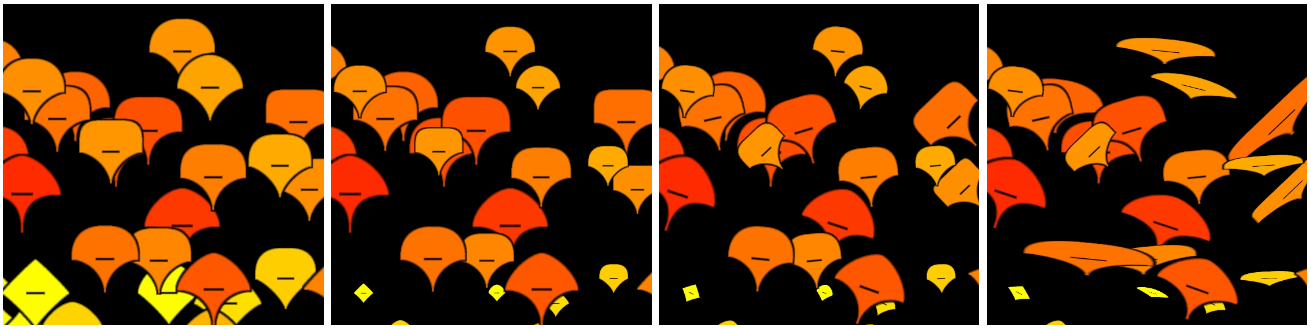
Critical Design and Realization Aspects of Glyph-based 3D Data Visualization

Andreas E. Lie*
University of Bergen
Norway, www.ii.UiB.no/vis

Johannes Kehrer†
University of Bergen
Norway, www.ii.UiB.no/vis

Helwig Hauser‡
University of Bergen
Norway, www.ii.UiB.no/vis



(a) Two data attributes are represented as the upper / lower glyph shape (b) Added data attribute to overall glyph size (c) Glyph rotation has been assigned a data attribute as well (d) A data attribute has been assigned to glyph aspect ratio

Figure 1: Adding more attributes to the glyph, while preserving the glyph's orthogonality.

Abstract

Glyphs are useful for the effective visualization of multi-variate data. They allow for easily relating multiple data attributes to each other in a coherent visualization approach. While the basic principle of glyph-based visualization has been known for a long time, scientific interest has recently increased focus on the question of how to achieve a clever and successful glyph design. Along this newer trend, we present a structured discussion of several critical design aspects of glyph-based visualization with a special focus on 3D data. For three consecutive steps of data mapping, glyph instantiation, and rendering, we identify a number of design considerations. We illustrate our discussion with a new glyph-based visualization of time-dependent 3D simulation data and demonstrate how effective results are achieved.

Keywords: Glyphs, Multi-variate, Simulation, Glyph Design, Visualization

1 Introduction

In scientific projects as well as in commercial applications we see an increased utilization of computational simulation for the investigation of natural phenomena. Compared to earlier years, current resulting datasets are 3D instead of 2D, time-dependent instead of single time step, only, and multi-variate with many values per space-time location, to name just three of more recent properties (which soon will be standard in many cases). This means that not only the large size of simulation datasets is challenging, but also its complexity. With this, it is getting more important and more difficult to enable users to "read between the lines", i.e., to better understand the relations between different data dimensions. A variety of useful visualization approaches have been proposed to reveal the information that is contained in high-dimensional data, and we refer to Ward [Ward 2008] and Bürger and Hauser [Bürger and Hauser 2007] for a review of some of these approaches.

An interesting approach is to use glyphs to represent multiple data variates per space-time location. A generic (and usually also relatively simple) shape is defined with a set of variable appearance properties, including shape characteristics, color, opacity, etc., that when instantiated is parameterized by a subset of the data variates per data item. Glyph-based visualization has been known for many years. It is more than 15 years ago, for example, that de Leeuw and van Wijk [de Leeuw and van Wijk 1993] proposed the so-called "local flow probe" as an interesting example for glyph-based visualization of 3D flow data. Recently, there is new interest in glyph-based approaches. See, for example, Ropinski and Preim [Ropinski and Preim 2008] for a recent survey. Several interesting examples of glyph-based visualization that have recently been published, see Oeltze et al. [Oeltze et al. 2008], Kindlmann and Westin [Kindlmann and Westin 2006], Ropinski et al. [Ropinski et al. 2007] and Meyer-Spradow et al. [Meyer-Spradow et al. 2008].

An important lesson learned from these recent works is that an appropriate glyph design is crucial for the success of a glyph-based visualization. It was a wide-spread opinion in the related research community for a long time, that "just" knowing the well-published basic principle of glyph-based visualization would well suffice to also utilize this approach successfully. More recently, however, it has been understood that only very well designed glyphs are actually useful. In this paper we therefore discuss critical design aspects

*e-mail: Andreas.Lie@student.uib.no
†e-mail: Johannes.Kehrer@uib.no
‡e-mail: Helwig.Hauser@uib.no

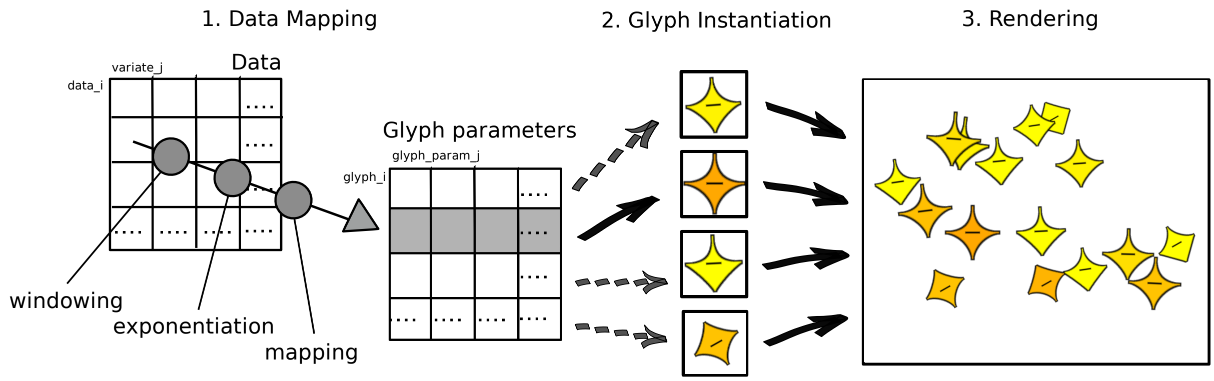


Figure 2: Data variates undergo data mapping stages; windowing, exponentiation and mapping. These values are then used to instantiate the corresponding glyphs, (e.g, determining shape, size) and finally the glyphs are rendered into the context.

of glyph-based visualization with the special focus on 3D data. We exemplify our discussion with a new, glyph-based visualization of time-dependent 3D simulation data and show how effective results can be achieved.

In section 2 we describe similar and related work. In sections 3 and 4 we will present considerations with respect to glyph representation. Section 5 will exemplify different datasets visualized by glyphs created according to these considerations.

2 Related Work

Ward [Ward 2002] discusses several different glyph types and placement strategies for glyphs. This work is considered highly relevant for glyph based visualization. Ropinksi et al. [Ropinski et al. 2007] propose glyph placement strategies, and use glyphs on surfaces in 3D for visualizing multi-variate data. Our focus is not placing glyphs on surfaces, but in a truly 3D environment. They also provide a thorough taxonomy for glyph-based visualizations in the medical domain [Ropinski and Preim 2008]. We aim to build upon this by extending the preprocessing step prior to the creation of the glyphs. Sawant and Healey [Sawant and Healey 2007] successfully map several attributes to their glyphs used in flow visualization. Our aim is to use more complex shapes to enable more variates to be mapped to glyph properties. Bertin [Bertin 1983] proposed six retinal variables: shape, size, orientation, color (value and hue) and texture, which stresses the importance of careful and well thought glyph design. Shaw et al. [Shaw et al. 1999] show that it is hard to distinguish similar shapes of super ellipsoids. This is a problem related to the orthogonality of the shapes. Wong et al. [Wong et al. 2002] fuse several layers of variate visualizations, but suffers slightly from cluttering and occlusion. De Leeuw and van Wijk [de Leeuw and van Wijk 1993] designed the Probe glyph which could properly visualize twelve different parameters simultaneously. Van Walsum et al. [van Walsum et al. 1996] describe features and attribute sets which are extracted from the regions of interest in the data, allowing local minima and maxima to be mapped to icons (or glyphs). We build upon the idea of features and attribute sets, and allow changing of mapping inside the extracted data ranges as well. Densely packed icons have been used to form visual textures [Pickett and Grinstein 1988], representing multi-variate data. Stolte et al. [Stolte et al. 2002] proposed a system named Polaris which does interactive visual analysis and allows data transformations to unveil hidden relations and data. Kindlmann [Kindlmann 2004] uses different kinds of super quadrics as glyphs to visualize data. Similar work has been done by T.J. Jankun-Kelly and Mehta [Jankun-Kelly and Mehta 2006] which use super ellipsoid glyphs to visualize variates in nematic liquid crystals. Piringer et al. [Piringer et al. 2004] show the importance of halos to enhance the depth perception and separability of points. Toutin [Toutin 1997] uses color to assist the viewer in interpreting spatial relation.

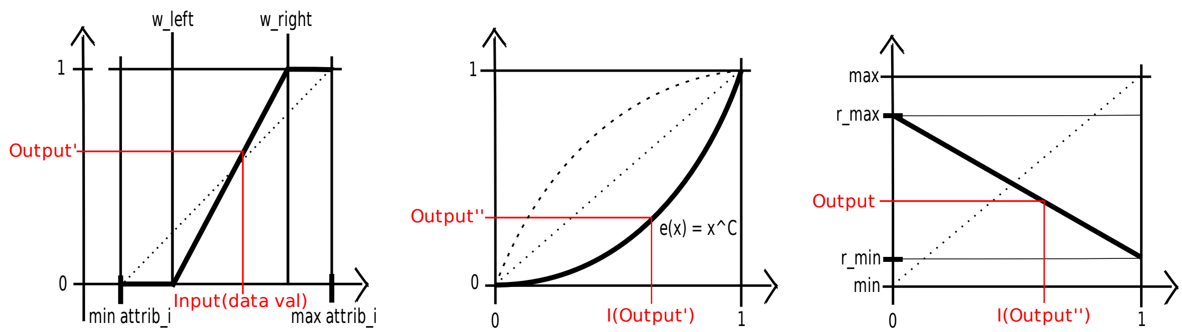
3 Overview

E. Tufte discusses in his work [Tufte 2001] that developing design techniques for enhancing graphical clarity is crucial. According to the model of the visualization pipeline [Hauser and Schumann 2009], we suggest to substructure the task of glyph-based 3D visualization into three separate steps, namely Data Mapping, Glyph Instantiation and Rendering. Figure 2 shows the flow from the data mapping step, to glyph instantiation and the final rendering step. It is generally very useful for glyphs to have normalized input from the variables the glyphs represent. While the interval $[-1,1]$ arguably can be used for such a normalization purpose, we choose to use the range $[0,1]$ for the design of our very comprehensible model. Accordingly, the data mapping step comes first in our pipeline, and the glyph instantiation and rendering steps successively. We think its generally useful to consider to data mapping as three elementary stages, namely, windowing, exponentiation, and mapping. Aiming for a glyph based visualization of 3D data, there is always a question on how to cope with the problem of occlusion and cluttering which results in information loss. We propose three generic options in order to deal with these challenges, namely, halos, chromadepth, and interactive slicing. The glyphs depicted in figure 1 are created by drawing two super ellipses (one for the upper, the other for the lower half) and combining them. These shapes are considered simple, meaning that they are easy to understand and allow for mental completion if they were to overlap or partially occlude each other. The glyphs can have data mapped to them, controlling the upper and lower half, color, rotation, size and aspect ratio. In section 6, we will describe the creation of these glyphs in more detail.

4 Selected generic Considerations with respect to Glyph Representation

4.1 Data Mapping

In this section we discuss three steps in the data mapping stage where the user has the ability to increase the value of the visualiza-



(a) Windowing (b) Exponentiation (c) Mapping

Figure 3: (a) User selects w_{left} and w_{right} which adjusts the data mapping ramp for input datavalues. The dotted line represents the default value for the ramp. (b) User can control the data mapping curve by adjusting the C term in x^C. The middle dotted line represents the default curve, the curve can be bent upwards or downwards. (c) The output range of the datamapping can be adjusted to fit the datavariates. User selects r_{min} and r_{max} to clamp the output range.

tion by adjusting the data variates according to his/her needs. We also cover several aspects of glyph design to provide some guidelines for making the most useful glyphs. Finally, we will cover how to cope with occlusion and cluttering in the visualizations.

Windowing The process of windowing serves the effect of enhancing differences in data values. This is achieved by clamping the range of selected values to be linearly distributed to the output range. Figure 3a displays such a mapping function for data values. Every value outside the clamped window results in either the mapping functions minimum or maximum output depending on the data value in question. The windowing allows the user to select the w_{left} and w_{right} giving the user complete control over how and where the mapping slope will reside in the data domain. Windowing is a method widely used in medical visualization where it is more commonly known as contrast enhancement, and is often applied to visualize 2D or 3D images. In figure 7 several examples of such windowing are illustrated. The default setting equals a simple linear mapping of the data values. (the dotted lines in figure 3a).

Exponentiation After the windowing all datavalues are transformed to the unit range, $[0,1]$, the next step in our pipeline is to adjust the function that further processes the data. This process is known as exponentiation. Here we allow the user to control the exponent c in the simple, yet powerful function x^c. Such functionality makes selected data values easier to distinguish based on which curve was chosen. Figure 3b displays such exponential mapping. This curve exists only inside the window and is by default x^1 which is linear mapping. Figure 7 is an example where both windowing and exponentiation have been applied to the data.

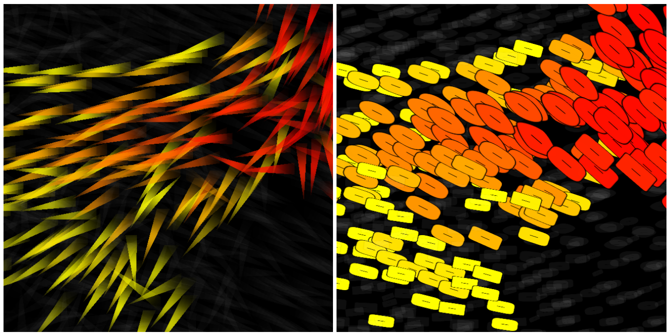
Mapping Data may have characteristics that may result in unwanted glyph behaviour. We therefore allow the glyph mapping to be altered according to the users wish for solving such problems. It would, for instance, make little sense to map time to a rotation attribute, which could rotate the glyph $\pm 45°$. Mapping allows us to change the output range of the data mapping pipeline, making this attribute more fitting by restricting the output range to (in this case) $[0.5,1]$ instead of the regular $[0,1]$ range. Now time $t=0$ equals zero rotation, and $t = max$ equals the equivalent max glyph rotation. An example where mapping applies well, is in the case where the user wants to focus on low values for a certain data variate. This variate can now be mapped to glyph size, resulting in more prominent glyphs for low values by reversing the mapping. Figure 3c shows an example of change in the output function. Figure 7d displays the effect of reversing the mapping.

Data mapping consists of three simple steps, which are intuitive and straight forward to specify. These steps are an important aspect for glyph based visualization. Users of glyph based visualizations will start by selecting a mapping (possibly choosing to invert data or clamp output ranges). After specifying the correct output ranges for the data mapping pipeline, they then proceed to change and fine tune the windowing and exponentiation data mapping stages.

4.2 Glyph Instantiation

2D vs. 3D Using glyphs for visualizing multi-variate variables in 3D is challenging for the user. To mentally reconstruct the particular values represented by glyphs is non-trivial. Size, orientation and geometric properties are often mapped to the data (because the properties represent strong visual cues). Effects from 3D projection complicate the interpretation of the glyph shapes. Therefore, we suggest to only use 3D glyphs if they are geometric properties and inherently related to the 3D position where they are placed. For example, it makes sense to show arrow based glyphs mapped with the three velocity components in 3D simulation domains. For other visualization we strongly propose to use billboarded 2D glyphs since they avoid distortion of other glyph properties. Figure 4a is such an example of arrow based glyph visualization.

Orthogonality A big challenge in glyph design is the orthogonality of the glyph components. If glyph parts are not visually separable, the interpretation is non-trivial. An example of such a mapping, is to map data values to individual RGB color components, since interpreting the individual color components from a color is very hard. Furthermore, large numbers of variates is hard to accommodate if the glyph shape is simple. The glyph size and complexity must be seen in a direct relation to the resolution of the visualization. If there are few datapoints large glyphs can be used. If there are many data points, simple shaped glyphs that can be displayed in a densely manner are required. Eventually it must be assumed that there exist some maximum number of glyph shapes and properties that the user can distinguish and discriminate. In our examples we found it challenging to have 5 or 6 different variates mapped to the glyph. In the figure 1 we can see how well the glyph avoids distortion when having more and more attributes mapped to it. In the first



(a) 3D arrow glyphs (b) 2D glyph billboards

Figure 4: (a) 3D glyphs are fine, if data with an inherent relation to the 3D space is shown (such as flow direction); (b) otherwise 2D (billboarded) glyphs are preferred

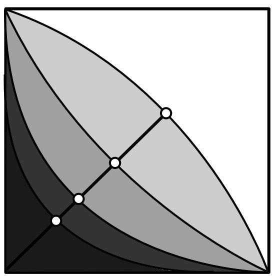


Figure 5: We calculate the difference between the innermost and outermost curve, and use this number to equally distribute the shape variations linearly along the diagonal.

image three variates are mapped. Two to (upper and lower) shape and one to color. The second image has included size as a parameter for the glyph. In the third rotation is introduced, and in the last visualization aspect ratio is mapped for a total of 6 variates. Mapping both size and aspect ratio should be handled with care, since they combined impose a perceptual challenge on the viewer. They can however be utilized, for instance, by using size as a selection parameter. Thus large glyphs have important characteristics (small glyphs do not), and having aspect ratio possibly depicting another parameter of interest. An example mapping can be seen in the Hurricane Isabel visualization, see figure 10, where size depicts amount of clouds, and aspect ratio depicts air velocity.

Normalization A variation of the glyph shape, for example, has the implicit effect of changing the size (i.e., area) of the glyph. We therefore suggest to normalize these effects against each other, e.g., to adapt the overall glyph size in order to compensate the otherwise implicit change of the size due to the shape variation. In our glyph design we also take into consideration that the shapes used for representing the glyphs are visually not equally spaced. Figure 6 shows how the shapes originally where, and how (the lower row) they became after shape equalization. For the shape equalization, we calculated the distance from the center of the shape, to its curve along the diagonal, and used these calculations to select shapes which would result in visually equally spaced shapes. In figure 5 you can see how measurements were made to normalize the shapes.

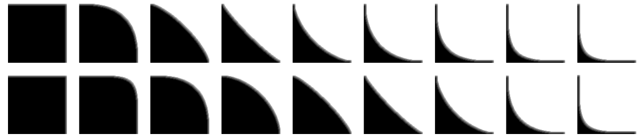


Figure 6: Upper row: the attribute is directly mapped to the shape exponent. The lower row: perceptual shape normalization along the diagonal.

Redundancy As mentioned above, it is challenging to read glyph based visualizations, even if designed with care. Using redundancy to depict especially relevant data characteristics is an useful way to emphasize important attributes, and decrease the chances for information loss. The glyphs design inherently displays the same shapes on the right and left side of the glyphs, which allows users to correctly understand how the glyphs are shaped, despite being able to see only partially occluded glyphs in visualizations. The densely packed glyphs in figure 9 relies strongly on this. Figure 9 has color and size mapped to temperature, emphasizing the importance of temperature in the visualization. We also use a vertical bar inside the glyphs, to assist users interpreting the rotation of a given glyph. Figures 1, 9 and 10 all show these bars that assist user interpretation of rotation.

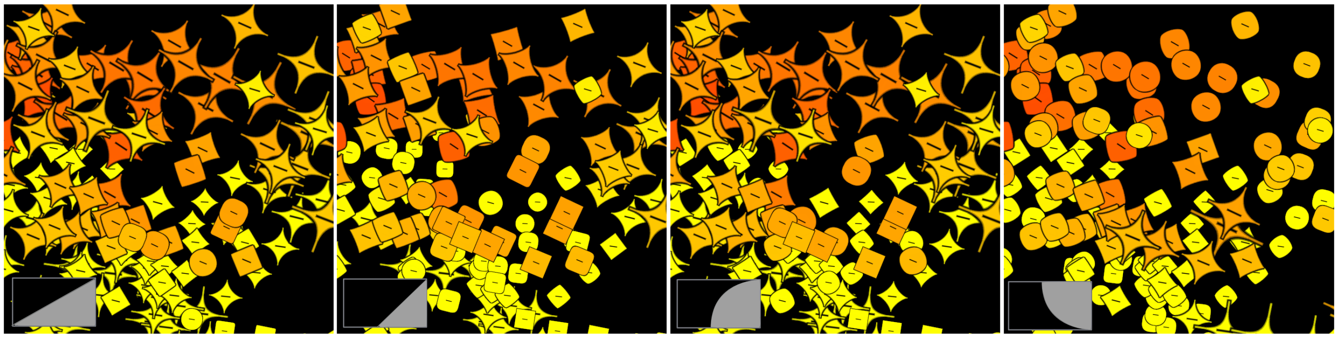
Glyph-based visualization is just one opportunity to visualize multivariate data. Glyphs are helpful to understand multiple variates simultaneously (e.g. "reading between the lines"). Therefore it is of much more importance to carefully think about inter property aspects of glyph design, i.e. how the different glyph characteristics are dependent on each other, than the individual glyph expressions of data variables. The size and aspect ratio characteristics is an excellent example of this. Eventually it is how all the different variations harmonize that will result in whether or not users are able to achieve their goals.

4.3 Rendering

A general problem in 3D visualization is occlusion, depth perception and visual cluttering. Glyph-based visualizations are most comprehensible when selecting (or placing) glyphs in such a manner that only a small number of glyphs are shown simultaneously and that they do not overlap or occlude one another. Often it is not trivial to achieve either small numbers of glyphs, or view angles where glyphs do not overlap. In these circumstances we suggest three approaches; halos, chromadepth, and interactive slicing, that by themselves have been proposed in different situations and scenarios.

Halos A simple but effective way to improve depth perception of discrete primitives in datasets is to include halos around the primitives. This will make the primitives stand out from other objects, and allow users to mentally complete the partially occluded shapes since the individual glyphs can be identified. This is a technique very common among illustrators for drawing attention toward objects. Piringer et al. [Piringer et al. 2004] and Interrante et al. [Interrante and Grosch 1998] use halos to emphasize discontinuity in depth and to draw the users attention towards objects.

Chromadepth Relative depth perception is hard to cope with by using only halos. The relation of two non-overlapping glyphs are diffucult to determine (which is in front of which). By allowing the



(a) (b) (c) (d)

Figure 7: Different changes to the datamapping stage done successively. 7a depict default datamapping. In 7b window adjustments have been done to achieve better contrast with respect to higher values. Exponentiation curve has been fine tuned in 7c to reveal more differences among the lower values for the selected range. In 7d the mapping output range has been inverted.

use of color to represent depth (instead of a data attribute), similar to the work of Toutin [Toutin 1997], and by using a color scale that either is complementary or has clear continuous change, depth perception can be achieved successfully. The figure 8c is an example of chromadepth and its effect.

Interactive Slicing Occlusion is a major problem when reading glyphs. Since halos and chromadepth does not cope with occlusion, we suggest to employ interactive slicing, a technique that allows for view dependent slice-based visualizations.The user specifies a plane in the 3d visualization, which determines whether the renderer should omit the data or not. This way we can avoid the occlusion problem by suppressing the occluding glyphs in the front of our specified plane. Figure 10 is an example of such slicing, where only the lowest layer of the hurricane Isabel is visualized. Interactive slicing is commonly used in volume visualization.

These three solutions enable the user to cope with a large amount of problems caused by occlusion and visual cluttering. They also enhance and attract the focus of the user to the glyphs with halos and may assist revealing hidden nuggets (valuable information) in the data by interactive slicing.

5 Demonstration

In this section we will demonstrate that glyphs can be used in conjunction with very different datasets and successfully depict several different characteristics simultaneously. The datasets are the Diesel Exhaust System-dataset and data from the hurricane Isabel. Both these data sets are studied thoroughly by Doleisch et al. [Doleisch et al. 2004a] and [Doleisch et al. 2004b].

5.1 Diesel Exhaust System

The diesel exhaust system includes a diesel particle filter which traps soot, and burns the soot at over 1000 degrees to oxidize it at different intervals. The dataset contains over 260.000 vertices, and is given at ten timesteps. In Figure 9 timestep five is visualized. We map data attributes to the glyph properties as specified in table 1.

Color	Flow Temperature
Glyph Upper	Soot amount
Glyph Lower	Soot amount second derivative
Glyph Size	Flow Temperature
Glyph Rotation	O^2 fraction

Table 1: Glyph property mapping for Diesel Exhaust System dataset

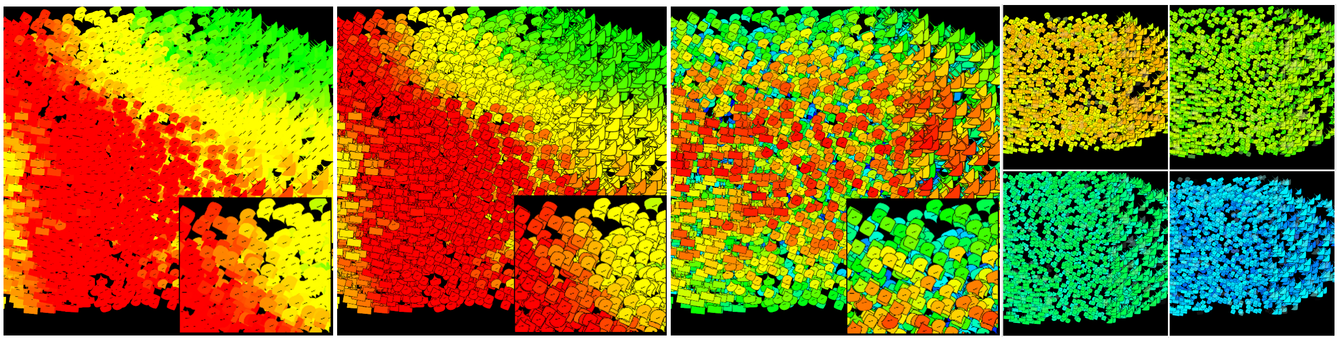
Exhaust and soot particles from the diesel engine is guided into a diesel particle filter where soot is trapped and oxidized at around temperatures from 600 up to over 1000 degrees celsius. Table 1 shows which data values are mapped to the glyph properties. The oxidation process moves from left to right.

From the visualization in figure 9, we can see that the temperature differences cause an uneven oxidation of the soot. Two areas with non-optimal temperature levels (green) can be located on the right side of the oxidation peak temperatures (red). In these areas one can see that the O^2 levels are low (rotation), a critical part (in addition to high temperature) to burn as much soot as possible. The visualization also can verify that soot amount left of the peak temperatures is very low.

5.2 Hurricane Isabel

The hurricane simulation contains meteorological data from the class 5 hurricane Isabel which ravaged in 2003. The dataset has 24 variables and contains 24 timesteps, each of 100.000 vertices. In Table 2 the glyph property mapping for the visualization in Figure 10. We choose to focus on fast moving air flows that exist close to the surface. Through the use of slicing, only the lowest layer of data (closest to the surface) is selected. Semi-transparent glyphs can be identified in this visualization, which is a direct result of brushing flow velocity vs clouds with smooth degree-of-interest in the SimVis framework.

We can identify the eye of the hurricane in the lower right corner of the visualization in figure 10. The eye is almost surrounded by a wall of precipitation and relatively colder airflows. From this visualization one can see that there is low pressure inside the hurricane, and that cold winds from north mix in with warm air from the south. A cold front can be identified in the higher left parts of the visualization. There exists also a very interesting area directly below this



(a) (b) (c) (d)

Figure 8: These visualizations are from the diesel particle filter dataset. Image 8a represents just simple colored glyphs. 8b emphasizes glyph differentiation by adding halos to the glyphs. In 8c color is used to assist the user in interpreting the spatial relation between the glyphs (depth perception). Various orthogonal slices can be seen in 8d where color indicates the depth of the slice taken, similar to 8c.

cold front, where a small subset of glyphs (four) identifies a region where there is a high amount of pressure and precipitation.

Color	Temperature
Glyph Upper	Pressure
Glyph Lower	Precipitation
Glyph Size	Clouds

Table 2: Glyph property mapping for Hurrican Isabel dataset

6 Technical Details

Our glyph visualization is integrated into the SimVis framework [Doleisch 2007] for assisting the user in visual datamining and analysis. The SimVis framework allows interactive visual analysis of large multi-variate datasets. The renderer allows user changes to the data pipeline to adjust for more optimal glyphs. See section 4.1 on windowing and exponentiation for a thorough explanation. The framework and the plugin was developed in programming languages C++, OpenGL and CG-shader.

We employ a glyph texture atlas containing all possible glyph variations. This texture allows us to externalize the glyph itself, making the glyphs fully independent of the framework. By using such atlases, new glyphs and variations are very likely to appear since the framework is unaware of a glyphs visual characteristics. In the atlas we only represent one quadrant of each glyph, to save valuable space by omitting redundant information. The glyph shapes can easily be reconstructed inside a shader by simple mirror and rotation operations.

Our glyph texture atlas was created by drawing various super ellipses, and saving them in the atlas. We also employed antialiasing to smooth the borders of the glyphs giving them a more visually pleasing look. Halos were saved in a separate channel of the atlas, enabling the shader to include the halo if the user specified so.

The shader would ultimately load the glyph atlas as a lookup texture, picking the correct quadrants (quarters of the glyph) from that texture and mirror and rotate these quadrants to completely draw the glyph itself. Since the texture coordinates can easily be modified inside the shader, we allow for different halves to be drawn in the glyph thus enabling more attributes to be mapped to the glyph. Size, rotation and aspect ratio is also performed by adjusting the texture coordinates to achieve the desired effect.

We choose Super Ellipses as a basis for our glyph shapes. These shapes are simple to understand, and easy to get to parameterized form. The super ellipses can be varied by changing their controlling exponent, from a square (low exponent) through circle and diamond shapes to star shape (with high exponents). This exponent is continuous and therefore well suited for having mapped data to it. These shapes are easy to distinguish from each other, and work very well to convey information of the data values they depict. By having two separate ellipses, one for top and one for bottom, we can map two different datavalues to these parameters. The glyph shapes were perceptually normalized to allow them to represent an even amount of change in the corresponding data. See figure 5 and figure 6. The area of cover of every glyph is also calculated, to allow for size normalization during the glyph instantiation step.

An advantage of simple shapes, is that the viewer of the visualization still can mentally complete the glyphs if they were to overlap and occlude each other. This quality, in addition to visual redundancy, makes simple glyphs very efficient in conveying their information.

We are able to map data to color, size, the two super ellipse halves, rotation and aspect ratio of the glyph. The glyphs can properly visualize six different parameters in addition to the DOI controlled opacity provided by the SimVis framework. These attributes are closely coupled with retinal variables described by [Bertin 1983]; shape, size, orientation and color (hue and value).

7 Summary and Conclusions

We present an effective way to allow user adjusting of data values that is both straightforward and comprehensible. All data variates may undergo the data mapping steps: windowing, exponentiation, and mapping. These are considered as easy to understand, but powerful tools that allow fine tuning the resulting glyph shapes. The data mapping stage inherently increases the value of the resulting glyphs.

The design of glyphs to be used in visualizations is both complex and crucial. We point out improvements of glyph design by discussing the glyphs inter property aspects (orthogonality and redundancy). We moreover propose to normalize glyph shapes both per-

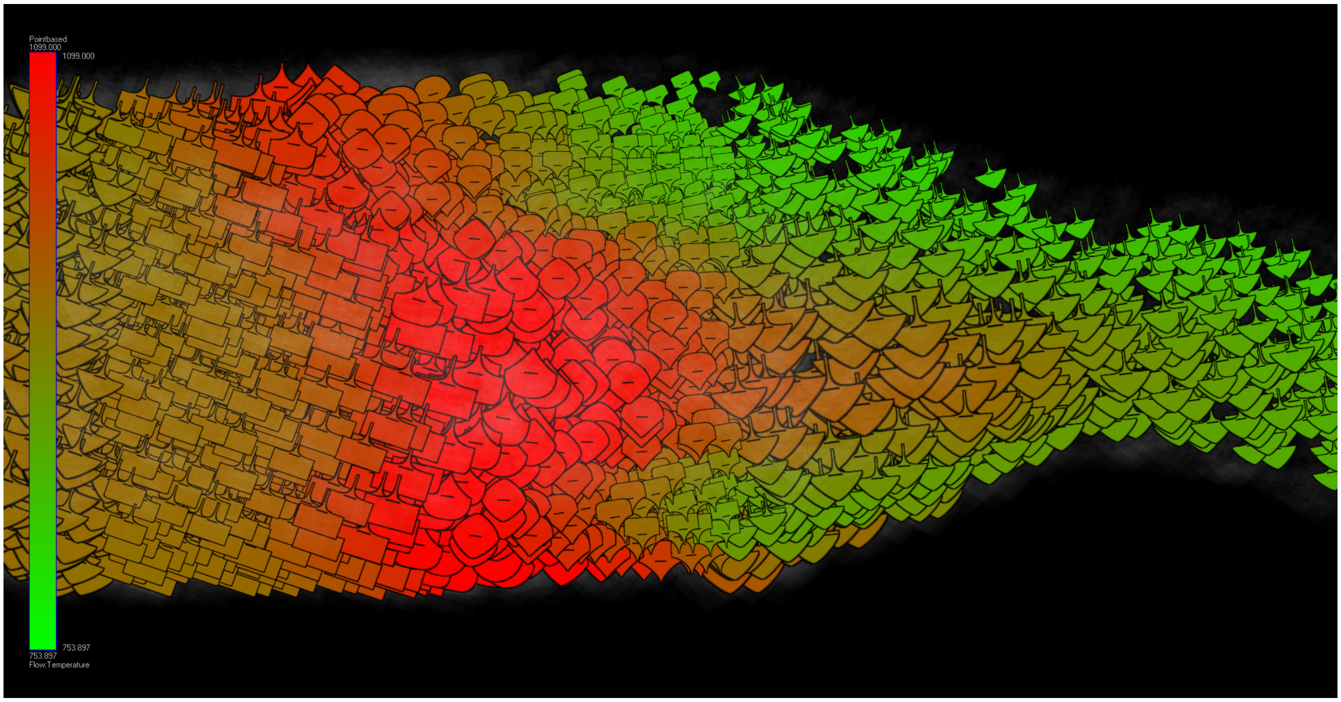


Figure 9: Diesel particle filter oxidizes soot at temperatures above 1000 degrees celsius. The color represents the temperature of the process. Upper glyph shape visualizes the amount of soot at that given point, and the lower shape the rate of rate of change. Rotation is the amount of O^2 which is needed for the oxidation process. Glyph size is mapped to temperature to achieve redundancy for the visualization. One can see that there exist two green areas right of the peak temperatures where there are high amounts of soot, and the oxidation is non-optimal because of lower temperatures.

ceptually and in size to avoid loss of orthogonality while maintaining clarity. 2D shapes are ultimately easier to interpret than their 3D counterparts, and we propose to only use the 3D glyphs when spatial relation is inherent.

We stress to use already existing techniques to help avoid problems with occlusion and cluttering. The use of halos help emphasize discontinuity. Chromadepth and interactive slicing help the user interpret depth.

8 Future Work

A user study would help emphasize the strengths and weaknesses of glyph-based visualizations, as well as feedback on the guidelines provided to create such glyphs. Another interesting angle would be to apply MPEG-7 shape descriptors to have perceptually focused metrics for glyphs.

9 Acknowledgments

We want to thank Helmut Doleisch and Phillip Muigg (SimVis GmbH, Austria) and Jean-Paul Balabanian (UIB, Norway) for guidance and help regarding SimVis and the implementation of our glyph based renderer. Thanks to Robert A. Johannessen (UIB, Norway) for proof-reading. The Diesel Exhaust System dataset is courtesy of AVL List GmbH, Graz, Austria. The Isabel hurricane dataset is courtesy of NCAR - The National Center for Atmospheric Research, USA.

References

BERTIN, J. 1983. *Semiology of graphics*. Univ. of Wisconsin Press.

BÜRGER, R., AND HAUSER, H. 2007. Visualization of multivariate scientific data. In *EuroGraphics 2007 State of the Art Reports (STARs)*, 117–134.

DE LEEUW, W. C., AND VAN WIJK, J. J. 1993. A probe for local flow field visualization. In *Proc. IEEE Visualization '93 (Vis '93)*, 39–45.

DOLEISCH, H., MAYER, M., GASSER, M., WANKER, R., AND HAUSER, H. 2004. Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proc. of IEEE TVCG/EG Symposium on Visualization 2004*, 91–96.

DOLEISCH, H., MUIGG, P., AND HAUSER, H. 2004. Interactive visual analysis of hurricane isabel with simvis. Tech. Rep. TR-VRVis-2004-058, VRVis Research Center, Vienna Austria. http://www.vrvis.at/simvis/Isabel/.

DOLEISCH, H. 2007. SimVis: interactive visual analysis of large and time-dependent 3d simulation data. In *Winter Simulation Conference*, 712–720.

HAUSER, H., AND SCHUMANN, H. 2009. Visualization pipeline. In *Encyclopedia of Database Systems*, Springer LINK, M. T. Özsu and L. Liu, Eds., 1145–1146.

INTERRANTE, V., AND GROSCH, C. 1998. Visualizing 3d flow. *IEEE Comput. Graph. Appl. 18*, 4, 49–53.

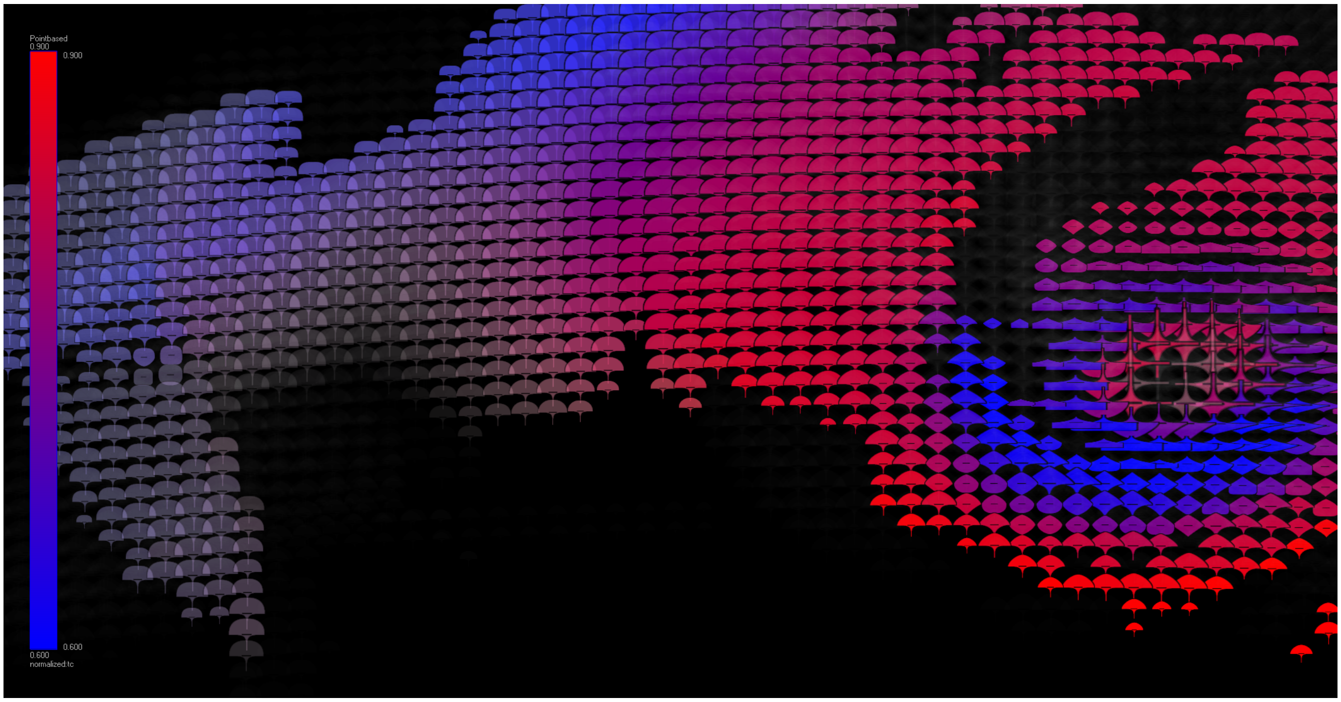


Figure 10: The hurricane Isabel dataset, timestep 12. Color represents temperature, and the amount of clouds is mapped to glyph size. The upper shape represents pressure, and the lower shape precipitation. The visualization depicts the fast moving clouds (specified via brushing), and the eye of the hurricane is visible in the lower right, surrounded by high amounts of precipitation and cold airflow.

JANKUN-KELLY, T., AND MEHTA, K. 2006. Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualization. *IEEE Trans. Vis. Comput. Graph. 12*, 5, 1197–1204.

KINDLMANN, G., AND WESTIN, C.-F. 2006. Diffusion tensor visualization with glyph packing. *IEEE Trans. Vis. Comput. Graph. 12*, 5, 1329–1336.

KINDLMANN, G. 2004. Superquadric tensor glyphs. In *Proc. of IEEE TVCG/EG Symposium on Visualization 2004*, 147–154.

MEYER-SPRADOW, J., STEGGER, L., DÖRING, C., ROPINSKI, T., AND HINRICHS, K. 2008. Glyph-based spect visualization for the diagnosis of coronary artery disease. *IEEE Trans. Vis. Comput. Graph. 14*, 6, 1499–1506.

OELTZE, S., MALYSZCZYK, A., AND PREIM, B. 2008. Intuitive Mapping of Perfusion Parameters to Glyph Shape. In *Bildverarbeitung für die Medizin (BVM2008)*, 262–266.

PICKETT, R., AND GRINSTEIN, G. 1988. Iconographic displays for visualizing multidimensional data. In *Proc. of IEEE, Systems, Man, and Cybernetics*, vol. 1. Aug, 514–519.

PIRINGER, H., KOSARA, R., AND HAUSER, H. 2004. Interactive focus+context visualization with linked 2d/3d scatterplots. In *Proc. of the Intl. Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2004)*, 49–60.

ROPINSKI, T., AND PREIM, B. 2008. Taxonomy and usage guidelines for glyph-based medical visualization. In *Proc. Simulation and Visualization 2008*, 121–138.

ROPINSKI, T., SPECHT, M., MEYER-SPRADOW, J., HINRICHS, K., AND PREIM, B. 2007. Surface glyphs for visualizing multimodal volume data. In *Proc. Vision, Modelling, and Visualization*, 3–12.

SAWANT, A. P., AND HEALEY, C. G. 2007. Visualizing flow data using assorted glyphs. *Crossroads 14*, 2, 1.

SHAW, C. D., HALL, J. A., BLAHUT, C., EBERT, D. S., AND ROBERTS, D. A. 1999. Using shape to visualize multivariate data. In *Workshop on New Paradigms in Information Visualization and Manipulation*, 17–20.

STOLTE, C., TANG, D., AND HANRAHAN, P. 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Vis. Comput. Graph. 8*, 1, 52–65.

TOUTIN, T. 1997. Qualitative aspects of chromo-stereoscopy for depth-perception. *Photogrammetric Engineering and Remote Sensing 63*, 2, 193–203.

TUFTE, E. R. 2001. *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press, Cheshire, Connecticut.

VAN WALSUM, T., POST, F. H., SILVER, D., AND POST, F. J. 1996. Feature extraction and iconic visualization. *IEEE Trans. Vis. Comput. Graph. 2*, 2, 111–119.

WARD, M. O. 2002. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization 1*, 3–4, 194–210.

WARD, M. 2008. Multivariate data glyphs: Principles and practice. In *Handbook of Data Visualization*, C.-H. Chen, W. Härdle, and A. Unwin, Eds. Springer, 179–198.

WONG, P. C., FOOTE, H., KAO, D. L., LEUNG, L. R., AND THOMAS, J. 2002. Multivariate visualization with data fusion. *Information Visualization 1*, 3–4, 182–193.