GRAPHICS INTERFACES FOR MIDI-EQUIPPED SYNTHESIZERS

Przemyslaw Prusinkiewicz
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada

ABSTRACT

This paper explores methods of controlling synthesizer parameters based on manipulating graphics objects on a bit-mapped screen. A variety of computer generated graphical controls is proposed. Programs implementing graphical controls on a Macintosh microcomputer interfaced with a DX7 synthesizer are outlined.

1. INTRODUCTION

"Mathematics and electronics afford us increasingly powerful means of dealing with data. But at the same time they multiply the number of arbitrary choices without changing our natural means of perception the slightest. It thus comes down to utilizing these natural means the best way."

Jacques Bertin (1983)

"The most important factors in the computer's seduction have to do with the specificity of the computer as a medium to support the desire, the needs and in extreme cases the obsession for 'perfect mastery'."

Sherry Turkle (1984)

Creating new voices (patches, instruments) using many contemporary synthesizers is difficult. There are several reasons for this. The number of controllable parameters is large — often more than one hundred. This is intended to maximize the flexibility of the instrument. At the same time, the number of physically manipulable knobs (or slide potentiometers) is kept small in order to minimize the instrument cost. Consequently, one knob can be used to input values of many parameters. This creates two problems. First, a parameter cannot be immediately accessed — a knob must be assigned to a parameter before its value can be changed. This routing may involve changing the state of several switches. Second, it is no longer possible to discern current values of parameters just by looking at the control panel because the assignment of knobs to parameters is not fixed. At the same time the user needs a particularly clear, well-organized and comprehensive access to data because of the large number of parameters involved. The need for a good user interface is reinforced if some parameters have a nonintuitive character and their impact on the final sound is difficult to conceptualize. If parameters cannot be made intuitively obvious, they should at least be easily controllable in order to facilitate experimentation.

In contemporary MIDI-equipped synthesizers, all or most parameters can be remotely controlled by a computer. This opens a world of new interactive techniques applicable to the design of interfaces between man and synthesizer. The purpose of this paper is to present some techniques based on manipulating graphical objects on a bit-mapped screen.

The idea of of using a graphics interface to control sound synthesis parameters is not new. Buxton et al. (1982) described the application of "graphic potentiometers", bar graphs and function plots for this purpose. A graphics interface was also included in the console for the Lucasfilm audio signal processor (Snell 1982). However, the growing availability of workstations and microcomputers with powerful interaction-handling hardware and software encourages further research in this area. Due to the decreasing cost of computer hardware, graphics interfaces need no longer to be confined to expensive computer music studios. They may become the standard tool for controlling commercially available synthesizers.

This paper presents five approaches to the design of graphical interfaces between man and synthesizer. They were implemented on a 512K Macintosh (*) controlling a DX7 synthesizer. An Apple IIe (**) with a Roland MPU-401 MIDI Processing Unit provided the RS-232 - MIDI interface (Fig. 1). (For debugging purposes, the exchanged messages were also displayed on the Apple IIe screen.) New controls were first drafted using MacPaint (**) and after a satisfactory appearance was achieved, they were implemented in software. All programs were written in C (Manx 1983, 1984, 1985) and rely heavily on the Macintosh firmware (Apple 1984, 1985).

(*) Macintosh is a trademark licenced to Apple Computer, Inc.
(**) Apple IIe and MacPaint are trademarks of Apple Computer, Inc.

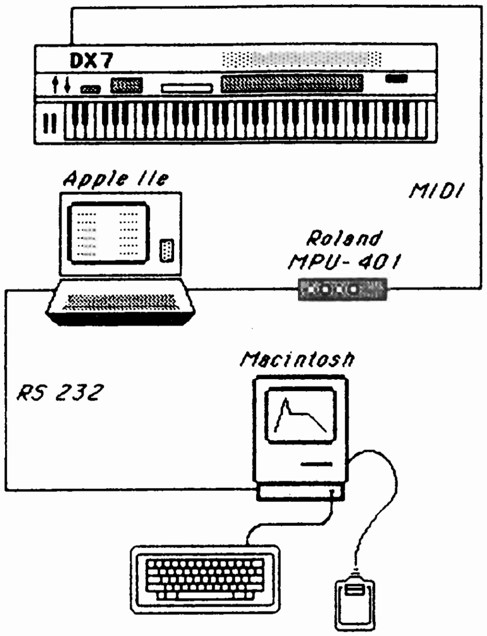

Fig. 1. The environment used for experimentation.

2. VIRTUAL FRONT PANELS

The most straightforward approach to the design of graphics interfaces for synthesizers uses graphical objects which visually resemble physical controls: bistable and multiposition switches, slide and rotary potentiometers, VU-meters etc. These elements are assembled to form an *electronic metaphor* of the front panel of a synthesizer. Like "real" panels, the virtual ones may contain a large number of controls (in the order of hundreds). At the same time the number of different types of controls is small (usually less then ten). Consequently, the actual definition of the panel from predefined controls can be left to the user. This idea has been implemented in a program named UCofA (a Universal Controller of Anything). The user has at his disposal a menu of three types of controls: slide potentiometers, rotary potentiometers and bistable switches. Copies of these items can be picked from the menu and placed in several windows, each representing a portion of the control panel (Fig. 2). At any time, a *dialog box* (Apple 1984, 1985) associated with a control can be opened, allowing for the interactive definition of the control name, the minimum and the maximum value of the controlled parameter, and the format of the MIDI message to be sent when the control is manipulated.

The main advantage of user-definable virtual control panels lies in their flexibility. A program such as UCofA can be used to control different types of synthesizers. Furthermore, a user can configure a panel in the way he finds the most convenient. The visual resemblance between real world controls and their electronic metaphors on the screen contributes to the intuitive character of the interface and helps in understanding its operation.

3. BAR GRAPHS

Graphical forms of virtual controls resembling real knobs, switches etc. are limited. A departure from the

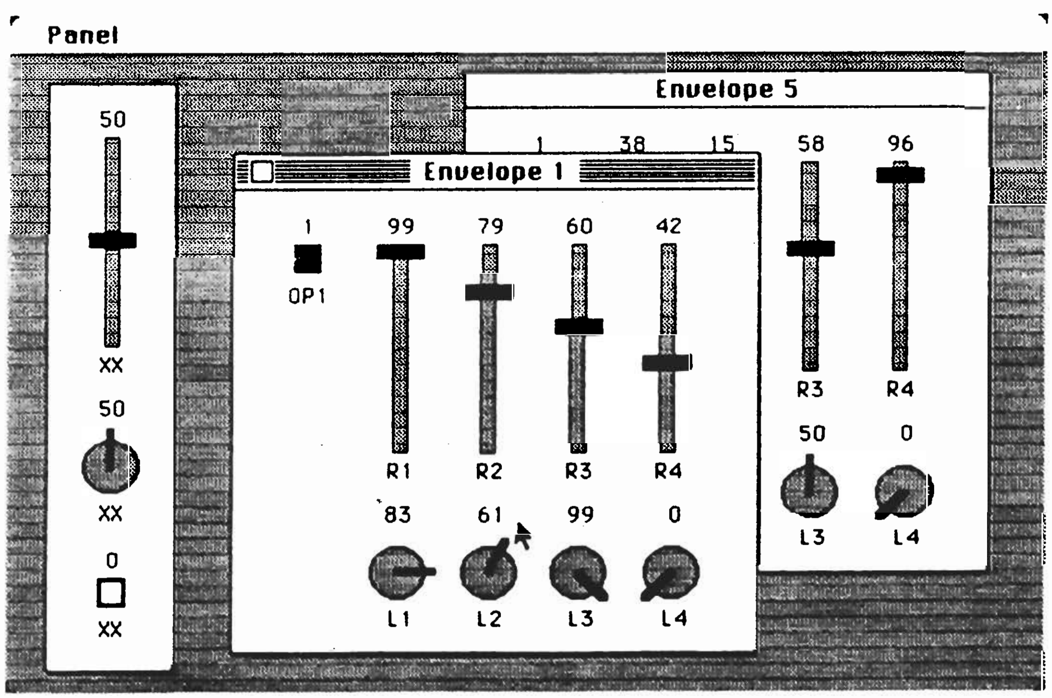

Fig. 2. Example of the UCofA screen.

mimetic design brings more freedom into the choice of control forms. This allows for an improved data presentation.

Displaying values of synthesizer parameters is a special case of graphical presentation of multiple variables. Commonly used methods in this area include different forms of bar graphs and pie charts. In musical applications, bar graphs are particularly suitable for presenting series of logically related parameters, such as amplitudes of harmonics or output levels of a number of signal sources (Fig. 3). They also allow for convenient data entry because it is intuitive to point to a bar and make it higher or lower using a mouse. (However, the zero value requires special attention - a bar should not disappear without leaving a trace to allow subsequent manipulation).

Fig. 3. Example of a bar controller.

There are many details which can make bar graphs more or less legible. Extensive studies by Bertin (1977, 1983) provide a valuable source of reference when designing bar controllers.

4. BERTIN DIAGRAMS

If parameters do not form a series of logically connected quantities, their inclusion in a single bar graph may lead to an illegible presentation. On the other hand, breaking data into several graphs creates the problem of their layout, and makes the relationships between parameters in different graphs obscure. Bertin suggests that in many cases a legible presentation can be achieved by arranging bar graphs into a two-dimensional array. For example, in the case of the DX7, such an array can be used to represent all eight parameters of the amplitude envelopes of the six operators (Fig. 4a). All parameters belonging to the same operator are arranged in one row, and the corresponding parameters of all operators are placed in one column. Consequently, the diagram reflects two types of relationships between parameters.

Bertin perceives the two-dimensional diagrams not only as a tool for representing multivariate data, but also as a method for exploring their relationships. To this end, he proposes to permute rows and columns of a diagram in such a way that the elements of maximum values would be grouped along the main diagonal of the array

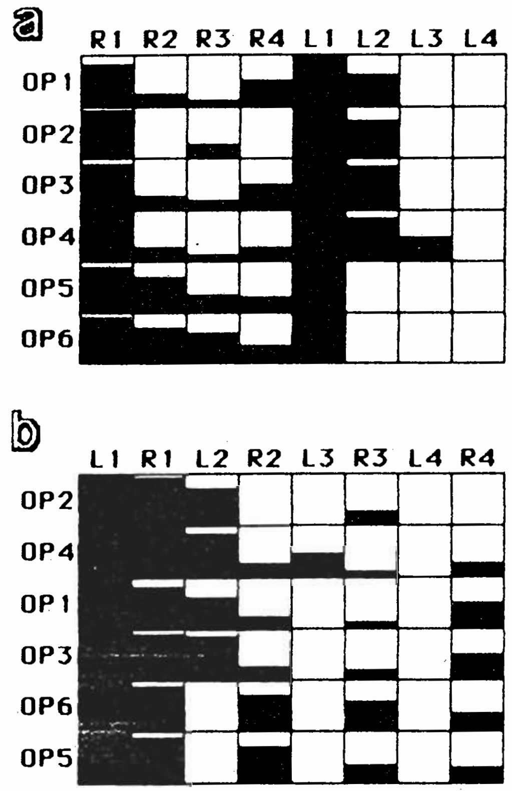

Fig. 4. Using Bertin diagrams to control DX7 amplitude envelopes.

(Fig. 4b). In the case of statistical data, these permutations lead to a more legible presentation and help in the understanding of the essential relationships between data assigned to the rows and columns. It is not obvious whether permutations are of the same cognitive value in the case of diagrams representing synthesizer controls. Still, they provide a simple mechanism which allows the user to interactively define the layout of controls in the way he perceives as the most convenient and logical.

Bertin diagrams are intended to represent values of many variables in one picture, thus facilitating conceptual associations between data. However, a computer implementation of these diagrams is limited by the size and resolution of the screen. For example, the maximum height of the bars in Fig. 4a and 4b equals 25 pixels. As a result, a bar can represent only 25 different values. In many cases this is not enough. Two methods for increasing the number of representable values without increasing the size of the diagram are shown in Fig. 5c and 5d. They are based on coding the additional information using the texture or the shape of the bars. Bigger values are represented by bars which are relatively darker or wider. In both cases the bar actually shown on the screen can be thought of as a partial view of a longer bar placed behind the Bertin diagram and clipped to the dimensions of one cell. This conceptuali-

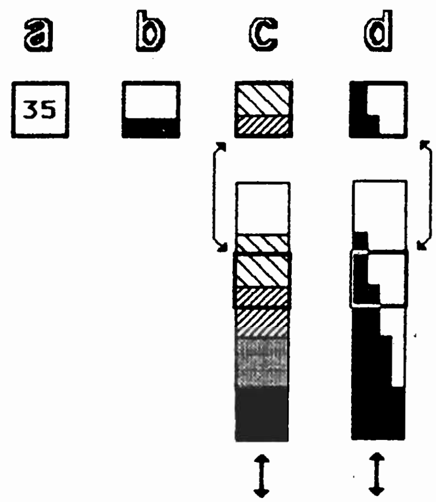

Fig. 5. Parameter representations implemented in the envelope controller based on Bertin diagrams. (a) Numerical values. (b) Standard bars. (c, d) Modified bars - the actual representations can be thought of as clipped views of longer bars.

zation is particularly convenient when manipulating a parameter with a mouse. The user has a consistent feeling of moving a long bar. It is not important that this bar is only partially visible.

An envelope controller for DX7 based on Bertin diagrams has been implemented in software. The user can interactively permute rows and columns, and select one of the four viewing modes available (Fig. 5).

5. FUNCTION PLOTS

Graphical forms of controls discussed up to this point are not closely related to their functions. In many cases, however, controls corresponding to a conceptual model of parameters' functions can be designed. For example, parameters defining an envelope are usually explained and conceptualized by referring to a function plot. Thus it is intuitive to control them by directly manipulating a plot on the screen. An implementation of the DX7 amplitude envelope controls which follows this approach is shown in Fig. 6a. The user modifies the envelope by dragging the break points, represented by the solid squares, with a mouse. (The hollow squares do not correspond to independent parameters and therefore cannot be dragged). The frequency envelope can be controlled in a similar way (Fig. 6b). However, its plot indicates that the frequency deviation can be positive or negative with respect to the frequency of reference.

Since a point in the plane has two degrees of freedom, its position can be used to control two independent parameters. The attachment of synthesizer parameters to the abscissa and the ordinate of a break point presents the simplest approach, but is confusing in the case of DX7 envelopes. Their parameters ("rates") do not specify time intervals directly. Instead, they define how fast the value of an envelope increases or decreases. Consequently, these parameters should be associated with the slopes of the envelope segments rather than with the coordinates of any particular points. However, given the limited screen resolution, the slopes cannot be used to enter nor represent rate values with the required precision (any integer value from 0 to 99). To remedy this situation, a modified graphical representation of envelopes was designed and

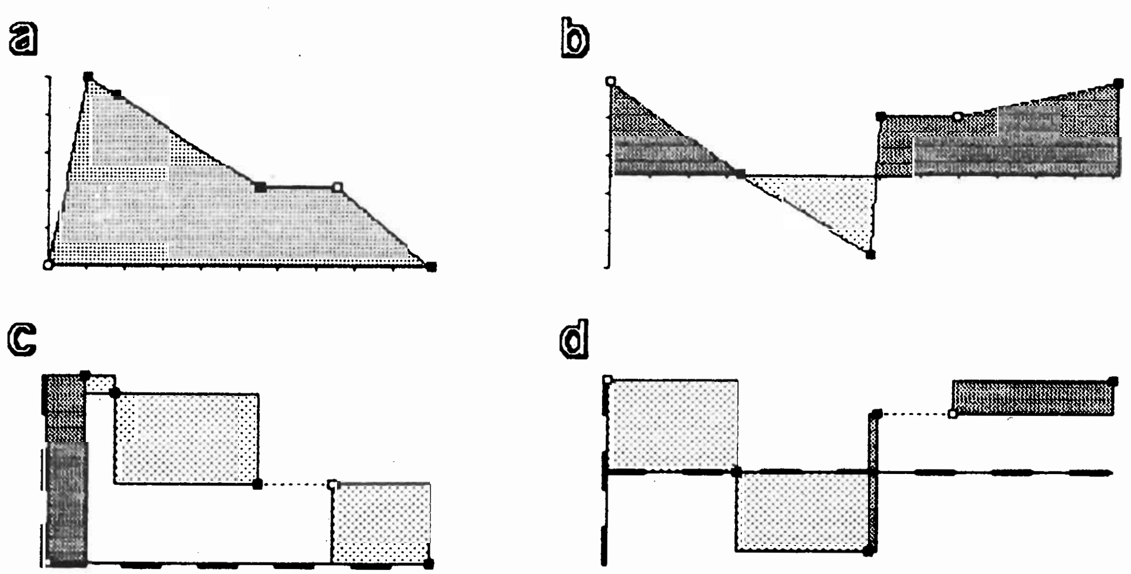

Fig. 6. Using function plots to manipulate DX7 envelopes. (a, b) Controls based on the usual representation of the amplitude and frequency envelopes. (c, d) Controls based on the modified representation of envelopes.

implemented (Fig. 6c and 6d). This design substitutes the usual line segments connecting consecutive break points by rectangles. The height of each rectangle represents the difference of envelope levels between two break points. The width represents the rate of transition between these levels. The area represents the transition time. Additionally, the pattern filling each rectangle indicates whether the envelope value increases or decreases during a particular transition. Because envelope levels and rates are represented directly as the coordinates of the break points, the resolution of the Macintosh screen is sufficient to control these parameters with full precision.

Function plots are also suitable to control sets of parameters other than envelopes. For example, in the DX7 they can be used to specify how the output level of an operator varies across the keyboard (i.e. in the function of the key pressed).

Function plots provide a very understandable method for controlling sets of interrelated parameters. They tend, however, to occupy a substantial area of the screen. Consequently, only a limited number of parameters can be simultaneously represented and manipulated. A global, comprehensive view including many parameters is difficult to achieve.

6. CHERNOFF FACES

Graphical forms of controls discussed in the previous section correspond with their functions expressed in technical terms. They refer to the physical aspects of a particular synthesis process. However, a musician is interested mostly in the final result, i.e. the synthesized sounds. Consequently, the relationship between graphical forms of controls and the actual sounds may have a more abstract character, deemphasizing the technical aspects of synthesis and encouraging free associations between sounds and pictures. These pictures may be quite arbitrary, providing that they represent multiple parameters legibly, and can be easily memorized. Several types of pictures satisfying these conditions were actually developed in the scope of multivariate analysis. The most popular are Chernoff faces (Chernoff 1973, 1975, Wang 1978). They represent up to 18 variables by such face features as the length of the nose, the vertical position, width and the curvature of the mouth, the separation and the size of the eyes, etc. Other types of pictures including trees, castles and asymmetric faces were also proposed as a tool for representing multivariate data (Kleiner & Hartigan 1981, Flury & Riedwyl 1981).

If a face is used not only for data display but also for data entry, all of its features have to be easy to manipulate with a mouse. A simple experimental face satisfying this requirement is shown in Fig. 7.

Faces, castles and other "exotic" controls may be particularly attractive for synthesizer users with no technical background (for example children).

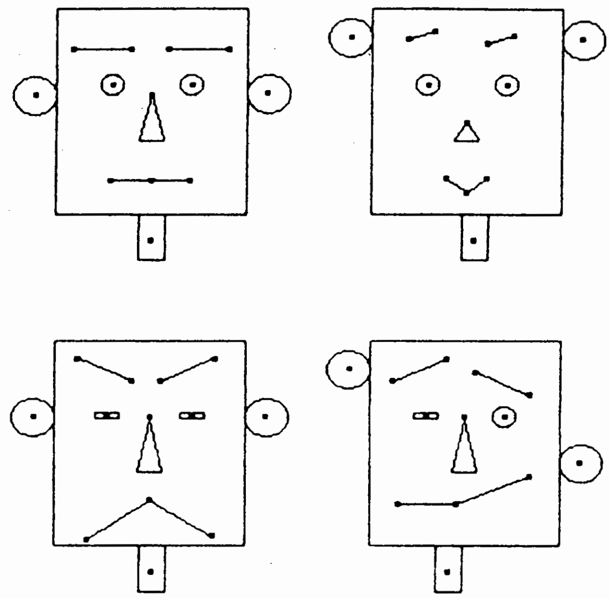

Fig. 7. An experimental face used to control a set of parameters. The manipulable face features are indicated by small squares.

7. CONCLUSIONS

A variety of computer generated graphical controls have been proposed. In their design, methods for multivariate data representation used in business applications, geography, statistics etc. were considered. Programs implementing graphical controls on the Macintosh microcomputer interfaced with the DX7 synthesizer were outlined. These programs are relatively simple due to the graphical and interaction-handling routines included in Macintosh firmware. The environment and the example controls described in this paper may be useful in a further study of human factors involved in the design of graphical interfaces for synthesizers.

ACKNOWLEDGMENT

This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Apple (1984, 1985). *Inside Macintosh.* Cupertino: Apple Computer Inc.

Bertin, J. (1977). *La Graphique et le Traitement Graphique de l'Information.* Paris: Flammarion.

Bertin, J. (1983). *Semiology of Graphics.* Madison: The University of Wisconsin Press.

Buxton, W. *& al.* (1982). Objed and the design of timbral resources. *Computer Music Journal 6 No. 2, 32-44.*

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68, 361-368.

Flury, B. & Riedwyl, H. (1981). Graphical representation of multivariate data by means of asymmetrical faces. *Journal of the American Statistical Association* 76, 757-765.

Manx (1983). *Aztec C65 Manual (Version V1.05b)*. Shrewsbury, N.J.: Manx Software Systems.

Manx (1984, 1985). *Aztec C for the Macintosh (Version 1.06D)*. Shrewsbury, N.J.: Manx Software Systems.

Snell, J. (1982). The Lucasfilm real-time console for recording studios and performance of computer music. *Computer Music Journal* 6 No. 3, 33-45.

Turkle, S. (1984). *The Second Self*. New York: Simon and Schuster.

Wang, P.C.C. (1978) (ed.). *Graphical Representation of Multivariate Data*. New York: Academic Press.