Viz3D: Effective Exploratory Visualization of Large Multidimensional Data Sets

Almir Olivette Artero Maria Cristina Ferreira de Oliveira

Instituto de Ciências Matemáticas e Computação
Universidade de São Paulo
Caixa Postal 668, 13.560-970, São Carlos – SP, Brasil
{almir,cristina}@icmc.usp.br

Abstract

We propose a multidimensional visualization technique, named Viz3D, that creates a 3D representation of n-dimensional data that may be interactively manipulated by users to handle visual cluttering and object occlusion. The projection performed in Viz3D is comparable in quality with the 3D projections obtained with well-known dimensionality reduction techniques, at a lower complexity cost. While a 3D projection conveys more information, giving the user more control of the visual representation and an additional dimension, as compared to 2D, visual cluttering and object occlusion are still a problem in handling large multidimensional data sets. To produce more effective visualizations, two strategies are introduced. Dimensionality is handled with a similarity clustering of attributes prior to projection. Data set size is handled with a new strategy of visualizing data densities, rather than individual data records. Both the direct and density Viz3D visualizations provide the basis for a user driven visual clustering approach applicable to high-dimensional data sets that is very simple, intuitive and effective.

1. Introduction

Advances in data collection technology and digital computing boosted the demand for more effective data analysis tools as data sets increase in number, size and complexity. Many techniques that create interactive visual representations of multidimensional data that can be explored by users in search of information have been proposed [12,3,14,5]. Information Visualization is the research field that encompasses the use of interactive graphical representations to support exploratory analysis of multidimensional data, so that analysis of complex data can benefit from the human ability of detecting patterns on images and domain knowledge. However, scalability of information visualization techniques so that data sets with more than a few thousand elements can be effectively represented is still an issue. Many multidimensional visualization techniques work by mapping each data element (a record, or a record attribute[1]) into a graphical marker and its visual attributes (e.g., shape and color). These graphical markers are then displayed on the two-dimensional screen space. For example, in the Parallel Coordinates technique [10], each record is mapped into a poly-line drawn on a parallel coordinates reference system; whereas in the technique known as *RadViz* [9], each record is mapped onto a small shape positioned on a two-dimensional radial arrangement of axes, as illustrated in Figure 1. It shows a visualization of the Iris flower data, often used to assess classifiers. The data set[2] consists of 150 4-attribute records with measures on three flower species. One of the three classes is linearly separable from the others, the other two are not.

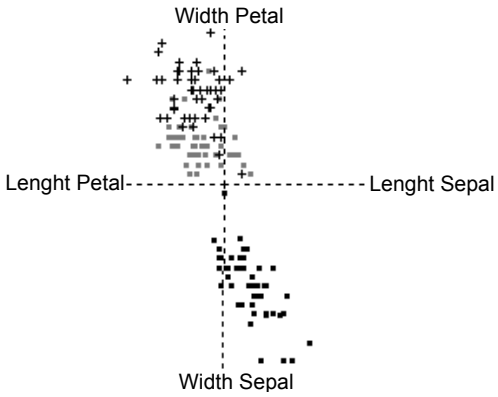


Figure 1 *RadViz* visualization of the *Iris* flower data.

Both Parallel Coordinates and *RadViz* project data, originally defined on an *n*-dimensional (*nD*) space (where *n* is the number of data attributes), on the two-dimensional (2D) screen space. However, mapping each attribute to a graphical marker becomes very ineffective when the number of data attributes and/or of data records increases even moderately. An excessive number of data records produces visualizations hampered by severe object

[1] We use the term data attribute, borrowing terminology from the statistics and data mining fields. Equivalent terms are data dimension (visualization terminology), or a data variate (statistics terminology).

[2] Available at http://lib.stat.cmu.edu/DASL/Datafiles/Fisher'sIris.html

occlusion and visual cluttering [5], which are aggravated if data dimensionality is also high. As a consequence, users may be discouraged from starting exploratory tasks, as visualizations are barely informative.

Creating 2D and 3D graphical representations of multidimensional data precedes the advent of Information Visualization. For example, Principal Component Analysis [15] aims at reducing data dimensionality for analysis and allows data to be displayed in 2D or 3D scatter plots. This and other dimensionality reduction techniques have been proposed, and typically preceed application of many analysis and mining algorithms that can not be effectively applied to high-dimensional data. Dimensionality is also a concern for visualization techniques, as it increases visualization complexity and hampers exploratory analysis.

This work describes contributions for handling the visualization of large high-dimensional data sets. We propose a multidimensional visualization technique that projects *n*-dimensional data into a 3D display space, creating a representation that may be interactively manipulated by users to handle visual cluttering and object occlusion. This technique, called *Viz3D*, extends *RadViz*, inheriting its ability to convey clusters present in data. The projection performed in *Viz3D* is comparable in quality with the 3D projections obtained with well-known dimensionality reduction techniques, at a lower complexity cost. While a 3D projection conveys more information, giving the user more control of the visual representation and an additional dimension, as compared to 2D, the same limitations of the one-record to one-marker mapping approach ultimately hold. To ensure more effective visualizations, two strategies are introduced. To handle dimensionality we apply a similarity clustering of attributes to *Viz3D*. To handle data set size, we propose visualizing data densities, rather than individual data records. These strategies provide the basis for a user driven visual clustering approach applicable to high-dimensional data that is very simple and effective.

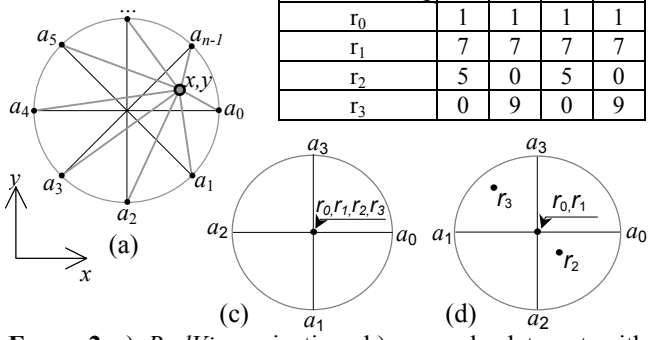
This paper is organized as follows. In Section 2 we provide the required background and discuss related work. In Section 3 we introduce *Viz3D*, presenting the projection approach adopted. The technique is illustrated by comparing its visualizations with those obtained with typical dimensionality reduction techniques, on several data sets. The strategy proposed for deriving an ordering of axes so that cluttering is reduced is introduced in Section 4. In Section 5 we describe a user-driven interactive visual clustering approach based on *Viz3D*, as well as density-based data visualizations that support cluster identification in the *n*-dimensional space. Finally, conclusions and further work are discussed in Section 6.

2. Related Work

We introduce two visualization techniques that inspired the proposal of *Viz3D*. These multidimensional visualizations work by projecting data on the 2D space, depicting data dimensions in a circular arrangement of axes A brief overview of dimensionality reduction techniques is also presented. Earlier attempts to derive an arrangement of data attributes to increase visualization effectiveness, as well as previous work on density-based visualization, are briefly reviewed.

2.1 Visualizing data on 2D space

RadViz (Radial Visualization) [9] displays *n*-dimensional data on a 2D display by mapping data elements on graphical markers projected within a 2D circle. In the visualization, *n* axes emanate from the circle center and terminate on its perimeter, as illustrated in Figure 2(a). Each axis represents a data attribute, and is associated with an attraction factor as in an imaginary spring system. The position of graphical markers is defined by weighting their respective attribute values so as to equilibrate the spring forces. The resulting projection constitutes a non-linear transformation that preserves some data symmetries.



records \ attributes	a_0	a_1	a_2	a_3
r_0	1	1	1	1
r_1	7	7	7	7
r_2	5	0	5	0
r_3	0	9	0	9

Fgure 2 a) *RadViz* projection; b) example data set with four 4-attribute records; c) *RadViz* mapping of different records to the same position, causing occlusion; d) a different disposition of the axes reduces overlapping.

In this technique [5]: in general, records are pushed towards the axis mapping the attribute that has a dominant value; records whose attributes have equal values are projected on the center of the radial system; records that have attributes with equal or almost equal values mapped to axes placed on opposite sides of the circle are also pushed towards the center. Major advantages are the low complexity $O(mn)$ (where m is the number of records and n is the number of attributes), and the fact that similar records in the nD space are projected close together on the 2D space, favoring identification of clusters.

Disadvantages are record overlapping, and the fact that very different records may be projected closed together. Star Coordinates [11] extends this idea allowing the user to control the direction and length of the radial axes in search for adequate data projections. However, interaction is hampered if the number of attributes (i.e., axes) is high, so that the exhaustive search for a good configuration becomes impracticable. Occlusion results from two distinct situations. First, even for moderately sized data sets distinct records may be projected on the same position in the screen. Jittering may alleviate this effect. However, when overlapping is a consequence of the great number of markers to be displayed, other approaches are required to handle the problem.

Dimensionality reduction techniques provide an alternative for producing 2D or 3D views of multidimensional data. Dimension reduction maps data described on an n-dimensional input space to a k-dimensional output space, $k<n$. A widely used technique is Principal Component Analysis (PCA) [15], in which a rotational transformation is applied to the data using the principal components obtained from the eigenvectors. The transformation attempts to maximize the variance of the projected data. Although it produces good results in general, some specific situations are not handled well. Moreover, computational cost is high, $O(m^2)$. FastMap [4] also attempts to maximize variance on the output space. However, it uses some heuristics to avoid the $O(m^2)$. complexity. Rather than computing the principal components, FastMap uses the cosines law to project data over a line segment obtained from the two data points farther apart. The results are good and the cost is reasonable, $O(mn)$.

Yang et al. [18] observe that dimension reduction techniques generate lower dimensional spaces that are not necessarily meaningful to users, and suggest a visual approach towards the problem. In their proposal, the original dimensions are hierarchically organized as a tree, so that similar ones (i.e., correlated attributes) are positioned close together. A user may navigate in the tree and define the most representative attributes. Another alternative to projection are feature selection techniques [13], which try to identify the relevant dimensions for a particular data analysis task.

Visualization techniques can also be improved by embedding information on the relevance of the attributes in the visualization. Ankerst et al. [2] observe that a disposition of visualization axes so that similar attributes are positioned close together is crucial to improve effectiveness of several visualizations, such as Parallel Coordinates, Circle Segments, Recursive Patterns and others. They show that finding an optimal configuration of the axes is an NP-Complete problem, and propose heuristics to handle the problem, describing a solution based on ant colony algorithms. In Section 4 we propose one approach for deriving a good axes disposition for *Viz3D* that produces enhanced data visualizations.

2.2 Handling Data Set Size with Density Information

The concept of data density is defined as follows. Let matrix D_{mxn} represent a set of data containing m records with n attributes each; thus each of its rows corresponds to an n-dimensional variable.

Definition 1 (Density): The density function of a data matrix D_{mxn} for an n-dimensional variable x, based on a kernel density estimation function K, is defined as:

$$f(\boldsymbol{x}) = \frac{1}{mh^n}\sum_{i=1}^{m} K\left(\frac{\boldsymbol{x}-d_i}{h}\right) \quad (1)$$

where d_i is the *i-th* record of the data set (the n-dimensional variable given by the *i-th* row in matrix D) and K is the kernel function defined for the n-dimensional variable x, which satisfies $\int_{R^n} K(\boldsymbol{x})d\boldsymbol{x} = 1$. Square wave and Gaussian functions are the most common kernels. Parameter h in the density function defines a smoothing factor or bandwidth.

Many approaches reported to identify clusters in large data sets are based on computing density estimates. For example, Denclue [7], HD-Eye [8] and HC-Cooperative [1], work with densities calculated over bidimensional projections from the multidimensional data. The multidimensional clusters are defined from the clusters observed in the projected spaces, where the major difficulty is precisely the definition of the most adequate projections. Similarly, visualizations that use data frequency and density estimation to highlight areas with relevant information content in cluttered visualizations have also been proposed. Wegman and Luo [17] suggest density plots to help identifying clusters and uncommon features in Parallel Coordinates visualizations. In their approach, the pixels of the polygonal lines are painted with intensity proportional to the pixel's record overlapping. Their strategy allows identification of some patterns that would be hidden in a conventional visualization. However, overlapping resulting from the crossings of line segments is unduly highlighted, which aggravates cluttering. Rodrigues Jr. et al. [16] use frequency information to highlight information in cluttered Parallel Coordinates visualizations. In their technique, referred to as Frequency Plots, frequency is computed in the 1D space defined by each attribute. In tracing the polygonal lines, pixel intensities along each line segment are interpolated from the frequency values ascribed to the extreme points positioned in the parallel axes. Frequency Plots highlight the behavior of attributes characterizing classes, however, they are less effective to support identification of global patterns. In Section 5 we propose a density-based

visualization that enhances *Viz3D* to handle large volumes of data.

3. *Viz3D*: Multidimensional Visualization in 3D Space

Viz3D is a visualization technique that retains the *RadViz* ability to reveal clusters in multidimensional space, while handling record overlapping more effectively. Data records are projected on the surface and interior of a 3D cylinder, creating a representation that accommodates a larger number of graphical markers, and offers a more suitable alternative for exhibiting large high-dimensional data. Many situations of overlapping may be handled by interacting with the 3D data representation.

Assuming a data set stored on matrix D_{mxn}, a *Viz3D* representation is obtained by mapping the *n*-dimensional coordinates of the *m* data records into 3D coordinates (x_i,y_i,z_i) according to Eq. (2):

$$x_i = x_c + \frac{1}{n}\sum_{j=0}^{n-1}\frac{D_{i,j} - \min_j}{\max_j - \min_j}\cos\left(\frac{2\pi j}{n}\right)$$

$$y_i = y_c + \frac{1}{n}\sum_{j=0}^{n-1}\frac{D_{i,j} - \min_j}{\max_j - \min_j}\sin\left(\frac{2\pi j}{n}\right) \qquad (2)$$

$$z_i = z_c + \frac{1}{n}\sum_{j=0}^{n-1}\frac{D_{i,j} - \min_j}{\max_j - \min_j}$$

for $i = 0,...,m\text{-}1$ and $j = 0,...,n\text{-}1$. (x_c,y_c,z_c) is the origin of the radial axes system; and $max_j = Max(d_{k,j})$, $min_j = Min(d_{k,j})$ for $k = 1,...,m$.

Figure 3 illustrates the projection of an *n*-dimensional record with attributes $(a_0,\ldots,a_{n-1})$. Coordinates x_i and y_i of the graphical marker associated to the ith data record are mapped as in *RadViz*, while the z_i coordinate is obtained by averaging all the attribute values of the given record. Different arrangements of the radial axes produce different visualizations. Therefore, axes positioning may be manipulated to solve overlapping in some situations.

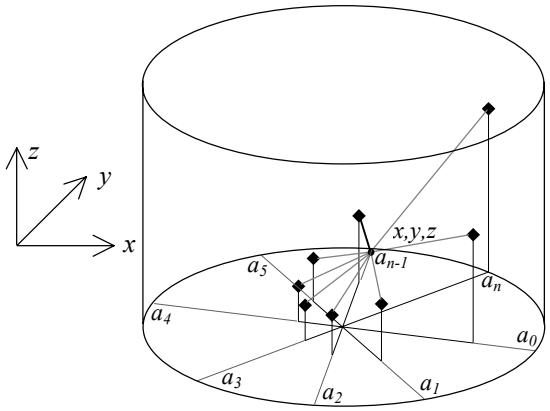


Figure 3 *Viz3D* projection.

Figures 4(a) to 4(d) shows visualizations of a 4-dimensional hypercube generated with *RadViz*, PCA, FastMap and *Viz3D*, respectively. They have been created from 1.281 records representing coordinates of points on the hypercube. The spatial structure lost in *RadViz* (Figure 4(a)) may be observed in all the other visualizations.

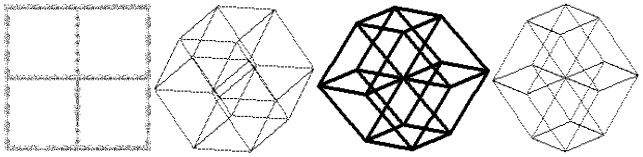


Figure 4 Projection of a 4D hypercube a) on the 2D space, with *RadViz*; on the 3D space with: b) PCA; c) FastMap; d) *Viz3D*.

Figure 5 shows a *Viz3D* visualization of a synthetic data set known as Pollen.data[3], that contains 3.749 records describing 5 measures of pollen grains plus 99 added records that define 6 clusters constituting the letters of the word EUREKA. Figure 5(a) shows the initial *Viz3D* visualization, whereas 5(b) shows the same visualization after a scaling operation, where one may identify the presence of a cluster in the central area. Figure 5(c) shows the result of applying additional scaling and rotation operations, until a configuration has been obtained that allows visually identifying the six clusters.

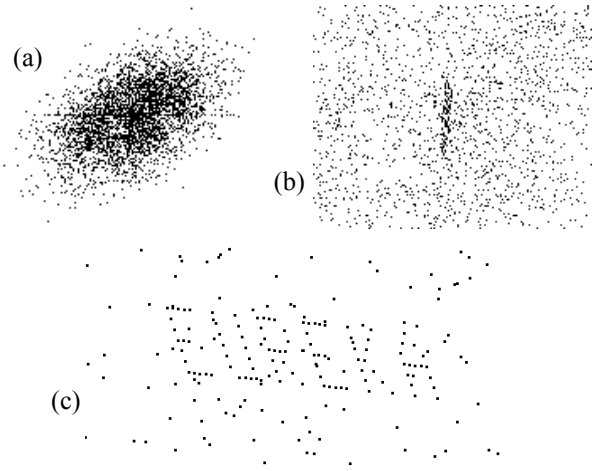


Figure 5 a) Initial *Viz3D* visualization of the *Pollen* data; b) visualization after a scaling operation to amplify the central area; c) visual identification of the clusters.

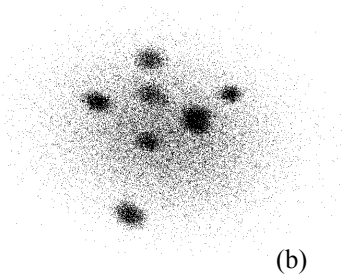
Although the 3D projection performed in *Viz3D* can accommodate more graphical elements, visual cluttering and occlusion are still severe when handling large data sets. This fact is illustrated in Figure 6(b), which shows a *Viz3D* visualization of a synthetic data set called *Sint3.data*

[3] Available at http://lib.stat.cmu.edu/datasets/

containing 38.850 20-atribute records. The data contains 8 clusters, as described in Figure 6(a). A total of 14.831 records belong to clusters, the remaining 24.019 records may be considered as noise. In this case, interacting with the 3D visual representation a user may easily identify the presence of 7 of the 8 clusters. Even though a user might try changing the arrangement of axes in search for more informative projections, an exhaustive search may be tiresome and unproductive. Moreover, if the user is not aware of the data contents, as it is usually the case in exploratory visualization, it is impossible to know if relevant patterns are being missed. This problem is handled with a strategy that generates a 'good' arrangement of the visualization axes, based on an analysis of attribute similarity, as discussed in Section 4.

Cluster#	Records
1	2,272
2	1,521
3	2,272
4	1,518
5	2,271
6	1,352
7	2,272
8	1,353

(a)



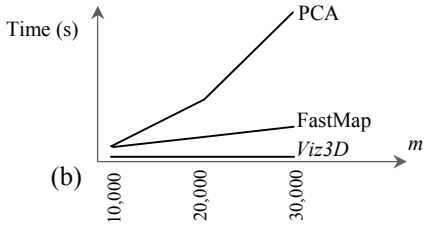
(b)

Figure 6 a) Number of records in each of the 8 clusters in the data set; b) *Viz3D* data visualization.

Figure 7 shows execution times, in seconds, to produce 3D views of data using PCA, FastMap and *Viz3D*, for data sets of size 1.000, 10.000, 20.000 and 30.000 11-attribute records, on a Pentium III with 192 MB RAM. Creating a *Viz3D* projection has complexity $O(mn)$, requires a single scan over the data set, thus ensuring low execution times even on large data.

Data set size	***PCA***	***FastMap***	***Viz3D***
10.000 x 11	10.71	10	0.0599
20.000 x 11	55.59	19	0.1099
30.000 x 11	139.18	29	0.2199

(a)



(b)

Figure 7 a) Execution times, in seconds, for PCA, FastMap and *Viz3D*; b) Line graphs of execution times.

4. Arrangement of Axes and Attribute Reduction

A situation that aggravates occlusion and cluttering both in *RadViz* and *Viz3D* is the displacement of visual markers towards the center of radial system. A possible approach to prevent this effect is to position axes associated to highly correlated attributes close together, avoiding placing them at opposite extremes of the radial system. We follow the suggestion by Ankerst et al. [2] of deriving an ordering of exhibition for the axes based on the similarity of their associated attributes.

An axes arrangement for exhibition that keeps highly similar attributes close together may be achieved computing information on attribute similarity from the data set. Our approach consists in deriving a similarity matrix S_{nxn} (a lower diagonal matrix) from the data matrix D_{mxn}. Marix S contains similarity measures amongst all pairs of data attributes, where $s_{i,j}$ gives a measure of similarity between data attributes i and j, computed, for example, with Eq. (3):

$$s_{i,j} = 1 - \frac{1}{m}\left[\sum_{k=1}^{m}\left|\frac{D_{k,i} - Min(D_i)}{Max(D_i) - Min(D_i)} - \frac{D_{k,j} - Min(D_j)}{Max(D_j) - Min(D_j)}\right|\right] \quad (3)$$

where $Min(Di)$ and $Max(D_i)$ are, respectively, the lowest and highest values in colum i. High values of $s_{i,j}$ indicate high similarity between attributes i and j. This measure is invariant to both scale and translation, but it presents problems in the presence of noisy data, an issue that may be handled by a suitable normalization. It is worth noting that Eq. (3) gives a very simplistic similarity measure, and is being used here to illustrate the proposed axes arrangement approach. Other measures capable of handling functionality between the attributes might be equally considered. Based on Eq. (3) (or an alternative preferred similarity measure), the axes are arranged in a sequence for exhibition based on the relative similarity of their associated attributes, so that highly similar attributes (axes) are placed next to each other.

One may obtain a simple similarity measure associated to the whole attribute set, for a given exhibition sequence *SS*, summing up the similarity values associated to each consecutive pair of axes, for example, $SS(\text{“}ijk\text{”}) = s_{i,j} + s_{j,k}$. In addition to providing a convenient arrangement of the axes for an initial data visualization, the sequence provides useful information to data analysts considering dimensionality reduction processes. A user may interactively remove one of two highly correlated axes from the visualization, i.e., axes corresponding to highly similar attributes, observing the effect on visualization complexity and assessing alternatives prior to applying a feature selection process. An automated process to remove highly correlated attributes from the visualization might be implemented, where potential candidates for elimination are determined based on the highest $s_{i,j}$ value in the sequence. The actual attribute to be removed, either i or j, is the one whose removal produces in the smaller value of *SS*. The process may be repeated, for example, until a user-defined minimum number of attributes is reached. Such a

procedure may be classified as a *Sequential Backward Selection* feature selection process [13] that starts with a complete set of attributes and gradually eliminates highly correlated ones. If this approach is used to create a *Viz3D* visualization of the same data exhibited in Figure 6, the 8 clusters become readily identifiable (see Figure 9), as opposed to the previous visualization in which attributes are arranged in the same sequence as they appear in the data set. Figure 8 shows visualizations of another synthetic data set, known as *Quadruped Mammals*[4]. This data, typically used to train classifiers, contains records with 72 attributes describing approximations of members (head, tail, four legs, body and neck) of four animals (1-cat, 2-dog, 3-horse and 4-giraffe). Each member is approximated by a cylinder and represented by nine attributes.

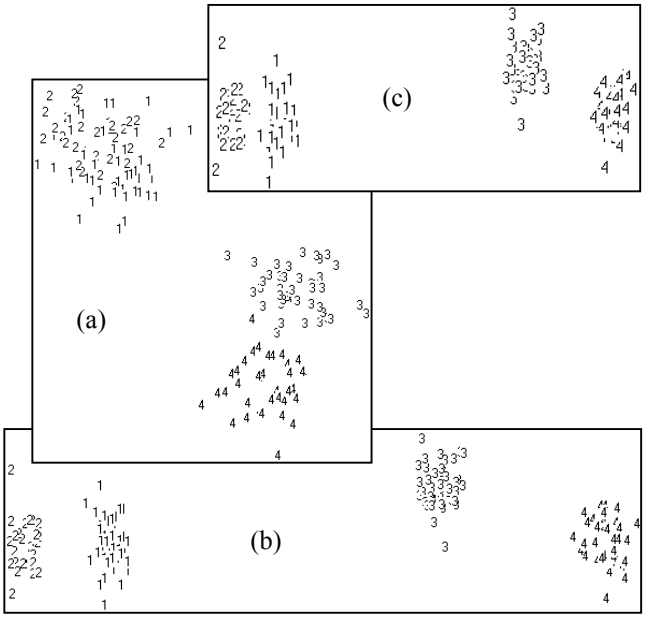


Figure 8 *Viz3D* visualizations of the *Quadruped Mammals* data: a) created with attributes arranged as in the data file; b) after a similarity-based sequencing of axes; and c) created using only the 40 'most relevant' attributes.

Figure 8(a) shows a *Viz3D* visualization created with attribute axes arranged in the same sequence as they appear in the data set, after some user interactions. The resulting visualization shows an overlapping of groups 1 and 2 (cat and dog) and a poor separation between groups 3 and 4 (horse and giraffe). Figure 8(b) shows the projection obtained with the similarity based arrangement of attributes, in which the four classes are well separated. Figure 8(c) shows the visualization created with the similarity-based arrangement approach, after reduction to 40 attributes by eliminating highly similar ones. Experiments with real and synthetic data sets showed that both *RadViz* and *Viz3D* visualizations always improve with the similarity-based axes arrangement strategy as far as cluster identification is concerned. The same strategy is applicable to other axes-based visualization techniques, such as Parallel Coordinates.

5. Visual Clustering and Density Visualization

Viz3D supports a user-driven visual clustering approach that is very effective for handling high-dimensional data. Two visual clustering approaches are introduced in this section. In the first one the user interacts directly with the *Viz3D* representation, visually delimiting data clusters. In the second approach the user interacts with a density enriched *Viz3D* visualization to identify clusters in very noisy data sets, which generate severely cluttered visualizations.

Visual clustering approaches [8][1] allow users to interact with visual representations to drive cluster identification. In our solution the user interacts with a *Viz3D* representation, using the mouse to delimit regions that possibly contain a cluster. An iterative process is conducted in which, once a cluster is visually delimited, its containing records are labeled accordingly and the user may proceed to identify other clusters. The process is illustrated in Figure 9 with the Sint3 data. Figure 9(a) shows an initial visualization, in which the user has delimited a rectangular region containing a projected cluster. All data elements contained in the rectangle are selected, and in the next step only selected elements are shown. Because selection is on the 2D projected space, as the user rotates the visualization s/he observes different projected views, showing records that clearly do not belong to the cluster, as illustrated in Figure 9(b). The user can once again delimit regions in these new projections, refining the selection of records in the cluster (Figures 9(c) and (d)), until a satisfactory choice has been made, i.e., until a 'precise' demarcation of the cluster, according to the user visual perception, has been achieved. This situation is illustrated in Figure 9(e), which shows the result of selecting the region shown in Figure 9(d). Once the group has been completely delimited, its records are labeled with a group identifier, e.g., Group 1. Then, the next visualizations show all data records but those already labeled as Group 1, and the process may be repeated for other groups. Figure 9(f) shows the *Sint3* data visualization without the records assigned to Cluster 1.

Cluttering gradually diminishes along the process, simplifying the task of identifying new clusters. Thus, the process is applicable even to very cluttered and noisy visualizations. However, in severe cluttering conditions and very noisy data, visual demarcation may become difficult and tiresome. Exhibiting density information, rather than the raw data, provides a more suitable representation for visual clustering. Density visualizations

[4] Available at http://lib.stat.cmu.edu/datasets/

'filter' the information exhibited, producing visualizations that are more effective for revealing patterns.

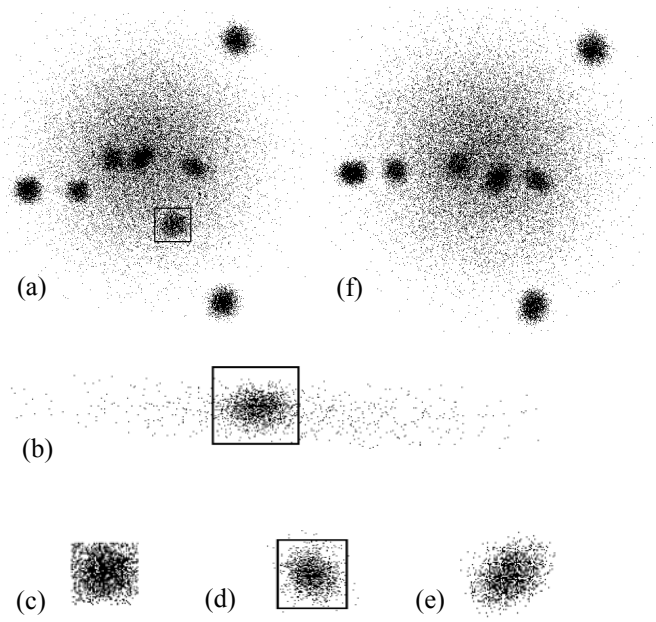


Figure 9 Interactive visual clustering with *Viz3D*.

Computing density on the original n-dimensional space is prohibitive for large values of n. We compute data density in the 3D discrete *Viz3D* projected space, creating a discrete density matrix data whose dimensions are determined by the resolution of the projected space. The algorithm is given in Box 1. The *Viz3D* projection generates the projected (x,y,z) coordinates of the m n-dimensional records. Data densities at each position of the 3D discrete space are then computed and stored on the density matrix. Parameter K_h in the algorithm is given by $(K_w - 1)/2$, where K_w is the width of the kernel density estimation filter. Our implementation uses a square wave filter, rather than a Gaussian. The resulting visualization is created from the density matrix. Each matrix element density(i,j,k) is projected on a voxel with coordinates (x,y,z), whose value is set proportional to the density value in the corresponding matrix position. Assuming a monochrome visualization, voxels mapping regions with greater data density will be projected in pixels painted with higher intensities, thus allowing a direct visual identification of regions with clusters of data. Inverting the mapping so that lower densities are mapped to higher pixel intensities allows the identification of data outliers.

In an exploratory process, a user controls two parameters: the width of the kernel density estimation filter (K_w) and the lower density threshold for the contents of a voxel to be projected. This threshold controls the noise level in the visualization, as only voxels with a minimum data occupation will contribute. Varying kernel width allows controlling the level of clustering process, as in a hierarchical clustering approach: a minimum width (one) corresponds to considering each voxel region as a group, whereas a maximum width corresponds to considering the contents of all voxels as a single group. Figure 10 shows varying density visualizations of the *Sint3* data, created using the optimal axes sequence. The two numbers shown at the bottom of each frame inform the kernel mask width (left), and the lower density threshold (right).

Viz3D density visualization algorithm

Let $D_{m \times n}$ be the data matrix, where m is the number of records and n is the data dimensionality;

1 - Construct the 3D density matrix $density_{w \times w \times w}$;
  // w is given by the volume voxel resolution

2 - Let $density_{p,q,r} = 0 \; \forall \; p,q,r \mid \; p \in [0,w]$, $q \in [0,w]$ and $r \in [0,w]$;

3 - **For** $i = \{1,2,...,m\text{-}1\}$ **do**
    // compute data frequency at each point in the 3D space
    3.1 compute projected coordinate (x,y,z) with Eq. 2
    3.2 let $f[x,y,z] = f[x,y,z] + 1$;

4 - **For** each $x,y,z = \{0,1,...,d\}$ **do**
    // compute density using the density estimation kernel
    For each $p,q,r = \{-K_h,...,K_h\}$ **do**
      $density\,[x,y,z] = f[x,y,z] + f[x+p,y+q,z+r]$;

5 - **If** $density[x,y,z]$ > threshold **then** project and display the 3D voxel volume (x,y,z) setting voxel intensity proportional to the value in the corresponding density matrix position.

Box 1 *Viz3D* Density visualization algorithm.

The sequence of steps (a) too (d) in Figure 10 illustrates the effect of increasing the density threshold: an initial value is gradually increased, so that at each step only regions with increasing data densities are being shown, thus highlighting the clusters in the data. The sequence (e) to (g) illustrates the effect of reducing kernel width, K_d. The same visual clustering process supported by conventional *Viz3D* visualizations, described in the beginning of this section, is also applicable to density-based visualizations.

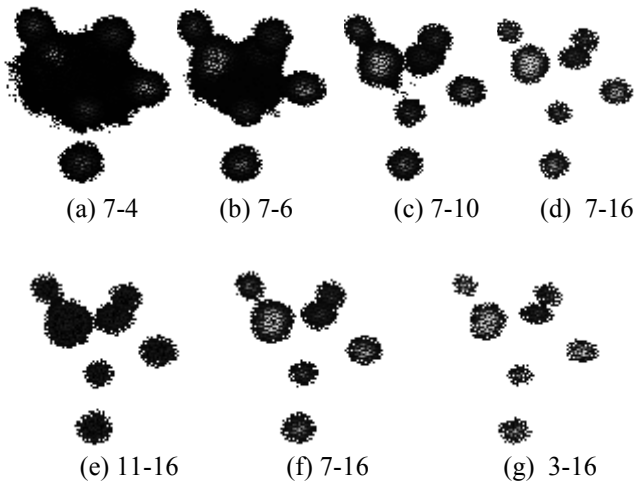


Figure 10 Density visualizations of the *Sint3* data, with increasing density threshold values.

6. Conclusions and Further Work

Viz3D creates an interactive 3D representation of multidimensional data that offers an interesting alternative to other projection techniques. It has linear complexity on both the number of data elements and data dimensionality. Projections created are comparable, in quality, to those produced by techniques such as PCA and *FastMap*. The technique is effective to convey clusters of data and handles the visual cluttering and occlusion problems better than 2D approaches. *Viz3D* and its density-based variation both support effective visual clustering of high-dimensional data in a way that allow users to perform a good qualitative evaluation of data prior to application of analytical algorithms.

Any projection strategy that maps data to a lower-dimensional space implies in some information loss. *Viz3D* ensures that records that are close in the input space are mapped close together in the projection space. However, distant records in the input space may be projected near to each other. Consequently, the visual clustering approach proposed may produce clusters that do not reflect precisely data organization in the *n*-dimensional space. A cluster quality evaluation on the input space could be conducted so that complementary information is given to the user, who may use it to refine cluster identification. Several metrics for evaluating the quality of clusters could be used in this context to provide users with additional feedback [6]. Complementary visualizations of the cluster data elements that allow a qualitative evaluation of data organization on the input space, such as Parallel Coordinates, could also be provided to complement information.

Acknowledgements

The authors acknowledge the financial support of FAPESP (The State of São Paulo Research Funding Agency) Grant 01/07566-2, and CNPq (The Brazilian National Research Funding Agency) Grants 521931/97-5 and 141584/01-7.

References

[1] Aggarwal, C. A Human-Computer Cooperative System for Effective High Dimensional Clustering, *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining* 2001, 221-226.

[2] Ankerst, M., Berchtold, S., Keim, D.A. Similarity Clustering of Dimensions for Enhanced Visualization of Multidimensional Data, *Proc. IEEE Symp. Information Visualization* 1998 52-60.

[3] Card, S.K., Mackinlay, J.D., Shneiderman, B., *Readings in Information Visualization, Using Vision to Think*, Morgan Kaufmann, 1999.

[4] Faloutsos, C., Lin, K. FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets. *Proc. SIGMOD Conf*. 1995, 163-174.

[5] Grinstein, G.G., Hoffman, P.E., Pickett, R.M. Benchmark Development for the Evaluation of Visualization for Data Mining. In: *Information Visualization in Data Mining and Knowledge Discovery*, Fayyad, U., Grinstein, G.G. (eds.), 2001.

[6] Halkidi, M., Vazirgiannis, M., Batistakis, I. Clustering Validation Techniques, *Intelligent Information Systems J*., 17(2-3), 2001, 107-145.

[7] Hinneburg, A., Keim, D.A. An Efficient Approach To Clustering in Large Multimedia Databases With Noise, *Proc. Int. Conf. Knowledge Discovery and Data Mining* 1998, 58-65.

[8] Hinneburg, A., Keim, D.A., Wawryniuk, M. HD-Eye: Visual Mining of High-Dimensional Data, *IEEE Computer Graphics and Applications,* 1999, 22-31.

[9] Hoffman, P.E. *Table Visualizations: A Formal Model and its Applications*. Doctoral Diss., Computer Science Dept., University of Massachusetts, Lowell, MA, 1999.

[10] Inselberg, A. The Plane with Parallel Coordinates, *The Visual Computer*, Special Issue on Computational Geometry, 1, 1985, 69-92.

[11] Kandogan, E. Visualizing Multi-Dimensional Clusters, Trends, and Outliers using Star Coordinates, *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining* 2001, 107-116

[12] Keim, D.A. Information Visualization and Visual Data Mining, *IEEE Trans. Visualization and Computers Graphics*, 8, 2002, 1-8.

[13] Molina, L.C., Belanche, L., Nebot A., Feature Selection Algorithms: A Survey and Experimental Evaluation, *Proc. IEEE Int. Conf. on Data Mining* 2002, 306-313.

[14] Oliveira, M.C.F., Levkowitz, H. From Visualization to Visual Data Mining: A Survey. *IEEE Trans. Visualization and Computer Graphics*, 9, 2003, 378-394.

[15] Pearson, K. On Lines and Planes of Closest Fit to System of Points in Space, *Philosophy Magazine*, 6, 1901, 559-572.

[16] Rodrigues Jr, J.F., Traina, A.J., Traina Jr, C., Frequency Plot and Relevance Plot to Enhance Visual Data Exploration. *Proc. XVI Brazilian Symp. Computer Graphics and Image Processing,* 2003, 117-124.

[17] Wegman, E.J., Luo, Q. High Dimensional Clustering using Parallel Coordinates and the Grand Tour. *Proc. Conf. German Classification Society*, Freiburg, Germany, 1996.

[18] Yang, J., Ward, M.O., Rundensteiner, E. A., Huang, S. Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets, *VisSym* 2003.