# Test Data Sets for Evaluating Data Visualization Techniques

*Daniel A. Keim*

*R. Daniel Bergeron*

*Ronald M. Pickett*

**Ludwig-Maximilians-Universität München**
**Institut für Informatik**
**Bericht 9411**

**Juni 1994**

# Test Data Sets for Evaluating Data Visualization Techniques

Daniel A. Keim
Institute for Computer Science, University of Munich,
Leopoldstr. 11B, 80802 Munich, Germany,
keim@informatik.uni-muenchen.de

R. Daniel Bergeron
Department of Computer Science,
University of New Hampshire,
Durham, NH 03824, USA, rdb@cs.unh.edu

Ronald M. Pickett
Institute for Visualization and Perception Research,
University of Massachusetts Lowell,
Lowell, MA 01854, USA, pickett@cs.uml.edu

## Abstract

In this paper we take a step toward addressing a pressing general problem in the development of data visualization systems — how to measure their effectiveness. The step we take is to define a model for specifying the generation of test data that can be employed for standardized and quantitative testing of a system's performance. These test data sets, in conjunction with appropriate testing procedures, can provide a basis for certifying the effectiveness of a visualization system and for conducting comparative studies to steer system development.

**Keywords:** Testing Data Visualizations, Generating Test Data, Visualizing Multidimensional and Multivariate Data, Perception of Visualizations

**to appear in:**
'Perceptual Issues in Visualization', Springer, 1994.

## 1 Introduction

Data visualization has captured very high interest among scientists and many commercial and public domain visualization systems have appeared in recent years including, for example, *AVS* [Ups 89], IBM's *Data Explorer*, Silicon Graphics' *Explorer*, *PV-Wave* from Precision Visuals, *IDL* from Research Systems, *Khoros* from the University of New Mexico, and *apE* from Ohio State [Dye 90]. All generally available visualization systems rely on conventional visualization techniques based primarily on two-dimensional displays, or two-dimensional surfaces in a three-dimensional world.

Considerable efforts have also been aimed at developing and prototyping non-traditional visualization techniques that attempt to present multivariate and multidimensional data in effective ways. Many different approaches have been demonstrated, and their potential value in numerous areas of application have been touted. Some examples include work by Grinstein et al. [PG 88, GPW 89, SBG 91], Beddow [Bed 90], LeBlanc et al. [LWW 90], Inselberg and Dimsdale [ID 90], Beshers and Feiner [BF 92, BF 93], Keim et al. [KKS 93, KK 94], and Mihalisin [MTS 91]. Lacking in all this activity is any quantitative evidence of just how effective the techniques are. Until we develop a basis for evaluation, we will not be able to get beyond this current demonstrational stage. To progress, we need to know with certainty what is working and what adjustments are leading to improvement.

The general purpose of a visualization system is to transform numerical data of one kind or another into pictures in which structures of interest in the data become perceptually apparent. By encoding and formatting the data into just the right kind of pictorial array, the structures, so the hope goes, will make themselves perceptually apparent. Conceivably, one might find a kind of coding and formatting that reveals many different kinds of structures in the data. But it is also conceivable that some structures might require very narrowly tuned codings and formats to become perceptible.

One of the big weaknesses of our present state of understanding is that we hardly know what we mean by *structure in the data*. We know something about, and even have a precise language for describing, some familiar and simple statistical structures. We turn to this familiar domain of structures for the test data sets proposed in this paper. But the field is in great need of a broader conception and language of structure. We need a taxonomy to inventory the world of structures that visualization systems might need to address. Creating an awareness of this lack of understanding is, indeed, one of the ancillary goals of this paper.

Visualization systems are actually just another instance of technology in science for detecting, analyzing and interpreting signals —albeit signals (what we are calling structures) of a rather broad and often ill-defined type. The need to provide a basis for quantitative evaluation of systems for signal detection and recognition is well recognized in many areas of science and technology. Evaluation of medical diagnostic systems provides a good case in point. Medical imaging systems are subject to various objective certification tests with standardized "phantom" images to verify that they can reveal the details of images that have to be resolvable for certain types of diagnoses. Even beyond such general certifications are standardized evaluations to determine how well the whole system, including the radiologist who does the reading of the images, performs in detecting and diagnosing particular diseases. In those situations, sets of test patterns (images of real cases) are assembled, and standardized tests are conducted to measure exactly how well the system performs (see [SP 82]). The need for, and approach to, evaluating visualization systems is almost exactly analogous. We want to know how well a given visualization system does in helping a scientist to detect and interpret a structure in his or her data. We need a standard set of test data sets and a standardized testing procedure. In this paper, we provide a start toward building this needed resource.

Our goal is to generate test data sets with characteristics similar to those of real data sets. Unlike real data sets, however, the characteristics of artificially generated data sets may be varied arbitrarily. We may, for example, vary the correlation coefficient of two dimensions, the mean and variance of some of the dimensions, the location, size and shape of clusters, etc. Varying the data characteristics in a controlled manner is crucial for evaluating different visualization techniques. For example, controlled test series allow us to find the point where data characteristics are perceivable for the first time, or the point where they are no longer perceivable. Also, the same test data may be used in comparing different visualization techniques, helping to determine their strengths and weaknesses.

## 2 Scientific Data

We are interested in generating test data that have characteristics similar to those of typical scientific data. Scientific data is characterized by its *data type*, the way in which it is *organized*, and the way in which the values relate to each other (the *distribution*).

### Data Types

Scientific data can consist of multiple values of various data types, which are typically described using terminology from programming languages, such as *float*, *integer*, and *string*. For our purposes we are more interested in the generic characteristics of the data types. These are best identified using terminology from the statistical domain, which defines the following standard types:

- *nominal* — data whose values have no inherent ordering
- *ordinal* — data whose values are ordered, but for which no meaningful distance metric exists
- *metric* — data which has a meaningful distance metric between any two values.

### Organization of the Data

Scientific data is often highly organized in that data values have some inherent physical or logical relationship to other data values, which might be called its *neighbors*. This organization is usually called its *data structure*. Note the distinction between the data structure (the structure *of* the data) and the patterns of values in the data that we are trying to see with a given visualization technique (the structures *in* the data). We are primarily interested in scientific data that is organized with only limited inherent structure —in particular, we consider here only data that can be represented in arrays. This restriction omits engineering-style data that is most naturally represented with more complex data structures.

The least-structured form of data is a set of records which have no particular *a priori* ordering among themselves. Conventional database records satisfy this requirement. Although there may be many fields in the records that *could* be used to order the records, there is no pre-defined ordering that is inherent in the data. Database keys which are used to uniquely identify and access database records, also do not provide a natural ordering since in most cases they only induce an artificial ordering of the records. Data sets having no inherent structure or organization can be considered to be 0-dimensional arrays.

Other data has underlying structure or organization, such that each data record has an inherent unique *position* relative to the other records. Often the record's position is related to a location in some geometric domain, or to a point in time. Such data can be generated by sampling of physical phenomena or from simulations and is commonly represented as arrays (perhaps multidimensional). A record can now be identified and accessed by its relative position in the data set which corresponds to the indices into its position in the (multidimensional) array. If this position is determined by the coordinate values of its placement in the geometric and/or time domain, these coordinate values are likely to be explicitly included in the data record. However, if the data elements are uniformly distributed over the range of indices of the array, their values can be computed from the indices of the record into the array, and need not be explicitly stored. If a data variable maps to an index into the data set's storage array, we say that that variable represents an *array dimension* of the data set.

Regardless of how the data is initially defined, the visualization may choose whether or not to place a record's visual representation on the display in a way that is consistent with the record's position in the data set. For example, consider a data set composed of carbon and nitrogen measurements on a two-dimensional x-y grid. A straightforward visualization might show the carbon value as a color or intensity at each position on the x-y grid; i.e., the x-y grid of the data is mapped to the x-y coordinates of the display. However, it might also be useful to produce a visualization in which the values of the carbon and nitrogen are mapped to the x-y coordinates of the display and the y-value of the grid is mapped to the intensity. (Note that this mapping need not produce a single-valued function: there may be multiple y-values for one pair of carbon/nitrogen values. If the visualization technique must have only a single value, some choice has to be made.)

**Examples for Typical Data Sets**

Our model of the data and the data generation process allows us to handle a wide range of types of data in a uniform way. In the following, we provide examples for typical data sets that may be generated using our model.

**Statistical Data**

We use the term statistical data to describe data sets whose data values are best defined by statistical parameters such as distribution functions, correlation coefficients, or cluster descriptions. Statistical data may have an arbitrary number of dimensions with none of them being an array dimension. The data may be scattered arbitrarily in multidimensional space and, in general, even duplicate data items are allowed. Examples of this kind of data are financial databases, product databases, personal databases, databases that record banking transaction, telephone calls or other events, and scientific databases (e.g., observations or simulations). Most of these data sets are typically stored in relational database systems.

For evaluating different visualization techniques, it is interesting to study how well different visualization techniques represent statistical patterns described by some statistical parameter. For controlled studies of this type, the statistical parameters should first be varied one at a time. After understanding their effects on the visualizations, more realistic test data sets may be built by using multiple statistical parameters to describe the test data. Examples of data sets that are best described by distribution functions include deviations of norm values which are best approximated by normal distributions, radioactivity which may be described by an exponential distribution, or periodic events which may often be assumed to follow a uniform distribution. Single dimensions with such distribution characteristics may be specified easily.

If something about the relationship between multiple dimensions is known, the data may be better described by correlation coefficients and functional dependencies. The relationship of solar radiation and temperature, for example, may be described by a high correlation coefficient and some functional dependency. Since there are usually complex relationships between multiple dimensions in real data, we also provide the ability to specify correlations between multiple parameters and complex functional dependencies. An example of a more complex relationship is the interdependencies between temperature, humidity, solar radiation, precipitation and wind speed.

Local correlations are also important features of many data sets. In a local correlation the correlation coefficient is much higher in a specific region than in the whole data set. One way of describing this kind of relationship is to specify the different partitions of the data space separately. Another way of describing complex relationships is to consider them to be multidimensional clusters in an otherwise homogeneous, possibly empty multidimensional data space. Examples of data sets that can be best described by a base data set and a set of clusters are data sets that contain a portion of data items having some clearly distinguishable properties. We may also have time series of statistical data. In most cases, the time dimension is an array dimension. This means that the cardinality of the data set is given by the considered time frame and no duplicate data items may occur.

**Image Data**

Another important class of test data is image data. Image data is two-dimensional in nature. In terms of our test data generation, normal two-dimensional image data is generated by setting the total number of dimensions to 3 and the number of array dimensions to 2. Depending on the application, however, image data may have a much higher dimensionality since multiple values for each point of the two-dimensional array may occur or different types of images for the same region may exist. In earth observation science, for example, researchers record many images at different wavelengths. To specify the test data, first the ranges for the array dimensions need to be specified. The ranges of the array dimensions determine the total number of data items. Then, the specific characteristics of the data can be specified using distributions, functional relationships, (local) correlations or cluster descriptions. Note that only the characteristics of the non-array dimensions may be specified since the array dimensions are dense and their values are given by the range definitions. In many cases, however, the distributions, functional relationships, (local) correlations or cluster descriptions include some dependency on the array dimensions. We may further have time series of image data which requires a third array dimension.

**Other Data**

Image data may be easily extended to volume data by using an additional array dimension for the third dimension of the volume. Volume data and other types of data such as geographic, geometry, molecular, fluid dynamics or flow data have specific characteristics which can only be specified by our method to a very limited extend. For molecular data, we may, for example, generate a set of atoms and some random 3D structure. However, for such molecule data to be realistic, many physical, chemical, and biological constraints apply which have to be modeled explicitly. In general, generation of arbitrary realistic test data sets would require lengthy descriptions or complex simulations reflecting all constraints and interdependencies.

At this point, we want to stress that our goal is to test and compare visualization techniques for statistical and image data. We do not intend to produce test data sets that are completely realistic for some application domain. Instead, we want the test data sets to have only a few characteristics of real data sets. Important, however, is the possibility to vary the characteristics of the test data gradually. Although real data sets are very important in testing and comparing visualization techniques, we believe that an in-depth evaluation of their strengths and weaknesses is only possible with generated test data sets whose characteristics can be precisely controlled.

## 3 Structures in the Data

In order to generate large amounts of data, we need to have an automatic mechanism for generating the data with carefully controlled statistical variations. In some cases, we want to generate the values of a particular data field without regard to other neighboring values, or values of other fields; more often we want to model actual data that has some kind of *correlation* among the various data fields.

#### Probability Distributions, Correlations and Functional Dependencies

A test generation utility needs to support the ability to specify that data generation should be driven by a variety of probability distributions, including at least the well-known distributions such as poisson, gamma, gaussian, etc. [Dev 87]. These distributions require the user to specify parameters such as the maximum, minimum, mean, and standard deviation.

More complicated (and more realistic) data generation requires that the values of different fields in the data have some functional relationship to values of other fields. In fact, the quintessential goal of scientific data visualization is to assist the scientist in determining the nature of some phenomenon by understanding the relationships present among the data values that represent that phenomenon. By generating test data containing known relationships, we hope to be able to evaluate visualization techniques to see if these relationships produce a distinctive recognizable pattern in the visual presentation of the data.

The standard measure of correlation used in today's statistics packages is the *correlation coefficient* which is defined as a measure of the linear relationship between two variables. As useful as this measure is, it does not serve to identify more complicated relationships such as non-linear dependencies and dependencies based on 3 or more variables simultaneously. Since we are generating new data, rather than analyzing existing data, we can easily generalize the notion of correlation coefficients to specify more complex interrelationships. The basic mechanism for controlling the generation of interrelated data fields is to have the user define functional dependencies among these data fields. The *functional dependencies* allow the user to specify a formula to generate a set of *initial* values for a data record, but the user can also specify that a random perturbation should be applied to these values in order to approximate real data more realistically. The randomizing function parameters are under user control.

**Data Clusters**

Our model of a visualization evaluation environment is based on the notion that the test data set should contain subsets that have data characteristics which are distinctive from the rest of the data. The visualization test then presents the data (perhaps in a variety of formats) to see whether the distinctive subset produces a distinctive and recognizable visual effect. We use the term *data cluster* to refer to a subset of data with distinctive data characteristics. The specification of a data cluster requires the specification of a region of the data space as well as the data generation parameters to be used for generating data in that region.

In its most general form, a *region* is any contiguous subset of the n-dimensional data space defined by the set of fields in the data records of the data set. In its simplest form, we can define a *rectangular* region by identifying a specific range of data values of a subset of the fields. For example, a 2-dimensional rectangular region could be defined by specifying $23 \le x \le 45$ and $102 \le y \le 150$, for the fields x and y.

A precise definition of the notion of *distinctive data characteristics* is difficult to achieve and perhaps not even desirable. What constitutes significantly different data characteristics in one domain may not be significant in another. For our purposes we simply allow a user to designate a *different* set of data generation parameters for each region.

There are two major categories of data clusters as defined by the data generation parameters — *value clusters and density clusters*. A value cluster occurs when the differentiation of data characteristics is determined by *values* of fields of the data records defined in the region. For example, the values of the temperature field inside the cluster could be defined to have a mean of 34.5 with a standard deviation of 2.3, whereas outside the region, the mean might be 46.4 with a standard deviation of 5.6. A density cluster, on the other hand, is defined when the number of data records defined in the region has a significantly different density than the number of data records defined outside the region. For example, a cluster region could be defined by a range of temperatures between 0 and 32 degrees, such that the resulting data set should have approximately 3 data records per unit temperature range inside this region, but should average only 1 data record per unit temperature outside the region.

**Formalization**

Most scientific data can be described as unordered sets of multidimensional data. For the purpose of test data generation, we therefore assume a test data set to be an unordered set of n-dimensional data vectors (or data elements). Each data element can be seen as a point in n-dimensional space being defined along dimensions $x_1, x_2, ..., x_n$.

A cluster inside such test data sets can be defined as a set of points with some common characteristics that differ from the remaining points. A cluster may also be defined as a region in n-dimensional space with each of the data points inside the region having some characteristics that are clearly distinguishable from the rest of the data set. In this case, the cluster may be defined as a connected geometric object using a subset of the data dimensions. Sometimes, there may be no sharp border between the cluster region and the remaining data set. In this case, a threshold may be used to determine whether a data item belongs to the cluster or not. The dimensions that are used in the definition of a region are called *region dimensions*. If the region is defined by m dimensions, we call it an m-dimensional cluster where $0 \leq m \leq n$.

In addition to region dimensions, we also identify the dimensions that have the property of being *dense* such as the x and y coordinates in image data or the time dimension in time series data. We call such dimensions *array dimensions*. Without loss of generality, we assume that the first k data dimensions are the array dimensions $(x_1,...,x_k)$ and the dimensions $x_{k+1}$, ..., $x_n$ are the non-array dimensions. For each of the array dimensions (i=1..k), a range $[x_i^l; x_i^h]$ is defined with the number of data values in the range being $n_i$. Note that for each value $(v_1, ..., v_k)$ in the cross product of the ranges $[x_1^l; x_1^h]$ x ... x $[x_k^l; x_k^h]$, there is exactly one data item in the data set that has $v_1$, ..., $v_k$ as the values for its first k dimensions. In other words, the first k dimensions are array dimensions if the projection of the n-dimensional data set onto the k array dimensions is bijective and the projection yields a k-dimensional rectangle covering each value inside that rectangle. In the case of using array dimensions, the number (N) of data items in the data set is given by the number of array dimensions and their ranges. It is the product of the $n_i$:

$$N = \prod_{n=1}^{k} n_i .$$

The array dimensions only contain information on the position of a data item inside the k-dimensional rectangle spanned by the ranges of the k array dimensions. By imposing an ordering on the data items and using the $n_i$ as well as their ordering as meta-information, the same information is available without storing the array dimensions as part of the data vectors. For space efficiency reasons, many formats for storing data with array dimensions (e.g., image data) use some kind of convention which allows the array dimensions to be omitted.

In testing existing data sets for array dimensions, a necessary precondition that is easy to test is to get the ranges of each possible array dimension, to multiply the corresponding $n_i$, and to compare it with the number of data items in the data set. The sufficient condition for several dimensions to be array dimensions is much harder to test. It also requires a check for duplicate combinations of values in the possible array dimensions. In cases where no array dimensions can be identified, it may be interesting to extend or reduce the data set to allow some dimensions to be array dimensions. For this purpose, additional data items may be introduced using interpolation techniques or unnecessary and redundant data items may be omitted (or averaged). In some cases, it may even be desirable to turn data items with varying intervals between values into array dimensions. This can be done by artificially introducing an array dimension according to the ordering of the data items. The same can also be done for ordinal types whose data values are ordered but have no constant interval between values.

For visualization purposes, often a subset of the array dimensions is mapped to the dimensions of the visualization. Image data (#ArrayDimensions $\geq 2$), for example, is usually mapped to the two dimensions of the display; time series of image data (#ArrayDimensions $\geq 3$) are usually mapped to the two dimensions of the display plus time; time series of three-dimensional geometric data (#ArrayDimensions $\geq 4$) are usually mapped to three display axes plus time, and so on. In these examples, the mappings are natural, but there are many other mappings possible, especially if $k \gg 4$ or $n \gg k$, which means that there are many more array dimensions than the three dimensions of the display plus time or that there are many non-array dimensions which are difficult to visualize if only the array dimensions are mapped to the three dimensions of the display plus time. For low array dimensionality ($k < 4$) or no array dimensions ($k = 0$), the task of visualizing the data is to find some meaningful mapping from non- array dimensions to the dimensions of the display plus time (which are basically all metric array-like dimensions in the visualization domain).

## 4 Test Data Generation

In generating multidimensional test data sets, it is important to distinguish data sets according to the number of array dimensions, the number of clusters, and the method used for describing them (data, value cluster or density cluster regions). All three aspects are important not only for determining the data generation parameters but also for the data generation process itself, especially for the constraints that apply in generating the data.

**Constraints**

Constraints in generating the data are especially important if one or more array dimensions are involved. One constraint is that the number of data items is given by the number and ranges of the array dimensions. Also, the number of data items for each data value in one array dimension is given as the product of the $n_i$ of the remaining array dimensions. Similar constraints apply to any combination of the array dimensions. The constraints may also be expressed in terms of uniqueness and coverage of the value combinations for all array dimensions. The easiest way to fulfill these constraints is to generate the test data in an ordered fashion covering the allowed ranges for all array dimensions uniquely. An independent generation of the array dimensions would require checking the constraints for each generated data item which is computationally intensive. Still, in some cases it may be necessary to check some constraints. For example, if multiple region clusters are defined using array and non-array dimensions, then conflicts between the cluster definition and the constraints introduced by the array dimensions may occur.

**Data generation parameters**

Independently from the method used to describe the clusters, several data generation parameters are needed. Among the basic data generation parameters, there are the overall number of dimensions (n), the number of array dimensions (k) and their ranges, the number of clusters, and, in case k = 0, the number of data items. In order to generate test data, we need at least some more information about the non-array dimensions, namely their distribution function (uniform, normal, gaussian, ...) in case it is an independent dimension, or the correlation coefficient or functional dependency in case it is a dependent dimension. Array dimensions are considered independent dimensions which allows them to be used in defining the dependent ones. The different distribution functions are defined by specifying the necessary parameters: lower and upper limit for the uniform distribution, mean and standard deviation for the gaussian distribution, rho and lambda for the gamma distribution, and so on. Functional dependencies may be defined by an arbitrary function plus a randomness factor which is used to perturb the results of the functional dependency.

**Cluster regions**

A different way of describing the characteristics of the test data set is to explicitly define the cluster regions and their properties. Depending on the kind of clustering used, we distinguish between value cluster and density cluster regions. *Value cluster regions* are defined by identifying the region dimensions, defining the geometric shape of the region, the number or percentage of data items in the region, and the distribution function, correlation coefficient or functional dependency plus randomness factor for each region dimension. In our test data generation, regions are m-dimensional rectangles in n-dimensional space. This allows the regions to be defined by specifying some range for each region dimension. *Density cluster regions* are defined by identifying the region dimensions, defining the geometric shape of the region, and the density of elements in the region. The actual number of data items that are in each region and outside all regions is determined relative to each other.

The data items belonging to non-overlapping regions can be generated independently from each other. Regions that partially overlap with other regions require special consideration. The specified data characteristics for overlapping regions may be conflicting and may not be satisfiable by any data set. In order to interpret overlapping region specifications unambiguously, the order of defining the regions determines a priority ordering for the regions. The regions that are defined first have the highest priority. In case of cluster density regions, the regions which have the highest priority are filled with data items according to the desired density. For subsequent cluster density regions, only the non-overlapping part of the region is filled with data items according to the desired density.

We define the *base region* as the region in multidimensional space that includes all other regions. Assume, that the range of each region for dimension i is given by $[l_i, h_i]$. Then, the base region includes at least the multidimensional space defined by

$$[\min\{l_1\}, \max\{h_1\}] \ \times \ \ldots \ \times \ [\min\{l_n\}, \max\{h_n\}].$$

If some dimension is not used in any region definition, the range for that dimension is arbitrary. Note that in general, the base region is sparse since the number of data items may be low compared to its volume — it may even be empty.

Clusters that are defined using distribution functions such as normal or gaussian distributions provide smooth transitions into the region. Other cluster definitions, including density clusters may result in rather sharp transitions into the region. Such transitions may need to be smoothed to resemble real data. Defining smooth transitions into regions is not always straightforward, especially in the case of overlapping regions. We do not address this issue at the present time.

## 5 Examples

Tools that partially implement the described test generation facilities have been implemented at the University of Munich and the University of New Hampshire at Durham. The tool developed at the University of New Hampshire is described in a related paper in this volume [WB 94]. It is primarily oriented towards generating what we identified as *image data*. The tool developed at Munich focuses on the generation of *statistical data* as described in section 2. In the case of statistical data, the number of array dimensions is assumed to be zero. Different kinds of relationships may be defined between different dimensions in each of the clusters and the base region. Figure 1 shows visualizations produced by the VisDB system [KKS 93] using generated test data. The data of the four dimensions is generated such that only the first of the four dimensions is independent; the other three dimensions are functional dependant on dimension one. Dimension two is linear, dimension three quadratic, and dimension four is cubic dependant on dimension one. The generated data set consists of 6000 data items and the distribution of values for the independent dimension is uniform in the range [0, 100]. The data used to produce figure 1a has a randomness factor of zero which is increased to 0.5 in figure 1b and to 1.0 in figure 1c. Despite the linear functional dependency between dimension one and two, the corresponding visualizations in figure 1a are identical. This is due to the normalization and mapping of the different value ranges to a fixed color range. The main difference between dimension one and the dimensions with a higher order functional dependency is that the region of light coloring is larger. This is due to the unequal distribution of values in the extended value ranges of dimensions two and three. The increasing randomness factor results in some distortion of the visualization which also induces minor distortions in the visualization for dimension one. This results from a different ordering of data items

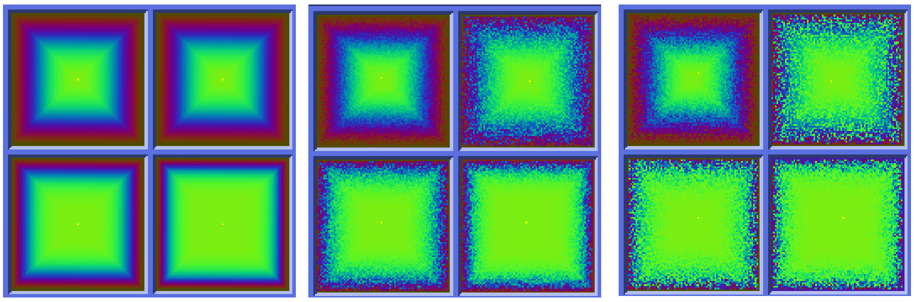
a. Randomness Factor = 0.0 b. Randomness Factor = 0.5 c. Randomness Factor = 1.0

**Figure 1:** Visualizations from Test Data with Functional Dependencies

which is caused by data items that have a high deviation from the functional dependency. More visualizations produced by the VisDB system using generated test data with different base region and cluster sizes can be found in a related paper of this volume [KK 94].

## 6 Conclusions and Future Work

In this paper we have described a model for test data generation that can be used to evaluate visualization techniques. The data sets are constructed from specifications that identify clusters of data that have different characteristics. Users can define clusters based on the density of data in the region or based on the values of the data. Statistical distributions, correlations, and functional dependencies can be used to determine the characteristics of the data in each region. Aspects of our model have been incorporated into two different systems for generating test data.

Our intent in defining our test data generation model is to begin to develop tools that can be used to provide support for rigorous evaluation of visualization techniques — especially those that present multivariate and/or multidimensional data. Our work is just a small step in this direction. The kinds of data sets that we can generate do not necessarily represent any particular kind of 'real data'. There are many other kinds of distributions that may be needed in order to provide truly meaningful tests for a particular domain. It would be nice provide arbitrarily shaped regions, to develop rigorous definitions of alternative interpretations of how to handle overlapping regions and to define smooth transitions across region boundaries. Finally, the most difficult work is the development a complete methodology for evaluating the effectiveness of visualization techniques.

**Acknowledgments**

This research has been supported in part by the National Science Foundation under grant IRI-9117153.

**References**

[Bed 90] Beddow J.: *'Shape Coding of Multidimensional Data on a Microcomputer Display'*, Visualization '90, San Francisco, CA., 1990, pp. 238-246.

[BF 92] Beshers C., Feiner S.: *'Automated Design of Virtual Worlds for Visualizing Multivariate Relations'*, Visualization '92, Boston, Mass., pp. 283-290.

[BF 93] Beshers C., Feiner S.: *'AutoVisual: Rule-based Design of Interactive Multivariate Visualizations'*, Computer Graphics & Applications, Vol. 13, No. 4, 1993, pp. 41-49.

[Dev 87] Devore J. L.: *'Probability and Statistics for Engineering and the Sciences'*, Brooks/Cole, Monterey, California, 1987.

[Dye 90] Dyer D. S.: *'A Dataflow Kit for Visualization'*, Computer Graphics & Applications, Vol. 10, No. 4, 1990, pp. 60-69.

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

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

[KKS 93] Keim D. A., Kriegel H.-P., Seidl T.: *'Visual Feedback in Querying Large Databases',* Visualization '93, San Jose, CA., 1993, pp. 158-165.

[KK 94] Keim D. A., Kriegel H.-P.: *'Possibilities and Limits in Visualizing Large Amounts of Multidimensional Data',* in: Perceptual Issues in Visualization, Springer, Berlin, 1994.

[LWW 90] LeBlanc J., Ward M. O., Wittels N.: *'Exploring N-Dimensional Databases'*, Visualization '90, San Francisco, CA, 1990, pp. 230-237.

[MTS 91] Mihalisin T., Timlin J., Schwegler J.: *'Visualizing Multivariate Functions, Data and Distributions'*, Computer Graphics & Applications, Vol. 11, No. 3, 1991, pp. 28-35.

[PG 88] Pickett R. M., Grinstein G. G.: *'Iconographic Displays for Visualizing Multidimensional Data'*, Proc. IEEE Conf. on Systems, Man and Cybernetics, Beijing and Shenyang, China, 1988.

[SGB 91] Smith S., Grinstein G. G., Bergeron R. D.: '*Interactive Data Exploration with a Supercomputer'*, Visualization '91, San Diego, CA, 1991, pp. 248-254.

[SP 82] Swets J. A., Pickett R. M.: *'Evaluation of Diagnostic Systems: Methods from Signal Detection Theory'*, Academic Press, New York, 1982.

[Ups 89] Upson C., et al.: *'The Application Visualization System: A Computational Environment for Scientific Visualization'*, Computer Graphics & Applications, Vol. 9, No. 4, 1989, pp. 30-42.

[WB 94] Wong P. C., Bergeron R. D.: *'A Multidimensional Multivariate Image Evaluation Tool'*, in: Perceptual Issues in Visualization, Springer, Berlin, 1994.