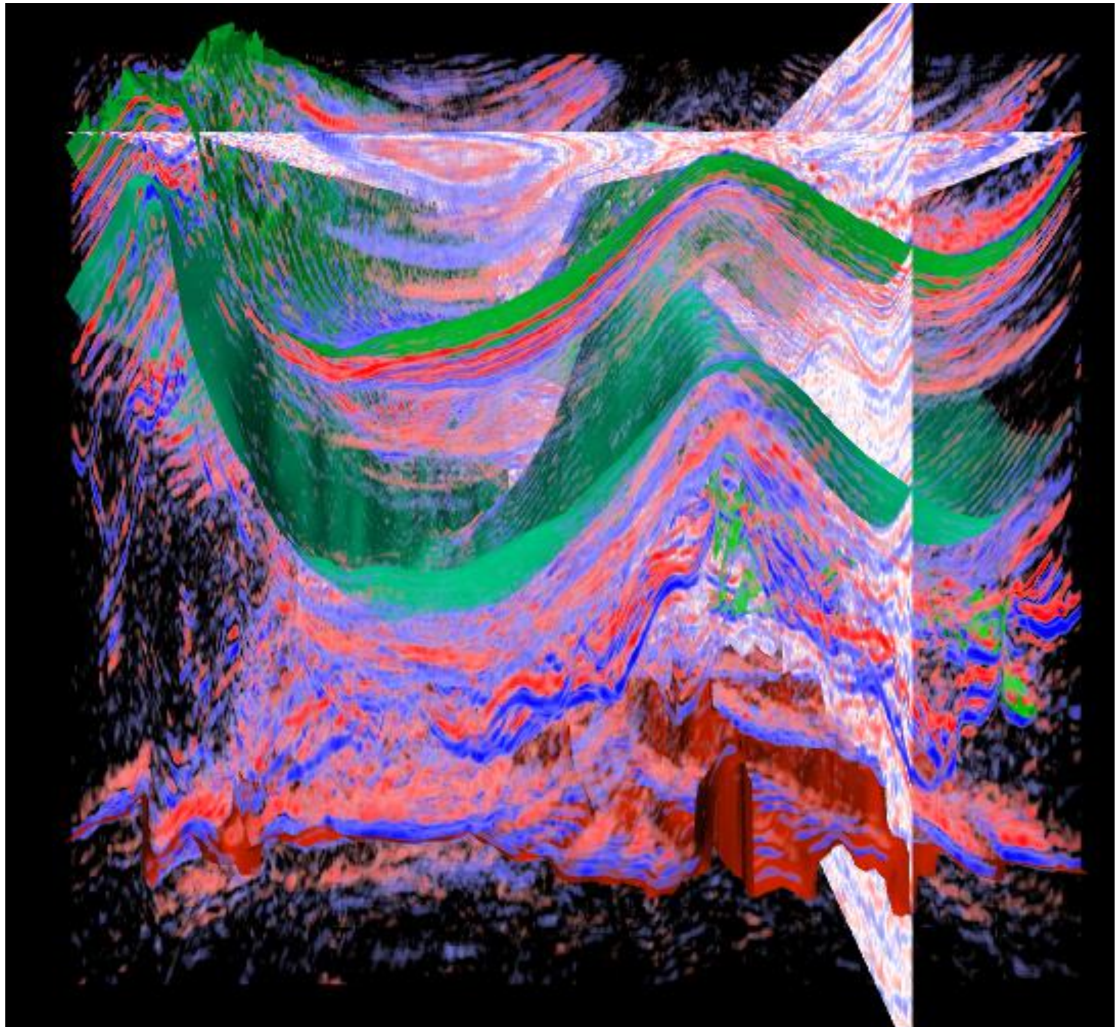


Move lens through volume

Iso Surface Approximations

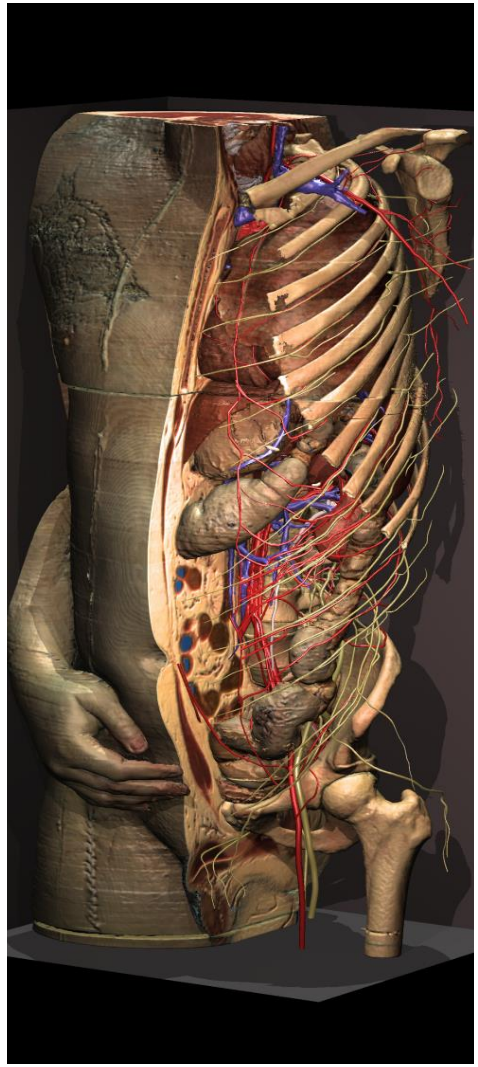


An Overview of a Large Volume



Problem

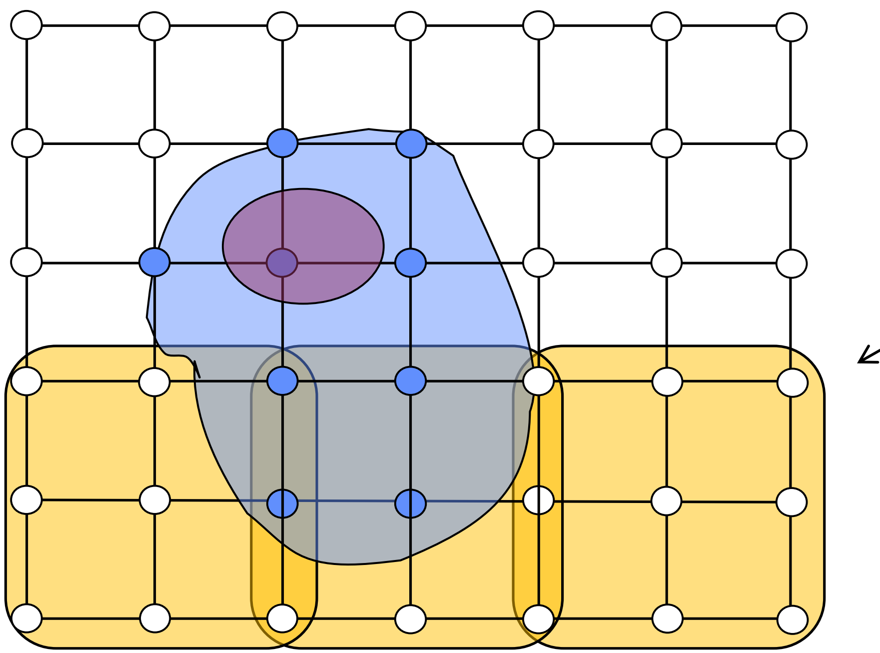
- Entire volume data does not fit
  - into graphics card's memory or
  - even into main memory
- Approach
  - Multi-resolution techniques and paging
  - Select a level of detail for each volume region and perform rendering based on this representation



Visible Human 50GB

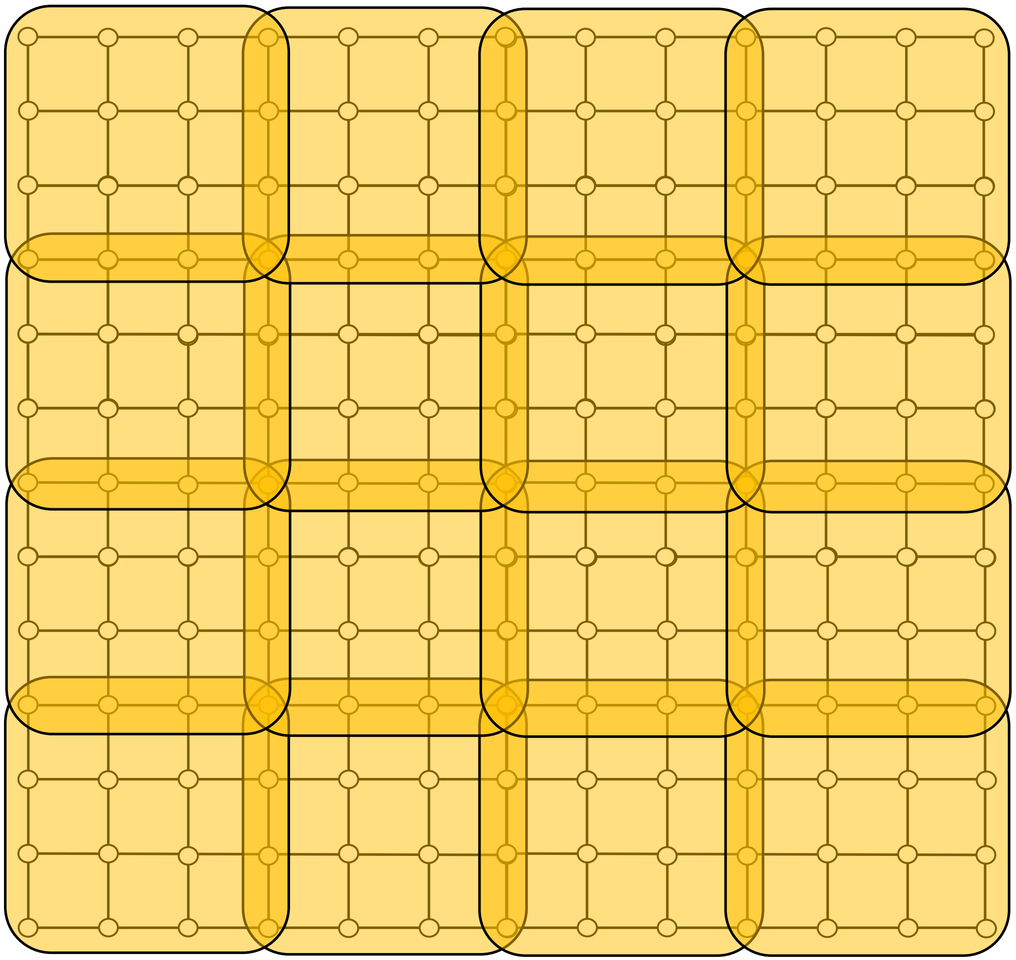
Bricks

- A brick is a set of adjacent voxels, typically consisting of 16x16x16, 32x32x32 or 64x64x64 voxels. The entire volume can be split into bricks. Bricks overlap by at least one row/column to allow for linear interpolation.



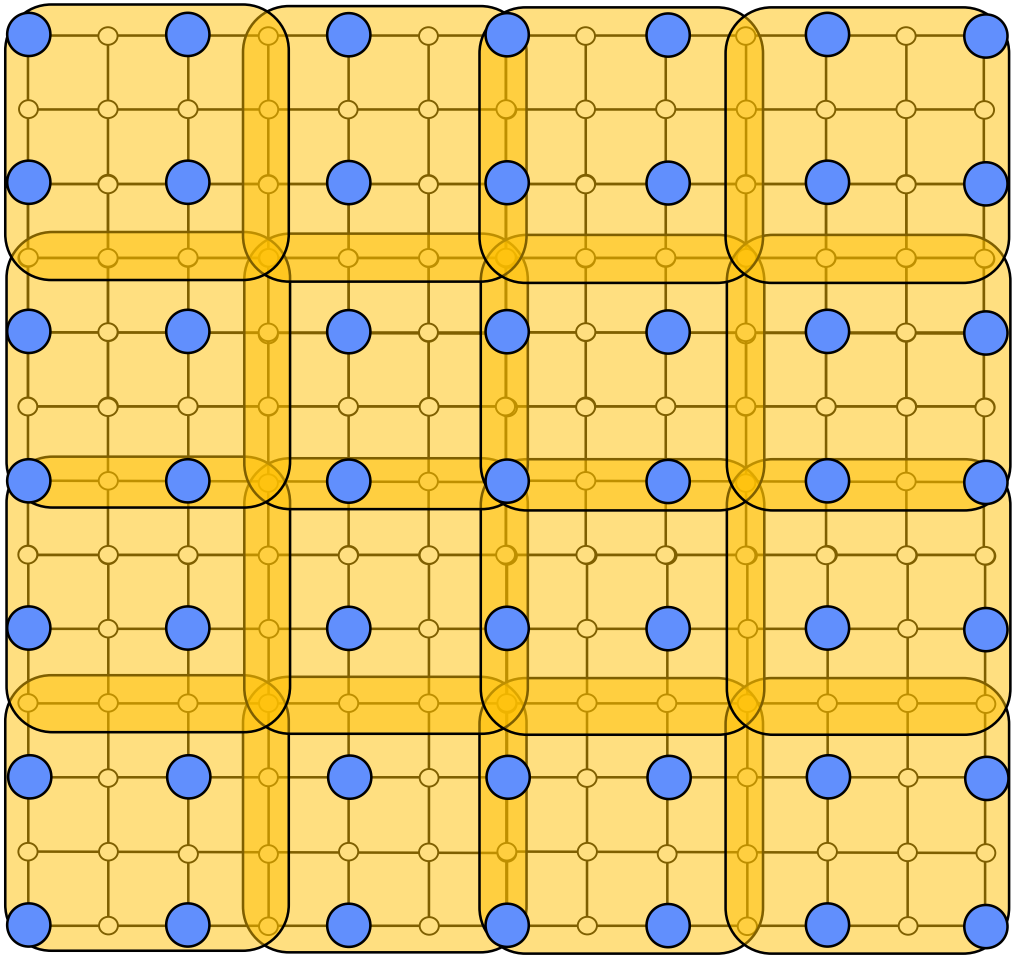
brick of 3x3 voxels

Octree Generation Bottom up



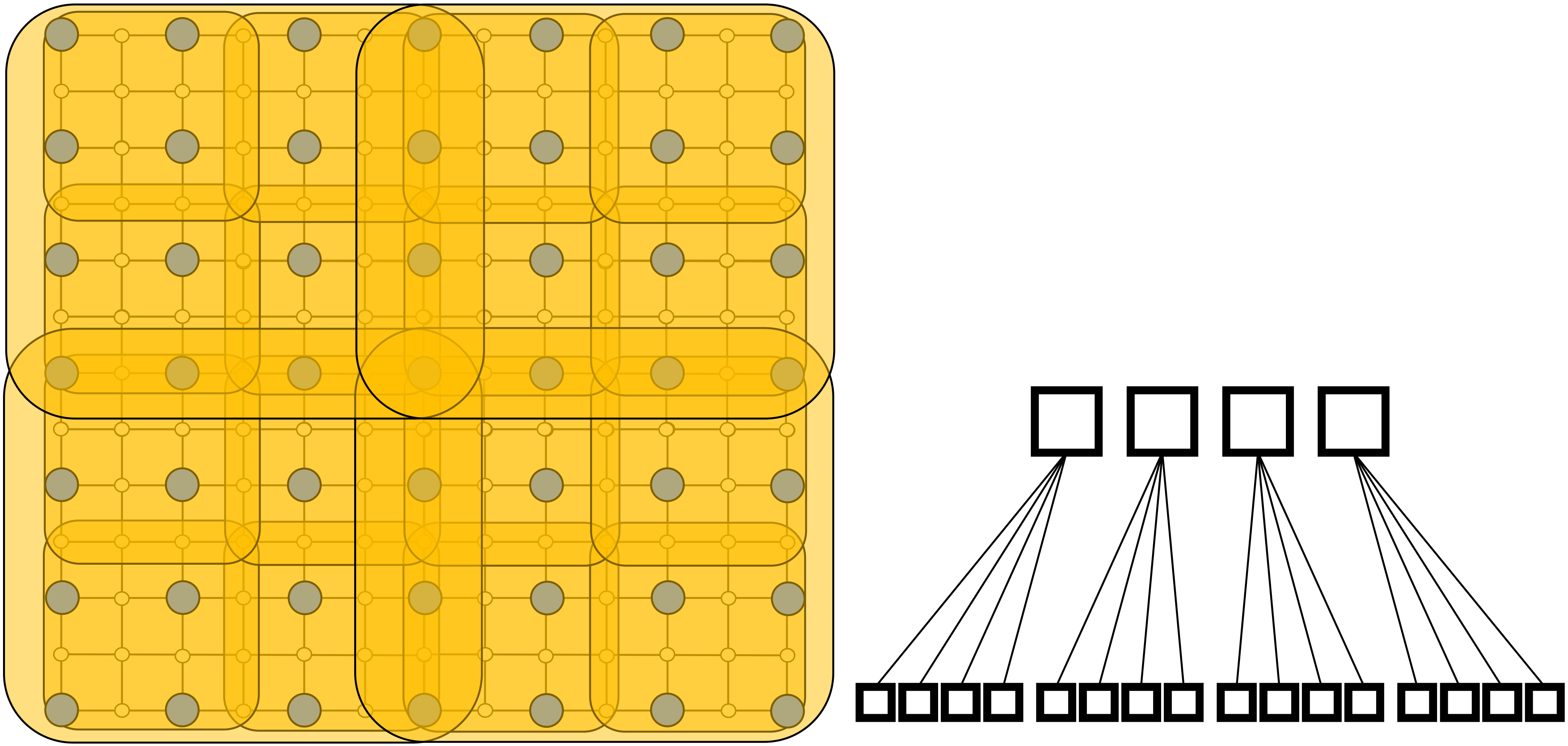
- Create bricks containing 4x4 voxels of finest resolution

Octree Generation Bottom up



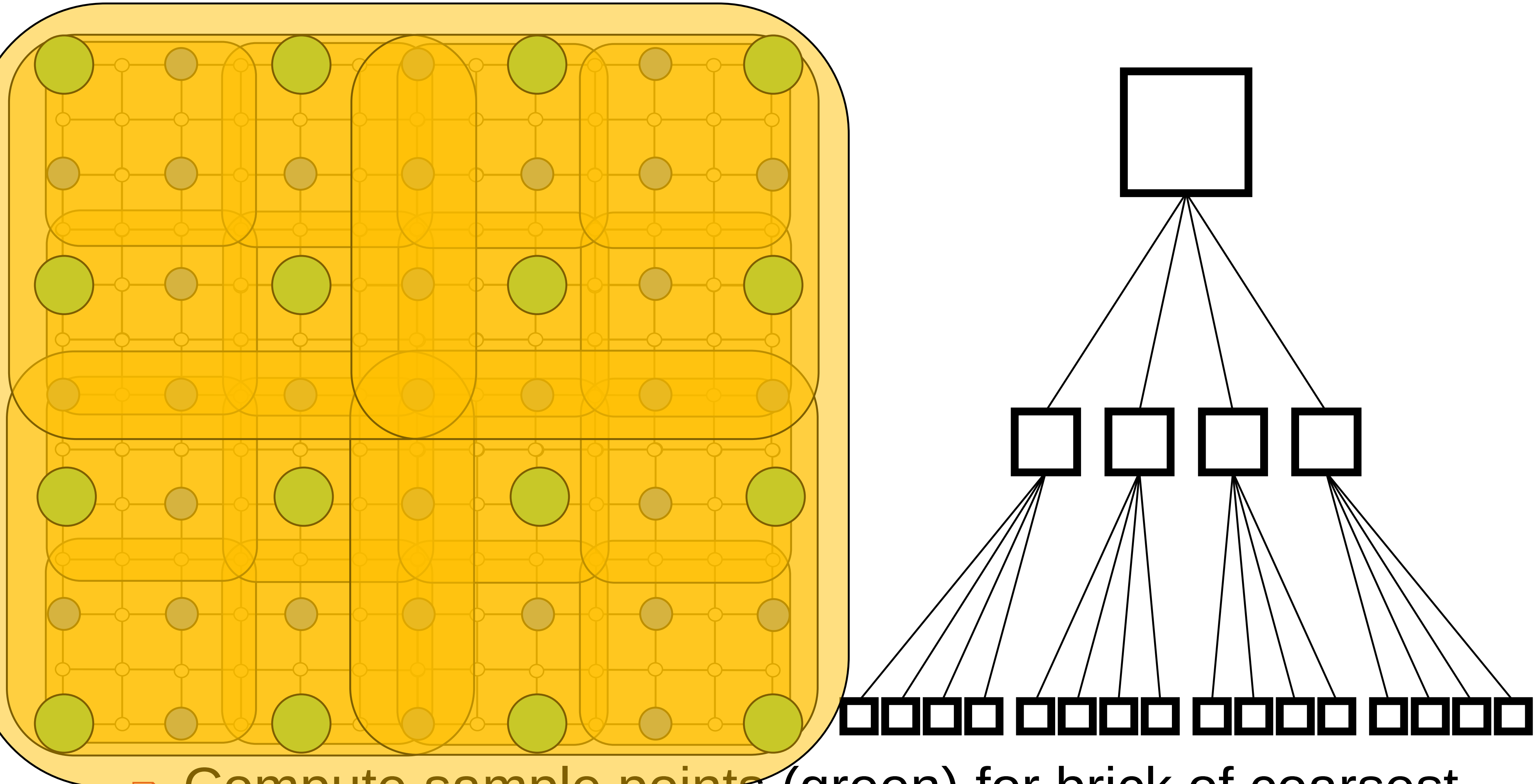
- Compute sample points (blue) for bricks of coarser resolution

Octree Generation Bottom up



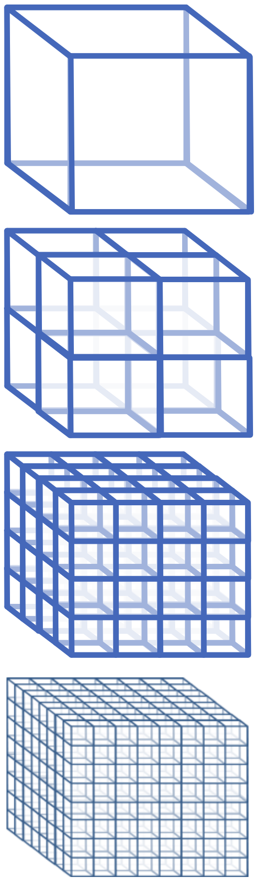
- Create bricks containing 4x4 voxels of next coarser resolution

Octree Generation Bottom up



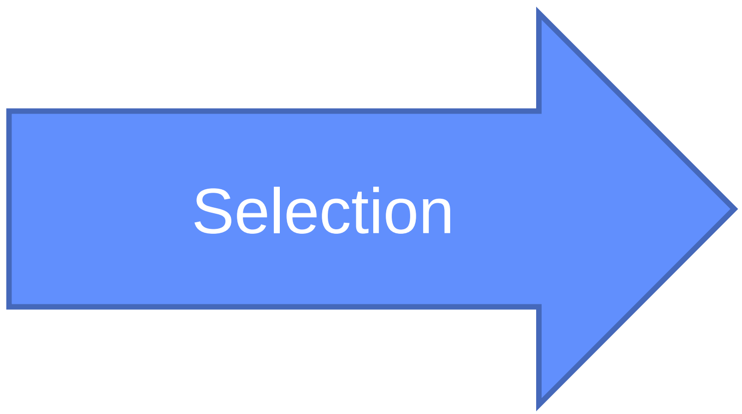
- Compute sample points (green) for brick of coarsest resolution

Multi-Resolution Approach

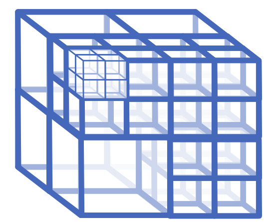


Lowest LOD
(1 brick)

Highest LOD
(512 bricks)

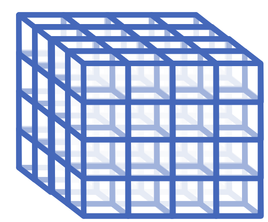


Selected bricks:
"the cut"

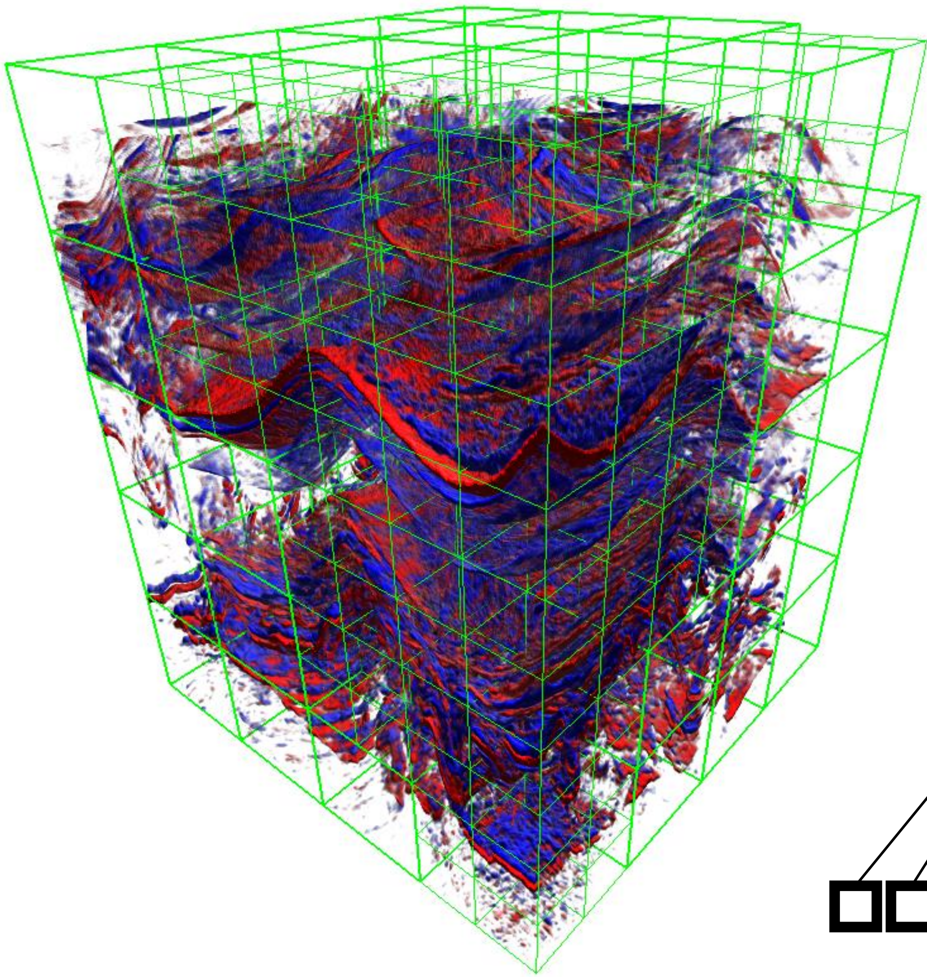


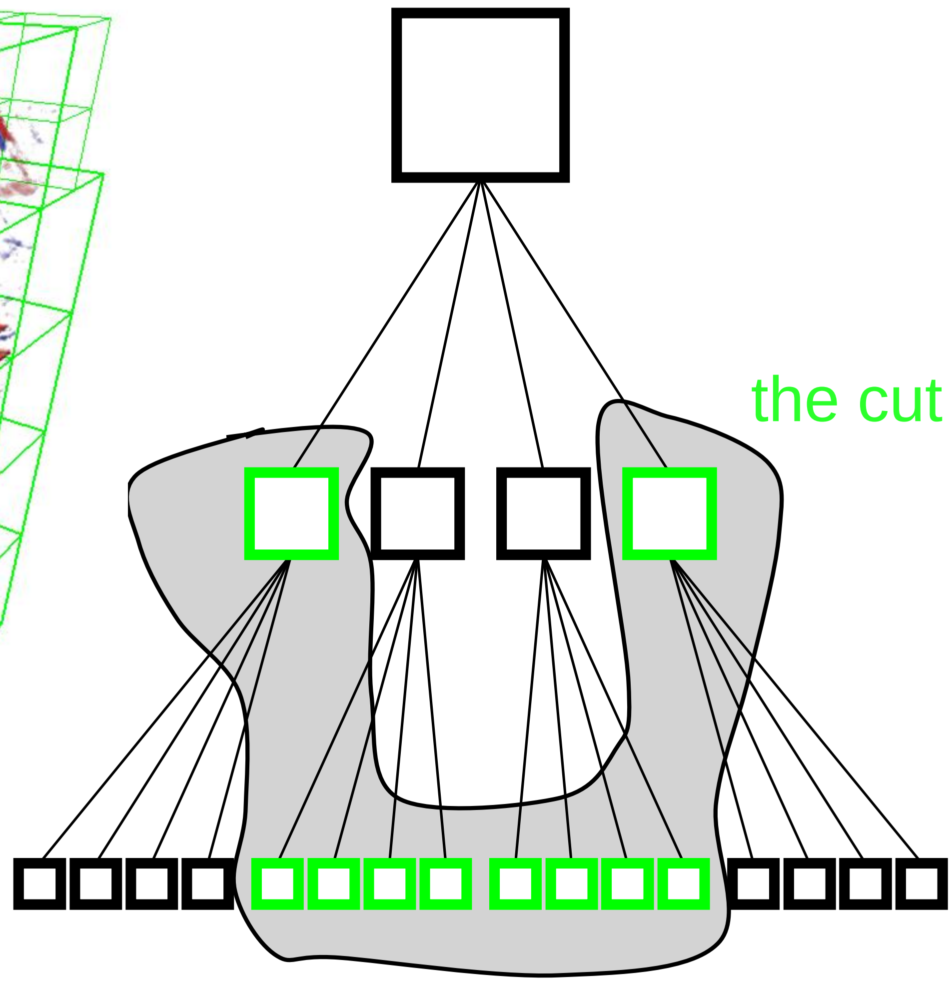
36 bricks
(≈7%)

All bricks of same size in memory
e.g. 32x32x32 voxels



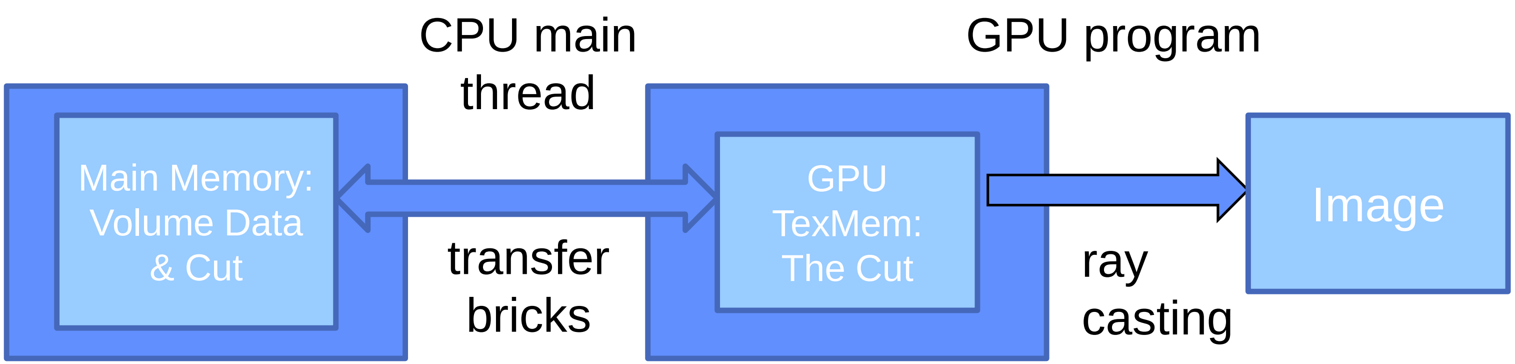
Octree Representation





Basic Out-of-Core Multi-Resolution Rendering Algorithm

- On a frame-to-frame basis
  - **Selection**: Select / update the cut based on an a priority function. The cut defines the level of detail for each volume region.
  - **Transfer**: Upload this data representation (the cut) to the graphics card
  - **Rendering**: Perform rendering based on the current cut



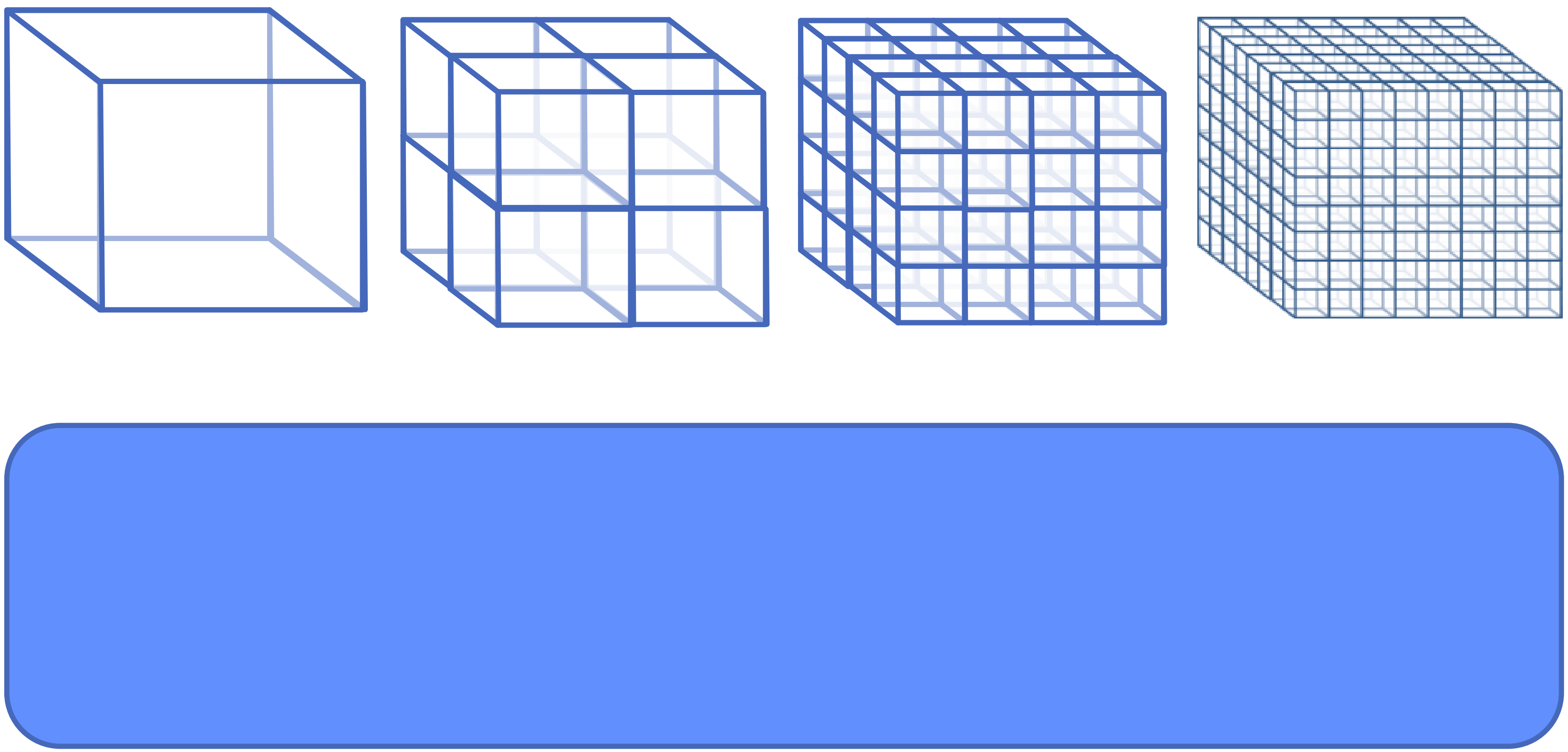
Selection Algorithms

- Use a priority function that defines a priority for each brick
- Use priority queue to organize bricks by priority
- Start with inserting root node into priority queue
- Remove bricks with highest priority and reinsert children into sorted priority queue (could also use heap data structure) until the bricks fill up the available memory of the graphics card (N bricks)

Selection Algorithms Based on Priority Queues

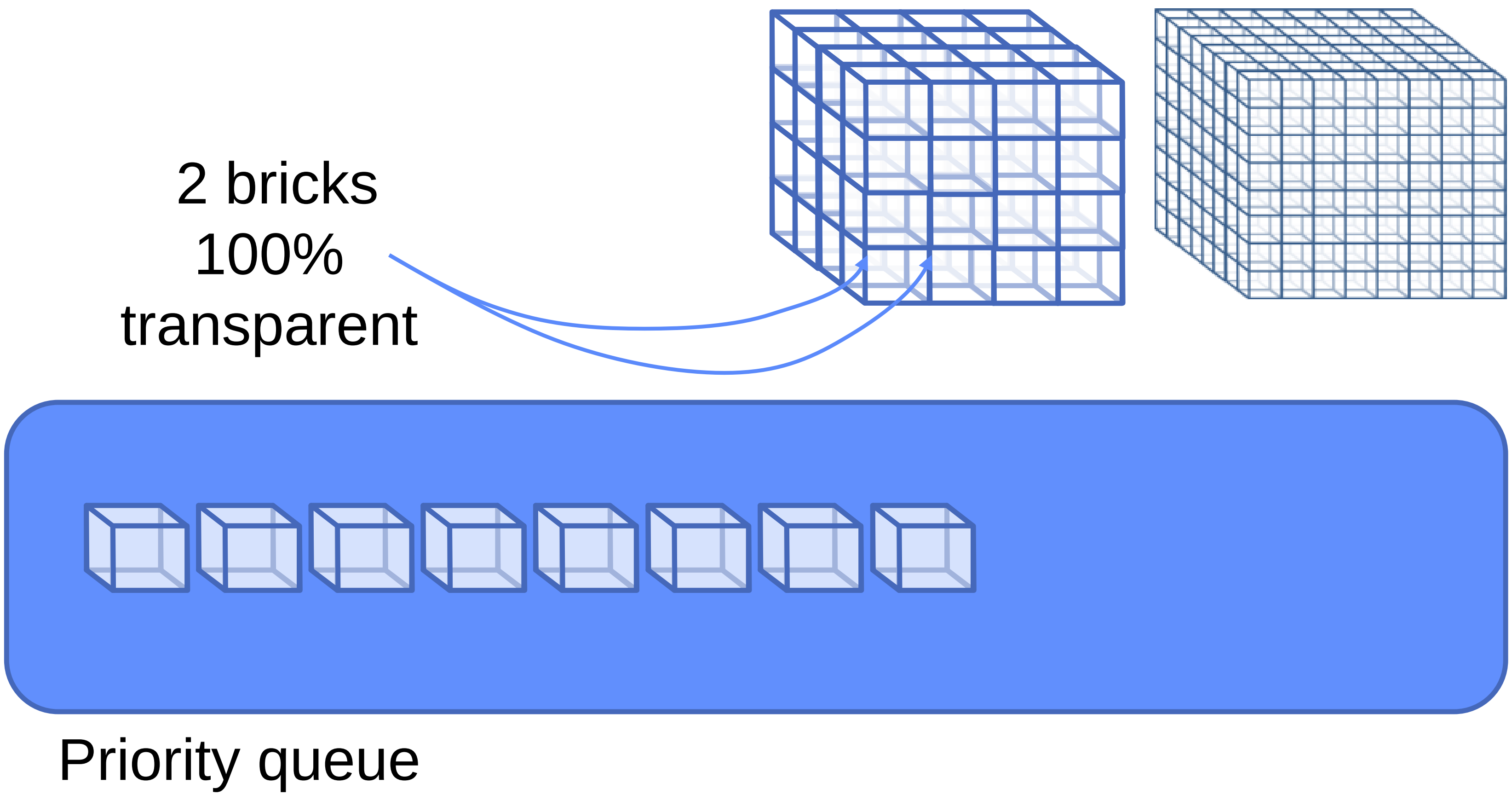
- Boada I., Navazo I. and Scopigno R. “Multiresolution Volume Visualization with Texture-based Octree”. The Visual Computer, vol. 17, pp. 185–197, 2001.
- Guthe Stefan, Wand Michael, Gonser Julius and Straßer Wolfgang. “Interactive Rendering of Large Volume Data Sets”. Proc. Visualization '02, pp. 53–60, Boston-USA, 2002.
- Younesi H., Möller T. and Carr H. “Improving the quality of Multiresolution Volume Rendering”. En Proc. Eurographics / IEEE VGTC Symposium on Visualization, pp. 251–258, Lisboa–Portugal, 2006.
- Wang Chaoli, Garcia Antonio and WeiShen Han. “Interactive Level-of-Detail Selection Using Image-Based Quality Metric for Large Volume Visualization”. IEEE Transactions on Visualization and Computer Graphics. Vol.13 – No.1, pp. 122–134, 2007.

Selection Algorithms



Priority queue

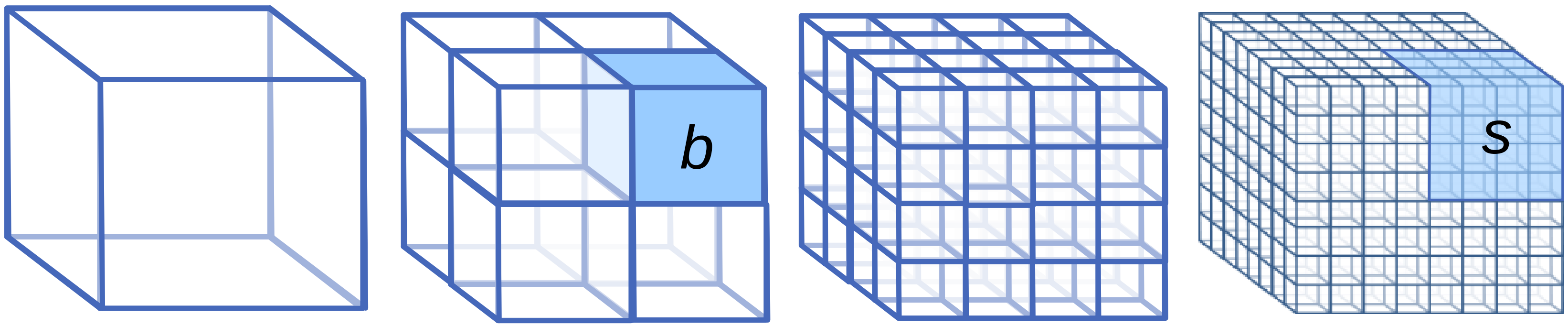
Selection Algorithms



Selection Algorithms

- Greedy algorithm
  - Selects the most promising next step based on the locally available information
  - In general it does not result in the global optimum
- Node with highest error = best choice?
- Global error has not been defined
- How close to the global optimal algorithm does the greedy selection algorithm perform?
  - Global optimum is defined by the cut that has the lowest global error within the constraints of the problem – here the number of bricks that can be stored in the graphics card
- Frame-to-frame coherence?
- Bandwidth is limited – and thus the number of bricks that can be exchanged per frame

Data-based Multi-Resolution Distortion



$$D(b) = \sum_{i=1}^{n_s} D(s_i, b_i)$$

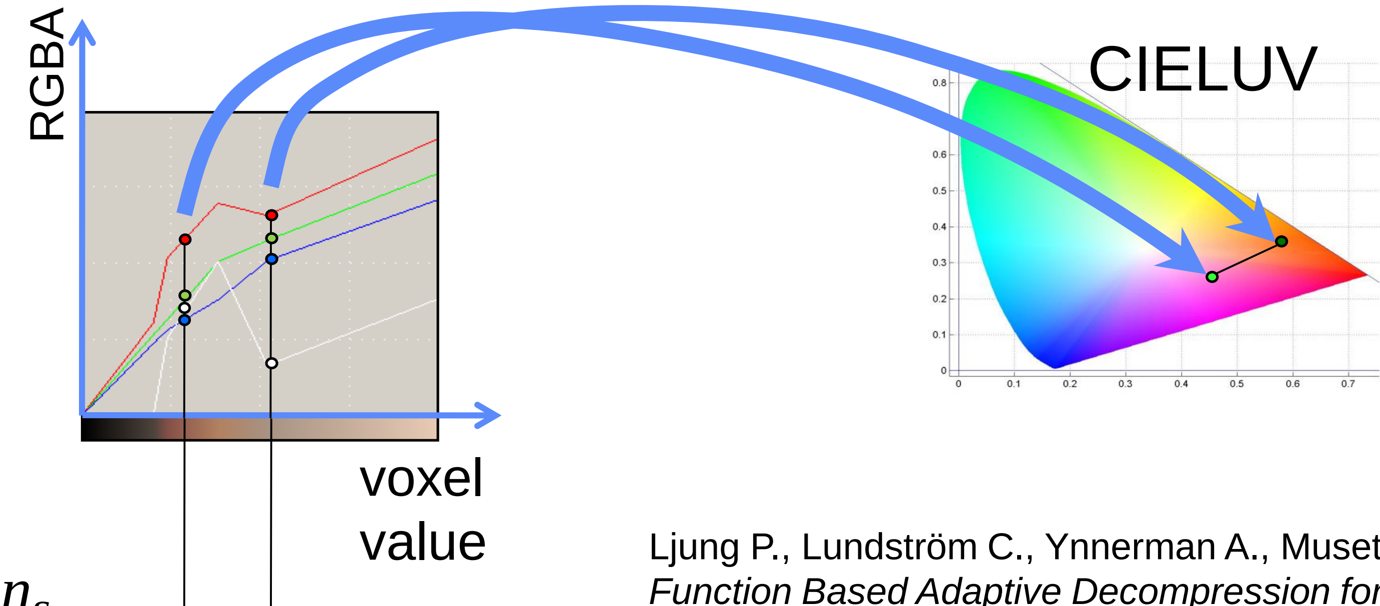
Distortion *D(b)* defines the error introduced by approximating brick set *s* by brick *b*

n_s number of voxels in brick set *s*
s_i data value of voxel *i* in brick set *s*
b_i is corresponding data value to s_i in brick *b*
$D(s_i, b_i)$ error of representing s_i by b_i

Guthe S., Wand M., Gonser J., Straßer W. "*Interactive Rendering of Large Volume Data Sets*". En Proc. Visualization '02, pp 53–60, Boston-USA, 2002.

Data-based Multi-Resolution Distortion

- Data-based multi-resolution distortion



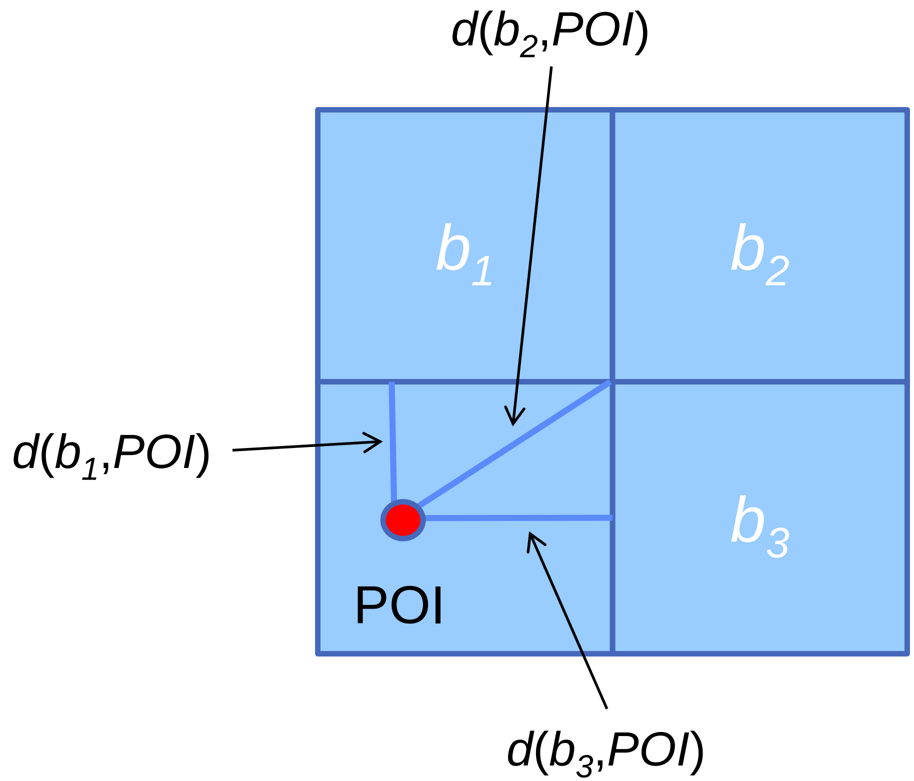
$$D(b) = \sum_{i=1}^{n_s} D(s_i, b_i)$$

$D(s_i, b_i)$ is a distance in the CIELUV domain. It is computed after applying the transfer function to s_i and b_i

Ljung P., Lundström C., Ynnerman A., Museth K. “*Transfer Function Based Adaptive Decompression for Volume Rendering of Large Medical Data Sets*”. En Proc. IEEE Volume Visualization and Graphics Symposium, pp. 25–32, **2004**.
Wang Chaoli, Garcia Antonio, WeiShen Han. “*Interactive Level-of-Detail Selection Using Image-Based Quality Metric for Large Volume Visualization*”. IEEE Transactions on Visualization and Computer Graphics. Vol.13 – No.1, pp. 122–134, **2007**.

Level of Importance

- Refinement priority based on
  - Point of interest (POI)
  - Region of interest (ROI)
  - View point



$$I(b) = \frac{diag(b)}{diag(b) + d(b, POI)}$$

$d(b,POI)$ is the minimum distance from brick b to the POI
$diag(b)$ is the length of a brick's diagonal.
$I(b) \in (0,1]$ and coarser bricks closer to the POI have higher priority to be refined.
$d(b,PI)$ is zero if POI is inside b.
Instead of POI we may use ROI or the view point or a weighted mix

Global Error

Combining Distortion & Level of Importance results in a weighted distortion for each brick

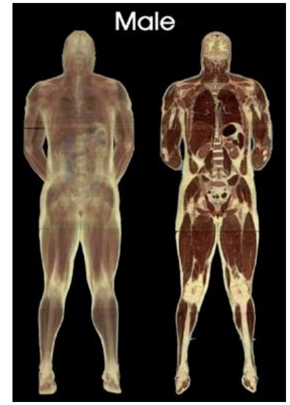
$$E(b) = D(b) * I(b)$$

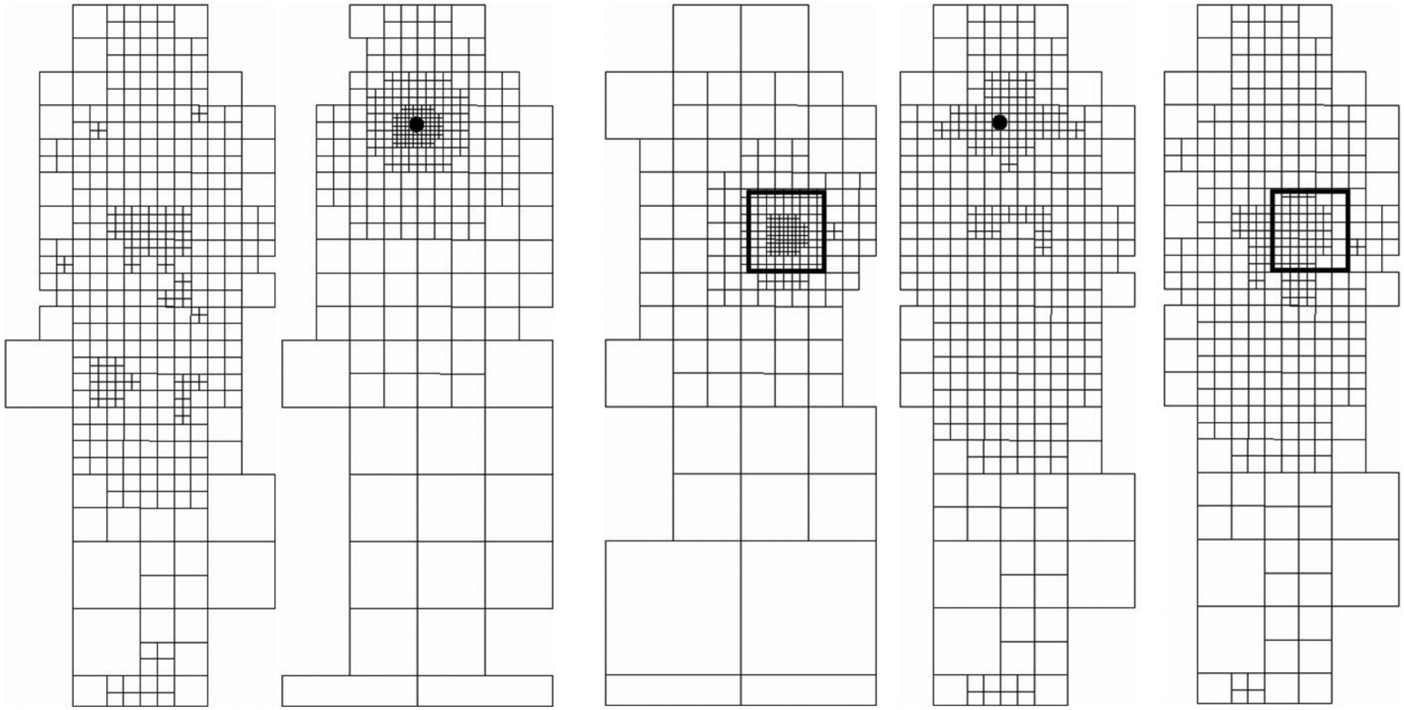
Global error of a cut *X* is the average per voxel error with *n(b)* number of voxels in brick b

$$E(X) = \frac{1}{\sum_{b \in X} n(b)} \sum_{b \in X} E(b)$$

Problem: find the cut *X* that minimizes *E*(*X*) under the given constraints (mainly memory constraint)
difficult problem – can be solved in polynomial time
use heuristic for real-time rendering – greedy approaches

Global Error

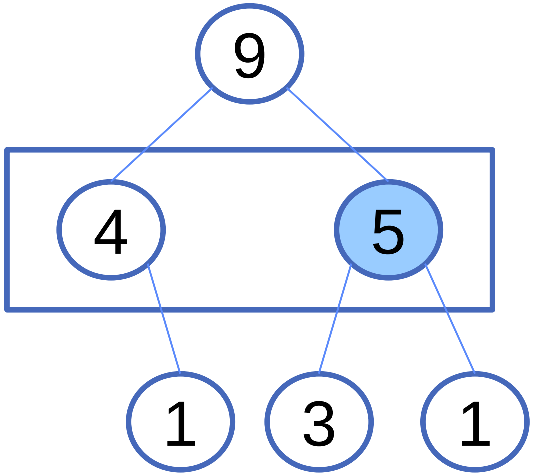




$E(b) = D(b)$	$E(b) = I(b)$	$E(b) = I(b)$	$E(b) = D(b)I(b)$	$E(b) = D(b)I(b)$
Distortion only	Importance only for POI	Importance only for ROI	Distortion times Importance for POI	Distortion times Importance for ROI

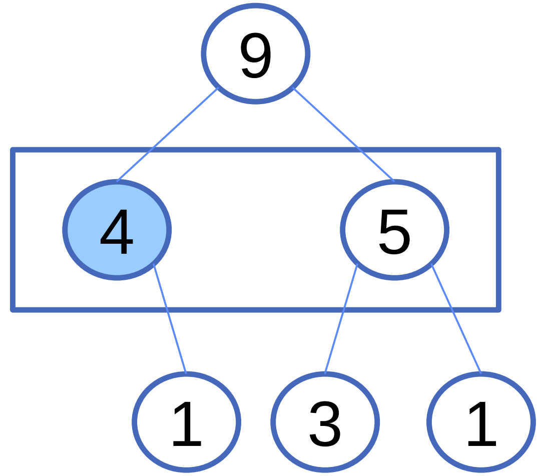
Improved Cut Selection

<u>Naïve Greedy Cut Selection</u>:
Split the node with highest error



Global Error = 4+5 = 9

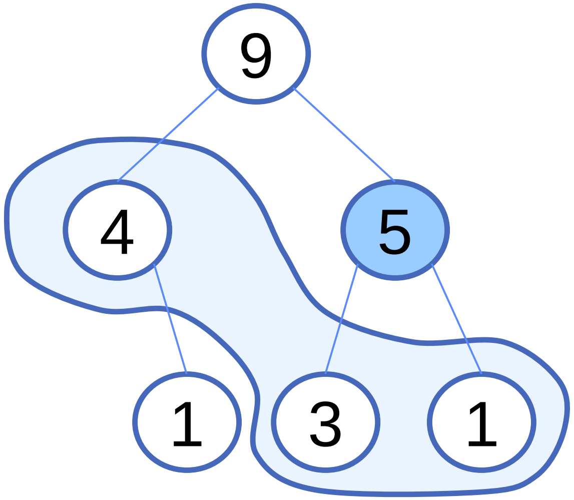
<u>Improved Greedy Cut Selection</u>:
Split the node with highest error reduction per child



Global Error = 4+5 = 9

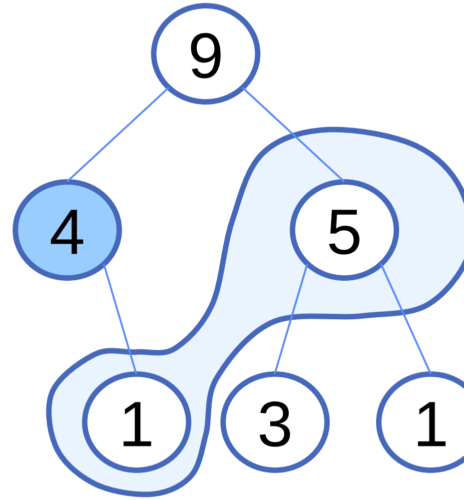
Improved Cut Selection

<u>Naïve Greedy Cut Selection</u>:
Split the node with highest error



Error = 4+3+1 = 8

<u>Improved Greedy Cut Selection</u>:
Split the node with highest error reduction per child



Error = 1+5 = 6

Split the node with highest error reduction is often better than simply splitting the node with highest error

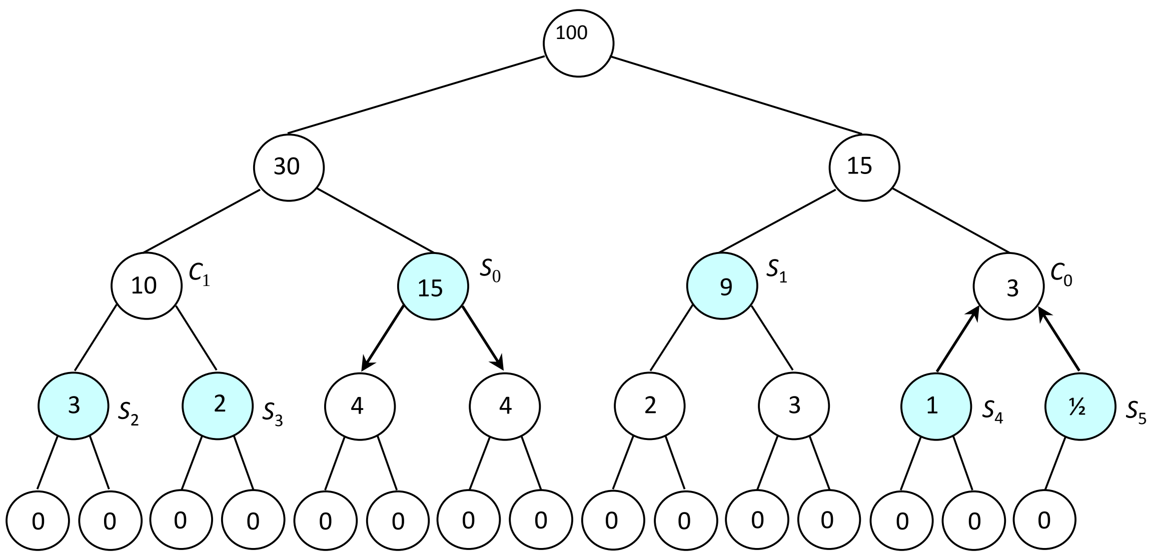
Incremental Cut Update

- Selecting a new cut every frame may be necessary due to moving view point, region of interest or point of interest
- Typically not everything changes
  - Use frame-to-frame coherence
  - Update cut instead of computing a completely new cut
- Bricks need to be stored on the graphics card for fast rendering
  - Limit download into graphics card to avoid large frame rate variations

Incremental Greedy Selection Algorithm

Split-and-Collapse Algorithm

- Consider a maximum cut size (number of bricks) (N)
- Limit number of brick replacements (downloads) per frame (M)
- Maintains ordered sets of splitable nodes S and collapsible nodes C

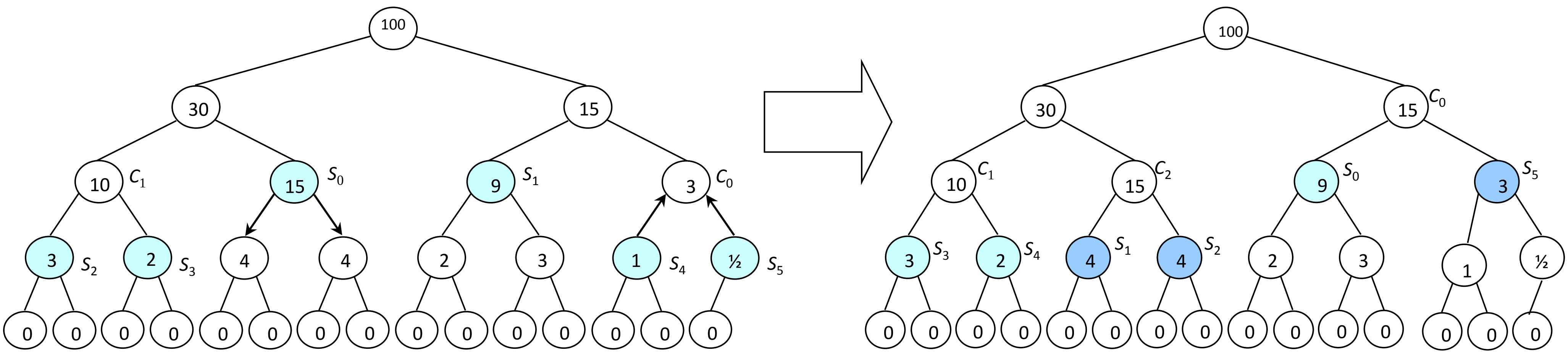


- The priority p in each node: error increase or error reduction
  - c_0 is collapsible node from set C with lowest error increase
  - s_0 is splitable node from set S with highest error reduction

Incremental Greedy Selection Algorithm

Split-and-Collapse Algorithm

- Starts from previous cut X and performs the following in a loop
  - Try to split the node s_0 with highest priority $p(s_0)$
  - If split is not possible, then collapse node c_0 with lowest priority $p(c_0)$
  - Stop when N or M reached or $p(c_0)>p(s_0)$

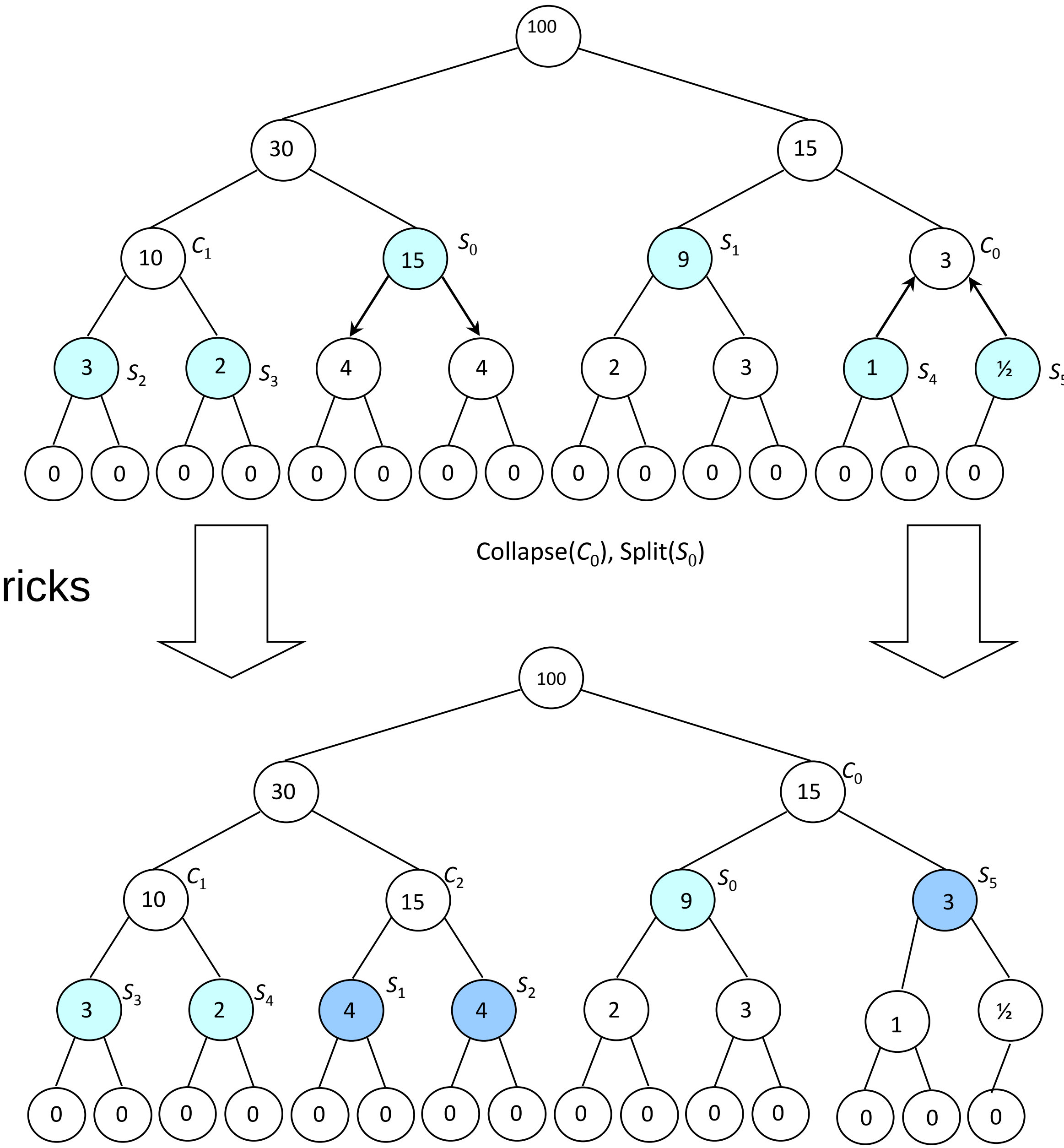


Details: Carmona, R., Froehlich, B.
Error-controlled real-time cut updates for multi-resolution volume rendering
Elsevier Computers & Graphics, Volume 35, Issue 4, pp 931-944, August 2011.

Error = 30,5

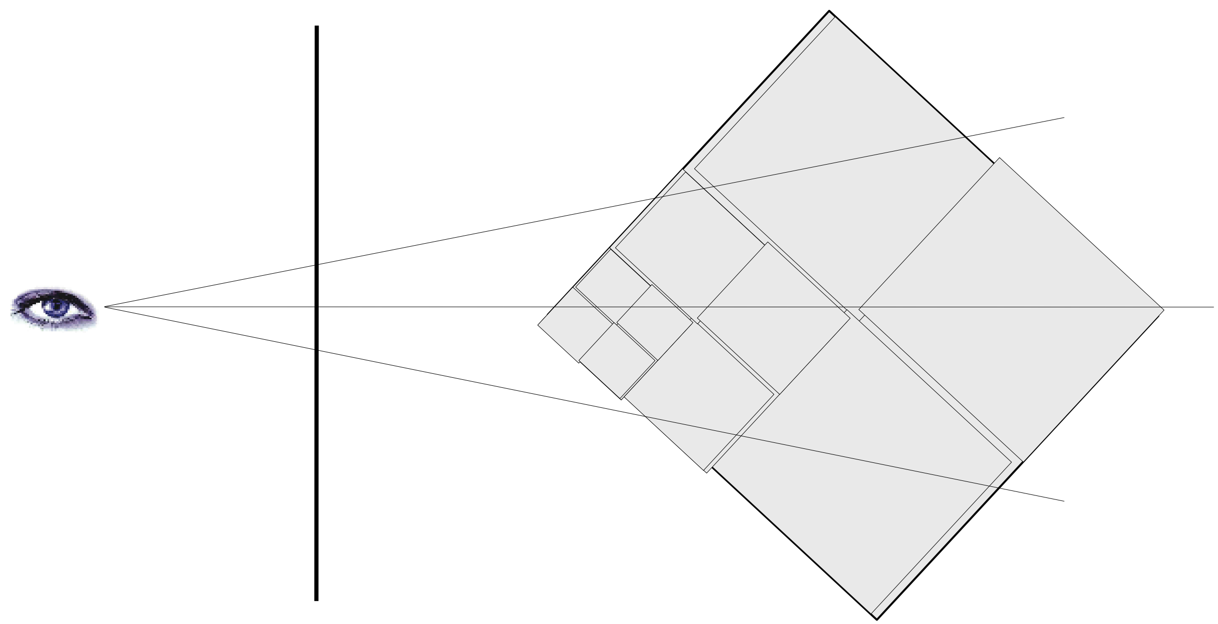
Memory constraint 6 bricks

Error = 25



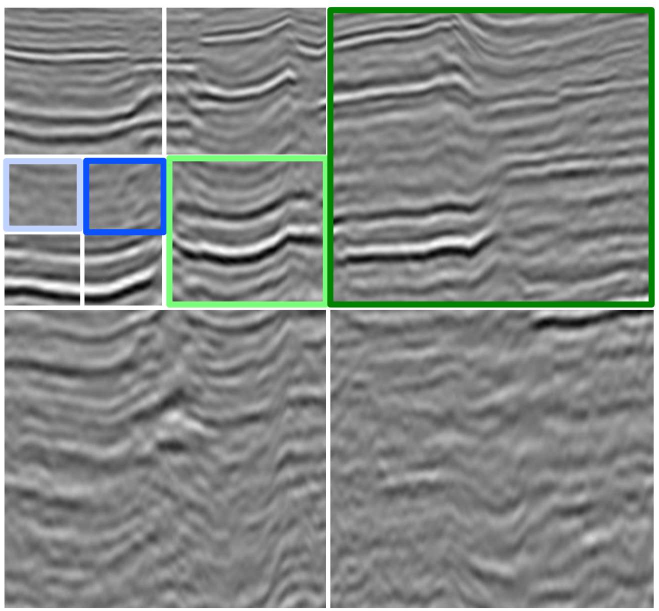
Multi-Resolution Single Pass Ray Casting

- Rays have to traverse multi-resolution volume consisting of a set of bricks of different level of detail

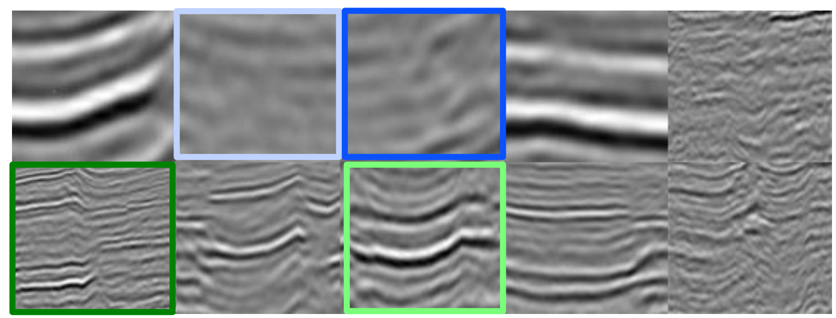


Multi-Resolution 3D-Texture Atlas

- One big 3D-texture holding current working set of bricks
- Each brick is of the same size in memory
- Individual bricks are replaced on a frame-by-frame basis
- All bricks accessible at once through a single texture



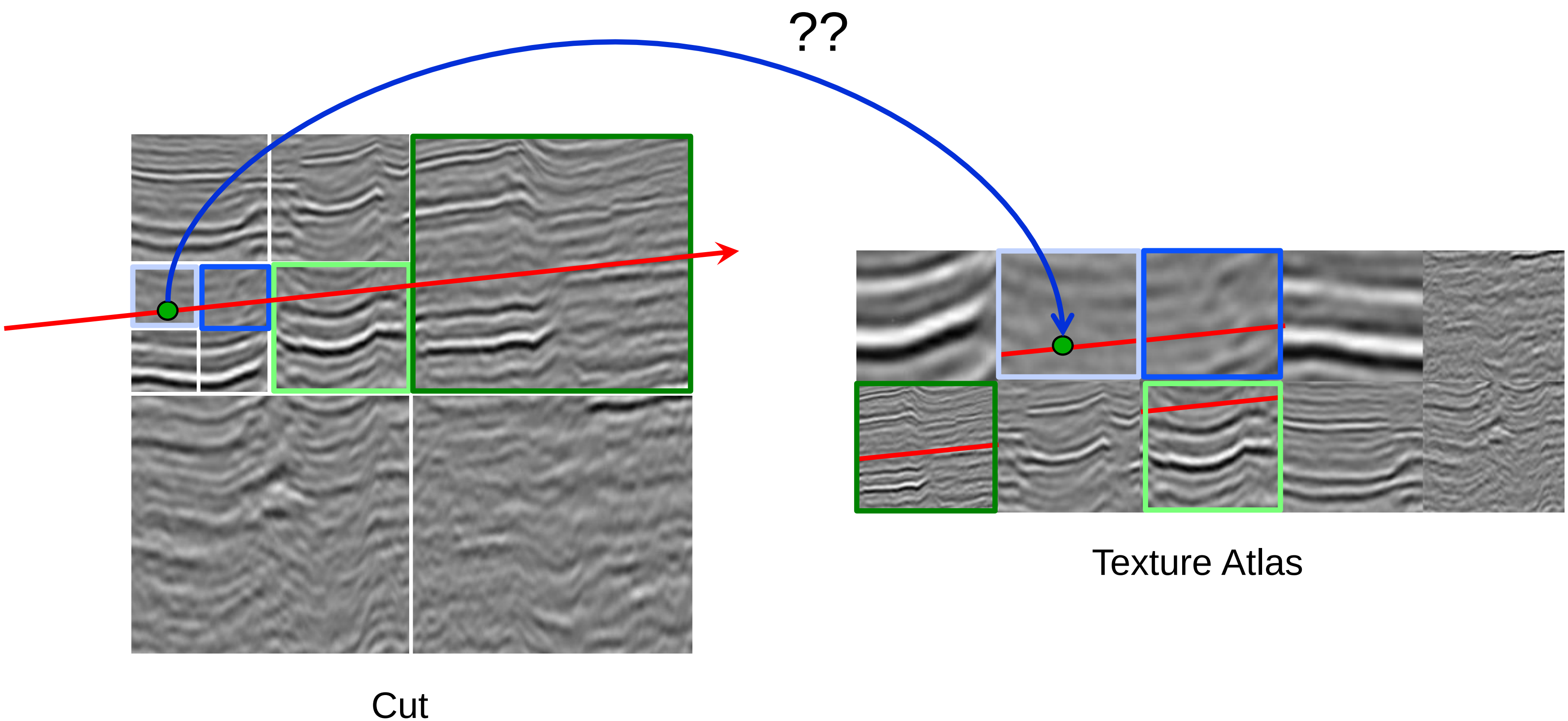
Cut



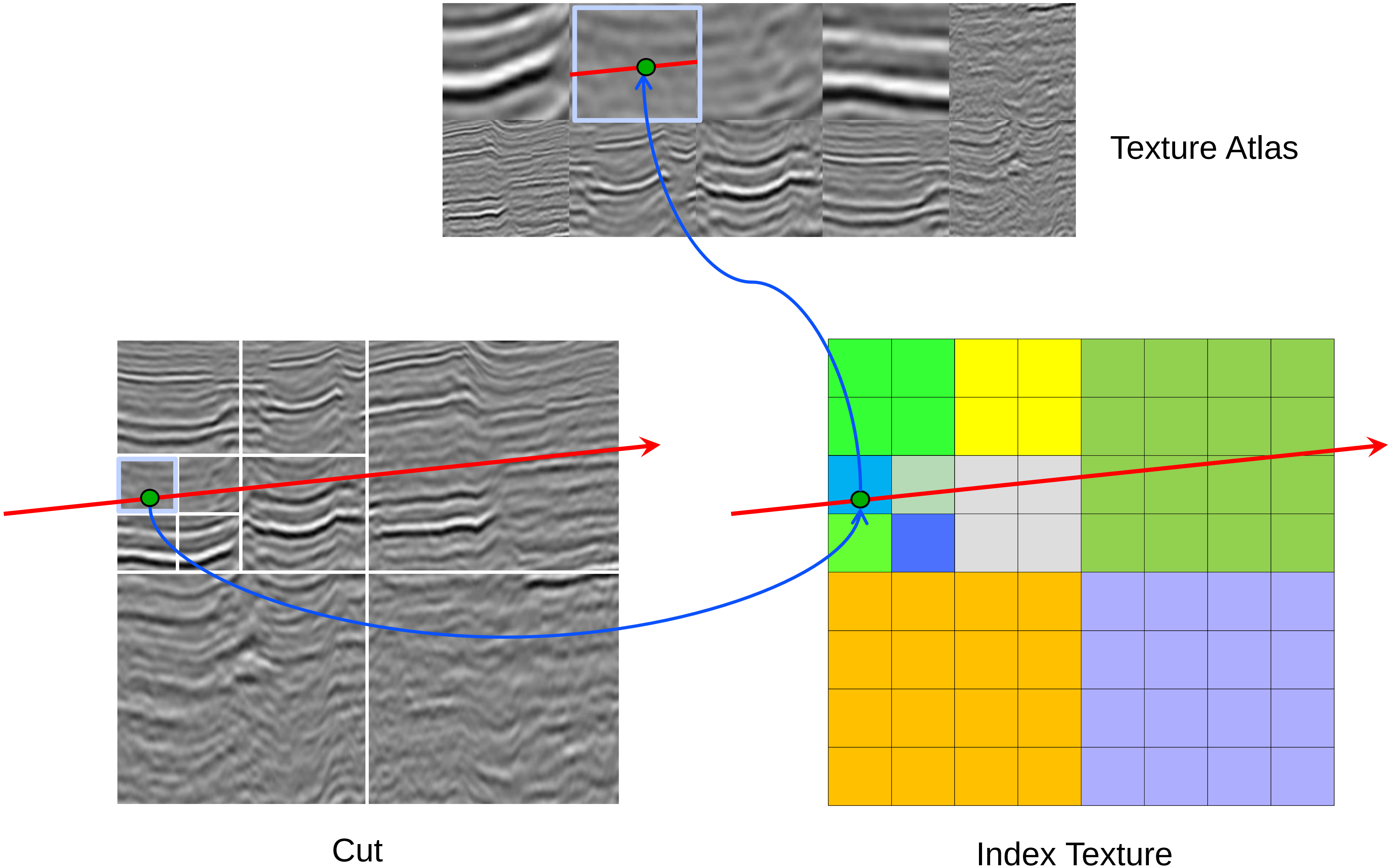
Texture Atlas

Octree Ray Traversal

- Ray needs to traverse bricks in texture atlas
- Problem: How to find the brick that corresponds to a 3D point on the ray?

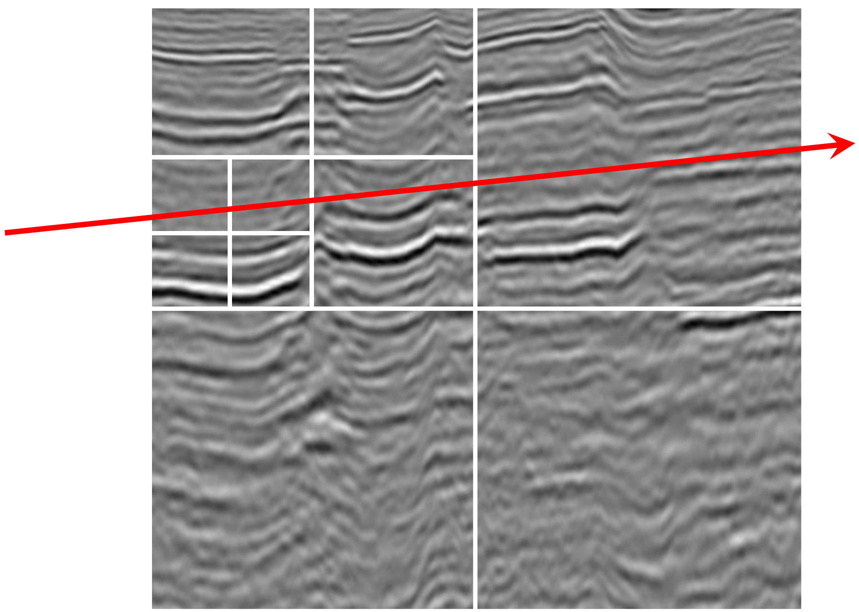


Point Localization via Index Texture

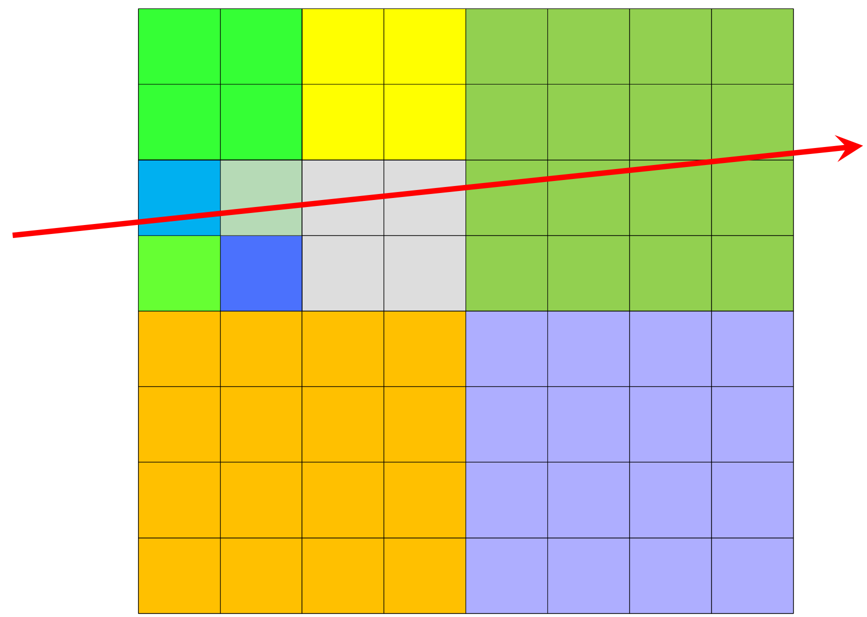


Index Texture

- Index texture is a 3D array (texture)
- Size of index texture: Number of bricks on the finest octree level
- Stores index into texture atlas and octree level of brick
- For each sample point a lookup into the index texture is performed to find the relevant brick
- Needs to be partially updated on a frame-by-frame basis



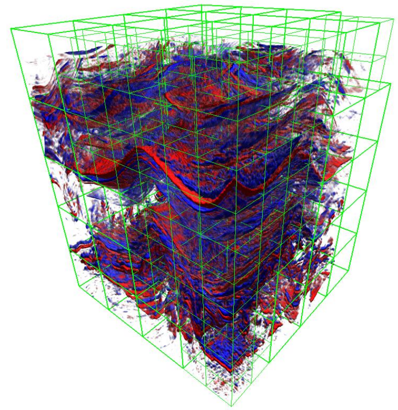
Cut

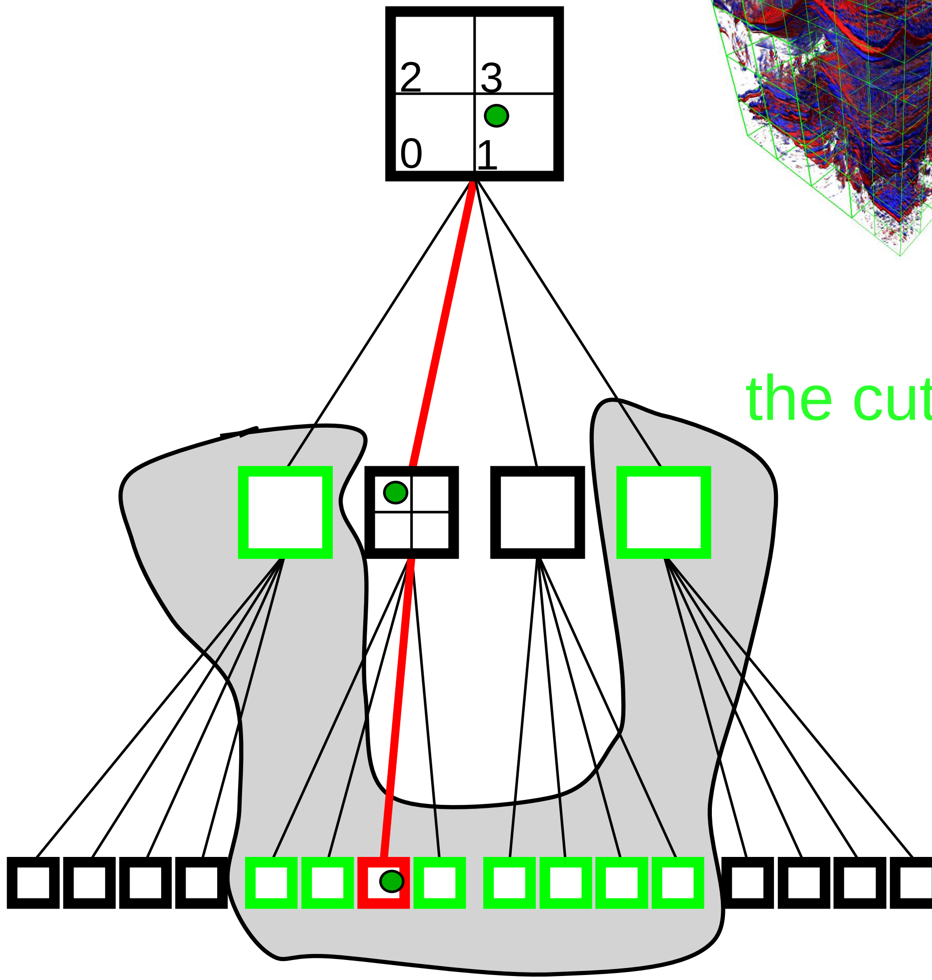


Index Texture

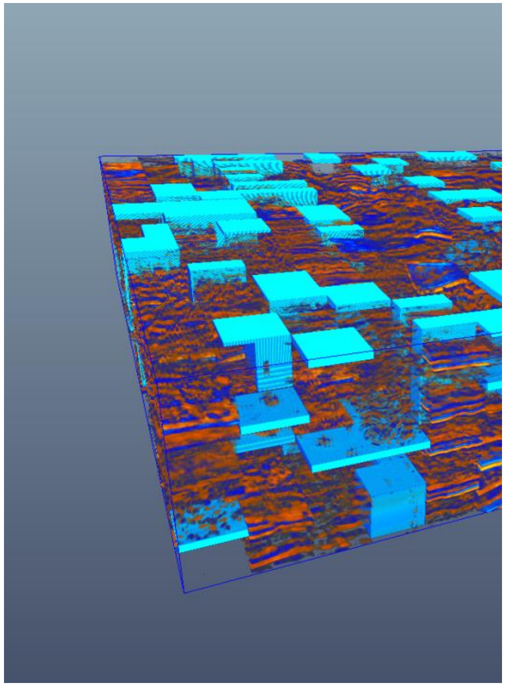
Point Localization via Octree Traversal

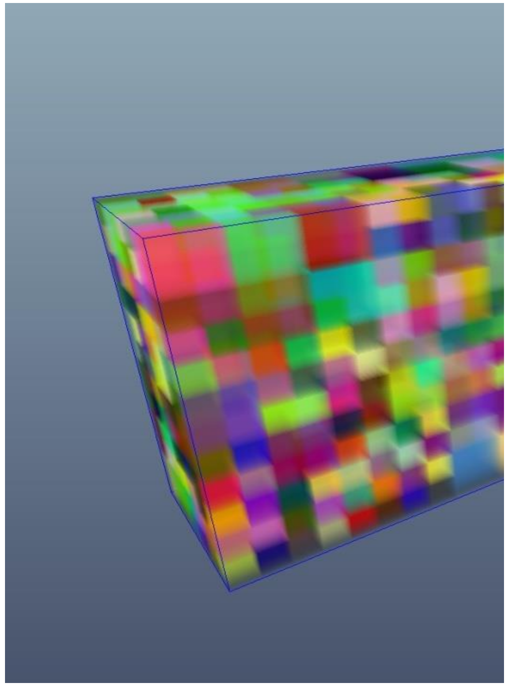
- Requires serialized octree structure from root to nodes of the cut on the graphics card
- Inner nodes store the address of the children
- Leaf nodes store the address of the brick in the index texture atlas
- Point localization starts from the root until a node of the cut is reached
- On each level it selects the child which contains the point to be localized by comparing its coordinates to the center of the octree node

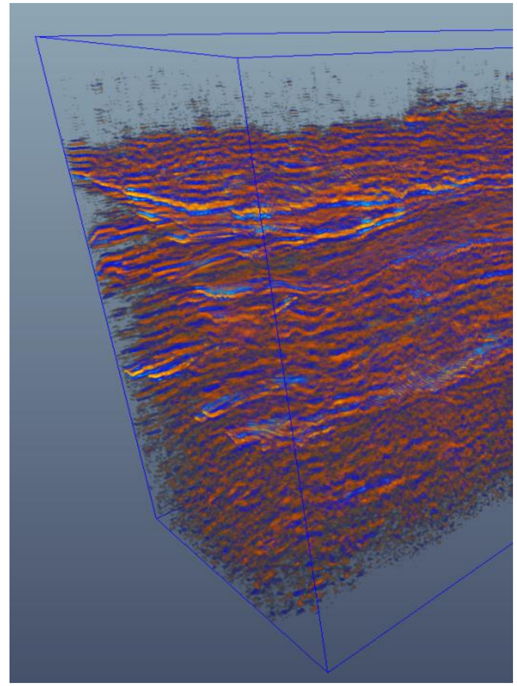




Multi-Resolution Ray Casting

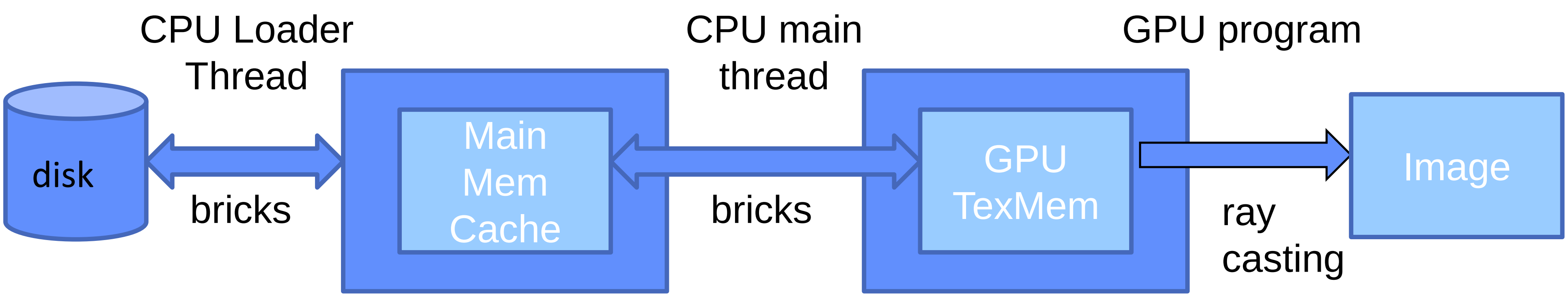

texture atlas


index texture

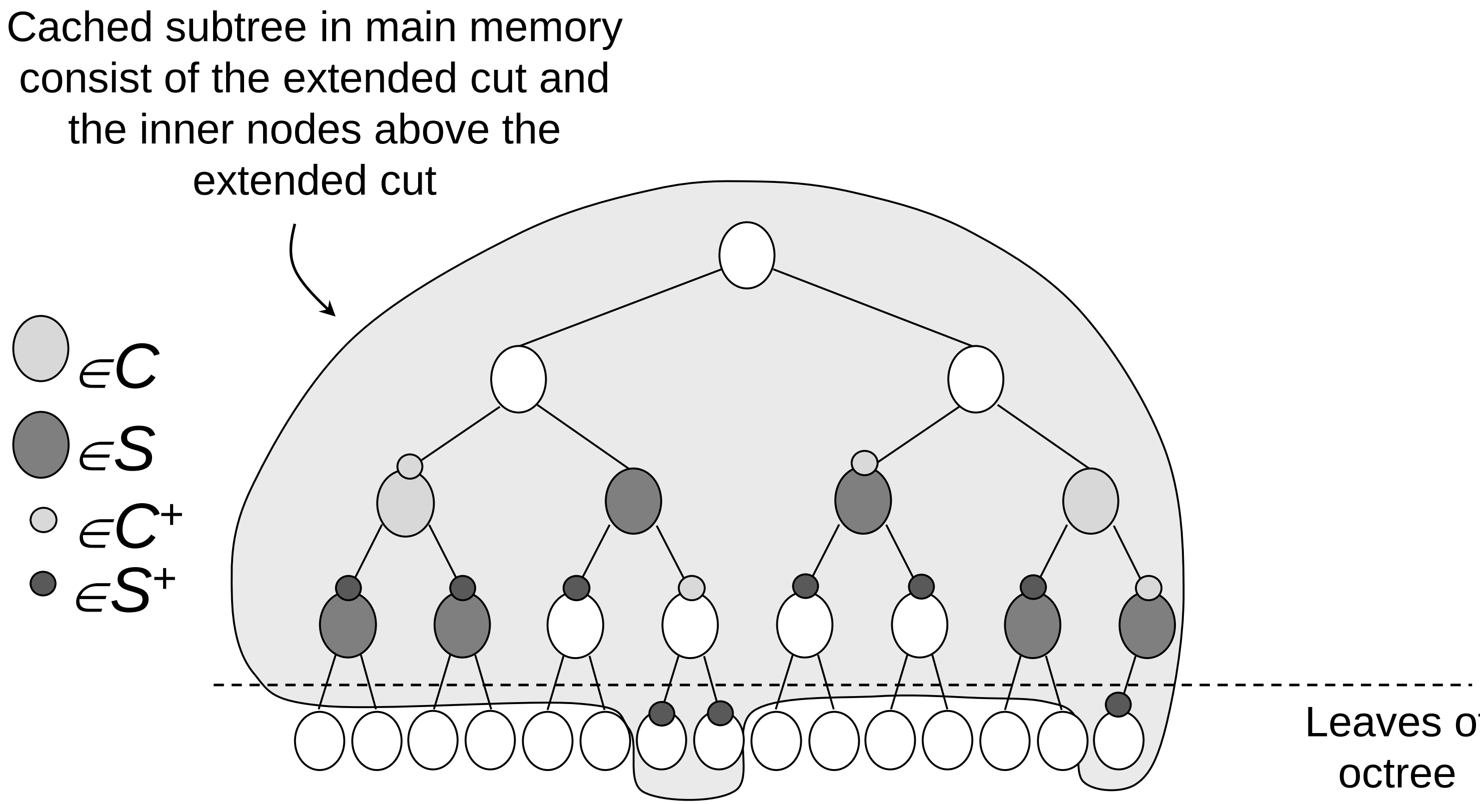

final volume rendering

Out-of-Core Rendering

- Split-and-Collapse algorithm over extended cut X+ is used to define cut in main memory that serves as cache
- Separate loader thread pages bricks from disk / network into main memory
- Main thread pages bricks from main memory into GPU memory
- Always keep subtree above cut in main memory
  - Always allows collapse operations



Out-of-Core Rendering



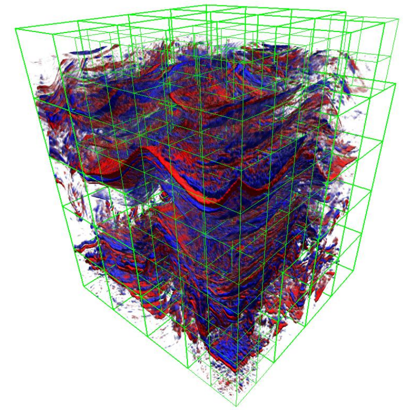
C: collapsible nodes of cut X

S: splitable nodes of cut X

C^+: collapsible nodes of cut X^+

S^+: splitable nodes of cut X^+

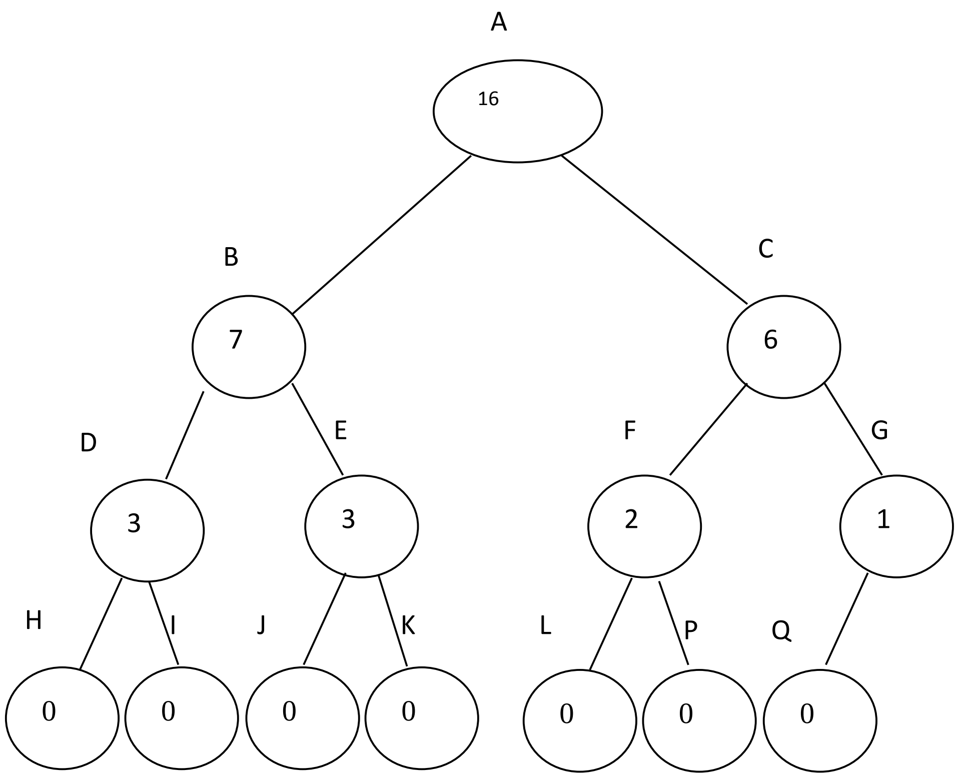
Summary Multi-Resolution Volume Rendering



- Entire volume data does not fit
  - into graphics card's memory or
  - even into main memory
- Approach
  - Multi-resolution techniques
    and paging based on bricks
  - On a frame-to-frame basis
    - Select / update a level of detail for each
      volume region based on an a priority function
    - Upload this representation to graphics card
    - Perform rendering based on this representation

End

Optimal Cut Selection



Optimal Selection Algorithm:

Find S, |S|≤*N*, with minimal distortion per voxel

(a) Backtracking:

test recursively with all possible splits. On each node (*K* nodes), *N* possibilities ⇒ O(K^N).

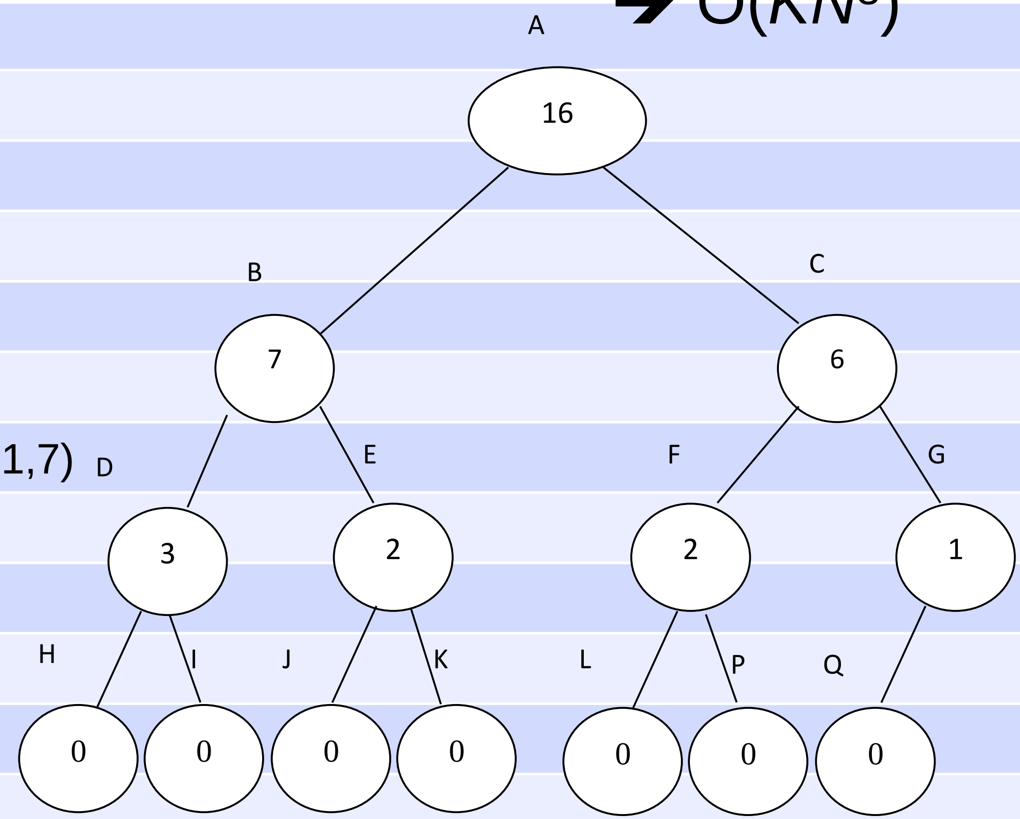
(b) Dynamic Programming:

Performs bottom-up, on each node obtain the optimal selection for each possible *n*≤*N* combining optimal solutions from child nodes. At root node

Optimal Cut Selection

Concat $O(N)$
Combining $O(N^2)$
K nodes
→ $O(KN^3)$

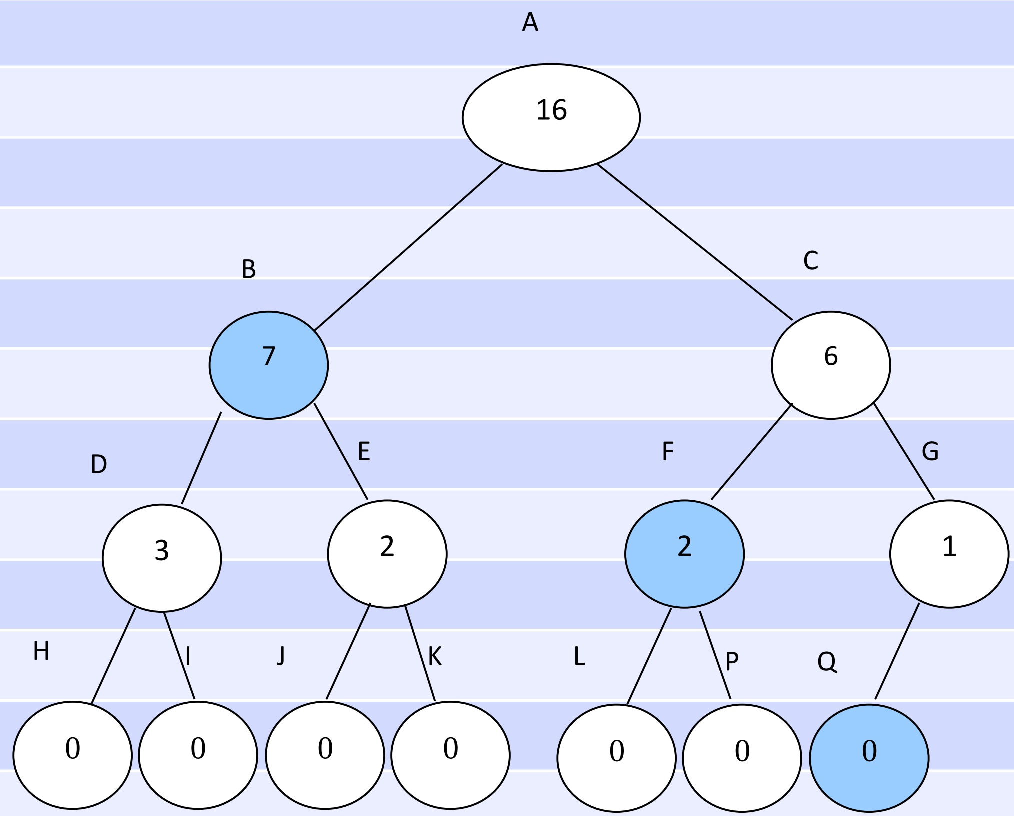
node	Optimal solutions (nodes, #nodes ≤ 3, distortion $D(S)$)
H	(H,1,0)
I	(I,1,0)
D	(HI,2,0), (D,1,3)
J	(J,1,0)
K	(K,1,0)
E	(JK,2,0), (E,1,2)
B	~~(HIJK,4,0)~~, (HIE,3,2), ~~(DJK,3,3)~~, (DE,2,5), (B,1,7)
L	(L,1,0)
P	(P,1,0)
F	(LP,2,0), (F,1,2)
Q	(Q,1,0)
G	(Q,1,0), (G,1,1)
C	(LPQ,3,0), ~~(LPG,3,1)~~, (FQ,2,2), ~~(FG,2,3)~~, (C,1,6)
A	~~(HIELPQ,6,2)~~, ~~(HIEFQ,5,4)~~, ~~(HIEC,4,8)~~, ~~(DELPQ,5,5)~~, ~~(DEFQ,4,7)~~, ~~(DEC,3,11)~~, ~~(BLPQ,4,13)~~,(BFQ,3,9),(BC,2,13),(A,1,16)



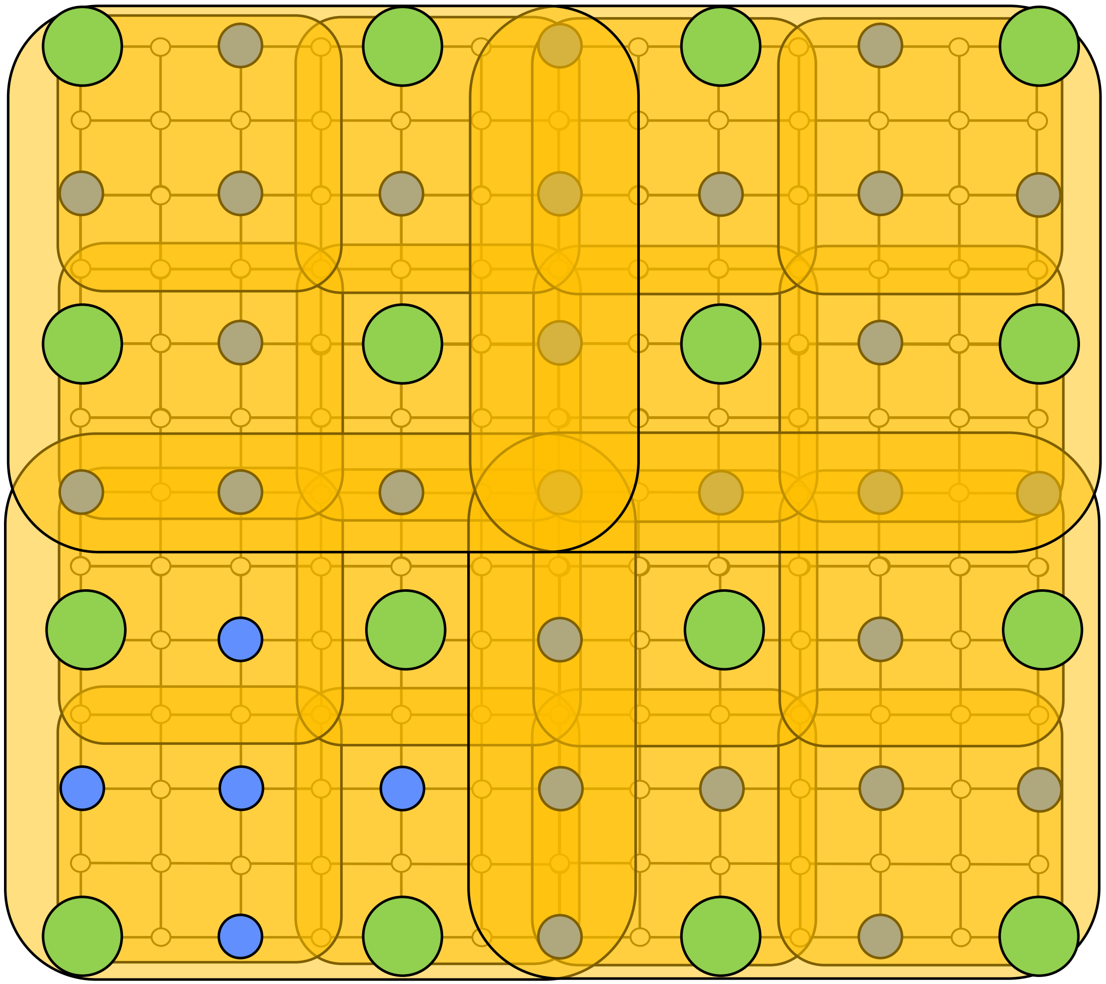
Optimal Cut Selection

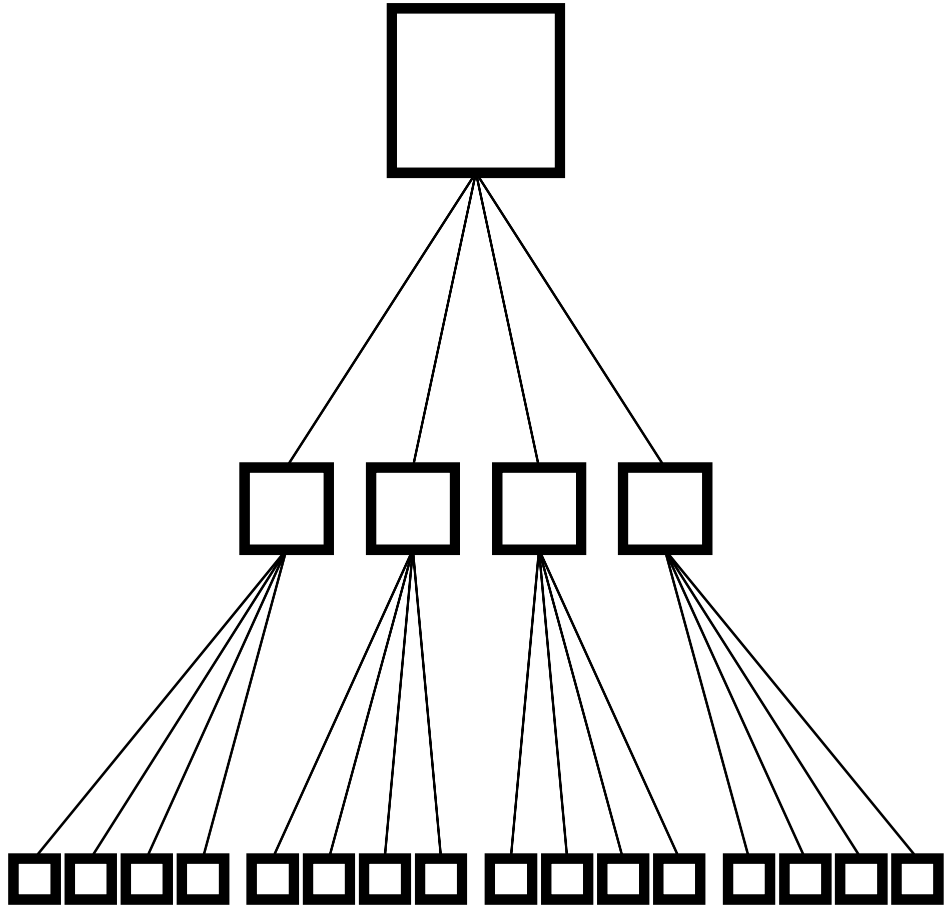
Concat O(N)
Combining O(N
K nodes
➔ O(KN^3)

node	Optimal solutions (nodes, #nodes ≤ 3, distortion $D(S)$)
H	(H,1,0)
I	(I,1,0)
D	(HI,2,0), (D,1,3)
J	(J,1,0)
K	(K,1,0)
E	(JK,2,0), (E,1,2)
B	(HIE,3,2), (DE,2,5), (B,1,7)
L	(L,1,0)
P	(P,1,0)
F	(LP,2,0), (F,1,2)
Q	(Q,1,0)
G	(Q,1,0), (G,1,1)
C	(LPQ,3,0), (FQ,2,2), (C,1,6)
A	**(BFQ,3,9)**,(BC,2,13),(A,1,16)

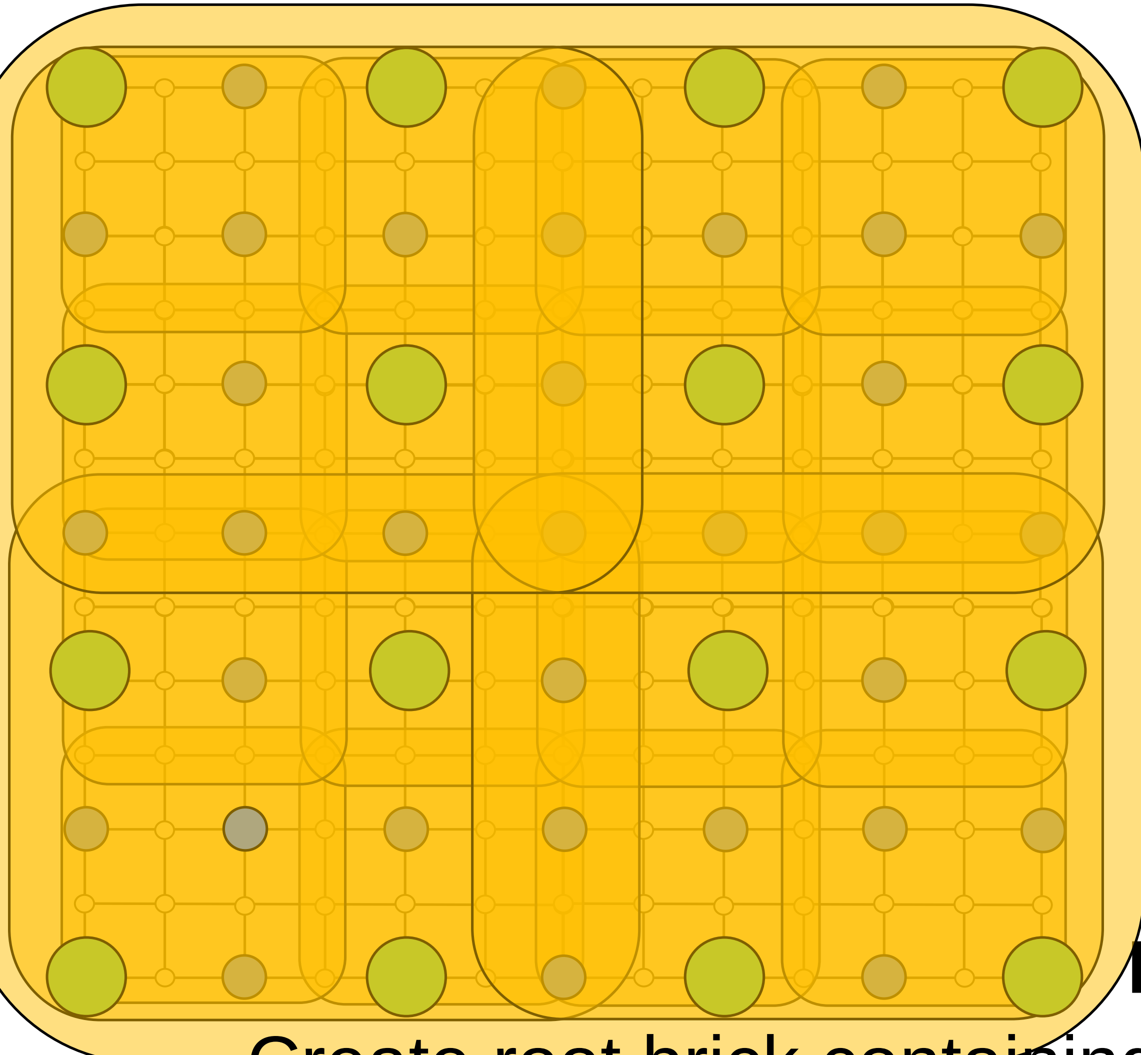


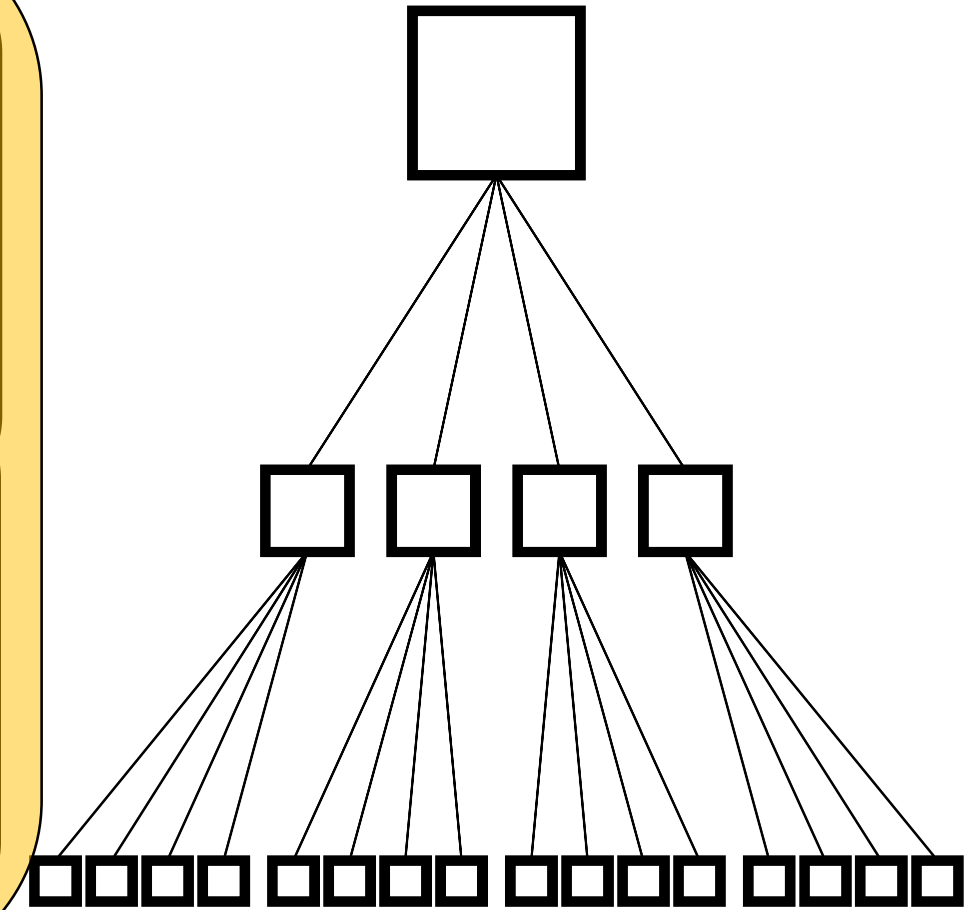
Octree Generation Bottom up





Octree Generation Bottom up





- Create root brick containing 4x4 voxels of coarsest resolution

Visualization

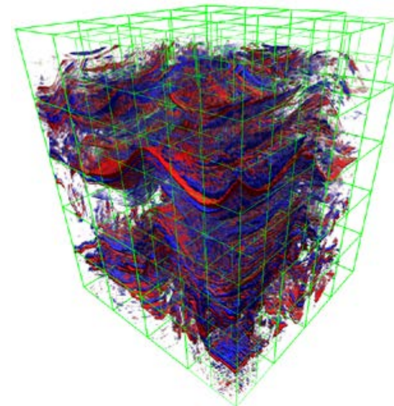
Vector Visualization Basics

SS 2015

Bernd Fröhlich

Virtual Reality and Visualization Research
Faculty of Media
Bauhaus-Universität Weimar

Summary Multi-Resolution Volume Rendering



- Entire volume data does not fit
  - into graphics card’s memory or
  - even into main memory
- Approach
  - Multi-resolution representation of the volume and paging based on bricks
  - On a frame-to-frame basis
    - Select / update the level of detail for each volume region based on a priority function
    - Upload this representation to graphics card
    - Perform rendering based on this representation

Acknowledgements

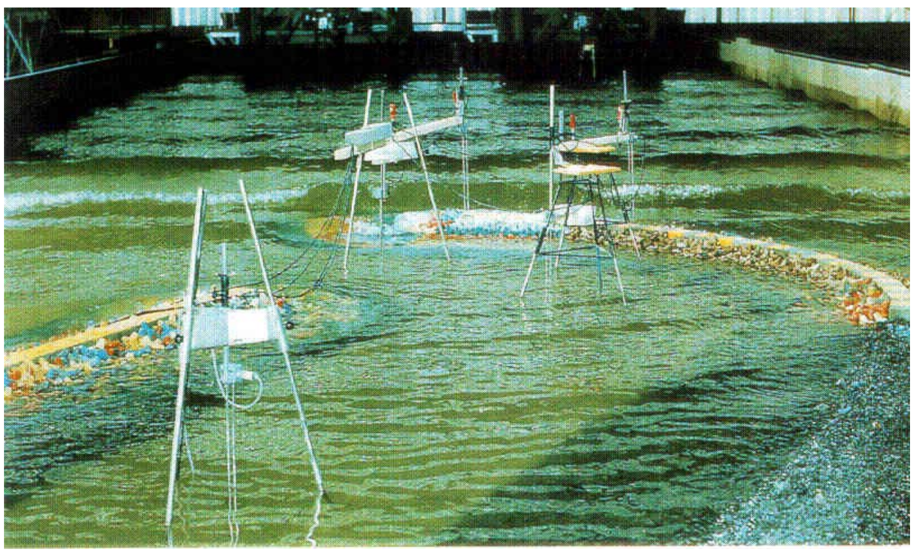
- This lecture is based on
  - Lectures by
    - John Morris, UWA
    - W. Shen, Ohio State University
    - Torsten Moeller, Simon Fraser University, Vancouver
    - Chris Johnson, Scientific Computing and Imaging group at the University of Utah
  - Book Visualisierung by Schumann and Mueller

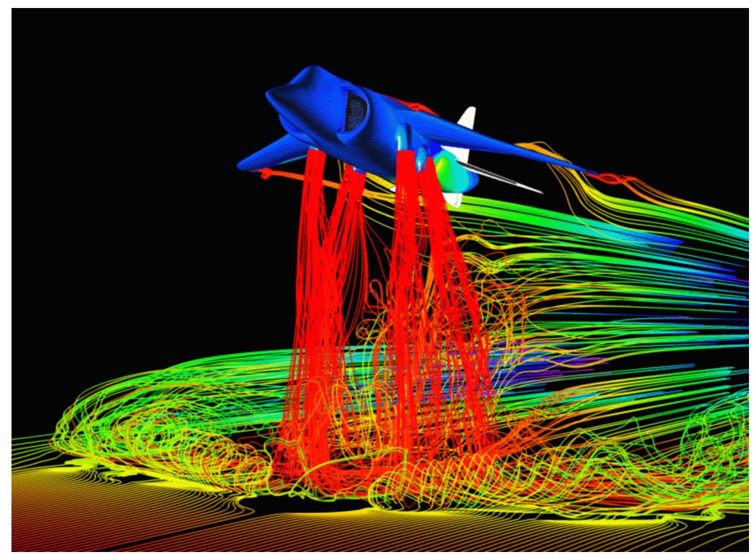
Further Reading

- Visualization Analysis and Design. Munzner. AK Peters / CRC Press, Oct 2014.
  - *Chap 8: Arrange Spatial Data*
- Overview of flow visualization. Weiskopf and Erlebacher. In The Visualization Handbook, edited by Charles Hansen and Christopher Johnson, pp. 261–278. Elsevier, 2005

Vector Visualization - Origin

- Rich field of fluid flow visualization
- Hundreds of years old!!
- Numerical flow simulation
  - CFD - Computational Fluid Dynamics





Flow Visualization

Strömungsvisualisierung

- Gaseous flow
  - Development of cars, aircraft, spacecraft
  - Design of machines - turbines, combustion engines
  - Wind and weather
- Liquid Flow
  - Naval applications - ship design
  - Civil engineering - harbor design, coastal protection
  - Currents in the sea: e.g. Gulf Stream, …
  - Chemistry - fluid flow in reactor tanks
  - Medicine - blood vessels, SPECT, fMRI
- Numerical simulation - most of the time
  - e.g. PDEs, Navier-Stokes equations, …
- Experimental techniques - photographs etc ...

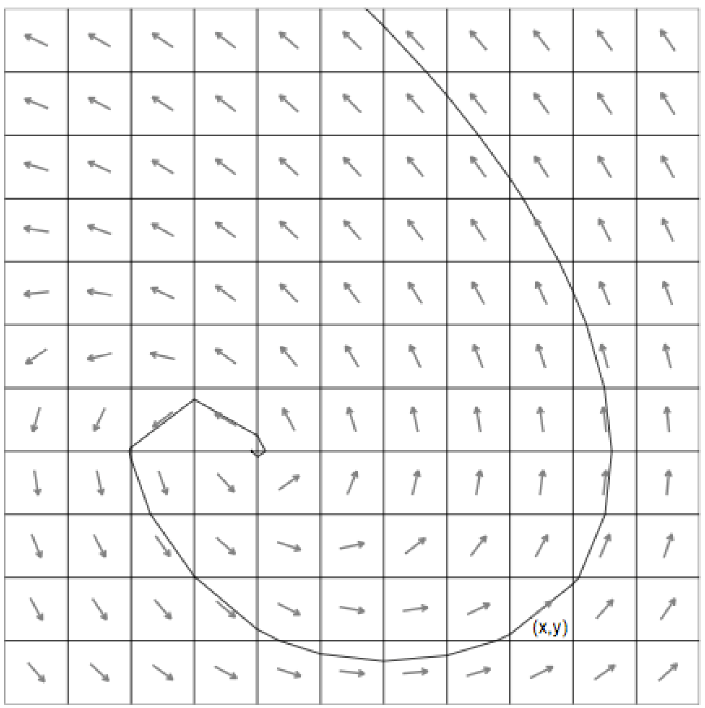
Flow Visualization

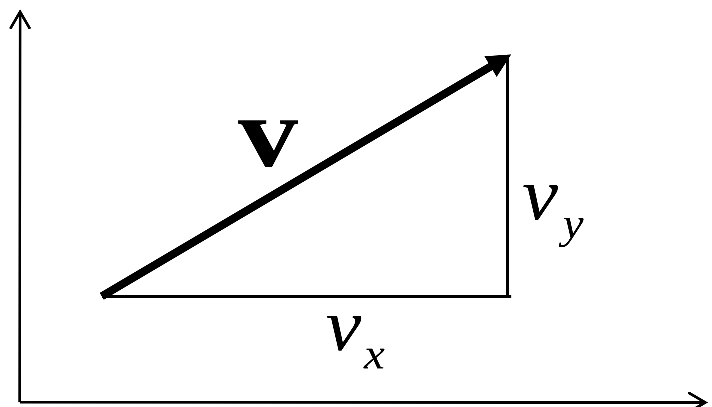
- What is the problem definition?
- Given (typically):
  - Physical position (point in 2D or 3D)
  - Velocity (vector in 2D or 3D)
  - Additionally:
    - pressure (scalar)
    - density (scalar)
    - entropy (scalar)
- Steady flow - vector field is constant
  - “stationäres Vektorfeld”
- Unsteady - vector field changes with time
  - “instationäres Vektorfeld”
  - “zeitveränderliches Vektorfeld”

Velocity Fields

- Vectors represent
  - Direction of flow
  - Velocity of flow in each direction (unit: m/s, km/h, …)
  - Total velocity $|\mathbf{v}| = \sqrt{v_x^2 + v_y^2}$ (unit: m/s, km/h, …)

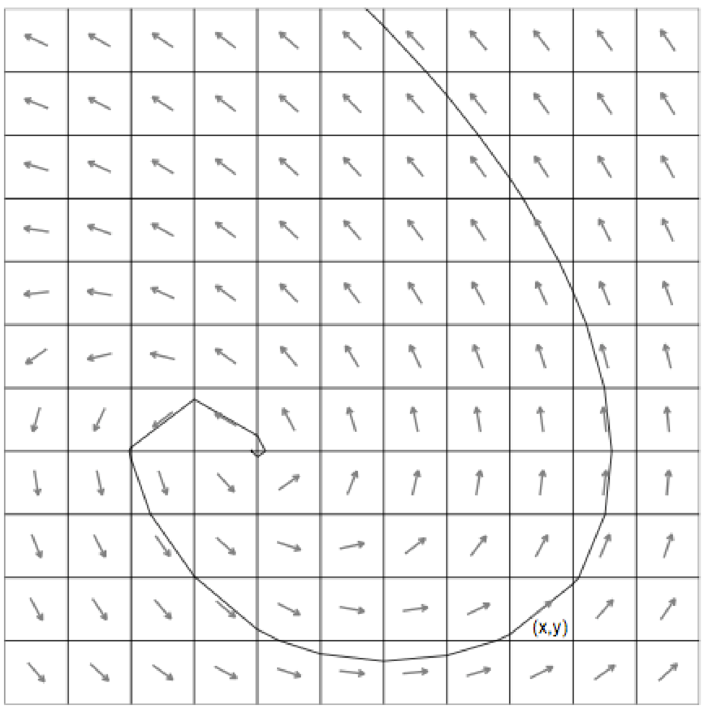
$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

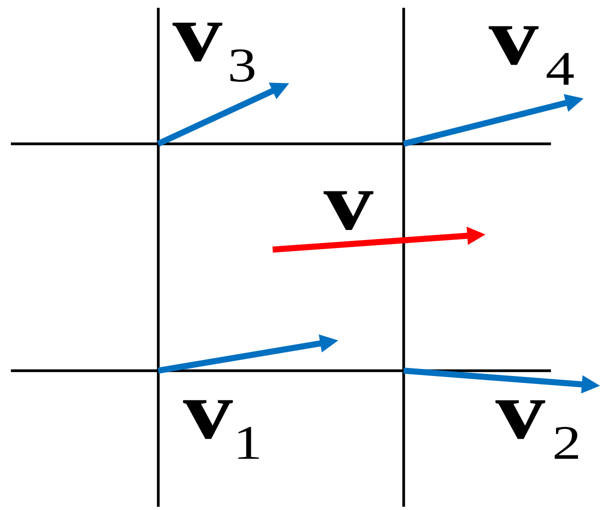




Velocity Fields

- Velocity grids
  - Computing a velocity at an arbitrary location requires interpolation (bi-linear, tri-linear, quad-linear)





$$\mathbf{v} = (1-v)((1-u)\mathbf{v}_1 + u(\mathbf{v}_2)) + v((1-u)\mathbf{v}_3 + u(\mathbf{v}_4))$$

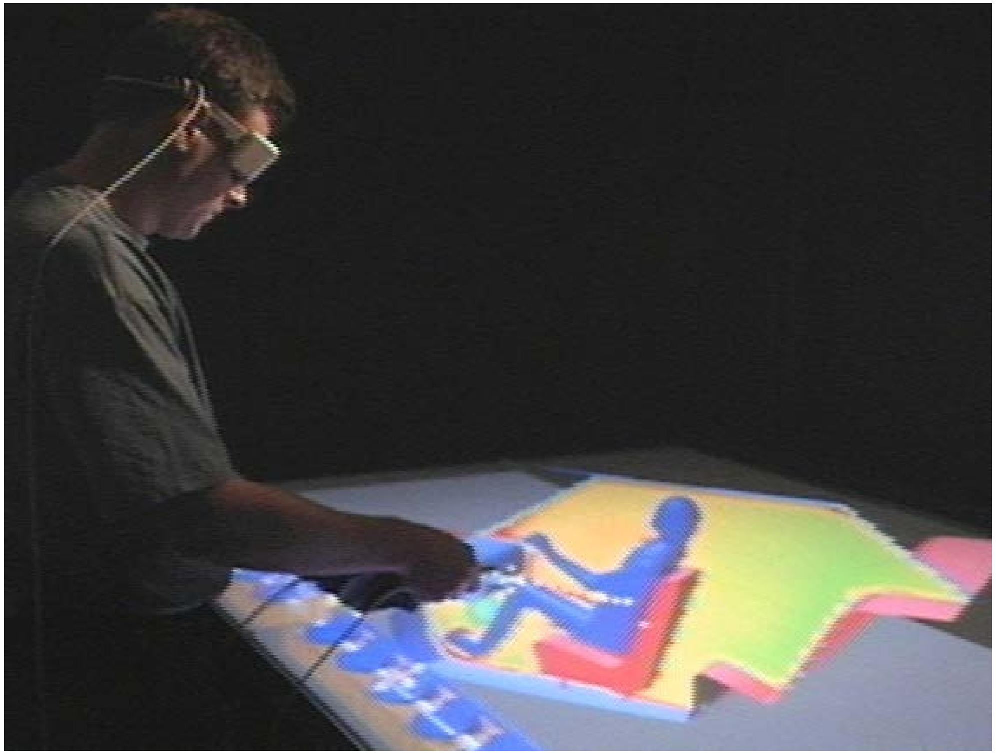
Traditionally: Experimental Flow Visualization

- Purpose
  - Get an impression of flow around a scale model of a real object
  - As a source of inspiration for the development of new and better theories
  - To verify a new theory or model

- Three basic techniques
  - Adding foreign material (smoke, oil, …)
  - Optical techniques (glowing particles, …)
  - Adding heat and energy

- Problems:
  - The flow is affected by experimental technique
  - Not all phenomena can be visualized
  - Expensive (wind tunnels, small scale models)
  - Time consuming

Visualization of Simulations



Air conditioning simulation: temperature and flow

Wind tunnel simulation

Vector Visualization - Pipeline

Data Generation (Acquisition)

↓

Data Preparation and Selection
vector magnitude, vorticities, …

↓

Visualization Mapping
(Transformation)

↓

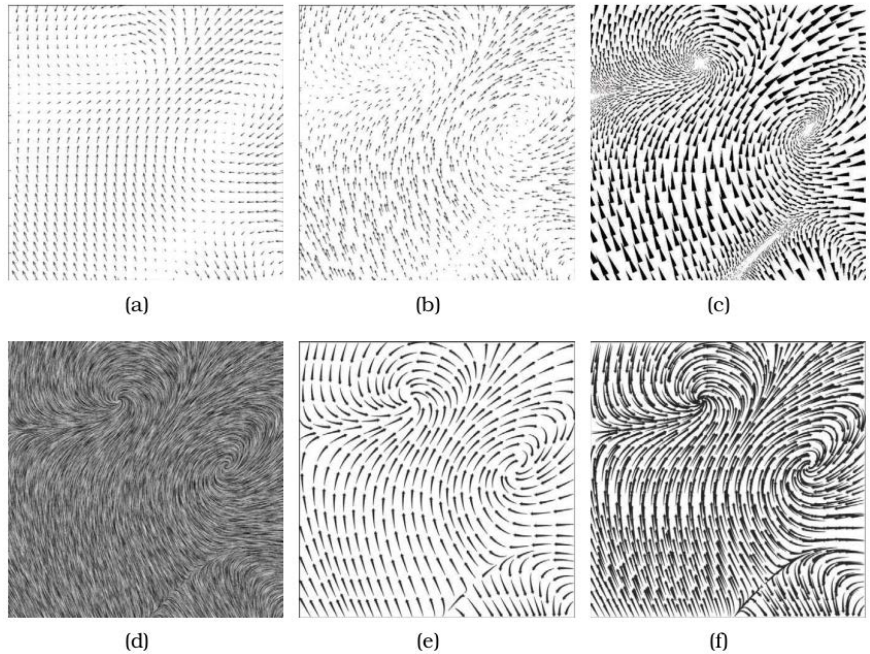
Rendering (Shading, Compositing)

↓

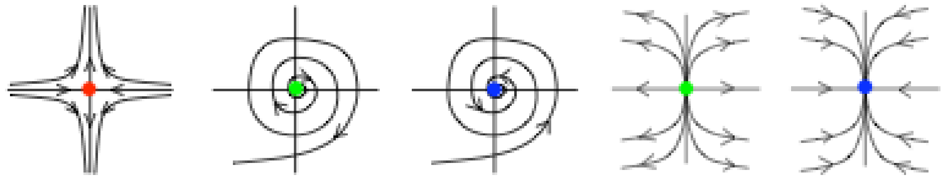
Display

Vector Fields

- Data
  - Vector per cell
- Visualization techniques
  - Flow glyphs
    - Purely local
  - Geometric flow
    - Trace particle trajectories
    - Sparse set of seed points
  - Texture flow
    - Dense seeds
  - Flow features
    - Global computation to detect features
    - Encoded with one of methods above



[Comparing 2D vector field visualization methods: A user study. Laidlaw et al. IEEE Trans. Visualization and Computer Graphics (TVCG) 11:1 (2005), 59–70.]

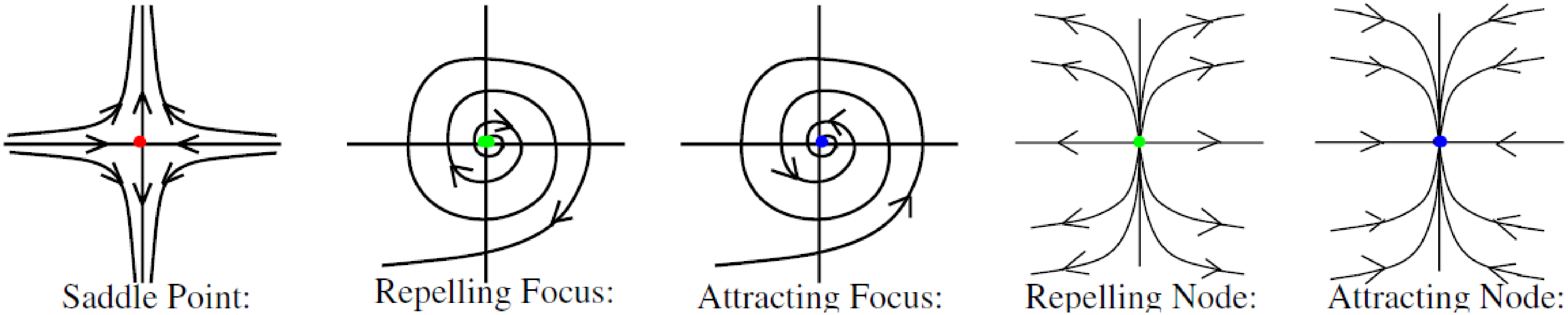


[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. Computers & Graphics 26:2 (2002), 249–257.]

Figure on Previous Slide

- Local glyph techniques
  - a) arrow glyphs on a regular grid.
  - b) arrow glyphs on a jittered grid.
  - c) triangular wedge glyphs inspired by oil painting strokes
- Dense texture technique
  - d) LIC: dense texture-based idiom
- Geometric technique using streamlines
  - e)curved arrow glyphs with image-guided streamline seeding.
  - f) curved arrow glyphs with regular grid streamline seeding.

Vector Fields



[Topology tracking for the visualization of time-dependent two-dimensional flows. Tricoche, Wischgoll, Scheuermann, and Hagen. Computers & Graphics 26:2 (2002), 249–257.]

- Critical points
  - A critical point is a position in a flow field domain where the velocity vanishes.
  - Sinks: attracting nodes and foci
  - Sources: repelling nodes and foci

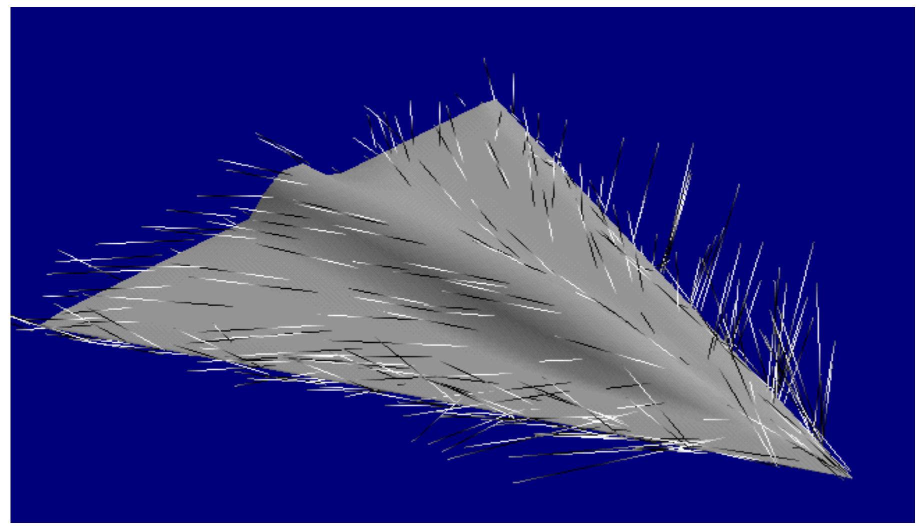
- Tasks in an empirical study
  - Finding critical points and identifying their types
  - Identifying what type of critical point is at a specific location
  - Predicting where a particle starting at a specified point will end up (advection)
  - Result: arrow glyphs are worst

Vector Visualization - Mappings

- Local glyph techniques
  - Hedgehogs/glyphs/Icons
- Geometric technique using streamlines
  - Particle tracing
  - Stream-, streak-, time- and path-lines
  - Stream-ribbon, stream-surfaces, stream-polygons, stream-tube
- Dense texture technique
  - Line integral convolution

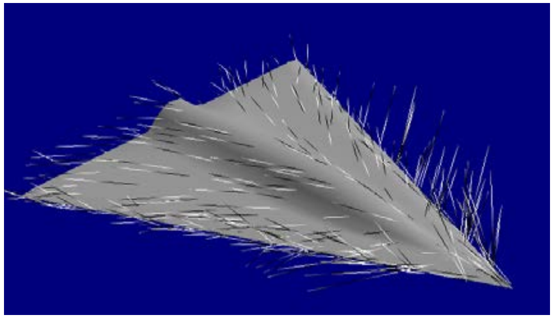
Hedgehogs and Glyphs

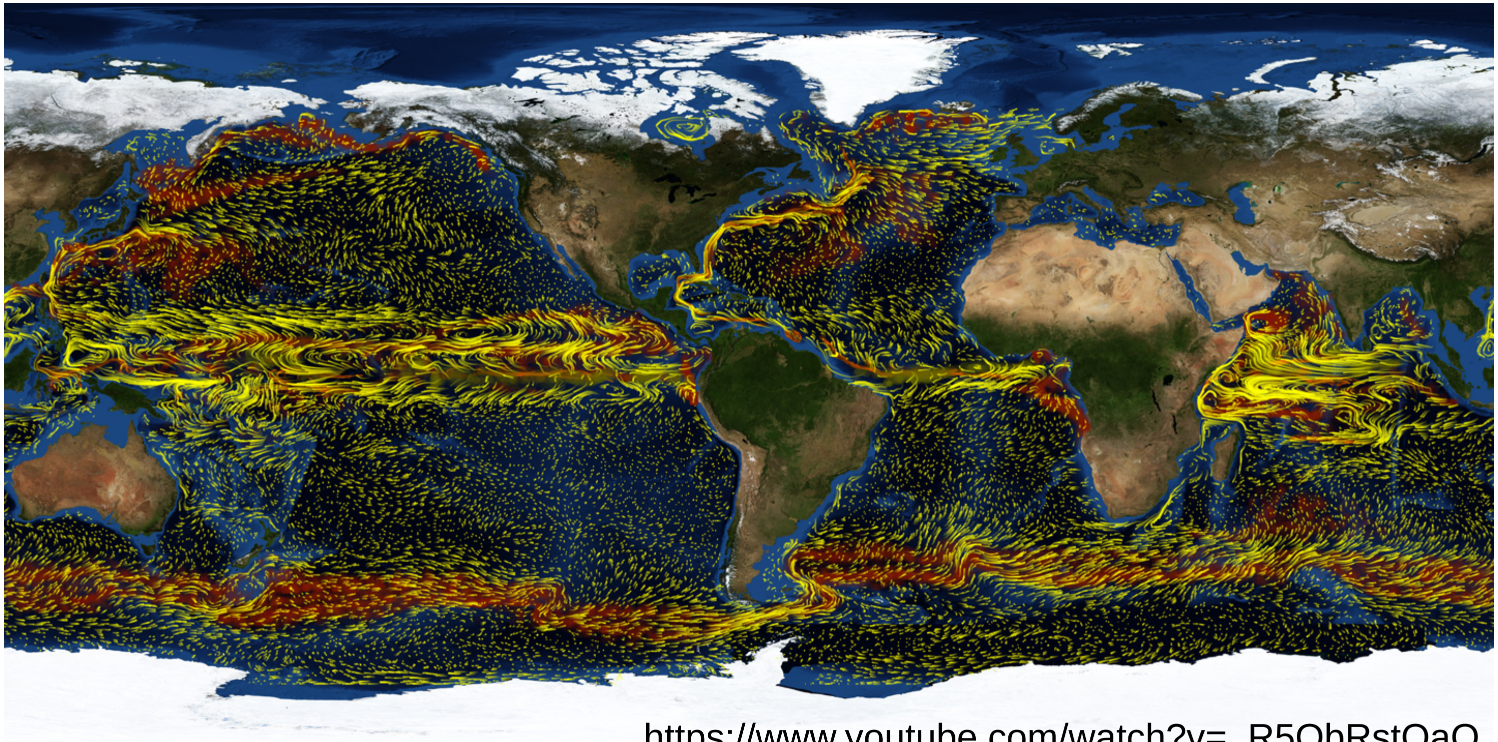
- Analogous to tufts or vanes from experimental flow visualization
- Clutter the image very quick
- Work reasonably well in 2D
- Not very informative



Hedgehogs and Glyphs

- Put "icons" at certain places in the flow
- Hedgehogs use arrow icons
  - Represent direction & magnitude
- Other primitives are possible
  - 3D primitives (glyphs)

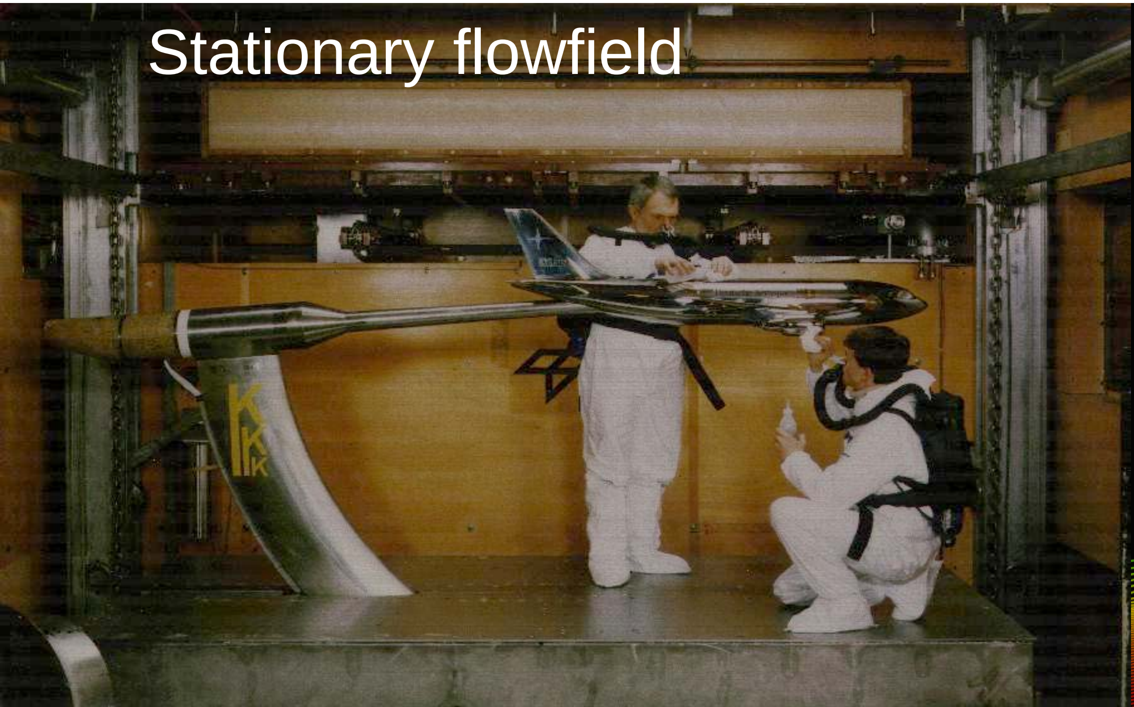


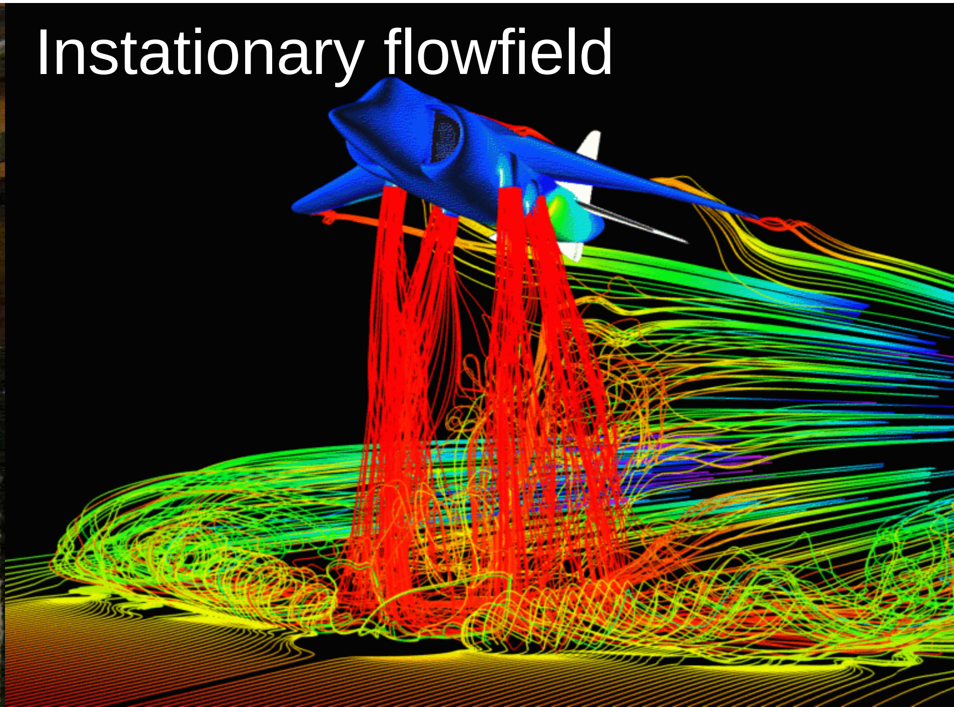


https://www.youtube.com/watch?v=_R5ObRstQaQ

Stream Lines

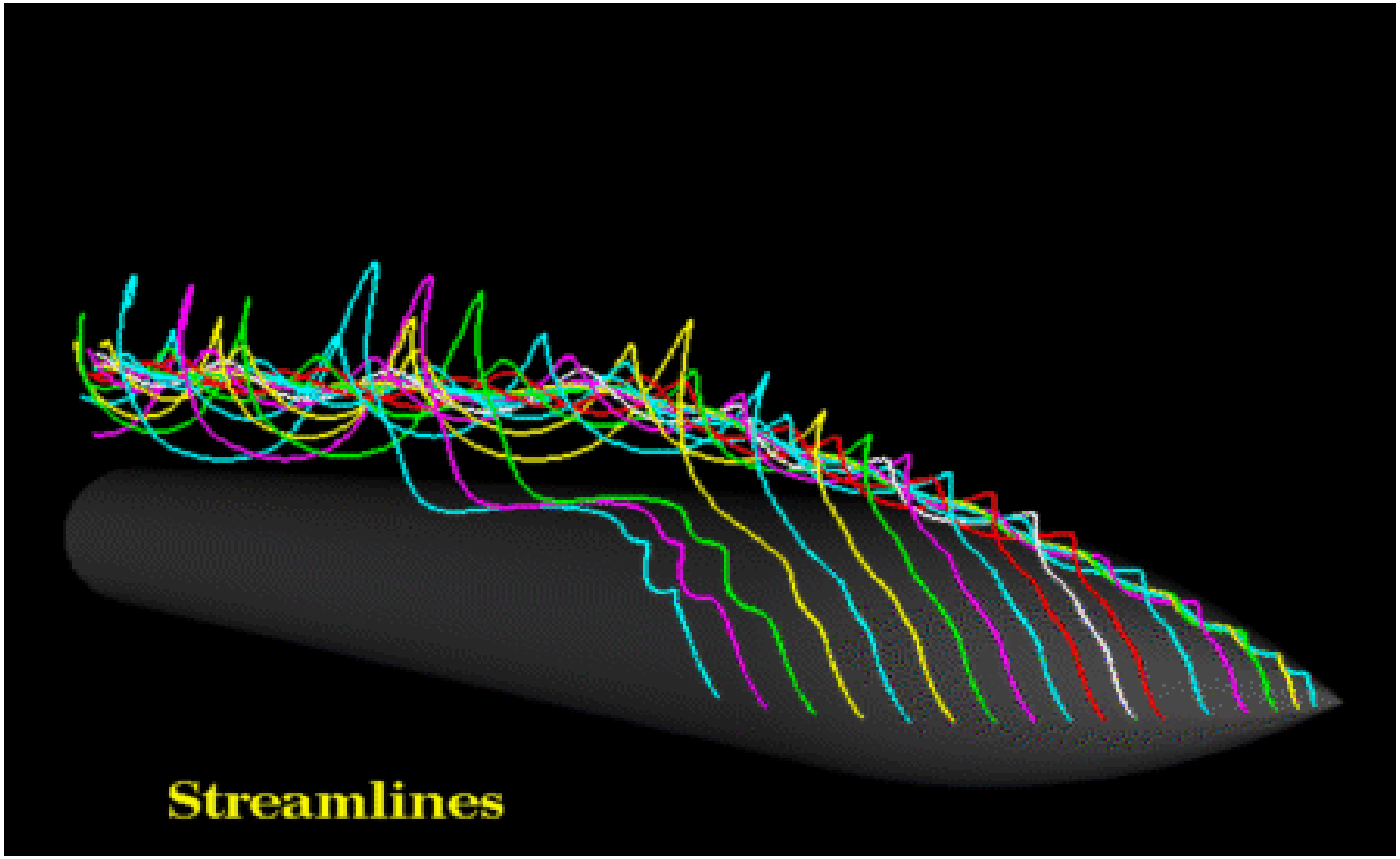
- Curves that connect the particle trace through a steady flow field
- Lines that are everywhere tangent to the velocity field
- Real experiment: insert glowing particle into field and keep the shutter of the camera open for the duration of the particle flight





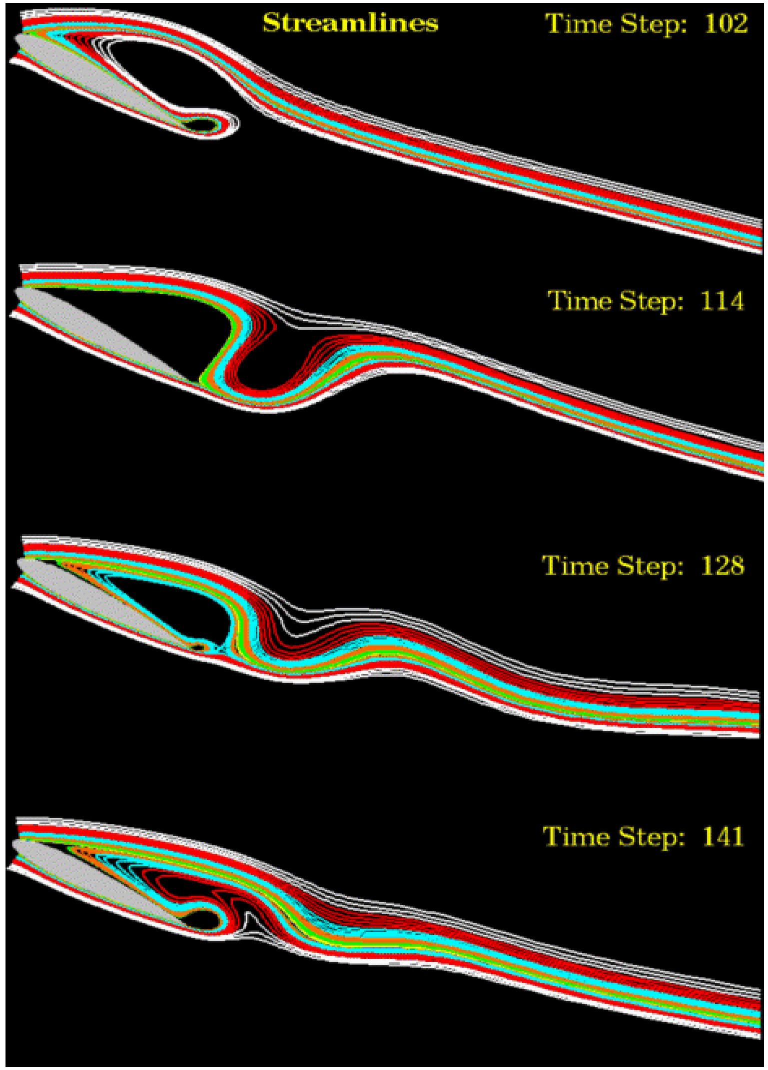
Stream Lines

- Everywhere tangent to the steady flow
- Only defined for steady flow fields



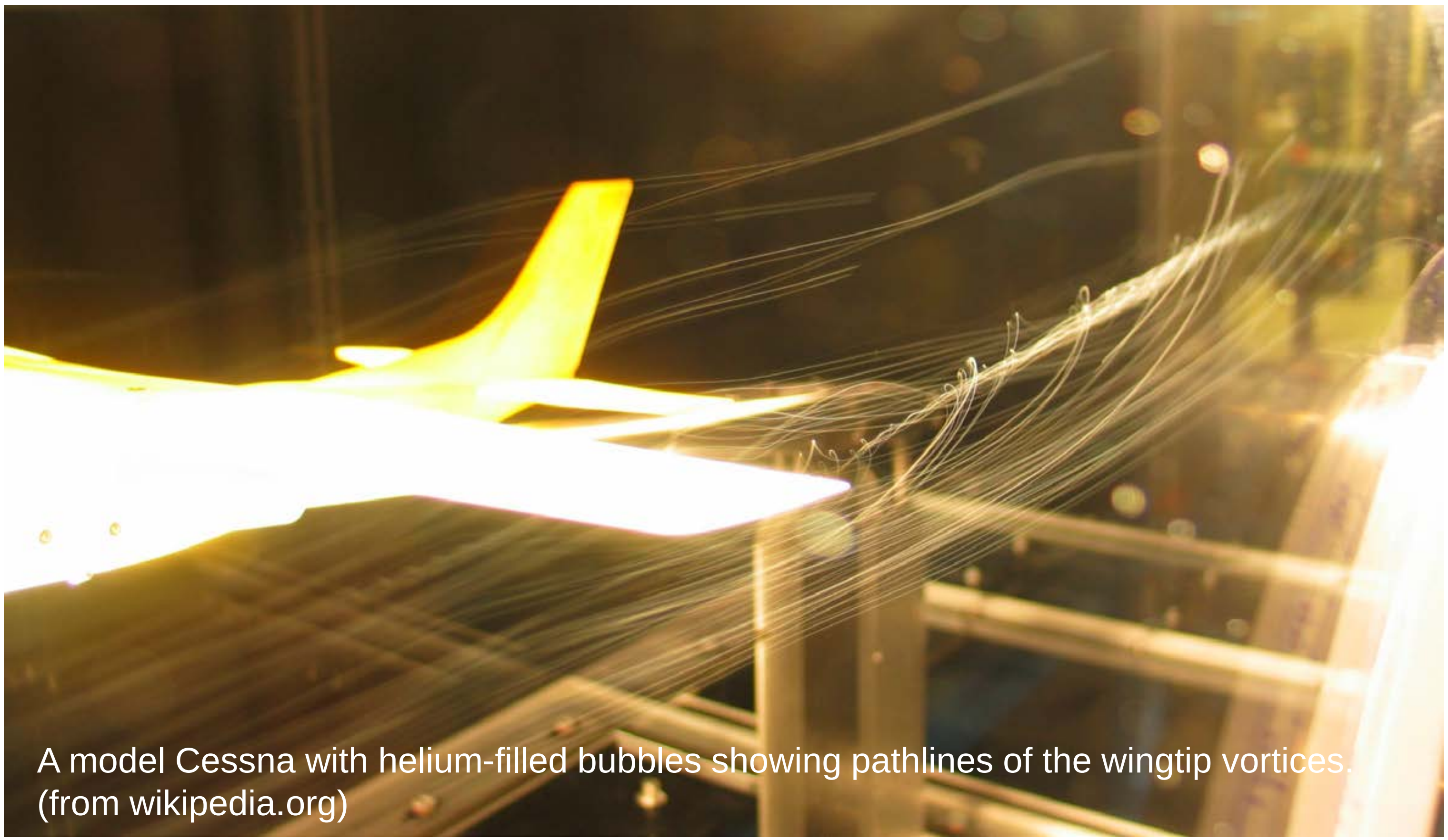
Stream Lines

- Streamlines showing the flow field of a fixed time step but the flow of an instationary flow field would never look like that

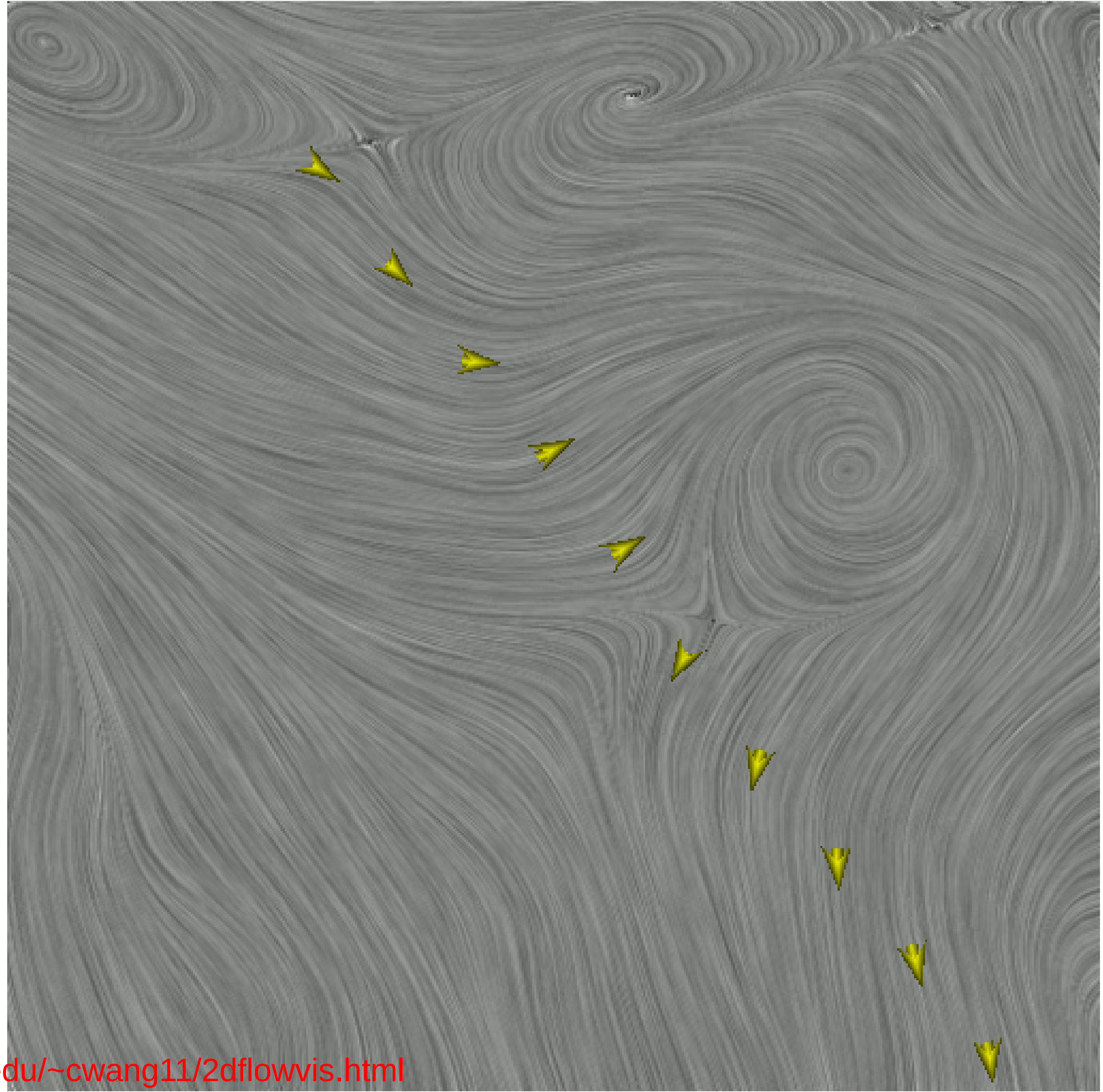


Path Lines

- Path of a particle in the flow field (fluid, gas). Imagine a light emitting particle in the flow. A path line is obtained when a photographic plate is exposed for several seconds.
- Collection of particle traces gives sense of time evolution of flow



A model Cessna with helium-filled bubbles showing pathlines of the wingtip vortices.
(from wikipedia.org)

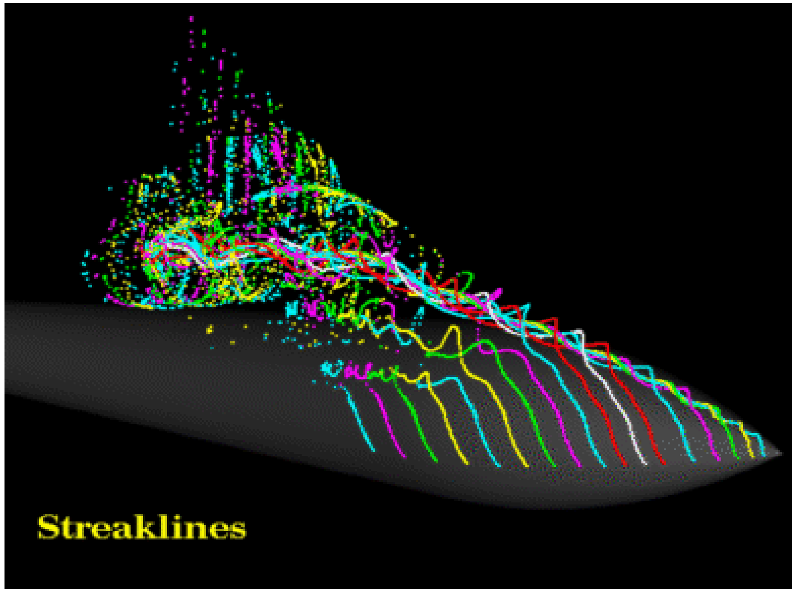


http://www3.nd.edu/~cwang11/2dflowvis.html

Streak Lines

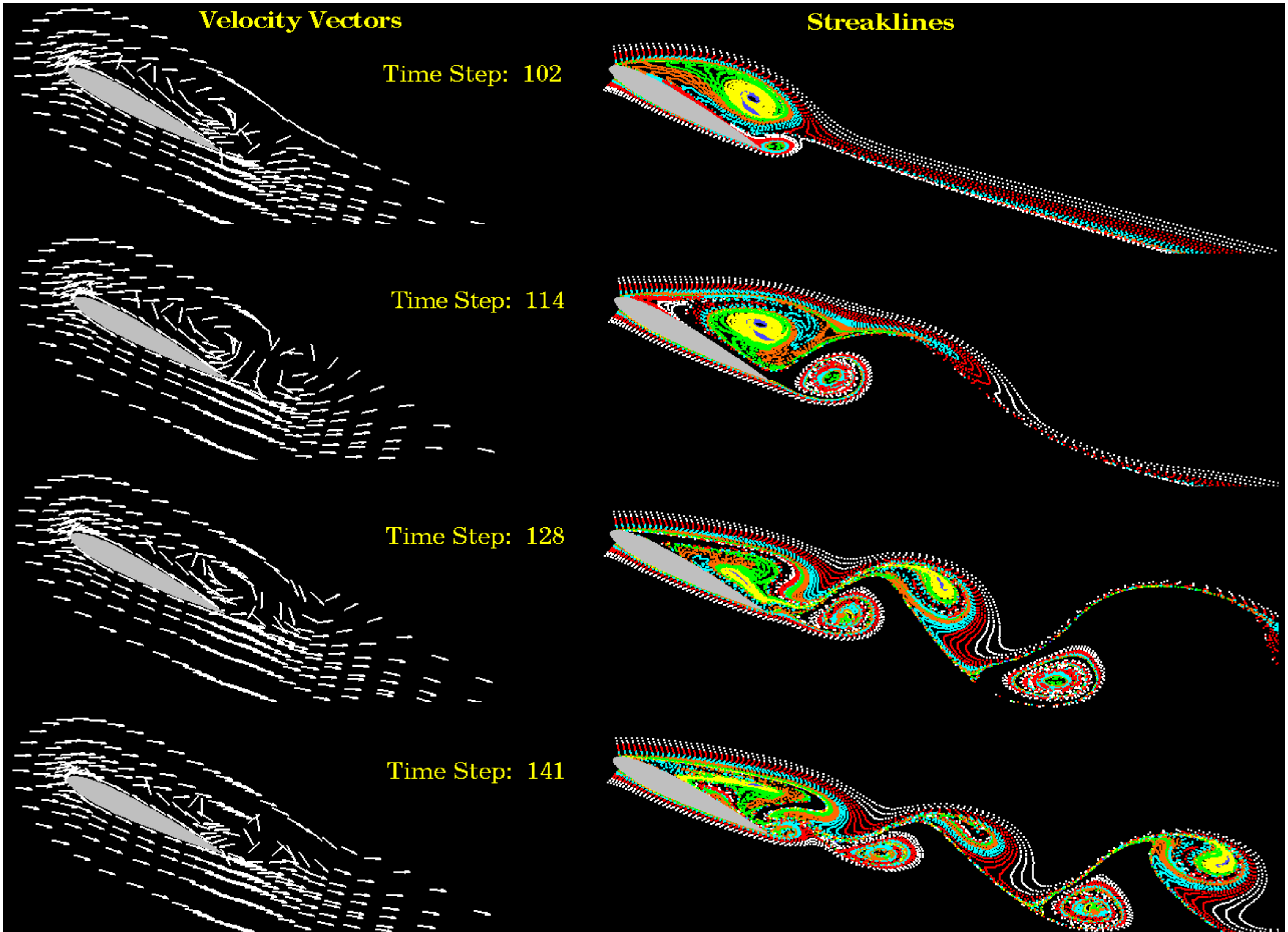
German: Streichlinien

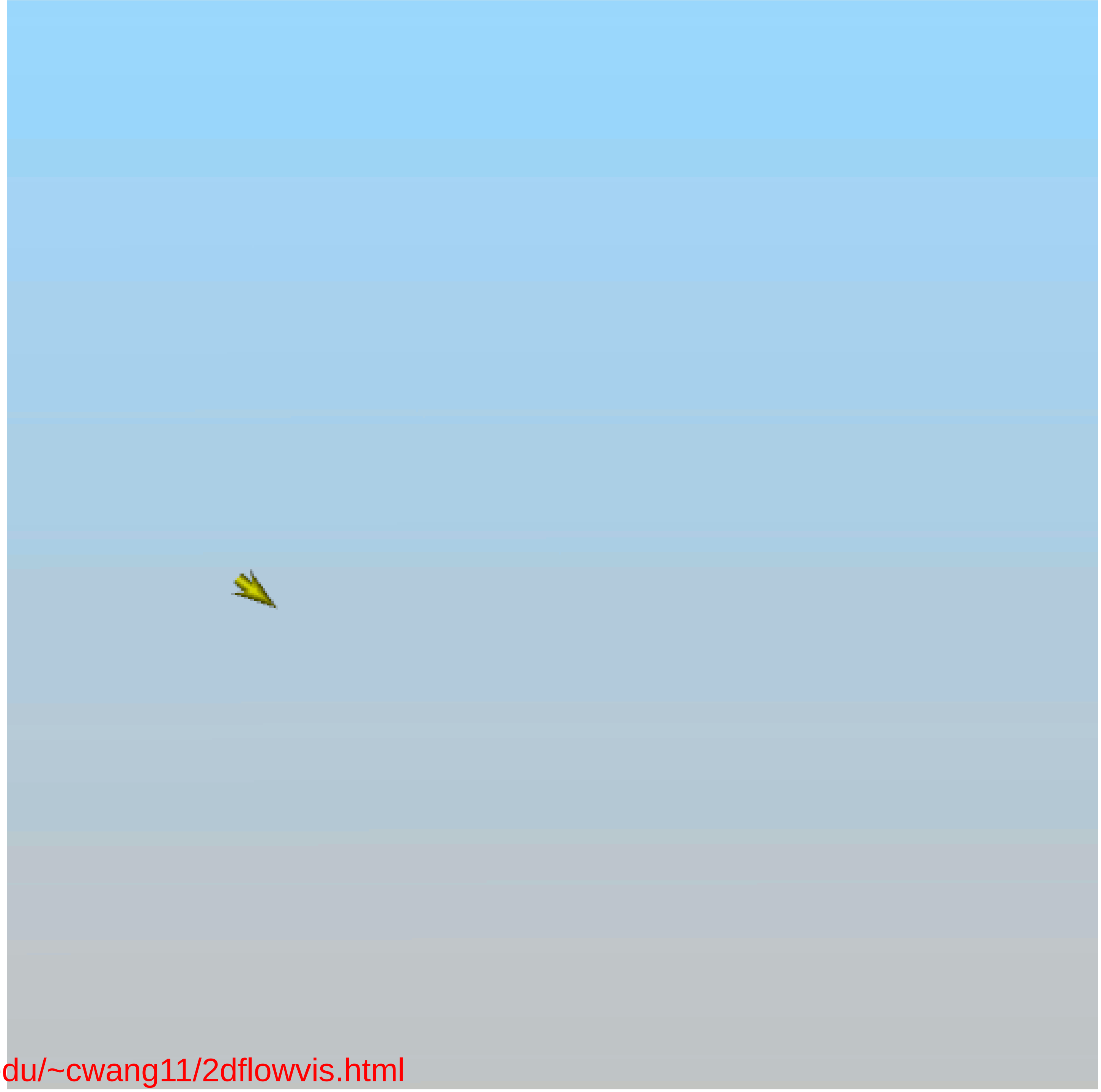
- Real world practice: Injecting dye for a period of time gives a line of dye in the fluid, from which the fluid flow can be seen.
- Created by injecting dye in the flow from a fixed position x_0.
- Location at time t_0 of all fluid elements that have previously passed through the position x_0
- Information of the past history of the flow
- Identical to stream lines for steady flow



Streak Lines

dye is injected in the flow from a fixed position

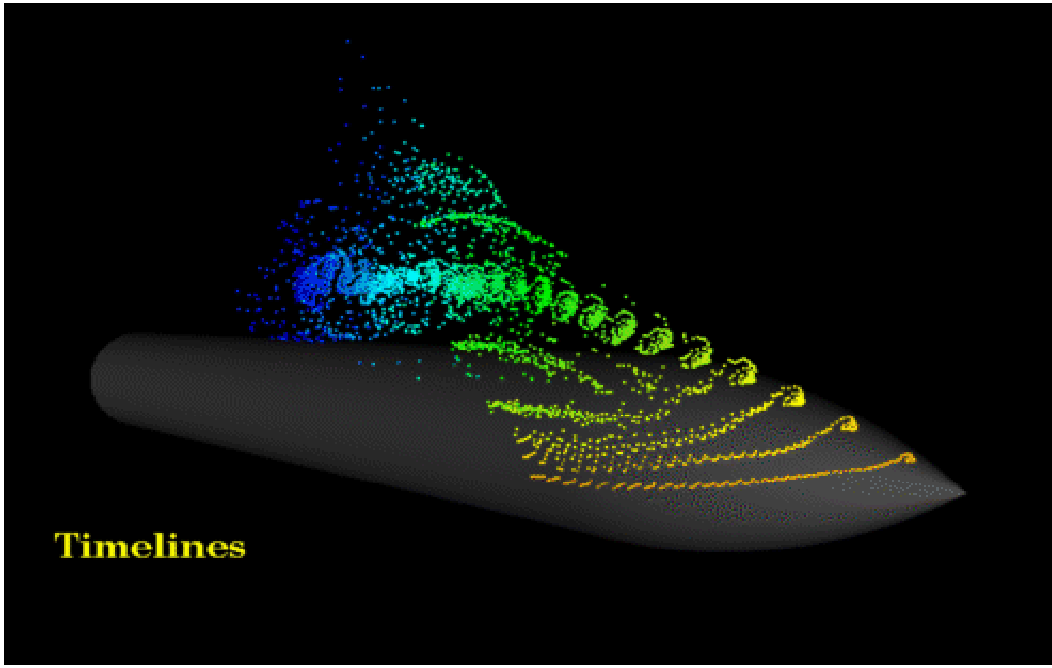


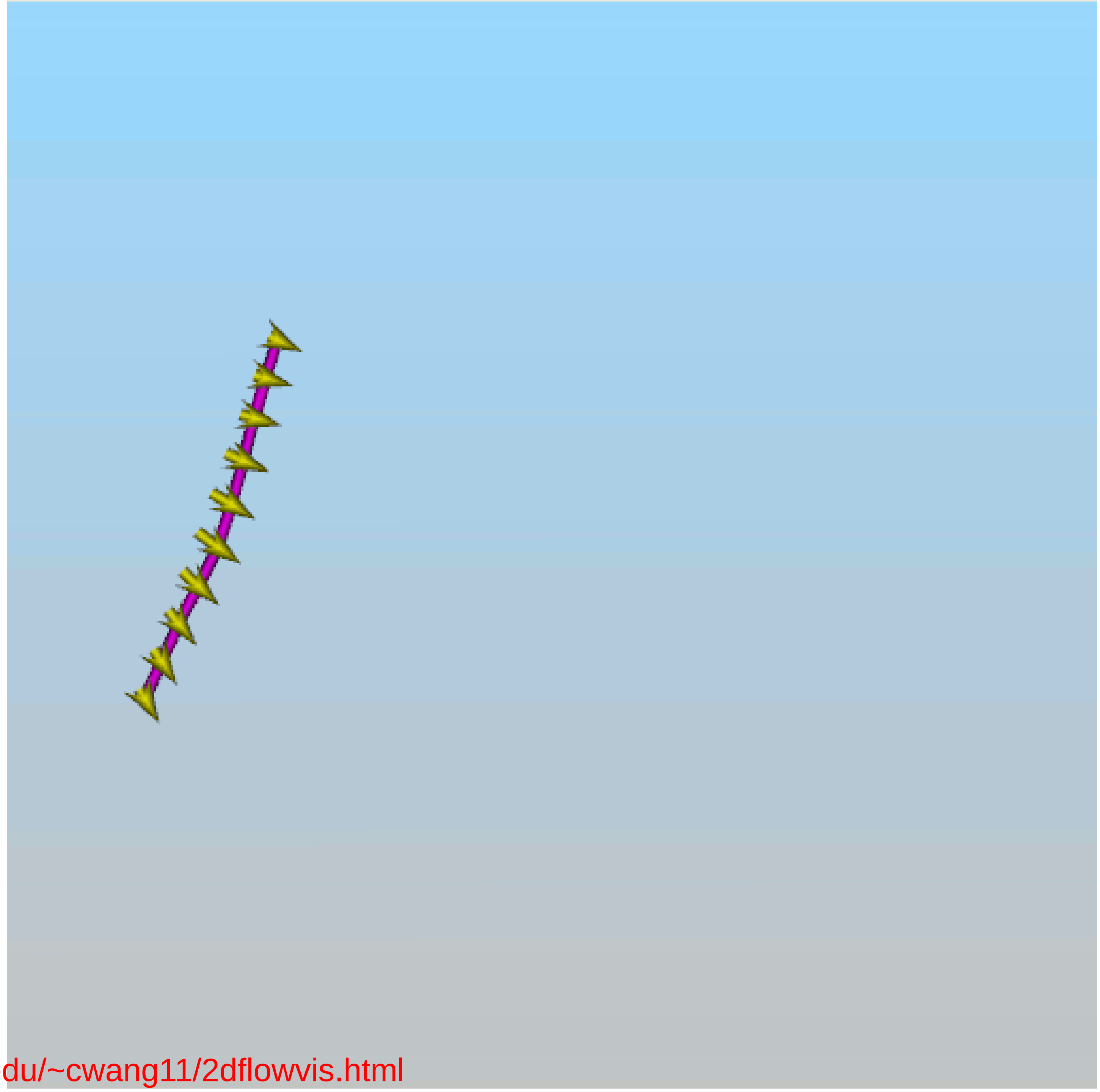


http://www3.nd.edu/~cwang11/2dflowvis.html

Time Lines

- Lines of particles that once released in the fluid, are moved and transformed by the flow. The motion and formation of the line, which is often released perpendicular to the flow, shows the flow.
- Real world experiments - often consist of row of small particles, such as hydrogen bubbles, which are simultaneously released into the flow.

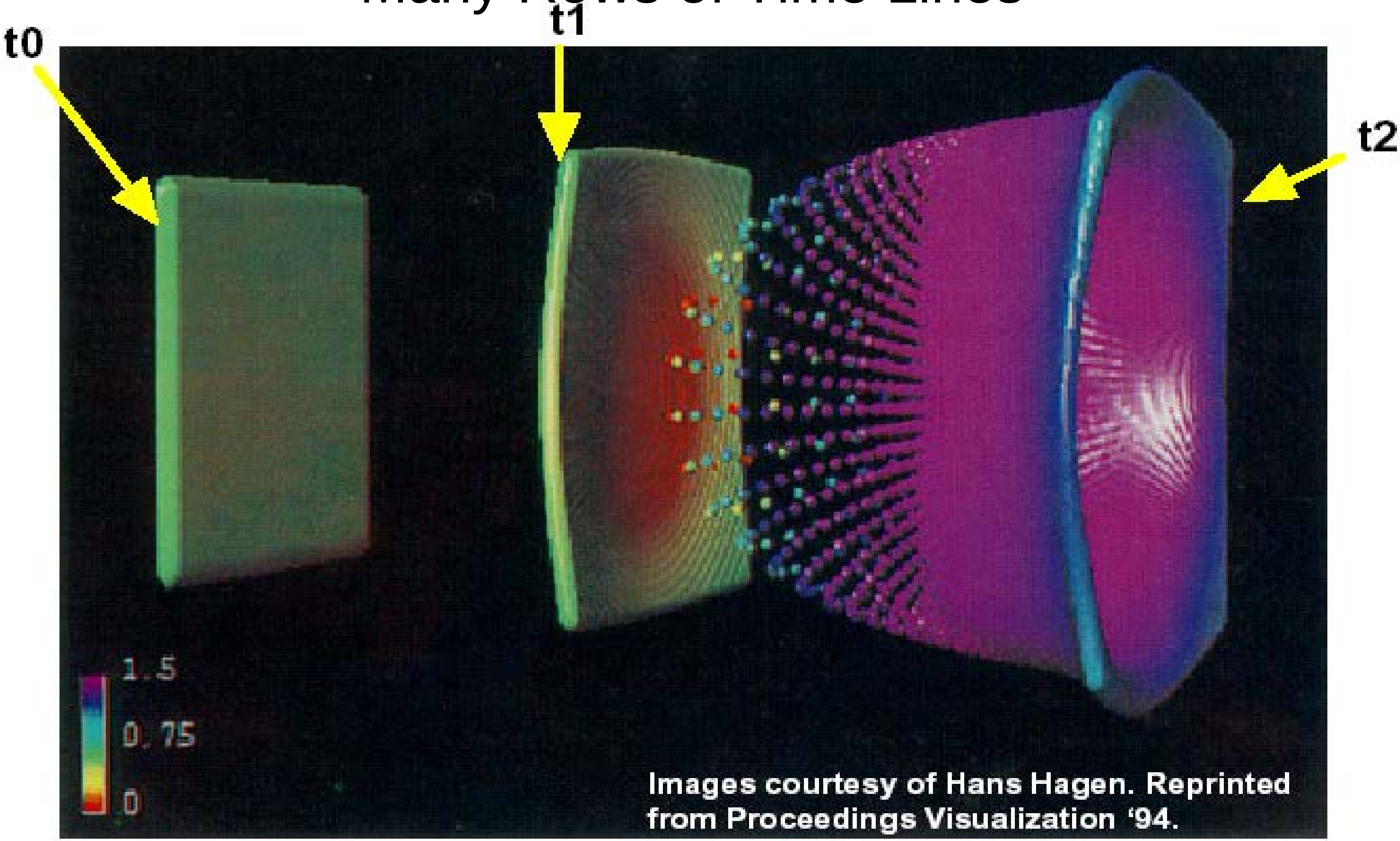




http://www3.nd.edu/~cwang11/2dflowvis.html

Time Surface

Many Rows of Time Lines

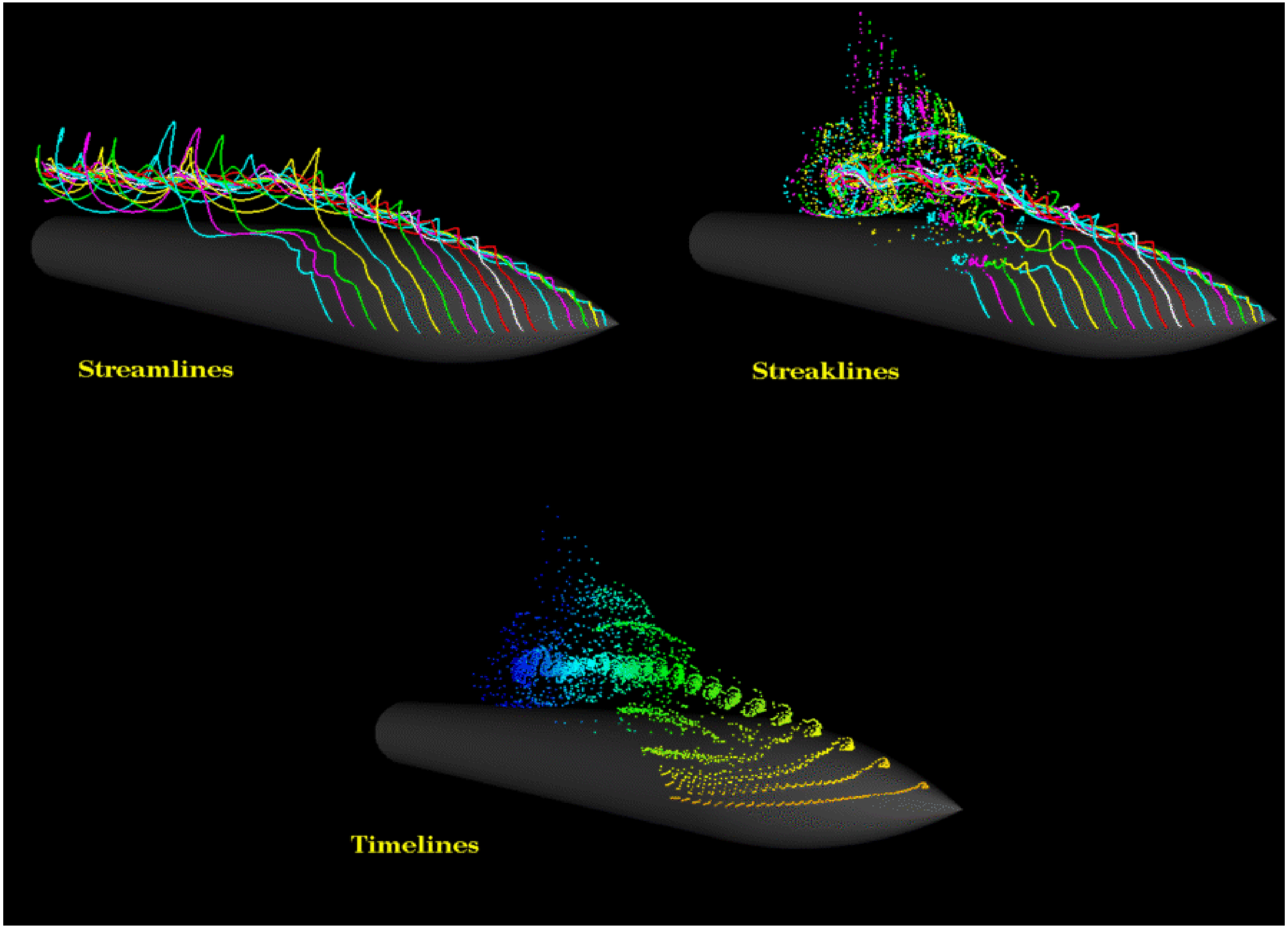


Overview Flow Vis Line Techniques

- **Stream line:** Curves that connect the particle trace through a **steady** flow field
- **Streak line:** generated by continuously injecting particles from a fixed location.
- **Path line:** generated by tracing the path of a single particle (also called a particle path).
- **Time line:** generated by tracing a line of particles which are all released at the same time.

For **Steady flows:** path and streak lines are identical to stream lines - lines that are everywhere tangent to the velocity field.

Comparison of Mappings



Stream Ribbon

- We really would like to see vorticities, i.e. places were the flow twists.
- A point primitive or an icon can hardly convey this
- Idea: trace adjacent particles and connect them with polygons
- Shade those polygons appropriately and one will detect twists

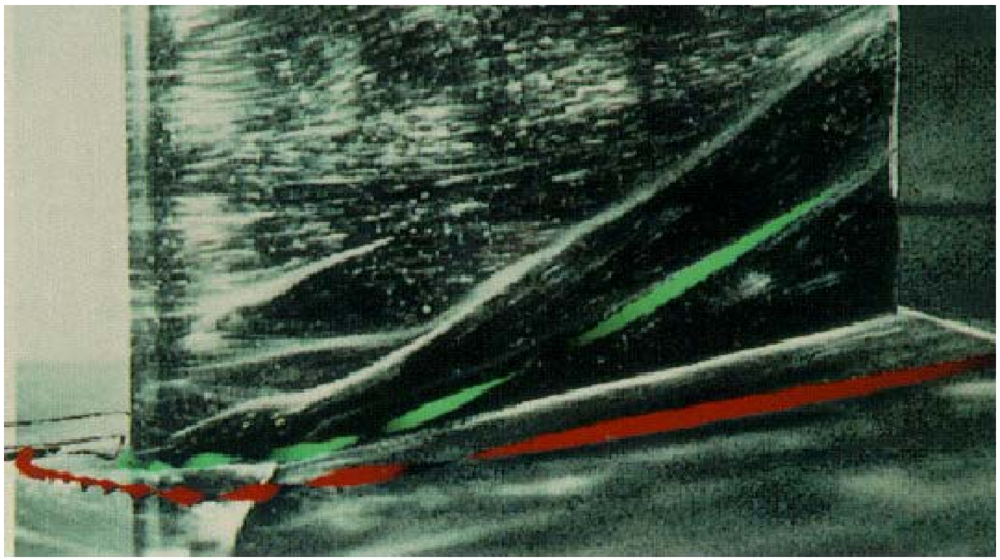
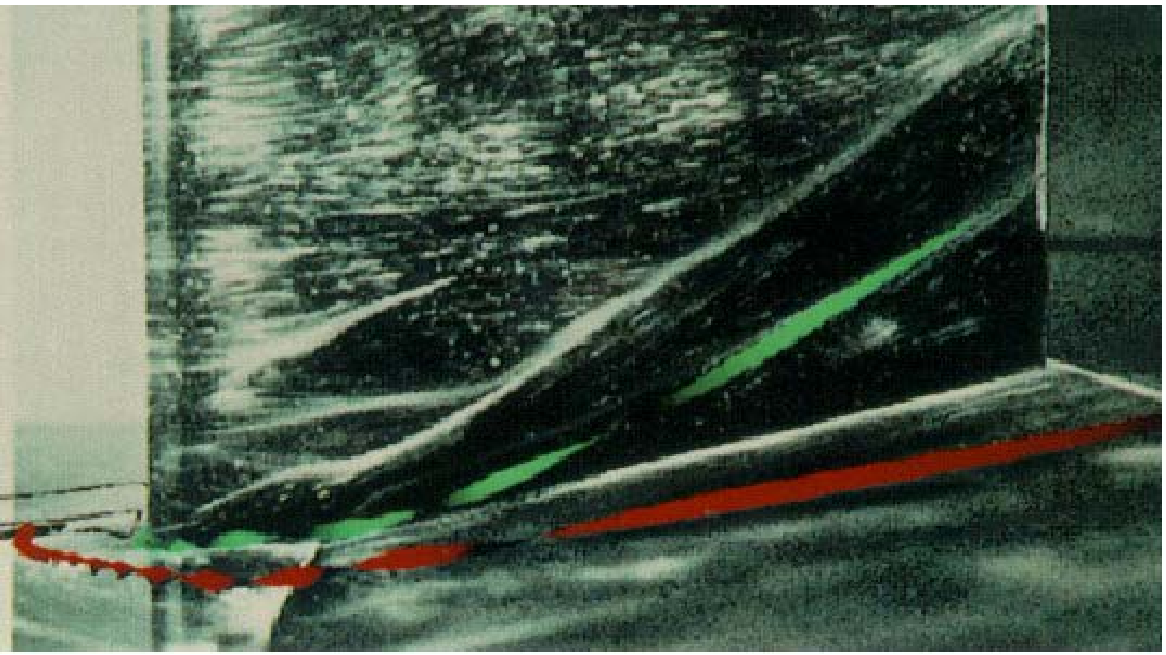


Image courtesy of H. Pagendarm, DLR.

Stream Ribbon

- Problem - when flow diverges
- Solution: Just trace one streamline and a constant size vector with it



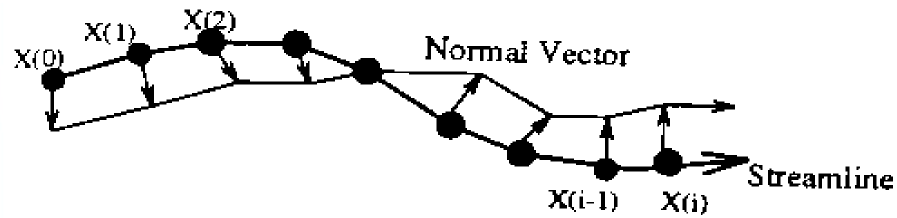
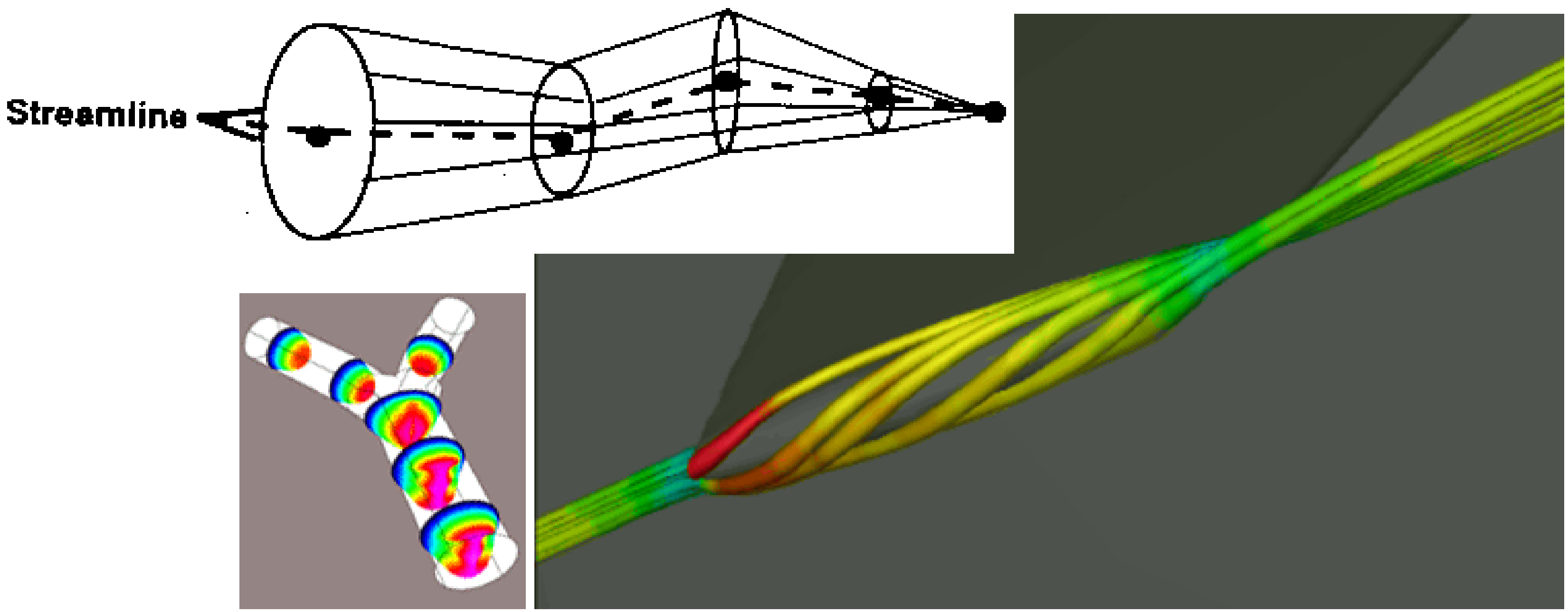


Image courtesy of H. Pagendarm, DLR.

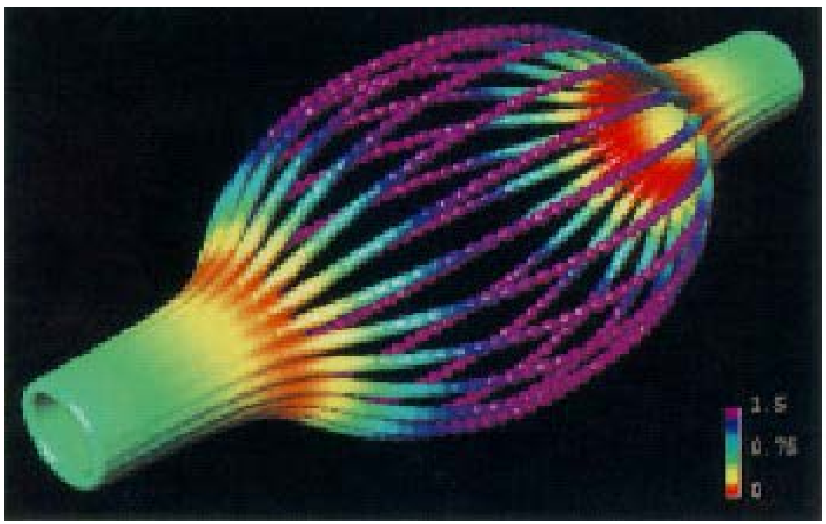
Stream Tube

- Generate a stream-line and connect circular crossflow sections along the stream line



Stream Balls

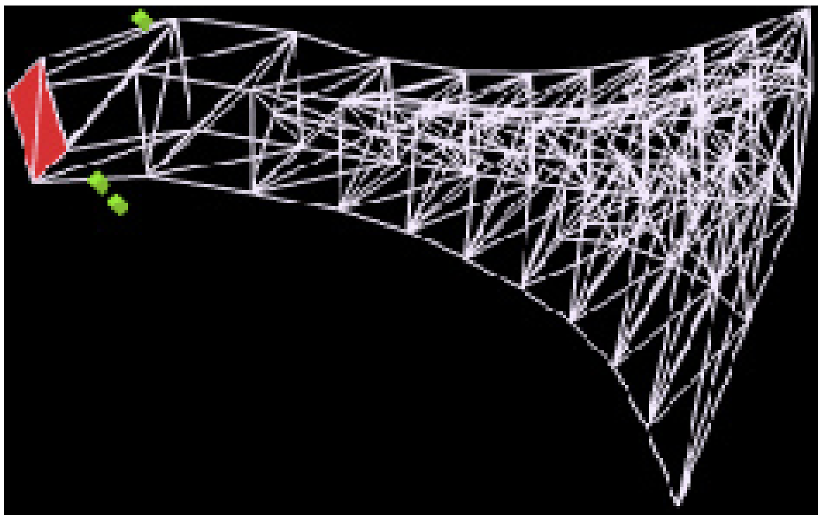
- Another way to get around diverging stream-lines
- Simply put implicit surface primitives at particle traces - at places where they are close they'll merge elegantly and form tubes

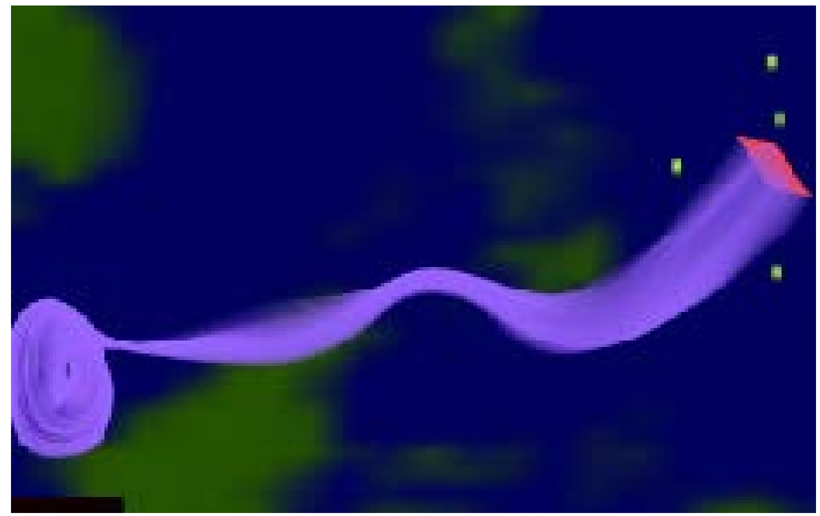


Images courtesy of Hans Hagen. Reprinted from Proceedings Visualization '94.

Mappings - Flow Volumes

- Instead of tracing a line - trace a small polygon





Particle Tracing
(Particle Advection)

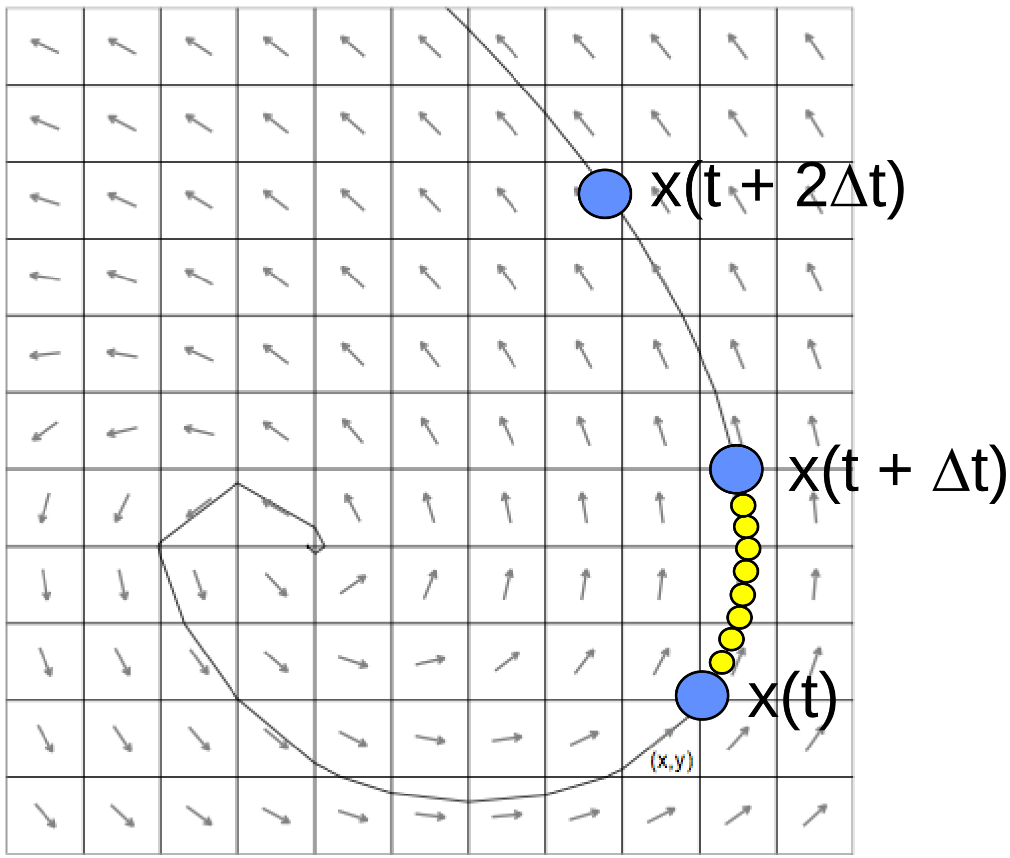
- We need to solve

$$\frac{dx}{dt} = v\big(x(t)\big) \quad \text{or} \quad x(t+\Delta t) = x(t) + \int_{t}^{t+\Delta t} v(x(\tau))\,d\tau$$

- **v**(x(t)) : particle velocity at a certain position in the vector field and at a certain point in time t
- **x**: position of a particle
- Numerical integration used: Euler, Runge-Kutta, …

Particle Advection

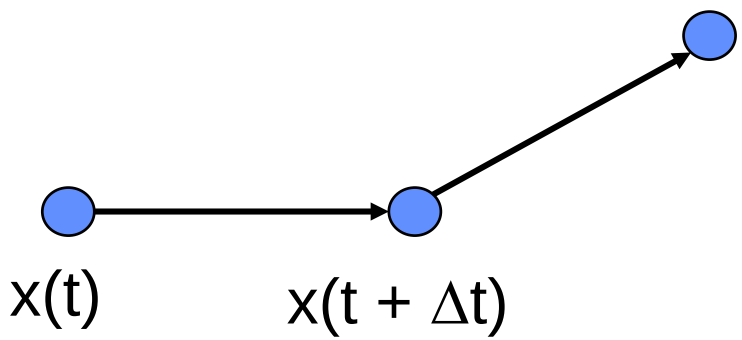
$$x(t+\Delta t) = x(t) + \int_{t}^{t+\Delta t} v(x(\tau))\,d\tau$$



Euler

$$x(t+\Delta t) = x(t) + v(x(t))\Delta t$$

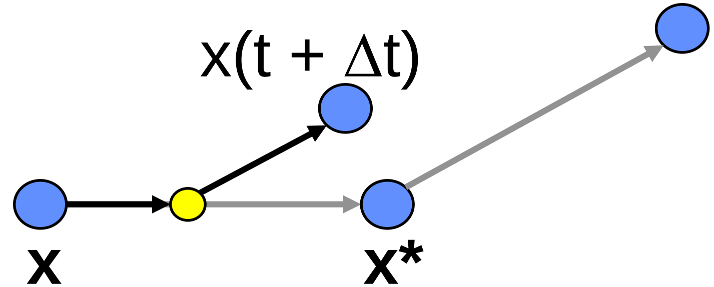
- Often not good enough, need to resort to higher order methods

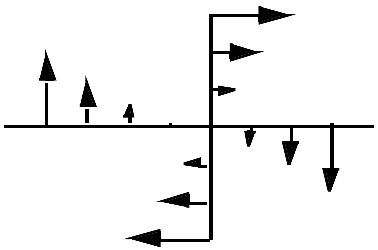


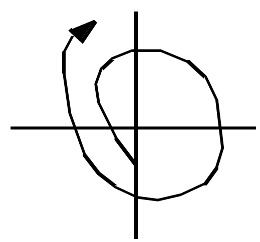
Second Order Runge-Kutta

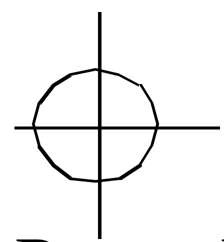
$$x^{*}(t+\Delta t) = x(t) + v(x(t))\Delta t$$

$$x(t+\Delta t) = x(t) + \frac{\left(v(x(t)) + v\left(x^{*}(t+\Delta t)\right)\right)}{2}\Delta t$$




vector field


Euler


Runge-Kutta

4th Order Runge-Kutta

- Often used – samples the flow field at 4 locations

$$x(t+\Delta t) = x_0 + \tfrac{1}{6}\left(v(x_0) + 2v(x_1) + 2v(x_2) + v(x_3)\right)\Delta t$$

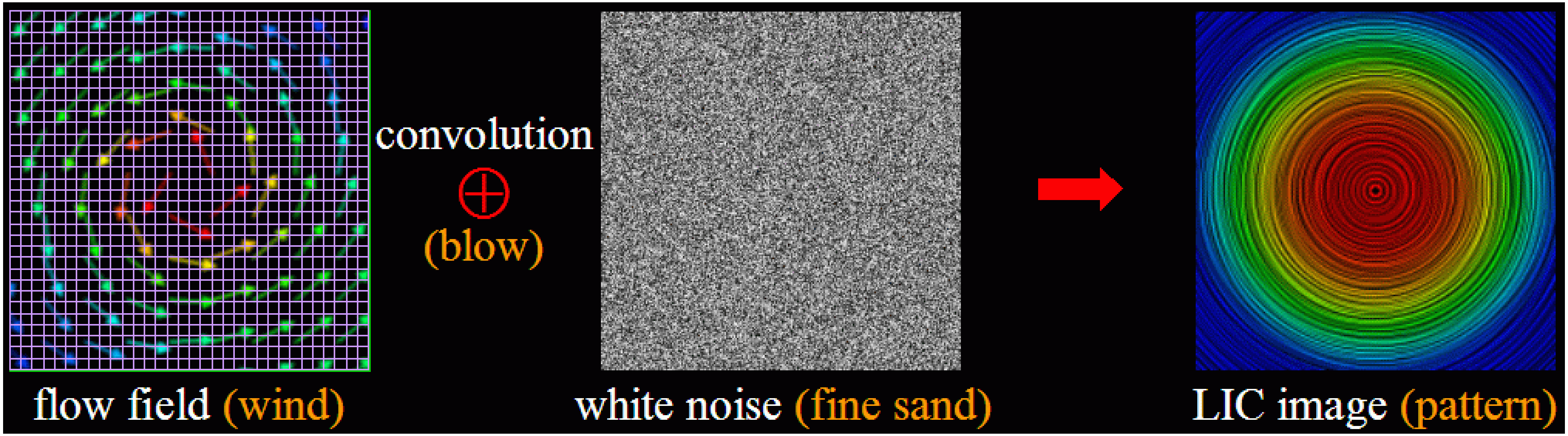
$$x_0 = x(t)$$

$$x_1 = x(t) + \tfrac{1}{2}v(x_0)\Delta t$$

$$x_2 = x(t) + \tfrac{1}{2}v(x_1)\Delta t$$

$$x_3 = x(t) + v(x_2)\Delta t$$

Line Integral Convolution (LIC)



- Image courtesy
  http://www.zhanpingliu.org/research/flowvis/lic/lic.htm

Line Integral Convolution (LIC)

- Given
  - Vector field
  - Texture image
- Output: colored field correlated in the flow direction
- Texture image is in general white noise.

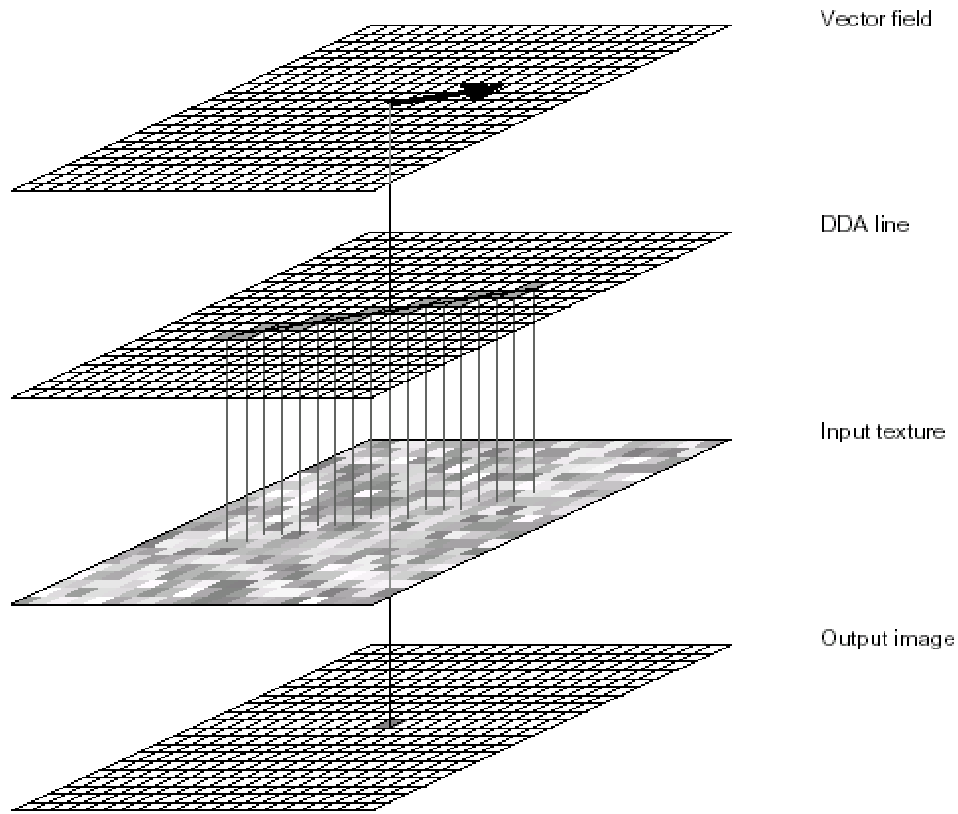
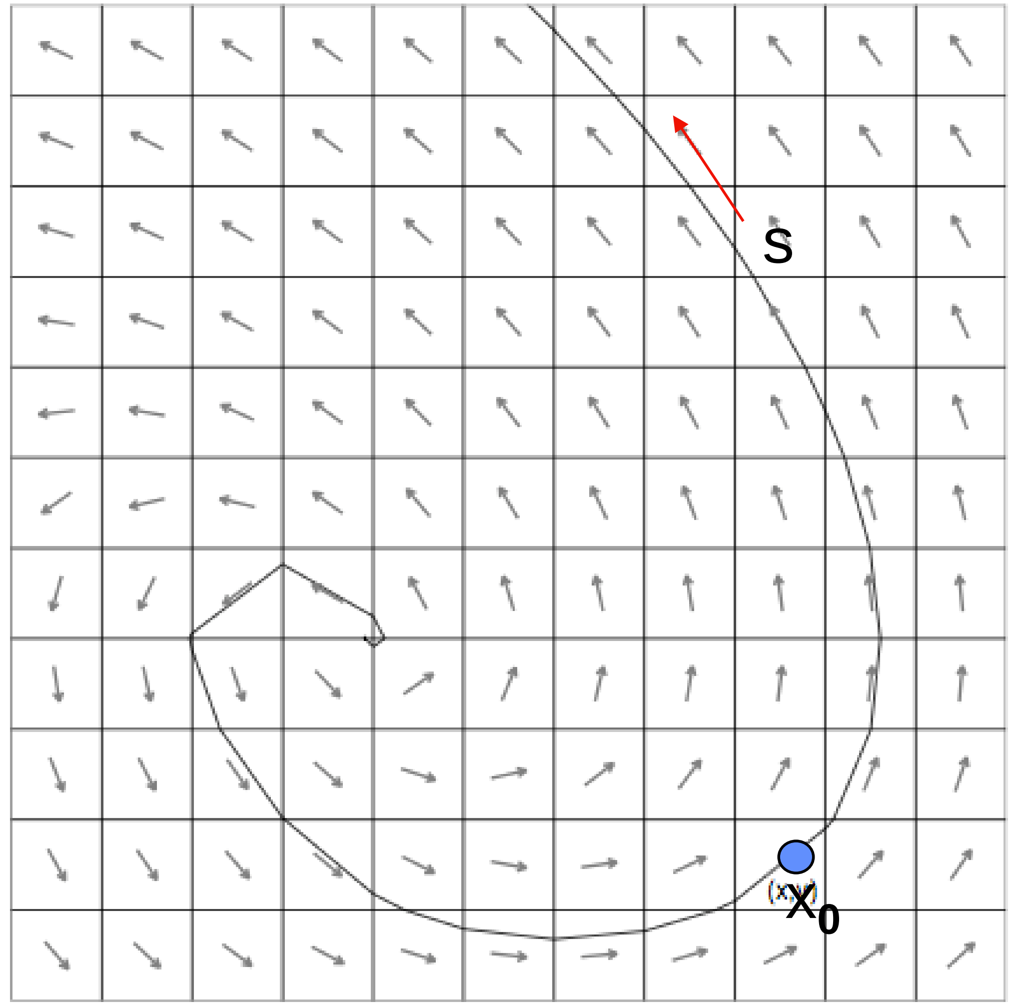


Image Courtesy Brian Cabral

LIC

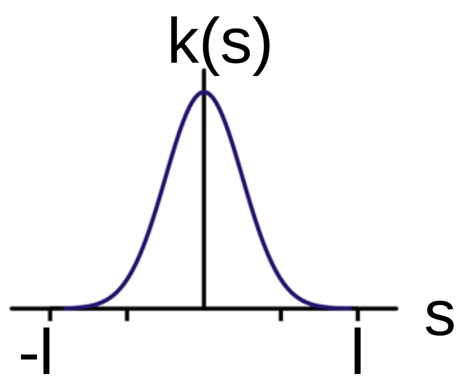
- Assume input texture (intensity), vector field and output images are all the same resolution.
- For each output pixel/voxel, generate a streamline both forwards and backwards of a fixed length.
- Integrate the intensity that the streamline passes through.



LIC

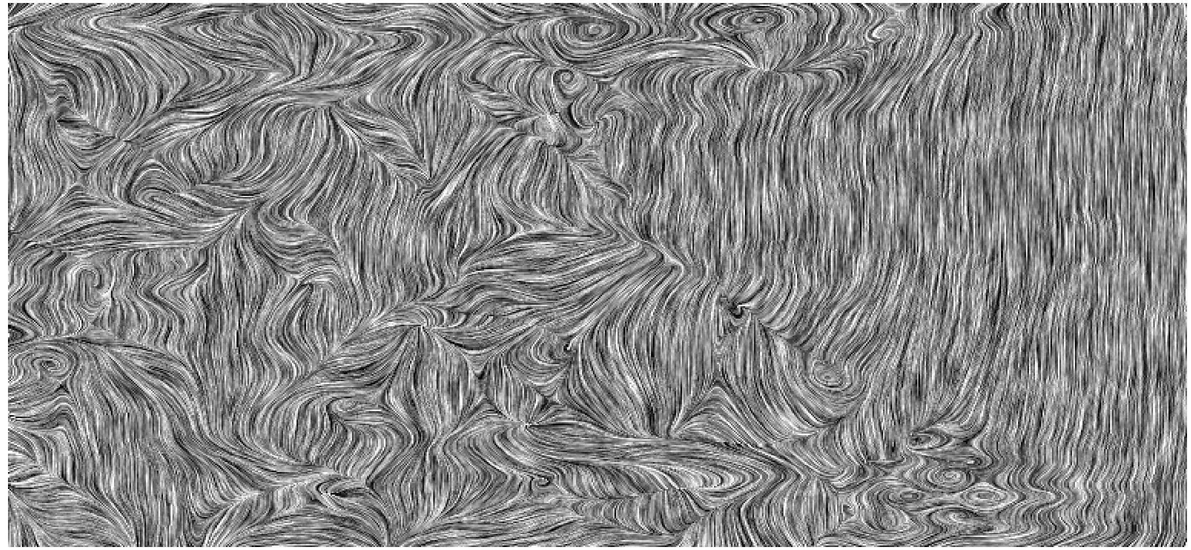
$$I(x_0) = \int_{s_0-l}^{s_0+l} k(s-s_0)T(L(s))ds$$

- $I(x_0)$: Intensity of pixel x_0
- $k(s)$: filter kernel of length $2l$ – weighting function
- $T(x)$: noise texture at location x
- $L(s)$: stream line parameterized by arc length
- $x_0=L(s_0)$

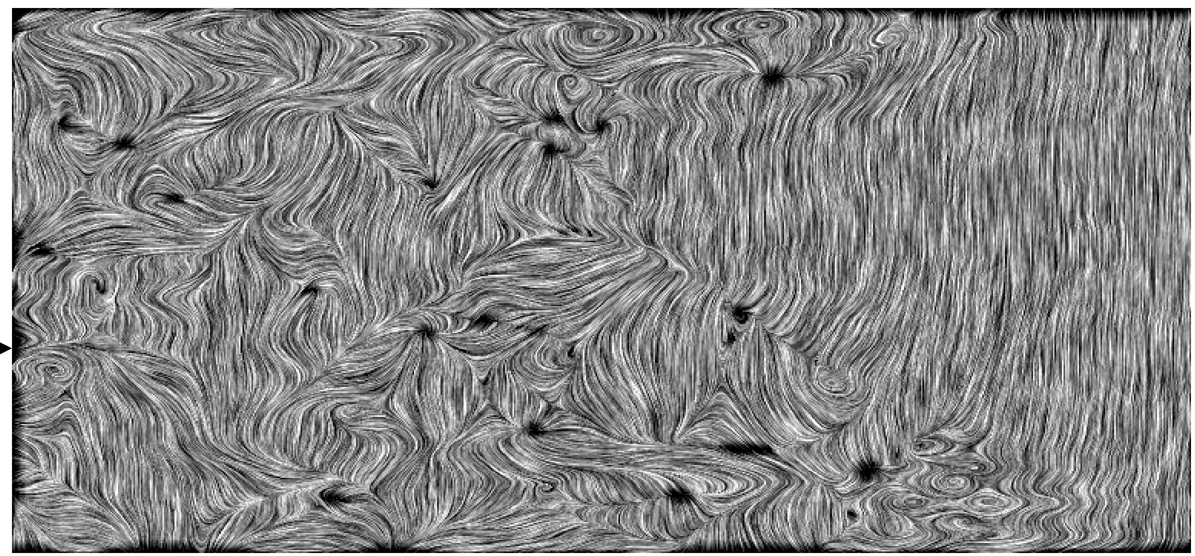


LIC - Normalization

- We need to normalize by the sum of the filter weights

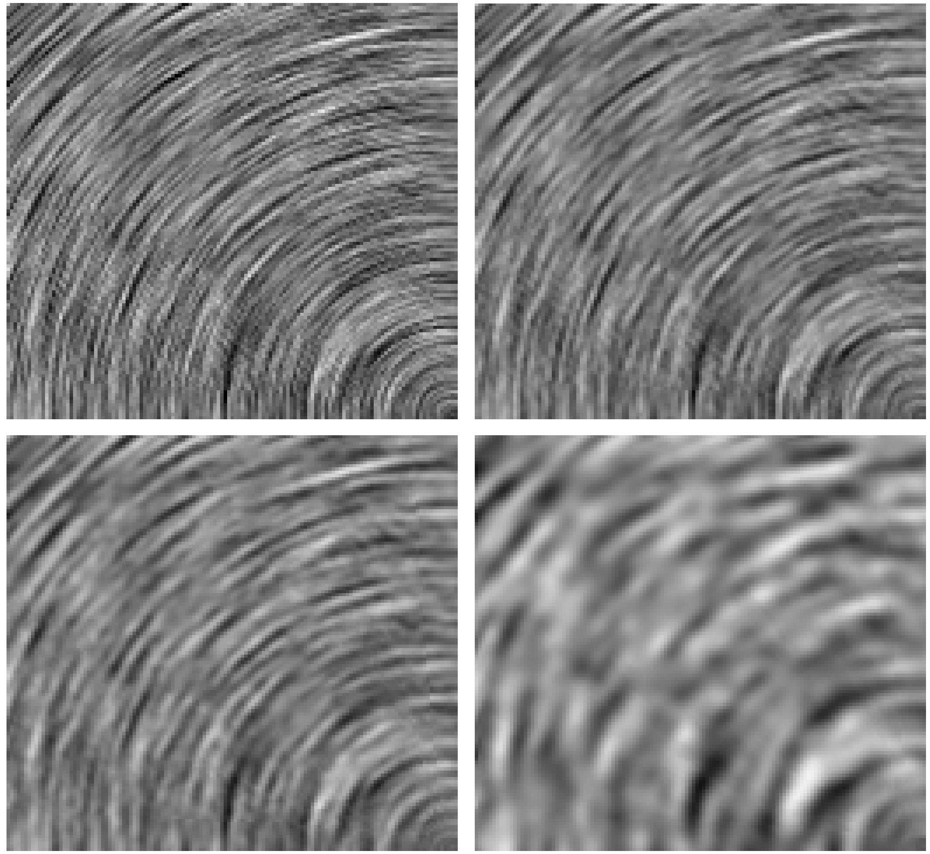


No normalization

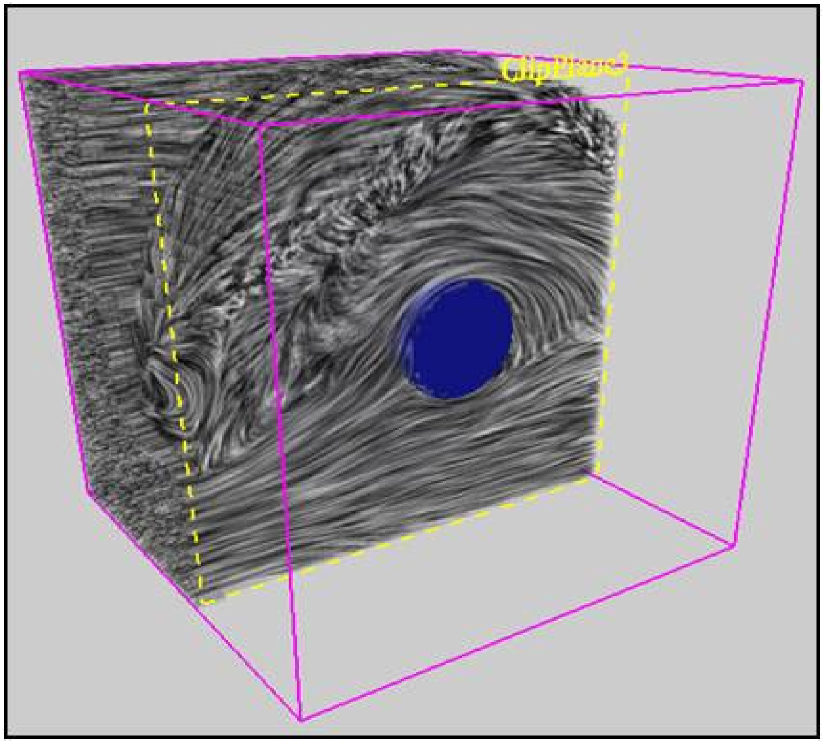


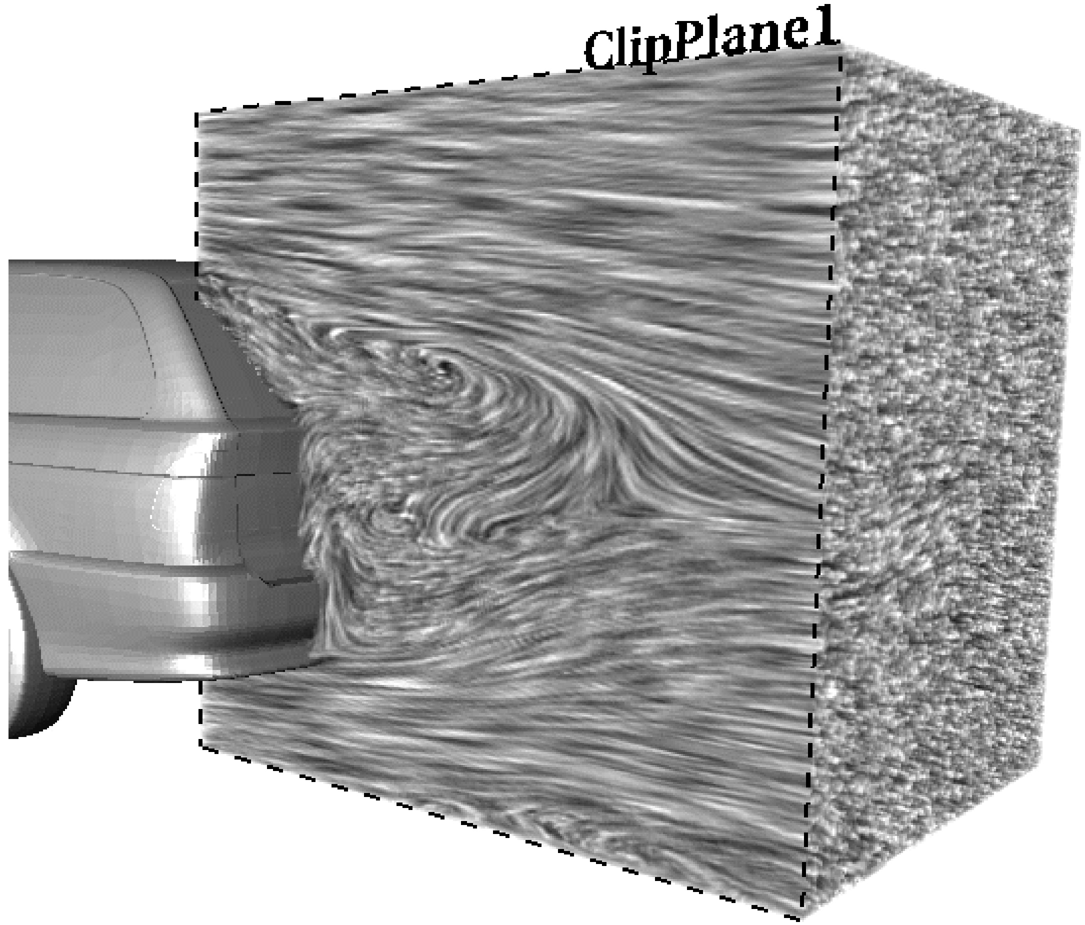
LIC

- Aliasing can be a problem
- Use low-pass noise!
- DEMO!



3D LIC





END