Proc. Visualization '96, Hot Topic Session, San Francisco, CA, 1996.

'Circle Segments': A Technique for Visually Exploring Large Multidimensional Data Sets

Mihael Ankerst, Daniel A. Keim, Hans-Peter Kriegel

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 Munich, Germany
{ankerst, keim, kriegel}@informatik.uni-muenchen.de

ABSTRACT

In this paper, we describe a novel technique for visualizing large amounts of high-dimensional data, called 'circle segments'. The technique uses one colored pixel per data value and can therefore be classified as a pixel-per-value technique [Kei 96]. The basic idea of the 'circle segments' visualization technique is to display the data dimensions as segments of a circle. If the data consists of k dimensions, the circle is partitioned into k segments, each representing one data dimension. Inside the segments, the data values belonging to one dimension are arranged from the center of the circle to the outside in a back and forth manner orthogonal to the line that halves the segment. Our first results show that the 'circle segment' technique is very powerful for visualizing large amounts of data, providing more expressive visualizations than other well-known techniques such as the 'recursive pattern' technique and traditional 'line graphs'.

1. Introduction

One of today's problems in exploratory data analysis is the rapidly increasing amount of data that needs to be analyzed. The automation of activities in business, engineering, science, and government produces a rapidly increasing stream of data which is usually stored in large databases. Unless we find effective ways for exploring the databases, however, the collection of the data is useless and the databases become data dumps. Data exploration of very large databases is a difficult task and many researchers working in the data mining area are currently trying to find possibilities for extracting useful information from large databases.

Well-known techniques for visualizing large amounts of multidimensional data which have no standard mapping into the Cartesian coordinate system are: geometric projection techniques such as scatterplot matrices and coplots [Cle 93], parallel coordinates [ID 90], and others (e.g., [AC 91]), iconic display techniques (e.g., [Che 73, PG 88]), hierarchical techniques (e.g., [Shn 92]), dynamic techniques (e.g., [BMMS 91, AWS 92]), and combinations hereof. The research in this area also resulted in data analysis and exploration systems which implement some of the mentioned techniques. Examples include statistical data analysis packages such as S Plus/Trellis [BCW 88], visualization oriented systems such as ExVis [GPW 89] and XmdvTool [Ward 94], as well as database oriented systems such as TreeViz [Shn 92], the Information Visualization and Exploration Environment (IVEE) [AW 95], and the VisDB system [KK 95].

In most of the approaches proposed so far, the number of data items that can be visualized on the screen at the same time is still quite limited (in the range of 100 to 5,000 data values). In our work, we focus on visualization techniques that allow a visualization of much larger amounts of data. The basic idea of our pixel-per-value techniques is to map each data value to a colored pixel and present the data values belonging to each of the dimensions in separate portions of the screen. Since in general our techniques only use one pixel per data value, the techniques allow us to visualize the largest amounts of data which are possible on current displays (up to about 1,000,000 data values). If each data value is represented by one pixel, the main question is how the pixels are arranged on the screen. Our previous work focuses on supporting the data exploration and analysis process by providing query-dependent visualizations of the data, presenting the most relevant data items in the center of the display [Kei 94]. In this paper, we propose a new pixel-per-value technique for visualizing large amounts of multidimensional data, called 'circle segments'. In the rest of this paper, we describe the 'circle segments' technique, provide examples of visualizations, and compare it to traditional 'line graph' visualizations.

2. The 'Circle Segments' Visualization Technique

The fundamental idea of the 'circle segments' visualization technique is to display the data dimensions as segments of a circle. If

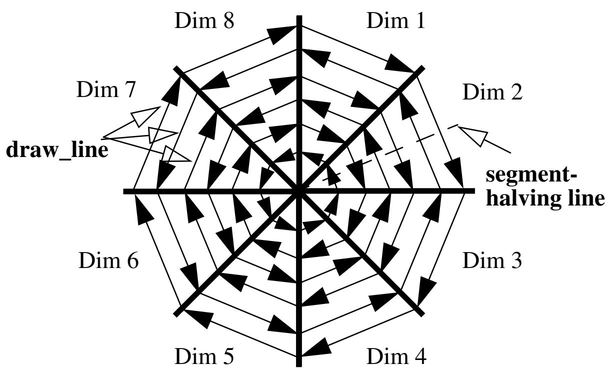

Figure 1: 'Circle Segments' Technique for 8-dimensional Data

```
void fill_segment(line l1,line l2)
{int x,y,direction = 1,
int record_count = initial_pixels(l1, l2, x, y);
    while (record_count<RECORD_ALL)
    {   while ((point_betw_lines(l1,l2,x,y)) && (record_count<RECORD_ALL))
        {   record_count++;
            Setpixel(x,y,color);
            draw_line.compute_next_point(x,y,direction);
        }
    draw_line.move();
    draw_line.compute_next_point(x,y,direction);
    direction *= -1;
    while(!point_betw_lines(l1,l2,x,y))
        draw_line.compute_next_point(x,y,direction);
    }
}
```

Figure 2: 'Circle Segments' Algorithm

the data consists of k dimensions, the circle is partitioned into k segments, each representing one data dimension. The data items within one segment are arranged in a back and forth manner along the so-called 'draw_line' which is orthogonal to the line that halves the two border lines of the segment (cf. figure 1). The 'draw_line' starts in the center of the circle and draws the pixels from one border line of the segment to the other. Whenever the 'draw_line' hits one of the border lines, the 'draw_line' is moved in parallel along the segment-halving line to the outside of the circle and the direction of the 'draw_line' changes. This process is repeated until all data items of one dimension are visualized and then the whole procedure is restarted for the remaining dimensions.

The algorithm (cf. figure 2) is called with the two border lines of the segment, and first steps into the subfunction 'initial_pixels'. The function 'initial_pixels' draws the first pixels of a segment until the following 'draw_lines' have at least one pixel between the two border lines. The return value of 'initial pixels' is the number of pixels already drawn. The 'initial_pixels' function is necessary — especially for data sets with many dimensions — since the following part of the algorithm assumes that there are 'drawable pixels' on subsequent 'draw_lines'. The function 'draw_line.compute_next_point' — implemented using a variant of the Bresenham-algorithm to avoid float operations — moves ahead on the 'draw_line'. The function 'point_betw_lines' checks whether a point is still in the segment. If the point is not in the segment any more, the 'draw_line' is moved one pixel to the outside in parallel along the segment-having line. The new 'draw_line' draws the pixels in the opposite direction of the previous 'draw_line'. Note that the 'circle segments' technique requires that the data set consists of at least three dimensions.

An additional feature of the 'circle segments' technique is that the assignment of dimensions to the segments of the circle can be changed by the user. Our first experience shows that this possibility is very important because changing the order of the dimensions helps the user to compare related dimensions and to group relevant dimensions for further analysis.

3. Comparing the 'Circle Segment' Technique with other Techniques

We tested the 'circle segments' technique and compared it to other pixel-per-value techniques as well as traditional 'line graphs'. The data base for our experiments is a stock exchange database containing 10 years of stock data (5,328 data records) from the Frankfurt stock exchange. For our comparison with the 'line graph' visualization, we used seven stock price developments. In the example visualization (cf. figure 4), seven different colors are assigned to the lines corresponding to the seven stock prices. Due to the width of the screen this technique is limited to represent at most about 1,000 records, which means in our example only every fifth database entry. The information the user is able to derive from the 'line graph' visualization depends on the degree of overlap. In the example, the user is able to see that at the beginning one stock price development has a high fluctuation and largely differs from the others. It is also easily perceivable that right after half of the time period most of the stock prices have their peaks.

When using the 'circle segments' technique on the same data set (cf. figure 3[1]), the oldest data items for all dimensions are in the middle of the 'circle' and the most recent ones are at the outside. The coloring maps high data values to light colors and low data values to dark colors, so the user gets an intuitional view of the represented data set. In comparison to the 'line graph' visualization, the 'circle segments' technique shows much better that the fifth, sixth and seventh stock all have its highest price right at the end of the visualized time period. In addition, with our technique the user is able to easily follow the development of the stock prices and to find analogical tendencies between the dimensions which are not easily detectable in the 'line graph' visualization. For example, the first and the second stock show a similar development in recent times, the fifth and sixth stock progress similarly in the first half of the time period, and the third and forth stock both have a high price at the same time with the price of the third stock remaining high a bit longer.

Figure 5 shows our technique with 50 stock prices from the Frankfurt stock index, representing about 265,000 data values. Because of the high degree of overlap, 'line graphs' are not suitable for visualizing this many dimensions, and therefore we compared the 'circle segments' technique with other pixel-per-value techniques such as the 'spiral' [Kei 94] and 'recursive pattern' techniques [KKA 95]. The main advantage of our new technique is that the overall representation of the whole data set is better perceivable — including potential dependencies, analogies, and correlations between the dimensions. According to our experience in working with the VisDB system which implements different pixel-per-value techniques, the 'circle segments' technique is the technique which is best suited for exploratory data analysis in high-dimensional data space.

1. Unfortunately, structures in the visualizations which are easy to perceive in the color version are difficult to perceive in the B/W version. A color postscript version of the paper may be obtained from our ftp-server (URL: 'ftp://arcadia.informatik.uni-muenchen.de/pub/local/dbs/pubs/Vis96.ps'). Readers who do not have access to the world wide web may obtain a paper version upon request.

4. Conclusions

In this paper, we introduced the ‘circle segments’ visualization technique as an approach for visualizing large amounts of data. Using our technique, the user may generate visualizations of very large amounts of multidimensional data, providing a good overview of the data. Our first experiments show that the ‘circle segment’ technique is very powerful for visualizing large amounts of data and provides more expressive visualizations than previous pixel-per-value techniques. A further advantage of our technique is that it allows the user to control the arrangement of the dimensions, which is important especially for comparing multiple dimensions. In our future work, we will apply the ‘circle segments’ technique in different applications to explore its strengths and weaknesses and to further improve the technique.

REFERENCES

[AC 91] Alpern B., Carter L.: *‘Hyperbox’*, Visualization ‘91, San Diego, CA, 1991, pp. 133-139.

[AW 95] Ahlberg C., Wistrand E.: *‘IVEE: An Environment for Automatic Creation of Dynamic Queries Applications’,* Proc. ACM CHI Conf. Demo Program (CHI95), 1995.

[AWS 92] Ahlberg C., Williamson C., Shneiderman B.: *‘Dynamic Queries for Information Exploration: An Implementation and Evaluation’,* Proc. ACM CHI Int. Conf. on Human Factors in Computing, Monterey, CA, 1992, pp. 619-626.

[BCW 88] Becker R., Chambers J. M., Wilks A. R.: ‘T*he New S Language*’, Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, CA, 1988.

[BMMS 91] Buja A., McDonald J. A., Michalak J., Stuetzle W.: *‘Interactive Data Visualization Using Focusing and Linking’*, Proc. Visualization ‘91, San Diego, CA, 1991, pp. 156-163.

[Che 73] Chernoff H.: *'The Use of Faces to Represent Points in k-Dimensional Space Graphically',* Journal Amer. Statistical Association, Vol. 68, 1973, pp. 361-368.

[Cle 93] Cleveland W. S.: *‘Visualizing Data’,* AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit NJ, 1993.

[GPW 89] Grinstein G., Pickett R., Williams M. G.: *‘EXVIS: An Exploratory Visualization Environment’,* Proc. Graphics Interface ‘89, London, Ontario, Canada, 1989.

[ID 90] Inselberg A., Dimsdale B.: *‘Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry’*, Visualization ‘90, San Francisco, CA, 1990, pp. 361-370.

[Kei 94] Keim D. A.: *‘Visual Support for Query Specification and Data Mining’*, Ph.D. thesis, University of Munich, July 1994, Shaker Publishing Company, 1995.

[Kei 96] Keim D. A.: *‘Pixel-oriented Visualization Techniques for Exploring very large Databases’*, Journal of Computational and Graphical Statistics, March 1996.

[KK 95] Keim D. A., Kriegel H.-P.: *‘VisDB: A System for Visualizing Large Databases’,* System Demonstration, Proc. ACM SIGMOD Int. Conf. on Management of Data, San Jose, CA, 1995, p. 482.

[KKA 95] Keim D. A., Kriegel H.-P., Ankerst M.: ‘*Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data*’, Proc. Visualization ‘95, Atlanta, GA, 1995, pp. 279-286.

[PG 88] Pickett R. M., Grinstein G. G.: *‘Iconographic Displays for Visualizing Multidimensional Data’,* Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ, 1988, pp. 514-519.

[Shn 92] Shneiderman B.: ‘*Tree Visualization with Treemaps: A 2-D Space-filling Approach*’, ACM Trans. on Graphics, Vol. 11, No. 1, 1992, pp. 92-99.

[Ward 94] Ward M. O.: *‘XmdvTool* M. G.: *Integrating Multiple Methods for Visualizing Multivariate Data’,* Proc. Visualization ’94, Washington, DC, 1994, pp. 326-336.

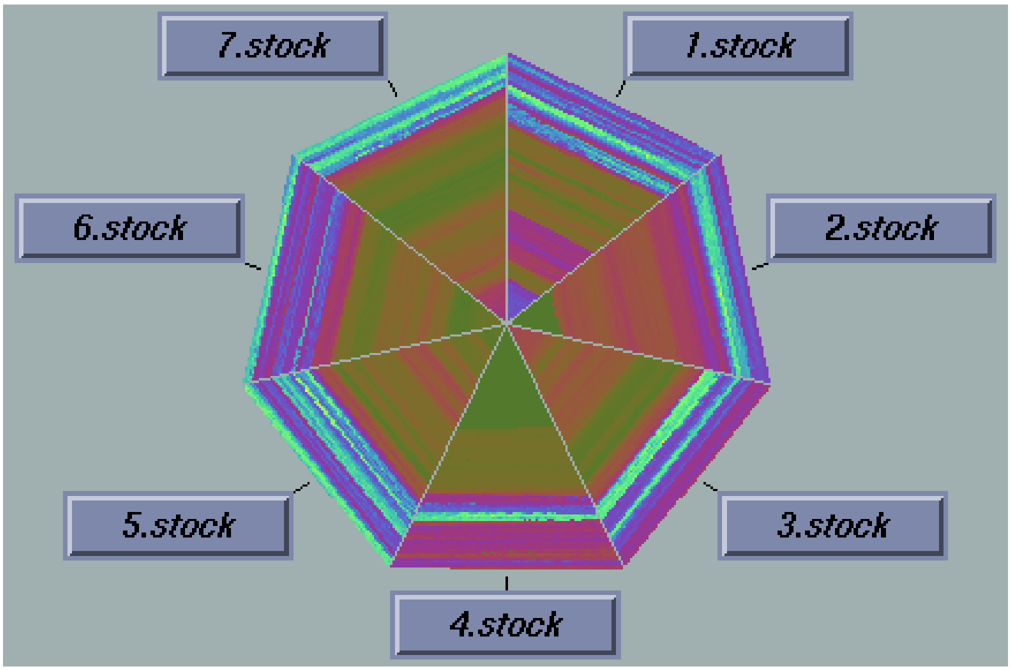

Figure 3: Visualizing 7-dimensional Data with the ‘Circle Segments’ Technique

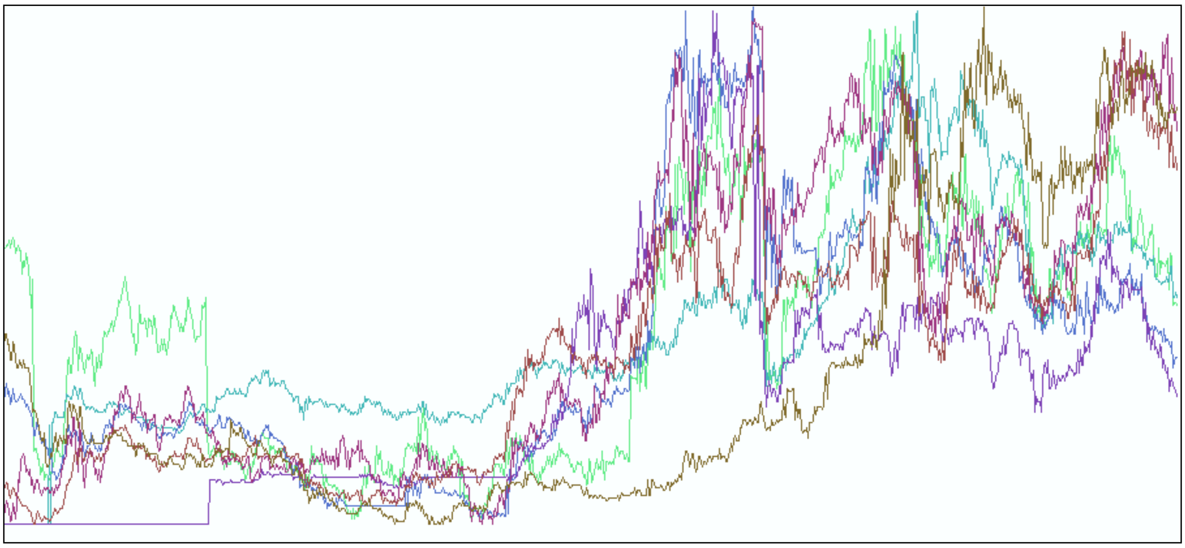

Figure 4: Visualizing 7-dimensional Data using the ‘Line Graph’ Visualization Technique

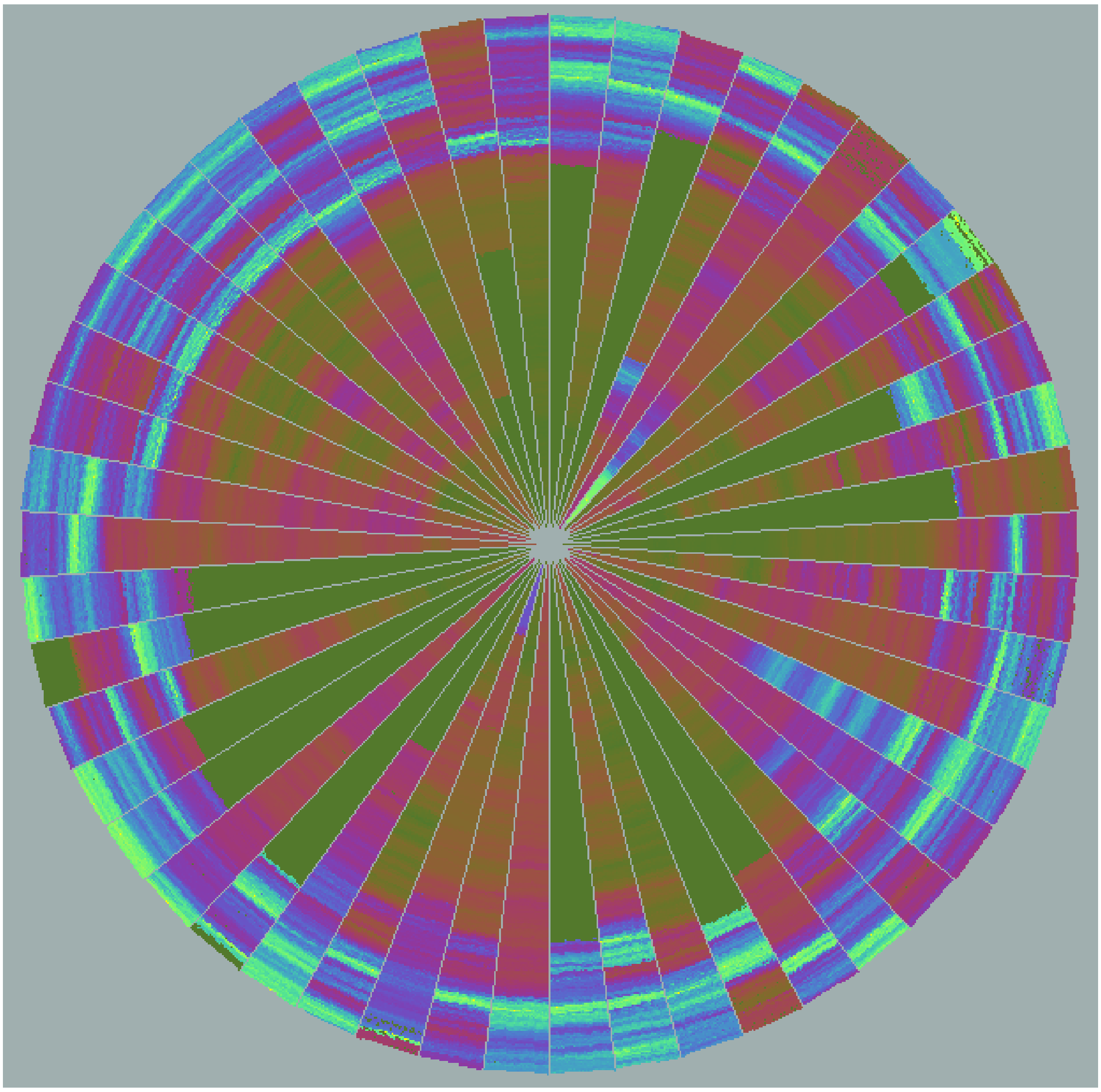

Figure 5: Representing about 265,000 50-dimensional Data Items with the ‘Circle Segments’ Technique