1. Methods

As stated earlier, glyphs represent different data variates by a set of visual properties including shape, size, color, orientation, etc. It was a wide-spread opinion for a long time that "just" knowing these basic principles of glyph-based visualization would suffice to its successful usage. More recently, however, it has been understood that only well-designed glyphs are actually useful. Some visual properties, for instance, are more prominent and thus can be easier perceived and interpreted than others (compare to preattentive visual stimuli [CM84, HBE96]). An effective glyph visualization should, therefore, carefully chose and combine different visual properties. In this section, we discuss critical design aspects and guidelines for glyph-based visualization.

1.1. Design and Usage Guidelines for Glyphs

A number of guidelines for glyph-based visualization have been proposed by Ward [War08], Ropinski and Preim [RP08], and Lie et al. [LKH09]:

- **Parameter mappings should**
  - visually emphasize important variables (e.g., using redundant mappings / one-to-many [War08, LKH09])
  - guide the user’s focus of attention, e.g., using size or transparency to encode relevance
  - incorporate semantics of the data, e.g., arrows encoding flow directions, glyph orientation in DTI data, color representing temperature
  - be mentally reconstructible / glyph orthogonality
  - glyph normalization (account for distortions introduced by other glyphs parameters)

- **glyph placement should avoid unwanted glyph aggregations in image space (e.g., using jittering)**
  - glyph shapes should be unambiguously perceivable independent of the viewing directions (e.g., superquadrics [Kin04])
  - glyph visualizations should support quantitative analysis in the attentive phase (visual exploration using brushing)
  - Hybrid visualizations should be exploited to provide spatial context
  - Enhance depth perception (e.g., chromadepth or contours)

Also, ordering of data dimensions can be correlation-, symmetry-, data- or user-driven [War08].

In the context of information visualization, Ward [War02] discusses glyph placement strategies such as data- or structure-driven placement. Ropinski and Preim [RP08] propose a perception-based glyph taxonomy for medical visualization. The authors categorize glyphs according to 1) preattentive visual stimuli such as glyph shape, color and placement, and 2) attentive visual processing, which is mainly related to the interactive exploration phase (e.g., changing the position or parameter mapping of a glyph). Additional usage guidelines are proposed, for instance, that parameter mappings should focus the user’s attention and emphasize important variates in the visualization. Also, glyph shapes should be unambiguous when viewed from different viewing directions. Kindlmann [Kin04], for example, use superquadric glyph shapes that fulfill the latter criterion.

1.2. A pipeline for glyph-based visualization

Inspired by the work of Ropinski and Preim, Lie et al. [LKH09] propose further guidelines for glyph-based 3D visualization. Aligned with the visualization pipeline [HS09], the task of creating a glyph-based 3D visualization is divided into three stages as shown in Fig. 1: 1) during data mapping, the data variates are remapped (to achieve, for example, some contrast enhancement) and
mapped to the different glyph properties; 2) glyph instantiation creates the individual glyphs, properly arranged across the domain; and 3) during rendering, the glyphs are placed in the visualization, where one has to cope with issues such as visual cluttering or occlusion. In the following, we discuss critical design aspects for each of these steps.

1.2.1. Data Mapping

Similar to Ward [War02], Lie et al. [LKH09] consider it useful that the glyphs expect normalized input from the depicted data variates such as values in the range \([0, 1]\). During data mapping, the authors identify three consecutive steps. First, the data values within a user-selected range \([w_{\text{left}}, w_{\text{right}}]\) are mapped to the unit interval. Values outside this range are clamped to 0 or 1, respectively. This allows to enhance the contrast of the visualization with respect to a range of interest (sometimes called windowing). A natural default choice for this step would be a linear map between \([w_{\text{left}}, w_{\text{right}}]\) and \([0, 1]\), but also other forms of mapping could be considered (for example, a ranking-based or discontinuous mapping). After the windowing, an optional exponential mapping \(e(x) = x^2\) can be applied in order to further enhance the contrast on the one or the other end of the spectrum. Finally, a third mapping step enables the user to restrict or transform the output range that should be depicted by a glyph property. Here, also semantics of the data variates can be considered (compare to the usage guidelines of Ropinski and Preim [RP08]). Using a reverse mapping, for instance, smaller data values that are possibly more important can be represented in an enhanced style while larger values are deemphasized.

1.2.2. Glyph Instantiation

- visual channel (color, shape, size, texture, and opacity) vs.
- visualization dimensionality (2D, 2.5D, 3D)

Several considerations are important for the instantiation of individual glyphs. When using a 3D glyph shape, one has to account for possible distortions introduced when viewing the glyph from a different point of view [Kin04]. In order to avoid this problem, Lie et al. suggest to use 2D billboard glyphs instead. In certain scenarios, however, it makes sense to use 3D glyphs, for example, when depicting a flow field via arrow glyphs. Another challenge in glyph design is the orthogonality of the different glyph components, meaning that it should be possible to perceive each visual cue individually (or to mentally reconstruct them as suggested by Preim and Ropinski [RP08]). When representing a data variate by glyph shape, for example, this affects the area (size) of the glyph as well. Accordingly, such effects should be normalized against each other, for instance, by altering the overall glyph size in order to compensate for implicitly changes of the glyph shape.

However, it is not always easy to design a glyph-based visualization such that the different data-to-property mappings are independent and do not influence each other (the interpretation of shape details, for example, is usually influenced by the size of the glyph). In this context, the number of data variates that can be depicted must be seen in relation to the available screen resolution. Large and complex glyphs such as the local probe [DLvW93] can be used when only a few data points need to be visualized. If many glyphs should be displayed in a dense manner, however, a more simple glyph may be desirable [KW06]. Another design guideline is the usage of redundancies, for instance, to use symmetries that ease the reconstruction of occluded parts of the glyph. Important properties can, however, be mapped to multiple glyph properties in order to reduce the risk of information loss.

1.2.3. Rendering

Important aspects when rendering many glyphs in a dense 3D context are depth perception, occlusion, and visual cluttering.
tering. In cases where many glyphs overlap, halos can help to enhance the depth perception and to distinguish individual glyphs (compare to Piringer et al. [PKH04]). For improving the depth perception for non-overlapping glyphs a special color map (called *chroma depth* [Tou97]) can be used to represent depth. Finally, appropriate glyph placement [RP08, War02], interactive slicing, or filtering via brushing are strategies for dealing with occlusion and cluttering issues.

2. Application Domains

2.1. Tensor Visualization

David / Bob Here is a number of recent papers that may be interesting in this context:

- “Fused DTI/HARDI Visualization” by Prkocska et al. [PPvA*11];
- “Sampling and Visualizing Creases with Scale-Space Particles” by Kindlmann et al. [KESW09];
- “Superquadric Glyphs for Symmetric Second-Order Tensors” by Schultz and Kindlmann [SK10];
- “GPU-Based Ray-Casting of Spherical Functions Applied to High Angular Resolution Diffusion Imaging” by van Almsick et al. [vAPP*11]

2.2. Medical Visualization

2.2.1. SPEC Data

- Ropinski and Preim [RP08];
- Meyer-Spradow et al. [MSSD*08]

3. Time-dependent Data / Flow Data

- Helix glyphs [TSWS05];
- arrow glyphs, e.g., Treinish [Tre99];
- Peng et al.

Since glyphs are often not placed in a dense way, the space between them can be used for additional information. Kirby et al. [KML99], for instance, use concepts from painting for visualizing 2D flow. They combine different image layers with glyphs, elongated ellipses, and color. Treinish [Tre99] visualizes multi-variate weather data using color contouring on vertical slices and isosurfaces that represent cloud boundaries. At user-defined locations (vertical profiles), the wind velocity and direction are represented by a set of arrow glyphs. Streamlines following the wind direction are seeded at each arrow.

3D helix glyphs on maps [TSWS05] are used to represent the spatio-temporal relations of health data. The time-dependent data values are color coded and mapped to the ribbons, which increases in angle and height for each time step in order to create a “tunnel view”. Thereby, multiple data attributes can be represented by subdividing each ribbon into smaller sub-ribbons. Interaction techniques such as altering the helix’ diameter (in order to reveal cyclic behavior) and navigating through the 3D representation are also incorporated.

3.1. Uncertainty Visualization

- “Glyphs for visualizing uncertainty in vector fields” by Wittenbrink et al. [WPL96];
- “Multidimensional visual representations for underwater environmental uncertainty” by Schmidt et al. [SCB*04];
- “Multivariate Glyphs for Multi-Object Clusters” by Chlan and Rheingans [CR05];
- “A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets” by Sanyal et al. [SZB*09]; later they publish NOODLES [SZD*10];

[Rephrase: Multi-run simulations are an important step in the development of simulation models, where one aims to identify model parameters that have the most influence on the simulation output. In such a sensitivity analysis [Ham04, Hel08], the values of certain model parameters are changed systematically and multiple simulation runs are computed, accordingly. In the resulting data, a distribution of values is given for the same data variable at each position in space and time (one value for each run). The representation of such multi-run data is rather new to the visualization community [LPK05, KDP01, KLDP02]. It is especially challenging since the data are often time-dependent, higher-dimensional, multi-variate, and large at the same time [WP09]. A direct visualization of such time-varying volumes of data distributions is often not feasible. Accordingly, the individual distributions of multi-run values need to be analyzed first, and then derived statistical properties can be visualized (compare to the visual analytics mantra [KAF*08]).]

[Rephrase: The visualization of multi-run data is especially interesting since it is an alternative approach for representing uncertainty [WP09, PKRJ10]. General approaches for uncertainty visualization are discussed by Pang et al. [PW1L97], Johnson and Sanderson [JS03], and Griethe and Schumann [GSO06]. MacEachren et al. [M*05], moreover, review approaches for geospatial uncertainty visualization. In the following, we describe coordinated multiple views and abstraction techniques for multi-run data such as visualization of statistical parameters, shape descriptors, or operators.]

Data distributions are often represented with box plots [MTL78], encoding minimum and maximum values, mean, median, and other quartile or percentile information. Kao et al. [KDP01, KLDP02] extend this approach to 2D multi-run data. In some cases, the distribution can be represented adequately by statistical parameters such as mean, standard deviation, interquartile range, skewness or kurtosis. The computed statistics are visualized on 2D surfaces using colorcoding and bar glyphs. Recently, Potter et al. [PKRJ10] survey different box plot variations and propose another extension.
to this approach. The so-called summary plot includes additional statistics of the multi-run data such as skewness, kurtosis and tailing information. These plots, however, cannot be placed in a dense spatial context. In recent work, Kehrer et al. [KMDH11] visualize aggregated multi-run properties using glyphs that are based on super ellipses (see Fig. 3). The glyphs are carefully designed in order to be placed in a 3D context [LKH09]. The approach is integrated in a visual analysis framework with linking and brushing.

Spaghetti plots [DHLZ02] are utilized by meteorologists to investigate multi-run data, where a contour line is visualized for each run at a selected time step (resembling a pile of spaghetti noodles). Sanyal et al. [SZD+ 10] combine spaghetti plots with a ribbon- and glyph-based uncertainty visualization. The uncertainty glyphs consist of a number of concentric colored circles that represent the standard deviation, interquartile range, and the width of the 95% confidence interval.

In PlanningLines [AMTB05], temporal intervals (e.g., certain activities) and related uncertainties are visualized (maybe Figure). Each task consists of two encapsulated bars representing the the minimum and maximum duration, which are bounded by two caps depicting the start and end intervals. Thereby, also uncertainties can be visualized, i.e., when planning future activities one often does not know exactly how long a task can take or when it will start or end.

References


The Eurographics Association 2010.


[26] vAPP’11 VAN ALSMICK M., PEETERS T., PRCKOVSKA V., VILANOVA A., TER HAAR ROMENY B.: Gpu-based ray-casting of spherical functions applied to...


