Shear:

A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a shear. Two common shearing transformations are those that shift coordinate x values and those that shift y values.

An x-direction shear relative to the x axis is produced with the transformation matrix

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-53}$$

which transforms coordinate positions as

$$x' = x + sh_x \cdot y, \qquad y' = y \tag{5-54}$$

Any real number can be assigned to the shear parameter sh_x. A coordinate position (x, y) is then shifted horizontally by an amount proportional to its distance (y value) from the x axis ($y = 0$). Setting sh_x to 2, for example, changes the square in Fig. 5-23 into a parallelogram. Negative values for sh_x shift coordinate positions to the left.

We can generate x-direction shears relative to other reference lines with

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-55}$$

with coordinate positions transformed as

$$x' = x + sh_x(y - y_{ref}), \qquad y' = y \tag{5-56}$$

An example of this shearing transformation is given in Fig. 5-24 for a shear parameter value of 1/2 relative to the line $y_{ref} = -1$.

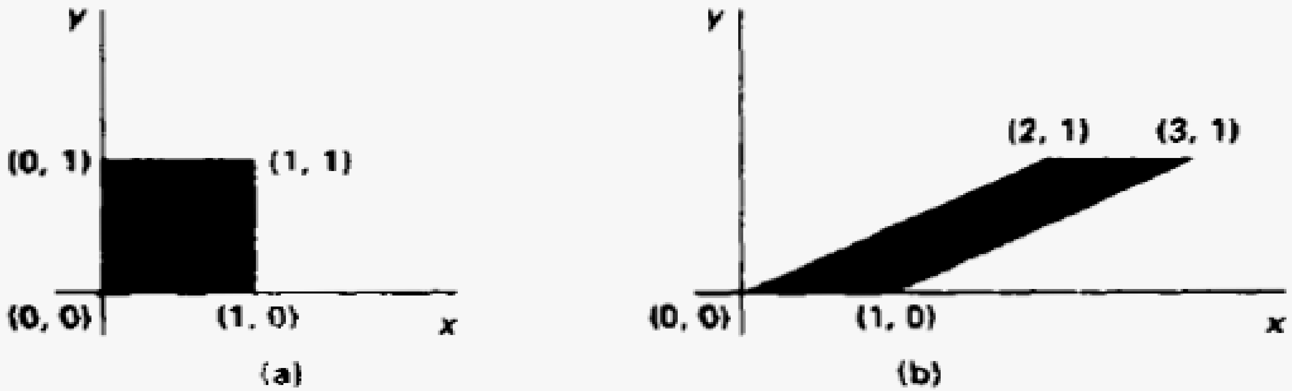

Figure 5-23
A unit square (a) is converted to a parallelogram (b) using the x-direction shear matrix 5-53 with $sh_x = 2$.

A y-direction shear relative to the line $x = x_{ref}$ is generated with the transformation matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{bmatrix} \tag{5-57}$$

which generates transformed coordinate positions

$$x' = x, \qquad y' = sh_y(x - x_{ref}) + y \tag{5-58}$$

This transformation shifts coordinate position vertically by an amount proportional to its distance from the reference line $x = x_{ref}$. Figure 5-25 illustrates the conversion of a square into a parallelogram with $sh_y = 1 / 2$ and $x_{ref} = -1$.

Shearing operations can be expressed as sequences of basic transformation. The x-direction shear matrix 5-53, for example, can be written as a composite transformation involving a series of rotation and scaling matrices that would scale the unit square of Fig. 5-23 along its diagonal, while maintaining the original lengths and orientations of edges parallel to the x axis. Shifts in the positions of objects relative to shearing reference lines are equivalent to translations.

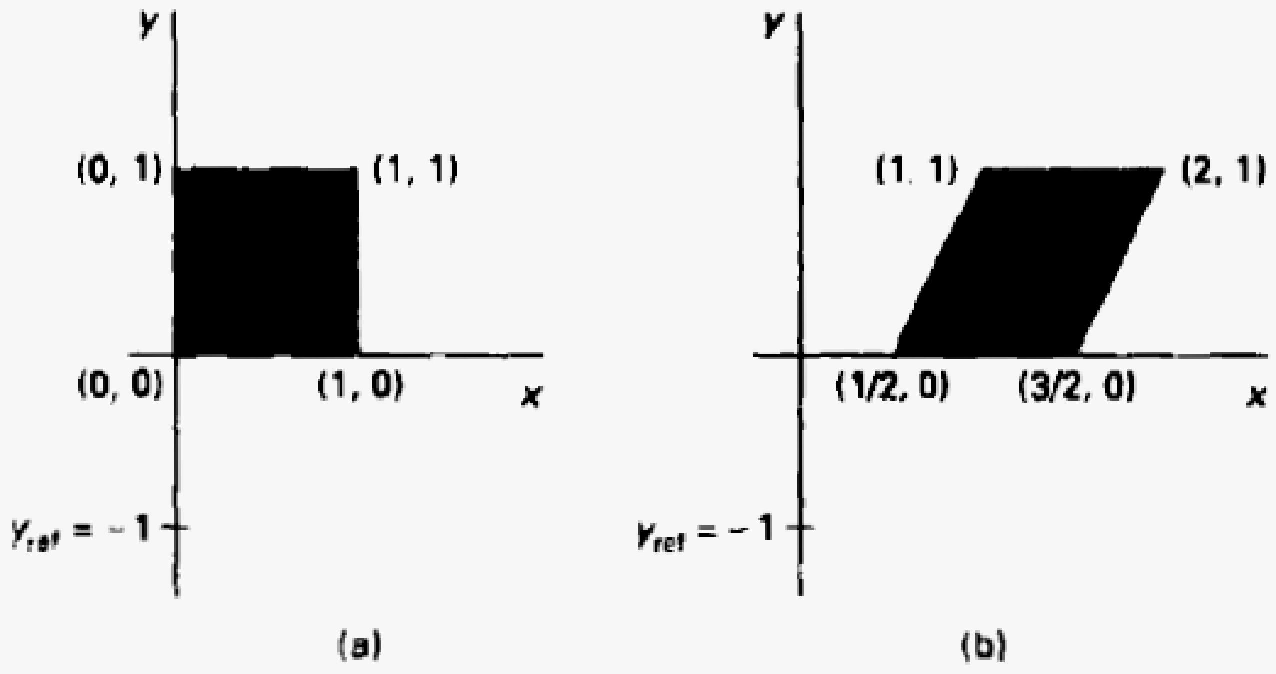

Figure 5-24
A unit square (a) is transformed to a shifted parallelogram (b) with $sh_x = 1/2$ and $y_{ref} = -1$ in the shear matrix 5-55.

Matrix Representations and Homogeneous Coordinates:

Many graphics applications involve sequences of geometric transformations. An animation, for example, might require an object to be translated and rotated at each increment of the motion. In design and picture construction applications, we perform translations, rotations, and scaling to fit the picture components into their proper positions. Here we consider how the matrix representations discussed in the previous sections can be reformulated so that such transformation sequences can be efficiently processed.

We have seen in Section 5-1 that each of the basic transformations can be expressed in the general matrix form $P' = M_1 * P + M_2$

with coordinate positions P and P' represented as column vectors. Matrix M1 is a 2 by 2 array containing multiplicative factors, and M2 is a two-element column matrix containing translational terms. For translation, M1 is the identity matrix.

For rotation or scaling, M2 contains the translational terms associated with the pivot point or scaling fixed point. To produce a sequence of transformations with these equations, such as scaling followed by rotation then translation, we must calculate the transformed coordinates one step at a time. First, coordinate positions are scaled, then these scaled coordinates are rotated, and finally the rotated coordinates are translated. A more efficient approach would be to combine the transformations so that the final coordinate positions are obtained directly from the initial coordinates, thereby eliminating the calculation of intermediate coordinate values. To be able to do this, we need to reformulate Eq. 5-15 to eliminate the matrix addition associated with the translation terms in M2.

We can combine the multiplicative and translational terms for two-dimensional geometric transformations into a single matrix representation by expanding the 2 by 2 matrix representations to 3 by 3 matrices. This allows us to express all transformation equations as matrix multiplications, providing that we also expand the matrix representations for coordinate positions. To express any two-dimensional transformation as a matrix multiplication, we represent each Cartesian coordinate position (x, y) with the homogeneous coordinate triple (xh, yh, h), where

$$x = \frac{x_h}{h}, \qquad y = \frac{y_h}{h} \tag{5-16}$$

Thus, a general homogenous coordinate representation can also be written as (h*x, h*y, h). For two-dimensional geometric transformations, we can choose the homogenous parameter h to be any nonzero value. Thus, there is an infinite number of equivalent homogeneous representations for each coordinate point (x, y).

A convenient choice is simply to set h = 1. Each two-dimensional position is then represented with homogeneous coordinates (x, y, 1). Other values for parameter h are needed, for example, in matrix formulations of three dimensional viewing transformations.

The term homogenous coordinates is used in mathematics to refer to the effect of this representation on Cartesian equations. When a Cartesian point (x, y) is converted to a homogeneous representation (xh, yh, h), equations containing x and y, such as f(x, y) = 0, become homogeneous equation in the three parameters xh, yh, and h. This just means that if each of the three parameters is replaced by any value v times that parameter, the value v can be factored out of the equations.

Expressing positions in homogeneous coordinates allows us to represent all geometric transformation equations as matrix multiplications. Coordinates are represented with three-element column vectors, and transformation operations are written as 3 by 3 matrices. For Translation. we have

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-17}$$

which we can write in the abbreviated form

$$\mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P} \tag{5-18}$$

with $\mathbf{T}(t_x, t_y)$ as the 3 by 3 translation matrix in Eq. 5-17. The inverse of the translation matrix is obtained by replacing the translation parameters t_x and t_y with their negatives: $-t_x$ and $-t_y$.

Similarly, rotation transformation equations about the coordinate origin are now written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-19}$$

or as

$$\mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P} \tag{5-20}$$

The rotation transformation operator $\mathbf{R}(\theta)$ is the 3 by 3 matrix in Eq. 5-19 with rotation parameter θ. We get the inverse rotation matrix when θ is replaced with $-\theta$.

Finally, a scaling transformation relative to the coordinate origin is now expressed as the matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-21}$$

or

$$\mathbf{P}' = \mathbf{S}(s_x, s_y) \cdot \mathbf{P} \tag{5-22}$$

where S(sx, sy) is the 3 by 3 matrix in Eq. 5-21 with parameters sx and sy. Replacing these parameters with their multiplicative inverses (1/sx, 1/sy) yields yields the inverse scaling matrix.

Matrix representations are standard methods for implementing transformations in graphics systems. In many systems, rotation and scaling functions produce transformations with respect to the coordinate origin, as in Eqs. 5-19 and 5-21. Rotations and scalings relative to other reference positions are then handled as a succession of transformation operations. An alternate approach in a graphics package is to provide parameters in the transformation functions for the scaling fixed-point coordinates and the pivot-point coordinates General rotation and scaling matrices that include the pivot or fixed point are then set up directly without the need to invoke a succession of transformation functions.