— COMPUTER TECHNOLOGY —

Software-based visual psychophysics using the Commodore Amiga with Deluxe Paint III

PETER WENDEROTH

University of Sydney, Sydney, New South Wales, Australia

Often it is useful to produce visual displays that allow for pilot testing of experimental hypotheses, which let the experimenter examine effects reported by others, or which can be used for at least some serious experimentation (e.g., decrement in visual illusions with practice). The combination of the Amiga computer and Deluxe Paint III software is ideal for this purpose. The ease with which graphic displays can be produced, coupled with the new animation facility in the software, enables method-of-adjustment psychophysical experiments to be produced in a very short time. Worked examples are given, and the relative merits and shortcomings of the Amiga computer for these purposes are discussed.

This paper demonstrates methods of setting up visual psychophysical experiments in which a whole experiment can be ready for implementation in only minutes. Furthermore, absolutely no knowledge of computers is necessary, and no programming languages are used. Thus, the techniques are ideal for preliminary testing of ideas, for use in introductory classes by undergraduate students, and for serious research in areas such as Müller-Lyer or Poggendorff illusion decrement over repeated trials.

Recently, Anstis and Paradiso (1989) have shown how the joint use of commercially available software, namely Deluxe Paint and The Director, could be used to run psychophysical experiments. Their methods enable data to be collected and stored to file, but the use of a Basic-like language is required. The methods described here operate entirely within the software package Deluxe Paint III, so that it is not possible to store data; rather, values are read from the screen and written down. However, as stated at the outset, the aim is to describe methods that anyone can use very easily.

The general technique is as follows: The latest version of Deluxe Paint (Deluxe Paint III, Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404; or, for a Pal version that completely fills the European screen: 11/49 Station Road, Langley, Berkshire, England SL3 8YN) includes an animation facility. This allows different pictures to be drawn on a series of "frames" (screens), which can then be flipped through at variable speeds to create ap-

parent movement. For example, one could create two frames and put a black square to the left on Frame 1 and the same square to the right on Frame 2. Selecting CONTROL/PLAY from the ANIM menu (see below) would then cause repeated cycling across the two frames to show a black square in horizontal apparent motion.

This animation facility can easily be used to do experiments using the method of adjustment. First, one uses the mouse and relevant menus to draw the fixed part of a stimulus—for example, the vertical parallels and the left oblique of a Poggendorff illusion display (see Robinson, 1972). These fixed segments are represented by the solid rectangle and the oblique line labeled A in Figure 1. Next, the variable right oblique is created and made to move vertically from obviously too high to obviously too low over a series of, say, 50 frames. In Figure 1, the right-hand obliques represent the positions of the variable segments on Frames 1, 16, (physically aligned with A), 27, and 43. Next, the standard or fixed stimuli—the parallels and left oblique—are copied to all 50 frames. This is all that needs to be done.

The keyboard (using the 1 and 2 keys) can be used to step through the frames one at a time until the subject judges that the obliques are aligned. Knowing which frame contains the truly aligned stimulus (e.g., Frame 16/50), the experimenter can read off the frame number at which the subject judged alignment (say 20/50). If the variable stimulus moves 1 pixel per frame (see below), the error is 4 pixels. Since each pixel (in low resolution, or Lo-Res) measures 0.78 mm vertically and 0.82 mm horizontally, the vertical misalignment would be $4 \times 0.78 = 3.12$ mm.¹

Let us produce a version of this Poggendorff experiment as a first example. The required hardware items are

This work was supported by Australian Research Council Research Grant A78831871. The author wishes to thank three unknown reviewers for insightful and helpful comments. Correspondence may be addressed to Peter Wenderoth, Department of Psychology, University of Sydney, N.S.W. 2006, Australia.

Figure 1. The fixed (black bar and Line A) and variable segments of the Poggendorff illusion. Lines 1, 16, 27, and 43 represent the positions of the line on Frames 1, 16, 27, and 43 of the animation. The line on Frame 16 is physically aligned with A.

an Amiga 500, 1000 (Kickstart 1.2 or later), or 2000, with 1MB RAM and two floppy disk drives (or a hard disk). The first floppy drive (internal: DF0:) is used to boot up the Deluxe Paint III program; the second, external drive (DF1:) is used to copy created files for later use. Anstis and Paradiso (1989) fully describe the basics of using disks and disk drives. All of the demonstrations discussed below were created on an Amiga 2000 with 3MB RAM and a 40MB hard disk.

The Poggendorff Illusion

Open the Deluxe Paint III disk by clicking twice on its icon. Double click on the Deluxe Paint program icon to start the application. Select the default settings in the menu that appears (Lo-Res 320×200, 32 colors) by clicking on OK. When the paint screen appears, the foreground is in a fawn color and the background is black. In order to obtain a white background and black foreground, press the p (for Palette) key on the keyboard. In the Color Palette box that appears, the colors will be referred to by numbers (not shown) from the top of the left column (1, black) vertically downward (2, fawn; 3, red; etc.), column by column, until the last, bottom right color (32) is reached. When the palette appears, click with the left mouse button on the fawn foreground color (Color 2). The RGB sliders show that this color consists of a lot of red, less green, and even less blue. To change the fawn color to white, use the mouse pointer to move all three sliders up to 15. The foreground color is now white. Click OK on the palette requester and it will disappear. Finally, note that just above the color palette at the bottom right of the screen there is a white circle (foreground color) on a black background (background color). To reverse these, click the right mouse button on Palette Color 2 (white). Click on CLR with the left mouse button and the screen turns

white. Click the left mouse button on Color 1 (black). Now we have a white background and can draw in black.

Now we will draw the fixed portion of the Poggendorff figure. Go to the menu bar at the top of the screen just to the right of where DELUXE PAINT COLOR is printed, and hold down the right mouse button. Select ANIM/FRAMES/SET #. When the requester appears, change the frame count to 50 and click OK. Now, the top left of the screen shows that you are on Frame 1/50. Go to the PREFS menu and select COORDS. Now, as you move the mouse around on the screen, you are given its coordinates, in pixels, at the top right of the screen. We shall need to use not only the screen we are on but a spare screen called SCRATCH. To see it, press j. This screen is identified by the word *scratch* at the top left. By pressing j again and again, one can toggle back and forth from the anim (Frame 1) to the scratch screen. Remember that if you draw something incorrectly, you can undo it by clicking with the left mouse button on UNDO in the right-hand toolbox, or click on CLR to clear the screen, or click on the solid part of the square shape half way down the toolbox, select Color 2 (white) and paint a white quadrilateral over your mistake to erase it.

On the scratch screen, select the solid rectangle tool (click on the filled half of the square shape half way down the tools menu on the right of the screen) and draw a black rectangle, reading the coords as you do it. These will now be relative coordinates and will give you the x- and y-coordinates of the bottom right corner of the rectangle relative to the top left corner, which is 0,0. Make the rectangle about 30 pixels wide and 130 high. Next, using the smallest, usually the default, brush (the smallest shape at the very top of the toolbox), draw a line 40 pixels long, anywhere to the side of the screen, using the coords information to make it tilted exactly 45°, sloping down to the left.

Click the left mouse button on the incomplete square to the left of the letter A in the toolbox. Position the resulting crosshair to the top left of the line, hold down the left mouse button and pick up the line as a brush by moving the mouse beyond the line to the right and then down below it. Click it into place with the left button so that its upper end meets the solid rectangle about a quarter of the way up on the left-hand side. Get rid of all lines on the screen other than the Poggendorff components by clicking on white as the foreground color and selecting the solid rectangle to paint over them. Now select the menu item EFFECT/STENCIL/MAKE. When the requester appears, click on Color 1, black, and then MAKE. Now we have completed the nonvariable section of the Poggendorff figure, and, being a stencil, it cannot be erased. Check this by clicking on CLR.

Press p to bring up the palette requester. Click with the left mouse button on (a) Color 1, black (b) COPY and (c) Color 21. Color 21, previously the darkest gray, will now be black too, but it will not be a stencil color. Click OK. Press j to go to the Frame 1 screen. Select the smallest (1-pixel) brush at the top right of the toolbox, click on

Color 1 at the bottom right, and choose the straight line tool. Draw a line 40 pixels long, tilted 45° clockwise, and pick it up as a brush. Clear the screen and click on CURRENT FRAME. Press j, and, *without laying down the brush*, position it on the right side of the Poggendorff display, where it is obviously too high. *Without moving the mouse*, j back to the Frame 1 screen and click the left button to lay down the brush. Clear the screen (current frame). Even though there is nothing now on the screen, you have just told the program that when you animate this brush it should begin to move from the position it last occupied on the screen. Now select ANIM/MOVE. When the requester appears, set the y distance to -50. By making the count (number of frames) equal to the y (vertical) distance, you ensure that the progression from frame to frame is 1 pixel. Select PREVIEW and you should see your variable line gradually drifting downward, frame by frame. Select DRAW. Deluxe Paint now draws all 50 frames. When this is done, select ANIM/CONTROL/PLAY; you will see the animated line drifting down, over and over again. To stop the animation, click the left button or press the space bar. Select ANIM/CONTROL/GO TO and choose Frame 1.

Now we need to copy the fixed part of the illusion (which is still on the scratch screen) to every frame. Select PICTURE/SPARE/MERGE IN FRONT and choose ALL FRAMES. When the program has done this, you can play the animation and stop it at any point by pressing the left button. Note that you can decrease the speed, if you want, by changing the frames per second information in the ANIM/CONTROL/SET RATE requester or just by pressing the left/right arrow keys. Save the animation by selecting ANIM/SAVE. At the requester, select DF1: if that is where you want to save it, and type "pogg" under filename.

Next, we want the x-, y-coordinates for correct alignment. Use j to select the scratch screen. Select EFFECT/STENCIL/MAKE, click on Color 1 to release it from stencil, and click MAKE. Select the magnifying glass tool (above UNDO) and click the cursor over the left bottom end of the left oblique. Select the 1-pixel brush. Select red as the foreground color. Place the red mouse pixel over the bottommost pixel of the oblique. Write down the coordinates. Click on the magnify tool to deselect it, select the straight line tool, and replace the mouse pixel at the written-down coordinates. Press the left button and draw a 45° line upwards and across the solid Poggendorff parallels to emerge at the other side (45° can be ensured either by coords reading or just by ensuring that the line is not jagged). Magnify the area on the right of the parallels where the red line emerges. Place the mouse pixel on the first red pixel outside the parallels. Read off the coordinates; this is the point of true alignment, so write down these coordinates. Deselect the magnify tool.

Using j, go back to the Frame 1 screen. Magnify the right side of the figure at about the true alignment point and place the red pixel at that point. Using the 1 and 2 keys (previous, next frame), move frame by frame

through your animation until the variable line is at the true alignment point. Note the frame number and write it down.

That's it! Now you can clear all frames, ANIM/LOAD your program, and have subjects use the 1 and 2 keys to adjust to apparent alignment. When you do this, press F10 to hide the menu bar and toolbox. When each setting is complete, ask the subject to look away, press F10 to regain the menu bar, and write down the frame number. Different starting positions on different trials can be achieved by choosing ANIM/CONTROL/GO TO and selecting a particular frame: 1 will begin from too high, 50 from too low. From Frame 1, the 2 (next frame) key should be used, and from Frame 50, the 1 (previous frame) key should be used. Of course, one could choose any other frame number to begin.

The Müller-Lyer Illusion

Now that we have completed the first example and have dealt with general principles, it will become clear in this example how easy the variations are. It is assumed from here on that, on opening the program, the foreground and background colors are always set to black and white.

Open up as before, make 50 frames, and go to the scratch screen. Beginning at the left center of the screen, create a horizontal line 60 pixels long, using the smallest brush. Create a 45° line for an arrowhead and BRUSH/FLIP it to get the various other arrowheads you want, laying them onto the line as shown in Figure 2A, using the magnify tool. Note that the ZOOM tool alongside magnify can be used to vary the magnification factor by clicking on it with the right mouse button.

At the upper edge of the screen, create another line 80 pixels long and add outwardly going arrowheads as in Figure 2B. Pick this composite line with arrows up as a

Figure 2. The fixed [A] and variable [B] components used to create the frames of a Müller-Lyer illusion. [C] The appearance of the composite figure at physical equality.

brush, and place it alongside the inward arrowhead figure so that its left, unfeathered tip exactly touches the right side of that shaft (*do not click*). Press *j* to go to Frame 1, and lay down the brush by pressing the left button. Clear the frame, select *ANIM/MOVE*, and enter an *x* distance of -50 (to make the number of frames equal to the distance moved, in pixels). Choose *DRAW*. When the animation is completed, go to the scratch screen and remove all the extraneous figures leaving only the inward arrow figure. Press *j* to go to Frame 1, select *PICTURE/SPARE/MERGE IN FRONT*, and at the requester choose *ALL FRAMES*. You can then play the animation and save it as *ANIM/SAVE/ML*. Pressing Keys 1 and 2 allows adjustment settings. As I drew it, the point of physical equality was on Frame 22 (the two lines share the central pixel). Perceptual equality was at about Frame 34, and an overestimation of 12 pixels in 60 is an illusion of about 20%, as is often found. Figure 2C shows this powerful illusion; the shafts in this figure are equal in length.

Of course, the amount of illusion will depend on the arrowhead lengths selected. Finally, in this example, the 1 key moves the arrow to the right and the 2 key moves it to the left. Although this is not a problem for an experienced subject, the relationships could be reversed for a more ergonomically sound demonstration.

The Role of Elevation in the Ponzo Illusion

The Ponzo illusion is often presented in the form of two equal rectangles having different elevations on a background of linear perspective, as in Figure 3A. In this example, the variable stimulus will be the square shape at the horizon. First, we create the fixed stimulus on the scratch screen. Draw a large medium gray square, about 80×80 , at the bottom center of the screen. Pick it up as a brush and clear the screen. Select *EFFECT/PERSPECTIVE/DO*. A crosshair appears at the center of the screen representing the perspective center. The brush is now surrounded by a perspective wire frame. Locate the brush at the bottom of the screen, below the crosshair. Simultaneously press *SHIFT/8*, the 8 being that on the numerical pad to the right of the keyboard. The brush outline is now slanted 45° in depth. Lay it down. Select *EFFECT/PERSPECTIVE/FILLSCREEN*, and your linear perspective floor will be drawn.

Now draw a solid red square, 43 pixels on a side, at the bottom center of the screen. Press *F10* to delete all the menu bars before picking up the half screen (up to the horizon) as a brush. Save this as *BRUSH/SAVE/ponzo.v*. Select *BRUSH/ROTATE/90 DEGREES*. Save this as *BRUSH/SAVE/ponzo.h*. Clear all frames and set frame number to 40.

Load the *ponzo.v* brush, press *j*, press *F10*, and lay it down on the scratch screen. Press *j*. Draw a 50×50 square, using the same color (red) as that of the squares in the brushes you have saved. Pick it up as a brush and clear the screen. Go to the scratch screen and place the brush so that it just touches the horizon, directly above

Figure 3. The Ponzo illusion induced [A] by perspective and elevation and [B] by perspective alone.

the lower square (*do not click*). Press *j* and lay down, then delete, the brush. Choose *EFFECT/PERSPECTIVE/CENTER* and click the large crosshair at the very center of the bottom edge of the square you have laid down. This ensures that any change in size will occur about this point. Clear the screen and select *ANIM/MOVE*. Under *Dist Z*, enter 150. Select *DRAW*. Once the gradually shrinking (receding) square has been drawn to the 40 frames, select *PICTURE/SPARE/MERGE IN FRONT* and *ALL FRAMES*.

Now you have a 40-frame sequence in which the observer can use the 1 and 2 keys to set the "far" square apparently equal to the "near" one. Save this sequence as *ANIM/SAVE/ponzo.v*.

Clear all frames and load the brush *ponzo.h*. Repeat the procedure exactly, except: (1) align the variable square horizontally with the fixed square (i.e., on the horizon of what is now the "wall" rather than the "ground"),

and (2) make the perspective center the center of the side of the square facing the fixed square. Save this as ANIM/SAVE/ponzo.h (Figure 3B).

In my version, the point of physical equality is on Frame 9. Over various observers, the PSE is around Frame 17 for ponzo.v and around Frames 14–15 for ponzo.h. This suggests that the illusion of size is slightly larger in ponzo.v, probably due to the extra cue of elevation in that case, which adds to the linear perspective effect.

Simultaneous Contrast

There is also a clever way to demonstrate simultaneous contrast. The essence of the method is to create animation frames and to lay down on these frames small squares that are of identical shape and in the identical position, which gradually change in lightness. The observer can then flip through the anim frames until a lightness match with a comparison is achieved.

Press p to call up the palette. Copy Color 1 (black) to Color 3 and copy Color 2 (set white) to Color 16. Click on Color 3, then SPREAD, and then Color 16. Now we have a smooth range of grays between 3 and 16. Color 9 will be used as the test color: copy 9 to 25. The light and dark backgrounds will be Colors 5 and 15; copy these to 26 and 27, respectively. Click off the palette. (Press OK.)

Next, make 14 frames. Click on the grid tool (above the magnify tool), which will help constrain the squares about to be drawn to the same position on each frame. Click on the solid rectangle tool and Color 3 and go to some fixed coordinates—say, 100, 120. Draw a square 25 on a side. Press 2 to move to Frame 2. Pick up Color 4 and repeat this. Continue doing this until you reach Frame 14 and Color 16. Turn off the grid. If this anim is now played, it will display a stationary square that cycles in luminance. Go to ANIM/ANIMBRUSH/PICK UP. Go to Frame 1, and, using the crosshairs, pick up the square just as you would an ordinary brush. Go to ANIM/ANIMBRUSH/SAVE. Save the animbrush as, say, "lightness." What you have now done is to save the whole 14 frames of the animation.

Now clear all frames. On the animation screen (the frames screen, not the scratch screen), select the solid rectangle tool and Color 27 and draw a large inducing square to the left of the screen. Select Color 26 and draw an identical square on the right. Select Test Color 25 and draw a square 25 on a side in the center of the right-hand inducing square. Select ANIM/Frames/COPY TO ALL to copy this fixed display to all 14 frames. Go to ANIM/ANIMBRUSH/LOAD and load your "lightness" brush. Now, Deluxe Paint III lets you paint frame by frame in a very simple fashion. Press L-Amiga, and, while holding it down, move the animbrush to the center of the left-hand inducing square. Press and hold down the left mouse button until you reach Frame 14. You now have 14 frames that are identical, except that the lightness of the left test square changes systematically from frame to frame.

Save this whole sequence as an animation (ANIM not ANIMBRUSH). The subject can then press Keys 1 and 2 to step through the frames to try to match the left to the right test square. A close subjective match is not possible with these selected values, but most observers will select somewhere between Frames 4 and 5 while the true match is Frame 7. This can be demonstrated by picking up the various colors and painting them alongside the test colors on both sides.

An interesting variation on the above techniques could enable one to obtain quasi-quantitative measurement of the McCollough effect. Draw a grid of thin black lines and superimpose a white circle in the center. The thin lines are now interrupted by the overlaid circle. Select a bright red and complete the gaps between the broken black lines in red. Neon color spreading occurs: The spaces between the truly red lines apparently fill with a neon desaturated pinkish color. Such neon colors are very similar to the colors experienced in the McCollough effect. By setting the palette to range from bright red through white to bright green, and using these different colors to join the lines on successive frames of an animation, one can use the 1 and 2 keys to step through a range of red, green, and white neon colors. The measure of the effect would be the difference between the frame chosen in a pretest and posttest match.

Variable Angles

Because there is severe aliasing on the screen ("jaggies"; i.e., lines are jagged off the cardinal axes and the obliques), this demonstration is best done in Hi-Res. On opening PICTURE/SCREEN FORMAT/HIRES, select Hi-Res 2 colors. Set the frame number to any value—to save time here, use 10. On the scratch screen, draw a vertical line about 80 pixels long, using coords. Pick this up as a brush and j to the Frame 1 screen. Select magnify and magnify any part of the screen near the center. Select BRUSH/HANDLE/PLACE and holding down the left mouse button, drag around until the brush handle is at the lower tip of the line. This will be so when the crosshairs form a solid T shape with no undershoot or overshoot of the stem of the T. Note that you can keep on selecting PLACE in the HANDLE menu over and over again, until you are where you want to be. On releasing the mouse key, you will see that the brush is held at its end. Still under magnification, select EFFECT/PERSPECTIVE/CENTER. This allows you to click the left mouse button to determine where the center of rotation will be. Move the line to a central screen position and click the left button *twice* in the same place. The first click indicates that this is the point about which to rotate; the second places the end of the brush at that point. Now, deselect magnification, select ANIM/MOVE, and enter angle Z 90 and DRAW. Over the 10 frames, the line rotates about its lower tip from vertical to near horizontal.

Now magnify again and choose BRUSH/HANDLE/PLACE to select the upper tip of the brush. When this is done, under magnification precisely about the upper tip of this

brush with the bottom end of the one already in Frame 1. Since the perspective center has not changed, simply lay it down, unmagnify, select ANIM/MOVE, and change the Z angle to -90 . Select DRAW. This adds the lower half of the chevron or angle to the upper halves already created. At the end of the sequence, you will have an animation that begins as a vertical 160-pixel line and changes in large steps to a very small angle.

As usual, you can save this ANIM and use it to step through angle sizes using the 1 and 2 keys. Using some larger number of frames would, of course, give much more finely grained angular changes.

Conclusions

These examples should suffice to show not only how simple it is to do elementary psychophysics using Deluxe Paint III, but also how perceptual effects can be demonstrated to students and pilot experiments carried out. There are, of course, many other possible examples. Spatial frequency can be varied by creating a grating that recedes in depth and by creating a white solid circle on a black surround on the scratch screen; merging in front on all frames can make this grating visible through an aperture. In principle, too, one could create an animation in which the disparity in a random dot stereogram (Anstis, 1986) changes over frames. Aside from psychophysics, the program also allows creation of a crude crossed drifting plaid: although orthogonal gratings can be interleaved on successive frames, the alternation rate is not fast enough to avoid flicker. However, if the whole plaid is created on a single brush, it can be drifted across frames. Certainly, for testing potentially good ideas and for interactive demonstrations in introductory classes, the Amiga and Deluxe Paint III provide an inexpensive, user-friendly interface.

At present, the Amiga computer does not offer the high-resolution, flicker-free display required to display non-jagged tilted lines sufficient for some serious research. However, the release of a new operating system (Workbench 1.4) and new graphics chips is imminent. These will allow a flicker-free high-resolution display ($1,280 \times 512$ PAL, $1,280 \times 400$ NTSC) suitable for complex memory-hungry displays with less obvious aliasing. At present, a hardware device called Flickerfixer allows Hi-Res displays without flicker; it provides 704×580 (PAL) or 704×480 (NTSC) in the overscan mode on Deluxe Paint III. Also imminent is the release of the Amiga 3000

with a fast 68030 processor. Even now, however, some researchers have used the Amiga to conduct published research that is both elegant and innovative (e.g., Shimojo, Silverman, & Nakayama, 1989). Finally, there are many research teams working on problems such as decrement of the Müller-Lyer illusion with practice. The methods described here could easily be implemented for such experiments, because all contours are vertical, horizontal, or oblique and thus lack aliasing. By varying the distance of the subject from the screen, the 0.82-mm horizontal pixel width can provide a very precise measure of illusion and decrement in visual angles terms.

REFERENCES

- ANSTIS, S. (1986). Visual stimuli on the Commodore Amiga: A tutorial. *Behavior Research Methods, Instruments, & Computers*, **18**, 535-541.
- ANSTIS, S., & PARADISO, M. (1989). Programs for visual psychophysics on the Amiga: A tutorial. *Behavior Research Methods, Instruments, & Computers*, **21**, 548-563.
- ROBINSON, J. O. (1972). *The psychology of visual illusion*. London: Hutchinson.
- SHIMOJO, S., SILVERMAN, G. H., & NAKAYAMA, K. (1989). Occlusion and the solution to the aperture problem for motion. *Vision Research*, **29**, 619-626.

NOTE

1. One reviewer asked why squares on the screen were never truly square. I have a PAL version of Deluxe Paint in which the Lo-Res display is 320×256 pixels. The NTSC (US) system has 320×200 pixels. As I understand it, this means that the same number of pixels on NTSC will be spread over more screen space and hence appear more distorted. There is a menu command in Deluxe Paint called BE SQUARE (PREFERENCES menu) that corrects for this. Alternately, presumably the aspect ratio of the monitor itself could be adjusted. In my PAL version, the BE SQUARE command makes no obvious difference: All squares are almost square, and all circles are almost circles. However, if I choose the NTSC format, the bottom of the screen will be missing, and choosing BE SQUARE will make all squares and circles appear vertically squashed.

Note inserted at proof: Since arriving on leave at the Center for Visual Science at Rochester, NY, I have measured a 100×100 pixel "square" on an Amiga monitor. It was 8.09 cm wide and 8.89 cm high. Thus, on that monitor, each pixel was 0.81 mm wide and 0.89 mm high. Choosing BE SQUARE from the Preferences menu made squares truly square and made circles truly circular.

2. To delete the 1 from the frame number box, click to its right in the box with the left mouse button and backspace. Enter 50. For any other basic details of Amiga conventions, consult an Amiga and/or Deluxe Paint manual.

(Manuscript received January 11, 1990;
revision accepted for publication May 21, 1990.)