An Interactive Visualization Model for Large High-dimensional Datasets

Wei Ding
Division of Computing and Mathematics,University of Houston-Clear Lake,
2700 Bay Area Blvd. Houston, TX 77058. Email: ding@cl.uh.edu
Ping Chen
Department of Computer and Mathematical Sciences, University of Houston-Downtown,
One Main St. Houston, TX 77002. Email: chenp@uhd.edu

Abstract Data visualization gives a direct view of complex data, which is especially helpful for analysis of large high dimensional datasets. However, existing methods often lose simplicity and clarity while rendering large amount of complex data. In this paper, we discuss some essential properties that a data visualization system should have. Also we present an interactive data visualization model which can effectively and efficiently visualize large high dimensional datasets. We evaluate our system with an oil exploration dataset.

Keywords Interactive Visualization, Glyph, Knowledge Discovery

I. Introduction

Nowadays many companies and public organizations use powerful database systems for collecting and managing information. And huge amount of data records are often accumulated within a short period of time. Valuable information is embedded in these data, which could help discover interesting knowledge and significantly assist in decision-making process. However, human beings are not capable of understanding so many data records which often have lots of attributes. The need for automated knowledge extraction is widely recognized, and leads to a rapidly developing market of data analysis and knowledge discovery tools.

In spite of many advances from knowledge discovery and data mining area, the human eye-brain system remains the best existing pattern recognition device for information extraction, and human analysis and insight are still the most important way to interpret and utilize the knowledge obtained from automated data mining tools. Data visualization transforms data into direct views and plays a very important role in knowledge discovery. Data visualization is a rapidly expanding research area, and its techniques range from simple histogram plots to large 3D visual reality systems.

A. Related work: Traditionally, many simple methods are designed to render small amount of data or statistical features of big data sets, such as histogram, pie, tree, etc.

To visualize more complex data, modern scientific visualization utilizes more advanced techniques. Visualization techniques, such as EXVIS [10], Chernoff Faces [4], icons [13] and m-Arm Glyph [14], are often called glyph-based methods. Glyphs are graphical entities whose visual features, such as shape, orientation, color and size, are used to encode attributes of an underlying dataset, and glyphs are often used for interactive exploration of data sets [18]. Glyph-based techniques range from representation via individual icons to the formation of texture and color patterns through the overlay of many thousands of glyphs [5]. Chernoff used facial characteristics to represent information in a multivariate dataset [4]. Each dimension of the data set encodes one facial feature, such as nose, eyes, eyebrows, mouth, or jowls. Glyphmaker proposed by Foley and Ribarsky visualize multivariate datasets in an interactive fashion [9]. Levkowitz described a prototype system for combining colored squares to produce patterns to represent an underlying multivariate dataset [12]. In [13] an icon encodes six dimensions by six lines of different colors within a square icon. In [5] Levkowitz describes the combination of textures and colors in a visualization system. The m-Arm Glyph by Pickett and Grinstein [14] consists of a main axis and m arms, and the length and thickness of each arm and the angles between each arm and main axis are used to encode different dimensions of a data set. [3] describes a glyph-based system for large high dimensional datasets. These techniques are incapable of visualizing large amount of high dimensional data because:

- Lack of human computer interaction.

- Lack of integration with other data mining and knowledge discovery (KDD) tools. The goal of data visualization is to help data analysis and knowledge discovery. There are many successful techniques in KDD, and integration of these techniques will be a great benefit. In this paper we will show how clustering plays an important role in revealing interesting details of a data set.
- Incapable to deal with large amount of data. Nowadays, a data set can easily have millions or even billions of records. How can we visualize both local details and general overview of such data sets?
- Incapable to dynamically assign data dimensions to visual elements. Most existing methods use only one visual object to visualize one data record. When a record has lots of dimensions, the visual object becomes too complex (that is, many visual properties of this visual object have to be used) for human beings.

B. General requirements for a data visualization system: A visualization system should be as automatic or intelligent as possible. On the other hand, viewers should be able to fine tune the display manually as much as they want. A visualization process involves initial automatic analysis and rendering, and the following finetuning and interaction with viewers. Nowadays a visualization system should satisfy the following requirements:

- Rendering a large data set efficiently.
- Rendering a high-dimensional data set.
- Dealing with both numerical and categorical values.

To satisfy these requirements a visualization system needs to be interactive and have an open architecture for easy integration of other data analysis components. In the rest of this paper, section II explains visualization process and some related problems. Section III describes general properties for an interactive visualization system. Section IV discusses how to use summary icons to render large amount of data. Section V gives one case study based on a real oil exploration data set, and conclusion is given in section VI.

II. Data visualization process

Data visualization is a graphic presentation of a data set, with the goal of providing a viewer with a qualitative understanding of the embedded information in a natural and direct way. Graphic presentation involves the usage of visual objects and its elements. One visual element is one visual feature of a visual object. The visual objects (these objects are differentiated by their shapes and styles) could be: point, line, polyline, glyph, 2-D or 3-D surface, 3-D solid, image, text, etc. And one visual object may have the following visual elements: color, location, shape/style, texture, size, orientation, position/motion, etc. We can divide visualization process into three stages:

1) Rendering data(forward transformation) stage
2) Backward transformation stage
3) Knowledge extraction stage

A. Rendering data stage: The basic requirement for rendering data is that different values should be displayed differently, the more different the original values are, the more different they should look. There include two steps:

1) *Association step:* Associate data dimensions/columns with visual elements:

$$D = \{d_1, d_2, \cdots, d_n\}, V = \{v_1, v_2, \cdots, v_m\}, F_a : D \rightarrow V \tag{2.1}$$

where D is the set of n dimensions in a data set, and d_i is the i^{th} dimension in D; V is the space of m visual elements which include visual objects and their features, and v_j is the j^{th} element in V.
If $n < m$, some visual elements are shown but do not represent any information, and they unnecessarily attract the viewer's attention, so usually this case is undesirable.
If $n > m$, at least one visual element need encode two or more data dimensions, which will make the display hard to understand, so this case is seldom used.
If $n = m$, one visual element represents one dimension of a data set. This case is used by most visualization methods. In the rest of this paper, we only consider this approach.

2) *Transformation step:* In this step we will choose a transformation function for each dimension-visual element pair which maps each value in that dimension to a member in that visual element domain. The transformation function can be expressed as:

$$F_i : d_i \rightarrow v_i (i = 1, 2, \cdots, n) \tag{2.2}$$

where d_i is the set of values of i^{th} dimension, and v_i is the set of domain members of i^{th} visual element.

These two steps are straightforward and it seems that we can make almost arbitrary choices, however, a visualization system is a human-computer system, which brings two constraints:

1) Human eyes can not distinguish very small visual differences, so a visualization system should not use very small visual differences to carry any information.
2) Human eyes have difficulty to handle a display with overwhelmingly rich visual features, which makes understanding and extraction of information difficult and hurt the motivation of visualization.

These constraints require a visualization system designer to choose association and transformation functions carefully, and in next section we will discuss some approaches.

1) Choosing visual objects and features: Choosing visual features are very important to a visualization system. In human visual system some visual properties are processed preattentively, without the need for focused attention. Typically, tasks that can be performed on large multi-element displays in less than 200 ms to 250 ms are considered preattentive. Eye movements take at least 200 ms to initiate, and random locations of the elements in the display ensure that attention cannot be prefocused on any particular location, but usually they can be completed with very little effort. This means that certain visual information is processed in parallel by the low-level visual system [13]. If we avoid feature conjunction which inhibit a user's low-level visual system, we can develop tools making use of preattentive vision, and they can offer a number of important advantages:

- Visual analysis of preattentive tasks is rapid (within 200 ms or less), accurate and relatively effortless.
- The time required for preattentive task analysis is independent of display size, which means that more data elements in a display will not increase the time required to analyze the display. This property is especially important in data mining field that usually involves huge amount of data.

2) Non-uniform data distribution problem: In this section we will discuss why a clustering step is necessary before data is visualized. Within a data set, it is common that the data values are clustered along one or several dimensions, which means that data distribution is not uniform. The problem of non-uniform data distribution has to be handled for better visualization quality, which is shown by the following example.

Suppose we have a one-dimensional dataset as {1, 1, 1, 1, 1, 5, 10, 10, 100}, and we choose the color of icon "bar" to represent it, and our transformation function is:

$\{value|1 \leq value \leq 10\} \longrightarrow \mathrm{r}ed$

$\{value|11 \leq value \leq 20\} \longrightarrow \mathrm{o}range$

$\ldots$

$\{value|91 \leq value \leq 100\} \longrightarrow \mathrm{b}lue$

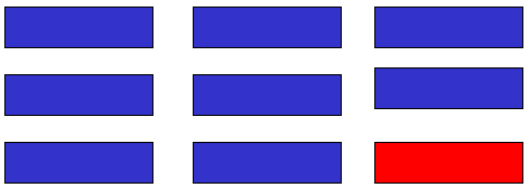

Fig. 1. The problem of non-uniform data distribution

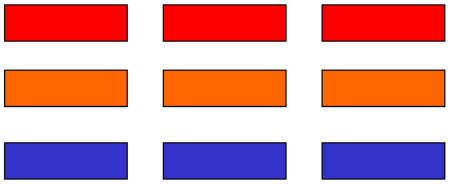

Fig. 2. The problem of non-uniform knowledge distribution

And the dataset will be visualized as Figure 1, although the visualization system can use ten different colors, most icons are blue because most data values fall into the interval $[1, 10]$ represented by blue. We can not tell the difference of these data values from the display, and such a visualization is less effective. We could use more colors, but too many colors may hurt visualization quality as we explained in Section II-A.1. Instead, a better option is to find the data clusters first for each dimension i. For a one-dimensional data set clustering we have plenty of clustering algorithms to choose from, such as BIRCH[20], Fractal Clustering[1], etc.

Let k_i be the number of clusters for the i^{th} dimension. Then, we divide v_i (set of members of i^{th} visual element used to render i^{th} dimension) into k_i groups, i.e.

$v_i = \{v_{ij}|1 \leq j \leq k_i\}$.

The transformation between the i^{th} data dimension and its visual element will be determined according to the cluster which the data value belongs to. c_{ij} denotes the j^{th} cluster of data in the i^{th} dimension, and we have:

$$C_i = \{c_{ij}\}(1 \leq i \leq n, 1 \leq j \leq k_i) \tag{2.3}$$

where C_i is the set of clusters in dimension i, n is the number of dimensions in a data set, k_i is the number of clusters in dimension i. We divide members in visual element V_i into k_i groups:

$$V_i = \{v_{ij}\}(1 \leq i \leq n, 1 \leq j \leq k_i) \tag{2.4}$$

where v_{ij} is a group of members in visual element i. Number of members in v_{ij} should be proportional to number of members of c_{ij}. For example, if we choose visual element "bar size", as shown in Figure 3, we could divide different sized bars into three groups, and each group has three members of visual element "bar size".

Then the transformation between data dimensions and visual elements will be:

$$F_{ij} : C_{ij} \rightarrow V_{ij}(1 \leq i \leq n, 1 \leq j \leq k_i) \tag{2.5}$$

With a clustering step we can assign members of visual elements more reasonably to data values, and a visualization system is able to reveal more information from a data set.

B. Backward transformation stage: Viewers have to be aware of and understand the association and transformation steps during the "Rendering data" stage, and be able to reverse the transformed display and restore the original picture in their mind. Such a backward transformation process is done solely by human beings, so it can not be too complex. Otherwise the user can not understand what the display represents and a visualization process will be useless. This requirement makes a complex association or transformation in the "Rendering data" stage infeasible, and it is why most visualization systems associate one visual element to only one data dimension and why studying of human visual system is very important.

C. Knowledge extraction stage: Rendering millions of icons is computationally expensive, and interpretation and analysis to be performed by the user is even harder. A visualization system has to provide not only a "loyal" picture of the original dataset, but also an "improved" picture to a viewer for easier interpretation and knowledge extraction. Integration of analysis functionality is important and necessary to help the viewer to extract knowledge from the display. In section II-A we specified the basic requirement about a visualization system as:

"Different data values should be visualized differently, and the more different the data values are, the more different they should look".

But what a viewer wants to find with a visualization system is not data values themselves, instead, it is the information or knowledge represented by data values. So, the above requirement can be better stated as:

"Different information should be visualized differently, and the more different the information is, the more different it should look".

To help a viewer on knowledge extraction a visualization system has to deal with the problem of non-uniform knowledge/information distribution. It is common in some data sets or fields that a small difference of a value could mean a big difference, which means the knowledge and information is not distributed uniformly within data values. A user would like a visualization system to be able to show these knowledge differences clearly. To be specific, two differences of same amount in data values may not necessarily be rendered by the identical difference in visual elements on the screen. Instead the difference representing more information should be displayed more significantly to get attention from a viewer. We give an example as follows.

Suppose we have a one-dimensional data set which saves human body temperatures, {36.5, 37.0, 37.5, 38.0, 38.5, 39.0, 39.5, 40.0, 40.5, 41.0, 41.5, 42.0}, and this data set is uniformly distributed. We still use a bar's color to visualize the data set, and after a clustering step our transformation function will map the values uniformly since the dataset has a uniform distribution:

$\{value|36.0 \leq value < 38.0\} \longrightarrow$ red

$\{value|38.0 \leq value < 40.0\} \longrightarrow$ o$range$

$\{value|40.0 \leq value \leq 42.0\} \longrightarrow$ blue

And the dataset will be visualized in Figure 2. The visualization system visualizes the data loyally. Both 40.0 and 42.0 are represented by blue, but as human body temperatures 40.0 and 42.0 could mean a difference of life and death, and this important information is lost. This example clearly show how important it is to integrate

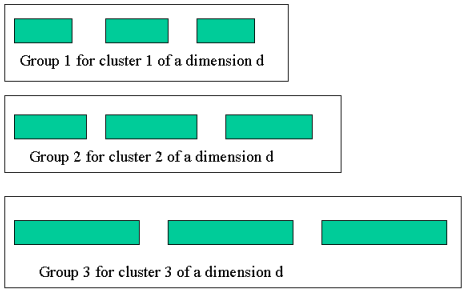

Fig. 3. We divide members of "bar size" into three groups, each group has three members.

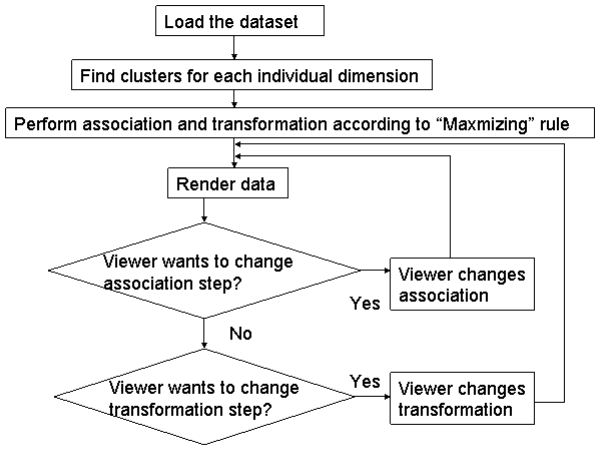

Fig. 4. An interactive visualization system model

domain knowledge in a visualization system. Such an integration can be achieved through interaction between viewers and a visualization system which will be discussed in next section.

III. Interactive visualization model

In Figure 4 we give an interactive visualization model which has the following properties:

1) Interaction: From the example in Section II-C, it is clear that integration of domain knowledge to a visualization system is very important due to the problem of non-uniform knowledge distribution. To a visualization system integration of domain knowledge can be achieved by choosing proper association function and transformation function during visualization process. However, there is no universal technique for all fields, data sets or users, and a visualization system should be interactive and provide a mechanism for views to adjust or change association and transformation functions during visualization process. And each data set or field has to be studied individually and visualized interactively before its important information can be revealed, which can only be performed by viewers or domain experts. By interaction a viewer can guide a visualization system step by step to display what he is interested in more and more clearly.
2) Correctness: We propose the following criteria for "correct" visualization:
   a) If possible a visualization system should show different dimensions of a data set differently through different visual objects or visual elements of one visual object.
   b)The more different the values are, the more differently they should be rendered. Since we may not know the distribution of a dataset, assigning data values to visual elements/properties may not make full usage of available visual elements/properties, a clustering step is preferred.
   c)The more different the information represented by data values are, the more differently they should be rendered. A distinguished visual difference between different information can help viewers better, which can be achieved by interaction between a visualization system and viewers. In this interaction process, viewers can finetune the transformation between data values and visual elements, and domain knowledge is obtained and reflected through a more customized display.
3) "Maximizing" rule: To optimize the rendering quality, the maximal range of visual objects/elements should be used as default settings.

IV. Utilizing summary icons

With multiple icons located in one position we can visualize a data set with high dimensions. In this section we discuss how to render a large dataset effectively. The first option is to simply display all data records, but:

1) Rendering a large number of icons at a time will make the icons indistinguishable
2) Rendering a huge set is computationally expensive

The second option is sampling. Basically, we have the following choices:

- A viewer specifies a value range for each dimension, only icons fall into this range can be displayed
- A viewer specifies the types of dimensions/icons to be displayed
- A viewer chooses a sampling rate to choose and display data records randomly
- Some domain-related criteria, such as choosing icons between two horizons in geophysics

However, sampling has the disadvantage of potentially missing infrequently occurring details of a data set, which may be of user's interests. Also sampling can not provide an overall picture of a data set all the time.

To display the local details and overall context of a data set at the same time, we use summarization. We use "summary" icons to display summarized data for "uninteresting" parts of a dataset, and regular icons to display the "interesting" parts of a data set which will show all details. One feature of a summary icon do not represent one field in a data record, instead it represents a statistical parameter (summary) of the fields from multiple underlying data records, such as sum, mean, median. By this way, we can build a hierarchical structure of icons as in Figure 5. The icons in low level represent only one record, the icons in high level will be a summary of icons/records below it. The icons on the high level are more general, they summarize information from a lot of records, and the icons on the low level are more specialized or local, and they represent and visualize only one record. We show how to use summary icons in visualization by an example.

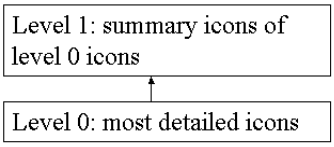

Fig. 5. Hierarchical structure of icons

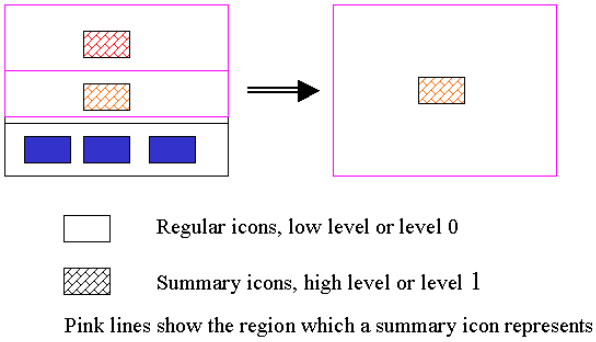

Fig. 6. Figures with summary icons

We still use the one-dimensional human body temperature set we displayed in Figure 2. But we use some summary icons to summarize some data values shown in Figure 6. The left figure in 6 shows two level-one summary icons, the top bar represents the average of first three values and middle bar represents the average of value 4 ,5 and 6. The right figure in 6 use one summary icon to represent the average of all data values in this set. Summarization can be very flexible, we can assign higher weight to regular icons whose values a viewer is more interested in, so they will be shown more significantly in summary icons, and it is easier for a viewer to notice these interesting data.

V. Case study

In this case study we visualize a large dataset that encodes multiple data fields at a single spatial location. This set of 12-dimensional geophysical data was obtained with man-made earthquakes to discover oil underground. These data is recorded as nine SGY files. Each file includes some headers and 6,172,871 one-dimensional records. These records are data samples of signals from 111 $\times$ 111 locations within 2 seconds after an explosion. The sampling rate is 4 ms. Data recorded in the 9 SGY files represent three different properties in geophysical science, which are interval velocity, amplitude of the 5-45 degree angles of incidence and amplitude of the 35-55 degree angles of incidence. Each property has three dimensions(nine dimensions in total). To represent these properties in 3D space, we used three different 3D icons: parallelogram, box, and pyramid. In Table 7, we list features of each icon and the data dimension they represented. And in Figure 8 six records are rendered for illustration purpose. We do not display any records with fast interval velocity equal to 0, and the number of records we need show is reduced to 4,262,747. The loading time is 149 seconds. And view rendering (move, rotate, zoom) can be done in real time. In Figure 9 we performed clustering for each data dimension and displayed data with summary icons, so a viewer can have a general idea about the data directly. If a user is interested in a specific area, he can drill to that area and have a detailed display similarly as Figure 8.

Icons	Visual features	Data dimensions
parallelogram		Interval Velocity
	size	Fast Interval Velocity
	orientation	Azimuth of the fast interval velocity
	color	(Fast-Slow) Interval Velocity
box		Amplitude of the 5-45 degree angles of incidence
	size	Large Amplitude Variation with Offset (AVO) Gradient
	orientation	Azimuth of the large AVO Gradient
	color	Azimuthal variation in the Gradient (Large minus small)
pyramid		Amplitude of the 35-55 degree angles of incidence
	size	Large Amplitude Variation with Offset Gradient
	orientation	Azimuth of the large AVO Gradient
	color	Azimuthal variation in the Gradient (Large minus small)

Fig. 7. Association between dimensions and visual elements

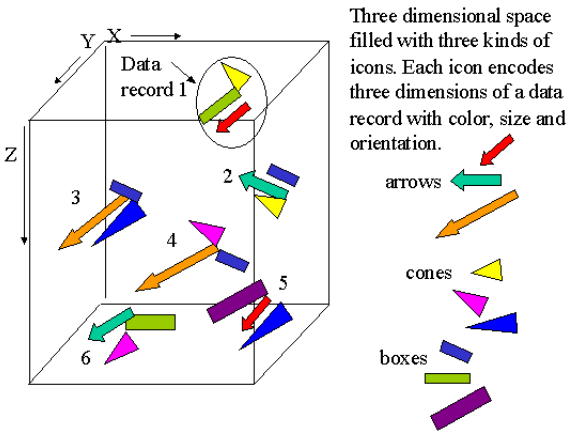

Fig. 8. A sample figure to visualize a twelve-dimensional data set with six records. Each icon uses color, size and orientation to represent three dimensions of a data record, three icons located in the same position can represent nine dimensions of a data record, the position itself can encode three dimensions, so a group of three icons can represent twelve dimensions. Totally there are six icon groups, which represent six data records in the data set.

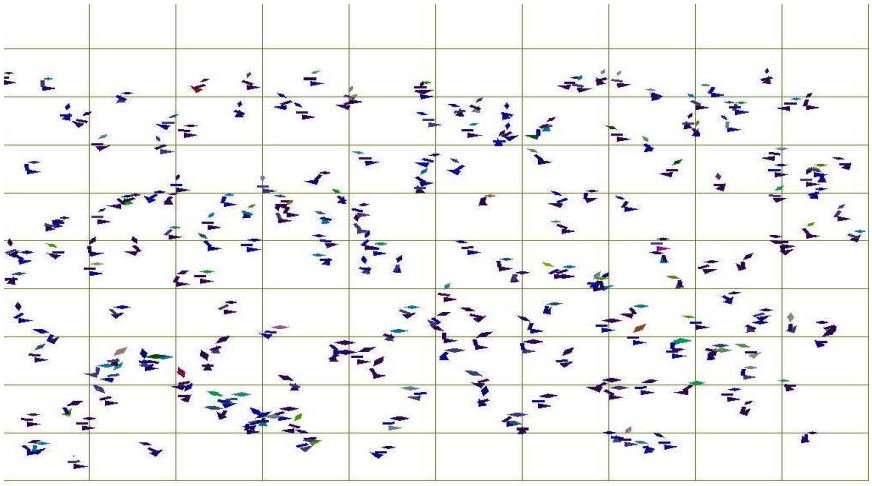

Fig. 9. A screen capture of the display which visualizes the samples of a data set(Grids are drawn to help locate icons.)

VI. Conclusion

In this paper we examine some important properties of a visualization system. We propose an interactive visualization model, and we discuss how a clustering step and interaction between viewers and a visualization system can solve the problems of non-uniform data distribution and non-uniform knowledge distribution. We implemented our interactive model and show its effectiveness with a case study.

References

[1] Barbara, D., Chen, P., Using Self-Similarity to Cluster Large Data Sets, Data Mining and Knowledge Discovery 7(2): 123-152, 2003

[2] Buja, A., Cook, D., and Swayne, D. F., Interactive high-dimensional data visualization. Journal of Computational and Graphical Statistics 5, pages 78-99, 1996.

[3] Chen, P., Hu, C., Ding, W., Lynn, H., Yves, S.,Icon-based Visualization of Large High-Dimensional Datasets, Third IEEE International Conference on Data Mining, Melbourne, Florida, November 19-22, 2003.
[4] Chernoff, H. The use of facesto represent points in k-dimensional space graphically. Journal of the American Statistical Association 68, 342, pages 361-367, 1973.
[5] Christopher, G. Healey, James T. Enns, Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. IEEE Transactions on Visualization and Computer Graphics, Volume 5, Issue 2, 1999.
[6] Ebert, D.,Rohrer,R.,Shaw, C.,Panda, P.,Kukla, D.,Roberts,D., procedural shape generation for multi-dimensional data visualization. Computers and Graphics, Volume 24, Issue 3, Pages 375-384, June 2000.
[7] Elizabeth M. Wenzel , Frederic L. Wightman , Scott H. Foster, Development of a three-dimensional auditory display system. ACM SIGCHI Bulletin, v.20 n.2, pages 52-57, Oct. 1988.
[8] Enns, J. T. Three-Dimensional Features that Pop Out in Visual Search. In Visual Search, Brogan, D., Ed., Taylor and Francis, New York, New York, pages 37-45, 1990.
[9] Foley, J., and Ribarsky, W. Next-generation data visualization tools. Scientific Visualization: Advances and Challenges, L. Rosenblum, Ed. Academic Press, San Diego, California, pages 103-127, 1994.
[10] Grinstein, G. G., Pickett, R. M. and Williams, M., EXVIS: An Exploratory Data Visualization Environment. Proceedings of Graphics Interface ’89 pages 254-261, London, Canada, 1989.
[11] Julesz, B. and Bergen, J.R. Textons, the Fundamental Elements in Preattentive Vision and Perception of Textures. The Bell System Technical Journal 62, 6, pages 1619-1645 ,1983.
[12] Laidlaw, D. H., Ahrens, E.T., Kremers, D., Avalos, M.J., Jacobs, R.E., and Readhead, C. Visualizing diffusion tensor images of the mouse spinal cord. Proceedings of Visualization ’98, pages 127-134, 1998
[13] Levkowitz, H. Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameter, Proceedings of IEEE Visualization’91 Conference, San Diego, CA, Oct. 1996
[14] Pickett, R. M. and Grinstein, G. G., Iconographics Displays for Visualizing Multidimensional Data. IEEE Conference on Systems, Man and Cybernetics. China, 1988.
[15] Triesman, A. and Gormican, S. Feature Analysis in Early Vision: Evidence from Search Asymmetries. Psychological Review 95, 1, pages 15-48, 1988.
[16] Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., Koudas, N.. Non-Linear Dimensionality Reduction Techniques for Classification and Visualization. KDD ’02, Edmonton, Canada, 2002.
[17] Ward, M. O., Xmdvtool: Integrating multiple methods for visualizing multivariate data. In Proceedings of Visualization ’94, pages 326-333, October 1994.
[18] Wegenkittl, R., Lffelmann, H., Grller, E., Visualizing the behavior of higher dimensional dynamical systems. Proceedings of the conference on Visualization ’97, 1997 , Phoenix, Arizona, United States
[19] Wong, P., Bergeron, R., 30 years of multidimensional multivariate visualization, In G. M. Nielson, H. Hagan, and H. Muller, editors, Scientific Visualization Overviews, Methodologies and Techniques, Los Alamitos, CA, 1997.
[20] Zhang, T., Ramakrishnan, R., and Livny, M. BIRCH: An efficient data clustering method for very large databases. In Proc SIGMOD96, Montreal, Canada, 1996.

Wei Ding is a Ph.D. student in Computer Science at University of Houston. Her research interests include data mining and machine learning. Ms. Ding received her BS degree on Computer Science from Xi’an Jiao Tong University in 1993 and MS degree on Software Engineering from George Mason University in 2000.

Dr. Ping Chen is an Assistant Professor in Department of Computer and Mathematics Science of University of Houston-Downtown. His research interests include computational semantics and text mining. Dr. Ping Chen received his BS degree on Information Science and Technology from Xi’an Jiao Tong University, MS degree in computer science from Chinese Academy of Sciences, and Ph.D degree on Information Technology at George Mason University