Metadata of the chapter that will be visualized in SpringerLink

Book Title	Scientific Visualization	
Series Title		
Chapter Title	Glyph-Based Multifield Visualization	
Copyright Year	2014	
Copyright HolderName	Springer-Verlag London	
Corresponding Author	Family Name	**Chung**
	Particle	
	Given Name	**David H. S.**
	Prefix	
	Suffix	
	Division	
	Organization	Swansea University
	Address	Swansea, UK
	Email	d.h.s.Chung@swansea.ac.uk
Author	Family Name	**Laramee**
	Particle	
	Given Name	**Robert S.**
	Prefix	
	Suffix	
	Division	
	Organization	Swansea University
	Address	Swansea, UK
	Email	r.s.Laramee@swansea.ac.uk
Author	Family Name	**Kehrer**
	Particle	
	Given Name	**Johannes**
	Prefix	
	Suffix	
	Division	
	Organization	University of Bergen
	Address	Bergen, Norway
	Email	johannes.kehrer@uib.no
Author	Family Name	**Hauser**
	Particle	
	Given Name	**Helwig**
	Prefix	
	Suffix	
	Division	
	Organization	University of Bergen
	Address	Bergen, Norway

	Email	helwig.hauser@uib.no
Abstract	■■■	

Chapter 13
Glyph-Based Multifield Visualization

David H. S. Chung, Robert S. Laramee, Johannes Kehrer and Helwig Hauser

~~**Abstract** ■■■~~

13.1 Introduction

The visualization of data that are given as fields of values is a classical topic in visualization research. A substantial amount of relevant work has been done, offering a wealth of well-proven techniques for revealing insight into such data fields. When visualizing multiple fields of data that co-exist with respect to a joint domain of reference, additional challenges are faced. One the one hand, there is a *technological challenge* of how to realize a visualization mapping that can reveal multiple fields of data at a time. On the other hand, there is a *perceptual challenge* of how easy it is to understand and correctly interpret such a visualization. AQ1

Glyph-based visualization is one possible approach to realize such a visualization of multi-field data (and other chapters of this book part describe alternative approaches). A parameterized visualization object is considered—called a *glyph* (or sometimes also an icon)—such that certain specifics with respect to its form, e.g., its shape, color, size/orientation, texture, etc., are given according to data values which this glyph should represent. A glyph-based visualization is then created by arranging a certain number of these glyphs across the domain of reference (these could be just a few, or just one, or many, even so many that they merge into a dense visualization) such that every glyph becomes a visualization of the data at (or nearby) the location where the glyph is placed.

D. Chung (✉) · R.S. Laramee
Swansea University, Swansea, UK
e-mail: d.h.s.Chung@swansea.ac.uk

J. Kehrer · H. Hauser
University of Bergen, Bergen, Norway
e-mail: helwig.hauser@uib.no

C.D. Hansen et al. (eds.), *Scientific Visualization*, Mathematics and Visualization,
DOI 10.1007/978-1-4471-6497-5_13

326752_1_En_13_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:2/7/2014 Pages: 9 Layout: T1-Standard

Glyph-based visualization approaches span a certain spectrum from, for example, dense arrangements of relatively simple shapes (stick figures would be an example) to individual instances of complex glyphs that reveal a lot of information (but only for few, selected places)—the local flow probe would be an example for this type of a glyph-based data visualization. Glyph-based visualization approach also vary with respect to whether they are constructed in a 2D or 3D visualization space. We think that it also makes sense to consider glyph-based visualization approaches, which are based on the placement of glyphs on surfaces within 3D (called 2.5D in the following). Additionally, we can differentiate visualization solutions according to which form aspects are varied according to the data, and how many different values a glyph eventually represents (usually this number is not too large, often 2–4, but then also examples exist where dozens of values are represented).

A property of all glyph-based visualization approaches is that a discrete visualization is created (instead of a continuous representation like a color map)—only at certain locations across the domain individual glyphs are instantiated to represent the data. This means that this approach is only suitable, when it is possible to assume a certain minimal degree of continuity of the data such that a mental reconstruction of the data, in particular also in the space between the glyphs, is at least principally possible. In scientific visualization, this often is possible, making glyph-based visualization particularly interesting for this particular field of application. Alternatively, a glyph-based visualization also makes sense for discrete data, if a one-to-one relation between every instance of the data and the glyphs is established.

In the following, we first review a selection of techniques that have been proposed for glyph-based data visualization. Then, we continue with a discussion of critical design aspects of glyph-based visualization, not at the least oriented at opportunities to deal with the perceptual challenge that is inherently associated with this form of visualization approach.

13.2 State-of-the-Art

This section presents a selection of important papers with a focus on glyph-based multifield visualization. A categorization is given based on the visual channels such as color, shape, size, texture and opacity occupied by the glyph in requirement for mapping each data attribute. We further cluster the techniques with respect to the spatial dimensionality of the visualization e.g., 2D, 2.5D and 3D. Texture can be subjective in terms of glyph-based classification, however, we find that it is very relevant in the research of multifield. The following work can be acknowledged without the use of this classification, but we include this in the table for completeness.

Table 13.1 Table illustrating a classification of multi-variate glyph-based visualization techniques based on the visualization dimensionality and the visual channels required to depict the data set

Visual channel	Visualization dimensionality		
	2D	2.5D	3D
Color	[5] [11] [21]	[3] [6] [16]	[20] [12] [9] [2] [10] [15] [8]
Shape	[8]		[1] [13] [20] [9] [7] [10] [15]
Size	[24] [21] [19]	[3] [16]	[20] [9] [2] [15]
Texture	[21]	[3] [6]	
Opacity	[11] [21]		[15]

13.2.1 Spatial Dimensionality: 2D

A common technique for representing multi-field data is to overlay multiple visualizations onto a single image. Kirby et al. [11] stochastically arrange multiple visualization layers to minimize overlap. Given a permutation of layers, a user-specified importance value is attached to each visualization of increasing weights in order to provide greater emphasis to higher layers. Visual cues such as color and opacity indicate regions and layers of importance (e.g., Rate of strain tensor example emphasized the velocity more by using black arrows). This method enables the simultaneous depiction of 6–9 data attributes, in which the authors apply to a simulated 2D flow field past a cylinder at different reynolds number. The example shows the visualization of velocity, vorticity, rate of strain tensor, turbulent charge and turbulent current.

Visualizing Multiple Fields on the Same Surface by Taylor [21] provides an overview of successful and unsuccessful techniques for visualizing multiple scalar fields on the same surface. The author first hypothesizes that the largest number of data sets that can be displayed by mapping each field to the following: a unique surface characteristic, applying a different visualization technique to each scalar field or by using textures/glyphs whose features depend on the data sets. This framework is limited to visualizing up to four scalar fields. The author then describes two techniques that prove effective for visualizing multiple scalar fields, (1) *data-driven spots (DDS)*—using different spots of various intensities and heights to visualize each data set, and (2) *oriented slivers*—using sliver like glyphs of different orientations that are unique to each data set along with various blending.

13.2.2 Spatial Dimensionality: 2.5D

A Scientific Visualization Synthesizer by Crawfis and Allison [3] introduces a novel approach for visualizing multiple scientific data sets using texture mapping and raster operations. The authors present an interactive programming framework that enables users to overlay different data sets by defining raster functions/operations. Using a generated synthetic data, the author presents a method for reducing the visual clutter by mapping color to a height field and using a bump map to represent the vector plots and contour plots. The final texture is mapped onto a 3D surface.

Peng et al. [16] describes an automatic vector field clustering algorithm and presents visualization techniques that incorporate statistical-based multi-variate glyphs. In summary, the authors clustering algorithm is given by: (1) derive a mesh resolution value for each vertex, (2) encode vector and mesh resolution values into R, G, B and α in image space. Clusters naturally form in this space based on pixel values. (3) The clusters are merged depending on a similarity value derived using euclidean distance, mesh resolution, average velocity magnitude and velocity direction. Several clustering visualizations are given, using $|v|$-range glyph that depicts the local minimum and maximum vector, and a θ-range glyph that shows the variance of vector field direction along with the average velocity direction and magnitude. Other visualization options include streamlets that are traced from the cluster centre, and color coding with mean velocity. The authors demonstrate their clustering results on a series of synthetic and complex, real-world CFD meshes.

13.2.3 Spatial Dimensionality: 3D

Geometric shapes are often used to represent multiple data values. Superquadrics and Angle-Preserving Transformations by Barr [1] introduces such an approach for creating and simulating three-dimensional scenes. The author defines a mathematical framework used to explicitly define a family of geometric primitives (superquadrics) from which their position, size, and surface curvature can be altered by modifying a set of different parameters. Example glyphs include a torus, star-shape, ellipsoid, hyperboloid or toroid. In addition, the paper describes a group of invertible transforms developed to bend and twist mathematical objects in three dimensions into a new form where shape properties such as volume, surface area and arc length is conserved.

de Leeuw and van Wijk [13] present an interactive probe-glyph for visualizing multiple flow characteristics in a small region. In particular, the authors focus on visualizing six components: velocity, curvature, shear, acceleration, torsion and convergence. The construction of the glyph is given by, (1) a curved vector arrow where the length and direction represents the velocity, and the arc shape is mapped to the curvature, (2) a membrane perpendicular to the flow where its displacement to the center is mapped to acceleration, (3) candy stripes on the surface of the velocity arrow illustrates the amount of torsion, (4) a ring describes the plane perpendicular to the flow over time (shear-plane), and finally (5) the convergence and divergence of the flow is mapped to a "lens" or osculating paraboloid. Placement of such probes are interactively placed by users along a streamline to show local features in more detail.

Data Visualization Using Automatic, Perceptually-Motivated Shapes by Shaw et al. [20] describes an interactive glyph-based framework for visualizing multi-dimensional data through the use of superquadrics. The author uses the set of superquadrics defined by Barr [1] and describes a method for mapping data attributes appropriately to shape properties such that visual cues effectively convey data dimensionality without depreciating the cognition of global data patterns. They map in decreasing order of data importance, values to location, size, color and shape (of which two dimensions are encoded by shape). Using superellipsoids as an example, the authors applied their framework on two different data sets.

Superquadric Tensor Glyphs by Kindlmann [9] introduces a novel approach of visualizing tensor fields using superquadric glyphs. Superquadric tensor glyphs address the problems of asymmetry and ambiguity prone in previous techniques (e.g. cuboids and ellipsoids). The author provides an explicit and implicit parameterization of the primitives defined by Barr [1] that uses geometric anisotropy metrics c_l, c_p, c_s to quantify the certainty of a tensor based on shape, and a user-controlled edge sharpness parameter γ. The parametrization forms a barycentric triangular domain of tensor glyphs that change in shape, flatness and orientation under different tensor eigen vectors. A subset of the family of superquadrics is chosen and applied towards visualizing a DT-MRI tensor field which is then compared against an equivalent ellipsoid visualization.

13.3 Critical Design Aspects of Glyph-Based Visualization

It was a wide-spread opinion for a long time that "just" knowing the basic principles of glyph-based visualization would suffice to its successful usage. More recently, however, it has been understood that only well designed glyphs, where different glyph properties are carefully chosen and combined, are actually useful. In this section, we discuss critical design aspects and guidelines for glyph-based visualization.

326752_1_En_13_Chapter ✓ TYPESET ☐ DISK ☐ LE ✓ CP Disp.:2/7/2014 Pages: 9 Layout: T1-Standard

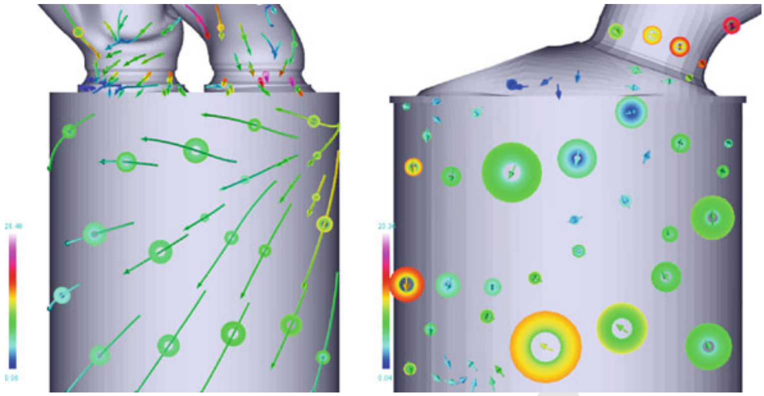

Fig. 13.1 Visualization of the flow in an engine using composite glyphs that depict the range of vector magnitude and direction in each cluster by Peng et al. [16]

In the context of information visualization, Ward [23] discusses glyph placement strategies such as data- or structure-driven placement. Ropinski and Preim [18] 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 [9], for example, use superquadric glyph shapes that fulfill the latter criterion.

Inspired by the work of Ropinski and Preim, Lie et al. [14] propose further guidelines for glyph-based 3D visualization. Aligned with the visualization pipeline [4], the task of creating a glyph-based 3D visualization is divided into three stages as shown in Fig. 13.2: (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.

Similar to Ward [23], Lie et al. 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 $[w_{left}, w_{right}]$ 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 $[w_{left}, w_{right}]$ and [0, 1], but also other forms of mapping could be considered (for example, a

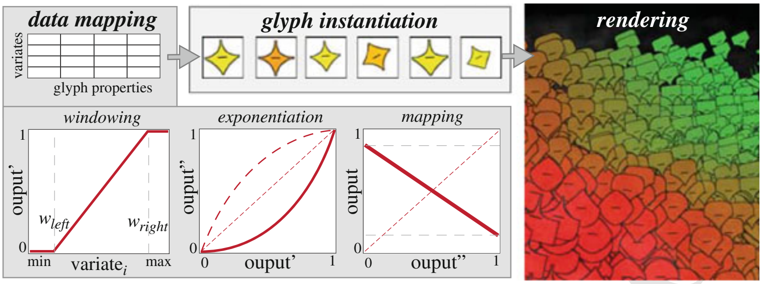

Fig. 13.2 Each data variate is subject to three stages of data mapping: windowing, exponentiation and mapping. The values are mapped to different glyph properties and used to instantiate the individual glyphs. Finally, the glyphs are rendered in their spatial context

ranking-based or discontinuous mapping). After the windowing, an optional exponential mapping $e(x) = x^\gamma$ 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 [18]). 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.

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 [9]. In order to avoid this problem, Lie et al. suggest to use 2D billboard glyphs instead.[1] 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 [18]). 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 [13] 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 [10]. Another design guideline is the usage of *redundancies*, for

[1] A billboard is a planar structure placed in a 3D scene, which automatically adjusts its orientation such that it always faces the observer.

instance, to use symmetries that ease the reconstruction of occluded parts of the glyph. Important properties can, moreover, be mapped to multiple glyph properties in order to reduce the risk of information loss.

Important aspects when rendering many glyphs in a dense 3D context are depth perception, occlusion, and visual cluttering. In cases where many glyphs overlap, *halos* can help to enhance the depth perception and to distinguish individual glyphs (compare to Piringer et al. [17]). For improving the depth perception for non-overlapping glyphs a special color map (called *chroma depth* [22]) can be used to represent depth. Finally, appropriate glyph placement [18, 23], interactive slicing, or filtering via brushing are strategies for dealing with occlusion and cluttering issues.

References

1. Barr, A.H.: Superquadrics and angle-preserving transformations. IEEE Comput. Graph. Appl. **1**(1), 11–23 (1981)
2. Chlan, E.B., Rheingans, P.: Multivariate glyphs for multi-object clusters. In: INFOVIS, p. 19 (2005)
3. Crawfis, R.., Allison, MJ.: A scientific visualization synthesizer. In: Proceedings of the IEEE visualization, pp. 262–267 (1991)
4. Hauser, H., Schumann, H.: Visualization pipeline. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of database systems, pp. 3414–3416. Springer, Berlin (2009)
5. Healey, C., Booth, K., Enns, J.: High-speed visual estimation using preattentive processing. ACM Trans. Comput.-Human Interact. **3**, 107–135 (1996)
6. Healey, C.G., Enns, J.T.: Large datasets at a glance: combining textures and colors in scientific visualization. IEEE Trans. Vis. Comput. Graph. **5**(2), 145–167 (1999)
7. Jankun-Kelly, T., Mehta, K.: Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualization. IEEE Trans. Vis. Comput. Graph. **12**(5), 1197–1204 (2006)
8. Kehrer, J., Muigg, P., Doleisch, H., Hauser, H.: Interactive visual analysis of heterogeneous scientific data across an interface. IEEE Trans. Vis. Comput. Graph. **17**(7), 934–946 (2011)
9. Kindlmann, G.: Superquadric tensor glyphs. In: Deussen, O., Hansen, C., Keim, D., Saupe, D. (eds.) Joint Eurographics—IEEE TCVG symposium on visualization, Konstanz, Germany, pp. 147–154 (2004)
10. Kindlmann, G., Westin, C.F.: Diffusion tensor visualization with glyph packing. IEEE Trans. Vis. Comput. Graph. **12**(5), 1329–1336 (2006)
11. Kirby, R.M., Marmanis, H., Laidlaw, D.H.: Visualizing multivalued data from 2D incompressible flows using concepts from painting. In: Proceedings of the IEEE visualization, pp. 333–340 (1999)
12. Kraus, M., Ertl, T.: Interactive data exploration with customized glyphs. In: WSCG (Posters), pp. 20–23 (2001)
13. de Leeuw, W.C., van Wijk, J.J.: A probe for local flow field visualization. In: Proceedings of the IEEE visualization conference (Vis'93), pp. 39–45 (1993)
14. Lie, A.E., Kehrer, J., Hauser, H.: Critical design and realization aspects of glyph-based 3D data visualization. In: Proceedings of the spring conference on computer graphics (SCCG 2009), pp. 27–34 (2009)
15. Meyer-Spradow, J., Stegger, L., Döring, C., Ropinski, T., Hinrichs, K.: Glyph-based SPECT visualization for the diagnosis of coronary artery disease. IEEE Trans. Vis. Comput. Graph. **14**(6), 1499–1506 (2008)

16. Peng, Z., Grundy, E., Laramee, R., Chen, G., Croft, N.: Mesh-driven vector field clustering and visualization: an image-based approach. IEEE Trans. Vis. Comput. Graph. (TVCG) (2011) AQ2
17. Piringer, H., Kosara, R., Hauser, H.: Interactive focus+context visualization with linked 2D/3D scatterplots. In: Proceedings on coordinated and multiple views in exploratory visualization (CMV 2004), pp. 49–60 (2004)
18. Ropinski, T., Preim, B.: Taxonomy and usage guidelines for glyph-based medical visualization. In: Proceedings on simulation and visualization, pp. 121–138 (2008)
19. Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., Moorhead, R.: Noodles: a tool for visualization of numerical weather model ensemble uncertainty. IEEE Trans. Vis. Comput. Graph. **16**(6), 1421–1430 (2010)
20. Shaw, C.D., Ebert, D.S., Kukla, J.M., Zwa, A., Soboroff, I., Roberts, D.A.: Data visualization using automatic, perceptually-motivated shapes. In: SPIE 3298, visual data exploration and analysis (1998)
21. Taylor, R.: Visualizing multiple fields on the same surface. IEEE Comput. Graph. Appl. **22**(3), 6–10 (2002)
22. Toutin, T.: Qualitative aspects of chromo-stereoscopy for depth-perception. Photogram. Eng. Remote Sens. **63**(2), 193–203 (1997)
23. Ward, M.O.: A taxonomy of glyph placement strategies for multidimensional data visualization. Info. Vis. **1**(3–4), 194–210 (2002)
24. Wittenbrink, C.M., Pang, A., Lodha, S.K.: Glyphs for visualizing uncertainty in vector fields. IEEE Trans. Vis. Comput. Graph. **2**(3), 266–279 (1996)

Author Queries

Chapter 13

Query Refs.	Details Required	Author's response
AQ1	If you wish to have 'abstract' in the chapter opening pages, please provide it. This is mandatory.	
AQ2	Please supply page ranges for Ref. [16].	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	