THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 56, No. 3, March 1977
*Printed in U.S.A.*

# Computer Displays Optically Superimposed on Input Devices

By K. C. KNOWLTON

(Manuscript received August 3, 1976)

*A set of pushbuttons on a console may appear to have computer-generated labels temporarily inscribed on them if the button set and computed display are optically combined, for example, by means of a semitransparent mirror. This combines the flexibility of light buttons with the tactile and kinesthetic feel of physical pushbuttons; it permits a user to interact more directly with a computer program, or a computer-mediated operation, in what subjectively becomes an intimately shared space.*

*A console of this design can serve alternately as a typewriter, computer terminal, text editor, telephone operator's console, or computer-assisted instruction terminal. Each usage may have several modes of operation: training, verbose, abbreviated, and/or special-privilege. Switching from one mode or use to another is done by changing the software rather than hardware; each program controls in its own way the momentary details of visibility, position, label, significance, and function of all buttons.*

*Several demonstrations are described, including a prototype of a proposed Traffic Service Position System (TSPS) console, and an interactive computer terminal resembling a Picturephone® set with a Touch-Tone® pad. Also suggested are combinations of computed displays with* x-y *tablets and other input devices.*

In interactive use of computers, a large number of advantages result from virtually superimposing the computed display on an input device such as a two-dimensional array of pushbuttons.[1,2] A display so arranged can be used effectively to label buttons or relabel them with new meanings; indeed the buttons themselves may seem to appear and disappear according to their momentary significance or nonsignificance to the program. The same composite console—display plus input device—may have vastly different uses depending on the program that labels buttons and reacts to them. Thus combined are complete flexi-

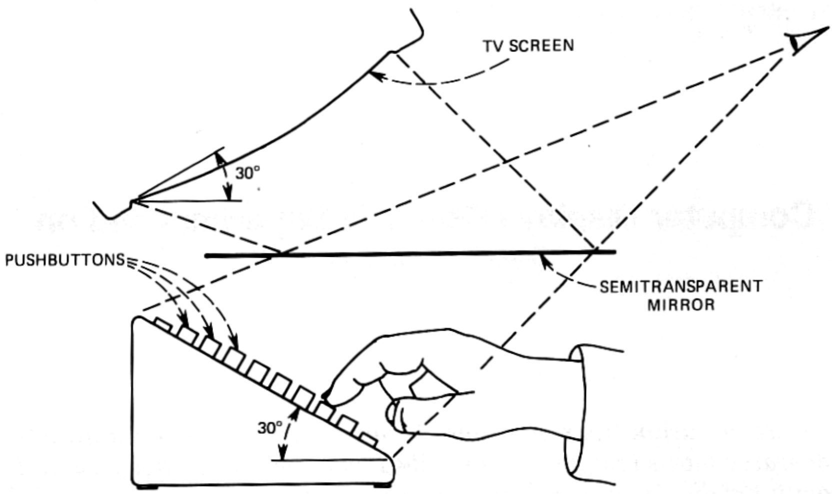



Fig. 1—Basic arrangement for superimposing a computed display on a two-dimensional array of buttons.

bility, normally associated with light buttons, and the tactile and kinesthetic feel of physical buttons that move, as on a typewriter. A button set may thus "be" a typewriter, calculator, telephone operator's console, computer-assisted instruction terminal, or music keyboard. An $x$-$y$ tablet or other two-dimensional input device may likewise have a computed display superimposed on it. In all cases, the user enjoys a sense of close interaction with the computer in an intimately shared input-output space.

## I. BASIC PRINCIPLES

A straightforward way of superimposing a display on a button set involves a semitransparent mirror, as illustrated schematically in Fig. 1. The user looks through the mirror and views his/her hand directly as it pushes buttons. The display—a television monitor in the illustration—and mirror are so arranged that the virtual image of displayed light buttons conforms in three-dimensional space exactly with the position of the physical keytops. Perceived spatial congruence is so precise that if the display is a bulging TV screen, then it is best for the button tops to conform to a convex envelope, as in Fig. 1, so that the central buttons do not seem too soft (i.e., so that the finger meets the physical button top at exactly the same depth as the image position). When displayed image, button set, and mirror are properly aligned, there is no parallax effect and bystanders perceive interactions exactly as the user does. In fact, the actual buttons need not be seen—it is best if they are painted a dull black so that they seem to disappear when the corresponding light

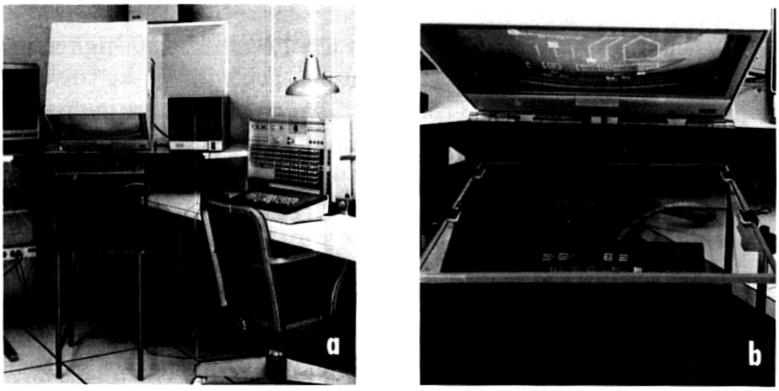


Fig. 2—(a) Laboratory setup for experimenting with displays superimposed on button tops. (b) Closeup of TV screen, mirror, and 12 by 10 set of buttons.

button is extinguished. For proper hand-eye coordination, the user does want to see his/her hand; therefore, lighting from the side is useful. If it is strong enough, then the mirror used need be only slightly transmitting (say 10 percent) and may thus be highly reflecting (say 75 percent) to maintain high visibility of the virtual display. Ambient light is no problem except that strong room illumination, direct or reflected, should be kept off the display screen.

The display, of course, needs to be generated upside down so that it appears right side up when viewed in the mirror. This is no fundamental problem except that one commonly imports or implements software in which the assumption of right-side-up generation (of alphabetic characters, for instance) may be embedded deep in the code.

One curiosity of these systems is that the hand seems transparent to light buttons, since it does not intervene in the path of reflected light. We can read through our fingertip the current label of the button pushed, as well as see other buttons beneath the hand. This is not in the least confusing to a user who has been at the machine for a few seconds; on the contrary, it is definitely helpful not to have to remove your hand to see what's beneath it.

Figure 2a is a photo of one generally useful laboratory prototype for experimenting with usages of virtual pushbutton consoles; Fig. 2b is a closeup of display, mirror, and button set as seen from farther away and lower than the user's normal head position. The computer used has 32K 24-bit words of core storage; programming is done in FORTRAN and an assembly language. The display is a normal 525-line TV monitor, with separate red, green, and blue (RGB) inputs, refreshed 30 times per second by specially built hardware from a separate core memory that holds 3 bits per picture cell.[3,4] (The displayed picture is only 496 lines of 528 pixels per line.) Each of the eight logical colors is program-definable to 128 levels per primary. The button set is a 12-wide by 10-high array of pushbuttons, ½-in. square on 1-in. centers, each with 3/16-in. travel. The computer reads only rows and columns in which buttons are momentarily depressed—all single hits are clearly decodable, as are multiple hits in the same row or column and some patterns produced by progressively adding buttons. The mirror is 16 in. square by ¼ in. thick; it is first-surface 75 percent reflecting and 10 percent transmitting.

## II. PROTOTYPE FOR A TELEPHONE OPERATOR'S CONSOLE

The setup of Fig. 2 has been used to implement an experimental demonstration of a flexible telephone operator's console—in particular, a possible replacement of the present Traffic Service Position System (TSPS) station and/or future versions of it.[5] Figures 3 and 4 illustrate many of the features that such a console might have. In the demonstration, button tops are ½-in. green squares, containing green labels; lines connecting logical groups of them are blue—sometimes these alone appear where an entire set of button tops has vanished in order to preserve a sense of orientation and geography. Lights at the very top of the board, indicating what type of call is presently being processed, are red, as are occasional wide frames around buttons, pointing out mandatory operator actions.

The sequence of Figs. 3a through 3e illustrate the handling of a particular call, which is being charged to a third phone. Figure 3a shows the board before the call arrives, with only a few buttons present, indicating the limited number of things an operator can take initiative on with no call to process, such as to inquire as to the time of day (lower left). In Fig. 3b, a "zero +" call has arrived from a non-coin phone (the calling party has dialed the called number, but asked for operator assistance by the leading zero). The operator asks "May I help you?" and the calling party requests that the call be charged to a third phone, whereupon the operator prepares to push the special calling (SPL-CLG) button. With the class of charge thus declared (Fig. 3c), this button lights brightly and other class-of-charge buttons disappear; also, the key pulse special (KP-SPL) button lights with a red frame, in effect insisting that the operator enter a third phone or credit card number. (Notice that the KP-SPL with red frame can be read through the hand.) When the operator pushes this button, the keyset appears and is used for entering the third phone number (Fig. 3d). The number entered appears in the center of the panel and the SPL-NO display button appears, indicating that henceforth there is a special number, which may be redisplayed at a later time. The ST-TMG button with red frame means that no more information is needed; if it is pushed, the telephone machinery may start timing the call as soon as the called party answers the phone. When this button is pressed, it

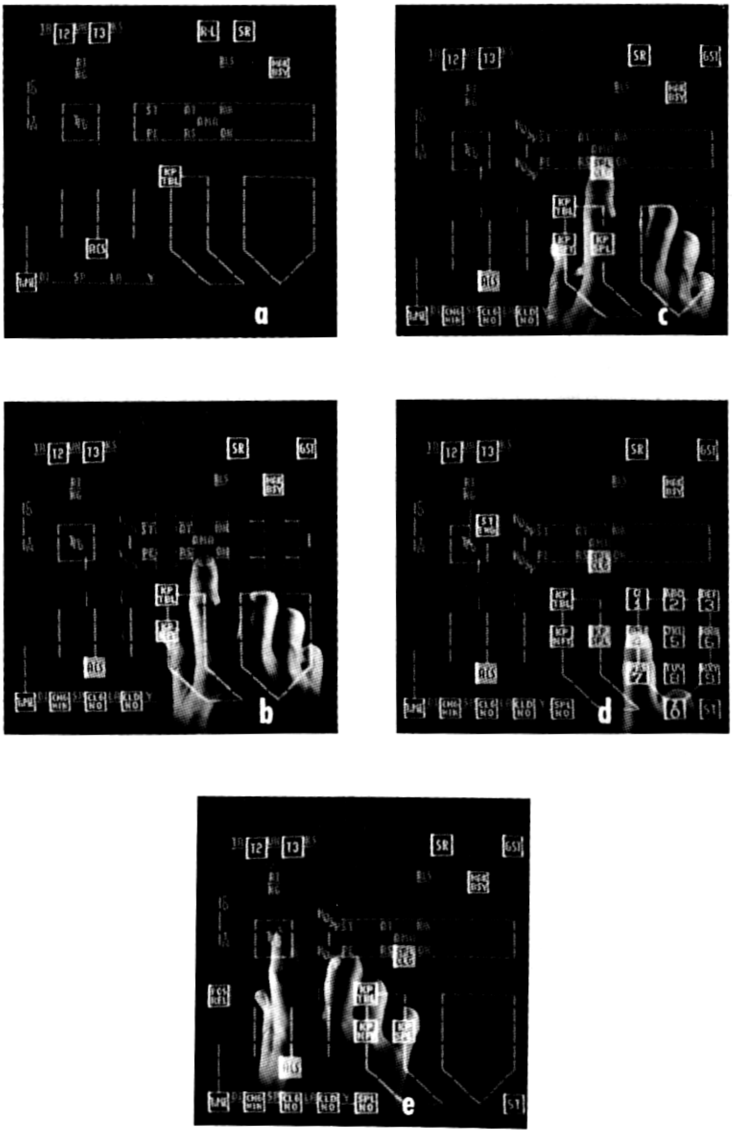


Fig. 3—Demonstration of a charge to third phone. (a) Quiescent board. (b,c) Operator pushes SPL-CLG button. (d) Third phone number is keyed in. (e) ST-TMG button pushed, permitting POS-REL.

disappears (Fig. 3e) and the POS-REL button appears, permitting the operator to release the call from this position, whereupon the board reverts to the quiescent state of Fig. 3a.

The sequence in Fig. 3 shows that this console is dynamic even during the processing of a phone call; all of the buttons that have meaning at any moment, and only those buttons, are visible. The sequence in which they appear, in fact, tends to lead the operator through the required series of decisions and actions; this should be a significant help in the training of new operators (there could be a verbose mode, or a HELP button to spell out in words or phrases the meaning of buttons or situations encountered.)

Figure 4 illustrates more features of the TSPS demonstration. For the sake of comparison, Fig. 4a shows the complete set of buttons corresponding approximately with the currently used board. It is not immediately obvious here that the KP-TBL (meaning, "prepare to key in a trouble code") is one of the few meaningful actions when no call is present. (Figure 4d, on the other hand, is the recommended appearance of the quiescent board, with only valid buttons showing; here, the operator is inquiring as to the time, which is displayed while the TIME button is pushed.) Figures 4b and 4e compare the left- and right-handed boards; a left-handed operator presumably will want the keyset for entering phone numbers, etc., on the left. (Programming, as one should expect, is done in terms of logical buttons—*where* each button happens to be at any time is a matter of mapping. Individual operators might even be allowed to make their personal rearrangements of the board.) Figure 4c shows the entire board temporarily turned into a typewriter keyboard for possible future applications requiring alphanumeric input. Figure 4f repeats something that already appeared in Fig. 3: a call originates from a patient at a hospital, an institution which does not want to be the collecting agent for phone calls and therefore requests the phone company not to let the call be billed to the calling phone. Thus, in the spot where the operator might have expected a class-of-charge button PAID (by the calling phone) there is an explanatory note, HOSP, which neither appears nor functions like a button.

The key word is flexibility, including the option of introducing new buttons for new services or functions. All such alterations, including adding, modifying, rearranging, relabeling, or deleting buttons, are changes in *software;* they would be much easier to implement on any or all of the consoles than would be equivalent changes in hardware (once the changeover has been done).

One should finally note that, for the handling of phone calls, a great deal of electronic equipment is already needed, including circuitry and other means for detecting and decoding button pushes. The significant addition suggested by the present prototype is that the main call-handling mechanism could tell the console what buttons to light and extinguish, and where. In other words, the console would show what button presses are legal, something which is already implicit in the program. Operators would be less likely to do things out of order, simply because

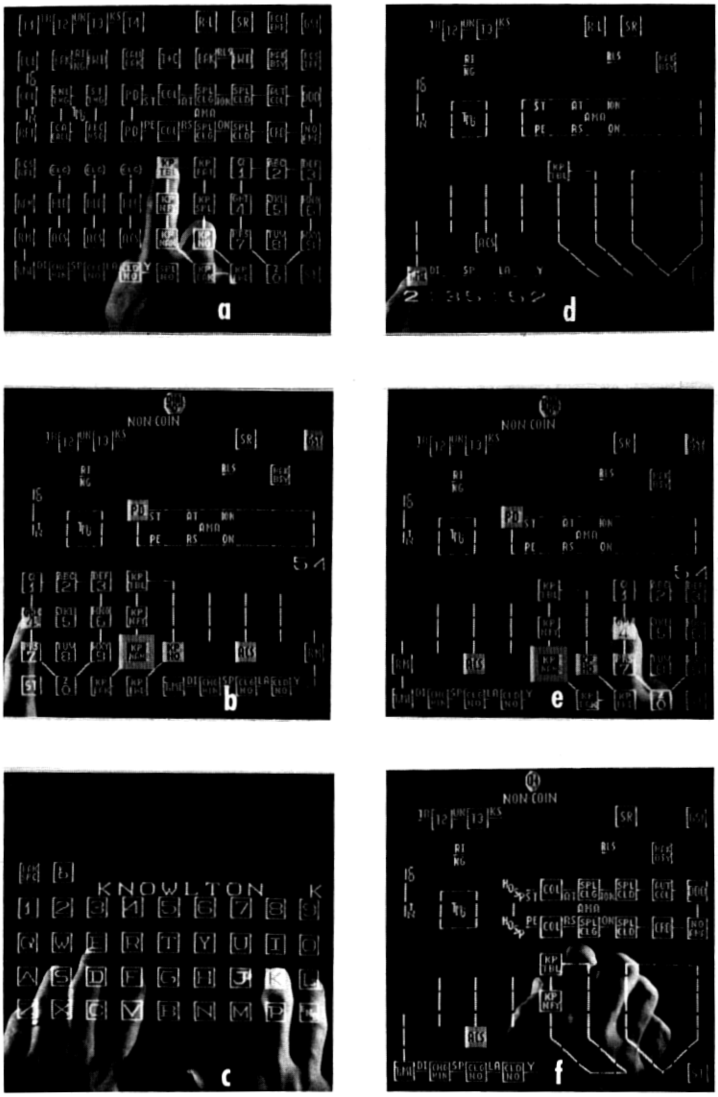



Fig. 4—Features of TSPS demonstration.

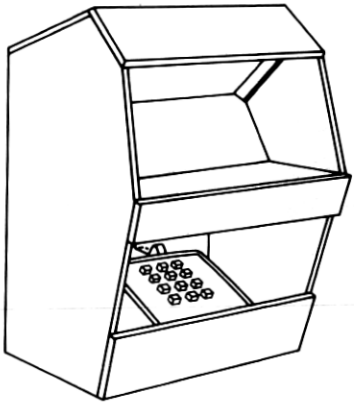
they would not expect anything to happen in response to pushing an unlighted button. A simple case in point is : if the start timing (ST-TMG) button is not present but the class-of-charge panel is entirely illuminated, it should be immediately obvious even to the beginner that the class of charge still needs to be declared (perhaps among other things) before it is legal or possible to start the timing of the call.

## III. A RELABELABLE "TOUCH-TONE" PAD AS AN INTERACTIVE CONSOLE

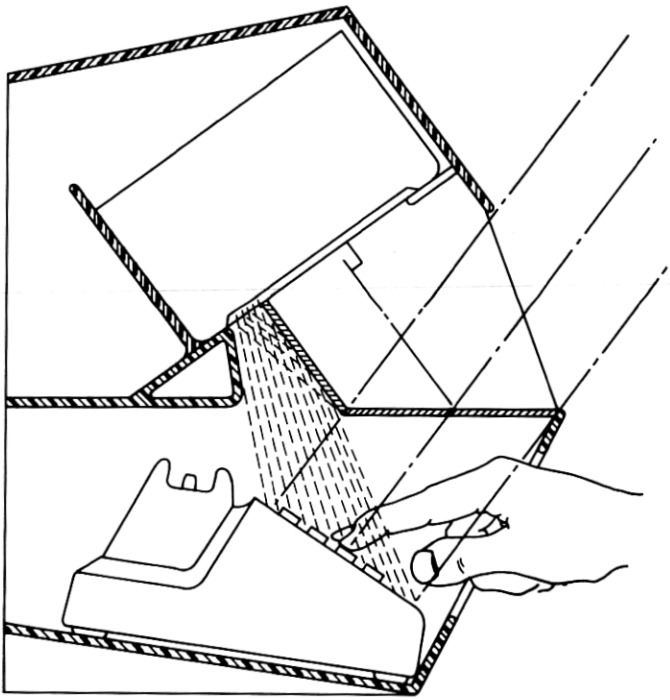
A set similar to a Picturephone set could be used as an interactive remote computer console, with the Touch-Tone pad as the input keyboard. A schematic for a mockup is shown in Fig. 5, where a semitransparent mirror effectively puts the computed image on the Touch-Tone buttons so that the 12 buttons can have several labelings and a correspondingly extended range of functions. A proposed new feature, as illustrated in Fig. 5b, is the use of the bottom quarter of the screen as a light source for illuminating the hand, but only when function buttons are displayed, not when some other graphic program result is being shown. In the latter instance, when the hand should not be seen, the screen "light" goes off. A partial cabinet hides the hand from room light.

Such a console has been built using a computer, a Touch-Tone pad as the keyboard, and a color television RGB monitor so modified that each of its three color signals is essentially a Picturephone signal.[6] The picture is again 3 bits per pixel with a total of 254 lines, 240 pixels per line. The front-surface mirror is 45 percent transmitting, 45 percent reflecting, with an antireflective magnesium fluoride coating on the second surface. Figure 6a shows a distant view of the button set and (inverted) display, whereas Fig. 6b shows the user's view for this same circumstance. These buttons are not black, yet the computer-generated image effectively obliterates the intrinsic labels on the buttons to the extent that one could turn the pad into a normal calculator, high numbers on top, with no confusion as to current numbering.

Two demonstration graphics programs have been written for this system. The first, whose three basic button labelings appear in Fig. 7, provides for drawing electronic circuit schematic diagrams, such as the one shown in Fig. 7d, by the juxtaposition and combination of basic patterns. A pattern is selected by pressing a key labeled by a small picture of the pattern (see Fig. 7a); the button marked NEXT EIGHT causes paging through several such sets of eight patterns. When pressed, the pattern is framed, as shown. The user may branch to that part of the program which places the new element on the faintly visible current picture by pushing PLACE. (Program branches involving relabeling buttons are indicated by symbols resembling miniature sets of 12 buttons with a label alongside.) Figure 7b shows the result: buttons for moving the new instance, for saying "OK, add it," and for seeing the result. The LET'S SEE button causes the button set and labels to go out, likewise the light illuminating the hand, whereas the faint circuit diagram in the background comes up to full brilliance, as in Fig. 7d. The speckling of unused buttons serves to obscure the intrinsic Touch-Tone labels. Another program branch provides for the redefinition of a pattern (see Fig.



(a)

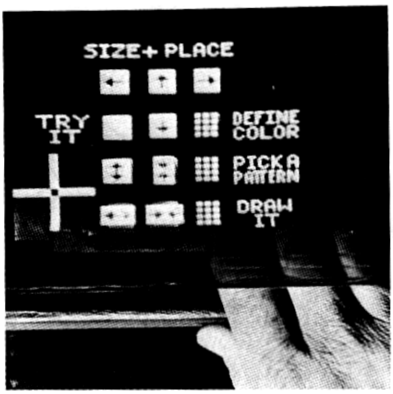


(b)

Fig. 5—(a) Console setup with a relabelable Touch-Tone pad as an iterative graphics console. (b) Side view of console showing extra mirror for reflecting bottom-of-screen light source.

7c). Here, an enlarged pattern appears on the right with a 3 by 3 window drawn on it. Contents of this window are displayed on the top nine buttons, where a button press flips the cell. The arrow buttons move the

(a)




(b)

Fig. 6—(a) Distant view of simulated Touch-Tone console setup showing inverted image on screen, mirror, and user's hand operating the buttons. (b) User's view (through mirror) of situation in (a) with Touch-Tone buttons effectively relabeled by the computed display.

window over the pattern; OK returns control to the rest of the program, with the pattern redefined.

A more elaborate program uses the capabilities of the color monitor to generate four-color designs like the one shown in Fig. 8 by selecting and applying variously stretched and positioned instances of basic patterns. Figure 9 shows its seven basic button labelings. The user may start with any of the four colors as background (Fig. 9a) and selects a pattern as before (Fig. 9b). In addition to placing it anywhere on the screen, the user may change its height and width independently (Fig. 9c). Before finally drawing the addition onto the picture, an optional

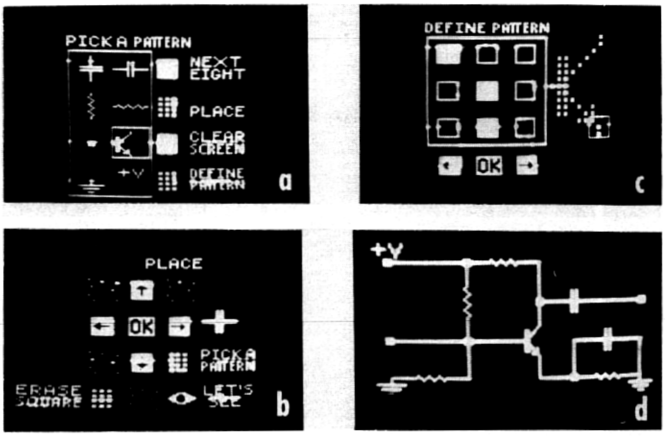



Fig. 7—(a) through (c). Button labelings of a program for composing electronic circuit diagrams. (d) Resulting diagram.

border, of width 0, 1, 2, or 3, and its color, are chosen (Fig. 9g). The colors themselves may be redefined (Fig. 9f) by increments or decrements of the three primary colors, the ± button serving to flip between modes ADD and SUBTRACT. Throughout the program, buttons which are temporarily meaningless disappear: if border width is zero, the color buttons vanish; if no more red can be added in defining the currently selected color, RED goes out; if position or size are extreme, the corresponding cursor arrow button disappears.

To summarize the Touch-Tone demonstrations, a complicated interactive graphics program can be run by means of a 12-button Touch-

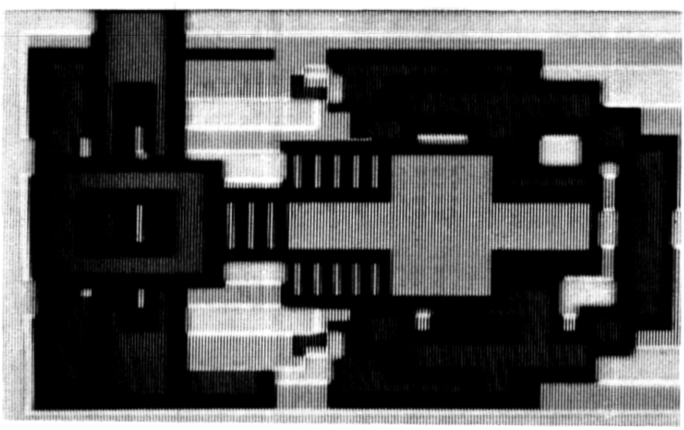


Fig. 8—Sample result of a more general program for production of four-color designs made of variously stretched and positioned geometric patterns.

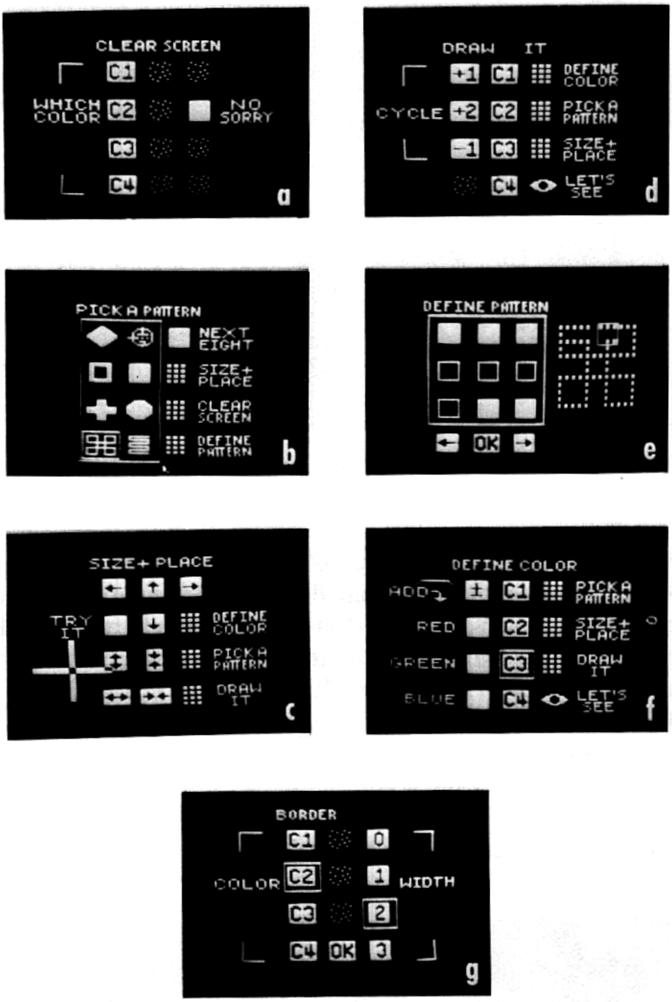



Fig. 9—The seven basic labelings for design-generating graphics system used to produce Fig. 8.

Tone pad if the buttons are easily relabeled to provide a rich variety of functions. Button forms and features found useful are:

Solid square with a 1- to 3-word label alongside.

Small pictures of symbols significant in the program.

Other iconic symbols:

An eye meaning "Let's see the picture."

Miniature 12-button set with label: a program goto involving relabeling.

Arrows for positioning a cursor and setting its size.

Picture cells in a basic pattern being defined.
Speckles to hide the intrinsic label of a nonfunctioning button.
A frame marking the current selection.
Alternation or cycling:
    Paging through sets of patterns.
    Add vs subtract for defining colors.
    Black vs white cells in pattern being defined.
Buttons that disappear when not meaningful:
    Cursor arrows when at extreme size or position.
    Color increments when at limit of range.
    Border color when width = 0.
Button set that disappears for viewing program result.
Frame around logical groups of buttons:
    Enlarged window of pattern cells.
    Pattern set.

## IV. COMPARISONS WITH OTHER DEVICES

The existing system closest in form and function to this one is the touch panel[7,8] developed as a part of the Plato computer-assisted instruction project,[9] where the user's finger on the screen intercepts one vertical and one horizontal light beam, with position decoded to one spot out of 16 by 16. Virtual light buttons have the following advantages over the touch panel:

(*i*) The hand is transparent; it does not obscure buttons pressed or buttons below it.

(*ii*) The keyboard has tactile feedback through button motion, but it does *not* respond to fingers passively resting on unpressed buttons—both characteristics are very important in typing.

(*iii*) Keyboards can easily be made for the simultaneous detection of more than one button hit (as in defining a pattern, typing a capital letter, or inputting a musical chord).

(*iv*) Only one kind of electronic-detection circuitry is required, that for detecting contact closings. (Almost every commonly used system has a keyboard).

A light pen, like the touch panel, also is a single-position indicator, and it obscures part of the region pointed to, but it does permit drawing free-hand curves and precise positioning on the picture cell level. Virtual light buttons do not lend themselves well to these tasks; for such operations we might use instead an $x$-$y$ tablet with virtually superimposed display. We can, however, use a button array for pointing with much finer resolution than button spacing, by any of a variety of protocols:

(*i*) A first button hit can position a cursor at the button center, and thereafter a small panel of four buttons in the left or right lower corner may serve to step the cursor in fine increments. In addition, the cursor may slew in any of these four directions if a slew button is simultaneously depressed.

(*ii*) Alternatively, after a single button hit positions the cursor to the center of the button, a second nearby button depressed, before the first is released, can mean "so many subdivisions in this direction." This method is quickly learned and provides for easy positioning on a grid five or seven times finer in both directions than button spacing. Subsequent button hits, relative to either the first or second button, could mean picture cell displacements in the corresponding direction.

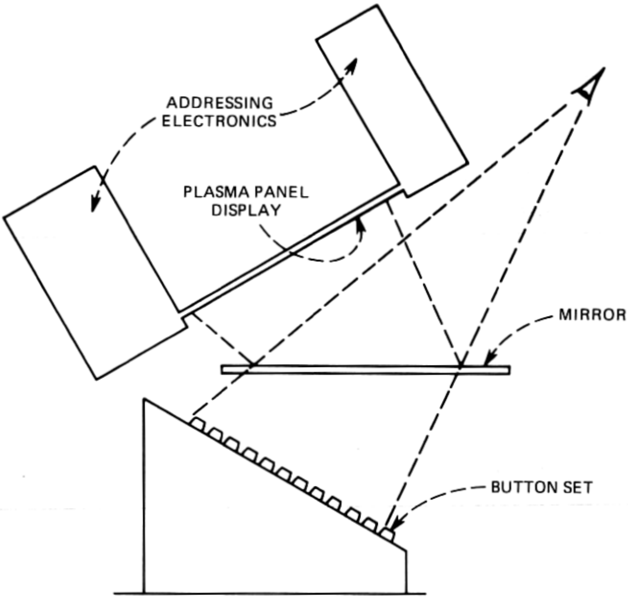
## V. ONGOING WORK

Experimental and developmental work is continuing with hardware and with both general and specific software, as follows:

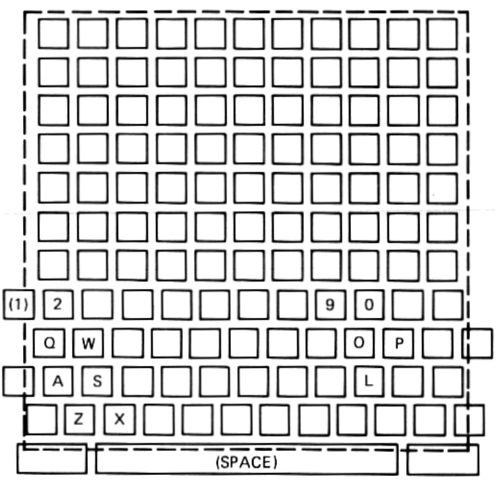
(*i*) The virtual light button setup is being considered as a possible form for a TSPS console. It would have the following advantages for operating companies: it would be expected to reduce training time; additions or changes in service or protocol would not require hardware changes to the consoles; one design would serve many purposes—handling phone calls, maintenance, traffic control, clerical work. Implications of the latter are that the operator's job could be restructured considerably, with periods of instruction or other jobs easily interleaved with normal call handling in off-peak hours.

(*ii*) A versatile and economical console is being designed, and a mockup built, using a 128-button panel, and a 512 by 512 60 pel/inch plasma panel[10] as the display, as shown schematically in Fig. 10. (The plasma panel needs no external refresh system—cells may be lit or extinguished individually, and each retains its state until changed. The panel is flat, permitting the button set to be flat, and since picture cell positions are defined by the structure of the device, the overall setup cannot become misaligned by electronic drift.) The board is arranged basically as an 11 by 11 array on ¾-in. centers (normal typewriter spacing) with slight adjustment of the bottom rows so they conform closely to a regular typewriter. Some buttons hang partially off the 8½ by 8½-in. display area; their meanings will normally be understood. The labels in Fig. 10 indicate how a typewriter keyboard is intended to be mapped onto the buttons set; the buttons can also be used as an 11 by 11 rectangular array but with some distortion in the lower lines, at least for the lowest 12-button row. Arrangements are being made to read all combinations of simultaneous button presses.

Commercially available plasma panels, sadly, are monochrome




(a)




(b)

Fig. 10—Console under construction using plasma panel display. (a) Side view. (b) Button layout: 128 buttons basically positioned on ¾-inch centers. Labels illustrate usage of lower board as a typewriter. Large square indicates virtual position of 8½-inch square plasma panel display area.

(neon orange). They are, however, much more economical than color TV monitors plus refresh buffers.

The ultimate flexible-but-economical console is expected to be close to the above design of keyboard + plasma panel, with addressing and driving electronics of the plasma panel in the lower cabinet, not in a bulky frame around the display. It would contain a character-generator capable of generating (hierarchically) parts of buttons, button tops, and sets of button tops. The remote computer would then need to designate only which light button(s) to put up and/or extinguish, and where, whereas the console would report button hits, perhaps with local culling of hits on nonlighted buttons. Both are low-capacity channels; a twisted pair phone line would suffice. An 11 by 11-in. mirror would be big enough for a single user (larger mirrors are useful for demonstrations).

(*iii*) A general-purpose software package under development will ultimately facilitate the writing of specific usage programs. It will permit convenient design and labeling of light buttons, plus facilities for conveniently defining mappings between logical and physical buttons and describing changes in state: appearance and disappearance of buttons, changes in values of variables, and flow of control. It will also provide a testing ground for one of the author's basic attitudes about usage of such a system: that there should always be exact correspondence between buttons which appear and those responded to. This ground rule should aid the development of complex systems like TSPS where there is a huge number of combinatoric states of the board, and where it is a big and difficult job to define precisely and completely which button hits are or should be legal from instant to instant.

(*iv*) One application nearing completion is a text editor, where text being worked on appears in the top part of the screen, while the bottom part serves as a typewriter keyboard. The novel feature of the setup is that pointing (to lines or words or positions for deleting, changing, or inserting) is done by pointing into the text. Text is displayed with three lines of five characters on each button top, and one designates a character by a sequence of two button hits: first the button on which the character appears, followed by the same button if the character is centered on this button, or by a nearby button in the direction that the character is off-center (the second button is always one of the 3-high by 5-wide subarray centered on the first).

## VI. ACKNOWLEDGMENTS

Dan Franklin designed and implemented the text editor. I thank Max Mathews and Peter Denes for facilitating the hardware developments and for suggesting the telephone operator console application, and Steve Bauman and Stu Silverberg for helping to formulate a meaningful TSPS demonstration. I thank Marie Hill, former Manager of Operator Services at Morristown, N. J., for several helpful consultations and for arranging for me to observe and monitor operators processing actual calls.

## REFERENCES

1. K. C. Knowlton, US Patent No. 3,879,722 Issued Apr. 22, 1975: "Interactive Input-Output Computer Terminal with Automatic Relabeling of Keyboard."
2. K. C. Knowlton, "Virtual Pushbuttons as a Means of Person-Machine Interaction," Proc. IEEE Conference on Computer Graphics, Pattern Recognition, and Data Structures, Beverly Hills, California, May 1975, pp. 350–351 (Abstract).
3. P. B. Denes, "A Scan-Type Graphics System for Interactive Computing," Proc. IEEE Conference on Computer Graphics, Pattern Recognition, and Data Structures, Beverly Hills, California, May 1975, pp. 21–24 (Abstract).
4. P. B. Denes, "Computer Graphics in Color," Bell Laboratories Record, *52*, No. 5 (May 1974), pp. 138–146.
5. R. J. Jaeger, Jr., and A. E. Joel, Jr., "TSPS No. 1—System Organization and Objectives," B.S.T.J., *49*, No. 10 (December 1970), pp. 2417–2443.
6. A. M. Noll, "Scanned Display Computer Graphics," Commun. ACM, *14*, No. 3 (March 1971) pp. 29–32.
7. F. A. Ebeling, R. S. Goldhor, and R. L. Johnson, "A Scanned Infra-red Light Beam Touch Entry System," Digest of Technical Papers, SID International Symposium, June 1972, pp. 134–135 (Abstract).
8. B. L. Richardson, "X-Y Coordinate Detection Using a Passive Stylus in an Infra-red Diode Matrix," Digest of Technical Papers, SID International Symposium, June 1972, pp. 132–133 (Abstract).
9. D. L. Bitzer and R. L. Johnson, "PLATO—A Computer Based System Used in Engineering Education," Proc. IEEE, 59, No. 6 (June 1971), pp. 960–968.
10. W. E. Johnson and L. J. Schmersal, "A Quarter Million Element AC Plasma Display With Memory," Proc. SID, *13*, No. 1 (First Quarter 1972).