PIXEL-ORIENTED TECHNIQUE: A TECHNIQUE TO VISUALIZE MULTIDIMENSIONAL DATA

Jaspreet Kaur[1], Dr. Dharmender Kumar[2]

[1]Student, [2]Assistant Professor of CSE Deptt

Guru Jambheshwar University of Science and Technology, Hisar

[1]jaspreet11cse@gmail.com, [2]dharmindia24@gmail.com

ABSTRACT

In the world of knowledge, the amount of data is rising day by day. To explore this huge data various data mining techniques are available, but the result of these techniques is not easily understandable. To simplify the result, different type of visualization technique are available i.e. pixel-oriented, parallel coordinate, stick figure , scatterplot matrics, dimensional stacking and star glyph technique to visualize multidimensional data. Pixel-oriented technique is one of them. Flat and hierarchical pixel-oriented technique widely used, is described here. The XmdvTool is open source software used for the visualization of dataset using pixel-oriented technique.

Keywords: Visualize multidimensional data, data visualization, hierarchical clustering.

INTRODUCTION

Our ability to generate and accumulate information has far exceeded our ability to effectively process them. The omnipresent exchange of information in this technology age is a strong reason for data growth. The proliferation of the internet today could only hint at the potential of tomorrow's exponential information growth. Advancements in data acquisition technologies results in acquiring data at far greater densities and resolutions than ever before. Such evidence clearly puts forward a case that data size is ever raising; and that in view of this growth it will be increasingly difficult to process data in search of anomalies, patterns, features, and ultimately knowledge and extrapolation of that knowledge. Interpreting data, be it overwhelming or not, is a hard task. But it is most difficult to find patterns in the voluminous data when it is unclear. Automated analysis is hopeless when such analysis criteria cannot even be explicitly formulated. To deal with very huge amount of data, automated analysis methods [29] and visualization techniques are combined for effective exploration of high dimensional data. [30] This is where visualization plays a crucial and most effective role by relying upon the power of the trained human eye.

Many techniques have been proposed by researchers to visualize data effectively. Since computers are used to create visualizations, many novel visualization techniques have been developed and existing techniques have been extended to work for larger data sets and make the displays interactive in multidimensional data sets. [18] There are so many challenges in visual data analysis like Interpretability, to provide semantics for future analysis tasks and decision-centered visualization, Evaluation and User acceptability. [19]

MULTIDIMENSIONAL DATA

A multidimensional data set consists of a collection of *N*-attributes and huge amount of tuples, where each entry of an *N*-tuple is a nominal or ordinal value corresponding to an independent or dependent variable.[34] We distinguish ourselves from the domain-specific class of visualization methods. Our research can be termed as non-domain-specific multivariate visualization, a radically different mode of displaying multidimensional data. Its non-domain specific nature results in generality and as such may be used to display a much larger class of datasets. Examples of such datasets include results from censuses, surveys and simulations. Most analysis of such data to date still relies on the application of statistical computations. Statistical methods, though precise, lack the richness of graphical depictions. But more importantly, they require the user to explicitly define sets of parameters for analysis. This is an arduous task, if not impossible, if the user has little knowledge or intuition about the characteristics of the data that they are about to analyze. Visualization serves to complement rather than to replace traditional data analysis. Visualization is the graphical representation of information, with the goal of providing the viewer with a qualitative understanding of the information contents. A visual sense of the data provides an interpretation unprecedented by statistical methods.

TECHNIQUES TO VISUALIZE MULTIDIMENSIONAL DATA

Researchers have developed so many approaches to visualize multi-dimensional data. Large datasets poses two problems to interactive displays. One is how to show the elements on the screen as fast as enough. Second is if it is drawn on the screen, can the user able to understand it. Visual occlusion [21] is a problem in general for visualization. Edward R. Tufte provides many examples of visualization techniques that have been used for many years. They have been broadly categorized [3] as:

Geometric Projection

These techniques aim to find interesting geometric transformations and projections of multidimensional data sets. Examples include Scatterplot matrices [7], Landscapes [31], Hyperslice [16] and Parallel Coordinates [1].

Icon-based

The idea here is to map each multidimensional data item to a shape, where data attributes control shape and color attributes. Examples include Chernoff-faces [12], Stick figures [26], Shape Coding [15], and Color Icons [13].

Hierarchical

These techniques subdivide the k-dimensional space and present the subspaces in a hierarchical fashion. Examples include Dimensional Stacking [24], Worlds-within-worlds [28], Treemap [2], and Cone Trees [11].

Pixel-based

Here, each attribute value is represented by one colored pixel and the values for each attribute are presented in separate sub windows. Examples are spiral [4] [5], recursive pattern [6], and circle segment techniques. [23]

The first three techniques do not scale well with respect to the size of the dataset. The main problem with applying such techniques to large data sets is display clutter that the amount of clutter obscures or occludes any visible trends in the data display.

FLAT PIXEL-ORIENTED TECHNIQUES

The basic idea of pixel-oriented techniques is to map each data value to a colored pixel and present the data values belonging to one attribute in separate windows. Since, in general, our techniques use only one pixel per data value, the techniques allows us to visualize the largest amount of data, which is possible on current displays (up to about 1,000,000 data values). If each data value is represented by one pixel, then question arises how to arrange the pixels on the screen. Our pixel-oriented techniques use different arrangements for different purposes. If a user wants to visualize a large data set, the user may use a query-independent visualization technique which sorts the data according to some attribute(s) and uses a screen-filling pattern to arrange the data values on the display. The query-independent visualization techniques are especially useful for data with a natural ordering according to one attribute (e.g., time series data). However, if there is no natural ordering of the data and the main goal is an interactive exploration of the database, the user will be more interested in feedback to some query. In this case, the user may turn to the query-dependent visualization techniques which visualize the relevance of the data items with respect to a query. Instead of directly mapping the data values to color, the query-dependent visualization techniques calculate the distances between data and query values, combine the distances for each data item into an overall distance, and visualize the distances for the attributes and the overall distance sorted according to the overall distance. The arrangement of the data items centers the most relevant data items in the middle of the window, and less relevant data items are arranged in a spiral-shape to the outside of the window. A11 pixel-oriented techniques partition the screen into multiple windows. For data sets with m attributes (dimensions), the screen is partitioned into m windows one for each of the attributes. (figure. 1)In case of the query-dependent techniques, an additional (m + 1) th window is provided for the overall distance. Inside the windows, the data values are arranged according to the given overall sorting which may be data-driven for the query-independent techniques or query-driven for the query-dependent techniques. Correlations, functional dependencies, and other interesting relationships between attributes may be detected by relating corresponding regions in the multiple windows. [18]

Query-Independent Pixel-Oriented Technique

Simple query-independent arrangements are to arrange the data from left to right in a line-by-line fashion or top down in a column-by-column fashion. If these arrangements are done pixel wise, in general, the resulting visualizations do not provide useful results. For data mining even more important are techniques that provide nice clustering properties as well as an arrangement which is semantically meaningful. An example for a technique which has these properties is the recursive pattern technique. The recursive pattern is based on a generic recursive scheme which allows the user to influence the arrangement of data items.

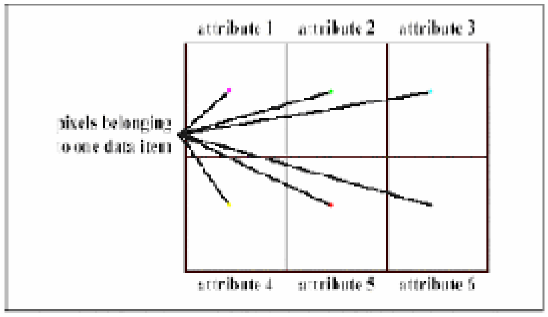

Figure 1: Arrangement of Attribute Windows for Data with Six Attributes [18]

It is based on a simple back and forth arrangement: First, a certain number of elements is arranged from left to right, then below backwards from right to left, then again forward from left to right, etc. [18] The 'recursive pattern' visualization technique is based on a simple back and forth arrangement: First, a certain number of elements is arranged from left to right, then below backwards from right to left, then again forward from left to right, and so on. The same basic arrangement is done on all recursion levels with the only difference that the basic elements which are arranged on level i are the patterns resulting from level(i-l)-arrangements. Let w_i be the number of elements arranged in the left-right direction on recursion level i and h_i be the number of rows on recursion level i. Then, the pattern on recursion level i consists of w_i x h_i level (i-1)-patterns, and the maximum number of pixel that can be presented on recursion level k is given by πw_i x $h_{i.}$(Figure. 2) [6]

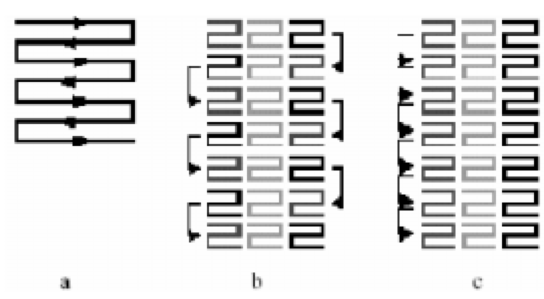

Figure 2: Illustration of the Recursive Pattern Technique [6]

Query-Dependent Pixel-Oriented Techniques

Instead of directly mapping attribute values to colors, the distances of attribute values to the query are mapped to colors. To make this concept clear we shall try to describe it in details. Let a relational database has attribute a_1, a_2… a_k. Now we take this database as a set of tuples (a_1, a_2….a_k).Simple queries against this database can be described as a region in the k dimensional space with the k attribute .The number of resulted data item found in this k-dimensional space can be empty or quite large but it cannot be determined before the query. So for this exact result one cannot understand about the behavior of the data. So to give more feedback besides data items within the query region, data item 'close' to the query region means which are approximately fulfilling the query are also presented. Now to get approximate results distances between data and query values are calculated. Now let for the k attributes we get the distance tuples(d_1,d_2,…..d_k).Then we can calculate the overall distance d_{k+1},where the value of d_{k+1} is zero if the data item is within the query region otherwise it gives the distance of the data item to the query. For calculating d_{k+1} user provides weighting factor according to importance(w_1,w_2,……w_k).Then the distance tuples are sorted according to overall distance d_{k+1}.Now this distance tuples are mapped to color.[5] There are three types of query-dependent techniques: Spiral Technique, Axis Technique and circle segment technique. In spiral technique, data items having the highest relevance factor are presented in the center of the display and then data items with lower relevance's in a rectangular spiral shape around the center. This technique is improved for getting a high degree of clustering by using a generic spiral form which may have a snake-peano-hilbert or Morton[10,8,9] like local pattern of certain degree (1,2,4,8,16) instead of rectangular spiral shape. This technique is clearly shown from figure 3. [22] Another technique is axis technique, where it includes the direction of the distance into visualization. The basic idea is

to assign two attribute to two axes. Then arrange data items according to direction of the distance see figure 4(a).

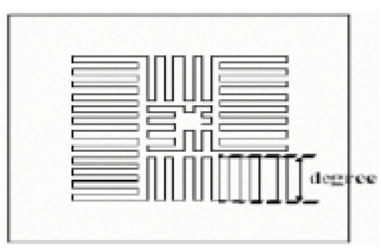

Figure 3: Generalized-Spiral Technique (degree of local pattern is 8, [22]

It is an improvement over the spiral technique. The problem for this technique is that though it represents high degree of expressiveness it can visualize slightly lower number of data items.

Last one is circle segment technique, in this the arrangement of the pixels starts at the center of the circle and continues to the outside by plotting on a line orthogonal to the segment halving line in a back and forth manner (figure 4(b)). [23] The advantage of this approach is that close to the center all attributes are close to each other enhancing the visual comparison of their values. But XmdvTool only implements spiral technique not axes and circle segment technique.

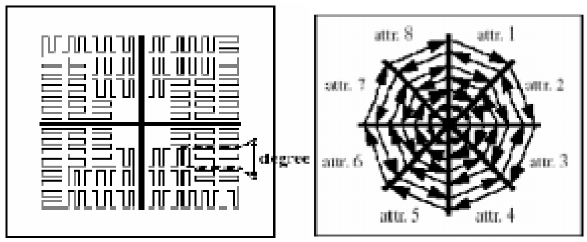

Figure 4: a) Generalized Axes Arrangement of one Attribute (Snake Spiral, degree of local pattern is 4), b) Illustration of the Circle Segments technique for 8-dimensional data [18]

HIERARCHICAL PIXEL-ORIENTED TECHNIQUE

The flat visualizations become very crowded when they are applied to large-scale data sets. To overcome this clutter problem, an Interactive Hierarchical Display framework is developed [17]. The underlying principle of this framework is to develop a multi-resolution view of the data via hierarchical clustering, and to use hierarchical variations of traditional multivariate visualization techniques to convey aggregation information about the resulting clusters. Users can then explore their desired focus regions at different levels of detail, using our suite of navigation and filtering tools.

Hierarchical Clustering

The primary purpose for building a cluster hierarchy is to structure and present the data at different levels of abstraction. A hierarchical approach is a convenient mechanism for organizing large datasets. By recursively clustering or partitioning data into related groups and identifying suitable representative information (summarizations) for each cluster, we can examine the data set methodically at different levels of abstraction, moving down the hierarchy (drill-down) when interesting features appear in the summarizations and up the hierarchy (roll-up) after sufficient information has been gleaned from a particular sub tree. A clustering algorithm groups objects or data items based on measures of proximity between pairs of objects [20]. In particular, a hierarchical clustering algorithm constructs a tree of nested clusters based on proximity information.

Let E be the a set of k N-dimensional objects, i.e.,

$$E = \{e_1, e_2, e_3 \dots E_k\}$$

Where e_i is an N-vector:

$$e_i = \{x_{i1}, x_{i2}, x_{i3}, \dots x_{iN}\}$$

An m-partition P of E breaks E into m subsets $\{P_1, P_2, \dots P_m\}$ satisfying the following two criteria:

$$P_i \cap P_j = \phi \text{ for all } 1 \leq I, j \leq m, i \neq j; \text{ and}$$

$$\bigcup_{i=1}^{m} P_i = E$$

A partition Q is nested into a partition P if every component of Q is a proper subset of a component of P. That is, P is formed by merging components of Q. A hierarchical clustering is a sequence of partitions in which each partition is nested into the next partition in the sequence. A hierarchical clustering may be organized as a tree structure: Let P_i be a component of P, and Q be the m partitions of P_i. Let P_i be instantiated by a tree node T_i. Then, the components of Q form the children nodes of T_i. The approaches that impose a hierarchical structure to a data set can be broadly categorized as either *explicit* or *implicit* clustering. In *explicit clustering*, hierarchical levels correspond to dimensions and the branches correspond to distinct values or ranges of values for the dimension. Hence different orders of the dimensions give different hierarchical views. On the other hand, *implicit clustering* tries to group similar objects based on a certain metric, for instance the Euclidean distance.[34]

Interactive Tools in Hierarchical Visualization

There are several interactive tools, such as the structure based brush, drill-down/roll-up operations, extent scaling, and dynamic masking, to help users interactively explore the hierarchical visualizations. These are described briefly below. [32, 33, 25, 17]

- Structure-based brush: A structure-based brush allows users to select subsets of a data structure (for example, a hierarchy) by specifying focal regions as well as a levels-of-detail on a visual representation of the structure.
- Drill-down/Roll-up Operations: Drill-down/roll-up operations allow users to change the level of detail of interactive hierarchical displays intuitively and directly. Drill-down refers to the process of viewing data at an increased level of detail, while roll-up refers to the process of viewing data with decreasing detail [34].
- Extent Scaling: Extent scaling solves the problem of overlapping among the bands by decreasing the extents of all the bands in each dimension by scaling them uniformly via a dynamically controlled extent scaling parameter. Users can still differentiate between clusters with large and small extents after the bands have been scaled.
- Dynamic Masking: Dynamic masking refers to the capability of controlling the relative opacity between brushed and unbrushed clusters. It allows users to deemphasize or even eliminate brushed or unbrushed clusters. With dynamic masking, the viewer can interactively fade out the visual representation of the unbrushed clusters, thereby obtaining a clearer view of the brushed clusters while maintaining the context of unbrushed areas. Conversely, the bands of the brushed clusters can be faded out, thus obtaining a clearer view of the unbrushed region.

Navigation and Filtering Tools in Flat and Hierarchical approaches

- Dimension Zooming: The use of distortion techniques [35, 27] has become increasingly common as a means for visually exploring dense information displays. Distortion operations allow the selective enlargement of subsets of the data display while maintaining context with surrounding data. We introduce a distortion operation that we term dimension zooming. We scale up each of the dimensions independently with respect to the extents of the brushed subspace, thus filling the display area. The subset of elements may then be examined as an independent data set. This zooming operation may be performed as many times as desired. For a data set occupying a large range of values, this operation is invaluable for examining localized trends.

- Dimension reordering

SOFTWARE

The software used in this work is XmdvTool. XmdvTool is a public-domain software package for interactive visual exploration of multivariate data sets. XmdvTool is a multivariate visualization system developed by Mathew O. Ward that integrates several techniques for displaying and visually exploring multivariate data. XmdvTool version 8.0 is based on OpenGL and Tck/Tk and is available for Windows95/98/NT/2000/XP and Linux platforms. It already supports four classes of techniques for displaying both (non-hierarchical) flat form data and hierarchically clustered data, namely scatterplots, star glyphs, parallel coordinates, pixel-oriented and dimensional stacking. XmdvTool also supports a variety of interaction modes and tools, including brushing in screen, data, and structure spaces, zooming, panning, and distortion techniques, and the masking and reordering of dimensions. Finally, color themes and user customizable color assignments permit tailoring of the aesthetics to the users. Xmdv-Tool has been applied to a wide range of application areas, such as remote sensing, financial, geochemical, census, and simulation data. More information about the XmdvTool project can be obtained from http://davis.wpi.edu/~xmdv. [14]

RESULT AND ANALYSIS

This segment contains the result and analysis of pixel-oriented technique. Before presenting our result of visualization, in the following we briefly introduce how data characteristics such as correlations, functional dependencies, and clusters may be identified in our visualizations generated by the spiral technique. Note that these techniques are interactive in nature and are therefore difficult to describe in written form. Properties that hold for all or most of the data can be deduced from the overall brightness and color distribution of the visualizations. The size of the yellow portion of the visualization indicates the number of data items fulfilling the query for the corresponding attribute. The brightness of the visualization of some attribute indicates the degree of fulfilling the corresponding query predicate, and the overall color distribution shows the distribution of distance values for the corresponding attribute. Sharp borders between colors, which indicate discontinuities in the value range of an attribute, are especially interesting. Usually the visualizations of the attributes are not independent. Using our visual data mining techniques, correlations and functional dependencies between attributes may be identified by the similarity of their visualizations (see Figure 12).

A bit problematic are visualizations that do not show any structure. The human perceptual system may consider such visualizations as being similar although the pixels of the visualizations do not correspond to each other and no correlation between the attributes exists. In experimenting with various data sets, we found that the problem of misleading similarity of visualizations only occurs in visualizations without structure. In cases where existing structure in the data is not revealed by the visualization, the structure can usually be made visible by arbitrarily changing the weighting factors of some attributes. Changing the weighting factors makes it easier to perceive similarities in the visualizations and to determine correlations and functional dependencies.

Another important group of interesting data characteristics is clusters of data items with similar properties. In our visualizations, clusters usually appear as rectangular regions (possibly partial rectangles) which may have different colors for different attributes (See Figure 13). Clusters may have a lower dimensionality than the whole data set. Therefore, the clustering may appear only in the visualizations of certain attributes.

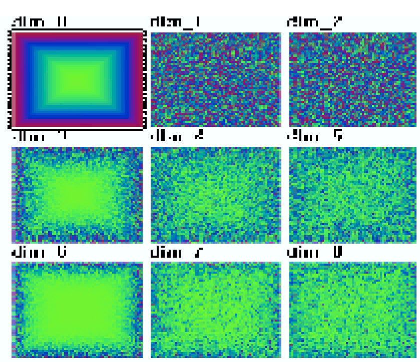

Figure 12: Linear functional dependency between dimensions 0 and 3 and quadratic functional dependency between dimensions 0 and 6. Note that there is no dependency between dimension 0, 1 and 2 as they are completely dissimilar in appearance.

Therefore, the clustering may appear only in the visualizations of certain attributes. Note that the visualizations generated by the spiral technique in general depend on the chosen query region. Changing the query region has a major impact on the resulting visualizations. For example, if the query region is moved away from a cluster, the cluster becomes less visible in the visualization. By interactively changing the query region, different clusters can be made visible.

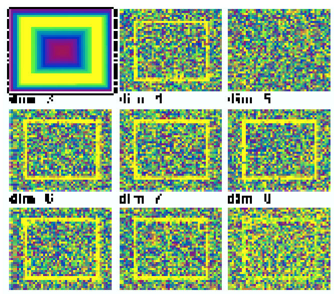

Figure 13: Clustering can be easily seen in this view. Also note that the visualization tells us that the cluster is of a lower dimension than the dataset.

Figure 14 shows flat and hierarchical pixel-oriented visualization of voy dataset. In this figure date and s/c distance is strongly correlated because they have very similar attributes. There is also strong correlation between the attributes of BR in RTN, BT in RTN, BN in RTN and B Magnitude because they all are similar.

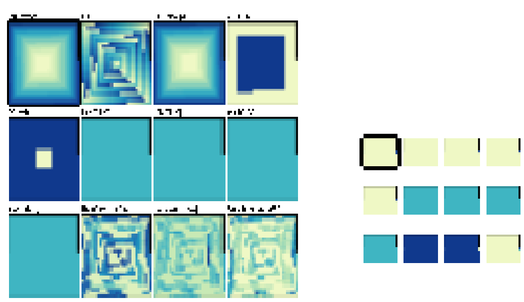

Figure 14: Flat and Hierarchical Pixel-oriented visualization of Voy dataset

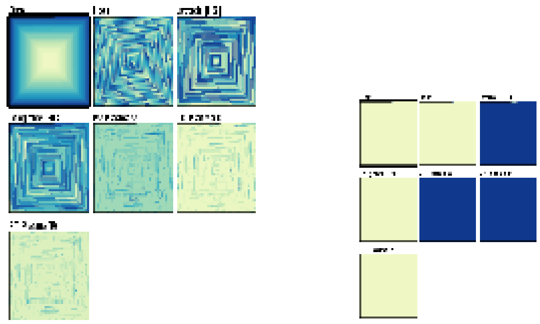

Figure 15: Flat and Hierarchical Pixel-oriented visualization of Venus dataset

There is less correlation between the attributes of Plasma Density and Plasma Temperature and weaker correlation between the attributes of Hour and Plasma Velocity. There are linear functional dependency between date and S/C Distance and no functional dependency between Date and Hour. In hierarchical pixel-oriented visualization data points are combined to form cluster. Clusters may have a lower dimensionality than the whole data set. Therefore, the clustering may appear only in the visualizations of certain attributes. So there is stronger correlation between the attributes of dimension 1, 2,3,4,5 and 12 and also between 6,7,8,9 and 10 and 11.

There is another dataset shown in figure 15 is Venus dataset. In this, visualization there is strong correlation between the attributes of latitude and longitude and weak correlation between the attributes of hour and latitude. There is also some correlation between the attributes of PD Plasma and PT Plasma. In hierarchical visualization due to low dimensionality of cluster, there is stronger correlation between the attributes of date, hour, latitude and PT Plasma and also Latitude, PV Plasma and PD Plasma.

CONCLUSION AND FUTURE WORK

Data visualization techniques are used for exploration and analysis of large amount of data. An XmdvTool is described which is used to explore multidimensional data. Pixel oriented techniques i.e. spiral technique is a useful technique because this technique avoids overlapping of data points and provide us good results. The study concludes that the ability to visualize large data sets in maximum resolution and find patterns in the data is enhanced when pixel oriented visualization is used in conjunction with other multivariate techniques. Future work will focus on the improvement of the proposed technique and its application to a variety of visualization techniques, including geometric and iconic techniques, extend the pixel-oriented technique to axes and circle segment technique in XmdvTool.

REFERENCES

[1] A. Inselberg and B. Dimsdale, Parallel coordinates: A tool for visualizing multidimensional geometry. *Proc. of Visualization '90, p. 361-78*,
1990.

[2] B. Shneiderman. Tree visualization with tree-maps: A 2d space-filling approach. *ACM Transactions on Graphics, Vol. 11(1), p. 92-99*, Jan, 1992

[3] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications, In *IEEE Transactions on Visualization and Computer Graphics*, volume 6, pages 1-20, January-March 2000

[4] D.A. Keim and H.P. Driegel, VisDB: Database exploration using multi-dimensional visualization. In *Computer Graphics & Applications*, pages 40-49, Sept. 1994

[5] D. A. Keim. Pixel-oriented visualization techniques for exploring very large databases, *J. Computational and Graphical Statistics*, 5(1):58-77, 1996.

[6] D. A. Keim, H.P. Kriegel and Mihael Ankerst, Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. of the 6th IEEE Visualization Conference*, pages 279-286, 1995

[7] D. F. Andrews Plots of high-dimensional data. *Biometrics*, 29:125-136, 1972

[8] D Hilbert, "ber stetige Abbildung einer Lmie auf ein Flchenstck," *Math Annalen,* vol *38,* pp 459-460, 1891

[9] G M. Morton, A *Computer Oriented Geodetic Data Base and a New Technique in File Sequencing* Ottawa, Canada IBM Ltd, 1966

[10] G Peano, "Sur une courbe qui remplit toute une aire plame," *Math Annalen,* vol 36, pp 157-160, 1890

[11] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d Visualization of hierarchical information, *Proc. of Computer-Human Interaction '91, p. 189-194*, 1991

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

[13] H. Levkowitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. *Proc. Visualization '91*, 1991

[14]http://davis.wpi.edu/~xmdv/index.html

[15] J. Beddow. Shape coding of multidimensional data on a microcomputer display. *Proc. Visualization '90*, pages 238-246, 1990.

[16] J. J. Wijk and R. D. van Liere, Hyperslice *Proc. Visualization '93*, Pages 119-125, 1993

[17] J. Yang, M. O. Ward, and E. A. Rundensteiner, Interactive hierarchical displays: A general framework for visualization and exploration of large multivariate data sets. *Computer and Graphics Journal*, 27:256-283, 2002.

[18] Keim D. A., Kriegel H.-P.: 'Visualization Techniques for Mining Large Databases: A Comparison*', Trans.on Knowledge and Data Engineering,* vol 8.pp.923-938, 1996.

[19] Keim, D.A.; Mansmann, F.; Schneidewind, J.; Ziegler, H.; "Challenges in Data Visualization"*10th international conference on data visualization,*pp:9-16,2006

[20] K. Jain and C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988

[21] Lida Tang, Ben Shneiderman.: *'Dynamic Aggregation to Support Pattern Discovery: A case study with web logs.*

[22] M.A.A Farque, A.S. Munni, M.M.R. Mozumdae:"A Comparative Study Multidimensional Visualization", *International Conference on Computer & Information Technology ICCIT*, vol: 1, pp: 405-409, 2003

[23] M. Ankerst, D. A. Keim, and H..P. Kriegel, Circle segments: A technique for visually exploring large multidimensional dataset. *Proc. Visualization '96*, 1996

[24] M.O. Ward, J.T. LeBlanc, and R. Tipnis, N-land: a graphical tool for exploring n-dimensional data. *CG194 Proc: Insight through Computer Graphics, p. 130-41*, 1996.

[25] M. O. Ward, J. Yang, and E. A. Rundensteiner, Hierarchical exploration of large multivariate data spaces, In *Proc. Dagstuhl Seminar on Scientific Visualization*, 2000.

[26] R. M. Pickett and G. G. Grinstein, Iconographic displays for visualizing multidimensional data. *Proc. IEEE Conf. Systems, Man and Cybernetics*, pages 514-519, 1988.

[27] R. Rao and S. Card, Exploring large tables with the table lens, Proc. of ACM CHI'95 Conference on Human Factors in Computing Systems, Vol. 2, p. 403-4, 1995

[28] S. Feiner and C. Beshers, Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. *Proceedings ACM Symposium on User Interface Software and Technology*, pages 76-83, 1990.

[29] Tatu, A.; Albuquerque, G.; Eisemann , M.; Bak, P.; Theisel, H.; Magnor, M.; Keim, D.; "Automated Analytical Methods to Support Visual Exploration of High-Dimensional Data", *IEEE Transaction on visualization and graphics,*vol:17,pp:584-597,2011.

[30] Tatu, A.; Albuquerque, G.; Eisemann, M.; Schneidewind, J.; Theisel, H.; Magnork, M.; Keim, D.; '"Combining automated analysis and visualization techniques for effective exploration of high-dimensional data"*IEEE symposium on visual analytic science and technology ,*pp: 59-66,2009.

[31] W. Wright, Information animation applications in the capital markets, *Proc. Int'l Symp. Information Visualization*, pages 19-25, 1995.

[32] Y. Fua, M.O. Ward, and E.A. Rundensteiner, Navigating hierarchies with structure-based brushes, *Proc. of Information Visualization '99, p. 58-64*, Oct. 1999.

[33] Y. Fua, M.O. Ward, and E.A. Rundensteiner, Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Visualization and Computer Graphics, Vol. 6, No.2, p. 150-159*, 2000.

[34] Y. Fua, M.O. Ward, and E.A. Rundensteiner, Hierarchical parallel Coordinates for exploration of large datasets, *Proc. of Visualization '99, p. 43-50*, Oct. 1999

[35] Y. Leung and M. Apperley, A review and taxonomy of distortion-oriented presentation techniques, ACM Transactions on Computer-Human Interaction Vol. 1(2), June 1994, p. 126-160, 1994