

REFLECTION BEFLECTION

JON CHOATE

What follows is a direct result of the work I have been doing over the past five years on a National Science Foundation project, grant number ESI9618029. The project, known as CAS-Intensive Mathematics, was based at Penn State and the University of Iowa and was headed by Dr. M. Kathleen Heid, and Dr. Rose Mary Zbiek. The goal of the project was to create materials for the secondary school curriculum that took full advantage of the latest technologies. We ended up making extensive use of Geometer's Sketchpad and the computer algebra system found on the TI-92 and more recently on the TI-89. We found that given these technologies we could have students work with geometric transformations in new and exciting ways. Because of this I found myself becoming more and more interested in computer graphics and their underlying mathematics. The study of transformations is greatly enhanced by the use of geometric construction packages such as Cabri and Geometer's Sketchpad, but there has been a lot written about that. So, this issue's column will focus on how you can use a computer algebra system, Mathematica, to study a particular geometric transformation, or reflection, in new and rewarding ways. Almost all that follows can be done using the computer algebra systems now available on handheld calculators such as the TI-89 and is in no way dependent on using Mathematica.

A word about design software

Macromedia's Freehand is typical of the type of software graphic designers are currently using. It offers the artist a variety of tools for creating and transforming designs including those that will translate, rotate, reflect, scale and shear (or skew) an image. **Figure 1** shows the Freehand 10 toolbar that contains its transformation tools. In what follows we will look at how this tool works and the underlying mathematics.

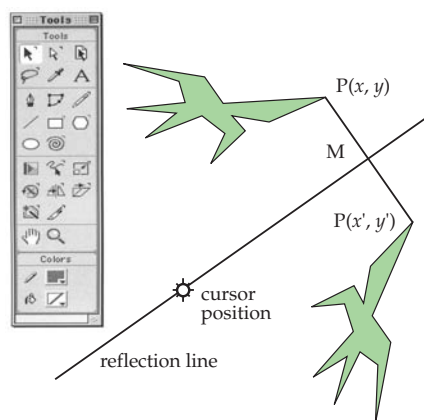


FIGURE 1. THE REFLECTION TOOL

To use the reflection tool in Freehand, you first select it, which changes the cursor to a circular shape. You then move the cursor so that it lies on a point on the reflection line and click the mouse. At this point a line appears. You now move the cursor until you get the line oriented the way you want it and then you release the mouse, at which point the reflected image appears and the reflection line disappears. Underlying this tool is a set of routines that take as input two points on the reflection line and which then create the appropriate transformation. As you know the transformations that are used to rotate, reflect, scale and shear (or skew) an image are all linear affine transformations and can be expressed in matrix form as

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}.$$

Actually, when matrices are used in computer graphics to transform planar figures, the computations are done using 3×3 matrices. With this in mind note that the general affine transformation shown above can be re-expressed as

$$\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where the point (x, y) is represented as $(x, y, 1)$. There are many advantages to using this representation. First, it allows you to perform translations using matrix multiplication. For example, to translate a point $P(x, y)$ 4 units in the x -direction and 10 units in the y -direction you could represent P by the column vector

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

and multiply by a 3×3 as shown below

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + 4 \\ y + 10 \\ 1 \end{bmatrix}$$

The technical name for this new coordinate system is homogeneous coordinates. They are used extensively in computer graphics for a variety of reasons and will be the subject of a future Modeler's Corner column.

The mathematics of reflection

In order to derive these formulas, one must first figure out how to reflect about any line through the origin. Note that the way this tool works, the computer is probably keeping track of two points: The first is the location of the cursor, call this P_1 , where the mouse button is depressed, and the second, P_2 , the location of the cursor when the mouse button is released. Knowing this, it is easy to calculate the Cartesian equation of the line or the vector parallel to P_1P_2 . Assume for the sake of argument that the line has

equation $y = mx + b$. To perform this reflection it is helpful to first derive the equations for reflecting a point about the line $y = mx$. To do this, assume that $\tan^{-1}(m) = \alpha$, and that the point you wish to reflect has polar coordinates $P[r; \theta]$ and Cartesian coordinates (x, y) (see **Figure 2** below).

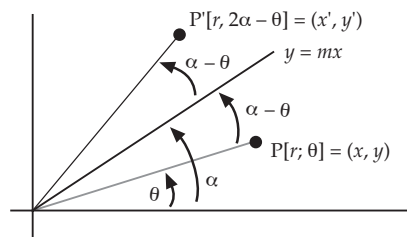


FIGURE 2.

REFLECTING AROUND THE LINE $y = mx$

Since the angle between OP and the line $y = mx$ is $\alpha - \theta$, the angle between OP' , where P' is the image of P , is also $\alpha - \theta$ so the reflected point OP' has polar coordinates $P'[r, \alpha - \theta + \alpha - \theta] = P'[r, 2\alpha - \theta]$. Converting to Cartesian coordinates, you get $P' = (x', y') = (r \cos(2\alpha - \theta), r \sin(2\alpha - \theta))$. Applying the double angle formulas to the preceding yields

$$\begin{aligned} x' &= r \cos(2\alpha - \theta) \\ &= r(\cos(2\alpha) \cos(\theta) + \sin(2\alpha) \sin(\theta)) \\ &= r \cos(2\alpha) \cos(\theta) + r \sin(2\alpha) \sin(\theta) \\ &= \cos(2\alpha)(r \cos(\theta)) + \sin(2\alpha)(r \sin(\theta)) \\ &= \cos(2\alpha)x + \sin(2\alpha)y. \end{aligned}$$

Using an analogous argument you can show that

$$y' = \sin(2\alpha)x - \cos(2\alpha)y$$

Expressing this in matrix form,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(2\alpha) & \sin(2\alpha) & 0 \\ \sin(2\alpha) & -\cos(2\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

To reflect around $y = mx + b$, all you need to do is first shift the origin to $(0, -b)$, reflect around $y = mx$, and then translate back. Expressing this in matrix form and calculating using Mathematica gives you **Figure 3**.

Another way to develop these equations is through the use of some of the properties of reflection. In this case we will reflect around any line $y = mx + b$. In figure 1, $P'(x', y')$ is the reflected image of $P(x, y)$. The following facts about reflections will be useful in what follows and are easy to prove.

- a) The midpoint M of PP' must lie on the reflection line, so

$$\frac{y + y'}{2} = m\left(\frac{x + x'}{2}\right) + b$$

- b) The segment PP' must be perpendicular to the reflection line, so

$$\frac{y - y'}{x - x'} = -\frac{1}{m}$$

This gives you two equations in two unknowns, which could be solved by hand but can easily be done using a CAS as we have done using Mathematica in **Figure 4**.

Cleaning the preceding up a bit, letting $xx = x'$ and $yy = y'$ and expressing it in matrix notation gives

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1-m^2}{m^2+1} & \frac{-2m}{m^2+1} & \frac{-2mb}{m^2+1} \\ \frac{-2m}{m^2+1} & \frac{m^2-1}{m^2+1} & \frac{2b}{m^2+1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

A third way to drive the reflection formulas is to make use of the formulas for reflecting around the x -axis. The image of the point (x, y) when reflected about the x -axis is $(x, -y)$. Expressed in matrix form this becomes

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ -y \\ 1 \end{bmatrix}$$

To reflect a point about the line $y = mx + b$ do the following

1. Translate the origin to $(0, -b)$
2. Rotate the plane about the origin through an angle of α , where $\alpha = \tan^{-1}(m)$.
3. Reflect the rotated point about the x -axis.

Figure 3 shows a Mathematica session. The input `In[4]:` is a matrix multiplication of three matrices: a translation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix}$, a rotation matrix $\begin{pmatrix} \cos[2\alpha] & \sin[2\alpha] & 0 \\ \sin[2\alpha] & -\cos[2\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and another translation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{pmatrix}$. The output `Out[4]:` is the resulting matrix $\begin{pmatrix} \cos(2\alpha) & \sin(2\alpha) & -b\sin(2\alpha) \\ \sin(2\alpha) & -\cos(2\alpha) & \cos(2\alpha)b+b \\ 0 & 0 & 1 \end{pmatrix}$.

FIGURE 3.

Figure 4 shows a Mathematica session. The input `In[52]:` is `Solve[{(y + yy) / 2 == m ((x + xx) / 2) + b, (y - yy) / (x - xx) == (-1 / m)}, {xx, yy}]`. The output `Out[52]:` is a list of rules: $\{ \{xx \rightarrow -\frac{x m^2 + 2 b m - 2 y m - x}{m^2 + 1}, yy \rightarrow -\frac{-y m^2 - 2 x m - 2 b + y}{m^2 + 1}\} \}$.

FIGURE 4.

Figure 5 shows a Mathematica session. The input `In[7]:` is a matrix multiplication of three matrices: a translation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix}$, a rotation matrix $\begin{pmatrix} \cos[\alpha] & -\sin[\alpha] & 0 \\ \sin[\alpha] & \cos[\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and another translation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. The output `Out[7]:` is the resulting matrix $\begin{pmatrix} \cos^2(\alpha) - \sin^2(\alpha) & 2 \cos(\alpha) \sin(\alpha) & -2 b \cos(\alpha) \sin(\alpha) \\ 2 \cos(\alpha) \sin(\alpha) & \sin^2(\alpha) - \cos^2(\alpha) & b - b(\sin^2(\alpha) - \cos^2(\alpha)) \\ 0 & 0 & 1 \end{pmatrix}$.

FIGURE 5.

Figure 6 shows a Mathematica session. The input `In[1]:` is `$Line = 0`. The output `Out[0]:` is `0`. The input `In[1]:` is `reflect[α_] :=` followed by a matrix multiplication of three matrices: a rotation matrix $\begin{pmatrix} \cos[\alpha] & -\sin[\alpha] & 0 \\ \sin[\alpha] & \cos[\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix}$, a translation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and another rotation matrix $\begin{pmatrix} \cos[-\alpha] & -\sin[-\alpha] & 0 \\ \sin[-\alpha] & \cos[-\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix}$. The input `In[2]:` is `translate[x_, y_] :=` followed by a matrix multiplication of three matrices: $\begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}$. The input `In[3]:` is `Simplify[translate[x1, y1].reflect[α].translate[-x1, -y1]]`. The output `Out[3]:` is a matrix $\begin{pmatrix} \cos(2\alpha) & \sin(2\alpha) & 2 \sin(\alpha) (x1 \sin(\alpha) - y1 \cos(\alpha)) \\ \sin(2\alpha) & -\cos(2\alpha) & 2 \cos(\alpha) (y1 \cos(\alpha) - x1 \sin(\alpha)) \\ 0 & 0 & 1 \end{pmatrix}$. The input `In[4]:` is `Simplify[translate[x1, y1].reflect[α].translate[-x1, -y1].` followed by a vector $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. The output `Out[4]:` is a vector $\begin{pmatrix} x1 + (x - x1) \cos(2\alpha) + (y - y1) \sin(2\alpha) \\ y1 + (y1 - y) \cos(2\alpha) + (x - x1) \sin(2\alpha) \\ 1 \end{pmatrix}$.$

FIGURE 6.

4. Rotate the plane about the origin through an angle of α .
5. Translate by $(0, b)$.

Expressing this in matrix form and using Mathematica yields **Figure 5**.

Apply the double angle formulas to the output and you get the same result we got earlier.

I suspect that the Reflection Tool in Freehand does not use the $y = mx + b$ form because it can't be used to reflect about vertical lines. Going back to how the tool actually works, we need to come up with formulas that take as input the coordinates of the two points $P_1(x_1, y_1)$ where the mouse is held down and the point $P_2(x_2, y_2)$ where the mouse is released. Here is what I think happens after P_2 is defined.

1. The angle $\alpha = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right)$ is calculated.
2. The origin is shifted to $P_1(x_1, y_1)$.
3. The point is reflected using the matrix we developed for reflecting around the line $y = \tan(\alpha)x$ (or if $\alpha = \pi/2$ the line $x = 0$).
4. The origin is shifted back.

To do this it is really helpful to be able to define matrix functions with parameters. **Figure 6** shows you can use Mathematica to enter the reflection matrix we just defined and give it the name reflect as well as a translation matrix.

So this gives you another way to reflect points: Find two points on the reflection line P_1 and P_2 and use the formulas

$$x' = \cos(2t)(x - x_1) + \sin(2t)(y - y_1) + x_1$$

$$y' = \sin(2t)(x - x_1) - \cos(2t)(y - y_1) + y_1$$

One final thought about reflections for those of you who use vectors. There is a very nice way to derive the reflection formulas using vectors and the projection of one vector on another. Here is a brief outline. In **Figure 7** below, O is the origin, Q is some point on the line $y = mx$, P is the point you want to reflect, and P' is the image of P .

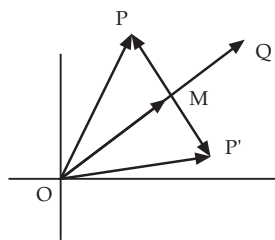


FIGURE 7. VECTOR OP' IS THE REFLECTION OF OP ABOUT THE VECTOR OQ

Let OM be the projection of OP on OQ which means that $OM = kOQ$, where $k = (OP \cdot OQ / OQ \cdot OQ)$, (\cdot stands for the dot product). The vector $MP = OP - OM$. Since P' is the reflection of P , $MP' = -MP$. Therefore,

$$\begin{aligned} OP' &= OM + MP' \\ &= OM - MP \\ &= OM - (OP - OM) \\ &= 2OM - OP. \end{aligned}$$

This method is neat and shows the real power of using vectors. To see this assume that $OQ = (2, 1)$ and $OP' = (x, y)$ then

$$\begin{aligned} OM &= ((2,1) \cdot (x, y) / (2, 1) \cdot (2, 1))(2, 1) \\ &= ((2x + y)/5)(2, 1) \\ &= ((4x + 2y)/5, (2x + y)/5) \end{aligned}$$

and

$$\begin{aligned} OP' &= 2OM - OP \\ &= 2((4x + 2y)/5, (2x + y)/5) - (x, y) \\ &= ((8x + 4y)/5, (4x + 2y)/5) - (x, y) \\ &= ((3x + 4y)/5, (4x - 3y)/5) \end{aligned}$$

Expressed in matrix form this becomes

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ \frac{4}{5} & \frac{-3}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Equating all the preceding expressions for reflecting about a given line is the source of some nice trig exercises that I encourage you all to try. Also, armed with the rotation formulas, you can begin to analyze algebraically how reflections combine. For example, you can show that

- a) reflection in two parallel lines is a translation, and the order in which you reflect about the lines makes a difference.
- b) reflection in two intersecting lines is a rotation, and the order in which you reflect about the lines makes a difference. In order to find the center of rotation C note that C must be a fixed point for the transformation

I hope the preceding will encourage some of you to play more with transformations, and to start using computer algebra systems if you haven't already done so. \square

