

Icon-based Visualization of Large High-Dimensional Datasets

Ping Chen* Chenyi Hu† Wei Ding‡ Heloise Lynn, Yves Simon§

Abstract

High dimensional data visualization is critical to data analysts since it gives a direct view of original data. We present a method to visualize large amount of high dimensional data. We divide dimensions of data into several groups. Then, we use one icon to represent each group, and associate visual properties of each icon with dimensions in each group. A high dimensional data record will be represented by multiple different types of icons located in the same position. Furthermore, we use summary icons to display local details of viewer's interests and the whole data set at meantime. We show its effectiveness and efficiency through a case study on a real large data set.

1 Introduction

Data visualization plays an important role in discovering knowledge since the human eye-brain system is still the best existing pattern recognition device. Data visualization is a rapidly expanding research area due to the huge increase of number and size of datasets that need to be visualized and interpreted. Data visualization techniques may range from simple scatter plots and histogram plots over parallel coordinates to 3D visual reality systems.

Visualization techniques, such as EXVIS [8], Chernoff Faces [1] [3], icons [5] and m-Arm Glyph [7], are often called glyph-based methods. Glyphs are graphical entities whose visual features such as shape, orientation, color and size are controlled by attributes in an underlying dataset, and glyphs are often used for interactive exploration of data sets [9]. 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 [2]. Chernoff used facial characteristics to represent information in a multivariate dataset [1] [3]. Each dimension of the data set controls one facial feature such as nose, eyes, eyebrows, mouth, and jowls. Glyphmaker proposed by Foley and Ribarsky can visualize multivariate datasets in an effective, interactive fashion [4]. Levkowitz described a prototype system for combining colored squares to produce patterns to represent an underlying multivariate dataset [6]. In [5] an icon encodes six dimensions by color coding six different lines within a square icon. In [2] Levkowitz describes the combination of textures and colors in a visualization system. The m-Arm Glyph by Pickett and Grinstein [7] 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.

2 Visualizing high dimensional data

Data visualization is the graphic presentation of a data set, with the goal of helping and providing the viewer with a qualitative understanding of the embedded information in a natural and direct way. And a visualization process includes Rendering data(forward transformation) stage, Reverse transformation stage, Knowledge extraction stage.

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

1. Association step

Associate data dimensions/columns with visual elements. The association is as the following:

$$\begin{aligned} D &= \{d_1, d_2, \dots, d_n\} \\ V &= \{v_1, v_2, \dots, v_m\} \\ F_a &: D \rightarrow V \end{aligned} \tag{1}$$

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

*Department of Computer and Mathematical Sciences, Univ. of Houston-Downtown, 1 Main St. Houston, TX 77002

†Computer Science Department, Univ. of Central Arkansas, 201 Donaghey Ave. Conway, AR 72035

‡Division of Computing and Mathematics, Univ. of Houston-Clear Lake, 2700 Bay Area Blvd. Houston, TX 77058

§Lynn Inc., 14732 F Perthshire Rd., Houston, TX 77079

of m visual elements which include visual objects and their features, and v_j is the j^{th} element in V .

2. Transformation step

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 function is:

$$F_i : d_i \rightarrow v_i (i = 1, 2, \dots, n) \quad (2)$$

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

During the association step of rendering data a visualization system associates visual elements with dimensions from a data set. 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 for each visual object, we may choose from the following visual features, color, location, shape/style, texture, size/length/width/depth, orientation, relative position/motion, etc.

Existing methods use only one icon's visual features to represent a data record, as the dimensions of the data record become higher, more features from the icon has to be used if no dimension reduction techniques adopted, which results in a complex icon. A complex icon is hard to understand and computationally expensive, which will hurt visualization quality and make large data set visualization impractical.

In our technique, we use a group of icons' visual features to represent one data record. This group of icons are located in the same position to tell a viewer that this whole group represent the same data record. Icons in one group should be of different types (shapes). And for each icon of this group, we associate one of its selected visual features with one dimension in a data set. In three-dimensional space the position of one icon can represent three dimensions, but since all icons in the group will share the same position, only three dimensions can be represented by the position of a icon group. If there are some identical triples in the three dimensions of a data set selected to be displayed by position, we can not use position to encode any dimensions any more. Instead we only associate dimensions with icon visual features and display groups of icons uniformly (or any way specified by a viewer) in the space, in this case, icon positions do not represent any information except icons in the same position are for the same data record. Suppose the number of selected icon types is N , and the number of selected features from each icon is M , and we are able to choose three dimensions which

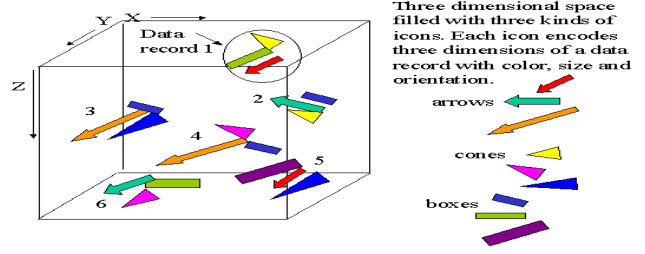


Figure 1. A sample figure to visualize a twelve-dimensional data set with six records.



Figure 2. A visualization system can not reach its potential with the problem of non-uniform data distribution

do not have identical triples, the number of dimensions which can be displayed is: $3 + M \times N$. If we want the user to be able to associate the features of icons with original data dimensions easily, M and N can not be too large. Our estimation of M and N could be up to around 10, and our technique could handle up to around 50 dimensions.

If we choose color, size and orientation as the visual features for icon "box", "arrow" and "cone", we will have a sample figure as Figure 1. By using multiple icons located in the same position our method can effectively visualize a data set with higher dimensions than existing methods. Within the data set, it is common that the data values are clustered, and the data distribution is not uniform. Non-uniform data distribution can hurt our visualization efforts, which is shown by the following example. Suppose we have a one-dimensional dataset as $\{1, 1, 1, 1, 5, 10, 10, 100\}$, and we choose the color of icon "bar" to represent it, and our transformation function is:

$$\{value | 1 \leq value \leq 10\} \rightarrow red$$

...

$$\{value | 91 \leq value \leq 100\} \rightarrow blue$$

And the dataset will be visualized as Figure 2, 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 any more, and visualization is less effective. Instead, we find the data clusters first for each dimension i with data clustering techniques. For one-dimensional data set clustering we have lots of clustering algorithms to choose from.

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

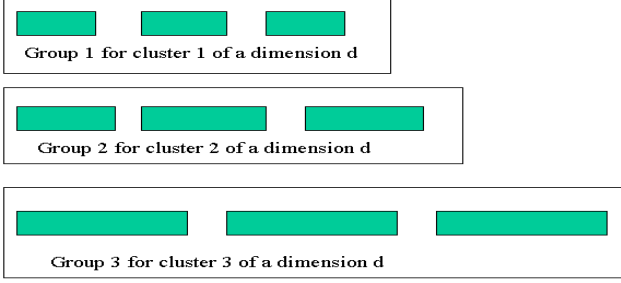


Figure 3. We divide members of “bar’s size” into three groups, each group has three members.



Figure 4. Failure to reveal critical information from a data set with non-uniform knowledge distribution problem

element) 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. Let us use c_{ij} to denote the j^{th} cluster of data in the i^{th} dimension, then we have:

$$C_i = \{c_{ij} | (1 \leq i \leq n, 1 \leq j \leq k_i)\} \quad (3)$$

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:

$$V_i = \{v_{ij} | (1 \leq i \leq n, 1 \leq j \leq k_i)\} \quad (4)$$

where v_{ij} is a group of members in visual element i . For example, if we choose visual element “bar’s 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’s size”.

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

$$F_{ij} : C_{ij} \rightarrow V_{ij} (1 \leq i \leq n, 1 \leq j \leq k_i) \quad (5)$$

In “Rendering data” stage we perform association and transformation on the original data and then perform rendering. Viewers have to be aware of and understand the association and transformation during the visualization process, and be able to reverse the transformed display and restore the original picture in their mind.

This requirement makes a complex transformation in the first stage infeasible.

Rendering millions of icons is computationally expensive, and interpretation and analysis done by the user is even harder. A visualization system has to provide not only a “loyal” picture of the original dataset, but also a “better” picture for easier interpretation and knowledge extraction. Previously we specified the basic requirement for 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 really want is the information or knowledge represented by the 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. Of course, a user would like a visualization system to be able to show these meaningful 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. 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 our transformation function will map the values uniformly since the dataset has a uniform distribution:

$$\begin{aligned} \{value | 36.0 \leq value < 38.0\} &\rightarrow red \\ \{value | 38.0 \leq value < 40.0\} &\rightarrow orange \\ \{value | 40.0 \leq value \leq 42.0\} &\rightarrow blue \end{aligned}$$

And the dataset will be visualized as the above figure. 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. From the above example, it is clear that integration of domain knowledge into a visualization system is very important due to non-uniform knowledge distribution. To a visualization system integration of domain knowledge can be achieved by choosing proper association function and transformation functions during visualization process.

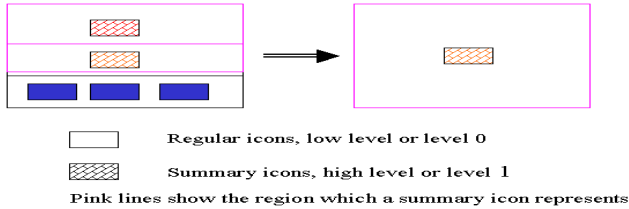


Figure 5. Figures with summary icons

3 Summary icons and a case study

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.

Using temperature set Figure 4, we use some summary icons to summarize some data values shown in Figure 5. The left figure in 5 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 5 use one summary icon to represent the average of all values in this set.

In case study we use 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 were recorded as nine SGY files. Each file includes some headers and 6,172,871 one-dimensional records. These records are data samples from 111×111 locations within 2 seconds after an explosion. Data represents 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. We used three different icons: parallelogram, box, and pyramid. The experiment is run on a PC with Pentium III 1GHz CPU, 256 MB RAM, and a 16 MB video card. View rendering (move, rotate, zoom) can be done in real time.

4 Conclusion

Using multiple icons located in one position is an effective and efficient method for large high dimensional data set visualization. Summary icons can help display local data details and overall context at the same time.

References

- [1] Bruckner, L.A., On chernoff faces. In Graphical Representation of Multivariate Data, P.C.C. Wang, Ed. Academic Press, New York, New York, pages 93-121, 1978.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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
- [6] 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
- [7] Pickett, R. M. and Grinstein, G. G., Iconographics Displays for Visualizing Multidimensional Data. Proceedings of the 1988 IEEE Conference on Systems, Man and Cybernetics. Beijing and Shenyang, People's Republic of China, 1988.
- [8] 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.
- [9] 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