Multidimensional Data Visual Exploration by Interactive Information Segments

Francisco J. Ferrer-Troyano, Jesús S. Aguilar-Ruiz, and José C. Riquelme

Department of Computer Science, University of Seville
Avenida Reina Mercedes s/n, 41012 Seville, Spain
{ferrer,aguilar,riquelme}@lsi.us.es

Abstract. Visualization techniques provide an outstanding role in KDD process for data analysis and mining. However, one image does not always convey successfully the inherent information from high dimensionality, very large databases. In this paper we introduce VSIS (Visual Set of Information Segments), an interactive tool to visually explore multidimensional, very large, numerical data. Within the supervised learning, our proposal approaches the problem of classification by searching of meaningful intervals belonging to the most relevant attributes. These intervals are displayed as multi–colored bars in which the degree of impurity with respect to the class membership can be easily perceived. Such bars can be re–explored interactively with new values of user–defined parameters. A case study of applying VSIS to some UCI repository data sets shows the usefulness of our tool in supporting the exploration of multidimensional and very large data.

1 Introduction

Visualization techniques provide an important support to extract knowledge from huge amounts of data incorporating ingenuity, analytic capability, and experience of the user in order to steer the KDD process [15]. From graphic representations of a query or data set, the user carries out an interactive visual exploration from which interesting subsets and data relationships can be identified, and new hypotheses and conclusions can be drawn. Such hypotheses can be later verified by data mining techniques. Through a visual exploration, the user can intuitively have a good idea of the result interpretation. Different graphic views of the same data set can give the user a better understanding about it, and an easy way to detect patterns, outliers, and noise. In addition, visualization tools can be also used to reduce the search space and therefore, to obtain simpler models for complex sub–domains.

An important concern for multidimensional data visualization techniques is to avoid different entities overlapping on the screen. These individual entities can be data–items or examples, data–values or attribute–values, or data aggregations based on the former ones. If the values are directly displayed, they usually are a significantly small portion of the entire available data. Otherwise, it is likely that the resulting image cannot convey the data properties appropriately and the

Y. Kambayashi et al. (Eds.): DaWaK 2004, LNCS 3181, pp. 239–248, 2004.

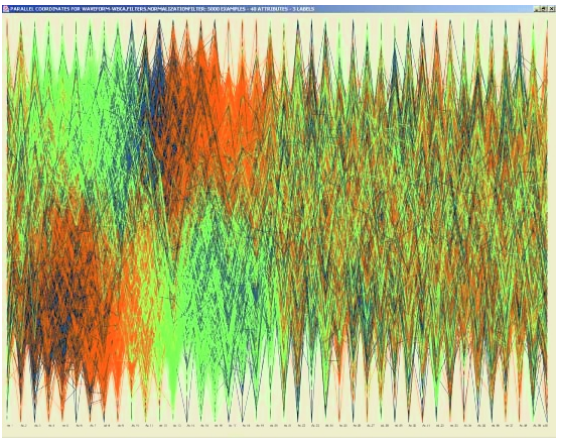

Fig. 1. Wave–Form database (40 attributes, 5000 examples, 3 class labels) in Parallel Coordinates.

exploration becomes a difficult task. As an example, Figure 1 shows the Wave–form data set displayed using the well–known Parallel Coordinates technique [11]. Because of the high width and depth of this data set, individual examples cannot be clearly seen from this display, also preventing the detection of relevant patterns and attributes. We think it is more interesting to display as few graphic entities as possible in order to represent as large amount of data as possible. The smaller number of graphical entities containing higher number of examples, the easier and more meaningful interpretation of results.

In addition, many visualization techniques have restrictions regarding the data size, with respect either to the number of examples or the number of attributes. In this paper we introduce VSIS (Visual Set of Information Segments), an interactive tool to explore multidimensional and very large databases. Handling enormous amount of data might seem risky to graphically represent every different value in only one image, not only because of the screen limitations, but also the human ability to understand a complex image. Therefore, our goal is also to incorporate user's constraints, so the display can become more significant for the expert.

2 VSIS: Visual Set of Information Segments

Within the supervised learning, the problem of classification is generally defined as follows. An input finite data set of training examples is given. Every training example is a pair $e = (x, y)$ where x is a vector of m attribute values (each of which may be numeric or symbolic) and y is a class discrete value named label. The goal is to obtain a model $y = f(x)$ to classify or decide the label for new non–labelled test examples named queries. VSIS supports the problem of classification with numerical attributes by displaying only the most relevant

attributes with only the most meaningful intervals. The set of intervals is as small as possible depending on the user demand. For each interval is displayed the distribution of labels within it and the relationship with other intervals. We name these graphic entities information segments.

Henceforth, the next notation is used to describe VSIS. Let m be the number of continuous attributes $(\mathcal{A}_1, \ldots, \mathcal{A}_m)$. Let $Y = \{y_1, \ldots, y_z\}$ be the set of class labels from one nominal attribute previously selected by the user. Let $\mathcal{T}$ be the training set so that: $\mathcal{T} = \{e_1, \ldots, e_n\};\ e_i = (x_i, y_i);\ x_i \in \mathcal{R}^m;\ y_i \in Y; i \in \{1, \ldots, n\};\ m, n \in \mathcal{N}$.

Definition 1 (Empty Segment) *An empty segment $S_{j,k}$ represents an interval I of the j^{th} attribute $\mathcal{A}_j$ for which no training example has a value within I: $\forall\ e_i \in \mathcal{T} \cdot x_{ij} \notin I$.*

Definition 2 (Pure Segment) *A pure segment $S_{j,k}$ represents an interval I of the j^{th} attribute $\mathcal{A}_j$ for which all the training examples are associated with the same class label: $\nexists\ e_i,\ e_{i'} \in \mathcal{T} \cdot x_{ij} \in I \wedge x_{i'j} \in I \wedge y_i \neq y_{i'}$.*

Definition 3 (Impure Segment) *An impure segment $S_{j,k}$ represents an interval I of the j^{th} attribute $\mathcal{A}_j$ for which there are training examples associated with different class labels: $\exists\ e_i,\ e_{i'} \in \mathcal{T} \cdot x_{ij} \in I \wedge x_{i'j} \in I \wedge y_i \neq y_{i'}$.*

The segments are displayed as colored bars. The color represents the class label and it is previously selected by the user. Empty segments are not displayed, pure segments are displayed with one color and impure segments are displayed with a number of colors. Every color takes up an area inside a rectangle which is proportional to the number of examples with the label associated to such a color in the respective information segment.

The process is divided into two steps: first, an initial set of segments is calculated; second, the minimal set of segments meaningful for the user is obtained. Both sets are displayed and can be interactively re–explored.

2.1 ISIS: Initial Sets of Information Segments

This first phase builds m initial sets $ISIS_j$ $(j \in \{1, \ldots, m\})$, one per attribute. Each set $ISIS_j$ is formed by α information segments and provide the user with insight about the label distribution of input data. α is a user parameter (integer) which splits the continuous attributes of $\mathcal{T}$ into α equal–width intervals. The higher value for α, the higher accuracy is obtained.

Each information segment $S_{j,k}$ $(k \in \{1, \ldots, \alpha\})$ is composed by three elements:

- $I_{j,k} = [l_{j,k}, u_{j,k})$ is a left–closed, right–open interval in $\mathcal{R}$, such that $u_{j,k} = l_{j,k+1}$ $(\forall k < \alpha)$.
- $H_{j,k} = \{H_{j,k^1}, \ldots, H_{j,k^z}\}$ is a histogram with the number of examples for each label that are covered by $S_{j,k}$. An example e_i is covered by a segment $S_{j,k}$ if the j^{th} attribute value of the example (x_{ij}) belongs to the interval $I_{j,k}$.

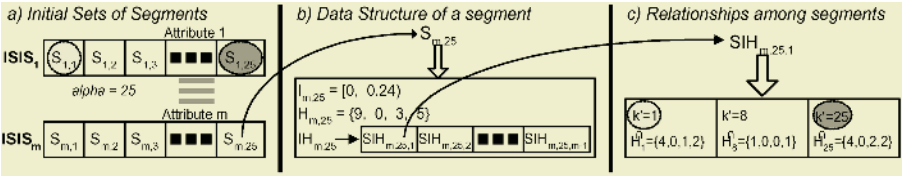

Fig. 2. Diagram of the data structure used to build the initial sets of information segments ($\alpha = 25$).

– $IH_{j,k}$ is a set of $m-1$ elements $SIH_{j,k,j'}$, one per each attribute $\mathcal{A}_{j'}$ different to $\mathcal{A}_j$. Each element $SIH_{j,k,j'}$ is composed by a set of pairs $(k', H^{\cap}_{k'})$, related to segments for other attributes containing examples covered by S_{jk}. The element k' is the index of a segment $S_{j',k'}$, and $H^{\cap}_{k'}$ is the histogram of class labels for examples in the intersection $H_{j,k} \cap H_{j',k'}$. The purpose of this data structure is to compute the minimal set of segments in the next phase.

The initial sets of segments are built by one only scan, previously generating α empty segments $S_{j,k}$ for each attribute with $H_{j,k^p} = 0$ ($p \in \{1, \ldots, z\}$) and $IH_{jk} = \emptyset$. Then every example $e_i = (x_i, y_i)$ updates the class–labels histogram H_{jk} of the segment S_{jk} that covers x_i (increasing by one $H_{jk^{y_i}}$), and the relationships IH_{jk} among such updated segments. The computational cost of the process is not expensive since the index k of the segment S_{jk} associated to a value x_{ij} can be calculated directly:

$$k = \lfloor norm(x_{ij}) \cdot \alpha \rfloor;\ norm(x_{ij}) = \frac{x_{ij} - MIN_j}{MAX_j - MIN_j};\ MIN_j = l_{j1};\ MAX_j = u_{j\alpha}$$

Figure 2 shows a diagram of the data structure used to build the initial set of information segments, using $\alpha = 25$ initial intervals and 4 class labels ($Y = \{A, B, C, D\}$). For each attribute $\mathcal{A}_j$, each $ISIS_j$ has 25 equal–width segments. The last segment of the last attribute ($S_{m,25}$) is associated to the real interval $[0, 0.24]$. This interval covers 17 examples, 9 of them with label A, 3 with label C, and 5 with label D. These 17 examples are covered by three segments in the attribute 1 ($S_{1,1}$,$S_{1,8}$ and $S_{1,25}$). The first segment, $S_{1,1}$, covers 7 examples (4 with class A, 0 with class B, 1 with class C and 2 with class D), 2 the second and 8 the third one.

When all the examples have been processed, all the empty segments are removed. Next, every pair of consecutive segments with equal label distribution is joined ($\frac{|H_{j,k-1^p}|}{\sum_{p=1}^{z}|H_{j,k-1^p}|} = \frac{|H_{j,k^p}|}{\sum_{p=1}^{z}|H_{j,k}|}, \forall p \in \{1, \ldots, z\}$). Given $S_{j,k-1}$ and $S_{j,k}$ as consecutive segments, $H_{j,k-1}$ and $IH_{j,k-1}$ are updated with $H_{j,k}$ and $IH_{j,k}$, respectively, and the right segment $S_{j,k}$ is removed. Finally, when all the attributes have been examined, a ranking of them is obtained as a function of the number of pure segments, the number of impure ones, and the impurity level of them, by means of the next heuristic:

$$Weight(\mathcal{A}_j) = n^{-ns_j} \sum_{k=1}^{ns_j} (\max_{p=1}^{z} |H_{j,k^p}|)$$

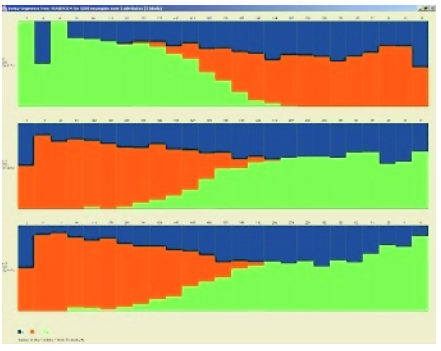

Fig. 3. Wave–Form database. Initial segments for the three most significant attributes (x7, x15, and x16), using $\alpha = 25$. There are not empty segments in any attribute.

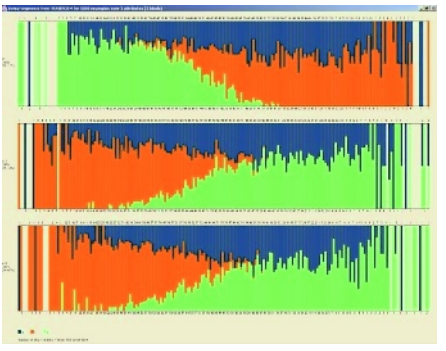

Fig. 4. Wave–Form database. Initial segments using $\alpha = 200$. There are empty segments in every attribute: x7 (32 segments), x15 (25), and x16 (29).

where n is the number of training examples, z is the number of different class labels and ns_j is the number of non–empty initial segments in $\mathcal{A}_j$.

Figures 3 and 4 show the initial segments obtained from Wave-form data set, for the three most significant attributes according to the above heuristic: x7, x15 and x16. Attributes are graphically shown in order of relevance, from top to bottom. The display shows similarities among attributes -when two or more of them have similar shapes- and their relevance -when the distribution is homogeneous, that is, intervals are impure-. The higher value for α, the greater number of empty intervals are displayed. That is the reason why in Figure 4 the number of initial segments are not equal to the number of initial intervals. Having a look to the images, we can know the label distribution and the overlap level inside the attributes. It is interesting the correlation between attributes x15 and x16 (two at the bottom of figure on the left). That correlation is stronger when the class *orange* is present, and we can observe that shapes for class *green* are different for these two attributes from the middle of the attribute until the right bound. In addition, the initial segments provide the user with an insight about the potential complexity of the *Minimal Set of Information Segments*.

2.2 MSIS: Minimal Set of Information Segments

In the second phase, consecutive segments belonging to the attributes selected in the first phase are joined, trying to take advantage of attributes with least number of segments and smaller intersection among them. The goal now is to find the least number of segments from which to describe the label distribution, transforming thousands of examples with dozens of attributes into several colored segments clearly separated in the image.

Definition 4 (Support) *The support of a segment $S_{j,k}$ is the number of examples covered by $S_{j,k}$.*

Definition 5 (Purity) *The purity of a segment $S_{j,k}$ is the percentage of examples covered by $S_{j,k}$ with a majority label with respect to its support.*

Definition 6 (Minimal Support γ) *The minimal support γ is the lowest support that a segment must surpass to belong to the MSIS.*

Definition 7 (Minimal Purity δ) *The minimal purity δ is the lowest percentage of examples with a majority label with respect to the number of covered examples that an impure segment must surpass for to it be a part of the MSIS.*

Therefore, a pure segment $S_{j,k}$ takes $\delta_{j,k} = 100$ whereas an impure segment $S_{j',k'}$ takes $\delta_{j',k'} = 100\frac{\max_{p=1}^{z}(|H_{j',k'^p}|)}{\sum_{p=1}^{z}|H_{j',k'^p}|}$. γ and δ are two user parameters.

The MSIS is built from the $ISIS_j$ sets by two iterative procedures. For each attribute $\mathcal{A}_j$, the algorithm looks for the two consecutive impure segments in $ISIS_j$ whose union is possible and whose resulting support is the highest. Two consecutive impure segments can be joined if the resulting purity is greater than or equal to the minimal purity δ.

Next, another iterative procedure adds joined segments from the $ISIS_j$ sets to MSIS. In each iteration, a new segment is included in the MSIS: the one with the largest number of examples that are not yet covered by another segments already included in the MSIS. Thus, the first segment to be included will be the one with the highest support. The procedure ends when either all the examples have been covered or there is no segment that covers examples uncovered by the MSIS. The number Δ of examples that a segment $S_{j,k}$ in $ISIS_j$ can provide for the MSIS is computed by the intersection among $IH_{j,k}$ and the histograms $H^{\cap}_{k'}$ associated with the segments $S_{j',k'}$ already included in the MSIS, according to the equation 1:

$$\Delta = (\sum_{p=1}^{z}|H_{j,k^p}|) - |\bigcap_{\forall\ S_{j'k'}\in\ MSIS}(SIH_{j,k,j'}, H_{j',k'})\ | \qquad (1)$$

In each new iteration, the number of examples uncovered by non–included segments may change with respect to the earlier iteration, and every segment is re–visited again. If a segment $S_{j,k} \in ISIS_j$ does not contain examples uncovered by MSIS, then it is removed from $ISIS_j$ so that the next iteration will have lower computational cost.

When the above procedure ends, the MSIS is displayed. For each attribute with at least one information segment, a horizontal attribute–bar shows its segments in increasing order of values, from left to right. Every attribute–bar is equal in size, both in width and height. To the left of each bar, the name of the associated attribute is displayed, along with the total number of examples covered by the segments belonging to such an attribute, and the total number of exclusive examples that all the segments provide. We allow interactive capability to VSIS tool so that the user can keep on exploring the examples belonging to impure segments (both in ISIS and MSIS) until finding a meaningful visual description, through both Parallel Coordinates technique and new segments over different dimensions with higher purity.

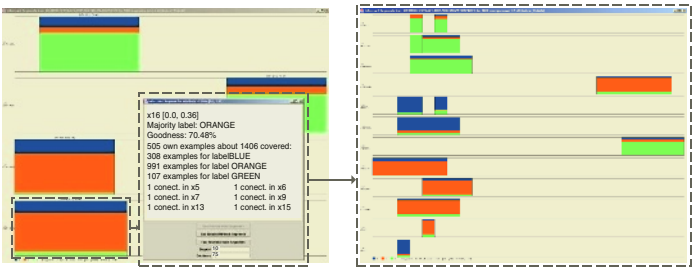

Fig. 5. Wave–Form data set. To the left, first exploration level using α=25, γ=1000 and δ=70. To the right, second exploration level after selecting one segment from the image to the left, using α=25, γ=10 and δ=75.

An example of this capability of VSIS is shown in Figure 5. To the left, we try to get more insight from Figure 3. To select the most significant segments we set γ=1000, so only segments containing at least 1000 examples will be displayed. In addition, as the label distribution provides many impure segments, we will relax the criterion by setting δ=70, so only segments containing at least 70% of examples with the same label must be displayed. Only four attributes were selected by the algorithm (x6, x13, x15 and x16), and one segment for each one. Attribute x7 appeared in Figures 3 and 4, but not in Figure 5, because the new intervals offer fewer intervals with better label distribution.

Now we are interested in the last attribute x16 (bottom), so we can click on it and set new values for δ and γ. This means we are going to analyze only that segment, and therefore, the examples covered by it (exploration level = 2). The new values for δ and γ are 75 and 10, respectively. The result is shown on the right, where more segments are displayed due to the reduction of the value of γ (only 10 examples). However, the purity has been increased up to 75. New attributes appear in this image (x4, x5, x6, x9, x10, x11, x13, x15, x17, x18, x19 and x38), and also new segments (some of them were already in the exploration level =1), which provide more insight about the data, as this might mean that x6 and x13 are decisive for class *green* and x15 for class *orange*.

Re–exploring impure segments gives a powerful insight of data, as we can achieve a higher accuracy on a specific domain. In this way VSIS cedes the control to the user in order to find, group and validate decision rules with the detail level needed. When a segment is explored, the new segments represent sub–domains satisfying new user specifications. Each new exploration level gives a new description of one attribute condition in a decision rule, since when we decide to explore a segment, the subset defined by that attribute condition is visualized. The process is completely interactive, reducing the support and increasing the accuracy every time.

3 Displaying Very Large Databases

We have selected two databases from the UCI repository [5] to show the usefulness of our tool for visualizing great amount of data: one with large number of

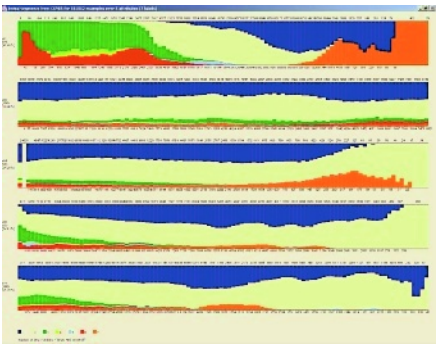

Fig. 6. Covtype database (581012 examples, 54 attributes, and 7 class labels). Initial segments for the best five attributes according to ranking: a01, a12, a14, a36, a37 (α=50).

Fig. 7. Covtype database: MSIS using $\alpha = 50$, $\gamma = 1000$ and $\delta = 70$. Fifteen information segments (5 pure segments and 10 impure segments).

examples (Covtype) and another one with large number of attributes (Isolet). Figures 6 and 7 show two visualization examples of Covtype database. Such displays represent the manner to explore and detect significant sub–domains and irrelevant attributes, and to validate patterns or rules extracted by learning algorithms. These displays also give a good estimate with respect to accuracy and complexity of the model to be extracted by a learning algorithm.

In Isolet database, the high degree of overlapping among different class labels (most of segments are impure with a very low purity) shows the difficulty to obtain both a non–complex and accurate knowledge model for (Figures 8 and 9). To tackle 617 attributes and 26 class labels is computationally expensive for many visualization techniques, however VSIS performance is satisfactory.

4 Related Work

In [15], Information Visualization and Visual Data Mining techniques are classified according to three criteria:

- The data type to be visualized: *one–dimensional data* [19], *two–dimensional data* [20], *multidimensional data* [16], *text & hypertext* [19], *hierarchies & graphs* [6,4], and *algorithms & software* [8].
- The data representation: *standard 2D/3D displays* [20], *geometrically transformed displays* [1,10,11], *icon–based displays* [7], *dense pixel displays* [14], *stacked displays* [12], and hybrid techniques.
- The user interaction way: *projection* [3], *filtering* [20], *zooming* [17], *spherical & hyperbolic distortions*, and *linking & brushing*.

According to the data type, our proposal VSIS can visualize multidimensional data sets with numerical attributes. With respect to the second group, VSIS

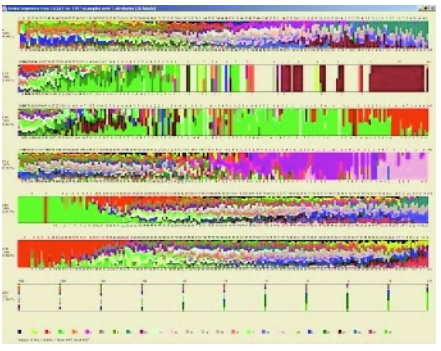

Fig. 8. Isolet database (7797 examples, 617 continuous attributes, and 26 class labels). Initial segments for the best seven attributes using $\alpha = 200$.

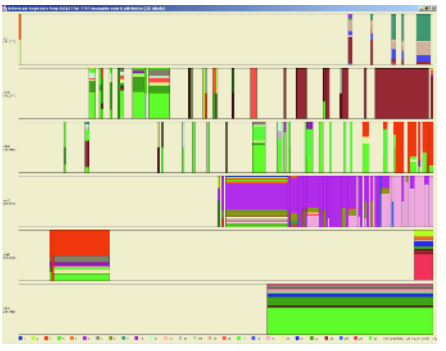

Fig. 9. Isolet database: MSIS using $\alpha = 200$, $\gamma = 10$ and $\delta = 50$.

belongs to standard 2D techniques. Regarding the third category, VSIS provides data projections to the user, zooming and filtering to detect and validate relevant and meaningful attributes and subdomains. Dimensionality reduction has been dealt by three major approaches: Principal Component Analysis (PCA) [13], Multidimensional Scaling (MDS) [18], and Kohonen's Self Organizing Maps (SOM) [9]. Recently, new dimensionality reduction techniques have been proposed to process very large data sets with high dimensionality [2]. VSIS reduces the dimensionality in an interactive manner so as to find meaningful subdomains according to user measures.

5 Conclusions and Future Work

VSIS is a visualization tool to explore multidimensional numerical data. Through visual interaction and feedback, the user decides how many examples and what level of purity a segment must fulfil to be considered representative of a significant subdomain. The information segments can be *seen* as decision rules with a number of disjunctions equals the number of exploration levels, whose support is greater than or equal to the last value of γ, and whose purity is greater than or equal to the last value of δ. This representation helps the user with the identification of the most relevant attributes. Results are very interesting as the tool is very flexible and allows the user to go into the level of exploration needed.

We are currently improving VSIS by using dense pixel approach to display segments with variable color intensity degree in such a way the support of the segments can be easily perceived. Several similarity measures are being addressed to re–order segments on the display regarding the number of shared examples, making the graphical information understandability easier.

References

1. D. F. Andrews. Plots of high–dimensional data. *Biometrics*, 29:125–136, 1972.
2. M. Ankerst, S. Berchtold, and D.A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *InfoVis'98*, pages 52–60, 1998.
3. D. Asimov. Grand tour: A tool for viewing multidimensional data. *SIAM J. Science and Statistical Computing*, 6:128–143, 1985.
4. G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing.* Prentice Hall, 1999.
5. C. Blake and E. K. Merz. Uci repository of machine learning databases, 1998.
6. C. Chen. *Information Visualization and Virtual Environments.* Springer-Verlag, 1999.
7. H. Chernoff. The use of faces to represent points in k–dimensional space graphically. *Journal of Am. Statistical Assoc.*, 68:361–368, 1973.
8. M. C. Chuah and S. G. Eick. Managing software with new visual representations. In *IEEE Information Visualization*, pages 30–37, 1995.
9. Arthur Flexer. On the use of self-organizing maps for clustering and visualization. In *PKDD'99*, pages 80–88, 1999.
10. P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–474, 1985.
11. A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Visualization'90.*
12. B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information. In *Visualization'91*, pages 284–291, 1991.
13. J. Jolliffe. *Principal Component Analysis.* Springer Verlag, 1986.
14. D.A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans. Visualization and Computer Graphics*, 6(1):59–78, 2000.
15. D.A. Keim. Information visualization and visual data mining. *IEEE Trans. Visualization and Computer Graphics*, 8(1):1–8, 2002.
16. D.A. Keim and M.C. Hao. Hierarchical pixel bar charts. *IEEE Trans. Visualization and Computer Graphics*, 8(3):255–269, 2002.
17. N. Lopez, M. Kreuseler, and H. Schumann. A scalable framework for information visualization. *IEEE Trans. Visualization and Computer Graphics*, 8(1):39–51, 2002.
18. A. Mead. Review of the development of multidimensional scaling methods. *The Statistician*, 33:27–35, 1992.
19. L. Nowell, S. Havre, B. Hetzler, and P. Whitney. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Visualization and Computer Graphics*, 8(1):9–20, 2002.
20. D. Tang, C. Stolte, and P. Hanrahan. Polaris: A system for query, analysis and visualization of multidimensional relational databases. *IEEE Trans. Visualization and Computer Graphics*, 8(1):52–65, 2002.