Informatics in Education, 2002, Vol. 1, 129–148
© 2002 *Institute of Mathematics and Informatics, Vilnius*

Data Visualization: Ideas, Methods, and Problems

Vydūnas ŠALTENIS,
Vilnius Pedagogical University
Studentu 39, LT-2600 Vilnius, Lithuania
e-mail: saltenis@ktl.mii.lt

Jūratė AUŠRAITĖ
Institute of Mathematics and Informatics
Akademijos 4, LT-2021 Vilnius, Lithuania
e-mail: juratea@mail.lt

Received: July 2002

Abstract. The science of data visualization requires both a theory of perception and of computer graphics. This article explores the convergence sciences of perception and techniques of visualization. Bertin's Image Theory of visualization research offers a promising technology for transforming an indigestible mass of numbers into a medium which human beings can understand, interpret, and explore.

Visualization techniques are of increasing importance in exploring and analyzing large amounts of multidimensional information. In the article we try to set some classes of visualization techniques. We describe some of the most popular multidimensional data research methods, and present some examples.

Visualization is applied in education of old. It must make connections between knowledge the learner has and the knowledge being taught. Therefore in order to design effective visualizations it is necessary to know (or at least have a theory about) what the learner knows. This is especially important in the context of education.

Key words: data visualization, proximity visualization, clustering methods, projection methods, multidimensional scaling, self-organizing map.

1. Introduction

Visualization, a term used in the industry since the publication of the National Science Foundation report *Visualization in Scientific Computing* (McCormick *et al.*, 1987), is a form of communication that transcends application and technological boundaries. With the advent of computer graphics, researchers can convert entire fields of variables (representing density, pressure, velocity, entropy and so on) to color images. The information conveyed to the researcher undergoes a qualitative change because it brings the eye-brain system, with its great pattern-recognition capabilities, into play in a way that is impossible with numerical data alone. Data visualization is an established technique for exploration, analysis, and presentation of data.

For example, an observer instantly sees the vortices, shock systems, and flow patterns in the visualization of hydrodynamic calculation, while these same patterns are invisible in mere listings of several hundred thousand numbers, each representing field quantities at one moment of a time. When computing the space-time solution of the laws of physics, particular numerical quantities at each event in time-space are not important; rather, what is important is understanding of the global structure of the field variables that constitute the solution and the causal interconnections of the various components of it.

At present, visualization is used in various scientific and engineering research areas (DeFanti *et al.*, 1989), for example:

- the use of interactive computer graphics in chemistry, which began in 1964;
- scientific computation applied to medical imaging has created opportunities in diagnostic medicine;
- visualization helps mathematicians to understand various equations that are too complex to conceptualize otherwise;
- meteorologists obtain information on natural behavior that cannot be safely observed: differing vertical wind, temperature, pressure and moisture structures, etc.;
- the planetary study involves the accumulation of huge volumes of data on the planets in the solar system.

2. Theory of Perception in Data Visualization

The science of data visualization requires both a theory of perception and of computer graphics. However, visualization designers have paid relatively little attention to perceptual issues. Bertin's (1983) *Image Theory* can serve as a guide to visualization design. Visualization research offers a promising technology for transforming an indigestible mass of numbers into a medium which humans can understand, interpret, and explore.

As shown in Fig. 1, the first stage maps numbers (data/process) to images by means of some algorithmic technique. The second stage maps images to insight by means of perception.

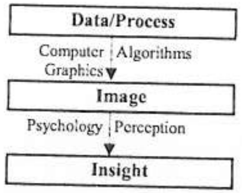

Fig. 1. The transformation from numbers to insight requires two stages.

2.1. *Bertin's Image Theory Concepts*

Bertin's key concept is *image*, from which the theory derives its name. An *image* is the fundamental perceptual unit of visualization. In terms of Bertin, three stages of information are extracted (see Fig. 2):

1) external identification is that the graph is showing something, for example, height and weight;
2) internal identification is perception that the height component is mapped onto the vertical axis of the plane and weight onto the horizontal axis; and
3) perception of correspondence is noting that the data point is located at a particular intersection of visual variables X and Y.

According to Bertin, there are two functionally different classes of visual variables: planar and retinal. Components may be represented by six retinal variables: size, color, shape, orientation, texture, and brightness. Each retinal variable can be used with three types of implantation. Bertin believes that an efficient visualization is limited to three-component visualizations. It is not possible to create an efficient four-component visualization by adding a second retinal variable: you could not, for example, add a component for country of birth by coding each mark with the shape reflecting nationality.

Bertin believes that the major problem of most data visualizations is the choice of visual variables with an inappropriate length or level of organization. Length refers to the number of categories or steps (distinguishing different colors, brightness levels, etc.) A visual variable must have a length equal to or greater than the component it represents. A major problem for the visualization designer is to match the data to a visual variable with the correct length.

The level of perceptual organization specifies the type of data scale: nominal, order, or ratio, which each visual variable can portray (Table 1). That is, suppose the goal is to allow the viewer to extract ratios from the visualization, e. g., to immediately see that one value is twice as much as another. Brightness could not be used, for example, because doubled intensity produces only about a 1.4 factor increase in the perceived brightness.

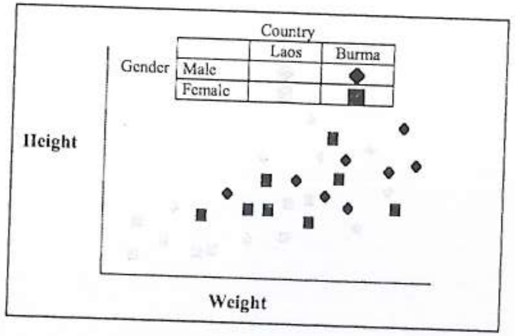

Fig. 2. Height/weight graph, with additional components represented by marks of different shape (gender) and brightness (nationality).

Table 1

Bertin levels of perceptual organization

	Associative	Selective	Ordered	Quantitative
Planar	Yes	Yes	Yes	Yes
Size		Yes	Yes	Yes
Brightness		Yes	Yes	
Texture	Yes	Yes	Yes	
Color	Yes	Yes		
Orientation	Yes	Yes		
Shape	Yes			

Associative Organization. Bertin subdivides the nominal scale level into two subcategories: associative and selective. Associative perception is the lowest organization level, allowing grouping of all the elements of a variable in spite of different values. According to Bertin, planar dimensions, texture, color, orientation (point and line implantations only), and shape are associative, while size and brightness are not.

Selective Organization. The concepts of associative and selective organization have implications for a wide range of issues in data visualization, such as the presentation of multiple data views.

Fig. 3 shows a typical example. One window shows data in graph form, while the other shows a table. It is obvious if the two views are linked by color. If the informational view in the right window had a different color than it's corresponding object in the left view, the viewer would incorrectly form a nominal scale grouping.

Ordered Organization. The associative and selective perception provides only a nominal scale classification. Bertin suggests that some variables allow the data to be ranked. Observers can see that one value of a variable represents a larger or a smaller quantity than another. For example, the increased age could be represented by data points with increased brightness. In this case, it would be possible to make decisions about relative height/weight of elder or younger people. Bertin considers the planar variables: brightness, size, and texture variables as ordered while excluding shape, color, and orientation.

Quantitative Organization. The highest level of organization is quantitative which permits a direct extraction of ratios. That is, the ratio of variable values maps directly to the ratio of the data values. If age were represented by data points of different size, which is a quantitative variable, the viewer could immediately see which individuals were twice as old as others. While ordered data permit only a relative magnitude of the represented quantities, quantitative variables support the perception of ratios. Only size and the planar dimensions are quantitative.

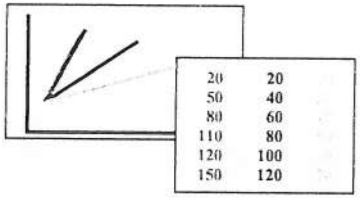

Fig. 3. One window shows data in graph form, while the other shows a table.

2.2. *M. Green's Extension Beyond the Bertin Image Theory*

In his work M. Green (1998) tries to show that Bertin bases his theory solely on his introspection and makes no attempt to provide any empirical data support. He outlines some basic researches on the human vision and suggests the ways of using them to guide data visualization.

M. Green states that:

1) advent of computer graphics permits the use of several new visual variables that Bertin does not discuss;
2) there are several secondary perceptual effects which complicate a direct application of *Image* theory; and
3) perhaps the most important, Bertin's belief that *images* are always limited to three components is false.

In Table 2, M. Green uses research data to propose corrections and extensions to Bertin's original level of organization assignments (Table 1).

Shape can be selective. The results of research (Green, 1991) show that the list of primitives largely reflects those coded by the brain in the primary visual cortex. Accordingly, the list agrees well, but not perfectly, with Bertin's list of retinal variables. The image is typically processed by a series of cascaded linear and nonlinear filters, which

Table 2

M. Green's updated Bertin levels of perceptual organization

	Associative	Selective	Ordered	Quantitative
Planar	Yes	Yes	Yes	Yes
Size		Yes	Yes	Yes
Brightness		Yes	Yes	Yes-if scaled
Texture	Yes	Yes	Yes	
Color (Hue)	Yes	Yes	Yes-limited	
Orientation	Yes	Yes		
Shape	Yes	Yes		
Motion: Velocity		Yes	Yes	Yes-if scaled
Motion: Direction		Yes		
Flicker: Frequency		Yes	Yes	Yes-if scaled
Flicker: Phase		Yes		
Disparity		Yes	Yes	

presumably correspond to neuronal levels in the brain. The decision as to whether two shapes/textures are associative or selective is determined by the similarity of their outputs in the model. In theory, such a model should make it possible to predict whether shapes/textures would be associative or selective. The applicability of such models to predicting immediate visualization of diagrams is unknown.

Color can be ordered. Color consists of three properties: hue, saturation, and brightness (or hue, chroma, and lightness). It is true that hue is in general a nominal variable: red, green, blue do not form an ordered scale. But, over small ranges, hue can be ordered. For each hue, there is a specific example, called a "unique hue", which is perfect in that it is not tinged by any other hue. For example, unique yellow contains no trace of either red or green. An ordered scale of yellow could therefore be constructed starting around unique yellow and extending to either (but not both) unique red or unique green. Secondly, saturation, the amount of white mixed with a spectral, i.e., the purest, hue, is also an ordered variable.

Brightness can be "quantitative". Brightness is ordered but not quantitative because it produces a psychometric function with an exponent less than one. It should be easy, using existing psychophysical data, to promote brightness to a quantitative level of organization by rescaling the intensity axis. The visual system's brightness exponent varies with conditions and can be as low as 0.33.

M. Green (1998) proposes, that data can be displayed by means of several new visual variables, including motion, flicker, and disparity (stereo depth).

Motion. One new retinal variable is motion, which can be split into two subvariables: velocity and direction. It is unlikely that motion is associative. Objects with different motions are generally perceived as lying on different surfaces. Segmentation of an image into constituent surfaces is perhaps the earliest and the most primitive perceptual function. Motion velocity is likely to be ordered, since it is a continuum of magnitude, and observers can readily discriminate steps of increasing value. Direction of motion is likely to be unordered, since it is not a variable of magnitude. Velocity is not quantitative because it produces a compressive psychometric function, through it might be made quantitative by appropriate rescaling

Flicker. Like motion, flicker also has two subvariables – frequency and phase. Flicker frequency, the speed of the on-off cycles, is selective by common sense, can be readily ordered, and could be promoted to quantitative by rescaling. There are some problems with flicker frequency. First, it has a very short length. Secondly, apparent brightness varies with frequency, making these somewhat "integral dimensions". Flicker frequency is not associative.

Binocular Disparity. A third new variable is binocular disparity, which can be created by giving the left and right eyes slightly different views of the same visualization. Instead, disparity provides good relative depth information, revealing whether one object is closer or further than another, and is a very powerful cue for image segmentation. Its main role is to help break scenes into meaningful surfaces, so it is likely to be poor for association but ideal for selection. Since it provides only relative depth information, it is a poor choice to represent quantitative data but good for representing ordered data.

Bertin asserts that *images* can contain no more than three components; however, M. Green's studies have found that four or more component *images* are possible. A subset of visual variables permits the conjunction search. Observers can effortlessly search for conjunctions of shape or color with disparity (Nakayama and Silverman, 1986) or motion. These searches have four components: 2 planar, 1 Bertin retinal variable, plus disparity or motion, variables which Bertin does not include in his analysis.

3. Proximity Visualization

Proximity Visualization is the visualization of abstract data collections. Database tables with many attributes, graphs, and multimedia collections are types of data collections for which it is difficult to design useful visual representations. However, similarity between elements of such collections can be measured, and a good overview picture should respect this proximity information by positioning similar elements close to each other, while far from dissimilar ones. Such a visualization is, in effect, a topology, preserving map of the underlying data collection.

3.1. *Data Types*

The proximity of a pair of objects from a data collection can be expressed either as their similarity, mutual agreement, or dissimilarity. A measure of dissimilarity between a pair of objects from a collection is a dissimilarity coefficient. A number of coefficients could be used with a particular data collection. Each data type could be serviced by different dissimilarity coefficients.

A quantitative variable can either be measured on an interval or a ratio measurement scale, for example temperature.

An ordinal scale is weaker than quantitative in that it also induces an ordering of objects, but does not make any statement about the magnitude of the differences. An example of an ordinal scale is a well defined ordering A > B > C > . . . >F. It is impossible to know how the differences between the pairs of grades relate, for instance, whether the difference between A and B is greater than that between B and C. However, the difference between A and C must be greater than the difference between A and B, due to the inequalities, and an effective dissimilarity coefficient for a pair of objects measured on an ordinal scale can be based on differences in their rank.

A nominal scale allows for the weakest form of measurement, as it does not enforce any ordering of objects. A number of non-overlapping categories are defined and an object can belong to any single one of these. The similarity of a pair of objects is defined in terms of their category membership: if they belong to the same category, they are considered as equivalent, otherwise, they are completely different. An example of such an exclusive categorization is a person's zodiac sign.

Binary. A binary scale of measurement is a special case of a nominal scale with only two categories, for example, male–female.

Heterogeneous. A collection of entity descriptions may be conveniently represented by a set of objects with appropriate attributes. The utility of relational and object databases is based on this premise. The canonical representation of such data is a table with one row for each object, and a column for each attribute. In general, the attributes cannot be expected to be homogeneous, and thus a dissimilarity coefficient for data tables has to combine attributes measured on arbitrary scales, to give an overall dissimilarity between pairs of objects (rows).

Relationships. Binary relationships xRy within a set V of entities can be expressed as a graph $G(V, E = \{(a,b): a, b \in V, aRb\})$, with an edge (a,b) whenever entities a and b enter into a relationship, and an entity being synonymous to a vertex. In general, a pair of entities might enter into an indirect relationship through one or more intermediate entities. Such a chain of vertices defines a path through the graph G that has a length equal to the number of chained vertices minus one, i.e., the number of edges that have to be traversed.

Images. Humans are adept at determining similarity between a given pair of images, by establishing semantic relationships between these objects. This process is difficult to emulate on a computer system. However, it is practical to automatically extract low-level features from the images, such as color and texture, and compare their similarity.

Text Corpus. It is common practice to index a corpus of documents on themes occurring within it, e.g., nouns or phrases, to facilitate querying for relevant themes. In a classic model of Information Retrieval – the vector model – each document is represented by a weight vector

$$\vec{w} = (w_1, \ldots, w_q)^T,$$

where element w_a specifies the relative importance of a theme t_a in the document, and q is the total number of themes and the dimensionality of the weight vector space (Baeza-Yates and Ribeiro-Neto, 1999). If the theme t_a does not occur in the document, the element w_a is simply set to 0, otherwise w_a is made proportional to the frequency of t'_as occurrence in the document, and inversely proportional to the commonness of t_a within the corpus. Thus, a theme frequently occurring in a given document will be assumed to have a high relevance, e.g., "computer" in a corpus of computer science articles.

Proximity. There are data collections that consist solely of proximity measurements, for example, results of an experiment where a subject has been asked to rate how similar pairs of stimuli are. The only meaningful visual representation for such data is the one based on proximity. The judgements of dissimilarity are unlikely to obey Euclidean or even metric properties, – most importantly, triangle inequality. However, by adding a sufficiently large positive constant to every dissimilarity, they can be made metric and Euclidean (Gower and Legendre, 1986; Borg and Groenen, 1997).

4. Classification of Multidimensional Visualization Techniques

Many novel visualization techniques have been developed and existing techniques have been extended to work with 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 have no inherent two- or three-dimensional semantics. In general, relational databases can be seen as multidimensional data sets with the attributes of a database corresponding to the dimensions.

The visualization techniques can be divided into geometric projection techniques (Inselberg, 1985; Inselberg and Dimsdale, 1990), icon-based techniques (Pickett and Grinstein, 1988; Beddow, 1990), pixel-based techniques (Keim, 1995; Keim and Kriegel, 1994; Keim *et al.*, 1995), hierarchical techniques (LeBlanc *et al.*, 1990; Robertson *et al.*, 1991; Shneiderman, 1992), graph-based techniques (Eick and Wills, 1993; Becker *et al.*, 1995), and combinations thereof (Asimov, 1985; Ahlberg and Shneiderman, 1994).

Geometric projection techniques display includes techniques from exploratory statistics, such as scatterplot matrices (Andrews, 1972; Cleveland, 1993). Other geometric projection techniques include projection views (Furnas and Buja, 1994; Spence *et al.*, 1995), hyperslice (Wijk and Liere, 1993), landscapes (Wright, 1995), and parallel coordinates visualization technique (Inselberg, 1985; Inselberg and Dimsdale, 1990). The parallel coordinate technique maps the k-dimensional space onto the two-display dimensions by using k equidistant axes which are parallel to one of the display axes. The axes correspond to the dimensions. Each data item is presented as a polygonal line, intersecting each of the axes at that point which corresponds to the value of the considered dimension.

Another class of visual data exploration techniques is the *icon-based* techniques. The idea is to map the attribute values of a multidimensional data item to the features of an icon. Classes of icon-based techniques are stick figures (Pickett and Grinstein, 1988), shape-coding (star icons, Chernoff faces) (Beddow, 1990), and color icons (Levkowitz, 1991; Keim and Kriegel, 1994). In the case of the stick figure technique, for example, two dimensions are mapped to the display dimensions and the remaining dimensions are mapped to the angles and/or limb length of the stick figure icon. If the data items are relatively dense with respect to the two-display dimensions, the resulting visualization presents texture patterns that vary according to the characteristics of the data.

The basic idea of *pixel-based* techniques (Keim, 1996) is to map each dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas. Since, in general, pixel displays use one pixel per data value, the techniques allow the visualization of the largest amount of data possible (up to about 1,000,000 data values). Pixel techniques use different arrangements for different purposes. By arranging the pixels in an appropriate way, the resulting visualization provides detailed information on local correlations, dependencies, and hot spots.

Stacked display techniques are tailored to present data partitioned in a *hierarchical* fashion. In the case of multidimensional data, the data dimensions to be used for partitioning the data and building the hierarchy have to be selected appropriately. An example of a stacked display technique is dimensional stacking (LeBlanc *et al.*, 1990). The basic idea is to embed one coordinate system inside another coordinate system, i.e., two attributes form the outer coordinate system, two other attributes are embedded into the outer coordinate system, and so on. The display is generated by dividing the outmost level coordinate systems into rectangular cells and, within the cells, the next two attributes are

Table 3

An attempt to compare multidimensional visualization techniques
(++: very good, +: good, 0: neutral, -: bad, - -: very bad)

		Clustering	Multivariate hot spot	No. of attributes	No. of data objects	Categorical data	Visual overlap	Learning curve
Geometric Techniques	Scatterplot Matrices	++	++	+	+	-	0	++
	Landscapes	+	+	-	0	0	+	+
	Prosection Views	++	++	+	+	-	0	+
	Hyperslice	+	+	+	+	-	0	0
	Parallel Coordinates	0	++	++	-	0	- -	0
Icon-based Techniques	Stick Figures	0	0	+	-	-	-	0
	Shape Coding	0	-	++	+	-	+	-
	Color Icon	0	-	++	+	-	+	-
Pixel-oriented Techniques	Query-Independent	+	+	++	++	-	++	+
	Query-Dependent	+	+	++	++	-	++	-
Hierarchical Techniques	Dimension Stacking	+	+	0	0	++	0	0
	Treemap	+	0		0	++		0
	Cone Trees	+	+	0	+	0	+	+

used to span the second level coordinate system. This process may be repeated one more time. The usefulness of the resulting visualization largely depends on the data distribution of the outer coordinates and, therefore, the dimensions which are used for defining the outer coordinate system have to be selected carefully. A rule of thumb is to choose the most important dimensions first. Other examples of hierarchical techniques are a treemap (Shneiderman, 1992; Johnson, 1993), and cone-trees (Robertson *et al.*, 1991).

This brief classification is aimed at providing a more structured understanding of a large number of available multidimensional visualization techniques. It can also be used as a starting point to compare the available techniques, to improve the existing ones, and to develop new techniques. Table 3 is trying to compare a number of visualization techniques. The comparison of the visualization techniques is based on their suitability for certain:

- **task characteristics**, such as clustering and multivariate hot spots,
- **data characteristics**, such as number of dimensions (attributes), number of data objects, and suitability for categorical data,
- **visualization characteristics**, such as visual overlap and learning curve.

Ripley (1996) divides statistical data-analysis methods into clustering methods, projection methods, and multidimensional scaling (MDS) methods.

5. Clustering Methods

The goal of clustering is to reduce the amount of data by categorizing or grouping similar data items together. Such grouping is pervasive in the way humans process information and one of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories or taxonomies (Jardine and Sibson, 1971; Sneath and Sokal, 1973). The methods may also be used to minimize the effects of human factors in the process.

Clustering methods can be divided into two basic types: hierarchical and partitional clustering.

Hierarchical clustering proceeds successfully by either merging smaller clusters into larger ones, or by splitting larger clusters. The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. The end result of the algorithm is a tree of clusters called a dendrogram, which shows how the clusters are related.

Partitional clustering attempts to directly decompose the data set into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

A commonly used partitional clustering method, K-means clustering (MacQueen, 1967), uses the criterion function – the average squared distance of the data items x_k from their nearest cluster centroids:

$$E_K = \sum_k ||x_k - m_{c(x_k)}||^2, \tag{1}$$

where $c(x_k)$ is the index of the centroid that is closest to x_k . One possible algorithm for minimizing the cost function begins by initializing a set of K cluster centroids denoted by m_i, $i = 1, \ldots, K$. The positions of m_i are then adjusted iteratively by first assigning the data samples to the nearest clusters and then recomputing the centroids. The iteration is stopped when E does not change markedly any more. In an alternative algorithm, each randomly chosen sample is considered in succession, and the nearest centroid is updated.

Eq. 1 is also used to describe the objective of a related method, vector quantization (Gersho, 1979; Gray, 1984; Makhoul *et al.*, 1985). In the vector quantization, the goal is to minimize the average (squared) *quantization error*, the distance between a sample x and its representation $m_{c(x)}$. The algorithm for minimizing (1) that was described above is actually a straightforward generalization of the algorithm proposed by Lloyd (1957) in a one-dimensional setting.

The interpretation of clusters may be difficult. Most clustering algorithms prefer certain cluster shapes, and the algorithms will always assign the data to clusters of such shapes even if there were no clusters in the data. Usually the goal is not just to compress the data set, but also to make inferences about its cluster structure. The results of the cluster analysis need to be validated (Jain and Dubes, 1988).

Another potential problem is that the choice of the number of clusters may be critical: quite different kinds of clusters may emerge when K is changed. Good initialization of the cluster centroids may also be crucial; some clusters may even be left empty if their centroids lie initially far from the distribution of data.

6. Projection Methods

The goal of the *projection methods* is to represent the input data items in a lower-dimensional space in such a way that certain properties of the structure of the data set are preserved as faithfully as possible.

6.1. *Linear Projection Methods*

Principal Component Analysis (PCA) is a multivariate statistical method for linearly transforming a sample of N p-variate vectors

$$\vec{X} = [x_{ik}\colon i = 1, \dots, N;\ k = 1, \dots, p]$$

into a new sample of $q-$variate vectors

$$\vec{Y} = [y_{il}\colon l = 1, \dots, q],$$

such that the columns of $\vec{Y}$ are uncorrelated, and $q \leqslant p$ (Flury, 1997). Each of the q derived variables is expressed as a linear combination of p correlated, measured variables (Hotelling, 1933).

The q derived variables are referred to as Principal Components (PCs), and form a system of orthogonal axes. If we consider $\vec{X}$ to be a configuration of N points

$$\vec{x}_i = (x_{i1}, \dots, x_{ip})^T$$

in a p-dimensional Euclidean space, then the first PC defines a line through this space that minimises the sum of squared distances of the points from it, and thus maximises the variance of coordinates $\{x'_i = y_{i1}\}$ of an orthogonal projection of $\{\vec{x}_i\}$ on this line (Pearson, 1901). The second PC is a line that maximises the variance of the projection coordinates $\{x''_i = y_{i2}\}$ of the points, subject to being perpendicular to the first PC. Taken together, the first two PCs give a plane of the closest fit to the configuration $\vec{X}$, i.e., one that minimises the sum of squared distances of the points to that plane. The remaining PCs are defined recursively in this manner, and individually account for a smaller amount of the total variance. The first $q' < q$ principal components give the best linear approximation.

6.2. *Nonlinear Projection Methods*

PCA cannot take into account nonlinear structures, consisting of arbitrarily shaped clusters or curved manifolds. Projection pursuit tries to express some nonlinearities, but if the data set is high-dimensional and highly nonlinear, it may be difficult to visualize it with linear projections onto a low-dimensional display even if the projection angle is chosen carefully.

Several approaches have been proposed for reproducing nonlinear higher-dimensional structures on a lower-dimensional display. The most common methods allocate a representation for each data point in the lower-dimensional space and try to optimize these representations so that the distances between them would be as similar to the original distances as possible. The methods differ in how the different distances are weighed and how the representations are optimized.

6.2.1. *Multidimensional Scaling*

MDS refers to a group of methods that is widely used especially in behavioral, econometric, and social sciences to analyze subjective evaluations of paired similarities of entities. The starting point of MDS is a matrix consisting of paired dissimilarities of the entities.

The goal of MDS methods is not merely to create a space that would represent the relations of the data faithfully, but also to reduce the dimensionality of the data set to a sufficiently small value to allow visual inspection of the set There exists a multitude of variants of MDS with slightly different cost functions and optimization algorithms.

The algorithms of MDS can be roughly divided into two basic types: metric and nonmetric MDS.

In the original *metric MDS* (Torgerson, 1952; Young and Householder, 1938), the distances between the data items have been given, and the configuration of points that would give rise to the distances is sought. Often a linear projection onto a subspace obtained by PCA is used. The key idea of the method to approximate the original set of distances to distances corresponding to the configuration of points in a Euclidean space can, however, also be used for constructing a nonlinear projection method. If each item x_k is represented by a lower-dimensional, say, two-dimensional data vector x'_k, then the goal of projection is to optimize the representations so that the distances between the items in the two-dimensional space would be as close to the original distances as possible. If the distance between x_k and x_l is denoted by $d(k, l)$ and the distance between x'_k and x'_l in the two-dimensional space by $d'(k, l)$, the metric MDS tries to approximate $d(k, l)$ by $d'(k, l)$. If a square-error cost is used, the objective function, called a *stress* function, to be minimized can be written as

$$E_M = \sum_{k \neq l} [d(k, l) - d'(k, l)]^2. \tag{2}$$

Function (3), called a *sstress* function, is more suitable for optimization:

$$E_M = \sum_{k \neq l} \Big[(d(k, l))^2 - (d'(k, l))^2\Big]^2. \tag{3}$$

Many authors minimize different versions of the *stress* function by means of either local techniques or their modifications. Some of these modifications are derived to achieve non-local behavior of the method. The most widely known method of this kind is SMACOF, based on the majorization of an objective function (DeLeeuw, 1977; DeLeeuw, 1988). The tunneling method is known as one of global optimization methods (Törn and Žilinskas, 1989). It was adapted to the problems of multidimensional scaling and investigated by P. Groenen (1993).

A perfect reproduction of the Euclidean distances may not always be the best possible goal, especially if the components of the data vectors are expressed on an ordinal scale. The *rank order* of the distances between the vectors is meaningful only, not the exact values. The projection should try to match the rank order of the distances in the two-dimensional output space to the rank order in the original space. The best possible rank ordering for a given configuration of points can be guaranteed by introducing a monotonically increasing function f that acts on the original distances. *Nonmetric MDS* (Kruskal, 1964; Shepard, 1962) uses such a function (Kruskal and Wish, 1978), whereby the normalized cost function becomes

$$E_N = \frac{1}{\sum_{k\neq l} [d'(k,l)]^2} \sum_{k\neq l} \left[f(d(k,l)) - d'(k,l)\right]^2. \tag{4}$$

For any given configuration of the projected points x'_k, f is always chosen to minimize (4).

Another nonlinear projection method, *Sammon's mapping* (Sammon, 1969), is closely related to the metric MDS version described above. It, also, tries to optimize the cost function:

$$E_S = \sum_{k\neq l} \frac{\left[d(k,l) - d'(k,l)\right]^2}{d(k,l)}. \tag{5}$$

The only difference between Sammon's mapping and the nonlinear metric MDS (2) is that the errors in distance preservation are normalized. Because of the normalization, the preservation of small distances will be emphasized.

The principal curves (Hastie and Stuetzle, 1989) are smooth curves that are defined by the property that each point of the curve is the average of all data points that project to it, i.e., for which that point is the closest point on the curve. Otherwise, the curves pass through the center of the data set Principal curves are generalizations of principal components, extracted using PCA, in the sense that a linear principal curve is a principal component.

7. The Self-Organizing Map Algorithm

Competitive learning is an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space (Kohonen, 1982; Kohonen, 1984; Nass and Cooper, 1975). A kind of division of labor emerges in the network when different neurons specialize to represent different types of inputs.

The specialization is enforced by a competition among the neurons: when an input x arrives, the neuron that is best able to represent it wins the competition and is allowed to learn it even better.

If there exists an ordering between the neurons, i.e., the neurons are located on a discrete lattice, a self-organizing map, a competitive learning algorithm can be generalized: if not only the winning neuron, but also its neighbors are allowed to learn, neighboring neurons will gradually specialize to represent similar inputs, and the representations will become *ordered* on the map lattice.

The neurons represent the inputs with reference vectors m_i, the components of which correspond to synaptic weights. One reference vector is associated with each neuron called a *unit* in a more abstract setting. The unit, indexed by c, whose reference vector is nearest to the input x, is the winner of the competition:

$$c = c(x) = \arg\min_i \left\{\|x - m_i\|^2\right\}. \tag{6}$$

The winning unit and its neighbors adapt to represent the input even better by modifying their reference vectors towards the current input. If the locations of units i and j on the map grid are denoted by the two-dimensional vectors r_i and r_j, respectively, then

$$h_{ij}(t) = h\left(\|r_i - r_j\|; t\right),$$

where t denotes time.

During the learning process at time t the reference vectors are changed iteratively according to the following adaptation rule:

$$m_i(t+1) = m_i(t) + h_{ci}(t)\left[x(t) - m_i(t)\right], \tag{7}$$

where $x(t)$ is the input at time t and $c = c(x(t))$ is the index of the winning unit.

8. A Sample of Using Multidimensional Scaling

Although improved local search procedures are used for some applications of multidimensional scaling, certain applications can be solved only with global optimization. An example of such applications is described in (Mathar, 1996).

A frequently used test problem for multidimensional scaling algorithms (Mathar and Žilinskas, 1994; Green *et al.*, 1989) is based on the experimental results, on testing several soft drinks (Green *et al.*, 1989). 38 students have tested ten different Cola brands. Each pair was judged on its dissimilarity on a 9-point scale (1 – very similar, 9 – completely different). The accumulated dissimilarities are the data of the problem. The scaled down accumulated dissimilarities are given in Table 4.

Table 4

Scaled accumulated dissimilarities among 10 Cola brands

	Pepsi	Coke	Classic Coke	Diet Pepsi	Diet Slice	Diet 7-Up	Dr. Pepper	Slice	7-Up	Tab
Pepsi	...									
Coke	1.27	...								
Classic Coke	1.69	1.43	...							
Diet Pepsi	2.04	2.35	2.43	...						
Diet Slice	3.09	3.18	3.26	2.85	...					
Diet 7-Up	3.20	3.22	3.27	2.88	1.55	...				
Dr. Pepper	2.86	2.56	2.58	2.59	3.12	3.06	...			
Slice	3.17	3.18	3.18	3.12	1.31	1.64	3.00	...		
7-Up	3.21	3.18	3.18	3.17	1.70	1.36	2.95	1.32	...	
Tab	2.38	2.31	2.42	1.94	2.85	2.81	2.56	2.91	2.97	...

The goal of this multidimensional scaling problem is to find the configuration of 10 objects, representing each Cola brand in a two-dimensional space, which would help to interpret the data. It is shown in (Mathar, 1996) that there are many local minima and, interpreting the data on the basis of the achieved configuration from local minima, leads to different results. So it is necessary to find the global minimum and the corresponding configuration which explains the data best.

The number of objects for this global optimization problem is $N = 10$. The configuration of objects in $m = 2$-dimensional space should be found. The number of variables is $n = 17$. The feasible region is $([0, 4]^2, [-4, 4]^{n-2})$. The objective function of the problem is (3).

The representing of the Cola problem is shown in Fig. 4.

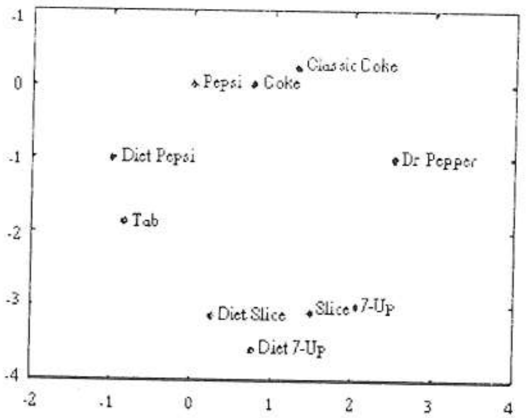

Fig. 4. The solution of the Cola problem.

9. Visualization in Education

Education can be viewed as the externally facilitated development of knowledge. This external influence can take many forms (a teacher, textbook, article, movie, TV show, computer program...). The purpose of any visualization to be used in an educational context is to facilitate the learning of some knowledge (idea, concept, fact, algorithm, relationship ...). In order to accomplish this, visualization should relate knowledge the learner has and the knowledge being taught. Therefore, in order to design effective visualizations, it is necessary to know (or at least have a theory about) what the learner knows.

The purpose of graphical display is to provide the viewer with a visual means of processing the information. It is important to note, that for a visualization to be effective it must make use of the knowledge base of the viewer. If the viewer does not possess the knowledge to understand the graphical entities and relations between them, then the visualization will not achieve its goal. For the most part visualization technoques can be classified into two categories:

- Data exploration,
- Communicating information.

Data Exploration is the practice of using visualization techniques to find unforeseen relationships between data points or sets of points in large databases. Once a relationship has been found, the same visualization can be used to communicate that relation to others. Visualization techniques can also be applied to information that is already known.

In them work (1992), Hiebert and Carpenter present a framework for discussing the representation of mathematical concepts in the context of teaching for understanding. The concepts in the framework are not specific to mathematics and can be applied to the representation of knowledge in general. Their framework also provides a useful means for discussing the role that external representations play in learning. The main points of their framework are:

- relationships exist between external and internal representations;
- the form of the external representation with which the student interacts influences how knowledge is represented internally;
- the form in which a student externally represents knowledge if influenced by their internal representation;
- internal representations of knowledge are connected to form networks of knowledge;
- networks can be hierarchical – some representations subsume others;
- networks can be graph-like – the nodes represent pieces of information and the arcs represent the relationships between them;
- most likely, a combination of the two of them;
- understanding occurs when an idea is well integrated into a richly connected network.

10. Conclusions

In this paper, we have presented an overview of major visualization techniques. Each of these techniques is based on distinct concepts and produces distinct images. No optimal visualization method can be selected. The variety of visualization methods exists because different data and different practical problems require a corresponding visualization technique. The overview is, of course, incomplete and does not include, for example, incremental MDS. Gathering of visualization methods and techniques lets us to investigate a new method or technique.

References

Ahlberg, C., and B. Shneiderman (1994). Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proc. Human Factors in Computing Systems CHI'94 Conf.*, pp. 313–317.

Andrews, D.F. (1972). Plots of high-dimensional data, *Biometrics*, **29**, 125–136.

Ankerst, M., D.A. Keim and H.-P. Kriegel (1996). Circle segments: a technique for visually exploring large multidimensional data set. In *Proc. Visualization'96*.

Asimov, D. (1985). The grand tour: a tool for viewing multidimensional data. *SIAM J. Science and Statistical Computing*, **6**, 128–143.

Baeza–Yates, R., and B. Ribeiro–Neto (1999). *Modern Information Retrieval*. Addison–Wesley, Harlow.

Beddow, J. (1990). Shape coding of multidimensional data on a microcomputer display. In *Proc. Visualization'90*, pp. 238–246.

Bertin, J. (1983). *The Semiology of Graphics*. Univ. Wisconsin Press, Madison.

Borg, I., and P. Groenen (1997). *Modern Multidimensional Scaling*. Springer–Verlag, New York.

Cleveland, W.S. (1993). *Visualizing Data*. Summit, N. J., Hobart Press.

DeFanti, T.A., M.D. Brown and B.H. McCormick (1989). Visualization Expanding Scientific and Engineering Research Opportunities. Visualization in Scientific Computing. In *IEEE Computer Society Press Los Alamitos*. California, pp. 34–36.

DeLeeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In J.R. Barm *et al.* (Eds), *Recent Developments in Statistics*. North Holland, pp. 133–145.

DeLeeuw, J. (1988). Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, **5**, 163–180.

Eick, S., and G.J. Wills (1993). Navigating large networks with hierarchies. In *Proc. Visualization'93*, pp. 204–210.

Flury, B. (1997). *A First Course in Multivariate Statistics*. Springer–Verlag, New York.

Furnas, G.W., and A. Buja (1994). Prosections views: dimensional inference through sections and projections. *J. Computational and Graphical Statistics*, **3**(4), 323–353.

Gersho, A. (1979). Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, **25**, 373–380.

Gower, J.C., and P. Legendre (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, **3**, 5–48.

Gray, R.M. (1984). Vector quantization. *IEEE ASSP Magazine*, April, 4–29.

Green, M. (1991). Visual search, visual streams, and visual architectures. *Perception and Psychophysics*, **50**, 388–403.

Green, M. (1998). Toward a perceptual science of multidimensional data visualization: Bertin and Beyond. *ERGO/GERO Human Factors Science*. http://www.ergogero.com/dataviz/dviz0.html

Green, P., F. Carmone and S. Smith (1989). *Multidimensional Scaling: Concepts and Applications*. Allyn and Bacon, Boston.

Groenen, P. (1993). *The Majorization Approach to Multidimensional Scaling*. DSWO Press.

Hastie, T., and W. Stuetzle (1989). Principal curves. *Journal of the American Statistical Association*, **84**, 502–516.

Hiebert, J., and Th. P. Carpenter . (1992). Learning and teaching with understanding. *Handbook on Mathematics Teaching and Learning*.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417–441.

Inselberg, A. (1985). The plane with parallel coordinates, special issue on computational geometry. *The Visual Compuer*, **1**, 69–97.

Inselberg, A., and B. Dimsdale (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. Visualization'90*, pp. 361–370.

Jain, A.K., and R.C. Dubes (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

Jardine, N., and R. Sibson (1971). *Mathematical Taxonomy*. Wiley, London.

Johnson, B. (1993). *Visualizing Hierarchical and Categorical Data*. PhD thesis, Dept. of Computer Science, Univ. of Maryland.

Keim, D.A. (1995). *Visual Support for Query Specification and Data Mining*. Shaker–Publishing Company, Aachen.

Keim, D.A. (1996). Pixel-oriented visualization techniques for exploring very large databases. *J. Computational and Graphical Statistics*, **5**(1), 58–77.

Keim, D.A., and H.-P. Kriegel (1994). VisDB: Database exploration using multidimensional visualization. In *IEEE Computer Graphics & Applications*, pp. 40–49.

Keim, D.A., and H.-P. Kriegel, M. Ankerst (1995). Recursive pattern: a technique for visualizing very large amounts of data. In *Proc. Visualization'95*, pp. 279–286.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59–69.

Kohonen, T. (1984). *Self-Organization and Associative Memory*, (3rd edition 1989). Springer, Berlin.

Kruskal, J.B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, **29**, 1–27.

Kruskal, J.B., and M. Wish (1978). Multidimensional scaling. *Sage University Paper series on Quantitative Applications in the Social Sciences*, number 07–011, Sage Publications, Newbury Park, CA.

LeBlanc, J., M.O. Ward and N. Wittels (1990). Exploring N-dimensional databases. In *Proc. Visualization'90*, pp. 230–239.

Levkowitz, H. (1991). Color icons: merging color and texture perception for integrated visualization of multiple parameters. In *Proc. Visualization'91*, Oct.

Lloyd, S.P. (1957). Least squares quantization in PCM. Unpublished memorandum, Bell Laboratories. Published 1982. In *IEEE Transactions on Information Theory*, **28**, 129–137.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L.M. Le Cam, J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, *Statistics*, pp. 281–297.

Makhoul, J., S. Roucos and H. Gish (1985). Vector quantization in speech coding. In *Proceedings of the IEEE*, **73**, pp. 1551–1588.

Mathar, R., and A. Žilinskas (1994). On global optimization in two-dimensional scaling. *Acta Appl. Math.*, **33**(1), 109–118.

Mathar, R.A. (1996). Hybrid global optimization algorithm for multidimensional scaling. *Classification and Knowledge Optimization. In Proceedings of the 20th Annual Conference of the Gesellschaft für Klassification e. V.*, University of Freiburg, pp. 63–71.

McCormick, B.H., T.A. DeFanti and M.D. Brown (Eds.) (1987). Visualization in scientific computing. *Computer Graphics*, **21**(6).

Nakayama, K., and G. Silverman (1986). Serial and parallel processing in visual feature conjunctions. *Nature*, **320**, 264–365.

Nass, M.M., and L.N. Cooper (1975). A theory for the development of feature detecting cells in visual cortex. *Biological Cybernetics*, **19**, 1–18.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine, Sixth Series*, **2**, 559–572.

Pickett, R.M., and G.G. Grinstein (1988). Iconographic displays for visualizing multidimensional data. In *Proc. IEEE Conf. Systems, Man, and Cybernetics*, pp. 514–519.

Ripley, B.D. (1996). *Pattern Recognition and Neutral Networks*. Cambridge University Press, Cambridge.

Robertson, G.G., J.D. Mackinlay and S.K. Card (1991). Cone trees: animated 3d visualizations of hierarchical information. In *Proc. Human Factors in Computing Systems CHI'91 Conf.*, pp. 189–194.

Sammon Jr., J.W. (1969). A nonlinear mapping for data structure analysis. In *IEEE Transactions on Computers*,

18, pp. 401–409.

Shepard, R.N. (1962). The analysis of proximities: multidimensional scaling with an unknown distance function. *Psychometrika*, **27**, 125–140; 219–246.

Shneiderman, B. (1992). Tree visualization with treemaps: a 2d space-filling approach. *ACM Trans. Graphics*, **11**(1), 92–99.

Sneath, P.H.A., and R.R. Sokal (1973). *Numerical Taxonomy*. Freeman, San Francisco, CA.

Spence, R. *et al.* (1995). Visualization for functional design. In *Proc. Intl'l Symp. Information Visualization (InfoVis '95)*, pp. 4–10.

Torgerson, W.S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, **17**, 401–419.

Törn, A., and A. Žilinskas (1989). Global optimization. *Lecture Notes in Computer Science*, **355**, Springer.

Van Wijk, J.J., and R.D. van Liere (1993). Hyperslice. In *Proc. Visualization'93*, pp. 119–125.

Wright, W. (1995) Information animation applications in the capital markets. In *Proc. Int'l Symp. Information Visualization*, pp. 19–25.

Young, G., and A.S. Householder (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, **3**, 19–22.

V. Šaltenis graduated from the Kaunas Technological Institute, Lithuania, in 1959. He received a Ph.D. degree from the Moscow Energy Institute of the USSR Academy of Sciences in 1966 and the Degree of Habil. Doctor from the Institute of Mathematics and Informatics, Vilnius in 1998. He is a principal researcher of the Optimization Department at the Institute of Mathematics and Informatics, Lithuania. His present research interests include both theory and applications of the structure of optimization problems, multicriteria decision support systems.

J. Aušraitė received an M.S. degree in informatics from Vilnius Pedagogical University, Lithuania, in 1999. Since 2001 she has been a doctoral student at the Optimization Department of the Institute of Mathematics and Informatics, and a junior lecturer at the Department of Informatics of Vilnius Pedagogical University. Her research interests include visualization of multidimensional databases, visualization technologies and methods.

Duomenų vizualizacija: idėjos, metodai, problemos

Vydūnas ŠALTENIS, Jūratė AUŠRAITĖ

Straipsnis skirtas duomenų vizualizavimui, atkreipiant dėmesį į psichologinius vaizdų suvokimo aspektus. Apžvelgiama Bertino suvokimo teorija *Image Theory*, kurioje tyrinėjama, kaip atvaizduoti didelius duomenų kiekius. Jis išskyrė dvi vaizduojamų kintamųjų klases: išdėstomi plokštumoje *(planar)* bei besiskiriantys savo išvaizda (*retinal*). Bertinas teigė, kad efektyviausias yra trijų komponenčių vizualizavimas.

Bertino teoriją praplėtė M. Greenas. Jis teigė, kad yra daugiau nei du vaizduojamų kintamųjų tipai, apie kuriuos Bertinas neužsimena ir, kad trijų komponenčių vizualizavimas yra nebūtinai efektyviausias (gali būti keturių ir daugiau). Duomenys gali būti vaizduojami tokiais kintamaisiais, kaip judesys (motion), mirgėjimas (flicker) bei binokuliarinis regėjimas (disparity).

Kalbant apie duomenų atvaizdavimą svarbu teisingai įvertinti duomenis, tam apžvelgiami duomenų tipai. Toliau klasifikuojami dažniausiai naudojami vizualizacijos būdai. Jie literatūroje skirstomos į geometrinės projekcijos (geometric projection), piktograminę (icon-based), taškinę (pixel-based), hierarchnę (hierarchical), o taip pat bandoma juos jungti.

Aprašomi keturi populiariausi daugiamačių duomenų vizualizavimo metodai: klasterių (clustering), projekcijos (projection), daugiamačių skalių (multidimensional scaling) bei save organizuojančius žemėlapius (self-organizing map). Vizualizavimo metodai ganėtinai skirtingi – tiek savo sprendimo idėjomis, tiek gautu vaizdu.