Technique and Cue Selection for Graphical Presentation of Generic Hyperdimensional Data

by

Lee Howard

A thesis proposal submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
Brigham Young University
December 2010
ABSTRACT

Technique and Cue Selection for Graphical Presentation of Generic Hyperdimensional Data

Lee Howard

Department of Computer Science

Master of Science

The process of visualizing n-D data presents the user with four problems: finding a hyperdimensional graphics package capable of rendering n-D data, finding a suitable presentation technique supported by the package that allows insight to be gained, using the provided user interface to interact with the presentation technique to explore the information in the data, and finding a way to share the information gained with others. Many graphics packages have been written to address the first problem. However, existing packages do not sufficiently solve the other three problems.

A hyperdimensional graphics package that sufficiently solves these problems is expected to simplify the user experience and allow the user to easily explore, interact with, and share the data. I propose a package that will sufficiently solve all four problems. The package will be able to render n-D data through appropriate encapsulation of presentation techniques and their associated visual cues. All known presentation techniques will be accommodated through the use of an extensible plugin system. Desirable features will be supported by the user interface to allow the user to easily interact with the data. Sharing of visualizations and annotations will be included to allow users to share information with each other. By providing a hyperdimensional graphics package that accommodates all presentation techniques and includes desirable features, including those that are rarely or never supported, the user will benefit from tools that will allow better interaction with multivariate data to extract information and share it with others.
1. Introduction

Data visualization is the process of visually rendering data so that the user can infer information about the data. In order to convey the information in the data effectively, visual cues are used to represent variable values. A visual cue is an attribute of the rendering, such as position, size, or color, which can be changed to communicate the value of the variable it represents (see Section 2.1). Presentation techniques dictate and allow the selection of visual cues and their layout in the rendering. Many presentation techniques are available for graphing data. However, traditional graphing packages typically limit the number of variables to three or less. Due to this restriction, it is difficult to see relationships in the data and extract information from the data when more than three variables are represented.

1.1. The Value of Hyperdimensional Presentation Techniques

Hyperdimensional presentation techniques have been developed to allow the simultaneous rendering of more than three variables. This allows the user to see more or all of the variables of each piece of data simultaneously. The alternative is to suppress one or more variables so that traditional 3-D graphing methods can be employed. Valuable information can be gleaned when none of the variables are suppressed, permitting observation of all of the variables and their relationships.

Figure 1 shows four plots of the same data set of some car information using different hyperdimensional techniques. All of the plots were rendered using the XmdvTool [ward1994xmdvtool]. Figure 1(a) is drawn using parallel coordinates [inselberg1985plane]. In parallel coordinates, each dimension is shown as a vertical axis, and data points are rendered as polylines that cross each axis at the appropriate value for that dimension. Trends can be seen by analyzing where the lines tend to cross adjacent axes. For example, with the ‘MPG’ (Miles Per Gallon) axis next to the ‘Cylinders’ axis, you can see that as the number of cylinders increases, the miles per gallon decreases. This relationship could be anticipated, but is used to show the power of this technique. Figure 1(b) is drawn using scatterplot matrices [tukey1981graphical]. In scatterplot matrices, variables are taken two at a time and graphed on an x,y plot. All such plots are placed together in a matrix format to allow visual determination of how variables relate to each other. Figure 1(c) is drawn using starplots [tukey1981graphical]. Starplots consist of lines emanating from a center point with adjacent line end points connected by lines for visual clarity. Each line represents a variable, and the length of each line communicates the value for that variable in a data point. Starplots are useful for finding similar data points in a data set since similar data points have similar overall shape. Figure 1(d) is drawn using dimensional stacking [leblanc1990exploring]. Dimensional stacking builds a hierarchy of the dimensions. Buckets are defined for each dimension, and data points are associated with the buckets into which their values fall. Points are plotted by splitting the rendering area into horizontal spaces equal to the number of buckets for the first dimension. Then, each of those spaces is divided vertically by the number of buckets in the second dimension, and so on, until all dimensions are accounted for. The point is plotted by following the hierarchy based on point values and marking the appropriate spot in the grid. Dimensional stacking helps to show clusters and patterns in the data set.

Each presentation technique provides potentially unique insight into the data. However, if only one or two of the techniques are utilized, some of the available information might not be seen. A graphics package that supports all available presentation techniques can be expected to provide the user with more information than a graphics package that doesn’t support all available presentation techniques.
1.2. The Visualization Process

The user is confronted with four problems when trying to visualize n-D data:

- finding a hyperdimensional graphics package capable of rendering n-D data
- finding a suitable presentation technique supported by the package that allows insight to be gained
- using the provided user interface to interact with the presentation technique to explore the information in the data
- finding a way to share the information gained with others.

Many graphics packages have been written to address the first problem. Some of the available packages are listed in Table 2.

The second problem, finding a suitable presentation technique, is case specific. One presentation technique may present the data in a more suitable way than another in one situation, while the reverse may be true for a different situation. The same is true about the user’s perception of which technique is suitable. Each user may have a different preconception about which technique is more suitable for a given data set. Since it cannot be known which techniques will prove suitable, a general-purpose package should accommodate as many techniques as possible. One solution to this problem is to allow extensibility through a plugin system. Presentation techniques can then be added through coding a simple plugin, which can be easily distributed to users of the package.

Interaction with the visualization is an important way of exploring the information inherent in the data. Since interaction is provided through the user interface (UI), the UI should be simple and easy to use. The user should be provided with standard features that will allow manipulation of the visualization. For example, panning, rotation, and zooming are important features that allow the user to modify the visualization to find interesting information. Other similar desirable features, such as annotations, should be included to allow the user to focus on the process of visualization rather than on the package.
Once the user has found information in the data using various presentation techniques, he may wish to share the information with others, either for collaborative purposes, for reports, or for publications. If the graphics package does not support sharing of visualizations with other, the user must find some other way to share the information. Consider the user who needs to include a visualization in a report. If the package does not provide a way of saving the visualization as an image, the user may need to take a series of screen shots of the visualization and crop the resulting image in an image manipulation program. It saves the user some effort if the functionality is supported by the package.

A hyperdimensional graphics package that sufficiently addresses each of these problems can be expected to simplify the user experience and allow the user to easily explore, interact with, and share the data.

1.3. Research Approach

While 3-D graphic environments and techniques are commonplace and well established, techniques for higher dimensional graphics are not widely used despite a significant, but often unrealized repertoire of hyperdimensional capabilities. Many techniques have been developed to render hyperdimensional data. However, current graphics packages utilize at most a few of these techniques. I propose to provide a framework that will allow the user to easily select presentation techniques and their associated visual cues. I propose to do so by

- identifying presentation techniques that represent the spectrum of available presentation techniques (Section 2.1)
- identifying desirable features of application-independent, wide technique- and cue-spectrum user interfaces for three-dimensional graphics (Section 2.2)
- identifying desirable features of application-independent, multiple technique- and cue-spectrum user interfaces for n-dimensional graphics (Section 2.3)
- showing that existing graphics packages which accommodate multiple n-dimensional presentation techniques and cues do not cover the spectrum of presentation techniques and do not possess all of the desirable features (Section 2.4)
- overcoming these deficiencies by
  - specifying a file format and a common data representation suitable for all known hyperdimensional presentation techniques (Section 4.1-Section 4.2)
  - specifying a rendering context that is independent of any specific graphics API (Section 4.3)
  - specifying a programming interface that will provide extensibility through the use of plugins (Section 4.4, Section 4.5.6)
  - defining an intuitive, application-independent, wide technique- and cue-spectrum user interface for n-dimensional graphics which accommodates the spectrum of presentation techniques and includes the desired features (Section 4.5)

In Section 2, several visual cues and presentation techniques are discussed. Desirable features from the user interfaces of some three-dimensional graphics packages are identified. Those features are combined with features from n-dimensional graphics packages to form a list of desirable features hyperdimensional graphics packages could include to support useful interaction and exploration. Current graphics packages are reviewed and evaluated based on the desirable features previously mentioned. Section 3 contains my thesis statement. Section 4 describes the design of the proposed graphics package. Section 5 establishes how achievement of my objectives will be validated. Section 6 contains the anticipated schedule for completion of this thesis.

2. Related Work

This section consists of four parts. First, I briefly review the techniques that I will include in this research, together with a justification for their inclusion. Various visual cues are mentioned, and their role
in the different presentation techniques is shown. Then I review some of the graphics packages available for 3-D graphics to identify desirable features. Next I identify a list of features that will be used to evaluate hyperdimensional graphics packages. Finally I review the state of the art in hyperdimensional graphics packages and evaluate them based on the aforementioned features.

2.1. Multi-spectrum Hyperdimensional Presentation Techniques and Visual Cues

To easily accommodate all presentation techniques in a graphics package, a common representation must be found. One feature all techniques have in common is that they use visual cues to display the data. A visual cue is an attribute of the rendering that can be changed to communicate the value of the variable it represents.

Visual cues that have been used in existing techniques include the following:

- Angle – The angle between two lines varies according to the value.
- Attributes – Varying the attributes of an object indicates the values. For example, with Chernoff faces, varying attributes like eye size, eye position, face shape, mouth shape, etc. conveys the values of the variables.
- Color – Different colors for discrete values, or a blending of colors for numerical values.
- Curvature – The curvature of a line varies according to the value.
- Distance – The distance from a point of reference communicates the value (or collection of values in the case of Star Plots, for example).
- Function – The values are combined into a function, and the function is plotted.
- Hierarchal Structure – Coordinate systems can be nested. For example, at (x,y,z) positions, smaller coordinate systems can be placed to convey one set of values, and (t,u,v) positions within those smaller coordinate systems can convey an additional layer of values.
- Length – Varying the length of a line segment or part of a geometric shape communicates the value.
- Level – The amount an object is filled in indicates the value.
- Orientation – The orientation varies with the value.
- Position – The location along each axis shows the value.
- Quantity – The quantity of attributes or objects indicates the value.
- Rotation – The faster the point or geometric shape rotates, the greater the value.
- Shading – Varying the shading can indicate the value.
- Shape – A different shape per value or blending different shapes for varying numeric values.
- Size – Varying the size of the point or geometric shape can indicate the value.
- Structure – The structure of the object changes according to the value or values (of multiple variables).
- Symbol – Varying the symbol or symbol attributes can indicate the value.
- Taper – The rate of taper indicates the value.
- Texture – A different texture per value or blending different textures together.
- Thickness – The thickness of a line or lines varies according to the value.
- Transparency – The transparency varies according to the value.

Many different methods of visualization have been proposed to view data with more than three dimensions. The presentation techniques found in Table 1 will be included. The presentation techniques have been chosen based on commonality (as shown by prevalence in the literature), diversity, and relevance to this research. The techniques are listed with the visual cues that pertain to each technique, as well as references that give detailed explanations of the technique. While this list does not contain every available presentation technique, it is a heavy sampling and covers the spectrum of hyperdimensional presentation techniques.
<table>
<thead>
<tr>
<th>Name of Technique</th>
<th>Angle</th>
<th>Attributes</th>
<th>Color</th>
<th>Curvature</th>
<th>Distance</th>
<th>Function</th>
<th>Hierarchal</th>
<th>Length</th>
<th>Level</th>
<th>Orientation</th>
<th>Position</th>
<th>Quality</th>
<th>Rotation</th>
<th>Shading</th>
<th>Shape</th>
<th>Symbol</th>
<th>Texture</th>
<th>Thickness</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrews’ Plot [andrews1972plots]</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boxes [hartigan1975printer]</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cartesian Hyperspace Graphics [burton1988algorithms]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Castle Symbols [kleiner1981representing]</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chernoff Faces [chernoff1973use]</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circle Segments [ankerst1996circle]</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circular Parallel Coordinates [hoffman2000table]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dimensional Stacking [leblanc1990exploring]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glyphs [anderson1960semigraphical]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grand Tour [asimov1985grand]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gridviz [hoffman2000table]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heat Map (Data Image) [weinstein1997information, minnotte1998data]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height Map [grinstein2001high]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hyperbox Display [alpern1991hyperbox]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iconographic Display (Stick Figures) [pickett1988iconographic]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modified Circle Segments [hoffman1999dimensional]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mosaic Plot [hartigan1981mosaics]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-Line Graph (on separate axes) [hoffman2000table]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Coordinates [inselberg1985plane]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Planes Coordinates [xiao1992parallel]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Volumes Coordinates [xun1997parallel]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Piles [melville1997piles]</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pixel-Oriented Techniques [keim1996pixel]</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polar Glyphs [grinstein2001high]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polar Lines [grinstein2001high]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polyviz (Expanded Radviz) [hoffman2000table]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Profile Plots [chambers1983graphical]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projection Pursuit [friedman1974projection]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radviz [hoffman1997visualization]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammon Plot [sammon1969nonlinear]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scatterplot Matrices (Generalized Draftman’s Display) [tukey1981graphical]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Coordinates [kandogan2001visualizing]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Coordinates (3-D) [cooprider2007extension]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Plot [tukey1981graphical]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey Plot [hoffman2000table]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic Scatterplots [chambers1983graphical]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Lens [rao1994table]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tree Symbols [kleiner1981representing]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worlds within Worlds (N-Vision) [feiner1990worlds]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 1- Hyperdimensional presentation techniques with their associated visual cues.*
2.2. Wide Spectrum, Three-dimensional Presentation Techniques and Visual Cues

Many lower-dimensional presentation techniques have been developed that allow presentation of data that has three or fewer variables. Some of these include the bar chart, histogram plot, line graph, pie chart, and scatterplot. These techniques are well established and widely implemented. Years of interaction using these techniques have lead to standard patterns of interaction. Exploring these patterns of interaction will identify useful features that can be included in the development of a user-friendly hyperdimensional graphics package. After reviewing several 3-D graphics packages, I chose to review two of those packages that represent the majority of 3-D graphics packages and their user interfaces. In the following subsections, these two 3-D graphics packages will be briefly reviewed to provide insight to their user interaction.

2.2.1. ADVIZOR® Analyst

ADVIZOR® Analyst [advizor2010desktop] is a commercial business intelligence tool that is focused on helping people see important trends in their data. It supports many presentation techniques, including the bar chart, line chart, histogram chart, pie chart, and scatterplot [advizor2010technical]. The user interface employed allows users to quickly look through their data and find interesting trends and correlations [advizor2010data]. Some of the features of the user interface include

- Easy selection of data in a plot through clicking and dragging a rectangle over the interesting data points.
- Ability to exclude data points or draw them differently from other data points.
- Ability to view a data point’s variable values by hovering over it.
- Ability to save interaction state, such as selection, coloring, view transforms, etc.

2.2.2. Centrifuge®

Centrifuge® [centrifuge2010data] is a commercial business intelligence tool. It focuses on allowing users to find linking information in their data. It supports many presentation techniques, including the bar chart, bubble chart, line chart, and pie chart. It is written as a Flex application that runs in Flash player, thus leveraging the interaction supported by Flash player. Some of the features of the user interface include

- Easy zoom in a plot through clicking and dragging a rectangle over the desired zoom area.
- Ability to exclude data points.
- Ability to view a data point’s variable values by hovering over it.
- Ability to save interaction state, such as selection, coloring, view transforms, etc.
- Drag-and-drop interface allows easy manipulation of the application.

2.2.3. Summary

These two packages represent most of the 3-D graphics packages available. The review of these packages has identified some useful interaction features. The useful features that will be included in this research are

- **Point Display** – Hovering over a data point displays its values.
- **Point Exclusion** – Data points in a plot can be excluded from the plot either by not drawing them or by drawing them differently than other data points.
- **Saving State** – Interaction state, such as selection, coloring, view transforms, etc. can be saved for later reuse.
- **Selection** – Selection of data in a plot should be easy. It is shown in ADVIZOR® Analyst that selection using a click-and-drag rectangle is sufficiently easy.
• **Zoom** – Zooming in on a portion of a plot should be easy. It is shown in Centrifuge® that using a click-and-drag rectangle can be sufficiently easy.

### 2.3. Hyperdimensional Graphics Package Features

One of the goals of this research is to establish a comprehensive list of features that are meaningful and appropriate for inclusion in a hyperdimensional graphics package. Many of the features included below are found in current graphics packages. Others are not found in any hyperdimensional graphics package. All of them are intended to help the user of the graphics package to easily navigate and interact with the data that is visualized. The list includes the features identified in Section 2.2. Current graphics packages will be evaluated based on the following features:

**Annotations** – As users try to understand more complex data, it is often useful to have others look at the same visualizations to verify discoveries or to comment on patterns. When users are not at the same computer, this can be difficult. Annotations are meant to allow users to communicate directly using the graphics package and are rendered directly on the visualization. Color may be specified for all annotations. Several types of annotations should be supported:

- **Text** – Textual annotations allow notes to be taken or labels to be made on the visualization. A position and/or width may be specified to allow commenting and labeling to show up next to the point of interest in the view.
- **Highlighter/Pen Tool** – A highlighter tool or a pen tool allows the user to circle or highlight points [denisovich2005software]. This can be useful in showing interesting data points to another person or in documenting a discovery for later evaluation.
- **Shapes** – Tools that draw preset shapes are also useful. Drawing with a highlighter or pen may not be considered professional in some instances, whereas a perfectly drawn box or circle may be more appropriate. Shapes that should be included are circles, boxes, lines, polylines, and polygons.

**Brushing/Selection** – Brushing is the process of selecting data points based on the values of their variables. Brushing is usually done in one of two ways: using a tool or through selection of a range of values. It should also allow a change in selection.

- **Brush Tool** – A brush tool is commonly provided that is placed or dragged over the desired data points, resulting in the selection of those points. Examples of brush tools are the click-and-drag rectangle tool and the click-and-drag lasso tool.
- **Range Brush** – A range of values can be specified by using sliders, where the ends of the slider signify the range values, with one slider per variable. Values that satisfy all of the specified ranges are selected.
- **Change Selection** – Since it is beneficial to be able to select points in a view and see them in other views (see ‘Linking’ below), the user should be able to select any points at any time in any view. It is also useful to be able to add more points to the group of selected points after the initial selection or to remove points from the selected points. Such functionality is accomplished through holding a key, like the control key, and dragging with the tool over the desired/undesired points.

**Coloring** – Color is an attribute that can help distinguish parts of a visualization from others. At least 10 different colors should be supported in each of the following:

- **Point Color** – The package should allow the data point’s color to be changed, regardless of whether or not color is used as a visual cue or to signify something else in the view. This allows one group of points to be distinguished from other groups of points.
• **Background Color** – Changing the background color can help change the contrast and overall feel of the visualization. The user may prefer different colors for the background than the default or different background colors for each view.

**Consistent Tools** – The primary tools that are available to one view should be available to any view. Tools such as zoom, pan, annotation, and rotation (where appropriate) should be allowed for any view at any time. Users may experience frustration if tools available to one view are not available to others.

**Extensibility** - Given that it may be difficult, infeasible, or undesirable for one team of programmers to implement all existing and future techniques, there should be a way to easily add functionality and presentation techniques to the system such that others in the community may add to the functionality of the package without having to add directly to the source code of the package. An example of this can be seen in GIMP, an open-source image manipulation program [gimp2010gimp]. GIMP supports plugins and scripting. Due to this, there is a massive amount of plugins and scripts available for download that greatly enhance the functionality of GIMP. Users can simply download the plugin or script and place it in the correct folder. GIMP then reloads the plugins/scripts on command, and then the new functionality is available. No change to the source code of GIMP or to the installed program is needed.

**Image Support** – Visualization packages allow users to interact with and analyze the data. When it comes time to show the results to others, images are often created to add to reports, websites, etc. Screenshots of the view are inconvenient and lead to unnecessary image cropping. The package should allow the user to save the visualization as an image in a common format or copy the image to the clipboard so that it may be used by another program.

**Labels** – Visualizations can become unintelligible if labels are not utilized appropriately.

• **Default Labels** – Labels should be clear and convey necessary information, such as the labeling of axes in a plot. They should be added by default wherever is appropriate.

• **Label Editing** – Labels should also be editable because the user may want to change the text or format of the label or remove the label entirely without changing the names of variables or other attributes. Labels for each technique should be editable.

**Linking** – Linking is the process by which selection of data points in one view highlights the same data points shown in other views. As was mentioned in Section 1.1, seeing the data using different presentation techniques can lead to additional insight not afforded by a single technique. The same is true when highlighting points. Highlighting points in all views shows the same data in ways specific to the technique used, allowing further insight.

**Multiple Data Sets** – Information is often assimilated into different sets of data. It can be helpful to view information from more than one data set simultaneously. A graphics package should allow the rendering of more than one dataset.

**Multiple Views** – To be able to get the most out of visualizations, it is often necessary to view multiple views side by side. A graphics package should allow at least several views to be seen simultaneously on the screen. This allows the user to simultaneously view the data using different presentation techniques and/or cues to quickly see the data in different ways. It also allows more than one data set to be visualized on the same screen.

**Panning** – There are times when the zoom level doesn’t allow all of the points to fit in the view. Panning is the process of moving the center of the visualization such that points that are outside of the bounds of the view can be seen. Common methods for panning include scrollbars or a hand tool that can ‘grab’ the visualization and move it.
Point Display – Most presentation techniques do not display all of the values of a data point in the view. To accommodate a user’s desire to see all of the values of a data point, the information should be displayed when either the mouse is hovering over the data point or when the data point is clicked. Using a tool-tip dialog box is a fairly common method for displaying the information.

Rotation – Rotation is the process of pivoting the view around a point or line. A rotation tool can help to define where the pivot is and how much to rotate around it. Rotation should always accompany the techniques where it is appropriate.

Saving State/Sharing – Users interacting with their data will want to save the state of the visualization so they can come back to it later. A graphics package should allow saving of state so that the visualizations can be returned to the same state when they were saved. The views open, the zoom, pan, and rotation, the coloring, the selection of points, and the annotations made should all be saved. This also allows visualizations to be shared with others. Much can be gained when more than one person looks at the same visualization. Saving operation sequences may be useful for other types of interaction, as well. For example, the user could save a sequence of operations that were used to obtain a particular view of the data. This sequence of operations could then be used later on a different data set.

Shadowing – Having all of the points drawn in the view can lead to a cluttered display. Shadowing is the ability to select some data points and mark them as ‘shadowed’. Shadowed points are rendered as grayed-out points, rendered using transparency so that they are less visible than non-shadowed points, or not rendered at all. Shadowing points is a good way to reduce clutter by only showing points of interest in the view.

Subsetting – Subsetting is similar to shadowing. The selected points that are of interest are kept, while the rest are discarded, reducing clutter in the view. The difference is that this set of selected points is added as a new data set in the program. New views can be created using this data set like other data sets. This new set of data points may also be saved to file, reducing file sizes due to having removed unwanted data points.

Technique Changing – Changing the presentation technique permits the exploitation of the particular strengths of multiple techniques. Any view should be able to change the presentation technique that it is using. This will aid in viewing the data in many different forms without the need to always create a new window for a different presentation technique.

Tool Tips – There are often tools that the user doesn’t understand or recognize. A tool tip is a small rectangle with a small amount of text that appears next to the mouse when it is hovering over something important. By showing tool tips, the programmers of the package can communicate to the user the intent of various items in the user interface. This may help the user become familiar with the user interface faster.

Variable Changing – A common interaction with presentation techniques is changing the variable assignments. This allows the user to explore the data and view the relationships between the data points. The variable assignment should take place separately per view so that the user may change the variable ordering in a way that is conducive to analysis for that user. As this is a frequent interaction, it should be prominent in the user interface.

Zoom – The default zoom is rarely what the user wants. To allow more detail in the view to be seen, a zoom tool must be provided. As mentioned in Section 2.2.3, a zoom tool using a click-and-drag rectangle is sufficiently easy to use. A secondary form of zoom may be provided, as well.
2.4. Hyperdimensional Graphics Packages

Even with good hyperdimensional visualization techniques available, it is typically difficult for a user to graph hyperdimensional data. The typical user lacks the awareness and the skill necessary to implement most, or even any presentation technique. While there are implementations for many presentation techniques, most were developed as prototypes, presented as proofs-of-concept, and never refined for general usage.

Graphics packages have been written which attempt to simplify and facilitate the graphing of hyperdimensional data. These packages provide access to presentation techniques and allow appropriate user interaction. However, most packages are limited in scope, usability, and functionality. Many packages are limited to a single technique. Some are complicated or non-intuitive to use. A good visualization system should take the burden from the user and allow the user to easily choose or alter the features used to display the data. It should provide access to the spectrum of available presentation techniques in a user-friendly manner.

Several visualization systems are found in literature and on the market that are not discussed in depth for the following reasons. First, there are systems that are not general purpose or that support three or less of the hyperdimensional presentation techniques shown in Table 1. These systems will not be discussed in depth. Included among this group of systems are Antaeus [janina2010antaeus], DataDesk® [data2010data], DataMontage™ [stottler2010datamontage], Glyphmaker [ribarsky1993using], Grapheur [reactive2010grapheur], Gisharp [avs2010gisharp], Leadscape® [leadscape2010leadscape], Miner3D® [miner3d2010miner3d], Mondrian [theus2010mondrian], Omniscope [visokio2010omniscope], Parallax [kdnugetts2010parallax], Partek [partek2010partek], ScienceGL [sciencegl2010advanced], Style Intelligence [inetsoft2010intelligence], Style Scope [inetsoft2010scope], Visalix [xerox2010visalix], Visipoint [vispoint2010visipoint], VisIt [lawrence2010visit], Visual i/o [designing2010designing], and VisuMap [visumap2010visumap]. This is not to suggest that these packages are not useful or that they lack desirable functionality, such as statistical capabilities. This research focuses on hyperdimensional presentation techniques, and these packages don’t meet the desired qualifications. Other visualization systems are no longer available for purchase or download or are no longer supported or actively being developed. Included among this group of systems are MANET [hofmann2000manet], ORCA [sutherland2000orca], VisDB [keim2010visdb], and XGobi [swayne1998xgobi, swayne1991xgobi]. Finally, there are some systems for which sufficient information could not be obtained, despite significant effort. These systems include Analyst’s Workstation [i22010analysts], Oculus [oculus2010oculus], OpenViz [avs2010openviz], Spotfire® [ticbo2010spotfire], Tableau® [tableau2010fast], and Think EMD [think2010think]. In the absence of information to the contrary, it is assumed that these systems are similar to the systems that will be reviewed.

The following subsections provide overviews of the current hyperdimensional graphics packages that accommodate four or more hyperdimensional presentation techniques. The subsections provide a small summary of each package and briefly covers whether the package supports the desired features. A summary of supported techniques for each package is shown in Table 2, including those packages that will not be reviewed. A summary of the supported features for each of the reviewed packages is shown in Table 3.

2.4.1. GGobi

GGobi is a project written by Swayne, Cook, Lang, and Buja and is the successor to the XGobi system [swayne2003ggobi]. XGobi supported parallel coordinates and scatterplot matrices, but it had many problems. XGobi stored information across many different files, which quickly became cluttered. It was written using the X-Window system, which limited its use to computers that had the X-Window system running. Another issue was that to view data using two different windows, two different processes were set up, requiring inter-process communication. GGobi was written to overcome the shortcomings of the XGobi system. GGobi was written using the GTK+ toolkit to update the look and feel of the user
interface and to help with cross-system compatibility. The file format was changed to use XML. This was done to allow the information that was originally in many files to be stored in one file. Another change is that the windows all run on the same process. This allows many plots to be generated without the overhead of setting up different processes. Another improvement is that GGobi was designed to be extendable. Functionality can be added by creating plugins, including new types of plots. Figure 2 shows a screenshot of GGobi plotting the same data set using three different techniques.

The evaluation of the desired features for GGobi 2.1.8 can be summarized as follows:

**Annotations** – Not supported as defined.

**Brushing/Selection**
- **Brush Tool** – A brush tool is provided which allows selection of points. The brush starts as a fixed size, but can be resized by right-clicking on the bottom-left corner. Adding to the selection of points is supported by making the brushing persistent. Once the points are persistently brushed, it can be difficult to remove the brushing.
- **Range Brush** – Brushing a range of values is supported using sliders, where the ranges are placed next to each other, and sliders are moved to change the range boundaries.
- **Change Selection** – Points can be added to the selection of points, but they cannot be removed from the selection.

**Coloring** – Color is an attribute that can help distinguish parts of a visualization from others. At least 10 different color should be supported in each of the following:
- **Point Color** – Points that are brushed may be assigned a color.
- **Background Color** – The background color of the view may be changed, although it seems to only be available when the brushing interaction mode is selected, and it affects all views.

**Consistent Tools** – The zoom and pan tools are not available in some of the views.

**Extensibility** – Plugins may be written to extend functionality.

**Image Support** – Not supported as described.

**Labels**
- **Default Labels** – Default labels are generally supported where appropriate.
- **Label Editing** – Not supported as described.

**Linking** – Selection and highlighting in one view highlights in the other views.

**Multiple Data Sets** – Multiple data sets can be loaded and rendered.

**Multiple Views** – Multiple views may be opened for any loaded data set.

**Panning** – Panning is supported for those views that support zooming.

**Point Display** – Information for the point is shown next to the mouse, but only one field shows up at a time. The user can choose which field is shown in the tool tip.

**Rotation** – Rotation is supported for those views where it is appropriate.
**Saving State/Sharing** – GGobi saves the points and their properties, but it doesn’t save some of the other desired interactions. For example, it doesn’t save which views are open, or if it does, those views are not opened when the file is loaded.

**Shadowing** – Points that are brushed can be marked as ‘shadow’ points and excluded from the view.

**Subsetting** – A form of subsetting is supported, but not exactly as described above. The subset is not from the brushed points. Instead, points must be clicked individually to give them a ‘sticky label’, which can then be subsetted. The subsetted points are not added to a new data set.

**Technique Changing** – Not supported as described.

**Tool Tips** – There are tool tips on various tools and buttons.

**Variable Changing** – Variables can be changed by clicking on the appropriate button next to the variables in the main window.

**Zoom** – Zoom is provided for some of the views.

---

**Figure 2. A screenshot of GGobi with 3 different plot types of the same data set.**

---

**2.4.2. NovoSpark® Visualizer**

NovoSpark® Visualizer is a visualization package written by NovoSpark® Corporation [novospark2010novospark]. The package mainly focuses on using curves to represent the data. It is one of the few packages that supports Andrews’ Plot. The main visualization area is used to render Andrews’ plot, parallel coordinates, and standard curves. Auxiliary displays, running in separate windows, are used to graph multi-line plots, scatterplot matrices, and polar coordinates. These displays can also be dragged into the main window and attached in predefined locations. Figure 3 shows a screenshot of NovoSpark® Visualizer plotting the same data set using four different techniques.

The evaluation of the desired features for NovoSpark® Visualizer 2.1 can be summarized as follows:
Annotations – Not supported as defined.

Brushing/Selection
- **Brush Tool** – Not supported as defined.
- **Range Brush** – The color of the curves can be changed based on the value. However, this is only one dimension, not the full n-dimensional brush.
- **Change Selection** – Not supported as defined.

Coloring
- **Point Color** – Not supported as defined.
- **Background Color** – The background color of the main view may be changed, but the other views’ background colors can’t be changed.

Consistent Tools – Zoom and panning are available in the main window, but not in the other views. Rotation is available in the main window, and it is the only place it is needed.

Extensibility – Not supported as described.

Image Support – The rendered image for the main view can be saved to file. The rendered image can also be copied to the clipboard for all views.

Labels
- **Default Labels** – Default labels are generally supported where appropriate.
- **Label Editing** – Not supported as described.

Linking – Selection in one of the auxiliary views highlights the data point in the main view but not in other auxiliary views. Selection in the main window also does not highlight points in other views.

Multiple Data Sets – Multiple data sets can be loaded and rendered.

Multiple Views – Multiple views may be opened for any loaded data set. However, only one view may be opened for a presentation technique. For example, only one view of a scatterplot matrices plot may be open at a time. Another limitation is that the Andrews’ plot and the parallel coordinates plot are both plotted in the main window, which means that the user can only view one or the other.

Panning – Panning is supported for those views that support zooming.

Point Display – A tool tip appears that displays the data point number, but no other information for the point is shown in the tool tip.

Rotation – Rotation is included where appropriate.

Saving State/Sharing – The data set and general state seem to save correctly. However, selection, some coloring, and other interactions are not saved.

Shadowing – Not supported as described.

Subsetting – Not supported as described.
Technique Changing – The main window allows the user to switch between Andrews’ plot and parallel coordinates, but the auxiliary windows cannot change the presentation technique used.

Tool Tips – There are tool tips on various tools and buttons.

Variable Changing – There doesn’t seem to be a way to change variables in all of the views.

Zoom – Zoom is provided for some of the views.

Figure 3. NovoSpark® Visualizer plotting a data set using Andrews’ plot, multi-line graphs, polar coordinates, and scatterplot matrices.

2.4.3. Tulip

Tulip is a visualization package developed primarily at LaBRI, University of Bordeaux I, France, and maintained by David Auber, the original author, and Patrick Mary [auber2010Tulip]. Tulip can plot hyperdimensional data using circular parallel coordinates, parallel coordinates, pixel-oriented techniques, and scatterplot matrices. All plots are drawn in separate windows that are contained in the main window. Figure 4 show a data set of a poker game using some of the presentation techniques.

The evaluation of the desired features for Tulip 3.4.1 can be summarized as follows:

Annotations – Not supported as defined.

Brushing/Selection

• Brush Tool – A selection tool is provided that allows a click-and-drag rectangle to select points.
• Range Brush – Not supported as defined.
• Change Selection – Points may be added or removed from the selected points by selecting them using the brush tool while holding the ‘control’ (ctrl) button to add and the ‘shift’ button to remove.

Coloring

• Point Color – Not supported as defined.
• Background Color – The background color of each view may be changed. The option is found in one of the tabs in the view editor pane.
Consistent Tools – Each view can be panned and rotated. However, there are some instances where zoom is not allowed. For example, in the scatterplot matrices plot, zoom is not allowed when viewing all of the plots. The user must double click on one of the plots in the matrix, which will change the view to only show that scatterplot. Zoom is then allowed for that plot.

Extensibility – A full plugin system is supported to add functionality to the package.

Image Support – The rendered image for each view may be saved to file.

Labels
- Default Labels – Default labels are generally supported where appropriate.
- Label Editing – Not supported as described.

Linking – Selection of points in one view highlights them in all other views.

Multiple Data Sets – Multiple data sets can be loaded and rendered.

Multiple Views – Multiple views may be opened for any loaded data set.

Panning – Panning is supported for all views.

Point Display – A tool tip appears that displays the data point number, but no other information for the point is shown in the tool tip or another appropriate location.

Rotation – Rotation is supported for all views. 2-D renderings are rotated about the z-axis.

Saving State/Sharing – The data set and general state seem to save correctly. However, selection and other interactions are not saved.

Shadowing – Not supported as described.

Subsetting – Not supported as described.

Technique Changing – Not supported as described.

Tool Tips – There are tool tips on various tools and buttons.

Variable Changing – Variable changing is supported, but it is located in a tab that may not always visible. Variable reordering may require many clicks of the mouse because the variables are moved up or down one position per click.

Zoom – Zoom is provided for some of the views.
Figure 4. Tulip plotting a data set using circular parallel coordinates, parallel coordinates, pixel-oriented techniques, and scatterplot matrices.

2.4.4. XmdvTool

XmdvTool is a program originally written by Ward [ward1994xmdvtool]. It is intended to make hyperdimensional graphing accessible and easy to use. XmdvTool allows the user to view hyperdimensional data using scatterplots, glyphs, parallel coordinates, and hierarchical techniques, as well as other techniques that have been added [cui2006enhancing, patro2004pixel, yang2003interactive]. Figure 5 shows one of the default data sets plotted using a few of the techniques.

The evaluation of the desired features for XmdvTool 8.0 can be summarized as follows:

Annotations – Not supported as defined.

Brushing/Selection

- Brush Tool – Some brushing in the scatterplot matrices plot is allowed by permanently leaving the brushing region on the display. However, this is not a specific brush tool, and this functionality is not supported in all views.
- Range Brush – Range brushing is supported on the parallel coordinates display. The range is displayed for each axis, with the brushed region drawn in the background. Other range brushing is not available from the user interface.
- Change Selection – Points can be added to the selection of points if they fall in the range of the brushing region in the parallel coordinates display or the scatterplot matrices plot, but they cannot be removed from the selection if they fall in the brushing region.

Coloring

- Point Color – Points that are brushed may be assigned a color, and points that are not brushed can be assigned a color, but those colors do not stay assigned to the points if the brushing region is changed.
- Background Color – The background color of the view may be changed, but it affects all views.

Consistent Tools – The available tools are consistent. XmdvTool doesn’t support any techniques that require rotation, so that tool is not included.

Extensibility – Not supported as described.
**Image Support** – The rendered image for each view can be saved to file.

**Labels**
- **Default Labels** – Default labels are generally supported where appropriate.
- **Label Editing** – Not supported as described.

**Linking** – Selection and highlighting in one view highlights in the other views.

**Multiple Data Sets** – Multiple data sets can be loaded and rendered.

**Multiple Views** – Multiple views may be opened for any loaded data set.

**Panning** – Panning is supported for those views that support zooming.

**Point Display** – Information for the point is shown in a display at the bottom of the main window, but only the fields used in the presentation technique show up.

**Rotation** – No techniques that require rotation are supported, so it is not included.

**Saving State/Sharing** – There doesn’t seem to be an option to save in the user interface.

**Shadowing** – Not supported as described.

**Subsetting** – Not supported as described.

**Technique Changing** – The user can change the presentation technique for any view by selecting the view to change and clicking on the technique buttons located on the right side in Figure 5.

**Tool Tips** – There are tool tips on various tools and buttons.

**Variable Changing** – Variables can be changed in a dialog. Changing the reordering of the variables requires many clicks, since a click moves a variable down one space. The reordering also occurs for all associated views, not just the current one.

**Zoom** – Zoom is provided for all of the views.
Figure 5. XmdvTool plotting a data set of car information using dimensional stacking, starplots, scatterplot matrices, and parallel coordinates.

| Package Name                  | Andrews' Plot | Boxes | Cartesian Hyperplane Graphics | Castle Symbols | Circular Segments | Circular Parallel Coordinates | Dimensional Stacking | Glyphs | Grand Tour | Gridviz | Heat Map | Heatmap Display | Hexagon Display | Iconographic Display | Modified Circular Segments | Mosaic Plot | Multi-Line Graph (on separate axes) | Nature Scenes | Parallel Coordinates | Parallel Planes Coordinates | Parallel Volumes Coordinates | Piles | Point Clouds | Point Fingers | Projections | Projection Pursuit | Radar Plot | Scatter Matrices | Star Coordinates | Star Coordinates (3D) | Surface Plot | Surface Plot | Tree Lens | Worlds Within Worlds | Worlds Within Worlds | Yeast Symbols | Yeast Symbols | Yeast Symbols |
|-------------------------------|---------------|-------|-------------------------------|----------------|-------------------|-----------------------|-----------------------|--------|-------------|----------|----------|------------------|------------------|--------------------------|--------------------------|-------------------|-------------------|------------------|-------------------|-----------------|---------------|-----------------|----------------|-------------------|-----------------|-----------------|-----------------|-----------------|------------------|-----------------|
| ADVIZOR® Analyst              |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| [advizor2010desktop]          |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Antaeus [janina2010antaeus]    |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Centrifuge® [centrifuge2010data] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| DataDesk® [data2010data]      |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| DataMontage™ [stottler2010datamontage] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| GGobi [swayne2003ggobi]        | X             |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Glyphmaker [riharsky1993using] |               | X     |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Grapheur [reactive2010gropheur] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Gsharp [avls2010gsharp]       |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Leadscope® [leadscope2010leadscope] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Miner3D® [miner3d2010miner3d] | X             | X     |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Mondrian [theus2010mondrian]  | X             | X     |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| NovoSpark® Visualizer         | X             |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| [novospark2010novospark]      |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Omniscope [visokio2010omniscope] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Parallax [kdnuggets2010parallax] |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
| Partek [partek2010partek]     |               |       |                               |                |                   |                       |                       |        |             |          |          |                  |                  |                          |                          |                  |                  |                  |                  |                 |               |                 |                 |
### Table 2 – Current hyperdimensional graphics packages and their supported presentation techniques. The values are marked as accurately as possible, but due to the nature of user interfaces (things can be easily hidden), as well as the nature of commercial software (the software is not freely available), some values may not be marked correctly.

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Annotations</th>
<th>Brushing</th>
<th>Coloring</th>
<th>Consistent Tools</th>
<th>Interactive Support</th>
<th>Labels</th>
<th>Multiple Data Sets</th>
<th>Multiple Views</th>
<th>Point Display</th>
<th>Rotation</th>
<th>Saving State/Sharing</th>
<th>Shadows / Transparent</th>
<th>Technique Changing</th>
<th>Tool Tips</th>
<th>Variable Change</th>
<th>Zoom</th>
</tr>
</thead>
<tbody>
<tr>
<td>GGobi (Version 2.1.8) [swayne2003ggobi]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NovoSpark® Visualizer [novospark2010novospark]</td>
<td>P</td>
<td>P</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>P</td>
<td>X</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X</td>
<td>P</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tulip [auber2010Tulip]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>P</td>
<td>P</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XmdvTool [ward1994xmdvtool]</td>
<td>P</td>
<td>P</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 3 – Hyperdimensional graphics packages with their supported features. Only packages with more than four of the specified presentation techniques are shown. Packages where sufficient information is not available are not shown. Features implemented as described in Section 2.3 are marked with an ‘X’, while features that only partially fulfill the description are marked with a ‘P’. The values are marked as accurately as possible, but due to the nature of user interfaces (things can be easily hidden), some values may not be marked correctly.

### 2.4.5. Summary

Given the set of all hyperdimensional presentation techniques, no package has all or even most of the techniques implemented. Some packages have been reviewed and evaluated against a set of desirable features. The review of these graphics packages show that there are some features that are not in any of the reviewed packages or are only supported in some of them. Annotation and label editing features are noticeably missing from all of the reviewed packages. Few packages support changing the selection as described earlier. There are also few packages that provide completely consistent tools, shadowing and
subsetting. The other features seem to be better supported in one form or another. While only a few
graphics packages have been reviewed, it can be assumed that most other packages on the market would
have similar feature characteristics.

2.5. What This Research Will Show
A primary purpose of this research is to show that a comprehensive system can be built that will
give users the tools they need to collaboratively visualize and share multivariate data. The proposed
system will be superior to existing systems as follows:

• It will have a specific mechanism to accommodate all known presentation techniques. Each
  presentation technique provides potentially unique insight into the data and may provide the user
  with more information about the data.
• Most of the tools for interaction will be readily visible in the UI, not hidden in menus. This will
  allow the user to quickly find and use tools.
• Tools that are available in one view will be available in others where appropriate. This
  consistency will allow users to focus on the visualization rather than on the package itself.
• Desirable features not widely supported in hyperdimensional graphics packages will be included
  to increase functionality and usability.
• Specifically, the system will support annotations of visualizations, which can be a significant tool
  to share information about visualizations with others.
• There will be support for sharing visualizations and annotations such that saving the state of the
  visualizations and annotations to file and reopening the file results in the visualizations and
  annotations appearing the same as before they were saved. This will allow a user to share a file
  knowing that another user will be shown the same visualizations and annotations when the file is
  opened.

3. Thesis Statement
The process of visualizing n-D data presents the user with four problems: finding a
hyperdimensional graphics package capable of rendering n-D data, finding a suitable presentation
 technique supported by the package that allows insight to be gained, using the provided user interface to
interact with the presentation technique to explore the information in the data, and finding a way to share
the information gained with others. Many graphics packages have been written to address the first
problem. However, existing packages do not sufficiently solve the other three problems.

A hyperdimensional graphics package that sufficiently solves these problems is expected to
simplify the user experience and allow the user to easily explore, interact with, and share the data. I
propose a package that will sufficiently solve all four problems. The package will be able to render n-D
data through appropriate encapsulation of presentation techniques and their associated visual cues. All
known presentation techniques will be accommodated through the use of an extensible plugin system.
Desirable features will be supported by the user interface to allow the user to easily interact with the data.
Sharing of visualizations and annotations will be included to allow users to share information with each
other. By providing a hyperdimensional graphics package that accommodates all presentation techniques
and includes desirable features, including those that are rarely or never supported, the user will benefit
from tools that will allow better interaction with multivariate data to extract information and share it with
others.

4. Project Description
This paper focuses on a general purpose approach to hyperdimensional rendering. Specifically,
the spectrum of higher-dimensional presentation techniques and their associated visual cues will be usable
to render the n-D data sets.
My research will include developing a framework design to accommodate techniques and their associated cues, including anticipated future techniques, without dependence on the details of any specific technique. A plugin architecture with an accompanying programming interface will be provided to accomplish this. The design will include a user interface that allows the user to enter a data set from a file, choose a presentation technique, choose visual cues, and view the results. The desirable features listed in Section 2.3 will also be described.

4.1. File Format

It is important to easily accommodate data from many sources. However, the framework will require that the file format be able to contain information regarding annotations, view transformations, selections, coloring, etc. To achieve both of these ends, two file formats will be supported: the CSV (comma-separated-value) format and a custom XML (Extensible Markup Language) format.

4.1.1. CSV Format

The CSV format is widely known, easy to understand, and is fairly efficient to process. Data will be stored using the rows and columns. Each column will represent a variable, and each row (line in the file) will represent a data point. The names of the variables will be included as the first row in the file, and data point values will be included in the subsequent lines. Variables may be either numeric or categorical. A numeric variable is one whose value is treated as a number in the normal fashion. A nominal variable, also known as a categorical variable, is one whose values are of a fixed set. For example, a variable that represents manufacturers of automobiles might have values like Ford, GMC, Honda, and Toyota. These values cannot be interpreted as conventional numbers and need to be handled differently. Variables will be determined to be numeric or nominal based on whether values contain just numbers or numbers and letters. The system will assign numeric values for each nominal value so that all presentation techniques can plot the data, but each presentation technique can choose which value to use. In this example, values could be assigned as follows: Ford = 1, GMC = 2, Honda = 3, and Toyota = 4. The user interface will allow these assigned values to be changed, if desired. Values for time will be represented as another column of data in the file. By using this simple format, and by allowing nominal variables to be plotted, the graphics framework will be able to accommodate data from a wide spectrum of sources.

4.1.2. XML Format

An XML format allows great flexibility at the cost of readability. Since it naturally lends itself to encoding objects of any description, it is able to easily store annotations, view transformations, selection, etc. Each of these required elements can be defined using the nodes in the XML. A root node will encapsulate the files contents. Other nodes can be added as children to the root node to represent annotations, view transformations, selection, etc. An exact specification for each node will not be given, since it is just an implementation detail. The data point information will be stored within a tag marked as “data”. Inside of this tag, the data will be stored exactly the same as the CSV format. This allows the flexibility of XML to be utilized for annotations and other objects while still allowing reasonably fast parsing of the data.

4.1.3. Other File Formats

These file formats are two of many that could have been chosen. The CSV file format was chosen for simplicity, ease of use, readability, and universality. The XML format was chosen to allow flexibility in storing annotations and other objects. Other file formats may be equally viable. An example of another way to store the data is in a database. This can allow flexibility in selecting which data is to be displayed. Each way of storing the data has its benefits. The file formats chosen are sufficient for this research, and other file formats can easily be added later by writing a file loader for a given format that conforms to the programming interface.
4.2. Model

Once the data is read in from file, the model class will use a three-dimensional array to store the data. The columns will represent each variable, the rows will hold elements or instances of data, and the third dimension will be used for successive points in time, if applicable. The objects stored in this array will be structs holding a float and a pointer to a string. This will allow models to store both numeric and categorical data. Multiple data sets will be handled by creating a new model for each data set.

The model will have a set of methods that will allow presentation techniques to interact with it. Getter methods will allow the techniques to request information about data points. Setter methods will allow the techniques to change various attributes of the data points, such as selection, color, and visibility. Linking will be accomplished by informing the model that an attribute has changed. The model will then inform all of the techniques listening to it that the attribute has changed. All techniques will then update their rendering.

4.3. Rendering Context

When a window is created for a presentation technique, a rendering context will be created and provided to the presentation technique implementation. This rendering context will be a class that implements a simple drawing interface. The technique implementation will call methods on this class to do its graphical rendering. The rendering context class will be a wrapper for the underlying rendering code. This will allow the graphics package to run using different renderers. For example, OpenGL or DirectX may be used for normal rendering, or a raytracer may be used for high-quality images.

4.3.1. Selection

To aid in the selection of data points, each draw method will require a point id along with the other parameters. When the user clicks on the visualization, the program will tell the technique implementation to redraw itself. The rendering context will receive the draw calls, but instead of redrawing the picture, it will find the data point that should be selected. Selection will be done this way to simplify the technique implementation and to allow the graphics package to handle all annotations (Section 4.5.2) and labels (Section 4.5.8) in a consistent way.

4.3.2. Consistent Tools

The rendering context will be responsible for providing panning, rotation, and zoom functionality. This functionality will be consistent in all views, meaning that panning, rotation, zoom, and annotation functionality will be available in all views, regardless of the presentation technique used. This will primarily be accomplished using a virtual camera and standard graphics operations. Matrices can be used to perform affine transformations on all rendering calls. For example, a matrix may be set up to translate everything 2 units down. Similarly, rotation and zoom can be set up in matrices to accomplish the desired effect.

4.3.3. Canvas Type

The technique implementation will inform the rendering context that it will use either a 2-D or 3-D canvas. The camera operations will be set up accordingly to provide the standard interaction for the type of canvas used. For example, the arrow keys on the keyboard may be mapped to panning operations. When using a 2-D canvas, the up and down keys could mean to pan the view up and down. When using a 3-D canvas, the up and down keys might be used to go forward and back in 3-D space.

4.3.4. Image Support

The rendering context will have a method that will provide access to the rendered image as an array of pixels. This will allow the user to save the rendered image to file. Image file exporters will be written as plugins that register themselves using the programming interface (Section 4.4). When the user requests to save the rendered image, a file chooser with the available file formats will appear, allowing
the image to be saved. The BMP format will be supported by default. Other file formats may be supported later by writing appropriate plugins.

4.4. Programming Interface

A programming interface will be provided that will allow appropriate access to the user interface and other functions necessary to plot the data and allow interaction. A plugin system will be provided to allow custom functionality. The plugin system will allow plugins to simply be placed in the correct directory. After restarting the program, any new plugin functionality will be available. All presentation techniques will be programmed as plugins that adhere to the programming interface. The interface will allow the technique implementations to specify the name of the technique, the visual cues available, and the types of interaction supported, along with any other technique specific information. The technique implementation may also specify callbacks that are activated through menu items. This will allow the technique to provide custom options and iterations. Through the programming interface, data points may be added or modified, allowing future custom file loaders.

4.5. User Interface (UI)

A user interface will be designed that will allow simple and easy interaction for users for the more common user interactions. For the purposes of this research, ‘simple and easy interaction’ will be determined by how difficult the functionality is to find in the UI. Specifically, functionality that can be found in toolbars or otherwise immediately visible in the UI or that is available through common keystrokes is considered to be ‘easy and simple’. Functionality found in menus or that is hard to find is considered to be ‘hard or difficult’ because the user has to search through the menus.

Most user interaction should be ‘easy and simple’ because interacting with the data is an important aspect of graphics packages. Interactions such as annotations, panning, rotation, zooming, selection of presentation techniques, and selection of visual cues are included in this group. Other options that are provided by the presentation techniques, such as special configuration that differs from the default, are allowed to utilize menus and other less easy interaction methods.

Common user interaction, as mentioned above, will be supported in the UI. For example, the user will be able to select a presentation technique and assign cues to variables to be presented. After initially displaying and possibly manipulating the data using the selected presentation technique and assignment of cues, the user will be able to select a different presentation technique and/or assignment of visual cues. The user interface will allow such interaction, as well as brushing (selecting and deselecting) interesting points of data, panning, zooming, rotating, etc. as appropriate for the selected presentation technique.

Figure 6 shows mockups of a preliminary UI design. In Figure 6(a), the presentation techniques will be laid out so that the user can see all of them easily and can select the data set and the desired presentation technique. Only a few are shown for illustration purposes. Once a technique has been chosen, a separate window will appear that will allow the user to assign visual cues to the variables in the data set, shown in Figure 6(b). The first window will remain to allow more presentation techniques to be selected for viewing. By using separate windows for each presentation technique, the user is able to view the data simultaneously using different presentation techniques, or perhaps use the same presentation technique with a different view or cue selection. The cue selection is made simple by placing the cue selection area next to the visualization area. The assignment of cues to variables and viewing the results will be straightforward. Interaction with the view will be enhanced through the tools in Figure 6(c). The tools will allow zooming, rotation, panning, annotations, and other interactions.
4.5.1. Interaction

The UI will be responsible to handle all user interaction. The UI will catch user input, such as typing and mouse clicks, and forward them to the technique implementation as appropriate. This alleviates duplicate code that would have been necessary in the technique implementation, allowing the plugin coding process to be simpler. When the technique implementation receives a user input, it will be responsible to check the current state of the tools and other settings through the programming interface and act accordingly.

4.5.2. Annotations

All annotations will be handled by the UI. The user creates new annotations by selecting one of the annotation tools shown in Figure 6(c). As the user performs clicking and dragging actions on the visualization, the UI will make draw calls to the rendering context for that view. The annotations will be given 3-D positions relative to the view position and orientation. The annotations will be stored in the rendering context. The rendering context can then use the annotation information to decide whether to select a data point or an annotation, as described in Section 4.3.1. The user may select an annotation using the arrow tool. The user may edit annotations using the corresponding annotation tool by double-clicking on the annotation. Appropriate interaction will follow to allow modifications of the annotations. For example, for a text annotation, a cursor will appear in the text, and the user will be able to change the text in the normal way. The properties of the annotations, such as color, line thickness, text size, or text font, may be changed by using the options in the ‘Properties’ section of the tool palette in Figure 6(c). The properties options will change according to which annotation tool is selected.

4.5.3. Brushing

Brushing may be done using either the brushing tools (brush or lasso) or by clicking on the range brush tool, located above the cue selection box in Figure 6(b). The brush tool will allow selection of points in a view using a click-and-drag rectangle. All points that fall in the rectangle will be selected. Clicking and dragging a second time will remove the selection of the first box and select the points that fall in the new box. However, if the user holds the ‘control’ (ctrl) key, any point falling in the new box will be added to the selection from the first box. If the user holds the ‘shift’ key, any points falling in the new box will be deselected, leaving the other points selected still. This interaction will allow the user to easily select the desired points in the view. The lasso tool will behave similarly. The lasso tool is used to draw a path around the desired points. The path is not conformed to any particular shape, allow great
flexibility in selection. The ‘control’ and the ‘shift’ keys will modify the lasso tool in the same way as the brush tool.

The range brush tool will allow the user to specify the range of values used for selection of points. Clicking the range brush tool will open a dialog that will have sliders corresponding to each of the possible variables. A checkbox below the slider will allow the user to choose whether or not to have that variable constrain the selection. The user may use some or all of the variable sliders. The selection of points will be updated in all corresponding views as the user moves the sliders.

4.5.4. Color

Color selection for each view will be available by clicking the color button above the cue selection box in Figure 6(b). A dialog box will appear, allowing the color of the background, the unselected points, and the selected points to be changed. The color of the background will only change the current view’s background color. This will allow the user to customize each view. If no points are selected, the option to change the color of the selected points will be grayed out. When the color of the selected points is changed, the model for the view will be notified. The model will store the color for those points in a special field. Other views that share the same model can either use that color or ignore it. It will also be possible for a technique implementation to allow other color selection, such as a default label color. The technique will inform the UI of other options by using the programming interface.

4.5.5. Consistent Tools

Consistent tools mean that all of the tools are available to all views. Annotations, panning, rotation, and zoom will be available to all views through the tools included in the tool palette, shown in Figure 6(c). The tools will use the functionality provided by the rendering context, described in Section 4.3.2.

4.5.6. Extensibility

The package will be extensible through a plugin architecture. The architecture will allow users to download a plugin, restart the application, and begin using the new functionality. The programming interface (Section 4.4) will provide an API for the plugin implementation to use to access the necessary methods. Functionality such as image exporters, data file loaders, and presentation techniques may be added using plugins.

4.5.7. Image Support

Image support will be handled by the rendering context as explained in Section 4.3.4.

4.5.8. Labels

All modification of labels will be handled by the UI. The technique implementation will register labels with the rendering context. Labels will then be treated as special types of text annotations. When the user clicks inside of the label bounds using the text annotation tool, editing of the label will follow the interaction describe in Section 4.5.3, including changing any of the properties, such as color, text size, and text font. Upon the completion of the editing of the label, the rendering context will inform the technique implementation of the new label value. This will allow the technique implementation to provide a default label, yet it will also allow the user to change the label’s value, if desired.

4.5.9. Linking

Linking will be handled using the model. A view will notify the model of changes, and the model will notify all of the corresponding views. The views will then refresh their rendering.
4.5.10. Multiple Data Sets

Multiple data sets may be opened using the open button on the main window, shown in Figure 6(a). Each data set will correspond to a model instance, and all views for that data set will link to the same model.

4.5.11. Multiple Views

Multiple views may be opened by selecting the data set and clicking on the desired presentation technique in the main window, shown in Figure 6(a). A new window will open where the presentation technique will be displayed. Interaction with the presentation technique will occur as previously explained.

4.5.12. Panning

Panning will be supported in two different ways. The tool palette in Figure 6(c) has a hand tool that will allow the user to ‘grab’ the visualization and move it using the mouse. The visualization will also have scrollbars when appropriate that the user may use to move the visualization around.

4.5.13. Point Display

The UI will handle displaying point values. When the mouse hovers over a data point, a tool-tip will appear and display the data point’s variable values. The data point under the mouse will be found using the algorithm described in Section 4.3.1.

4.5.14. Rotation

A rotation tool is provided in the tool palette, shown in Figure 6(c). The rotation tool will allow the user to rotate the view, as appropriate to the rendering context canvas type (Section 4.3.3). The ‘control’ (ctrl) key will be used to toggle the axis about which the visualization will be rotated. The rotation tool makes calls to the rendering context, which will change how the visualization is drawn.

4.5.15. Saving State/Sharing

The user will be allowed to save the state of all of the visualizations. When the save button on the main window is clicked, a dialog box will provide options to customize what is saved. For example, the user may have more than one data set open but may only want to save one of the data sets to the file. Another option may be to save user interactions that can later be applied to a different data set. After the options are selected, the information, including selection of points, coloring of points, view transformations, etc., will be saved in the XML format described in Section 4.1.2. The file may then be shared with others. When the file is later opened, all visualization will appear in the exact state that they were saved in.

4.5.16. Shadowing

Shadowing will be implemented using a special value stored in the model. Points that are selected may be marked as ‘shadowed’ by clicking the shadow button, located above the cue selection box in Figure 6(b). The view will inform the model that the selected points are ‘shadowed’. The model will then inform all corresponding views of the change. The views may choose to ignore the ‘shadowed’ setting.

4.5.17. Subsetting

Points that are selected may be subsetted into a new data set by clicking the subset button, located above the cue selection box in Figure 6(b). This is accomplished by creating a new model that contains only the selected points. The view will have its model switched to the new model, and the view will continue as normal.
4.5.18 Technique Changing

Technique selection will be provided through a drop-down menu, shown in Figure 6(b). The UI will be responsible to provide the drop-down menu and populate it with the available presentation techniques. When a user changes techniques, the rendering will be refreshed to reflect the change.

4.5.19. Tool Tips

The UI will support tool tips for all of the tools and buttons. A small rectangle with a short amount of text will appear next to the mouse to describe the functionality or name of the item when the mouse is hovering over it. Each presentation technique button in the main window, shown in Figure 6(a), will show the name each technique in the tool tip. Other items will show the name of the item and/or an appropriate message.

4.5.20. Variable Changing/Cue Selection

Cue selection will be provided through drop-down menus, as shown in Figure 6(b). The UI will be responsible to provide the drop-down menus and user interaction with them. When a user changes one of the visual cues, the UI will notify the technique implementation that a change occurred. The technique implementation will then refresh the rendering, which will reflect the change.

4.5.21. Zoom

The zoom tool provided in the tool palette, shown in Figure 6(c), will allow the user to selectively zoom in on the data. If the user clicks on the visualization without dragging, the view will zoom in a default amount, making the point where the user clicked be in the center of the view. If the user clicks and drags the mouse, a rectangle will appear that shows how much to zoom in. The view will be scaled according to the size of the rectangle. If the user right-clicks, the view will zoom out to the normal zoom. All zoom functionality is implemented using the rendering context as described in Section 4.3.2.

4.5.22. Summary of Features

The UI will tie all of the pieces of the graphics package together. The UI will support all of the features listed in Section 2.3. These features have been shown to be desirable and will allow the user to interact and explore his or her data.

4.5.23. Dimensionality Reduction

An additional, novel feature will be included in the package. The feature will use the persistence of vision of the retina to explore whether or not patterns in data may be seen while interactively changing the dimensionality of the display. A slider will be provided in the UI, as shown in Figure 6(b), that will allow the user to change the dimensionality of the presentation. The user will click on the slider, look at the visualization, and move the slider to see whether patterns may be visible when dropping and adding dimensions. The persistence of vision of the retina will merge the two images, allowing the brain to detect differences and similarities in the images.

4.6. Assumptions

Some assumptions about the data will be made. The framework will accept data that is correctly formatted and display it accordingly, but it will assume that the variables are independent. At times, the user is graphing the data in an effort to determine interdependency of the variables. It will be the user’s responsibility to determine and deal properly with any interdependency among the variables. It will be the user’s responsibility to correctly discretize continuous data. Similarly, it will be the user’s responsibility to determine the quantity of data that is sufficiently large for discovery of information and sufficient small to facilitate meaningful graphic presentation. These assumptions are fair and necessary for a generic hyperdimensional graphics system.
4.7. Deliverables

Using the design described here, a prototype system will be built that demonstrates the interface of the framework. A simple plugin API will be specified that allows plugins appropriate access to the package functionality. A file loader plugin and a presentation plugin will be written to show how the package may be expanded using the plugin architecture. To evaluate the user interface design, the user interface will be built out using a GUI library. It will be sufficiently developed to show how all of the features work together to enhance the visual exploration process and allow sharing of insights with others. It will be sufficiently developed to have the basic functionality of all of the tools in the toolbar, shown in Figure 6(c). The dimensionality slider will be included and will permit experimentation with the persistence of vision. A test data set will be procured and used to show the usage of the system.

4.8. Limitations

The prototype system will be developed only as a proof-of-concept. Only one data set is needed because it is straight-forward to apply the principles and the system to a different data set. Only two plugins will be written. This will be sufficient to demonstrate the basic usage of the plugin architecture. Writing similar plugins will be simple activities of copy, paste, and modify. The user interface will be developed only sufficient to show the basic functionality of the tool set and to demonstrate the utility of the tools in data exploration. Extra menu items that might accompany a production system may be excluded. While they would add to the functionality of the system, they are not requisite in showing the package’s intended purpose or in proving concepts.

5. Validation

A new, improved design for a graphics package for presentation of generic hyperdimensional data will be produced which overcomes the deficiencies of current hyperdimensional graphics packages. Specifically, the proposed graphics package will be capable of accommodating all hyperdimensional presentation techniques discussed here and will possess all of the features outlined in Section 2.3. The design will show that it can accommodate all known presentation techniques by providing a specific way to add functionality. The design shows this in Section 4.5.6. The design describes a plugin architecture that will allow plugins to be created that extend the base functionality of the package, including adding new presentation techniques. To show that the graphics package has all of the desired features, the design will provide a detailed list showing how each of the features has been satisfactorily provided for (see Section 4.5.1-Section 4.5.22). The UI will provide ‘simple and easy interaction’, as defined in Section 4.5. It will do so by providing a tool palette, tool bars, and other UI elements that allow most interactions to occur without looking through menus. A prototype system will be built to show proof-of-concept, as outlined in Section 4.7-Section 4.8.

6. Thesis Schedule

Completion of my thesis will include the following:

• 12/8/10 – Thesis proposal completion and acceptance
• 2/1/11 – Completion of the prototype system
• 3/1/11 – Preparation of the written thesis
• 4/12/11 – Thesis defense and submission
• 5/1/11 – Preparation and submission of a paper intended for publication
Annotated Bibliography


This is the technical overview of the ADVIZOR® package, which contains a description of what features are included. The plot types mentioned were of interest to this research.


ADVIZOR® is a business intelligence graphics package that allows the user to plot business data. The webpage referenced describes the products they offer and contains links to other useful webpages. The package is quickly reviewed to show desirable user interface features.


This webpage contains a gallery of videos that demonstrate the ADVIZOR® package. The videos show the functionality of the user interface.


This paper describes a presentation technique called the hyperbox, which is useful for rendering n-D data. It is one of the presentation techniques included in this research.


A method for rendering n-D data as glyphs is presented. These glyphs form the basis of other glyph rendering techniques. Glyphs are one of the presentation techniques included in this research.


Described in this article is a presentation technique that takes as input n-D data and outputs functions that show the relationship of the data. It is one of the presentation techniques included in this research.


Circle segments are a presentation technique that uses single pixels to display data values. The pixels are laid out in a circle segment for each dimension, creating a full circle. It is one of the presentation techniques included in this research.


The grand tour is a method of presenting the user with different projections of n-D data that allows the user to get a general idea of the distribution of points. It is one of the presentation techniques included in this research.

Tulip is an open-source, free n-D graphics package allowing n-D data to be rendered using different presentation techniques. It is one of the packages reviewed in this paper.

Gsharp allows the user to create and use different plot types to render data. It is mentioned in this paper but not reviewed.

Openviz allows the user to create and use different plot types to render data. It is mentioned in this paper but not reviewed.

This article describes methods for rendering 4-D data. It is one of the presentation techniques included in this research.

This webpage contains videos demonstrating the user interface. The videos are used to find desirable functionality in the user interface.

Centrifuge is a system that provides link analysis of data. It allows the user to discover relationships in the data using link graphs and charts. The package is quickly reviewed to show desirable user interface features.

This book describes various methods for graphing n-D data. Some presentation techniques in the book are included in this research.

Chernoff describes a method for representing n-D data using facial characteristics. Various attributes, such as eye shape, eye size, head shape, nose shape, nose size, nose position, may each be used to represent a dimension. This is one of the presentation techniques included in this research.

The star coordinate presentation technique is extended to the third dimension. This is one of the presentation techniques included in this research.

Since the plots on the diagonal of scatterplot matrices don’t convey new information, this paper explores the effect of putting other plots in place of the diagonal plots. This paper is mentioned in this research.


Data Desk is a statistical package that is augmented with charts for data visualization. This is one of the packages mentioned, but not reviewed, in this paper.


A framework is described which allows visualizations to be annotated using free-hand drawings drawn directly on the visualization. This paper is referenced in the section regarding annotations.


This is the website for Visual I|O, a data visualization package. This package is mentioned, but not reviewed, in the paper.


Worlds Within Worlds is a presentation technique that presents n-D data using hierarchal coordinate systems. This is one of the presentation techniques included in this research.


The projection pursuit algorithm is presented which attempts to find meaningful projections of the data. This is one of the presentation techniques included in this research.


This is the website for GIMP, an image manipulation program. GIMP is referenced in the paper in relation to its plugin system.


Various presentation techniques are reviewed in this paper. Many of these techniques are included in this research.


This paper describes a presentation technique called boxes. The length of the 3 dimensions of the first box convey the values of 3 variables. The height of subsequent boxes conveys other values of variables. Boxes is one of the presentation techniques included in this research.

The presentation technique called mosaics is described. Mosaics involves drawing boxes of varying sizes based on the value of the variable. This is one of the presentation techniques included in this research.

[Hillery2011multivariate] Hillery, B. Multivariate Data Visualization Via Outdoor Scenes. Accepted for presentation at the Conference on Visualization and Data Analysis 2011.

Variable values are mapped onto nature scenes. For example, a sparse scene might mean a small value, while a scene full of vegetation might mean a large value. This is one of the presentation techniques included in this research.


A presentation technique called Radviz is presented. n points are equally spaced on a circle. Springs are then attached to those points and to each of the data points. The spring strength is determined by the value of the variable for that spring. The points are plotted where the springs equal out. This is one of the presentation techniques included in this research.


This paper includes a presentation technique called modified circle segments, which uses alternating color and grayscale wedges. This is one of the presentation techniques included in this research.


Various presentation techniques are presented and/or reviewed. This includes Circular Parallel Coordinates, Gridviz, Multi-Line Graphs, Polyziv, and Survey Plots, which are presentation techniques included in this research.


MANET is a visualization package that is no longer being actively developed. It is referenced in this paper as such.


Analyst’s Workstation is a visualization package. Due to insufficient information, the package is only referenced in this paper.


Style Intelligence is a visualization package that supports a few of the presentation techniques relevant to this research. It is mentioned in this paper.


Style Scope is a visualization package that supports a few of the presentation techniques relevant to this research. It is mentioned in this paper.
The presentation technique called Parallel Coordinates is first described in this paper. Parallel Coordinates involves using a coordinate axis per variable that are set up vertically, with polylines crossing the axes at the data points’ values. It is one of the presentation techniques included in this paper.

Antaeus is a visualization package that uses scatterplot matrices to plot the data. It is mentioned in this paper.

A presentation technique called star coordinates is presented. The technique consists of an axis for each dimension coming out of a center point and equally spaced apart. Points are plotted as the vector sum all of the values. Moving the axes changes the direction of the vector for that value. This is one of the presentation techniques included in this research.

Parallax is a software package that supports parallel coordinates. It is mentioned in the paper.

Pixel-oriented techniques consist of encoding the value for a variable as a single pixel. Each variable would then have a collection of pixels representing the values of the data points for that variable. This is one of the presentation techniques included in this research.

VisDB is a visualization package that is no longer being actively developed. It is mentioned in the paper as such.

Trees and castles are two presentation techniques to display n-D data, so named for their appearance. Trees involves encoding values as branches of a tree. Castles involves drawing lines for each value, similar to bar graphs, with the lines connected. These presentation techniques are included in this research.

VisIt is a visualization package that supports parallel coordinates. It is mentioned in this paper.

Leadscope is a visualization package that seems to be geared mostly to chemical visualization. However, it does seem to support n-D data. It is mentioned in this paper.
This paper describes a presentation technique called dimensional stacking. It is one of the presentation techniques included in this research.

A presentation technique called piles is presented. An example is given using star plots. One dimension is the location on the center axis, while the other dimensions are shown as normal using star plots at the location on the axis. It is one of the presentation techniques included in this research.

This paper describes a presentation technique called the data image, also known as a heat map. It is one of the presentation techniques included in this paper.

NovoSpark® Visualizer is a visualization package that supports a few of the presentation techniques considered in this research. It is reviewed in this paper.

This paper describes a process of adding the pixel-oriented visualization presentation technique to the XmdvTool. It is mentioned as one of the papers describing additions to XmdvTool.

This paper describes iconographic displays. The example is of stick figures, which are placed based on data point values. This is one of the presentation techniques included in this research.
This paper describes a presentation technique known as the table lens. The table lens allows the user to focus on various parts of the table while retaining the context of the rest of the table. This is one of the presentation techniques included in this research.

Grapheur is a visualization package. Grapheur is mentioned, but not reviewed, in this paper.

Glyphmaker is a visualization package involving the use of glyphs to plot the data. A few glyph types are supported and can be switched by the user. Glyphmaker is mentioned, but not reviewed, in this paper.

The presentation technique called the sammon plot is described. Points are plotted such that the approximate structure of the data is relatively unchanged. It is one of the presentation techniques included in this research.

Visualization simulations can be accomplished using this system. Visualization simulations can be mentioned, but not reviewed, in this paper.

Orca is a visualization package written in Java that supports a few different types of visualizations. It features a pipeline structure to allow flexibility in the rendering process. It is mentioned in this paper as a package that is no longer being actively developed.

GGobi is a visualization package that evolved from the XGobi package. This paper describes the changes that were made, along with the reasons for doing so. GGobi is reviewed in this paper.
XGobi is the predecessor of the Ggobi package. It supports a few visualization techniques, including the grand tour and projection pursuit. It is no longer being actively developed and is mentioned in this paper as such.

This paper describes the XGobi package and its functionality. XGobi is no longer being actively developed and is mentioned in this paper as such.

This is the website for Tableau, a visualization package. It is mentioned, but not reviewed, in this paper.

This is the website for Mondrian, a visualization package. It is mentioned, but not reviewed, in this paper.

This is the website for Think EDM, a visualization package. It supports a couple of the presentation techniques included in this research. It is mentioned, but not reviewed, in this paper.

This is the website for Spotfire, a visualization package. It supports a few of the presentation techniques included in this research. It is mentioned, but not reviewed, in this paper.

Various presentation techniques are discussed, including scatterplot matrices and star plots. These are 2 of the presentation techniques included in this research.

This is the website for Visipoint, a visualization package. It is mentioned, but not reviewed, in this paper.

This is the website for Omniscope, a visualization package. Omniscope supports a couple of the presentation techniques included in this research. It is mentioned, but not reviewed, in this paper.

This is the website for VisuMap, a visualization package. VisuMap supports a couple of the presentation techniques included in this research. It is mentioned, but not reviewed, in this paper.


XmdvTool is a visualization package for n-D data that supports a few of the presentation techniques included in this research. It is one of the packages reviewed in this paper.


This is another source that, seemingly independently, presented heat maps. Heat maps is one of the presentation techniques included in this research.


Visalix is a visualization package, and this is its website. It is mentioned, but not reviewed, in this paper.


Parallel planes coordinates is a presentation technique that extends parallel coordinates using planes. It is one of the presentation techniques included in this research.


Parallel volumes coordinates is a presentation technique that extends parallel coordinates using volumes. It is one of the presentation techniques included in this research.


This describes hierarchal displays and how the XmdvTool has been extended to include hierarchal displays of all of its supported presentation techniques. It is mentioned as one of the papers describing the extension of XmdvTool.
BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis proposal submitted by

Lee Howard

This thesis proposal has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

___________________________________________________

Date  Robert P. Burton

___________________________________________________

Date  Bryan S. Morse

___________________________________________________

Date  William A. Barrett

___________________________________________________

Date  Kent E. Seamons  
Graduate Coordinator