

Visualizing structure in high-dimensional multivariate data

by F. W. Young
P. Rheingans

We present and discuss several dynamic statistical graphics tools designed to help the data analyst visually discover and formulate hypotheses about the structure of multivariate data. All tools are based on the notion of the "data space," a representation of multivariate data as a high-dimensional (hD) space which has a dimension for each variable (column of the data) and a point for each case (row of the data). The data space is projected orthogonally onto the "visual space," a three-dimensional space which is seen and manipulated by the data analyst. The visual space has a point-like object for each case and can have a vector-like object for each variable. The three dimensions of the visual space are orthogonal linear combinations of the variables. We discuss the notion of a "guided tour" of multivariate data space, and present guided-tour tools, including 1) 6D-rotation, a tool for dynamically rotating, in six-dimensional (6D) space, from one 3D portion of the data space to another while displaying the dynamically changing projection in the visual space; 2) hD-residualization, a tool that determines, at the user's request, the largest invisible 3D space—i.e., the largest 3D space is orthogonal to the visual space. This space is used with the visual space so that 6D-rotation can occur between two new 3D portions of the

data space; 3) projection-cueing, a group of three tools that use change in object brightness as a cue to show change in aspects of the projection of objects from the data space to the visual space during hD-rotation. In addition to these tools for touring high-dimensional multivariate space, we discuss tools for manipulating the 3D visual space, and a tool for examining the relationship between two data spaces. Finally, we present a guided-tour implementation in which the user manipulates joysticks and sliders to dynamically and smoothly control the graphics tools in real time. A video supplement demonstrates the implementation.

Introduction

Statistical graphics can be powerful data analysis tools for exploring scientific data for structure—powerful because they help the scientific explorer visualize structure. Dynamic statistical graphics—graphic methods in which the user interacts with a computer to create smoothly moving pictures of the data—can be especially powerful tools for exploring for structure when the data are more than two-dimensional. Again, the power of these methods stems from their ability to help a scientist visualize the structure of data, even when the structure may exist in more than three dimensions. Since the early stages of scientific inquiry involve exploration, and since scientific

©Copyright 1991 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of this paper must be obtained from the Editor.

exploration leads to scientific hypotheses, graphical methods are central to the process of gaining scientific insight.

In this paper we discuss dynamic statistical graphics. In particular, we discuss a set of dynamic statistical graphics tools for exploring and visualizing structure in high-dimensional multivariate data. These tools are for "looking at data to see what it seems to say," to quote John Tukey, the founder of the branch of statistics [1] which focuses on, and is called, exploratory data analysis.

In the first section of this paper we present a number of considerations in designing dynamic statistical graphics tools for analyzing high-dimensional data. This discussion reflects the light shed by Hurley and Buja [2] in their paper describing *guided tours*, methods for visualizing high-dimensional data that are based on real-time dynamic graphics which the user guides through high-interaction, immediate-feedback actions.

In the second section of the paper we describe the conceptual and mathematical aspects of a set of guided-tour tools for exploring and visualizing high-dimensional data. These tools, some of which are presented here for the first time, implement rotation in up to six dimensions, provide the ability to perform a self-guided tour of high-dimensional data space, and provide ways to visualize the distance of a projection from high-dimensional to three-dimensional space. We also discuss tools for manipulating 3D space, and a tool for comparing two high-dimensional spaces. These guided-tour tools are designed to enable the data analyst to explore and visualize structure in high-dimensional space.

In the third section we discuss the software and hardware involved in an implementation of the guided-tour tools presented in the previous section. We call our system VISUALS/Pxpl, a new implementation on the Pixel-Planes computer [3, 4] of the VISUALS software reported earlier¹ [5-7]. Pixel-Planes is a special-purpose, one-of-a-kind, massively parallel graphics computer especially designed to optimize operations in 3D space.

In the fourth and final section we present an example and discuss a video of VISUALS/Pxpl being used to explore data concerning the rates of seven types of crime in the fifty United States. The guided-tour tools are demonstrated in the video. These data have been explored using the original VISUALS system, as reported by Young, Kent, and Kuhfeld [6] and by Young [7].

1. Guided tours

Hurley and Buja [2] define a *guided tour* as a way of exploring and visualizing multivariate data. The work

reported here is an example of a guided tour. Indeed, our work is very similar to earlier work reported by Young and his co-workers [5-7]. However, that earlier work differs from Hurley and Buja's definition of a guided tour in one important way, whereas the current work does not. The difference is discussed below.

• Data space

To define a guided tour, we begin by defining the multivariate data that are to be explored and visualized. Suppose that the multivariate data consist of h numerical variables observed on each of n cases. Suppose further that these data are collected together into the matrix $\mathbf{X}$, an $n \times h$ matrix of data with elements x_{ia} . This matrix has n rows, one for each of the n cases, and h columns, one for each of the h variables.

In order to understand the idea of a guided tour, we introduce the notion of a *data space*. A data space is an abstract view of the data. In the data space each case of the data is represented by an h -dimensional observation vector $\mathbf{x}_i$, whose a th element is the observation on variable a . Thus, abstractly, the entire set of data is represented by n points in an h -dimensional data space. The rows of the data matrix contain coordinates of the points in this space; the columns are the dimensions of the space. The canonical basis vectors of the data space $\mathbb{R}^h$ are denoted by $\mathbf{e}_a$, $a = 1, \dots, h$. They are in one-to-one correspondence with the observed variables. Without loss of generality, we assume that $\mathbf{X}$ is "column-centered," i.e., that the mean of each column is 0. In the abstract high-dimensional (h D) data space, this implies that the centroid of the space is at the origin.

• Visual space

An important aspect of a guided tour of data space is that the tour is visual: The purpose of the guided tour is to help the data analyst visualize the high-dimensional structure of the data space. Thus, a central part of the guided tour is the *visual space*: a 3D picture of the data formed by orthogonally projecting the data space $\mathbb{R}^h$ onto $\mathbb{R}^3$. The projection is orthogonal with respect to the canonical inner product in $\mathbb{R}^h$. Such orthogonal projections enable us to form 3D pictures which have mutually perpendicular x , y , and z axes. Numerically, the visual space is represented by the matrix $\mathbf{V}_p$, an $n \times 3$ matrix of data with elements v_{iap} . This matrix has n rows, one for each of the n cases, and 3 columns, one for each of the h variables. The visual space, and its matrix representation, involve dynamically varying projections, thus the subscript p .

The visual space contains points, one point for each case as it is projected from the high-dimensional data space into the visual space. The visual space may also contain vectors, one vector for each variable as it is

¹ VISUALS is also implemented on IBM and Macintosh microcomputers, and under X-Windows on UNIX workstations. These implementations include all features discussed in this paper except high-dimensional projection-cueing. Information on the availability of these implementations is available from the first author. Macintosh is a trademark of Apple Computer, Inc. The X-Window System is a trademark of MIT. UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

projected from the data space. (A vector has zero length when its variable is orthogonal to the 3D space.) If the plot contains only case points, the visual space is a 3D scatterplot. If it also contains variable vectors, it is a 3D biplot. Note that we specifically use orthogonal projections that do not imply that the variables are unit length, rather than orthonormal projections which do carry this implication. If we wish, all variables can be normalized to unit length.

Note that our definition of the visual space V_p differs from the corresponding notion in Hurley and Buja's guided tour in that their work deals with a sequence of orthonormal projections that are one- or two-dimensional. That is, their visualization is in $\mathbb{R}^1$ or $\mathbb{R}^2$, not in $\mathbb{R}^3$. Also, their work only considers projecting the cases as points in the visual space, and does not consider projecting the variables as vectors in the visual space. However, while our definition can be seen as a generalization of theirs, we do not view the generalization as fundamental.

• *Dynamic graphics*

As did Hurley and Buja, we restrict our consideration to moving plots produced by displaying a sequence of frames in which every frame is a different projection of the data space onto the visual space. Several frames are computed and shown every second. We consider only *dynamic* movement, movement which is smooth in real time and which is controlled by a data analyst through graphic, high-interaction, immediately effective actions. Here, the computer creates only one frame in the sequence before interrogating the analyst to see how the next frame should be produced, with the creation-interrogation cycle occurring several times per second. Dynamic movement is in contrast to *animated* movement, in which the computer creates a series of frames and then presents them in sequence to the viewer who passively views the "movie."

The dynamic plots consist of a sequence of projections displayed in rapid succession. We denote any one of these visualizations as V_p , the visualization based on projection p of the data space into the visual space. The projection is one of the series $V_1, V_2, \dots, V_{p-1}, V_p, V_{p+1}, \dots$, where each V_p is in $\mathbb{R}^3$.

• *Purpose*

A guided tour capitalizes on the pattern-recognition power of human vision and the computational power of graphics workstations to help data analysts look for structure (form hypotheses) in their high-dimensional multivariate data. The purpose is to aid in forming hypotheses about the high-dimensional geometric structure of the data, even though we can only see in three dimensions. To do this, a guided tour must

- Respect the data's high-dimensional geometry.
- Respect the data analyst's three-dimensional perception.
- Respect the workstation's computational limits.

Of course, while we can see in three dimensions, we can draw in only two dimensions on the computer screen. Thus, a guided tour must present high-dimensional information in two dimensions, such that our three-dimensional perception can understand the high-dimensional geometry. In order to do this, the sequence of projections should meet requirements that were emphasized by Hurley and Buja:

- The movement should be smooth, so that we can observe smooth movement of points and vectors in the visual space. This means that projections in the sequence should be "close enough" so that the movement of points and vectors from one to the next is small.
- Since the purpose of a guided tour is to help the data analyst visually explore data space for structure, the sequence of projections and the corresponding sequence of visual spaces should be generated under the control of the data analyst. Furthermore, the data analyst should control the sequence via highly interactive, immediately effective actions.
- The computation of the sequence of projections and visual spaces should be in real time. In particular, the projections in the sequence should be "rapid enough" so that the movement appears to be continuous.

• *Target spaces*

The central problem in designing a method for visually touring data space is how to construct the sequence of projections and their corresponding visual spaces. As has been discussed by Young, Kent, and Kuhfeld [6] and by Hurley and Buja [2], it is much simpler for the implementor of a visual data-space tour to construct the sequence of projections and visual spaces without regard to the data analyst, and to present them passively to the user. In fact, such a technique has been proposed by Asimov [8] and by Buja and Asimov [9]. However, this technique, which Asimov named the "grand tour," does not actively involve the data analyst, so it would seem to be less likely that the data analyst would find structure of interest.

Thus, the developer of a truly interactive "guided tour," as opposed to the noninteractive "grand tour," is faced with the problem of how to place the construction of the sequence of projections under the control of the data analyst, and how to do this in a way which is both fast and simple to use. Solutions to these problems have been provided by Young and his co-workers, and by Buja

and his co-workers. These two groups of investigators propose to provide the data analyst with tools for constructing a series of "target spaces," and with additional tools for smoothly interpolating between the target spaces. For both groups of researchers, the guided tour consists of the sequence of spaces produced by interpolating between successive target spaces.

Thus, the problem of how to construct the sequence of projections reduces to two more fundamental problems: First, what tools do we provide the data analyst to construct target spaces? Second, what tools do we provide the analyst to interpolate between the targets? In the next section we discuss these guided-tour tools.

2. Guided-tour tools

In this section we present a specific set of guided-tour tools. The tools include one for constructing target spaces, one for interpolating between target spaces, and a group of tools that use object brightness to represent information about the projection of an object from data to visual space. In addition, we present a group of tools for manipulating 3D visual space, and a tool that can be used to understand the relationship between two data spaces.

• Data

Before presenting and defining the guided-tour tools, we need to complete the definition of the data to be studied with the tools. In the previous section we defined the basic multivariate data as X , an $n \times h$ matrix with a row for each of the n cases and a column for each of the h variables. These data are assumed to be column-centered. We indicated in that section that we take the abstract view that the data are n points in an h -dimensional space whose centroid is at the origin.

The variables in the data correspond to the dimensions of the data space. We may represent the variables (dimensions) in the visual space by axes that extend between $\pm l_a$, $a = 1, \dots, h$. These axes necessarily run through the origin and centroid of the space. We define l_a as the standard deviation of the coordinates on the dimension. The standard deviation is proportional to the length of the axes, since the length of an axis is the square root of the sum of the squared coordinates, whereas the standard deviation is the square root of the mean of the squared coordinates. Because of the centering, the standard deviation is the average of the distances of points from the origin of the data space when the points are orthogonally projected onto the dimension.

We augment the data matrix X by vertically concatenating it with an $h \times h$ diagonal matrix L whose diagonal elements are l_a . This means that X is now an $(n + h) \times h$ matrix containing the multivariate data in the first n rows, and the standard deviations l_a of the h

axes on the diagonals of the last h rows. If we wish, we can further augment X with additional rows whose values represent the coordinates of supplemental points or variables. If there are s such supplemental rows or coordinates, X becomes an $(n + h + s) \times h$ matrix of coordinates.

Optionally, the dimensions of the data X may be "normalized": i.e., made to all have the same length (due to the centering, if they have the same length they will also have the same standard deviation and same variance). This is done by dividing each column of X (including augmented and supplemental values) by its length l_a by the equation $X =: XL^{-1}$, where all $(n + h + s)$ rows of X are included in the normalization. Note that for the augmented (but not supplemental) rows, the normalization process changes the nonzero coordinates to one.

• Initial spaces

Now that the data matrix X is completely defined, we define the initial visual space V_0 and the initial target space T_0 . The definition of the initial visual space is, simply, that V_0 is an $(n + h + s) \times 3$ matrix whose three columns equal three of the columns of X . The definition of the initial target space is equally simple: $T_0 = V_0$. The subscripts on the visual and target space matrices indicate that they vary, with the initial matrices indicated by 0. Note that the subscripts are different for the two matrices. For the visual space we use p to indicate that the visual space presents varying projections from the data space. For the target space we use t to indicate that the target changes over time.

• Visual representation

The rows of the multivariate data are represented by points in the data space, and are represented by "point-like" objects in the visual space. These objects could be spheres, cubes, 3D crosses, etc. The variables of the data are represented in the data space as dimensions. Thus, in the visual space they are shown as "axis-like" objects. Of course, we can think of planes in the data space (such as the plane formed by a pair of variables). Such a plane could be represented in the visual space by a "plane-like" object such as a grid. Since the h augmented rows of X represent the dimensions of the data space, their visual "objects" are lines drawn between $\pm l_a$. Finally, the supplemental rows of X may represent either cases or variables; thus, supplemental cases are represented in the visual space by point-like objects, whereas supplemental variables are represented by vector-like objects (lines from the origin).

• hD-residualization tool

This tool calculates T_{t+1} and T_{t+2} , the next two target spaces in the sequence of targets. This tool enables the

data explorer to create many alternative 3D views of the data space, these views being used as targets by the 6D-rotation tool discussed below. This tool was developed and discussed by Young [7] and his co-workers [5, 6].

The hD-residualization tool calculates the largest 3D space that is orthogonal to the visible space V_p , the largest *invisible* space. This space is "largest" in the sense that it contains the three longest mutually orthogonal dimensions which are also orthogonal to the visible space. It is also largest in the sense that it is the maximum-variance 3D space orthogonal to the visible space. This tool is called hD-residualization because it computes the largest "residual" space in the invisible portion of the high-dimensional data space.

The hD-residualization equations are based only on the n coordinates of the cases, not on the h coordinates of the variables nor on the s supplemental coordinates. The data space X (excluding the lower $h + s$ rows) is related to the visible space V_p by the equation (we omit the subscript on V_p for simplicity and because these equations hold for all values of p)

$$X = VB + R,$$

where R is an $(n \times h)$ matrix of residual information between the two spaces, and B is a $(3 \times h)$ matrix of coefficients of three orthogonal linear combinations of the h variables, determined by the equation

$$B = V^+X,$$

where $V^+ = (V'V)^{-1}V'$. Then

$$R = X - VV^+X$$

can be decomposed into

$$R = PQS'$$

using a singular value decomposition. We then define the T_{i+1} space as the old interpolation space V and the T_{i+2} space as the first three columns of PQ . Notice that residualization does not change the data X .

• 6D-rotation tool

This tool is used by the data explorer to rotate a 3D projection of the high-dimensional cloud of points back and forth between the two targets through a 6D portion of the data space. The user watches the dynamically changing projection of the cloud into the visual space, in order to understand the cloud's 6D structure. Our tool extends a 4D version of this tool developed and presented by Buja et al. [10].

This tool uses a trigonometric interpolation which Buja et al. have shown to be an orthonormal rotation in the six-dimensional space spanned by the two target spaces. The rotation follows the shortest geodesic path in 6-space. The equation is

$$V_p = T_{i+1}(\cos[U_p]) + T_{i+2}(\sin[U_p]),$$

where V_p is the $(n + h + s) \times 3$ matrix of coordinates v_{ia} of the objects seen in visual space, where the functions $\cos$ and $\sin$ are the cosine and sine functions applied to the diagonal of U_p , and where U_p is a diagonal 3×3 matrix with diagonal values

$$0^\circ \leq u_{paa} \leq 90^\circ,$$

where the values u_{paa} increment from 0° to 90° dynamically over p , the increment being 5° .

• hD-depth cueing tools

The three tools in this set of tools use object brightness to visually represent information about the projection "depth" of the object from data space to visual space. One of the hD-depth cueing tools uses brightness to represent distance information, another tool uses brightness to represent angular information, and the third tool uses brightness to represent fit information. These three tools are introduced for the first time in this paper.

Projection distance cueing

The definition of this hD-depth cueing tool depends on the fact that an orthogonal projection of a point i in data space onto the 3D visual space forms a right triangle, as portrayed in Figure 1. The hypotenuse of the triangle is denoted h_i , which is the distance in data space between the origin and the location of point i in data space. The sides of the triangle are denoted p_i , the distance in data space between the location of point i in data space and the nearest surface of visual space, and d_i , the distance in visual space between the origin and the projection of point i into visual space. The sides p_i and d_i form a right angle (as indicated in the figure), because of the orthogonality of the projection. Therefore,

$$h_i^2 = p_i^2 + d_i^2,$$

and it follows by substitution that

$$p_i^2 = \sum_a^h (x_{ia}^2) - \sum_{ia}^3 (v_{ia}^2).$$

We normalize the distance p_i so that it is always between 0 and 1 by dividing it by the maximum of the distances of the $n + h + s$ objects from the origin, say object m . We then represent the distance over which the object has been projected from data space to the visual space by the brightness of the object in the visual space. If the projection is very long (the data object is far from the visual space), the object is shown dimly. If the projection is very short (the data object is very close to the visual space), the object is shown brightly. Specifically, we define the brightness cueing value for projection distance to be

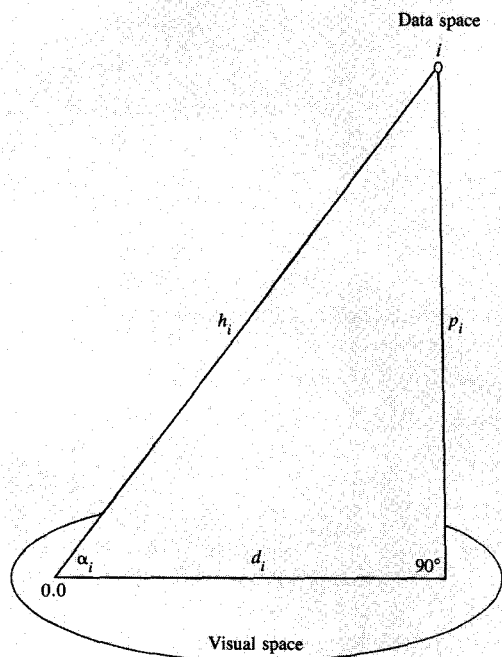


Figure 1

Orthogonal projection of point i in data space onto 3D visual space.

$$b_i = 1 - [p_i/d_m]$$

for object i .

Projection angle cueing

This hD-depth cueing tool uses brightness to visually cue the cosine of the angle between the line from the origin to the location of the object in data space and the line from the origin to the location of the object in visual space. With this definition of hD-depth cueing, the value for object i is defined as

$$b_i = \cos [\alpha_i].$$

It can be shown that this is equivalent to

$$b_i = 1 - [\sum_a^3 (v_{ia}^2) - \sum_a^h (x_{ia}^2)] / \sum_a^3 (v_{ia}^2).$$

When the cosine is 1, the object is bright. In this case the angle is 0, and the distance h_i between the object in data and visual space is 0, implying that the location of the object in visual space coincides with the location of the object in data space. If the angle is very small, the cosine is nearly 1, and the two lines are nearly colinear. This implies that the visual space very nearly contains the

object before it is projected from data space. Thus, the location of the object in visual space adequately represents the location of the object in data space. If the angle is large, the cosine is nearly 0 and the hypotenuse h_i is very long, indicating that the location of the object in visual space does not adequately represent the location of the object in data space. Here, the object is very dim.

Projection fit cueing

This tool uses brightness to visually depth cue the proportion of the total variance of the case which is represented in the visual space. The value of this hD-depth cueing tool is defined as

$$b_i = [\sum_a^3 (v_{ia} - v_i)^2 / 3] / [\sum_a^h (x_{ia} - x_i)^2 / h].$$

(Note that a subscript "dot" on v_i and x_i indicates the mean for row i . Also note that h with no subscript is the dimensionality of the high-dimensional space, not the distance h_i of point i from the origin.) If the proportion is 1, all of the variance of the case is represented in the visual space, and the object is very bright. If the proportion is 0, none of the variance of the case is represented in the visual space, and the object is very dim. The brightness varies linearly with the proportion.

• 3D tools

Since our definition of a guided tour of high-dimensional space is in terms of projections into a visual space which is three-dimensional, the data explorer needs to have a collection of tools for manipulating 3D space. We discuss here, briefly, a standard collection of such tools. We do not, however, define these tools mathematically, as their definition and development have been presented elsewhere. In fact, the tools presented in this section are available in a number of commercially available data analysis systems.

The visual data analyst must have tools to *spin* (rotate) and *move* (translate) the visual space. Ideally, it should be possible to combine spins and moves on each of the three axes of the space. The analyst should also be able to *rock* the spin and move motions to increase the depth illusion. A number of additional tools have been proposed to enhance the explorer's understanding of the data cloud's structure. These include *brushing*, the ability to move a rectangular "brush" across the screen to select and manipulate subsets of points inside the brush; *metamorphing*, which is changing the size, shape, or color of the object that represents the observation point in the 3D space; and *subsetting*, the creation of subsets of objects by their location in 3D space; by their color, size, or shape; by the value of an attached label; or by their observation number.

In addition to the (now) standard set of 3D tools, the implementation we discuss in the next section contains

two less-common 3D tools. These tools, which are designed to enhance the 3D effect, project the cloud of observation points in *perspective* onto the 2D graphics screen, and (optionally) in *stereo* perspective.

- *6D-interpolation tool*

In addition to the tools for exploring and visualizing one data space, the implementation discussed below has a tool for visually comparing two high-dimensional data spaces. This tool enables the data explorer to smoothly interpolate between two 3D portions of two hD data spaces. The user watches the dynamically changing interpolation in order to understand the relationship between the two hD spaces. This tool was developed and discussed by Young and his co-workers [5-7]. It is very similar to the 4D-rotation tool developed by Buja and his co-workers, and was in fact presented by Young to perform the functions performed by hD-rotation. However, it has been shown by Buja et al. [10] that when viewed as a rotation, the 6D-interpolation tool does not yield an orthogonal rotation; rather, it yields a sheared, nonorthogonal rotation. They go on to point out, however, that the tool is useful for comparing two 3D spaces, or two 3D portions of two separate hD data spaces.

The 6D-interpolation tool for dynamically moving from T_{i+1} to T_{i+2} is defined as

$$V_p = T_{i+1}(I - C_p) + T_{i+2}C_p,$$

where V_p is defined as above, and where C_p is a diagonal 3×3 matrix with diagonal values

$$0 \leq c_{paa} \leq 1,$$

where the values c_{paa} increment from 0 to 1 dynamically over p in increments of 0.05.

3. VISUALS/Pxpl implementation²

The tools for guiding a tour of high-dimensional data space that are defined above are implemented in a system we call VISUALS/Pxpl. We have chosen this name because many of the fundamental touring tool concepts were defined and implemented by Young [7] and his co-workers [5, 6] in a system they called VISUALS. The "Pxpl" suffix reflects that the software has been reimplemented on a special-purpose, massively parallel, custom graphics computer called Pixel-Planes. This computer was developed by Fuchs and his co-workers [3, 4].

Pixel-Planes is a raster graphics system for high-speed rendering of 3D objects and scenes. It features a "frame buffer" composed of custom logic-enhanced memory chips that can be programmed to perform most of the

time-consuming pixel-oriented tasks in parallel at each pixel. The novel feature of this approach is a unified mathematical formulation for these tasks and an efficient tree-structured computation unit that calculates inside each chip the proper values for every pixel in parallel.

The current system, Pixel-Planes 4 (Pxpl4), contains 512×512 pixels $\times$ 72 bits per pixel, implemented with 2048 custom 3- μ m nMOS chips (63 000 transistors in each, operating at 8 million microinstructions per second). There are a total of 262 144 separate processors, one for each pixel. These processors work in parallel.

The Pixel-Planes architecture is a novel approach to raster graphics in which the front part of the system specifies the objects on the screen in pixel-independent terms, and the frame-buffer memory chips themselves work from this description to generate the final image. Image primitives such as lines, polygons, and spheres are each described by expression (and operations) that are "linear in screen space," that is, by coefficients A, B, C such that the value desired at each pixel is $Ax + By + C$, where x, y is the location of the pixel on the screen. Thus, the information that is broadcast to the frame buffer is a sequence of sets (ABC , instruction), rather than the usual (pixel-address, RGB-data) pairs. In contrast to other raster systems, the most time-consuming pixel-level calculations are done neither by general-purpose processors nor by special hardware that executes only a particular set of graphics functions. Instead, Pixel-Planes is a fairly general-purpose raster engine, especially powerful when most of the pixel operations can be described in linear (or planar) expressions.

Pxpl4 contains a fairly conventional "front-end" graphics processor, implemented using the Weitek XL chip set, that traverses a segmented, hierarchical display list, computes viewing transformations, performs lighting calculations, clips polygons (or other primitives) that are not visible, and performs perspective division. For objects described by polygons, the graphics processor translates the colored-polygon-vertex description of each object into the form of data (A, B, C) for linear expressions, together with instructions for the "smart" frame buffer. An image-generation controller converts work-parallel data and instructions into the bit-serial form required by the enhanced memory chips. A video controller scans video data from the frame buffer and refreshes a standard raster display. The system is hosted by a conventional UNIX workstation that supports the system's user interface through various graphics input devices and provides system programming tools.

The heart of the Pxpl4 system is the "smart" frame buffer, an array of custom VLSI processor-enhanced memory chips. Each of these chips contains two identical 64-pixel modules. Each module has three main parts: a

² The material in this section on Pixel-Planes 4 is adapted from the Pixel-Planes Project Summary, Department of Computer Science, UNC, Chapel Hill, North Carolina.

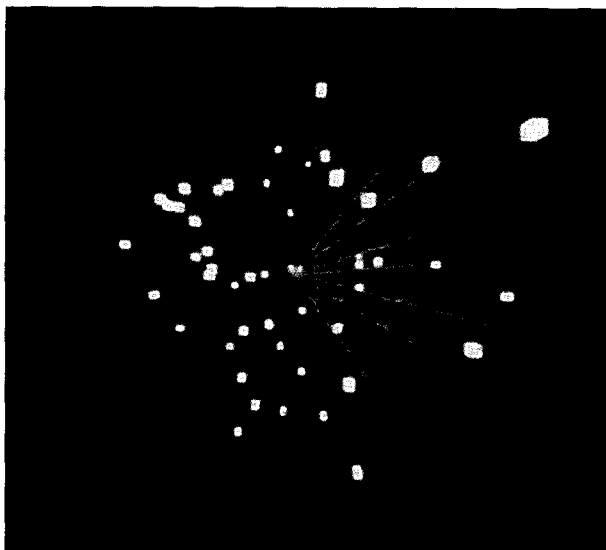


Figure 2

Initial display, showing the principal 3D space. Cubes represent the fifty states; the vectors represent seven crimes.

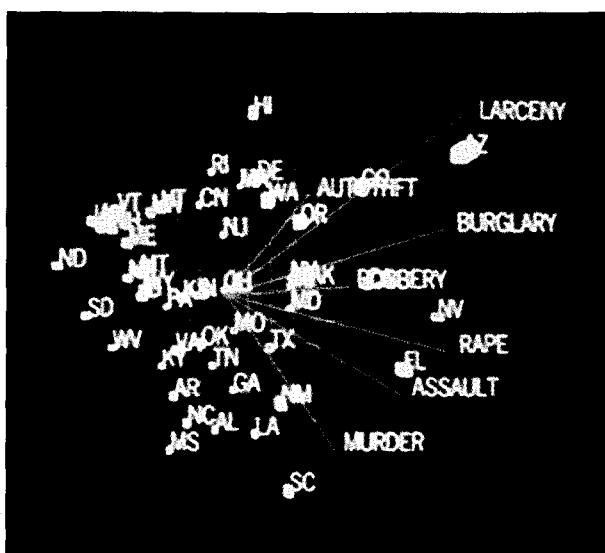


Figure 3

The display of Figure 2, with states and crimes labeled. Property crime vectors point upward, personal crime vectors point downward, and higher crime rates correspond to greater distance to the right along the vectors.

conventional memory array that stores all pixel data for a 64-pixel column on the screen, an array of 64 tiny one-bit ALUs, and a linear expression evaluator that generates $Ax + By + C$ simultaneously for all pixels. All

ALUs in the system execute the same microinstruction at the same time, and all memories receive the same address (each pixel ALU operates on its corresponding bit of data) at the same time. Pxl4 can process about 39 000 smooth-shaded, z-buffered triangles per second. Shadows are cast at about 13 000 triangles per second, using true shadow volumes. About 12 000 smooth-shaded, z-buffered, interpenetrating spheres are rendered per second.

Motion in VISUALS/Pxpl is controlled by two 3D joysticks and a slider. One joystick always controls 3D spinning. The other joystick has three modes: In one mode it controls 3D translation; in another mode it controls 6D-rotation; in the third mode it controls 6D-interpolation. The slider always controls viewing angle. Various keyboard commands implement other tools.

4. Video example

In this example we demonstrate using VISUALS/Pxpl to look for structure in seven-dimensional data. These data, which are shown in Table 1 of [7], report the crime rate for seven major types of crime in each of the fifty United States for 1977. The rate is per 100 000 population. (The data were gathered by the FBI and were published in the 1979 Statistical Abstract of the United States by the U.S. Department of Commerce.) We submitted these data to a principal components analysis and then used VISUALS/Pxpl to investigate the structure of the principal component scores and coefficients.

Figure 2³ (video scene 1) shows the initial display constructed by VISUALS/Pxpl. This is the initial 3D space V_0 . What we see is the 3D space formed by the first three principal components (the principal 3D space). The first component is displayed horizontally and the second vertically; the third is represented by the size of the cubes (large cubes are in the front of the space, small ones are in the back). In Figure 2 there are fifty cubes for the principal component scores of the fifty states, and seven vectors for the principal component coefficients of the seven crimes. The cubes represent the location of each state as projected into the principal 3D space, while the vectors represent the seven crimes as projected into this space. The length of a vector represents how close the crime is to the principal 3D space (i.e., the amount of variance in the crime that is associated with the plane). This type of plot, which shows the observations (states) as points (cubes) and the variables (crimes) as vectors, is called a biplot [11].

Figure 3 (also from video scene 1) shows the same display as Figure 2 except that the states and crimes have been labeled. We see that the crime vectors point to the right in the direction of the first principal component,

³ The photographs are taken from a video sequence made by the authors that is part of the video supplement to this issue.

indicating that it represents the overall crime rate, and that the states on the right have the highest overall crime rates and those on the left have the lowest overall crime rates. Note that the property crime vectors (auto theft, larceny, burglary, and robbery) point upward, and the personal crime vectors (rape, assault, and murder) point downward. Thus, the second principal component, which is vertical, separates property crime from personal crime.

Previous visual exploration of these data [7] has revealed that in the principal 3D space there is a cluster of southern states. This cluster does not include Florida, which has a crime pattern like northern states. In **Figure 4** (video scene 2), the same scene is shown as in Figures 2 and 3, except that the southern states are now represented by red cubes and Florida by a yellow cube, and the crime vectors have been made invisible. Since the southern states are at the bottom of the space, they are states with disproportionately high rates of personal crime. Spinning the space shown in **Figure 4** confirms that the cluster of southern states is at least three-dimensional.

We now use VISUALS/Pxpl to investigate whether the cluster of southern states is a cluster in all seven dimensions, or only in the three we see while spinning the space shown in **Figure 4**, or in some other dimensionality between three and seven. To do this, we prepare to take a guided tour of the high-dimensional data space. First we define the two targets T_0 and T_1 . We define $T_0 = V_0$, the space we have been examining. We then define T_1 to be the space whose dimensions are principal components 4, 5, and 6 (the largest space orthogonal to T_0). Spinning T_1 (video scene 4) indicates that the cluster of southern states is a cluster in this space as well.

However, further investigation suggests that the southern states may divide into three clusters, one consisting of North and South Carolina, another of Mississippi, Alabama, and Louisiana, and the third of Georgia, Arkansas, and Tennessee. If we count Florida, there appear to be four clusters of these southern states. This conclusion is reached by taking a depth-cued guided tour of the six-dimensional space formed by T_0 and T_1 .

This process (video scenes 5-7) involves the following steps. First, we focus on the states of interest by making all cubes except the southern ones invisible. This is shown in **Figure 5**, where the space has been spun into a new orientation. Second, we switch to parallel from perspective projection, to accurately portray paths of movement. The locations of the Southern states (including Florida) are shown in **Figure 6**. Here the red cubes are the Carolinas, the blue cubes are Georgia, Arkansas, and Tennessee, and the white cubes are Mississippi, Alabama, and Louisiana. (Note that colors were *not* assigned before discovering the clusters. After

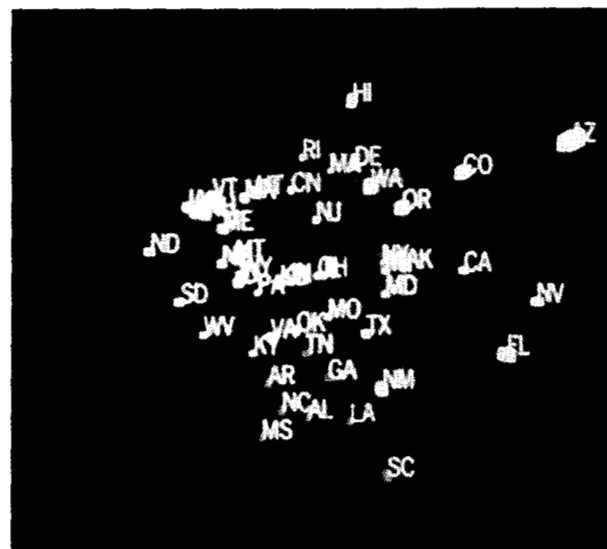


Figure 4

Same scene as Figures 2 and 3, with Florida represented by a yellow cube, the other southern states by red cubes, and the vectors made invisible.

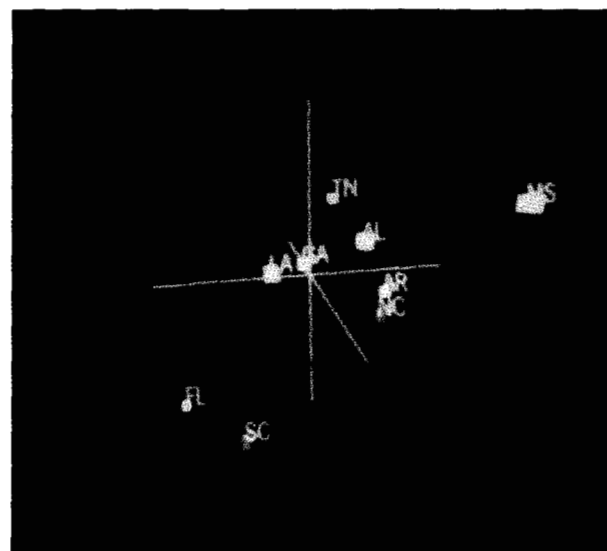


Figure 5

A different orientation of the 3D space, with only the southern states visible. Florida remains yellow, North and South Carolina remain red, and the remaining visible states are changed to white, reflecting the three apparent clusters within the group of southern states.

all, they presuppose knowledge of the clusters that did not exist prior to the visual exploration. Rather, they were assigned after visual exploration revealed the three

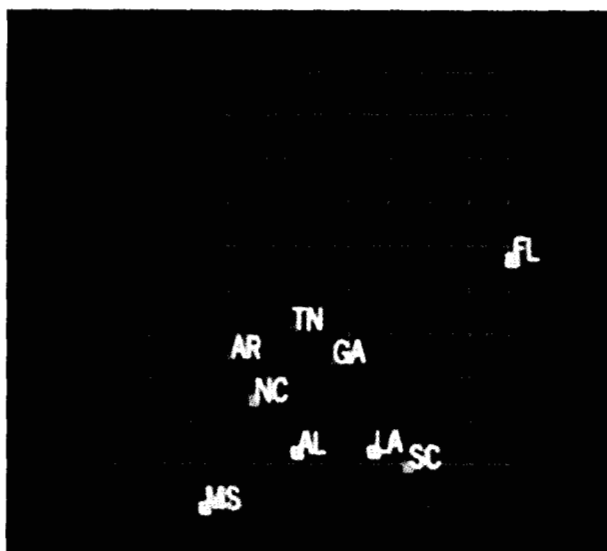


Figure 6

A parallel projection of the southern states, with color changed to blue for Georgia, Arkansas, and Tennessee.

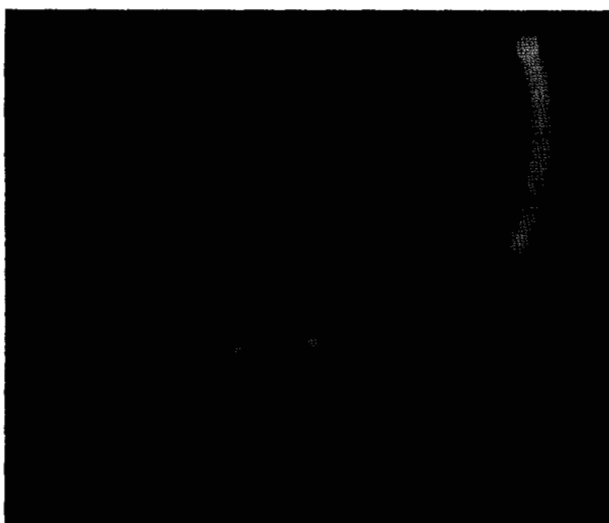


Figure 7

A time-lapse image of 6-dimensional rotation of the image of Figure 6, showing differing paths for each cluster but similar paths for the states within a cluster.

clusters.) Third, we use 6D-rotation to rock back and forth between T_0 and T_1 . A time-lapse photograph of the rocking process is shown in **Figure 7**. (Figure 6 is one end of the rocking, as can be seen by comparing the two photographs.)

What we look at during this high-dimensional depth-cued rocking is the paths of movement and the changing brightnesses of the cubes representing the nine southern states. Clusters of states which follow similar paths and which show similar changes in brightness are close together in six-dimensional space. States in different clusters will follow different paths and have different brightness changes.

This guided tour reveals that the Carolinas (red cubes) move along paths which are different from those taken by the other states but which are similar to each other; that Georgia, Arkansas, and Tennessee (blue cubes) move along another set of similar paths which are different from those taken by other states; and that the same is true for Mississippi, Alabama, and Louisiana (the white cubes). The paths are called "similar" because they show the same kind of movements.

Depth cueing further reinforces this structure. Figure 7 shows projection fit cueing, the tool which uses brightness to show the proportion of a state's total variance that is represented in the visual space. Bright portions of the paths represent projections of states which retain much of their variance in the visual space, dim portions represent projections which retain little. If the states in a cluster are located in about the same place in multivariate space, then, as they are projected onto the visual space as it rotates through six dimensions, the cubes for the states in a cluster should display similar brightness changes. By studying this photograph we see that the brightnesses of cubes that are the same color change in similar fashion (although one white-cube path—for Mississippi—seems to differ from the other two white paths), further suggesting that the states are clustered in high-dimensional space as we suspect.

The depth-cued guided tour allows us to draw a conclusion about the structure of these states in the six-dimensional space formed by the first six principal components. The conclusion is that these nine southern states appear to form four clusters in six dimensions. Apparently, there are four different patterns of crime in these nine southern states.

Finally, in video scene 9 (which is not shown in photographs here) we return to the full biplot of all fifty states and seven crimes to see if the depth-cued guided tour will reveal additional structure of interest. We notice that the Alaska cube moves in and out from the center to the upper right-hand corner as we rotate from the principal space (where Alaska is in the center) to the residual space (where Alaska is in the corner). We further notice that Alaska's brightness changes *opposite* to the way in which most cubes change: Alaska is dimmest when it is in the principal space, the space that accounts for the most variance, and it is brightest when it is in the residual space, a space which accounts for much less

variance. Thus, we see that Alaska is an outlier, and that much of its variance is accounted for by components that account for little other variance. Indeed, when we look at the original crime rate data, we see that Alaska's crime-rate profile is unusual: At least in 1977, when these data were obtained, it had the highest rate of rape per 100 000 population of any state in the country, but it did not have an extremely high rate for other personal crimes.

5. Conclusion

In the video we have seen dynamic statistical graphics that use changing brightness and movement. With the aid of these guided-tour tools we can discover and visualize structure in high-dimensional multivariate data. These graphics tools have helped us discover that Alaska has an unusual crime pattern, and that there are four six-dimensional clusters among the crime patterns of the southern states.

Note that we cannot see these four clusters as *spatial* clusters in any portion of the video. Rather, we see these clusters as *movement* clusters. Furthermore, the movement clustering is reinforced by similar patterns of changing brightness. Indeed, it may be that we cannot see a spatial structure that actually exists in a high-dimensional space in *any* of the infinity of different static projections of that space. For our example there may be no 2D (or 3D) projection that shows these four groups of states as spatially separated clusters. However, the movement and brightness clusters imply that spatial clusters *do* exist in 6D. Thus, VISUALS/Pxpl gives us a way of "visualizing" the structure of high-dimensional space by encoding the structure as movement and changing brightness in three-dimensional space.

Dynamic statistical graphics has helped us discover and visualize structure in high-dimensional multivariate data.

Acknowledgments

This research has been supported, in part, by grants 5-R24-RR-02170 from the National Institutes of Health and N00014-86-K-0680 from the Office of Naval Research.

References

1. J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley Publishing Co., Reading, MA, 1977.
2. C. Hurley and A. Buja, "Analyzing High-Dimensional Data with Motion Graphics," *SIAM J. Sci. & Statist. Comput.* **11**, 1193-1211 (1990).
3. H. Fuchs, J. Goldfeather, J. P. Hultquist, S. Spach, J. D. Austin, F. P. Brooks, Jr., J. G. Eyles, and J. Poulton, "Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes," *SIGGRAPH '85* **19**, 112-120 (1985).
4. H. Fuchs, M. Levoy, and S. M. Pizer, "Interactive Visualization of 3D Medical Data," *Computer* **22**, 46-51 (1989).
5. F. W. Young, D. P. Kent, and W. F. Kuhfeld, "VISUALS: Software for Dynamic Hyper-Dimensional Graphics," *Proceedings of the American Statistical Association, Section on*

Statistical Graphics, Alexandria, VA, 1986, pp. 69-74.

6. F. W. Young, D. P. Kent, and W. F. Kuhfeld, "Dynamic Graphics for Exploring Multivariate Data," *Dynamic Graphics for Statistics*, W. S. Cleveland and M. McGill, Eds., Wadsworth Publishing Co., Belmont, CA, 1988, pp. 391-424.
7. F. W. Young, "Visualizing Six-Dimensional Structure with Dynamic Statistical Graphics," *Chance* **2**, 22-30 (1989).
8. D. Asimov, "The Grand Tour: A Tool for Viewing Multidimensional Data," *SIAM J. Sci. & Statist. Comput.* **6**, 128-143 (1985).
9. A. Buja and D. Asimov, "Grand Tour Methods: An Outline," *Computer Science and Statistics: The Interface*, D. M. Allen, Ed., Elsevier Science Publishers, New York, 1986, pp. 63-67.
10. A. Buja, D. Asimov, C. Hurley, and J. A. McDonald, "Elements of a Viewing Pipeline for Data Analysis," *Dynamic Graphics for Statistics*, W. S. Cleveland and M. E. McGill, Eds., Wadsworth Publishing Co., Belmont, CA, 1988, pp. 277-308.
11. K. R. Gabriel, "The Biplot Graphical Display of Matrices with Application to Principal Components Analysis," *Biometrika* **58**, 453-467 (1971).

Received October 26, 1989; accepted for publication November 26, 1990

Forrest W. Young *L. L. Thurstone Psychometrics Laboratory, CB-3270 Davie Hall, University of North Carolina, Chapel Hill, North Carolina 27599.* Professor Young received his Ph.D. in quantitative psychology from the University of Southern California in 1967, and is Professor of Quantitative Psychology and Biostatistics at UNC. He has been President of the Psychometric Society, and is active in the Computational Statistics and Statistical Graphics sections of the American Statistical Association. His research interests focus on the development of methods for visualizing structure in high-dimensional data, and on the development of computer interfaces that will ease and increase the efficiency of the process of analyzing data.

Penny Rheingans *Department of Computer Science, Sitterson Hall CB-3175, University of North Carolina, Chapel Hill, North Carolina 27599.* Ms. Rheingans received a B.A. in computer science from Harvard University in 1985. She received an M.S. in computer science from the University of North Carolina at Chapel Hill in 1988 and is currently completing a Ph.D. in computer graphics. Her research interests include the application of interactive computer graphics techniques for the representation and exploration of quantitative information, multivariate data representation using color, the manipulation of virtual objects and representations, and the modeling of objects and their behaviors.