Visual Analysis of High Dimensional Real Data

Thesis submitted in partial fulfillment
of the requirements for the degree of

MASTERS OF SCIENCE BY RESEARCH
in
COMPUTER SCIENCE

by

NAHIL JAIN
200702023
nahil.jain@research.iiit.ac.in

CENTER FOR DATA ENGINEERING
International Institute of Information Technology
Hyderabad - 500 032, INDIA
MAY 2012

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Visual Analysis of High Dimensional Real Data" by Nahil Jain, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____

Date

Adviser: _____

Dr. Soujanya Lanka

_____

Dr. Kamalakar Karlapalem

Dedicated to my parents Mr. Ashok Jain and Mrs. Snehalata Jain & my siblings Anshul and Eashita

Acknowledgments

I would like to express my deep and sincere gratitude to my advisers, Dr. Soujanya Vadapalli and Dr. Kamalakar Karlapalem for their continuous support of my M.S. study and research, for their patience, motivation, enthusiasm, and immense knowledge. I am grateful to Dr. Kamalakar Karlapalem and CDE for funding my postgraduate education.

I would like to thank my lab mates Aditya, Shubhankar, Shashank, Gaurav, Prapula and Shubhangi for their terrific company and valuable discussions during my study. I also take this opportunity to thank my close friends Sachin, Rakshit and Rohit for their constant encouragement. I would specially like to thank Saket, Harshit, Raghvendra and Aditya who were always ready for discussion on my research.

I owe loving thanks to my mother Mrs. Snehalata Jain and my father Mr. Ashok Jain for their endless love, encouragement, support and understanding. I am also thankful to my parents for teaching me the value of knowledge and education.

Finally I would like to thank God for giving me the strength to complete this work.

Abstract

High Dimensional Real Data Visualization

Visual representation of data is called Data Visualization. The aim of data visualization is to provide viewers an understanding of the dataset. In high dimensional numerical data visualization, data is mapped from numerical form to visual objects. The simple line graph or scatter plot[11] has been used for visualization to understand the interaction of two variables. Over the time, more sophisticated data visualization techniques were developed to visually understand high dimensional datasets.

Cluster Visualization

Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar to each other than to those in other clusters. Visualizing shape and structure of data points within a cluster helps in understanding nature of the cluster (group of points which are close to each other). Two dimensional and three dimensional clusters can be visualized directly but visualizing higher dimensional clusters is difficult. Several techniques have been presented to visualize high dimensional data clusters. They include general multidimensional data visualization techniques, icon based representation techniques and interactive cluster exploration tools. Majority of these visualization techniques are based on a point level abstraction (every data point corresponds to one or more visual objects present on canvas). When number of data points increase, these techniques are crippled by over plotting, decline in legibility, inability to plot complete dataset or loss of speed and interaction.

Classification Data Visualization

Data classification consists of using variables with known class to predict class of other variables. Visualization of classification data is similar to cluster visualization. Techniques which are used for cluster visualization can also be used for classification visualization.

In this thesis, we present a tool called PEARLS which uses basic three-dimensional shapes like sphere and cube to visualize high dimensional data clusters. These shapes, called pearls are groups of points which represent a level of abstraction in between data point level and cluster level. We embed various interactive techniques like data dimension, swiss cheese view and attribute filtering to make the

cluster analysis informative and intuitive. Other tools like Parallel coordinates and scatter plot matrix are embedded for a detailed analysis of individual pearls. We demonstrate the use of PEARLS through case studies on a Singapore real estate data set and Baseball players data set. Our evaluation shows the benefits of PEARLS in cluster analysis and concept identification within clusters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Extracting meaningful information from large quantities of data is a difficult task. Effective visual representation of data makes this task easier. Visual representation of data is called Data Visualization. Aim of data visualization is to help users detect expected as well as unexpected from data and gain meaningful insights into data. Many techniques, domain dependent as well as domain independent have been developed for multi dimensional data visualization. Some of the well known techniques are parallel coordinates [22], scatter plot matrices [11] and RadViz [35].

Different data types require different visualization techniques. Multivariate datasets occur very often in domains ranging from science to finance. Data points in such datasets either already have classes or they are clustered or classified to extract meaningful information. Understanding these clusters or classes is a key component in the puzzle of understanding whole dataset. In a multi-dimensional real data cluster, data points have shape (structure of the cluster) and size (spread of the cluster). Visualizing shape and structure of these data points helps in understanding clusters. Understanding clusters helps in performing further data analysis and comparing one cluster with another cluster. A two dimensional or three dimensional data cluster can be visualized and shape and size of clusters preserved accurately. Feiner and Beshers [14] observe that human experience with spatial positioning is limited to 3-D. This leads to difficulty in visualizing data clusters with more than three dimensions.

Well known data visualization techniques like parallel coordinates [22], scatter plot matrices [11] and RadViz [35] have been used to visually explain the results of clustering algorithms and data clusters. Most of these techniques focus on a point level abstraction. Every point corresponds to one or more objects on the visual canvas. The point level abstraction suffers from the limitations when number of points becomes too large. Andrienko et. al [1] and others have identified several problems like over plotting, decline in legibility, inability to plot full dataset or loss of speed and interaction which lead to a decline in efficacy of techniques based on point level abstraction. Moreover, large number of visual

objects overwhelm user. Figure 1.1 shows parallel coordinate visualization of 5000 data points. In this visualization, problems like over plotting and decline in legibility are evident.

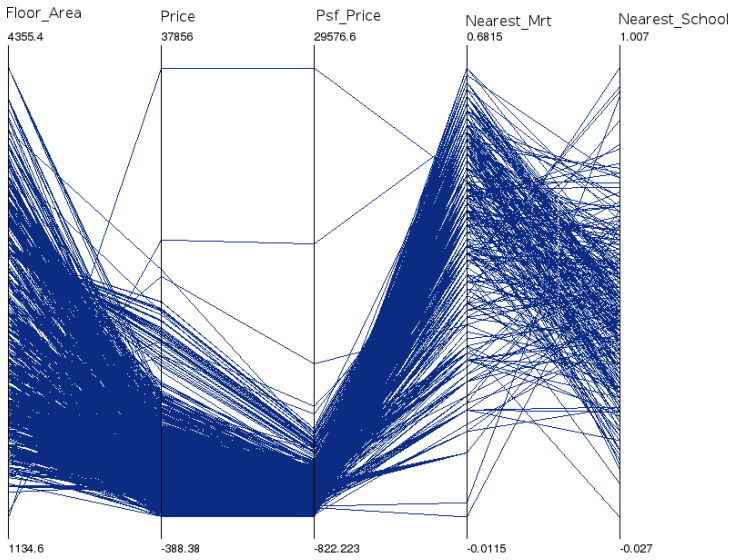


Figure 1.1: This image sequence shows a parallel coordinate plot for 5000 records of Singapore Real estate dataset. Problems of over plotting and decline in legibility are evident.

In this thesis, we propose PEARLS, a visualization technique that helps user understand clusters, by visualizing their shape and size using primitive easy to understand shapes like sphere, cube and rhombus. These shapes not only describe shape and size of data cluster, but they also represent concepts. As stated in Han and Kamber [17], *"A concept usually refers to a collection of data such as frequent_buyers, graduate_students, and so on and concept description generates descriptions for characterization and comparison of data."*. In PEARLS, concepts can be thought of as an abstract description of the cluster and a potentially closest such description.

Describing a data cluster and its points using an intermediate level of abstraction scales up when number of data points increase. Such a visualization also enables a user to search for groups of points which belong to a predefined concept and explore various intra point relationships in various subsets of dimensions by constantly refining how the cluster is divided into concepts.

1.1 Contributions of Thesis

The contributions of this thesis are :

- Pearl three dimensional visualization technique for high dimensional data.

- Intuitive user interactions techniques like Swiss-cheese view, filtering of dimensions, data dimension and Point Search technique.
- Visualization of clustering and classification results using PEARLS.
- Demonstration of how PEARLS can be used in concept description and discrimination, in searching for point groups representing various concepts and in exploring intra point relationships between points in various subsets of dimension. Concept description and discrimination are elaborated in in real life data experiments in Chapter 5.
- A PEARLS visualization tool kit built in Qt and OpenGL integrated with Xmdv toolkit to visualize parallel coordinates, scatter plot, star glyph, dimension stacking analysis and k-reverse nearest neighbor analysis on individual pearls.

1.2 Motivational Example

We illustrate how search results from a real-estate property web-site could be visualized using PEARLS and how the results could be explored using the interactive features of PEARLS. A user searching for the 'right' listing, often encounters thousands of listings as search results to the query posted. Browsing through the entire list to analyze and identify the 'right' one is non-trivial and time-consuming task. Hence, these exhaustive search results are under-utilized most often. The regular modes of displaying these search results are: (i) list-mode and (ii) map-mode based on the location. An intuitive and information mode of display is needed that helps the user to browse through the data without getting overwhelmed with information deluge. We suggest PEARLS as an alternative to the regular list-mode and believe that it brings more structure to the display of the search results. The idea of grouping search results is not new in the field of textual search. But the concept of clustering the search results, partitioning the clusters into pearls and displaying the pearls along with various interactive features for the user to zoom in and out is illustrated as a detailed case study in the experiments section.

For instance, when a query is executed on Singapore's property website property guru, the search results included over 35,000 listings. After clustering, 15 major clusters are identified. Each cluster representing set of listings that are close to each other based on attributes like size of the unit, location, nearest train station and school. The largest cluster is partitioned to obtain pearls. Example of a single pearl goes like this: pearl id 31, Alexis (59 units),LVIV (33 units), Stevens_Suites(22 units), City_Square_Residences (30 units), Visioncrest_Residence (27 units) and City_Loft(13 units). These

listings are located in one district and the clustering and partitioning helped group the listings accordingly. The visualization of the cluster is shown in Figure 1.2. The pearl in red represents the pearl 31.

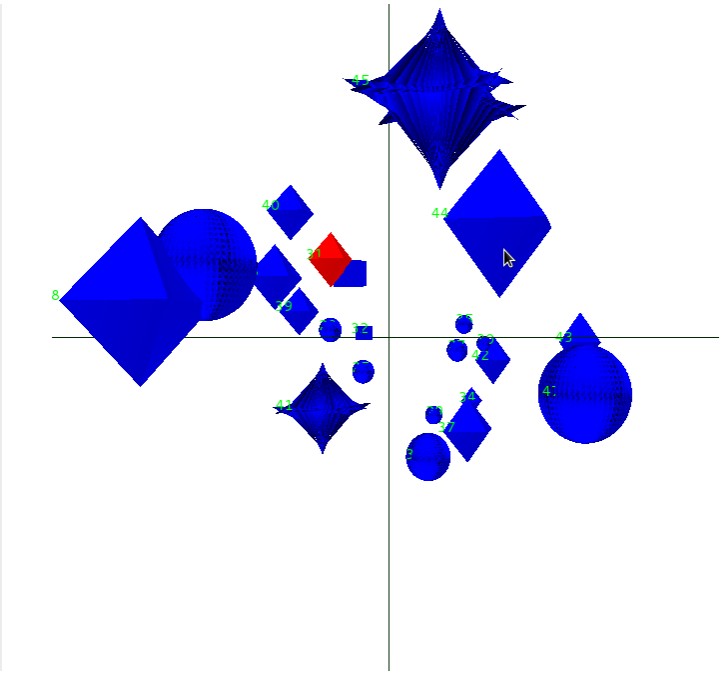


Figure 1.2: This image sequence shows pearls for a cluster of Singapore real estate dataset. Pearl with pearl id 31 is highlighted in red.

Using the PEARLS visualization tool, the user is first provided with 15 clusters to choose. Summary of each cluster is provided, so the user could identify clusters of interest. Upon choosing the cluster, the tool partitions cluster into pearls and obtains the PEARL visualization of that cluster. Clicking on each pearl, the user could obtain more information. Accordingly, the user could then explore pearls close to each other. Each pearl could be zoomed in and out, to obtain information at various levels (point-level, pearl-level and cluster-level). Thus, the information is presented in a hierarchical fashion; information could be broken down to individual listing level. The user could choose the attribute (or dimension) that matters most like 'distance to nearest MRT' and order the pearls along that dimension to identify groups of listing that satisfy the criteria. We believe that an overall system like PEARLS could help in situations when there is information overload (for numerical data sets) and with suitable data preparation for PEARLS, the features it provides could be used to explore, analyze and zoom into specific points for more detail. Searching across data sets and representing the results of the clusters is a good example of an application where PEARLS fits.

1.3 PEARLS and BEADS

PEARLS is based on some of the ideas presented in BEADS [47],[46]. BEADS

- is a framework to provide conceptual visualization of high dimensional data clusters.
- is an approach to compose 2-D visual metaphors to visualize high dimensional clusters.
- gives a formalism for shape definition and applies it for shape identification.
- has results on real and synthetic data sets to illustrate the approach.

A system called BEADS [46] was implemented in C++ and rendering was done using a set of Gnuplot commands. The BEADS system can at best be described as a proof of concept but it has limited applicability in data visualization and exploration tasks. BEADS framework was not developed to meet requirements of a visualization tool. Also, the bead shape identification algorithm was based on some approximations derived by computing ratio's of 2-D Lp norm shapes. We have found out that as we move to higher dimension shapes these approximations do not hold. For example ratio of area of 2-D sphere to 2-D plus (p = .5) is 4.75 is while that of a 4D hyper sphere and plus is 776.175. We developed PEARLS system to present next generation of BEADS system.

1.3.1 A Case for 3-D PEARLS

Reasons which compelled us to design a 3-D visualization instead of 2-D are:

1. **Overlap in beads** In 2-D, due a limited canvas size, lot of beads overlap. Larger beads completely overlap some smaller beads. In 3-D, a user can rotate the camera and view same visualization from various angles which addresses the problem of overlap.

2. In 2-D BEADS, the position of a bead conveys only its distance from center and the quadrant in which this bead lies. Position is one of the key features in conveying visual information and a 3-D visualization enables us to convey more information using position of beads. In 3-D, position of a pearl conveys its distance from cluster centroid, the quadrant in which it lies and its value in dimension with maximum standard deviation. An extra dimension also enables us to introduce data dimension interactive technique (described in section 4.1).

Figure 1.3 shows some plots generated by Beads System.

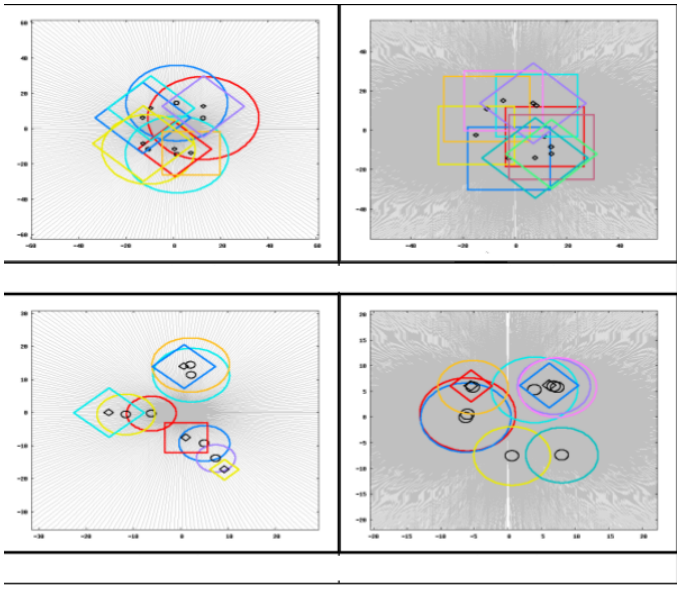


Figure 1.3: Some of the plots generated by BEADS for various clusters.

In [29], authors have identified baseline requirements of software systems built for data visualization. There are seven functional requirements (i.e., rendering scalability, information scalability, interoperability, customizability, interactivity, usability, and adoptability) and seven non functional requirements (i.e., views, abstraction, search, filters, code proximity, automatic layouts, and undo/history).

1.3.2 Non Functional Requirements

Among all non functional requirements both BEADS and PEARLS equally satisfy requirements of rendering scalability (since both compute in C++ and number of beads and pearls is much small as compared to number of data points in most cases) and interoperability.

Information Scalability A data visualization tool must be able to reduce and extend the amount of visualized information on demand. It should also allow user to select and view the information of interest. The PEARLS toolkit build in QT using OpenGL allows information scalability since QT and openGL support building dynamic graphic views where amount of visualized information can be reduced or extended. A user can change the number of pearls to increase/decrease the extent of visualized information. He/She could also hide certain pearls and generate the pearls again, which in effect removes the information generated from points contained within hidden pearls. Figure 1.4 shows examples of information scalability of PEARLS. In part a, only 15 pearls are generated; In part b, 20 pearls are generated ; In part c, some pearls are removed from original visualization shown in part b.

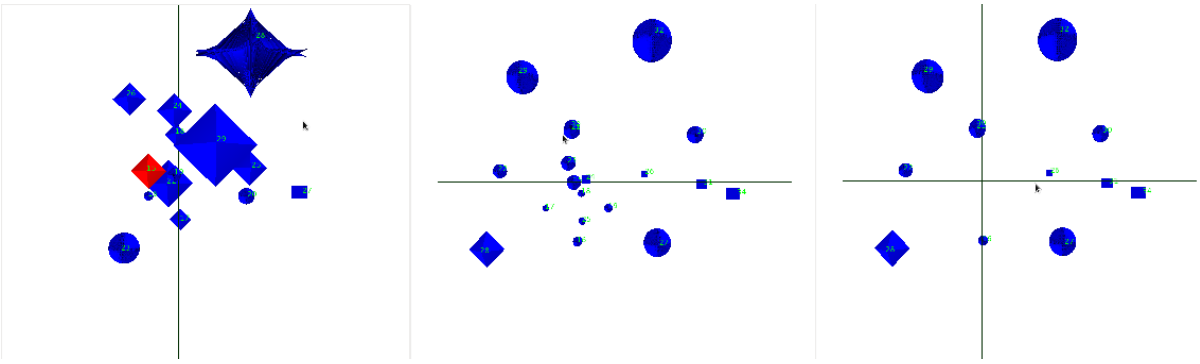


Figure 1.4: Information Scalability in PEARLS

Customizability Customizability enables to meet needs that cannot be foreseen by tool developers, for instance, if a tool is applied in a new context. Most visualization tools do not allow easy extension by new visualization techniques. Customizability depends on programming language, development framework and modularity of code. PEARLS implementation supports extensive customizability as it is written in C++ using QT/OpenGL framework in a modular manner. We have extended PEARLS framework by adding various standard multivariate data visualization techniques implemented in Xmdv toolkit[50] . This demonstrates that an QT/OpenGL based independent framework is much more customizable as compared to bead Plots drawn in Gnuplot.

Interactivity A visualization tool should be interactive and exploratory in nature. While Gnuplot is an interactive toolkit, its goal is plotting of mathematical functions. Hence, the nature of interactivity it supports does not meet requirements of a data visualization toolkit. PEARLS framework provides interactive features like Swiss cheese view, use of a dimension as data dimension, attribute filtering and browsing points in pearls and Clusters in tabular form.

Usability For any toolkit usability or ease of use is a very important requirement. User interface of a visualization toolkit should be intuitive to user. In the context of a code visualization toolkit, Storey et al. say that “available functionality should be visible and relevant and should not impede the more cognitively challenging task of understanding a program”. The PEARLS interface adheres to this guiding principle. User Interface of PEARLS is showing figure 4.1.

Adoptability Bull et al. [6] state that ease of use and adoptability to the tasks cause a tool to be adapted. Reiss [40] says that developers will adopt tools only if ”the cost of learning that tool does not exceed its expected rewards and the tool has been and can easily shown to provide real benefits”.

Tilley et al. [45] suggest that research tools might be more adoptable if they were more understandable, robust, and complete. While a number of qualities have been listed which makes a tool adoptable there is no exhaustive checklist which can guarantee adaptability. While designing PEARLS framework we aimed at designing an easy to use, understandable and robust toolkit. We believe that the toolkit will be adopted by users.

1.3.3 Functional Requirements

Among functional requirements, both BEADS and PEARLS follow identical abstraction model and permit similar code proximity (ability of the visualizer to provide easy and fast access to the underlying source code) .

Views A Visualization software provide multiple views to satisfy the need of different stakeholders and to emphasize on different aspects of data. Integration of multiple views is also an important aspect of visualization system as it supports ease of exploration. PEARLS has closely integrated dynamic textual and graphical views which are described in section 4.1.7. Gnuplot based implementation does not provides flexibility to develop such views. Examples of graphic Views and textual views of several datasets are given in Chapter 5 and 6.

Filters Filtering of information in visualization tools can be seen as a rudimentary form of (structural) querying [44]. Visualization tools support data filtering in several ways like interactive filtering or criterion function based filtering. Filtering allows users to reduce the amount of visualized data and to limit their analysis. PEARLS supports filtering using concept of swiss cheese view and attribute filtering as described in section 4.1 . Due to limitations of Gnuplot, BEADS could not support this feature. Figure 1.5 shows using swiss cheese filter to remove Pearls 30, 32, 27 and 29.

Undo/History Undo/History is a feature present in most interactive visualization toolkits and graph editing toolkits. Since user performs interactive manipulations that change the visualization, there should be an undo mechanism that allows them to revert to previous states. In PEARLS all interactive manipulations can be undone. The two major interactive manipulations are hiding pearls using swiss cheese view and regenerating the pearls by changing various parameters. The history of all interactive steps along with their results is stored to make it possible to revert back to previous states when needed. The BEADS implementation does not support it due to limitations on interactivity and static views. Figure 1.6 shows undoing the removal of Pearls 27 and 29 using Undo History button.

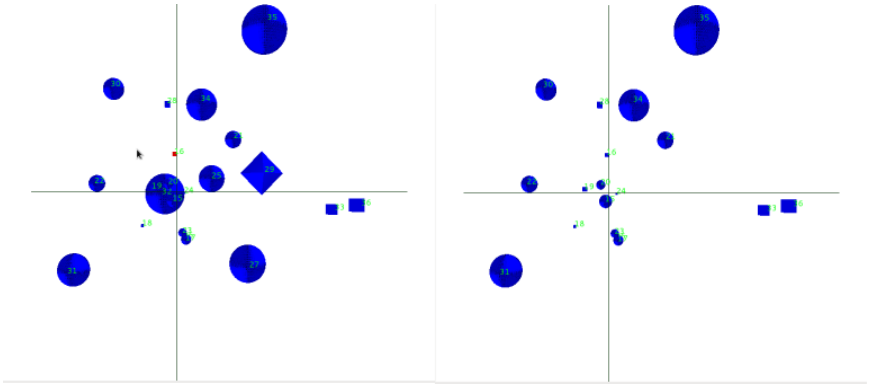


Figure 1.5: Filters in PEARLS

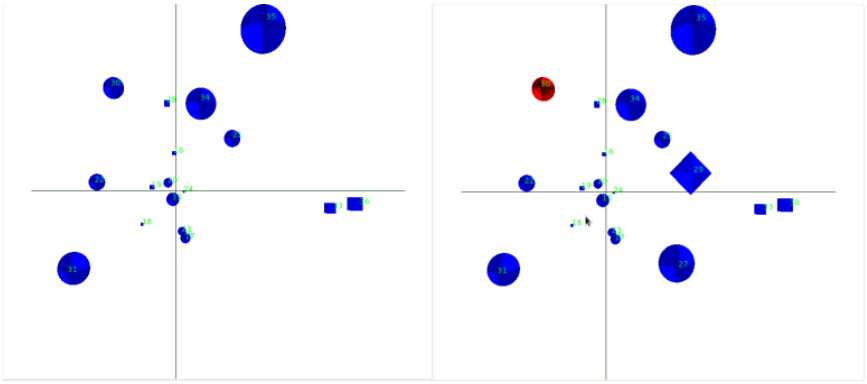


Figure 1.6: Undo in PEARLS

Search Storey et al. [44] observe that lack of a searching tool to find text strings from the data hinders the user. PEARLS visualizes multidimensional datasets where each point can/cannot have a textual name. PEARLS toolkit supports searching of points using their names while BEADS tool does not support it. Figure 1.7 shows searching ”Urbana“ using search box in Singapore real estate dataset.

Figure 1.7: Search in PEARLS

1.4 Organization of Thesis

The rest of the thesis is organized in the following manner.

In Chapter 2, we look at the **Related Work** in the field of multi dimensional data visualization and cluster visualization. We shall provide an exhaustive coverage of prominent work conducted in these areas in the past decade.

In Chapter 3, we give an overview of PEARLS. We describe the algorithms which generate PEARLS plot from a data cluster and present underlying concepts.

In Chapter 4, we describe PEARLS toolkit. We highlight the functionality and interactive features of the toolkit, discuss the non functional requirements which it satisfies and describe major implementation details.

In Chapter 5, we introduce concept description and concept discrimination and describe how PEARLS toolkit is useful for the same. We describe queries which PEARLS system can answer and give a sequence of steps to be followed for these queries. We also describe how such a sequence of steps can be constructed for new query types.

In Chapter 6, we provide data analysis case studies on singapore real estate dataset and nutrition dataset. In the case studies, we use PEARLS toolkit to perform concept search tasks on dataset. We show PEARLS visualization for each step of the task.

In Chapter 7, we highlight our conclusions followed by providing some ideas for the future work.

Chapter 2

Related Work

In this chapter we look at the related work in the field of multi dimensional numerical data visualization and cluster visualization.

2.1 Multidimensional Data Visualization

Chan [9] has done a comprehensive survey on techniques for multi variate data visualization. We describe the major techniques.

2.1.1 Scatterplot Matrix

In a scatter plot two attributes are projected along the x-y axes of the Cartesian coordinates. A Scatterplot matrix is collection of scatterplots organized in a matrix to provide correlation information among the attributes. Scatterplot matrix is primarily used to observe patterns in the relationships between pairs of attributes. Figure 2.1 shows a scatter plot matrix for a 5 dimensional data of 500 automobiles.

2.1.2 Prosection Matrix

Prosection [16] is an orthogonal projection where the data items that lie in the selected multidimensional range are colored differently. Prosection matrix is a collection of prosections organized in a matrix. Figure 2.2 shows a prosection and a prosection matrix.

2.1.3 Parallel Coordinates

Parallel coordinates [21] [20] [23] is a well known multidimensional data visualization technique. In parallel coordinates, a backdrop is drawn consisting of n parallel lines(n is number of dimensions of data), typically vertical and equally spaced. A point in n-dimensional space is represented as a poly

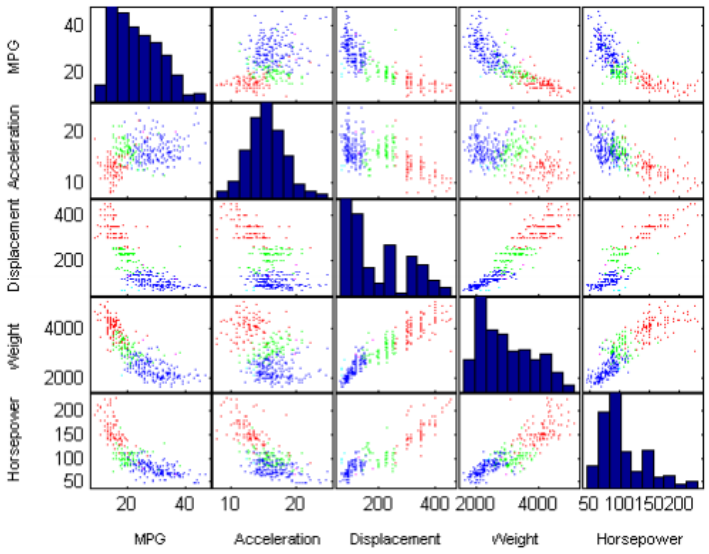


Figure 2.1: A scatterplot matrix for 5 dimensional data of 500 automobiles

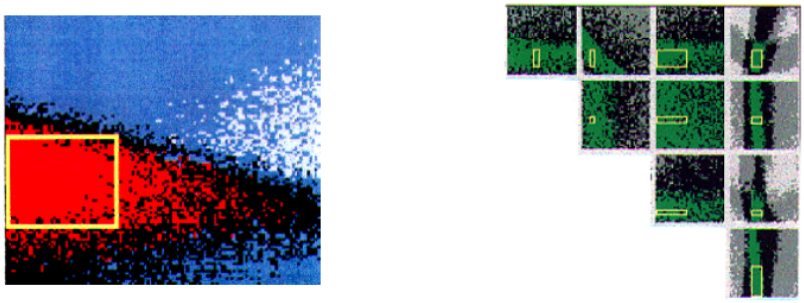


Figure 2.2: (a) A prosection, (b) A prosection matrix

line with vertices on the parallel axes; the position of the vertex on the i^{th} axis corresponds to the i^{th} coordinate of the point. Figure 2.3 shows a parallel coordinates plot.

2.1.4 Radical Coordinates Visualization

In Radical Coordinates Visualization [18], n lines emerge radically from the center of the circle and terminate at the perimeter. Each line represents one attribute. Spring constants attached to the data attribute values define the positions of the data points along the lines. Points with approximately equal or similar dimensional values lie closer to the center. Figure 2.4 shows an example of radical coordinates visualization.

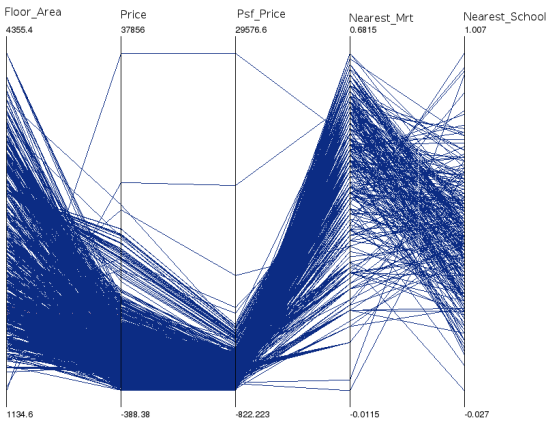


Figure 2.3: Parallel Coordinate Plot

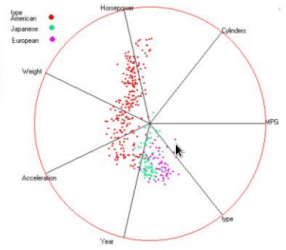


Figure 2.4: Radical Coordinates Visualization

2.1.5 Table Lens

In table lens visualization [39], each row represents a data item and the columns refer to the attributes. Each column is viewed as a histogram or as a plot as shown in figure 2.5. Table Lens allows users to spot relationships and analyze trends in dataset.

2.1.6 Pixel Oriented Techniques

In pixel oriented techniques, idea is to represent an attribute value by a pixel based on some color scale. For an n-dimensional dataset, n colored pixels are needed to represent one data item.

Spiral technique [27] is a pixel oriented technique to visualize query results over multidimensional dataset. In this technique, pixels are arranged in spiral form according to the overall distance from the query. Figure 2.7 shows an example of spiral technique visualization.

Circle segment [2] is also a technique to visualize query results over multidimensional dataset. It assigns attributes on the segments of a circle. Data items are arranged within a segment so that a single data item appears in the same position at different segments. The ordering and colors of the pixel

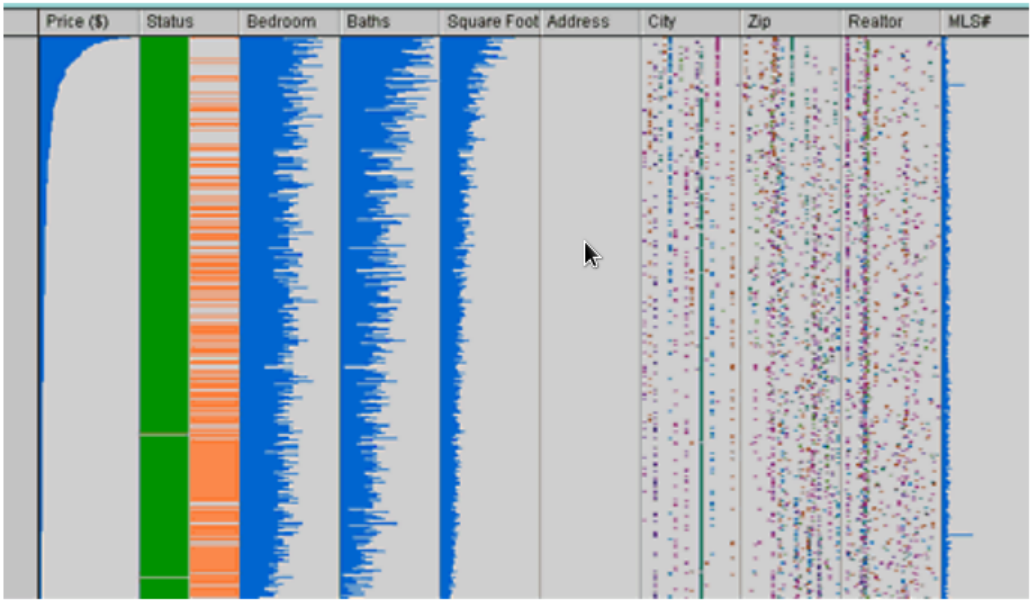


Figure 2.5: An example of Table Lens from Inxight

are determined using overall distance to the query. Figure 2.8 shows an example of circle segment visualization on a 8 dimensional dataset.

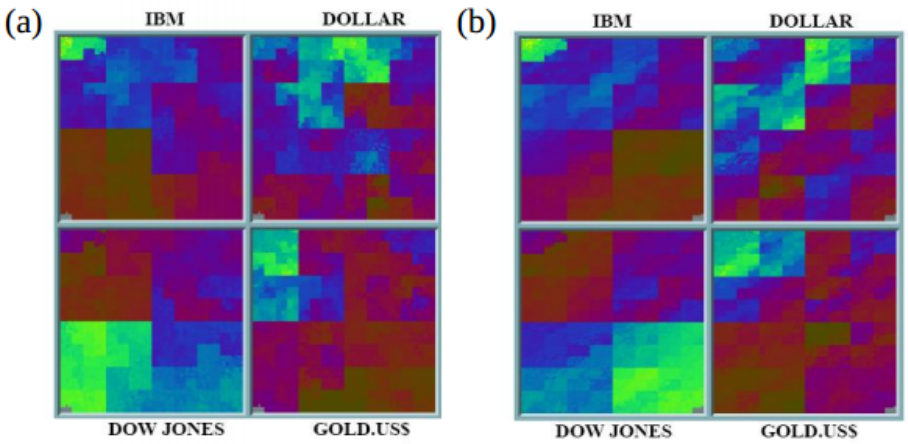


Figure 2.6: Space Filling Curves

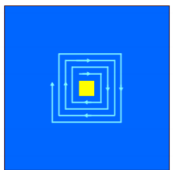


Figure 2.7: Spiral Technique

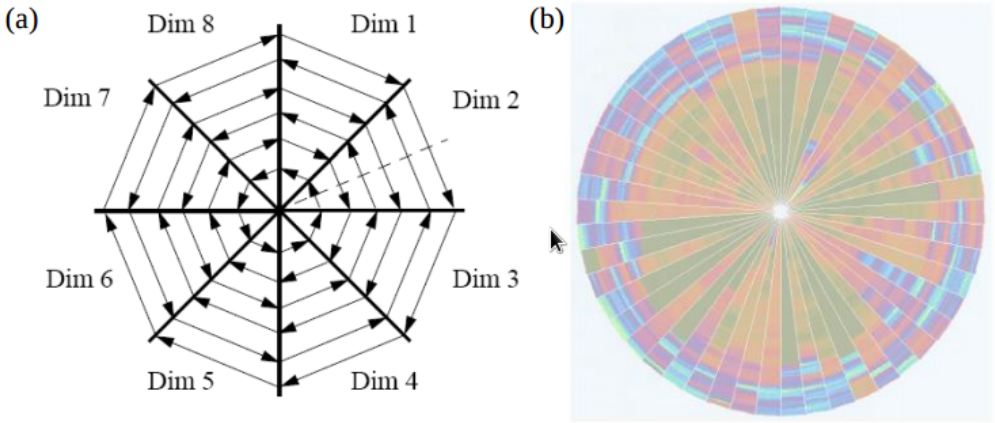


Figure 2.8: (a) Circle segment arrangement for 8-diemensaionl data,(b) An example of circle segments

2.1.7 Hierarchical Axis

In this technique [52] [34], axes are laid out horizontally in a hierarchical fashion. This technique can plot many attributes in one screen. A simple example is the histogram within histogram plot as show in figure 2.9.

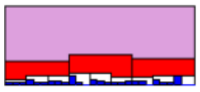


Figure 2.9: Histogram within Histogram Plot

2.1.8 Dimensional Stacking

Dimensional Stacking [31] technique partitions the data space into 2-dimensional subspaces which are stacked into each other. Important attributes should be choosen for outer levels of stack. Figure 2.10 shows a partition of dimensional stacking and a complete example of dimensional stacking technique.

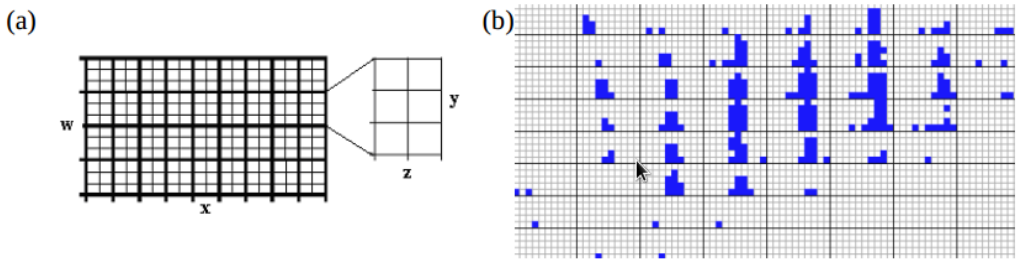


Figure 2.10: (a) Partition of dimensional stacking, (b) An example

2.1.9 Treemap

Treemap [42] partitions the screen hierarchically into regions depending on the attribute values. The sizes of the nested rectangles represent the attribute values. The color of the regions may encode an additional attribute. Figure 2.11 shows an example of Treemap visualization.

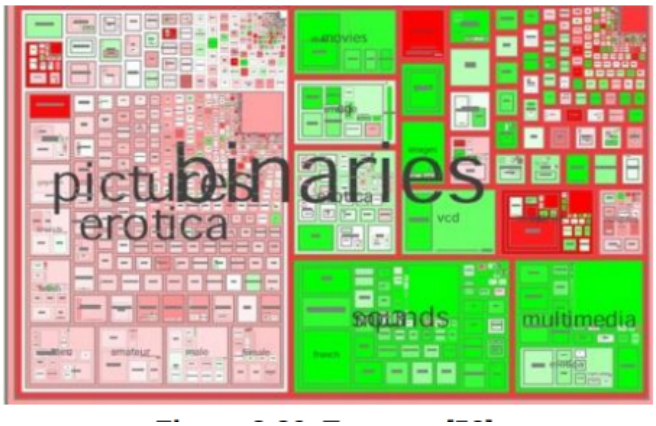


Figure 2.11: Treemap

2.1.10 Chernoff Faces

In this technique [10] two attributes are mapped to the 2D position of a face and remaining attributes are mapped to its properties like the shape of nose, mouth, eyes and that of the face itself. Chernoff faces can only visualize a limited amount of data items. Figure 2.12 shows an example of Chernoff Face visualization.

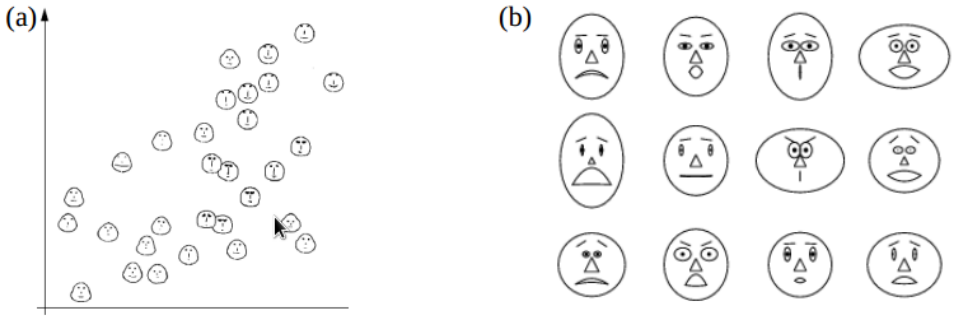


Figure 2.12: : (a) Chernoff faces in various 2D positions, (b) Different facial features

2.1.11 Star Glyphs

In Star Glyphs [8][19], the dimensions are represented as equal angular axes. An outer line connects the data value points on each axis, as depicted in Figure 2.13. Each data item is presented by one star glyph.

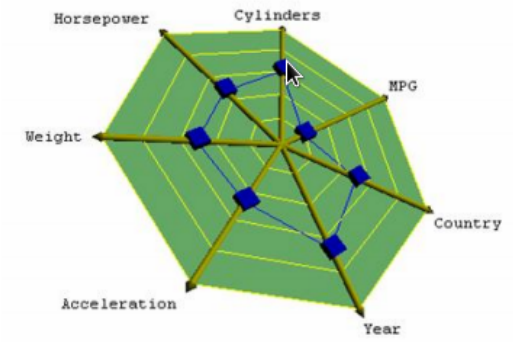


Figure 2.13: Star Glyph

2.1.12 Stick Figure

Stick figure [38] is a visualization technique that maps two attributes to the display axes of stick figure and the remaining to the rotation angle, length, thickness or color of the limbs. According to [9], "when the data items are relatively dense with respect to the display dimensions, the packed icons exhibit some texture patterns that vary according to the data features, which are detected by pre-attentive perception". Figure 2.14 shows stick figure visualization for a 5 dimensional dataset.

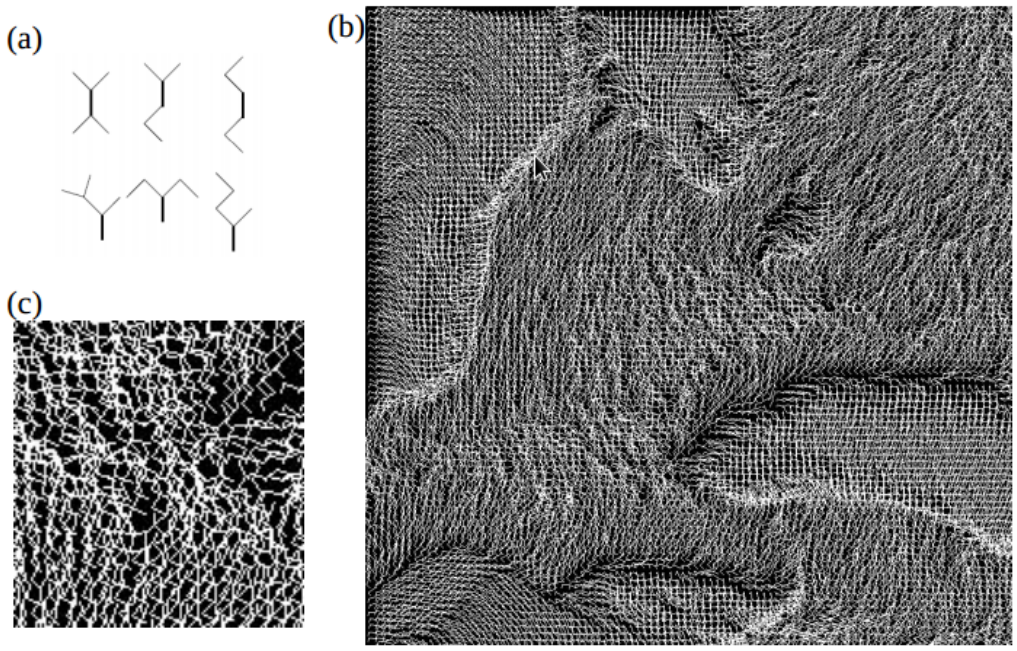


Figure 2.14: (a) Stick figure family, (b) 5D image data using stick figures, (c) Part of (b) in original size

2.1.13 Color Icon

Color icon [32] is a combination of the pixel-based spiral axes and icon based shape coding techniques. Pixels are replaced by arrays of color fields that represent the attribute. Various properties like color, shape, size, orientation, boundaries and area sub-dividers can be used to map the multidimensional data to these color fields. Figure 2.15 shows a color icon visualization for a 5 dimensional dataset.

2.2 Cluster Visualization

Related work in cluster visualization can be described in three categories. Using general purpose multidimensional data visualization techniques to visualize data clusters, Icon-Based Multivariate data visualization techniques and various interactive cluster exploration tools.

2.2.1 Multidimensional data visualization techniques

Parallel coordinates [22] is a well known multidimensional data visualization techniques. In parallel coordinates, a backdrop is drawn consisting of n parallel lines(n is number of dimensions of data), typically vertical and equally spaced. A point in n-dimensional space is represented as a poly line with

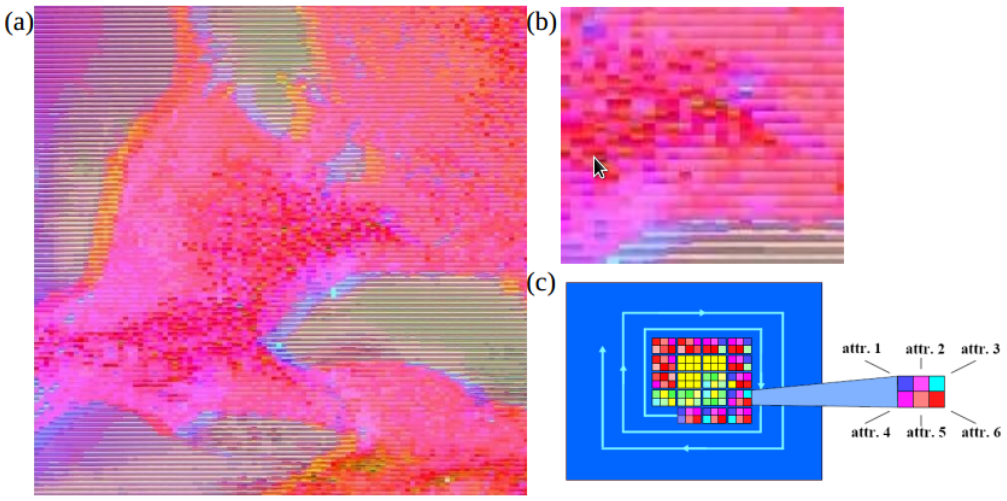


Figure 2.15: (a)5D image data using color icons, (b) Part of (a) in original size, (c) Color icon scheme

vertices on the parallel axes; the position of the vertex on the i^{th} axis corresponds to the i^{th} coordinate of the point. Several variations of parallel coordinates have been widely used to visualize multi dimensional clusters and to identify cluster patterns in large multidimensional datasets.

For example, Fua et al. [15] used hierarchical clustering to develop a multiresolutional view of the data and used a variation on parallel coordinates to convey aggregation information for the resulting clusters. Authors perform a hierarchical clustering on dataset. A level of detail control parameter w is chosen to select clusters from hierarchy tree which are to be displayed. ($S(w)$ is the collection of clusters whose size v_i is less than or equal to w but whose parent's size is greater than w.) Clusters are represented on parallel coordinate plot as variable-width opacity bands. The mean of the cluster stretches across the middle of the band and is encoded with deepest opacity which is a function of the density of a cluster. The top and bottom edges of the band have full transparency. The opacity across the rest of the band is linearly interpolated. The thickness of the band across each axis section represents the extents of the cluster in that dimension. These opacity bands are colored on the basis of cluster proximity. This proximity function is based on the structure of the hierarchical tree, that is sibling nodes are considered closer than non-sibling nodes. Figure 2.16 shows a series of images captured at six varying levels of abstraction for and 8 dimensional Fatal Accident Reports Dataset. The efficiency of this visualization technique is dependent of complexity of clustering technique. After clustering, for every cluster a visual object can be computed in constant time using meta data generated while clustering. Visual clarity depends on level of detail control parameter w. For equal number of clusters

and points, this technique and parallel coordinates have equivalent visual clarity. Figure 2.2.1 shows figure of PEARLS visualization for a cluster of same dataset.

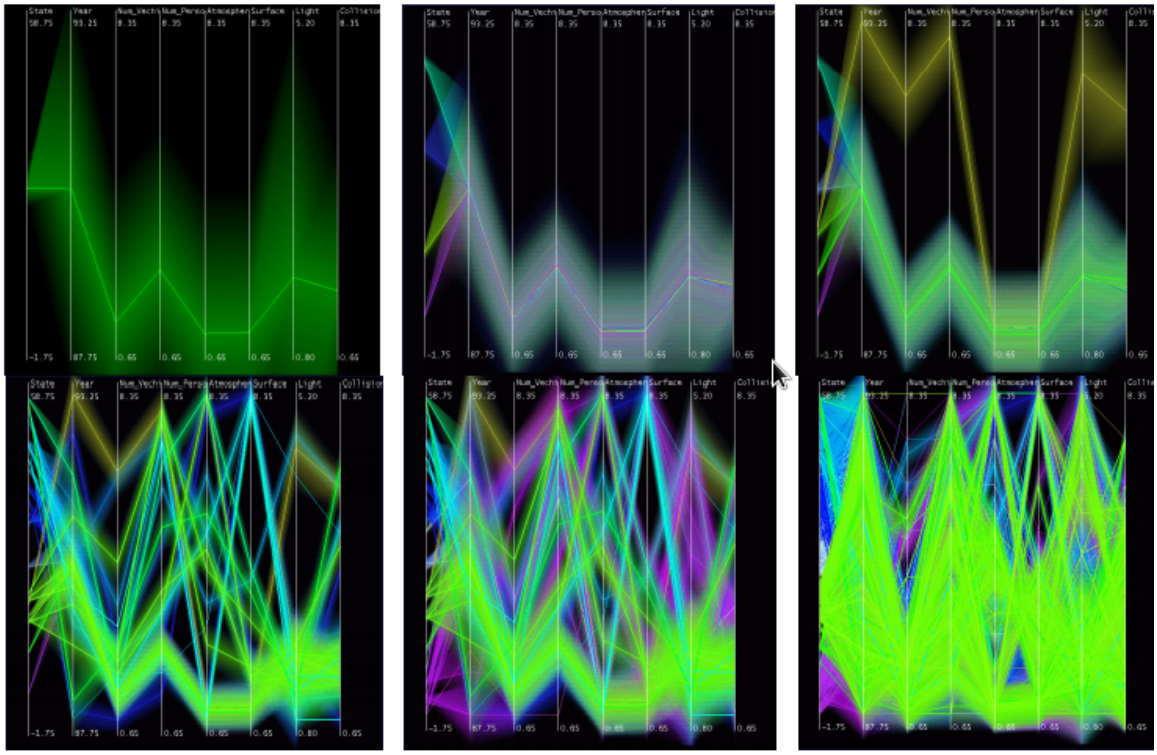


Figure 2.16: This image sequence shows a Fatal Accident data set at different level of details. The first image shows a cut across the root node. The last image shows the cut chaining all the leaf nodes. The rest of the images show intermediate cuts at varying levels-of-detail.

Novotny [36] partitioned data into clusters and represented each cluster as a polygonal area on a parallel coordinate plot and used both opacity values and textures to distinguish different clusters. Opacity values can be chosen based on three criterion : uniform mapping (all clusters have the same opacity 1/k, where k is the number of clusters), population mapping (the opacity value is the ratio of the cluster population to the number of samples in the whole data set) and density mapping (the opacity value is the ratio of the cluster population to the size of the cluster. Aim of coloring is to have colors as mutually different as possible and share the same intensity. A qualitative scheme acquired from [5] is applied to find color of the individual elements. The palette distinguishes the objects by hue while keeping the color intensity relatively the same for all colors. Figure 2.18 shows Novotny's visualization. Computation complexity is dependent on the complexity of clustering algorithm. For every cluster, visual objects can be computed in constant time. This technique has considerable improvements over Parallel Coordinates technique in terms of visual clarity.

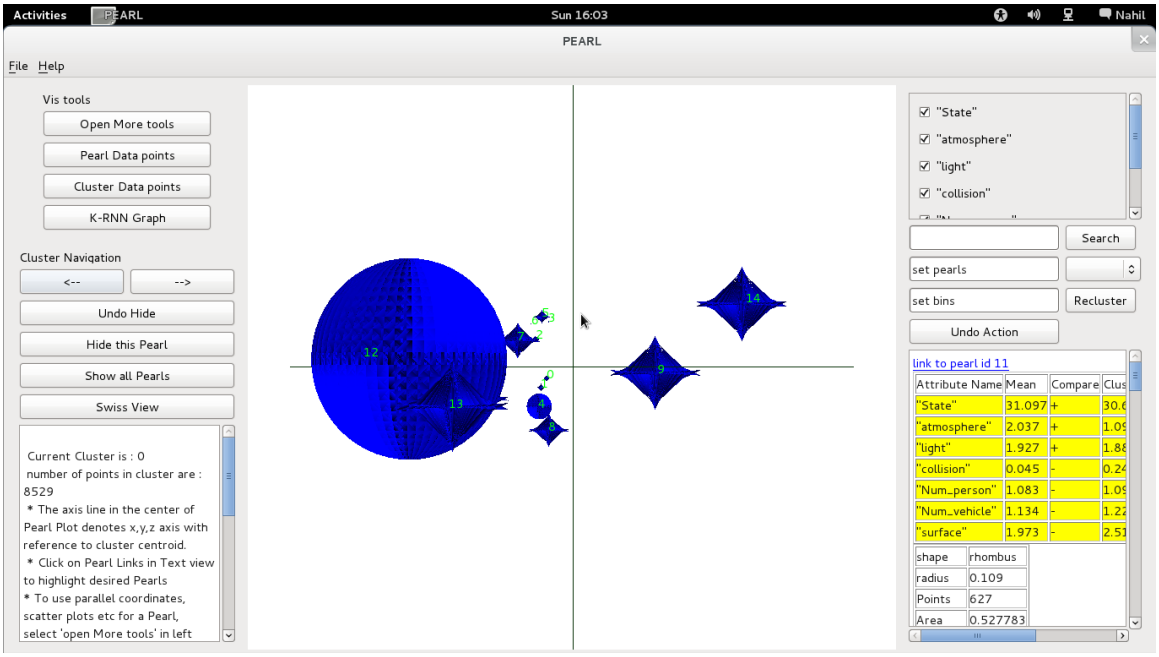


Figure 2.17: PEARLS visualization for a cluster of Fatal Accident dataset. Most attributes of fatal accident dataset like Number of persons, Number of vehicles, atmosphere, light take discrete integer values between very small range 0-15. This leads to a bias towards plus shapes.

Parallel coordinates technique suffers from over-plotting, resulting in an image which is far too cluttered to perceive any trends, anomalies or structure. [36] and [15] visualize clusters instead of data items on a parallel coordinate plot. But in order to fully investigate a data set, it is necessary that cluster representations must reveal detailed information about each individual cluster Johansson et. al. [24] proposes use of high precision textures to represent these clusters on a parallel coordinate plot. In this approach, for every cluster, all the data items belonging to it are rendered and the maximum number of data items intersecting anywhere on display, p_i is computed. p_{global}, the largest of the p_i values is then computed and intensity range is normalized using either p_{global} or p_i. If normalization is done using p_{global}, clusters with a small maximum intersecting value become more transparent. If p_i are used for each cluster, each cluster's maximum intensity is perceived as equally dense. Due to the sometimes large range of intensities, the human eye has difficulty in perceiving the smallest intensity values. A linear scaling may lead to clamping of highest intensity values. So authors propose use of transfer function (TF) which allows non-linear as well as user-defined mapping. Figure 2.19 shows visualization using various transfer functions. Computation complexity of this technique is dependent on clustering algorithm as well as transfer function. It is also dependent on GPU architecture for computation of p. Visual clarity as well as level of detail are dependent on transfer functions. A logarithmic function used

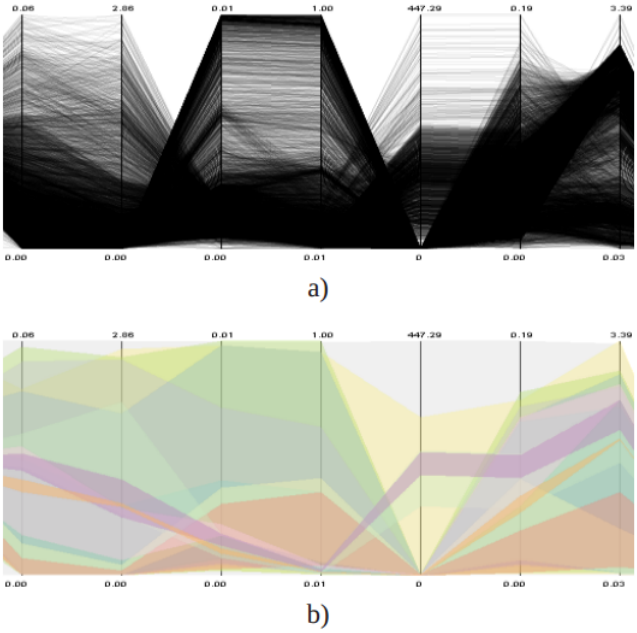


Figure 2.18: The same data is visualized by traditional parallel coordinates (top) and using visual abstraction (bottom) by Novotny

for Figure 2.19.(d) provides low visual clarity with high level of detail and a function used in Figure 2.19.(b) provides an overview of data with high visual clarity.

Principal Component Analysis, Multidimensional Scaling and RadVis [35] can represent multidimensional data into 2-D which makes it easier to identify clusters present in dataset but they result in loss of structure and shape of clusters. They fail to preserve the spatial relationships and much of the data analytics process is based on spatial geometries and densities. They also do not take account the apriori knowledge which a user might posses about semantics of dimensions.

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components.

Multidimensional Scaling(MDS) is a non-linear method for projecting n-D data down to a reduced number of dimensions [4]. n data points can, for example, be represented as 2-D display points. The MDS algorithm attempts to make the 2-D display points accurately reflect the relationships that exist

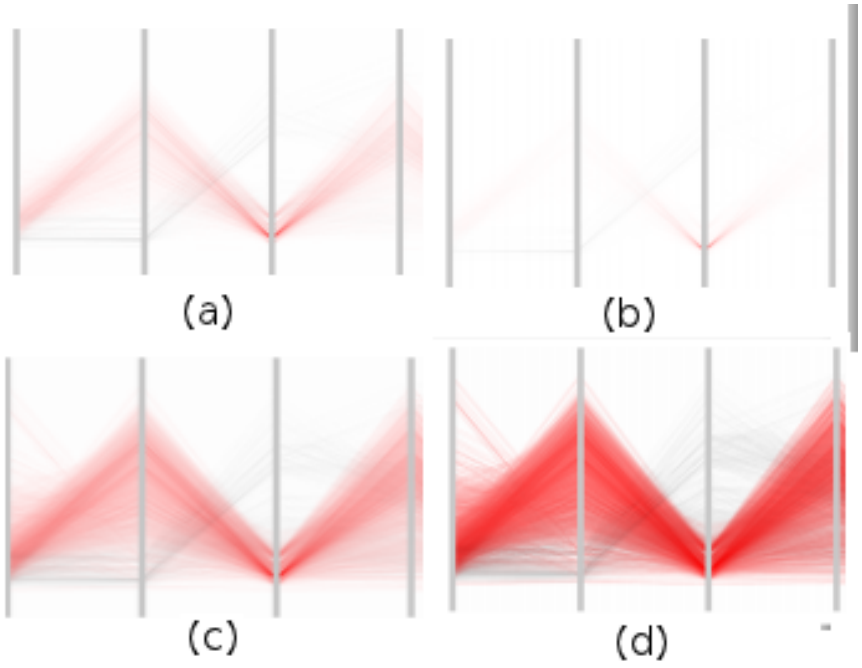


Figure 2.19: (a)A linear TF prevents cluttering and to provide overview of the data. (b) Square TF shows only dense regions. (c)A square root TF enhances low density regions and makes it easier to search for outliers. (d)A logarithmic TF puts even more emphasis on the lower density regions.

between the corresponding n-D points by iteratively evaluating the level of Stress in the configuration (high stress means the 2-D relationships are poorly correlated with the n-D relationships) and moving the 2-D points in a direction of reduced stress. Unfortunately, it is difficult for users to understand the semantics of the clusters as well as perform visual analysis on a MDS plot.

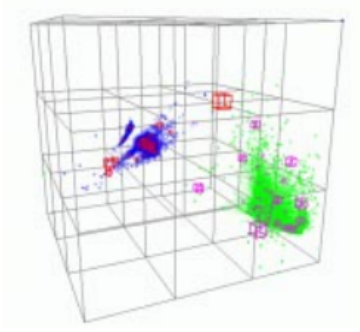
MDS and Principal component Analysis are computationally expensive techniques. The computation of PCA requires eigenvalue decomposition (EVD) of the co variance matrix of the feature vectors. One well-known EVD method is the Cyclic Jacobi's method which diagonalizes a symmetric matrix. It requires around O($d^3 + d^2n$) computations, where d is dimensionality and n is number of data points. Standard MDS operates by means of eigen vector analysis of an N x N matrix. It produces a layout based on a linear combination of dimensions. It is an O(n^3) procedure. 2 dimensional layouts generated by MDS and PCA consist of points only so they are of high visual clarity and free from over plotting.

RadViz (Radial Coordinate visualization) [35] is a visualization method, which uses the Hooke's law from physics for mapping a set of n-dimensional points into a plane. Each RadViz mapping of points from n dimensional space into a plane is uniquely defined by position of the corresponding n anchors (points S_j, which are placed in a single plane. The anchors are most often situated around a circle, but this is not necessary. It is supposed that each anchor holds its own virtual spring of variable stiffness

and all the loose ends of the springs are bound together. A point $[y_1, ..., y_n]$ from n dimensional space is mapped into a single point u in the plane of anchors as follows: for each anchor j the stiffness of its spring is set to y_j and the Hooke's law of mechanics is used to find the point u, where all the spring forces reach equilibrium (they sum to 0). The position of $u = [u_1, u_2]$ is given by the equations

$$u_1 = \frac{\sum_{j=1}^{n} y_j \cos(\alpha_j)}{\sum_{j=1}^{n} y_j}$$
$$u_2 = \frac{\sum_{j=1}^{n} y_j \sin(\alpha_j)}{\sum_{j=1}^{n} y_j}$$

In [26] author uses non linear magnification to visualize high dimensional data clusters. N-dimensional dataset is described in terms of 3D frames of the data. For a data set composed of N-dimensional points a user can select frames of the data with each frame representing three of the N possible dimensions. The key idea here is to use overlaying frames within a single three dimensional coordinate space (a single frame presents three dimensions of the data) using color cues to visually separate the dimensions. Selection of data records can either be done by entering the record ID number, or by selecting one or more data points by brushing with the mouse. When a record is selected, it is highlighted in all of the visible frames to show its position relative to the other records in the visible dimensions. Figure 2.20 shows visualization of two frames (six dimensions) in a single three dimensional coordinate space. For every individual record, its corresponding visual element can be computed in constant time. Visual clarity of this technique is similar to PCA and MDS as every record is rendered as a point in three dimensional space.



Visualization of 2 Frames (6 Dimensions)

Figure 2.20: Visualization using Non Linear Magnification

This visualization technique is difficult to interpret as for a viewer the task is still the same in terms difficulty for cognitive abilities and this technique is not effective when dataset is not sparse.

In [25] author uses star coordinates for visually exploring data and cluster discovery. In Star Coordinates, coordinate axes are arranged on a two-dimensional circle with origin as center of the circle

and with equal angles between the axes. Each multi-dimensional data element is represented by a point on a two dimensional plain, where each attribute of the data contributes to its location through uniform encoding. The Star Coordinate (SC) system is basically a curvilinear coordinate system, which can be formally mapped to the Cartesian Coordinates (CC) by defining a two-dimensional point representing the origin on $(x,y) = (o_x, o_y)$ and a sequence of n two dimensional vectors $A_n =< \vec{a}_1, \vec{a}_2, ..., \vec{a}_i, \vec{a}_n >$ representing the axes. The mapping of a data element (D_j) from a dataset D to a point (P_j) in the two-dimensional Cartesian Coordinates is determined by the sum of all unit vectors $u_i = (\vec{u_{xi}}, \vec{u_{yi}})$ on each coordinate multiplied by the value of the data element for that coordinate. Figure 2.21 shows an example of calculation of data point location for an 8 dimensional dataset and figure 2.22 shows star coordinates visualization for car specs dataset. Star coordinates technique require $O(n*d)$ computations to generate visual objects for data points; where n is number of data points and d is dimensionality. Visual clarity of this technique is similar to PCA and MDS since it renders a point on a two dimensional plot for every data point. Figure 2.23 shows PEARLS visualization for a cluster of the car specs dataset.

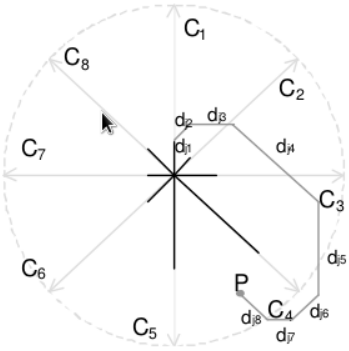


Figure 2.21: Calculation of data point location for an 8-dimensional dataset

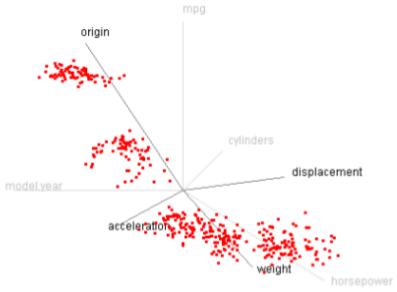


Figure 2.22: Star Coordinates Visualization of car specs dataset

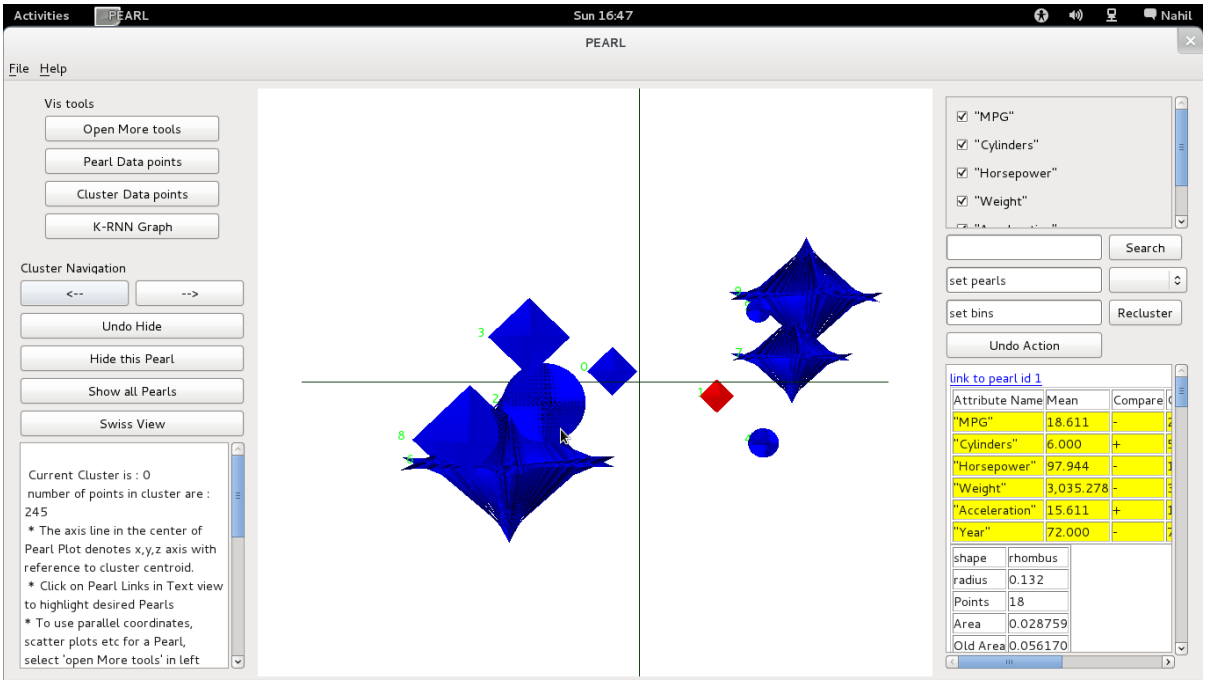


Figure 2.23: PEARLS Visualization for car specs dataset

Various transformations and mathematical operations are then applied to discover clusters, trends and outliers present in data. While this technique helps in cluster and pattern discovery, it is not designed for a comprehensive analytics and visual querying of already existing clusters.

While [36] and [24] follow a similar line of thought as ours, which is partitioning dataset into cluster and then visualizing it, we use a different visualization display and extend our technique to a comprehensive cluster exploratory tool.

2.2.2 Icon Based Techniques

Pickett et al. [37] first proposed to represent multi dimensional data as graphic icons. Every data item was represented using a graphic item, visual properties of graphic item are controlled by attribute values of data item.

In VisDB, authors [27] use a pixel to represent each data item resulting from a query. The query results give the user not only the data items fulfilling the query but also a number of data items that approximately fulfill the query. Data items are sorted according to their relevance with respect to the query and relevance factors are mapped to colors. Hundred percent correct answers are colored with yellow and placed in the middle of the visualization with the approximate answers creating a rectangular spiral around this region.The colors range from yellow to green, blue, red, and almost black to denote

increasing distance from the correct answers. Authors have chosen this color scale empirically. Separate windows are generated for each selection predicate of the query and and are arranged next to window with overall query result. In the separate windows, the pixels for each data item are placed at the same relative position as they appear for that data item in the overall result window. All the windows together make up the multidimensional visualization. By relating corresponding regions in the different windows, the user can perceive data characteristics such as multidimensional clusters or correlations. VisDB requires $O(n*d)+O(nlogn)$ computations to generate visual objects for a single window; where n is number of data points retrieved as result of query and d is dimensionality. $O(n*d)$ computations are required to compute distance of every point and $O(nlogn)$ computations are required to sort them. Figure 2.24 shows example of visualization generated by VisDB system for an eight dimensional dataset with 7000 data items.

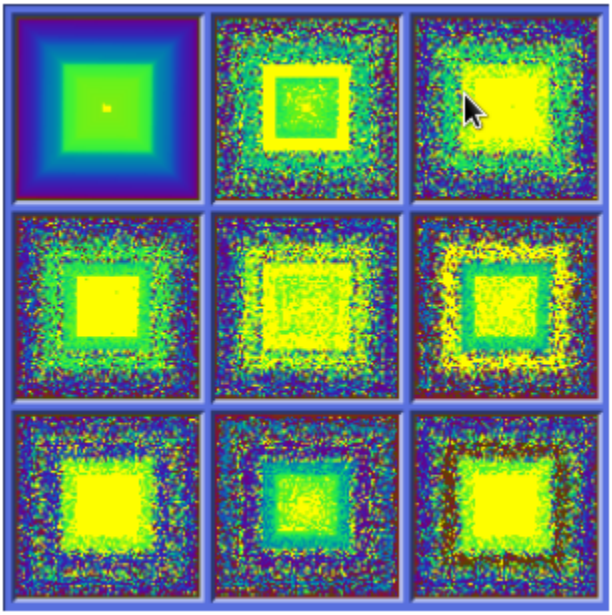


Figure 2.24: A VisDB visualization for Eight-dimensional data (7000 Data Items)

In Value and Relation (VaR) display [53], authors uses pixel-oriented techniques and density based scatter plots to create dimension glyphs to convey values. VaR is created by following four steps:

1. A dimension glyph, is generated to represent data values in each dimension, using pixel oriented techniques. In particular, each value is represented by a pixel whose color indicates a high or low value, and pixels representing values from the same dimension are grouped together to form a glyph. A spiral pixel layout is used to place pixels. In all glyphs, pixels representing values in the same row occupy the same position so that glyphs can be associated with each other.

2. The correlations among the dimensions are calculated and recorded into an $N * N$ matrix (where N is the dimensionality of the data set).

3. The $N * N$ relationship matrix is used to generate N positions in a 2D space, one position for each dimension. The proximity among the positions reflects relationships among the dimensions; that is, closely related dimensions are spatially close to each other, and unrelated dimensions are positioned far way from each other. In particular, an MDS algorithm [4] is used to create the 2D positions upon the relationship matrix.

4. The dimension glyphs are placed in the 2D space in their corresponding positions to form the VaR display. Fig. 2.25b shows an example of the VaR display. It shows the Image-89 data set of 89 dimensions and 10,417 data items.

Figure 2.25(a) illustrates construction of VaR display and 2.25(b) illustrates VaR visualization for a dataset with 89 dimensions and 10,417 items.

Complexity for VaR technique: O($nlogn * d$)+ O($n * d^2$)+ O(d^3); where d is dimensionality and n is number of points. O($nlogn * d$) is required to construct glyphs. O($n * d^2$) is required to compute correlation matrix and O(d^3) computations are required to place glyphs.

DICON, [7] is a dynamic icon-based visualization technique which represents clusters of multidimensional data as compact glyphs. Visual encoding for a patient dataset. In this encoding, an individual entity is described by a feature vector. Each feature in the vector is a numerical value depicted by a small cell. The cells are packed together to generate an individual icon. Individual icons are grouped together by splitting and re-grouping their features into categories. Figure 2.26 shows this visual encoding over a patient dataset. 2.27 shows DICON Visualization for a cars dataset. Authors have developed interaction techniques over visualization to understand, compare and adjust multidimensional data clusters. In the paper, authors do not comment on computational complexity of DICON technique. The computational complexity of generating icons for individual data records is O(d); where d is dimensionality. The complexity for packing these individual icons to form an icon for cluster is implementation dependent. There is no overlap in icons generated by DICON since the position of icon's do not represent the values of data items, hence amount of visual clarity is higher than parallel coordinate based techniques.

While Icon Based techniques can visually encode various statistical attributes of data points within a cluster they result in loss of structure and shape information. They are also crippled by problems hindering other point level abstraction techniques like over plotting, decline in legibility or inability to plot complete dataset.

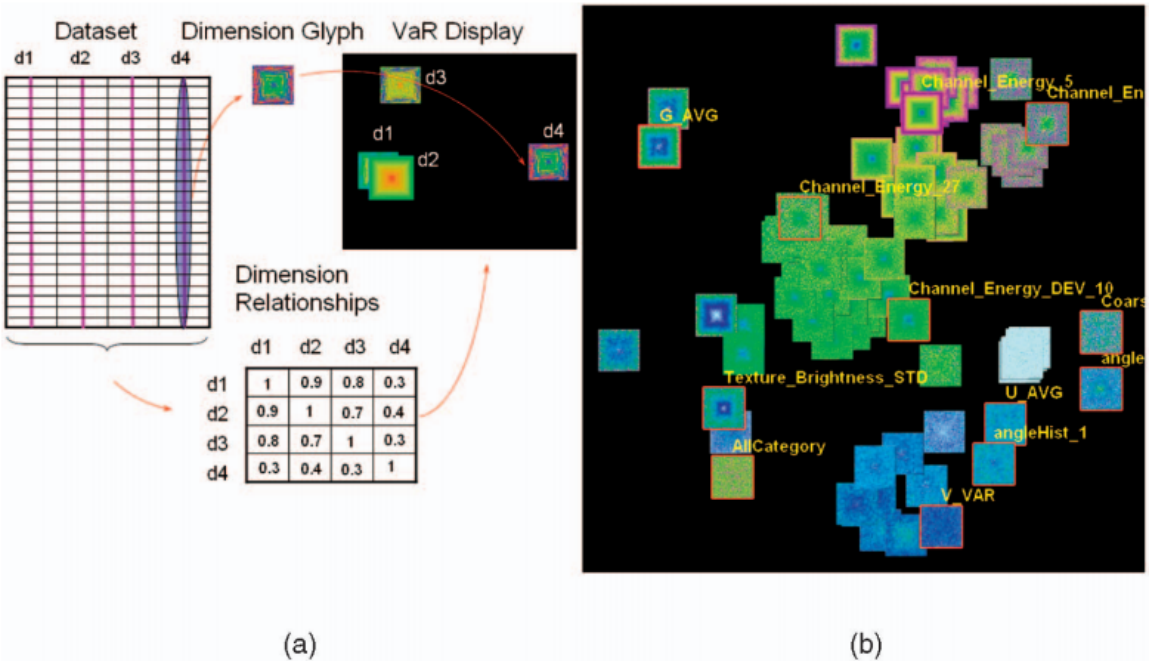


Figure 2.25: 1. (a) Illustration of the VaR display. On the left is the spreadsheet of a 4D data set with each column representing a dimension. At the bottom is a matrix that records the pairwise relationships (such as correlations) among the dimensions. In the middle is the glyph of the fourth dimension. On the right is the VaR display of the data set. (b) The Pixel MDS VaR display of the Image-89 data set (89 dimensions and 10,417 data items).

2.2.3 Interactive cluster exploration tools

Several researchers have designed cluster visualization and exploratory analysis tools by leveraging power of interactivity.

Seo et al. designed HCE(Hierarchical Cluster Explorer) [41] and proposed four interactive features for hierarchical multidimensional cluster analysis. Hierarchical Clustering Explorer's interactive features are

1. overview of the entire data set, coupled with a detail view so that high-level patterns and hot spots can be easily found and examined

2. dynamic query controls that let users eliminate uninteresting clusters and show the interesting clusters more clearly

3. coordinated displays that forge a bidirectional link from the overview mosaic to two-dimensional scatter graphs;

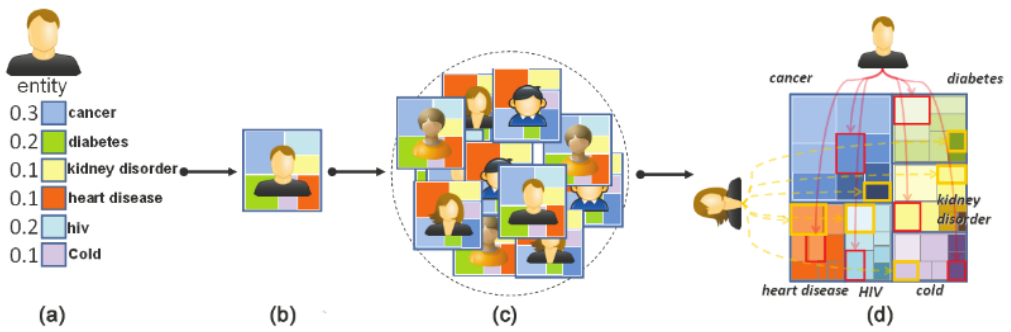


Figure 2.26: DICON encoding shown for a patient dataset

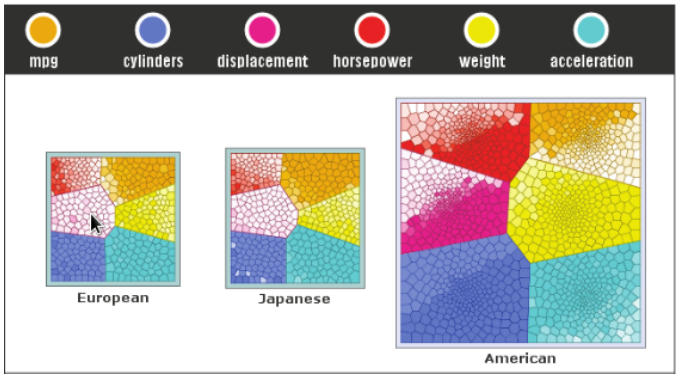


Figure 2.27: DICON Visualization of a cars dataset

4. cluster comparisons to let researchers see how different algorithms cluster the genes.

Primary purpose of HCE is to visualize and analyze dendrogram created by hierarchical clustering and compare hierarchical clusters.

Lex et al. [33] designed an interactive Caleydo matchmaker technique to allow users to split multidimensional datasets into subgroups, cluster them separately and analyze relations between the resulting clusters of each group. The Caleydo Matchmaker technique allows a visual comparison of multiple groups of clustered data. Since there is no inherent order of clusters and records in the clusters, both clusters and records within the clusters are sorted according to their mean value. In addition, having introduced a specific ordering, we can use the parallel coordinates metaphor. The heat maps of groups are generated and arranged side by side, where each group is equivalent to an axis in a parallel coordinates plot. Identical records are connected within the groups. A naive approach for connecting records, results in visual clutter rendering the visualization unusable. Hence authors use an edge bundling strategy. Figure 2.29 displays Caleydo visualization technique over a dataset with 1800 records and three clustering algorithms. Figure 2.29 shows that the k-means algorithm (used in (b)) assigned differently

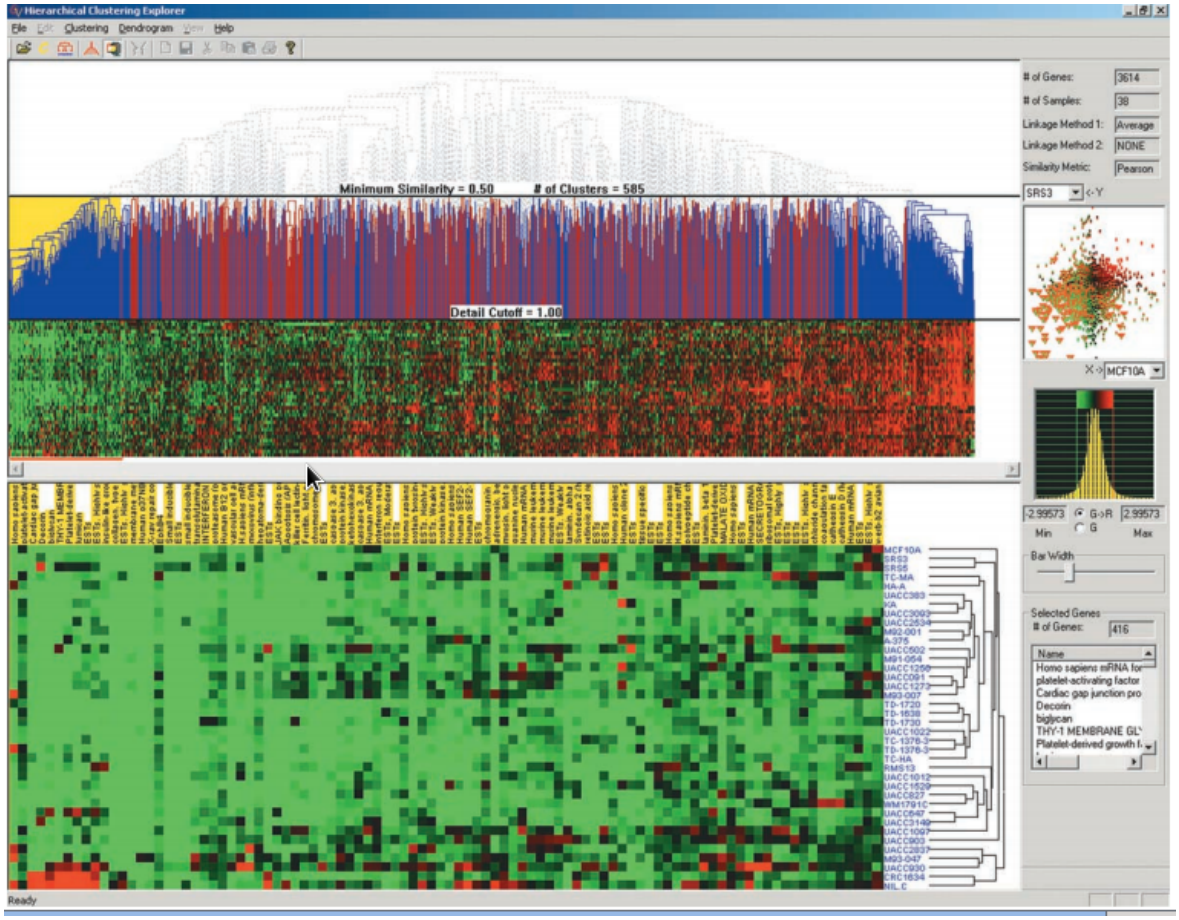


Figure 2.28: Hierarchical Clustering Explorer's compressed overview. The melanoma gene expression profile contains 3,614 genes and 38 experimental conditions. This view shows the entire hierarchy. The detail information of a selected cluster, shown as a yellow highlight in the upper left, appears below the overview, together with the gene names and the other dendrogram at the lower right

expressed genes to the same cluster, while affinity propagation and the hierarchical clustering algorithm created separate clusters (highlighted in yellow and orange respectively in 2.29.

In [48], author designed an interactive tool for exploration of hierarchical clusters in multidimensional data but the tool does not addresses challenge of describing the structure of clusters in terms of point distribution and spread. Also the resultant visualization fails to preserve and present the spatial distribution of the data points in cluster.

The underlying idea in these systems is to develop a key visualization technique and customized interactions for it. These interactions play a major role in ability of user to understand the visualization and perform exploratory analysis. PEARLS follows the similar approach and includes a key visualization technique and a set of unique interactions.

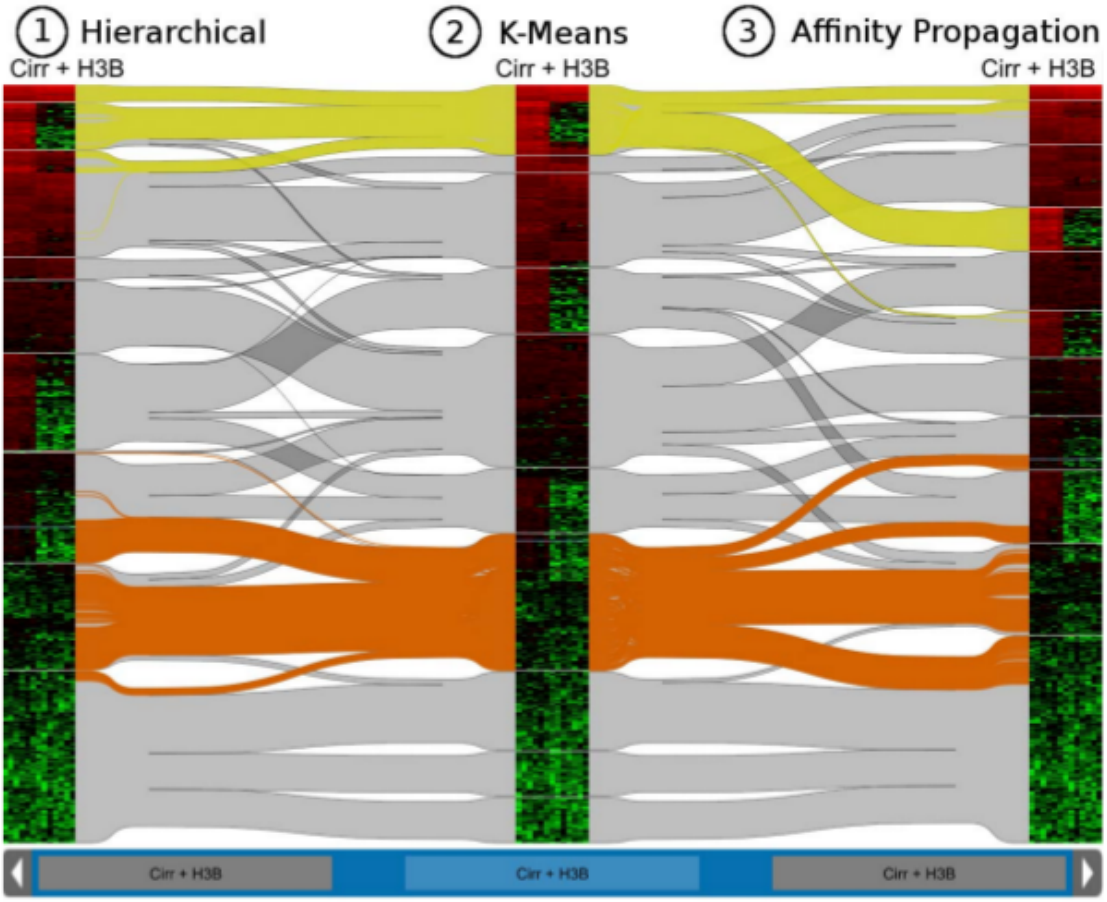


Figure 2.29: Comparison of hierarchical clustering, k-mean and affinity propagation algorithm, each was run on a dataset with 1800 records

Chapter 3

Overview of PEARLS

In this chapter we provide an overview of PEARLS. We describe various algorithms used in generating PEARL plots and underlying concepts for these algorithms. The process of obtaining the visualization of a set of clusters present in the data set, by building the shape of a high-dimensional cluster, is accomplished by three modules : Partitioning, 3-D Pearl Shape Identification and Pearl Placement. Pearls are stored in the PDAS(Pearls Data Abstraction Structure) data structure (section 4.2.1) and then displayed on screen space. Various interaction techniques (section 4.1) allow users to control the logic used to generate the visualization. Figure 3.1 shows a block diagram of the modules of PEARLS.

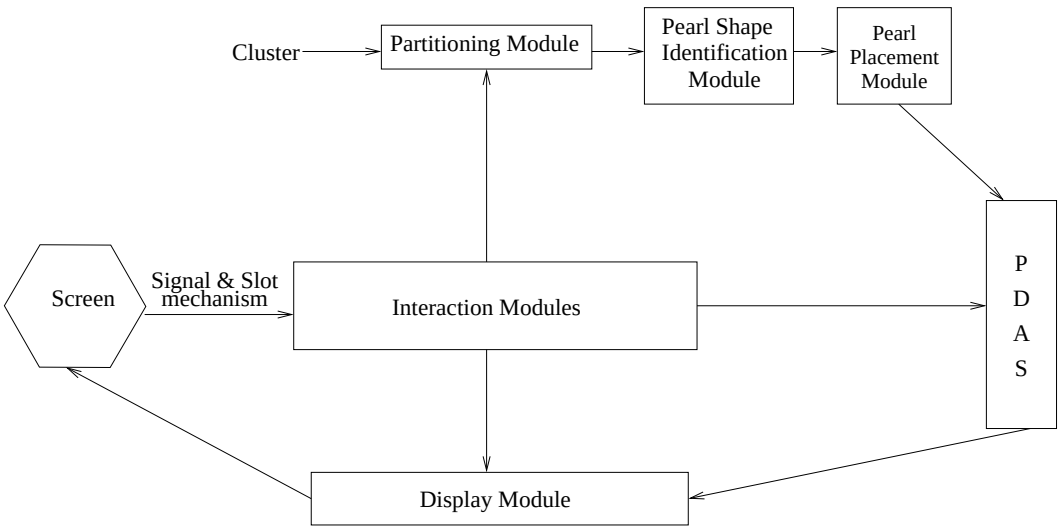


Figure 3.1: Module diagram of PEARLS

3.1 Cluster Division

Let X be a data set of d-dimensional points ($X \subset R^d$). Let the set of clusters (group of points which are close to each other) be represented as $C = \{C_1, C_2, ..., C_c\}$, where c is the number of clusters identified from the data set. Given a cluster $C_i (C_i \subseteq X)$, this step divides the cluster into a set of non-overlapping subsets of points. Each subset of the cluster is referred to as a *pearl*. The points in a pearl should be close to each other so that the shape-fitting for the pearl could be obtained accurately through a mathematical framework.

3.2 Pearl Shape Identification

In this module, a shape is obtained for every pearl using pearl shape identification algorithm. Best fitting Lp norm shape is assigned to a pearl by comparing distances between centroid of the pearl and the farthest point from centroid found using various Lp-norm distance measures. Rules for shape identification are given in [46].

By executing the best Pearl Shape Identification Algorithm (Algorithm 1) on each pearl, we get the shapes of pearls.

we have $P = \{0.25, 0.5, 1, 2, \infty\}$, as set of distinct values of p. While we can choose P to be any subset of values between 0 and ∞, the values we have choosen represent simple, commonly used shapes whose higher dimension can be understood by a user easily. In three dimensions .25 and .5 represent plus shape, 1 represent rhombus, 2 represents sphere and ∞ represents cube. Figure 3.2 shows the span and structure of these *Lp* norm shapes in 2 Dimensions

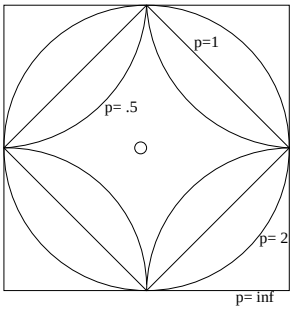


Figure 3.2: 2-D shapes for various Lp norms

In the algorithm 1, in line 2, farthest ten percent points are removed to eliminate shape distortion by outliers. In the while loop of line 3 - 8, for each p, the farthest point (f_p) in the pearl and its distance (d_p) from pearl centroid is computed. A tuple of three values $\{p, d_p, f_p\}$ is computed and maintained

for each p in the list T. From the list of the tuples, the various Lp -norm envelopes are then analyzed. Line 9 - 12 initialize minimum volume, $best_p$ and $best_r$ variables. In the loop of lines 13 - 18, volume is computed for all the tuples in T and the Lp norm shape with minimum volume is selected. Volume is computed by taking product of volume constant computed using Algorithm 2 with the radius of Lp norm shape raised to the power equal to dimensionality.

Algorithm 2 computes volume of i dimensional Lp norm shape by stacking up the i-1 dimensional shapes, each of which has a radius $(1-s^p)$, s ranging -1 to 1. The non recursive solution to this equation has been found by Wang in [49] as $Vc[d] = 2^d \frac{\Gamma(1+\frac{1}{p})^d}{\Gamma(\frac{1}{p}*d+1)}$ but we use recursive version since a recursive solution can be easily computed by a computer.

Since the corresponding Lp norms are known, we use 3-D shapes of these high dimensional norm envelopes i.e, use corresponding Lp norm envelope for 3-D space, to represent the d-dimensional envelope.

Algorithm 1 Pearl Shape Identification Algorithm

1: **Input** : A pearl, a list of Lp norms P
2: Farthest 10% points are removed from pearl to eliminate shape distortion by outliers.
3: **while** $P \neq \phi$ **do**
4: $p_{max} = Largest\_p\_in(P)$
5: (d_{pmax}, f_{pmax}) = $Farthest\ Distance, Farthest\ point\ from\ pearl\ Center\ using\ p_{max}\ Lp\ norm$
6: T.push($(p_{max}, d_{pmax}, f_{pmax})$)
7: P = P - p_{max}
8: **end while**
9: $t_1(p_1, f_1, r_1) = Get\ Tuple\ With\ Largest p(T) fromT$
10: $T = T - t_1$
11: $min\_volume = Vc_1 * r_1^d$
12: $best_p = p_1, best_r = r_1$
13: **while** all t $\in$ T considered, in decreasing order of p **do**
14: $t_1(p_1, f_1, r_1) = Get\ Tuple\ With\ Largest p(T)$
15: **if** $(Vc_1 * r_1^d < min\_volume)$ {where d is dimensionality} **then**
16: $min\_volume = Vc_1 * r_1^d$
17: $best_p = p_1, best_r = r_1$
18: **end if**
19: **end while**
20: **Output: Best fitting** Lp **norm** $best_p$ **and radius** $best_r$

We give an example of how Vc for a shape with Lp norm 2 and dimensions 4 is computed using Algorithm 2.

1. Vc[0]=1
2. Vc[1]= Vc[0] * $\int_{-1}^{1}(1-s^2)^{0/2}ds$ = 2

Algorithm 2 Volume Constant (Vc) Computation

1: **Input**: d (dimensionality) , p (l-p norm shape)
2: Vc[0]=1 {because for all l-p norm shapes, volume constant in 0-dimensions is 1}
3: **for** i between 1 to d **do**
4: $Vc[i] = Vc[i-1] * \int_{-1}^{1}(1-s^p)^{(i-1)/p}\mathrm{d}s$
5: **end for**
6: return(V[d])
7: {Where V[d] denotes volume constant for dimensionality d }
8: **Output**: Volume constant V_c for given p and d

3. Vc[2]= Vc[1] * $\int_{-1}^{1}(1-s^2)^{1/2}\mathrm{d}s$ = 3.141

4. Vc[3]= Vc[2] * $\int_{-1}^{1}(1-s^2)^{2/2}\mathrm{d}s$ = 4.187

5. Vc[4]= Vc[3] * $\int_{-1}^{1}(1-s^2)^{3/2}\mathrm{d}s$ = 4.933

Similarly Vc[4] for Lp norm 1 is 0.666, for Lp norm .5 is 0.00635 and for Lp norm infinity(100) is 16.015.

Table 3.1 shows example of pearl shape identification algorithm for a pearl with 5 points. Lp norm shape rhombus is chosen for this pearl as *Max_Lp_Vol* is lowest with Lp norm value as 1.

Table 3.1: Example of Pearl Shape Identification Algorithm

P1	2	-2	4	5		
P2	3	2	1	4		
P3	1	3	2	2		
P4	4	3	-1	4		
P5	1	5	4	1		
Pearl Centroid	2.2	2.2	2	3.2		
Lp Norm Values			0.5	1	2	100
Lp-Distance(P1,centroid)			27.58	8.2	**4.99**	**4.2**
Lp-Distance(P2,centroid)			10.47	2.8	1.52	1
Lp-Distance(P3,centroid)			9.51	3.2	1.87	1.20
Lp-Distance(P4,centroid)			23.64	6.4	3.67	3
Lp-Distance(P5,centroid)			**32.10**	**8.2**	4.25	2.8
Max_Lp_dist			**32.10**	**8.2**	**4.99**	**4.2**
Vc[4]			0.00635	0.666	4.933	16.015
$Max\_Lp\_Vol_{(Vc[4]*Max\_Lp\_dist^4)}$			6747.14	**3011.13**	3063.42	4983.38
Farthest Points			P5	P5	P1	P1
Shape selected			**rhombus**	◆		

3.3 Shape Composition

3-D Pearls Plot

Till now, we have a set of pearls for a cluster. We now generate a 3-D plot in which pearls are placed. The origin of this plot maps to the cluster centroid. The range of axes are defined by the original distance between pearl centers and cluster centroid and radius of pearls.

Sectors in 3-D Pearls Plot

The orthogonal axes divisions of the original d dimensional space are represented as sectors in the x-y plane of 3-D plot. By orthogonal axes divisions we refer to the division of space by the orthogonal axes. For example, in 4-D the space is divided into 16 parts by the w, x, y and z axes. Some examples of parts are [+w,+x,+y,+z], [-w,-x,+y,+z], [+w,-x,-y,-z] etc. So, in 3-D,we divide the space into 16 parts on x and y axes. Given a d-dimensional space, then the number of axes divisions of the space is 2^d and correspondingly, the 3-D plot is divided into 2^d sectors with each sector angle being $\frac{2\pi}{2^d}$. The sectors are numbered from 0 to 2^d starting from +x-axis in the Pearls Plot to 360° (anti-clockwise); so the binary representation of an integer between 0 and 2^d can map to a distinct axes division. While constructing the overall shape of the clusters, pearls lying in various axes-divisions are plotted in the corresponding sector in the 3-D Pearls Plot. The algorithm to map the d-dimensional pearl onto the 3-D Pearls Plot is Algorithm 3.

Pearl Placement Algorithm (Algorithm 3) is used to locate position of pearls on 3-D canvas. For loop in line 6-13 finds the sector angle which determines the sector of pearl. Elevation angle of pearl is found in line 14. Line 15-19 find the position of shape using its radius, sector angle and elevation angle. The closest pearl to the centroid of the cluster is placed first near $BO(0,0)$, at a distance $d(C_i, B_{ic})$. The shape of the pearl is identified by the corresponding best-fit Lp norm described earlier in Algorithm 1. Pearls are placed using the sector angle and the distance of the bead to the centroid of the cluster.

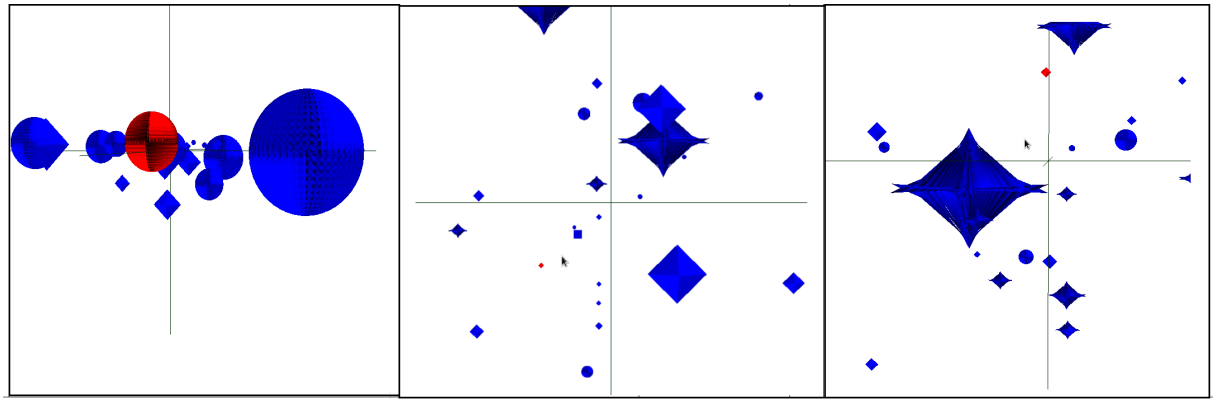


Figure 3.3: PEARLS: Image (a) shows pearls from Baseball hall of fame dataset, Image (b) and (c) show pearls from Singapore real estate dataset

Algorithm 3 Pearl Placement Algorithm

1: **Input**: Cluster C_i, set of pearls P_i
2: Identify the closest pearl to the centroid of the cluster, say P_{ic}.
3: Compute distance of pearl to cluster centroid, $d(C_i, P_{ic})$
4: Obtain radius of the pearl, r_{ic} such that ratio of volume of original n-D shape and 3-D shape remains a constant across all pearls.
5: Obtain the position of the pearl in axes-division of space as:
6: **for** each dimension **do**
7: **if** $C(P_{ic})'s$ d-dimension value is greater than $C(C_i)'s$ d-dimension value **then**
8: set the bit corresponding to d-dimension to 1
9: **else**
10: else the bit corresponding to d-dimension to 0
11: **end if**
12: **end for**
13: Based on the integer value of the bit-vector (say i), the sector angle θ is calculated as $2\pi * i/2^d$.
14: The elevation angle ϕ is calculated as angle between pearl centroid and the unit vector in dimension with maximum standard deviation.
15: Obtain the Lp norm information of pearl P_{ic}.
16: Based on the value of p, get the corresponding 3-D shape and set the radius of the 3-D shape as r_{ic}.
17: The position of pearl (b_x, b_y) is set as
18: $b_x = d(C_i, P_{ic})$ * cos(θ)* sin(ϕ)
$b_y = d(C_i, P_{ic})$ * sin(θ)* sin(ϕ)
$b_z = (C_i, P_{ic}) * cos(\phi)$
19: Plot pearl at $((b_x, b_y, b_z)$.

3.4 Differences between PEARLS and BEADS

As mentioned in chapter 1 PEARLS is based on some of the ideas presented in BEADS [47],[46]. Figure 3.4 shows module diagram of BEADS. Partitioning module of PEARLS and BEADS are similar. Bead shape identification module of BEADS is different from PEARL shape identification module of PEARLS. Instead of computing actual volumes of high dimensional Lp norm shapes to find the best fit, BEADS computes the volume by assuming that ratio of volume of Lp norm shapes of same radius remains same as dimensionality increases. We found that this assumption does not holds when dimensionality increases beyond 4. In the Bead placement module, BEADS places the beads on a 2D beads plot by using only their distance from cluster centroid and their quadrant in high dimensional space. Algorithm 4 shows the Bead Shape identification algorithm used by BEADS and Algorithm 5 shows Bead Placement Algorithm used by BEADS.

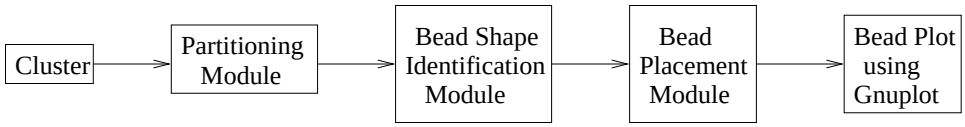


Figure 3.4: Module Diagram of BEADS

Algorithm 4 Bead Shape Identification Algorithm

Input : A bead, a list of Lp norms P
Farthest 10% points are removed from bead to eliminate shape distortion by outliers.
while $P \neq \phi$ **do**
  $p_{max} = Largest\_p\_in(P)$
  (d_{pmax}, f_{pmax}) = $Farthest\ Distance, Farthest\ point\ from\ bead\ Center\ using\ p_{max}\ Lp\ norm$
  T.push($(p_{max}, d_{pmax}, f_{pmax})$)
  P = P - p_{max}
end while
while all t ∈ T considered, in decreasing order of p **do**
  $t_1(p_1, f_1, r_1) = Get\ Tuple\ With\ Largest p(T)$
  $T = T - t_{t1}$
  $t_2(p_2, f_2, r_2) = Get\ Tuple\ With\ Largest p(T)$
  if $(r_2 < \alpha * r_1)$ **then**
    $best_p = p_2, t_1 = t_2$, goto step 12 in this loop
  else
    $best_p = p_1$ and quit
  end if
end while

3.5 Outline

There are two parts to this work

1. **Visualization technique based on pearls** (covered in sections 3.1, 3.2, 3.3, 4.1 and 4.2). In sections 3.1,3.2 and 3.3 we describe algorithms to generate pearl plot from data cluster. In section 4.1 we describe interaction techniques implemented in PEARLS toolkit and in section 4.2 we give various implementation details.

2. **Exploratory scenarios to use PEARLS for data exploration and analytics** (covered in sections of chapter 5 and chapter 6). In sections 5.1,5.2 and 5.3 we describe how PEARLS can be used for concept search, concept description and exploring intra point relationships respectively. In section 6.1 we present case studies on real life datasets.

Algorithm 5 Bead Placement Algorithm

Input: Cluster C_i, set of beads B_i
Identify the closest bead to the centroid of the cluster, say B_{ic}.
Compute distance of bead to cluster centroid, $d(C_i, B_{ic})$
Obtain radius of the bead, r_{ic}.
Obtain the position of the pearl in axes-division of space as:
for each dimension **do**
  if $C(B_{ic})'s$ d-dimension value is greater than $C(C_i)'s$ d-dimension value **then**
    set the bit corresponding to d-dimension to 1
  else
    else the bit corresponding to d-dimension to 0
  end if
end for
Based on the integer value of the bit-vector (say i), the sector angle θ is calculated as $2\pi * i/2^d$.
Place the bead in the corresponding sector in the 2-D plot as:
$b_x = d(C_i, B_{ic})$ * cos(θ)
$b_y = d(C_i, B_{ic})$ * sin(θ)
Plot the 2-D shape of the bead at (b_x, b_y).

Chapter 4

PEARLS Toolkit

4.1 Interaction Functionality

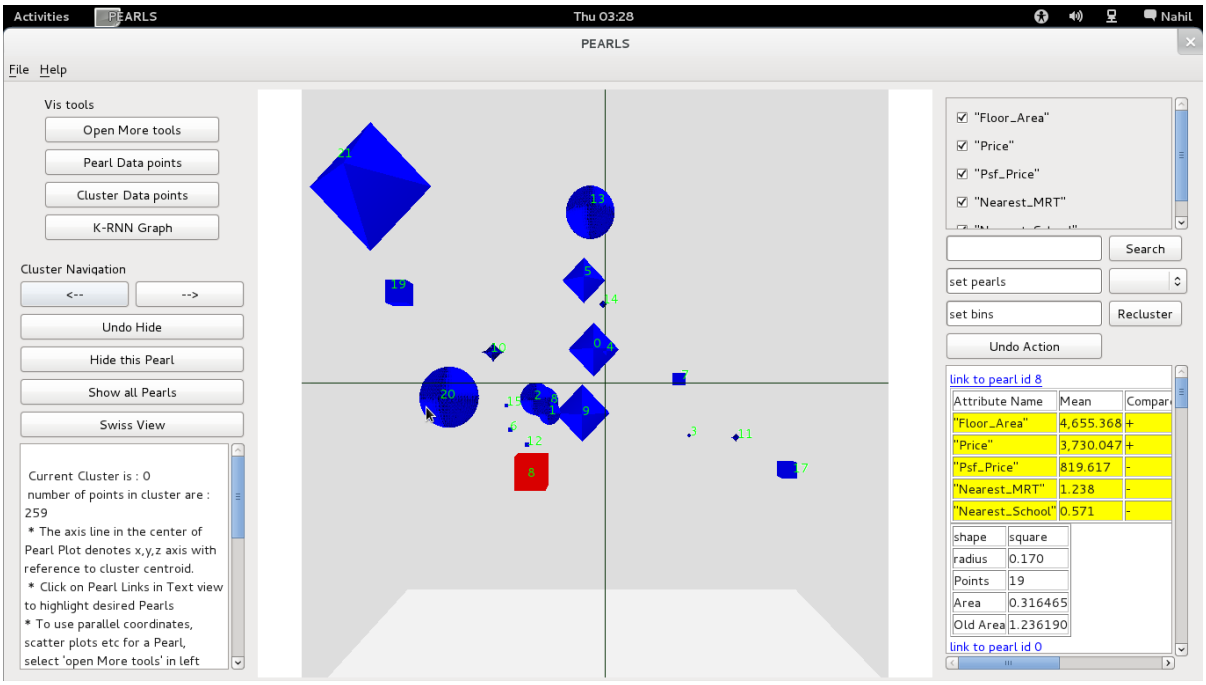


Figure 4.1: Screenshot of PEARLS System

PEARLS is designed as an interactive visualization tool [Figure 4.1]. It has a core visualization technique and a set of interaction techniques specifically designed for pearl visualization technique. PEARLS aims to give a user enough flexibility to control the scope and flow of analysis.

4.1.1 Swiss Cheese

Brushing technique is implemented as Swiss View in PEARLS system. Brushing operation refers to ability to interactively select subsets of data and perform various operations on them. The principles of Brushing were first explained by Becker and Cleveland [3].

In the context of PEARLS, every pearl is a subset of data. As the user navigates across pearl plot and explores various pearls, he/she can highlight and delete various pearls according to his/her interest. This helps in pruning the unnecessary data and focusing on analyzing the data points of interest in contrast to other points.

A user has an option of choosing between a simple view where removed pearls are not completely invisible or a specially designed swiss cheese view which is shown in Figure 4.2. In the swiss cheese view, a transparent bounding box represents the complete range of dataset and the hidden pearls are represented by semi transparent shapes.

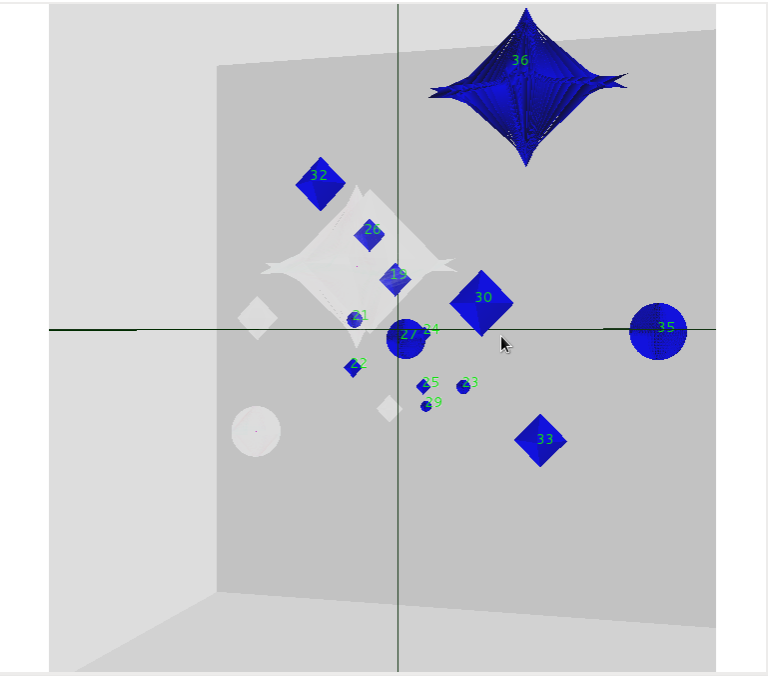


Figure 4.2: Swiss Cheese View

4.1.2 Attribute filtering

PEARLS allows users to filter the set of attributes used for cluster division to generate pearls. By default, all data attributes are used during cluster division. For multidimensional datasets with many

attributes, users can apply filter to focus on a subset of the attribute space. This allows users to explore intra point relationships in various subsets of dimensions.

4.1.3 Data Dimension

We introduce a user-driven data dimension approach where user chooses a dimension(d) to act as data dimension. The data points in cluster are binned according to their value in the data dimension. In current implementation, bins can be created in two ways: either to ensure that every bin contains equal number of data points or ensuring that bin span across equal range in d(data dimension). PEARLS algorithm is run over every bin. PEARLS can be extended to use various techniques to divide data into bins. In the pearl composition phase, the height of screen is equally divided in these bins. Position of a pearl is determined by using algorithm 6.

Algorithm 6 Data Dimension Approach

Input: Distance between Pearl's center (P_{ic}) and Cluster centroid(C_i), sector angle (θ); both were determined in Algorithm 3
$b_x = d(C_i, P_{ic}) * cos(\theta)$
$b_y = d(C_i, P_{ic}) * sin(\theta)$
$b_z =$ Value of P_{ic} in data dimension
Output: (b_x, b_y, b_z), position of pearl p

4.1.4 Reclustering

Re-clustering or regenerating Pearls is done when a user changes number of pearls in which a cluster must be divided or applies attribute filtering or data dimension technique. Koschke [30] said that "the user needs the ability to control the logic used to produce the visualization in order to speed the process of trying different combinations of techniques". Combination of Swiss Cheese View, Data Dimension, attribute filtering and changing number of pearls give users precisely this ability to control and ability to comprehend data cluster. Case1 in section 6.1.1 presents an example of use of these techniques to search for points forming a predefined concept.

4.1.5 Detail View Techniques

In PEARLS, there is no one to one mapping between a data-point and a visualization object. Hence, PEARLS suffers from the problem of level of detail (LOD) culling. However, every visualization object represents a set of data points which are already aggregated by clustering, so in-spite of LOD culling, PEARLS successfully visualizes trends within a cluster.

Nevertheless, for a comprehensive analysis, it is necessary to be able to explore the individual data records. Accordingly, following Shneiderman's overview first, zoom and filter, details-on-demand principle [43] PEARLS provides interactive detail views for individual pearls. Individual pearls can be explored using following detail views in PEARLS: Scatter plots [11], Star Glyphs, Parallel Coordinates [22], Dimensional Stacking [31], Pixel-oriented Display [28] and k-RNN Graphs. Scatter plots, Star Glyphs, Parallel Coordinates, Dimensional Stacking and Pixel-oriented Display are implemented by using parts of Xmdv toolkit [50].

k-RNN Graphs For every point, all the points from which edges are directed towards it (in a k nearest neighbor graph) belong to its k-reverse neighbor set. The *k-RNN* set of point p gives the set of points that consider p as their k-nearest, for a given value of k. If the point p has higher number of *k-RNNs* than another point q, then p has a denser neighborhood than q. k-reverse nearest neighbor graph helps in exploring and understanding the distribution of points inside a particular pearl.

Complexity: Let the number of points in pearl be M.

1. Complexity of building the distance matrix is $O(M^2)$, complexity of sorting the distance matrix is $O(M^2 \log M)$ and complexity of building k-RNN graph from sorted distance matrix is $O(k * M)$

2. Total complexity = $O(M^2 \log M) + O(M^2) + O(k * M) = O(M^2 \log M)$

Graphviz library[13] is used to render the graph from this adjacency matrix.

4.1.6 Point Search Technique

For many datasets, individual points have specific names, for example in a sports dataset every data point may represent a unique player name. In such scenarios, it is highly desirable to have an interface which allows users to search for points of interest by name. Point search feature in PEARLS allows user to search for particular points, locate the pearl in which they lie and then perform analysis as required. If a user knows an interesting data point, user can find its pearl, analyze it using detail view techniques described above and check whether the complete pearl shares the same interesting features as that of a particular data point.

Point Search has been implemented using find() function of C++ string library. A *point_pearl_index* is stored for every state in cluster state index. For every point this index has a corresponding pearl id.

Complexity We use string::find() operation in c++ in PEARLS implementation. string::find() operation has a complexity of O(m)*O(n); where m is number of characters in search string and n is number of characters in point name. In worst case, if search string is not present, all points names must be

Algorithm 7 Point Search

Input: A search string s
for every cluster c in dataset **do**
  for every point p in cluster c **do**
    if isTrue(p.find(s)) **then**
      return *pearl_index* corresponding to p from current *point_pearl_index* mapping for cluster c
    end if
  end for
end for
Output: Index of the pearl in which point is found.

searched. So worst case complexity is O(p)*O(n)*O(M); where p is number of points, and M is number of characters in largest point name.

4.1.7 Views

To satisfy different users, a visualization software needs multiple integrated views. Views are primarily divided into graphical and textual views. PEARLS, apart from the central graphical view, has a textual view in right side which is interactive in nature and is closely integrated to graphical view. Two other textual views are provided to look at data points of individual pearls and whole cluster in tabular format.

The textual view in right side has a information box for every pearl which contains its id, mean value and ranges in various dimensions, a +/- sign comparing mean value in dimension to cluster centroid, number of points in pearl and shape, radius and area of pearl. A hyperlink is provided for every pearl id in the textual view. A user can click on the link and the pearl will be highlighted in the graphical view. Textual view also highlights the dimension selected as data dimension in Green and dimensions filtered by using attribute filtering technique in Yellow color.

Textual View provides a comprehensive summary of pearls of cluster and helps to locate pearls which are overshadowed by other pearls or are too small to be noticed. Small pearls which are have a large point to volume ratio represent dense regions in cluster and may represent interesting concepts.

4.2 Implementation

PEARLS toolkit has been implemented in a modular manner. The partition module (Section 3.1), pearl shape identification module (Section 3.2) and pearl placement module (Section 3.3) are run on the cluster data points and the results are stored in a Pearl Data Abstraction Structure. A display module

renders these pearls on the screen. Using signals and slots technique of Qt, various interactions result in calling of various interaction modules. These interaction modules either interact with partitioning module, Pearl Data Abstraction Structure (Section 4.2.1) or display module.

4.2.1 Pearl Data Abstraction Structure

Undo operations are a very important component of any visualization toolkit. In PEARLS, performing an undo of re-clustering operation (see section 4.1.4) is costly. Re-clustering happens when number of pearls in which cluster should be divided is changed or when attribute filtering or data dimension technique is used.

Our data structure called PDAS (Pearls Data Abstraction Structure) is designed to optimize undo operation and memory usage.

Primitive data structures in our system are

1. Pearl structure (Figure 4.3) : Every pearl data structure contains some points and data like pearl's location and size. The points in a pearl originally belong to a cluster and hence only a mapping(pearl_point_index) to cluster points is stored. A group of pearls represents a data cluster.

2. A mapping between points and pearls (point_pearl_index) to support point search operation (as described in section 4.1.6).

After every re-clustering action, pearls of a cluster are regenerated and old pearls are required only when user performs undo operation on re-clustering. Cluster points remain same after re-clustering operation and they should not be stored again in memory. Since pearls only store mapping to cluster points we can achieve this. The point_pearl_index is generated again after re-clustering and old point_pearl_index is used only if an undo operation is performed.

Undo action can be efficiently supported by a LIFO(Last in First Out) based data structure. PDAS (Figure 4.3) makes use of a stack data structure called *cluster_state_stack* which stores cluster states. After every re-clustering operation a new state is created in the stack. A state primarily consists of a list of pearl data structures and a mapping *point_pearl_index*. The original cluster data points are stored only once in the memory to minimize the memory consumption.

Benefits of PDAS Storing *pearl_point_index* instead of actual points inside pearls saves huge memory overhead. For a dataset with *n* points and *d* dimensions, if we perform *m* operations which require re-clustering, a naive implementation with points inside pearls will require O(n*d*m) memory. Storing

pearl_point_index requires only O($n * m$) memory. Storing previous states in stack makes undo a constant time operation. If previous states were not stored, undo operations will require running clustering algorithm over cluster with old parameters to partition the data, running pearl shape identification algorithm over every pearl to identify shape and then running pearl placement algorithm to place pearls at their correct positions.

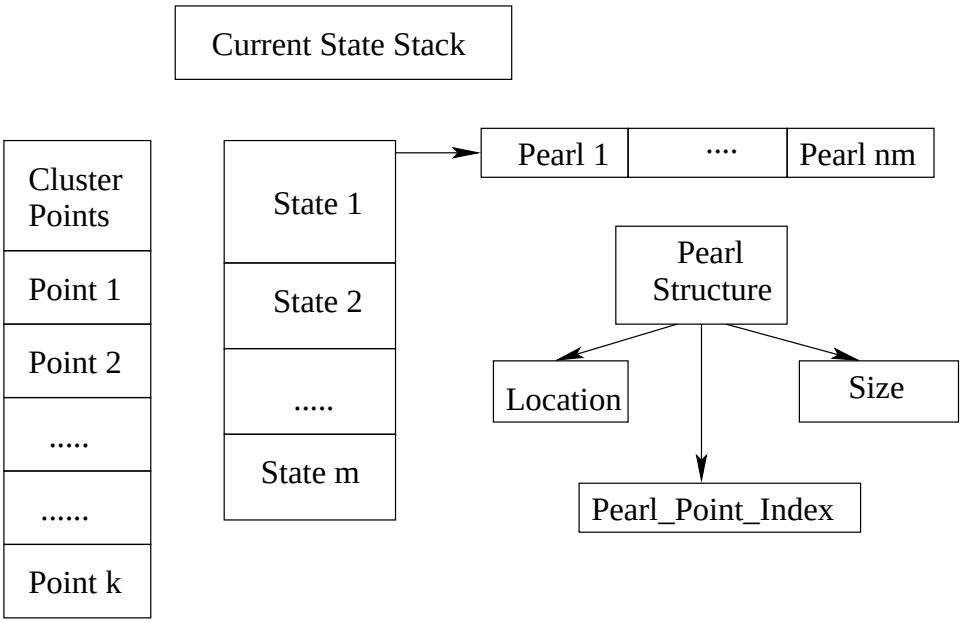


Figure 4.3: Pearls Data Abstraction Structure

4.2.2 Implementation Details

The PEARLS technique is implemented in C++ using QT framework. For rendering, we use the bindings for OpenGL. The images were produced using a published real-life Baseball Hall of fame dataset [12] and a Singapore real estate dataset crawled from *http://www.propertyguru.com.sg/*.

Current PEARLS implementation performs well for datasets with up to 20 dimensions and up to 35000 data records on standard hardware (an Intel Core Duo CPU with 2.2 GHz processor, 1.9 GB RAM and a 19 inch screen with a resolution f 1366x768). There is no hard limit on number of dimensions as well as number of points and the only noticeable impact is on performance of k-mean algorithm to generate pearls. For an eight dimensional dataset with 35000 data records in 15 clusters, computing 25 pearls for every cluster took 54 seconds.

Chapter 5

Cluster and Class Visualization and Exploration

As stated in Han and Kamber [17], *"A concept usually refers to a collection of data such as frequent_buyers, graduate_students, and so on and concept description generates descriptions for characterization and comparison of data. Concept description generates descriptions for characterization and comparison of data."*

PEARLS toolkit is suitable for task of concept search and concept description.

5.1 Concept Search

Given a concept description and a dataset, **Concept Search** can be viewed as finding groups of points from the dataset which fit this concept description. For example: Give a dataset X of Real Estate Unit Information with attributes Price, Area, Psf Price, Nearest School and Nearest MRT Station and a concept description "Find all those data points which are moderately priced, have large area and are close to Schools"; concept search means finding the group of data points which satisfy this concept description.

In most of the concept tasks on real life data clusters, adjectives like 'moderate', 'high', 'low', 'close' etc are loosely defined. This lack of strict definition is a major challenge in Concept Search task. A naive solution for a user is to plot the data points according to every dimension and understand the distribution of data points and then come up with strict mathematical and algebraic equations to describe the concepts. So to accomplish the task given above: a user must plot the data points with respect to Price, remove the data points which are priced highly, then again plot the remaining points with respect to area and remove low area data points and finally plot remaining data points with respect to Nearest School and then prune unwanted points. Doing these steps manually is overwhelmingly

difficult. The complexity further increases manifold if the user wants to change his concept description during intermediate steps or he/she wants to analyze which data points are close to each other.

PEARLS toolkit is suitable for task of concept search as well as concept description. Formal description of a concept query "Given a data cluster X with R ($R_1, R_1, ..., R_n$) dimensions. A concept can be described as a group of points which are t_1 in R_a, t_2 in R_b, t_3 in R_c and so on; where R_a, R_b, R_c $\subset$ ($R_1, ..., R_n$) and t_1,t_2,t_3 etc are adjectives describing values of data points in respective dimensions." Figure 5.1 shows the flowchart for applying interactive techniques to perform concept search. In chapter 6.1 we demonstrate how to perform concept search by case studies.

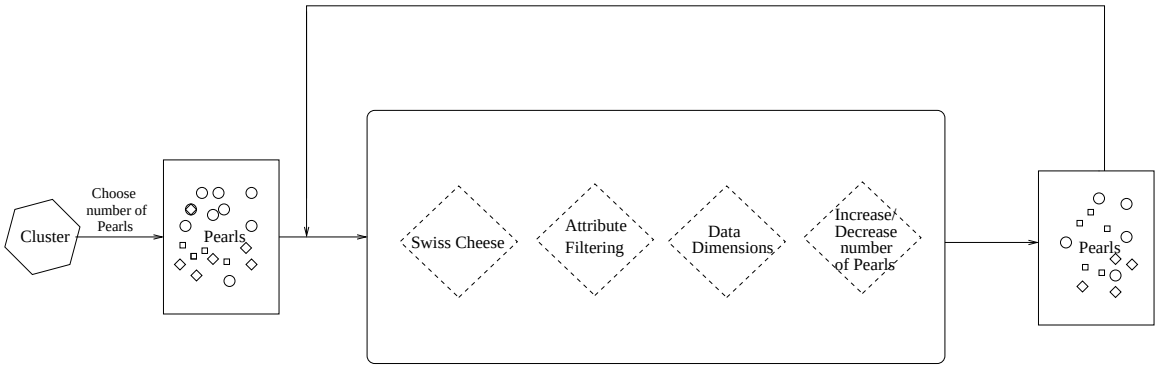


Figure 5.1: Steps for Concept Search and Concept Description

5.2 Concept Description

PEARLS is also suitable for task of concept description. Combination of textual view (Section 4.1.7) and graphical view can be used to describe a cluster in terms of underlying concepts.

Figure 5.2 displays Pearls for a cluster of nutrition dataset and table 5.1 shows textual view for this cluster. The first hand pearls visualization divides the cluster into pearls which represent some interesting concepts which can be used to learn more about cluster and find interesting point groups. After looking at the preliminary Pearls plot a user may wish to look at these cluster points from a different perspective. For example, user might just be interested in vitamin_a, vitamin_c and cholesterol of various food items and might wish to have a concept division on only these attributes. Or user may be interested in a deeper level concept analysis of points lying in a few pearls. Or user may just be interested in more finer division of cluster.

As Han and Kamber state [17], "Users like the ease and flexibility of having data described at different levels of granularity and from different angles."

The interactive techniques of PEARLS are capable of describing the clusters from various angles as well as different levels of granularity. By varying the number of pearls, granularity of description can be changed. By omitting certain dimensions in the Pearl generation, one can do a concept description which is focused only on attributes of interest. By removing uninteresting pearls using swiss cheese view, a user can do a concept description for only interesting points. And by combining all of them "a user can generate a concept description focusing on points and attributes of interest at various granularity levels". Figure 5.1 shows the flowchart for applying interactive techniques to perform concept description.

So to generate concepts which are distinguishable on the basis of calcium, vitamin_a, vitamin_c and cholesterol, a user can use attribute filtering technique and filter the remaining attributes. To analyze points lying in particular pearls user can use swiss cheese view, hide remaining pearls and regenerate the pearls. Figure 5.3 displays Pearls for the cluster after attribute filtering and Table 5.2 shows textual view.

The flowchart for both concept search and description tasks is similar. The major difference is in frequency of use of swiss cheese. In concept search task, after most attribute filtering or data dimension operations, user prunes some pearls which do not satisfy search criteria. While in concept description tasks, there is no predefined criteria or need to prune pearls. The aim is just to describe cluster in terms of concepts at various levels of granularity and from different angles.

We agree that textual view only gives a summary of the pearl with high,low and average values. In a PEARL, points will have values below as well as above the average values. Hence, the pearls generated are susceptible to contamination with not so interesting points to a certain extent. But if average, low and high values are carefully used and a detail analysis is conducted using detail view techniques; such contamination can be minimized.

Table 5.1: Textual View of nutrition cluster (Graphic view in Figure 5.2) : It shows information for nine pearls farthest from cluster centroid. In the right side of table there is a small textual interpretation drawn mostly from tables and from looking at Parallel coordinate plot of pearls in few cases. Description of Pearls close to cluster centroid has been omitted since they have similar values as cluster mean.

Pearl id 57

Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High
calorie	196.182	-	233.704	155.347	251.033
protein(g)	13.102	+	11.304	2.07	20.862
carb(g)	4.615	-	19.436	0.08	9.961
fat(g)	14.053	+	12.473	7.479	19.727
cholesterol(mg)	76.189	+	61.043	18.619	425.956
calcium(mg)	93.48	-	98.664	10	247.351
iron(mg)	1.709	-	1.942	0.04	4.058
vit_a(IU)	445.029	-	1228.936	1	2873.913
vit_c(mg)	3.844	+	3.369	0.2	13.301
shape	plus				
radius	0.168				
Points	40				

Pearl 57 contains food items with carb content considerabely lower than cluster average. The average carbohydrate content in cluster is 19.43 gm but food items in Pearl 57 have carb content ranging from .08 to 9.961 with an average of 4.615

Pearl id 58

Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High
calorie	314.609	+	233.704	264.868	443.978
protein(g)	15.589	+	11.304	4.4	55.502
carb(g)	26.619	+	19.436	20.622	32.899
fat(g)	16.04	+	12.473	13.001	20.202
cholesterol(mg)	74.949	+	61.043	1	572.052
calcium(mg)	296.999	+	98.664	235.261	397.403
iron(mg)	1.427	-	1.942	0.657	3.16
vit_a(IU)	572.861	-	1228.936	194.019	1990.737
vit_c(mg)	1.254	-	3.369	0.1	3.53
shape	plus				
radius	0.082				
Points	15				

Points in pearl 58 are rich source of calcium, have above average carbohydrate, fat and calories. Some of them can have very low protein but remaining have high protein.

Pearl id 59

Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High
calorie	240.758	+	233.704	200.546	282.625
protein(g)	6.565	-	11.304	0.667	11.208
carb(g)	25.812	+	19.436	13.28	36.66
fat(g)	12.494	+	12.473	5.424	16.522
cholesterol(mg)	26.722	-	61.043	1.757	94.997
calcium(mg)	53.615	-	98.664	2.739	172.387
iron(mg)	0.971	-	1.942	0.24	1.737
vit_a(IU)	508.945	-	1228.936	53.765	3226.243
vit_c(mg)	23.892	+	3.369	13.953	55.895
shape	plus				
radius	0.199				
Points	11				

Points in Pearl 59 are rich source of vitamin C and are low in cholesterol. Parallel coordinates plot show that apart from a single point(which has 94.997 mg cholestrol) all others have cholestrol below 38.37 mg.

Pearl id 60

Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High
calorie	246.249	+	233.704	147.991	374.955
protein(g)	15.446	+	11.304	10.329	22.701
carb(g)	11.164	-	19.436	0.28	20.002
fat(g)	15.79	+	12.473	1	31.2
cholesterol(mg)	47.828	-	61.043	15.999	137.01
calcium(mg)	353.411	+	98.664	226.583	711.938
iron(mg)	1.002	-	1.942	0.05	3.701
vit_a(IU)	632.586	-	1228.936	163.96	2166.23
vit_c(mg)	2.431	-	3.369	0.1	9.91
shape	plus				
radius	0.256				
Points	25				

Points in pearl 60 are rich source of calcium.

Pearl id	61					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
calorie	310.762	+	233.704	263.51	352.981	
protein(g)	11.268	-	11.304	0.6	19.122	Points in pearl 61 represent food with high calorific value, high fat content and below average carbohydrate
carb(g)	4.167	-	19.436	1.5	13.4	
fat(g)	27.571	+	12.473	21.487	37.003	
cholesterol(mg)	98.15	+	61.043	4	255.016	
calcium(mg)	47.05	-	98.664	4	196.023	
iron(mg)	3.104	+	1.942	0.03	11.902	
vit_a(IU)	4271.3	+	1228.936	9.999	23777.719	
vit_c(mg)	5.28	+	3.369	0.1	19.899	
shape	rhombus					
radius	0.141					
Points	26					

Pearl id	62					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
calorie	182.214	-	233.704	135.001	276.963	
protein(g)	23.94	+	11.304	10.86	47.678	
carb(g)	2.41	-	19.436	0.03	15.519	
fat(g)	7.907	-	12.473	0.4	18.601	Points in pearl 62 have high protein content.
cholesterol(mg)	320.647	+	61.043	55.005	3100.259	
calcium(mg)	27.292	-	98.664	5.001	180.027	
iron(mg)	4.916	+	1.942	0.6	13.908	
vit_a(IU)	4790.866	+	1228.936	20.003	39058.277	
vit_c(mg)	4.581	+	3.369	0.1	16.4	
shape	circle					
radius	0.211					
Points	52					

Pearl id	63					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
calorie	246.272	+	233.704	194.019	310.319	
protein(g)	5.711	-	11.304	2.4	12.067	
carb(g)	40.631	+	19.436	35.456	45.318	
fat(g)	7.263	-	12.473	2.34	12.901	Points in pearl 63 represent food items with high carbohydrate.
cholesterol(mg)	18.568	-	61.043	0.852	57.517	
calcium(mg)	83.297	-	98.664	2.747	249.025	
iron(mg)	1.51	-	1.942	0.049	4.6	
vit_a(IU)	670.01	-	1228.936	5.999	7521.527	
vit_c(mg)	1.784	-	3.369	0.2	13.599	
shape	rhombus					
radius	0.048					
Points	27					

Pearl id	64					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
calorie	176.563	-	233.704	133.02	272.988	
protein(g)	22.039	+	11.304	15.052	28.416	
carb(g)	3.515	-	19.436	0.8	6.32	Points in Pearl 64 have high protein content and low carbohydrate content.
fat(g)	7.747	-	12.473	1.95	20.542	
cholesterol(mg)	335.928	+	61.043	52.993	562.937	
calcium(mg)	22.188	-	98.664	5.001	91.994	
iron(mg)	15.469	+	1.942	5.109	30.526	
vit_a(IU)	35524.424	+	1228.936	144.004	96015.414	
vit_c(mg)	18.062	+	3.369	0.7	30.399	
shape	rhombus					
radius	0.699					
Points	16					

Pearl id	65					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
calorie	483.214	+	233.704	405.006	564.023	
protein(g)	8.691	-	11.304	1	57.906	
carb(g)	4.259	-	19.436	0.03	7.36	
fat(g)	48.022	+	12.473	31.797	61.392	Points in Pearl 65 have very high fat content and high calorie values.
cholesterol(mg)	60.738	-	61.043	21.425	150.001	
calcium(mg)	40.429	-	98.664	14.999	70.17	
iron(mg)	1.131	-	1.942	0.09	5.499	
vit_a(IU)	619.906	-	1228.936	33.995	3570.94	
vit_c(mg)	1.592	-	3.369	0.1	6.801	
shape	rhombus					
radius	0.095					
Points	14					

Table 5.2: Textual View of nutrition cluster after Attribute filtering (Graphic view in Figure 5.3) : It shows information for pearls farthest from cluster centroid. Attributes removed are calorie, fat, protein, carb and iron. In the right side of table there is a small textual interpretation drawn mostly from tables and from looking at Parallel coordinate plot of pearls in few cases. Description of Pearls close to cluster centroid has been omitted as they are similar in value to cluster centroid.

pearl id	57					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	283.834	+	61.043	105.012	564.023	Points in this pearl have high cholesterol content, low calcium content and high vitamin a content.
calcium(mg)	16.583	-	98.664	5.001	69.99	
vit_a(IU)	13596.831	+	1228.936	8134.79	17490.358	
vit_c(mg)	5.967	+	3.369	1.3	11.4	
shape	square					
radius	0.014					
Points	12					

pearl id	58					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	29.913	-	61.043	12.501	86.993	Points in this pearl have low cholestol, high calcium content varying vitamin A content and above average vitamin c content.
calcium(mg)	230.187	+	98.664	187.523	321.48	
vit_a(IU)	368.974	-	1228.936	74.994	1373.662	
vit_c(mg)	7.726	+	3.369	4.637	13.199	
shape	square					
radius	0.008					
Points	21					

pearl id	59					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	119.297	+	61.043	16.586	564.93	Points in this pearl have low calcium content, varying cholestrol and vitamin a content and very high vitamin c content.
calcium(mg)	55.576	-	98.664	13.998	172.387	
vit_a(IU)	521.744	-	1228.936	13.001	5869.544	
vit_c(mg)	16.066	+	3.369	12.001	22.103	
shape	circle					
radius	0.034					
Points	23					

pearl id	60					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	64.044	+	61.043	8.001	572.052	Points in pearl 60 have varying cholesterol levels, low vitamin a and vitamin c content.
calcium(mg)	317.263	+	98.664	267.003	417.959	
vit_a(IU)	580.873	-	1228.936	163.96	1990.737	
vit_c(mg)	1.834	-	3.369	0.1	8.302	
shape	rhombus					
radius	0.032					
Points	26					

pearl id	61					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	400.597	+	61.043	210.987	515.01	Points in Pearl 61 have very high cholestrol, low calcium, very high vitamin A content and varying Vitamin C content.
calcium(mg)	11.001	-	98.664	5.001	43.006	
vit_a(IU)	30288.833	+	1228.936	23777.719	39908.023	
vit_c(mg)	4.97	+	3.369	0.7	13.7	
shape	circle					
radius	0.03					
Points	10					

pearl id	62					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	160.736	+	61.043	1.757	562.937	Points in Pearl 62 have low calcium content, very high vitamin C content and varying cholesterol and vitamin A content.
calcium(mg)	18.195	-	98.664	2.739	46.873	
vit_a(IU)	6337.521	+	1228.936	112.992	21648.773	
vit_c(mg)	31.171	+	3.369	23.6	55.895	
shape	rhombus					
radius	0.153					
Points	9					

pearl id	63					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	54.867	-	61.043	15.999	93.995	Point in Pearl 63 have low vitamin C content, very high calcium content and varying Vitamin A and cholestrol content.
calcium(mg)	564.954	+	98.664	458.767	711.938	
vit_a(IU)	1183.587	-	1228.936	617.7	2166.23	
vit_c(mg)	1.054	-	3.369	0.1	2.471	
shape	rhombus					
radius	0.033					
Points	5					

pearl id	64					
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High	
cholesterol(mg)	420.788	+	61.043	331.016	511.055	Points in Pearl 64 have high cholesterol content, very low vitamin C content , extremely high vitamin a content and varying vitamin c content.
calcium(mg)	6	-	98.664	5.001	7	
vit_a(IU)	76252.935	+	1228.936	66980.8	96015.414	
vit_c(mg)	14.299	+	3.369	0.7	24.497	
shape	circle					
radius	0.053					
Points	5					

pearl id	65				
Attribute Name	Mean	pearl mean vs cluster mean	Cluster Mean	Low	High
cholesterol(mg)	3055.117	+	61.043	3009.976	3100.259
calcium(mg)	26.003	-	98.664	9	43.006
vit_a(IU)	132.006	-	1228.936	117.018	146.997
vit_c(mg)	10.601	+	3.369	10.5	10.701
shape	circle				
radius	0.003				
Points	2				

Pearl 65 has only two points both of which have extremely high cholesterol content.

5.3 Exploring data point relationships in subsets of Dimensions

PEARLS toolkit is suitable for exploring data point relationships in various subsets of dimensions. Data point relation ship tasks which can be performed using PEARLS can be primarily divided in two types. A user can either search for a data point and explore points similar to it or he/she can browse various pearls and understand various data point groups.

In the first case, when a user wants to look for data points similar to specific data point, user can select dimensions in which similarity is desired using Attribute filtering technique and then generate suitable number of Pearls. User can then search for the data point of interest and find to which Pearl does this point belong to. Other points in this pearl will have values similar to this point in the selected dimensions.

Lou Gehrigwas an American baseball first baseman who played 17 seasons in Major League Baseball (MLB) for the New York Yankees (19231939). Gehrig is chiefly remembered for his prowess as a hitter and his durability, a trait which earned him his nickname ”The Iron Horse”. Gehrig accumulated 1,995 runs batted in (RBIs) in 17 seasons, with a career batting average of .340. Gehrig holds several records like most runs batted-in by a first baseman (184 runs in 1931), most runs scored by a first baseman (167 in 1936) and others. [51]

We want to analyze the Baseball Hall of dataset and find players which are similar to Lou in Runs_scored, Hits, Home_runs, Runs_Batted_in and batting_average. The Baseball dataset cluster has 1340 data points. We do attribute filtering and select only the required 5 attributes. We select number of Pearls as 25 and generate Pearls. Figure 5.4 shows the generated Pearls. We then do a Point Search for Lou and Pearl 24 is highlighted as a result. Pearl 24 is the farthest Pearl from cluster centroid which implies that, for this subset of five attributes, the distance between centroid of Pearl 24 and cluster centroid is maximum. Figure 5.5 shows parallel coordinate detail view for this pearl and Table 5.3 displays the list of player records in this Pearl. After an overview from graphical and textual view, if a user wants to do analysis on individual data points within Pearl, parallel coordinate plot is useful.

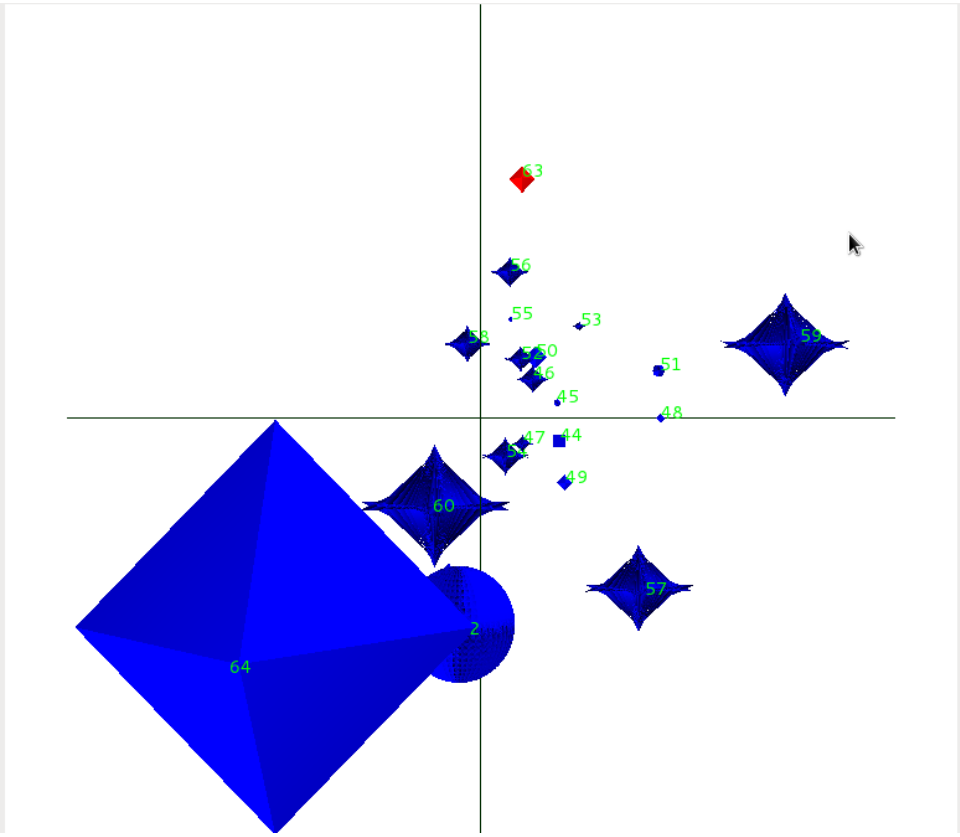


Figure 5.2: Pearls image of cluster 2 of nutrition dataset

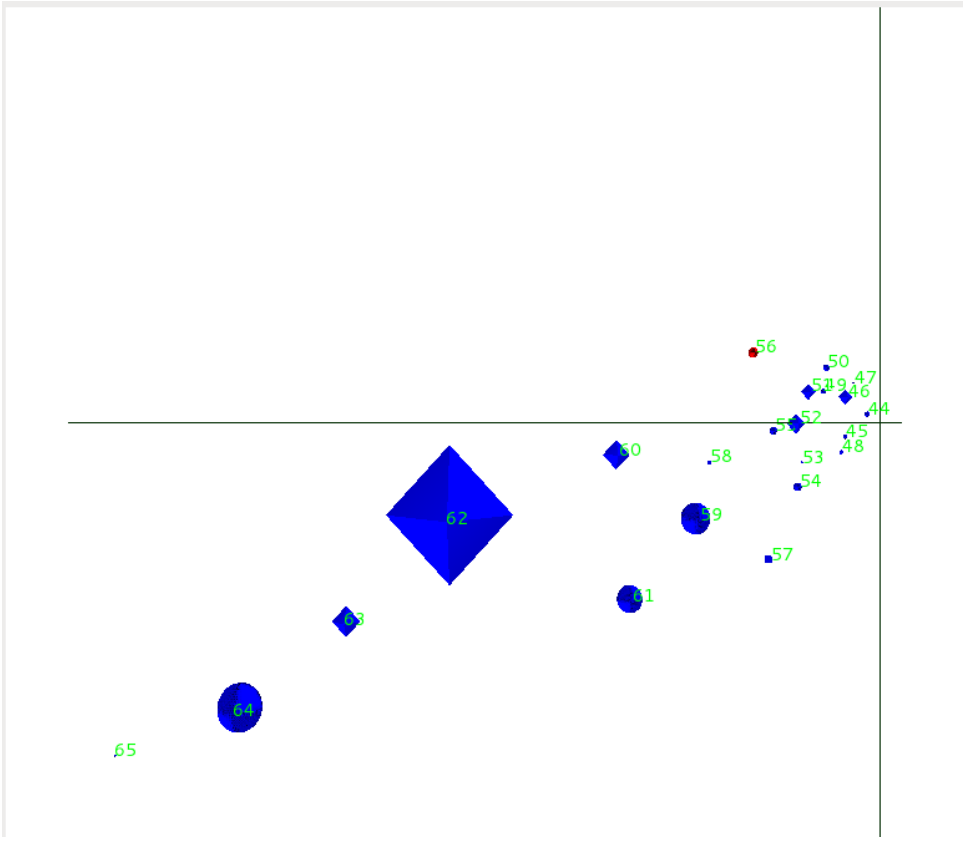


Figure 5.3: Pearls image of cluster 2 of nutrition dataset

Table 5.3: Pearl from baseball dataset : Pearl 24 which contains Player Lou Gehrig

point_names	Runs_Scored	Hits	Home_Runs	Runs_Batted_In	Batting_Average	Walks
LOU_GEHRIG	1888	2721	493	1995	0.34	1508
STAN_MUSIAL	1949	3630	475	1951	0.331	1599
JIMMIE_FOXX	1751	2646	534	1922	0.325	1452
MEL_OTT	1859	2876	511	1860	0.304	1708
TED_WILLIAMS	1798	2654	521	1839	0.344	2019
HANK_AARON	2174	3771	755	2297	0.305	1402
WILLIE_MAYS	2062	3283	660	1903	0.302	1464
FRANK_ROBINSON	1829	2943	586	1812	0.294	1420
BABE_RUTH	2174	2873	714	2213	0.342	2056
CARL_YASTRZEMSKI	1816	3419	452	1844	0.285	1845

In the second case, when user is not looking for points similar to any specific data point, user can specify a subset of dimensions using Attribute filtering technique and generate suitable number of pearls. User can then browse individual pearls using detail view techniques or look at the list of data points in pearls. Such an exploration will enable a user to understand what data points are close to each other in the selected subset of dimensions.

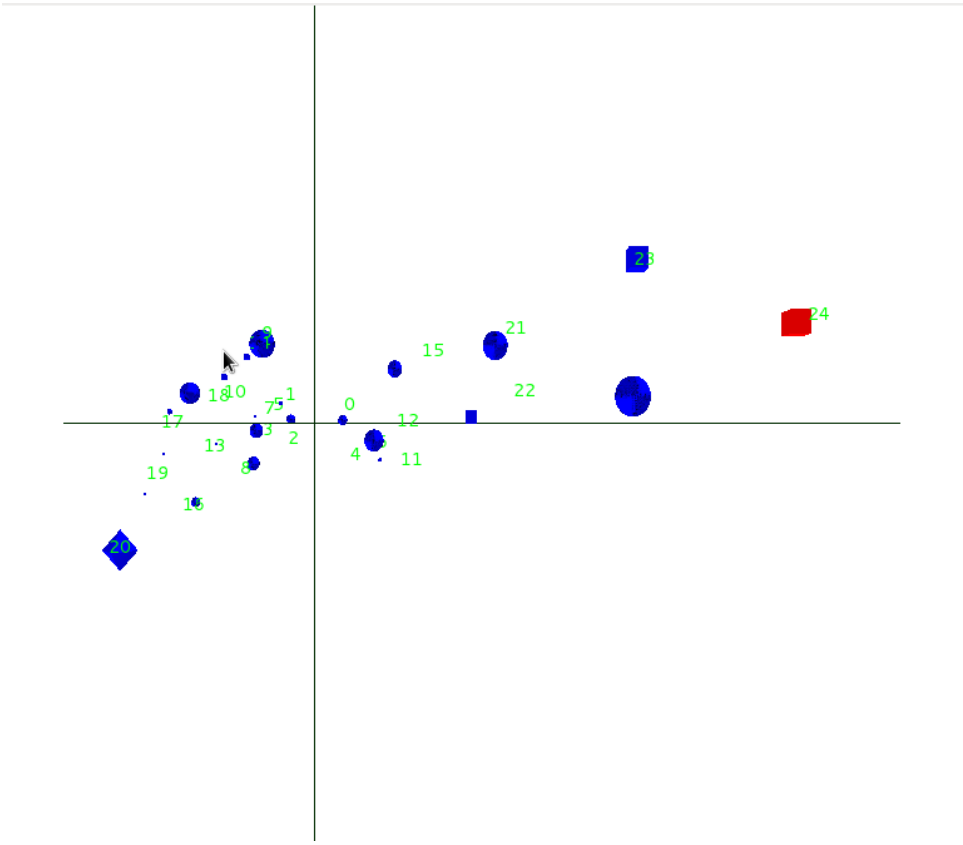


Figure 5.4: Pearls image for Baseball dataset: The highlighted pearl, ie pearl id 24, contains the player Lou Gehrigwas

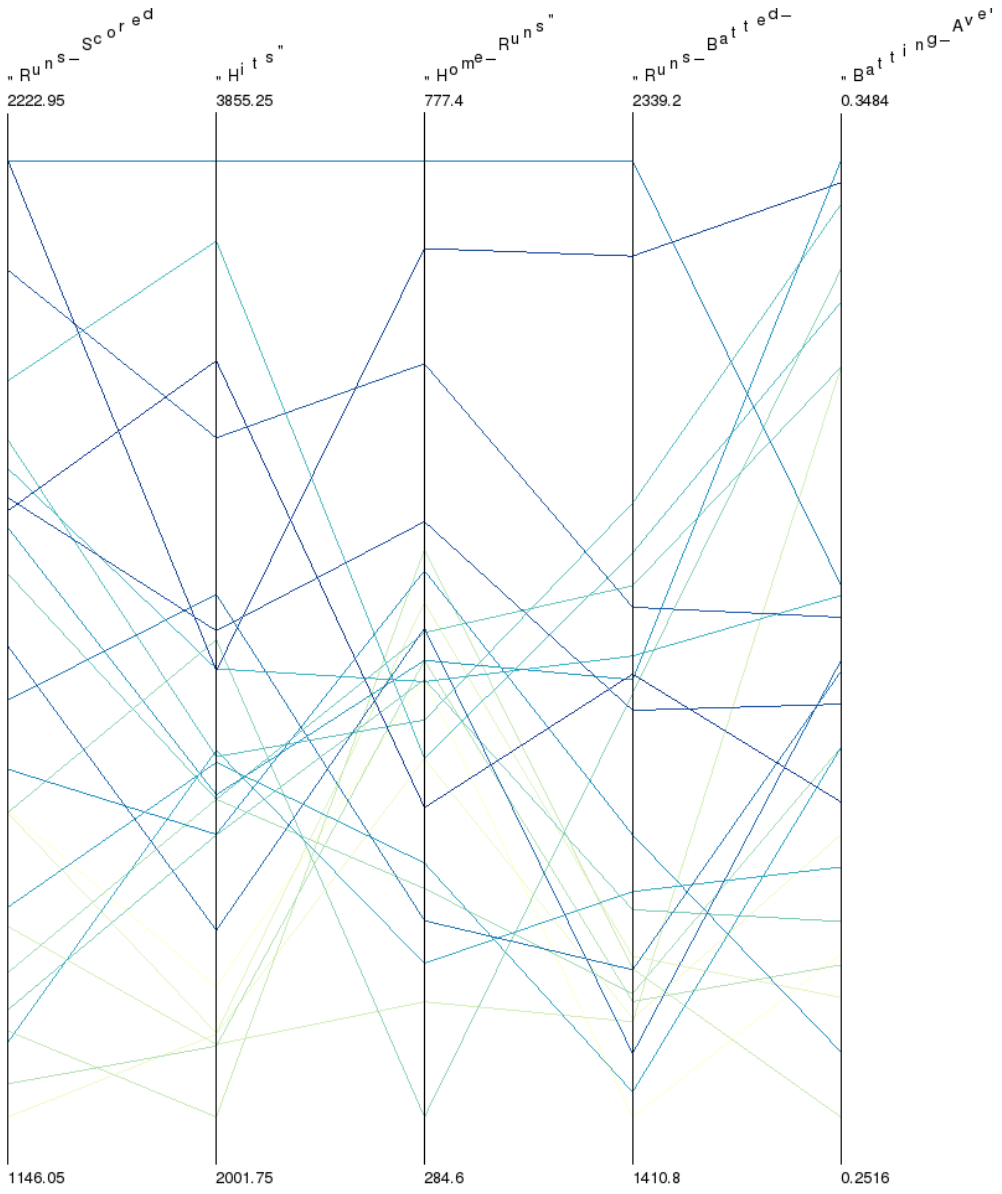


Figure 5.5: Parallel coordinate plot for Pearl 24. This Pearl contains player Lou Gehrigwas and other similar players. This plot shows values for individual points in Pearls in various dimensions. After an overview from graphical and textual view, if a user wants to do analysis on individual data points within Pearl, parallel coordinate plot is useful.

Chapter 6

Case Studies

6.1 Exploratory Data Analysis Case Studies

In this section we perform exploratory data analysis tasks on real life datasets. The first task in on a cluster of Singapore real estate dataset and the second task is on a cluster of Baseball dataset. These tasks are focused on finding data points which represent answers to common questions asked by people looking at these datasets. The first task on Singapore dataset is a question asked by many real estate buyers and the second task is a question which occurs in mind of many sport enthusiasts.

6.1.1 Singapore Dataset

We use the methodology described in section 5.1 in figure 5.1 to perform a task on Singapore real estate dataset. The exact set of operations are described below.

Dataset Description Singapore dataset consists of information about 35,000 real estate homes. Each record has a name and Floor_area, Price, Psf_Price, Nearest_MRT and Nearest_School dimensions. The dataset is divided in 15 clusters using a standard clustering algorithm.(Since real life datasets are sparse all 35,000 records can't be treated as a single cluster) The clustering results are loaded in PEARLS toolkit.

Task Description The task is to find properties which are close to school, are not high priced and have moderate area in cluster 1. Following operations complete this task. Every image in Table 6.3 is result of a single step.

1. select *Nearest_School* as data dimension, specify number of bins as 5, number of pearls as 3-4 pearl per bin and filter attributes other than Price and floor_area.

2. Look at the ranges of bins and prune pearls which lie in region of high school distance.

3. Select *price* as data dimension, specify bins and pearls, prune attributes other than floor_area.

4. Prune pearls which have extremely high valuation prices. From the remaining pearls prune pearls which have points with low floor area.

5. Select floor_area as data dimension and specify number of bins and pearls.

6. Prune pearls which have points with low floor area.

It is to be noted that high and low are determined while exploring the pearls by looking at various ranges and with help of some apriori knowledge. Table 6.1 shows statistical overview of original cluster and Table 6.2 shows statistical overview of remaining pearls after step 6.

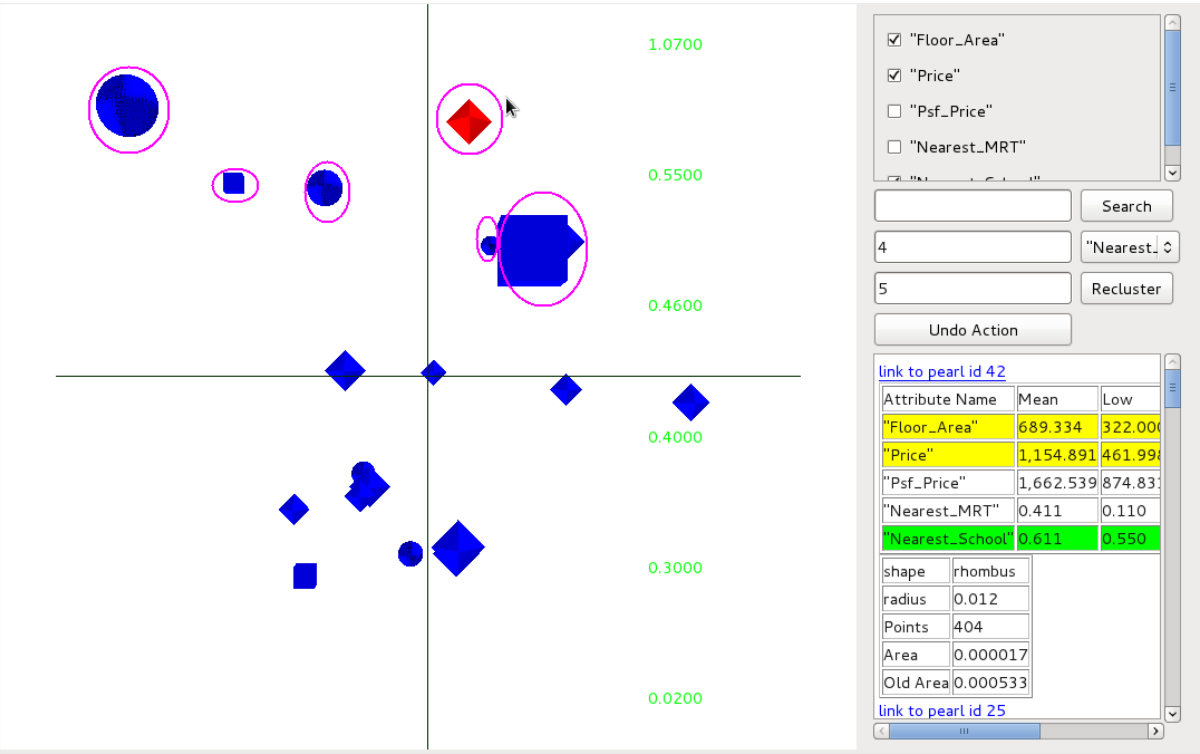
Table 6.1: Original Cluster from Singapore Dataset

Points	5523		
Attribute_Name	Mean	Low	High
Floor_Area($sqft$)	1,364.024	322.000	4,209.000
Price($in\ thousands\ of\ Singapore\ dollars$)	2,881.842	4.097	36,117.548
Nearest_School(km)	0.439	0.020	1.070

Table 6.2: Description of pearls remaining after completing all steps of task 1 ; Task: find properties which are close to school, are not high priced and have moderate area

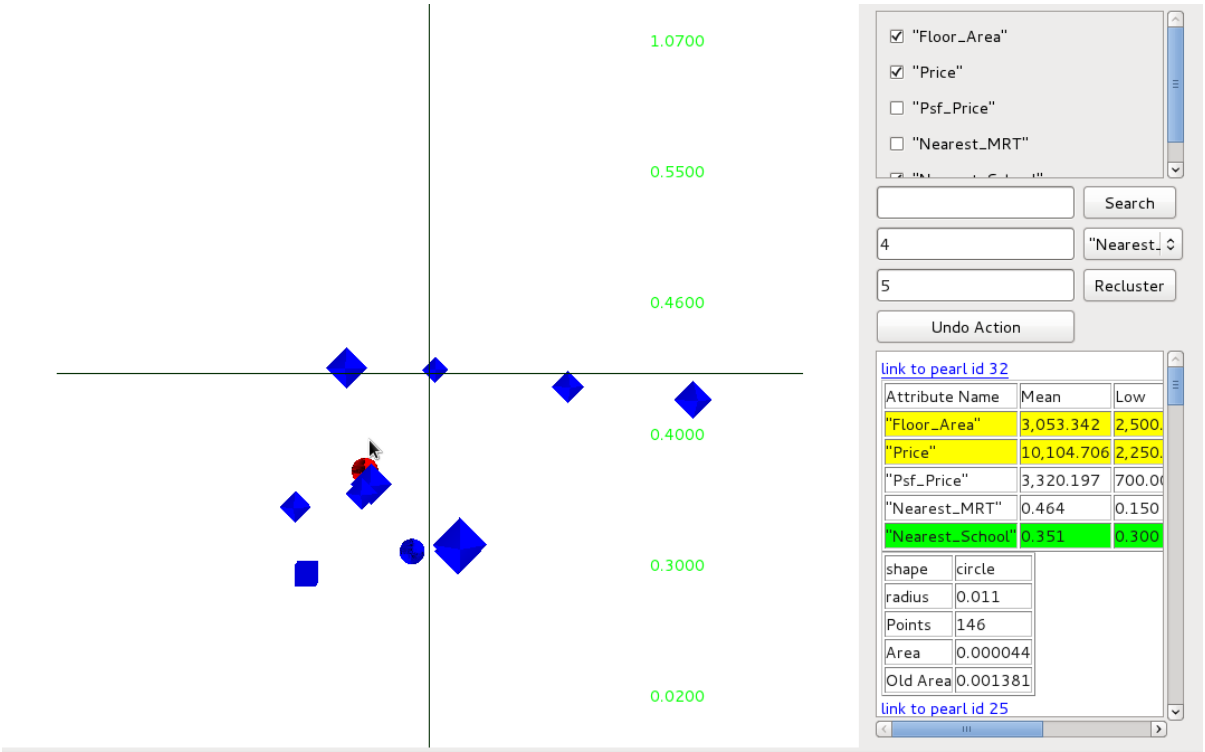
Pearl id	34		
Points	124		
Attribute Name	Mean	Low	High
Floor_Area($sqft$)	1356.347	1174	1915
Price($in\ thousands\ of\ Singapore\ dollars$)	1970.541	1150.001	2435.548
Nearest_School(km)	0.436	0.39	0.46
Pearl id	33		
Points	110		
Attribute Name	Mean	Low	High
Floor_Area($sqft$)	1271.1	1152	1700
Price($in\ thousands\ of\ Singapore\ dollars$)	1757.271	1080.003	2449.999
Nearest_School(km)	0.342	0.29	0.39
Pearl id	35		
Points	141		
Attribute Name	Mean	Low	High
Floor_Area($sqft$)	1415.44	1152	1918
Price($in\ thousands\ of\ Singapore\ dollars$)	1809.329	1200.002	2449.999
Nearest_School(km)	0.238	0.1	0.3
Pearl id	36		
Points	33		
Attribute Name	Mean	Low	High
Floor_Area($sqft$)	2185.394	1819	2900
Price($in\ thousands\ of\ Singapore\ dollars$)	1887.818	1350	2449.999
Nearest_School(km)	0.362	0.24	0.46

Table 6.3: Task on Singapore Dataset



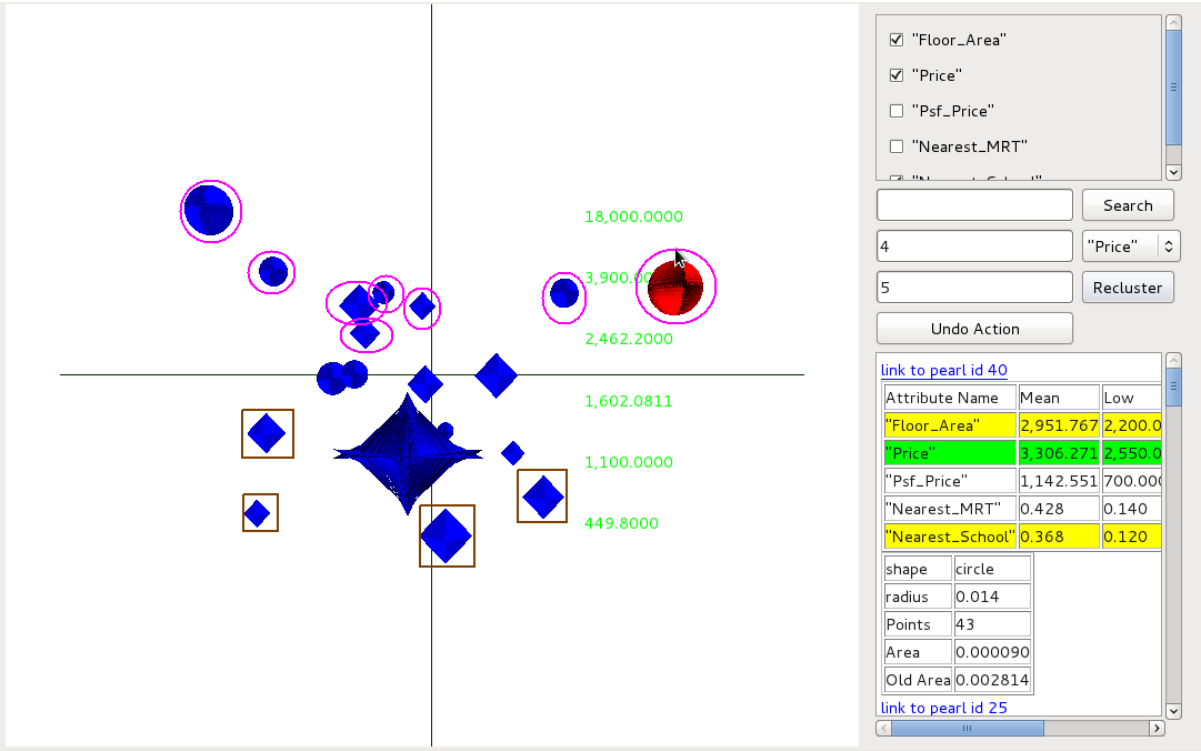
Result of Step1: z-dimesnion maps to *Nearest_school* which was selected as the data dimension. There are five consecutive bins in z-dimension. Clustering is done only on *Floor_Area* and *Price*.

Pearls which are enclosed in a pink circle are candidates for pruning as they represent real estate with large *Nearest_school* distance.



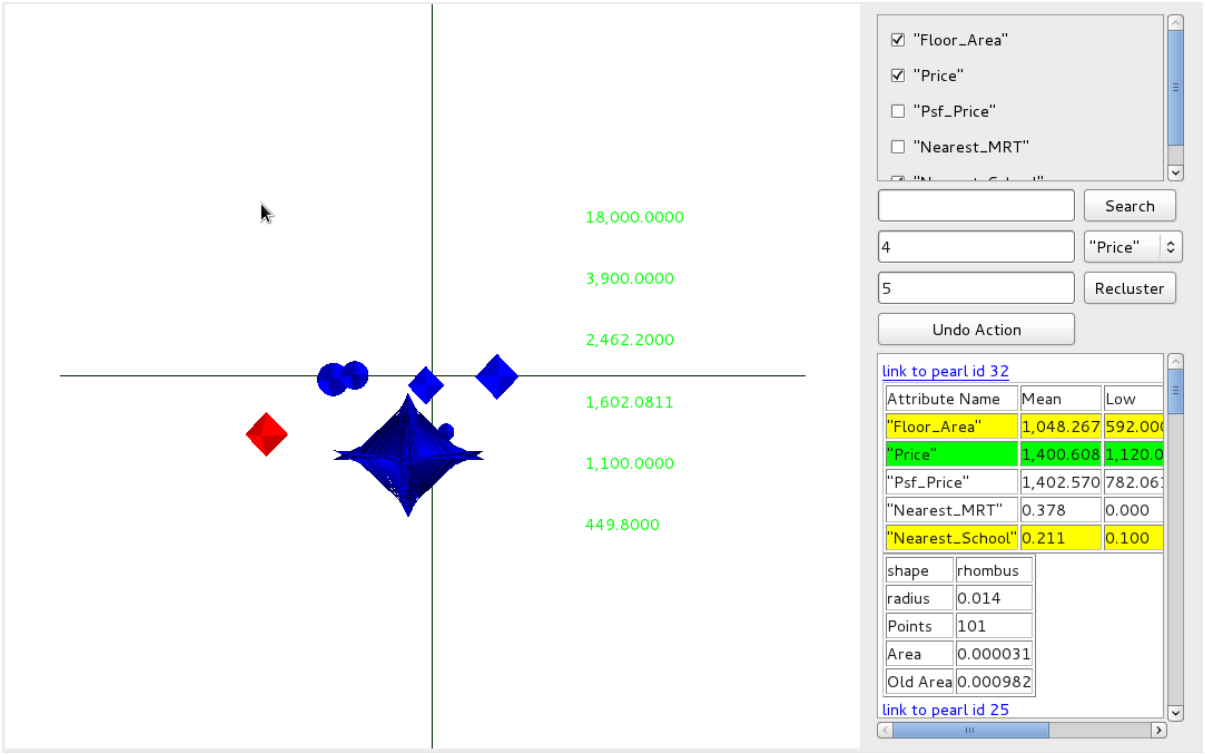
Result of Step2: Pearls enclosed in pink in previous image were removed.

Among the remaining pearls, our aim is to find pearls representing data points with high price as well as low area. Exploration reveals that pearls representing such data points do not exist in this view.

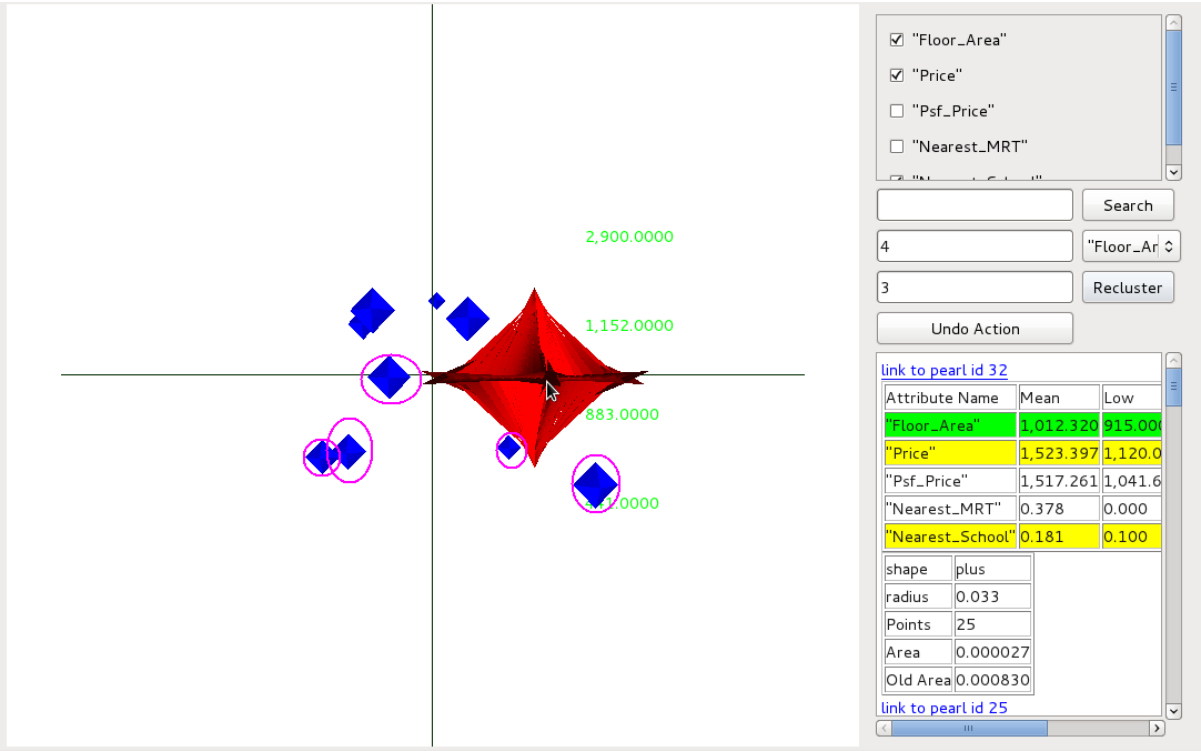


Result of Step3: z-dimension maps to *Price* which was selected as the data dimension. There are five consecutive bins in z-dimension.

Pearls in bin-1 and bin-2 from top are marked in pink circles, they represent data points with high *Prices* and they will be removed. Pearls which area marked with brown rectangular region represent data points with below acceptable $floor\_area$ and they will also be removed.

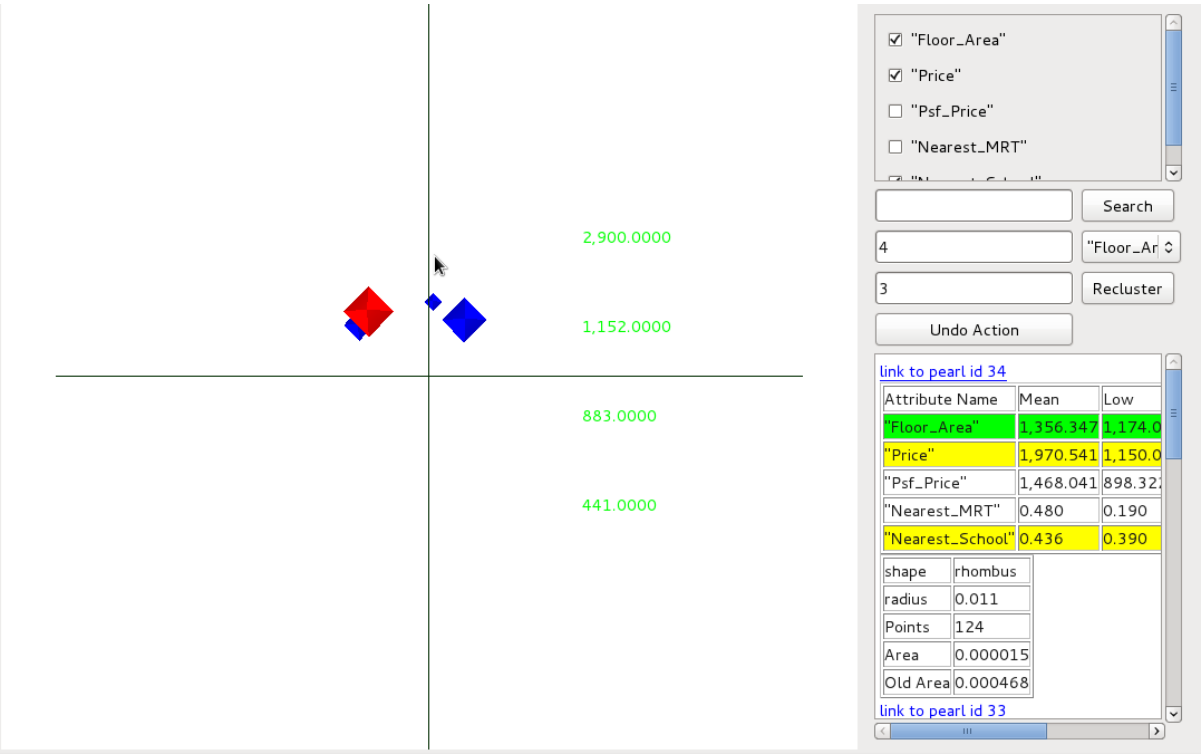


Result of Step4: Pearls enclosed in pink circle or brown rectangle in last image are removed. Further exploration using textual view and detail views of pearls reveal presence of many data points with small $floor\_area$.



Result of Step5: *Floor_Area* is used as data dimension with 3 bins and 4 pearls. Three bins consecutive bins in z-dimensions show three ranges of data-points in *floor_areas*. Clustering is done on *Nearest_School* and *Price*.

Exploration using textual view and detail view reveals that first range from top (1152 - 2900) has approximately 400 data points which satisfy required criterion. Hence, other pearls will be removed.



Result of Step6: The remaining four pearls can be analyzed in detail using detail view techniques and user can narrow down to specific properties which he/she may want to buy.

Figure 6.1 shows the parallel coordinate plot for cluster 1 of Singapore dataset. This is the parallel coordinate visualization of cluster 1 of Singapore 1 dataset on which we performed the Task 1. This visualization is created using Xmdv toolkit. While it is possible to see that there exist some points in the cluster which satisfy our query, it is not possible to

1. determine how many such points exist and what is their distribution?

2. determine the cluster structure since there are around 5000 points in the cluster over plotting leads to problem a) and b)

3. group these points according to values in particular dimensions. Parallel coordinate algorithm does not supports any grouping whereas PEARLS groups the points in various pearls. Since number of points satisfying the query might be large it is essential to group these points and show summary of every group.

4. a new query on the output of first query.

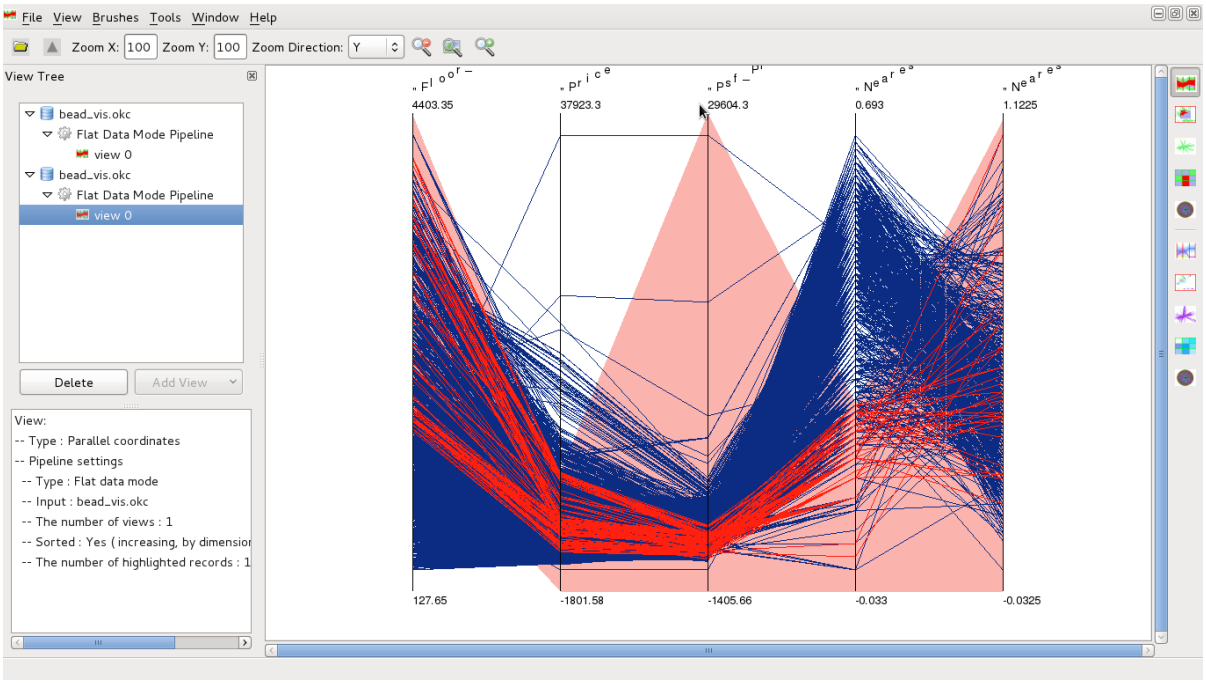


Figure 6.1: Parallel coordinate visualization of cluster 1 of Singapore dataset

6.1.2 Baseball Hall of Fame Dataset

We use the methodology described in section 5.1 in figure 5.1 to perform a task on baseball Hall of Fame dataset. The exact set of operations are described below.

Dataset Description : Baseball Dataset contains following attributes : *Runs_Scored*, Hits, Doubles, Triples, *Home_Runs*, *Runs_Batted_In*, *Batting_Average*, Walks, Strikeouts, *Stolen_Bases* and *Fielding_Average*. It consists of information about 1340 major league baseball players who had retired prior to the 1993 and who were were eligible for the Major League Baseball Hall of Fame (had played in at least ten seasons). The dataset is loaded as a single cluster in PEARLS toolkit. Original dataset contained 26 attributes, we use only 11 attributes to generate pearls and perform the task.

Task Description : The task is to find a group of players who are approximately among top 25% in *home_runs* as well as *Fielding_Average* and who have above average doubles and triples.

Following operations complete this task. The pearls in image 6.2 represent step 2 of these operations.

1. Use data dimension technique on *Fielding_Average* with number of bins as 4 and number of pearls as some convenient number. Note the range of upper most bin (bin with fielders with highest *Fielding_Average*, ie .9830 - 1.00)

2. Use data dimension technique on *home_runs* with number of bins as 4 and filter all other attributes apart from *Fielding_Average*, *doubles* and *triples*.

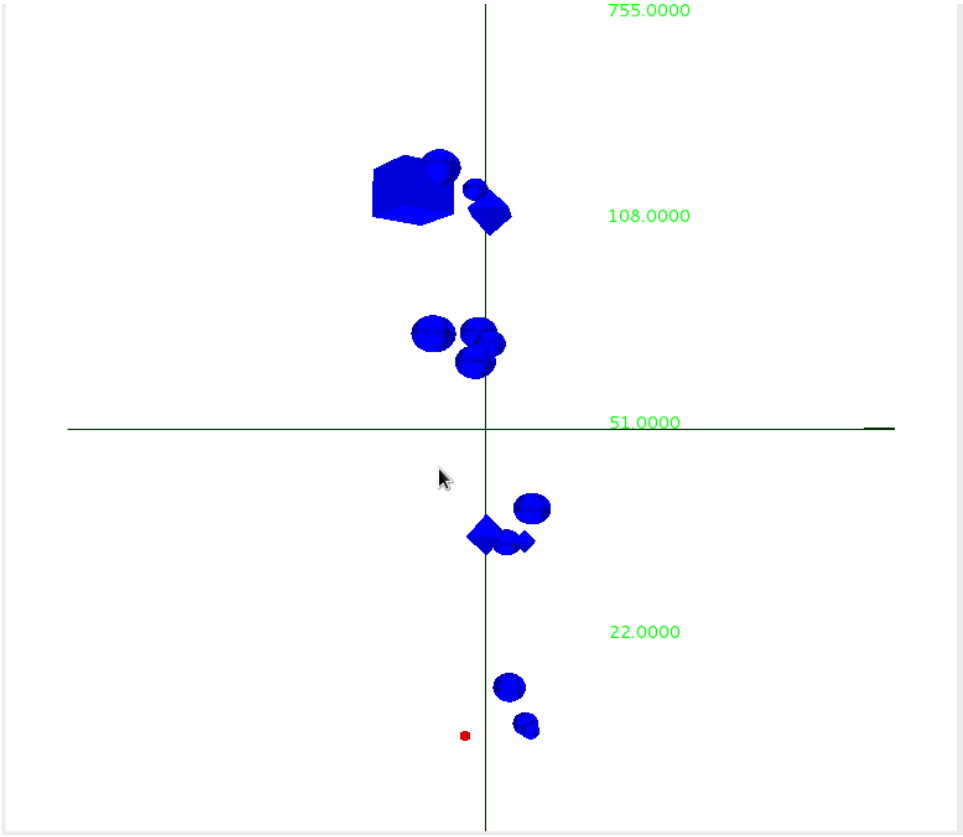


Figure 6.2: Pearls in the plot are from step 2 of case study on Baseball Hall of Fame dataset. Ranges 0-22, 22-51,52-108 and 108-755 represent various bins for *home_runs* dimension.

3. Prune pearls which belong to first three bins (*home_runs* range : (0,108.8)). From the remaining pearls, prune pearls where number of doubles and triples is lower than cluster average.

4. Use data dimension technique on *Fielding_Average* with number of bins as 4 and number of pearls as 1 (since number of points left are already low and only points with below required fielding average need to be removed)

5. Perform pruning by using attribute filtering or data dimension on doubles, triples if required.

Table 6.4 displays list of such players with their *home_runs*, *Fielding_Average*, *doubles* and *triples*.

6.2 Querying using PEARLS

In the previous section, we performed two tasks of answering queries on various datasets. PEARLS can be used to visually query datasets. Queries with selection predicate over multiple dimensions can be answered using pearls. The advantage of PEARLS over traditional querying is that a user can analyze

the data by looking at the visualization and then specify query. Instead of hard ranges a user deals with relative distances of various points from cluster centroid in the selected dimensions and prunes the points which are too far or too close. Also a user can divide the dataset in the bins and look at the distribution of data in various bins. This gives a better understanding of data distribution which is helpful in deciding what further interactive actions are required to give the desired query.

The queries which specify relative distances from cluster centroid in various dimensions can be interactively answered using PEARLS. For example: a query on vehicle dataset, select vehicles with large horsepower,high acceleration, moderate cost and good average is a query which specifies relative distances from cluster centroid.

For such a query,

- A user must arrange the query predicates according to his priority order over dimension.
- He/She should then use data dimension technique to prune the dimensions not required in query and generate Pearls over the level of granularity required. The location of every pearl specifies in which quadrant does the pearl lies. For a dimension order [D1,D2,D3], if a pearl is in dimension (+,+,+), this implies it has higher average that cluster average in D1,D2 and D3. By looking at location of the pearl and text view, a primary pruning of uninteresting pearls can be done.
- For further analysis, various dimensions can be either selected as data dimensions or sets of dimensions can be pruned using attribute pruning to generate pearls.
- This process continues iteratively till only interesting points remain.

6.3 Discussions

In PEARLS toolkit, Pearl id's are assigned to pearls on the basis of their distance from cluster centroid. Farther the pearl, larger the pearl id. A user can choose to display the id's of pearls on top of them in visual plot. Since we are viewing a three dimension plot on a two dimensional computer screen, some pearls which are far from cluster centroid may appear relatively closer from certain viewing angles. Displaying the pearl id is helpful in avoiding confusion for these instances. Shape and size of pearl are computed after removing farthest ten percent points which are assumed to be true. Hence, the pearls which are larger in size have a relatively large number of points far from cluster centroid as compared to pearls which are smaller in size. The pearls which are close to cluster centroid are comprised of points which are similar to average and pearls which are far from centroid are comprised of points which are far

from average. Pearls which are far from cluster centroid tend to have large size. This can be attributed to the fact that regions close to cluster centroid tend to be more dense as compared to regions which are far from cluster centroid.

Data dimension technique, Swiss cheese view and filtering can together be effectively used for visual data analysis because they support expression of useful multidimensional queries through interaction and aid of data mining. Trained domain experts can follow complex lines of inquiry using sequences of simple interactions and perform a wide range of visual analysis tasks.

In a lot of data analysis tasks, it is difficult to specify data points of interest as set of mathematical and Boolean rules. It is also difficult to update these rules when new interests are found. Moreover, a viewer may not know apriori what they will find interesting. PEARLS visualization along-with the set of interactions described here is highly effective in such data analysis tasks.

In PEARLS visualization, whole data cluster can be viewed at a time which makes it easier to locate elements of interest. PEARLS does not suffer from drawbacks suffered by point abstraction tools like inability to plot complete dataset or loss of speed and interaction since number of visual objects (pearls) is significantly lesser than number of data points in most cases. It may suffer from over plotting and decline in legibility when some pearls are overshadowed by larger pearls but an effective text based view and ability to rotate the 3-D visualization vertically and horizontally solves this problem.

This technique provides intuitiveness and efficiency by leveraging well established data mining techniques of clustering and user's apriori knowledge of the dimensions' semantics and hypotheses about relationships among dimensions.

Different visualization techniques show different trends and patterns from same dataset. It is very difficult to design a visualization technique which can reveal all the trends and interesting patterns from a dataset. Hence, visualization techniques should not be looked as techniques competing with each other. To derive the maximum benefits from various visualization techniques, it is essential to find how various techniques can be integrated and can complement each other. In PEARLS toolkit, we have integrated parallel coordinates, scatter plots and other visualization techniques because we believe that they are helpful in providing additional analysis of the dataset.

Table 6.4: Results of Task 2: List of data points in the remaining pearls after completion of task 2. Task: Find players who are approximately among top 25% in *home_runs* as well as *Fielding_Average* and who have above average doubles and triples

point_names	Home_Runs	Fielding_Average	Doubles	Triples
BOBBY_DOERR	223	0.98	381	89
GABBY_HARTNETT	236	0.984	396	64
CHET_LEMON	215	0.984	396	61
FRED_LYNN	306	0.988	388	43
FRANK_WHITE	160	0.984	407	58
CESAR_CEDENO	199	0.985	436	60
STAN_MUSIAL	475	0.984	725	177
CARL_YASTRZEMSKI	452	0.981	646	59
HANK_AARON	755	0.98	624	98
BILL_TERRY	154	0.992	373	112
JIM_BOTTOMLEY	219	0.988	465	151
JOHNNY_MIZE	359	0.992	367	83
CHRIS_CHAMBLISS	185	0.993	392	42
ERNIE_BANKS	512	0.994	407	90
CECIL_COOPER	241	0.992	415	47
JOE_TORRE	252	0.99	344	59
GARY_CARTER	324	0.991	371	31
KEITH_HERNANDEZ	162	0.994	426	60
JOE_KUHEL	131	0.992	412	111
AMOS_OTIS	193	0.991	374	66
YOGI_BERRA	358	0.989	321	49
BOB_WATSON	184	0.991	307	41
BILL_DICKEY	202	0.988	343	72
RON_FAIRLY	215	0.991	307	33
GEORGE_SCOTT	271	0.99	306	60
LEE_MAY	354	0.994	340	31
GIL_HODGES	370	0.992	295	48
DICK_ALLEN	351	0.989	320	79
VIC_POWER	126	0.994	290	49
RUDY_YORK	277	0.99	291	52
BILL_WHITE	202	0.992	278	65
ROY_SIEVERS	318	0.991	292	42
JOE_ADCOCK	336	0.994	295	35
NORM_CASH	377	0.992	241	41
ROY_WHITE	160	0.988	300	51
GEORGE_McQUINN	135	0.992	315	64
HAL_TROSKY	228	0.993	331	58
GEORGE_KELLY	148	0.992	337	76
HANK_GREENBERG	331	0.991	379	71
LOU_GEHRIG	493	0.991	534	163
TONY_PEREZ	379	0.992	505	79
JIMMIE_FOXX	534	0.992	458	125

Chapter 7

Conclusions

In this thesis, we presented PEARLS, an interactive multidimensional cluster visualization toolkit. It provides a novel approach to cluster structure and shape visualization and cluster exploratory analysis. PEARLS leverages well established data mining technique of clustering to generate pearls from data clusters. Pearls could be thought as a layer of abstraction between point level and cluster level. PEARLS scales well to support large number of data points as well as large number of pearls. PEARLS satisfy well established functional and non functional requirements for interactive data visualization toolkits. Several interaction techniques specific to pearls visualization have been developed for cluster manipulation and intuitive exploratory analysis. Swiss Cheese view, data dimension and attribute filtering are major interaction techniques. These techniques help the user to control logic used to produce visualization. We demonstrate the use of PEARLS toolkit in cluster visualization, concept search within a cluster, concept description and data point search. Our evaluation through case studies on two real datasets (Singapore real estate dataset and nutrition dataset) demonstrates the effectiveness of PEARLS.

7.1 Future Work

Future work includes

- Performing additional experiments with a variety of clustering algorithms to generate pearls.
- exploring computational geometry techniques to find shapes not restricted to Lp norm shapes.
- performing further user studies to evaluate aspects of the design and improving coloring of pearls.
- building a web based version of toolkit.

Related Publications

- Nahil Jain, Soujanya Lanka and Kamal Karlapalem, “Exploratory Visual Analysis for Data Enthusiast” , Under Review, IEEE VAST 2012 .

Bibliography

[1] G. Andrienko and N. Andrienko. Blending aggregation and selection: Adapting parallel coordinates for the visualization of large datasets. 2005.

[2] M. Ankerst, D. A. Keim, and H. peter Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. 1996.

[3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987.

[4] C. Bentley and M. Ward. Animating multidimensional scaling to visualize n-dimensional data sets. In *Information Visualization '96, Proceedings IEEE Symposium on*, pages 72 –73, 126, oct 1996.

[5] C. A. Brewer. The color brewer.

[6] R. I. Bull, A. Trevors, A. J. Malton, and M. W. Godfrey. Semantic grep: Regular expressions + relational abstraction. In *Proceedings of the Ninth Working Conference on Reverse Engineering (WCRE'02)*, WCRE '02, pages 267–, Washington, DC, USA, 2002. IEEE Computer Society.

[7] N. Cao, D. Gotz, J. Sun, and H. Qu. Dicon: Interactive visual analysis of multidimensional clusters. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2581 –2590, dec. 2011.

[8] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.

[9] W. W. Y. Chan. A survey on multivariate data visualization.

[10] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, June 1973.

[11] W. C. Cleveland. *Dynamic Graphics for Statistics (Wadsworth & Brooks/Cole Statistics/Probability Series)*. Springer, 1 edition, jul 1988.

[12] J. J. Cochran. Career records for all modern position players eligible for the major league baseball hall of fame. In *Journal of Statistics Education*, 2000.

[13] J. Ellson, E. Gansner, L. Koutsofios, S. North, G. Woodhull, S. Description, and L. Technologies. Graphviz open source graph drawing tools. In *Lecture Notes in Computer Science*, pages 483–484. Springer-Verlag, 2001.

[14] S. Feiner and C. Beshers. Worlds within worlds : Metaphors for exploring n-dimensional virtual worlds. pages 76–83. ACM Press, 1990.

[15] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, pages 43–50, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[16] G. W. Furnas and A. Buja. Prosection views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3:323–385, 1994.

[17] J. Han and M. Kamber. *Data Mining - Concepts and Techniques*. Morgan Kaufmann Publishers.

[18] P. E. Hoffman. *able Visualizations: A Formal Model and Its Applications*. Doctoral Dissertation. University of Massachusetts at Lowell, 1999.

[19] P. E. Hoffman. *Table visualizations: a formal model and its applications*. PhD thesis, 2000. AAI9950455.

[20] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985. 10.1007/BF01898350.

[21] A. Inselberg. Multidimensional detective. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 100 –107, oct. 1997.

[22] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[23] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[24] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 125 – 132, oct. 2005.

[25] E. Kandogan. 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*, KDD '01, pages 107–116, New York, NY, USA, 2001. ACM.

[26] T. A. Keahey and T. A. Keahey. Visualization of high-dimensional clusters using nonlinear magnification. In *In Visual Data Exploration and Analysis VI, volume 3643 of SPIE*, 1999.

[27] D. Keim and H.-P. Kriegel. Visdb: database exploration using multidimensional visualization. *Computer Graphics and Applications, IEEE*, 14(5):40 –49, sept. 1994.

[28] D. A. Keim, M. Ankerst, and H.-P. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the 6th conference on Visualization '95*, VIS '95, Washington, DC, USA, 1995. IEEE Computer Society.

[29] H. M. Kienle and H. A. Muller. Requirements of software visualization tools: A literature survey. *Visualizing Software for Understanding and Analysis, International Workshop on*, 0:2–9, 2007.

[30] R. Koschke. Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey. *Journal of Software Maintenance*, 15(2):87–109, Mar. 2003.

[31] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 230–237, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[32] H. Levkowitz. Color icons-merging color and texture perception for integrated visualization of multiple parameters. In *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, pages 164 –170, 420, oct 1991.

[33] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1027 –1035, nov.-dec. 2010.

[34] T. Mihalisin, J. Timlin, and J. Schwegler. Visualization and analysis of multi-variate data: a technique for all fields. In *Proceedings of the 2nd conference on Visualization '91*, VIS '91, pages 171–178, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[35] L. Nováková and O. Štepánková. Multidimensional clusters in radviz. In *Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization*, SMO'06, pages 470–475, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[36] M. Novotny. Visually effective information visualization of large data. In *In 8th Central European Seminar on Computer Graphics (CESCG 2004*, pages 41–48. CRC Press, 2004.

[37] R. Pickett and G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on*, volume 1, pages 514 –519, aug 1988.

[38] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pages 514–519, 1988.

[39] R. Rao and S. K. Card. 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*, CHI '94, pages 318–322, New York, NY, USA, 1994. ACM.

[40] S. P. Reiss. The paradox of software visualization. In *Proceedings of the 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis*, VISSOFT '05, pages 19–, Washington, DC, USA, 2005. IEEE Computer Society.

[41] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, July 2002.

[42] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, Jan. 1992.

[43] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336 –343, sep 1996.

[44] M.-A. D. Storey, D. Čubranić, and D. M. German. On the use of visualization to support awareness of human activities in software development: a survey and a framework. In *Proceedings of the 2005 ACM symposium on Software visualization*, SoftVis '05, pages 193–202, New York, NY, USA, 2005. ACM.

[45] S. Tilley, S. Huang, and T. Payne. On the challenges of adopting rots software, 2003.

[46] S. Vadapalli. *Reverse Nearest Neighbors Driven Clustering and Visualization of High Dimensional Data*. PhD Thesis. IIIT Hyderabad, 2011.

[47] S. Vadapalli and K. Karlapalem. Beads: High dimensional data cluster visualization. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 235 –236, oct. 2009.

[48] T. Van Long and L. Linsen. Multiclustertree: Interactive visual exploration of hierarchical clusters in multidimensional multivariate data. In *Computer Graphics Forum*, 28, pages 823–830, 2009.

[49] X. Wang. Volumes of generalized unit balls. *MATHEMATICS MAGAZINE*, 78:390–394, 2005.

[50] M. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *Visualization, 1994., Visualization '94, Proceedings., IEEE Conference on*, pages 326 –333, oct 1994.

[51] Wikipedia. Lou gehrig — wikipedia, the free encyclopedia, 2012. [Online; accessed 16-April-2012].

[52] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. pages 3–33. IEEE Computer Society Press, 1997.

[53] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, and W. Ribarsky. Value and relation display: Interactive visual exploration of large data sets with hundreds of dimensions. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):494 –507, may-june 2007.