Chapter 7
Survey of Visualization Techniques

> Today we live in an information-based technological world. The problem is that this is an invisible technology. Knowledge and information are invisible. They have no natural form. It is up to the conveyor of the information and knowledge to provide shape, substance, and organization [...]
>
> Norman (1993, p. 104)

A major part of this book is dedicated to a survey of existing visualization techniques for time and time-oriented data. The complexity of the visualization problem, which results from the multitude of aspects having an impact on the visual representation, already suggests that there must be a variety of techniques – and indeed there are numerous ones. The following survey lists many techniques, some of them very specific to a particular application domain, others more general with potential applicability in other fields than the one described. We are aware that our survey cannot be exhaustive. This is due to the fact that visualization of time-oriented data is a hot research area which is constantly yielding new techniques. Moreover, we have seen that visualization solutions might be highly application-dependent (what and why aspects, see Chapter 4), and hence, it is virtually impossible to dig out every tiny variation of existing visualization approaches that might be hidden in the vast body of scientific literature across application domains. Therefore, we took care to include a wide spectrum of key techniques, both classic ones with proven usefulness and contemporary ones with potential impact.

The survey lists the techniques on a per-page basis. This allows for easy access when a quick reference to a particular technique is sought by the reader. Each page briefly describes the background, explains the main idea and concepts, and indicates the application of a particular technique. The description is accompanied with a reference to the original publication or a list of references in the case that multiple publications propose or make use of the same approach. As this is a visualization book, a figure demonstrates the technique in use or the conceptual construction of the visual representation. Additionally, we provide a side-bar that categorizes the technique. To keep the categorization at a manageable level, we do not use the full-

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_7,

scale classification introduced in the previous chapters, but instead focus on three key criteria: data, time, and vis(ualization). For each key criterion, we introduce further sub-criteria and corresponding characteristics. The side-bar information follows this pattern:

- **data**
  - *frame of reference* – abstract vs. spatial
  - *variables* – univariate vs. multivariate
- **time**
  - *arrangement* – linear vs. cyclic
  - *time primitives* – instant vs. interval
- **vis**
  - *mapping* – static vs. dynamic
  - *dimensionality* – 2D vs. 3D

Where possible, a distinct classification will be given. However, this is not always possible, particularly for more general and flexible visualization approaches. In such cases, we will indicate that multiple characteristics hold per category.

7.1 Techniques

It was not easy to decide on a good order for the techniques in the survey. If we sorted by the name of a technique or by the year of first publication, we would lose the semantic relationships of techniques, and similar techniques would be scattered across the survey just because they have different names. Therefore, we use the categories provided in the side-bar to structure the survey. At the top-most levels, the order is determined by the data characteristics. The survey will start with techniques for abstract time-oriented data, and later we will move on to techniques for spatial data. Accordingly, techniques for univariate time-oriented data will precede those for multivariate data. At the subsequent levels, we order the techniques by their affiliation to the categories *arrangement*, *time primitives*, *mapping*, and *dimensionality*.

Table 7.1 provides an overview of all techniques that are included in our survey along with their categorization. This table might also be used to search for techniques that fulfill certain criteria. Suppose we have an abstract univariate dataset containing time instants as well as intervals and would like to find appropriate visualization techniques. In this case, we would look at the columns *abstract*, *univariate*, *instant*, and *interval* in Table 7.1 and search for lines that match these criteria. For the given example, we would find the techniques Gantt chart, perspective wall, and spiral display as primary matches. For quick access to a technique that is known by name, the index at the end of the book provides the corresponding page number.

	data				time				vis				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
Point Plot	■		■		■		■		■		■		152
Line Plot	■		■		■		■		■		■		153
Bar Graph, Spike Graph	■		■		■		■		■		■		154
Sparklines	■		■		■		■		■		■		155
SparkClouds	■		■		■		■		■		■		156
Horizon Graph	■		■		■		■		■		■		157
TrendDisplay	■		■		■		■		■		■		158
Decision Chart	■		■		■		■		■		■		159
TimeTree	■		■		■		■		■		■		160
Arc Diagrams	■		■		■		■		■		■		161
Interactive Parallel Bar Charts	■		■		■		■		■			■	162
TimeHistogram 3D	■		■		■		■		■			■	163
Intrusion Monitoring	■		■		■		■			■	■		164
Anemone	■		■		■		■			■	■		165
Timeline	■		■		■			■	■		■		166
Gantt Chart	■		■		■		■	■	■		■		167
Perspective Wall	■		■		■		■	■	■			■	168
DateLens	■		■		■			■	■		■		169
TimeNets	■		■		■			■	■		■		170
Paint Strips	■		■		■			■	■		■		171
PlanningLines	■		■		■			■	■		■		172
Time Annotation Glyph	■		■		■			■	■		■		173
SOPO Diagram	■		■		■			■	■		■		174
Silhouette Graph, Circular Silhouette Graph	■		■		■	■	■		■		■		175
Cycle Plot	■		■		■	■	■		■		■		176
Cluster and Calendar-Based Visualization	■		■		■	■	■		■		■	■	177
Tile Maps	■		■		■	■	■		■		■		178
Multi Scale Temporal Behavior	■		■		■	■	■		■		■		179
Recursive Pattern	■		■	■	■	■	■		■		■		180
GROOVE	■		■		■	■	■		■		■		181
SolarPlot	■		■		■	■	■		■		■		182
SpiraClock	■		■			■		■		■	■		183
Enhanced Interactive Spiral	■		■			■	■		■		■		184
Spiral Graph	■		■	■		■	■		■		■		185
Spiral Display	■		■	■		■	■	■	■		■	■	186
VizTree	■		■	■	■		■		■		■		187

continued on next page

Table 7.1: Overview and categorization of visualization techniques.

	data				time				vis				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
TimeSearcher	■			■	■		■		■		■		188
TimeSearcher 3, River Plot	■			■	■		■		■		■		189
BinX	■			■	■		■		■		■		190
LiveRAC	■			■	■		■		■		■		191
LifeLines2	■			■	■		■		■		■		192
Similan	■			■	■		■		■		■		193
CareCruiser	■			■	■		■		■		■		194
Layer Area Graph	■			■	■		■		■		■		195
Braided Graph	■			■	■		■		■		■		196
ThemeRiver	■			■	■		■		■		■		197
3D ThemeRiver	■			■	■		■		■			■	198
Stacked Graphs	■			■	■		■		■		■		199
TimeWheel	■			■	■		■		■		■		200
MultiComb	■			■	■		■		■		■		201
VIE-VISU	■			■	■		■		■		■		202
Timeline Trees	■			■	■		■		■		■		203
Pixel-Oriented Network Visualization	■			■	■	■	■		■		■		204
CiteSpace II	■			■	■		■		■		■		205
history flow	■			■	■		■		■		■		206
PeopleGarden	■			■	■	■	■		■		■		207
PostHistory	■			■	■	■	■		■		■		208
MOSAN	■			■	■		■		■		■	■	209
Data Tube Technique	■			■	■		■		■			■	210
Kiviat Tube	■			■	■		■		■			■	211
Temporal Star	■			■	■		■		■			■	212
Time-tunnel	■			■	■		■		■			■	213
Parallel Glyphs	■			■	■	■	■		■			■	214
Worm Plots	■			■	■		■		■			■	215
Software Evolution Analysis	■			■	■		■		■			■	216
InfoBUG	■			■	■		■		■	■	■		217
Gravi++	■			■	■		■		■	■	■		218
CircleView	■			■	■		■		■	■	■		219
Trendalyzer, Animated Scatter Plot	■			■	■		■			■	■		220
TimeRider	■			■	■		■			■	■		221
Process Visualization	■			■	■		■			■	■		222
Flocking Boids	■			■	■		■			■		■	223

continued on next page

Table 7.1: Overview and categorization of visualization techniques.

	data				time				vis				
	frame of reference		variables		arrangement		time primitives		mapping		dimensionality		
	abstract	spatial	univariate	multivariate	linear	cyclic	instant	interval	static	dynamic	2D	3D	page
Time Line Browser	■			■	■		■	■	■		■		224
LifeLines	■			■	■		■	■	■		■		225
PatternFinder	■			■	■		■	■	■		■		226
Continuum	■			■	■		■	■	■		■		227
EventRiver	■			■	■		■	■	■		■		228
FacetZoom	■			■	■		■	■	■		■		229
Midgaard	■			■	■		■	■	■		■		230
VisuExplore	■			■	■		■	■	■		■		231
KNAVE II	■			■	■		■	■	■		■		232
Circos	■			■	■	■	■	■	■		■		233
Kaleidomaps	■			■		■	■		■		■		234
Intrusion Detection	■			■		■	■		■			■	235
Small Multiples	■	■	■	■	■	■	■		■		■	■	236
EventViewer	■	■	■	■	■	■	■	■	■		■		237
Ring Maps	■	■	■	■	■	■	■	■	■		■	■	238
Time-Oriented Polygons on Maps		■	■		■	■	■		■		■		239
Icons on Maps		■	■	■	■	■	■	■	■		■	■	240
Value Flow Map		■	■		■		■		■		■		241
Flow Map		■	■		■		■	■	■		■		242
Time-Varying Hierarchies on Maps		■	■		■		■		■			■	243
VIS-STAMP		■		■	■		■		■		■		244
Space-Time Cube		■	■	■	■		■	■	■			■	245
Spatio-Temporal Event Visualization		■	■	■	■		■		■			■	246
Space-Time Path		■	■	■	■		■	■	■			■	247
GeoTime		■		■	■		■		■	■		■	248
Pencil Icons		■		■	■		■	■	■			■	249
Data Vases		■		■	■		■	■	■		■	■	250
Wakame		■		■	■		■		■			■	251
Helix Icons		■		■		■	■	■	■			■	252
count	87	17	47	65	94	25	93	30	94	11	80	28	

Table 7.1: Overview and categorization of visualization techniques.

Let us now start the survey with the simple, but most widely used techniques for visualizing time-oriented data – the point plot and the line plot.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

Point Plot

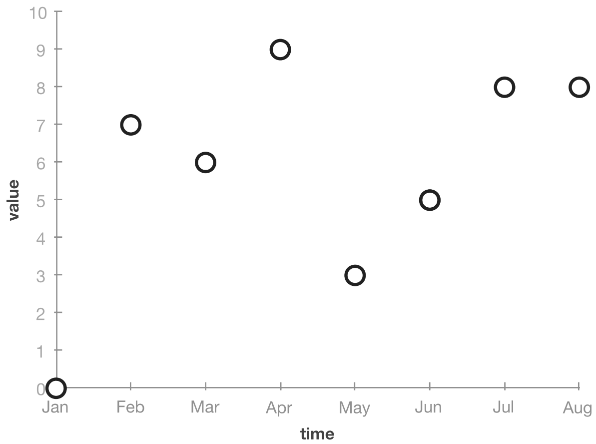


Fig. 7.1: Data are displayed as points in a Cartesian coordinate system where time and data are mapped to the horizontal axis and the vertical axis, respectively.
Source: Authors.

One of the most straightforward ways of depicting time-series data is using a Cartesian coordinate system with time on the horizontal axis and the corresponding value on the vertical axis. A point is plotted for every measured time-value pair. This kind of representation is called point plot, point graph, or scatter plot, respectively. Harris (1999) describes it as a 2-dimensional representation where quantitative data aspects are visualized by distance from the main axis. Many extensions of this basic form such as 3D techniques (layer graph) or techniques that use different symbols instead of points are known. This technique is particularly suited for emphasizing individual values. Moreover, depicting data using position along a common scale can be perceived most precisely by the human perceptual system.

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

Line Plot

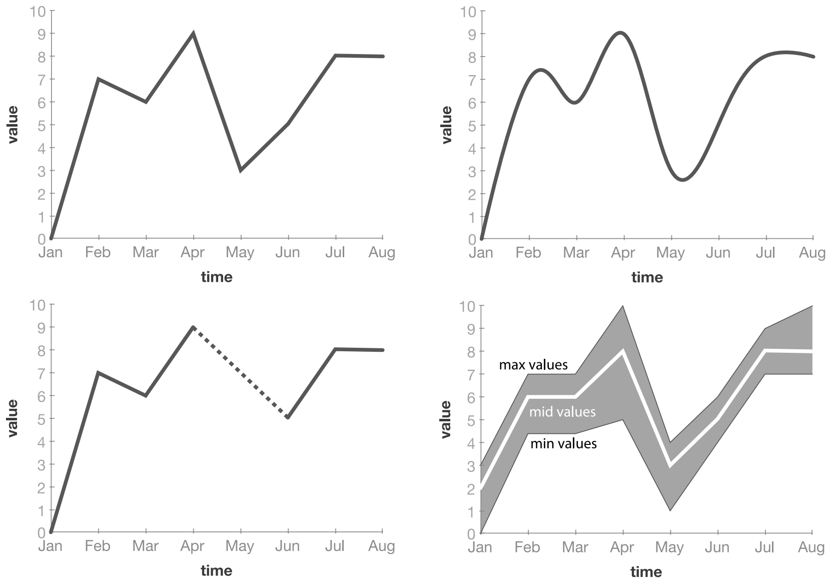


Fig. 7.2: Successive data points are connected with lines to visualize the overall change over time. (top-left: straight lines; top-right: Bézier curves; bottom-left: missing data; bottom-right: band graph).
Source: Authors.

The most common form of representing time-series are line plots. They extend point plots (↪ p. 152) by linking the data points with lines which emphasizes their temporal relation. Consequently, line plots focus on the overall shape of data over time. This is in contrast to point plots where individual data points are emphasized. As illustrated in Figure 7.2, different styles of connections between the data points such as straight lines, step lines (instant value changes), or Bezier curves can be used depending on the phenomenon under consideration. However, what has to be kept in mind is that one can not be sure in all cases about the data values in the time interval between two data points and that any kind of connection between data points reflects an approximation only. A further point of caution is missing data. Simply connecting subsequent data points might lead to false conclusions regarding the data. Therefore, this should be made visible to the viewer, for instance by using dotted lines (see Figure 7.2, bottom-left). There are many extensions or subtypes like fever graphs, band graphs (see Figure 7.2, bottom-right), layer line graphs, surface graphs, index graphs, or control graphs (see Harris, 1999).

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

Bar Graph, Spike Graph

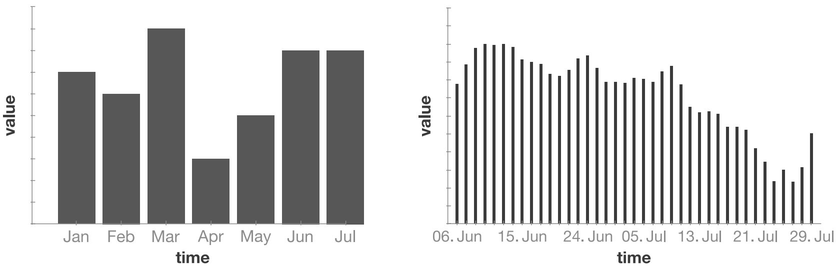


Fig. 7.3: Bar length is used to depict data values. Right: if bars are reduced to spikes the graph is also called a spike graph.
Source: Adapted from Harris (1999).

Bar graphs (see Harris, 1999) are a well known and widely used type of representation where bars are used to depict data values (see Figure 7.3, left). This makes comparisons easier than with point plots. As bar length is used to depict data values, only variables with a ratio scale (having a natural zero) can be represented. Consequently, the value scale also has to start with zero to allow for a fair visual comparison. In contrast to line plots, bar graphs emphasize individual values as do point plots. A variant of bar graphs often used for graphing larger time-series (e.g., stock market data such as price or volume) are spike graphs. As illustrated in Figure 7.3 (right), the vertical bars are reduced so as to appear as spikes, where spike height is again used to encode data values. This way, a good visual balance is achieved between focusing on individual values and showing overall development of a larger number of data values.

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

Sparklines

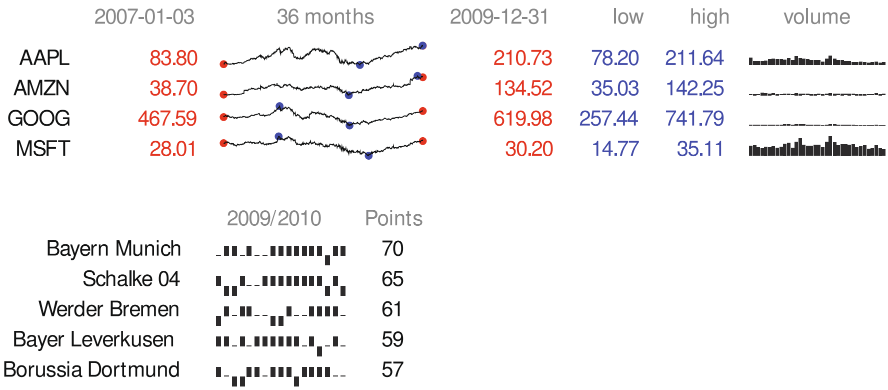


Fig. 7.4: Simple, word-like graphics intended to be integrated into text visualize stock market data (top). Bottom: Soccer season results using ticks (up=win, down=loss, base=draw).
Source: Generated with the sparklines package for LaTeX.

Tufte (2006) describes sparklines as simple, word-like graphics intended to be integrated into text. This adds richer information about the development of a variable over time that words themselves could hardly convey. The visualization method focuses mainly on giving an overview of the development of values for time-oriented data rather than on specific values or dates due to their small size and the omission of axes and labels. Sparklines can be integrated seamlessly into paragraphs of text, can be laid out as tables, or can be used for dashboards. They are increasingly adopted to present information on web pages (such as usage statistics) in newspapers (e.g., for sports statistics), or in finance (e.g., for stock market data). Usually, miniaturized versions of line plots (↪ p. 153) and bar graphs (↪ p. 154) are employed to represent data. For the special case of binary or three-valued data, special bar graphs can be applied that use ticks extending up and down a horizontal baseline. One use for this kind of data are wins and losses of sports teams where the history of a whole season can be presented using very little space. For line plots, the first and last value can be emphasized by colored dots (•) and printing the values themselves textually to the left and right of the sparkline. Moreover, the minimum and maximum values might also be marked by colored dots (•). Besides this, colored bands in the background of the plot can be used to show normal value ranges as for example here 4.8 8.3.

References

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, CT.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

SparkClouds

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

Fig. 7.5: Display of the 25 most important keywords in a series of twelve measurements. The bigger the font size is, the more important is a keyword. Keywords that are not among the current top 25 important keywords but have been among them at an earlier point in time are attenuated by using dimmed color and smaller font size.
Source: Lee et al. (2010). © 2010, IEEE. Used with permission.

Tag clouds visualize a set of keywords weighted by their importance. To this end, a layout of the keywords is computed. By varying font size, color, or other visual variables important keywords are emphasized over less-important keywords. Classic tag clouds, however, are incapable of representing the evolution of keywords. Lee et al. (2010) integrate sparklines (↪ p. 155) into tag clouds in order to visualize temporal trends in the development of keywords. The idea is to visually combine a keyword (or tag) and its temporal evolution. The keyword's importance is encoded with the font size used to render the text, where the size can correspond either to the overall importance of the keyword for the entire time-series or to the importance at a particular point in time. Attached to the keyword is a sparkline that represents the keyword's trend. A color gradient is shown in the background of each keyword-sparkline pair to make this design perceivable as a visual unit. Lee et al. (2010) conducted user studies with sparkclouds and could confirm that sparkclouds are useful and have advantages over alternative standard methods for visualizing text and temporal information.

References

Lee, B., Riche, N., Karlson, A., and Carpendale, S. (2010). SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189.

Horizon Graph

data

frame of reference: **abstract**
variables: **univariate**

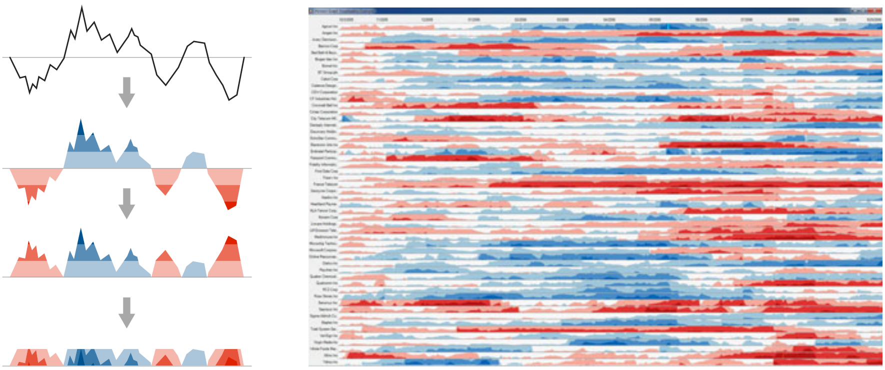


Fig. 7.6: The construction of a horizon graph from a line chart is illustrated on the left. Because horizon graphs require only little screen space they are very useful for comparing multiple time-dependent variables as shown to the right for stock market data.
Source: Left: Adapted from Reijner (2008). Right: Image courtesy of Hannes Reijner.

time

arrangement: **linear**
time primitives: **instant**

Reijner (2008) describes horizon graphs as a visualization technique for comparing a large number of time-dependent variables. Horizon graphs are based on the two-tone pseudo coloring technique by Saito et al. (2005). The left part of Figure 7.6 demonstrates the construction of horizon graphs (from top to bottom). Starting from a common line plot, the value range is divided into equally sized bands that are discriminated by increasing color intensity towards the maximum and minimum values while using different hues for positive and negative values. Then, negative values are mirrored horizontally at the zero line. Finally, the bands are layered on top of each other. This way, less vertical space is used, which means data density is increased while the resolution is preserved. A study by Heer et al. (2009) has shown that mirroring does not have negative effects and that layered bands are more effective than the standard line plot (↪ p. 153) for charts of small size, as for example in the right part of Figure 7.6.

vis

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

References

Heer, J., Kong, N., and Agrawala, M. (2009). Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1303–1312, New York, NY, USA. ACM Press.

Reijner, H. (2008). The Development of the Horizon Graph. In *Electronic Proceedings of the VisWeek Workshop From Theory to Practice: Design, Vision and Visualization*.

Saito, T., Miyamura, H., Yamamoto, M., Saito, H., Hoshiya, Y., and Kaseda, T. (2005). Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 173–180, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

TrendDisplay

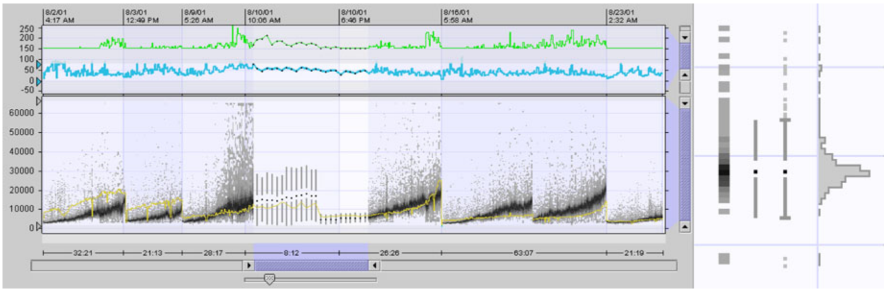


Fig. 7.7: The main panel shows the raw data (drug discovery data) and the top panel depicts derived statistical values. Depending on the available screen space, four different levels of visual abstraction are used: density distributions, thin box plots, box plots plus outliers, and bar histograms (as illustrated to the right).
Source: Brodbeck and Girardin (2003), © 2003 IEEE. Used with permission.

The TrendDisplay technique by Brodbeck and Girardin (2003) allows the analysis of trends in larger time-series. The technique is used for the drug discovery process and in quality control. Basically, the TrendDisplay window is composed of two panels. The main panel on the bottom shows the measured (raw) data and the top panel depicts derived statistical values (see Figure 7.7, left). Four different levels of detail are used in order to cope with large numbers of time points: density distributions, thin box plots, box plots plus outliers, and bar histograms (from low to high level of detail) (see Figure 7.7, right). In the temporal dimension, bifocal focus+context functionality is used for enlarging areas of interest without losing context information about neighboring data. The different levels of detail are chosen automatically depending on the available screen space. Moreover, brushing & linking as well as smooth transitions complete the highly interactive interface.

References

Brodbeck, D. and Girardin, L. (2003). Interactive Poster: Trend Analysis in Large Timeseries of High-Throughput Screening Data Using a Distortion-Oriented Lens with Semantic Zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 74–75, Los Alamitos, CA, USA. IEEE Computer Society.

Decision Chart

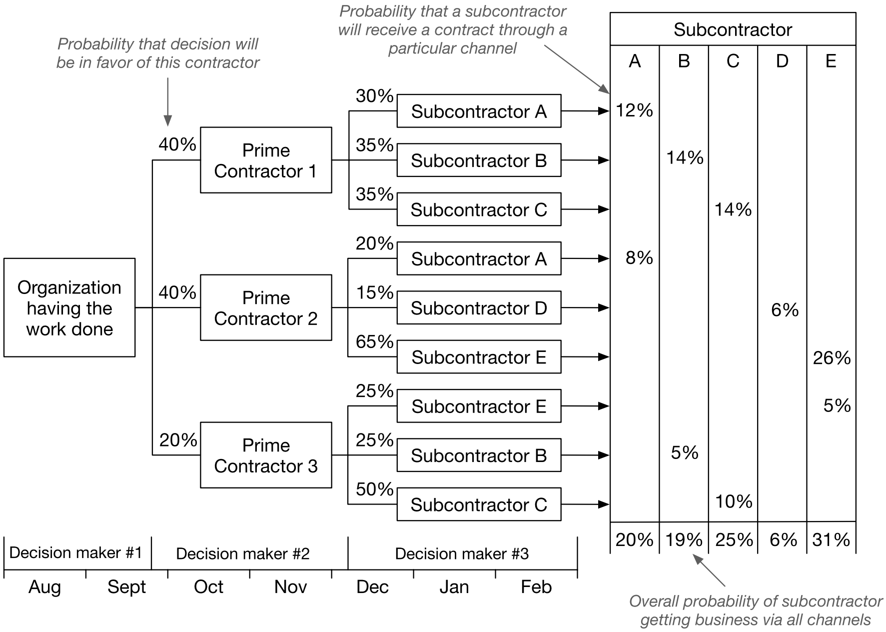


Fig. 7.8: Future decisions and corresponding alternative outcomes are depicted over time along with their probabilities.
Source: Adapted from Harris (1999).

Harris (1999) describes decision charts as a graphical representation for depicting future decisions and potential alternative outcomes along with their probabilities over time. It is one of very few techniques for time-oriented data that use the *branching time* model (see Section 3.1.1). Decision charts use a horizontal time axis along which information elements (decisions and probabilities) are aligned. Multiple decisions for a particular time interval are stacked on top of each other, indicating that they are possible alternatives for that interval. However, the temporal context itself is not of prime interest and is just indicated by a simple time scale on the bottom of the chart. The main advantage of the decision chart is that it allows planners to investigate possible outcomes and implications before decisions are made.

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

TimeTree

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

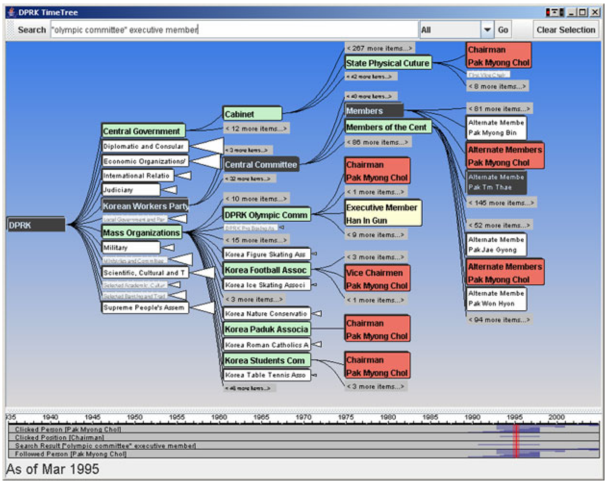


Fig. 7.9: The figure shows changes in the organizational structure of officials in the DPR Korea with focused and important nodes highlighted. The time slider (currently set to March 1995) shows selected personal events and serves for interactive navigation.
Source: Card et al. (2006), © 2006 IEEE. Used with permission.

TimeTree by Card et al. (2006) is a visualization technique to enable the exploration of changing hierarchical organizational structures and of individuals within such structures. The visualization consists of three parts: a time slider, a tree view, and a search interface (see bottom, center, and top of Figure 7.9, respectively). The time slider's main purpose is to allow users to navigate to any point in time. Additionally, it shows information strips with events for a selected set of individuals, and thus, provides insight into where in time interesting things have happened. The tree view shows the snapshot of the organizational structure corresponding to the selected time point. The tree visualization uses a degree-of-interest (DOI) approach to highlight important information. To this end, a specific color scheme is used to indicate certain data characteristics, as for instance, important nodes, nodes matching with search queries, or recently clicked nodes. Low-interest nodes, with regard to the user's current focus and search query, are ghosted, and entire subtrees may be represented as triangular abstractions in order to de-clutter the display and maintain the readability of important nodes. The search interface supports a textual search for individuals in the represented organization.

References

Card, S. K., Suh, B., Pendleton, B. A., Heer, J., and Bodnar, J. W. (2006). Time Tree: Exploring Time Changing Hierarchies. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, Los Alamitos, CA, USA. IEEE Computer Society.

Arc Diagrams

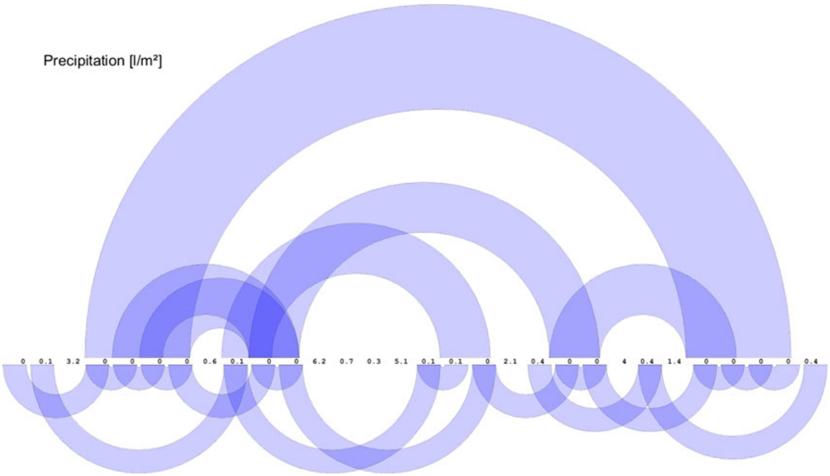


Fig. 7.10: A sequence of data values is shown along the horizontal axis. Matching subsequences are connected by arcs, where arc thickness and height encode subsequence size and occurrence distance, respectively.
Source: Image courtesy of Michael Zornow.

Patterns in sequences of data values can be visualized using arc diagrams. They were introduced as an interactive visualization technique by Wattenberg (2002). Given a sequence of values, the goal is to extract significant subsequences that occur multiple times in the original sequence. The visualization displays the sequence of data values in textual form along the horizontal (time) axis. Occurrences of significant subsequences are visually connected by spanning arcs. The arcs' thickness represents the size of the subsequence, that is, the number of data values in the subsequence. The height of an arc indicates the distance between two successive occurrences of the subsequence. To express data-specific aspects, one can separately use the space above or below the data sequence. This also helps to reduce overlap of arcs. Additionally, transparency is used to allow users to see through overlapping arcs. The visualization can be controlled interactively via several parameters (e.g., minimum size of subsequences or tolerance threshold for fuzzy pattern extraction) to keep the number of arcs at an interpretable level.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Wattenberg, M. (2002). Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 110–116, Los Alamitos, CA, USA. IEEE Computer Society.

Interactive Parallel Bar Charts

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

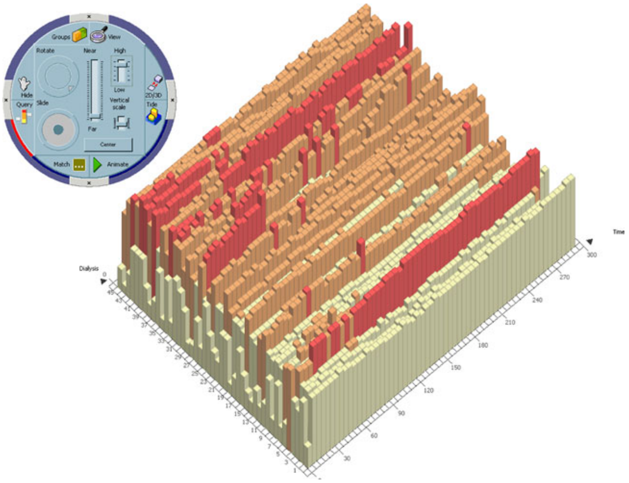


Fig. 7.11: Visualization of clinical time-dependent data where one axis represents different hemodialysis sessions and the other axis represents the series of time steps. One time-dependent variable (e.g., blood pressure) is encoded to the height of the bars.
Source: Chittaro et al. (2003), © 2003 Elsevier. Used with permission.

Chittaro et al. (2003) present a technique for visualizing time-dependent hemodialysis data. To keep the visualization of multiple hemodialysis sessions simple and easy to use for physicians, the design is based on common 3D bar charts, where the height of bars encodes individual data values. Multiple bar charts (one per hemodialysis session) are arranged on a regular grid in a parallel fashion. This visual display is easy to interpret despite the 3D projection. Visual exploration and analysis are facilitated through a wealth of interaction tools. Dynamic filtering combined with a color coding mechanism supports visual classification. To manage occlusions, interactive features are provided, such as flattening individual bars or groups of bars, or leaving only colored squares in the grid. The water level interaction is particularly helpful for comparison tasks: virtual water engulfs all bars with a height below a user-defined threshold. This eases the assessment of similarities and differences with respect to dialysis sessions and time. Additional visual cues support physicians in detecting anomalies or special events in the data and enable them to take necessary actions quickly. For multivariate analysis, an integration with parallel coordinate plots is supported.

References

Chittaro, L., Combi, C., and Trapasso, G. (2003). Data Mining on Temporal Data: A Visual Approach and its Clinical Application to Hemodialysis. *Journal of Visual Languages and Computing*, 14(6):591–620.

TimeHistogram 3D

data
frame of reference: **abstract**
variables: **univariate**

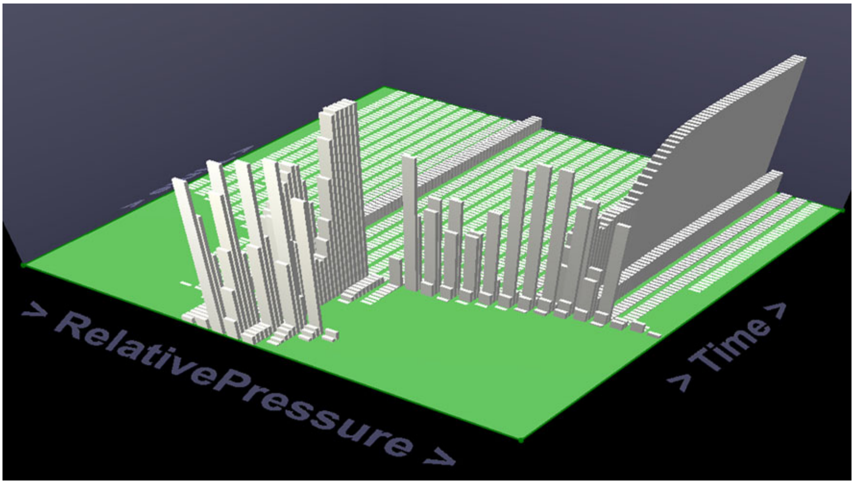


Fig. 7.12: Time is encoded along the x-axis, while a time-dependent variable (RelativePressure) is represented along the y-axis. For each time-value pair in the resulting grid in the x-y plane, the height of a cuboid represents the frequency of data items per grid cell.
Source: Kosara et al. (2004), © 2004 IEEE. Used with permission.

time
arrangement: **linear**
time primitives: **instant**

Kosara et al. (2004) proposed an interactive extension of well-known histograms called TimeHistogram 3D. The TimeHistogram is especially designed for time-oriented data. It has been developed to give an overview of complex data in the application context of computational fluid dynamics (CFD). A design goal of this technique was to show temporal information in static images while maintaining the easy readability of standard histograms. In the TimeHistograms in Figure 7.12, the x-axis encodes time and the y-axis encodes a time-dependent variable (RelativePressure), effectively creating a grid in the x-y plane, where each grid cell corresponds to a unique time-value pair. In order to visualize the number of data items (i.e., their frequency) per time-value pair, cuboids are shown for each cell, where the height of a cuboid encodes the frequency. This way, the user can see where in time and in which value range data items accumulate. Several interactive features such as brushing, scaling, and a 2D context display, which is shown in the background of the histogram, are part of this technique.

vis
mapping: **static**
dimensionality: **3D**

References

Kosara, R., Bendix, F., and Hauser, H. (2004). TimeHistograms for Large, Time-Dependent Data. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym)*, pages 45–54, Los Alamitos, CA, USA. IEEE Computer Society.

Intrusion Monitoring

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

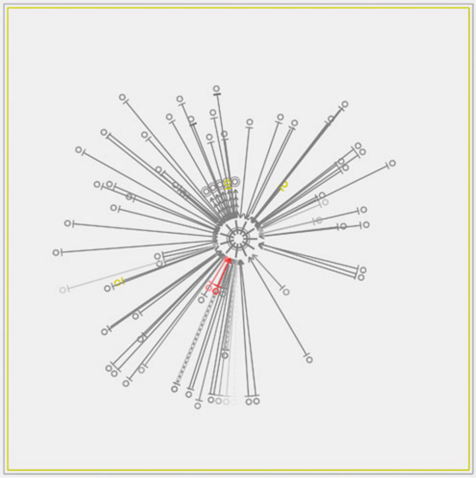


Fig. 7.13: The glyph in the center represents the monitored system. Connections to remote hosts are depicted as radially arranged lines. Critical and suspicious connections are visualized by red and yellow color, respectively.
Source: Image courtesy of Robert F. Erbacher.

Erbacher et al. (2002) describe a system that visualizes time-stamped network-related log messages that are dynamically generated by a monitored system. These messages correspond to events in a linear continuous time domain. The visualization shows the monitored server system as a central glyph encoding the number of users and the server's load (see Figure 7.13). Events are shown as radially arranged lines at whose end the remote host is shown as a smaller glyph. Regular network activities are drawn with a shade of gray. Unexpected or suspicious activities result in a change of color: Hosts that try to open privileged connections are colored in red, hosts that fail to respond turn yellow, lines representing timed-out connections or connections that failed the authentication procedure are shown in red, and connections that have been identified as intrusions are represented with even brighter red. To preserve a history of connections that have been terminated, the corresponding lines are faded out gradually. This kind of visual representation helps administrators in observing network communication. Presenting colored (red or yellow) lines among gray lines attracts the attention of administrators to suspicious activity and actions can be taken quickly to counter network attacks from remote hosts.

References

Erbacher, R. F., Walker, K. L., and Frincke, D. A. (2002). Intrusion and Misuse Detection in Large-Scale Systems. *IEEE Computer Graphics and Applications*, 22(1):38–48.

Anemone

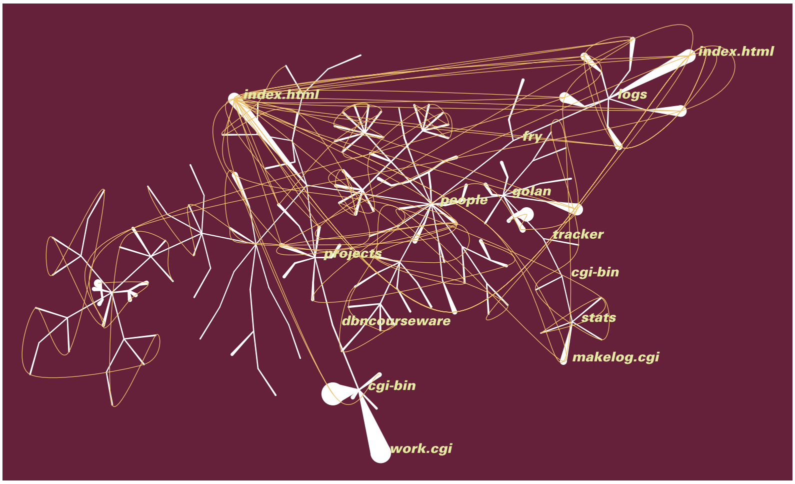


Fig. 7.14: Snapshot of Anemone showing traffic patterns of people visiting the web site of the Aesthetics & Computation Group at the M.I.T. Media Lab. The structure of the web site is shown as a node-link representation. Nodes vary in size, depending on how frequently a page is visited by users. Rarely-visited parts fade out slowly.
Source: Image courtesy of Ben Fry, MIT Media Laboratory, Aesthetics + Computation Group, © 1999-2005.

Anemone by Fry (2000) is a technique related to the visualization of structured information. It is a dynamic, organic representation designed to reveal not only the static structure of a website, which is based on its organization into folders and files, but also to reveal dynamic usage patterns. To this end, a classic node-link representation is visually enriched with dynamically updated usage statistics to form a living representation that truly reflects the restless nature of a website. The static structure is shown as nodes that are connected via straight branches. At the tip of a branch resides the actual web page. Additional labels can be used to identify nodes by their corresponding page's name. Nodes dynamically change size depending on how often they are visited by users. When a user follows a link from one page to another, a thin curved line is drawn connecting both pages. If parts of a site have not been visited for a long time, they shrink in size and slowly fade out. To allow users to concentrate on particular items of interest, it is possible to select nodes and to lock them to a dedicated position. This is quite useful, because the dynamic character of the technique implies that visual representation constantly changes its appearance.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

References

Fry, B. (2000). Organic Information Design, Master's thesis, Massachusetts Institute of Technology.

Timeline

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **interval**

vis

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

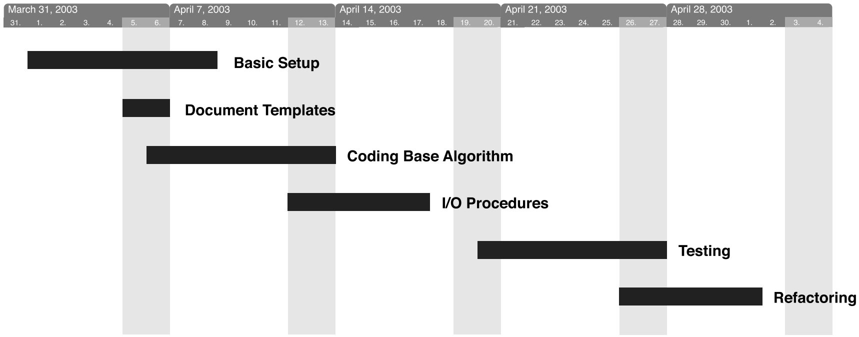


Fig. 7.15: Bars are arranged relative to a time axis to visualize both the location and duration of intervals. One can also see how intervals are related to each other.
Source: Generated by the authors.

If the time primitives of interest are not points but intervals, the visualization has to communicate not only where in time a primitive is located, but also how long it is. A simple and intuitive way of depicting incidents with a duration is by marking them visually along a time axis. This form of visualization is called timeline. Most commonly, a visual element such as a line or a bar represents an interval's starting point and duration (and consequently its end). Figure 7.15 shows an example with bars. If multiple intervals share a common time axis, as in this example, it is even possible to discern how the various intervals are related to each other (for possible relations of intervals see Section 3.1.2). Timelines are a very powerful visualization technique that, according to Tufte (1983), had been used long before computers even appeared (see also Chapter 2). Many different variants of timelines exist in diverse visualization tools. Additional interaction techniques often allow users not only to view time intervals, but also to create and edit them. Prominent examples are LifeLines (↪ p. 225) and Gantt charts (↪ p. 167).

References

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

Gantt Chart

data

frame of reference: **abstract**
variables: **univariate**

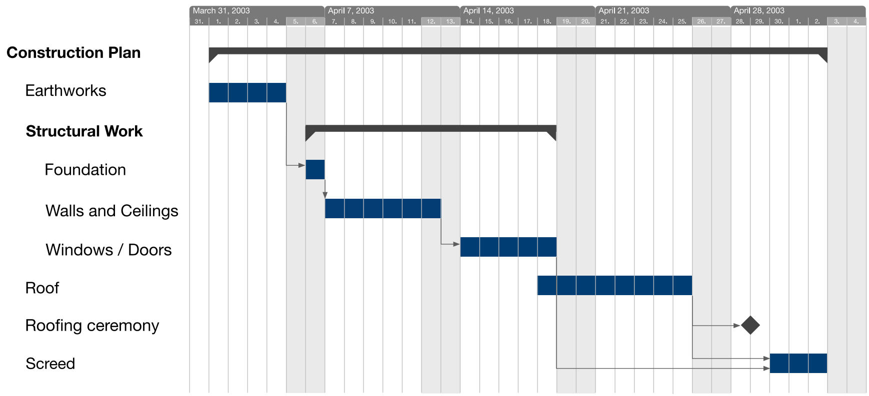


Fig. 7.16: The Gantt chart shows a project plan for construction works. To the left, the chart provides an indented list of tasks. In the main panel, timelines show position and duration of tasks in time, where black and blue bars stand for groups of tasks and individual tasks, respectively. Additionally, diamonds indicate milestones.
Source: Authors.

time

arrangement: **linear**
time primitives: **instant, interval**

Planning activities, people, and resources is a task that is particularly important in the field of project management. One of the common visualization techniques used for such tasks are Gantt charts. This kind of representation was originally invented by Gantt (1913) who studied the order of steps in work processes (see also Chapter 2, p. 25). Mainly work tasks with their temporal location and duration as well as milestones are depicted. The tasks are displayed as a textual list in the left part of the diagram and might be augmented by additional textual information such as resources, for example. Related tasks can be grouped to form a hierarchy, which is reflected by indentation in the task list. For displaying the position and duration of tasks in time, timelines (↪ p. 166) are drawn at the corresponding vertical position of the task list. This leads to an easily comprehensible representation of information from the past, present, and future. Hierarchically grouped tasks can be expanded and collapsed interactively. Summary lines are used to maintain an overview of larger plans. Sequence relationships are represented by arrows that connect tasks (e.g., an arrow from the end of task A to the beginning of task B shows that task B may start only after task A is finished). Milestones indicating important time points for synchronization within a project plan are visually represented by diamonds. The fact that tasks are mostly ordered chronologically, typically leads to a diagonal layout from the upper left to the lower right corner of the display.

vis

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

References

Gantt, H. L. (1913). *Work, Wages, and Profits*. Engineering Magazine Co., New York, NY, USA.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **3D**

Perspective Wall

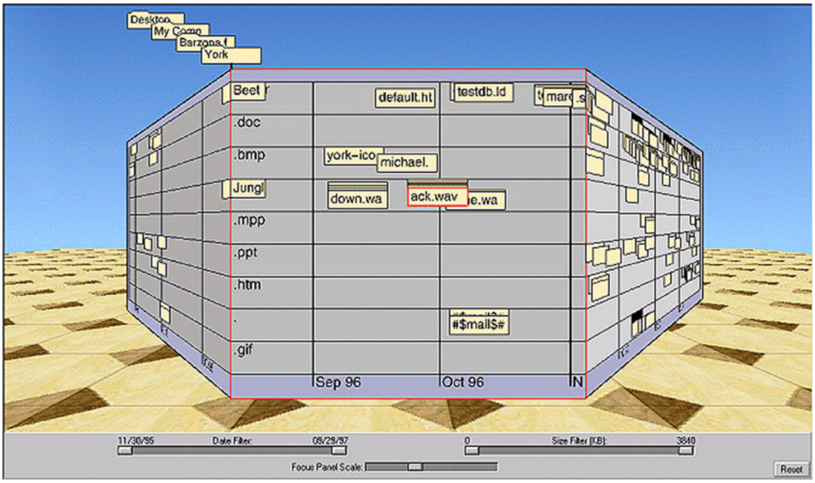


Fig. 7.17: A perspective wall representing time-related information of a file system for a period of several months. The focus (currently set to September/October 1996) shows detailed text labels for files, whereas the context regions only indicate files as yellow boxes.
Source: © Inxight Federal Systems.

Time-oriented data that are linked to a longer time axis (i.e., wide span in time or many time primitives) are usually difficult to represent visually because the image becomes very wide and exhibits an aspect ratio that is not suited for common displays. The perspective wall by Mackinlay et al. (1991) is a technique that addresses this problem by means of a focus+context approach. The key idea is to map time-oriented data to a 3D wall. For a user-selected focus, full detail is provided in the center of the display. Two context representations show the data in the past (to the left) and in the future (to the right) with regard to the current focus. The context is bent perspectively to reduce the display space occupied by these regions, effectively allowing for better space utilization in the focus (see center of Figure 7.17). Interaction methods are provided to enable users to navigate in time in order to bring different time spans into focus. The actual data representation on the wall may vary across applications; the only requirement is that time is mapped linearly from left to right. For example, one can use bars as in the figure or more advanced visual representations such as the ThemeRiver (↪ p. 197).

References

Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The Perspective Wall: Detail and Context Smoothly Integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 173–179, New York, NY, USA. ACM Press.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear**
time primitives: **interval**

vis

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

DateLens

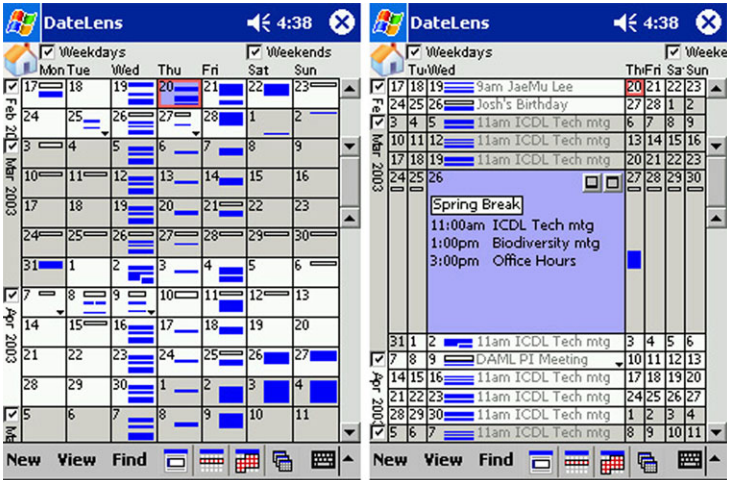


Fig. 7.18: A calendar grid shows the items of one's personal schedule as colored bars (left). Fisheye distortion is applied to show detailed textual information at the point the user is focusing on, and to maintain the context at less graphical detail (right).
Source: Images courtesy of Ben Bederson.

Most people use calendars to plan their daily life, for instance, to maintain a list of appointments or bookmark future events. Bederson et al. (2004) developed a tool to make it easier to work with a personal schedule on small handheld devices. Because display space is limited on such devices (compared to common desktop displays), focus+context mechanisms are applied to present temporal information at different levels of detail. Based on a common tabular representation of a calendar (see Figure 7.18, left), the DateLens magnifies table cells so as to provide more display space to important information that is currently in the user's focus (see Figure 7.18, right). The fisheye distortion magnifies the focus and reduces graphical detail in the context of the display. If sufficient display space is available, calendar entries are shown in textual form. Otherwise, temporal intervals of calendar entries are indicated by bars that visualize the starting point and temporal extent of appointments and events stored in the calendar. Various interaction mechanisms allow users to view the calendar at different temporal granularities and to navigate forward and backward in time.

References

Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. (2004). DateLens: A Fisheye Calendar Interface for PDAs. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(1):90–119.

data

frame of reference: **abstract**
variables: **univariate**

TimeNets

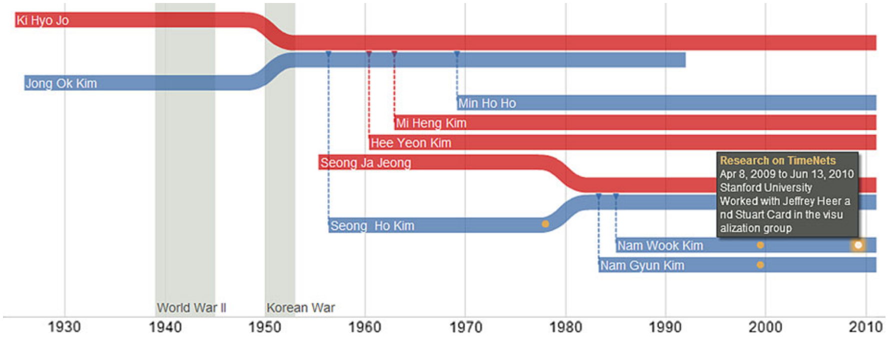


Fig. 7.19: TimeNets visualize temporal and structural aspects of genealogical data. Bands that extend along the horizontal time axis visualize individuals. Marriage and divorce are indicated by converging and diverging bands, respectively. Children are connected to their parents via drop lines. Labels are shown for the persons' names as well as for historical and personal events.
Source: Image courtesy of Jeffrey Heer and Nam Wook Kim.

time

arrangement: **linear**
time primitives: **interval**

Genealogical data are an interesting source of time-oriented information. In such data, not only family structures are of interest, but also temporal relationships. Kim et al. (2010) propose the TimeNets approach, which aims to visualize both of these aspects. TimeNets represent persons as individual bands that extend horizontally along a time axis from left to right. Each band shows a label of the person's name and different colors are used to encode sex: red is reserved for females, and males are shown in blue. Marriage of persons is visualized by converging the corresponding bands, while divorce is indicated by diverging bands. When a child is born, a new band is added to the display. A so-called drop line connects the band of the child to the parents' bands to convey the parent-child relationship. In order to allow users to focus on relevant parts of the data, a degree of interest (DOI) algorithm is applied. Bands below the DOI threshold are filtered out or smoothly fade in where they are linked to bands of relevant persons. Users can select multiple persons to focus on. On each change of the focus, the visualization shows a smooth transition of the display to keep users oriented.

vis

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

References

Kim, N. W., Card, S. K., and Heer, J. (2010). Tracing Genealogical Data with TimeNets. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 241–248, New York, NY, USA. ACM Press.

Paint Strips

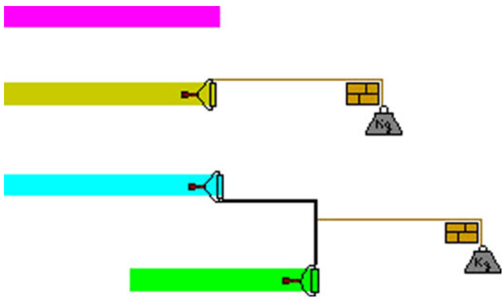


Fig. 7.20: Paint strips indicate the location and duration of time intervals, effectively allowing users to assess relationships of intervals. Temporal indeterminacy of intervals is indicated by paint rollers that can move flexibly within certain constraints, which are represented by wall elements. *Source: Image courtesy of Luca Chittaro.*

Chittaro and Combi (2003) designed paint strips to represent relations between time intervals for visualizing queries on medical databases. The technique is strongly related to timelines (↪ p. 166), but here paint strips are used as equivalents of bars to indicate time intervals, and optionally, the indeterminacy of intervals is communicated by placing paint rollers at either end of the paint strips. A paint roller with a weight attached to it means this interval can possibly extend in time. Graphical depictions of wall elements represent constraints on the extension. This way, the maximum duration and earliest start or latest end of intervals are defined, depending on which end of the painting strip the paint rollers are attached to. It is also possible to link strips, which means if one strip moves, the other one moves to the same extent as well. This relationship is indicated graphically by connecting the involved paint rollers before attaching them to the rope that holds the weight (see bottom of Figure 7.20). Paint strips were especially developed for medical applications but can be used anywhere where indeterminate time intervals have to be visualized. Thanks to the simplicity of the paint strip metaphor, there is room for application-dependent enhancements, such as textual annotations for start and end points as well as for durations of intervals.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear**
time primitives: **interval**

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

References

Chittaro, L. and Combi, C. (2003). Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation. *Data and Knowledge Engineering*, 44(2):239–264.

data
frame of reference: **abstract**
variables: **univariate**
time
arrangement: **linear**
time primitives: **interval**
vis
mapping: **static**
dimensionality: **2D**

PlanningLines

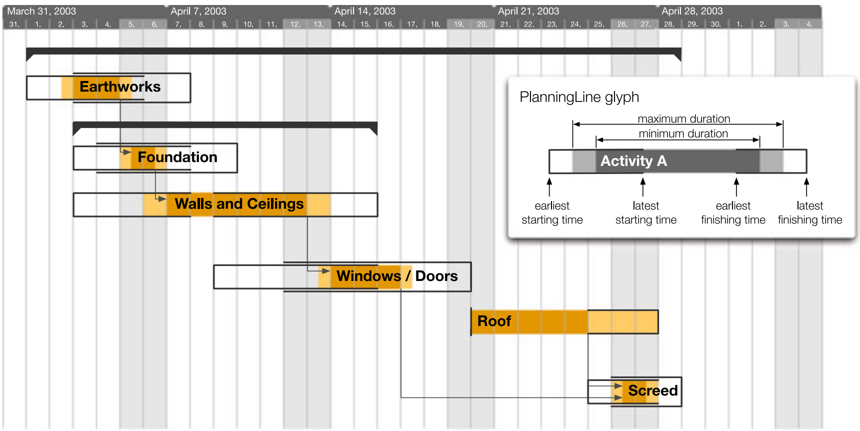


Fig. 7.21: Project plan of construction works that represents temporal uncertainties via PlanningLines. A PlanningLine glyph consists of two encapsulated bars, which represent minimum and maximum duration. The bars are bounded by two caps encoding the start and end intervals.
Source: Adapted from Aigner et al. (2005).

Since the future is always inherently connected with possible uncertainties, delays, and the unforeseen, these issues need to be dealt with in many domains like project management or medical treatment planning. PlanningLines by Aigner et al. (2005) allow the representation of temporal uncertainties, thus supporting project managers in their difficult planning and controlling tasks. PlanningLines have been designed to be easily integrated into well-known timeline-based visualization techniques such as Gantt charts (↪ p. 167). A single glyph (see glyph explanation in Figure 7.21) provides a visual representation of the temporal indeterminacies of a single activity, facilitates the identification of (un)defined attributes, supports in maintaining logical constraints (e.g., bars may not extend caps), and gives a visual impression of the individual and overall uncertainties. Uncertainties might be introduced by explicit specifications, usually connected with future planning (e.g., "The meeting will start between 12 p.m. and 2 p.m.." – which might be any point in time between noon and 2 p.m.) or is implicitly present in cases where data are given with respect to multiple temporal granularities (e.g., data given on a granularity of days and shown on an hourly scale).

References

Aigner, W., Miksch, S., Thurnher, B., and Biffl, S. (2005). PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 457–463, Los Alamitos, CA, USA. IEEE Computer Society.

Time Annotation Glyph

data

frame of reference: **abstract**
variables: **univariate**

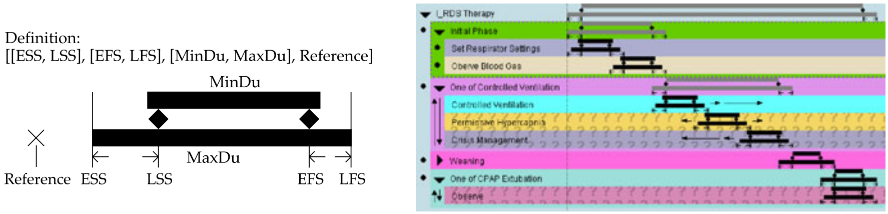


Fig. 7.22: The time annotation glyph was designed to represent the temporal constraints of medical treatment plans. It uses the metaphor of bars that lie on pillars. Left: Single glyph and associated parameters. Right: Usage in a tool for representing the temporal and hierarchical aspects of a medical treatment plan as well as the execution order of individual parts.
Source: Images courtesy of Robert Kosara.

time

arrangement: **linear**
time primitives: **interval**

The time annotation glyph by Kosara and Miksch (2001) uses the simple metaphor of bars that lie on pillars to represent a complex set of time attributes. Four vertical lines on the base specify the earliest and the latest starting and ending times. Supported by these pillars lies a bar that is as long as the maximum duration. On top of the maximum duration bar, a bar that represents the minimum duration lies upon two diamonds indicating the latest start and the earliest end. Furthermore, undefined parts and different granularities are indicated visually. Because of this metaphor, a few simple time parameter constraints can be understood intuitively. For example, the minimum duration cannot be shorter than the interval between latest start and earliest end – if it was, the minimum duration bar would fall down between its supports. All of the parameters might be defined relative to a reference point that is also represented graphically. In summary, the following parameters are shown: earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS), latest finishing shift (LFS), minimum duration (MinDu), and maximum duration (MaxDu). The technique is used to represent the time annotations of medical treatment plans within the AsbruView application (see Figure 7.22, right) as described in Kosara and Miksch (2001).

vis

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

References

Kosara, R. and Miksch, S. (2001). Metaphors of Movement - A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131.

data
frame of reference: **abstract**
variables: **univariate**
time
arrangement: **linear**
time primitives: **interval**
vis
mapping: **static**
dimensionality: **2D**

SOPO Diagram

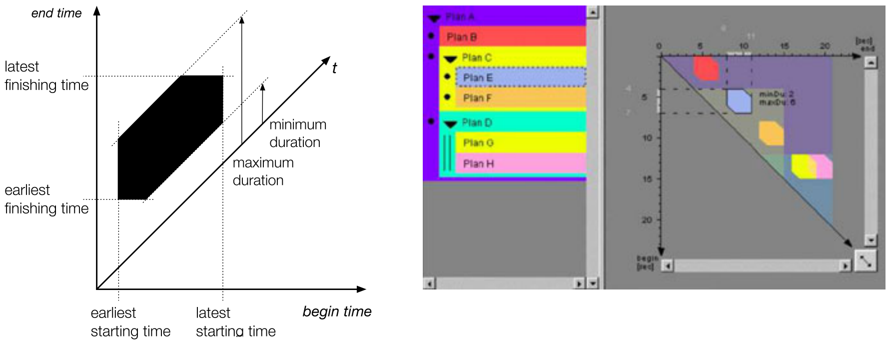


Fig. 7.23: A SOPO diagram shows the possible configurations of the begin and end times of an event via a constrained polygonal shape. Right: SOPOView – an interactive visualization tool for working with SOPOs applied for medical treatment plans.
Source: Images courtesy of Robert Kosara.

For planning and scheduling, the temporal extents of events can be characterized by sets of possible occurrences (SOPOs), i.e., a set of possible begin and end times during which an event may happen. Rit (1986) defined a theoretical model for the definition and propagation of temporal constraints for scheduling problems. A graphical representation of SOPOs was introduced as a visual aid for understanding and solving such problems. In this representation, the extent of temporal uncertainty is expressed via a polygonal shape. The axes of a SOPO diagram represent begin time (x-axis) and end time (y-axis). Points in this diagram do not represent points in time, but complete intervals specified by their begin (x-coordinate) and end time (y-coordinate). Hence, the extent of an interval is represented by its position, not its visual extent. The area an item covers reflects all intervals that fit the specification given by means of earliest start, latest start, earliest end, latest end, minimum, and maximum duration (see Figure 7.23, left). Moreover, the exact occurrence of an event may be constrained by other related events which further modify the sets of possible occurrences. The propagation of such constraints is aided graphically, e.g., via overlaps of individual SOPOs. Later, this idea was interactively enhanced and further developed to be applied for visualizing medical treatment plans in the tool SOPOView (see Figure 7.23, right and Kosara and Miksch, 2002).

References

Kosara, R. and Miksch, S. (2002). Visualization Methods for Data Analysis and Planning. *International Journal of Medical Informatics*, 68(1–3):141–153.

Rit, J.-F. (1986). Propagating Temporal Constraints for Scheduling. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 383–388, Los Altos, CA, USA. Morgan Kaufmann.

Silhouette Graph, Circular Silhouette Graph

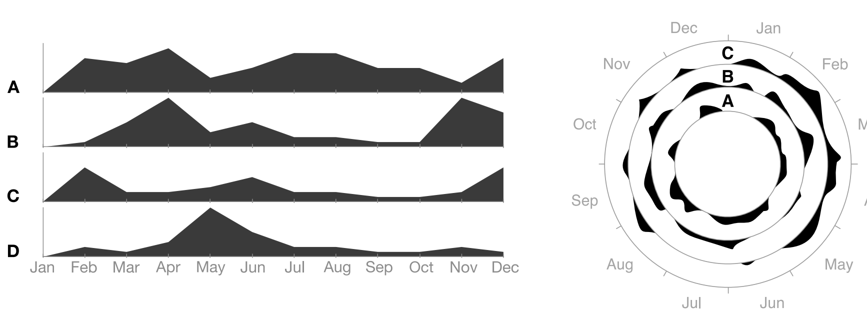


Fig. 7.24: Silhouette graphs are filled line plots that can be used for enhancing the comparison of multiple time-series that are put side-by-side.
Source: Adapted from Harris (1999).

Silhouette graphs emphasize the visual impression of time-series by filling the area below the plotted lines (see Harris, 1999). This leads to distinct silhouettes that enhance perception at wide aspect ratios of long time-series compared to line plots (↪ p. 153) and allow an easier comparison of multiple time-series. On the left of Figure 7.24, time is mapped to the horizontal axes and multiple time-series are stacked upon each other. Other layouts of the axes might be used for reflecting different time characteristics. One example are circular silhouette graphs (see Figure 7.24, right) that represent silhouette graphs on concentric circles in order to emphasize periodicities in time.

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

data
frame of reference: **abstract**
variables: **univariate**

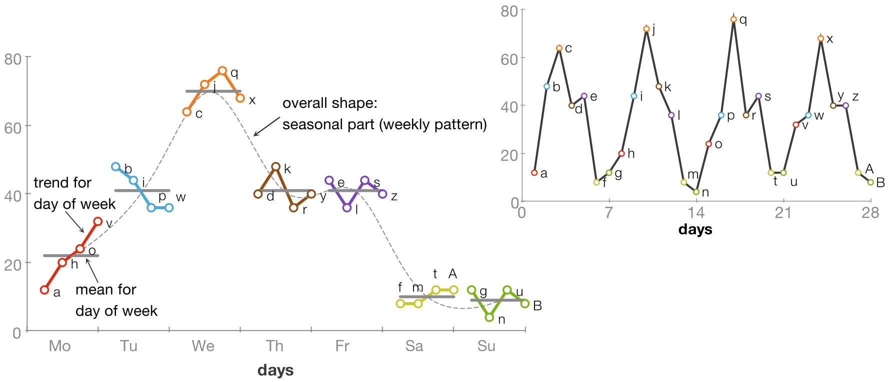
time
arrangement: **linear, cyclic**
time primitives: **instant**

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

Cycle Plot

data

frame of reference: **abstract**
variables: **univariate**



time

arrangement: **linear, cyclic**
time primitives: **instant**

Fig. 7.25: Both seasonal and trend components of a time-series can be discerned from cycle plots (left), which is hardly possible when using line plots (right).
Source: Adapted from Cleveland (1993) with permission of William Cleveland.

Time-series data may contain a seasonal as well as a trend component, which is also reflected in many statistical models. Cleveland (1993) describes cycle plots as a technique to make seasonal and trend components visually discernable. This is achieved by showing individual trends as line plots embedded within a plot that shows the seasonal pattern. For constructing a cycle plot, one has to define the time primitives to be considered for the seasonal component. The horizontal axis of the cycle plot is then subdivided accordingly. Figure 7.25 demonstrates this using weekdays. We are interested in the trend for each weekday and the general weekly pattern. The data for a particular weekday are visualized as a separate line plot (e.g., data of the 1st, 2nd, 3rd, and 4th Monday). This allows the identification of individual trends for each day of the week. In the figure, we see an increasing trend for Mondays, but a decreasing trend for Tuesdays. Additionally, the cycle plot shows the mean value for each weekday (depicted as gray lines). Connecting the mean values as a line plot (dashed line in the figure) reveals the seasonal pattern, which in this case is a weekly pattern that clearly shows a peak on Wednesday.

vis

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

References

Cleveland, W. (1993). *Visualizing Data*. Hobart Press, Summit, NJ, USA.

Cluster and Calendar-Based Visualization

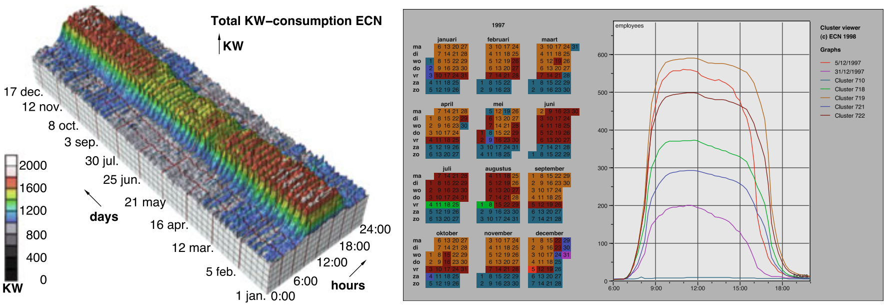


Fig. 7.26: The 3D visualization on the left represents daily power consumption patterns for several weeks of a year. The calendar representation on the right shows the cluster affiliation of daily patterns and allows users to discern days with uncommon patterns, in this case office hours of employees.
Source: Van Wijk and Van Selow (1999), © 1999 IEEE. Used with permission.

Temporal patterns can indicate at which time of the day certain resources are highly stressed. Relevant applications can be found in computing centers, traffic networks, or power supply networks. An approach that allows for finding temporal patterns at different temporal granularities has been proposed by Van Wijk and Van Selow (1999). The starting point of the approach is to consider the course of a day as a line plot (↪ p. 153) covering the 24 hours of a day. Multiple daily courses of this kind are visualized as a three-dimensional height field (see left part of Figure 7.26), where the hours of the day are encoded along one axis, individual days are encoded along the second axis, and data values are encoded as height (along the third axis). This allows users to detect short term daily patterns and long term trends of the data at higher temporal granularity. To assist in the analysis of the data, Van Wijk and Van Selow further suggest grouping similar daily courses into clusters (see also Section 6.2, p. 130). The data belonging to a particular cluster are aggregated to define the representative for that cluster, i.e., the representative again forms a daily course. Cluster affiliation of individual dates is then color-coded into a calendar as depicted in the right part of Figure 7.26. The user can adjust the number of clusters to be shown so as to find the level of abstraction that suits the data and the task at hand. The combination of analytical and visual methods as applied here is useful for identifying days of common and exceptional daily behavior.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **2D, 3D**

References

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, Los Alamitos, CA, USA. IEEE Computer Society.

Tile Maps

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

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

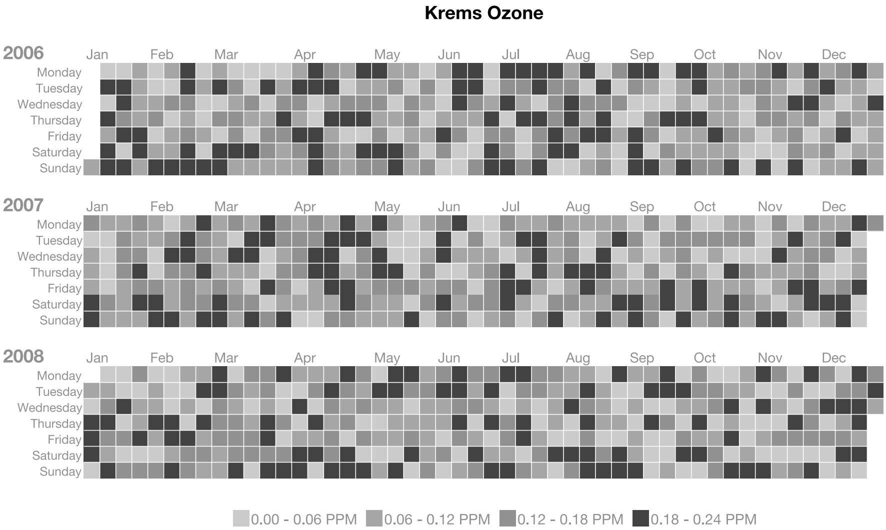


Fig. 7.27: The matrices show ozone measurements made in the course of three years, where data values for individual days are encoded to the brightness of matrix cells.
Source: Adapted from Mintz et al. (1997) with permission of David Mintz.

Tile maps as described by Mintz et al. (1997) represent a series of data values along a calendar division. The idea behind this technique is to arrange data values according to different temporal granularities. For example, data values measured on a daily basis are displayed in a matrix where each cell (or tile) corresponds to a distinct day, a column represents a week, and a row represents all data values for a particular weekday (see Figure 7.27). One additional level of granularity can be integrated by stacking multiple matrices as shown in the figure. Data values are visualized by varying the lightness of individual tiles. A visual representation constructed this way can be interpreted quite easily, because it corresponds to our experience of looking at calendars. The arrangement as a two-dimensional matrix allows users to identify long-term trends by considering the matrix as a whole, to discern individual trends for Mondays, Tuesdays, and so forth by looking at the matrix rows, and to derive weekly patterns by investigating matrix columns. For example, the U.S. Environmental Protection Agency provides a web tool that generates tile maps automatically for specified air pollutants and locations.

References

Mintz, D., Fitz-Simons, T., and Wayland, M. (1997). Tracking Air Quality Trends with SAS/GRAPH. In *Proceedings of the 22nd Annual SAS User Group International Conference (SUGI97)*, pages 807–812, Cary, NC, USA. SAS.

Multi Scale Temporal Behavior

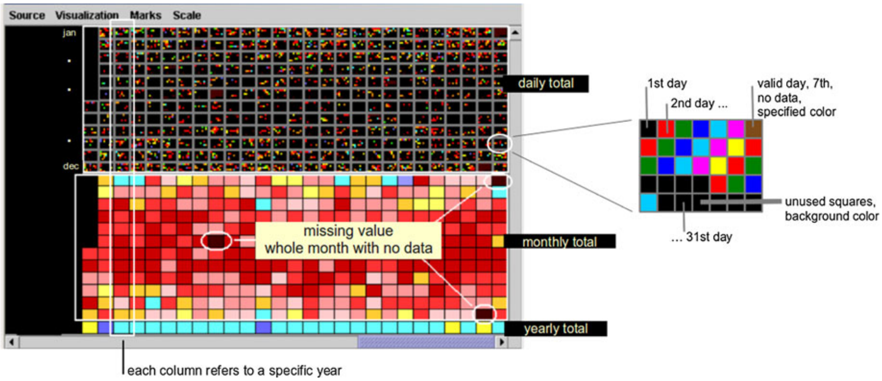


Fig. 7.28: Different levels of granularity are shown simultaneously in one display to allow users to explore patterns in precipitation data from Brazil at different temporal levels.
Source: Image courtesy of Milton Hirokazu Shimabukuro.

The Multi Scale Temporal Behavior technique by Shimabukuro et al. (2004) comprises different levels of granularity and aggregation to explore patterns at different temporal levels. The basis for the visualization is a matrix that is divided vertically into three regions, one for each of the three scale levels: daily data, monthly data, and yearly data (see Figure 7.28, top to bottom). Each column of the matrix represents a year worth of data. The cells in the topmost region represent months. They show full detail by color-coding individual pixels within a cell according to daily values. The middle region shows aggregated data. Here cells are no longer subdivided into pixels, but are colored uniformly, where the color represents the aggregation of daily values to a single monthly value. The same principle is applied for the bottom region (in fact, the bottom row). Twelve monthly values are aggregated into a single value for the year, which can again be represented by color. A significant and non-trivial problem in dealing with real world datasets are missing data values. This issue is tackled by the authors by preprocessing the data and marking missing values as such in the display.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

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

References

Shimabukuro, M., Flores, E., de Oliveira, M., and Levkowitz, H. (2004). Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pages 107–117, Los Alamitos, CA, USA. IEEE Computer Society.

Recursive Pattern

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

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

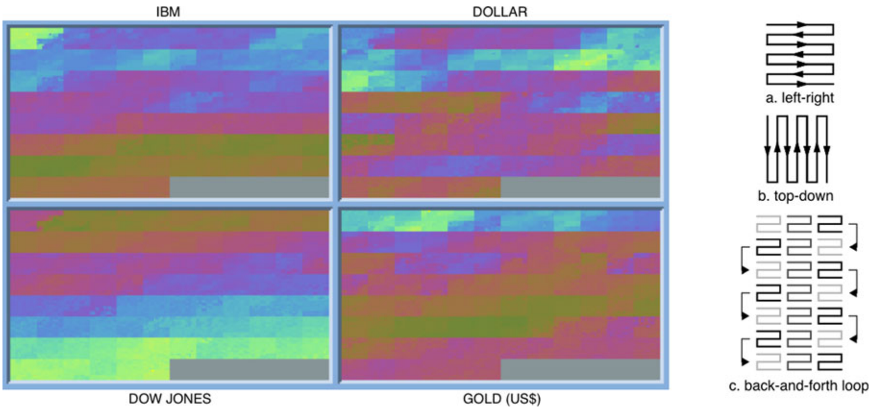


Fig. 7.29: Four time-series are shown as recursive patterns, where the color of a pixel represents stock price. Pixels are arranged to semantically match the hierarchical structure of the data: A row represents a year with its 12 months, each month is further subdivided into weeks, which in turn consist of five workdays, each of which represents nine data values for a single day. Right: Examples of possible pixel arrangements.
Source: Keim et al. (1995), © 1995 IEEE. Used with permission.

The most space-efficient way of visualizing data is to represent them on a per-pixel basis. Keim et al. (1995) suggest a variety of pixel-based visualization approaches of which the recursive pattern technique is particularly suited to display large time-series. The key idea behind the recursive pattern technique is to construct an arrangement of pixels that corresponds to the inherently hierarchical structure of time-oriented data given at multiple granularities. Figure 7.29 shows financial data as a pixel-based visualization. The initial step is to map nine data values collected per day to a 3x3 pixel group. This group is then used to form a larger group for a week of workdays containing 5x1 day groups. Recursively, groups for months, years, and decades can be created by arranging groups of the next lower granularity in a semantically meaningful way (e.g., 12 months are grouped into a year). In the resulting pattern, each pixel is color-coded with regard to a single data value in the time-series. Multiple dense pixel displays of this kind can be combined to get an overview of large multivariate datasets.

References

Keim, D., Kriegel, H.-P., and Ankerst, M. (1995). Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In *Proceedings of IEEE Visualization (Vis)*, pages 279–286, Los Alamitos, CA, USA. IEEE Computer Society.

GROOVE

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

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

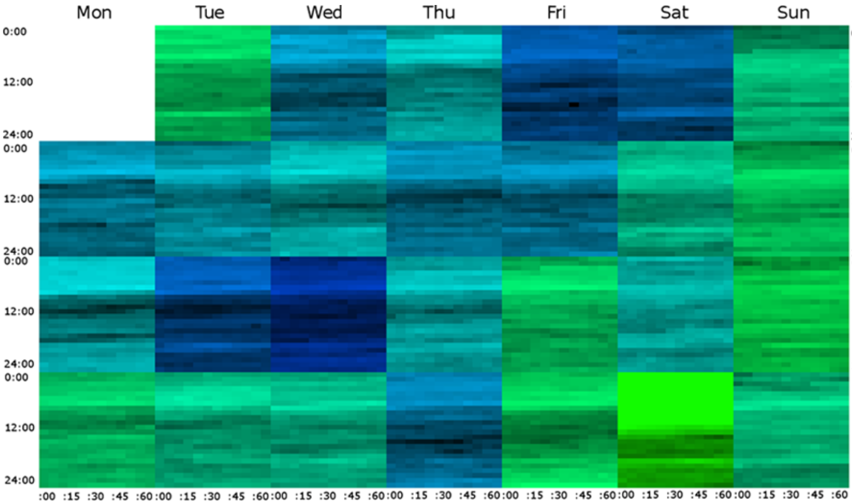


Fig. 7.30: GROOVE visualizations combine detail and overview readings and use regular layouts based on time granularities. Here, police assignments are shown over the course of one month. For each 5-minute interval the number of units assigned to a duty is given. Each block shows a day, rows of blocks represent weeks, columns of blocks show the day of the week, and within each block a row of pixels displays one hour. Color components are used to show detail values for pixels (lightness – bright: low, dark: high) and average values per block (hue – green: low, blue: high). *Source: Generated with the GROOVE software.*

GROOVE (Granularity Overview OVErlay) visualizations as presented by Lammarsch et al. (2009) utilize a user-configurable set of four time granularities to partition a dataset in a regular manner. That is, a recursive layout is achieved that shows columns and rows of larger blocks and a pixel arrangement within blocks for the detail structure. Following the concept of recursive patterns (↪ p. 180) different arrangements might be chosen (e.g., row-by-row or back-and-forth). The specific of GROOVE visualizations is the combination of overview (aggregated values) and details in one place using one of three kinds of overlays. This allows micro and macro readings and avoids eye movements between the overview and detail representations. First, color components can be employed with color-based overlay (see Figure 7.30). Second, opacity overlay applies interactive crossfading between the overview and the detail display. Third, spatial overlay can be used for viewing the data selectively at different levels of aggregation by expanding and collapsing areas.

References

Lammarsch, T., Aigner, W., Bertone, A., Gärtner, J., Mayr, E., Miksch, S., and Smuc, M. (2009). Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 44–49, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **univariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

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

SolarPlot

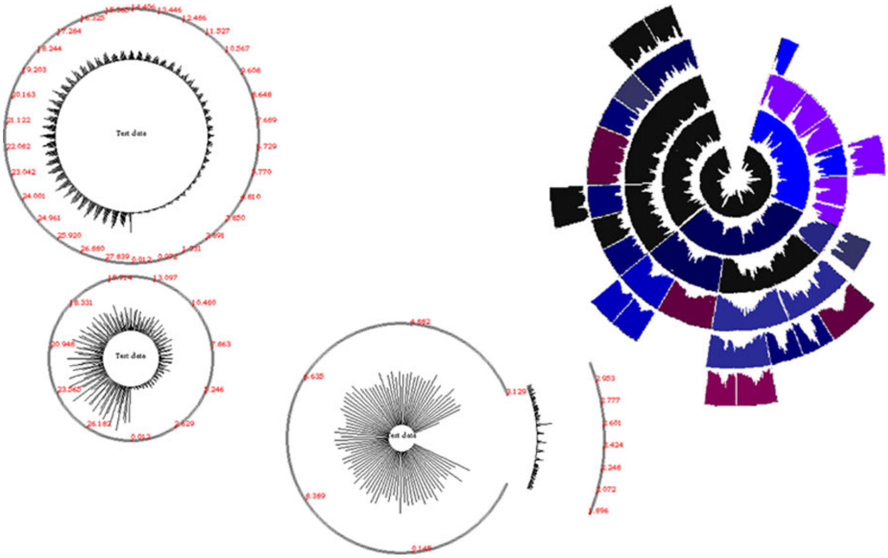


Fig. 7.31: Daily values of ticket sales data over a period of 30 years are plotted around the circumference of a resizable circle (left). For a selected interval, a more detailed arc can be shown (center). The technique can be enhanced so as to combine the representation of data and a hierarchical structure, in this case email traffic and company organization (right).
Source: Chuah (1998), © 1998 IEEE. Used with permission.

With the SolarPlot technique introduced by Chuah (1998), values are plotted around the circumference of a circle as shown left in Figure 7.31. Much like in a circular histogram, the first step is to partition the data series into a number of bins. Each bin is represented by a sunbeam whose length encodes the frequency of data items in the corresponding bin. The SolarPlot determines the number of bins dynamically depending on the size of the circle. Users are allowed to expand or contract the circle in order to get more or fewer bins, or in other words, to get a more or less detailed representation of the data. This way it is possible to explore the data at different levels of abstractions and to discern patterns globally across aggregation levels. The SolarPlot also supports locally switching to a more detailed plot for a user-selected focus interval as shown in the center of Figure 7.31. Chuah (1998) further suggests a variation of the SolarPlot called SolarPlot + Aggregate TreeMap, where the display of data is combined with a visual representation of a hierarchical structure (see Figure 7.31, right).

References

Chuah, M. C. (1998). Dynamic Aggregation with Circular Visual Designs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 35–43, Los Alamitos, CA, USA. IEEE Computer Society.

SpiraClock

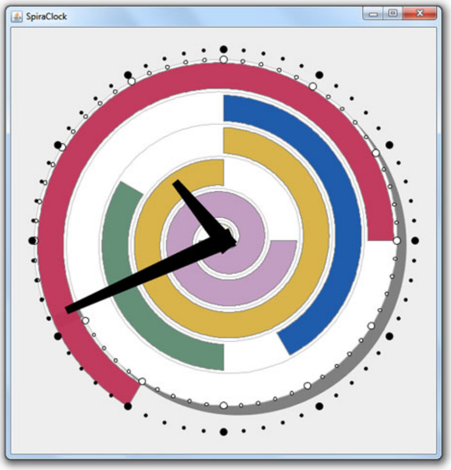


Fig. 7.32: The SpiraClock builds upon the clock display. The minute hand currently points to a meeting that has already started. Future appointments are aligned along a spiral on the clock face. *Source: Adapted from Dragicevic and Huot (2002) with permission of Pierre Dragicevic.*

The SpiraClock invented by Dragicevic and Huot (2002) visualizes time by using the clock metaphor. The visual representation consists of a clock face and two hands indicating hour and minute. The interior of the clock shows a spiral that extends from the clock's circumference toward its center. Each cycle of the spiral represents 12 hours, with the current hour shown at the outermost cycle and future hours displayed in the center (about nine future hours in Figure 7.32). Time intervals (e.g., meetings) are represented as thick segments along the spiral shape. These segments show when intervals start and end. Users can also see if certain appointments are in conflict because they *overlap*, or if the agenda is too tight, because many appointments *meet* (for further interval relations see Section 3.1.2). As time advances, the spiral is constantly updated and future intervals gradually move outward until they are current. Past intervals gradually fade out. In this sense, the SpiraClock enhances classic clocks with a preview of the near future and a brief view to the past. The SpiraClock allows users to drag the clock handles to visit different points in time, and intervals of interest can be highlighted and corresponding textual annotations can be displayed.

References

Dragicevic, P. and Huot, S. (2002). SpiraClock: A Continuous and Non-Intrusive Display for Upcoming Events. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 604–605, New York, NY, USA. ACM Press. Extended Abstracts.

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **cyclic**
time primitives: **interval**

vis
mapping: **dynamic**
dimensionality: **2D**

Enhanced Interactive Spiral

data
frame of reference: **abstract**
variables: **univariate**

time
arrangement: **cyclic**
time primitives: **instant**

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

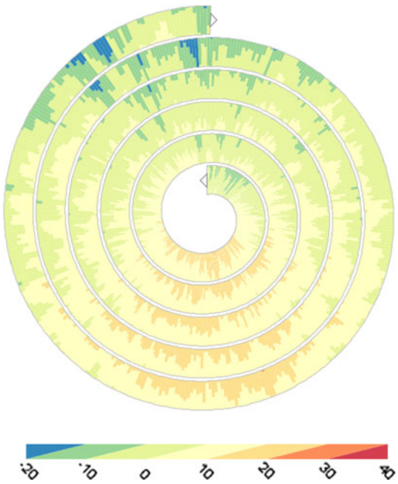


Fig. 7.33: The spiral shows the daily temperature measured in Rostock from 2006 to 2010. The blue and green colors in the upper left part of the spiral represent the colder winters of 2009 and 2010.
Source: Generated with the enhanced interactive spiral display tool.

Tominski and Schumann (2008) apply the enhanced two-tone color-coding by Saito et al. (2005) to visualize time-dependent data along a spiral. Each time primitive is mapped to a unique segment of the spiral. Every segment is subdivided into two parts that are colored according to the two-tone coloring method. The advantage of using the two-tone approach is that it realizes the overview+detail concept by design. The two colors used per spiral segment allow users to quickly recognize the value range of that segment (overview). If the value range is of interest, the proportion of the two colors indicates the particular data value more precisely (detail). The enhanced spiral can be adjusted interactively in various ways. The number of time primitives, the number of cycles, and additional geometrical parameters influence the shape of the spiral and thus the mapping of the time domain. The data representation is mainly controlled by the color scales applied and parameters such as the number of colors, the direction of the mapping, and the mapping function (linear vs. logarithmic). Navigation in time is possible via direct manipulation of the spiral.

References

Saito, T., Miyamura, H., Yamamoto, M., Saito, H., Hoshiya, Y., and Kaseda, T. (2005). Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 173–180, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C. and Schumann, H. (2008). Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity*, pages 53–56, Linköping, Sweden. Linköping University Electronic Press.

Spiral Graph

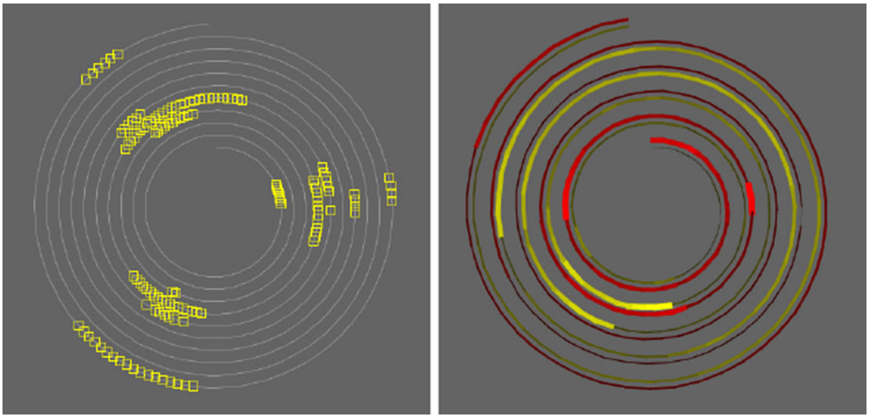


Fig. 7.34: A spiral graph encodes time-series data along a spiral, thus emphasizing the cyclic character of the time domain. Actual data values are visualized using symbols for nominal data (left) as well as color and line thickness for quantitative data (right).
Source: Weber et al. (2001), © 2001 IEEE. Used with permission.

The spiral graph developed by Weber et al. (2001) is a visualization technique that focuses on cyclic characteristics of time-oriented data. To this end, the time axis is represented by a spiral. Time-oriented data are then mapped along the spiral path. While nominal data are represented by simple icons (see Figure 7.34, left), quantitative data can be visualized by color, line thickness, or texture. One can also visualize multivariate time-series by intertwining several spirals as shown for two variables in the right part of Figure 7.34. In this case, a distinct hue is used per spiral so that individual variables can be discerned. Weber et al. (2001) further envision extending the spiral to a three-dimensional helix, in order to cope with larger time-series. The main purpose of the spiral graph is the detection of previously unknown periodic behavior of the data. The user can interactively adjust the spiral's cycle length (i.e., the number of data values mapped per spiral cycle) to explore the data for cyclic patterns. As an alternative, it is also possible to smoothly animate through possible cycle lengths. In this case, periodic behavior of the data becomes immediately apparent by the emergence of a pattern. When such a pattern is spotted, the user stops the animation and an interesting cycle length has been found (see also Section 4.2.1, p. 84).

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **cyclic**
time primitives: **instant**

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

References

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 7–14, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D, 3D**

Spiral Display

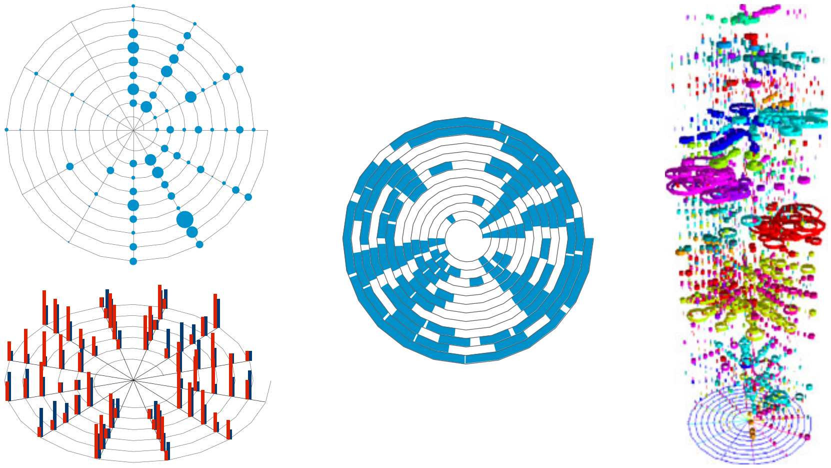


Fig. 7.35: Data can be visualized along a spiral in different ways: by the area of circular elements (top-left), by the sizes of multiple spikes (bottom-left), as bars marking start and end of intervals (center), or by the volume of hollow cans aligned at different layers along the vertical axis (right). *Source: Carlis and Konstan (1998), © 1998 ACM. Used with permission.*

The interactive spiral display by Carlis and Konstan (1998) uses Archimedean spirals to represent the time domain. Data values at particular time points are visualized as filled circular elements whose area is proportional to the data value (see Figure 7.35, top-left). In the case of interval-based data, filled bars are aligned with the spiral shape to indicate start and end of intervals (see Figure 7.35, center). If multivariate data are given at time points, the spiral is tilted and data values are visualized as differently colored spikes, where spike color indicates variable affiliation and spike height encodes the corresponding data value (see Figure 7.35, bottom-left). Alternatively, one can use the vertical z-axis to separate the display of multiple variables (see Figure 7.35, right). In this case, each time-dependent variable has its own layer along the z-axis and is represented with a unique color. Within a layer, data values are encoded to the volume of cans, which are lidless and hollow to prevent occlusion. The system implemented by Carlis and Konstan (1998) allows users to display multiple linked spirals to perform comparison tasks. The cycle lengths of spirals can be adjusted interactively and can also be animated automatically for discovering periodic patterns.

References

Carlis, J. V. and Konstan, J. A. (1998). Interactive Visualization of Serial Periodic Data. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 29–38, New York, NY, USA. ACM Press.

VizTree

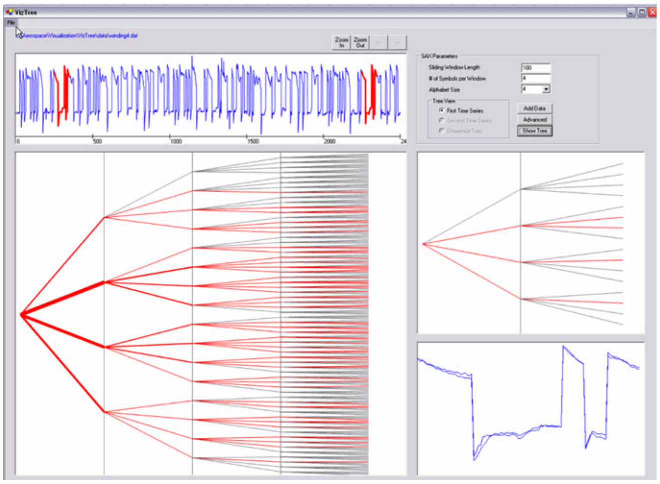


Fig. 7.36: Time-series of arbitrary length are represented as fixed-length subsequence trees. The figure shows a winding dataset that records the angular speed of a reel. Top-left: input time-series; top-right: parameter setting area for discretization and subsequence length; left: subsequence tree for the time-series; center-right: detail view of the tree shown in the left panel; lower-right: subsequences matching a particular string representation (e.g., subsequences starting with ‘ab’) whereas positions of matched subsequences are highlighted in the top-left panel.
Source: Image courtesy of Eamonn Keogh.

VizTree by Lin et al. (2005) is a time-series pattern discovery and visualization system for massive time-series datasets. It uses the time-series discretization method SAX (symbolic aggregate approximation) developed earlier by Lin et al. (2007). SAX discretizes time-series into a sequence of symbols (e.g., ‘abacacc’). Subsequences (patterns) are generated by moving a sliding window along the sequence. These subsequences are combined and represented by a horizontal tree visualization where the frequency of a pattern is encoded by the thickness of a branch (or light gray when frequency is zero). The VizTree interface consists of multiple coordinated views that show the input time-series along with the subsequence tree as well as control and detail-on-demand panels. VizTree can be used to accomplish different pattern discovery tasks interactively: finding frequently occurring patterns (i.e., motif discovery) and surprising patterns (i.e., anomaly detection), query by content, and the comparison of two time-series by calculating a difference tree.

data
frame of reference: **abstract**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Lin, J., Keogh, E. J., and Lonardi, S. (2005). Visualizing and Discovering Non-Trivial Patterns in Large Time Series Databases. *Information Visualization*, 4(2):61–82.

Lin, J., Keogh, E. J., Wei, L., and Lonardi, S. (2007). Experiencing SAX: A Novel Symbolic Representation of Time Series. *Data Mining and Knowledge Discovery*, 15(2):107–144.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

TimeSearcher

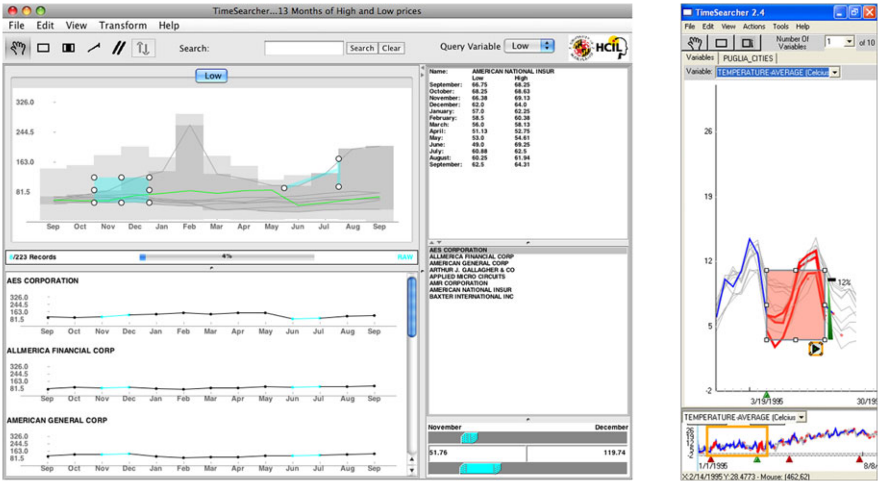


Fig. 7.37: Left: TimeSearcher 1 showing stock price data (top-left: Multiple line plots and query boxes; bottom-left: line plots for individual stocks matching the query). Right: TimeSearcher 2 with query-by-example using a searchbox (light red background) and matching items in red.
Source: Generated with the TimeSearcher software with permission of University of Maryland Human-Computer Interaction Lab.

Hochheiser and Shneiderman (2004) implemented TimeSearcher as a visual exploration tool for multiple time-series. While employing a straightforward visual representation using line plots, its main objective is to enable users to identify and find patterns in the investigated data. To this end, the so-called *timebox query model* has been developed. It allows the specification of a rectangular query region that defines both a time interval and a value range of interest. Those time-series that comply with a query (i.e., overlap with the timebox) are displayed, whereas all others are filtered out. Users can combine multiple timeboxes to refine the query further and other query functionalities such as leaders and laggers, angular queries, and variable timeboxes are also part of TimeSearcher. To provide contextual information, the data envelope and the query envelope can be displayed. Buono et al. (2005) extended these features in TimeSearcher 2 by allowing the representation of heterogeneous datasets and providing a *searchbox query model* that effectively implements a query-by-example functionality. Here, occurrences of a brushed portion of the time-series are searched, whereas the similarity threshold of matches can be adjusted.

References

Buono, P., Aris, A., Plaisant, C., Khella, A., and Shneiderman, B. (2005). Interactive Pattern Search in Time Series. In *Proceedings of the Conference on Visualization and Data Analysis (VDA)*, pages 175–186. SPIE.

Hochheiser, H. and Shneiderman, B. (2004). Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization*, 3(1):1–18.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

TimeSearcher 3, River Plot

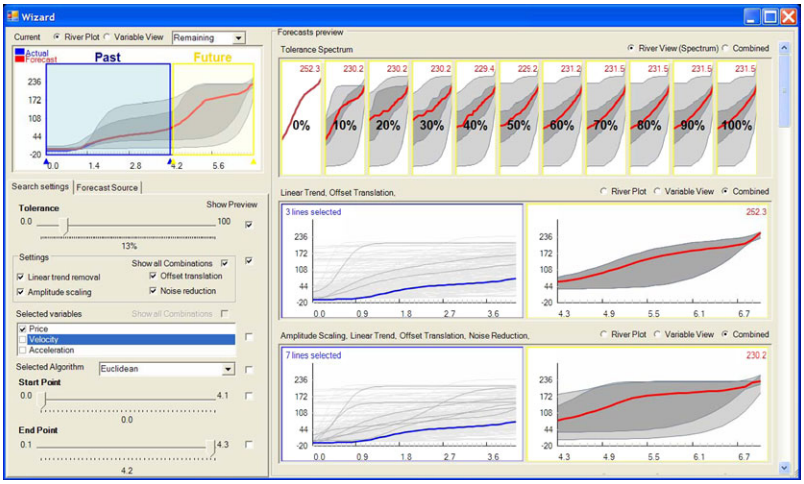


Fig. 7.38: Forecasting of online auction data. Top-left: selection of time interval for similarity search; bottom-left: selection of variables to consider and parameters to vary in the previews; right: preview area that assists users in understanding the impact of parameters – varying tolerance levels as river plots and different combinations of applied transformations as line plots and river plots.
Source: Image courtesy of Paolo Buono.

Buono et al. (2007) developed TimeSearcher 3 as a tool to support similarity-based forecasting of multivariate time-series. Similarity-based forecasting is a data-driven method using the similarity to a set of historical data for predicting future behavior. The outcome of the algorithm is affected by a number of options and parameters, for instance, the transformations applied or the tolerance threshold used for matching. As results, the median of the matched subsets becomes the forecast and descriptive statistics measures reflect the uncertainty associated with the forecast. This is displayed graphically as a simplified, continuous box plot, called a river plot. It uses superimposed, colored regions, for which light gray indicates the range between the minimum and maximum and dark gray the range between the 25% and 75% percentiles, and a line in the center, where red indicates the forecast, brown shows the median during the matching period, and black is the median before this period. TimeSearcher 3 builds upon TimeSearcher (↪ p. 188) and adds a preview interface to allow users to interactively explore the effects of adjusting algorithm parameters and to see multiple forecasts simultaneously.

References

Buono, P., Plaisant, C., Simeone, A., Aris, A., Shneiderman, B., Shmueli, G., and Jank, W. (2007). Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time Series Forecasting. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 191–196, Los Alamitos, CA, USA. IEEE Computer Society.

BinX

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

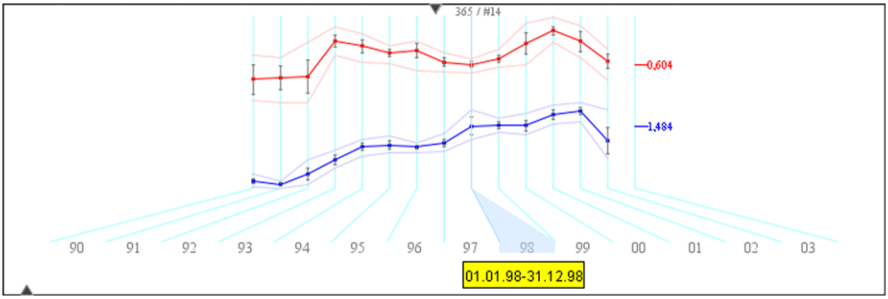


Fig. 7.39: Exchange rates for two currencies are compared using the BinX tool. Each bin aggregates the daily rates for a whole year. A selected bin is highlighted and its position on the global time scale is marked accordingly.
Source: Generated with the BinX tool with permission of Tamara Munzner.

Large time-series require the application of abstraction methods in order to reduce the number of time points to be displayed, thus keeping visualization costs at a manageable level. Finding a suitable degree of abstraction, however, is not an easy task. The BinX tool developed by Berry and Munzner (2004) is interesting in that it supports the exploration of different aggregations of a time-series. The aggregation is based on constructing bins, each of which holds a user-defined number of time points. Easy-to-use interaction is offered to quickly try out differently sized bins. BinX visualizes one or two quantitative time-dependent variables using common chart elements. An overview of the time axis is preserved at all times at the bottom of the BinX representation. The central chart view displays the two time-series in an aggregated fashion according to currently chosen bin size. In order to faithfully represent aggregated information, line plot (↪ p. 153), box-plots, and a min-max band are used in combination. The correspondence between a point in the chart and a time span (bin) on the time axis is represented upon user request. BinX supports clustering of bins as an additional mechanism for analytic abstraction. In this case, cluster affiliation of bins is encoded via color.

References

Berry, L. and Munzner, T. (2004). BinX: Dynamic Exploration of Time Series Datasets Across Aggregation Levels. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 5–6, Los Alamitos, CA, USA. IEEE Computer Society.

LiveRAC

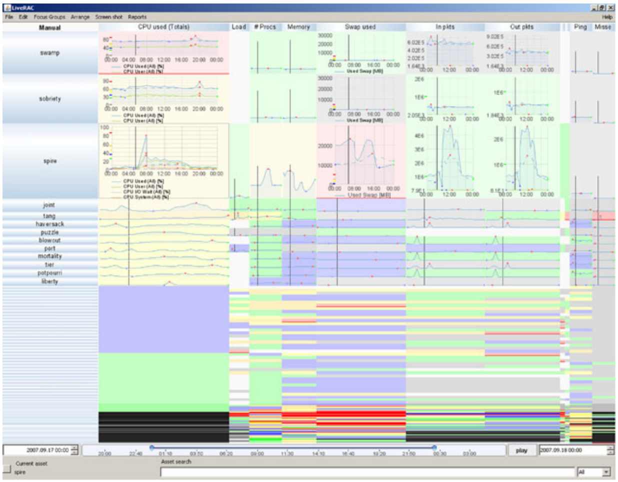


Fig. 7.40: A full day of system management time-series data showing more than 4000 devices in rows and 11 columns representing groups of monitored parameters. Representations within cells adapt to the available screen space by using different representations with more or less detail. *Source: McLachlan et al. (2008), © 2008 ACM. Used with permission.*

McLachlan et al. (2008) developed LiveRAC, a system for analyzing system management time-series data. LiveRAC scales to dozens of parameters collected from thousands of network devices. Familiar representations such as line plots (↪ p. 153), bar graphs (↪ p. 154), and sparklines (↪ p. 155) appear as the cells of a spreadsheet-like matrix. Rows and columns of the matrix are associated with monitored network devices and monitored parameters, respectively. Each cell contains an area-aware chart showing time on the horizontal axis and parameters on the vertical axis. To ensure that all cells remain visible at all times (i.e., to avoid scrolling), LiveRAC uses a so-called *stretch and squish* layout, which dynamically compresses and expands cells according to user interaction. Moreover, the individual charts adapt to the available screen space. This semantic zoom functionality ranges from charts with detailed labels, to smaller charts with fewer curves and less labeling, and ultimately to colored blocks for the smallest view. The cell background color represents changeable thresholds of minimum, maximum, or average values of the displayed parameters. Aggregation is applied if cells would overlap due to space restrictions, which is reflected in color intensity.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

References

McLachlan, P., Munzner, T., Koutsofios, E., and North, S. (2008). LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, CHI ’08, pages 1483–1492, New York, NY, USA. ACM Press.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

LifeLines2

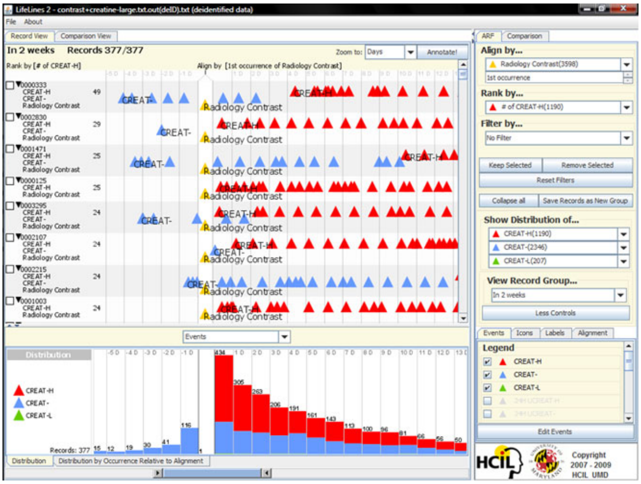


Fig. 7.41: LifeLines2 show patient records in stacked rows, where triangles indicate health-related events. A histogram view visualizes the number of events over time. Interaction operators (i.e., align, rank, filter) support visual exploration.
Source: Image courtesy of Taowei David Wang.

LifeLines2 by Wang et al. (2009) is an interactive visual exploration interface for instantaneous events based on categorical, health-related data (e.g., high, normal, or low body temperature). Events are displayed as triangles along a horizontal time axis, where color indicates event categories and data of different patient records are stacked vertically. An aggregation of events is represented as a histogram showing the number of occurrences over time. LifeLines2 introduces three powerful operators for interactive exploration: align, rank, and filter. The align operator can be used to arrange all records along a specific event type in temporal order, for example, to align a group of patients with regard to their first heart attack. Additionally, the time axis switches from an absolute time representation to relative time originating from the specified event (e.g., one week before, or two weeks after the first heart attack). The rank operator is useful for ordering records according to the number of occurrences of a specified event type. The filter operator allows searching of particular sequences of events including both the presence of events and the absence of events (e.g., patients having had a heart attack but no stroke following it).

References

Wang, T. D., Plaisant, C., Shneiderman, B., Spring, N., Roseman, D., Marchand, G., Mukherjee, V., and Smith, M. (2009). Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 15:1049–1056.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

Similan

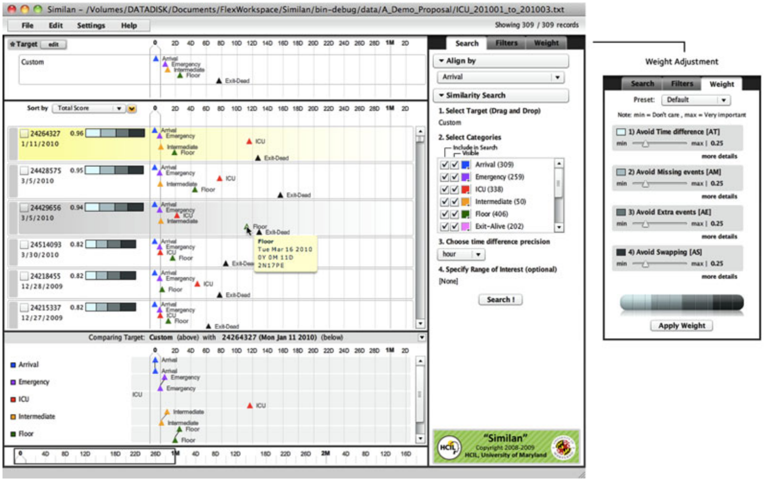


Fig. 7.42: Similan ranks patient records (center) according to their similarity to a target record (top). Individual records can be compared directly with the target (bottom). Various interaction operators, including adjustment of similarity weights (right), can be used to refine the visualization. *Source: Image courtesy of Krist Wongsuphasawat.*

Wongsuphasawat and Shneiderman (2009) describe Similan as a system for exploring patient records. Patient records are stacked upon each other and show health-related events as triangles, where color indicates event categories (e.g., arrival, emergency, ICU). Similan uses the same visual representation as LifeLines2 (↪ p. 192) but provides a different approach to data exploration. Instead of interactive filtering, records are ranked according to their similarity to a given event sequence (query-by-example). In Figure 7.42, the topmost record is the record that is most similar to the user-specified event sequence. This can be used to search for groups of patients who share similar temporal patterns. A dedicated view is provided to allow a direct comparison of the target query record with any particular record in the data. Another scenario is to search for an event sequence that the user is not certain whether it exists in the data; in this way the tool can give the most similar results if the exact event sequence does not exist. For determining the similarity of event sequences a similarity measure (M&M measure) has been developed. The weights of factors that determine the similarity measure can be adjusted interactively by the user.

References

Wongsuphasawat, K. and Shneiderman, B. (2009). Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 27–34, Los Alamitos, CA, USA. IEEE Computer Society.

CareCruiser

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

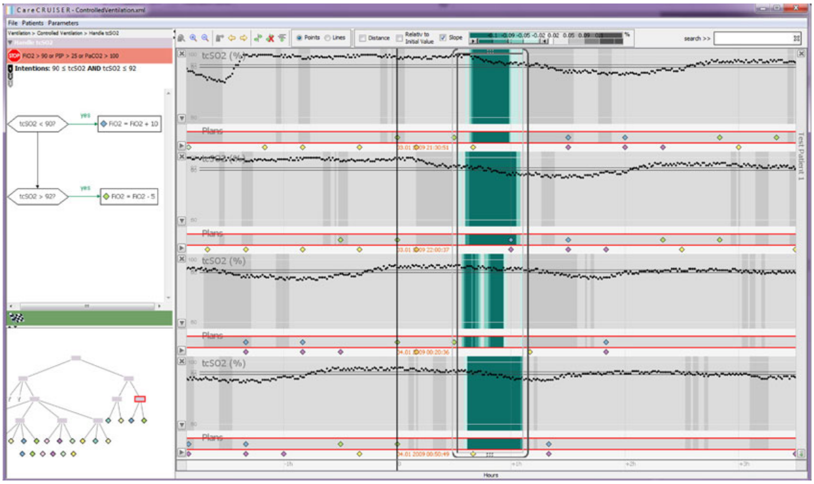


Fig. 7.43: A patient's parameters are displayed together with the applied clinical actions. In the selected area on the right, a delayed drop of the patient's $tcSO_2$ values after applying a specific clinical action is revealed. Contextual views are shown on the left – top-left: flow-chart like representation of the treatment plan logic; bottom-left: hierarchical decomposition of treatment plan. *Source: Generated with the CareCruiser software.*

CareCruiser by Gschwandtner et al. (2011) is a visualization system for exploring the effects of clinical actions on a patient's condition. It supports exploration via aligning, color-highlighting, filtering, and providing focus and context information. Aligning clinical treatment plans vertically supports the comparison of the effects of different treatments or the comparison of different effects of one treatment plan applied on different patients. Three different color-schemes are provided to highlight interesting portions of the development of a parameter: highlighting the distance of the actual values to the intended value helps to identify critical values; highlighting the progress of the actual values relative to the initial values shows to what extent the applied treatment plan has the intended effect; and highlighting the slope of a value helps to explore the immediate effects of applied clinical actions. A range slider is provided to filter the color-highlighting for selected events (see Figure 7.43, top) and a focus window which grays out the color-information outside its borders is used to support a focused investigation of a region of specific interest.

References

Gschwandtner, T., Aigner, W., Kaiser, K., Miksch, S., and Seyfang, A. (2011). CareCruiser: Exploring and Visualizing Plans, Events, and Effects Interactively. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis 2011)*, pages 43–50, Los Alamitos, CA, USA. IEEE Computer Society.

Layer Area Graph

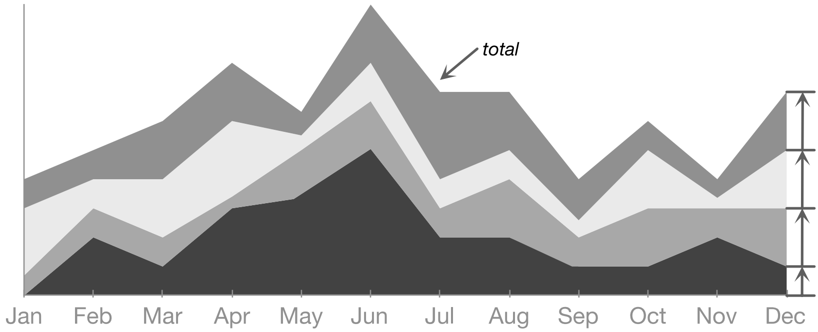


Fig. 7.44: Shows the developments of four variables as layered bands while emphasizing the total sum.
Source: Adapted from Harris (1999).

Layer area graphs might be used when comparing time-series that share the same unit and can be summed up (see Harris, 1999). A layer area graph is a stacked visualization where time-series plots are drawn upon each other as layered bands. Caution needs to be exercised for this kind of representation because it is sensitive to the order of the layers. Different orders influence the visual appearance of the individual layers because only the bottommost layer has a straight baseline. All subsequent layers are drawn relative to the layers below. An advantage of layer area graphs is the fact that they emphasize the total sum of values while providing information about the parts that constitute it. More advanced visualization techniques such as the ThemeRiver (↪ p. 197) or stacked graphs (↪ p. 199) build upon the basic principle of layer area graphs.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

Braided Graph

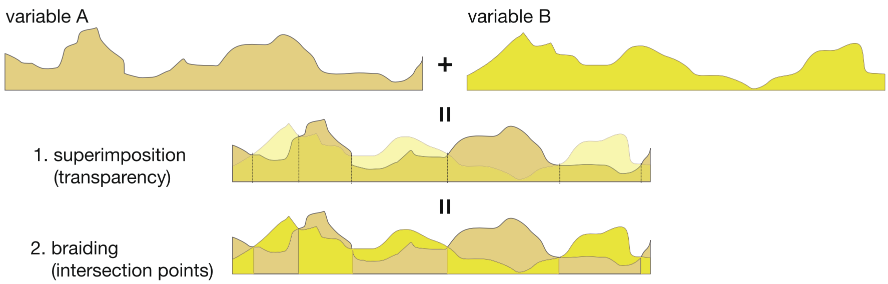


Fig. 7.45: Construction scheme of a braided graph for two variables. Top: individual variables as silhouette graphs; center: superimposed silhouette graphs using transparency (dashed lines show intersection points); bottom: braided graph (segments between intersection points are sorted to ensure visibility of all fragments).
Source: Adapted from Javed et al. (2010). © 2010 IEEE. Used with permission.

Braided graphs allow for superimposing silhouette graphs to show multivariate data. They were developed in order to take advantage of the enhanced perception of silhouette graphs (↪ p. 175) and at the same time avoiding the disadvantage of varying baselines of layered graphs (↪ p. 195). Simply drawing silhouette graphs on top of each other would lead to occlusion problems where a silhouette for larger data values occludes silhouettes for smaller values. The solution to this problem is to identify the points at which silhouettes intersect and to adapt the drawing order in between two intersections individually so that smaller silhouettes are always in front of larger ones. This ensures that all segments of all variables remain visible for the complete time-series. In a user study, Javed et al. (2010) compared line plots (↪ p. 153), silhouette graphs (↪ p. 175), horizon graphs (↪ p. 157), and braided graphs along the three tasks of determining local maxima, comparing global slopes, as well as locating and comparing values at specific time points. Besides the visualization type, the number of displayed time-series, and the height of the representation was varied. Interestingly, the type of visualization was not found to have a significant effect on task correctness in all conditions. However, subjects using line plots and braided graphs were significantly faster when searching for local maxima. For value and slope comparison tasks this was not the case. In general, higher numbers of time-series caused decreased correctness and increased completion time. Decreasing display space had a negative effect on correctness but little impact on completion time.

References

Javed, W., McDonnel, B., and Elmqvist, N. (2010). Graphical Perception of Multiple Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–34.

ThemeRiver

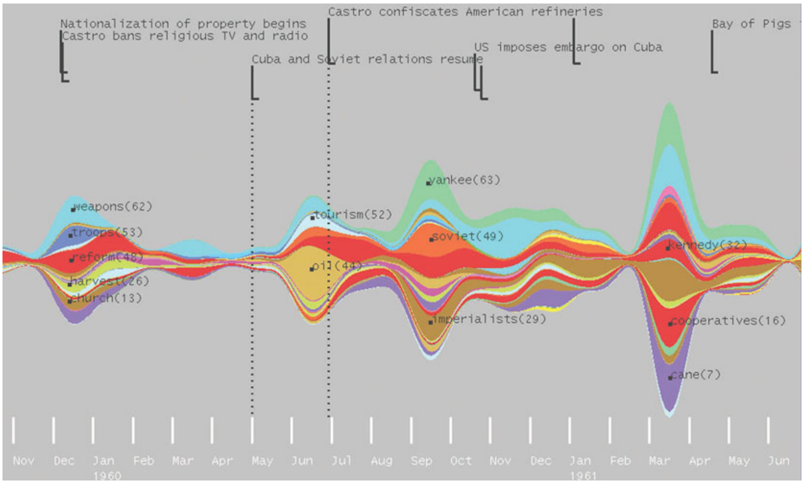


Fig. 7.46: The ThemeRiver representation uses the metaphor of a river that flows through time. Colored currents within the river reflect thematic changes in a document collection, where the width of a current represents the relevance of its associated theme.
Source: Havre et al. (2002), © 2002 IEEE. Used with permission.

The ThemeRiver technique developed by Havre et al. (2000) represents changes of news topics in the media. Each topic is displayed as a colored current whose width varies continuously as it flows through time. The overall image is a river that comprises all of the topics considered. The ThemeRiver provides an overview of the topics that were important at certain points in time. Hence, the main focus is directed towards establishing a picture of an easy to follow evolution over time using interpolation and approximation. Moreover, ThemeRiver representations can be annotated, e.g., with related major historical events, and raw data points with exact values can be shown. Even though the ThemeRiver was originally invented to visualize thematic changes in document collections, it is also suited to represent other multivariate, quantitative data. Because perception of data differs depending on where in the river individual variables are shown, it is important to provide interaction techniques to allow users to rearrange the horizontal position of variables.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

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

References

Havre, S., Hetzler, E., and Nowell, L. (2000). ThemeRiver: Visualizing Theme Changes Over Time. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 115–124, Los Alamitos, CA, USA. IEEE Computer Society.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

3D ThemeRiver

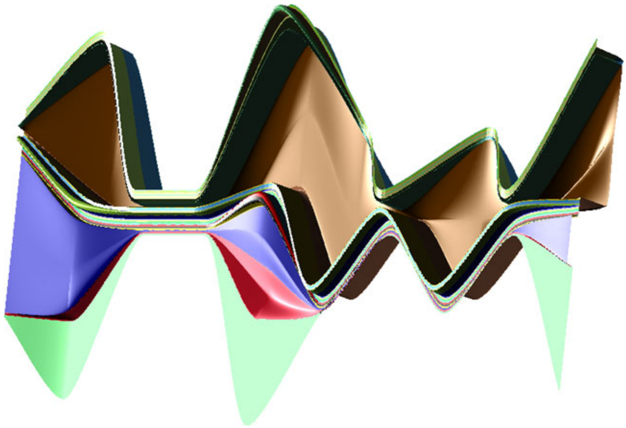


Fig. 7.47: Distinctly colored currents form the overall shape of the 3D ThemeRiver. The width, and additionally the height of currents is varied to visualize time-oriented data. In this figure, width encodes the overall distribution of 17 clusters of aerosol data and height indicates the incidence of zinc.
Source: Imrich et al. (2003), © 2003 IEEE. Used with permission.

Imrich et al. (2003) propose a 3D variant of the ThemeRiver technique (↪ p. 197). The 3D approach inherits the basic visual design from its 2D counterpart: multiple time-oriented variables are encoded to the widths of individually colored currents that form a river flowing through time along a horizontal time-axis. In the 2D variant, only one data variable can be visualized per current, namely by varying the current's width. Imrich et al.'s extension addresses this limitation. By extending the design to the third dimension it is possible to use an additional visual encoding: the height (in 3D) of a current can be varied to encode further information. This design is particularly suited to visualizing ternary covariate trends in the data. Imrich et al. conducted user tests to evaluate the usefulness of the 3D encoding, and indeed got positive results that indicate that the 3D variant has advantages over the 2D variant. Specifically, the availability of appropriate interactive 3D navigation tools is highlighted as an important factor contributing to the success of the 3D ThemeRiver.

References

Imrich, P., Mueller, K., Imre, D., Zelenyuk, D., and Zhu, W. (2003). Interactive Poster: 3D ThemeRiver. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society.

Stacked Graphs

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

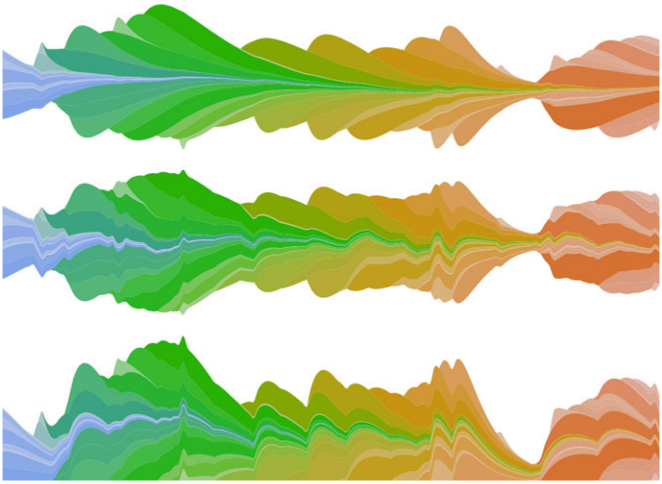


Fig. 7.48: Multivariate time-series visualized as stacked graphs with different designs: Streamgraph design (top), ThemeRiver layout (center), and traditional stacking (bottom).
Source: Generated with the streamgraph_generator code base.

Stacking multiple graphs on top of each other is a suitable approach to visualizing multiple time-dependent variables (see Harris, 1999). Elaborate variants of stacked graphs have been investigated in detail by Byron and Wattenberg (2008). To visualize the evolution of an individual variable, data values are encoded to the height of a so-called layer that extends along the horizontal time axis. A special color map is applied to visualize additional data variables and to make individual layers distinguishable. Several layers are then stacked on top of each other, effectively creating an overall graph that represents the visual sum of the entire dataset. Layout and sorting of layers can be done in various ways, resulting in quite different designs such as the Streamgraph design, the ThemeRiver layout (↪ p. 197), or traditional layer area graphs (↪ p. 195). The Streamgraph design (Figure 7.48, top) is notable because it received quite positive feedback when it appeared on the New York Times web site as a visual representation of box office revenues. In that version, individual layers were also outfitted with text labels.

References

Byron, L. and Wattenberg, M. (2008). Stacked Graphs – Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252.

Harris, R. L. (1999). *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, New York, NY, USA.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

TimeWheel

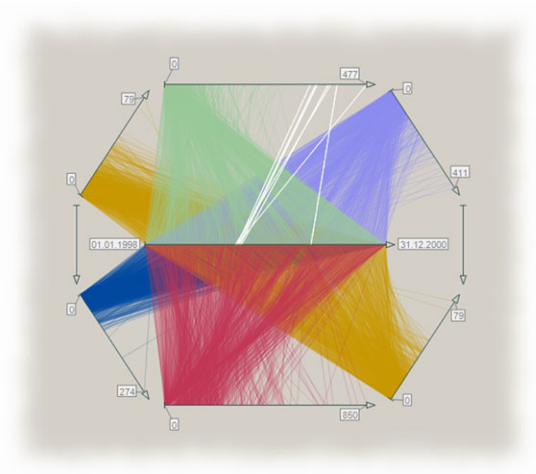


Fig. 7.49: The TimeWheel's central axis represents time. The axes in the periphery represent time-dependent variables; here we see the number of cases for eight diagnoses. Days with particularly high numbers of influenza cases are highlighted.
Source: Generated with the VisAxes software.

Tominski et al. (2004) describe the TimeWheel as a technique for visualizing multiple time-dependent variables. The TimeWheel consists of a single time axis and multiple data axes for the data variables. The time axis is placed in the center of the display to emphasize the temporal character of the data. The data axes are associated with individual colors and are arranged circularly around the time axis. In order to visualize data, lines emanate from the time axis to each of the data axes to establish a visual connection between points in time and associated data values. These lines form visual patterns that allow users to identify positive or negative correlations with the time axis, trends, and outliers. Such patterns can be best discerned for those data axes that are parallel to the time axis. To bring data axes of interest into this focus, users can rotate the TimeWheel. Focused data axes are further emphasized by stretching them, effectively providing them with more drawing space. Data axes that are perpendicular to the time axis are more difficult to interpret and are, therefore, attenuated using color fading and shrinking. Interactive exploration, including navigation in time, is supported through different types of interactive axes.

References

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

MultiComb

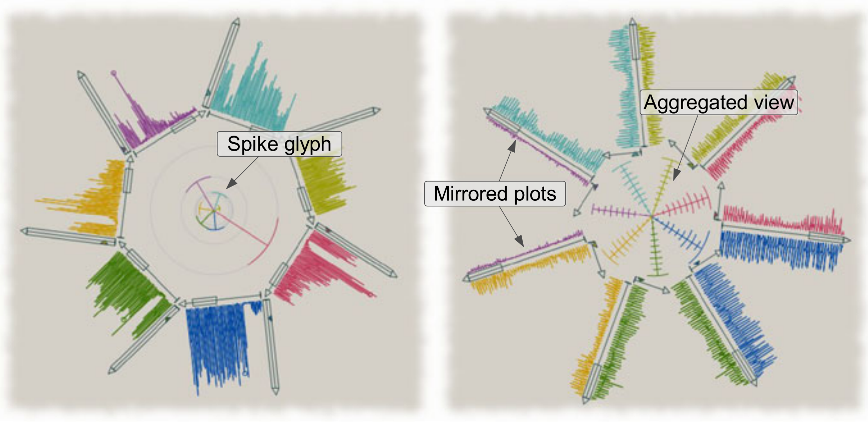


Fig. 7.50: Two MultiComb representations visualize seven time-dependent variables. In the left MultiComb, line plots are arranged around the display center, in the right one, they extend outwards. The very center of the MultiCombs can be used to display additional information via a spike glyph or an aggregated view.
Source: Generated with the VisAxes software.

Line plots (↪ p. 153) are expressive visual representations for univariate data. The rationale behind the MultiComb visualization is to utilize this expressiveness for representing multiple time-dependent variables. Tominski et al. (2004) describe the MultiComb as a visual representation that consists of multiple radially arranged line plots. Two alternative designs exist: time axes are arranged around the display center (see Figure 7.50, left) or time axes extend outwards from the MultiComb's center (see Figure 7.50, right). In the latter case, optional mirror plots duplicate plots of neighbor variables to ease visual comparison. To maintain a certain aspect ratio for the separate plots, the axes do not start in the very center of the MultiComb. The screen space in the center can therefore be used to provide additional views: a spike glyph can be shown to allow a detailed comparison of data values for a selected time point, or an aggregated view might display the history of a temporal data stream in an aggregated fashion. Various possibilities for interaction allow users to browse in time, to zoom into details of the time axes, as well as to add, remove, and reorder plots, and to rotate the MultiComb.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Tominski, C., Abello, J., and Schumann, H. (2004). Axes-Based Visualizations with Radial Layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1242–1247, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

VIE-VISU

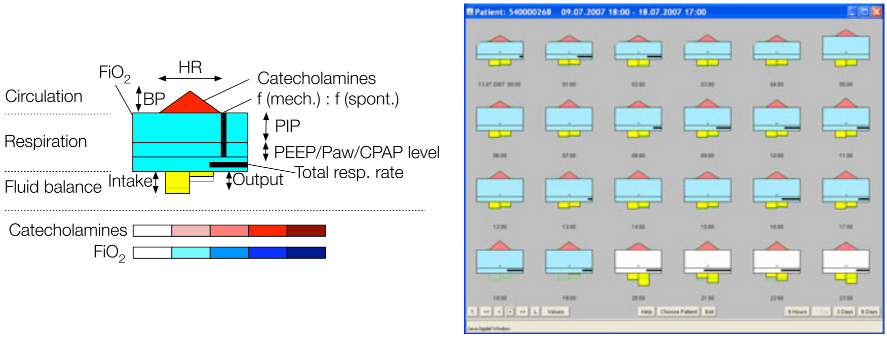


Fig. 7.51: VIE-VISU encodes fifteen health-related patient parameters to different visual attributes of a glyph (left). The example on the right shows neonatal patient information on an hourly basis for the course of the day.
Source: Left: Adapted from Horn et al. (2001). Right: Image courtesy of Werner Horn.

time
arrangement: **linear**
time primitives: **instant**

Paper-based analysis of patient records is hard to conduct because many parameters are involved and an overall assessment of the patient's situation is difficult. Therefore, Horn et al. (2001) developed VIE-VISU, an interactive glyph-based visualization technique for time-oriented patient records. The glyph consists of three parts that represent circulation, respiration, and fluid balance parameters. All in all, 15 parameters are visualized using different visual attributes (i.e., length, width, color) as illustrated in the left part of Figure 7.51. For example, the circulation parameter heart rate (HR) is encoded to the width of the triangle on top of the glyph and the triangle's color encodes catecholamines (color legend is given at the bottom). Each glyph represents a one hour period and 24 glyphs are combined in a small multiples display (↪ p. 236) as shown in the right part of Figure 7.51. Interaction controls support navigation in time and switching to different periods for the small multiples view. VIE-VISU helps users to combine different measurements, maintain their relationships, show their development over time, and make specific, possibly life threatening situations easy to spot.

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

References

Horn, W., Popow, C., and Unterasinger, L. (2001). Support for Fast Comprehension of ICU Data: Visualization using Metaphor Graphics. *Methods of Information in Medicine*, 40(5):421–424.

Timeline Trees

Day	Market basket and money spent
Mon:	milk $1, bananas $3
Tue:	cheese $1, apples $3
Wed:	milk $1, bananas $1, grapes $2
Thu:	milk $1
Fri:	milk $1, cheese $3

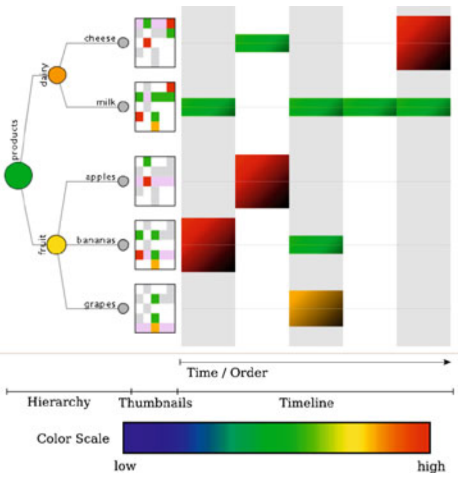


Fig. 7.52: A smaller set of products in a market basket is visualized using timeline trees. One can see that milk is bought regularly (green boxes for all but one day), and that cheese, apples, and bananas are more expensive (higher red-colored boxes).
Source: Image courtesy of Michael Burch.

Data that describe items which are related to each other are quite common. An example of such data are transactions in on-line shopping systems where products being bought together are considered to be related. Burch et al. (2008) visualize temporal sequences of transactions by means of so-called timeline trees. The visual representation consists of three parts: a display of an information hierarchy, a timeline representation of temporal sequences, and thumbnail pictures. The information hierarchy is a static hierarchical categorization of data items (e.g., a system of product groups), where groups can be expanded or collapsed interactively to view the data at different levels of detail. The timeline view shows multiple sequences of boxes for the current level of detail, where color and box size are used to encode data values (e.g., product price) of an item (or group) at a particular point in time. Thumbnails for each leaf of the information hierarchy show an overview of transactions masked by the corresponding leaf node. Enhanced with several interaction facilities, timeline trees help users to understand trends in the data and to find relations between different levels of abstractions (e.g., different product groups, or product groups and specific products).

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Burch, M., Beck, F., and Diehl, S. (2008). Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 75–82, New York, NY, USA. ACM Press.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

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

Pixel-Oriented Network Visualization

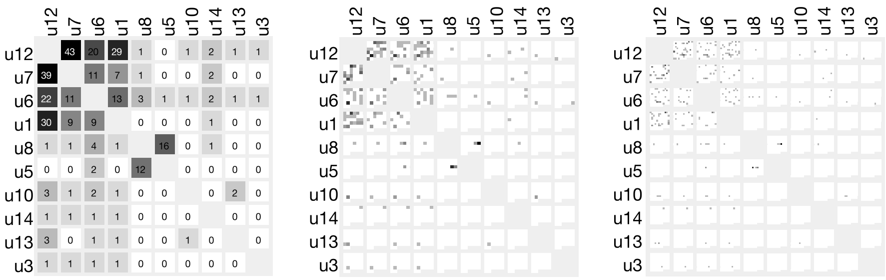


Fig. 7.53: Wiki collaboration patterns. Left: adjacency matrix showing users as rows and columns and collaboration intensity by color brightness of cells that connect two users (darker means more collaboration); center: pixel-oriented view where collaboration dynamics are shown in a 6x6 pixel array laid out row by row and each pixel represents a four week period; right: more fine-grained configuration that shows weekly steps.
Source: Images courtesy of Klaus Stein.

Social networks consist of actors and relationships between them. Unlike most static node-link representations of graph-like structures would suggest, these networks are dynamically changing over time. The two most common forms of visualizing time-varying networks are applying animation to node-link diagrams or applying the concept of small multiples (↪ p. 236) by showing snapshots of different points in time. An alternative display is suggested by Stein et al. (2010). They developed a pixel-oriented visualization of networks (PONV) that reveals interaction patterns between actors by integrating pixel-based representations (↪ p. 180) within the cells of an adjacency matrix. An adjacency matrix can be represented visually as a matrix table whose rows and columns represent the nodes of the network. The table cell at the intersection of a particular column and row visualizes information about the relationship between the corresponding nodes. Figure 7.53 (left) shows an example of an adjacency matrix where the darkness of a cell represents the collaboration intensity of two individuals (the value 0 and a white cell background indicates that there is no relationship between two individuals). In order to show the temporal evolution of the relationships, Figure 7.53 (center) uses an alternative representation. Now each cell contains a 6x6 pixel glyph, where each pixel represents the aggregated collaboration intensity of a four-week period. The user can interactively control the visualization, including the color scale, the pixel pattern arrangement, and the time period to be covered by each pixel (e.g., daily values as in Figure 7.53, right).

References

Stein, K., Wegener, R., and Schlieder, C. (2010). Pixel-Oriented Visualization of Change in Social Networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 233–240, Los Alamitos, CA, USA. IEEE Computer Society.

CiteSpace II

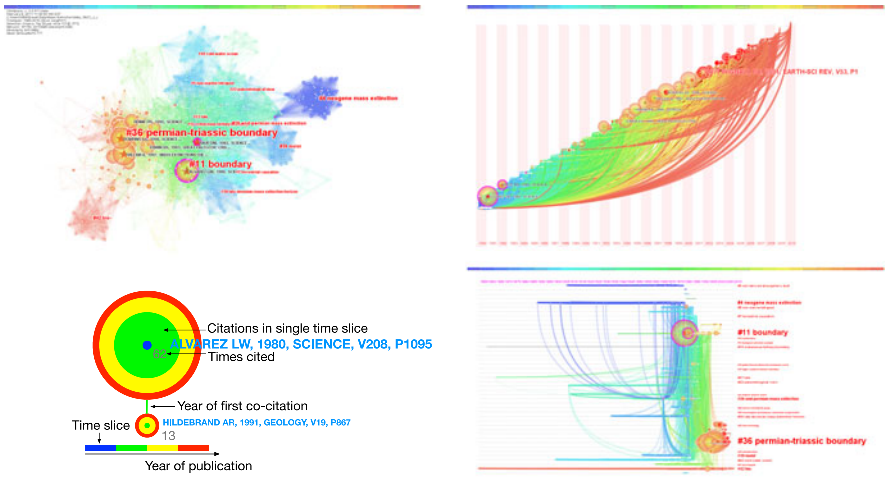


Fig. 7.54: A network of 750 most cited articles on mass extinction (1980–2010). Top-left: cluster view; bottom-left: legend – node size reflects overall amount of citations and colored rings show citations per time slice; top-right: time-zone view; bottom-right: timeline view.
Source: Images courtesy of Chaomei Chen.

CiteSpace II by Chen (2006) is a system that supports visual exploration of bibliographic databases. It combines rich analytic capabilities to analyze emerging trends in a knowledge domain with interactive visualization of co-citation networks. Three complementary views are provided for the visual representation: a cluster view, a time-zone view, and a timeline view. The cluster view represents a network as a node-link diagram using a force-directed layout. Node size shows how often an article or cluster was cited overall and citation tree rings of a node display the citation history from the center outward. The color of a ring represents a time period and its thickness is proportional to the number of citations in this period. Colors of links represent the time slice of the first co-citation. The time-zone view displays a network by arranging its nodes along vertical strips representing time zones using a modified spring-embedder layout that controls only the vertical positions of nodes freely. In the timeline view, time is mapped to the horizontal position and clusters are arranged along horizontal lines. Users can adjust a complex set of parameters to control the analysis process as well as interact and manipulate the visualization of a knowledge domain. CiteSpace II also provides clustering and labeling functions to help the user to interpret various structural and temporal patterns.

References

Chen, C. (2006). CiteSpace II: Detecting and Visualizing Emerging Trends and Transient Patterns in Scientific Literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

history flow

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

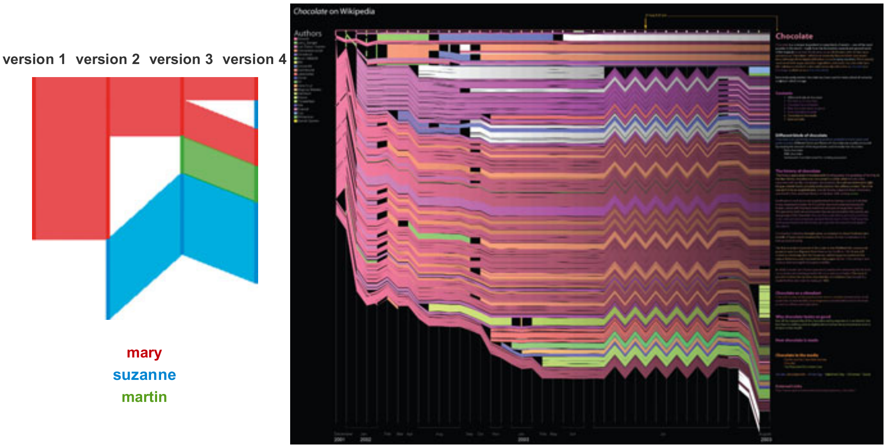


Fig. 7.55: The history flow shows vertical revision lines, one for each revision, where colored sections reflect the different authors of a document as schematically depicted on the left. This method is applied to the visualization of the Wikipedia entry on chocolate as shown on the right. *Source: Images courtesy of Fernanda B. Viégas.*

Viégas et al. (2004) designed history flow to be an exploratory wiki article analysis tool for finding author collaboration patterns, showing relations between document versions, revealing patterns of cooperation and conflict, as well as making broad trends immediately visible. The basis for the representation are so-called revision lines. These top-aligned, vertical lines are displayed for every version of a document. The length of revision lines is proportional to the document length. Individual sections of a revision line are colored differently to visualize which authors worked on which parts of a document. The sections associated with a particular author are visually connected from one revision to the next. One can discern stable sections and splits of sections. Gaps in connections clearly indicate deletions and insertions. Two different layouts can be used for spacing revision lines: uniform spacing (space by occurrence / event-based) or spacing according to time (space by date / time-based). The first layout shows each document change equally spaced without showing time intervals between versions proportionally. The second alternative additionally gives information about the exact timing.

References

Viégas, F. B., Wattenberg, M., and Dave, K. (2004). Studying Cooperation and Conflict between Authors with history flow Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 575–582, New York, NY, USA. ACM Press.

PeopleGarden

Fig. 7.56: This PeopleGarden shows about 1200 posts entered to a discussion board during a period of two months. Users are represented by flowers whose petals, in turn, represent individual messages posted by a user.
Source: Xiong and Donath (1999), © 1999 ACM. Used with permission.

Xiong and Donath (1999) developed PeopleGarden as a graphical representation of users' interaction histories in discussion groups. PeopleGarden visualizes data about users and messages posted to an online interaction environment. It integrates information on the time of posting, amount of response, and whether a post starts a new conversation. For intuitive understanding, PeopleGarden uses the metaphor of a garden of flowers. The garden represents the whole environment and flowers represent individual users within the environment. The petals of a flower stand for the messages posted by a user, the time of posting is mapped to the ordering and saturation of the petals, the amount of response is represented by circles that are stacked on top of petals, and color is used to depict whether a post starts a new conversation. Furthermore, the height of the flower gives information about how long a specific user has been a member of the discussion group. Using these visual representations, one can easily spot dominant voices, long time participants, or very active groups.

References

Xiong, R. and Donath, J. (1999). PeopleGarden: Creating Data Portraits for Users. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 37–44, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant**

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

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

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

PostHistory

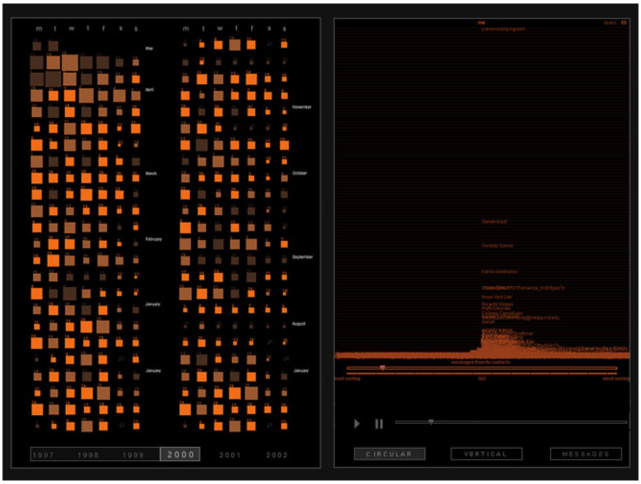


Fig. 7.57: The calendar panel on the left shows e-mail activity on a daily basis, where the number of emails and their average directedness are mapped to box size and color, respectively. The contacts panel on the right displays the names of people who sent messages to the user.
Source: Image courtesy of Fernanda B. Viégas.

Viégas et al. (2004) developed PostHistory with the goal of visually uncovering different patterns of e-mail activity (e.g., social networks, e-mail exchange rhythms) and the role of time in these patterns. PostHistory is user-centric and focuses on a single user's direct interactions with other people through e-mail. The social patterns are derived from analyzing e-mail header information. So, not the content of messages, but the tracked traffic is used as the basis for the analysis of people's e-mail conversations over time. Basically, the user interface visualizes a full year of e-mail activity and is divided into two main panels: a calendar panel on the left and a contacts panel on the right. The calendar panel shows the intensity of e-mail activity on a daily basis whereas a square represents a single day and each row of squares represents a week. The size of a square is determined by the quantity of e-mail received on that day and its color represents the average directedness of messages, i.e., whether a mail was received via TO, CC, or BCC. The brighter the color, the more directed the messages are that are present on that day. The contacts panel is used for displaying the names of people who sent messages to the user.

References

Viégas, F., Boyd, D., Nguyen, D., Potter, J., and Donath, J. (2004). Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, pages 109–118, Los Alamitos, CA, USA. IEEE Computer Society.

MOSAN

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D, 3D**

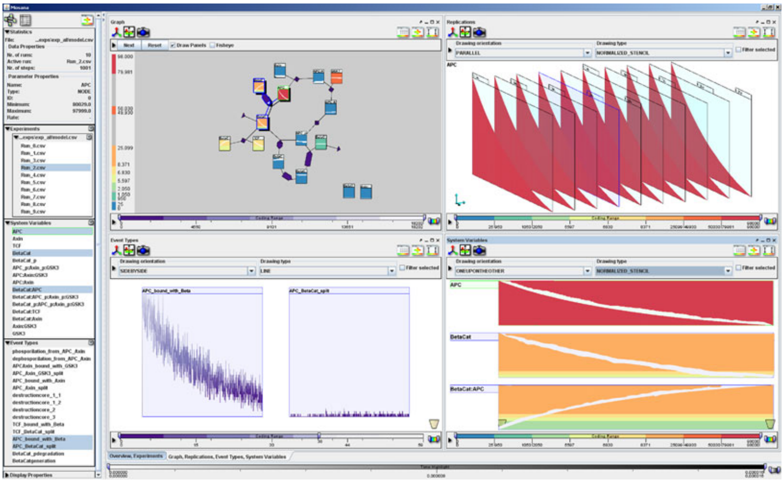


Fig. 7.58: The top-left view shows a simulated reaction network. An overview of the time-dependent simulation data is given by the small line plots in the boxes of the network. Three coordinated linked views are provided for the comparison of simulation runs and variables.
Source: Image courtesy of Andrea Unger.

MOSAN is a tool for visualizing multivariate time-oriented data that result from simulation of reaction networks. Due to the stochastic multi-run simulation, each variable comprises multiple time-series. In order to facilitate the understanding of the complex dependencies in the data it is necessary to jointly visualize structural information and stochastic simulation data together. To this end, Unger and Schumann (2009) combine different views within a single interactive interface. In an overview, time-oriented data are shown along with the structural relations among the variables in the reaction network. The structural relations are shown by a graph layout, where boxes correspond to variables, and simulation data are visualized by small line plots within the boxes (see Figure 7.58, top-left). The small line plots provide a highly aggregated view of the stochastic simulation data, thus focusing on the communication of the general temporal trends. Furthermore, advanced color-coding is applied to the plots to support the comparison of heterogeneous value ranges among variables. In addition to the overview, coordinated linked views support the inspection of individual time-series of the same variable (see Figure 7.58, top-right) as well as the detailed inspection and comparison of temporal developments of different variables selected from the overview (see Figure 7.58, bottom).

References

Unger, A. and Schumann, H. (2009). Visual Support for the Understanding of Simulation Processes. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 57–64, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

Data Tube Technique

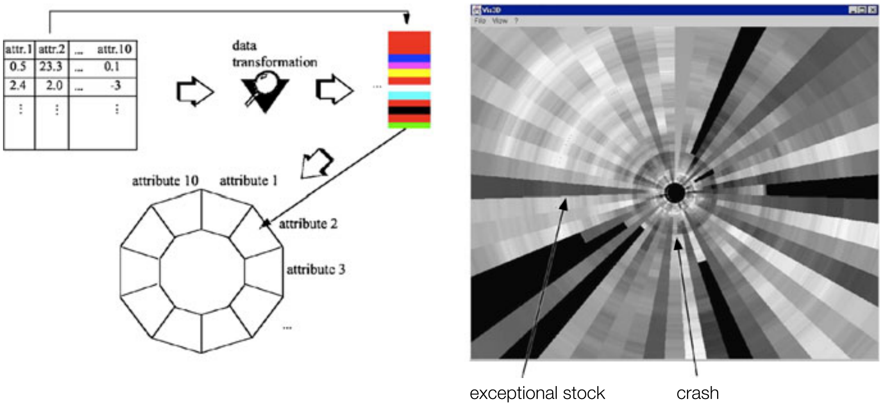


Fig. 7.59: Data are mapped onto the inside of a 3D tube using a tabular layout. Each slice of the tube represents an instant and each cell represents a data parameter by color. Left: visual mapping schema; right: exploring 50 different stocks.
Source: Images courtesy of Mihael Ankerst.

In the data tube technique by Ankerst (2001) multiple time-oriented variables are mapped to bands that follow the inside of a 3D tube (see Figure 7.59, left). Each slice of the tube represents an instant and each cell represents a data value by color. The tube is viewed from above and time is flowing to or from the center of the tube. The user is able to explore the data by interactively moving through the 3D tube. Because of the 3D perspective distortion, cells that are further away appear to be smaller in size, much like in a focus+context display. As a result of this, the number of displayed parameters and the number of displayable records can be quite large. Later, Ankerst et al. (2008) also developed a comprehensive temporal data mining architecture called DataJewel that is closely integrated with pixel-oriented visualization techniques.

References

Ankerst, M. (2001). Visual Data Mining with Pixel-oriented Visualization Techniques. In *Proceedings of ACM SIGKDD Workshop on Visual Data Mining*, New York, NY, USA. ACM Press.

Ankerst, M., Kao, A., Tjoelker, R., and Wang, C. (2008). DataJewel: Integrating Visualization with Temporal Data Mining. In Simoff, S., Böhlen, M., and Mazeika, A., editors, *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 312–330. Springer, Berlin, Germany.

Kiviat Tube

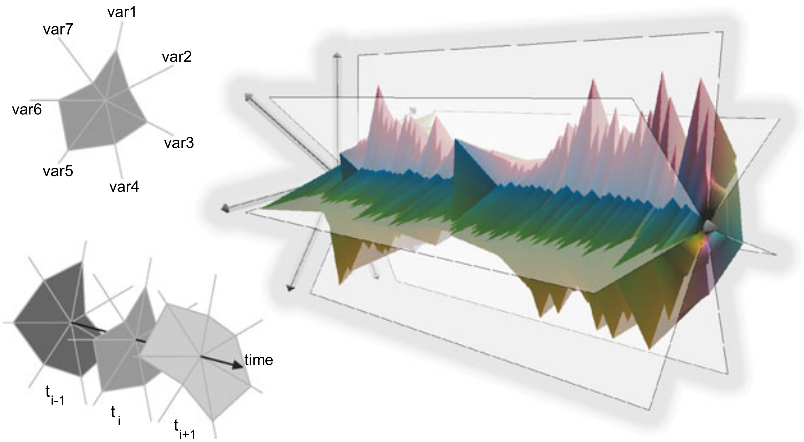


Fig. 7.60: Construction of a three-dimensional Kiviat tube representing seven time-dependent variables. Peaks and valleys indicate ups and downs in the evolution of the data over time. Wings assist in associating features of the Kiviat tube to particular variables in the data.
Source: Generated the VisAxes3D tool.

The Kiviat tube by Tominski et al. (2005) visualizes multiple time-dependent variables. The construction of a Kiviat tube is as simple as stacking multiple Kiviat graphs (see Kolence and Kiviat, 1973) along a shared time axis. Each Kiviat graph represents the data for multiple variables for a specific point in time. But instead of drawing individual Kiviat graphs, a three-dimensional surface is constructed. This way, multiple, otherwise separated time points are combined to form a single 3D body that represents the dataset as a whole. The spatial characteristics of a Kiviat tube can be recognized easily, as it allows users to identify peaks or valleys in the data over time. Additional semitransparent wings assist in relating identified patterns to particular variables. Common interaction methods can be used for zooming and rotation around arbitrary axes. Rotation specifically around the time axis enables users to quickly access variables on all sides of the Kiviat tube. Interactive axes allow users to navigate back and forth in time to visit different intervals of a possibly large time-series.

References

Kolence, K. W. and Kiviat, P. J. (1973). Software Unit Profiles & Kiviat Figures. *SIGMETRICS Performance Evaluation Review*, 2:2–12.

Tominski, C., Abello, J., and Schumann, H. (2005). Interactive Poster: 3D Axes-Based Visualizations for Time Series Data. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, pages 49–50, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

Temporal Star

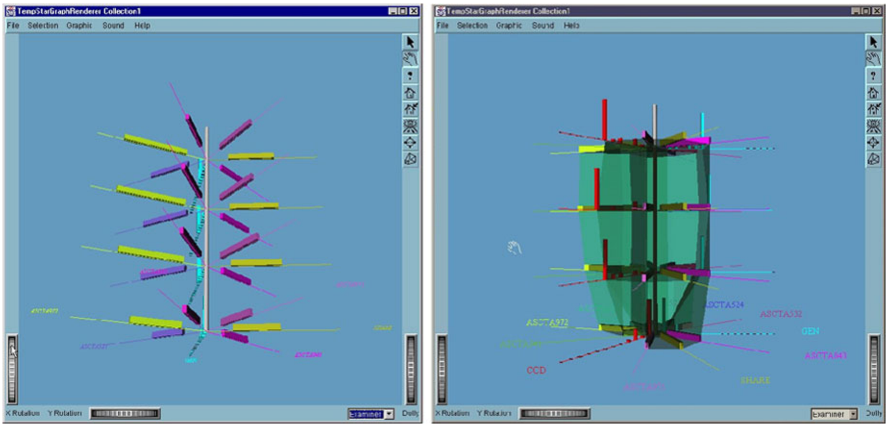


Fig. 7.61: 3D representation of circular column graphs that are arranged in a row to represent each time step. A transparent veil can be displayed to enhance the perception of the dataset's evolution. *Source: Images courtesy of Monique Noirhomme.*

The temporal star technique by Noirhomme-Fraiture (2002) visualizes multivariate data structures in 3D. For each point in time, a circular column graph is drawn that represents each variable's value as a bar length in a circular arrangement. These graphs are aligned in a row to represent the development of the dataset over time (see Figure 7.61, left). A unique color is assigned to each variable to aid recognition of variables across time. Moreover, a transparent veil can be displayed to enhance the perception of the dataset's evolution as a whole (see Figure 7.61, right). The concept used is similar to that of the Kiviat tube (↪ p. 211), which uses Kiviat graphs instead of circular column graphs. In the temporal star technique, difference plots are also integrated, showing the relative differences between variables rather than absolute values. The rendering parameters, the shown time intervals, and the configuration of the axes can be adjusted interactively. Furthermore, the temporal star technique is integrated with a data warehousing application that provides rich data manipulation features.

References

Noirhomme-Fraiture, M. (2002). Visualization of Large Data Sets: The Zoom Star Solution. *Journal of Symbolic Data Analysis*, 0(0).

Time-tunnel

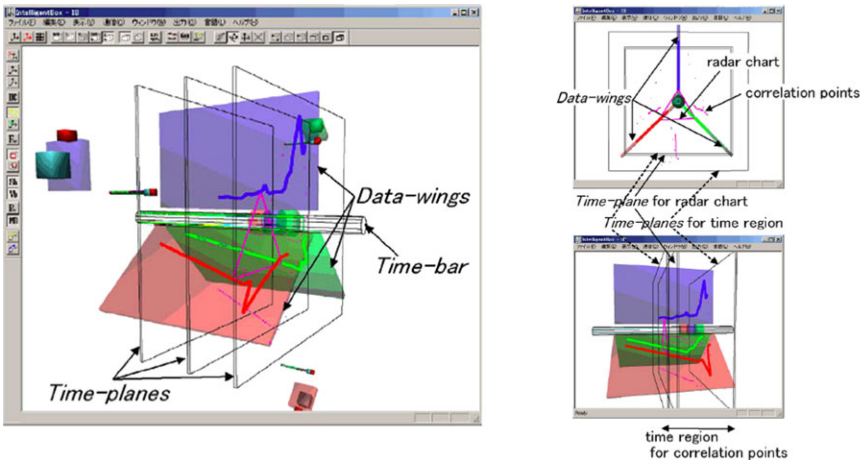


Fig. 7.62: Left: individual plots are put onto semi-transparent planes (data-wings) that are positioned around a central time-bar. Right: selected time intervals can be viewed as radar charts.
Source: Akaishi and Okada (2004), © 2004 IEEE. Used with permission.

Akaishi and Okada (2004) developed time-tunnel as a data analysis technique for visualizing a number of time-series plots in a 3D virtual space. The individual plots are put onto semi-transparent planes (data-wings) that are positioned around a central time-bar in a fan-like manner (see Figure 7.62, left). In the example above, line plots are used for visualizing data but any other linear time visualization might also be used. Multiple planes can be overlapped and compared due to their transparency. Furthermore, single time slices or selected time intervals can be viewed as radar charts (see Figure 7.62, right) and provide a multivariate point of view. Additionally, not only time-series, but any kind of information can be put onto a plane, making it a multimedia presentation tool.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Akaishi, M. and Okada, Y. (2004). Time-tunnel : Visual Analysis Tool for Time-series Numerical Data and its Aspects as Multimedia Presentation Tool. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 456–461, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

Parallel Glyphs

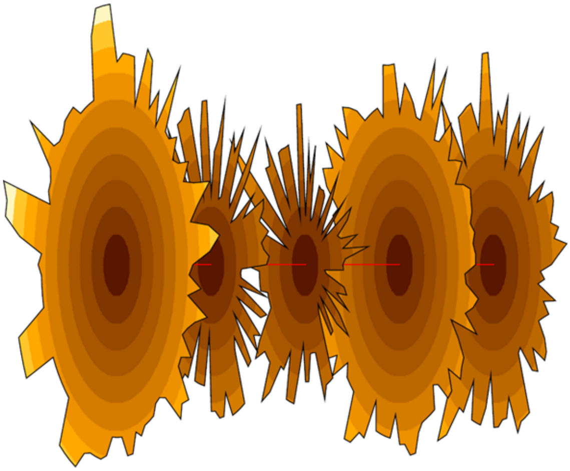


Fig. 7.63: Parallel glyphs are used to visualize five different variables over time. Each radial glyph shows a single variable over time whereas each point on the outside of a glyph corresponds to a data value measured at a specific point in time. Differently colored rings assist in comparing data values.
Source: Fanea et al. (2005), © 2005 IEEE. Used with permission.

Multivariate time-series can be visualized as parallel glyphs. Fanea et al. (2005) synthesized this technique as a combination of parallel coordinates and star glyphs. The visualization uses multiple star glyphs, each of which consists of as many radially arranged spikes as there are time points in the data. The length of a spike corresponds to the data value measured at the spike's associated time point. The tips of subsequent spikes are connected via a polyline, effectively creating a polygonal shape, which visualizes the data of one variable in a radial fashion. As an alternative representation to the spikes, the shape can be filled with differently colored rings to make the data values easier to compare. Multiple such star glyphs are generated, one for each variable of the dataset. These glyphs are then arranged in a three-dimensional space along a shared axis in a parallel fashion. To assist users in identifying correlations among variables, polylines can be used to connect the same time step along the glyphs. In order to avoid clutter, it is possible to restrict this feature to a user-selected number of time steps. The technique offers various ways of manipulating the display, including switching the role of variables and data records, rotation and zooming in the 3D presentation space, and adjustment of colors.

References

Fanea, E., Carpendale, M. S. T., and Isenberg, T. (2005). An Interactive 3D Integration of Parallel Coordinates and Star Glyphs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 149–156, Los Alamitos, CA, USA. IEEE Computer Society.

Worm Plots

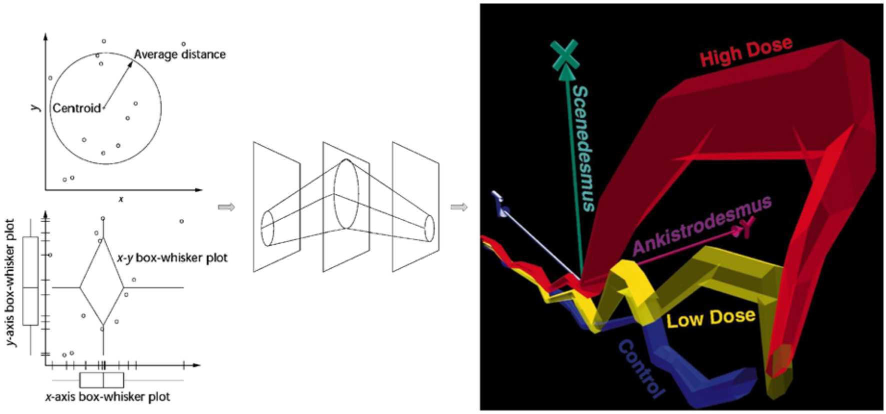


Fig. 7.64: Worm plots are generated by creating visual abstractions of point groups at multiple time steps and by assembling a 3D surface that resembles a worm. The worm plot on the right shows three groups (control, high dose, low dose) of data points of toxicology experiments plotted against the two variables Scenedesmus and Ankistrodesmus.
Source: Matthews and Roze (1997), © 1997 IEEE. Used with permission.

Worm plots have been developed by Matthews and Roze (1997) to help scientists gain qualitative insights into the temporal development of groups of points in scatter plots. The initial step necessary to construct a worm plot is generating a visual abstraction of multiple points. One way to do this is to compute the centroid of a group of points and the average distance of points to the centroid. The visual abstraction is then a circle with a radius equal to the average distance and located at the centroid. Alternatively, a 2D generalization of box-whisker plots can be used to form a diamond-shaped visual abstraction. Such abstractions are computed for each time step (i.e., each scatter plot). Subsequently, a three-dimensional surface (worm) is assembled from the visual abstractions of each group. This procedure is illustrated in the left part of Figure 7.64. Presented in an interactively manipulatable virtual world, worm plots allow users not only to see where in the variable space point groups are located, but also to discern the compactness of point groups, and to understand the development of these characteristics over time.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Matthews, G. and Roze, M. (1997). Worm Plots. *IEEE Computer Graphics and Applications*, 17(6):17–20.

Software Evolution Analysis

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

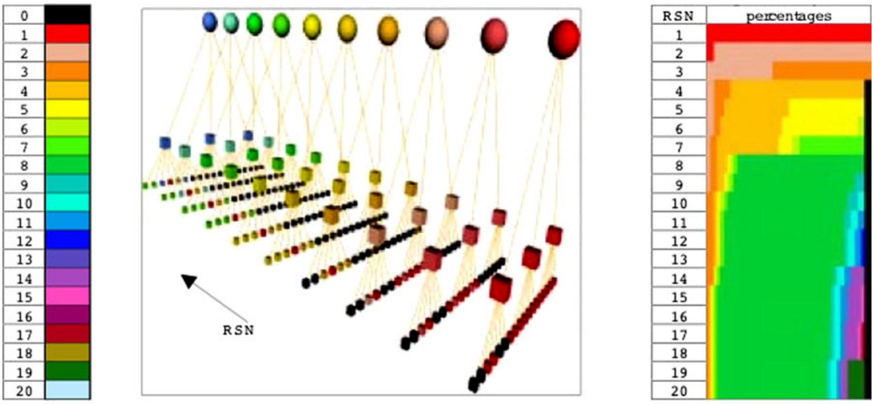


Fig. 7.65: 3D visualization to analyze software systems and product families. Left: color legend for different versions; center: hierarchical decomposition by modules, packages, and files (color represents current version); right: individual file where each row represents a single version colored by percentage of code originating from a particular (previous) version.
Source: Gall et al. (1999), © 1999 IEEE. Used with permission.

The software evolution analysis technique by Gall et al. (1999) uses 3D visualization to analyze software systems or product families respectively. The information is decomposed hierarchically into modules, packages, and files or similar concepts. This hierarchy is depicted as a three dimensional tree structure in which the leaf nodes represent individual files. Multiple such trees are aligned in layers in the 3D space, with one layer for each revision of the software. Color is used to distinguish different versions and to show changes over time. Furthermore, individual files might be inspected in more detail to explore the evolution of changes over time using proportionally colored version lines (see Figure 7.65, right). This way, patterns are formed that can be used to identify, for example, stable parts of a system, frequently changed parts, similarities, and more.

References

Gall, H., Jazayeri, M., and Riva, C. (1999). Visualizing Software Release Histories: The Use of Color and Third Dimension. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 99–108, Los Alamitos, CA, USA. IEEE Computer Society.

InfoBUG

data
frame of reference: **abstract**
variables: **multivariate**

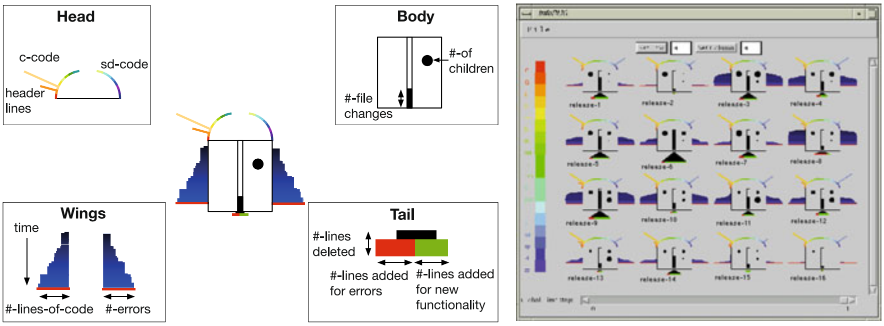


Fig. 7.66: The InfoBUG encodes data about software to the wings, head, tail, and body of a glyph (left). The wings show lines of code and number of errors over time, whereas body, head, and tail show further data for a selected time point. A small multiples view can be used to compare different software releases over time (right).
Source: Chuah and Eick (1998), © 1998 IEEE. Used with permission.

time
arrangement: **linear**
time primitives: **instant**

Chuah and Eick (1998) developed InfoBUG for visualizing changes in software projects. The InfoBUG is an information-rich graphic that combines a multitude of different heterogeneous data values. The glyph resembles an insect with wings, head, tail, and body. The different parts of the glyph are used to represent four different classes of information about software projects (see Figure 7.66, left): code lines and errors (wings), types of code (head), added and deleted lines of code (tail), and number of file changes and children (body). The wings represent the lines of code (left wing) and the number of errors (right wing) over time as vertical silhouette graphs. While the wings show data over time, the other parts of the glyph show only the data of a user-selected time point, which is indicated as a red line at the wings. Antennas on the InfoBUG's head represent different types of code, where color indicates the type of code and the relative sizes of different types are encoded by antenna length. The bug's tail represents the number of deleted and added lines. Finally, the InfoBUG's body visualizes information about the number of altered files via a bar in the middle of the body and the number of child objects via filled circles. Small multiples (↪ p. 236) can be used to compare the different releases of a software product over time (see Figure 7.66, right). Furthermore, the representation can be animated to follow the course of time.

vis
mapping: **static, dynamic**
dimensionality: **2D**

References

Chuah, M. C. and Eick, S. G. (1998). Information Rich Glyphs for Software Management Data. *IEEE Computer Graphics and Applications*, 18(4):24–29.

Gravi++

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static, dynamic**
dimensionality: **2D**

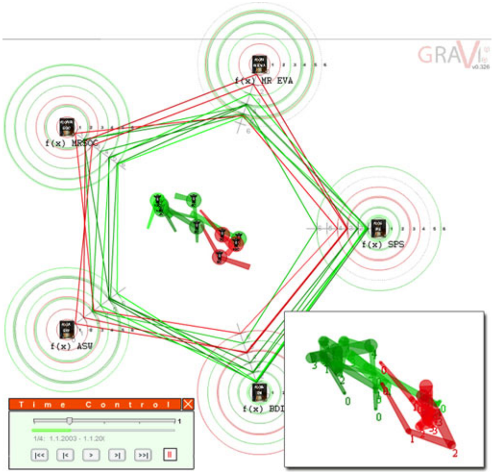


Fig. 7.67: Patient icons in the middle of the display are positioned relative to the surrounding parameters (in this case items of a questionnaire) following a spring-based model. Individual answers to questionnaire items are shown as concentric rings and star glyphs that show a set of answers as polygonal line might be displayed. The user can step through time manually or can use animation, which can be steered via the control panel on the lower left. Furthermore, traces might be displayed that convey information about the evolution of values over time as shown in detail on the lower right.
Source: Image courtesy of Klaus Hinum.

Hinum et al. (2005) designed Gravi++ to find predictors for the treatment planning of anorexic girls. It represents patients and data gathered from questionnaires during treatment over the course of several weeks or months. Patients are represented by icons that are laid according to a spring-based model relative to the surrounding icons that represent items of a questionnaire. This leads to the formation of clusters of persons who gave similar answers. To visualize the changing values over time, animation is used. The position of each person's icon changes over time, making it possible to trace, compare, and analyze the changing values. Alternatively, the change over time can be represented by traces. The size and path of the person's icon is shown corresponding to all time steps or only to a restricted subset like the previous and the next time step. To visualize the exact values of each question, rings around the question's icon can be drawn and star glyphs might be shown.

References

Hinum, K., Miksch, S., Aigner, W., Ohmann, S., Popow, C., Pohl, M., and Rester, M. (2005). Gravi++: Interactive Information Visualization to Explore Highly Structured Temporal Data. *Journal of Universal Computer Science*, 11(11):1792–1805.

CircleView

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static, dynamic**
dimensionality: **2D**

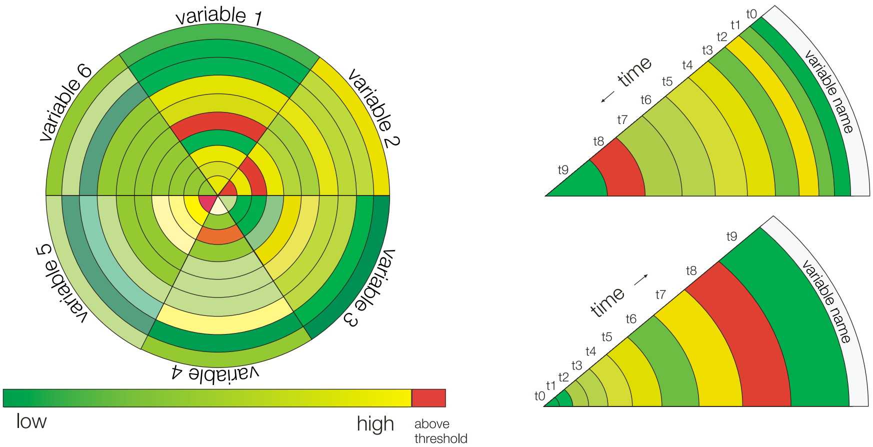


Fig. 7.68: Multiple variables are shown as segments of a circle. Each segment is further subdivided into time slots that represent data values using color. Left: six variables over ten time steps; right: different time axes arrangements (outside–in vs. inside–out) and space assignment that emphasizes more recent values.
Source: Adapted from Keim et al. (2004).

Keim et al. (2004) developed CircleView for visualizing multivariate streaming data as well as static historical data. Its basic idea is to divide a circle into a number of segments, each representing one variable. The segments are further divided into slots covering periods of time, and color shows the (aggregated) data value for the corresponding interval. Thus, time is mapped linearly along the segments. The user can interactively adjust the number of time slots, the time span per slot, different layouts for the time axis, and might emphasize more recent slots by assigning increasingly more space to them (see Figure 7.68, right). Since the order of segments is important for the visual appearance and comparison, it can be adjusted by the user or set automatically using similarity measures. For streaming data the segments of the circle are shifted automatically from the center to the edge (or vice versa). Keim and Schneidewind (2005) also presented a multi-resolution approach on top of CircleView, where time slots for coarser granularities are shown besides detail values.

References

Keim, D. A. and Schneidewind, J. (2005). Scalable Visual Data Exploration of Large Data Sets via MultiResolution. *Journal of Universal Computer Science*, 11(11):1766–1779.

Keim, D. A., Schneidewind, J., and Sips, M. (2004). CircleView: A New Approach for Visualizing Time-Related Multidimensional Data Sets. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 179–182, New York, NY, USA. ACM Press.

Trendalyzer, Animated Scatter Plot

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

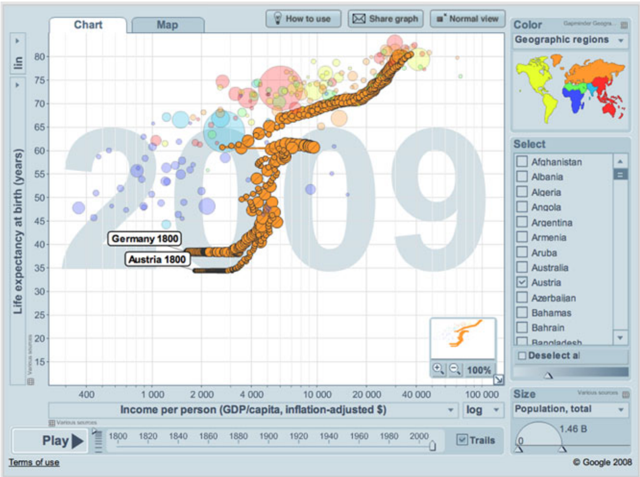


Fig. 7.69: Two data variables are mapped to the horizontal and vertical axes, symbol size represents a third variable, and animation is used to step through time. Additionally, trails are activated for the selected countries, Austria and Germany, which help to preserve the path of a variable through time.
Source: Generated with Trendalyzer with permission of the Gapminder Foundation.

Trendalyzer by Gapminder Foundation (2010) is an interactive visualization and presentation tool that is based on scatter plots. In contrast to point plots (↪ p. 152) where time is mapped on the horizontal or vertical axis, animation is used to represent time. Hence, two data variables are mapped onto the axes of the Cartesian coordinate system and animation is used to step through time. The size of a dot represents a third variable and color is used for distinguishing groups. The animation can be controlled via a time slider, a play/pause button, and a slider for adjusting animation speed. Furthermore, trails might be displayed, which help to preserve the path of a variable through time. This means that dots stay visible and are connected over time. Robertson et al. (2008) evaluated the use of animation in conveying trends over time and compared Trendalyzer with a modified version of trails, and small multiples (↪ p. 236). The results show that animation is both slower and less accurate than the other representations but is well suited as a presentation aid.

References

Gapminder Foundation (2010). Gapminder Trendalyzer. URL, http://www.gapminder.org/world/. Retrieved Feb., 2011.

Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J. (2008). Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1325–1332.

TimeRider

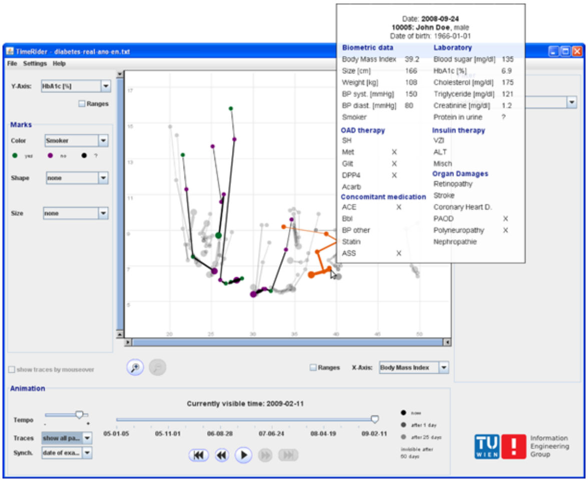


Fig. 7.70: Patients are represented as marks in a scatter plot that can be animated over time. Body mass index is mapped to the horizontal axis, HbA1c to the vertical axis, and mark color shows whether a patient smoked. Time controls for animation and synchronization settings are visible at the bottom. Additionally, traces are displayed that connect values over time. A detail-on-demand window showing further patient data is displayed when hovering over a patient mark.
Source: Generated with the TimeRider software.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **2D**

TimeRider by Rind et al. (2011) is an enhanced animated scatter plot (↪ p. 220) for exploring multivariate trends in cohorts of diabetes patients. The enhancements tackle three challenges of medical data: irregular sampling, data wear (i.e., decreasing validity over time), and patient records covering different portions of time. Animation of irregularly sampled data is achieved via interpolation of individual values along a linear trajectory. To account for data wear and to maintain temporal context, transparency and traces are used to enrich the visual encoding of time. For comparing patient histories that cover different portions of time, TimeRider provides four synchronization modes: by calendar date, patient age, start of treatment, and end of treatment. To take better advantage of animation, TimeRider is highly interactive; apart from common interactions to select, pan, zoom, filter, and show details on demand, the user can change the visual mapping of axes, color, shape, and size (see Figure 7.70, left). Other task-specific features are value ranges that can be highlighted in the background of the scatter plot and dynamic queries on data variables.

References

Rind, A., Aigner, W., Miksch, S., Wiltner, S., Pohl, M., Drexler, F., Neubauer, B., and Suchy, N. (2011). Visually Exploring Multivariate Trends in Patient Cohorts using Animated Scatter Plots. In *Proceedings of the International Conference on Human-Computer Interaction (HCI-I)*, Berlin, Germany. Springer. To appear.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **dynamic**
dimensionality: **2D**

Process Visualization

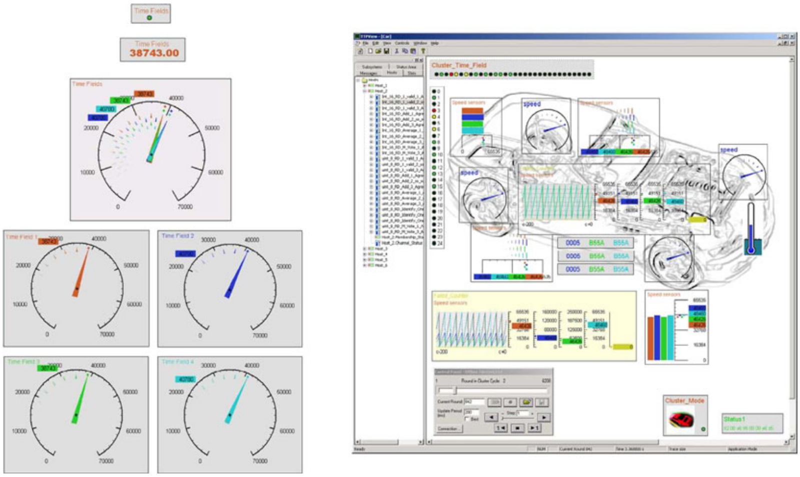


Fig. 7.71: Process visualization can be supported by providing virtual instruments at different levels of detail (left). This helps users to stay focused on important variables of an automotive process, while less relevant information is presented at a higher level of abstraction only (right). *Source: Matković et al. (2002), © 2002 IEEE. Used with permission.*

Process visualization, for instance in automotive environments, has to deal with a multitude of time-varying input variables to be monitored. Matković et al. (2002) suggest a focus+context approach to help users keep track of the important changes of a process. The key idea is to provide virtual instruments that represent monitored variables at different levels of detail. Instruments representing focused variables provide more detailed information, for example, a brief view on a variable's history, which is not possible with classic gauges. On the other hand, less relevant variables are visualized using heavily abstracted virtual instruments that might show just the numeric value or even only a colored dot. Multiple such instruments are arranged in a virtual environment that is used as visual reference for the monitoring scenario. Focus and context within the environment can change dynamically during monitoring, either upon detection of certain events in the data or via user interaction. The approach of Matković et al. (2002) is an excellent example of visualization of dynamic temporal data (see Section 3.3).

References

Matković, K., Hauser, H., Sainitzer, R., and Gröller, E. (2002). Process Visualization with Levels of Detail. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 67–70, Los Alamitos, CA, USA. IEEE Computer Society.

Flocking Boids

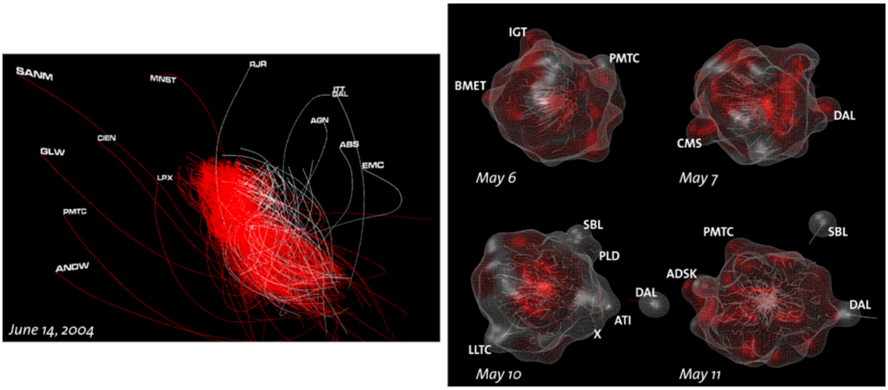


Fig. 7.72: Stock market data are represented as flocking boids that move in a three-dimensional presentation space. Left: boids leaving the flock indicate that the corresponding stock price behaves differently than the majority of prices; right: implicit surfaces surrounding boids help users to recognize the spatial structure of the flock.
Source: Vande Moere (2004), © 2004 IEEE. Used with permission.

Stock market data change dynamically during the day as prices are constantly updated. Vande Moere (2004) proposes to visualize such data by means of information flocking boids. The term boids borrows from the simulation of birds (bird objects = boids) in flocks. In order to visualize stock market prices, each stock is considered to be a boid with an initially random position in a 3D presentation space. Upon arrival of new data, boid positions are updated dynamically according to several rules. These rules attempt to avoid collisions of boids, to move boids at the same speed as their neighbors in the flock, to move boids toward the flock's center, to keep similar boids close to each other, and to let boids stay away from boids that are dissimilar. The visual representation is inherently dynamic and aims at the users' capability to perceive emergence of patterns as the visualization updates. To this end, boids and corresponding traces are visualized as animated curves, as shown on the left in Figure 7.72. This 3D visual representation is enhanced by enclosing boids within implicit surfaces, which help users recognize the spatial structure of the flock (see Figure 7.72, right). The flocking boids visualization can be useful for detecting various patterns in the data such as the emergence of clusters, the separation of boids from the main flock, or a general chaotic behavior of boids.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **dynamic**
dimensionality: **3D**

References

Vande Moere, A. (2004). Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 97–104, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **abstract**
variables: **multivariate**

Time Line Browser

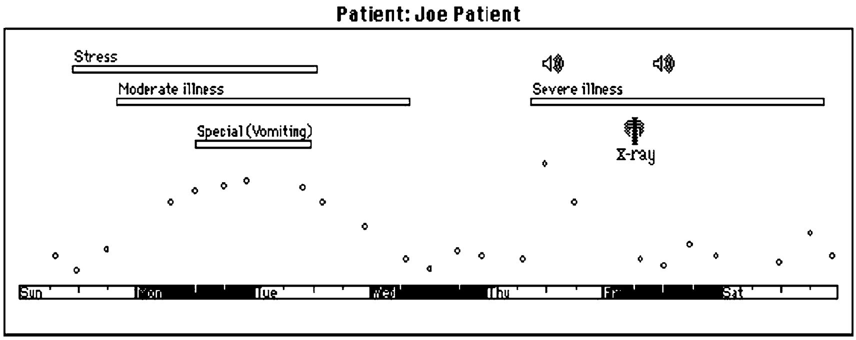


Fig. 7.73: Heterogeneous patient information is visualized along a common horizontal time axis. Intervals are displayed as labeled bars and events are displayed as icons. The small circles form a point plot that shows the patient's blood glucose over time.
Source: Cousins and Kahn (1991), © 1991 Elsevier. Used with permission.

time

arrangement: **linear**
time primitives: **instant, interval**

Cousins and Kahn (1991) developed the time line browser for visualizing heterogeneous time-oriented data. The time line browser integrates qualitative and quantitative data as well as instant and interval data into a single coherent view. To this end, Cousins and Kahn (1991) distinguish simple events, complex events, and intervals. Simple events are represented as small circles, whereas complex events are shown as icons. Bars are used to indicate location and duration of intervals. These depictions are aligned with respect to a common horizontal time axis (↪ p. 166), where textual labels might be used to display further details. In addition to the visualization, a formal system for timeline elements and timeline operations has been developed. It defines five basic operations (i.e, slice, filter, overlay, add, new) for manipulating timelines and also supports composite operations. These operations are useful for addressing the issues of different temporal granularities and the calendar mapping problem.

vis

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

References

Cousins, S. B. and Kahn, M. G. (1991). The Visual Display of Temporal Information. *Artificial Intelligence in Medicine*, 3(6):341–357.

LifeLines

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

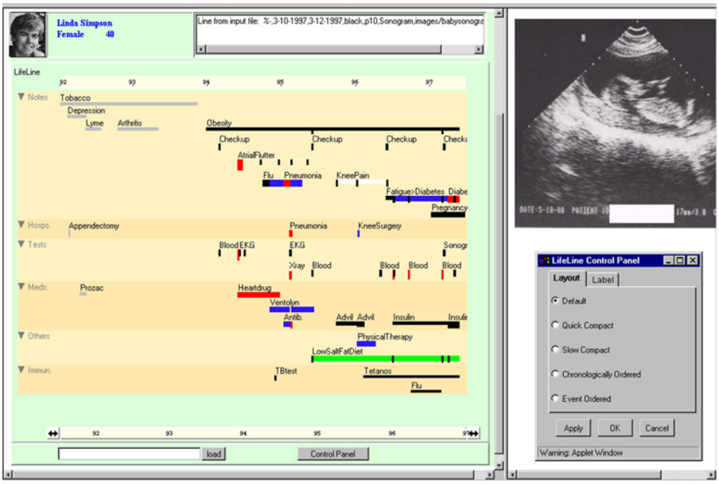


Fig. 7.74: Horizontal bars are used to show the temporal location and duration of health-related incidents. The example shows several facets of patient information and an additional linked sonogram on the right.
Source: Image courtesy of Catherine Plaisant and University of Maryland Human-Computer Interaction Lab.

A simple and intuitive way of depicting incidents is by drawing a horizontal line on a time scale for the time span the incident took. This form of visualization is called timeline (↪ p. 166). Plaisant et al. (1998) apply and extend this concept for visualizing health-related incidents in personal histories and patient records. Consequently, they call their approach LifeLines. Horizontal bars are used to show the temporal location and duration of incidents, treatments, or rehabilitation. Additional information can be encoded via the height as well as the color of individual bars. In order to structure the displayed information in groups, so-called facets are introduced. Multiple such facets are stacked vertically. Depending on the information sought by the user, facets can be expanded and collapsed. When collapsed, only a very small and geometrically as well as semantically down-scaled visual representation without textual labels is shown. When expanded, a facet shows full detail. External information related to certain incidents might be provided on demand in a linked view, as for example x-ray images or sonograms.

References

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., and Shneiderman, B. (1998). LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, pages 76–80, Bethesda, MD, USA. American Medical Informatic Association (AMIA).

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

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

PatternFinder

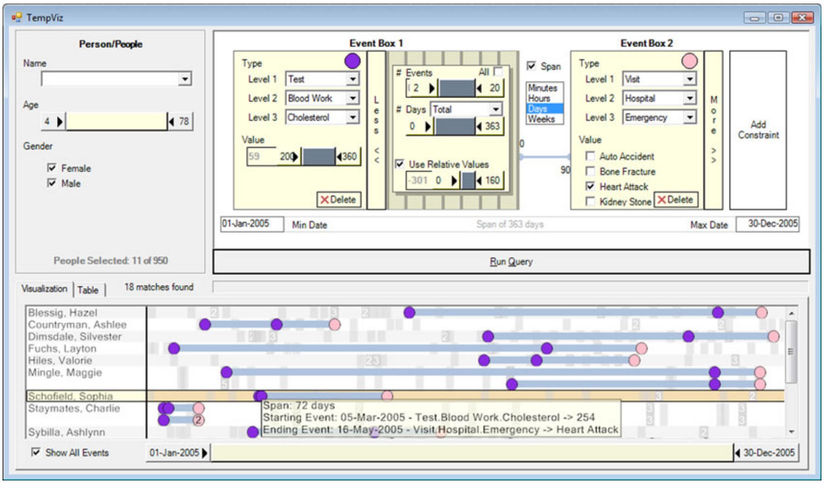


Fig. 7.75: The query formulated in the visual interface (top) relates a cholesterol test to a subsequent emergency visit in a hospital. The resulting visualization (bottom) shows several patient records that match with the query as vertically stacked ball-and-chain representations.
Source: Image courtesy of Jerry Alan Fails.

PatternFinder by Fails et al. (2006) is used for constructing queries to find temporal patterns in medical record databases. The temporal patterns consist of events that are associated with data, and time spans that separate events. Users formulate queries by imposing constraints on events and time spans. Events can be selected from a hierarchically structured vocabulary and constraints for associated variables can be specified in a visual interface along with temporal constraints. This way, users can build queries for the existence of events (e.g., persons with heart attack), temporally ordered events (e.g., heart attack followed by stroke), temporally ordered value changes (e.g., BMI of 25 or higher followed by BMI of 20 or lower), and trends over time (e.g., BMI decreasing). Event sequences might not only be specified in terms of temporal order, but also in terms of temporal distance (e.g., time span of 28 days or less between heart attack and stroke). Moreover, all of the mentioned query types can also be combined. For visualizing query results, a so-called ball-and-chain representation is used: results are shown as vertically stacked timelines, where colored circles represent matched events and bars stand for matched time spans.

References

Fails, J., Karlson, A., Shahamat, L., and Shneiderman, B. (2006). A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 167–174, Los Alamitos, CA, USA. IEEE Computer Society.

Continuum

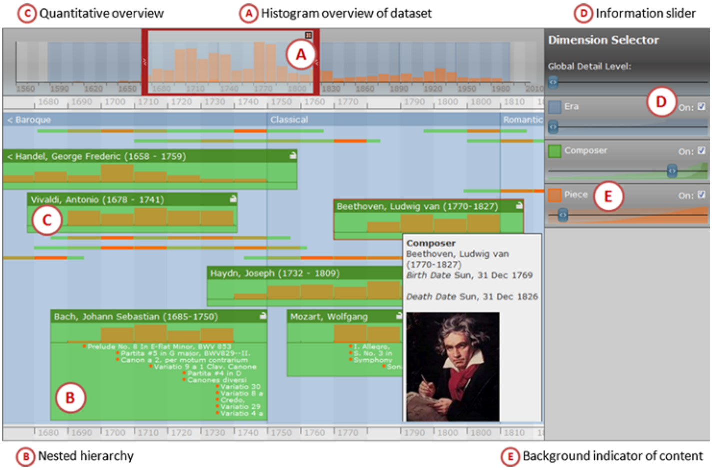


Fig. 7.76: Continuum showing a music dataset with the three variables era, composer, and piece. Scalable histograms provide a complete representation of the data and quantify the focal data item. Top-left: limeline overview; bottom-left: timeline detail view; right: dimension selector panel. *Source: Image courtesy of Paul André.*

Collections of small events often constitute larger, more complex events, like for example talks at conferences or legs of a race. Moreover, events might also be related to other events at other points in time (e.g., a paper written at some point in time and referenced later). Continuum by André et al. (2007) is a timeline visualization tool to represent large amounts of hierarchically structured temporal data and their relationships. It addresses the three problems of scale, hierarchy, and relationships by using scalable histogram overviews, flattening high-dimensional data into dynamically adjustable hierarchies, and arching connection lines for representing non-hierarchical relationships. The interface consists of three main panels that show overview, detail, and the dimension configuration. The timeline overview always represents the complete timespan of the dataset using scalable histograms where the vertical axis quantifies the user-selected focal data item. The timeline detail view shows hierarchical relationships as nested elements and applies semantic zooming depending on the amount of information to be displayed. With the dimension selector, users can interactively control the hierarchical buildup and the level of detail to be shown.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

References

André, P., Wilson, M. L., Russell, A., Smith, D. A., Owens, A., and schraefel, m.c. (2007). Continuum: Designing Timelines for Hierarchies, Relationships and Scale. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 101–110, New York, NY, USA. ACM Press.

EventRiver

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

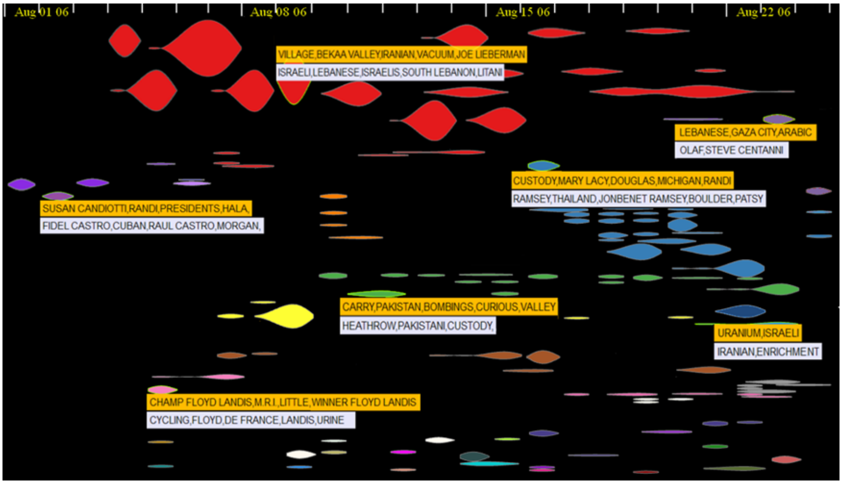


Fig. 7.77: CNN news data from August 2006. Event bubbles flow in a horizontal river of time, where important events are highlighted in red in the top part of the river.
Source: Luo et al. (2011). © 2011, IEEE. Used with permission.

Text collections such as news corpora or email archives often contain temporal references, which embed the text's information into a temporal context. Luo et al. (2011) describe a technique, called EventRiver, for exploring such text collections interactively in terms of important events and the stories that these events constitute. In a first phase, events are extracted from the data using a number of analytical steps, including keyword identification and temporal locality clustering. This phase yields a set of events which are characterized by their position in time, by their duration, and by several other measures (e.g., temporal influence, strength, co-strength). The visual design of EventRiver is based on so-called event bubbles that flow in a horizontal river of time (i.e., along a horizontal time axis). The bubbles are placed horizontally where events are located in time. A bubble's shape illustrates how an event has emerged and disappeared over time. Colors and the bubbles' vertical positions in the river are chosen so as to highlight important interconnected events that constitute long term stories in the text documents. While tooltip labels show the important keywords of events, document details are provided on demand in separate views. Analysts can adjust the EventRiver by using various interaction techniques including dynamic filtering, semantic and temporal zooming, and manual relocation of event bubbles.

References

Luo, D., Yang, J., Krstajic, M., Ribarsky, W., and Keim, D. (2011). EventRiver: Visually Exploring Text Collections With Temporal References. *IEEE Transactions on Visualization and Computer Graphics*. To appear.

FacetZoom

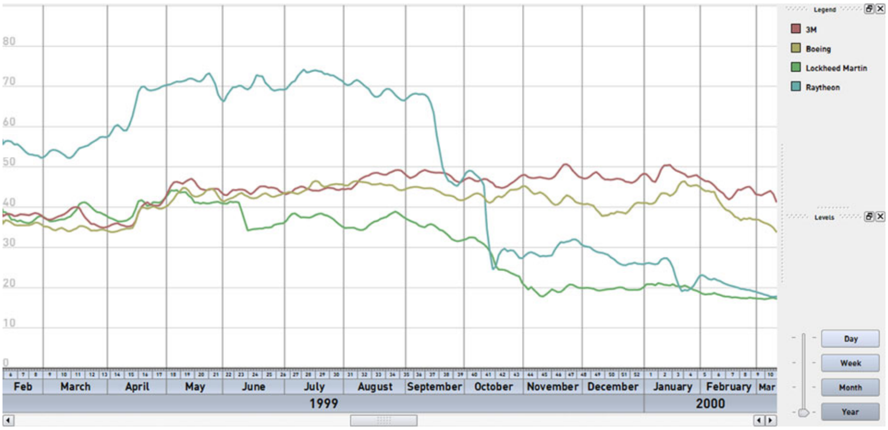


Fig. 7.78: The hierarchical structure of time is shown as an interactive horizontal time axis widget that has a data view attached to it, in this case a visualization of stock market data.
Source: Image courtesy of Raimund Dachselt.

FacetZoom is a technique that enables users to navigate hierarchically structured information spaces (see Dachselt et al., 2008). The hierarchical structure of time is a natural match for this technique. What Dachselt and Weiland (2006) originally called TimeZoom is a visual navigation aid for time-oriented data. The basic idea is to display a horizontal time axis that represents different levels of temporal granularity as stacked bars (e.g., decades, years, months, weeks, days). The time axis is an interactive widget that can be used to access data from different parts of the time domain at different levels of abstraction. In addition to continuous zooming and panning via mouse, it is also possible to simply select discrete intervals from the time axis. Depending on the user's selection, the time axis display is altered to accommodate the selected part of the time axis with more display space. Accordingly, the data view, which is attached to the time axis, can use the extra space to represent more data items in greater detail. While the actual mapping of time is static, the navigation steps of the user, including the visual adjustment of the time axis, are smoothly animated.

References

Dachselt, R., Frisch, M., and Weiland, M. (2008). FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1353–1356, New York, NY, USA. ACM Press.

Dachselt, R. and Weiland, M. (2006). TimeZoom: A Flexible Detail and Context Timeline. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 682–687, New York, NY, USA. ACM Press. Extended Abstracts.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

Midgaard

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

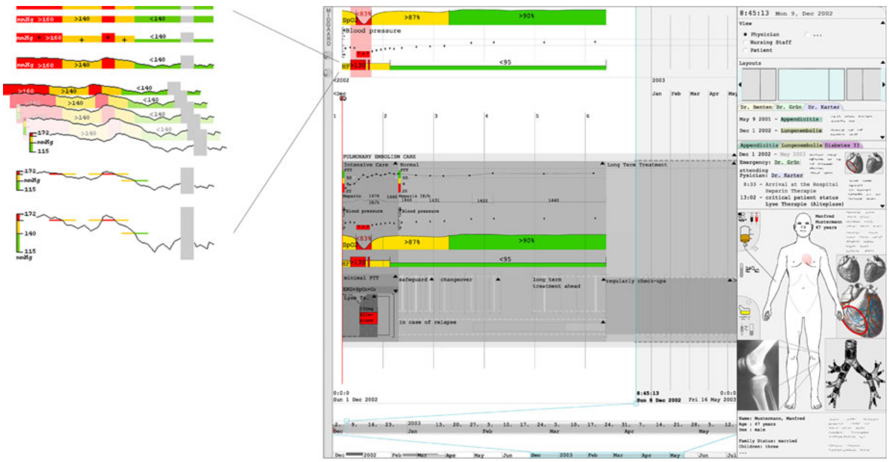


Fig. 7.79: Midgaard integrates the display of time-oriented patient data and treatment plans. Right: main user interface showing different measurements (e.g., blood gas measurements, blood pressure) using line plots, their corresponding temporal abstractions using color, and treatment plans as well as additional patient information on the right, and an interactive multi-scale time axis at the bottom; left: different steps of semantic zooming of a time-series from a broad overview (top) to a detailed view with details of fine structures (bottom).
Source: Authors.

Several tightly integrated visualization techniques have been developed in the Midgaard project by Bade et al. (2004) to enhance the understanding of heterogeneous patient data. To support the user in exploring the data and to capture as much qualitative and quantitative information as possible on a limited display space, Midgaard supports different levels of abstractions for time-oriented data (see Section 6.3). Switching between these levels is achieved via a smoothly integrated semantic zoom functionality (see Figure 7.79, left). These methods were designed to allow users to interact with data and time. Navigation in time is done using three linked time axes (see Figure 7.79, bottom-right). The first one (bottom) provides a fixed overview of the underlying time interval covering its full range. Selecting a subrange in that time axis defines the temporal bounds for the main display area and the second (middle) time axis. Selecting a further subrange in the middle time axis defines detail and surrounding context areas in time. By interactively adjusting the subranges, users can easily zoom and pan in time.

References

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 105–112, New York, NY, USA. ACM Press.

VisuExplore

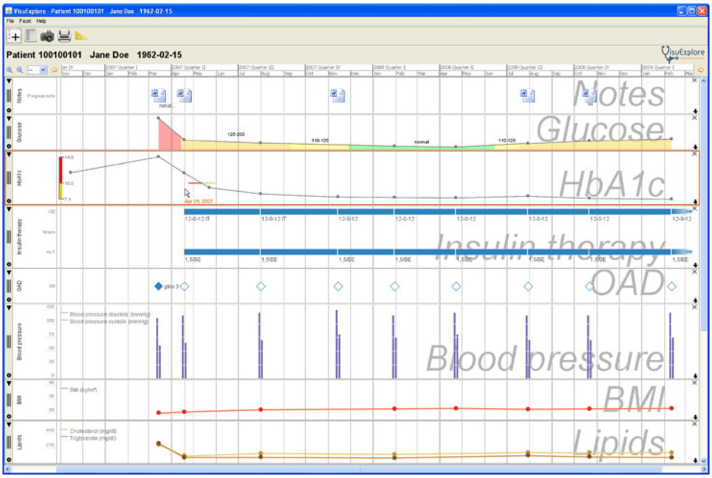


Fig. 7.80: Visualization of heterogeneous medical parameters of a diabetes patient. Beneath a panel that shows patient master data, eight visualization views show progress notes in a document browser, glucose and HbA1c as line plots with semantic zoom, insulin therapy as timelines, OAD as event chart, blood pressure as bar graphs, and BMI as well as lipids as line plots (top to bottom). *Source: Generated with the VisuExplore software.*

VisuExplore by Rind et al. (2010) is an interactive visualization system for exploring a heterogeneous set of medical parameters over time. It uses multiple views along a common horizontal time axis to convey the different medical parameters involved. VisuExplore provides an extensible environment of pluggable visualization techniques and its primary visualization techniques are deliberately kept simple to make them easily usable in medical practice: line plots (↪ p. 153), timeline charts (↪ p. 166), bar graphs (↪ p. 154), event charts, line plots with semantic zoom (see p. 112), and document browsers (see Figure 7.80, top). Furthermore, data might also be presented as textual tables to augment the visual representations. VisuExplore's interactive features allow physicians to get an overview of multiple medical parameters and focus on parts of the data. Users may add, remove, resize, and rearrange visualization views. Additionally, a measurement tool is integrated that makes it possible to determine time spans between user selected points of interest and this works not only within one but also across different views.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

References

Rind, A., Miksch, S., Aigner, W., Turic, T., and Pohl, M. (2010). VisuExplore: Gaining New Medical Insights from Visual Exploration. In Hayes, G. R. and Tan, D. S., editors, *Proceedings of the 1st International Workshop on Interactive Systems in Healthcare (WISH@CHI2010)*, pages 149–152, New York, NY, USA. ACM Press.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

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

KNAVE II

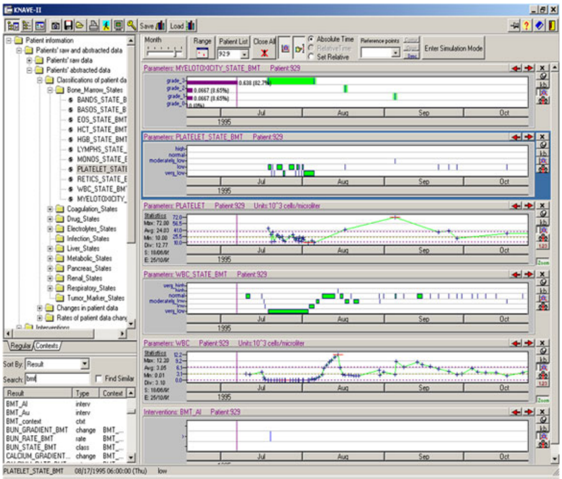


Fig. 7.81: KNAVE II supports the visualization and knowledge-based navigation of patient data and abstractions thereof. Left: tree view of clinical domain ontology for navigation; right: visualization panels for raw data and abstractions; bottom-left: search panel.
Source: Shahar et al. (2006), © 2006 Elsevier. Used with permission.

KNAVE II by Shahar et al. (2006) enables visual browsing and exploring of patient's data (raw measured values and external interventions such as medications). The system focuses mainly on the visual display of temporal abstractions of the data (see Section 6.3) and shows domain-specific concepts and patterns. In order to abstract the raw data, a predefined knowledge base is used that defines three types of interpretations: classification of data (e.g., low – normal – high), change of data (e.g., increasing – decreasing), and rate of change (e.g., slow – fast). Colored timelines (↪ p. 166) depict abstracted intervals as bars where a bar's vertical position within a panel encodes its qualitative value. For example, the second panel (blue frame) in Figure 7.81 shows the platelet state abstracted to the qualitative values: very low, low, moderately low, normal, high (from bottom to top). KNAVE II also allows users to view the raw data as line plots (↪ p. 153). Moreover, statistics for parameters and corresponding abstractions can be superimposed as bar graphs (↪ p. 154) as shown in the topmost panel, or can be given as text labels, as shown in the third and fifth panels. A granularity-based zoom allows users to quickly navigate in time simply by clicking individual granules, for instance the month *Aug*.

References

Shahar, Y., Goren-Bar, D., Boaz, D., and Tahan, G. (2006). Distributed, Intelligent, Interactive Visualization and Exploration of Time-Oriented Clinical Data and their Abstractions. *Artificial Intelligence in Medicine*, 38(2):115–135.

Circos

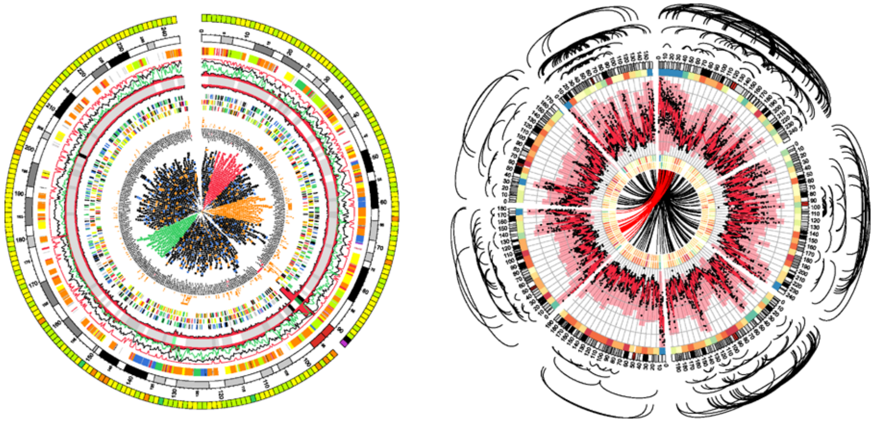


Fig. 7.82: Display of multivariate data using data tracks in a radial layout. Images show human chromosome data using point plots, line plots, tiles, histograms, heatmaps, and connectors that link points on the circle (right).
Source: Images courtesy of Martin Krzywinski.

Circos by Krzywinski et al. (2009) uses a circular design to generate multivariate displays. It uses concentric bands (data tracks) as display areas and is capable of displaying data as point plots (↪ p. 152), line plots (↪ p. 153), histograms, heat maps, tiles, connectors, and text. In this sense, time is mapped circularly to the circumference of data tracks. The configuration of a visual representation is handled via plain-text files where rules that can be defined for each data track filter and format data elements based on position, value or previous formatting. For communicating quantitative data via color, perceptually uniform color schemes based on the work of Brewer (1999) are used. Circos was initially developed for genomics and bioinformatics data to visualize alignments, conservation, and intra- and inter-chromosomal relationships. Relationships between pairs of positions are represented by the use of ribbons that connect elements. In the same way, relational data encoded in tabular formats can be shown. Due to its flexible approach, Circos has also been applied to numerous other application areas, such as urban planning, and has been used for infographics in newspapers and ads to display complex relationships.

data

frame of reference: **abstract**
variables: **multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant, interval**

vis

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

References

Brewer, C. A. (1999). Color Use Guidelines for Data Representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60, Baltimore, MD, USA. American Statistical Association.

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S. J., and Marra, M. A. (2009). Circos: An Information Aesthetic for Comparative Genomics. *Genome Research*, 19(9):1639–1645.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **cyclic**
time primitives: **instant**

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

Kaleidomaps

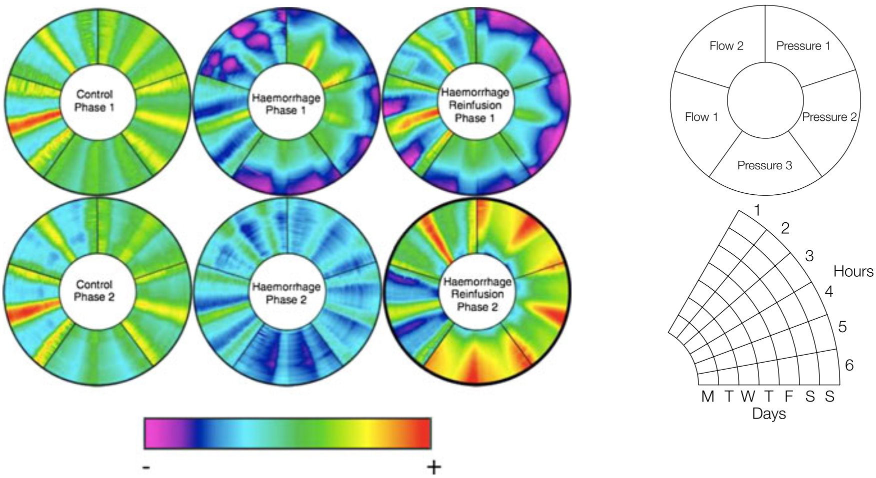


Fig. 7.83: On the left, six kaleidomaps show the morphology of blood pressure and flow waves over two experimental phases. Top-right: illustration of the layout of variables within a kaleidomap; bottom-right: layout of time within a segment.
Source: Bale et al. (2006), © 2006 IEEE. Used with permission.

Kaleidomaps by Bale et al. (2007) visualize multivariate time-series data and the results of wave decomposition techniques using the curvature of a line to alter the detection of possible periodic patterns. The overall idea of kaleidomaps is similar to the rendered output of a kaleidoscope for children, from whence the name comes. A base circle is broken into segments of equal angles for different variables. Each circle segment has two axes representing time, one along the radius and one along the arc of the segment. The data values and categories are represented using color. Due to the circular nature of kaleidomaps, the number of variables in one circle is limited to a maximum of six to eight. Interaction techniques within the kaleidomaps allow an analyst to drill down both in time and frequency domains in order to uncover potential relationships between time, space, and waveform morphologies. Kaleidomaps were developed in the domain of critical care medicine, but case studies have shown their usefulness in other domains as well, like in environment analysis.

References

Bale, K., Chapman, P., Barraclough, N., Purdy, J., Aydin, N., and Dark, P. (2007). Kaleidomaps: A New Technique for the Visualization of Multivariate Time-Series Data. *Information Visualization*, 6(2):155–167.

Bale, K., Chapman, P., Purdy, J., Aydin, N., and Dark, P. (2006). Kaleidomap Visualizations of Cardiovascular Function in Critical Care Medicine. In *Proceedings of International Conference on Medical Information Visualisation - BioMedical Visualisation (MediVis)*, pages 51–58, Los Alamitos, CA, USA. IEEE Computer Society.

Intrusion Detection

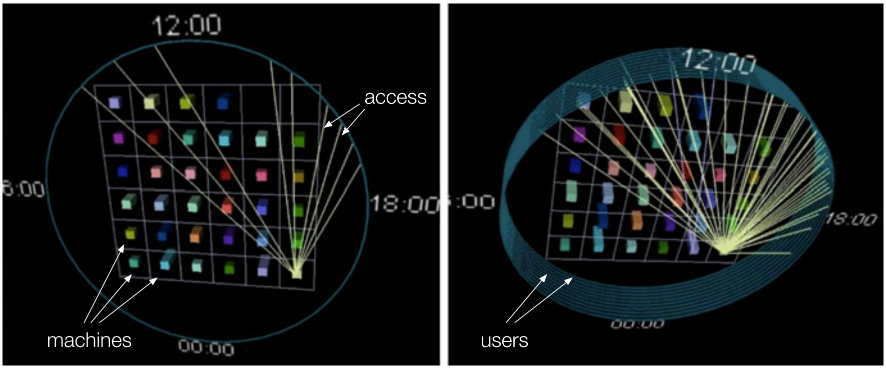


Fig. 7.84: Machines in a network are represented by a matrix of 3D cubes in the center of the display. Time is mapped to the circumference of a circle enclosing the matrix of machines. When a particular machine is accessed, a line is drawn that links the particular machine with a point in time on the circular time axis.
Source: Images courtesy of Kovalan Muniandy.

A 3D visualization technique by Muniandy (2001) helps to analyze user access to computers in a network over time for intrusion detection. The different parameters time, users, machines, and access are mapped onto a 3D cylinder. In Figure 7.84, time is mapped onto the circumference of a circle showing the 24 hours of a day. The units along the circle can be configured to represent either hours, months, or years. Different users are represented by individual cylinder slices that are stacked upon each other and machines are represented as cubes that are arranged in a matrix. Access to a machine by a user is visualized by a line connecting the user slice at the corresponding access time with the accessed machine. This way, certain patterns of network access can easily be spotted visually and suspicious behavior can be revealed. Details-on-demand are displayed when hovering with the mouse over an element of the visualization. To mitigate occlusion, the representation can be zoomed and rotated freely by the user. Moreover, filtering can be applied to remove clutter.

data
frame of reference: **abstract**
variables: **multivariate**

time
arrangement: **cyclic**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Muniandy, K. (2001). Visualizing Time-Related Events for Intrusion Detection. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA. IEEE Computer Society. Late Breaking Hot Topics.

Small Multiples

data

frame of reference: **abstract, spatial**
variables: **uni-, multivariate**

time

arrangement: **linear, cyclic**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **2D, 3D**

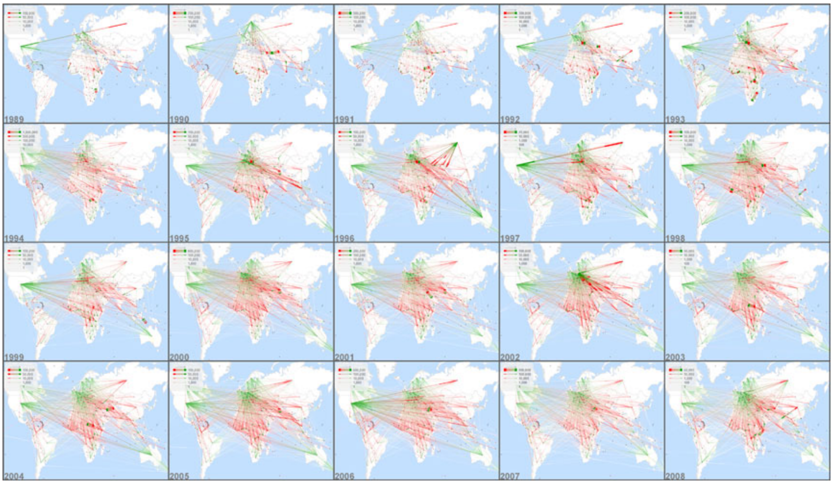


Fig. 7.85: Small multiples showing migration data. Each miniature map visualizes migration of people as links between countries. Green color indicates the origin, red color the destination, and line width the volume of migration.
Source: Generated with the JFlowMap software with permission of Ilya Boyandin.

Small multiples are more a general concept than a specific technique. Tufte (1983, 1990) describes small multiples as a set of miniature visual representations. For time-oriented data, each miniature visualizes a selected time point. The concrete depiction may show a single variable or multiple variables in an abstract or spatial context using a 2D or 3D presentation space. Particularly relevant is the arrangement of the small multiples as it dictates how the time axis is perceived. Linear or circular arrangements can be used, or specific arrangement patterns can be applied to account for different granularities of the time axes. Small multiples provide an overview of the data and allow users to visually compare the data at different time points. Another advantage of small multiples is that the concept can be applied to virtually any existing visualization technique; the only thing to do is to create a thumbnail from an existing visual representation for each time step. Depending on the amount of screen space occupied by each thumbnail, however, the number of representable time steps could be rather moderate. Or, if the images are shrunk to fit more time steps, less details are visible.

References

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.
Tufte, E. R. (1990). *Envisioning Information*. Graphics Press, Cheshire, CT.

EventViewer

data
frame of reference: **abstract, spatial**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

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

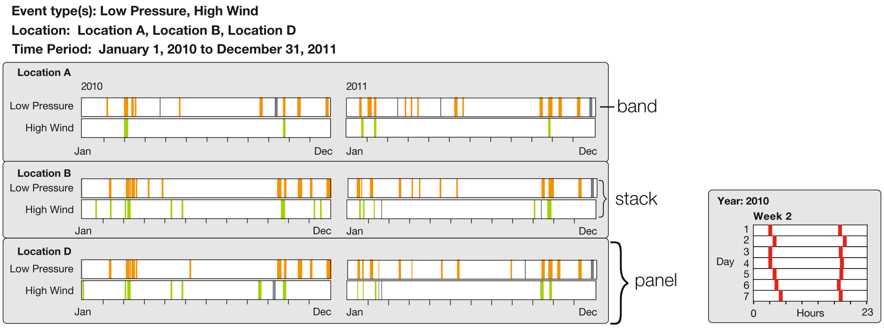


Fig. 7.86: Visual exploration of event data. Spatial, temporal, and thematic dimensions of events can flexibly be assigned to configurations of bands, stacks, and panels. Left: low pressure and high wind events are shown along three locations and two years; right: display configuration to reveal temporal patterns along hours of a day.
Source: Adapted from Beard et al. (2008).

EventViewer by Beard et al. (2008) is a framework that has been developed to visualize and explore spatial, temporal, and thematic dimensions of sensor data. The system supports queries on events that have been extracted from such data and are stored in an events database. The spatial, temporal, and thematic categories of selected events can flexibly be assigned to three kinds of nested display elements called bands, stacks, and panels. Bands are the primary graphic object and act as display container for a set of events. The horizontal dimension of a band represents time and bars within a band represent instances of events. The length of a bar corresponds to the event's duration and color can be used to encode other data values. Furthermore, missing data is shown by using gray bars to make a clear visual distinction to areas without events (shown as empty areas). Stacks consist of event bands that are placed on top of each other and panels are collections of stacks. Each of the three data dimensions space, time, and theme can be modeled along hierarchies or lattices. For time, calendric systems consisting of time granularities like hours, days, weeks, and years are used (see Section 3.1.2). The configuration of display elements are broken down along these hierarchical and lattice structures and form small multiples (↪ p. 236). The assignments can be changed interactively by the user via direct manipulation, thus revealing different kind of patterns as for example periodic patterns, spatial and temporal trends, or event-event relationships.

References

Beard, K., Deese, H., and Pettigrew, N. R. (2008). A Framework for Visualization and Exploration of Events. *Information Visualization*, 7:133–151.

Ring Maps

data
frame of reference: **abstract, spatial**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D, 3D**

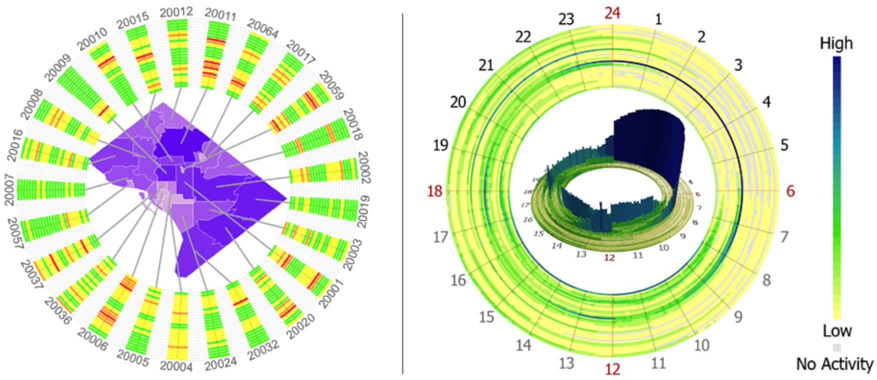


Fig. 7.87: Ring maps representing health related alert levels (green, yellow, orange, red) for various zip code regions during a period of 24 weeks (left) and degree of activity during the course of a day for 96 human activities, one shown per ring (right).
Source: Left: Image courtesy of Guilan Huang. Right: Image courtesy of Jinfeng Zhao.

The basic idea of ring maps is to create multiple differently sized rings, each of which is subdivided into an equal number of ring segments (see Zhao et al., 2008; Huang et al., 2008). The rings and their segments as well as the center area of the overall visual representation can be used in various ways. One can utilize ring maps to visualize spatio-temporal data. To this end, a map is shown in the center and the ring segments of a particular angle are associated with a specific area of the map. This is depicted in the left part of Figure 7.87, where different angles show the data for different zip code regions. A time-series for each region can then be represented by the rings, for instance, by assigning the first series entry to the inner-most ring and the last one to the outer-most ring. The actual data visualization is done by color coding. There are other ways of mapping information to rings and segments. The right part of Figure 7.87 shows an application of ring maps where the hours of the day are mapped to the ring segments and the rings represent different activities a person can be busy with during the course of a day. The degree of activity is encoded by color. This time the center of the display is used to show a complementary 3D representation to assist users in spotting highly active regions.

References

Huang, G., Govoni, S., Choi, J., Hartley, D. M., and Wilson, J. M. (2008). Geovisualizing Data With Ring Maps. *ArcUser*, Winter 2008.

Zhao, J., Forer, P., and Harvey, A. S. (2008). Activities, Ringmaps and Geovisualization of Large Human Movement Fields. *Information Visualization*, 7(3):198–209.

Time-Oriented Polygons on Maps

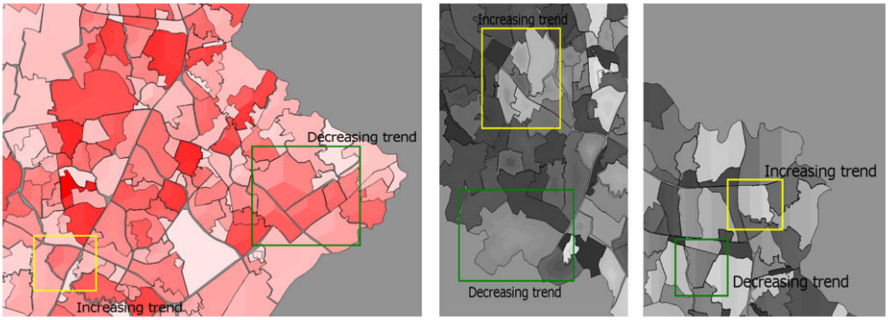


Fig. 7.88: Development of high school population in school districts over the years 2005, 2006, and 2007. Three different layouts to represent change over time of values associated with areas of a map. Left: wedges; center: rings; right: time slices. Data values are shown using color.
Source: Shanbhag et al. (2005), © 2005 IEEE. Used with permission.

Three time-oriented visualization methods are presented by Shanbhag et al. (2005) to analyze and support effective allocation of resources in a spatio-temporal context. Wedges, rings, and time slices are the three basic layouts used to display changes of data values over time on a map. For all three variants, data values and categories are represented using color components (hue, saturation, and brightness). In the layout of the wedges the area of a polygon is partitioned in a clock-like manner into radical sectors equal to the number of time points (see Figure 7.88, left). The ring layout is based on the idea of the concentric rings of a tree trunk where the innermost ring corresponds to the earliest time point and the outermost ring corresponds to the latest time point (see Figure 7.88, center). The time slices layout divides a polygon into vertical slices that are ordered from left to right according the progress in time (see Figure 7.88, right). The wedges, rings, and time slice layouts are applied to polygonal areas of a map. These visualizations were used for example to repartition school districts (seen as planning polygons) where variables such as student population by grade, number of students requiring free meals, and test scores needed to be analyzed to plan the future allocation of resources.

References

Shanbhag, P., Rheingans, P., and desJardins, M. (2005). Temporal Visualization of Planning Polygons for Effcient Partitioning of Geo-Spatial Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 211–218, Los Alamitos, CA, USA. IEEE Computer Society.

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear, cyclic**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **2D, 3D**

Icons on Maps

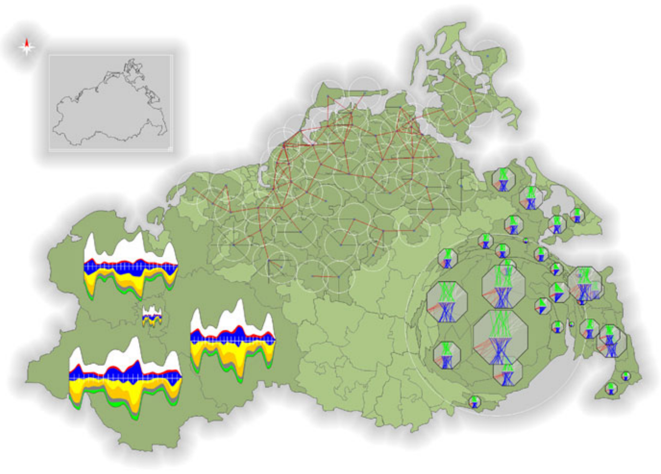


Fig. 7.89: The figure illustrates the embedding of time-representing icons into a map in order to visualize spatio-temporal data. This illustration shows ThemeRiver icons and TimeWheel icons in the left and the right part of the map, respectively. The northern part of the map illustrates a conflict graph as used for local optimization of icon positions.
Source: Generated with the LandVis system.

When time-oriented data additionally contain spatial dependencies, it is necessary to visualize both the temporal aspects and the spatial aspects. A sensible approach to achieving this is to adapt existing solutions. Maps are commonly applied to represent the spatial context of the data. In order to apply existing visualization techniques to represent the temporal context, they must be made compatible with the map display. First and foremost, this implies a reduction in size, which effectively means creating icons from otherwise full-window visual representations. In a second step, it is then possible to place multiple icons on the map, where the data are anchored in space. If there are too many icons on a map, they most likely occlude each other. Therefore, additional methods are applied to resolve occlusions by the global or local optimization of icon positions. The problem of finding suitable icon positions is very much related to the cartographic map labeling problem. Tominski et al. (2003) and Fuchs and Schumann (2004) demonstrate the integration of the ThemeRiver (↪ p. 197) and the TimeWheel (↪ p. 200) into a map display.

References

Fuchs, G. and Schumann, H. (2004). Visualizing Abstract Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 139–144, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2003). Visualisierung zeitlicher Verläufe auf geografischen Karten. In *Kartographische Schriften, Band 7: Visualisierung und Erschließung von Geodaten*, pages 47–57. Kirschbaum Verlag, Bonn, Germany.

Value Flow Map

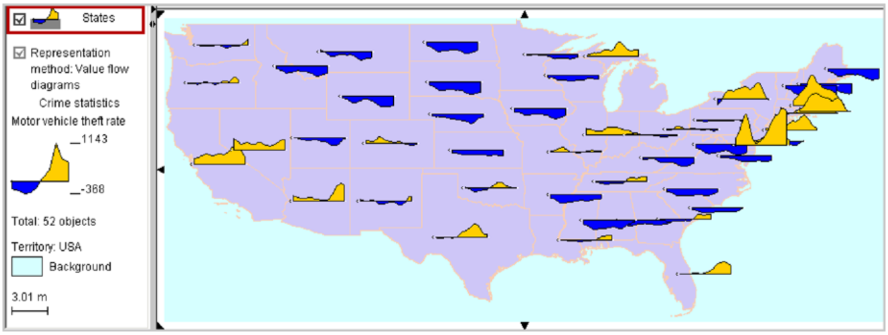


Fig. 7.90: Univariate spatio-temporal data are represented by embedding multiple miniature silhouette graphs into regions of a map. The graphs use a specific encoding where yellow color corresponds to a positive deviation of the variable from the data's mean and blue color indicates a negative deviation.
Source: Image courtesy of Gennady Andrienko.

What Andrienko and Andrienko (2004) call value flow map is a technique to visualize variation in spatio-temporal data. A value flow map shows one miniature silhouette graph (↪ p. 175) for each area of a cartographic map to represent the temporal behavior of one data variable per area. Typically, temporal smoothing is carried out by replacing the values of a point-based time scale with the mean values of an interval-based time scale. In this way, small fluctuations are disregarded and major trends become visible. Moreover, a number of data transformations can be applied to define the mapping of the graphs. An example of such a transformation is to show the variable's deviation from the data's mean, rather than the raw data, that is, data values are replaced by their differences to the mean in order to represent positive and negative variations. This way, the silhouette graphs visualize quite well how the data values flow in time and space (hence value flow map). This is a necessary requirement to enable analysts to detect patterns, and thus to support exploring spatial distributions, comparing data evolution at different locations, as well as finding similarities and outliers.

data
frame of reference: **spatial**
variables: **univariate**

time
arrangement: **linear**
time primitives: **instant**

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

References

Andrienko, N. and Andrienko, G. (2004). Interactive Visual Tools to Explore Spatio-Temporal Variation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, pages 417–420, New York, NY, USA. ACM Press.

data

frame of reference: **spatial**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

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

Flow Map

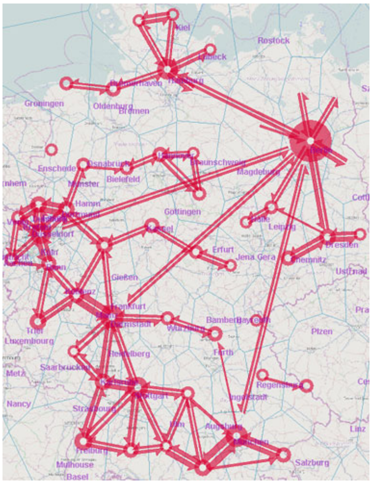


Fig. 7.91: The flow map shows characteristic movements of photographers over time extracted from metadata of more than 590,000 geo-referenced and time-stamped photographs.
Source: Image courtesy of Gennady Andrienko.

Flow maps show movements of objects over time, that is, they show a change of positions over time, rather than a change of data values. Usually, such movements form directed (optionally segmented) trajectories connecting the starting point of a movement and its end point. Such trajectories can be represented visually as more or less complex arrows or curves, where width, color, and other attributes can be used to encode additional information (see Kraak and Ormeling, 2003). A famous example is Minard's flow map of Napoleon's Russian campaign (see Figure 2.8 on page 21). A high number of flows, however, leads to overlapping trajectories and thus to visually cluttered flow maps. In order to represent massive data, flow maps can show abstractions of movements, rather than individual movements (see Andrienko and Andrienko, 2011). To faithfully communicate the underlying data, characteristic movements need to be extracted. First, time points are aggregated to larger time intervals and individual places are substituted with larger regions so as to arrive at abstracted trajectories that show mean trends. Secondly, the trajectories are grouped based on a similarity search, e.g., by applying cluster analysis or self organizing maps (SOM). In this way, places with similar dynamics are merged, and individual trajectories are replaced by trajectories associated with the groups.

References

Andrienko, N. and Andrienko, G. (2011). Spatial Generalization and Aggregation of Massive Movement Data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219.

Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Pearson Education, Harlow, England, 2nd edition.

Time-Varying Hierarchies on Maps

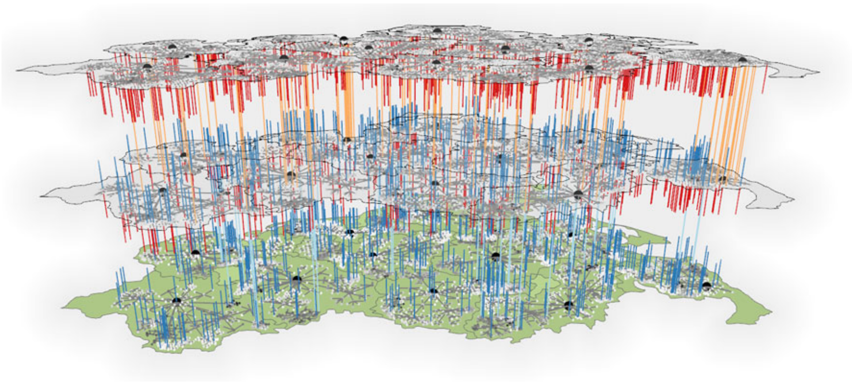


Fig. 7.92: Hierarchy layouts are embedded into areas of a map, where each map layer corresponds to one time step. Colored links and spikes between layers indicate significant changes from one time step to the other.
Source: Generated with the LandVis system.

Hierarchical structures can be found in many application areas. A technique for visualizing hierarchies that change over time in a geo-spatial context is described by Hadlak et al. (2010). This technique follows the idea of using the third dimension of the presentation space to represent the dimension of time, which is analog to the space-time cube approach (↪ p. 245). For a series of time steps, individual map layers are constructed, where each map region shows an embedded hierarchy layout and each node's color visualizes a data value. To facilitate the identification of changes between two layers, visual cues are added. Differently colored links between subsequent layers are used to indicate nodes that have moved or whose attribute values have changed significantly. Significance is determined by a user-selectable threshold. Positive attribute changes are shown as red links and negative changes are shown in blue. Links representing node movements are colored with a shade of gray. Addition or deletion of nodes and edges is indicated by spikes. Spikes that represent deletion leave a layer in the direction of the time axis and are shown in blue. Those that mark addition enter a layer and are shown in red. The layering approach in combination with the described visual cues allows users to compare successive time steps more closely.

data

frame of reference: **spatial**
variables: **univariate**

time

arrangement: **linear**
time primitives: **instant**

vis

mapping: **static**
dimensionality: **3D**

References

Hadlak, S., Tominski, C., and Schumann, H. (2010). Visualization of Attributed Hierarchical Structures in a Spatio-Temporal Context. *International Journal of Geographical Information Science*, 24(10):1497–1513.

VIS-STAMP

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

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

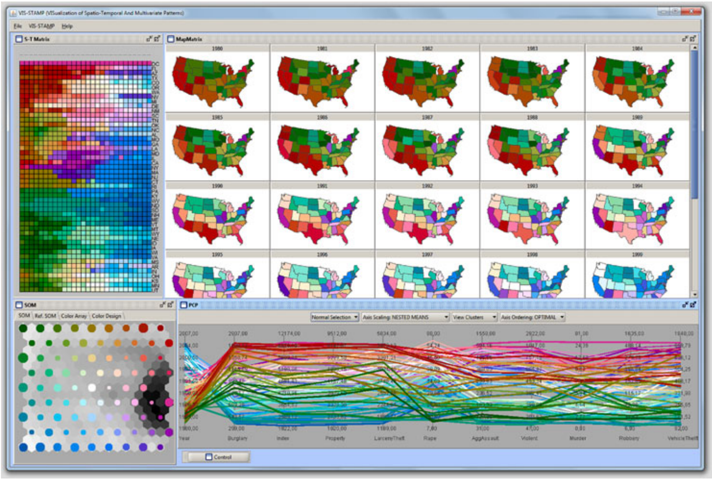


Fig. 7.93: Multiple views show multivariate spatio-temporal crime data that have been clustered by a self-organizing map (SOM). An enhanced color coding schema is used consistently among all views to visualize cluster affiliation.
Source: Generated with the VIS-STAMP system.

Spatio-temporal data can be complex and multi-faceted. Guo et al. (2006) developed a system called VIS-STAMP that integrates computational, visual, and cartographic methods for visual analysis and exploration of such data. At the heart of the system is a self-organizing map (SOM) that is used for multivariate clustering, sorting, and coloring. The visual ensemble comprises a matrix view (top-left in Figure 7.93), a map view (top-right), a parallel coordinates view (bottom-right), and a SOM view (bottom-left). The matrix view's columns represent time points and its rows stand for geographic regions. Cluster affiliation of the matrix cells is visualized by means of an enhanced color coding schema. The color coding is consistent across all views. The map view follows the small multiples approach (↪ p. 236) and shows color-coded map thumbnails, one for each time point. The parallel coordinates view addresses the multivariate character of the data. Finally, the SOM view offers a detailed view and control interface of the underlying self-organizing map. A number of automatic and interactive manipulation techniques (e.g., reordering and sorting) facilitate the data analysis.

References

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474.

Space-Time Cube

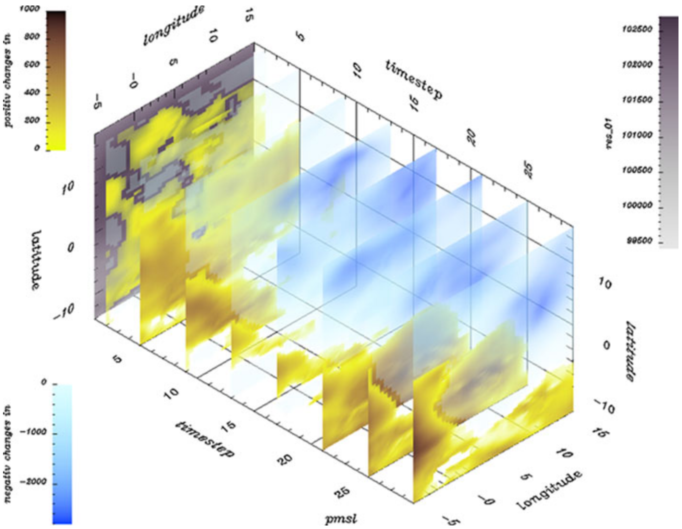


Fig. 7.94: This space-time cube represents two spatial dimensions (latitude and longitude) along the y-axis and the z-axis, and time along the x-axis. Multiple color-coded layers are embedded into the cube to visualize spatio-temporal climate data.
Source: Image courtesy of Thomas Nocke.

A classic concept that combines the visualization of space and time is the space-time cube, which is attributed to the pioneer work of Hägerstrand (1970). The basic idea is to map two spatial dimensions to two axes of a virtual three-dimensional cube and to use the third axis for the mapping of time. The spatial context is often represented as a map that constitutes one face of the space-time cube. The three-dimensional space inside the cube is used to represent spatio-temporal data, where possible visual encodings are manifold. One can place graphical objects in the cube in order to mark points of interest, or one can construct trajectories that illustrate paths of objects (↪ p. 247). Associated data can be encoded to the properties of graphical objects and trajectories, where color and size are common candidates. Another technique is to place multiple layers along the time axis, each of which encodes the data for a specific time point. Space-time cubes usually rely on appropriate interaction to allow users to view the data from different perspectives. A contemporary review of the concept can be found in the work by Kraak (2003).

References

Hägerstrand, T. (1970). What About People in Regional Science? *Papers of the Regional Science Association*, 24:7–21.

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

Spatio-Temporal Event Visualization

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

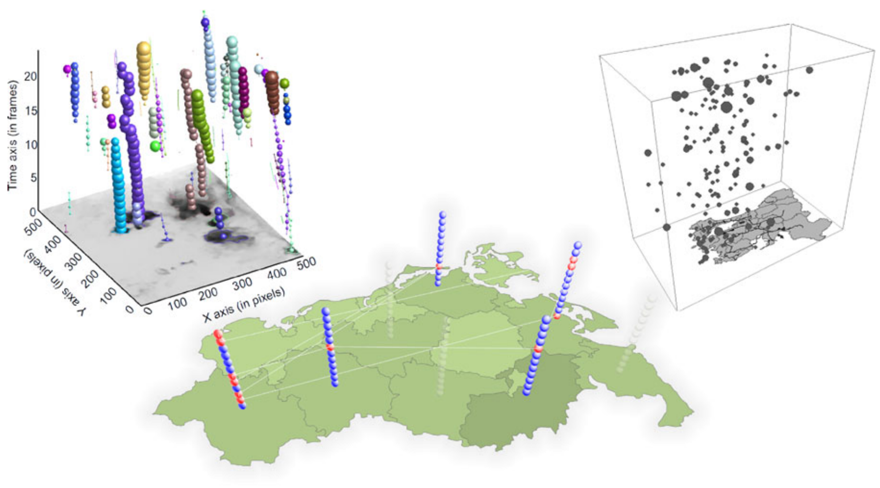


Fig. 7.95: Events in space and time are visualized by embedding graphical objects of varying size and color into space-time cubes. From left to right, the cubes show events related to convective clouds, human health data, and earthquakes.
Source: Left: Turdukulov et al. (2007), © 2007 Elsevier. Used with permission. Center: Generated with the LandVis system. Right: Gatalsky et al. (2004), © 2004 IEEE. Used with permission.

Events usually describe happenings of interest. In order to analyze events in their spatial and temporal context, one can make use of the space-time cube concept (↪ p. 245). The actual events are visualized by placing graphical objects in the space-time cube at those positions where events are located in time and space. Attributes associated with events can be encoded, for example, by varying size, color, shape, or texture of the graphical objects. Marking events in a space-time cube is a general concept with a wide range of applications: Turdukulov et al. (2007) explore events related to the development of convective clouds, Tominski et al. (2005) consider maxima in human health data as events of interest, and Gatalsky et al. (2004) visualize earthquake events.

References

Gatalsky, P., Andrienko, N., and Andrienko, G. (2004). Interactive Analysis of Event Data Using Space-Time Cube. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 145–152, Los Alamitos, CA, USA. IEEE Computer Society.

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

Turdukulov, U. D., Kraak, M.-J., and Blok, C. A. (2007). Designing a Visual Environment for Exploration of Time Series of Remote Sensing Data: In Search for Convective Clouds. *Computers & Graphics*, 31(3):370–379.

Space-Time Path

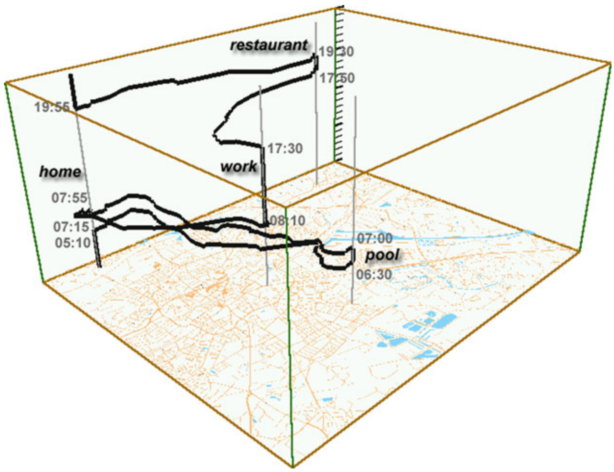


Fig. 7.96: A space-time path is embedded into a space-time cube and shows a person's movement For better orientation, important places are marked by vertical lines and annotations.
Source: Kraak (2003), © 2003 International Cartographic Association (ICA). Used with permission.

The space-time path is a specific representation of data in a space-time cube (↪ p. 245). The roots of the concept of space-time paths can be found in the work by Lenntorp (1976). Kwan (2009) describes contemporary visual representations that are based on the classic concept. A space-time path is constructed by considering the location of an object as a three-dimensional point in space and time. Multiple such points ordered by time describe the path that an object has taken. The path can be rendered as a polyline that connects successive points. In order to encode data along a space-time path, one can vary the line's color, use differently dashed line segments, or employ other visual attributes. Alternatively, a space-time path can be represented as a three-dimensional tube, where the tube's radius can be varied to encode additional data values. Today's implementations usually offer interaction to allow for virtual movements through space and time, or for rotation and zoom.

References

Kraak, M.-J. (2003). The Space-Time Cube Revisited from a Geovisualization Perspective. In *Proceedings of the 21st International Cartographic Conference (ICC)*, pages 1988–1995, Newcastle, UK. The International Cartographic Association (ICA).

Kwan, M.-P. (2009). Space-Time Paths. In Madden, M., editor, *Manual of Geographic Information Systems*, chapter 25, pages 427–442. American Society for Photogrammetry and Remote Sensing, Bethesda, MD, USA.

Lenntorp, B. (1976). Paths in Space-Time Environments: A Time Geographic Study of Movement Possibilities of Individuals. In *Lund Studies in Geography*, number 44 in *Series B: Human Geography*. Royal University of Lund, Lund, Sweden.

data
frame of reference: **spatial**
variables: **uni-, multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static, dynamic**
dimensionality: **3D**

GeoTime

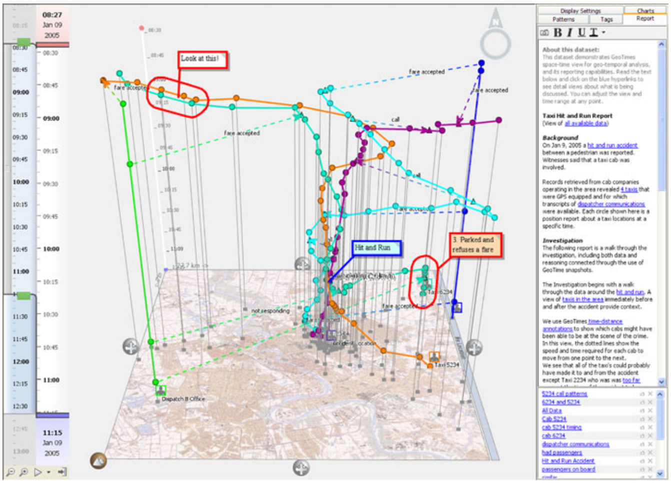


Fig. 7.97: The visualization shows taxis involved in a hit and run accident as colored points and paths. Patterns of interest are annotated in the scene, and a narrative can be authored on the right. *Source: Image courtesy of William Wright. GeoTime is a registered trademark of Oculus Info Inc.*

Kapler and Wright (2005) describe GeoTime® as a system to visualize data items (e.g., objects, events, transactions, flows) in their spatial and temporal context. It provides a dynamic, interactive version of the space-time cube concept (↪ p. 245), where a map plane illustrates the spatial context and time is mapped vertically along the third display dimension. Items and tracks are placed in the space-time cube at their spatial and temporal coordinates. GeoTime provides a variety of visual and interactive capabilities. Time intervals of interest can be selected by the user and events are smoothly animated along the time axis. Alternative projections of the display allow users to focus more on either temporal or spatial aspects. Notable about GeoTime are its annotation, storytelling, and pattern recognition features (see Eccles et al., 2008). They enable automatic as well as user annotation of the representation with findings, as well as the creation of stories about the data for analytic exploration and communication. Additional functionality allows the analysis of events and transactions in time above a network diagram (see Kapler et al., 2008).

References

Eccles, R., Kapler, T., Harper, R., and Wright, W. (2008). Stories in GeoTime. *Information Visualization*, 7(1):3–17.

Kapler, T., Eccles, R., Harper, R., and Wright, W. (2008). Configurable Spaces: Temporal Analysis in Diagrammatic Contexts. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 43–50, Los Alamitos, CA, USA. IEEE Computer Society.

Kapler, T. and Wright, W. (2005). GeoTime Information Visualization. *Information Visualization*, 4(2):136–146.

Pencil Icons

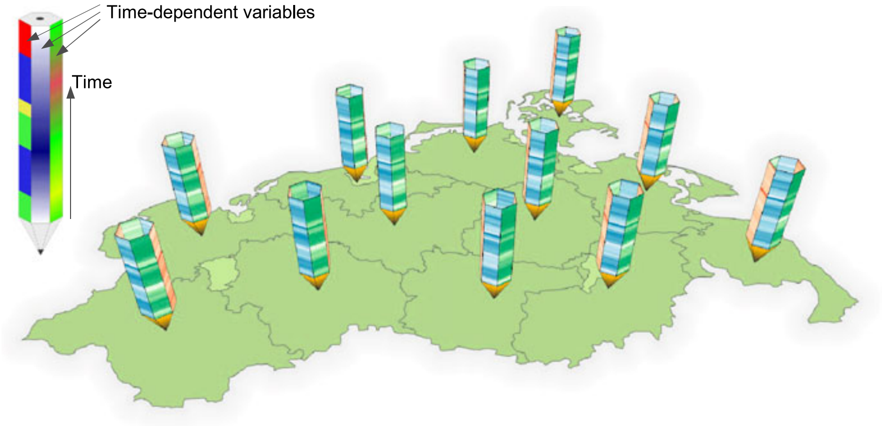


Fig. 7.98: Multiple time-dependent variables are mapped onto the faces of pencil icons to visualize temporal dependencies in the data. By placing the icons on a map, the spatial dependencies are communicated.
Source: Generated with the LandVis system.

Pencil icons have been developed by Tominski et al. (2005) to visualize multivariate spatio-temporal data. The technique is based on the space-time cube concept (↪ p. 245), where the spatial frame of reference is represented as a map in the x-y plane of a virtual three-dimensional cube. The dimension of time is mapped along the cube's z-axis. Within the cube, pencil icons are positioned where data is available. This way, the spatial context is communicated. Each pencil icon represents the temporal context and multiple time-dependent variables by mapping time along the pencil, starting at the tip, and by associating each face of the pencil with an individual time-dependent variable. Color coding is applied to visualize the data. Color lightness is varied according to data values, and different hues are used to help users identify particular variables. The linear shape of the pencil is suited to represent linear characteristics of the underlying time axis. Heterogeneous data can be depicted by using appropriate color scales. In order to deal with occlusion and information displayed at the pencils' back faces, several interaction techniques are provided, including navigation in the virtual world as well as the individual and linked rotation of pencils.

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant, interval**

vis
mapping: **static**
dimensionality: **3D**

References

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

data

frame of reference: **spatial**
variables: **multivariate**

time

arrangement: **linear**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **2D, 3D**

Data Vases

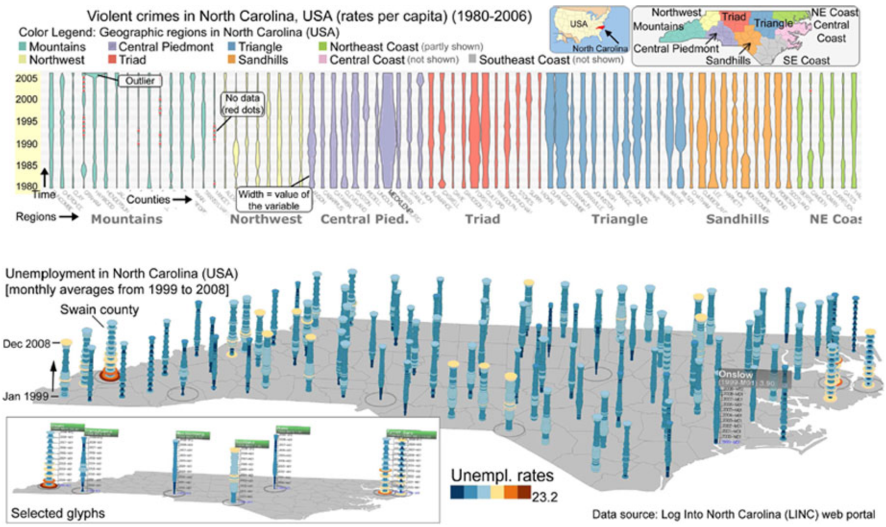


Fig. 7.99: Shape and color of a data vase encode time-varying data values. The spatial component of the data can be communicated by a vertical alignment of 2D data vases (top), or by embedding 3D data vases into a space-time cube (bottom).
Source: Thakur and Hanson (2010), © 2010 IEEE. Used with permission.

The data vases technique has been designed to visualize multiple time-varying variables. Thakur and Rhyne (2009) describe two alternative designs: a 2D and a 3D variant. A 2D data vase is basically a graph constructed by mirroring a line plot (↪ p. 153) against the time axis, effectively creating a symmetric shape that can be filled (segment-wise) with a data-specific color. Such data vases can then be arranged on the screen to create a meaningful visualization. For spatio-temporal data, one can use multiple vertically aligned data vases, each of which represents an individual geographic region. Thakur and Hanson (2010) further elaborate the idea of extending data vases to the third dimension. In 3D, data vases are constructed by stacking discs along a vertical time axis, where each disc maps the data for a particular time primitive to disc size and color. Such 3D data vases can then be embedded into a space-time cube (↪ p. 245), i.e., a virtual 3D world where two dimensions are used to show a geographic map and the third dimension encodes time.

References

Thakur, S. and Hanson, A. J. (2010). A 3D Visualization of Multiple Time Series on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 336–343, Los Alamitos, CA, USA. IEEE Computer Society.

Thakur, S. and Rhyne, T.-M. (2009). Data Vases: 2D and 3D Plots for Visualizing Multiple Time Series. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 929–938, Berlin, Germany. Springer.

Wakame

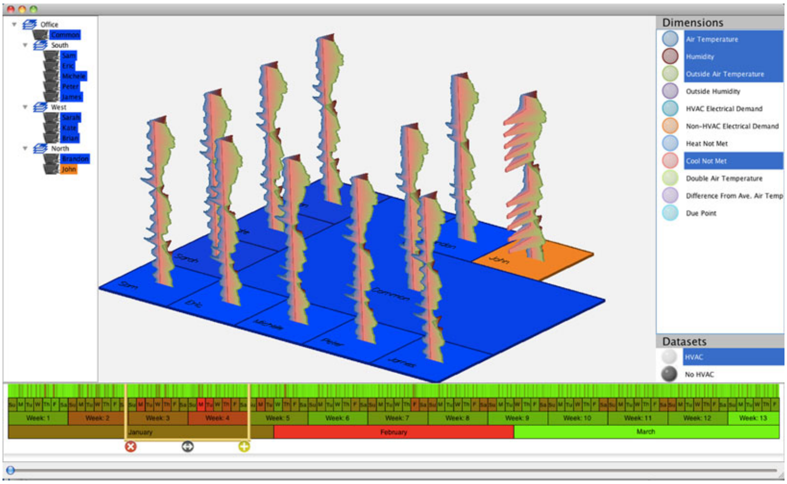


Fig. 7.100: Three-dimensional visualization objects, so-called Wakame, show multiple variables along time. Placing multiple Wakame on a map allows analysts to make sense of spatio-temporal data. A time scale widget on the bottom can be used to select intervals of interest.
Source: Image courtesy of Clifton Forlines.

Forlines and Wittenburg (2010) describe an interactive system for visualizing multivariate spatio-temporal data. The temporal aspects are encoded to multivariate glyphs, so-called Wakame. A single Wakame basically corresponds to a radar chart that has been extruded along the third dimension. In a radar chart, different variables are represented on radially arranged axes that are connected to form a polyline. Wakame are constructed as solid three-dimensional objects whose shape indicate temporal trends and relations among time-dependent variables (also see Kiviat tube, ↪ p. 211). Embedding multiple Wakame into a map display facilitates the understanding of spatial aspects. What is noteworthy about the Wakame system are its interaction and animation facilities. An intelligent camera positioning mechanism supports users in finding perspectives on the Wakame that most likely bear interesting information. A hierarchical time axis widget (↪ p. 229) denotes by color how "different" each time primitive is from its neighbors. This allows users to pick interesting time primitives easily. Upon interaction, animation is used to smoothly interpolate between views. Additional animation schemes are offered to switch between the three-dimensional Wakame view and traditional two-dimensional radar charts and line plots (↪ p. 153), which might be better suited for certain analysis tasks.

data
frame of reference: **spatial**
variables: **multivariate**

time
arrangement: **linear**
time primitives: **instant**

vis
mapping: **static**
dimensionality: **3D**

References

Forlines, C. and Wittenburg, K. (2010). Wakame: Sense Making of Multi-Dimensional Spatial-Temporal Data. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 33–40, New York, NY, USA. ACM Press.

data

frame of reference: **spatial**
variables: **multivariate**

time

arrangement: **cyclic**
time primitives: **instant, interval**

vis

mapping: **static**
dimensionality: **3D**

Helix Icons

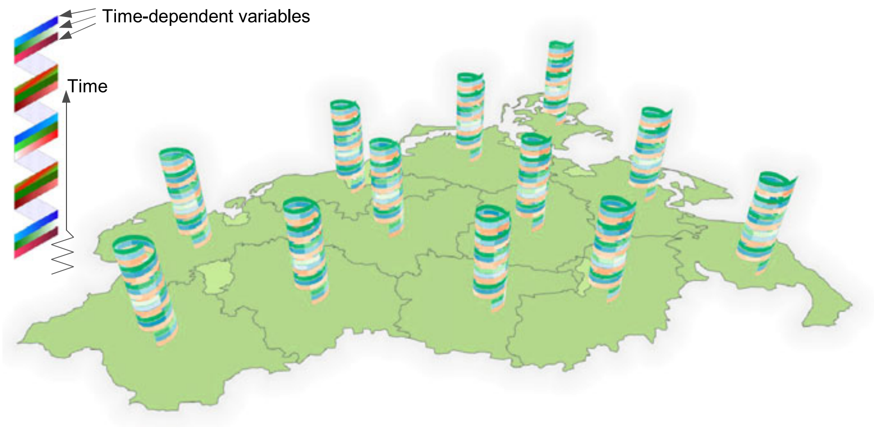


Fig. 7.101: Helix icons use color-coding to visualize multivariate spatio-temporal data along helix ribbons, which emphasize the data's cyclic temporal character. The spatial aspect of the data is illustrated by embedding helix icons in a space-time cube.
Source: Generated with the LandVis system.

Helix icons by Tominski et al. (2005) are useful for emphasizing the cyclic character of spatio-temporal data. The underlying model of this technique is the space-time cube (↪ p. 245), which maps the spatial context to the x-axis and the y-axis, and the dimension of time to the z-axis of a virtual three-dimensional cube. The actual data visualization is embedded into the cube. To this end, a helix ribbon is constructed to unroll the time domain along the z-axis. Each segment of the helix ribbon visualizes a specific instant (or interval) in time by means of color-coding. Multiple time-dependent variables can be visualized by subdividing the helix ribbon into narrower sub-ribbons, each of which represents a different variable. Using unique hues for each sub-ribbon helps the user distinguish variables. As for spiral graphs (↪ p. 185), interaction techniques help users in finding an appropriate number of segments per cycle so that periodic patterns in the data are revealed. The inherent 3D representation problems (i.e., information displayed on helix back faces or inter-icon occlusion) are dealt with by offering 3D navigation through the space-time cube and rotation of helix icons.

References

Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3D Information Visualization for Time Dependent Data on Maps. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 175–181, Los Alamitos, CA, USA. IEEE Computer Society.

7.2 Summary

This chapter reviewed 101 existing techniques for visualizing time and time-oriented data. Since there are observable balances and imbalances with regard to our categorization, it is worth taking a closer look at possible explanations.

Data – frame of reference In this book we mainly focus on abstract data, which is also reflected by our survey. Showing time-oriented data in a spatial frame of reference significantly increases the design efforts because more information has to be packed into the visual mapping. Particularly, the disciplines of cartography and geo-visualization, which are established, independent fields of research, have developed approaches to combining the visualization of temporal and spatial aspects of data (see Kraak and Ormeling, 2003; Andrienko and Andrienko, 2006).

Data – variables The number of techniques for univariate and multivariate data are almost balanced. While classic techniques often consider simpler univariate data, modern approaches take on the challenge of dealing with multiple variables. The survey also contains several techniques that cope with multiple variables simply by the repetition of a basic visualization design that only addresses univariate data (↪ p. 180).

Time – arrangement Most of the techniques in the survey support linear time; the approaches with cyclic time are significantly outnumbered. Reasons for this might be that users are usually interested in trends evolving from past, to present, to future, rather than in finding cycles in the data. The latter aspect, however, is important to fully understand the data, and therefore, expert data analysts need effective cyclic representations as well (↪ p. 186).

Time – time primitives Instants in time are the most commonly used time primitive in our survey. This seems natural because data are often measured at a particular point in time. Intervals occur less often, for example, in planning scenarios, where it is important to know how long certain activities will take (↪ p. 172). And as soon as it becomes necessary to abstract from individual instants to intervals in order to deal with bigger and bigger datasets, we have to prepare the visual representation accordingly (↪ p. 230).

Vis – mapping Apparently, the static pages of a book are better suited for showing static techniques. In this sense, our survey is a bit biased in that it contains mostly static approaches. However, dynamic animation is equally important and often it is the first solution offered when time-oriented data have to be visualized. Animation can also be an option in combination with static methods to extend the capacity of a technique in terms of the data that can be handled (↪ p. 248).

Vis – dimensionality Two-dimensional visual representations are often preferred over three-dimensional ones, because they are more abstract and thus easier to understand. Especially techniques developed in the early days of computer graphics tend to stick with two dimensions simply due to the limited computing power available then. However, modern technologies have made it easier both for visualization designers to implement three-dimensional visualization, and for visualization users to navigate and explore virtual 3D visualization spaces. This is particularly useful when data with spatial references have to be visualized (↪ p. 245).

There is another significant fact that can be derived from the survey: Most approaches address the model of an *ordered* time domain, while only a few of them explicitly consider the visualization of *branching* alternative strings of time, and none of them is capable of visualizing data that are based on the model of *multiple perspectives*. Therefore, particularly branching time and time with multiple perspectives deserve more research attention in the future.

From the review we can also see that some general concepts reoccur in several instantiations, as for instance the general application of line plots as the most basic visualization of time-dependent data, the utilization of the third display dimension to encode time, or the mapping of time to spiral shapes in order to visualize cyclic aspects. We also see that quite a number of publications is specific to a particular *what* and *why*, and as a consequence represent tailored solutions in terms of *how* the data are visualized. On the one hand, specific solutions are highly adapted and fine-tuned to be successful in supporting a specific set of users that try to solve a particular problem. On the other hand, however, these solutions are hard to adapt and reuse for other visualization problems, even when a new problem is similar to the original one and differs only in one aspect of our categorization schema. Therefore, existing techniques often lack broader applicability.

From the application perspective – a perspective that many users share as their day-to-day work is to make sense of constantly changing data – a general framework would be favorable. What other challenges need to be addressed in the future and what such a framework could look like will be discussed in the next chapter.

References

Andrienko, N. and Andrienko, G. (2006). *Exploratory Analysis of Spatial and Temporal Data*. Springer, Berlin, Germany.

Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Pearson Education, Harlow, England, 2nd edition.

Norman, D. A. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.