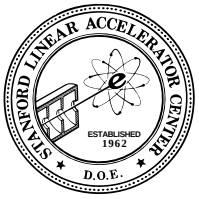
SLAC-PUB-5758
February 1992
(E/I)



HippoDraw and Hippoplotamus*

Michael F. Gravina, Paul F. Kunz,
Tomas J. Pavel, and Paul E. Rensing
Stanford Linear Accelerator Center
Stanford University
Stanford, CA 94309, U.S.A.

Abstract

HippoDraw is a NeXTstep application for viewing statistical data. It has several unique features which make viewing data distributions highly interactive. It also incorporates simple drawing tools. HippoDraw is written in Objective-C and uses the Hippoplotamus library package which handles the n-tuples and displays. Hippoplotamus is written in ANSI C.

1. Introduction

HippoDraw is a result of research into finding better ways to visualize the kind of statistical data that is so common in high energy physics analyses. In these analyses, frequency distributions are visualized as histograms, contour plots, scatter plots, *etc*. Traditionally, one used a library of subroutines, called a histogram package, within one's analysis programs to create and display such distributions. The problem with this approach was that one frequently selected parameters of the created plots that poorly represented the data distributions. Thus, one needed to re-run the analysis program, for example, just to change the number of bins in a histogram.

With the advent of powerful time-shared mainframe computers, it became possible for analysis programs to store the data that one wanted to visualize in a file and to use an interactive program to create and display the data distributions at a later time. In high energy physics, this was first demonstrated with the GEP[1] program at DESY in 1978. The utility of this technique was further proven at SLAC in 1983 with the IDA[2] program and at CERN in 1987 with the PAW[3] program. The most common way of storing this data has been in the form of a table of floating point numbers. The table has a fixed and small number of columns and an indefinite, perhaps large, number of rows. Technically, each row of the table is an ordered n-tuple, but commonly in high energy physics the whole table is called an n-tuple. Having the data represented in this form allows not only interactive creation of data distributions, but also allows one to select the data to be entered into a distribution depending on the value of a datum in the same row but another column. This selection process is commonly called "applying cuts to the data."

HippoDraw extends this basic technique by making the creation and display of plots even more interactive than the command line driven programs of the past. It achieves this by making every aspect of the creation and display controlled by a mouse in a Graphical User Interface (GUI) environment. The user's interaction with the plots is further enhanced by the use of sliders and the continuous updating of plots while the user drags a slider with a mouse.

In the next section, the interactive features of the HippoDraw will be described. HippoDraw was written using object oriented programming techniques in the Objective-C language for the NeXTstep environment. For the management of n-tuples and displays, a library package called Hippoplotamus (or Hippo for short) which is written in the ANSI C language is used. It has an object-oriented style and will be described in a following section.

* Work supported by Department of Energy, contract DE-AC03-76SF00515.

Presented at the 2nd International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, La Londe-Les-Maures, France, January 13-18, 1992.

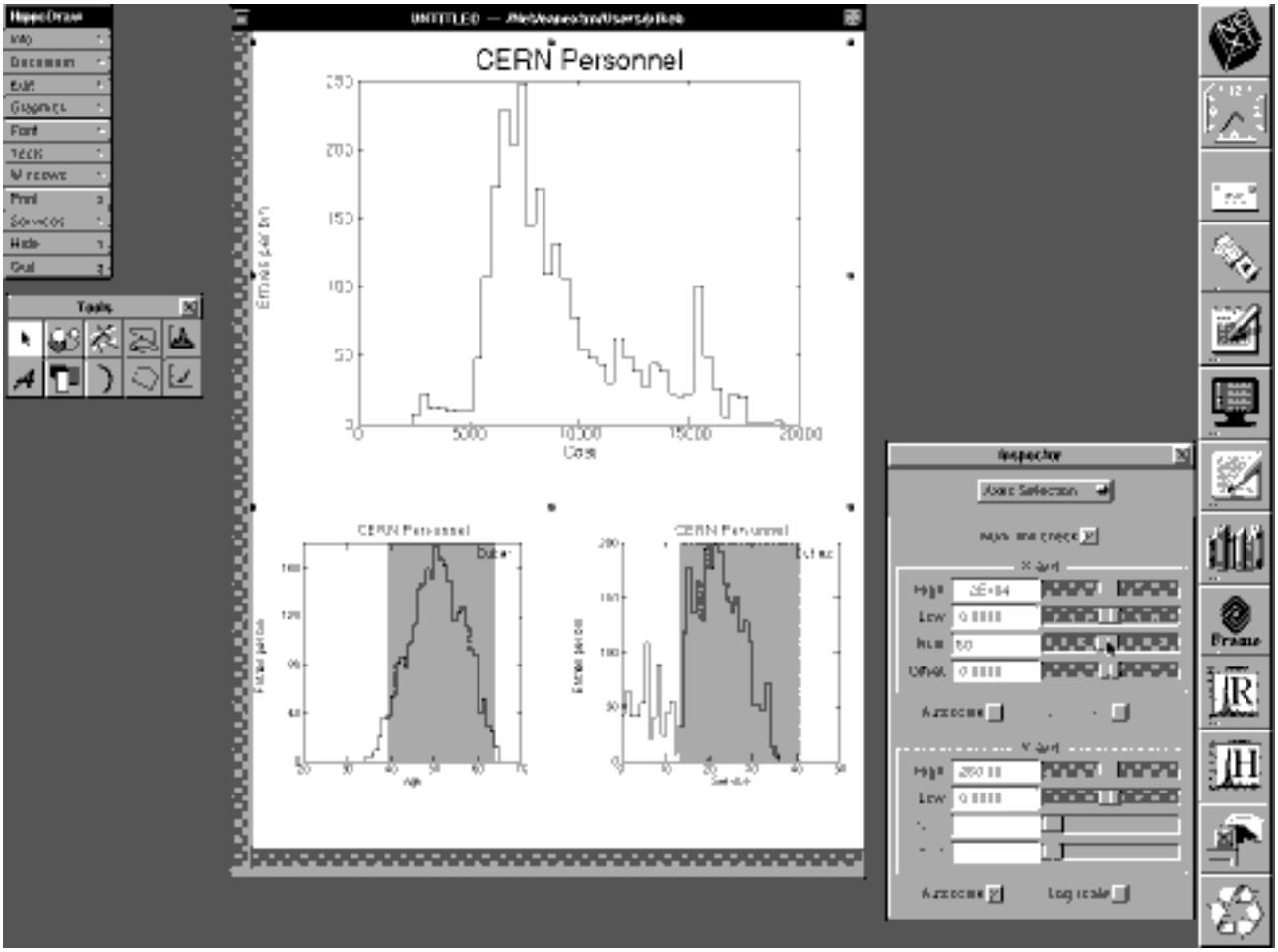


Figure 1. Screen image showing HippoDraw windows.

2. HippoDraw

The highly interactive nature of HippoDraw becomes apparent soon after the application is launched. The first step a user takes is to open a file containing the n-tuple. In the NeXTstep environment, this is done via a standard panel, called the Open Panel, which gives the user a view of the underlying UNIX file system and a means to navigate around it to find the desired file using only the mouse or keyboard shortcuts. When the file is found, the user can double click on its name, which appears in a scrollable list called a browser, click on a button labeled "OK," or use the carriage return key on the keyboard. All this functionality is provided by an object of the Open Panel class, one of the many classes which are provided in the NeXTstep environment that HippoDraw uses. There is also a Save Panel for saving the drawing document.

The next step is to create and display a distribution. This is where the "Draw" of HippoDraw comes in, because it is a simple drawing program like one might find on popular personal computers. A distribution plot is handled as a graphic object, just like a circle, rectangle, or polygon. On the left side of the screen image shown in Figure 1 is a panel labeled "Tools." It consists of 10 buttons with icons. Most of the buttons are used to invoke the standard drawing tools, but the two right-most buttons are used to create new plots. The upper button is for 1D histograms, and the lower button for 2D displays. Clicking on one of these buttons will create a new plot and insert it onto the drawing canvas.

Once a plot has been created, it can be moved within its window using the drawing program paradigm of clicking on it to select it, then dragging it with a mouse. It can be resized by dragging on one of the "handles" that are displayed when the plot is selected. Characteristics, such as line width and

color, can be changed just like any other graphic object and new plots can also be created by cut and paste operations.

Since it is very common for a user to want to see more than one distribution at a time, new plots are initially small in size, only about 5 cm square, which is half the width of the drawing window. This makes it easy for the user to arrange many plots in the window in any way that is desired. Thus, HippoDraw has quite an advantage over programs that only allow viewing one distribution at a time, or only allow viewing multiple distributions in some fixed arrangement.

A new plot is inserted onto the drawing canvas with a default set of parameters. For example, histograms are created with the first column of the n-tuple data being displayed in 50 bins. In HippoDraw, all these parameters can be changed with the point and click paradigm. For example, to change which column of the n-tuple is to be displayed in a distribution, one first invokes a panel which lists the labels of the columns in a scrollable list. This list serves two purposes. It is first a reminder to the user of what is contained in the data set. Secondly, it is the place to point and click so the user is never conscious of the column numbers.

All parameters of a histogram, such as the number of bins and the display range, are changeable via sliders and/or form fields. As the user drags a slider with the mouse, the plots are continuously updated. For histograms, this requires that the counts in the histogram bins be refreshed by re-accumulating them with each change of a parameter and that the plot completely be redrawn. However, this is a very fast operation. With 10K rows in the n-tuple data set, the refresh rate is about 2 histograms per second and dominated by the drawing speed. At 100K rows, the refresh rate is about 1 per second and the accumulation time starts to be noticeable. These measurements were made on a NeXTstation color machine which has a 25 MhZ 68040 CPU.

Changing the parameters in this manner is qualitatively different and an order of magnitude faster than a command line driven program. One should recall that a histogram is a means of projecting a frequency distribution onto two dimensions so it can be made visible. But in doing so, one loses some information in the process. If one uses too few bins, one might not see a detail in the data, but one will have good statistics. If one uses too many bins, then the plot might be dominated by fluctuations from bin to bin. There is no rule which says that any one histogram is better than another; it is the judgment of the user that counts. Thus having the ability to very rapidly change the histogram is important. Users of HippoDraw find that they are in much closer contact with the data and can understand it much more quickly, or they find things they did not know were there much more quickly.

One of the sliders for controlling the histogram is rather unique. It controls the offset of the bins by a fraction of the width of a bin. Sometimes, due to poor statistics, an apparent peak in the distribution is caused by accidental accumulation in one bin. By varying the placement of the edges of the bins interactively, one can very quickly see if a peak is real or an artifact of the binning.

The interactive nature of controlling histograms has also been used when applying cuts to the distributions. To apply a cut, the user of HippoDraw first chooses the column of the n-tuple that will be used as selection criteria for a histogram of another column and what kind of cut is to be made (*e.g.*, less than a value). A histogram of the column being used for selection is automatically displayed in the drawing canvas with a shaded region showing what range of values will be accepted. After the cut is applied to the target histogram, the selected range can be changed with sliders or typed into a form field. As the user drags the slider(s), the target histogram is continuously refreshed as well as the shading on the selection display. In Figure 1, the large histogram at the top of the window has two cuts applied to it, which are being displayed at the bottom of the window.

The use of cuts to select entries into distributions is an important tool in analyses. It can also be abused, in that sometimes one *tunes* the cuts to generate peaks that are not really meaningful. With HippoDraw, it is not possible to adjust a cut without having a histogram of the selection variable visible, because the authors feel that users must be conscious of the distribution of the data used for the cut in the region around the cutoff point(s). Also, if as one varies the cut parameters, one sees significant changes in the target histogram, then the one can immediately see what the cause might be by looking at the histogram of the selection variable.

A cut can be applied to one or more target histograms and one or more cuts can be applied to a target histogram. A cut can be applied to a histogram being used to display the selection variable of another cut, in which case, the cut is also applied to that cut's target histogram. Thus the interaction between different cuts can be readily studied. As the cut parameters are changed, all distributions to which that cut was applied are updated continuously. Thus, with HippoDraw, the control of cuts is qualitatively different and an order of magnitude faster than command line driven programs.

There are no commands in HippoDraw because everything can be controlled with the mouse or command-key equivalents. There is also no need for scripts with HippoDraw. When the drawing canvas is saved to disk, the parameters of all the distributions and cuts are saved, as well as a reference to the n-tuple used (or optionally a copy of it.) When this file is read back in, all the distributions and

cuts are re-established. To compare two data sets with a set of histograms and cuts, one only needs to open the drawing canvas linked to one data set, read in another similar data set, and replace the n-tuple used by the plots with the one from the other data set.

The source code of HippoDraw is a library of Objective-C classes of which only one needs to be instantiated when the program is launched. It has been organized so that its entire functionality can be incorporated into another application. For example, an analysis application to create n-tuples can incorporate HippoDraw to view them.

3. Hippoplotamus

Hippoplotamus (or Hippo for short) is an n-tuple management and display package with object-oriented flavor. The package is written in ANSI C and comes in two parts. The first part is the n-tuple manager and it is designed to be user friendly since it is likely to be incorporated in the user's analysis programs. The user only needs to know three functions in the package: for creating an empty n-tuple, for filling it, and for writing it to a disk file. Optionally, the user might use two additional functions for giving the n-tuple a title and for giving labels to the columns. A FORTRAN binding to these functions is provided with the package as well as a utility to convert n-tuples in HBOOK4 format to Hippo format. Utilities to convert files in a plain ASCII text representation to Hippo binary format and vice versa are also provided as part of the package.

The n-tuple files are saved in a machine-independent format. Hippo uses the industry standard XDR[4] format which is widely available on UNIX machines and on other systems as part of a TCP/IP networking package. XDR is also freely available with Sun's RPC 4.0 source distribution. Thus, to transfer a Hippo file between machines of different architectures, the standard FTP program is used in binary mode. Also, files that reside on NFS-mounted file systems can be shared even by machines of different architectures. When XDR is not installed, a plain ASCII text representation of the n-tuple can be used.

The display part of the Hippo package is designed to be friendly to the programmer who implements an interactive n-tuple display program. Thus, it consists of a library of functions to create and manipulate displays in the form of 1D and 2D histograms, x-y plots, scatter plots, and strip charts. These functions are at a rather low level. For example, there is a function to change the number of bins in a histogram display while leaving all other characteristics alone. In HippoDraw, the slider that controls the number of bins is clearly connected to this function. If a Hippo display is a histogram, the accumulation into bins is not done until the function to draw it is called. Functions that would change the accumulation mark the bins as "dirty," so that the accumulation will be refreshed just before the next display.

Hippo also supports the application of cuts on displays. Each display has a null terminated list of functions which determine whether a row should be accumulated or not. Standard cut functions such as less than or within two bounds are provided. The user can also provide his or her own. Also, each display can have a null terminated list of functions that are used to draw a function over the display.

Provided with the package are functions to drive graphics devices. Supported are line printer mode, Display PostScript (for NeXT computers), X-windows under InterViews, X-Windows with Motif, and UNIXPlot (Tektronix 4014 and others). The package does not make use of any higher level graphic packages like GKS; rather it is self contained. For the kind of drawing needed for plotting, a higher level package is not needed. It thus achieves a high degree of portability because it does not depend on licensing of external packages.

The entire Hippo package, including the external utilities, is about 10K lines of code. It has proven to be highly portable using very few conditional compilation statements. It was developed on the NeXT and Sun platforms, but has been tested on IBM RS/6000, DECStation, Silicon Graphics, DEC VAX/VMS, IBM VM/CMS, and Apple Macintosh. Since it is self contained, apart from requiring XDR, it could be used easily for many data collection purposes, such as embedded processors.

4. Conclusions

HippoDraw demonstrates several innovations that lead to a qualitatively different environment for visualizing statistical data. Interacting with the data is much faster with HippoDraw than with command line driven programs. The user is more likely to discover aspects of his or her data because the various changes may be done quickly, while they might not be done at all with a command line driven program. One might say that HippoDraw redefines the meaning of the word interactive when applied to histogramming.

The Hippoplotamus library package also breaks new ground in HEP in that it is the first such package written in the C language and it has an object-oriented flavor. Hippo is easy to use, light weight, and self contained (apart from XDR). It demonstrates the superiority of the C language for this kind of package over the FORTRAN language that has been used in other similar packages.

Acknowledgments

The authors would like to thank Charles Prescott for encouragement and support for equipment purchases. The initial design of Hippoplotamus was done by Jonas Karlsson when he was employed as a summer student. Other contributions to the design of Hippo come from William Shipley and Gary Word. Paul Hegarty of NeXT Computer, Inc. wrote the drawing application (as example code for developers) upon which HippoDraw is based. Contributions to the design, coding and testing of HippoDraw and Hippoplotamus come from David Aston, Don Briggs, George Irwin, Tony Johnson, and Imran Qureshi.

References

[1] E. Bassler, *Comput. Physics Commun.* **45** (1987) 201.

[2] T.H. Burnett, *Comput. Physics Commun.* **45** (1987) 195.

[3] R. Bock, et al, *Comput. Physics Commun.* **45** (1987) 181.

[4] Sun Microsystems, Inc., *XDR External Data Representation Standard*, RFC1014, (see also man pages on UNIX systems).