TWO-DIMENSIONAL TRANSFORMATIONS
(Chapter 5 in Computer Graphics)

• Basic Philosophy
• Basic Transformations
• Matrix Representations and Homogeneous Coordinates
• Composite Transformations
• Other Transformations
• Transformation Commands
• Raster Methods for Transformations
Basic Philosophy

- objects are defined by sets of vertices
- objects are transformed by transforming their vertices
- straight lines stay straight
Basic Transformations

- translation (reposition)
- scaling (reduce or enlarge)
- rotation (reorient)
translation

- straight line movement from one position to another
- \( x' = x + T_x \)
  \( y' = y + T_y \)
- translate objects by adding the translation vector to the coordinates of each endpoint
scaling

- altering the size of an object
- \[ x' = x \cdot Sx \]
  \[ y' = y \cdot Sy \]

- scale objects by multiplying the coordinates of each endpoint by the scaling factors
scaling, continued

- lengths and distances from the origin are scaled

- one point \((x_F, y_F)\) can remain fixed in position

\[
x' = x_F + (x - x_F)S_x \\
y' = y_F + (y - y_F)S_y
\]
rotation

- transformation along circular paths
  - $x' = x \cos \theta - y \sin \theta$
  - $y' = y \cos \theta + x \sin \theta$
- rotate objects by rotating each endpoint
rotation, continued

- rotation about an arbitrary pivot point \((x_r, y_r)\)
  - \(x' = x_r + (x - x_r)\cos\Theta - (y - y_r)\sin\Theta\)
  - \(y' = y_r + (y - y_r)\cos\Theta + (x - x_r)\sin\Theta\)

- efficient rotation for small angles
  - \(\cos\Theta = 1\)
  - \(\sin\Theta = \Theta\) in radians
Matrix Representations and Homogeneous Coordinates

- final coordinates are calculated directly from initial coordinates using matrix methods
- a homogeneous coordinates is added
  - \((x,y)\) becomes \([x \ y \ 1]\)
basic transformation matrices

- translation

\[
[x' \ y' \ 1] = [x \ y \ 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Tx & Ty & 1 \end{bmatrix}
\]

- scaling

\[
[x' \ y' \ 1] = [x \ y \ 1]\begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

- rotation

\[
[x' \ y' \ 1] = [x \ y \ 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]
inverses of basic transformation matrices

- translation

\[
[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

- scaling

\[
[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

- rotation

\[
[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} 0 \\ 0 \\ 0 & 0 & 1 \end{bmatrix}
\]
Composite Transformations

- a composite transformation matrix is the product of individual transformation matrices
- multiplying two matrices together is referred to as concatenation
two successive translations

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
Tx_1 & Ty_1 & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
Tx_2 & Ty_2 & 1
\end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
Tx_1 + Tx_2 & Ty_1 + Ty_2 & 1
\end{bmatrix}
\]
scaling relative to a fixed point

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-x_F & -y_F & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
S_x & 0 & 0 \\
0 & S_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
x_F & y_F & 1
\end{bmatrix}
= \\
\begin{bmatrix}
S_x & 0 & 0 \\
0 & S_y & 0 \\
(1 - S_x)x_F & (1 - S_y)y_F & 1
\end{bmatrix}
\]

(a) Original Position of Object and Fixed Point
(b) Translate Object so that Fixed Point \((x_F, y_F)\) Is at Origin
(c) Scale Object with Respect to Origin
(d) Translate Object so that Fixed Point IsReturned to Position \((x_F, y_F)\)
rotation about a pivot point

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-x_R & y_R & 1
\end{bmatrix} \cdot \begin{bmatrix}
\cos\theta & \sin\theta & 0 \\
-\sin\theta & \cos\theta & 0 \\
0 & 0 & 1
\end{bmatrix} \cdot \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
x_R & y_R & 1
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\cos\theta & \sin\theta & 0 \\
-\sin\theta & \cos\theta & 0 \\
(1 - \cos\theta)x_R + y_R \cdot \sin\theta & (1 - \cos\theta)y_R - x_R \cdot \sin\theta & 1
\end{bmatrix}
\]

(a) Original Position of Object and Pivot Point

(b) Translation of Object so that Pivot Point \( (x_R, y_R) \) is at Origin

(c) Rotation about Origin

(d) Translation of Object so that the Pivot Point \( (x_R, y_R) \) is returned to Position \( (x_R, y_R) \)
arbitrary scaling directions

- rotate so that the scaling axes coincide with the x and y axes
- scale
- rotate scaling axes back to their original positions
arbitrary scaling directions, continued

- example
  - $\Theta = 45^0$
  - $S_1 = 1$
  - $S_2 = 2$
concatenation properties

- matrix multiplication is associative
- matrix multiplication is not commutative
Other Transformations - reflection

- produce a mirror image relative to an axis of reflection
- reflection about the x axis

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

- reflection about the y axis

\[
\begin{bmatrix}
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

- reflection about the origin

\[
\begin{bmatrix}
-1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Other Transformations - reflection, continued

- reflection about the line $y = x$

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

- reflection about the line $y = -x$

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Other Transformations - shear

- produce a distortion
- x-direction shear

\[
\begin{bmatrix}
1 & 0 & 0 \\
SHx & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

- y-direction shear

\[
\begin{bmatrix}
1 & SHy & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Transformation Commands

- sometimes separate commands for each transformation operation
- sometimes a composite transformation command
  - create_transformation_matrix(xf, yf, sx, sy, xr, yr, a, tx, ty, matrix)
    assumes order
  - scale
  - rotate
  - translate
  - could be used for a single transformation
Transformation Commands, continued

- concatenation
  - accumulate_transformation_matrix (matrix_in, xf, yf, sx, sy, xr, yr, a, tx, ty, matrix_out)

- selecting previously defined matrices
  - set_transformation (matrix)
Raster Methods for Transformations

- some simple transformations can be carried out by manipulating the frame buffer contents

- translation (bit block transfer or bit-blt)
  - copy a block from one area of the frame buffer to another area of the frame buffer
  - fill the old area with background color
  - begin with an overlapped corner

- Boolean operations can be applied (exclusive-or is quite useful)

- rotation by 90° increments

- scaling by integer multiples
TWO-DIMENSIONAL TRANSFORMATIONS

- Basic Philosophy
- Basic Transformations
- Matrix Representations and Homogeneous Coordinates
- Composite Transformations
- Other Transformations
- Transformation Commands
- Raster Methods for Transformations