Visualization
– Information Visualization II

J.-Prof. Dr. habil. Kai Lawonn

Visualization of Trees and Hierarchies

Trees and Hierarchies

- What are trees?
  - Represent hierarchies
    - Items + structure (nodes + associations)
    - Ordering of items in which particular items are *parents* or *ancestors* of others
  - A mathematical model which applies to many topics
    - Represents the way nodes are arranged in many data structures
    - Represents the way many aspects of society are organized
  - A collection of nodes
    - Each may have extra information, such as label, data

Trees and Hierarchies – Examples

- Hierarchies
  - File systems, menus
  - Organization charts
  - Categorical classifications / taxonomy
  - Similarity and clustering
- Branching Processes
  - Genealogy and lineages, family trees
  - Phylogenetic trees
- Decision Processes
  - Indices or search trees
  - Decision trees

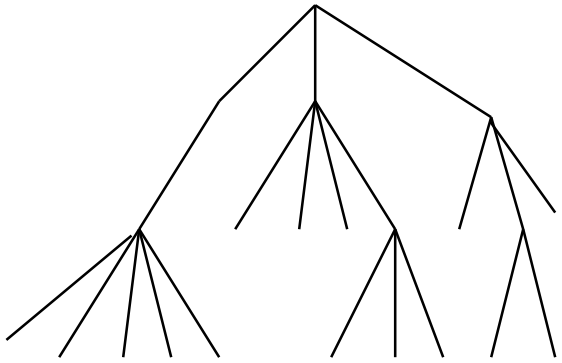
Trees and Hierarchies – Objectives

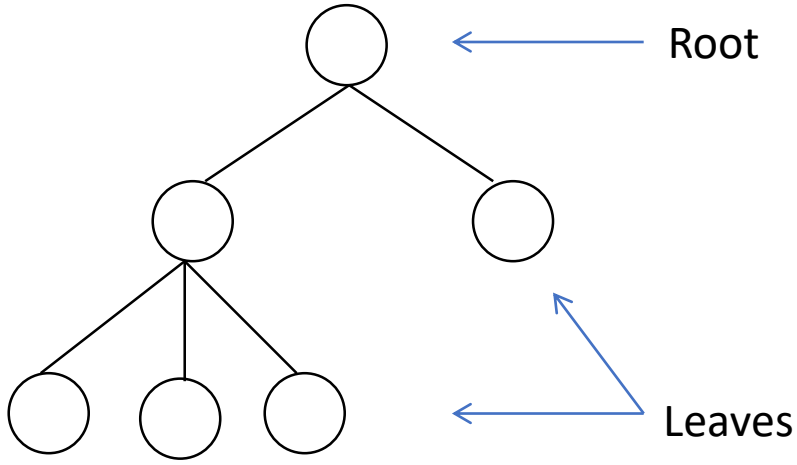
- Representing hierarchical data
  - Content information
  - Structural information

- Visualization objectives
  - Comprehension
  - Efficient usage of space
  - Esthetics
  - Exploration and interactivity

Trees and Hierarchies

- Hierarchies often represented as trees
- Root placed at the top, leaves at the bottom (top-down drawing)



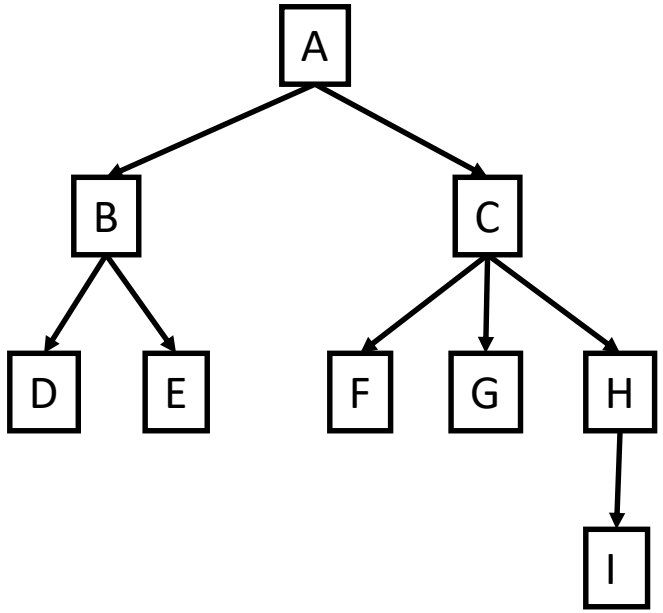


Terminology

- V = set of nodes (or vertices)
  - |V| = # of nodes or cardinality of V
- E = set of edges (or arcs), an edge is defined by two nodes
  - |E| = # of edges or cardinality of E
- There are several characterizations of a tree T=(V,E), such as
  - A tree is a connected graph with n nodes and n-1 edges
  - A tree is a connected graph that is no longer connected after removal of any edge.
  - A tree is connected and cycle free

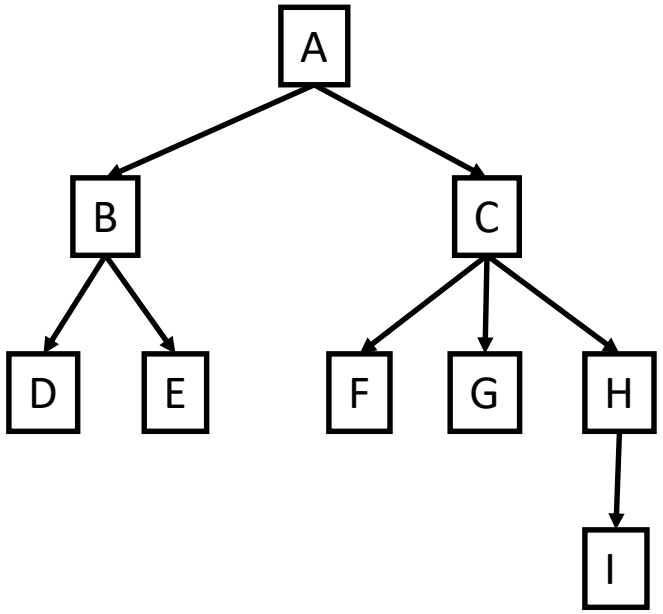
Terminology

- Number of nodes = number of edges+1
- A is the *root*
- B is the *parent* of D and E
- D and E are the *children* of B
- C is the sibling of B
  - Nodes with the same parent
- A, B, C, H are *internal nodes*
- D, E, F, G, I are *external nodes* or *leaves*



Terminology

- Depth of a node
  - Number of edges from the root to the node
  - Depth of root is always 0
  - depth(D)=2
  - If X is parent of Y
    - depth(Y) = depth(X) + 1
- Height of a node
  - Largest number of edges from the node to a descendant (deepest leaf)
  - Height of leaf is always 0
  - height(C)=2 (some paths from C are shorter than 2 edges)
- Height of tree is defined as height of root



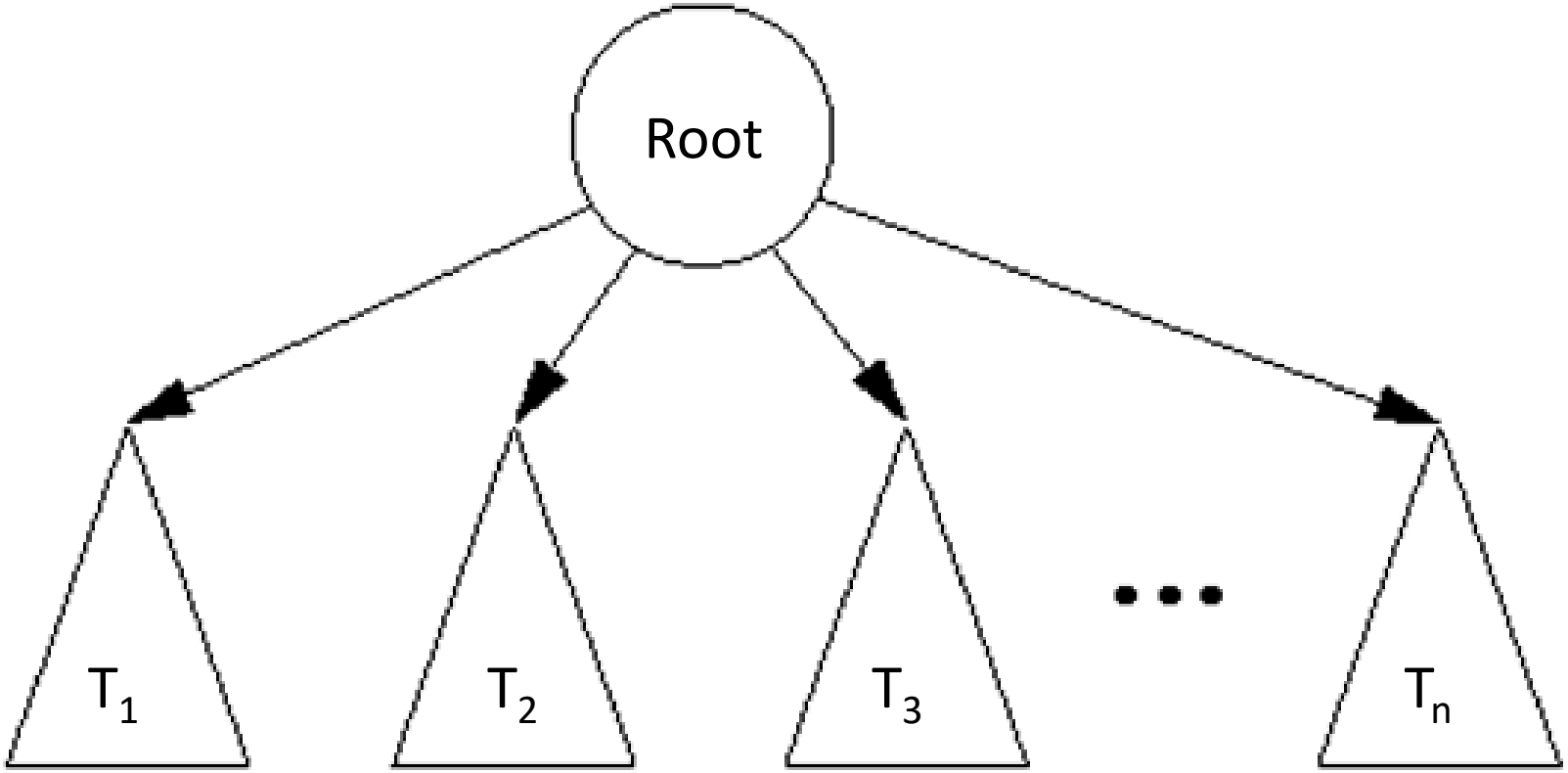
Terminology

- Rooted tree
  - Root is a distinguished node in the tree
  - Usually treated as a directed graph: all edges oriented away from the root
  - Direct edge u->v
    - u is the *parent* of v
    - v is the *child* of u
  - Leaf is a node with no child
- Ordered tree
  - Rooted tree with a (left-right) ordering for the children of every node

Terminology

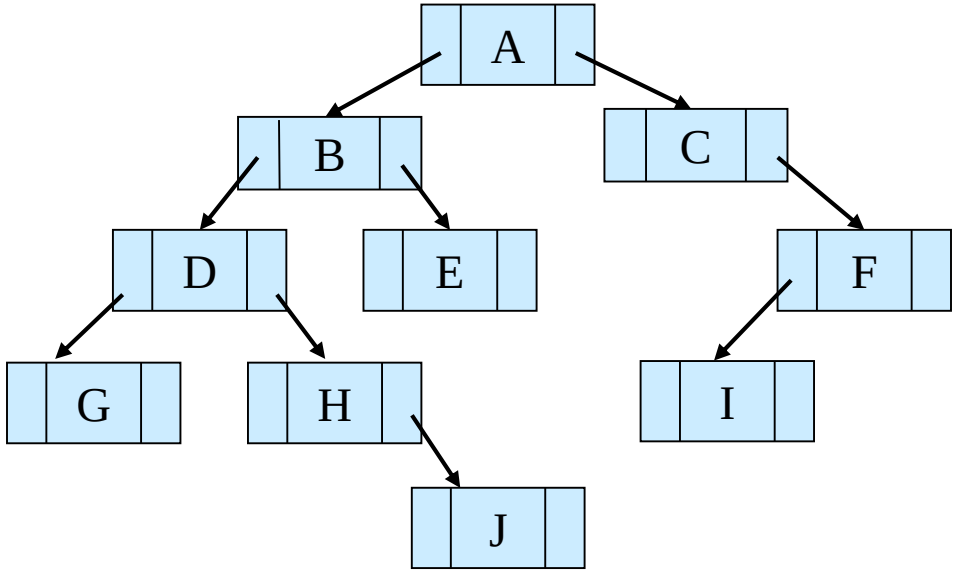
- Binary tree
  - Rooted tree with every node has at most two children
  - Left and right child or one child (either left or right)
- Subtree rooted at node v
  - The subtree induced by all nodes that have v as their “ancestor”
  - For example, in a binary tree: left and right subtree

Example: Subtrees of a Tree



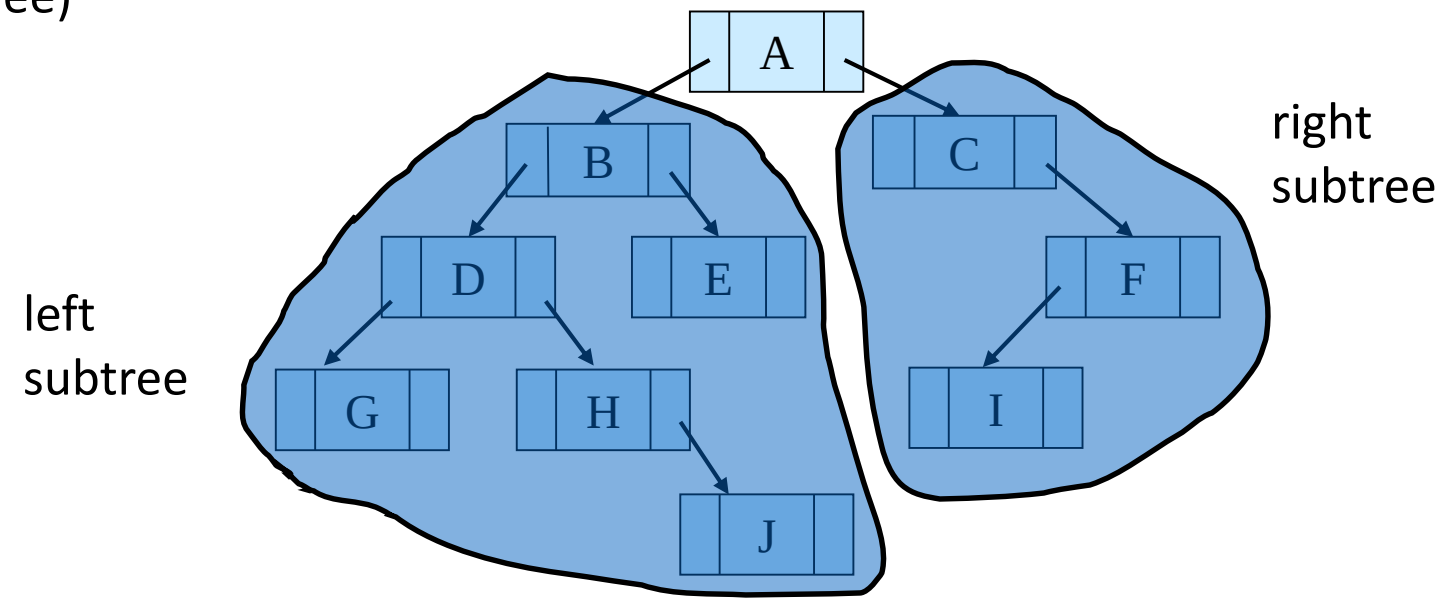
Binary Trees

- Recursive definition of binary tree
  - A binary tree is either
    - Empty or
    - An external node (leaf) or
    - An internal node (the root) and one or two binary subtrees (left subtree and/or right subtree)



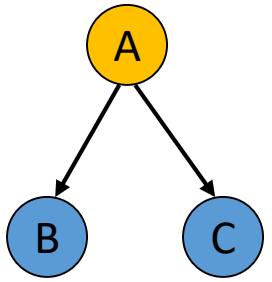
Binary Trees

- Recursive definition of binary tree
  - A binary tree is either
    - Empty or
    - An external node (leaf) or
    - An internal node (the root) and one or two binary subtrees (left subtree and/or right subtree)

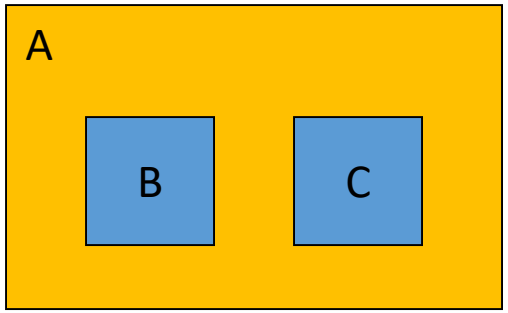


Two Visualization Approaches

- Connection
- Nodes and edges



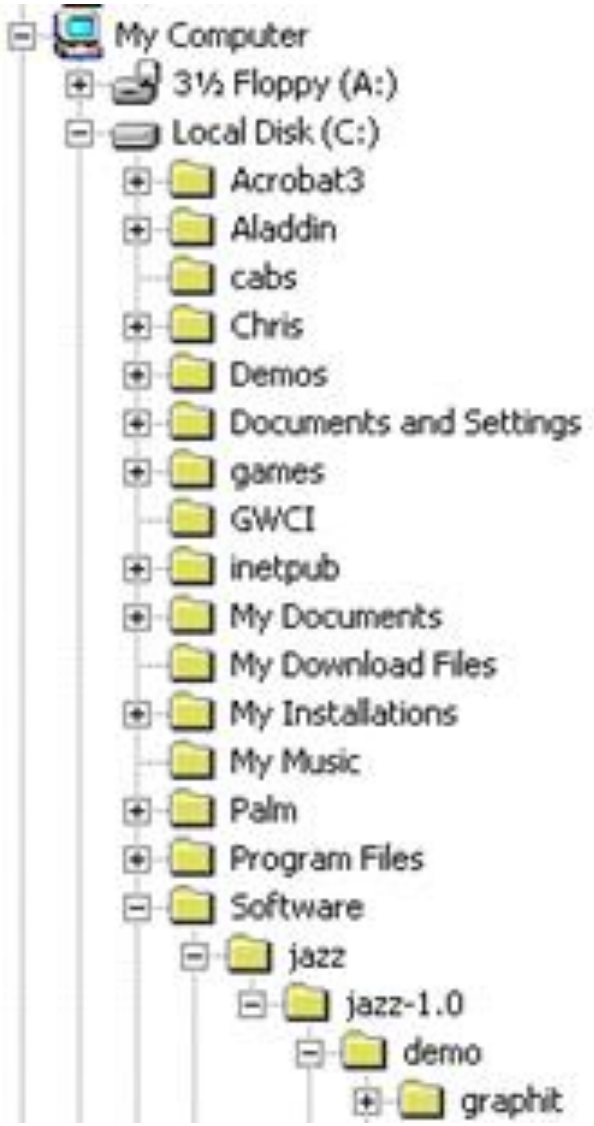
- Containment
- Nodes in node



Tree Visualization

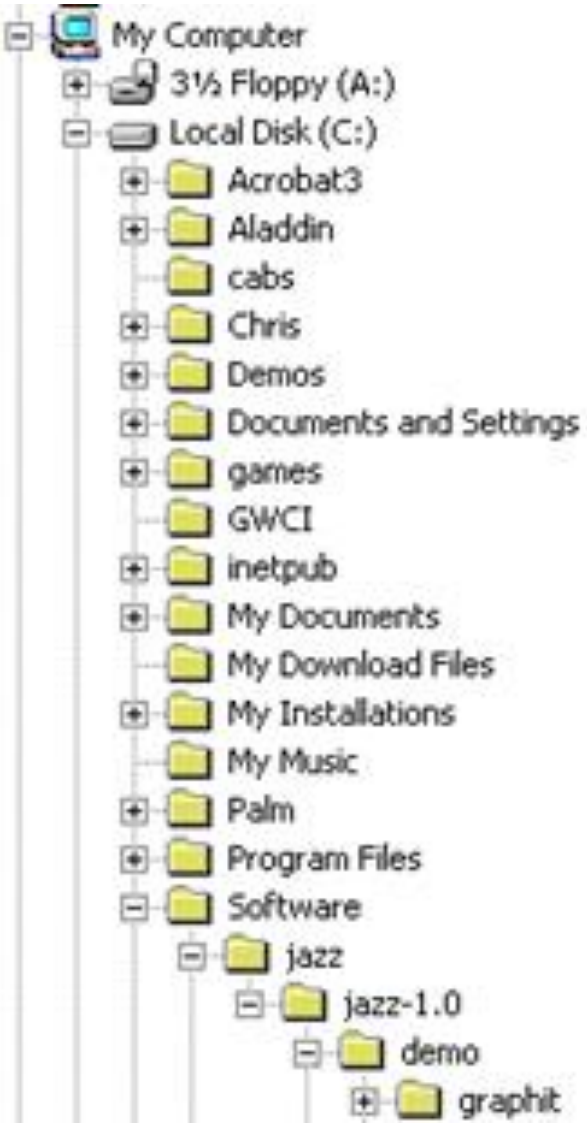
- Example: TreeView

- Why is tree visualization hard?
  - Show structure and items
  - Structure harder, consumes more space
  - Data size grows very quickly (exponential with height of tree)

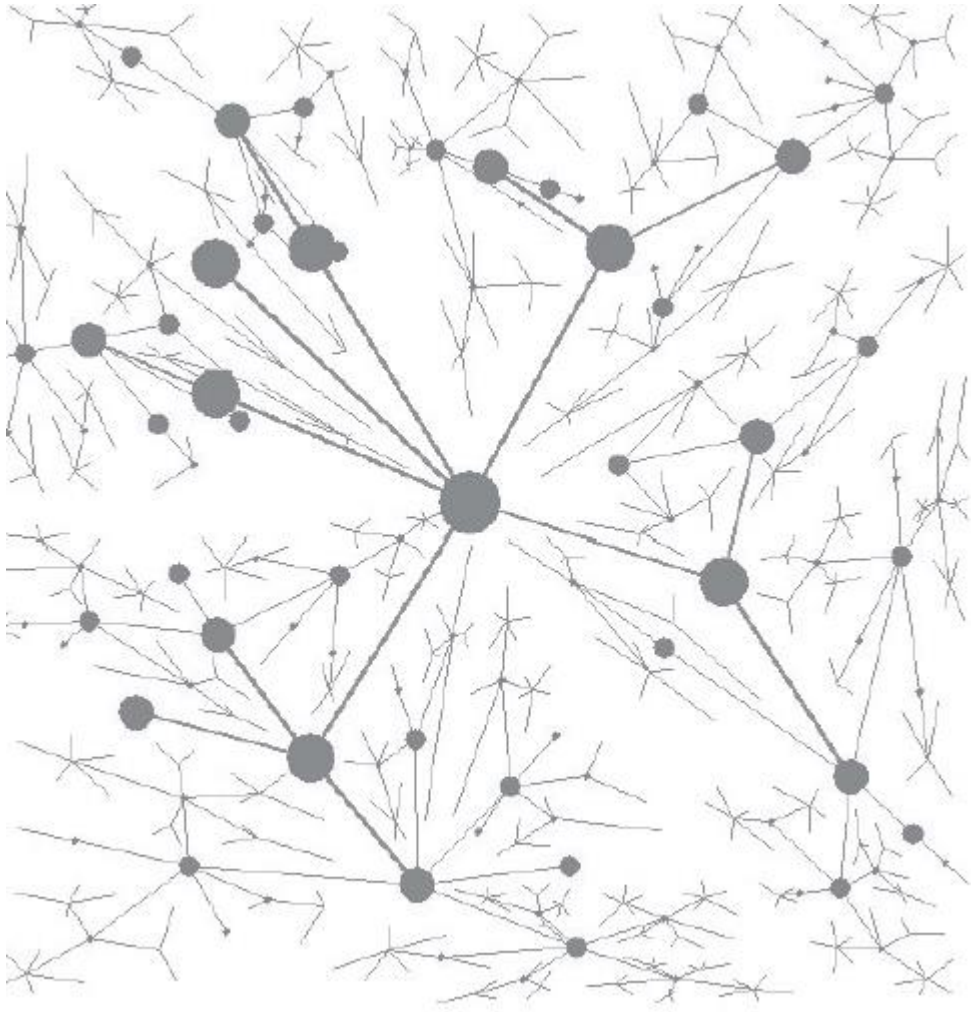


Example: TreeView

- Good for directed search tasks
- Filtering of subtrees
- Not good for investigating the tree structure
  - Structure partly hidden
  - Only a small number of items visible



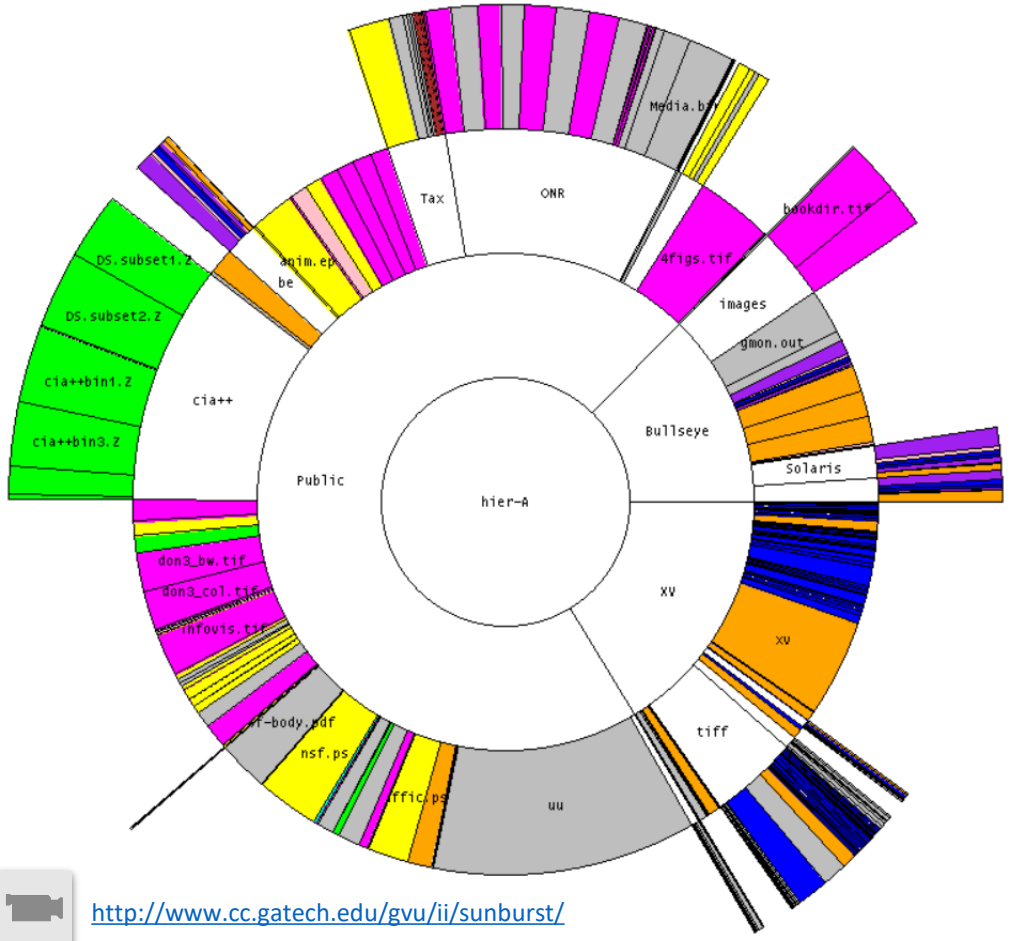
Example: Space Optimized Tree



[Nguyen & Huang 2002]

Example: Sunburst

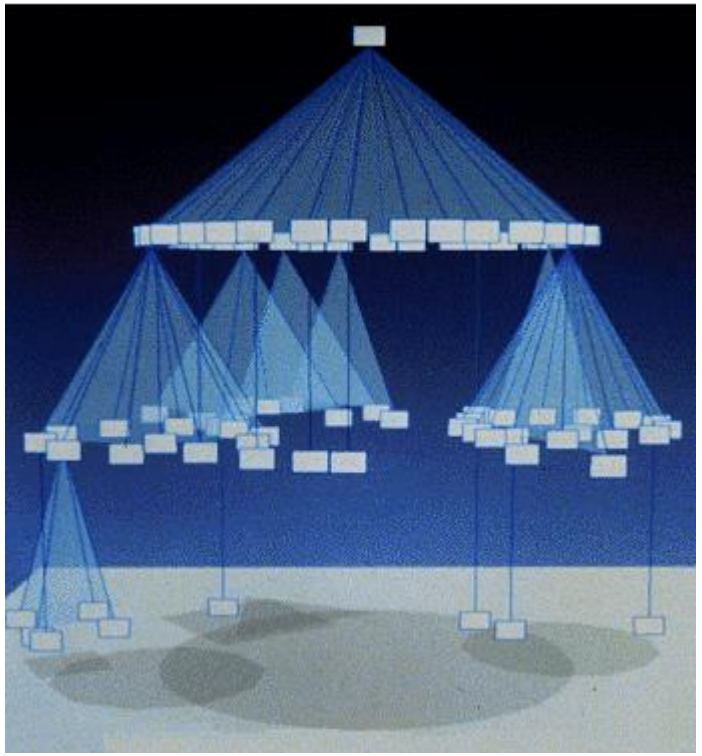
- Reduced visual representation, edges not explicitly shown
- Try it yourself here



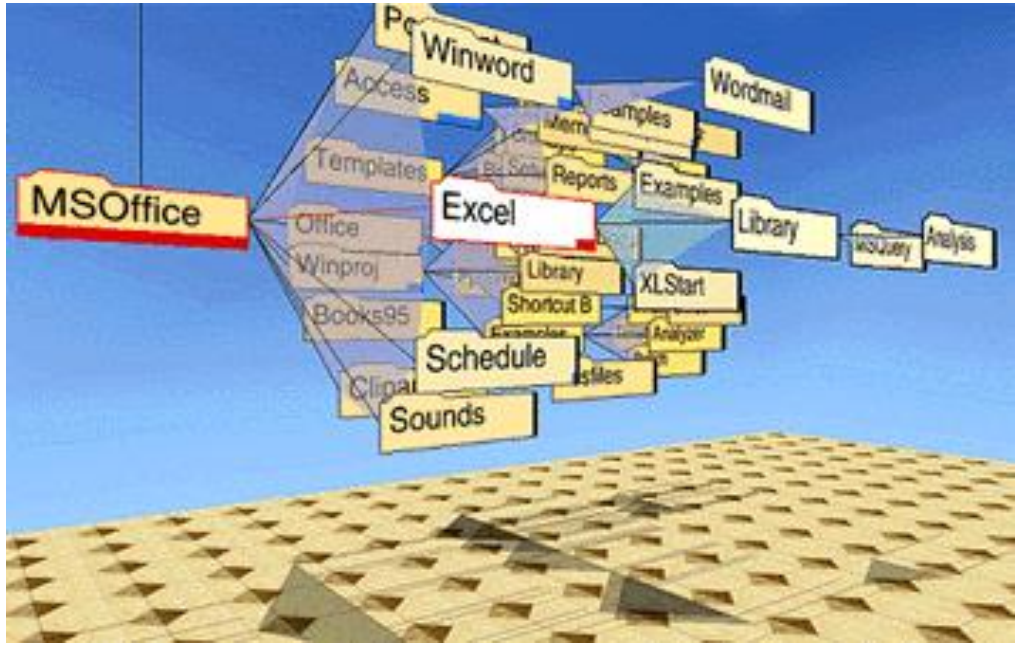
http://www.cc.gatech.edu/gvu/ii/sunburst/

[Stasko & Zhang 2000]

Example: Cone Trees

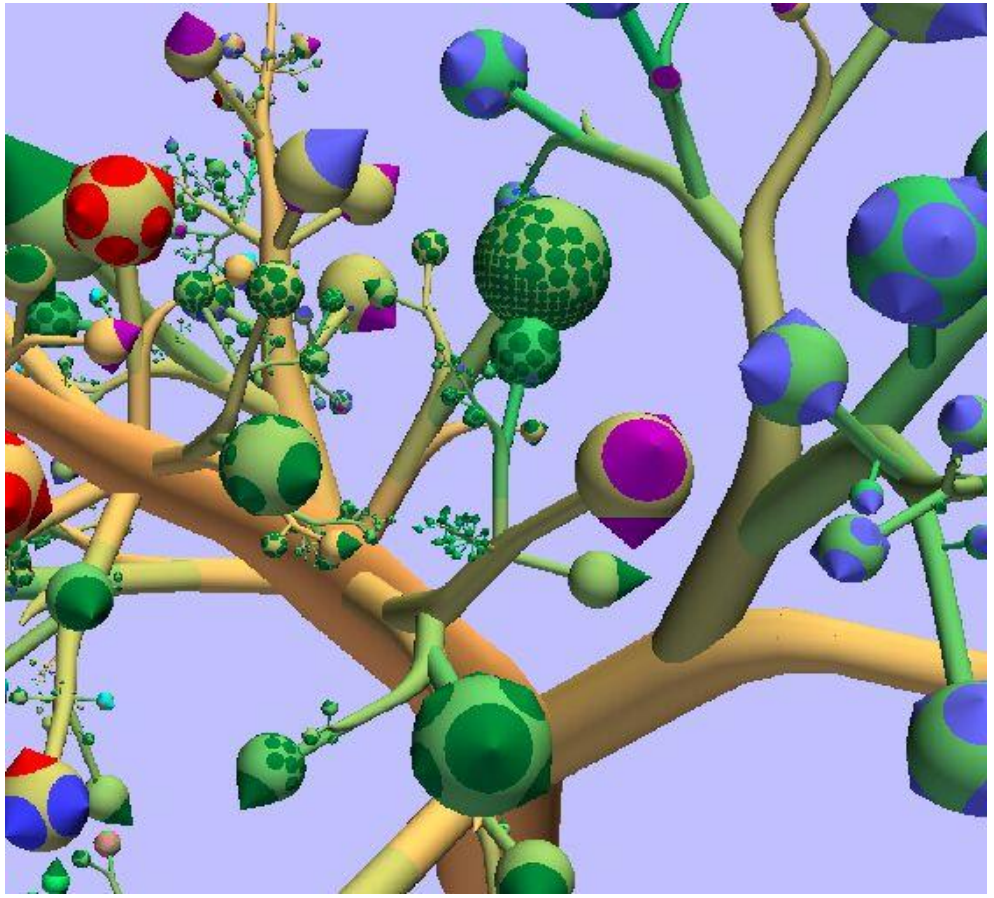


[Robertson, Mackinlay & Card 1991]



Example: Botanical Visualization

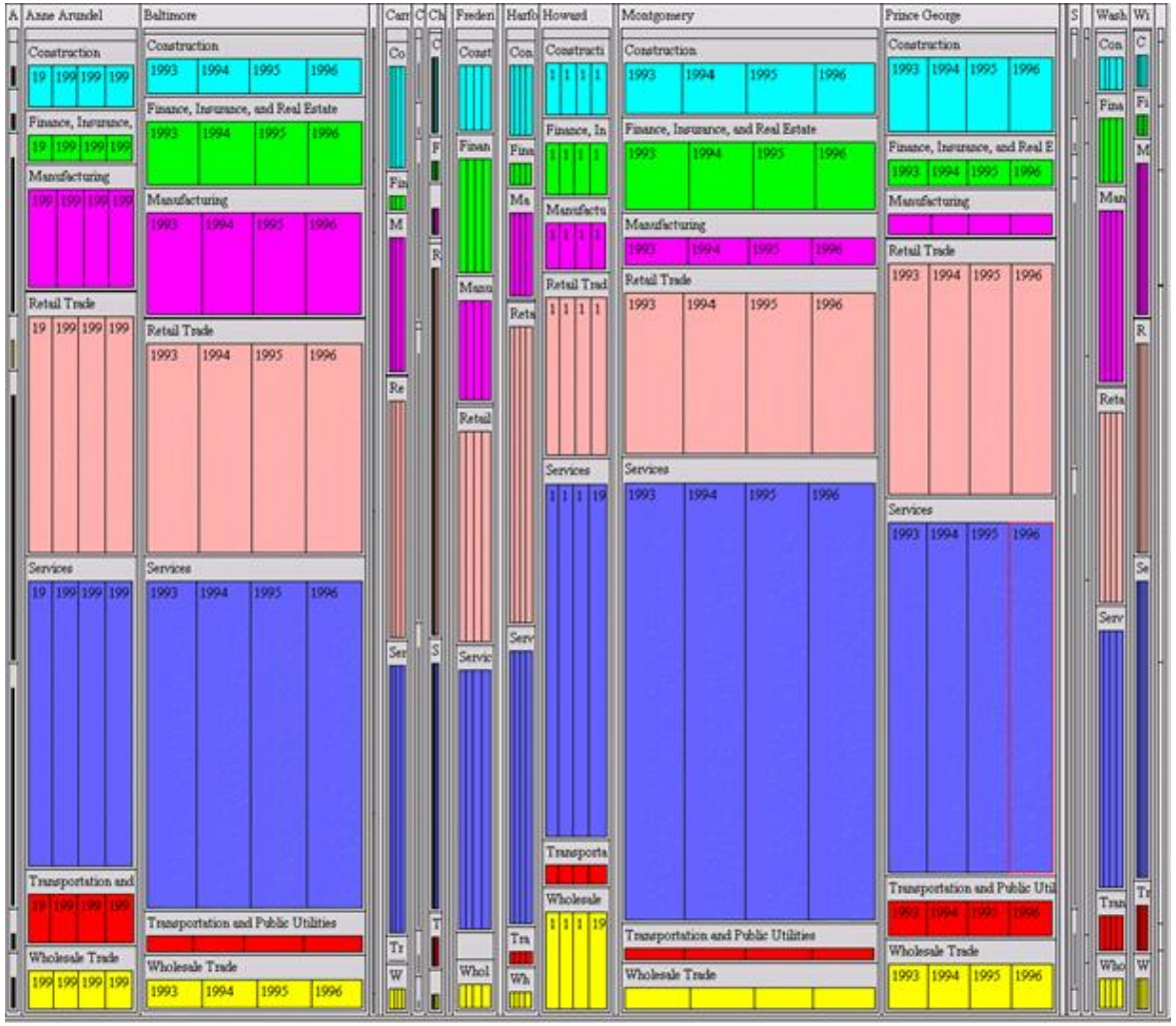
- The root is the tree stem
- Non-leave nodes are branches
- Leave nodes are “bulbs” at the end of branches



[Kleiberg, Wetering & van Wijk 2001]

Example: Treemap

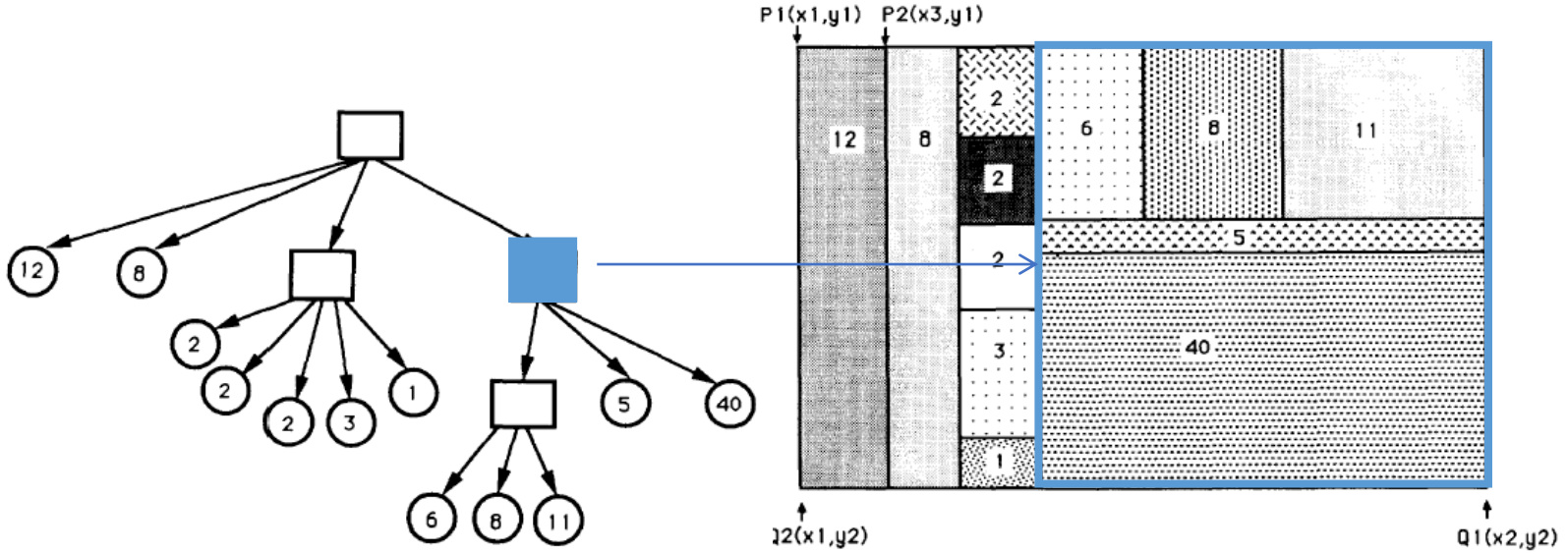
- Recursive subdivision of the area into rectangles
- Each rectangle corresponds to node
- Area and color correspond to node values
- Try it yourself here and here



[Johnson & Shneiderman 1991]

Treemap Algorithm

- Calculate node sizes
  - Recurse to children
  - Node size = sum children sizes
- Draw Treemap
  - Draw node rectangle in space using direction
  - Alternate direction (even or odd level)
  - For each child
    - Calculate child space as % of node space using size
    - Draw Treemap (child, child space, direction)



[based on Shneiderman 1992]

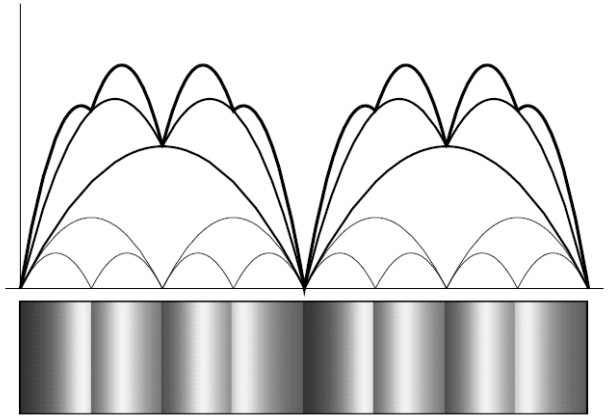
Treemap – Properties

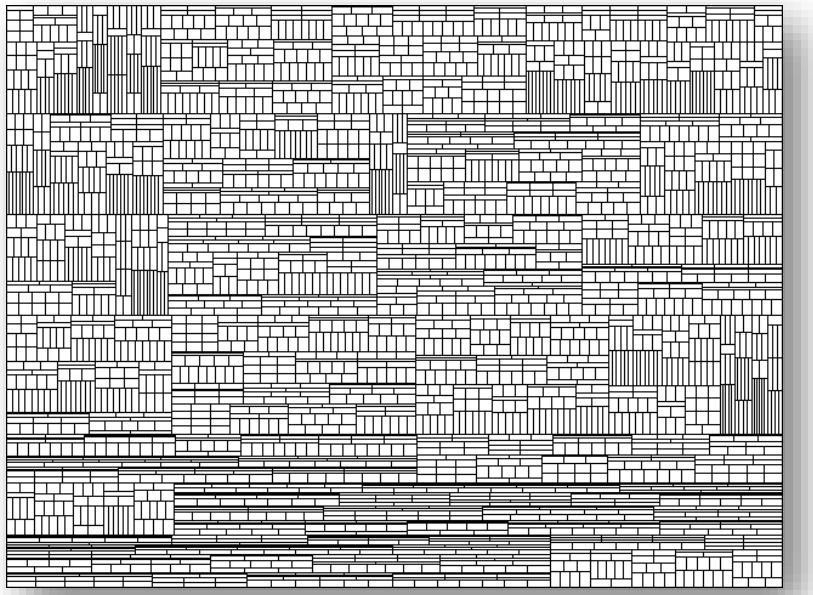
- Strengths
  - Uses all space to display tree
  - Shows nesting of hierarchical levels
  - Can represent node attributes (of leave nodes)
  - Scalable to large data sets

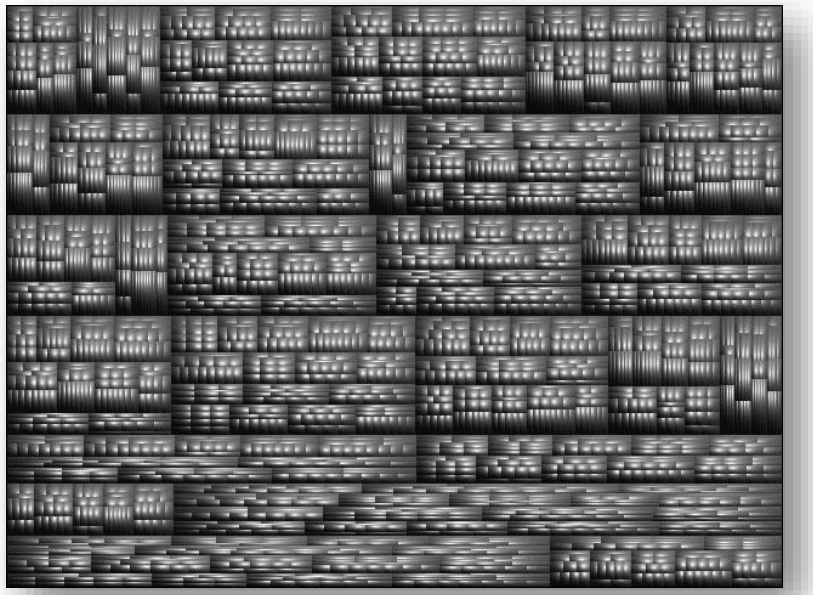
- Weaknesses
  - Size comparison is difficult
  - Difficult to see boundaries
  - Difficult to show labels of nodes
  - Shows only the information of the leaves

Treemap Variations

- Cushion treemap [van Wijk & van de Wetering 1999]
  - Using shading to help identify the levels in a treemap

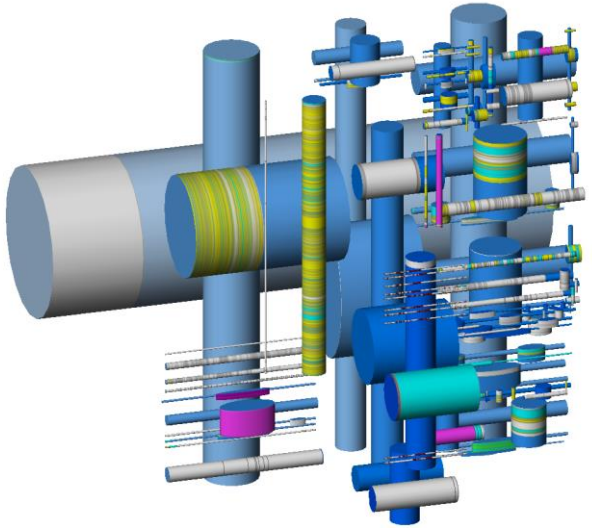




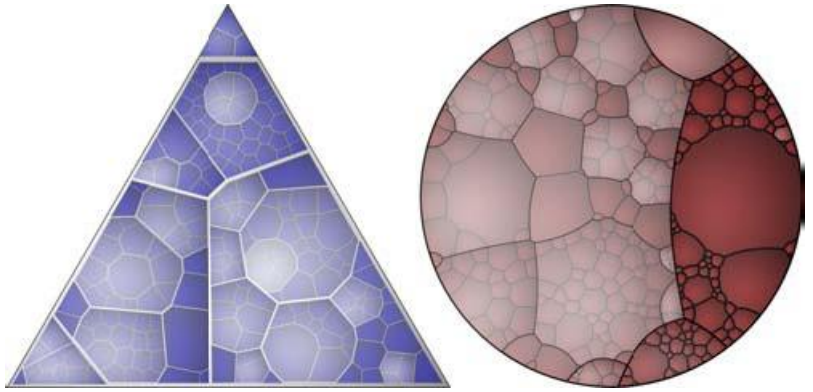


Treemap Variations

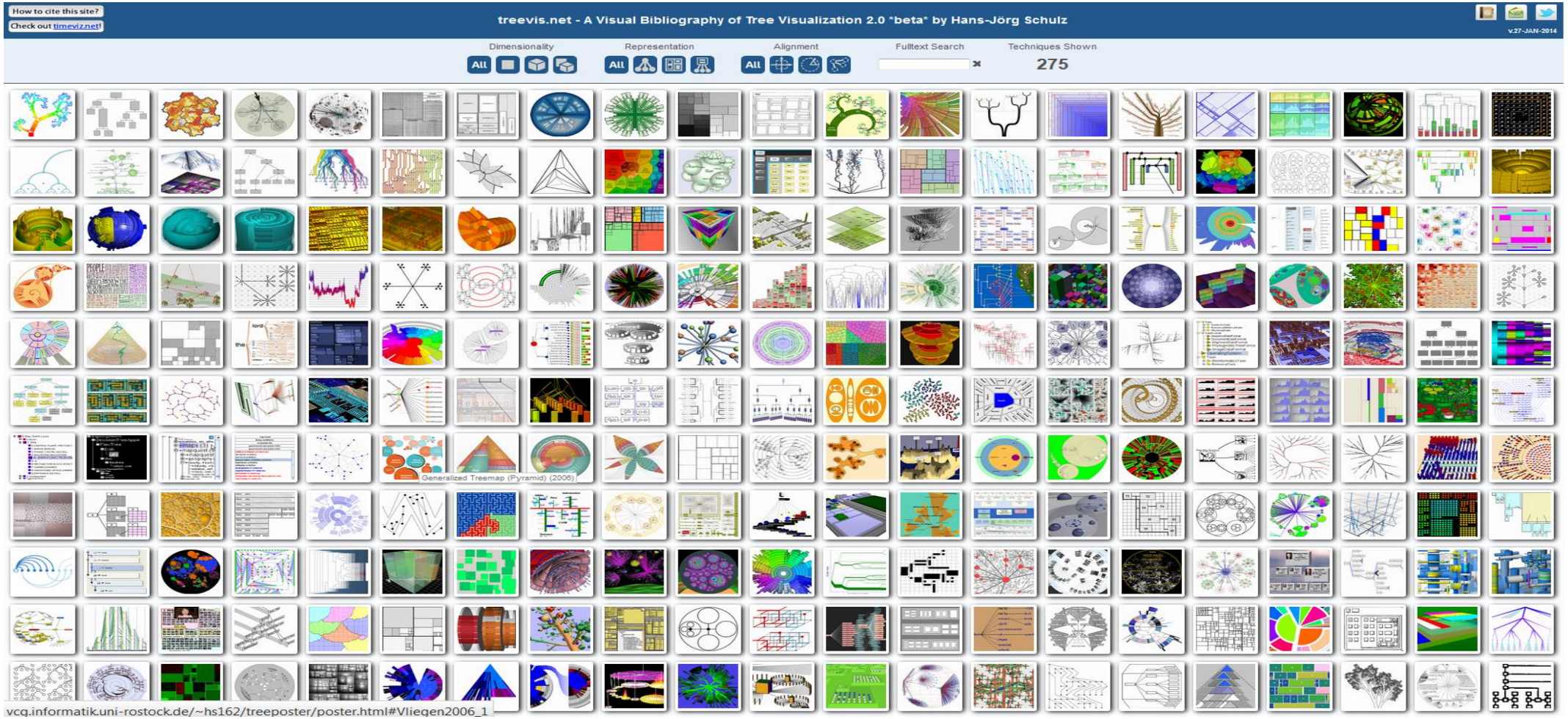
- Beamtree [van Ham & van Wijk 2002]
  - A variation of treemap in 3D
  - Using overlap instead of nesting to show the hierarchy



- Voronoi treemap [Balzer & Deussen 2005]
  - Similar idea but uses voronoi diagram as partition
  - The space does not have to be rectangular



Many More Approaches



See http://vcg.informatik.uni-rostock.de/~hs162/treeposter/poster.html#

Recursion and Trees

- Many operations and calculations are most easily expressed recursively
- Make sure there is a base case where you do not recurse
  - Avoid infinite calling
  - Usually base case is leaf or null
- Calculate answer for each call based on answers to recursive calls

Traversal

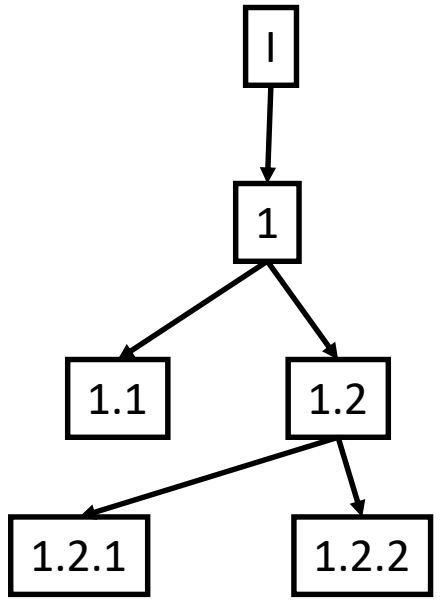
- Ways to go through nodes in order
  - Based on the structure of the tree
- There are many algorithms which need to do something with each node of a tree
  - E.g., count the elements
  - E.g., sum the elements' prices
  - E.g., find the highest mark of any element
- Sometimes, we care about the order of processing nodes
  - Calculate some function based on values for descendants or ancestors

Traversal

- A traversal is a particular order of processing (called “visiting”) all nodes
  - Three examples are
    - Preorder
    - Postorder
    - Inorder

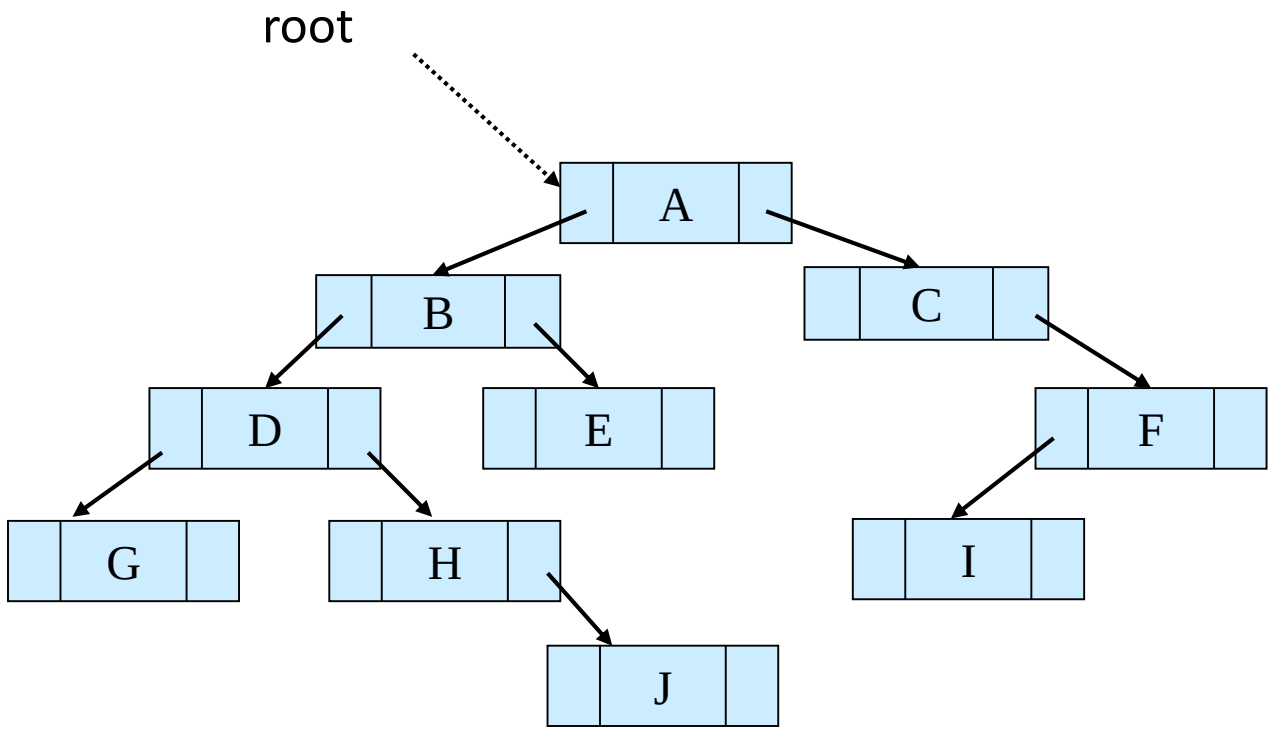
Preorder Traversal

- Each node is visited, immediately followed by its descendants
- To preorder traverse the subtree rooted at x
  - Visit x
  - For each child y of x
    - Preorder traverse the subtree rooted at y
- This matches the way a book lists table of contents



Preorder Traversal – Binary Trees

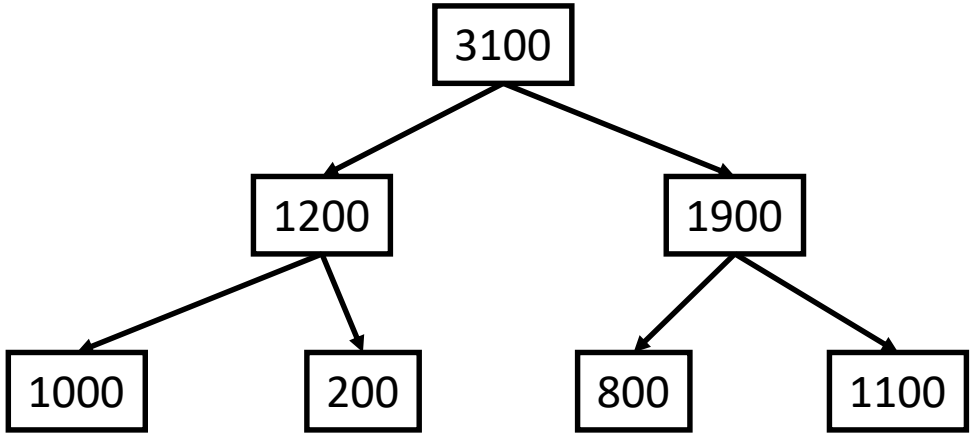
- Process the node
- Left child
- Right child



Postorder Traversal

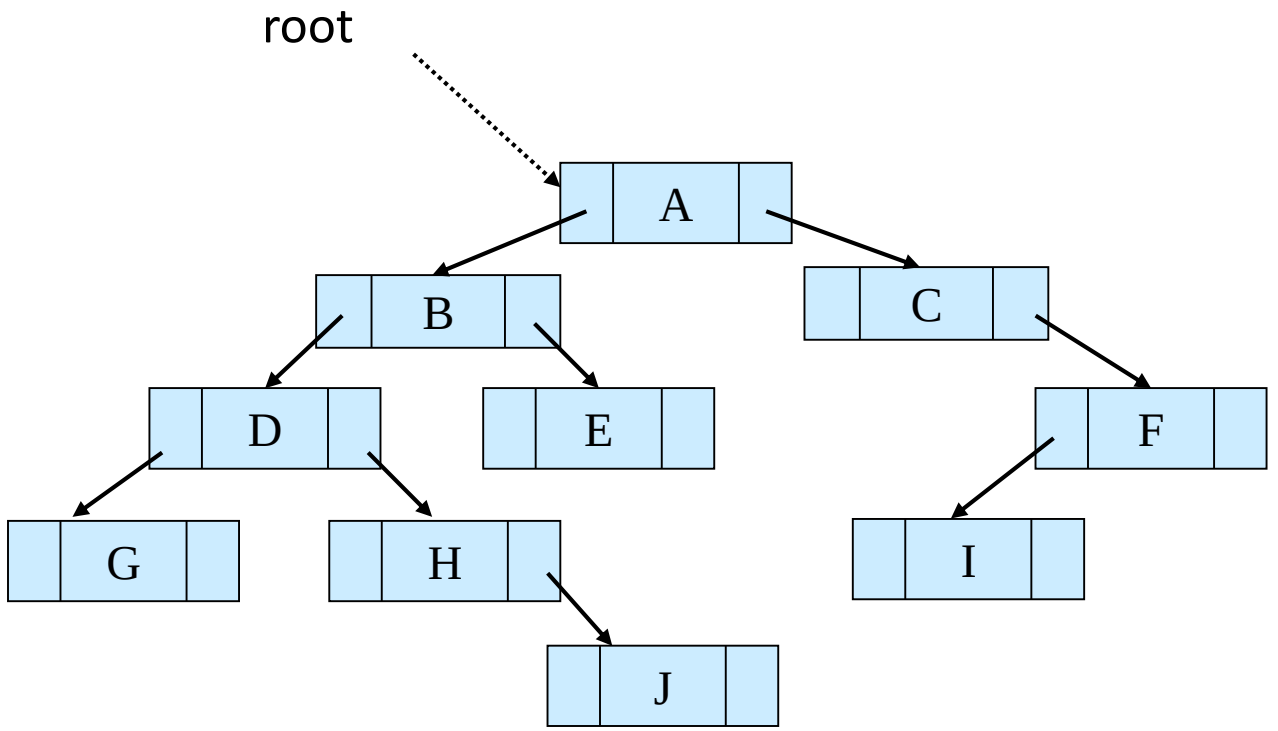
- Each node is visited immediately after all its descendants
- To postorder traverse the subtree rooted at x
  - For each child y of x
    - Postorder traverse the subtree rooted at y
  - Visit x
- This matches the way an account calculates subtotals

Graphic:	1,000
Memory:	200
Hardware total:	1,200
Licences:	800
Adaptation:	1,100
Software total:	1,900
Total cost:	3,100



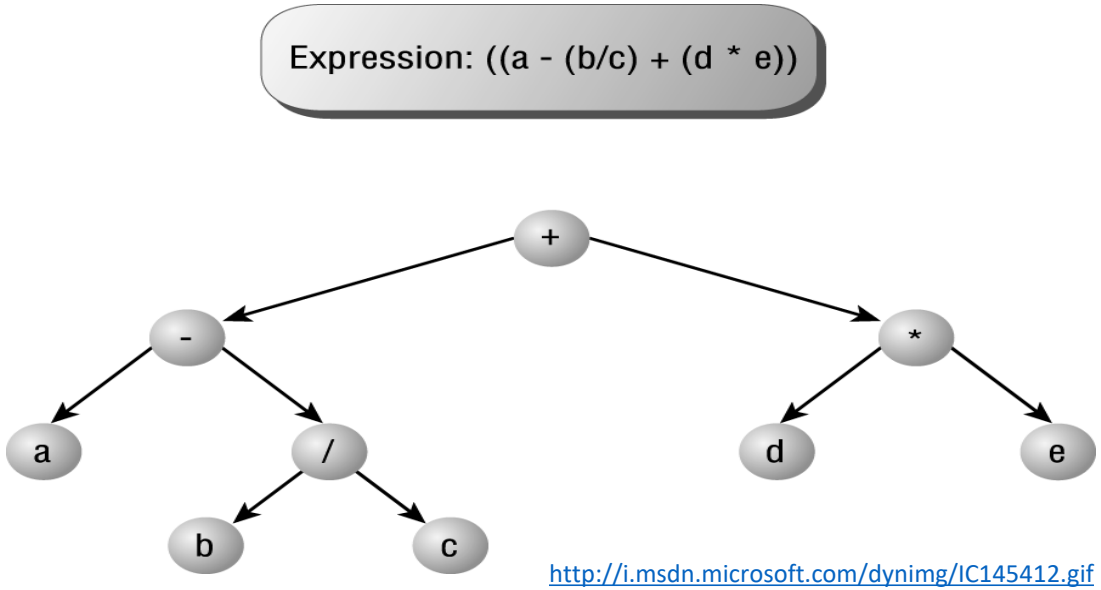
Postorder Traversal – Binary Trees

- Left child
- Right child
- Process the node



Inorder Traversal

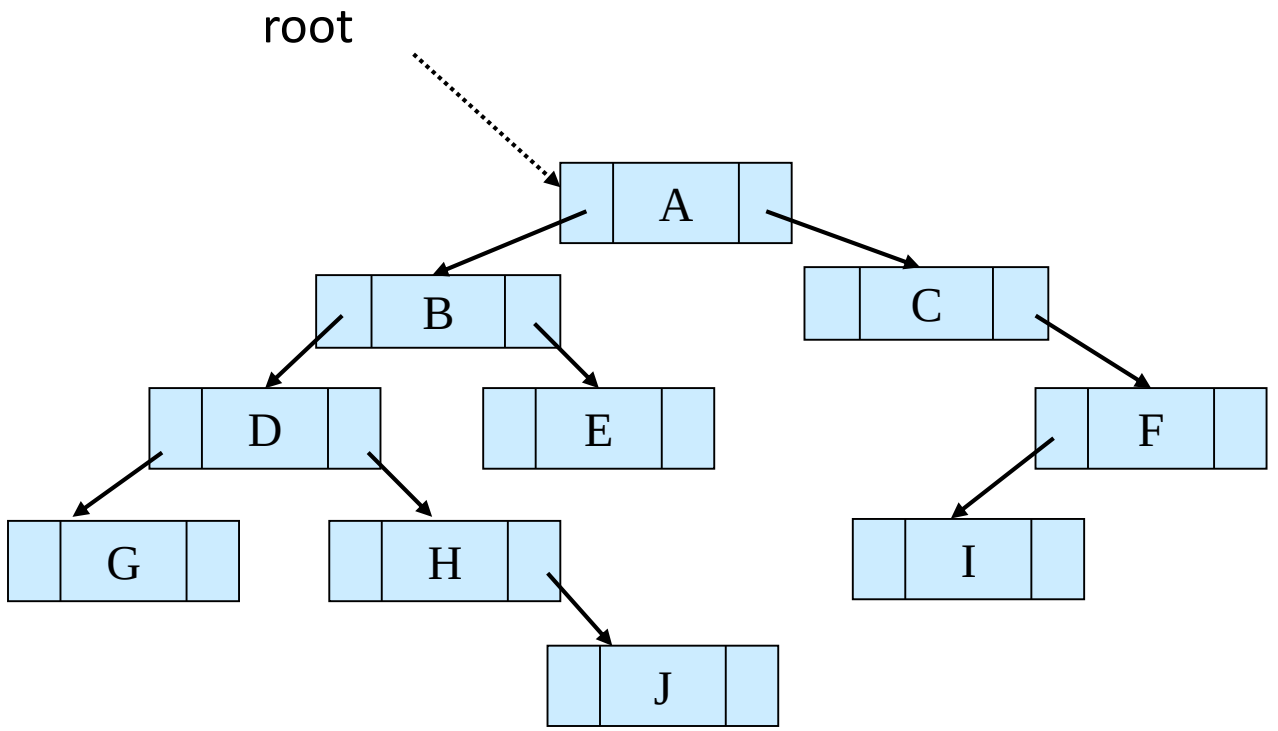
- Only for binary trees
- Each node is visited immediately after its left subtree, and immediately before its right subtree
- This corresponds to the way mathematical expressions are written



http://i.msdn.microsoft.com/dynimg/IC145412.gif

Inorder Traversal – Binary Trees

- Left child
- Process the node
- Right child



Literature

- http://vcg.informatik.uni-rostock.de/~hs162/treeposter/poster.html
- [Nguyen & Huang] Q.V. Nguyen and M.L. Huang. **A space-optimized tree visualization**. Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on, pages 85–92, 2002.
- [Stasko & Zhang] Stasko J., Zhang E. **Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations**. In Proceedings of the IEEE Symposium on Info Vizualization (Salt Lake City, USA, 2000), p. 57.
- [van Ham & van Wijk] F. van Ham, J. van Wijk. **Beamtrees: compact visualization of large hierarchies**. In Proc. of the IEEE Symposium on Information Visualization (Boston, USA, 2002), IEEE Press, pp. 93–100.
- [Robertson, Mackinlay & Card] Robertson, G.G., Mackinlay, J.D., Card, S.K. **Cone trees: animated 3d visualizations of hierarchical information**. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 189{194. CHI '91, ACM, New York, NY, USA (1991), http://doi.acm.org/10.1145/108844.108883
- [Johnson & Sneiderman] B. Johnson and B. Shneiderman. **Tree–maps: a Space–filling Approach**
- **to the Visualization of Hierarchical Information Structures**, Proceedings of IEEE Visualisation’91, IEEE CS Press, pp. 275–282, 1991.
- [Knuth] D. E. Knuth. **Fundamental Algorithms**, volume 1 of The Art of Computer Programming. Addison-Wesley, Reading, MA, 1st edition, 1968.
- [Wetherell & Shannon] **Tidy Drawings of Trees**, IEEE Trans. on Softw. Eng., 5(5):514-520, 1979.
- [Reingold & Tilford] E. Reingold and J. Tilford. **Tidier drawing of trees**, IEEE Trans. Softw. Eng., SE-7(2):223–228, 1981.
- [Kleiberg, van de Wetering & van Wijk] E. Kleiberg, H. van de Wetering, and J.J. Van Wijk. **Botanical visualization of huge hierarchies**. In Proc. of the IEEE Symposium on Information Visualization, 2001.

Literature

- [van Wijk & van de Wetering] J. van Wijk, H. van de Wetering. **Cushion treemaps: visualization of hierarchical information.** In Proceedings of IEEE Symposium on Information Visualization (San Francisco, USA, 1999), IEEE Press, pp. 73–78.
- [Kruskal & Landwehr] Kruskal, J. B., and Landwehr, J. M. **Icicle plots: Better displays for hierarchical clustering**. The American Statistician 37, 2 (1983), 162 – 168.
- [Sneiderman]Schneiderman, B. **Tree visualization with tree-maps: 2-d space-filling approach**. ACM Transactions on Graphics 11, 1 (1992), pp 92 - 99.
- [Balzer & Deussen] M. Balzer and O. Deussen. **Voronoi treemaps.** in Proceedings of the IEEE Symposium on Information Visualization, ser. INFOVIS'05. IEEE Computer Society, 2005.

Visualization of Time-Oriented Data

Characterization of Time-Oriented Data

- Differences to other data:
  - One extra independent dimension (x,y,z) → (x,y,z,t)
  - Events have a starting point and endpoint → duration
  - Events may overlap in time

- Attributes (A) associated with events may now be expressed as a function (f) of time (t)
  - A_1: $\{a_{1i} = f_1(t_i)\}$ mit $i \epsilon [1,n]$, n = number of points in time
  - A_2: $\{a_{2i} = f_2(t_i)\}$ mit $i \epsilon [1,n]$, n = number of points in time
  - …
- Attributes may be nominal, ordinal, quantitative

Examples of Time–Oriented Data

- Stock prices
- Weather data
- Sports data
- Electronic health records, e.g., medicines taken
- Electrocardiograms (ECG)
- Cities visited
- University courses attended

…

Modeling Time in Information Systems[Aigner11]

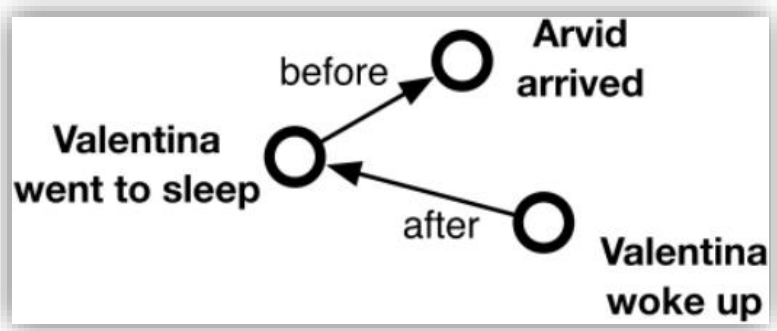
- Goal is not to perfectly imitate physical dimension
- Instead, provide a model of time that
  - reflects phenomena under consideration and
  - supports the analysis task at hand.

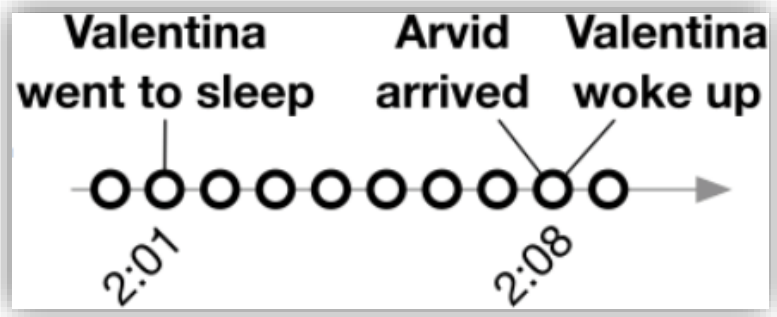
- Design criteria for modeling the time domain
  - **Scale**: ordinal vs. quantitative (discrete vs. continuous)
  - **Scope**: point-based vs. interval-based
  - **Arrangement**: linear vs. cyclic
  - **Viewpoint**: ordered vs. branching vs. multiple perspectives

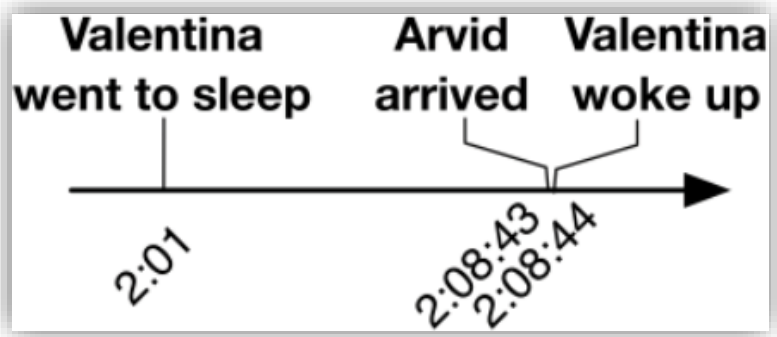
Scale

- Ordinal

- Quantitative
  - Discrete
  - Continuous



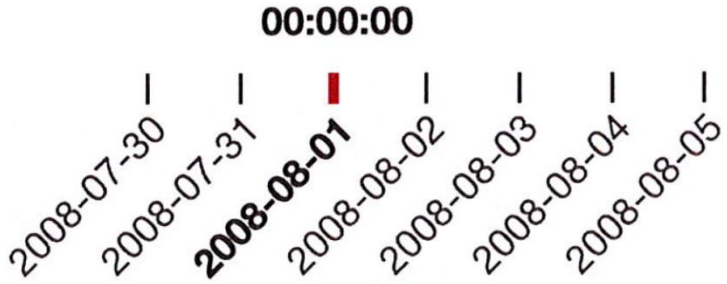


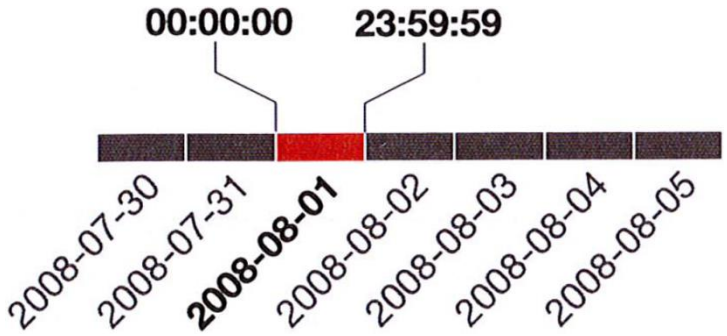


Source of images:
[Aigner11]

Scope

- Point-based
  - Similar to points in Euclidean space, i.e. temporal extent = 0
  - No information given about region between two points in time
- Interval-based
  - Subsections of time having a temporal extent > 0

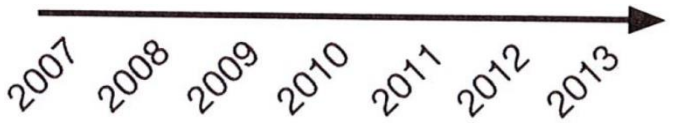




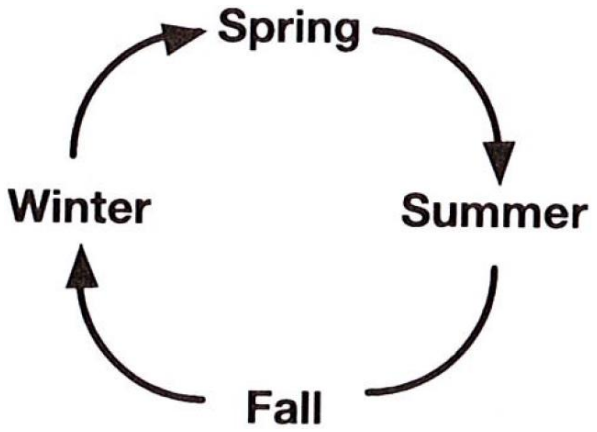
Source of images:
[Aigner11]

Arrangement

- Linear
  - Natural view, from past to future
  - Each time value has unique predecessor/successor



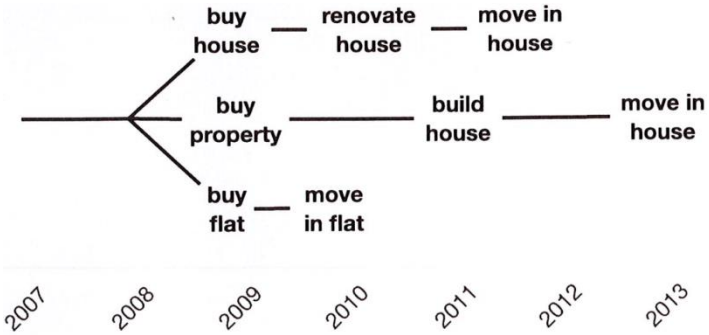
- Cyclic
  - Conveys periodicity
  - Common, e.g., in seasonal variations and monthly averages
  - A is predecessor as well as successor of B

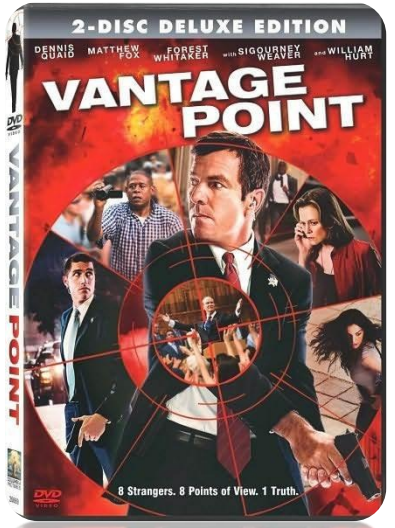


Source of images: [Aigner11]

Viewpoint

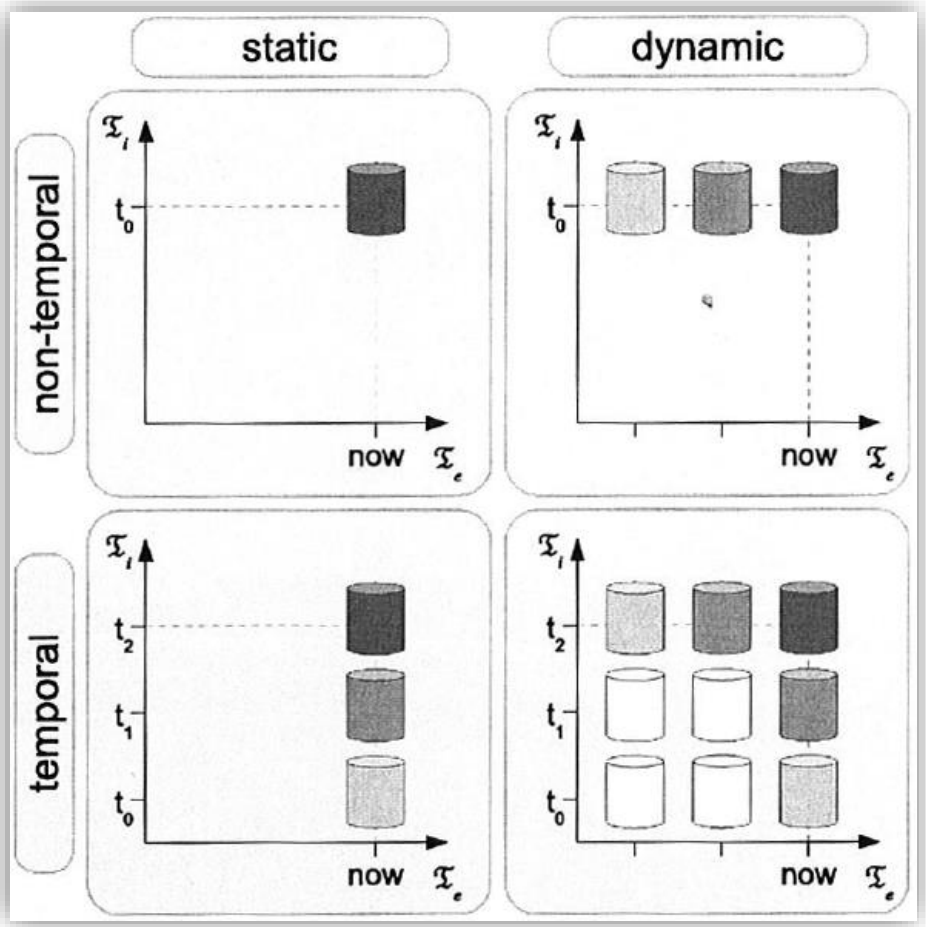
- Ordered
  - Things happen one after the other
- Branching
  - Multiple strands of time brunch out
  - Description of alternative scenarios for future but also for past (possible causes of a decision)
  - Only ONE alternative will happen or has happened
- Multiple Perspectives
  - Simultaneous, even contrary views of time
  - Common, e.g., in multi-run simulations and eyewitness reports





Relating Data and Time

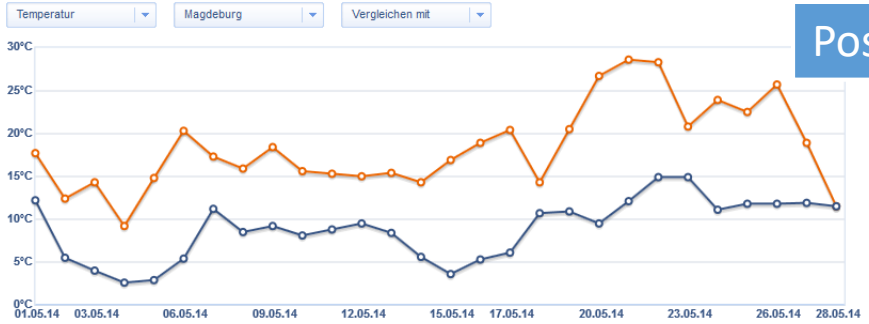
- Internal time $\mathfrak{T}_e$
  - Temporal dimension inherent in the data model
  - Describes data history
- External time $\mathfrak{T}_i$
  - Temporal dimensions extrinsic to the data model
  - Describes how data evolves over time



[Aigner11]

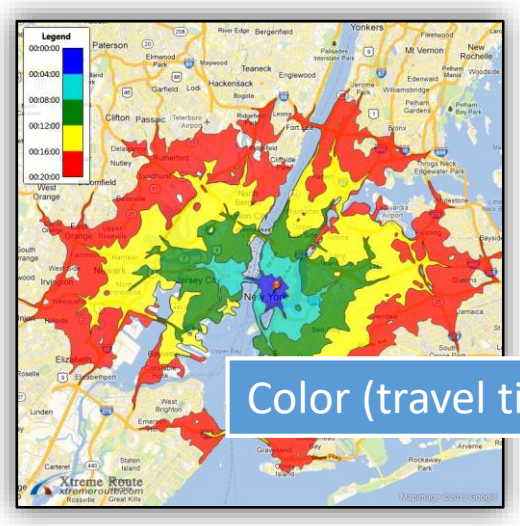
Mapping of Time

- **Static** display of temporal relations
  - Mapping time to visual variables



Position

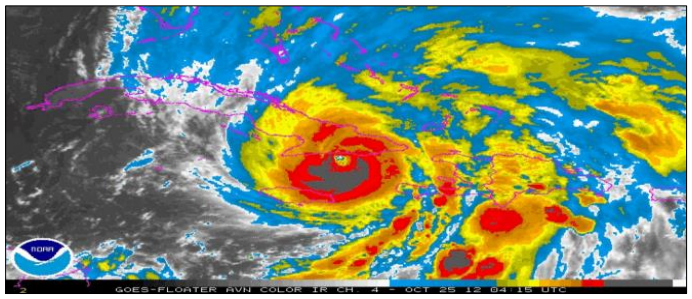
http://www.wetter.com/



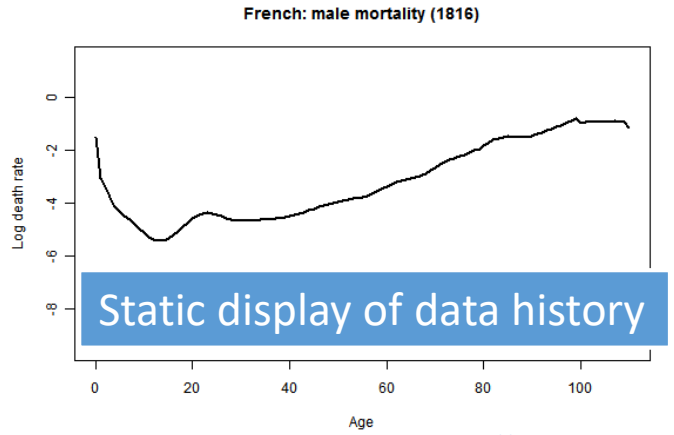
Color (travel time)

http://xtremeroute.com/

- **Dynamic** display of temporal relations
  - Animating visualization over time



http://photos.silive.com/



Static display of data history

http://robjhyndman.com/

Time–Oriented User Tasks [Dachselt11]

Tasks beyond Shneiderman's taxonomy (recall lecture 1)

- Find all events before, after or during a time span or a moment
- Sort data according to time
- Detect and compare periodic events
- Detect trends and patterns

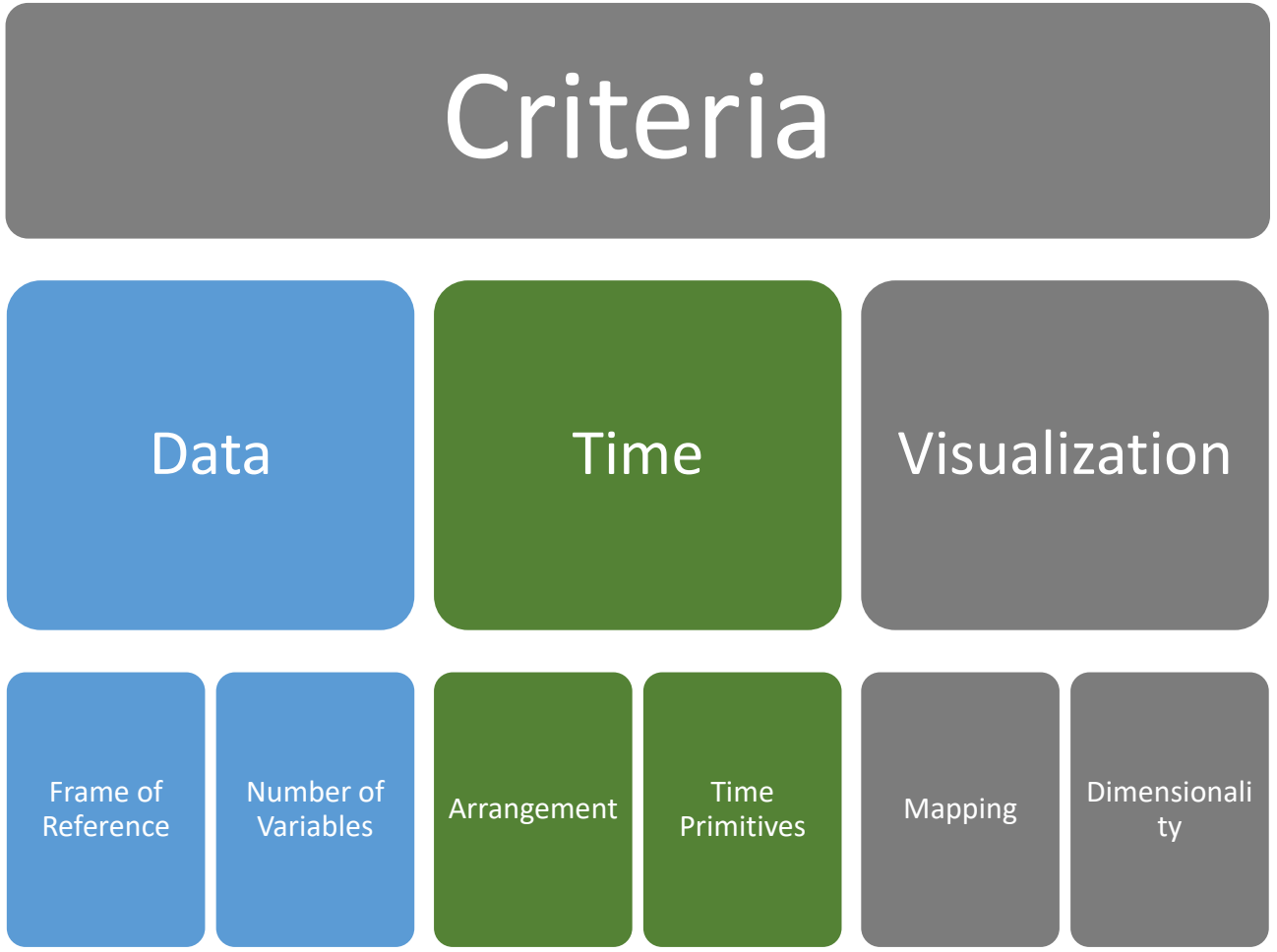
Time-Oriented User Questions [Aigner11]

- Does a data element exist at a particular point in time?
  (Existence of data element)
- When does a data element exist? Is there a cyclic behavior?
  (Temporal location)
- How long is the time span of a data element?
  (Time interval)
- How often does a data element occur?
  (Temporal pattern)
- How quickly does a data element change or how many differences exist between data elements over time?
  (Rate of change)
- In which order do data elements occur?
  (Sequence)
- Do data elements exist together?
  (Synchronization)

Visualization Techniques
– Taxonomy –

From “Visualization of Time-Oriented Data” by Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski, Springer, 2011.

Overview


Criteria
Data
Time
Visualization
Frame of Reference
Number of Variables
Arrangement
Time Primitives
Mapping
Dimensionality

Data: Frame of Reference

- Abstract
  - Data were collected in non-spatial context
  - Not connected to some spatial layout
  - Example: balance of bank accounts

- Spatial
  - Data contain inherent spatial layout
  - Often, geographical positions
  - Example: census data

Data: Number of Variables

reference:

Variables:

- Univariate
  - Single variable
  - Only one data value per temporal event
- Multivariate
  - Multiple variables
  - Multiple data values per temporal event

Time: Arrangement

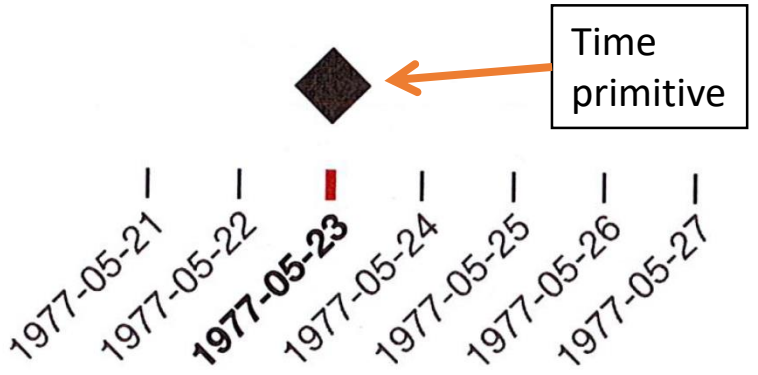
- Linear
  - Natural view, from past to future
  - Each time value has unique predecessor/successor

- Cyclic
  - Conveys periodicity
  - Finite set of recurring time elements, e.g., seasons
  - A is predecessor as well as successor of B

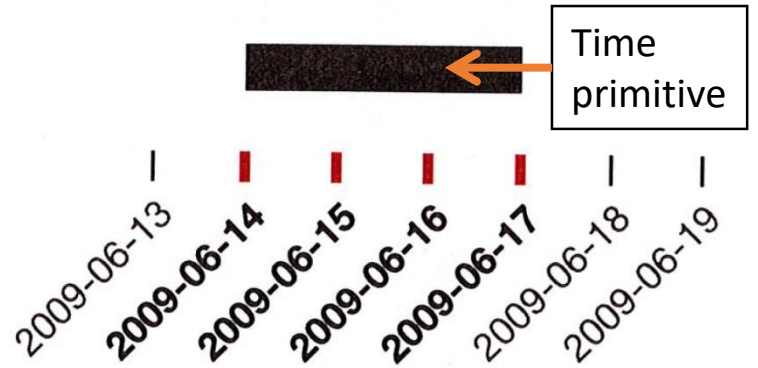
Time: Time Primitives

Time primitives relate data to time

- Instant
  - Refers to a single point in time
- Interval
  - Refers to portion of time
  - Can be represented by two instants denoting beginning and end of the interval



[June 14, 2009; June 17, 2009]



Source of images: [Aigner11]

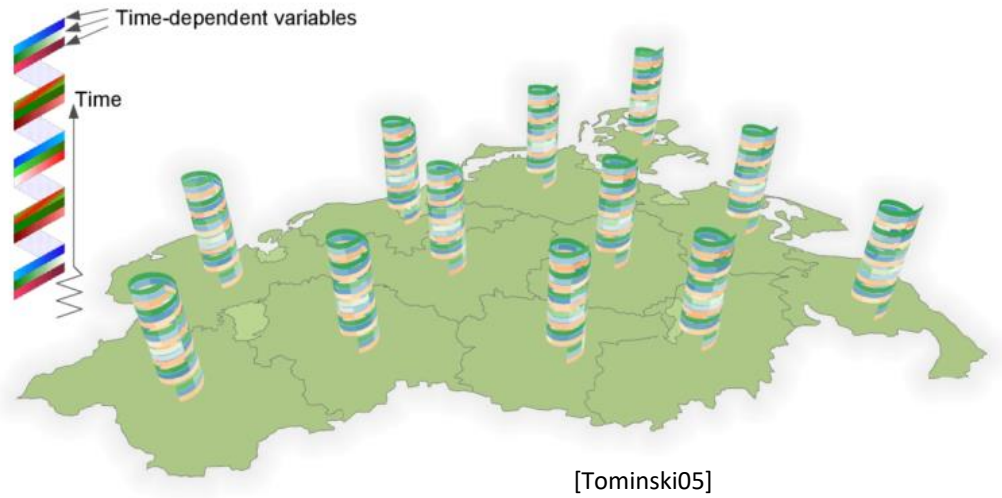
Visualization: Mapping

- Static display of temporal relations
  - Mapping time to visual variables

- Dynamic display of temporal relations
  - Animating visualization over time

Visualization: Dimensionality

- 2D
  - Time is mapped to visual axis on the display (unless dynamic mapping approach is used)
  - Time axis mostly aligned with a coordinate axis (but circular or spiral axes are also possible)
  - One axis free for data representation
- 3D
  - Additional axis to encode information
  - Particularly data with spatial frame of reference may benefit from 3D
  - Example: x- and y-axis represent longitude and latitude and z-axis encodes time



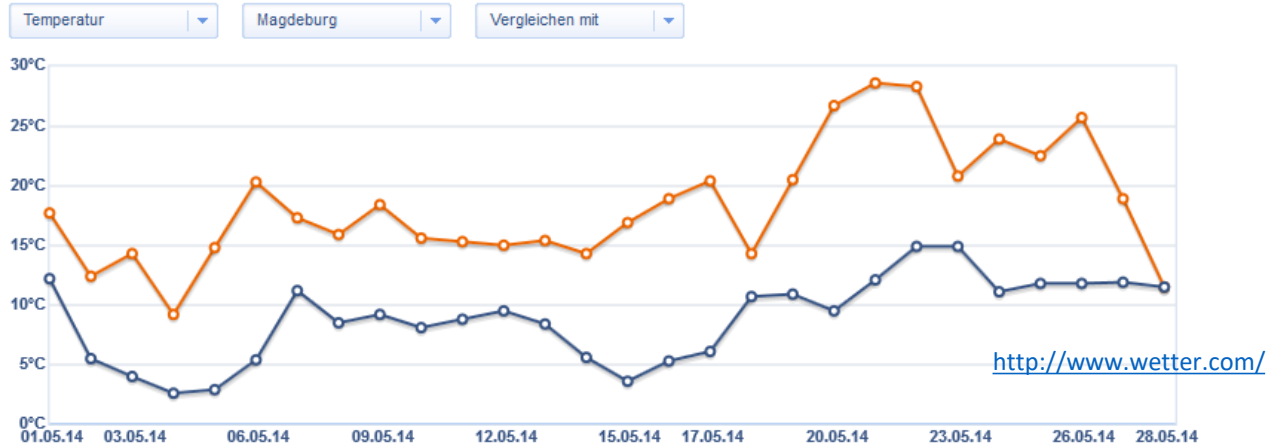
[Tominski05]

Visual Survey

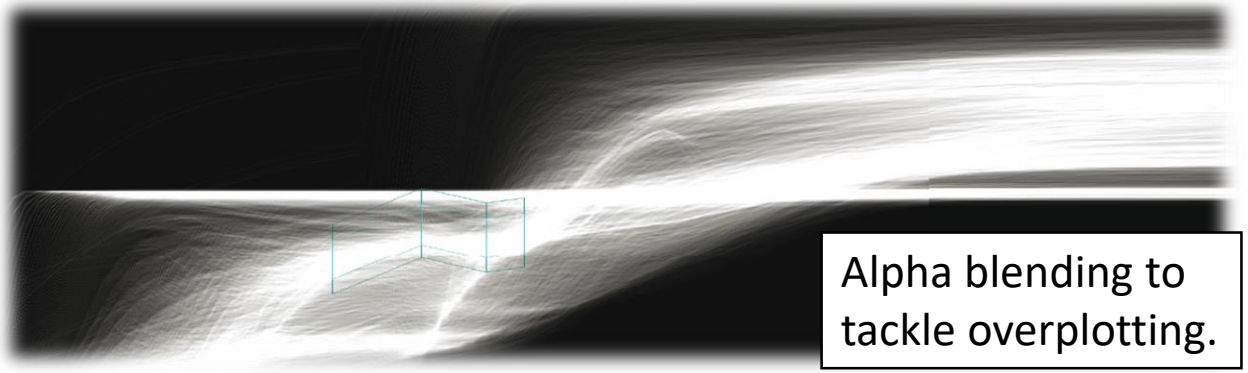
Visualization Techniques
– Examples –

Line Graph

- Time on x-axis and attribute on y-axis



http://www.wetter.com/



Alpha blending to tackle overplotting.

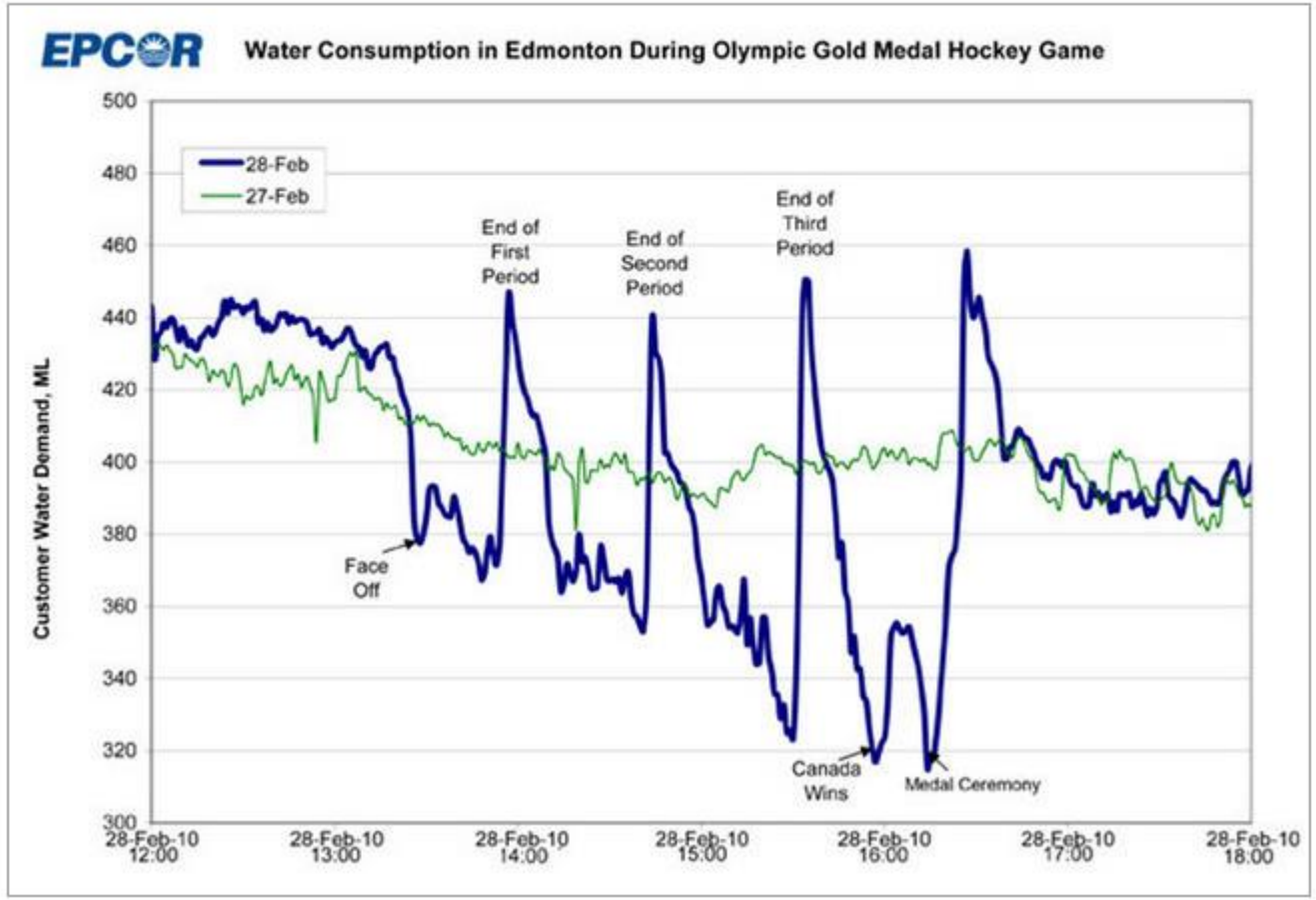
[Muigg08]

Data
reference: **abstract**
Variables: **both**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Funny Line Graph

What If Everybody in Canada Flushed At Once?

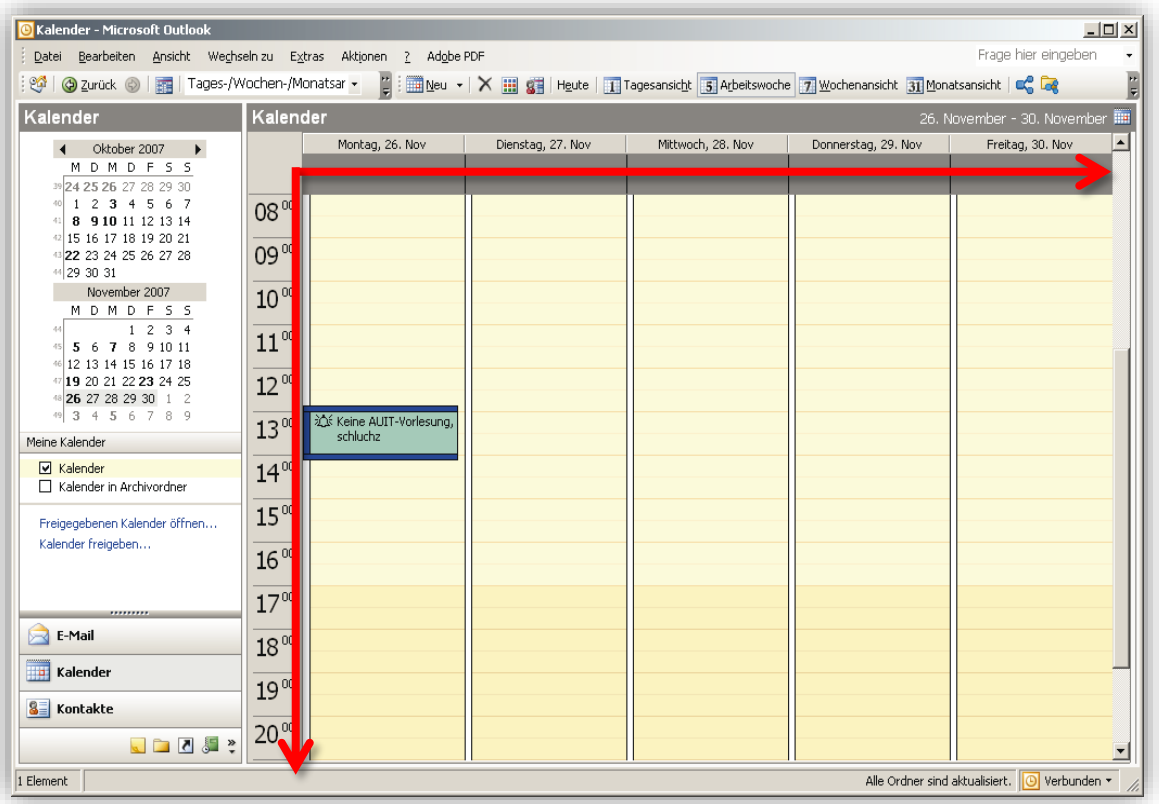
Written by Pats Papers | Monday, 8 March 2010 2:42 PM



Discovered by John Stasko at http://www.patspapers.com/

Calendar Matrix

- See available and blocked times
- Different views on the data: week, month, year



[Dachselt11]

Data

reference: **abstract** Variables: **both**

Time

arrangement: **linear** primitives: **both**

Vis

mapping: **static** dimensionality: **2D**

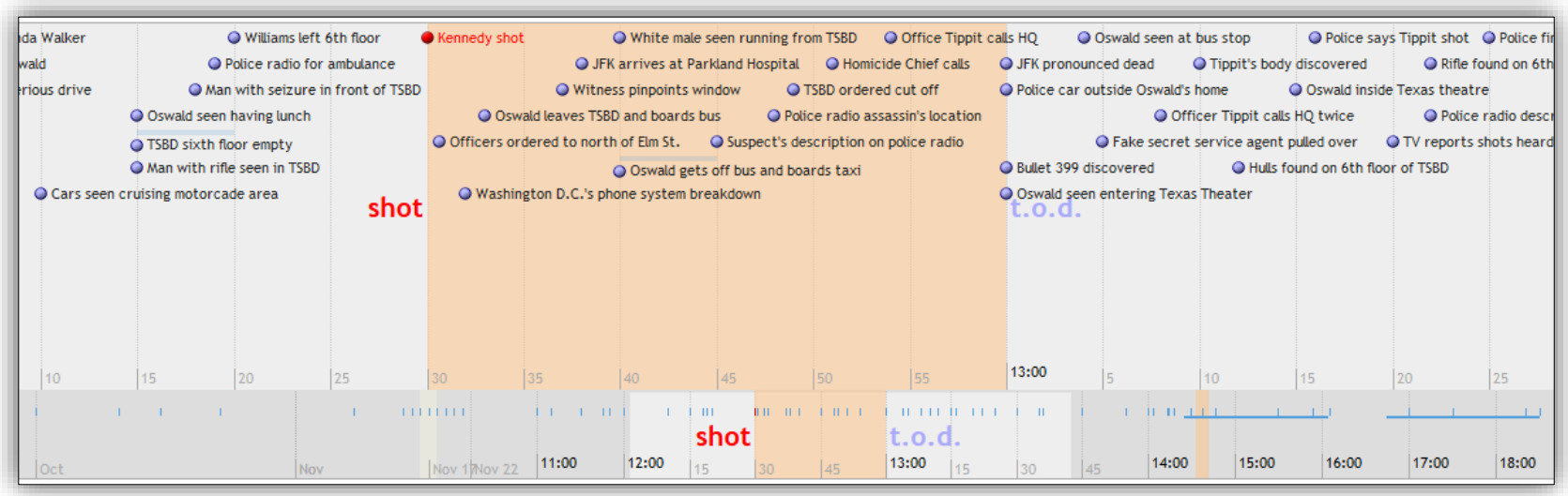
Timeline

- Overview of events in chronological order

By Jesus Diaz-Gizmodo http://infographiclist.com/

Interactive Timeline Example

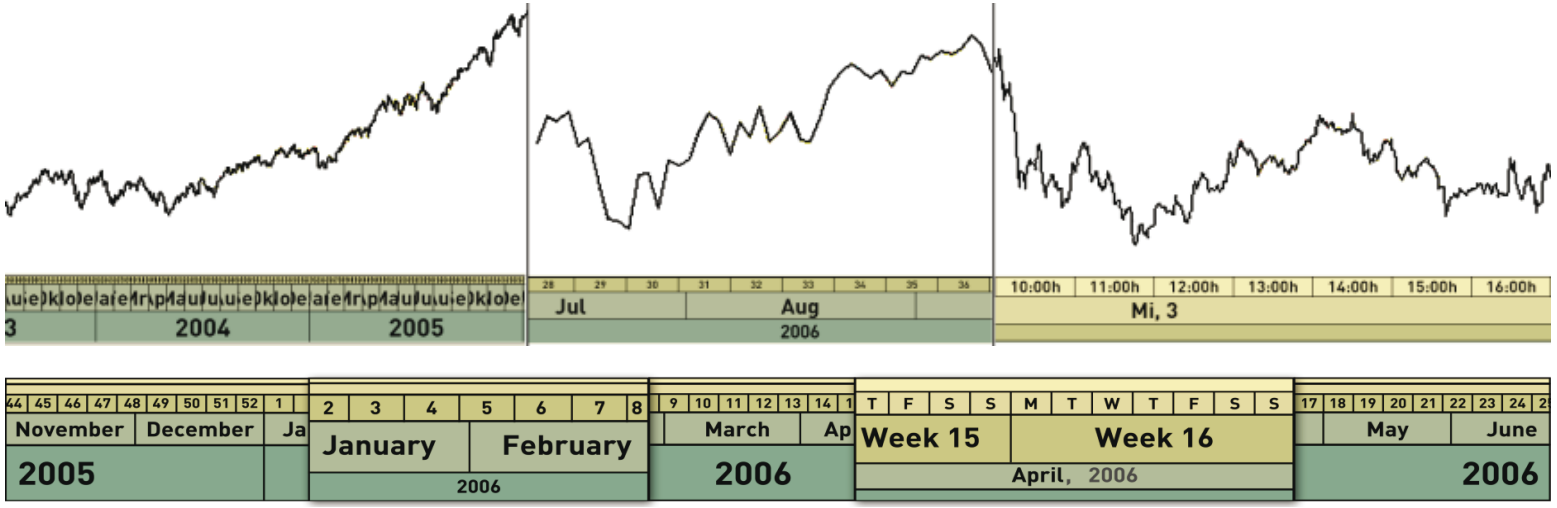
- Assassination of John F. Kennedy
- Combination of events and time spans
- Two widgets with different levels of detail
- Dragging interaction for traveling through time



http://www.simile-widgets.org/timeline/

Timeline Navigation Aid *TimeZoom* [Dachselt06]

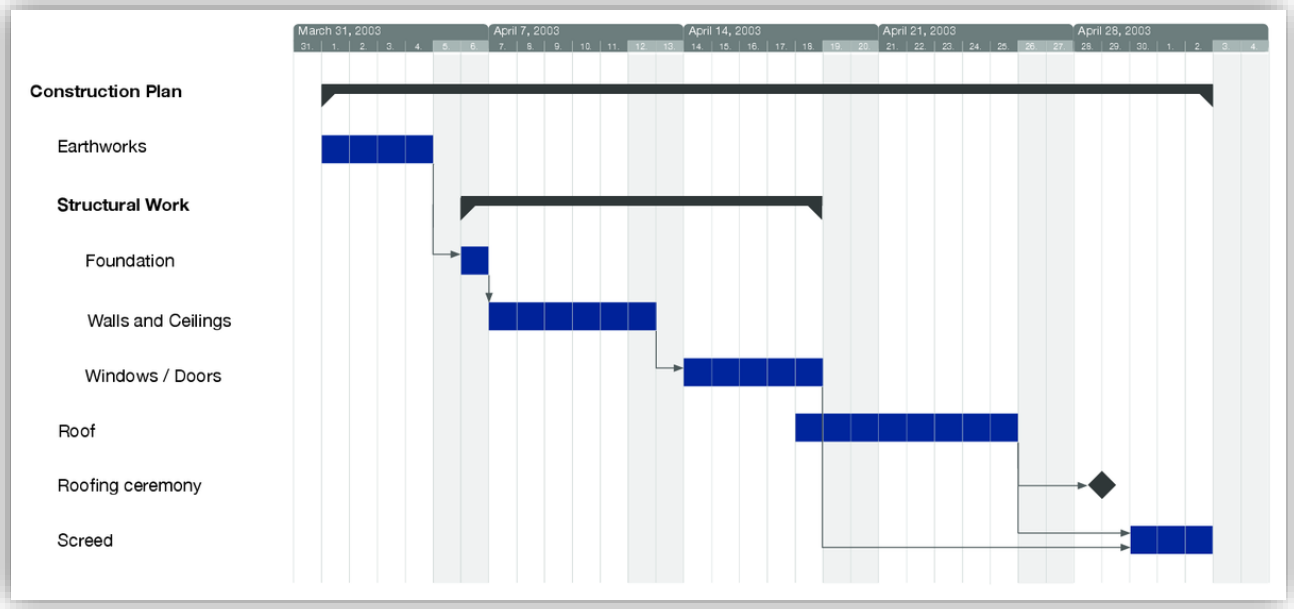
- Continuous zooming for traveling through different granularities of time: year → month → day → hour
- Restriction of zooming to focus regions → preserves context thereby simplifying orientation
- Multiple focus regions with different levels of detail



[Dachselt06]

Gantt Chart

- Widely used in project management
- Display of (possibly overlapping) tasks with start time and duration and display of milestones
- Tasks can be grouped resulting in a hierarchy

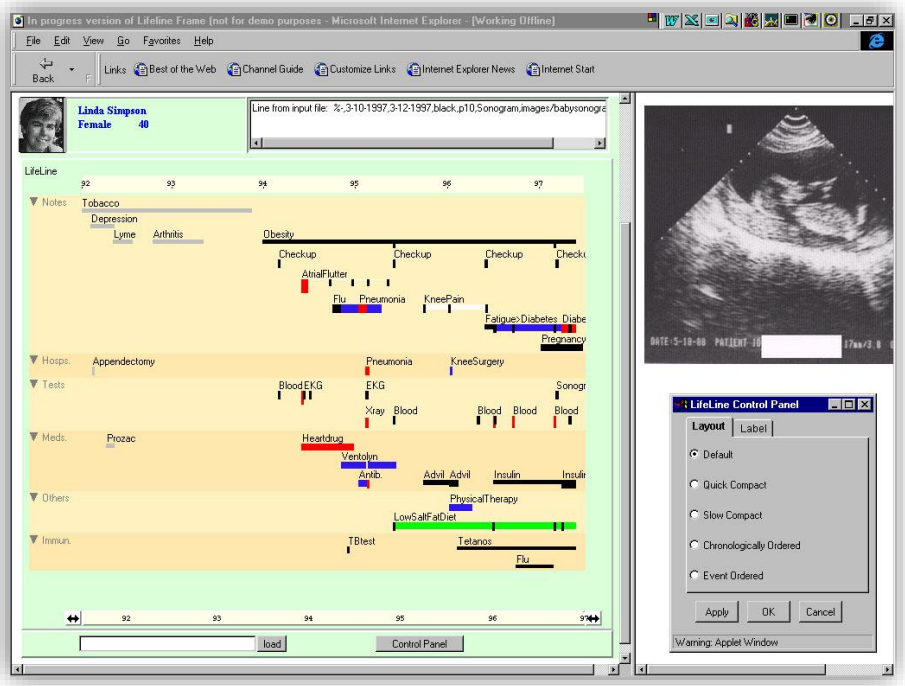


[Aigner11]

Data
reference: **abstract**
Variables: **univariate**
Time
arrangement: **linear**
primitives: **both**
Vis
mapping: **static**
dimensionality: **2D**

LifeLines[Plaisant98]

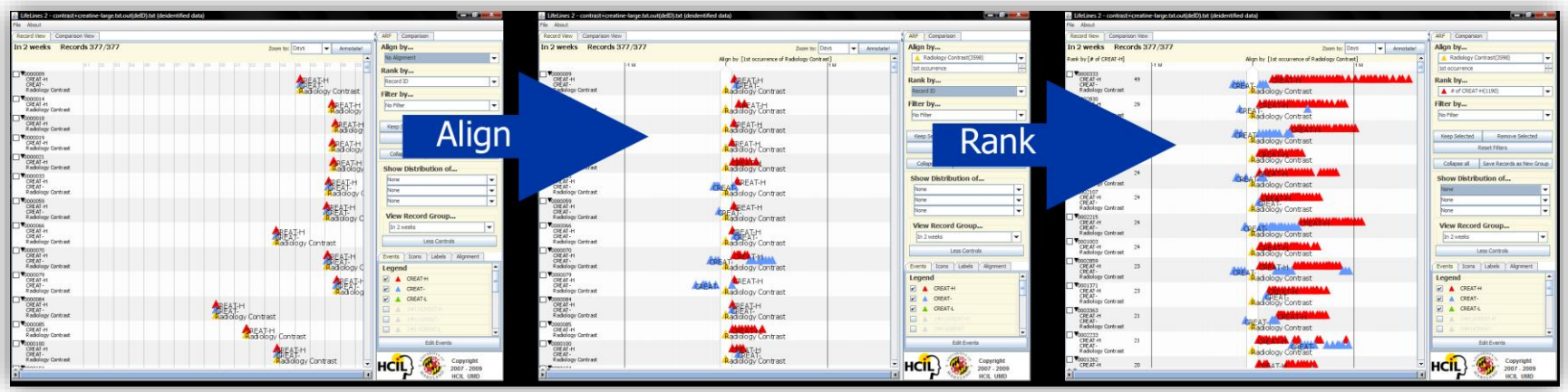
- Comprehensive visualization of health-related incidents in personal history and patient records
- Per incident, horizontal line on time scale with line length corresponding to duration of incident



[Plaisant98]

LifeLines2[Wang08]

- Comprehensive visualization of many records (LifeLines1 restricted to one record at a time)
- Need to align, rank, and filter the records
- Medical application:
  - Look for temporal coincidence of events
  - Avoid time-consuming pan&zoom

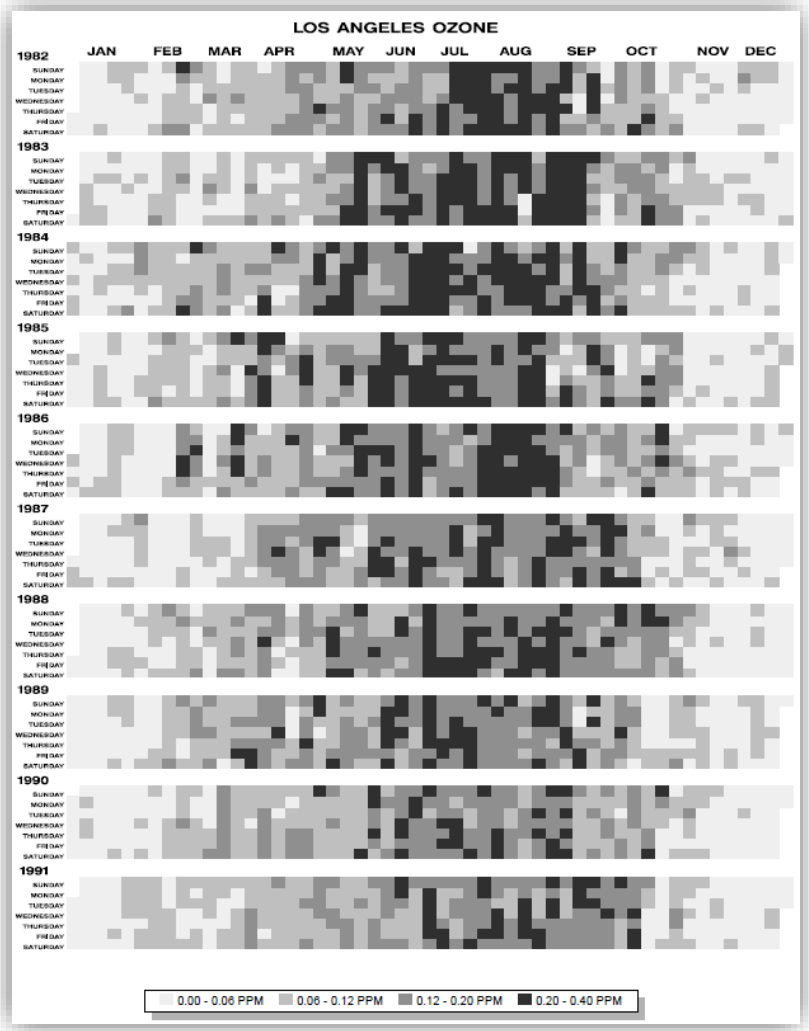


[Wang08]

Data
reference: **abstract**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Tile Maps[Mintz97]

- Arrange data values according to different temporal granularities in a matrix view
- Example: matrix of months and weekdays, cells represents days
- Map day-based attribute to lightness of cells
- Stack matrices to observe seasonal changes



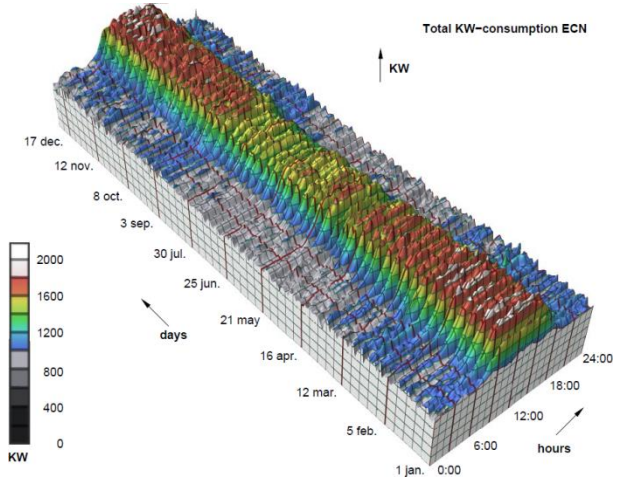
[Mintz97]

Data
reference: **abstract**
Variables: **univariate**
Time
arrangement: **cyclic**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Calendar View [Wijk&Selow99]

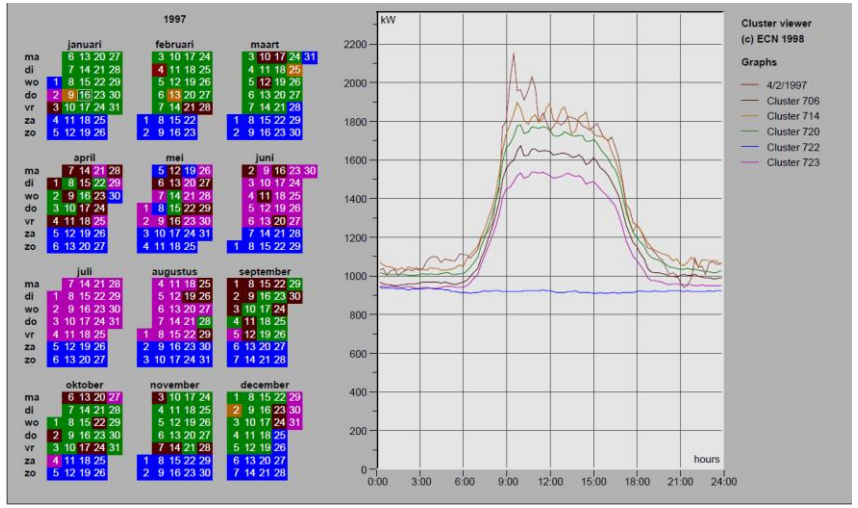
- Visualization of daily patterns over a year
- Example: power demand of a research facility
- Alternative visualizations: Height field of all daily patterns and calendar view of clustered patterns

•Shows all data
•Reveals seasonal trends



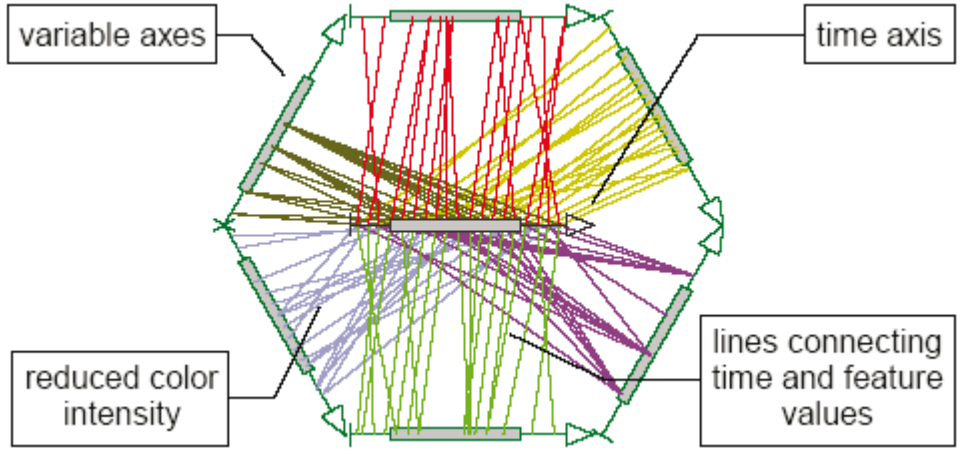
[Wijk&Selow99]

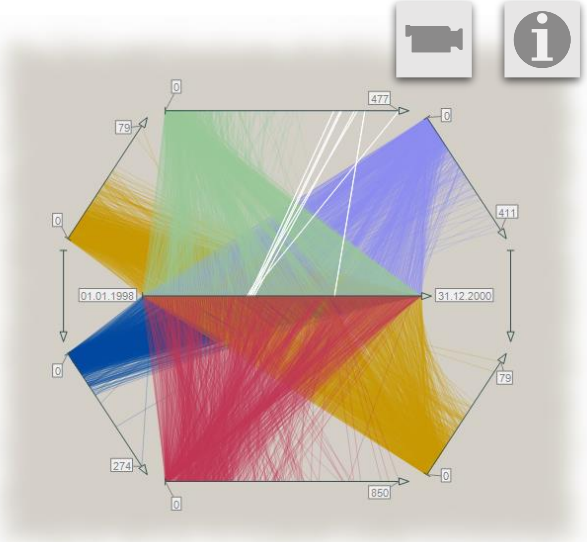
•Shows variation over the week
•Reveals distribution of typical daily patterns



TimeWheel [Tominski04]

- Time axis in the center of the view
- Variable axes arranged in circular fashion
- Colored lines connect time and attribute axes
- Zooming the domain supported for each axis
- Wheel rotation to parallel axes of interest

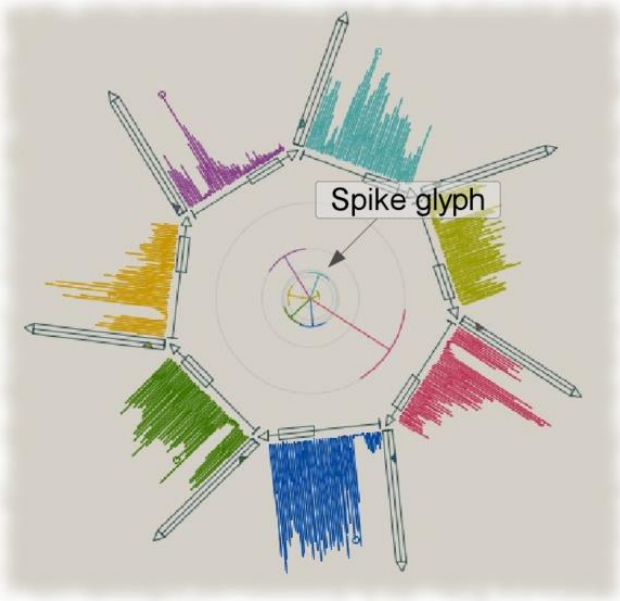


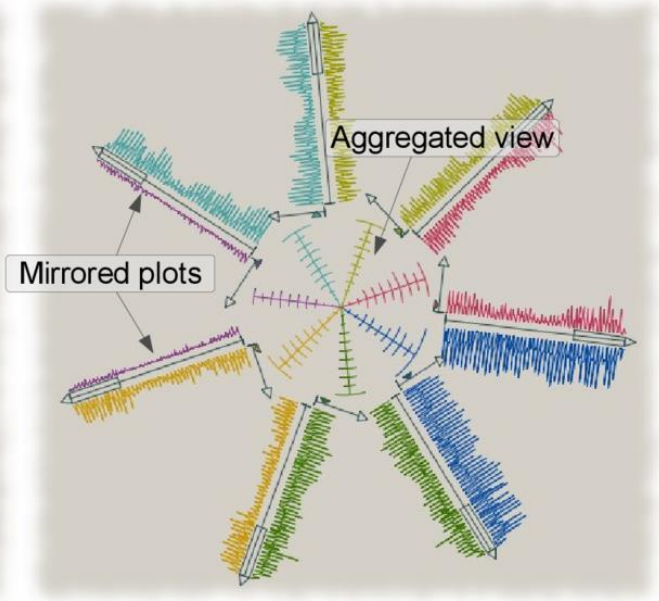


[Tominski04]

MultiComb

- One time axis is drawn per variable
- Axes are arranged in a circular fashion about the center or extend radially from it
- Glyphs for value comparison or aggregated value history are displayed in the center



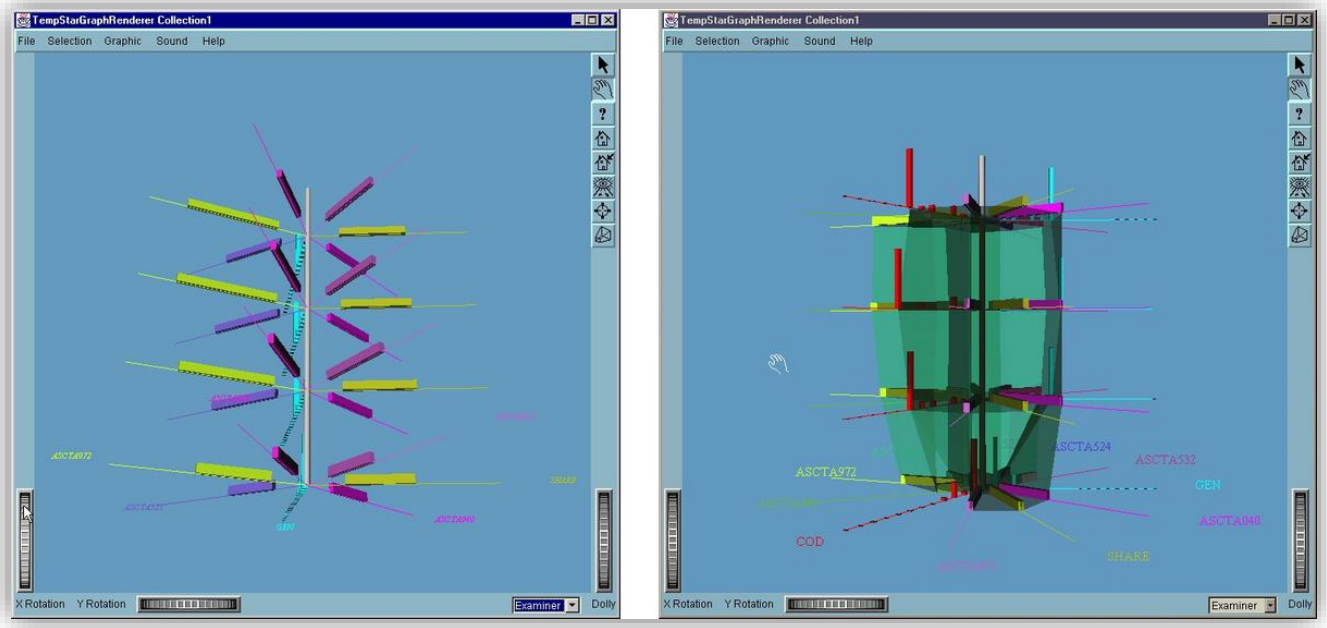


[Aigner11]

Data
reference: **abstract**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Temporal Star[Noirhomme-Fratuire02]

- Chronologically ordered stack of stars
- Each axis of a star shows single value, e.g., stock price, or histogram of quantitative attribute
- Transparent veil enhances perception of evolution

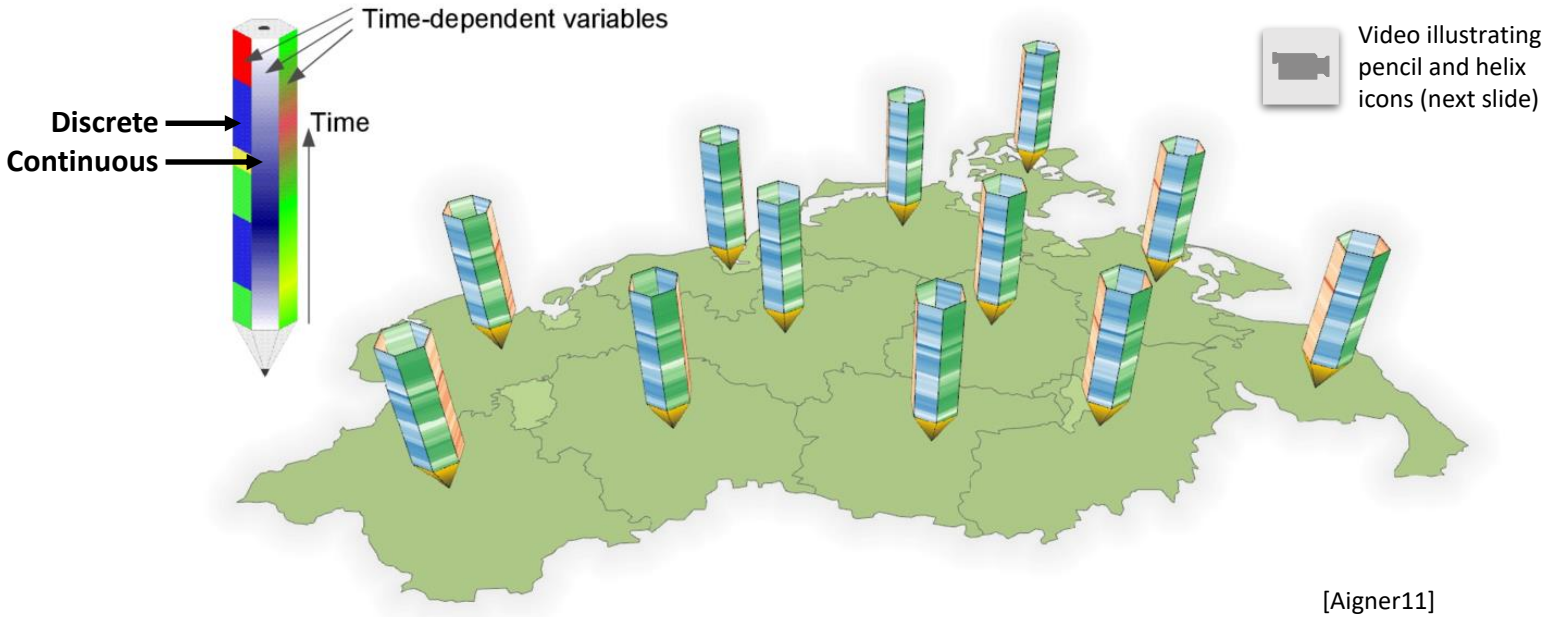


[Aigner11]

Data
reference: **abstract**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **3D**

Pencil Icons [Tominski05]

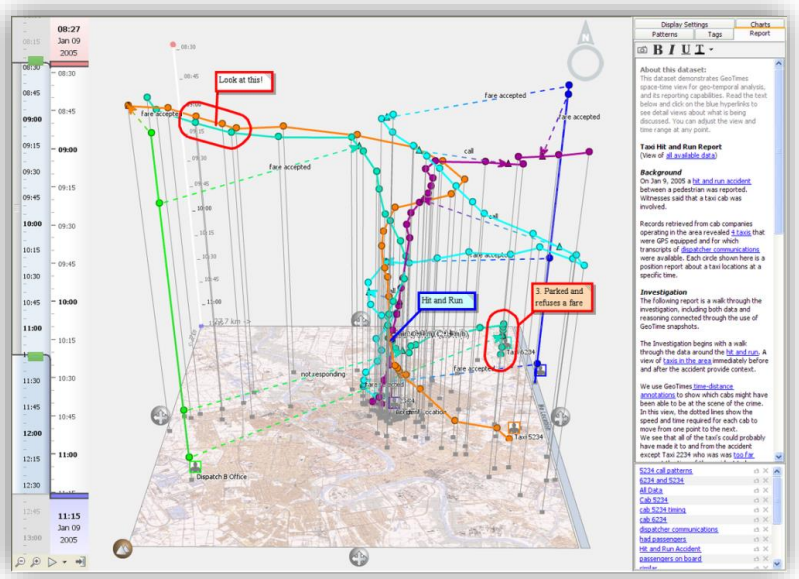
- Time-dependent attributes are mapped onto sides of a pencil starting at the tip
- Combination of discrete and continuous scales
- Positioning of multiple pencils on a map



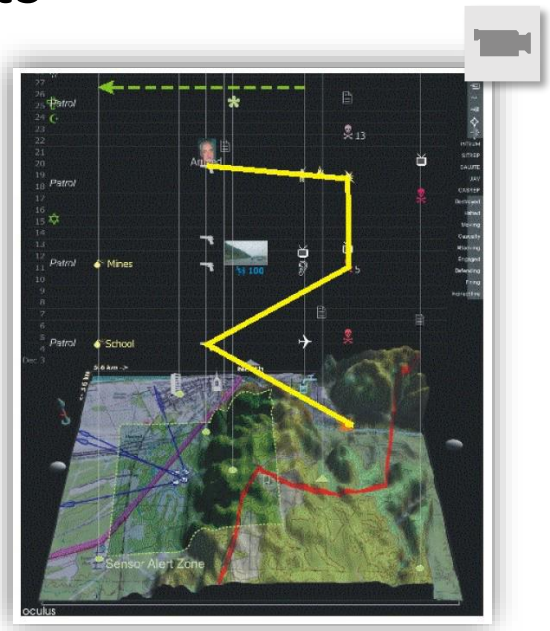
[Aigner11]

GeoTime [Kapler05]

- Map illustrates spatial context
- Time is mapped vertically along third dimension
- Order of events is visualized along routes
- Linking of geographic view with 2D charts



[Aigner11]

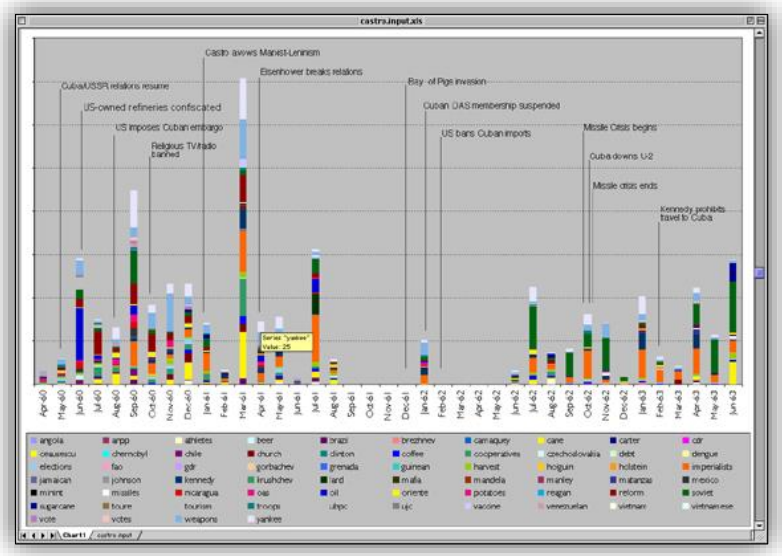


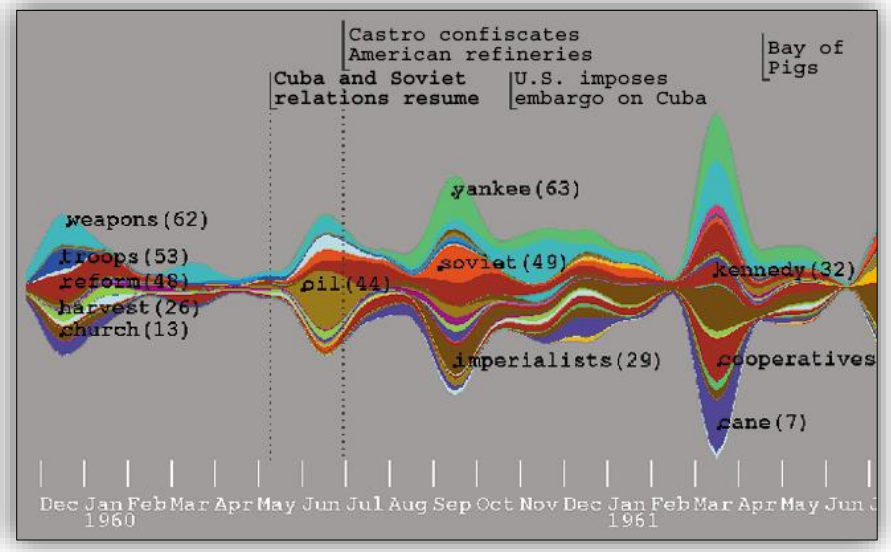
[Kapler05]

Data
reference: **spatial**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **dynamic**
dimensionality: **3D**

ThemeRiver [Havre00]

- Depicts changes in thematic content over time
- Continuous visualization of stacked bar chart
- Each topic is displayed as colored “current”
- Order of topics matters



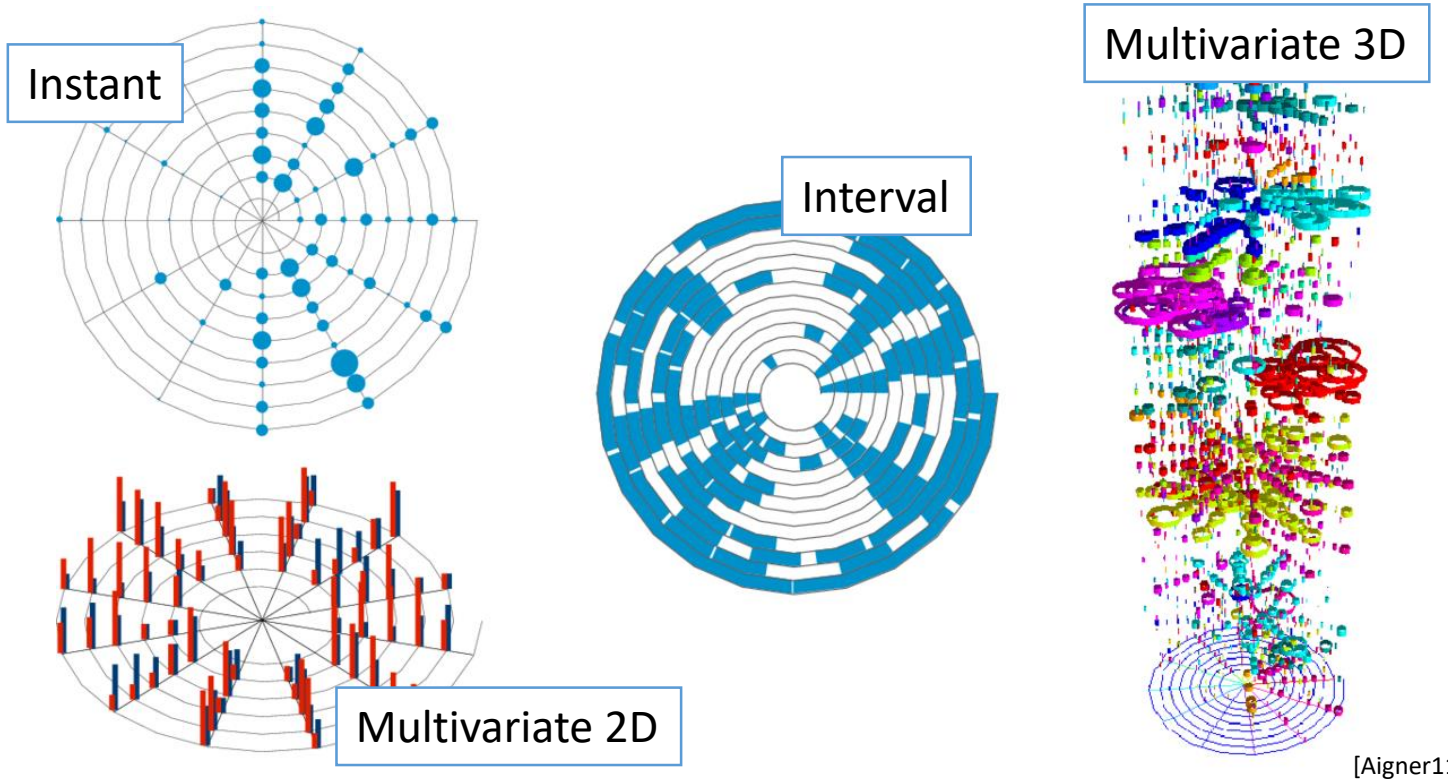


[Havre00]

Data
reference: **abstract**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Spiral Display[Carlis&Konstan98]

- Archimedean spirals to represent time domain
- Interactive adjustment of spiral's cycle length for discovering periodic patterns

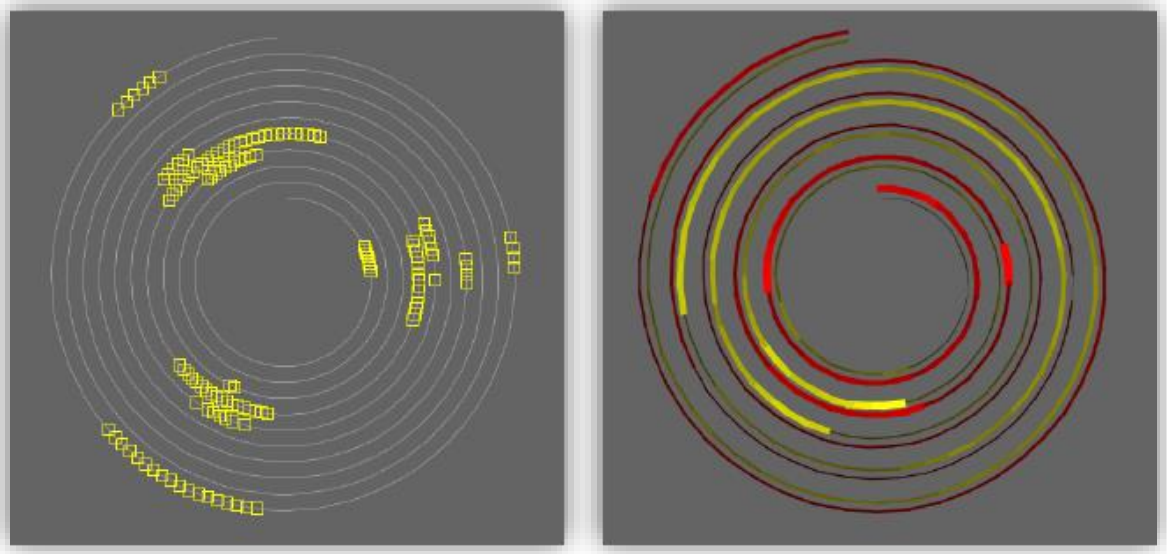


[Aigner11]

Data
reference: **abstract**
Variables: **both**
Time
arrangement: **cyclic**
primitives: **both**
Vis
mapping: **static**
dimensionality: **2D/3D**

Spiral Graph[Weber01]

- Spiral time axis for detection of periodic behavior
- Data mapped along spiral path
- Icons for nominal data and color, line thickness, or texture for quantitative data
- Intertwining spirals for multivariate data

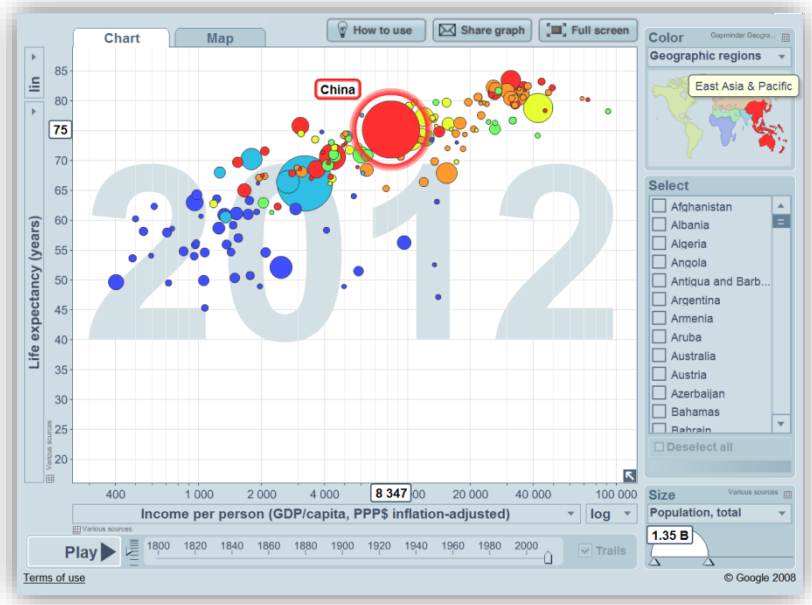


[Aigner11]

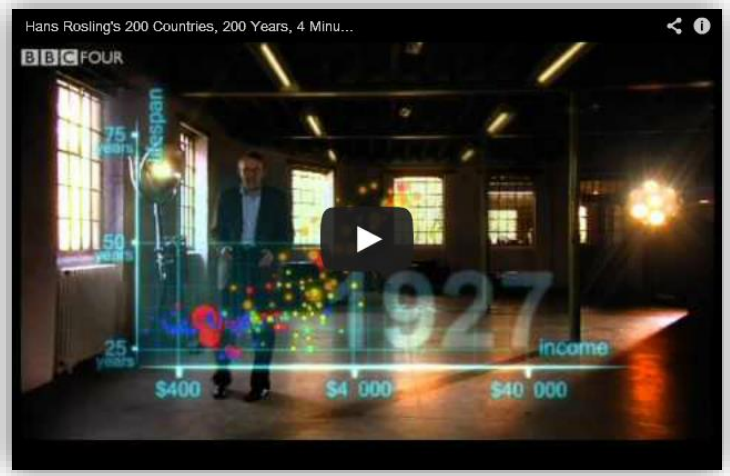
Data
reference: **abstract**
Variables: **both**
Time
arrangement: **cyclic**
primitives: **instant**
Vis
mapping: **static**
dimensionality: **2D**

Animated Bubble Chart

- 2D scatterplot with point size modified according to quantity or another third attribute
- Color may encode a fourth attribute
- Animation used to illustrate change over time



www.gapminder.org/world/



Hans Rosling's statistics:
200 countries, 200 years, 4 minutes

Data
reference: **abstract**
Variables: **multivariate**
Time
arrangement: **linear**
primitives: **instant**
Vis
mapping: **dynamic**
dimensionality: **2D**

Questions???