

## Computer Graphics

Transformations  
Part One

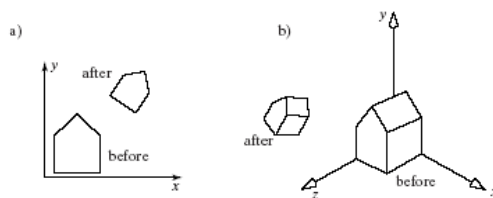
## Transformations

- We used the window to viewport transformation to scale and translate objects in the world window to their size and position in the viewport.
- We want to build on this idea, and gain more flexible control over the size, orientation, and position of objects of interest.
- To do so, we will use the powerful **affine transformation**.

Middle Tennessee State University

## Example of Affine Transformations

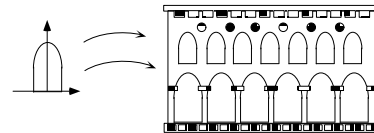
- The house has been scaled, rotated and translated, in both 2D and 3D.



Middle Tennessee State University

## Using Transformations

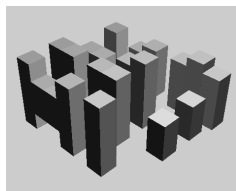
- The arch is designed in its own coordinate system.
- The scene is drawn by placing a number of instances of the arch at different places and with different sizes.



Middle Tennessee State University

## Using Transformations (2)

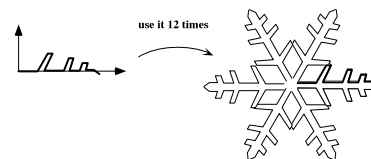
- In 3D, many cubes make a city.



Middle Tennessee State University

## Using Transformations (3)

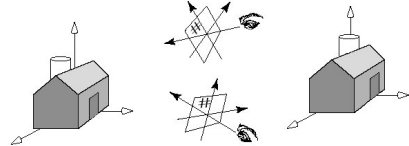
- The snowflake exhibits symmetries.
- We design a single **motif** and draw the whole shape using appropriate reflections, rotations, and translations of the motif.



Middle Tennessee State University

## Using Transformations (4)

- A designer may want to view an object from different vantage points.
- Positioning and reorienting a camera can be carried out through the use of 3D affine transformations.



Middle Tennessee State University

## Using Transformations (5)

- In a computer animation, objects move.
- We make them move by translating and rotating their local coordinate systems as the animation proceeds.
- A number of graphics platforms, including OpenGL, provide a graphics pipeline: a sequence of operations which are applied to all points that are sent through it.
- A drawing is produced by processing each point.

Middle Tennessee State University

## Graphics Pipeline (2)

- An application sends the pipeline a sequence of points  $P_1, P_2, \dots$  using commands such as:  

```
glBegin(GL_LINES);
glVertex3f(...); // send P1 through the pipeline
glVertex3f(...); // send P2 through the pipeline
...
glEnd();
```
- These points first encounter a transformation called the **current transformation (CT)**, which alters their values into a different set of points, say  $Q_1, Q_2, Q_3$ .

Middle Tennessee State University

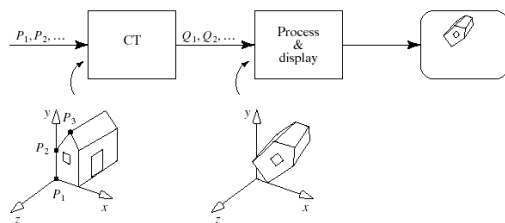
## Graphics Pipeline (3)

- Just as the original points  $P_i$  describe some geometric object, the points  $Q_i$  describe the transformed version of the same object.
- These points are then sent through additional steps, and ultimately are used to draw the final image on the display.

Middle Tennessee State University

## The OpenGL Graphics Pipeline

- This version is simplified.



Middle Tennessee State University

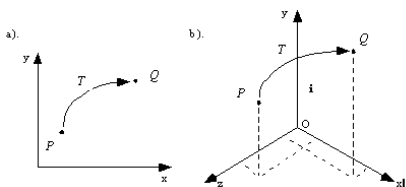
## Transformations

- Transformations change 2D or 3D points and vectors, or change coordinate systems.
  - An **object transformation** alters the coordinates of each point on the object according to the same rule, leaving the underlying coordinate system fixed.
  - A **coordinate transformation** defines a new coordinate system in terms of the old one, then represents all of the object's points in this new system.
- Object transformations are easier to understand, so we will do them first.

Middle Tennessee State University

### Transformations (2)

- A (2D or 3D) transformation  $T()$  alters each point,  $P$  into a new point,  $Q$ , using a specific formula or algorithm:  $Q = T(P)$ .



Middle Tennessee State University

### Transformations (3)

- An arbitrary point  $P$  in the plane is **mapped** to  $Q$ .
- $Q$  is the **image** of  $P$  under the mapping  $T$ .
- We transform an object by transforming each of its points, using the *same* function  $T()$  for each point.
- The **image** of line  $L$  under  $T$ , for instance, consists of the images of *all* the individual points of  $L$ .

Middle Tennessee State University

### Transformations (4)

- Most mappings of interest are continuous, so the image of a straight line is still a connected curve of some shape, although it's not necessarily a straight line.
- Affine transformations, however, *do* preserve lines: the image under  $T$  of a straight line is also a straight line.

Middle Tennessee State University

### Transformations (5)

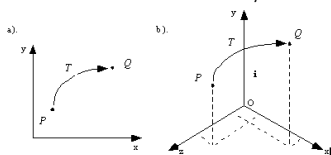
- We use an explicit coordinate frame when performing transformations.
- A coordinate frame consists of a point  $O$ , called the **origin**, and some mutually perpendicular vectors (called  $\mathbf{i}$  and  $\mathbf{j}$  in the 2D case;  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  in the 3D case) that serve as the axes of the coordinate frame.
- In 2D,

$$\tilde{P} = \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}, \tilde{Q} = \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix}$$

Middle Tennessee State University

### Transformations (6)

- Recall that this means that point  $P$  is at location  $= P_x \mathbf{i} + P_y \mathbf{j} + O$ , and similarly for  $Q$ .
- $P_x$  and  $P_y$  are the coordinates of  $P$ .
- To get from the origin to point  $P$ , move amount  $P_x$  along axis  $\mathbf{i}$  and amount  $P_y$  along axis  $\mathbf{j}$ .



Middle Tennessee State University

### Transformations (7)

- Suppose that transformation  $T$  operates on any point  $P$  to produce point  $Q$ :
- $\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = T \left( \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} \right)$  or  $Q = T(P)$ .
- $T$  may be any transformation: e.g., after transformation  $T$ :

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(P_x) e^{-P_x} \\ \ln(P_y) \\ 1 + P_x^2 \end{pmatrix}$$

Middle Tennessee State University

## Transformations (8)

- To make **affine** transformations we restrict ourselves to much simpler families of functions, those that are *linear* in  $P_x$  and  $P_y$ .
- Affine transformations make it easy to scale, rotate, and reposition figures.
- Successive affine transformations can be combined into a **single overall affine transformation**.

Middle Tennessee State University

## Affine Transformations

- Affine transformations have a compact matrix representation.
- The matrix associated with an affine transformation operating on 2D vectors or points must be a three-by-three matrix.
  - This is a direct consequence of representing the vectors and points in **homogeneous coordinates**.

Middle Tennessee State University

## Affine Transformations (2)

- Affine transformations have a simple form.
- Because the coordinates of  $Q$  are *linear combinations* of those of  $P$ , the transformed point may be written in the form:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11}P_x + m_{12}P_y + m_{13} \\ m_{21}P_x + m_{22}P_y + m_{23} \\ 1 \end{pmatrix}$$

Middle Tennessee State University

## Affine Transformations (3)

- There are six given constants:  $m_{11}$ ,  $m_{12}$ , etc.
- The coordinate  $Q_x$  consists of portions of both  $P_x$  and  $P_y$ , and so does  $Q_y$ .
- This *combination* between the  $x$ - and  $y$ -components also gives rise to rotations and shears.

Middle Tennessee State University

## Affine Transformations (4)

- Matrix form of the affine transformation in 2D:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

- For a 2D affine transformation the third row of the matrix is always  $(0, 0, 1)$ .

Middle Tennessee State University

## Affine Transformations (6)

- Vectors can be transformed as well as points.
- If a 2D vector  $\mathbf{v}$  has coordinates  $V_x$  and  $V_y$  then its coordinate frame representation is a column vector with third component 0.

$$\begin{pmatrix} V_x \\ V_y \\ 0 \end{pmatrix}$$

Middle Tennessee State University

## Affine Transformations (7)

- When vector  $\mathbf{V}$  is transformed by the same affine transformation as point  $P$ , the result is

$$\begin{pmatrix} W_x \\ W_y \\ 0 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ 0 \end{pmatrix}$$

- Important:** to transform a point  $P$  into a point  $Q$ , *post-multiply*  $M$  by  $P$ :  $Q = M P$ .

Middle Tennessee State University

## Affine Transformations (8)

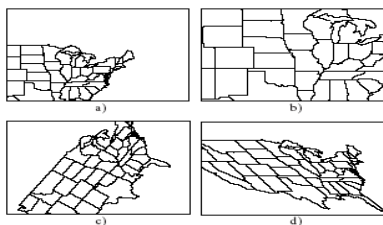
- Example: find the image  $Q$  of point  $P = (1, 2, 1)$  using the affine transformation

$$M = \begin{pmatrix} 3 & 0 & 5 \\ -2 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}; Q = \begin{pmatrix} 8 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 5 \\ -2 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

Middle Tennessee State University

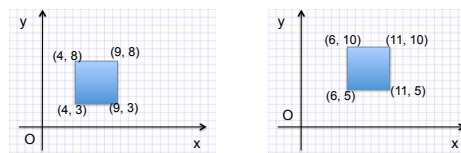
## Geometric Effects of Affine Transformations

- Combinations of four elementary transformations
  - (a) **translation**, (b) **scaling**, (c) **rotation**, and (d) **shear**



Middle Tennessee State University

## Translation



Translate the square by 2 along x axis and 2 along y axis

Middle Tennessee State University

## Translations

- To translate a point  $P$  by  $a$  in the  $x$  direction and  $b$  in the  $y$  direction use the matrix:

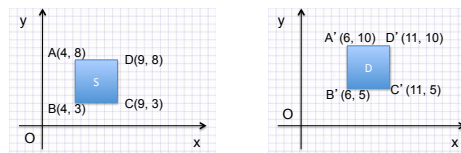
$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} = \begin{pmatrix} P_x + a \\ P_y + b \\ 1 \end{pmatrix}$$

- Only using homogeneous coordinates allow us to include translation as an affine transformation.
- It is meaningless to translate vectors.

Middle Tennessee State University

## Translation

Translate the square by 2 along x axis and 2 along y axis



What is the transformation matrix for this translate?  
Apply it to point A to see if it arrives at point A'?

Middle Tennessee State University

## Scaling

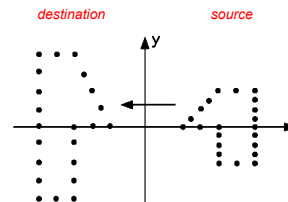
- Scaling is **about the origin**. If  $S_x = S_y$  the scaling is uniform; otherwise it distorts the image.
- If  $S_x$  or  $S_y < 0$ , the image is reflected across the y or x axis.
- The matrix form is

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

Middle Tennessee State University

## Scaling

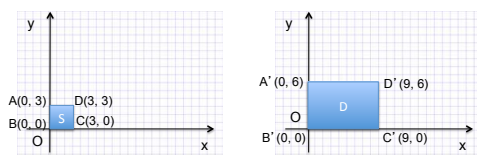
- The scaling  $(S_x, S_y) = (-1, 2)$  is applied to a collection of points. Each point is both reflected about the y-axis and scaled by 2 in the y-direction.



Middle Tennessee State University

## Scaling

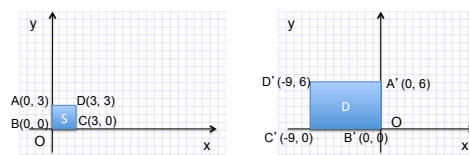
Scale the square by 3 along x axis and by 2 along y axis



What is the transformation matrix for this scale?  
Apply it to point D to see if it arrives at point D'?

Middle Tennessee State University

## Scaling

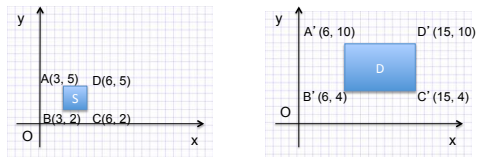


What is the transformation matrix for this scale?  
Apply it to point D to see if it arrives at point D'?

Middle Tennessee State University

## Practice Question

What is the transformation matrix?  
Apply it to point D to see if it arrives at point D'?



WILL COME BACK TO THIS ...

Middle Tennessee State University

## Types of Scaling

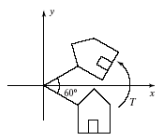
- Important:** Scaling is about the origin
- Pure reflections, for which each of the scale factors is  $+1$  or  $-1$ .
- A **uniform scaling**, or a magnification about the origin:  $S_x = S_y$ , magnification  $|S|$ .
  - Reflection also occurs if  $S_x$  or  $S_y$  is negative.
  - If  $|S| < 1$ , the points will be moved closer to the origin, producing a reduced image.
- If the scale factors are not the same, the scaling is called a **differential scaling**.

Middle Tennessee State University

## Rotation

- Counterclockwise around the z-axis, about the origin by angle  $\theta$ :

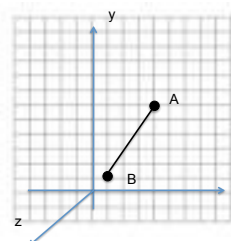
$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



Middle Tennessee State University

## Practice Question

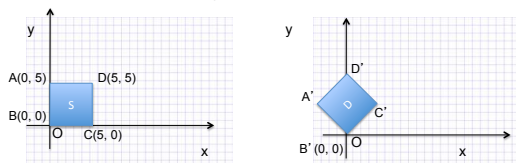
- Segment AB: Point A(4, 6), B(1, 2), rotate about the Z-axis for 30 degrees, compute the coordinates for the resulting points A'B'



Middle Tennessee State University

## Rotation

Rotate the square 45 degree counter-clockwise about Z-axis



What is the transformation matrix for this rotation?  
Apply it to points A, B, C, D to find the coordinates for A', B', C' and D'

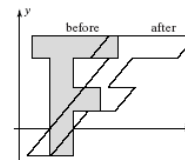
Middle Tennessee State University

## Shear

- Shear is the translation along an axis(say, X axis) by an amount that increases linearly with another axis (Y).

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & h & 0 \\ g & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

- Shear along x:  $h \neq 0$ , and  $P_x$  depends on  $P_y$  (for example, *italic* letters).
- Shear along y:  $g \neq 0$ , and  $P_y$  depends on  $P_x$ .

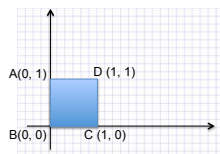


## Practice Question

- Given an unit square having its lower left corner on the origin point, what is the square after the following shear transformations?

$$(a) \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(b) \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Middle Tennessee State University

## Inverses of Affine Transformations

- $\det(M) = m_{11} * m_{22} - m_{21} * m_{12} \neq 0$  means that the inverse of a transformation exists.  
– That is, the transformation can be "undone".
- $M M^{-1} = M^{-1} M = I$ , the identity matrix.

Middle Tennessee State University

## Inverse Translation and Scaling

- Inverse of translation  $T^{-1}$ :

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

- Inverse of scaling  $S^{-1}$ :

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

## Inverse Rotation and Shear

- Inverse of rotation  $R^{-1} = R(-\theta)$ :

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

- Inverse of shear  $H^{-1}$ : generally  $h=0$  or  $g=0$ .

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -h & 0 \\ -g & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} \frac{1}{1-gh}$$

Middle Tennessee State University

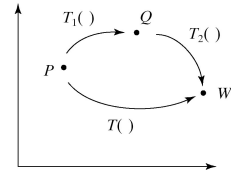
## Composing Affine Transformations

- Usually, we want to apply several affine transformations in a particular order to the figures in a scene: for example,
  - translate by (3, -4)
  - then rotate by  $30^\circ$
  - then scale by (2, -1) and so on.
- Applying successive affine transformations is called **composing** affine transformations.

Middle Tennessee State University

## Composing Affine Transformations (2)

- $T_1()$  maps  $P$  into  $Q$ , and  $T_2()$  maps  $Q$  into point  $W$ . Is  $W = T_2(Q) = T_2(T_1(P))$  affine?
- Let  $T_1=M_1$  and  $T_2=M_2$ , where  $M_1$  and  $M_2$  are the appropriate matrices.
- $W = M_2(M_1P)$   
 $= (M_2M_1)P$   
 $= MP$  by associativity.
- So  $M = M_2M_1$ , the product of 2 matrices (in reverse order of application), which is affine.



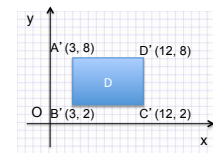
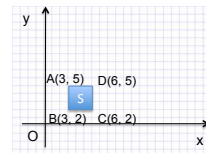
## Composing Affine Transformations: Examples

- For a point  $P$  to rotate around an arbitrary point  $V$ : translate  $V$  to the origin, rotate, translate  $V$  back to its original position:  
 $Q = T_V R T_{-V} P$
- Shear around an arbitrary point:  
 $Q = T_V H T_{-V} P$
- Scale about an arbitrary point:  
 $Q = T_V S T_{-V} P$

Middle Tennessee State University

## Translation + Scaling

What is the transformation matrix?  
 Apply it to point D to see if it arrives at point D'?

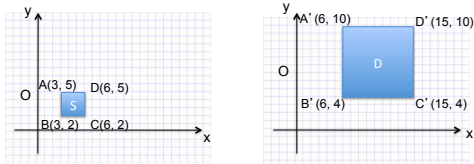


Middle Tennessee State University



## Translation + Scaling

What is the transformation matrix?  
Apply it to point D to see if it arrives at point D'?



Middle Tennessee State University

## Rotation about a Point

- Point  $P=(1, 2)$  rotates about point  $V=(-2, 3)$  around the Z-axis for 30 degrees.
  - Compute the Composite Matrix representing this transformation
  - Compute the coordinates for the image point Q

Middle Tennessee State University

## Composing Affine Transformations (Examples)

- Reflect across an arbitrary line through the origin O:

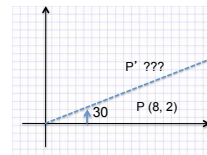
$$Q = R(\theta) S R(-\theta) P$$

- The rotation transforms the axis to the x-axis, the reflection is a scaling, and the last rotation transforms back to the original axis.

Middle Tennessee State University

## Practice Question

- Find the point  $P(8, 2)$  reflected across the line  $l$  from origin that is 30 degree from x-axis



Middle Tennessee State University

## Practice Question

For a line segment A(2, 1) and B(4, 1):

- What transformation matrix is involved in scaling this line by 2 along the X axis?
- What are the resulting coordinates of the two points on the transformed line segment?
- Given code:

```
void Draw_Line()
{
    glBegin(GL_LINE);
    glVertex3f(-1, 0, 0);
    glVertex3f(1, 0, 0);
    glEnd();
}
```

Middle Tennessee State University

## Fixed point scaling

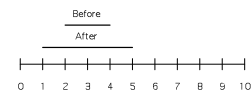
Scale by 2 with fixed point = (3,1)

- Translate the point (3,1) to the origin
- Scale by 2
- Translate origin to point (3,1)

$$\begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -3 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 1 \end{pmatrix}$$



Middle Tennessee State University

## More on Fixed Point Scaling

What if the fixed point is not in the middle of the line?

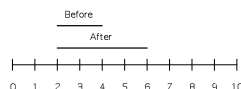
Scale by 2 with fixed point = (2,1)

1. Translate the point (2,1) to the origin
2. Scale by 2
3. Translate origin to point (2,1)

$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -2 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & -2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & -2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \end{pmatrix}$$

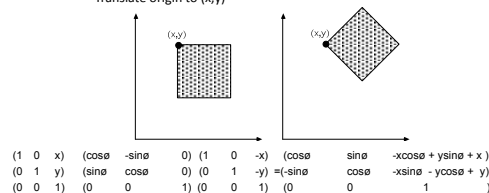


Middle Tennessee State University

## Hint on Rotation about a fixed point

Rotation Of  $\theta$  Degrees About Point (x,y)

- Translate (x,y) to origin
- Rotate
- Translate origin to (x,y)



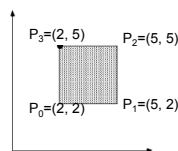
You rotate the box by rotating each vertex.

Middle Tennessee State University

## Practice Question

Given the square as shown, show the square after it is rotated 45 degree around  $P_3$ :

- What individual transformation matrices are involved?
- What is the composite transformation matrix?
- What are the coordinates of the 4 points of the square after the transformation?

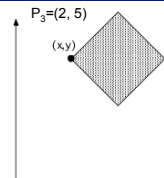


Middle Tennessee State University

## Practice Question

Given code to draw a square as:

```
void DrawSquare()
{
    // first point on origin, length=3
    glBegin(GL_POLYGON);
    glVertex3f(0, 0, 0);
    glVertex3f(3, 0, 0);
    glVertex3f(3, -3, 0);
    glVertex3f(0, -3, 0);
    glEnd();
}
```



- To draw the square as shown, i.e., 45 degree rotated around  $P_3$  and translated to the final position,
- What are the individual transformation involved?
  - What is the composite transformation matrix?

Middle Tennessee State University

## Practice Question

- Given the unit square centered at the origin point, what does the square look like and where is it located in the coordinate system after the following sequence of transformations (in the order as shown below):
  - Translate the figure along X-axis by 3, along Y-axis by 2
  - Rotate the figure 45 degrees along the Z-axis
  - Scale along X-axis by a factor of 3, along Y-axis by 2
- Compute the composite transformation matrix
- Compute the corner points of the resulting shape

Middle Tennessee State University

## Properties of 2D and 3D Affine Transformations

- Affine transformations *preserve* affine combinations of points.
  - $W = a_1P_1 + a_2P_2$  is an affine combination.
  - $MW = a_1MP_1 + a_2MP_2$
- Affine transformations preserve lines and planes.
  - A line through A and B is  $L(t) = (1-t)A + tB$ , an affine combination of points.
  - A plane can also be written as an affine combination of points:  $P(s, a) = sA + tB + (1-s-t)C$ .

Middle Tennessee State University

### Properties of Transformations (2)

- Parallelism of lines and planes is preserved.
  - Line  $A + \mathbf{b}t$  having direction  $\mathbf{b}$  transforms to the line given in homogeneous coordinates by  $M(A + \mathbf{b}t) = MA + M\mathbf{b}t$ , which has direction vector  $M\mathbf{b}$ .
  - $M\mathbf{b}$  does *not* depend on point  $A$ . Thus two different lines  $A_1 + \mathbf{b}t$  and  $A_2 + \mathbf{b}t$  that have the same direction will transform into two lines both having the direction, so they *are* parallel.
- An important consequence of this property is that *parallelograms map into other parallelograms*.

Middle Tennessee State University

### Properties of Transformations (3)

- The direction vectors for a plane also transform into new direction vectors independent of the location of the plane.
- As a consequence, parallelepipeds map into other parallelepipeds.

Middle Tennessee State University

### Properties of Transformations (4)

- The columns of the matrix reveal the transformed coordinate frame:
  - Vector  $\mathbf{i}$  transforms into column  $\mathbf{m}_1$ , vector  $\mathbf{j}$  into column  $\mathbf{m}_2$ , and the origin  $\mathbf{O}$  into point  $\mathbf{m}_3$ .
  - The coordinate frame  $(\mathbf{i}, \mathbf{j}, \mathbf{O})$  transforms into the coordinate frame  $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ , and these new objects are precisely the columns of the matrix.

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} = (\mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3)$$

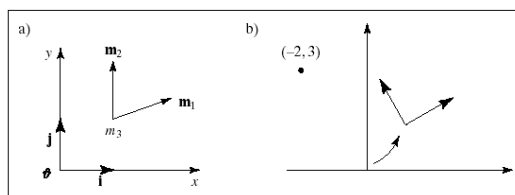
Middle Tennessee State University

### Properties of Transformations (5)

- The axes of the new coordinate frame are not necessarily perpendicular, nor must they be unit length.
  - They are still perpendicular if the transformation involves only rotations and uniform scalings.
- Any point  $P = P_x \mathbf{i} + P_y \mathbf{j} + \mathbf{O}$  transforms into  $Q = P_x \mathbf{m}_1 + P_y \mathbf{m}_2 + \mathbf{m}_3$ .

Middle Tennessee State University

### Properties of Transformations (6)

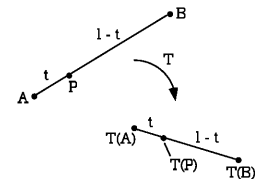


30° rotation to the left about point  $(-2, 3)$

Middle Tennessee State University

### Properties of Transformations (7)

- The transformed point,  $T(P)$ , lies the *same* fraction  $t$  of the way between images  $T(A)$  and  $T(B)$ .
- Relative ratios are preserved: consider point  $P$  lying a fraction  $t$  of the way between two given points,  $A$  and  $B$  (see figure).
- Apply affine transformation  $T(\ )$  to  $A$ ,  $B$ , and  $P$ .



### Properties of Transformations (8)

- How is the area of a figure affected by an affine transformation?
- It is clear that neither translations nor rotations have any effect on the area of a figure, but scalings certainly do, and shearing might.
- The result is simple: When the 2D transformation with matrix  $M$  is applied to an object, its area is multiplied by the *magnitude of the determinant* of  $M$ :

$$\frac{\text{area of transformed}}{\text{area before transformation}} = |\det M|$$

Middle Tennessee State University

### Properties of Transformations (9)

- In 2D the determinant of the matrix  $M$  is  $(m_{11}m_{22} - m_{12}m_{21})$ .
- For a pure scaling, the new area is  $S_x S_y$  times the original area, whereas for a shear along one axis the new area is the same as the original area.
- In 3D similar arguments apply, and we can conclude that the volume of a 3D object is scaled by  $|\det M|$  when the object is transformed by the 3D transformation based on matrix  $M$ .

Middle Tennessee State University

### Properties of Transformations (10)

- Every affine transformation is composed of elementary operations.
- A matrix may be factored into a product of elementary matrices in various ways. One particular way of factoring the matrix associated with a 2D affine transformation yields  
 $M = (\text{shear})(\text{scaling})(\text{rotation})(\text{translation})$
- That is, any  $3 \times 3$  matrix that represents a 2D affine transformation can be written as the product of (reading right to left) a translation matrix, a rotation matrix, a scaling matrix, and a shear matrix.

Middle Tennessee State University