Design considerations on glyph placement strategies

Dirk Streeb*
University of Konstanz

Johannes Fuchs†
University of Konstanz

ABSTRACT

While glyph design is well researched and several placement techniques have been suggested, how to place glyphs in practice is not straightforward. Based on literature, we structure the problem space of glyph placement in three main categories: context-driven placement, placement of data collections, and placement of data samples. Following this categorization, we discuss several design considerations. Additionally, we highlight dependencies on task, user and data that prohibit the formulation of generally applicable guidelines.

1 INTRODUCTION

Glyphs are a common technique for visualizing high-dimensional data. Accordingly, a lot of research has been done on how to design glyphs as well as how to choose appropriate glyph designs [2, 5, 6]. However, designing individual glyphs is only part of the work to be accomplished in order to achieve a useful glyph visualization [21]. One of the big strengths of glyphs is that they are independent of each other, and as a result offer the freedom to be placed arbitrarily. While we see a multitude of techniques for placing glyphs, we do not find a compiled set of design considerations aiding in the selection of a placement technique. Since position is a precise visual variable that may produce pre-attentively perceivable stimuli [2], an appropriate placement strategy is worth pursuing.

In this paper, we address the lack of guidance by providing a well-structured overview on the problem space. Although, we cannot offer a single guideline that "describes a process or a set of actions that may lead to a desired outcome or, alternatively, actions to be avoided to prevent an undesired outcome" [4, glyph on p. 105] as placing glyphs is strongly context-dependent [19]. Instead, we lay out design considerations pointing researchers and practitioners to trade-offs, and relevant decisions outlined in Fig. 1.

When thinking about glyph placement, it is important to note some fundamental distinctions (see Fig. 1). In the work of Ward [19] "glyph placement" clearly denotes the positioning of a given collection of data objects. For Ropinski and Preim [16], however, "glyph placement" is about choosing an appropriate sample from an—approximately—continuous data space, and placing this sample. In their case sampling and placement are interdependent, while in the case of Ward the sample is given and fixed. Lastly, Goffin et al. [7] research how to place glyphs when augmenting given textual structures. Contrary to the first two cases, glyphs are not positioned on an otherwise empty canvas.

Based on these classes, we constructed the diagram in Fig. 1. The first question to ask is whether a given context puts additional constraints into place. Only if glyphs are positioned on an empty canvas, there are no contextual constraints. Otherwise, glyph placement is largely determined by the preexisting context. Hence, other placement strategies become relevant, if there are enough degrees of freedom. Then, the major factor is the distinction described above: Is the collection of glyphs fixed, or does placement involve a sampling process?

*e-mail: streeb@dbvis.inf.uni-konstanz.de
†e-mail: fuchs@dbvis.inf.uni-konstanz.de

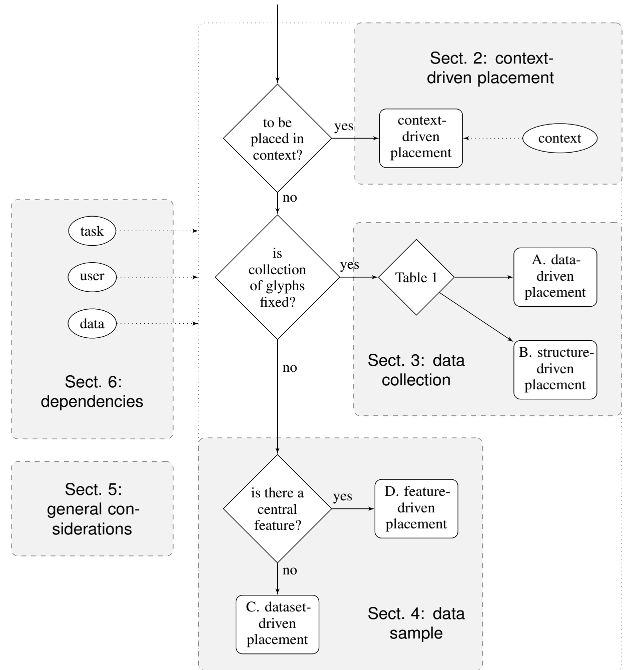

Figure 1: How to decide which placement strategies are appropriate. Many decisions not only depend on the data, but also on the task and the end user. Grey boxes denote sections of this paper, starting in the top right and continuing clockwise.

The remainder of this paper follows the structure of Fig. 1. Each grey box represents one of the main sections: First we turn to placing glyphs in restricted contexts in Sect. 2. Then, we investigate data collections (Sect. 3) and data samples (Sect. 4). In Sect. 5 we point out more general considerations. Finally, in Sect. 6 we put considerations introduced before in relation to the task, the user, and the data at hand. In Sect. 7 we conclude with pointing out relevant future work.

2 CONTEXT-DRIVEN PLACEMENT

Context-driven placement strategies can be employed when glyphs are to be placed in pre-defined structures. In this respect, they do not assume an empty canvas. Glyphs relate to some parts of the context, e.g., specific words in a text. Placing glyphs in close proximity to respective contextual items limits the area of reasonable positions. Besides that, the visual context itself puts constraints on attainable placements.

For example, Goffin et al. [7] compare numerous strategies for placing glyphs within a body of text. They advise to use the space between lines where available. Further, they found that placing visualizations to the right on the same line often saves space compared to adding more inter-line space. Obviously, adding space is only possible with text bodies that may be reflowed.

In other scenarios, like scanned text or augmented reality, the context is fixed and can not be modified. Thus, only available space can be used. While Goffin et al. [7] include the case of fixed textual contexts, we are not aware of an extensive set of glyph placement design considerations for augmented reality.

3 Placement of Data Collections

If there is no fixed context, glyphs can be positioned on an empty canvas without initial constraints, e.g., the glyphs may not occlude words in a text. However, we have to distinguish between glyphs generated from a fixed collection of observations, i.e., a data collection (this section), or sampled from a continuous data space, i.e., a data sample (Sect. 4).

Following Ward [19], we distinguish between *data-driven placement* and *structure-driven placement* of data collections. While the two categories are not clearly distinct, the question to ask is whether there is a linear, cyclic, hierarchical, geo-spatial, or otherwise relational structure to be highlighted. Otherwise, placing glyphs based on values of attributes is likely the way to go (see also Table 1).

A. Data-driven Placement With data-driven placement there are basically two options. First, selecting two highly-relevant attributes and placing glyphs in a scatter plot. Secondly, using a dimensionality reduction technique—like Principal Component Analysis—to embed glyphs on a plane. One aspect to consider is that in "data driven approaches, there are usually mappings of data dimensions to position that make more intuitive sense than others." [19, p. 207] Selecting those intuitive dimensions may help domain experts in working with the glyph visualization.

Overplotting is a common problem with data-driven placement strategies to which jittering or distorting positions may produce relief [19]. For example, Keim and Herrmann [9] exploit quadtrees to move conflicting glyphs to nearby locations.

B. Structure-driven Placement Contrary to data-driven placement, structure-driven placement highlights relationships between data points, as in a network, a linear or cyclic order, or a hierarchy. While the structure may be part of the original data set—as in the case of (seasonal) time series—it also may be calculated, e.g., using a clustering technique to gain a hierarchical structure.

"Similarly [to data-driven placement], the dimensions used to control structure-driven techniques, whether it be the key for sorting or the order of dimensions used for layering a hierarchy, are usually best left to selection by the user." [19, pp. 207f] Especially when imposing a structure, "user control of the process is vital, and this in turn requires users to be knowledgeable not only of the semantics of their data but also of the structuring algorithm being utilized." [19, p. 208] In that sense, Ward [19] points out that possibilities for user interaction are crucial in many cases as several placement design decisions are conditional on the structural patterns in the data.

An additional option to improve structure-driven placements is the insertion of white space to emphasize differences between adjacent glyphs [19]. For example, Meulemans et al. [11] use gaps in two-dimensional placement. Inserting white space can be seen as a combination of structure and data-driven methods [20]. As a result, we repeat the point already mentioned above: The distinction between data-driven placement and structure-driven placement is not clear cut, but it raises an important question, namely what is the relevant pattern to show.

4 Placement of Data Samples

Placing glyphs when sampling from an (approximately) continuous space complicates glyph positioning. The sampling process and the placement of glyphs depend on each other. This is different to placing glyphs generated from data collections because both cannot be separated. Choosing to sample a point from the continuum determines that the respective glyph should be positioned in close proximity to the sampling location. The other way round, placing a glyph at a specific location raises the expectation that underlying data was sampled nearby.

Ropinski and Preim [16] distinguish between two placement strategies: *dataset-driven placement* and *feature-driven placement*. Dataset-driven placement is guided by the data structures used to measure, simulate or store the continuous object of interest, e.g., the grid in a fluid dynamics simulation. Contrarily, feature-driven placement strategies position glyphs at locations that are meaningful features of the object of interest, e.g., isolines or isosurfaces.

C. Dataset-driven Placement The major problem with dataset-driven placement is that the rigid, often regular, structure may lead to misleading aggregations, overlapping glyphs, and sparse areas. Or, in the words of Ropinski, Oeltze and Preim: "the regular grid usually has a major influence on the visualization. Thus it can unintentionally emphasize or even feign a non-existent glyph aggregation." [15, p. 395] To avoid these negative effects at least a jittering should be applied [16]. In general, however, a feature-driven placement may be preferable.

Alternatively, glyph packing [22] places glyphs with densities depending on the glyphs' shapes instead of the grid structure, thus avoiding overplotting. Thereby, artifacts induced by the grid structure can be avoided as glyph packing reliably leads to textual appearance [16].

D. Feature-driven Placement Similar to data-driven placement strategies, feature-driven placement strategies consider values of underlying data for the placement of glyphs. An example from weather forecasts is placing arrow glyphs denoting wind direction and speed along streamlines [14]. This way feature-driven placement can be used to highlight correlations between features of the data space and data values encoded in glyphs.

Another example for feature-driven placement originates from medical imaging [17]. In this domain it is often key to place glyphs on the surface of objects. However, in three-dimensional medical images objects like a skull are not a priori distinct, but only distinguishable from surrounding tissue by its features. Therefore, feature-driven placement strategies often can offer more reasonable results than dataset-driven placement strategies.

5 General Considerations

Besides deciding for a placement strategy, there are some more general considerations to be taken into account. To begin with, Ropinski et al. [17] as well as Lie, Kehrer and Hauser [10] point out that glyph visualizations are more comprehensible whenever less glyphs are shown at a time. For example, the number of visible glyphs can be limited by interactive slicing in order to reduce overlap [10]. Generally, placement density should be put in relation to glyph complexity, the simpler the glyphs, the denser they can be placed [2].

Borgo et al. advise balanced glyph placement [2]. For example, glyph packing strategies usually are sufficient to avoid misleading aggregations, overlap and sparse regions [16]. Ward [19] suggests adding distortion to reduce overlap. Pickett and Grinstein [13], on the other hand, do not care too much about overlap. For them regular placement is crucial for producing emergent visual patterns. Also, Ward notes that since "distortion introduces error into the visual presentation, it is best to allow users to control the amount of distortion applied by either setting the maximum displacement for an individual glyph or the average among all glyphs, or by using animation to show the movement of glyphs from their original positions to their distorted positions." [20, p. 9]

Depending on the density of glyphs in the image, there is the option to add context to the glyph view. Such hybrid visualizations are rather common in scientific visualization, and should be exploited to augment the glyph visualization with spatial context [2, 16]. In information visualization, hybrid glyph visualizations are to the best of our knowledge only rarely used.

When placing glyphs in three dimensions, additional measures may aid in depth perception [2]. Such measures, for example, can

Data Characteristics	User Task	Recommended Strategy
Small to moderate size	univariate analysis	sorted structure-driven
Small to moderate size	bivariate analysis	raw data-driven
Small to large size	outlier detection	raw or derived data-driven
Moderate to large size	cluster analysis	derived data-driven, e.g., MDS
Small to moderate size	cluster analysis	hierarchical structure-driven after imposing a hierarchy
Small to moderate size, relational	link analysis	network structure-driven

Table 1: Initial guidelines for the choice of placement strategy for data collections by Ward [19, Figure. 13].

be halos added to glyphs [10]. Further, Borgo et al. suggest avoiding "perspective projections when using glyph size to encode a data variable" [2, DG14] as the placement along the depth dimension of the three dimensional space would otherwise distort size.

6 Dependencies on Task, User and Data

Design considerations discussed above point to more or less general aspects of glyph placement. Nonetheless, it has been highlighted repeatedly that glyph placement is strongly dependent on the problem context spanned by the task, the user and the data. In this section, we direct attention to these three aspects.

Task Most centrally, relevant tasks must be supported by good glyph placements [3]. Hence, the choice of visualization space—planar or three-dimensional—should be based on tackled tasks [2, 17]. As also discussed for other visualizations three dimensional visualizations are not automatically superior [3, 10]. The task will guide how to compromise between complexity of glyph design and density of placement [2, 10]. Similarly, placing bar chart glyphs on common baselines is beneficial for comparison tasks.

User The dependency on the user comes in two fashions. First, there is the repeated call for including the user in the placement of glyphs. Borgo et al. [2] even go as far as talking about user-driven placement. Specifically, users know best what they are interested in and can make trade offs, as for example put forward by Ward: "Placement techniques often force the user to trade off between efficient screen utilization, the degree of occlusion, and distortion of the values being used to position the glyphs. Ideally, the user should be able to dynamically adjust the mapping to strengthen or weaken the relative importance of each of these factors." [19]

Secondly, glyph placement is user-dependent as finally users are the ones who perceive the glyph visualization. Therefore, Carr [3] calls for testing designs with users.

Data Clearly data-driven placement depends on the data, but also other placement strategies depend on the main features of the dataset. As mentioned above, glyph size and spacing in between adjacent glyphs is dependent on the resolution of the dataset. Therefore, Ropinski and Preim conclude that "general guidelines cannot be proposed" [16, p. 14] for the placement of data samples. "The main point is that no single placement strategy is good for all tasks and data characteristics, which implies that for other than very limited circumstances, support should be provided for a variety of layout methods." [19, p. 208]

Usually these three types of dependencies appear in combination. In Table 1 we reproduce some guidelines provided by Ward [19, Figure. 13] that combine dataset size—as an example of data characteristics—and high level tasks. To the best of our knowledge, not much has been added since.

To give a practical example, Schreck, Keim and Mansmann [18] proposed grid treemap layouts to place glyphs according to a hierachy. They discuss three variants including nested, split lines and burst layouts, as well as quantum treemaps [1] (see also Fig. 2). As noted in the task paragraph, common baselines for bar chart glyphs can facilitate comparison. Thus, if the task involves comparing entities, the burst layout—which breaks the grid and thus the common baselines—should be avoided. On the other hand, it may be a good choice in case the hierarchical structure is more relevant or other glyph designs are used.

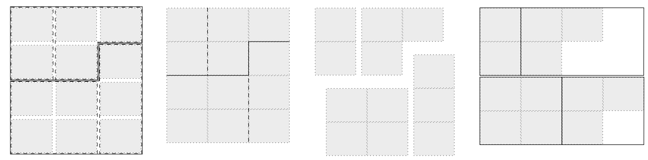

Figure 2: Layout sketches of nested grid treemap, split lines grid treemap, burst grid treemap, quantum treemap (left to right, cf. [1,18]).

Similarly, quantum treemaps and the burst layout of grid treemaps add whitespace. This can only be tolerated, if the number of glyphs is moderate relative to the available screen size. In case many glyphs need to be placed on limited space, the nested or split lines layout should be preferred. On the other hand, the added whitspace may facilitate the comprehension of the hierarchical structure. While quantum treemaps ensure that each level is represented as a rectangle, grid treemaps may add less whitespace, but do not enforce convex representations of levels. As you can see both, task and data demands, go hand in hand, and in the worst case need to be traded off against each other.

7 Future Work and Conclusion

Contrary to the design of glyphs, their placement has not attracted much research interest. Especially, when it comes to empirical studies, we have the impression that only few related works have been published. For example, Opach et al. [12] recently compared the performance of placing glyphs on a map and in a matrix. While glyph size is not glyph placement, it is one of the determining factors of overlap and occlusion, such that placement directly depends on it. Heer, Kong and Agrawala [8] investigated the minimal size required to display and interpret line charts. However, Goffin et al. found that "Unfortunately, similar readability or minimum size guidelines do not yet exist for most chart types, so the discrimination of whether the space is sufficient rests in the hands of the designer." [7]

There are various opportunities for more basic research on glyph placement. To begin with, the decision aid we propose in Fig. 1 and discussions in the literature referenced in Sect. 6 largely leave open how dependencies on task, user, and data come into play. It is generally acknowledged that visualizations need to support the analysis task in order to be effective. What "support" means, beyond showing the relevant information, yet is only partially understood.

As a result, general topics in visualization, such as glyph placement, often lack a sufficient amount of practically applicable guidelines. While we provide a decision aid on an abstract level in Fig. 1, which hopefully can serve as a starting point for practitioners, more concrete guidelines that are easy to understand and sufficiently general need to be developed. Perhaps, tabular approaches as, for example, started by Ward [19] are a good way to go (see also Table 1).

In conclusion, we presented some useful design considerations on the placement of glyphs. We distinguish between three scenarios: context-driven placement (Sect. 2), placing data collections (Sect. 3), and placement of data samples (Sect. 4). Further, we took more general considerations into account and related them with the dependencies imposed by practical tasks, users and data.

Acknowledgments

Part of this work was financed by the Graduate School of Decision Sciences at the University of Konstanz.

References

[1] B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *Transactions on Graphics*, 21(4):833–854, 2002.

[2] R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. *Eurographics State of the Art Reports*, pp. 39–63, May 2013. doi: 10.2312/conf/EG2013/stars/039-063

[3] D. Carr. Guidelines for designing information visualization applications. In *Proceedings of the 1999 Ericsson Conference on Usability Engineering*, 1999.

[4] M. Chen, G. Grinstein, C. R. Johnson, J. Kennedy, and M. Tory. Pathways for theoretical advances in visualization. *IEEE Computer Graphics and Applications*, 37(4):103–112, 2017. doi: 10.1109/MCG.2017.3271463

[5] D. H. S. Chung, P. A. Legg, M. L. Parry, R. Bown, I. W. Griffiths, R. S. Laramee, and M. Chen. Glyph sorting: Interactive visualization for multi-dimensional data. *Information Visualization*, 14(1):76–90, 2015. doi: 10.1177/1473871613511959

[6] J. Fuchs, P. Isenberg, A. Bezerianos, and D. A. Keim. A systematic review of experimental studies on data glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 23(7):1863–1879, 2016. doi: 10.1109/TVCG.2016.2549018

[7] P. Goffin, W. Willett, J. D. Fekete, and P. Isenberg. Exploring the placement and design of word-scale visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2291–2300, 2014. doi: 10.1109/TVCG.2014.2346435

[8] J. Heer, N. Kong, and M. Agrawala. 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 '09, pp. 1303–1312. ACM, New York, 2009. doi: 10.1145/1518701.1518897

[9] D. A. Keim and A. Herrmann. The gridfit algorithm: an efficient and effective approach to visualizing large amounts of spatial data. In *Proceedings Visualization '98*, pp. 181–188, October 1998.

[10] A. E. Lie, J. Kehrer, and H. Hauser. 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. Budmerice, Slovakia, April 2009. doi: 10.1145/1980462.1980470

[11] W. Meulemans, J. Dykes, A. Slingsby, C. Turkay, and J. Wood. Small multiples with gaps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):381–390, 2017. doi: 10.1109/TVCG.2016.2598542

[12] T. Opach, S. Popelka, J. Dolezalova, and J. K. Rød. Star and polyline glyphs in a grid plot and on a map display: Which perform better? *Cartography and Geographic Information Science*, 45(5):400–419, 2018. doi: 10.1080/15230406.2017.1364169

[13] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 514–519, August 1988. doi: 10.1109/ICSMC.1988.754351

[14] D. H. F. Pilar and C. Ware. Representing flow patterns by using streamlines with glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1331–1341, 2013. doi: 10.1109/TVCG.2013.10

[15] T. Ropinski, S. Oeltze, and B. Preim. Survey of glyph-based visualization techniques for spatial multivariate medical data. *Computers & Graphics*, 35(2):392–401, 2011. doi: 10.1016/j.cag.2011.01.011

[16] T. Ropinski and B. Preim. Taxonomy and usage guidelines for glyph-based medical visualization. In *Proceedings of the 19th Conference on Simulation and Visualization (SimVis08)*, pp. 121–138, 2008.

[17] T. Ropinski, M. Specht, J. Meyer-Spradow, K. H. Hinrichs, and B. Preim. Surface glyphs for visualizing multimodal volume data. In *Proceedings of the 12th International Fall Workshop on Vision, Modeling, and Visualization (VMV07)*, pp. 3–12, November 2007.

[18] T. Schreck, D. A. Keim, and F. Mansmann. Regular treemap layouts for visual analysis of hierarchical data. In *Proceedings of the Spring Conference on Computer Graphics (SCCG'2006)*. Casta Papiernicka, Slovak Republic, 2006.

[19] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002. doi: 10.1057/PALGRAVE.IVS.9500025

[20] M. O. Ward. *Multivariate Data Glyphs: Principles and Practice*, pp. 179–198. Springer, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-33037-0_8

[21] M. O. Ward and D. A. Keim. Screen layout methods for multidimensional visualization. In *Euro-American Workshop on Visualization of Information and Date*, 1997.

[22] C. Westin and G. Kindlmann. Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006. doi: 10.1109/TVCG.2006.134