SIGMOD RECORD special issue on 'Information Visualization', Dec. 1996.

Pixel-oriented Database Visualizations

Daniel A. Keim

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 Munich, Germany
Phone (+49) 89 2178-2225, Fax (+49) 89 2178-2192
E-mail: keim@informatik.uni-muenchen.de

Abstract

In this paper, we provide an overview of several pixel-oriented visualization techniques which have been developed over the last years to support an effective querying and exploration of large databases. Pixel-oriented techniques use each pixel of the display to visualize one data value and therefore allow the visualization of the largest amount of data possible. The techniques may be divided into query-independent techniques which directly visualize the data (or a certain portion of it) and query-dependent techniques which visualize the relevance of the data with respect to a specific query. An example for the class of query-independent techniques is the recursive pattern technique which is based on a generic recursive scheme generalizing a wide range of pixel-oriented arrangements for visualizing large databases. Examples for the class of query-dependent techniques are the generalized spiral and circle-segments techniques, which visualize the distances with respect to a database query and arrange the most relevant data items in the center of the display.

Keywords: Visualizing Large Databases, Visual Data Mining, Visualizing Multidimensional and Multivariate Data

1. Introduction

Visualization of data which have some inherent two- or three-dimensional semantics has been done even before computers were used to create visualizations (see for example the well-known books of Tufte [Tuf 83, Tuf 90]). 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. For most of the data stored in databases, however, there is no standard mapping into the Cartesian coordinate system, since the data has no inherent two- or three-dimensional semantics. In general, relational databases can be seen as multidimensional data sets with the attributes of the database corresponding to the dimensions of the multidimensional data set. There are a number of well-known techniques for visualizing multidimensional data sets. Those techniques may be classified into geometric projection techniques, iconic display techniques, hierarchical techniques, graph-based techniques, pixel-oriented techniques, dynamic techniques, and combinations hereof. An overview of many of those techniques is presented in [Kei 96]. The research also resulted in data exploration and analysis systems which implement some of the mentioned techniques. Examples include statistical data analysis packages such as S Plus / Trellis [BCW 88], XGobi [SCB 92], and Data Desk [Vel 92], visualization oriented systems such as ExVis [GPW 89], XmdvTool [Ward 94], and IBM's Parallel Visual Explorer, as well as database oriented systems such as TreeViz [Shn 92], the Information Visualization and Exploration Environment (IVEE) [AW 95], the VisDB system [KK 95] and SGI's MineSet™ [SGI 96].

In this article, we provide an overview of the pixel-oriented visualization 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 (cf. Figure 1). Since in general our techniques use only one pixel per data value, the techniques allow 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, the main question is 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.

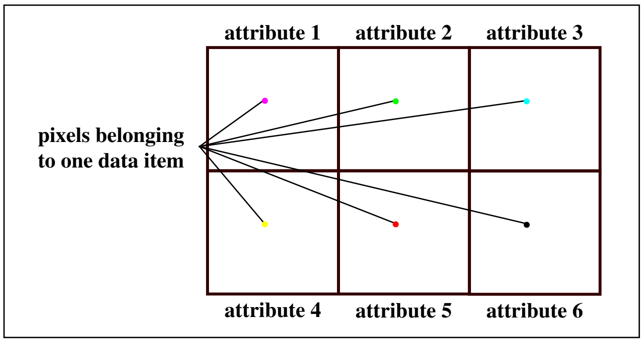

Figure 1: Arrangement of Windows for Data with Six Attributes

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 to the outside of the window.

In the rest of this paper, we briefly introduce three of our pixel-oriented visual database exploration techniques. Section 2 describes a query-independent techniques and section 3 two query-dependent techniques. In section 4, we briefly discuss the implementation of our techniques in the *VisDB* system and in section 5, we summarize our approach and point out some of the open problems for future work.

2. Query-Independent Visualization Techniques

As already mentioned, the basic idea of our visualization techniques is to present as many data values as possible at the same time with the number of data values being only limited by the number of pixels of the display. In dealing with arbitrary multidimensional data without any 2D- or 3D-semantics, one major problem is to find meaningful arrangements of the pixels on the screen. Even if the data has a natural ordering according to one attribute (e.g., time series data), there are many possibilities for arranging the data. One straightforward possibility is to arrange the data items from left to right in a line-by-line fashion. Another possibility is to arrange the data items top-down in a column-by-column fashion. If these arrangements are done pixelwise, in general, the resulting visualizations do not provide useful results [KKA 95]. More useful are techniques which provide a better clustering of closely related data items such as space-filling curves (e.g., the well-known curves by Peano & Hilbert [Pea 90, Hil 91] or Morton [Mor 66]). For data mining even more important are techniques which provide nice clustering properties as well as an arrangement which is semantically meaningful. An example for a technique which provides these properties is the recursive pattern technique. The recursive pattern technique is based on a generic recursive scheme which allows the user to influence the arrangement of data items. The basic arrangement is based on a simple back and forth movement: 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-1)-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. On recursion level i ($i \geq 1$), the algorithm draws w_i level(i-1)-patterns h_i times alternately to the right and to the left. The pattern on recursion level i consists of $w_i \times h_i$ level(i-1)-patterns, and the maximum number of pixels that can be presented on recursion level k is given by $\prod_{i=1}^{k} w_i \times h_i$. An example for a recursive pattern visualization of a database containing the 100 stocks of the FAZ index (Frankfurt Stock Index) from 20 years of stock price data (altogether 532,900 data values) is presented in Figure 4. For the details on the recursive pattern technique the reader is referred to [KKA 95].

Note that for the query-independent techniques, it is not mandatory that the data has some natural ordering. In searching for dependencies among attributes, one might sort the data according to one attribute and use our visualization technique for examining the dependencies of the other attributes. Consider, for example, a large database of personal data. If one wants to find dependencies between the parameter sales (of a person) and other attributes such as salary, age, and travel expenses, one might sort the data according to the sales parameter and visually examine the dependencies of the other attributes.

3. Query-Dependent Visualization Techniques

The query-independent visualization techniques visualize the attribute values by directly mapping them to color. The idea of the query-dependent visualization techniques is to visualize the data in the context of a specific user query to give the users feedback on their queries and direct their search. Instead of directly mapping attribute values to colors, the distances of attribute values to the query are mapped to colors. Since the focus of the query-dependent techniques is on the relevance of the data items with respect to the query, different arrangements of the pixels are appropriate. In developing query-dependent techniques, we experimented with several arrangements such as the left-right or top-down arrangements and compared the resulting visualizations. We found, that for visualizing the result for a database query it is most natural to present the data items with highest relevance to the query (e.g., data items fulfilling the query) in the center of the display. Our first approach described in [Kei 94, KK 94] arranges the data items with lower relevances in a rectangular spiral shape around the center. The techniques presented in this paper – the generalized-spiral and the circle-segments techniques – are generalizations of those techniques.

In case of the generalized-spiral technique, the original spiral arrangement [KK 94] is extended to a generic snake- or hilbert-like form, of which the user may choose the height (cf. Figure 2). As in case of the query-independent visualization techniques, a separate visualization for each of the selection predicates (attributes) is generated (cf. Figure 1). An additional subwindow shows the overall distances. In all subwindows, the pixels for each data item are placed at the same position as the overall distance for the data item in the overall

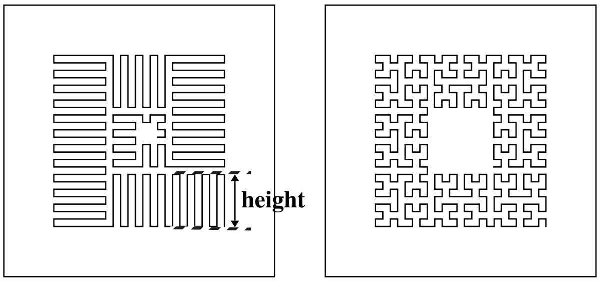

Figure 2: Generalized-Spiral Technique

distance subwindow. By relating corresponding regions in the different windows, the user is able to perceive data characteristics such as multidimensional clusters or correlations. Additionally, the separate subwindows for each of the selection predicates provide important feedback to the user, e.g. on the restrictiveness of each of the selection predicates and on single exceptional data items. Note that the original spiral technique is now the special case of the generalized-spiral technique with a height of one pixel. The advantage of the generalized-spiral technique is that the degree of clustering is higher. For details on the different variants of the generalized-spiral technique and a first examination of their effectiveness the reader is referred to [Kei 95].

A second query-dependent technique is the circle-segments technique. The basic idea of the circle-segments visualization technique is to display the distances for the attributes as segments of a circle (cf. Figure 3). If the data consists of k attributes, the circle is partitioned into k segments, each representing one the distances for one attribute. Inside the segments, the distance values belonging to one attribute 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. An example for a circle-segments visualization of 50 stock price developments from the FAZ index database used in the example above is presented in Figure 5. For the details on the circle-segments technique the reader is referred to [AKK 96].

4. The VisDB System

Several pixel-oriented visualization techniques including the techniques described in the previous subsections are implemented as part of the *VisDB* system [KK 95]. In addition to our pixel-oriented techniques, the *VisDB* system also supports the parallel coordinates technique developed by Inselberg & Dimsdale [Ins 81, ID 90] and the stick figure technique developed by Picket & Grinstein from the University of Massachusetts, Lowell. The *VisDB* system is implemented in C++/MOTIF and runs under X-Windows on HP 7xx machines. The system consists of an interactive interface which is divided into the visualization portion and the query specification portion (for the query-dependent techniques). The query specification portion provides a slider-based direct-interaction interface which allows an intuitive specification of queries [Kei 94, KKS 94]. Different types of sliders are available for different data types. Other options which support the data exploration process are the possibility to

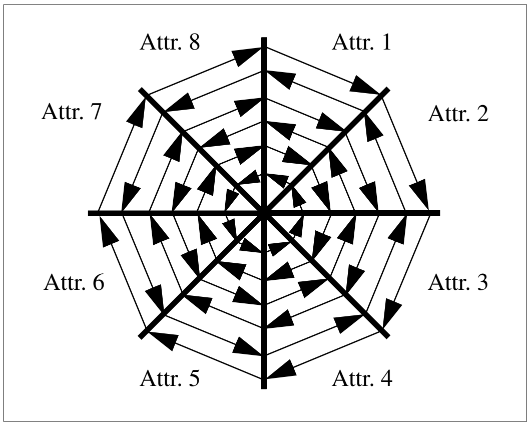

Figure 3: The Circle-Segments Technique for 8-dimensional Data

focus on certain colors and the possibility to get the data values corresponding to a pixel of the display. The current version of the *VisDB* system is main memory based and allows interactive query-dependent visualizations of very large databases. For databases with less than 100,000 data values, the recalculations can be considered to be truly interactive; for larger databases, the time is still in the range of a few seconds (for 1,000,000 data values, for example, the response time is about 20 seconds). When interfacing with current commercial database systems, however, performance problems arise since no access to partial results of a query is available, no support for incrementally changing queries is provided, and no multidimensional data structures are used for fast secondary storage access. We are currently working on improving the performance in directly interfacing to a database system. In the future, we plan to implement the *VisDB* system on a parallel machine which will be able to support interactive query modifications even for larger amounts of data.

The *VisDB* system has been successfully used in several application areas including a financial application where the system has been used to analyze multidimensional time-dependent data, a CAD database project where the system has been used to improve the similarity search, as well as a molecular biology project where the system has been used to find possible docking regions by identifying sets of surface points with distinct characteristics [Kei 94]. Currently, we explore several other data sets including a large database of geographical data, a large environmental database, and a NASA earth observation database. A first evaluation and comparison of multiple visualization techniques using different data sets and queries is reported in [KK 96].

5. Conclusions

Pixel-oriented visualization techniques which use each pixel of the display to visualize one data value provide a valuable help in exploring very large databases. The techniques described in this paper allow users to get a visual overview of large data sets and supports them in finding correlations, functional dependencies, and clusters. Our query-independent techniques directly visualize the attribute values by arranging the values according to some screen-filing curve. In addition, the recursive pattern technique allows the user to control the arrangement of the data values, providing the possibility to generate more meaningful visualizations. Our query-dependent techniques visualize the data attributes in the context of a specific query and provide visual feedback in querying the database. The techniques are especially helpful for interactively exploring large databases. We believe that the different techniques are useful for different data exploration tasks and for different stages of the data exploration process.

At this point, we want to stress that our visualization techniques are not designed to replace or substitute methods which have been developed in statistics or knowledge discovery. Also, we do not claim that our techniques are in general better than those techniques. Data visualization, statistics, and knowledge discovery have their advantages and we view them as being complementary to each other. Statistical analysis may, for example, be used to validate the hypotheses generated by the visualizations and vice versa. Future integrated tools for exploratory data analysis should therefore include not only statistical methods and knowledge discovery techniques but also data visualization techniques such as our pixel-oriented visualizations.

Acknowledgments

Developing a large number of diverse visualization techniques and implementing a complex system such as the *VisDB* system can not be done by a single person. My thanks goes to all my colleagues and students who contributed to the *VisDB* system, especially Thomas Seidl who implemented the first prototype of the system, Juraj Porada who implemented most of the current version, Mihael Ankerst who developed and implemented the recursive pattern and circle-segments techniques, and Professor Dr. Kriegel who provided the inspiring environment for doing the research reported in this paper.

References

[AKK 96] Ankerst M., Keim D. A., Kriegel H.-P.: *'Circle-Segments: A Technique for Visually Exploring Large Multidimensional Data Sets',* Visualization '96, Hot Topic Session, San Francisco CA, 1996.

[AW 95] Ahlberg C., Wistrand E.: *'IVEE: An Information Visualization and Exploration Environment',* Proc. Int. Symposium on Information Visualization, Atlanta, GA, 1995, pp. 66-73.

[BCW 88] Becker R., Chambers J. M., Wilks A. R.: *'The New S Language'*, Wadsworth & Brooks/Cole Advanced Books and Software, Pacific Grove, CA., 1988.

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

[Hil 91] Hilbert D.: *'Über stetige Abbildung einer Line auf ein Flächenstück'*, Math. Annalen, Vol. 38, 1891, pp. 459-460.

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

[Ins 81] Inselberg A.: *'N-Dimensional Graphics Part I: Lines & Hyperplanes',* IBM LA Science Center Report, # G320-2711, 1981.

[Kei 94] Keim D. A.: *'Visual Support for Query Specification and Data Mining',* Ph.D. Dissertation, University of Munich, July 1994, Shaker-Publishing Company, Aachen, Germany, 1995, ISBN 3-8265-0594-8.

[Kei 95] Keim D. A.: *'Enhancing the Visual Clustering of Query-dependent Databases Visualization Techniques using Screen-Filling Curves',* Proc. Int. Workshop on Database Issues in Visualization, Atlanta, GA, 1995.

[Kei 96] Keim D. A.: *'Databases and Visualization',* Tutorial, ACM SIGMOD Int. Conf. on Management of Data, Montreal, Canada, 1996, p. 543. A postscript version of the tutorial notes is available under "http://www.informatik.uni-muenchen.de/~keim".

[KK 94] Keim D. A., Kriegel H.-P.: *'VisDB: Database Exploration using Multidimensional Visualization',* Computer Graphics & Applications, 1994, pp. 40-49.

[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.

[KK 96] Keim D. A., Kriegel H.-P.: *'Visualization Techniques for Mining Large Databases: A Comparison',* Trans. on Knowledge and Data Engineering, Dec. 1996, to appear.

[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.

[KKS 94] Keim D. A., Kriegel H.-P., Seidl t.: *'Supporting Data Mining of Large Databases by Visual Feedback Queries',* Proc. 10th Int. Conf. on Data Engineering, Houston, TX, 1994, pp. 302-313.

[Mor 66] Morton .: *'A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing'*, IBM Ltd. Ottawa, Canada, 1966.

[Pea 90] Peano G.: *'Sur une courbe qui remplit toute une aire plaine'*, Math. Annalen, Vol. 36, 1890, pp. 157-160.

[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.

[SCB 92] Swayne D.F., Cook D., Buja A.: *'User's Manual for XGobi, a Dynamic Graphics Program for Data Analysis',* Bellcore Technical Memorandum, 1992.

[SGI 96] Database Mining and Visualization Group: *'MineSet^{TM}: A System for High-End Data Mining and Visualization'*, Silicon Graphics Inc., VLDB 1996, p. 595.

[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.

[Tuf 83] Tufte E. R.: *'The Visual Display of Quantitative Information',* Graphics Press, Cheshire, CT, 1983.

[Tuf 90] Tufte E. R.: *'Envisioning Information',* Graphics Press, Cheshire, CT, 1990.

[Vel 92] Velleman P. F: *'Data Desk 4.2: Data Description'*, Ithaca, NY, 1992.

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

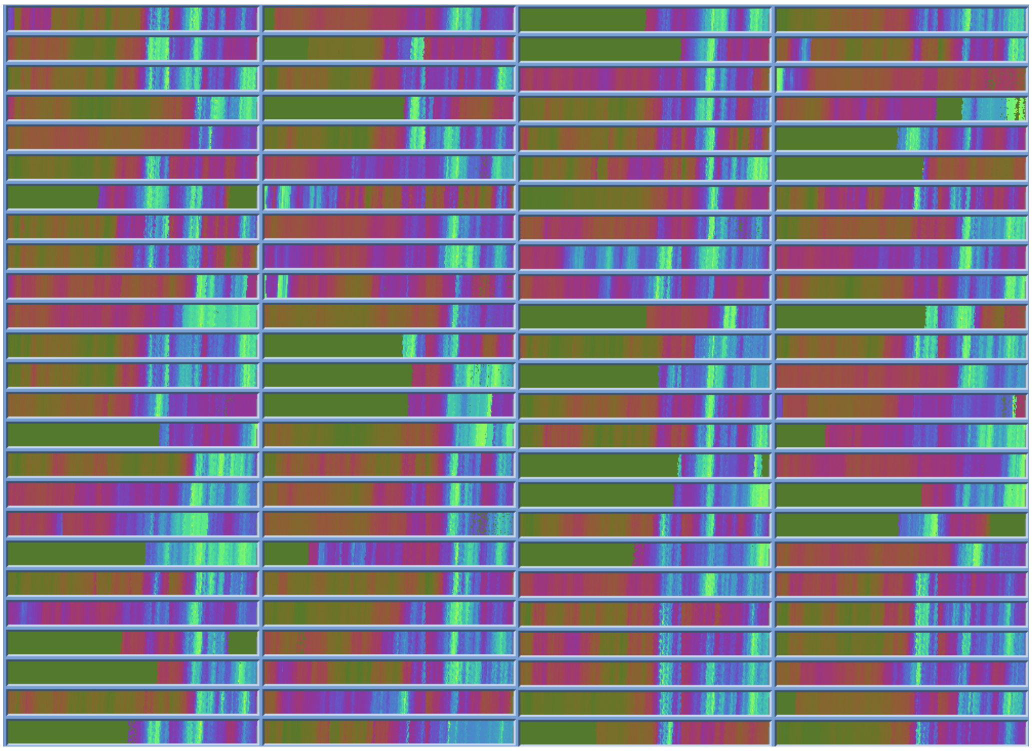

Figure 4: Recursive Pattern Visualization of 100 Stocks from Jan. '74 to Apr. '95 (about 530,000 Data Values) (cf. [KKA 95])

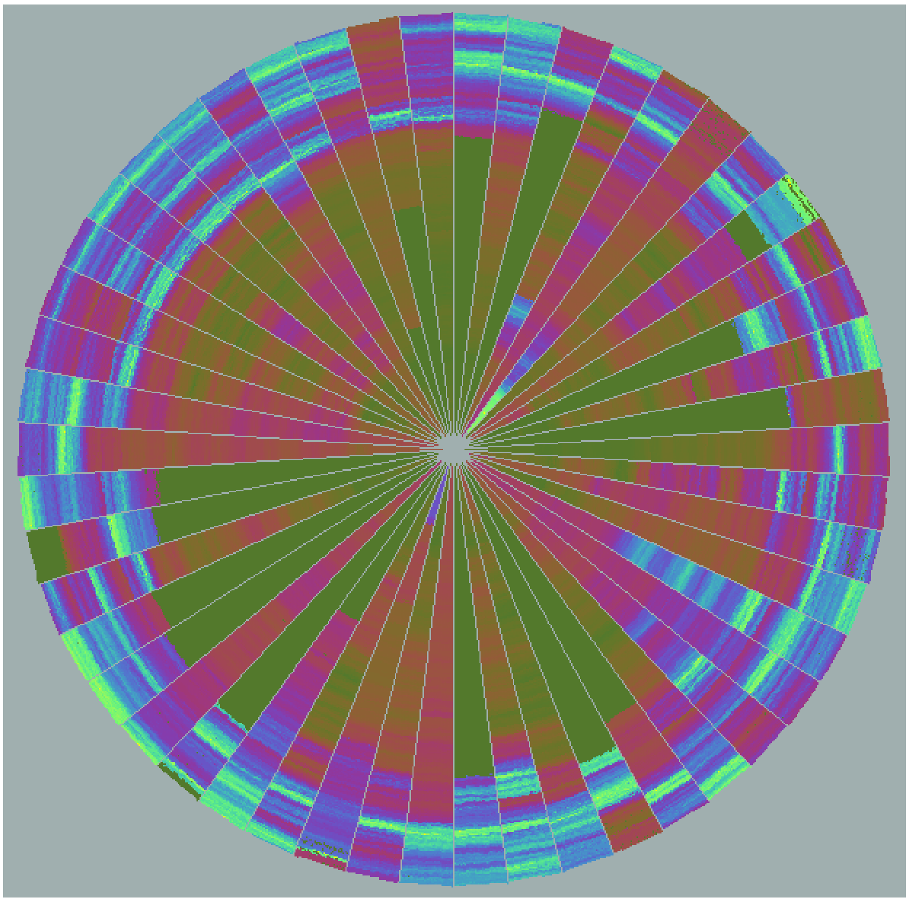

Figure 5: Circle-Segments Visualization of 50 Stocks from Jan. '74 to Apr. '95 (about 265,000 Data Values) (cf. [AKK 96])