Geometrical Transformations

- 2-D x 3-D
- Linear
- Affine
- Projective
- Matrix Composition

Linear Transformation

- Satisfies the following conditions (for u, v $\in \Re^n$, $\lambda \in \Re$):
  - L(u+v) = L(u) + L(v) and L(λu) = λL(u)
- Examples of linear transformations: Rotation, Scaling and Shearing
- Preserve parallel lines

Matrix Representation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Affine Transformation

- A(u) is *affine* iff A(u) = L(u) + c, where u $\in \Re^n$, L is a linear transformation and c is a constant
- Linear transformations are a special case of affine transformations
- Example of affine transform: Translation
- Preserve parallel lines

Matrix Representation

- The left (green) sub-matrix represents a linear transformation (rotation, scaling, shearing)
- The right (yellow) sub-matrix represents translation
- Non-square matrices have no general inverse

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

It has no physical meaning

Homogeneous Coordinates (2-D)

- Provide a way to represent affine transformations using square matrices
- n-dimensional position vector represented with (n+1) coordinates

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

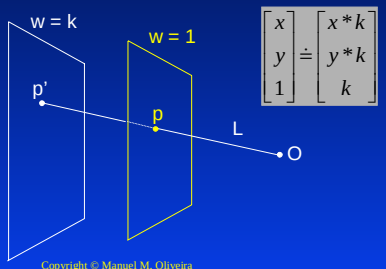
Homogeneous Coordinates

- Operates in projective space (point-line duality)
- The (n+1)th coordinate represents the plane upon which the transformation operates

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Projective Space

- All points along line L are projectively equivalent



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \doteq \begin{bmatrix} x*k \\ y*k \\ k \end{bmatrix}$$

Projective Transformation

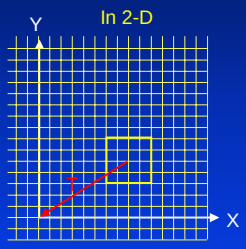
- If $[m_{31}\ m_{32}\ m_{33}] = [0\ 0\ 1]$ then orthographic projection (affine); otherwise, perspective projection
- Perspective projection only preserve parallel lines if they are parallel to the projection plane

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

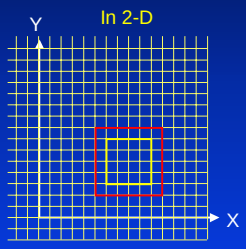
Translation

- x' = x + dx
- y' = y + dy
- z' = z + dz

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



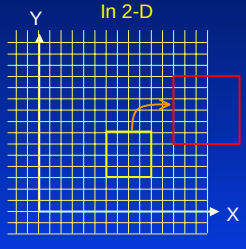
Scaling



Scaling

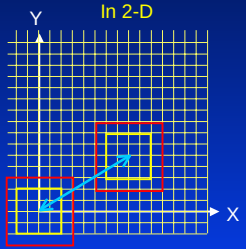
- $x' = S_x x$
- $y' = S_y y$
- $z' = S_z z$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Scaling Around a Point

- Translate the point to the origin, then scale and translate the origin back to the point



Shear (2-D)

- $x' = x + h_x y$
- $y' = y + h_y x$

$$H_x = \begin{bmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H_y = \begin{bmatrix} 1 & 0 & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad H_{xy} = \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

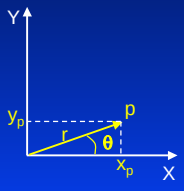
Shear (3-D)

- The same idea, but with more degrees of freedom
- Most interesting case: shear of x and y along z
- $x' = x + h_x z$
- $y' = y + h_y z$

$$H_{xy} = \begin{bmatrix} 1 & 0 & h_x & 0 \\ 0 & 1 & h_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

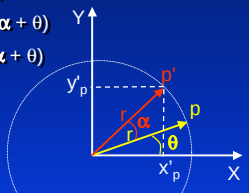
Rotation around the origin (2-D)

- $x_p = r \cos(\theta)$
- $y_p = r \sin(\theta)$



Rotation around the origin (2-D)

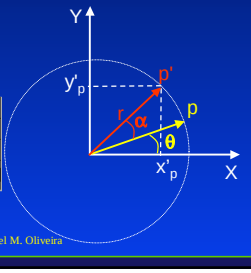
- $x_p = r \cos(\theta)$
- $y_p = r \sin(\theta)$
- $x'_p = r \cos(\alpha + \theta)$
- $y'_p = r \sin(\alpha + \theta)$



Rotation around the origin (2-D)

- $x'_p = r \cos(\alpha + \theta) = r (\cos(\alpha) \cos(\theta) - \sin(\alpha) \sin(\theta))$
- $y'_p = r \sin(\alpha + \theta) = r(\sin(\alpha) \cos(\theta) + \sin(\theta) \cos(\alpha))$
- $x'_p = x_p \cos(\alpha) - y_p \sin(\alpha)$
- $y'_p = x_p \sin(\alpha) + y_p \cos(\alpha)$

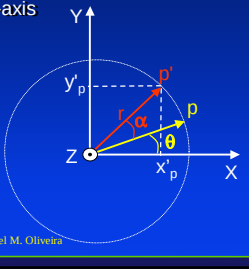
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Rotation (3-D)

- Rotations around the principal axis in 3-D can be derived from rotations in 2-D
- Example: rotation around Z-axis

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



Rotation (3-D)

- Rotation around X-axis

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

- Rotation around Y-axis

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

Rotation about arbitrary 3-D axis

- Difficult to compute the Euler angles that accomplish such rotation
- Easily accomplished using Quaternions

Matrix Composition

- More complex transformations are obtained by multiplying the matrices of the individual transformations
- Matrix multiplication is not commutative (AB ≠ BA), but it is associative ((AB)C = A(BC))
- Sometimes it is ok to switch the order of transformations:

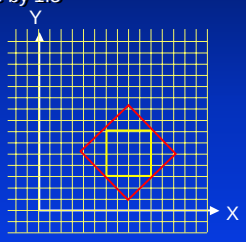
	T	Sc	R
T	ok	nok	nok
Sc	—	ok	ok*
R	—	—	ok

Orthogonal Matrices

- If all rows/columns of a matrix are LI and their lengths are all equal to 1, the matrix is called orthogonal
- The inverse of an orthogonal matrix is its transpose
- Rotation matrices are orthogonal
- Transformations involving only rotations and translations are called rigid body transformations

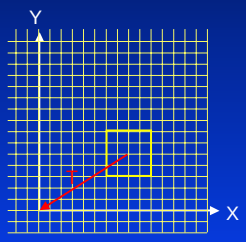
Composite Transformation

- Rotate the square by 45 degrees around its center and scale by 1.5



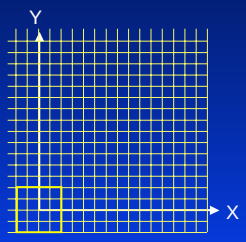
Composite Transformation

- Translate the center of the square to the origin (T)



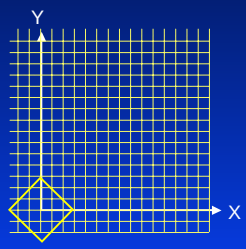
Composite Transformation

- Translate the center of the square to the origin (T)



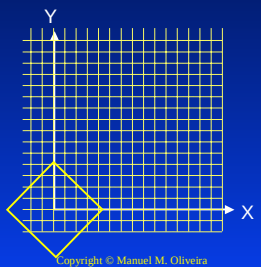
Composite Transformation

- Rotate (R) (or scale (S))



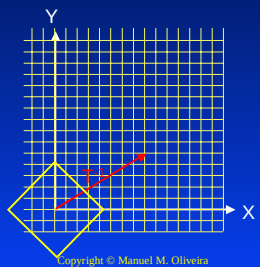
Composite Transformation

- Scale (S) (or rotate (R))



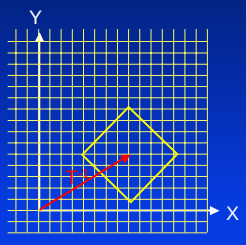
Composite Transformation

- Translate back (T^{-1})



Composite Transformation

- Composite: $T^{-1}SRT$ or $T^{-1}RST$



Composite Transformation

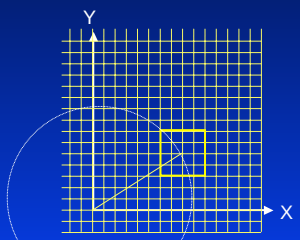
- Composite: $T^{-1}SRT$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 8 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -8 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Note: In OpenGL one specifies the transformation matrices in the reverse order they should be applied. In this case, one should first inform T^{-1} then S then R and finally T. They are composite as $(((I\ T^{-1})S)R)T$, producing the same result.

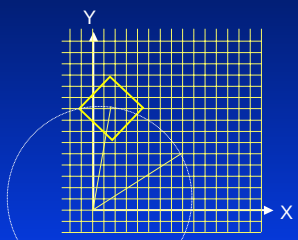
Composite Transformation

- Without translating to the origin: SR



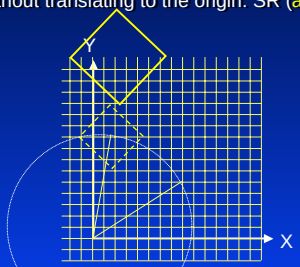
Composite Transformation

- Without translating to the origin: SR (after the rotation)



Composite Transformation

- Without translating to the origin: SR (after the scale)



Wrong result!