Generalized Hyper-cylinders: a Mechanism for Modeling and Visualizing N-D Objects

Matthew O. Ward[1] and Zhenyu Guo[1]

1 **Computer Science Department, Worcester Polytechnic Institute**
100 Institute Rd., Worcester, MA 01609 USA
{matt,zyguo}@cs.wpi.edu

Abstract

The display of surfaces and solids has usually been restricted to the domain of scientific visualization; however, little work has been done on the visualization of surfaces and solids of dimensionality higher than three or four. Indeed, most high-dimensional visualization focuses on the display of data points. However, the ability to effectively model and visualize higher dimensional objects such as clusters and patterns would be quite useful in studying their shapes, relationships, and changes over time. In this paper we describe a method for the description, extraction, and visualization of N-dimensional surfaces and solids. The approach is to extend generalized cylinders, an object representation used in geometric modeling and computer vision, to arbitrary dimensionality, resulting in what we term Generalized Hyper-cylinders (GHCs). A basic GHC consists of two N-dimensional hyper-spheres connected by a hyper-cylinder whose shape at any point along the cylinder is determined by interpolating between the endpoint shapes. More complex GHCs involve alternate cross-section shapes and curved spines connecting the ends. Several algorithms for constructing or extracting GHCs from multivariate data sets are proposed. Once extracted, the GHCs can be visualized using a variety of projection techniques and methods to convey cross-section shapes.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases N-Dimensional Visualization, Cluster Visualization

Digital Object Identifier 10.4230/DFU.SciViz.2010.1

1 Introduction

Visualization has been identified as a critical component to the process of interactive exploration and mining of large data repositories. The essential problems that need to be addressed when developing tools for interactive visual data analysis include:

- How to structure and process the data into a format and size that is manageable within the visualization environment, yet retains most, if not all the significant information content of the original data;
- How to best display information on the screen so as to provide users with useful insights into their data given the constraints of visual perception, screen resolution, and processing speed; and
- How to provide users the ability to effectively interact with the visualization to extract meaning from the data.

The field of data visualization can be roughly divided into two distinct subfields. Scientific visualization concentrates predominantly on the display of one, two, or three-dimensional spatial/physical data, while information visualization generally assumes data sets of arbitrary

Scientific Visualization: Advanced Concepts.
Editor: Hans Hagen; pp. 1–10

Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

dimensionality, usually without a spatial attribute but often with one or more relations defined between data items.

The display of surfaces and solids has usually been restricted to the domain of scientific visualization; however, little work has been done on the visualization of surfaces and solids of dimensionality higher than three. The primary focus in most high-dimensional visualization has been on the display of data points, rather than surfaces and solids. However, the ability to effectively visualize higher dimensional objects, whether defined analytically or derived from data samples, would be quite useful in studying the shapes and relationships of these so-called hyper-objects. For example, the richness of the description of a cluster of N-dimensional data points could be greatly enhanced beyond commonly used methods, which often just consist of the cluster center along with the hyper-box or hyper-ellipsoid encapsulating the data. Likewise, descriptions of differences or changes in these clusters over time would benefit from richer representations.

The focus of this paper is to describe a novel method for the description, extraction, visualization, and interactive exploration of N-dimensional surfaces and solids. The general concept is to extend generalized cylinders [8, 9], an object representation regularly used in geometric modeling and computer vision, to arbitrary dimensionality, resulting in what we term Generalized Hyper-cylinders (GHCs). In its simplest form, a GHC consists of two N-dimensional hyper-spheres connected by a hyper-cylinder (spine) whose shape at any point along the cylinder is determined by interpolating between the shapes of the endpoints. A broader class of GHCs can be defined by using alternate cross-section shapes as well as curved spines. We describe several algorithms for extracting GHCs from large multivariate data sets, with user-controllable parameters to adjust the accuracy at which the GHCs approximate the real data. Once extracted, the GHCs can be visualized in 2-D or 3-D using a variety of techniques, such as projecting the endpoints into the display space using PCA or MDS. A variety of object types to represent the shape of the GHC are being explored and evaluated. Finally, a suite of tools is being implemented for interactively exploring data displayed with GHCs, including operations for navigation, selection, filtering, and distortion.

This paper is organized as follows. Section 2 describes the work of others in visualizing N-dimensional points and objects. Section 3 defines generalized cylinders and their extension to generalized hyper-cylinders. Section 4 describes methods for visualizing GHCs, while Section 5 focuses on methods to extract GHCs from datasets via manual, semi-automated, and fully automated techniques. We conclude in Section 6 with a summary and a list of potential future research directions.

2 Related Work

Multivariate data can be found in most, if not all, disciplines of study, and a wide range of techniques have been developed for the visualization of such data. Popular techniques include scatterplot matrices, projected point methods [1], parallel coordinates [2, 13], and tabular views [7]. Other techniques that have been proposed include glyphs [12], pixel-oriented techniques [3], and dimensional stacking [5]. While these are useful for examining individual data records, they are less effective at providing a high-level description of the entire dataset or selected subsets of the data.

For example, if one were interested in describing the shape of a cluster in a 5-dimensional dataset, what techniques would be applicable? Most cluster descriptions in common use consist of a small number of attributes, such as the location of the cluster center, its population, and perhaps its dominant axis. Others represent a cluster by a representative

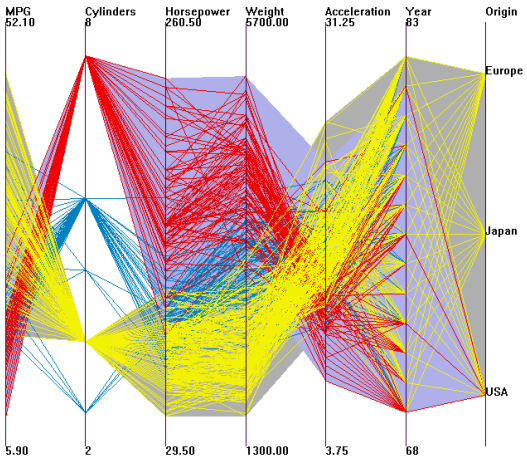

Figure 1 Two hyper-boxes in Parallel Coordinates. Two clusters have been isolated based on the second dimension. The brown and grey regions indicate the surrounding hyper-boxes for the clusters. Image generated with XmdvTool [6, 11].

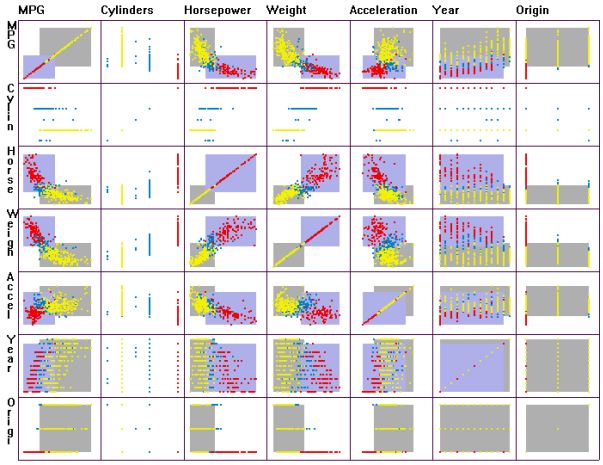

Figure 2 The same clusters in Scatterplot Matrices. Image generated with XmdvTool.

sampling of the data points contained in it. However, this is not a good representation for tasks such as comparing cluster shapes.

A simple approach to representing and visualizing N-dimensional clusters is to use the axis-aligned hyper-box that contains the points of the cluster. For example, in Figures 1 and 2, two clusters of points have been selected in the parallel coordinates and scatterplot matrix displays, respectively [6, 11]. The shaded regions indicate the extents in each dimension that contain the selected points (yellow points in the brown region, red points in the blue region). As can be seen, these regions overlap, so the user can only see the full extents of one of the clusters. Clearly an axis-aligned hyper-box is not a very accurate description of a cluster's shape.

Another approach is to decompose the N-dimensional space into a (potentially large) number of N-dimensional blocks or subspaces and represent the cluster as the set of subspaces that contain at least one data point. This is akin to the spatial enumeration technique for 3-D solid modeling [8], where a 3-D volume is represented by an array of volume elements

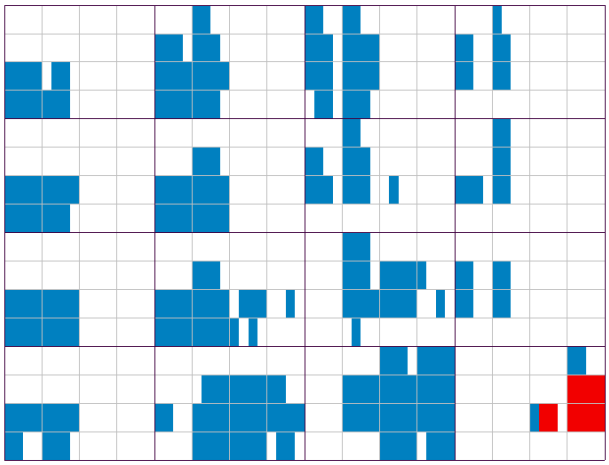

Figure 3 Spatial enumeration with dimensional stacking. Each bin represents a hyper-box in N dimensions. Those in red represent an isolated cluster [5, 4].

(voxels). One way to visualize this hyper-volume is with dimensional stacking [5, 4], where each dimension is divided into a small number of bins and the display space is recursively partitioned using pairs of dimensions. Figure 3 shows an example of a 5-dimensional dataset with one cluster (red) isolated. While separated in display space, the red bins are contiguous in N dimensions. The dark and light grid lines show the first two levels of nesting. Each dimension has four bins. This representation is not very compact, and in order to increase the accuracy of the representation the number of bins per dimension must be high. Also, for high dimensional data the number of occupied bins tends to be very small (the curse of dimensionality).

A method that is more accurate than hyper-boxes and more efficient than dimensional stacking is the H-BLOB method as described by Sprenger et al. [10]. They represent clusters via hierarchically nested hyper-spheres, which are then projected to three dimensions and visualized using implicit surfaces (see Figure 4). This generates a closed, smooth surface around each cluster, and thus provides a rich description of the shape. H-BLOBs have some similarities to GHCs; however, it would take a potentially large number of hyper-spheres to represent a shape that can be captured with a single hyper-cylinder, and we feel there are many shape features that can be derived from GHCs that would be difficult to extract with the H-BLOB representation.

3 Generalized Cylinders and Hyper-cylinders

Hyper-boxes and hyper-spheres are relatively coarse primitives to use in modeling shapes, especially those defined by groups of scattered points. In each case, there can be a significant amount of space within the model where there are no data points. In 3-D, a tapered cylinder can often come closer to encapsulating the points in a cloud, as the endpoints and radius can be adjusted to better fit the data. In geometric modeling and computer vision, this approach is known as *generalized cylinders* (GC) [9], which can be used to model axis-symmetric objects or object parts. A GC consists of two endpoints, a spine (straight or curved), and a cross-section (often a circle or ellipse). Many variants on GCs have been proposed over the years, including the use of non-convex cross-sections and varying the cross-section shape or size as one moves from one endpoint to the other. A wide range of complex object shapes

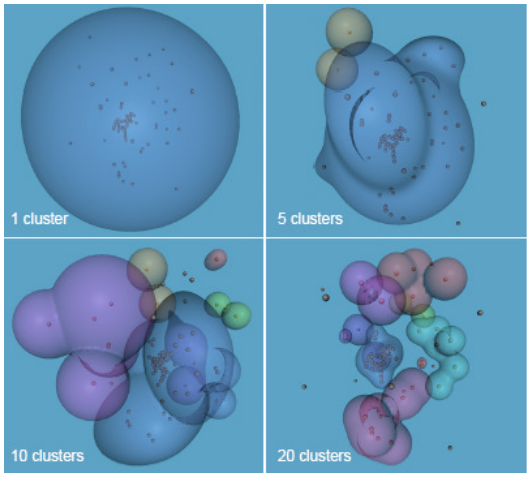

Figure 4 Implicit surfaces generated as hyper-spheres in N dimensions and projected to three dimensions. Image from [10] (used with permission).

can be represented using a small set of GCs. While GCs have been widely used in 2-D and 3-D, to the best of our knowledge, they have not been extended to higher dimensions.

In fact, it is not hard to imagine this extension, which we call a *generalized hyper-cylinder* (GHC). It is clear one can define two N-dimensional endpoints, along with a straight or curved spine connecting them. The shape of the cross-section, however, is not so straightforward. In the simplest form, we can use an (N-1)-dimensional hyper-sphere orthogonal to the spine, with either a constant or variable radius as one moves along the spine. This would result in hyper-planes at the ends of the GHC. Another alternative is to use an N-dimensional hyper-sphere at each end, similar to H-BLOBs, with an interpolated radius along the spine. As with GCs, we can also use ellipses for the cross-section shape. This can allow the GHC to fit a given dataset with increased accuracy. Finally, while it might be possible to use an arbitrary (N-1)-dimensional shape as a cross-section (e.g., represented as a GHC with one less dimension), we feel the resulting complexity would make interpretation, rendering, and analysis difficult.

4 Visualizing GHCs

There are many ways one could consider to render a set of GHCs; indeed. most multivariate data visualization techniques could be extended to convey the endpoints, spines, and cross-sections. As our initial attempt, we focused on GHCs with a straight spine and a hyper-sphere cross-section. For M GHCs we draw M 2-D scatterplots, each aligned with a particular GHC. Each GHC is represented as a trapezoid, where the width of the top and bottom are proportional to the radii at the two endpoints and the length of the trapezoid is proportional to the N-D distance between the endpoints. The endpoints are connected to represent the spine. All other GHCs in a given view are drawn relative to the focus GHC.

In the following, we describe how to generate a 2-D scatterplot to represent and visualize a hyper-cylinder. For each data point, we project it onto a scatterplot view by calculating the x and y coordinates relative to the two endpoints and the spine. Assume the two endpoints are A_1 and A_2, respectively, in N-dimensional space and A_m is the middle point of A_1A_2. For any data point B_i, it can be projected onto the spine A_1A_2. Assume the projection point is B_p, i.e., B_p is on A_1A_2 and B_iB_p is perpendicular to A_1A_2. The point B_p is calculated as

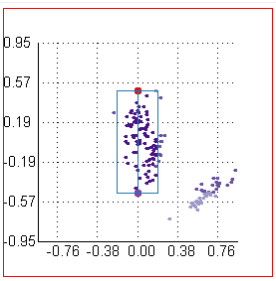

Figure 5 The perspective scatterplot view before rotating.

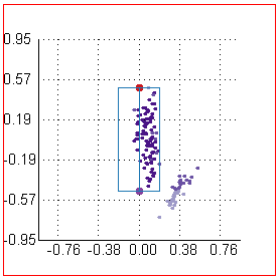

Figure 6 The perspective scatterplot view after rotating $1/3\pi$.

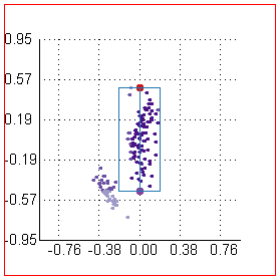

Figure 7 The perspective scatterplot view after rotating $2/3\pi$.

follows:

$$B_p = A_1 + \frac{(B_i - A_1) \cdot (A_2 - A_1)}{||A_2 - A_1||} \frac{A_2 - A_1}{||A_2 - A_1||}$$

The value of the y coordinate is the distance between A_m and B_p. If B_p is nearer A_2, the value is positive; if B_p is nearer A_1, the value is negative. The value of the x coordinate is computed as $||B_i - B_p||cos\theta$, where $||B_i - B_p||$ is the Euclidean distance from the data point and its projection point, and θ is the angle between vector $B_i - B_p$ and any fixed vector that is orthogonal to A_1A_2, say $T - T_p$ (T is any point in N-dimensional space and T_p is the projection point as described before):

$$\theta = arccos(\frac{(O - O_p) \cdot (T - T_p)}{||O - O_p||||T - T_p||})$$

Thus the x coordinate is the rotated distance from a data point to the spine which simulates a perspective view effect. When interactively increasing or decreasing all the angles by an offset, analysts are able to simulate viewing the hyper-cylinder from different orientations, i.e. by rotating the hyper-cylinder around the spine. Figures 5 to 7 show an example of the different perspective views from different orientations when viewing a hyper-cylinder in a three dimensional space. Point T is selected as the first data point of the dataset.

To visualize multiple GHCs in a single scatterplot view, we map the two endpoints of each non-focus GHC and connect them to represent the spine. We draw two perpendicular lines whose lengths are proportional to the two radii and connect the four corners (the end of the two perpendicular lines) to get a trapezoid. We fill the trapezoids with different colors to denote different GHCs.

Figure 8 shows a set of three GHCs for a 4-dimensional dataset. Each view is centered on one GHC (outlined), while the others are shown as filled colored trapezoids. The view with the red boundary indicates the GHC that is currently being edited (the focus GHC). The first GHC contains all of the data points; the amount of empty space indicates that this GHC does not fit the dataset accurately. The second and the third GHCs contain the two clusters, which are more accurate than the first GHC as they have smaller radii and shorter spine lengths. It is very easy to discover how well the data points fit the cluster and examine outliers that do not fit well in either GHC.

This representation is simple, yet flexible. Curved spines can easily be accommodated, as can cross-sections that change in a non-linear way. There are also many possible variations on this simple view, including:

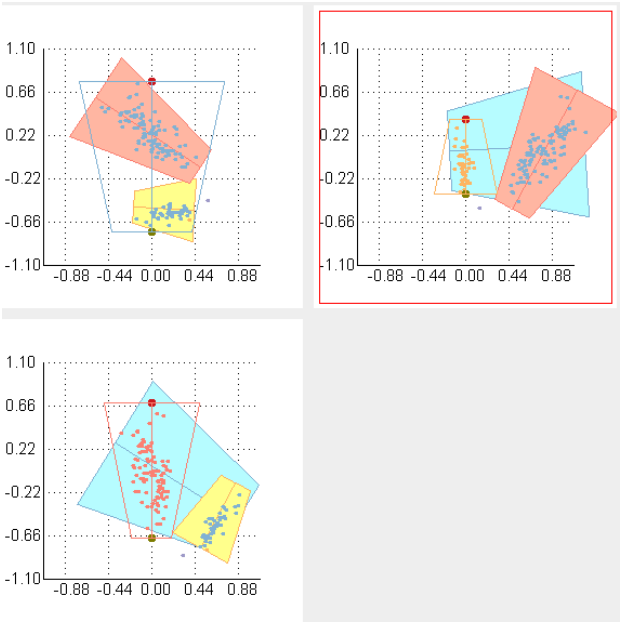

Figure 8 Visual representation of a set of GHCs .

- Placing endpoints, as well as intermediate axis points for curved spines, using dimensionality reduction techniques such as PCA and MDS.
- Rendering in 3-D, which can reduce the amount of occlusion.
- Using colored stripes to represent how each dimension is changing along the length of the spine.
- Extruding a 2-D star glyph along a 3-D spine, where the length of each branch of the star conveys the dimension size (for hyper-ellipsoidal cross-section).

We are experimenting with these and other variations for visualizing GHCs.

5 Extracting GHCs from Data

One of the biggest challenges with GHCs is deriving them from data, as a given dataset could be represented with varying degrees of accuracy, leading to varying numbers and shapes of GHCs. We can categorize potential approaches as either manual, automated, or semi-automated. In the manual case, the user defines the endpoints for each GHC as well as the cross-section size and shape. Data that fall within these specifications are assigned to the GHC being constructed. By coloring the points as they get covered by the GHC, the user can interactively adjust the position, shape, and orientation of the GHC to best fit the data. In this case, it is up to the user to decide when multiple GHCs are needed. Figure 9 shows such an interface for manual GHC specification. A subset of 5 dimensions are selected in the parallel coordinates view, and the endpoints of the GHC are specified in the scatterplot matrix view. In the projected view the user can adjust the radii of the GHC at each end.

In a semi-automatic approach, the user might specify pairs of endpoints and allow the system to compute the best cross-section size and shape to use. It would also be possible to automate the fitting of a curved spine, based on the distribution of points. Again, it would be up to the user to indicate how many GHCs should be used and approximately where they

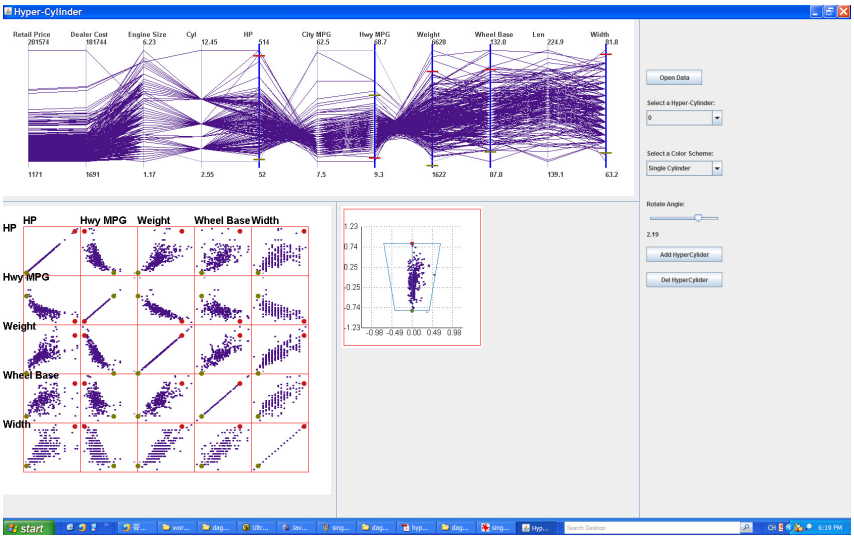

Figure 9 Interactive creation of GHCs to represent a dataset.

are located. Automated techniques could also be used to refine the endpoint location using a localized search.

A fully automatic method could start with a clustering of the data. It would then be assumed that each cluster could be represented by a single curved GHC or a set of connected linear GHCs. Starting with the extreme points of a cluster, the spine could be initialized to the straight line connecting the endpoints. Each point in the cluster would then be projected to this line to ascertain if any gaps exist that should result in the division of the cluster into multiple GHCs. Assuming no gaps exist, the distances and directions to each cluster point from the spine could be used to bend the spine towards the center of the data that project to that neighborhood of the spine. One challenge would be to identify where forks and joins must occur, e.g., when few points are close to the spine and there are two or more groups of points that share an approximate direction from the spine. One problem with this approach is that the initial choice of endpoints is critical; it is important to not choose outliers, and rather choose points that represent the dominant axis of the cluster.

We are studying ways to enhance our manual GHC creation tool as well as exploring algorithmic alternatives to some or all of the stages of extraction and refinement.

6 Summary and Conclusions

In this paper we have introduced a new method for approximately describing objects of dimensionality greater than three. By extending the notion of Generalized Cylinders from 3-D to N-D we can describe clusters and other patterns in multivariate datasets in a compact, yet descriptive form. GHCs can be useful for not only compressing a dataset, but also for comparing multiple datasets for change analysis and in specifying queries on data.

There are many unsolved problems and avenues for research in the definition, extraction, and use of GHCs. While space limitations prohibit us from going into detail here, a partial list of such topics includes:

- What forms of interaction should be available to create and explore GHCs? These might include navigation in data space, feature space, or display space, drill-down and roll-up to

get more or less detail on demand, and distortions such as bending, moving, and scaling to see objects more clearly without losing context.
- Are there other shapes that would capture the shape more accurately or that would be easier to extract with comparable shape accuracy?
- What error measures could be used?
- What object configurations would not be suitable for GHCs?

It is anticipated that a wide range of disciplines will benefit from this research, including bioinformatics, computational modeling, telecommunications, health care, and other scientific and industrial domains where the analysis of high dimensional data and models is important.

Acknowledgements

This work was primarily supported by the National Science Foundation under Award Number IIS-0812027.

References

1 Patrick Hoffman, Georges Grinstein, and David Pinkney. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In *NPIVM'99: Proceedings of the 1999 Workshop on New paradigms in Information Visualization and Manipulation (in conjunction with the 8th ACM International Conference on Information and Knowledge Management)*, pages 9–16, New York, NY, USA, 1999. ACM.

2 Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *VIS'90: Proceedings of the 1st Conference on Visualization 1990*, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

3 Daniel A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

4 John T. Langton, Astrid A. Prinz, and Timothy J. Hickey. Neurovis: combining dimensional stacking and pixelization to visually explore, analyze, and mine multidimensional multivariate data. In Robert F. Erbacher, Jonathan C. Roberts, Matti T. Gröhn, and Katy Börner, editors, *Proc. Visualization and Data Analysis*, page 64950H. SPIE, 2007.

5 Jeffrey LeBlanc, Matthew O. Ward, and Norman Wittels. Exploring n-dimensional databases. In *VIS'90: Proceedings of the 1st Conference on Visualization 1990*, pages 230–237, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

6 Allen R. Martin and Matthew O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *IEEE Visualization Conference*, pages 271–278, Los Alamitos, CA, USA, 1995. IEEE Computer Society.

7 Ramana Rao and Stuart K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI'94: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 318–322, New York, NY, USA, 1994. ACM.

8 Aristides G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys*, 12(4):437–464, 1980.

9 Steven Shafer and Takeo Kanade. The theory of straight homogeneous generalized cylinders and a taxonomy of generalized cylinders. In *Proceedings of the 1983 DARPA Image Understanding Workshop*, pages 210–218, 1983.

10 T. C. Sprenger, R. Brunella, and M. H. Gross. H-blob: a hierarchical visual clustering method using implicit surfaces. In *VIS'00: Proceedings of the Conference on Visualization 2000*, pages 61–68, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

11 Matthew O. Ward. Creating and manipulating n-dimensional brushes. In *Proceedings of Joint Statistical Meeting*, pages 6–14, 1997.

12 Matthew O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

13 Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85:664–675, 1990.