2012-06-22

Technique and Cue Selection for Graphical Presentation of Generic Hyperdimensional Data

Lee Mont Howard

Brigham Young University - Provo

Technique and Cue Selection for Graphical Presentation

of Generic Hyperdimensional Data

Lee Howard

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

Master of Science

Robert P. Burton, Chair
Bryan S. Morse
William A. Barrett

Department of Computer Science

Brigham Young University

August 2012

ABSTRACT

Technique and Cue Selection for Graphical Presentation
of Generic Hyperdimensional Data

Lee Howard
Department of Computer Science, BYU
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 solve the first problem. However, existing packages do not sufficiently solve the other three problems.

A hyperdimensional graphics package that sufficiently solves all these problems simplifies the user experience and allows the user to explore, interact with, and share the data. I have implemented a package that solves all four problems. The package is able to render n-D data through appropriate encapsulation of presentation techniques and their associated visual cues. Through the use of an extensible plugin system, presentation techniques can be easily added and accommodated. Desirable features are supported by the user interface to allow the user to interact easily with the data. Sharing of visualizations and annotations are included to allow users to share information with one another. By providing a hyperdimensional graphics package that easily accommodates presentation techniques and includes desirable features, including those that are rarely or never supported, the user benefits from tools that allow improved interaction with multivariate data to extract information and share it with others.

Keywords: hyperdimensional graphics package, extensible plugin system, presentation technique, visual cue, annotation, n-D data, hyperdimensional visualizations, Petrichor

ACKNOWLEDGEMENTS

I thank Dr. Robert Burton for all of his encouragement and support, and for helping me through all of the revisions of this thesis and other relevant papers. I also thank Dr. Bryan Morse and Dr. William Barrett for their comments and suggestions. I appreciate the Computer Science Department for helping to fund the project. I am also grateful for my wife Becky for her continual support and encouragement which made completion of this research possible.

Table of Contents

List of Tables

List of Figures

x

Chapter 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 2.3). 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 Need for and Value of Hyperdimensional Presentation Techniques

Hyperdimensional presentation techniques have been developed to allow the simultaneous rendering of more than three variables. These techniques allow the user to see several 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. However, valuable information can be lost when any of the variables are suppressed.

Hyperdimensional presentation techniques permit 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 [64]. Figure 1(a) is drawn using parallel coordinates [33]. In parallel coordinates, each dimension is shown as a vertical axis, and data points are rendered as polylines that intersect each axis at the appropriate value for that dimension. Trends can be seen by analyzing where the lines intersect

1

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 [59]. 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 [59]. 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 [40]. 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 in number 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 detected. A graphics package that easily accommodates available presentation techniques can be expected to provide the user with more information than a graphics package that doesn't easily accommodate available presentation techniques.

2
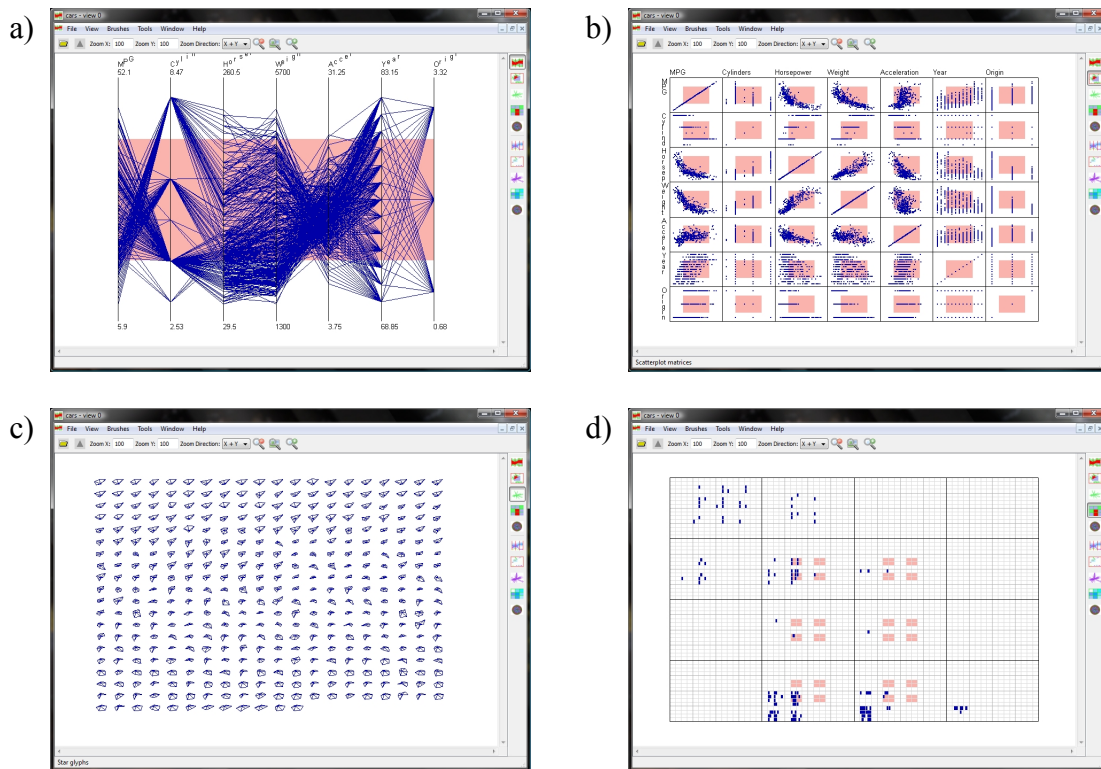
Figure 1. XmdvTool plots of the same data set using a) parallel coordinates, b) scatterplot matrices, c) star plots, and d) dimensional stacking. Images are screenshots of the XmdvTool program.

1.2 The Visualization Process

While many presentation techniques have been developed to visualize n-D data, obstacles remain to visualizing, exploring, and understanding n-D data. When considering the visualization process from the user's perspective, visualizing n-D data may seem daunting. The user may have several questions, such as 'What graph will show my data the best?' and 'Where do I find a program to graph my data?' Finding the answers to these and other questions will meaningfully improve the user's visualization experience.

In essence, the user with n-D data is confronted with four problems:

- finding a hyperdimensional graphics package capable of rendering that 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 in a meaningful way.

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

4

Chapter 2

Related Work

To determine if a package fits the needs of the user, we look first at current packages, both 3-D and n-D, to determine what is desirable in a package and if any existing packages meet all of the qualifications. All graphics packages have features that are intended to help the user of the graphics package to easily navigate and interact with the data that is visualized. However, the feature sets vary greatly between packages. By looking at the spectrum of available packages, we can compile a list of desirable features that are intended to help solve the problems a user faces when trying to visualize n-D data.

2.1 Lower-dimensional Presentation Technique Features

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. Exploring how users can interact with these presentation techniques can help to identify useful features that can be included in the development of a user-friendly hyperdimensional graphics package. In the following subsections, two graphics packages that support some of these lower-dimensional presentation techniques are briefly reviewed to provide insight to their user interaction.

2.1.1 ADVIZOR® Analyst

ADVIZOR$^{®}$ Analyst [3] is a commercial business intelligence tool that focuses on helping people see important trends in their data. It supports many presentation techniques,

5

including the bar chart, line chart, histogram chart, pie chart, and scatterplot [1]. The user interface employed allows users to quickly look through their data and find interesting trends and correlations [2]. 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 values by hovering over it, and

- Ability to save interaction state, such as selection, coloring, view transforms, etc.

2.1.2 Centrifuge®

Centrifuge® [12] 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 values by hovering over it, and

- Ability to save interaction state, such as selection, coloring, view transforms, etc.

6

2.1.3 Summary

While the review of these packages has not been exhaustive, it identifies some useful interaction features that would benefit the user in a hyperdimensional package, as well. These features include

- **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. One way to do this is by using a click-and-drag rectangle.

- **Zoom** – Zooming in on a portion of a plot should be easy. One way to do this is by using a click-and-drag rectangle.

2.2 Hyperdimensional Graphics Package Features

The interaction features found in Section 2.1 are a good foundation to build upon. However, more features are needed to adequately visualize, analyze, explore, and share n-D data. For example, some presentation techniques require more forms of interaction. 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. After researching current hyperdimensional graphics packages and the user interaction they provide, I have compiled a list of features that sufficiently solve the problems from Chapter 1. Some of the features included below are found in current graphics packages, while 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. For convenience, they are listed in alphabetical order.

2.2.1 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 drawn directly on the visualization and saved to file. This allows other users to see the annotations by simply sharing the file. 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 may be specified to allow commenting and labeling to appear 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 [18]. 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. Such shapes may include circles, boxes, lines, polylines, and polygons.

2.2.2 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 can also allow a change in selection, explained as follows.

- **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** – It is often beneficial to be able to select points in a view and see them in other views (see 'Linking' below). 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 usually accomplished through holding a key, like the control key, and dragging with the tool over the desired/undesired points.

2.2.3 Coloring

Color is an attribute that can help distinguish parts of a visualization from other parts. Many color options may be specified, including the following:

- **Point Color** – The data point's color may be changed, regardless of whether or not color is used as a visual cue or to signify something else in the view. This allows groups of points to be distinguished from other points.

9

- **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.

2.2.4 Consistent Tools

The primary tools that are available to one view should be available to any view. Tools such as zoom, pan, annotations, 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.

2.2.5 Dimensionality Reduction

There may be more variables than the user wants to see all at once. Allowing the user to reduce the number of variables shown provides the flexibility to view all or part of the data set. This can help the user find unexpected trends without having to change the data set.

2.2.6 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 graphics package such that others in the community may add to the functionality of the package without having to add directly its source code. An example of this can be seen in GIMP, an open-source image manipulation program [24]. GIMP supports plugins and scripting. Due to this, there is a massive collection of plugins and scripts available for download that greatly enhance the functionality of GIMP. Users can simply download the

المنارة للاستشارات

www.manaraa.com

plugin or script and place it in the correct folder. GIMP then reloads the plugins/scripts on command, and the new functionality is available. No change to the source code of GIMP or to the installed program is needed.

2.2.7 Image Support

After using the graphics package to visualize the data, the user may need to create an image of the visualization to include in reports, websites, etc. Screenshots of the visualization are inconvenient and lead to unnecessary image cropping. The graphics package should allow the user to save the visualization as an image in a common format so that the image can easily be used in other programs.

2.2.8 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.

2.2.9 Linking

Linking is the process by which the selection of data points in one view highlights the same data points shown in other views. Seeing the data using different presentation techniques can lead to additional insight not afforded by a single technique. The same is true when

11

highlighting points. Highlighting points in all views shows the same data in ways specific to the technique used, allowing further insight.

2.2.10 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. The rendering of more than one dataset should be supported.

2.2.11 Multiple Views

To be able to extract sufficient information from visualizations, it is often necessary to view multiple views side by side. At least several views should be allowed 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.

2.2.12 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.

12

2.2.13 Point Display

Many presentation techniques do not display all of the values of a data point in the view. To accommodate a user's desire to see the 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 common method for displaying the information.

2.2.14 Rotation

Rotation is the process of pivoting the view around a point or line, or parallel to a plane. 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.

2.2.15 Saving State/Sharing

Users interacting with their data will want to save the state of the visualization so they can revisit it later. The saving of state allows the visualizations to be returned to the same state as 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.

2.2.16 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 showing only points of interest in the view.

2.2.17 Subsetting

Subsetting is where points that are of interest are selected and copied to a new data set. The new data set may contain some or all of the original data set. This can potentially reduce clutter in the view by showing fewer data points. 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.

2.2.18 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 aids in viewing the data in many different forms without the need to always create a new window for a different presentation technique.

2.2.19 Tool Tips

There are often tools that the user doesn't understand or recognize immediately. 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.

14

2.2.20 Variable Changing/Cue Selection

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 view. As this is a frequent interaction, it should be prominent in the user interface.

2.2.21 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. A zoom tool using a click-and-drag rectangle is sufficiently easy to use. A secondary form of zoom may be provided, as well.

2.2.22 Summary

Many desirable features have been described which would benefit the user of a hyperdimensional graphics package. A graphics package that supports these features as outlined will help the user to interact with and explore the data in meaningful ways.

2.3 Multi-spectrum Hyperdimensional Presentation Techniques and Visual Cues

In addition to desirable features, a hyperdimensional graphics package must be able to accommodate many presentation techniques. This allows the user to explore the data using many different forms of visual presentation. To easily accommodate 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 or could be used in presentation techniques include the following, again listed in alphabetical order:

- Angle – The angle between two lines varies according to the value.

- Attributes – Varying the attributes of an object indicates the values. For example, in the presentation technique 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.

16

- 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. Table 1 contains a list of some of the presentation techniques that have been developed. The presentation 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.

Name of Technique	Angle	Attributes	Color	Curvature	Distance	Function	Hierarchal	Length	Level	Orientation	Position	Quantity	Rotation	Shading	Shape	Size	Structure	Symbol	Taper	Texture	Thickness	Transparency	Other
Andrew's Plot [6]						X																	
Boxes [26]								X								X							
Cartesian Hyperspace Graphics [11]																							X
Castle Symbols[38]																X							
Chernoff Faces [14]		X																					
Circle Segments [7]			X																				
Circular Parallel Coordinates [29]											X												
Dimensional Stacking [40]							X																
Glyphs [5]	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X	X	X	X	X	X
Grand Tour [8]																							X
Gridviz [29]																							X
Heat Map [43,65]			X																				
Height Map [25]			X																				
Hyperbox Display [4]			X								X							X					X
Iconographic Display (Stick Figures) [47]	X										X												
Modified Circle Segments [31]			X																				
Mosaic Plot [27]			X													X							
Multi-line Graph (on separate axes) [29]			X								X												
Nature Scenes [28]																							X
Parallel Coordinates [33]											X												
Parallel Planes Coordinates [55]																							X
Parallel Volumes Coordinates [67]																							X
Piles [41]								X			X												
Pixel-Oriented Techniques [37]			X																				
Polar Glyphs [25]											X												
Polar Lines [25]											X												
Polyviz (Expanded Radviz) [29]																							X
Profile Plots [13]											X					X							
Projection Pursuit [22]											X												X
Radviz [30]																							X
Sammon Plot [50]						X																	
Scatterplot Matrices [59]											X												
Star Coordinates [35]						X																	
Star Coordinates (3-D) [15]						X																	
Star Plot [59]								X															
Survey Plot [29]																X							
Symbolic Scatterplots [13]	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X	X	X	X	X	X
Table Lens [48]								X															
Tree Symbols [38]	X							X													X		
Worlds Within Worlds (N-Vision) [21]							X																

X = Presentation Technique Uses the Visual Cue

Table 1- Hyperdimensional presentation techniques with their associated visual cues.

2.4 Hyperdimensional Graphics Packages

Using the list of desirable features and the list of presentation techniques, we can now evaluate available hyperdimensional graphics packages. Several graphics packages can be found in literature and on the market. I have chosen to discuss some of these packages in detail based on their supported features, number of supported presentation techniques, availability, and relevance to this research. The following subsections provide overviews of these hyperdimensional graphics packages as well as images showing visualizations for each package.

2.4.1 GGobi

GGobi is a project written by Swayne, Cook, Lang, and Buja to overcome the shortcomings of its predecessor, the XGobi system [53]. GGobi is written using the GTK+ toolkit to help with cross-system compatibility. It uses an XML file format to allow the information that was originally in many files in the XGobi system to be stored in one file. All of the windows run on the same process which allows many plots to be generated without the overhead of setting up different processes. GGobi is 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.

2.4.2 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 [9]. 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 3 show a data set of a poker game using some of the presentation techniques.
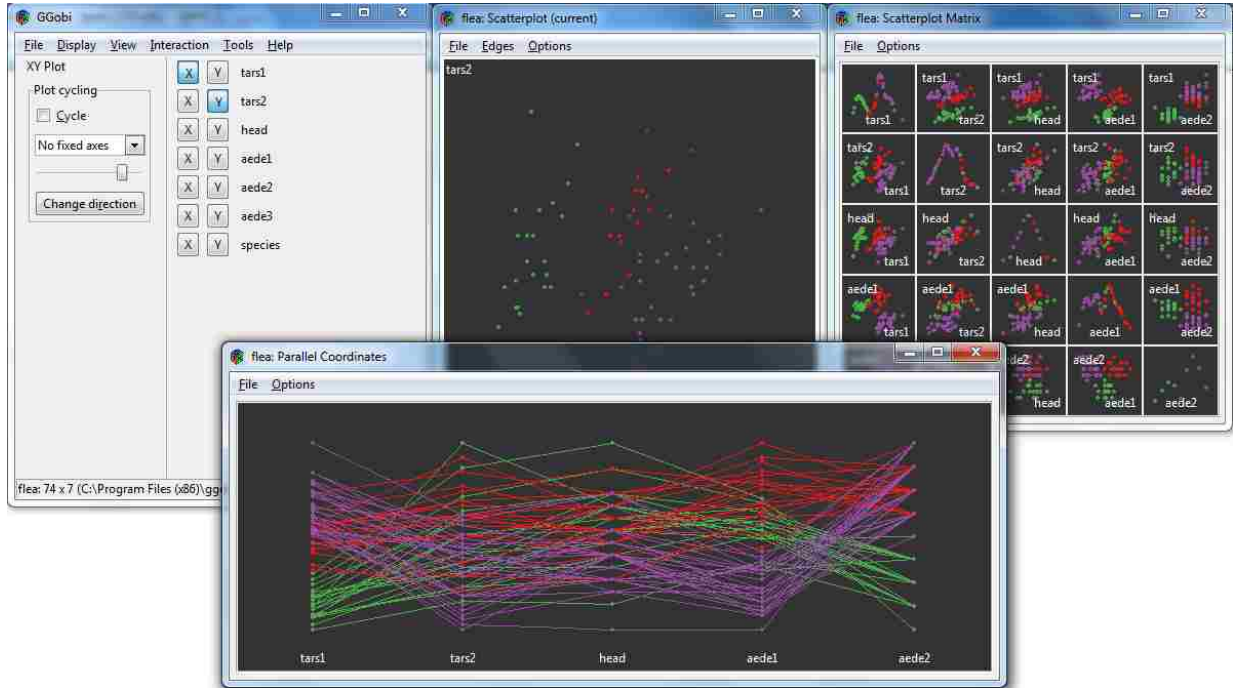
Figure 2. A screenshot of GGobi with 3 different plot types of the same data set.
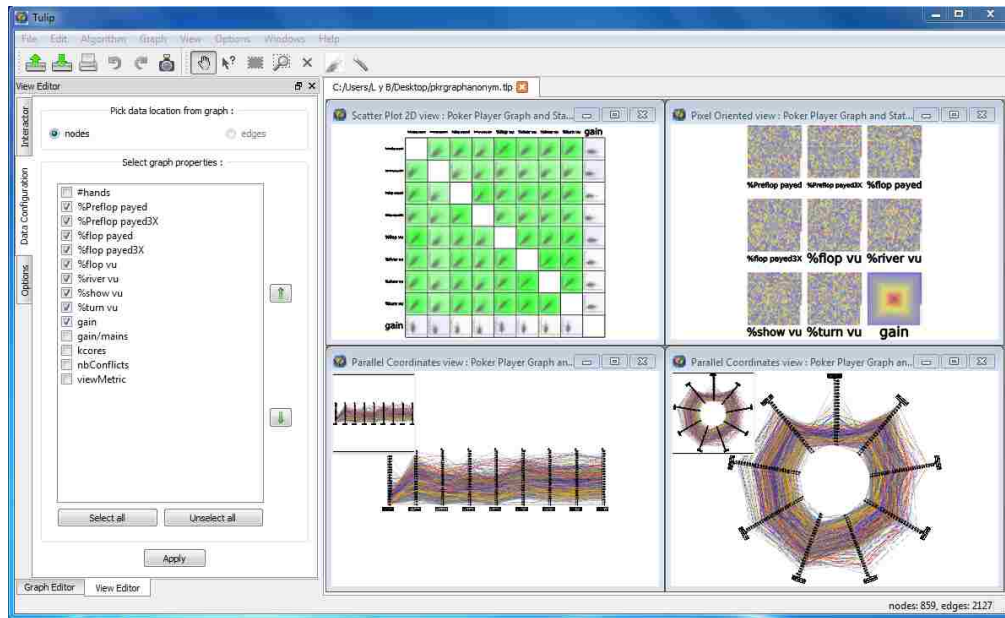
Figure 3. Tulip plotting a data set using circular parallel coordinates, parallel coordinates, pixel-oriented techniques, and scatterplot matrices.

20

2.4.3 XmdvTool

XmdvTool is a program originally written by Ward [64]. 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 [16, 46, 68]. Figure 4 shows one of the default data sets plotted using a few of the techniques.
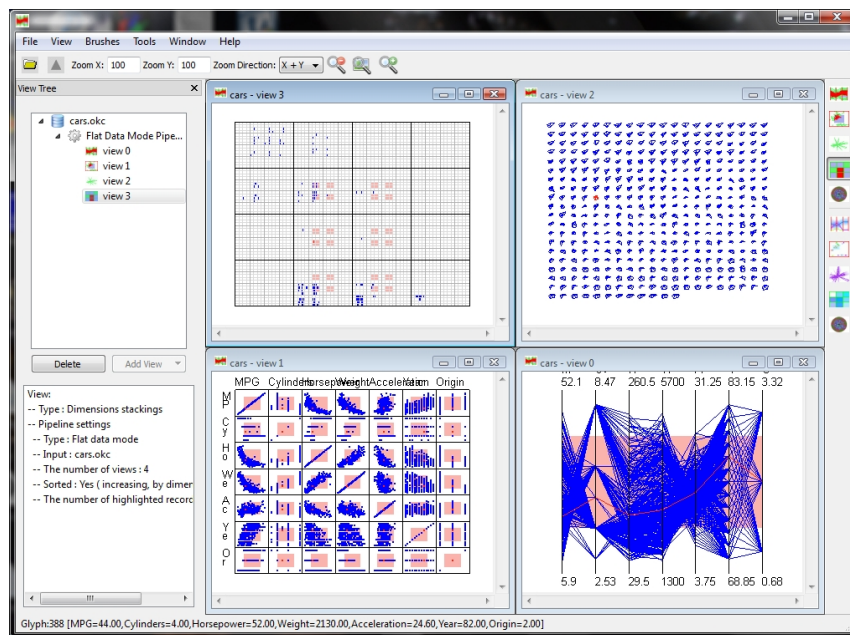
Figure 4. XmdvTool plotting a data set of car information using dimensional stacking, starplots, scatterplot matrices, and parallel coordinates.

2.4.4 Summary

Table 2 contains a listing of several graphics packages showing their supported presentation techniques. While this list of graphics packages is not exhaustive, it is significant that many presentation techniques are not supported in many of the graphics packages. Table 3 shows a summary of the supported desirable features for each of the packages discussed above. Many desirable features are not supported in these graphics packages. In my research, I was

unable to find a hyperdimensional graphics package that had more than about 50% of the

desirable features as described.

Package Name	Andrew's Plot	Boxes	Cartesian Hyperspace Graphics	Castle Symbols	Chernoff Faces	Circle Segments	Circular Parallel Coordinates	Dimensional Stacking	Glyphs	Grand Tour	Gridviz	Heat Map	Height Map	Hyperbox Display	Iconographic Display	Modified Circle Segments	Mosaic Plot	Multi-line Graph (separate axes)	Nature Scenes	Parallel Coordinates	Parallel Planes Coordinates	Parallel Volumes Coordinates	Piles	Pixel-Oriented Techniques	Polar Glyphs	Polar Lines	Polyviz	Profile Plots	Projection Pursuit	Radviz	Sammon Plot	Scatterplot Matrices	Star Coordinates	Star Coordinates (3-D)	Star Plot	Survey Plot	Symbolic Scatterplots	Table Lens	Tree Symbols	Worlds Within Worlds
ADVIZOR® Analyst [3]												S								S																				
Anteus [34]																																S								
Centrifuge® [12]												S																												
DataDesk® [17]																				S												S						S		
DataMontage™ [52]																		S																						
Ggobi [23,53]										S								S		S									S			S								
Grapheur [49]									S											S												S								
Gsharp [10]																										S						S								
Miner3D® [42]									S			S																				S								
Mondrian [56]																	S			S												S								
NovoSpark® Visualizer [44]	S																	S		S						S						S								
Omniscope [61]									S			S								S																				
Parallax [36]																				S																				
Partek [45]									S			S																												
ScienceGL [51]													S																											
Spotfire® [58]									S			S								S												S						S		
Style Intelligence [32]									S																							S						S		
Tableau® [54]									S			S					S															S						S		
Think EDM [57]									S			S																										S		
Tulip [9]							S													S					S							S								
Visalix [66]																																		S						
Visual i|o [19]									S			S																												
Visipoint [60]																															S									
VisIt [39]									S			S								S																				
VisuMap [62]									S			S																			S									
XmdvTool [63,64]								S												S					S							S	S							

S = Supported

Table 2 – A sampling of hyperdimensional graphics packages and their supported presentation techniques. The values are marked as accurately as possible, but due to the nature of user interfaces (features can be easily hidden), as well as the nature of software (software is not always freely available), some values may not be marked correctly.

Package Name	Annotations: Text	Annotations: Highlighter/Pen	Annotations: Shapes	Brushing: Brush Tool	Brushing: Range Brush	Coloring: Change Selection	Coloring: Point Color	Coloring: Background Color	Consistent Tools: Annotations	Consistent Tools: Panning	Consistent Tools: Rotations	Consistent Tools: Zoom	Dimensionality Reduction	Extensibility	Image Support	Labels: Default Labels	Labels: Label Editing	Linking	Multiple Data Sets	Multiple Views	Panning	Point Display	Rotation	Saving State/Sharing	Shadowing	Subsetting	Technique Changing	Tool Tips	Variable Changing	Zoom
GGobi [23,53]				S	S		S	P					S	S		S		S	S	S	S	P		P	S	P		S	S	S
Tulip [9]				S		S		S	S	S			S	S	S	S		S	S	S	S	S	S	P				S	P	S
XmdvTool [63,64]					P		P	P	S		S	S	S	S				S	S	S	S	P					S	S	P	S

S = Fully Supported   **P** = Partially Supported

Table 3 – Selected hyperdimensional graphics packages with their supported features. The values are marked as accurately as possible, but due to the nature of user interfaces (features can be easily hidden), some values may not be marked correctly.

Chapter 3

Project Description

After reviewing currently available graphics packages, we can conclude that none of
them sufficiently solves the problems a user faces when attempting to visualize data. Therefore, I
have designed and implemented a hyperdimensional graphics package that meets the user's
needs. The design of this package focuses on allowing good user interaction with the data,
including the desirable features from Chapter 2, as well as allowing for extensibility through a
plugin system.

The way a user interacts with a graphics package is limited to what is implemented by the
package. However, we don't want the user to feel limited. Instead, we want to provide an
environment where the user can become immersed in visualizing, exploring, and analyzing the
data. Previously, I outlined some desirable features that permit the user to interact with the data.
These features have been included in a new package called Petrichor. The word Petrichor means
the smell of rain. Just as the smell of rain is a simple pleasure, visualizing data should be simple
and pleasurable. The following sections describe the design and implementation used for
Petrichor's back end. Petrichor's user interface is described in Chapter 4.

3.1 File Format

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

3.1.1 Variable Types

The variables loaded into Petrichor may be classified as either numeric or categorical. A numeric variable is one whose value is treated as a number in the conventional fashion. A categorical variable, also known as a nominal variable, is one whose values are of a fixed set whose elements are not necessarily treated like numbers. 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. Petrichor assigns numeric values to each nominal value so that all presentation techniques can plot the data, but each presentation technique can choose to use the raw value directly. In this example, values could be assigned as follows: Ford = 1, GMC = 2, Honda = 3, and Toyota = 4. The user interface allows these assigned values to be changed, if desired. For example, some presentation techniques use only the assigned numeric value to render the data. This may result in an ordering that is not optimal. The user can change the assigned values to achieve an ordering that is optimal for the specific data set.

3.1.2 CSV Format

One of the file formats that Petrichor supports is the CSV format. This format is widely known, easy to understand, and efficient to process. Data is stored using rows and columns. Each column represents a variable, and each row (line in the file) represents a data point. The names of the variables are included as the first row in the file. An optional second row may be included to specify the types of each variable. Data point values are included in the subsequent lines. If the variable types are not specified in the optional second row, the variables are determined to be numeric or nominal, based on whether the variable values contain just numbers or numbers and

letters. By using this simple format, Petrichor is able to accommodate data from a wide spectrum of sources.

3.1.3 XML Format

Petrichor also supports a custom XML format. The XML format allows great flexibility in defining a structure to store information. 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 is defined using the nodes in the XML. The XML starts at a root document node containing information regarding plugin specific information, such as which plugin wrote the file, which version of the plugin was used, etc. The XML then has two distinct sections: one section for storing data sets, and one section for storing information relating to the visualization windows. In the data set section, the data is stored, including any color, selection, or shadowing information as well as the variable names and variable types. The visualization window section contains the view-specific information necessary to recreate the visualization. Annotation information, view variables, and plugin variables are stored in this section. Each window node also contains a reference to its data set. By using a more extensive file format, Petrichor is able to save the user's visualizations for later display and sharing.

3.1.4 Other 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. Each way of storing the data has its benefits. The file formats chosen are

26

sufficient to demonstrate Petrichor's functionality; other file formats can easily be added later by writing a file loader for a given format that conforms to the programming interface.

3.2 Model

Once the data is read in from file, it is added to a model class that uses a two-dimensional array to store the data. Each column represents a variable, and each row represents a point of data. The objects stored in this array are small class objects holding a number, the original text value, and other applicable information. Data points with missing values can be marked by setting a flag. This allows the model to store both numeric and categorical data, as well as whether or not the value is missing, in the same object. Multiple data sets are handled by creating a new model for each data set.

The model also calculates and stores various statistics related to the data. For example, after all of the points have been added, each variable's values are normalized. Other values, such as min, max, and mean, are calculated at the same time. These values are intended to facilitate rendering in the presentation techniques.

The model has a set of methods that allow presentation techniques to interact with it. Getter methods allow the techniques to request information about data points. Setter methods allow the techniques to change various attributes of the data points, such as color, selection, and shadow. Linking, the process of selecting points in one view and highlighting them in other views, is accomplished through a listener interface. The view in which points are selected informs the model that an attribute has changed. The model then informs all of the views listening to it that the attribute has changed. The views then tell the techniques to update their rendering to reflect the change.

27

3.3 Rendering Context

The rendering context is the base class that handles all of the rendering for Petrichor. The rendering context implements a simple drawing interface that allows presentation techniques to create visualizations. When a window is created for a presentation technique, a rendering context is created and provided to the presentation technique implementation. The technique implementation calls methods on this class to do its graphical rendering.

The rendering context class is written as a wrapper for the underlying rendering code. This allows the graphics package to potentially 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. Currently, the rendering context uses OpenGL to render the visualization.

3.3.1 Selection

To aid in the selection of data points, Petrichor requires a point id to be specified for the visual elements that are drawn for that point. The presentation technique plugin does this by passing the point id to the rendering context before the drawing calls for that point. When the user clicks on the visualization, the rendering context tells the technique to redraw itself. The rendering context then receives the draw calls, but instead of redrawing the picture, it finds the data point that should be selected. Selection is done this way to simplify the technique implementation and to allow the graphics package to handle all annotations (Section 4.2) and labels (Section 4.9) in a consistent way.

28

3.3.2 Consistent Tools

The rendering context is responsible for providing annotation, panning, rotation, and zoom functionality. This functionality is consistent in all views, meaning that annotation, panning, rotation, and zoom functionality are available in all views, regardless of the presentation technique used. This is accomplished by using a virtual camera in the rendering context and by using the selection technique outlined in Section 3.3.1. When a presentation technique renders a visualization, transformations are performed on all rendering calls based on the virtual camera. This allows the rendering context to "move" the content of the visualization without the presentation technique needing to change how the visualization is drawn. The code for the presentation technique is simpler because the rendering context handles all of the panning, rotation, and zoom functionality. Similarly, the rendering context handles all of the annotation functionality without involving the presentation technique.

3.3.3 Image Support

Petrichor supports saving the visualization as an image file through the plugin interface described in Section 3.4. When the user requests to save the rendered image, a file chooser with the available file formats appears, allowing the image to be saved. The BMP format is supported by default. Other file formats may be supported later as appropriate plugins are written.

3.4 Programming Interface

A programming interface is provided that allows appropriate access to the user interface and other functions necessary to plot the data and allow interaction. A plugin system allows custom functionality to be added to the system by third-party programmers. The plugin system

29

requires only that the plugins be placed in the correct directory. After restarting the program, any new plugin functionality is available. All presentation techniques have been programmed as plugins that adhere to the programming interface. The interface allows 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 allows the technique to provide custom options, such as toggling the visibility of a legend or other visual item. Through the programming interface, data points may be added or modified, allowing custom file loaders to be written.

Chapter 4

User Interface

The user interface (UI) for Petrichor has been designed and implemented to facilitate simple and easy interaction for the more commonly used features. For example, technique and cue selection have been given prominent locations in the UI so that the user doesn't have to search through menus to find the desired functionality. Other commonly used features, such as annotations, panning, rotation, and zooming, were placed in a tool palette to allow easy access. Figure 5 shows an overview of the UI.
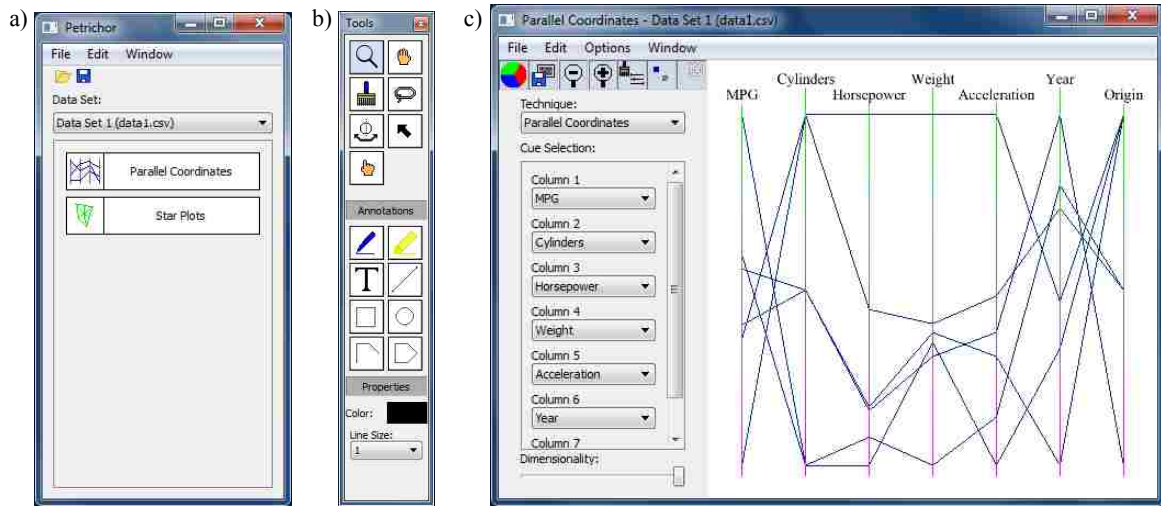
Figure 5. Basic user interface layout.

When the user starts up the program, the window in Figure 5(a) is shown. The user is then able to open a data file. Any data sets contained in the file can be selected from the drop-down menu. Once the user has selected the data set from the menu, he clicks on one of the presentation technique buttons below the menu. A new visualization window is opened and the data is rendered using the corresponding presentation technique plugin, as shown in Figure 5(c). The first window remains to allow more presentation techniques to be selected for viewing. Each click on a presentation technique button results in a new visualization window. By using separate

31

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 tool palette, Figure 5(b), and the UI elements in the left side of the visualization window are used to interact with the data. The user is able to assign cues to variables to be presented using the drop-down menus provided. After initially displaying and possibly manipulating the data using the selected presentation technique and assignment of cues, the user is able to select a different presentation technique and/or assignment of visual cues. The following subsections describe how Petrichor includes the desirable features outlined in Section 2.2.

4.1 User Interaction

The UI handles all user interaction. The UI catches user input, such as typing and mouse clicks and handles it appropriately. This alleviates duplicate code that would have been necessary in the technique implementation, allowing the plugin coding process to be simpler. If the technique requires mouse input from the user, a special tool in the tool palette can be used to forward mouse events to the technique plugin.

4.2 Annotations

All annotations are handled by the UI. The user creates new annotations by selecting one of the annotation tools shown in Figure 5(b). As the user performs clicking and dragging actions on the visualization, the rendering context for that window stores and updates the annotations. The annotations are given 3-D positions relative to the virtual camera position and orientation. The rendering context can use the stored annotation information to decide whether to select a

data point or an annotation, as described previously in Section 3.3.1. The user may select and move an annotation using the arrow tool. While the annotation is selected, the user may change annotation properties, such as color, line thickness, text size, or text font, by using the menu items in the 'Properties' section of the tool palette. The properties options change according to which annotation tool is selected to show the appropriate annotation properties. Figure 6 shows an example of annotations being used to add useful information to the visualization.
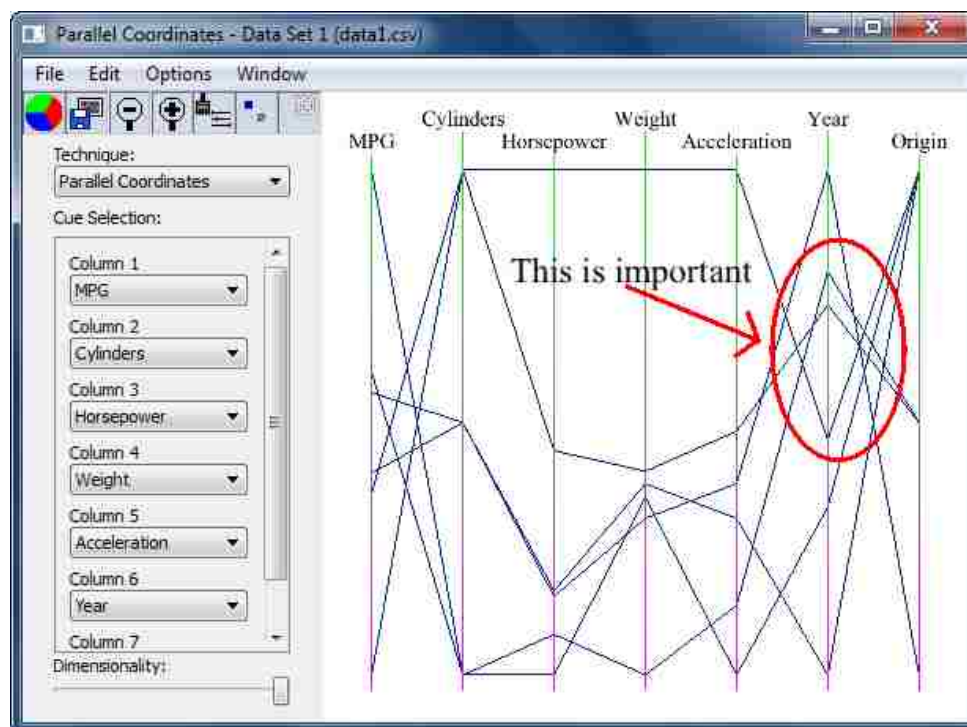
Figure 6. An example of a visualization enhanced by annotations. The text, arrow, and oval were added using the provided annotation tools.

4.3 Brushing

Brushing, or selection of data points, 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 5(c). The brush tool allows selection of points in a visualization using a click-and-drag rectangle,

shown in Figure 7(a). All points that fall in the rectangle are selected. Clicking and dragging a second time removes the selection of the first box and select the points that fall in the new box. However, if the user holds down the 'control' (ctrl) key, any point falling in the new box is added to the selection from the first box. If the user holds down the 'shift' key, any points falling in the new box are deselected, leaving the other points selected still. This interaction allows the user to select easily the desired points in the visualization. The lasso tool behaves similarly. The lasso tool is used to draw a path around the desired points, shown in Figure 7(b). The path is not conformed to any particular shape, allowing great flexibility in selection. The 'control' and the 'shift' keys modify the lasso tool in the same way as the brush tool.

The range brush tool, shown in Figure 8, allows the user to specify the range of values used for selection of points. Clicking the range brush tool opens a dialog that has sliders corresponding to each of the possible variables. A checkbox to the side of the slider allows 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 is updated in all corresponding views as the user moves the sliders.
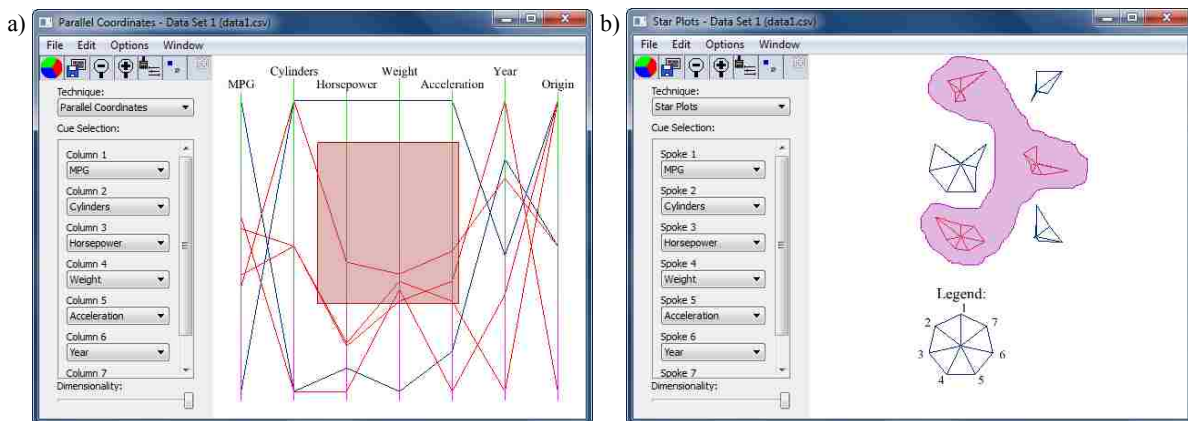
Figure 7. An example of the brush tools. Image (a) shows the brush tool selecting points. Image (b) shows the lasso tool selecting point.

34
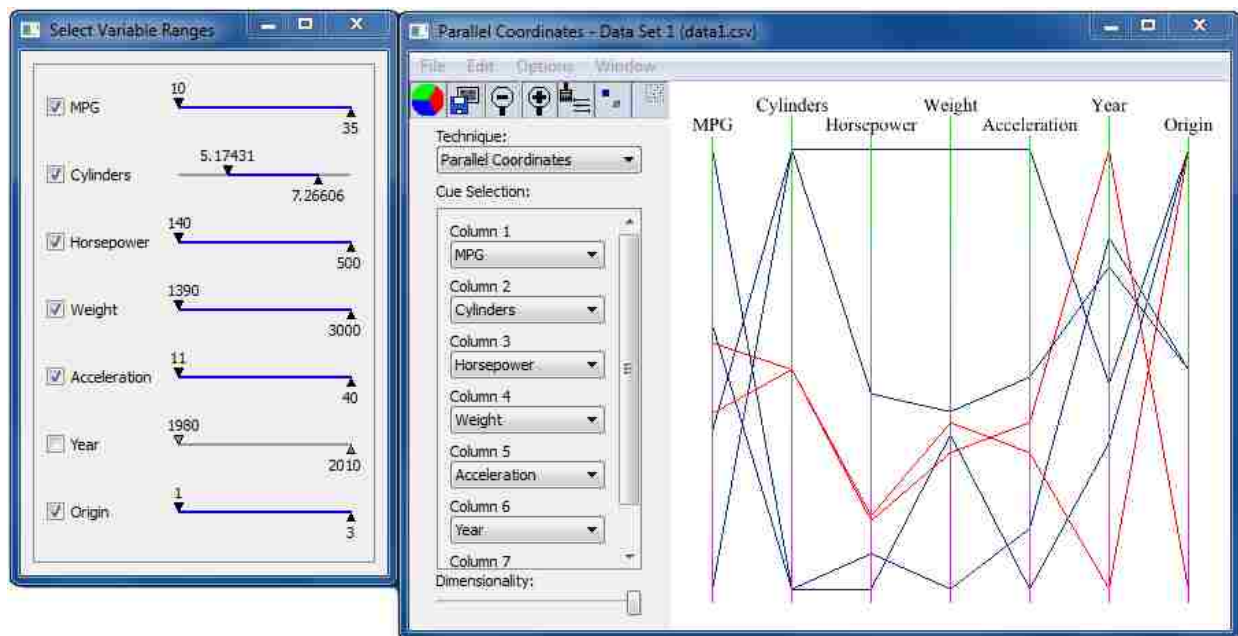
Figure 8. Brushing accomplished using range sliders.

4.4 Color

Color selection for each view is available by clicking the color button above the cue

selection box in Figure 5(c). A dialog box appears, Figure 9, allowing the color of the

background, the unselected points, and the selected points to be changed. Changing the color of

the background in the dialog box changes only the current view's background color. This allows

the user to customize each view. When the color of the selected points is changed, the model for

the view is notified. The model stores the color for those points. Other views that share the same

model can either use that color or ignore it. It is also possible for a technique implementation to

allow the user to choose other colors, such as a default label color. The UI asks the technique

what options are available using the programming interface.
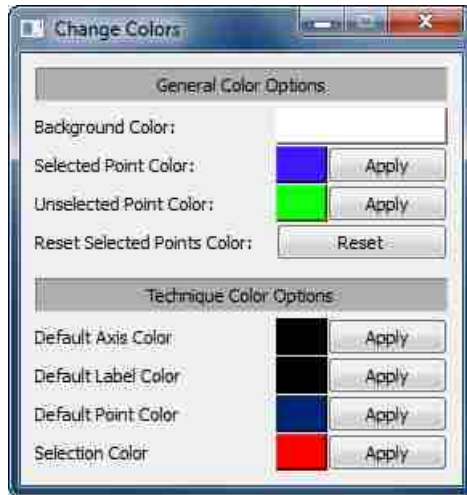
Figure 9. The color dialog box allows the user to specify the color of the background, selected points, and unselected points. If the presentation technique has color options, the user may specify those, as well.

4.5 Consistent Tools

Consistent tools mean that all of the tools are available in all visualization windows, including the tools in the tool palette, shown in Figure 5(b). The user can expect to be able to use all of the tools without worrying about which presentation technique is being used. The tools function as described in Section 3.3.2.

4.6 Dimensionality Reduction

A slider is provided by the UI, as shown in Figure 5(c). It allows the user to change the dimensionality of the presentation. As the user drags the slider, the dimensionality is changed in real time. The maximum and minimum dimensions are specified by the presentation technique. Figure 10 shows an example of dimensionality reduction.
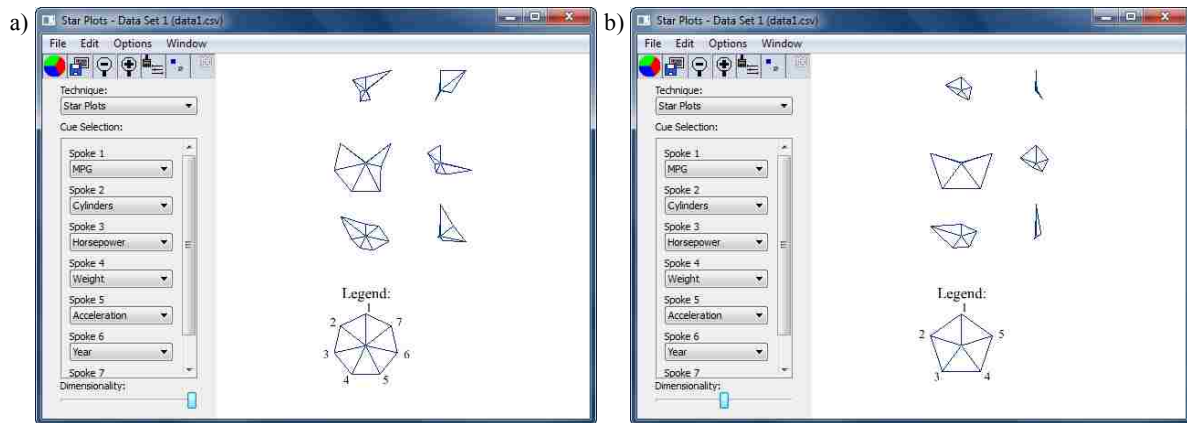
Figure 10. An example of dimensionality reduction using star plots. Image (a) shows the visualization with 7 dimensions. Image (b) shows the same visualization with 5 dimensions.

4.7 Extensibility

Petrichor is extensible through a plugin architecture. The architecture allows users to download a plugin, restart the application, and begin using the new functionality. The programming interface (Section 3.4) provides an API for the plugin implementation to use to access the necessary methods. The user may then access the new functionality through the UI. The UI loads all of the plugins and registers their functionality for use by the user. For example, the functionality provided by a new image exporter plugin is available when the user saves an image of a visualization. A new presentation technique plugin adds another presentation technique button in the main window, allowing the user to render the data using that technique. Functionality such as image exporters, data file loaders, and presentation techniques may be added using plugins.

4.8 Image Support

Image support is handled by the rendering context as explained in Section 3.3.3.

4.9 Labels

All modification of labels is handled by the UI. The technique implementation draws labels using the appropriate method for doing so. The method accepts an id that is used to identify the label to the rendering context. Labels are treated as special types of text annotations. When the user clicks inside of the label bounds using the text annotation tool, the label's text can be changed. The label can be moved and its properties can be changed using the arrow tool in the same fashion as normal annotations, as described in Section 4.2. Once the label's properties have been changed by the user, the rendering context ignores the values specified by the presentation technique unless the user resets the labels. This allows the technique implementation to provide a default label, yet it also allows the user to change the label's value, if desired. Figure 11 shows an example of the default label that has been changed to a user specified value.
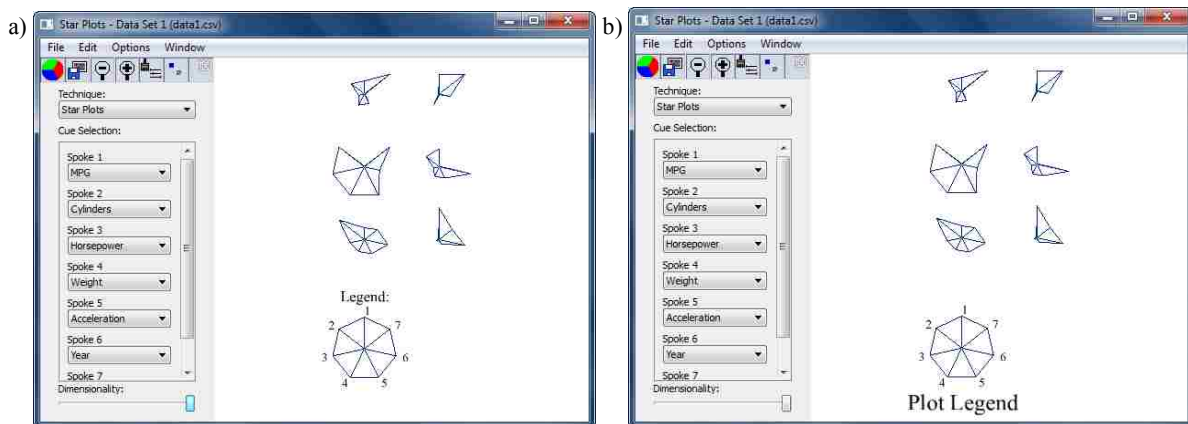
Figure 11. An example of label editing. (a) shows the visualization with the default labels. (b) shows the same visualization with the 'Legend' label's size, position, and text changed.

4.10 Linking

Linking in Petrichor is handled using the model, as described in Section 3.2. When the user performs certain actions, such as selecting points, the UI notifies the model of changes, and the model notifies all of the corresponding visualization windows. The visualization windows

then refresh their rendering to reflect the changes. Figure 12 shows two visualization windows that share the same model. The points that are selected in the first window are selected in the second window through linking.
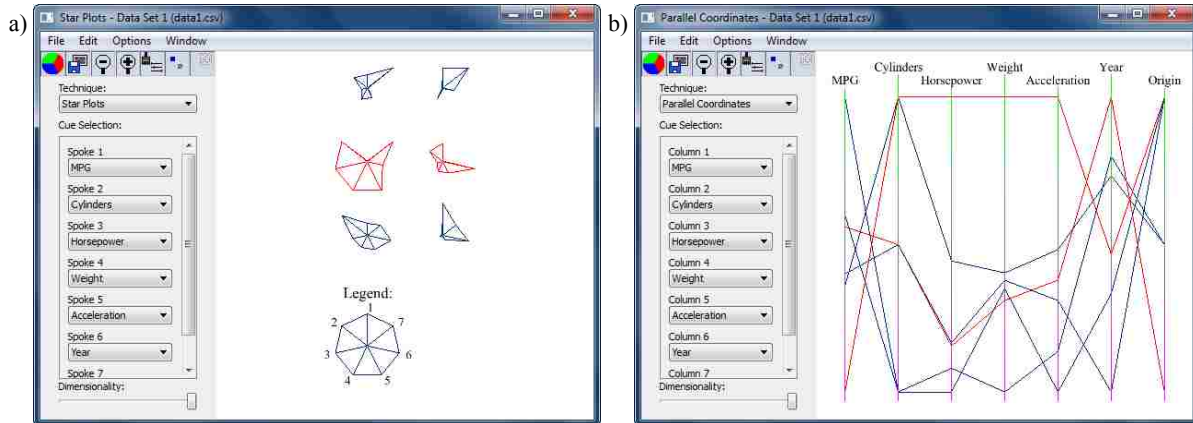

Figure 12. An example of linking. The selected points in the first window, shown in red, are selected in the second window through linking.

4.11 Multiple Data Sets

Multiple data sets may be opened using the open button on the main window, shown in Figure 5(a). Each data set corresponds to a model instance, and all views for that data set are linked to the same model.

4.12 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 5(a). A new visualization window opens, and the corresponding presentation technique plugin is used for rendering. Interaction with the presentation technique occurs as previously explained.

39

4.13 Panning

Panning is supported in two different ways. The tool palette in Figure 5(b) has a hand tool that allows the user to 'grab' the visualization and move it using the mouse. The rendering context also allows the visualization to be moved using keys on the keyboard.

4.14 Point Display

The UI handles displaying point values. When the mouse hovers over a data point, a tool-tip appears and displays the data point's variable values. The data point under the mouse is found using the algorithm described in Section 3.3.1.

4.15 Rotation

A rotation tool is provided in the tool palette, shown in Figure 5(b). The rotation tool allows the user to rotate the view about three different axes. The 'control' (ctrl) key is used to toggle the axis about which the visualization is rotated. The rotation tool adjusts the rendering context's virtual camera, changing how the visualization is drawn.

4.16 Saving State/Sharing

The user can save the state of all of the visualizations. When the save button on the main window is clicked, a dialog box, Figure 13, provides 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. After the options are selected, the information, including selection of points, coloring of points, view transformations, etc., are saved in the XML format described in Section

40

3.1.3. The file may be shared with others. When the file is opened later, all visualizations appear in the exact state that they were saved in.
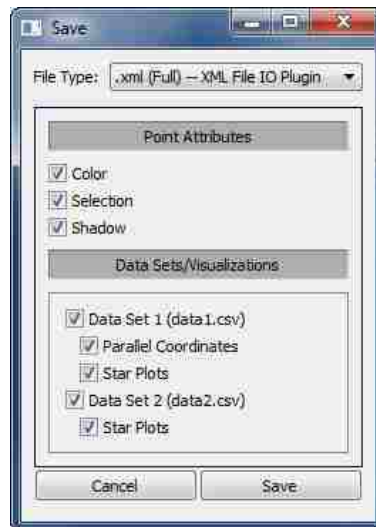

Figure 13. Save dialog box. The user may specify what is saved to the file using the options provided.

4.17 Shadowing

Shadowing is 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 5(c). The view informs the model that the selected points are 'shadowed'. The model then informs all corresponding views of the change. The views may choose to ignore the 'shadowed' setting.

4.18 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 5(c). This is accomplished by creating a new model that contains only the selected points. The view's model is switched to the new model, and the visualization continues as normal.

4.19 Technique Changing

Technique selection is provided through a drop-down menu, shown in Figure 5(c). The UI provides the drop-down menu and populates it with the available presentation techniques. When the user changes techniques, the rendering is refreshed to reflect the change.

4.20 Tool Tips

The UI provides tool tips for all of the tools and buttons. When the user hovers over a tool or button, a tool tip appears with a short amount of text to describe the functionality or name of the item. The user is able to learn what different tools and buttons do more easily, thereby reducing the time needed to learn the system.

4.21 Variable Changing/Cue Selection

Cue selection is provided through drop-down menus placed next to the visualization area, as shown in Figure 5(c). The UI provides the drop-down menus and populates them with the appropriate information from the presentation technique. When the user changes one of the visual cues, the UI notifies the presentation technique that a change occurred. The technique implementation refreshes the rendering, which reflects the change. Figure 14 shows the same presentation technique visualization with two different variable assignments.
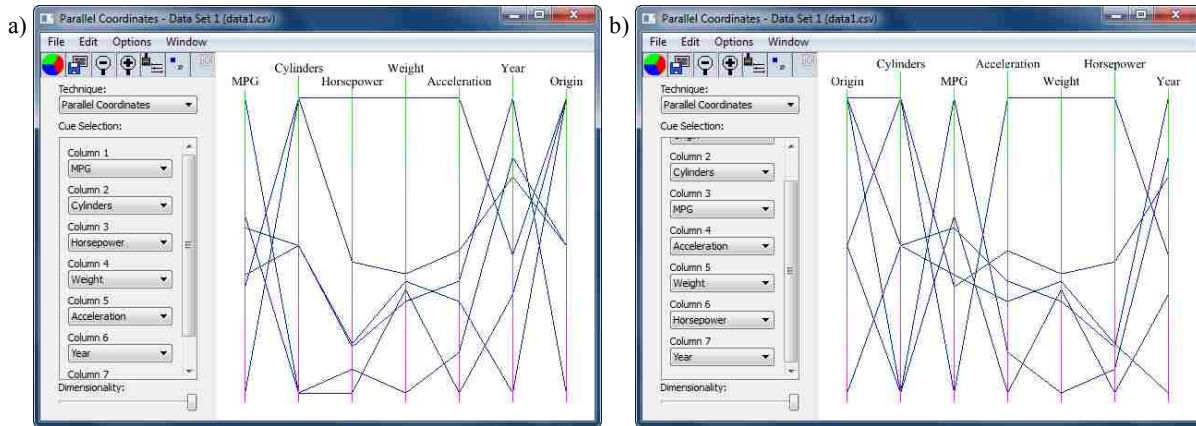
Figure 14. An example of cue selection. The two images show the same presentation technique with two different variable assignments.

4.22 Zoom

The zoom tool provided in the tool palette, shown in Figure 5(b), allows the user to selectively zoom in on the data. If the user clicks on the visualization without dragging, the view zooms in a default amount, centering on the point where the user clicked. If the user drags the mouse, a rectangle appears that shows how much to zoom in. The view is scaled according to the size of the rectangle. If the user right-clicks, the view zooms out a default amount. The user may also change the zoom for the visualization using the buttons provided at the top of each visualization window. All zoom functionality is implemented using the rendering context as described in Section 3.3.2.

43

Chapter 5

Results/Validation

Petrichor overcomes the deficiencies of existing hyperdimensional graphics packages using a design which includes many desirable features that can help the user visualize, explore, and share n-D data. Petrichor allows future expansion and is capable of easily accommodating hyperdimensional presentation techniques through its extensible plugin system. Sharing of visualizations and insights gained is enhanced through annotations, which allow the user to draw directly on the visualization. To demonstrate the usefulness of the design, Petrichor has been used to visualize a test data set.

For a good demonstration, an appropriate data set has been chosen. The data set is titled "Top 100 Private Colleges 2003" [20]. This data set is appropriate because it has many variables and a sufficient number of data points to show the benefits of some of Petrichor's features. For example, with many variables, dimensionality reduction can be useful in limiting what is rendered.

5.1 Data Set Visualization

One of the purposes of data visualization is to gain useful insights from data that is otherwise not easily discernible. Due to this, people often use visualization packages to answer questions related to the data. Since the data set chosen for this demonstration involves colleges, the questions to be answered might include "In which college are students most likely to succeed?" and "Which college is the best dollar value?" The search for answers to these questions will be demonstrated by visualizing the data using Petrichor.

5.1.1 In Which College Are Students Most Likely to Succeed?

The questions have been left vague and ambiguous intentionally. Generally, questions that are clear and precise are easy to answer by looking at the raw data. For example, the question "Which college costs the least?" is easily answered by looking at a column of data. To answer the question "In which college are students most likely to succeed?", we require more exploration of the data. First, we need to decide what might constitute success for a student attending college. A high graduation rate can indicate that the students were able to persevere through all of the classes they had to take. Alternatively, it might also mean that the college classes were easier to pass. So, looking at just one column of data might not be enough to answer the question. Let's look at the data and see if there are other ways of evaluating success. Figure 15 shows the data set rendered using Parallel Coordinates.
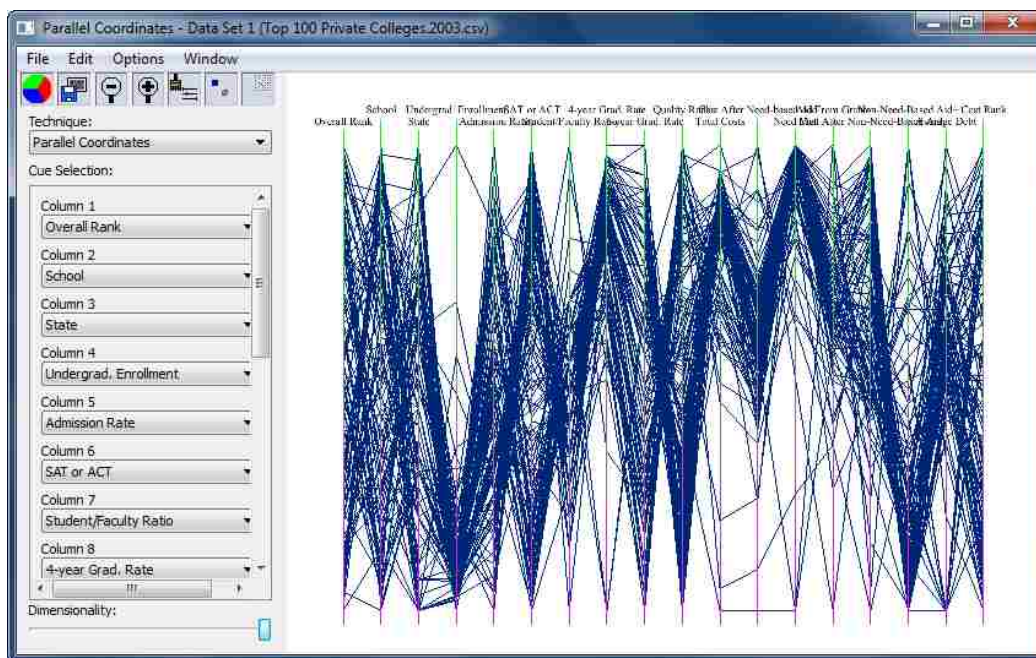

Figure 15. Initial view of the data set using parallel coordinates.

On first impression, the visualization appears cluttered. A lot of data is shown, and it is hard to see how the variables relate to each other. Using the dimensionality slider, we can reduce the number of dimensions shown to reduce the clutter resulting from so many variables being shown. After changing the number of dimensions shown, we will need to choose which variables will still be shown in the visualization. Since we are interested in factors relating to the success of students, we will choose the variables "Overall Rank", "4-year Grad. Rate", "6-year Grad. Rate", "Quality Rank", "Total Costs", "Need Met", "Average Debt", and "Cost Rank". Figure 16 shows the updated visualization.
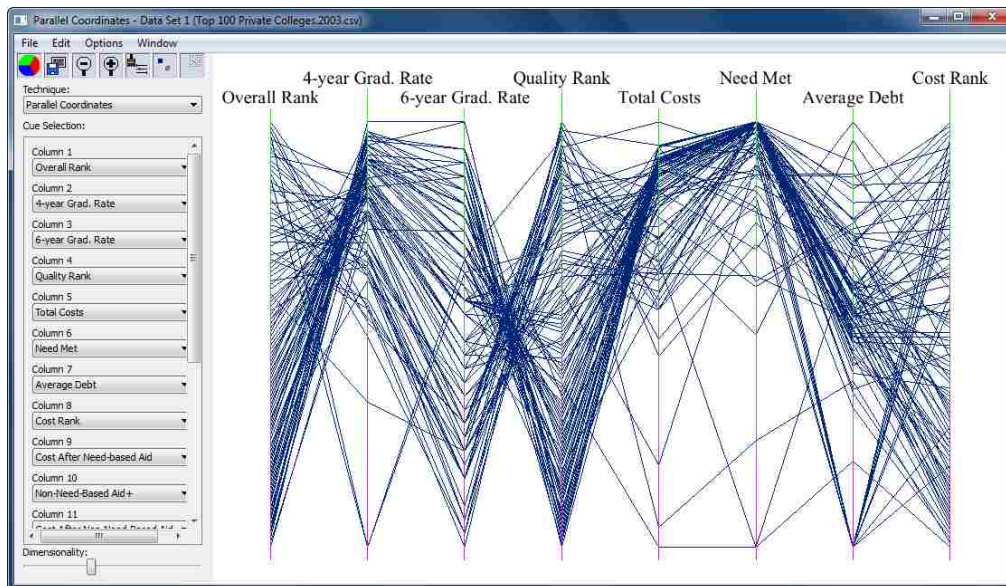

Figure 16. Updated visualization. The number of dimensions has been reduced, and the desired variables have been selected.

It is now easier to see the data. There are a few things to note about this particular display. First, the values have been normalized using the minimum and maximum values for each variable. This means that values in adjacent axes may not line up numerically. Second, there seems to be an inverse relationship between "Overall Rank" and "4-year Grad. Rate",

which is counter-intuitive. This inverse relationship occurs because ranking starts with 1 as the best with larger numbers being less good. The graduation rate starts with 100% as the best and goes down toward 0%. A similar relationship can be seen with the other variables dealing with ranking. With the Parallel Coordinates implementation in Petrichor, we can change the ordering for the variable by clicking on the appropriate axis using the Presentation Technique Tool. The result of flipping the rank variables is show in Figure 17.
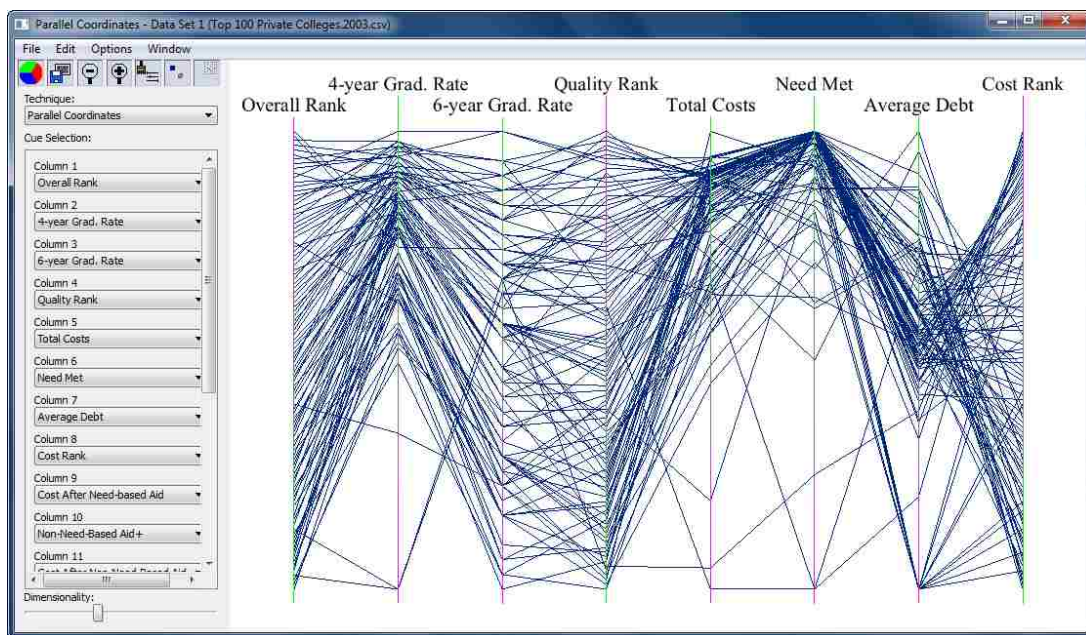

Figure 17. The rank axes have been flipped so that the best rank is on top.

The axes have been colored to show which way each axis is ordered. The purple side of each axis represents smaller values, and the green side represents larger values. Note that the purple side is on top for the axes whose orders have been flipped. Now we can look at the relationships in the data. Upon initial glance, we can see some trends in the data. First, there is a direct correlation between the overall rank and the graduation rate. This means that the people who compiled this data used the graduation rate as a measure of how good a college is. We can use the ranks that they have assigned as a starting point for our analysis. So, we'll filter out some

47

of the colleges that ranked lower in this data set and see what the remaining colleges have in common. To do this, we can either select the data points using one of the brushing tools in the tool palette, or we can use the range brush. We will use the range brush to filter the data points, shown in Figure 18. While we can see which data points are selected, the other points still add a bit of clutter. So we'll create a subset of the original data based on the data points we selected using the range brush, shown in Figure 19.
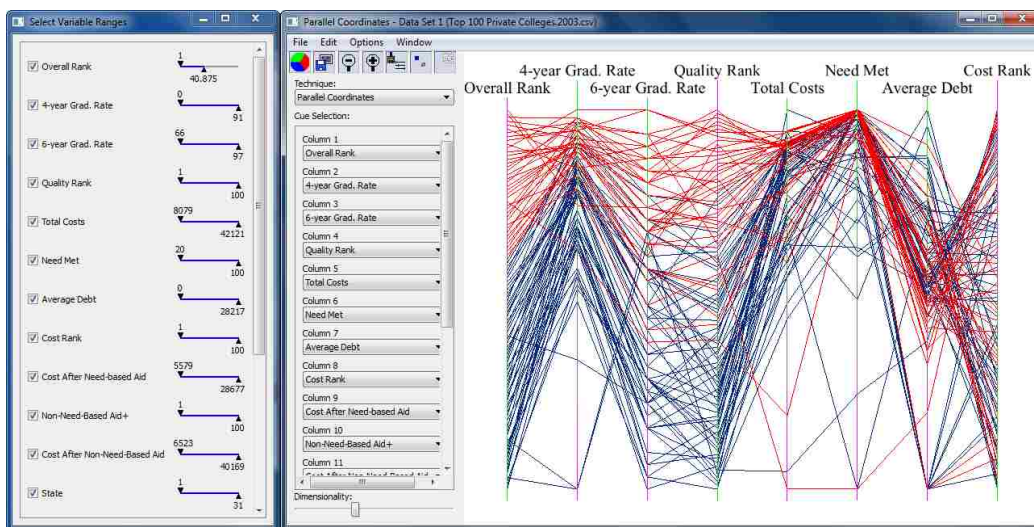

Figure 18. The range brush has been used to select the top 40 colleges based on overall rank.
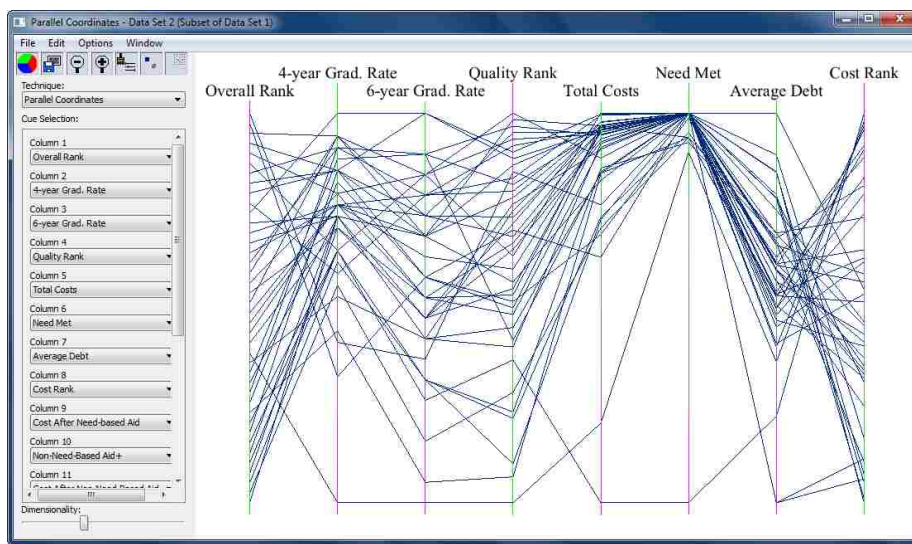

Figure 19. A new subset of the original data set.

Now that we have isolated the data points of interest, we can continue to analyze the remaining data points. Let's say that our measure of success is related to the 4-year graduation rate and the average debt. We want a high graduation rate while keeping the average debt to a minimum. We can use the range brush again to limit our choices based on these criteria. Once we have it narrowed to a few choices, we'll use the color options dialogue to change the color of the selected points to a purple color, shown in Figure 20. Based on our data exploration, we now have a few colleges that might rank highly in terms of student success.
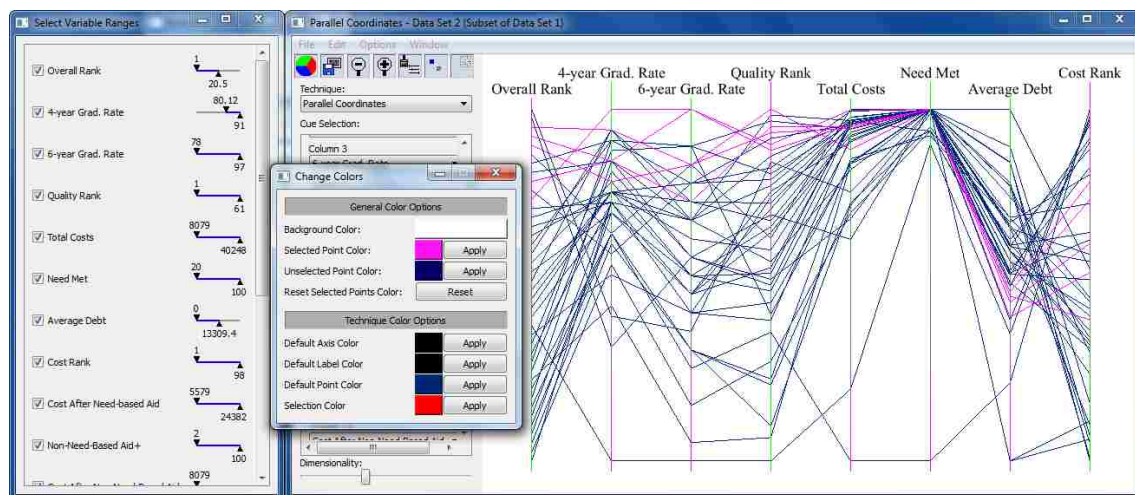

Figure 20. Some selected points have been colored to show the narrower selection.

5.1.2 Which college is the best dollar value?

Finding the best dollar value depends on what is considered valuable. Looking at the available variables, let's look at the variables "Overall Rank", "4-year Grad. Rate", "Quality Rank", "Total Costs", "Average Debt", "Aid from Grants", and "Student/Faculty Ratio". Figure 21 show the result of limiting the visualization to those fields.
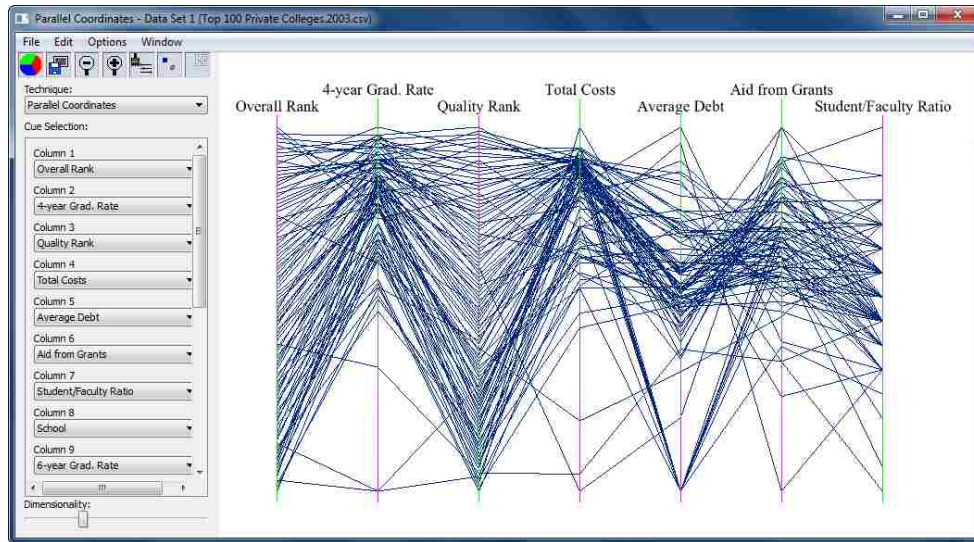
Figure 21. Initial visualization rendering.

As with the previous question, we will have to make some assumptions to answer the current question. Let's assume that it's desirable to have a relatively low student to faculty ratio, lower average debt, higher aid from grants, and a higher graduation rate. We can use the range brush to find some good candidates. Figure 22 shows the result of applying the filtering.
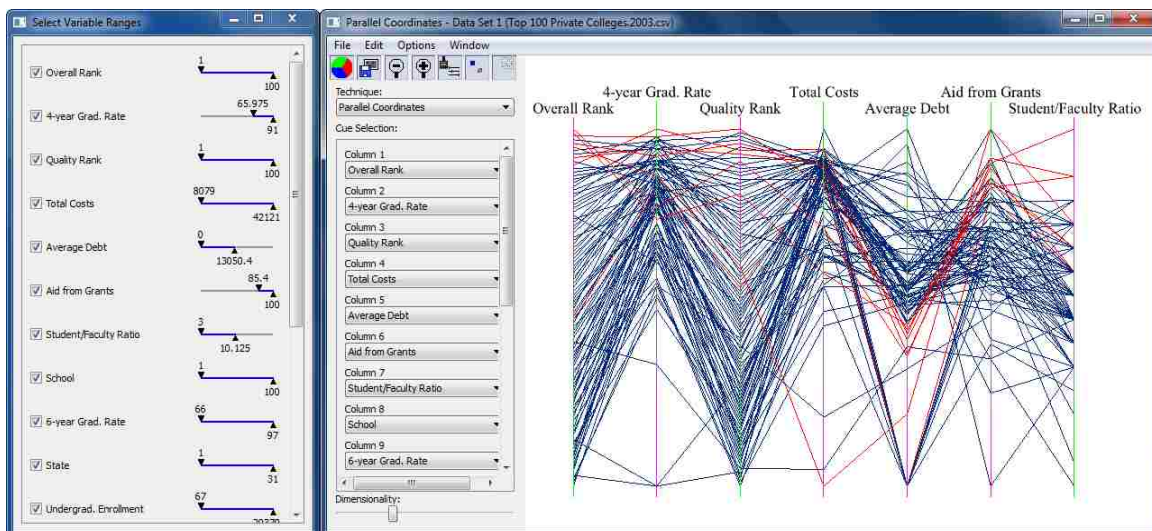

Figure 22. Range brush applied to the data points.

We now have a few choices to consider. Looking at Figure 22, we notice that the majority of the selected data points tend to cluster on the lower end of the median data points for average debt. We artificially created this clustering by limiting the average debt using the range brush. What if we eliminated a good college choice by strictly relying on the cutoff? We can find points that are similar to the ones we have selected by using a different presentation technique. Figure 23 shows the same data rendered using the Star Plots presentation technique.
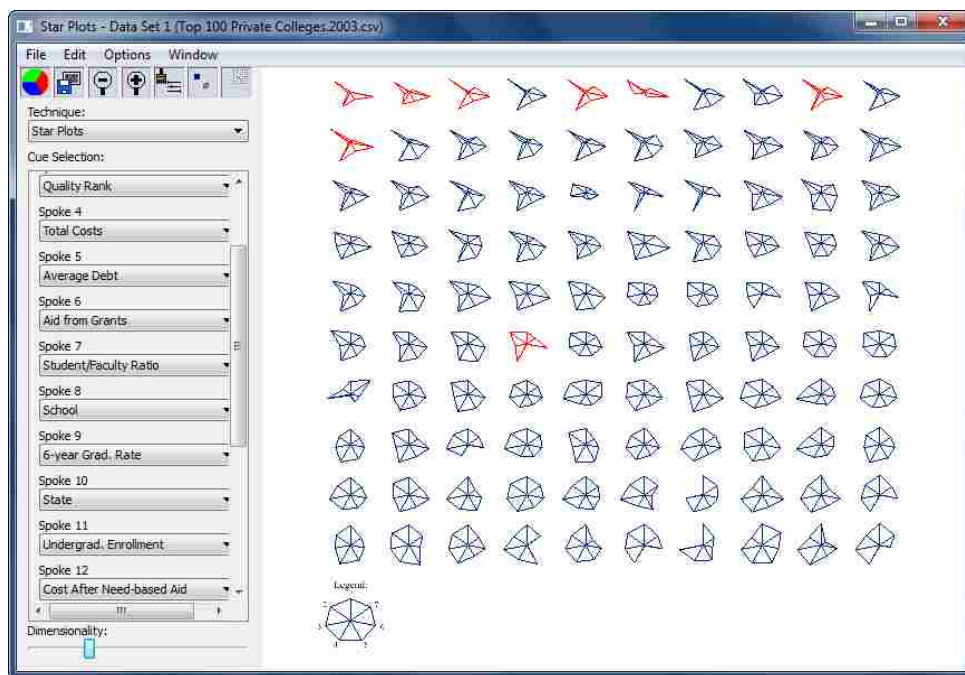

Figure 23. Data points rendered using star plots. The data points selected in Figure 22 are selected here due to linking.

Using the selected points we already have, we can look at the other points and see which ones are similar in shape. Points similar in shape tend to have similar values. Looking at a few of them, we see some we might like to add to the selection. We also notice that we might want to remove some from the selection. We can accomplish this change in selection using the lasso tool. By holding down the control key, we can add to the selection. By holding down the shift key, we

can remove points from the selection. After changing the selection, we can add some annotations to the rendering to take notes on our preferences for the selected colleges. Figure 24 shows the resulting visualization. At this point, we can save the data and view it later, or perhaps send it to a friend.
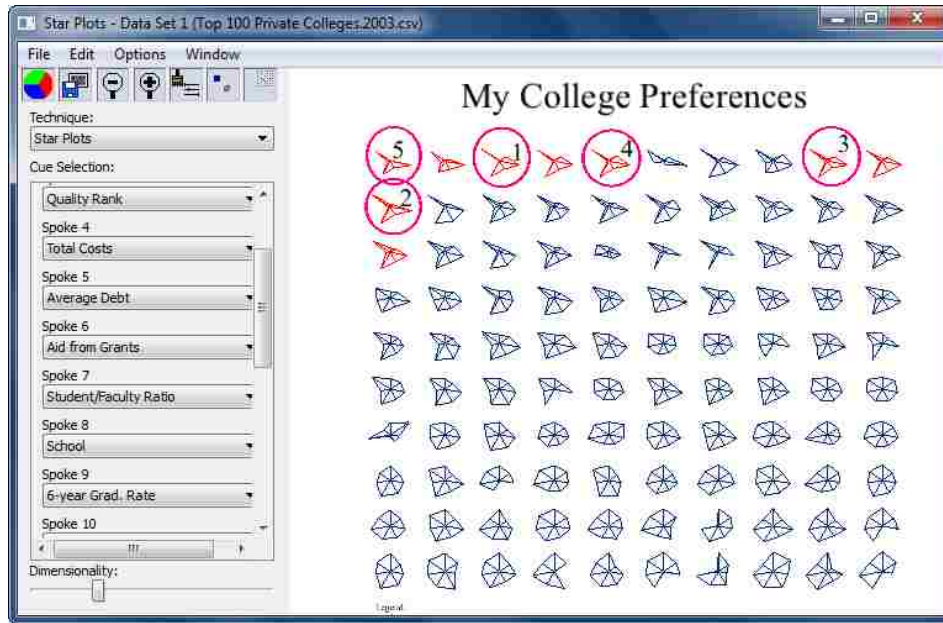

Figure 24. A change in selection of data points. Annotations have been added to show college preference.

5.2 Results

Some of the functionality of Petrichor has been demonstrated using a test data set. Various features of Petrichor helped to facilitate our exploration of the data. Note that the results of the demonstration were directly affected by some of the choices we made along the way. If other choices had been made, the outcome would probably be different. That is one of the benefits of graphing the data: we can try a few different options and see what works best. In exploring the data, we can get a feel for the answers to our questions. No other package on the market could fully facilitate this demonstration.

Chapter 6

Conclusion

This work has focused on solving the problems a person faces when trying to visualize n-D data. I have identified features a graphics package should support to adequately solve those problems. Those features have been incorporated into the design for a new hyperdimensional graphics package named Petrichor, whose ability to facilitate exploration of, interaction with, and sharing of n-D data has been demonstrated using a test data set.

In implementing Petrichor, I gained insights into the process of visualizing n-D data. First, there are several presentation techniques that have been developed to visualize n-D data. More n-D presentation techniques will be developed in the future. It is infeasible to support all available techniques in a package without investing a lot of energy. The best way to overcome this is to make it easy for other programmers to add to the supported techniques. Petrichor has been written to accomplish this. Petrichor makes the burden of programming lighter by handling the functionality that is the same across techniques. Second, there are many existing, disjoint ideas that can be combined to create a new, more useful package. For example, not many packages support annotations, despite the idea having been thoughtfully researched and reported [18]. Third, it is difficult to keep a UI design clean and simple while still providing the functionality needed. It takes a lot of thought to keep things organized in a way the user will find useful. Finally, existing packages can always be improved upon. The packages shown in Table 3 have been researched and built upon to improve the user's experience while exploring n-D data. Table 4 shows the result of Petrichor's design.

Package Name	Annotations: Text	Annotations: Highlighter/Pen	Annotations: Shapes	Brushing: Brush Tool	Brushing: Range Brush	Brushing: Change Selection	Coloring: Point Color	Coloring: Background Color	Consistent Tools: Annotations	Consistent Tools: Panning	Consistent Tools: Rotations	Consistent Tools: Zoom	Dimensionality Reduction	Extensibility	Image Support	Labels: Default Labels	Labels: Label Editing	Linking	Multiple Data Sets	Multiple Views	Panning	Point Display	Rotation	Saving State/Sharing	Shadowing	Subsetting	Technique Changing	Tool Tips	Variable Changing	Zoom
GGobi [23,53]				S	S		S	P					S	S		S		S	S	S	S	P	P	S	P			S	S	S
Tulip [9]				S		S	S		S	S			S	S	S	S		S	S	S	S	S	S	P				S	P	S
XmdvTool [63,64]					P		P	P	S		S	S	S	S		S		S	S	S	S	P					S	S	P	S
Petrichor	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S

S = Fully Supported   P = Partially Supported

Table 4 – Selected hyperdimensional graphics packages with their supported features, including a new package called Petrichor. The values are marked as accurately as possible, but due to the nature of user interfaces (features can be easily hidden), some values may not be marked correctly.

6.1 Contributions

This work has sufficiently solved the previously mentioned problems a person faces when trying to visualize n-D data. I have identified features a graphics package should support to solve those problems adequately. Those features have been implemented in Petrichor. Through the use of an extensible plugin system, presentation techniques can be easily added and accommodated. Consistent tools simplify the UI and the user's experience with the system and the data. Annotations help users to share notes and insights into the data with others. All of these features combine into a powerful, n-dimensional visualization package.

6.2 Future Work

While Petrichor supports all of the desirable features listed in this paper, there may be features that would further benefit the user. As new techniques are developed, new user interactions may be needed to fully immerse the user in the data. User testing may reveal these

potential improvements which ultimately may help to further enhance and/or streamline the UI. Another avenue of research may be to investigate how to add more user interaction to otherwise static presentation techniques. For example, in Petrichor's implementation of Parallel Coordinates, the user can click on the axes using the Presentation Technique Tool to change the ordering of each axis. More interactions of this type can be added to presentation technique plugins to increase the user's ability to interact with each technique. A further area of research is adding plugins that utilize animations in their visualization. Petrichor provides an interface that allows the presentation technique plugin to initiate time-based renderings, which allows animations to be drawn. However, a plugin that uses this feature has yet to be written and explored. Another area of research may involve intelligent cue selection, especially while changing the number of dimensions rendered. A final area of research is exploring ways to enhance the user interface. One example is the range brush tool. Currently, this tool is used to specify ranges for selection of data points based on their values for each variable. An enhancement to this tool would be to render a histogram above each range showing the distribution of the data points for each variable.

References

[1] ADVIZOR Solutions®. ADVIZOR®: A Technical Overview.

http://www.advizorsolutions.com/ADVIZOR%20Technical%20Overview%205.4.pdf, 2012.

Accessed May, 2012.

[2] ADVIZOR Solutions®. Data Visualization Gallery | ADVIZOR Solutions.

http://www.advizorsolutions.com/gallery/, 2012. Accessed May, 2012.

[3] ADVIZOR Solutions®. Desktop Business Analysis Software | ADVIZOR Solutions.

http://www.advizorsolutions.com/products/, 2012. Accessed May, 2012.

[4] Alpern, B. and Carter, L. The Hyperbox. In *Proceedings of the 2nd Conference on Visualization*, pages 133-139, 1991.

[5] Anderson, E. A Semigraphical Method for the Analysis of Complex Problems. *Technometrics*, Vol. 2(3), pages 387-391, 1960.

[6] Andrews, D. Plots of High-Dimensional Data. *Biometrics*, Vol. 28(1), pages 125-136, 1972.

[7] Ankerst M., Keim A., and Kriegel H. Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets. In *Proceedings of the 7th IEEE Visualization Conference*, Hot Topic Session, San Francisco, CA, 1996.

[8] Asimov, D. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, Vol. 6(1), pages 128-143, 1985.

[9] Auber, D. and Mary, P. Data Visualization Software | Tulip. http://tulip.labri.fr/TulipDrupal/, 2012. Accessed May, 2012.

[10] AVS. AVS/Gsharp Data Visualization Software. http://www.avs.com/products/gsharp/index.html, 2004. Accessed May, 2012.

[11] Burton, R. Algorithms and Display Techniques for Four-Dimensional Cartesian Graphics. In *Proceedings of the SPSE International Symposium and Exposition on Electronic Imaging Devices and Systems*, pages 95-102, 1988.

[12] Centrifuge Systems, Inc. Data Visualization - Data Analysis and Data Integration Software by Centrifuge Systems. http://www.centrifugesystems.com/index.php, 2012. Accessed May, 2012.

[13] Chambers, J., Cleveland, W., Kleiner, B., and Tukey, P. *Graphical Methods for Data Analysis*, pages 157-163. Wadsworth International Group, Belmont, California, and Duxbury Press, Boston, Massachusetts, 1983.

[14] Chernoff, H. The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association,* Vol. 68(342), pages 361-368, 1973.

[15] Cooprider, N. and Burton, R. Extension of Star Coordinates into Three Dimensions. In *Proceedings of the SPIE Conference on Visualization and Data Analysis*, Vol. 6495, 64950Q, 2007.

[16] Cui, Q., Ward, M., and Rundensteiner, E. Enhancing Scatterplot Matrices for Data with Ordering or Spatial Attributes. In *Proceedings of the SPIE Conference on Visualization and Data Analysis*, Vol. 6060, pages 248-258, 2006.

[17] Data Description, Inc. Data Desk: Statistical Analysis Software, Exploratory Data Analysis, Data Visualization, Data Exploration.
http://www.datadesk.com/products/data_analysis/datadesk/, 2012. Accessed May, 2012.

[18] Denisovich, I. Software Support for Annotation of Visualized Data Using Hand-Drawn Marks. In *Proceedings of the Ninth International Conference on Information Visualisation*, pages 807-813, 2005.

[19] Designing Analytics. GroupVisual.io. http://designinganalytics.com/, 2012. Accessed May, 2012.

[20] Drexel University. Data Sets.
http://mathforum.org/workshops/sum96/data.collections/datalibrary/data.set6.html, 2008. Accessed May, 2012.

[21] Feiner, S. and Beshers, C. Worlds Within Worlds: Metaphors for Exploring N-Dimensional Virtual Worlds. In *Proceedings of the 3rd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 76-83, 1990.

[22] Friedman, J. and Tukey, J. A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers*, Vol. C-23(9), pages 881-890, 1974.

[23] GGobi Team, The. Downloads. GGobi Data Visualization System. http://www.ggobi.org/downloads/#windows, 2012. Accessed May, 2012.

[24] GIMP Team, The. GIMP - The GNU Image Manipulation Program. http://www.gimp.org/, 2012. Accessed May, 2012.

[25] Grinstein, G., Trutschl, M. and Cvek, U. High-Dimensional Visualizations. In *Proceedings of the Workshop on Visual Data Mining, ACM Conference on Knowledge Discovery and Data Mining*, pages 1-14, 2001.

[26] Hartigan, J. Printer Graphics for Clustering. *Journal of Statistical Computation and Simulation*, Vol. 4(3), pages 187-213, 1975.

[27] Hartigan, J. and Kleiner, B. Mosaics for Contingency Tables. *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, pages 268-273. Springer-Verlag, New York, New York, 1981.

[28] Hillery, B. and Burton, R. Multivariate Data Visualization Via Outdoor Scenes. In *Proceedings of the SPIE Conference on Visualization and Data Analysis*, Vol. 7868, 78680X, 2011.

[29] Hoffman, P. Table Visualizations: A Formal Model and Its Applications. Doctoral Dissertation. University of Massachusetts, Lowell, Massachusetts, 1999.

[30] Hoffman, P., Grinstein, G., Marx, K., Grosse, I. and Stanley, E. DNA Visual and Analytic Data Mining. In *Proceedings of the 8th Visualization Conference*, pages 437-441, 1997.

[31] Hoffman, P., Grinstein, G. and Pinkney, D. Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM International Conference on Information and Knowledge Management*, pages 9-16, 1999.

[32] InetSoft Technology Corp. Style Intelligence: Business Intelligence Software | InetSoft Technology. http://www.inetsoft.com/products/StyleIntelligence/, 2012. Accessed May, 2012.

[33] Inselberg, A. The Plane with Parallel Coordinates. *The Visual Computer*, Vol. 1(2), pages 69-91, 1985.

[34] Janina Ratkowska Group, The. Antaeus • Data from the Ground Up. http://www.antaeus-data.com/, 2012. Accessed May, 2012.

[35] Kandogan, E. Visualizing Multi-Dimensional Clusters, Trends, and Outliers using Star Coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pages 107-116, 2001.

[36] KDnuggets. Parallax Software for Multidimensional Visualization and Automatic Classification. http://www.kdnuggets.com/software/parallax.html, 2012. Accessed May, 2012.

[37] Keim, D. Pixel-Oriented Visualization Techniques for Exploring Very Large Data Bases. *Journal of Computational and Graphical Statistics*, Vol. 5(1), pages 58-77, 1996.

[38] Kleiner, B. and Hartigan, J. Representing Points in Many Dimensions by Trees and Castles. *Journal of the American Statistical Association*, Vol. 76(374), pages 260-269, 1981.

[39] Lawrence Livermore National Laboratory. VisIt Visualization Tool. https://wci.llnl.gov/codes/visit/home.html, 2012. Accessed May, 2012.

[40] LeBlanc, J., Ward, M., and Wittels, N. Exploring N-Dimensional Databases. In *Proceedings of the 1st Conference on Visualization*, pages 230-237, 1990.

[41] Melville, J. and Burton, R. Piles for Hyperdimensional Graphics. *Computers & Graphics*, Vol. 21(1), pages 51-60, 1997.

[42] Miner3D, Inc. Miner3D - Visual Data Analysis Software for Business Intelligence and Science. http://www.miner3d.com/, 2012. Accessed May, 2012.

[43] Minnotte M. and West R. The Data Image: A Tool for Exploring High Dimensional Data Sets. In *Proceedings of the ASA Section on Statistical Graphics*, pages 25-33, 1998.

[44] NovoSpark Corporation. NovoSpark. http://www.novospark.com/, 2012. Accessed May, 2012.

[45] Partek Incorporated. Partek Incorporated | Next Generation Sequencing & Microarray Software. http://www.partek.com/, 2012. Accessed May, 2012.

[46] Patro, A. Pixel Oriented Visualization in XmdvTool. Masters Thesis, Worcester Polytechnic Institute, Worcester, Massachusetts, 2004.

[47] Pickett, R. and Grinstein, G. Iconographic Displays for Visualizing Multidimensional Data. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Vol. 1, pages 514-519, 1988.

[48] Rao, R. and Card, S. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence*, pages 318-322, 1994.

[49] Reactive Search S.r.l. Grapheur - The Data Mining and Interactive Visualization Tool. http://www.grapheur.com/, 2012. Accessed May, 2012.

[50] Sammon, Jr., J. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, Vol. C-18(5), pages 401-409, 1969.

[51] ScienceGL, Inc. Advanced 3D Graphics for 3D, 4D and Multiple Dimension Data Sets. Visualization Solutions for Industry, Business, Healthcare and Education. http://www.sciencegl.com/index.html, 2012. Accessed May, 2012.

[52] Stottler Henke, Inc. DataMontage Visualization of Complex, Time-Oriented Data. http://www.stottlerhenke.com/datamontage/, 2012. Accessed May, 2012.

[53] Swayne, D., Lang, D., Buja, A., and Cook, D. GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization. *Computational Statistics & Data Analysis*, Vol. 43(4), pages 423-444, 2003.

[54] Tableau Software. Fast Analytics and Rapid-Fire Business Intelligence from Tableau Software | Tableau Software. http://www.tableausoftware.com/, 2012. Accessed May, 2012.

[55] Tan, X. and Burton, R. Parallel Planes Coordinates for Hyperdimensional Computer Graphics. *Journal of Imaging Science and Technology*, Vol. 17(6), pages 591-595, 1992.

[56] Theus, M. Mondrian - Interactive Statistical Data Visualization in Java. http://rosuda.org/mondrian/, 2011. Accessed May, 2012.

[57] ThinkAnalytics Ltd. ThinkAnalytics - Think Enterprise Data Miner. http://www.thinkanalytics.com/productServices/edm/index.htm, 2012. Accessed May, 2012.

[58] TIBCO Software Inc. Business Intelligence Analytics Software & Data Visualization | TIBCO Spotfire. http://spotfire.tibco.com/default.aspx, 2011. Accessed May, 2012.

[59] Tukey, P. and Tukey, J. Graphical Display of Data Sets in 3 or More Dimensions. *Interpreting Multivariate Data*, chapters 10-12. Wiley, Chichester, United Kingdom, 1981.

[60] Visipoint Ltd. VisualData – Visipoint. http://www.visipoint.fi/visualdata.php, 2010. Accessed May, 2012.

[61] Visokio. Omniscope | Visokio. http://www.visokio.com/omniscope, 2012. Accessed May, 2012.

[62] VisuMap Technologies Inc. VisuMap Technologies: Visualizing High Dimensional Complex Data. http://www.visumap.net/, 2010. Accessed May, 2012.

[63] Ward, M. XmdvTool Home Page: Overview. http://davis.wpi.edu/xmdv/, 2010. Accessed May, 2012.

[64] Ward, M. Xmdvtool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of the Conference on Visualization*, pages 326-333, 1994.

[65] Weinstein, J., et al. An Information-Intensive Approach to the Molecular Pharmacology of Cancer. *Science*, Vol. 275(5298), pages 343-349, 1997.

[66] Xerox. Visual Interactive System for Annotation and Machine Learning, Data Mining, Analysis or Clustering. http://visalix.xrce.xerox.com/, 2012. Accessed May, 2012.

[67] Xun, P. and Burton, R. Parallel Volumes Coordinates for Hyperdimensional Computer Graphics. *Journal of Imaging Science and Technology*, Vol. 41(1), pages 59-70, 1997.

[68] Yang, J., Ward, M., and Rundensteiner, E. Interactive Hierarchical Displays: A General Framework for Visualization and Exploration of Large Multivariate Data Sets. *Computers & Graphics*, Vol. 27(2), pages 265-283, 2003.