SLAC-PUB-1408
April 1974

PRIM-9

An Interactive Multidimensional Data Display and Analysis System

Mary Anne Fisherkeller, Jerome H. Friedman
Stanford Linear Accelerator Center*
Stanford, California 94305

and

John W. Tukey
Princeton University**
Princeton, New Jersey 08540

ABSTRACT

(Submitted to A.E.C. Scientific Computer Information Exchange Meeting, May 2-3, 1974)

PRIM-9 is an interactive data display and analysis system for the examination and dissection of multidimensional data. It allows the user to manipulate and view point sets in up to nine dimensions. This is accomplished by providing all 36 two-dimensional projections along the original axes at the push of a button, along with the ability to rotate the data to any desired orientation. These rotations are performed in real time and in a continuous manner under operator control. From the parallax effect, arising from the dynamic aspect of this continuous rotation, one perceives a third dimension (depth into the screen).

PRIM-9 gives the operator the ability to perform manual projection pursuit. That is, by rotation and view change he can look at his data from all possible angles in the multidimensional space and try to find

those that provide interesting structure. The system also allows interactive masking and isolation. The user can conveniently mask on any or all of the current variables, thus isolating interesting structures found along the way. These interesting structures can then be further analyzed alone, or may be subtracted from the total sample to simplify the search for still other structures. In addition to this strategy, the user may invoke an automatic projection pursuit algorithm. Starting at any projection (view), this numerical algorithm will search (in much the same manner as a human operator) for those projections that provide interesting structure. The system also incorporates the ability to save either temporarily, or permanently, any interesting view that is found. The operator can return to these views at any later time or reproduce them on a hard copy device.

*Supported by the U.S. Atomic Energy Commission under Contract AT(043)515

**Prepared in part in connection with research at Princeton University supported by the U.S. Atomic Energy Commission.

INTRODUCTION

PRIM-9 is an interactive computer graphics program for Picturing, Rotation, Isolation, and Masking - in up to 9 dimensions. It is implemented on the Graphics Interpretation Facility of the Stanford Linear Accelerator Center, Stanford University. This facility consists of an Information Display's IDIIOM refresh CRT and a Varian 620/i minicomputer, linked to an IBM 360/91.

PRIM-9 is a result of a continuing program of research into techniques for applying computer graphics to exploratory data analysis. A general introduction to its properties and uses is documented in a 25 minute sound motion picture entitled "PRIM-9"[1], produced by the Computation Research Group of the Stanford Linear Accelerator Center, Stanford University. This note details its properties with emphasis on the human engineering aspects of its implementation and on the various data analysis problems to which it can be applied.

PRIM-9 has been developed toward ends of very different breadth and distance: first, to gain insight into what can be learned by looking at the numerical aspects of data in more than two aspects at a time and, second, to implement a tool for pictorial examination and dissection of multidimensional data. The development of PRIM-9 has grown through many stages, and many of the early techniques that were implemented and then later discarded may turn out to be central to other data display systems. The resulting system, as it is currently implemented is especially straightforward in concept. Its emphasis is on picturing and rotation on the one hand and masking and isolation on the other. Picturing means an ability to look at the data from several different directions in the multidimensional space. Rotation means, as a minimum, the ability to turn the data so that it can be viewed from any direction that is chosen. Picturing and Rotation are essential abilities, valuable alone, but their usefulness is greatly enhanced when combined with Masking and Isolation. Masking is the ability to select suitable subregions of the multidimensional space for consideration. That is, only those data points that lie in the subregion are displayed. Isolation is the ability to select any subsample of the data points for consideration. That is, only those points in the selected subsample are displayed. It is important to note that masking is tied to coordinates. If we rotate the data points, different points will enter and leave the masked region. Isolation, on the other hand, is tied to the data points. Under all operations of the system (except further isolation), the isolated sample remains the same. In the absence of rotation, masking and isolation are equivalent; however, in the presence of rotation the distinction is essential.

By interactively applying picturing, rotation, isolation and masking to his data, the user can, in particular, perform projection pursuit. That is, he can look for those views that display to him interesting structure. He can isolate any structures so found and study them separately and/or remove them from the remaining sample, simplifying the search for still further structures. In this way, he may gain considerable insight into the multidimensional properties of his data. As an aid to this process, the present version of PRIM-9 also provides an automatic projection pursuit algorithm.[2] This algorithm assigns to each view a numerical index that has been found to closely correspond to the degree of data structuring in the projection. When invoked, the algorithm will search for those views of the multidimensional data that maximize this projection index. At any point in the interactive session the user may invoke the automatic projection pursuit algorithm. Starting at the current view, the algorithm will find the view corresponding to the first maximum of the projection index, uphill from the starting view. Manual projection pursuit can then continue from this new (hopefully more structured) view. The algorithm can also be invoked at the beginning of the session starting with the various original or principal axes of the data. The resulting views can then provide useful starting points for further investigation.

The next section gives a brief description of the hardware and software configuration of the Graphic Interpretation Facility of the Stanford Linear Accelerator Center, on which PRIM-9 was developed. This is followed by several sections describing the implementation of the various features of the PRIM-9 system. The note then concludes with a section discussing various techniques for applying these features to some multidimensional data analysis problems.

THE GRAPHIC INTERPRETATION FACILITY

The Graphic Interpretation Facility of the Stanford Linear Accelerator Center (SLAC) is pictured in Figure 1 and is described in detail elsewhere.[3,4] A brief description emphasizing those properties relevant to the implementation of PRIM-9 is included here.

The primary computing resource at SLAC during the development of PRIM-9 was an IBM SYSTEM/360 Model 91, with 2048k bytes of storage, operating under OS/MVT with HASP.[*] The two basic components of the Graphic Interpretation Facility are a Varian Data Machines 620/i computer[5] with 8k 16-bit words of storage, and an IDIIOM Display Console[6] with a 21-inch CRT made by Information Displays, Inc. When the display is operating, the 620/i memory contains a program for the 620/i to execute and a program (display file) for

[*] the current resource includes, in addition to the S/360-91, twin IBM SYSTEM 370/168's, operating under VS/2R1.6 with ASP.

the IDIIOM to execute. These two programs run concurrently with the IDIIOM stealing cycles from the 620/i. The instructions (orders) in the display file may display characters, points, or straight line segments, perform unconditional or subroutine jumps, count and index, or interrupt the 620/i. Characters, points, and lines are positioned on a 1024 by 1024 raster unit grid on the face of the CRT. The 620/i instruction set includes, in addition to the usual set for a standard mini-computer, instructions to start and stop the IDIIOM in its execution of the display file, and instructions to read and reset registers associated with the display operation.

Interaction at the console is by means of a solid state alphameric keyboard, a light pen, and a function keyboard with 32 buttons. Under program control, portions of the display file may be designated as light pen sensitive or insensitive. When the light pen is pointed at a sensitive item on the CRT, the 620/i will be interrupted. The function buttons generate interrupts on both the closing and the opening of the switch. This means that software can produce either (1) single impulse for each depression of the button (cycling) or (2) a repeated impulse controlled by software timing, occurring so long as the button remains down. This last facility, especially when combined with automatic reversal (see below), is of great importance in allowing effective control. Plastic overlays may be placed on the function keyboard to identify the purpose of each button.

The 620/i and SYSTEM/360 are connected through an IBM 2701 Parallel Data Adapter Unit. Data may be transmitted in either direction through this link. The 620/i can interrupt the SYSTEM/360 and determine whether the SYSTEM/360 is trying to read or write; however, the SYSTEM/360 is not able to interrupt the 620/i.

Although the Facility can operate in a stand-alone fashion, most of our work (including the implementation of PRIM-9) has been done using it in conjunction with the SYSTEM/360. The SYSTEM/360 provides fast computation and mass storage, while the 620/i maintains the display.

A package of PL/1 procedures, the IDIIOM Scope Package,[4] has been provided for writing highly interactive programs on the IBM 360/91 without burdening the writer with all of the details of programming the 620/i and IDIIOM. The user of this package can, by means of procedure calls, control the display console in addition to having all of the facilities of PL/1 available to him. Procedures are supplied to construct graphic display elements which will display information on the IDIIOM CRT, and transmit elements as well as interrupts between the SYSTEM/360 and 620/i.

IMPLEMENTATION

A. Scaling and Coordinates

The data is stored (in the 360/91) in initial coordinates either as scaled before entry or as rescaled to fit into the display region. It remains in this form. Rotations to current coordinates are performed by a transformation matrix. An ongoing step of rotation involves (A) updating this transformation matrix (multiplication by a simple rotation matrix) and (B) passing the data through the modified transformation matrix and transferring the relevant coordinates to the 620/i. Thus in the PRIM-9 system, it is important to distinguish between the initial coordinates and the current coordinates. The initial coordinates are defined to be an orthogonal set of fixed axes in the multivariate space. The input data to PRIM-9 consists of a number of ordered sets of numbers. Each of these numbers is assigned to the initial axis corresponding to its position in the ordered set. These numbers then become the values of the initial coordinates and each ordered set of numbers becomes a point in the Eucleadian space spanned by the initial axes. The current coordinate axes are another orthogonal set spanning the same space that is connected to the original set by a similarity transformation. That is, each of the current coordinates is a particular linear combination of the initial coordinates. So long as only rotations are used (see F for exceptions), these linear combinations are restricted to those that do not change the interpoint distances in the multidimensional space.

Rotation has the effect of continuously changing the linear transformation between the current and initial coordinates. At the beginning of the session, the initial and current coordinates coincide, but as soon as a rotation is performed the current coordinates become distinct from the initial ones.

B. Picturing

For picturing, PRIM-9 provides the choice of any two of the current coordinates as horizontal and vertical axes. Two buttons cycle through the choices. One button changes the coordinate displayed in the vertical direction, while the other button changes the coordinate shown along the horizontal direction. In this way, it is easy to go through the $\binom{NDIM}{2}$ (NDIM is the total number of coordinates) possible displays, as well as getting from one particular display to another. Choosing a display involves selecting two integers i and j, where i is the y (vertical) coordinate and j is the x (horizontal) one. Pushing the first button increments i by one and the second, similarly, increments j, both modulo NDIM. In order to speed up the selection process, all unnecessary combinations have been eliminated by requiring that i be greater than j. That is, i cycles from 2 to NDIM while j cycles from 1 to i-1.**

C. Rotation

Being able to see projections on all coordinate pairs can be very useful. But it is not enough. To be able to get reasonably to any two-dimensional projection means either a way to call for the projection that we want, or a way to move about in the multidimensional space. Since we usually do not know just what we want, and when we do we will find it difficult to learn to call for it in a general way, we need a way to move about. Continuous controlled rotation is a natural way to move about (change projection) in a multidimensional space.

In the implementation of controlled rotation, one naturally thinks of turning a knob. However, the configuration of the Graphic Interpretation Facility does not support a knob. Thus, we are forced to implement rotation through pushing buttons. The naive approach to the control of a single rotation by buttons involves two buttons, with these responses:

- to one button: rotation "to the right" at a constant angular rate so long as the button is depressed.
- to the other button: rotation "to the left" at the same constant angular rate so long as the button is depressed.

** If we were programming this action again, we would use a constant-step version of the increasing-step-and-reversal control described under rotation.

This type of control has two serious flaws:

- if the rotation rate is slow enough for fine adjustment, the delay time for large rotations is undesirable -- so undesirable as to be nearly impractical.
- if used naively -- two buttons per rotation -- it is too easy to use up too many buttons. (PRIM-9, in its present version, makes as many as $\binom{9}{2} = 36$ different rotations available.)

Both of these disadvantages can be overcome, the first by an "increasing-step-and-reversal" control, the second by "time-sharing" the rotation drive.

The type of rotation control used in PRIM-9 has two features:

(1) rotation reversal -- rotation in one sense so long as the single button is held down -- when the same button is released and then again depressed, rotation in the opposite sense so long as the button is held down.

(2) rotation by increasing (accelerating) steps. These steps are presently taken as 1, 1, 2, 4, 8, 16, 32,... times a small (unit) angle. (Note the repetition of 1.) The largest possible step is limited by a speed limit settable at 1, 2, 4, 8, 16, 32, 64, or 128.

This combination of rotation reversal and acceleration gives the operator fast and easy approach to a desired data orientation. The rotation starts out slowly but quickly accelerates to the speed limit so long as the button remains depressed. When the operator sees an interesting data orientation on the CRT screen, he releases the button. Due to human reaction time, both in perception and in releasing the button, the rotation usually overshoots the desired data orientation. Depressing the same button again causes the rotation to proceed in the opposite direction, starting out at the slowest speed, and again accelerating. When the desired orientation again comes into view on the screen the button is released. Now, due to the slower speed (the acceleration usually has not reached the speed limit), the overshoot is much less (in the opposite direction). Again, depressing the button causes rotation reversal at the slowest speed allowing the operator to home in on the desired data orientation. In the usual case, at most two reversals are necessary. This strategy allows rotation in both directions, minimizes the delay time for large rotations, allows a slow rotation for fine adjustment, and requires only one button to drive the rotation.

In order to specify a rotation, one needs not only to specify the sign and magnitude of the rotation angle but also the rotation axis. A general rotation axis in a multidimensional space (for example, in terms of its direction cosines) is complicated to understand and time consuming to specify. In PRIM-9, directly available rotations are confined to those associated with pairs of the current coordinates. This makes both control and understanding relatively easy. A rotation axis is specified by two integers i and j. These integers specify the current coordinate axes that participate in the rotation. That is, coordinates i and j rotate while the other NDIM-2 coordinate axes, orthogonal to i and j, remain fixed. It is possible to get from any one two-dimensional projection to any other, with relative ease, by combining these selected rotations in the correct amount and sequence (this is the multidimensional analog of the Euler angle specification of a rotation in three dimensions.) The two integers (i and j) that specify the coordinates that participate in the rotation are selected in exactly the same manner, as discussed in the previous section for choosing the current projection axes. The selection is controlled by two buttons that cycle through the $\binom{NDIM}{2}$ possibilities. One button cycles through $2 \leq i \leq NDIM$ in increments of one while the other, similarly, cycles through $1 \leq j \leq i-1$.

If the axes that define the rotation coincide with the current projection axes, then the data points will simply move in circular orbits about their relative mean in the projection. If neither axis corresponds to a current projection axis, then the display will remain unchanged because the rotation is orthogonal to the current projection axes. Both rotation axes are "invisible" to the screen. Useful rotations occur when one of the axes that participates in the rotation corresponds to a current projection axis, while the other is one of the NDIM-2 axes orthogonal to the current display plane. In this case, the operator sees the projected points change their positions on the screen as the invisible coordinate is rotated against the visible one. When an interesting pattern emerges, as a result of the rotation, the operator can home in on it as described above. The invisible coordinate can then either be rotated against the other visible one or the operator can change to another invisible coordinate. He can then rotate these new coordinates to try to sharpen the structure even further. Continuing in this manner, the operator may manually iterate to a data orientation that provides an informative view of his multidimensional point cloud.

Easily recognizable continuous rotation provides an additional advantage. Its dynamic effects let one see an additional dimension, not instantaneously, but yet at the same time -- in the best sense of those words. This is due to the relative motions of the points associated with parallax. The points that are closer in the invisible rotation coordinate move across the screen more rapidly than those that are farther away. This parallax effect gives the illusion of a third dimension (depth into the screen) and this aspect seems to be a very useful complement to the two aspects (horizontal and vertical) provided directly by the screen.

To help in the interpretation of a particular projection, a display of the initial coordinate axes, as projected onto the current projection plane, is easily switched (with a pushbutton) in and out of view. This display allows the user to see graphically the linear combinations of the original coordinates that comprise the two axes of the current projection plane. Another button (neutral) returns the display to its initial state. That is, the current coordinates are reset to the initial ones.

D. Masking

Masking is the ability to select any subregion of the multidimensional space, and have only those data points that lie in the subregion displayed on the screen. Masking is used in connection with isolation and also has some interesting uses of its own (these are discussed in the last section). It is important to note that masking is tied to the coordinates. Under rotation, the data points will enter and leave the masked region.

The flexibility of PRIM-9's masking, like the flexibility of its rotation, is limited so that it is easy to control and understand. PRIM-9 allows simple masking on any or all of the current coordinates. That is, those points for which $X_i < F_i$ or those for which $B_i < X_i$, or both -- for a single i or several i's -- are caused not to appear on the screen. Here X_i $(1 \leq i \leq \text{NDIM})$ are the current coordinates and F_i, B_i are forward and backward bounds on each of these current coordinates.

Masking is controlled by five buttons. One button toggles the masking on and off. A second button cycles (one step per press) through the integer, i, $(1 \leq i \leq \text{NDIM})$ which identifies that current coordinate to which the mask is to be applied or altered. The third identifies which edge F (front), B (back) or J (joint) is to be driven. These three options are cycled through by successive pushes of the button. The fourth button drives the selected mask in a continuous motion using the same "increasing-step-and-reversal" control technique described above for rotation. The fifth button allows the rapid selective removal of the mask for a particular coordinate (identified using the second button) by resetting both the front and back mask edges to their outside positions.

This is in contrast to the effect of the first button which, when toggled off, removes the masks on all coordinates simultaneously. The J (joint) drive moves the front and back masks together in the same direction and speed, maintaining a fixed separation between them. This allows driving an unmasked zone of chosen width back-and-forth on a particular coordinate.

If the masking coordinate is one of the current projection axes, then driving the mask away from its outside position and toward the center of the screen will cause the points to disappear along an advancing line, accelerating to the speed limit so long as the drive button is held down. When the button is released and again depressed, the masking line will reverse, starting at the slow speed. The masked points will reappear as the mask boundary retreats. As for the case of rotation, this control allows the operator to arrive at and home in on a desired mask position easily and quickly.

A masked coordinate need not be a current projection axis. For example, one can mask on a current coordinate orthogonal to the projection plane. As the mask moves from its outside position on this "invisible" coordinate, points will disappear from various regions of the screen giving insight into the relationship contained in the data between the invisible coordinate and the two visible ones. In particular, the joint drive allows driving an unmasked zone of chosen width back-and-forth on the invisible coordinate. This will cause points to appear and disappear as the mask passes through the various values along the invisible coordinate. More sophisticated techniques using moving masks are discussed in the last section.

E. Isolation

Isolation is the ability to select an arbitrary subset of the data sample at any point in the analysis, and perform upon this subset or its complement (the full sample with the subset removed), all operations that one can perform on the full sample. Experience with PRIM-9 has shown that isolation is a most essential adjunct to picturing and rotation. It extends the system from being a purely linear device to a piecewise linear device, greatly increasing its effectiveness and power. Some of its more standard applications are discussed in the last section.

As implemented in PRIM-9, isolation begins with masking. The data points to be isolated are defined by constructing a mask (in terms of the current coordinates) that just contains the points to be isolated. As the mask is applied, the points that are masked out disappear from the screen, making it relatively easy to interactively construct a mask that just includes the desired subset of points to be isolated. It might seem, at first thought, that because of the limited flexibility of PRIM-9's masking, it would be difficult to construct mask boundaries that include arbitrary point subsets. This turns out not to be the case. This is due mainly to the fact that the current coordinates, on which the mask is defined, can be interactively rotated to an arbitrary orientation with respect to the initial coordinates, and that the isolation can be applied repeatedly in defining the subset. In this way, the user can construct a piecewise linear approximation to any boundary surface in the multidimensional space to sufficient accuracy to just include the points to be isolated.

When all of the undesired points have been masked out, so that only the subsample to be isolated appears on the screen, the user presses a button to invoke the isolation. This causes a menu of options to appear. Figure 2 is a photograph of this menu. The user selects the appropriate option by touching a light pen to those places on the screen that define the option.

The numbers at the right reference the sixteen isolates that can be simultaneously defined. Isolate "0=ALL" always references the total sample and "15=RESIDUAL" is reserved for special use with the "residual after" option. This leaves fourteen isolates that can be arbitrarily defined and stored by the user.

Touching the light pen to one of the seven options to the left causes a question mark to appear either to the right of the option or at a blank space within it. Touching the pen to one of the sixteen numbers causes the selected number to replace the question mark. When all of the question marks have been replaced by numbers and the command is correct, then touching "APPROVED" initiates the action. The commands are:

FILL n:	Save currently masked subset at n.
RECALL n:	Recall isolate saved at n and replace current subset with it.
SUBSELECT ON n1 FILL n2:	Save at n2 the intersection of the current subset with the isolate stored at n1.
INTERSECT n1 AND n2 FILL n3:	Save the intersection of isolates n1 and n2, at n3.
UNION n1 AND n2 FILL n3:	Save the union of isolates n1 and n2, at n3.
NOT n1 BUT n2 FILL n3:	Save the intersection of the complement of isolate n1 with isolate n2, at n3.

RESIDUAL AFTER n ALSO:	Replace isolate 15 by the intersection of the complement of isolate n with the current isolate 15.

If only a number is light-penned, "RECALL" is the default command. As each subset is isolated, the remainder may be saved with the last command. An asterisk appears to the left of those numbers that represent isolates that are currently in use. When an isolate has been saved or recalled, it becomes the current subset. If instead of a light pen hit the button invoking the isolation is simply pushed a second time, the display will alternate between the current subset and the entire data sample. The current isolate number always appears at the bottom of the screen.

F. Scale and Location Transformations

The location may be displaced in either the positive or negative direction and/or the scale can be expanded or contracted on any current coordinate axis. This is accomplished by specifying a "key" integer, i, $(1 \leq i \leq \text{NDIM})$ representing the coordinate to be transformed. A single button cycles through the possible values of i. Another button displaces the origin of the selected coordinate while a third button scales the coordinate. These latter two buttons drive the displacement or scale change in a continuous motion, using the "increasing-step-and-reversal" control technique described earlier.

G. Saving Views (Projections)

Quite often during a session, the user finds a projected view of his data that is sufficiently interesting to warrant saving it so that he may return to it later in the session, at another session, or perhaps record it permanently on a hard copy device.

PRIM-9 provides for both temporary (within a session) and permanent (between sessions) view saves. The permanent saves can also be transferred to a hard copy device at the user's discretion. This facility allows the user to continue with an analysis after he has found an interesting view. If further analysis should not improve the structuring, or perhaps even worsens it, the user can simply recall the saved view and begin again on a different track.

Saving a view (projection) consists of storing the transformation matrix connecting the initial coordinates to the current coordinates along with the mask bounds at the time of the save. Up to six different views may be saved on a temporary basis and another eighteen may be saved in a permanent disk data set. The temporarily saved views are simply identified by their number, i $(1 \leq i \leq 6)$, while the permanently saved views on the disk are given identifying names typed by the user on the keyboard. If no such name is typed, then the system assigns a default identifier which is the date and time of the save. To retrieve one of these views, the user depresses a button which presents a menu on the screen listing the names of all of his saved views. He selects a view by touching the light pen to the appropriate name.

H. Automatic Projection Pursuit

The PRIM-9 system provides the user with a sort of "automatic pilot" for rotation. That is, at the user's request, the system will invoke a numerical algorithm that automatically searches for data orientations (projection directions) that display interesting structure. This algorithm is detailed elsewhere[2] and only its general properties, as they relate to the implementation on the PRIM-9 system, are discussed here.

The automatic projection pursuit algorithm assigns to each projection a numerical index, $I(\hat{k},\hat{\ell})$, that corresponds to the degree of data structuring present in the projection. Here $\hat{k}$ and $\hat{\ell}$ are the two orthogonal unit vectors representing the particular two-dimensional projection of the NDIM-dimensional data. The more structure present in the projection, the larger $I(\hat{k},\hat{\ell})$ becomes. The essence of the algorithm is to find those projection directions ($\hat{k}$ and $\hat{\ell}$) that maximize $I(\hat{k},\hat{\ell})$, subject to the constraint $\hat{k} \cdot \hat{\ell} = 0$. This projection index, $I(\hat{k},\hat{\ell})$, is constructed to be a smooth function of its arguments so that sophisticated numerical maximization algorithms can be employed, minimizing the CPU cycles required.

The automatic projection pursuit algorithm can be invoked in two ways from the PRIM-9 system. At the simplest level the user simply depresses a button. Starting with the current projection (appearing on the screen), the algorithm finds (and displays on the screen), the projection corresponding to the first local maximum of the projection index, $I(\hat{k},\hat{\ell})$, uphill from it. In this search the horizontal coordinate, $\hat{k}$, is held fixed while the vertical coordinate is varied in the (NDIM-1)-dimensional subspace orthogonal to $\hat{k}$. Depressing the button a second time causes the search to continue, but this time the veritical coordinate is held fixed at the previous solution value, $\hat{\ell} = \hat{\ell}^*$, while the horizontal coordinate, $\hat{k}$, is varied in the (NDIM-1)-dimensional subspace orthogonal to $\hat{\ell}^*$, seeking a further maximum of $I(\hat{k},\hat{\ell}^*)$. Pressing the button a third time causes a further maximization, this time holding the horizontal coordinate $\hat{k}$ fixed at its solution value, $\hat{k} = \hat{k}^*$, and again varying the vertical one, but this time in the subspace orthogonal to the new horizontal coordinate $\hat{k}^*$. This procedure of alternately holding one coordinate fixed while varying the other in the subspace orthogonal to the first, can be continued (by repeated pushes of the button) until either an interesting view appears or until it converges (subsequent searches produce no change in the projection). At each stage in this iterative process, the results are displayed on the screen as the current projection.

The second mode of invoking automatic projection pursuit allows starting the search at projections other than the current one displayed on the screen. This is accomplished by depressing a different button. This causes a menu to appear on the screen listing the options available. These options allow the choice of any two of the initial coordinates or principal axes of the current data set (isolate) as starting axes for the automatic projection pursuit. This menu also allows the interactive modification of some of the parameters of the automatic algorithm as well as simply displaying the data along the various principal axes without the corresponding search. The user selects from among these options by touching the light pen to the appropriate places on the screen where the options appear. After starting the search at two of these alternate axes, the user continues it by repeatedly pressing the first button as described above.

DISCUSSION

PRIM-9 has been available for production data analysis for a relatively short time so that our experience with it is necessarily limited. We have probably not yet discovered many of its most interesting and useful applications. It has, so far, been employed in the exploratory analysis of multivariate high energy particle physics data, and in both supervised and unsupervised multivariate discrimination analysis in pattern recognition problems.

In the exploratory data analysis application, the system essentially functions as a cluster detection and separation device. By repeated application of view change and rotation (both manual and automatic), the user tries to find those data orientations that reveal to him interesting structure or clustering. Using the automatic projection pursuit algorithm (started at the larger principal axes of the data) to find interesting starting projections, the user then manually iterates the rotation to try to find structure or to sharpen any structure found. This iteration process usually proceeds as follows. A visible coordinate is rotated against one of the invisible ones until the clustering is as sharp as possible. This invisible coordinate is then rotated against the other visible one toward the same end. Another invisible coordinate is then chosen to be rotated against the two visible ones, hopefully increasing the cluster formation and separation. After all of the NDIM-2 invisible coordinates have been rotated against the current visible ones, the whole process can be repeated (since in the process of these rotations the current coordinates have been considerably changed) so long as progress is being made. At any point, the automatic projection pursuit algorithm can be invoked (this usually happens when progress being made by manual rotations is slow).

If a view is found that separates the data into two or more clusters, this view can be used (via masking and isolation) to isolate the clusters into separate data subsamples. These subsamples can then be analyzed individually to see if there are different projections that reveal still further clustering within each of the subsamples. Even if no further clustering is found in an isolate, rotating it in a controlled manner in the multidimensional space can still give considerable insight as to its multivariate properties. The parallax effect of the continuous real time rotation is very useful in providing the third-dimensional effect of depth into the screen.

Moving masks can also be useful in gaining insight into the multidimensional characteristics of data point sets. The simplest of these (as mentioned above) is the sliding of an unmasked zone of chosen width back and forth on the various coordinates invisible to the screen, one at a time. Another more sophisticated approach might be called the "concealed generalized episcotister". Its operation would be roughly as follows. Let each coordinate run between -1 and +1, let a, b, c and d be small fractions, and let the data for this example be six-dimensional.

picture = coordinates 1 and 2

masks = F_3 at -a, B_4 at +b, F_5 at -c, B_6 at +d

rotation = play with (4,3), (5,3), (6,3), (5,4), (6,4), (6,5)

The unmasked hypervolume is a hexadecant in the four coordinates 3, 4, 5 and 6. Rotating these coordinates among themselves essentially rotates this hyper-conical (linearly, not spherically hyper-conical) region around among the (four-dimensional projections of the) points, thus generalizing the rotation of a conventional (sector-disk) episcotister. This approach seems often rewarding in gaining the first clues as to major structure in a point cloud.

If, as another example, we suspect that the "core" (in one or more of the invisible coordinates) of a visible concentration of points is curved, the natural mode of inquiry is to (in order):

(1) rotate until the concentration seems as strong as possible (this will tend to eliminate "tilts" of the core),

(2) mask on one invisible coordinate, coming in from each side until perhaps one-sixth to one-fourth of all points are blotted out from each side,

(3) operate mask-countermask rapidly,

(4) rotate among the invisible coordinates at whim, repeating number (3) all the while.

Appearance of a displacement oscillating with the mask-countermask frequency is an indication of existence, direction, and amount of curvature.

In the pattern recognition applications, the system functions as both a linear and piecewise linear device in supervised and unsupervised discrimination analysis.

As an example of its use as a linear device in a supervised application, consider the problem of finding the best linear discriminant direction in the multivariate space for separating two known data classes. Here, the points corresponding to each class are correctly identified from some external source and the problem is to find the best direction separating the two classes when the data are projected onto that direction. The analysis proceeds as follows. A coordinate is added which contains the information identifying the class of each data point. This extra coordinate is pictured (vertically) against each of the data coordinates (horizontally) in turn, starting with the <u>second</u> coordinate.

NDIM coordinates with the objective of achieving maximal overlap of the two classes in each horizontally pictured coordinate. This process is repeated iteratively until it is impossible to increase the overlap in each of the NDIM-1 projections (2 through NDIM). At that point the direction of the current first coordinate represents the optimal discrimination axis, and picturing the extra (class identifying) coordinate against this first one, directly displays the discrimination achieved.

It is the presence of isolation in conjunction with rotation that gives the system its piecewise linear capability. This is easily demonstrated by considering the simple example of a two-dimensional data set, consisting of two classes whose boundaries are outlined in Figure 3. Clearly, there is no single one-dimensional discriminant direction that can completely separate the two classes (A and $B = B_1 + B_2 + B_3$).

Applying rotation (both manual and automatic), the user might find a projection that achieves a good partial separation (such as P_1 in Figure 3). Using this projection, a mask is constructed at M_1 and the B_1 sample is isolated from its complement $A + B_2 + B_3$. These two isolates are then rotated separately, searching for further structure. In the case of subsample B_1, this will yield a null result. For the subsample complement to B_1, however, the user may iterate to projection P_2 which does exhibit further structure. Using this projection to apply a mask at M_2, the B_2 sample is isolated from its complement $A + B_3$. Continuing with this procedure, the user can further iterate to projection P_3. Applying a mask at M_3 in this projection completely isolates Class A from the remainder of Class B (B_3). Thus, the application of the piecewise linear mask M_1, M_3, M_2 completely separates the two data classes. For the supervised case, where it is known in advance that there are only two classes, this completes the solution. For the unsupervised case, one can separately analyze the B_1, B_2 and B_3 isolates as well as their union ($B_1 + B_2 + B_3$), using rotation and masking to determine whether they represent separate classes or whether they are connected, thus representing a single class.

For this simple illustration, the dimensionality of the data was two while the projection pursuit was in one dimension. In PRIM-9, the data dimensionality can be as large as nine while the projection pursuit is in two dimensions. However, the basic principles are the same. One applies rotation and view change, trying to find projections with structure. When structures are found, they are isolated and the isolates are each individually studied seeking further clustering. If found, these are further isolated and so on. When no more structuring can be found among the isolates, then they are analyzed separately and together to try to understand their multivariate properties.

CONCLUSIONS

PRIM-9 has been in use for a short time examining and dissecting multivariate data. Its direct value for this purpose will have to be learned by experience. We have learned from its development that pictorial systems to be effective must, as did PRIM-9, go through many stages of trial and error learning. We now understand that the details of control can make or break such a system. We now recognize that these details of control must be adapted to what is available and to the people who are to use the system. If we had ten knobs and six switches instead of 32 buttons and a light pen, we would have realized the same four essentials in a quite different way. We now recognize the great value of the dynamic aspects of the display, especially easily recognizable rotation. Two aspects, horizontal and vertical, are always before us. We now have a strong feeling that the third aspect which supports these two best is this dynamic aspect of rotation, more useful than stereoscopy, color, flicker or distinctive characters. We have learned that this sort of pictorial facility can be useful in two quite different ways: first, directly in the interactive analysis of multivariate data, and second, as a source of ideas and approaches for the development of computer algorithms for multivariate analysis. The automatic projection pursuit algorithm, for example, was developed by observing the systematics of the interaction between the users and the PRIM-9 system. These systematics were encoded into a computer algorithm which has been very successful in seeking optimum projections.

Finally, and above all, we have learned that the four essentials of Picturing, Rotation, Isolation and Masking need to work together and that from them much can be learned.

REFERENCES

1. Film: "PRIM-9", produced by Stanford Linear Accelerator Center, Stanford California, S. Steppel, Ed., Bin 88 Productions, April 1973.

2. J.H. Friedman and J.W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," Stanford Linear Accelerator Center, Stanford, California, Report, SLAC PUB-1312, September 1973. (to be published in IEEE Trans. Computers)

3. R.C. Beach, M.A. Fisherkeller, and G.A. Robinson, "An On-Line System for Interactive Programming and Computer Generated Animation," Stanford Linear Accelerator Center, Stanford, California, Report, SLAC PUB-939, August 1971.

4. R.C. Beach, "The SLAC Scope Package for the IDIIOM--A Collection of PL/1 Procedures which may be used to control the IDDIOM Display Console," Stanford Linear Accelerator Center, Computation Research Group, Report No. CGTM No. 80, December 1969.

5. Varian Data 620/i Computer Manual. Bulletin No. 605-A, Varian Data Machines, 2722 Michelson Drive, Irvine, California.

6. IDIIOM Technical Description. Information Displays, Inc., 333 North Bedford Road, Mount Kisco, New York 10549.

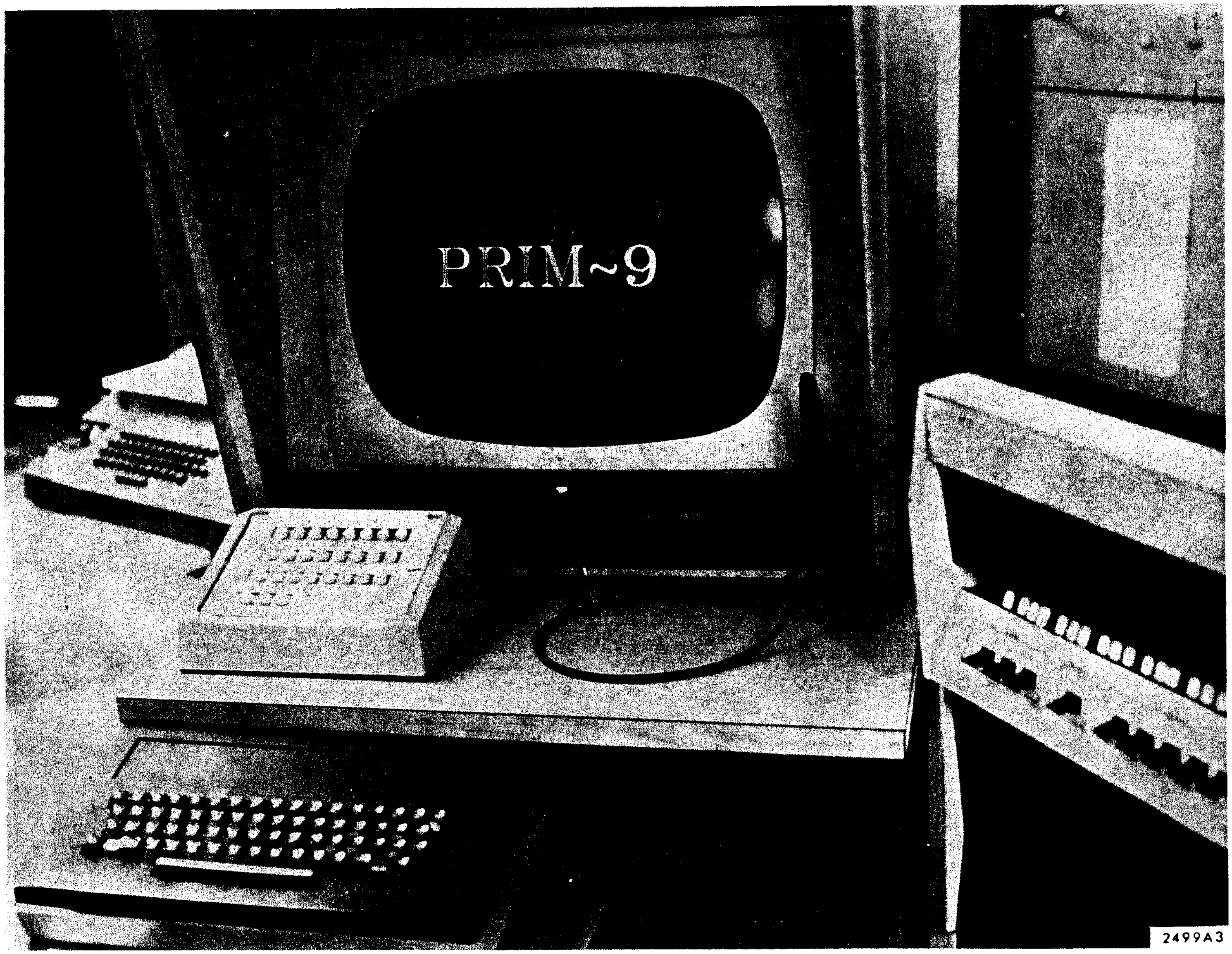
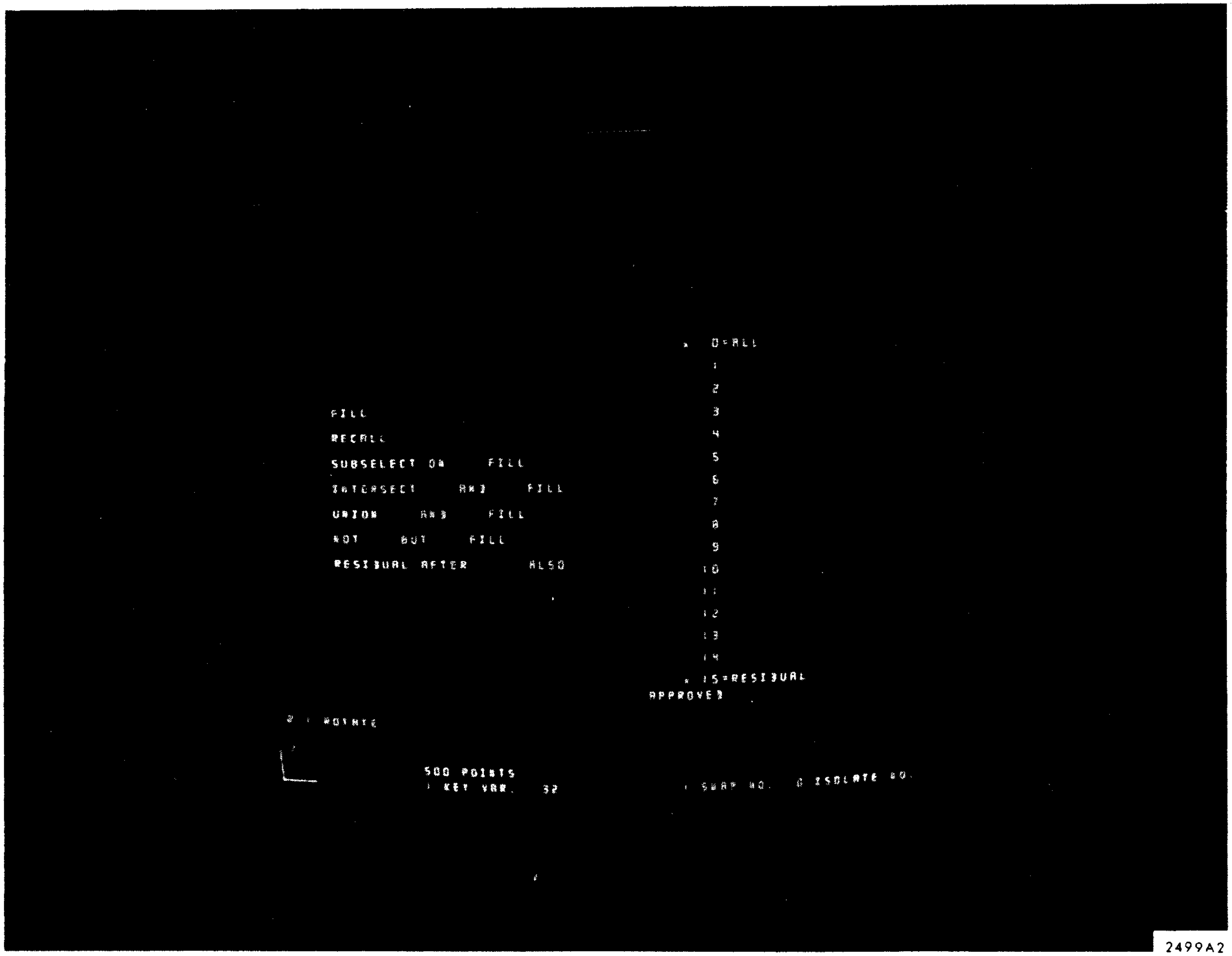


FIGURE 1

Sample



```
                                          *  0=ALL
                                             1
                                             2
FILL                                         3
RECALL                                       4
SUBSELECT ON     FILL                        5
INTERSECT     AND     FILL                   6
UNION     AND     FILL                       7
NOT     BUT     FILL                         8
RESIDUAL AFTER       ALSO                    9
                                            10
                                            11
                                            12
                                            13
                                            14
                                          * 15=RESIDUAL
                                        APPROVED
2 ↑ ROTATE

              500 POINTS
              ↑ KEY VAR.   32              ↑ SWAP NO.   6 ISOLATE NO.
```

2499A2

FIGURE 2

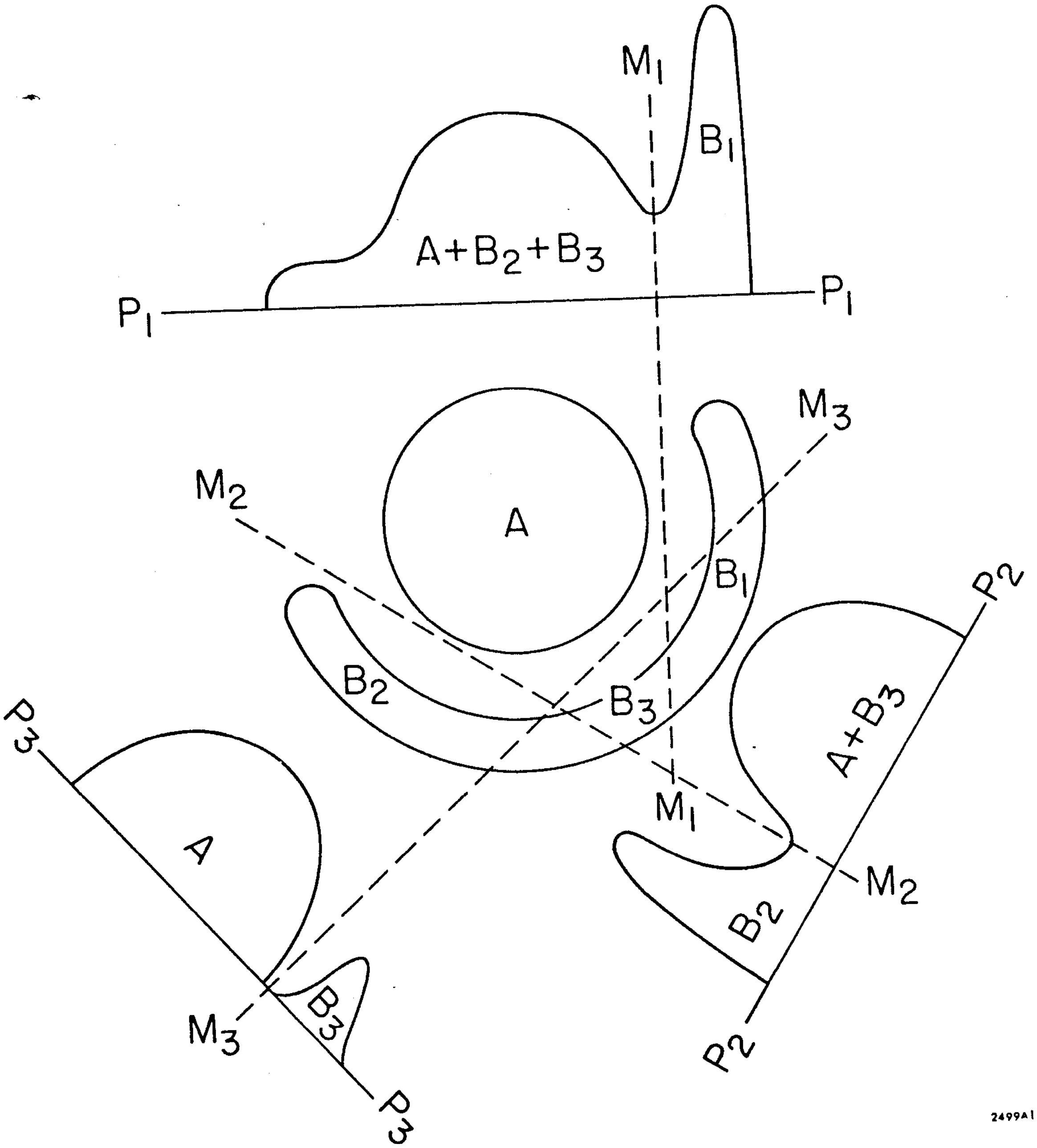

M1
B1
A+B2+B3
P1
P1
M3
M2
A
B1
P2
B2
B3
A+B3
P3
M1
A
M2
B2
M3
B3
P2
P3
2499A1

FIGURE 3