COMPUTER TECHNOLOGY

Random figures and mirror images with variable complexity, size, and orientation in Applesoft BASIC

JAMES W. ALDRIDGE and SERGIO A. FLORES
Pan American University, Edinburg, Texas

An algorithm is described for generating random figures of variable complexity, and programs are described for generating libraries of the figures and their mirror images encoded as Applesoft BASIC shape tables. The figures may be displayed by a user program with variable size and orientation under program control. A validating experiment of the mental rotation type is also reported. The figures in that experiment produced results typical of other types of stimuli. In addition, overall reaction time, mental rotation speed, and practice effects were sensitive to the complexity variable.

Applesoft BASIC, the language resident in all Apple II-series computers since the II+, has only one instruction for directly creating high-resolution graphics. HPLOT will plot a single point at specified coordinates, and HPLOT TO will draw a line between a current location and specified coordinates. Although it is possible to use this command for creation of stimuli by experiment-controlling programs, it is usually desirable to develop stimuli external to an on-line program by using binary "shape tables." Stimuli encoded as shape tables may then be loaded, drawn at specified coordinates to a specified scale, and rotated by a program using built-in Applesoft commands (see Apple Computer, 1981, for more information).

This paper describes a simple algorithm for generating random figures and a program that uses the algorithm to generate libraries of figures. The program encodes the figures and their mirror images as Applesoft shape tables and allows editing of the figures. Companion programs save the tables to disk and load them from disk. A validating experiment is also described, in which figures generated by the program were used in a mental rotation procedure.

Algorithm

Each figure is created from a series of vertical and horizontal lines. Conceptually, the algorithm begins at a starting point and then chooses the direction of the next line segment in a series of "turns." On the first turn, a decision is randomly made to move left, move right, or

not move. On this decision and each subsequent decision, the possible outcomes are equiprobable. If the decision is to not move, a decision is then randomly made to move up, move down, or not move. If the decision is still to not move, the process repeats, with alternating horizontal and vertical decisions, until a move is made and the turn ends. The second turn is similar, except that the decision process begins with a vertical move rather than a horizontal move. The process then continues until a figure of the desired complexity is created.

Programs

MIRROR SHAPES is a program written in Applesoft BASIC that incorporates the above algorithm for creating and editing tables for up to 127 shapes and their mirror images, at a specified complexity. Complexity is specified by means of an initial question, which is answered with an integer corresponding to the number of times the two-turn figure-creation algorithm is to be executed. Although the Applesoft RND function is used for each decision, it is properly seeded (see Aldridge, 1987).

One purpose of MIRROR SHAPES was the creation of figures that could be presented in variable orientations by using the Applesoft ROTATE command. However, ROTATE rotates a figure around the figure's starting point, which may not correspond to the figure's center. In order to create figures that will rotate about their actual centers, MIRROR SHAPES records the highest, lowest, leftmost, and rightmost points plotted during the figure-creation algorithm. After a figure is created, these values are used to calculate the figure's center. Nonprinting horizontal and vertical lines from the actual center to the starting point is then added to the beginning of the shape table, so that the actual center becomes the table's starting point and center of rotation.

Development of the programs, the experiment, and preparation of this paper were supported by grants BNS-7919801 from the National Science Foundation and S06RR08038 from the National Institute of Mental Health/MBRS to the senior author. Reprints and programs may be obtained from James W. Aldridge, Psychology Department, Pan American University, Edinburg, TX 78539.

Applesoft graphics routines also draw diagonal lines in a broader stroke than vertical or horizontal lines. Figures drawn at rotations that are not multiples of 90° therefore have a "fuzzier" appearance than those using vertical and horizontal lines. In the experiment reported below, however, there was no evidence that this difference had any effect on reactions to the figures.

The shape table bit patterns representing the sequence of drawing movements are encoded as BASIC strings of ones and zeros as each figure is created, and are subsequently translated into binary entries in a shape table. Although this technique greatly simplifies the programming, the use of strings creates some limitations. String length limits in Applesoft result in an absolute maximal complexity of 47 iterations of the algorithm, but 47 iterations are possible only if the starting point happens to coincide with the actual center and addition of nonprinting lines is not required. In practice, complexity values in excess of 36 require many attempts to create a single figure.

The practical limit of 36 iterations in figure complexity is not a problem in most applications. Figure 1 shows examples of figures generated at complexity levels of 4, 12, 20, and 36. Simpler figures, such as those at complexity 4, are suitable, for example, as pseudoletters. The more complex figures at complexity 12 are suitable for most visual memory experiments. Complexity 20 figures are quite complex, and the intricacy of complexity 36 figures is rarely needed. The effects of complexity levels were empirically examined in the experiment reported below.

Although each increment in complexity level adds two line segments in the figure-creation algorithm, the com-

Figure 1. Examples of figures generated by MIRROR SHAPES. Complexity refers to iterations of the shape-creation algorithm.

plexity level does not strictly correspond to the number of line segments visible in the figure. In order to avoid creation of an excessive number of squares in the figures and to permit protruding line segments, the algorithm permits overtracing. This has two additional consequences.

One consequence is that the Applesoft command XDRAW usually does not work properly with these figures. XDRAW plots a figure in the color complementary to the color already existing at each point. The most common application of XDRAW is in erasing existing figures without altering the background. If XDRAW is used with a figure that draws twice at the same location, XDRAW erases the line on the first pass, but redraws it on the second. It is therefore necessary to erase figures generated with MIRROR SHAPES by changing HCOLOR to the color of the background and redrawing the figure.

The second consequence of overtracing is that figures of the same nominal complexity differ in their number of visible line segments. This means that complexity should be interpreted as an average value, with some actual variability. It is sometimes the case, especially with low complexity levels, that a particular figure is simpler than is acceptable. For this reason, MIRROR SHAPES also permits editing of the shapes in a table after they have been generated.

After an entire shape table has been created, the program asks whether editing is desired. If so, the user may direct the program to step through each figure, or may specify one particular figure. In either case, eight examples of the figure are simultaneously drawn on the screen at rotations of 0° through 360° in increments of 45° . This not only permits examination of the basic structure of the figure, but also allows confirmation that the center of rotation of the figure corresponds to the figure's apparent center.

The user is then asked whether to save or replace the figure. If the figure is replaced, the figure-creation algorithm is executed, and the new figure is placed in the location of the old figure in the table. Because of the non-printing centering lines, however, the figures may differ in the number of bytes occupied in the table. If the new figure occupies fewer bytes than the original, a sequence of nonprinting up and down movements is added to the end of the figure. If, however, the new figure occupies more bytes than the original, it is truncated to fit into the available space. If truncation is a problem, this is revealed when the program returns to the edit function and displays the new figure.

Once the table is generated, a second program, SHSAVE, saves the table as a binary file on disk. Applesoft shape tables may be located anywhere in available memory, but it is usually desirable to protect them by placing them at the top of available memory. The system variable HIMEM may then be set to just below the tables, so that the operating system regards the tables as occupying unavailable memory. Because the length of a shape table is not known prior to generation and the table grows

upward in memory during generation, MIRROR SHAPES initially places a table just above high-resolution page 1. SHSAVE moves a new table to the top of available memory and then saves it to disk in such a way that the table automatically BLOADs to the same location.

In order for a program to use a shape table, it is not sufficient that the table be BLOADed with the standard command for loading binary files. It is also necessary to store the starting location of the table at hexadecimal locations \$E8 and \$E9 (decimal 232 and 233), low byte first. A third program, SHLOADN, does this automatically. SHLOADN, however, will reload a shape table only into the location from which it was previously saved. If it is necessary to load it into another location, the user program must BLOAD the table to the desired location and then store the starting address, using POKes to 232 and 233. It is normally necessary to do this only when loading several tables, as, for example, when figures with differing complexities are used simultaneously. If several tables are concurrently in memory, the user program must change the value in 232 and 233 to the appropriate location whenever a different table is referenced.

Experiment

In order to test the utility of figures generated by MIRROR SHAPES in a fairly rigorous situation, an experiment of the mental rotation type (e.g., Shepard & Metzler, 1971) was performed, using shapes with complexity levels of 4, 12, 20, and 36 at respective SCALES of 19, 11, 9, and 7. On each trial, while a cleared text page was displayed, a figure was drawn to the left of the HGR screen and either the same figure or (with a probability of 50%) its mirror image was drawn to the right. A machine language instruction-counting routine then displayed the HGR screen and measured the subject's reaction time (time to press either a button marked "same" or a button marked "different"). The buttons were connected to the Apple's standard game control button inputs.

Each level of complexity was presented to one group of 8 Pan American University students. On each trial, the left figure was presented at a rotation of either 0° or 180°. The right figure was presented rotated clockwise a multiple of 45° from the first, between 0° and 315°. For each set of eight trials, the eight rotation differences were randomly ordered, and the differences were not repeated within the set. Unscored practice sets of eight trials each were repeated until a set was completed without error or five sets had been completed. Following practice, three blocks of 64 trials each were used. Between the experiment-controlling program and the shape tables, there was room in memory for only 63 figures. The figures were randomly ordered for each subject on each block, and one figure was randomly selected to be repeated on the final trial. The instructions to the subjects emphasized accuracy.

The arithmetic mean reaction times are shown in Figure 2 as a function of degrees of rotation for each level of complexity. The main effect of complexity was reliable [$F(3,28) = 3.0, p < .05$], although in an unexpected

Figure 2. Mean reaction time (RT) as a function of orientation difference between shapes in each pair. Complexity (number of iterations of the shape-creation algorithm) is the parameter.

way. Reaction time was fastest for the simplest figures (complexity 4), but was almost equally fast for the most complex figures (complexity 36). Longest reaction times were for figures of intermediate complexity, the longest being for complexity 12.

The effect of rotation was highly reliable [$F(7,196) = 56.8, p < .001$], and there was also a complexity $\times$ rotation interaction [$F(21,196) = 2.9, p < .001$], which reflected the same differences revealed in the main effect. The effect of rotation was greatest for the intermediate complexities (especially complexity 12), with smaller effects of rotation for the simpler and more complex figures.

Effects of practice were studied by comparing mean reaction times between the three blocks of 64 trials, as shown in Table 1. There was a main effect of block [$F(2,56) = 7.5, p < .025$], with reaction time decreasing by about 200 msec per block. However, there was also a block $\times$ complexity interaction [$F(6,56) = 2.8, p < .025$]. There was no effect of practice for complexity 4 figures, which in fact produced a slight increase in reaction time between Blocks 1 and 3. The most complex figures showed an intermediate effect of practice, but the largest effect of practice was with complexity 12. In spite of this interaction, however, the ordering of com-

Table 1
Mean Reaction Time (sec) as a Function
of Trial Block and Complexity

Complexity	Block			Block 1-3
	1	2	3	
4	2.763	2.811	3.006	-0.243
12	5.368	4.971	4.439	0.929
20	4.008	3.893	3.573	0.435
36	3.494	3.222	2.995	0.499

Note.—Complexity refers to number of iterations of the shape-creation algorithm.

plexity reaction times remained approximately the same in Block 3 as in Block 1. Block did not interact with rotation, nor was there a block $\times$ rotation $\times$ complexity interaction ($F_s < 1$).

An analysis of error rates parallel to that performed on reaction times was performed, revealing a main effect of rotation [$F(7,196) = 14.7, p < .001$] as the only reliable effect or interaction. However, error rate effects may have been constrained by the fact that any subject with an error rate in excess of 10% was replaced.

Conclusions

The figures generated by MIRROR SHAPES can be used effectively in psychological research. The primary purpose of the experiment was to determine whether the complexity and orientation variables would have systematic behavioral consequences. Not only was this found to be the case, but the effects of complexity were more extensive than anticipated.

It appears that the range of complexity available with MIRROR SHAPES considerably exceeds the range needed, at least in this type of experiment. Maximal complexity effects were found when comparing simple figures with those of intermediate complexity. The present experiment does not suggest reasons for the reduction in complexity effects when complexity was increased beyond an intermediate level, although it is possible that past a certain point the figures became so complex that subjects ceased attending to the details of the figures. Alternatively, it is possible that subjects may have "rotated" only a portion of very complex figures to determine whether there was a match.

It also appears that effects of perceptual learning can be manipulated by selection of an appropriate level of complexity. In the above experiment, the simplest figures showed no effect of practice, whereas figures of intermediate complexity produced a 17% decrease in reaction time over the course of only three blocks of 64 trials.

Finally, with specific regard to mental rotation experiments, it appears that the MIRROR SHAPES figures produce results comparable to those of more conventional stimuli. The primary advantage of these figures, however, is that they are far simpler to generate and use than most stimuli currently available.

Availability

MIRROR SHAPES, SHSAVE, SHLOADN, and the programs for the experiment will be provided at no cost if a disk is sent to the senior author. Also available are two demonstration programs. Since MIRROR SHAPES and the experiment program are too large to run if they include REM statements, both documented and undocumented versions of these programs will be provided.

REFERENCES

- ALDRIDGE, J. W. (1987). Cautions regarding random number generation on the Apple II. *Behavior Research Methods, Instruments, & Computers*, **19**, 397-399.
- APPLE COMPUTER, INCORPORATED. (1981). *Applesoft II BASIC programming reference manual*. Cupertino, CA: Author.
- SHEPARD, R. N., & METZLER, J. (1971). Mental rotation of three-dimensional objects. *Science*, **171**, 701-703.

(Manuscript received September 29, 1987;
revision accepted for publication December 16, 1987.)