SLAC-PUB-5922
September 1992
(E/I)

HippoDraw*

**Michael F. Gravina, Paul F. Kunz,
and Paul E. Rensing**

**Stanford Linear Accelerator Center
Stanford University
Stanford, CA 94309, U.S.A.**

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

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 along with some user impressions of these features. 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[4], which is written in the ANSI C language, was used.

2. Basic Features

Figure 1 is a screen image with HippoDraw running. There is a large window in the center which is the drawing document and a smaller window in the lower right called the inspector panel. This is where the "Draw" of HippoDraw comes in, because it is a simple drawing program like one might find on popular personal computers. Drawing tools can be selected from the small

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

Presented at the conference on Computing in High Energy Physics, Annecy, France, September 21-25, 1992.

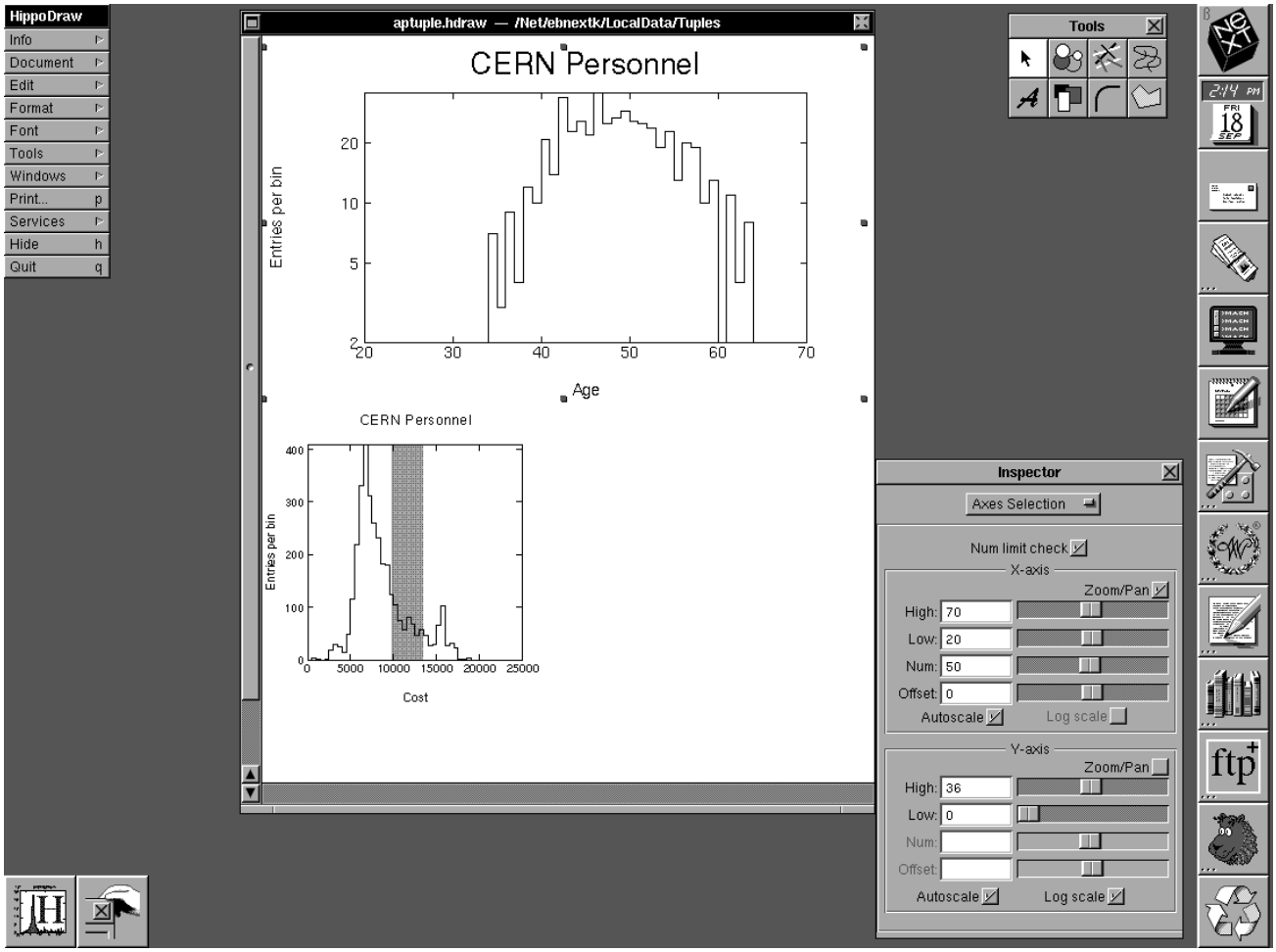

Fig. 1 Screen image showing HippoDraw windows.

panel in the upper right of the screen. Histogram plots and other data displays are treated like drawing objects in the drawing document. The inspector panel is used to display and change attributes of plots and other graphic objects. The design philosophy of HippoDraw was on the one hand to keep this panel small so it doesn't obscure the draw document, while on the other hand to make all attribute changes easily accessible. Thus, it is actually six sub-panels with only one being displayed at a time. The others are reachable via the pop up button near the top of the panel.

The first step a user takes in using HippoDraw 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.

The next step is to create and display a distribution. One of the inspector panels displays a browser with the labels of the n-tuple's columns. There are also 8 buttons to select what kind of data display is desired and in the case of 2D displays, which data columns will be used for which axis. A new plot is created by clicking on a *new plot* button. The association of data columns to axis can be changed at any time with a simple point and click operation. For each

such change, the selected histogram is thrown away and a new one put in it's place while keeping the display attributes intact. This operation is very fast, thus allowing the user to quickly browse the n-tuple columns.

Once a plot has been created, it can be moved within the drawing document 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. HippoDraw users make use of the drawing features to arrange and annotate plots for their private study, for presentation to collaborators or at conferences, or to make camera ready copies for publication.

A new plot is inserted onto the drawing canvas with a default set of parameters. For example, histograms are created with 50 bins and scaled so all data is visible. 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 redrawing the plot completely. 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 is 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 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 than with tradition applications.

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. Another feature is to change the action of the upper and lower range sliders into zoom and pan sliders. When operating in this mode, the user can quickly center a peak in the display then expand around it.

3. Other Features

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.*, within a range). 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 a cut applied to it, which is 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). Thus, if as one varies the cut parameters and 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. This leads to a quicker understanding of the data set.

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.

Multiple cuts on target histograms are effectively ANDed together. To be able to do more complex cuts and to allow display of data that can be derived by calculations on the n-tuple, HippoDraw allows functions that can be dynamically linked with the running application. These functions are given one row of the n-tuple as an argument and return the contents of that row for a new n-tuple column. The new column can be used for display or for a cut variable.

Two of the display types in HippoDraw are based on a 3D view of the data. They are the traditional lego plots and 3D scatter plots. Each of these have a user controlled perspective. It is felt that with perspective, users can better sense what data points are in the foreground versus the background. This is especially true as 3D plots are rotated since foreground points move faster on the screen than background points. The user also seems to be able to retain for a longer period of time this information than is the case with the traditional 3D views.

There are no commands in HippoDraw and there is no need for any 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. The authors are aware of six such applications and there may be more.

4. Conclusions

HippoDraw demonstrates several innovations that lead to a qualitatively different environment for visualizing statistical data. HippoDraw also attempts to be as easy to use as a *shrink-wrapped* application one might find on a personal computer. To test both the innovations and the ease of use, HippoDraw was made available by anonymous ftp in February of 1992. It has since gained widespread use both inside and outside of HEP. For example, HippoDraw is being used by academics in the field of economics and biology; it is even being used by a financial planning firm and other companies.

All users seem to agree that interacting with the data with HippoDraw is much faster than with command line driven programs. The user is more likely to discover aspects of his data because the various changes may be done quickly, while they might not be done at all with a command line driven program. Users also feel that the Draw aspects of HippoDraw are important as well, since it allows one to easily organize the displays for analysis or presentation. One might say that HippoDraw redefines the meaning of the word interactive when applied to data analysis.

Acknowledgments

The authors would like to thank Paul Hegarty of NeXT Computer, Inc. who wrote the drawing application (as example code for developers) upon which HippoDraw is based. Contributions to the design, coding and testing of HippoDraw come from David Aston, Don Briggs, Walter Innes, George Irwin, and Imran Qureshi.

References

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

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

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

[4] M. Gravina, et al, *SLAC-PUB-5921*, SLAC (1992).