Transformations in 2-D

Dr. V Masilamani

masila@iiitdm.ac.in

Department of Computer Science and Engineering
IIITDM Kancheepuram
Chennai-127

Overview

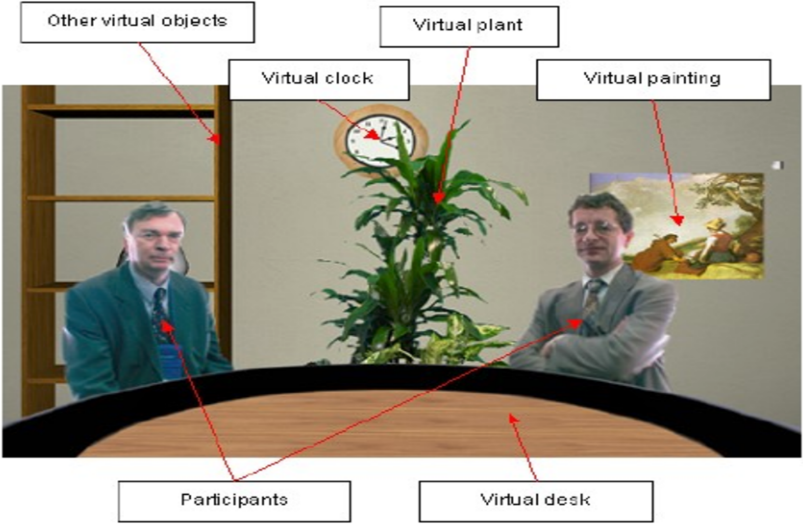
What is 2D Transform

- Coordinate System: (Origin, Axes), where the axes are basis vectors(eg. (1,0), (0,1))
- 2D-Coordinate System: (Origin, X, Y)
- Representation of 2D-Point: Given a coordinate system (Origin, X, Y), a 2D-point is represented as $\begin{bmatrix} x \\ y \end{bmatrix}$
- 2D-Transform: Transform that maps a 2D point to possibly another 2D-point

Need of 2D Transforms in Computer Graphics

- One of the objectives of computer graphics is to **simulate the manipulation of real world objects in images**
- The real world objects will undergo transformations
- Camera view or human view of such transformation is 2D transformation of objects
- Transformation of object is transformation of each of the points on the object.
- Application of 2D Transforms in computer graphics
  - To simulate the manipulation of objects
  - When each object is defined in its own coordinate system, to create scene with all those objects, all such objects need to be moved to a single coordinate system -This involves 2D transformation

Scene with objects moved from their own coordinate systems



Linear Transformation of 2D points:

- Linear Transformation : T is said to be linear if
  $T(ax + by) = aT(x) + bT(y)$
- Equivalent Definition of Linear Transform:
  T is said to be linear if
  - T(x+y)=T(x)+T(y)
  - T(ax)=aT(x)
- Characterization of Linear Transform: T is linear transform iff there exists a matrix A such that $T(X) = AX$, where $X = \begin{bmatrix} x \\ y \end{bmatrix}$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

x' = ax + cy
y' = bx + dy

$$\begin{bmatrix} x' \\ y' \end{bmatrix}^T = \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

x' = ax + cy
y' = bx + dy

Special cases of 2D Transformations:

- Identity Transform: $T(x, y) = (x, y)$
- In the Matrix form T(X) = AX, where $X = \begin{bmatrix} x \\ y \end{bmatrix}$

  a=d=1, b=c=0 => x'=x, y'=y**A = identity matrix, and**

  a=d=1, b=c=0 => x'=x, y'=y

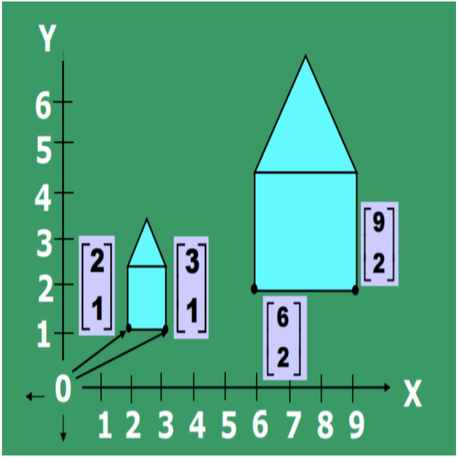
- **Scaling :**
  b=0, c=0 => x' = a.x, y' = d.y;
  This is scaling by a in x, d in y.

  If, a=d > 1, we have enlargement;
  If, 0 < a=d < 1, we have compression;

  If a = d, we have uniform scaling, else non-uniform scaling.

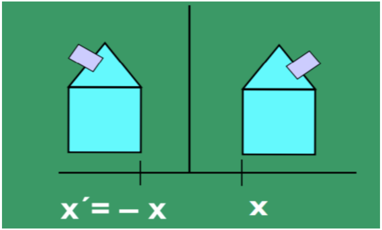
  Scale matrix: let $S_x = a$, $S_y = d$: $\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$

$S_x = 3$
$S_y = 2$



Y
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9
X
$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 3 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 6 \\ 2 \end{bmatrix}$
$\begin{bmatrix} 9 \\ 2 \end{bmatrix}$

What if S_x and/ or S_y<0 (are negative)?
Get reflections about an axis.



Only diagonal terms are involved in scaling and reflections.
Reflection
(about the Y-axis)

Special cases of Reflections ($|A| = -1$)

Matrix A	Reflection about
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	X-axis
$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	Y-axis
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	Y $=$ X line
$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$	Y $=$ -X line

Shearing: Off diagonal terms are involved in General Lin 2D-Transform

The General 2D Linear Transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

x' = ax + cy
y' = bx + dy

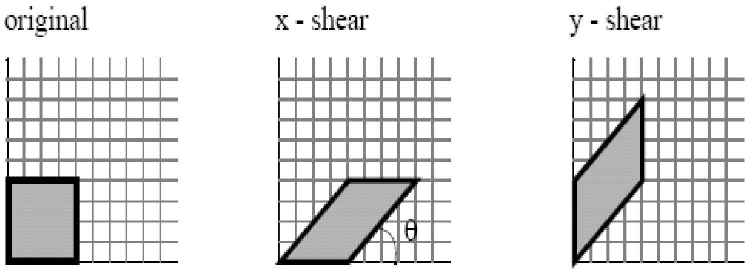
Substitute in the transform matrix

a = d = 1;
and b =0, $c \neq 0$
x' = x+cy
y' = y ;
This is called as Sheering in X Direction

Substitute in the transform matrix

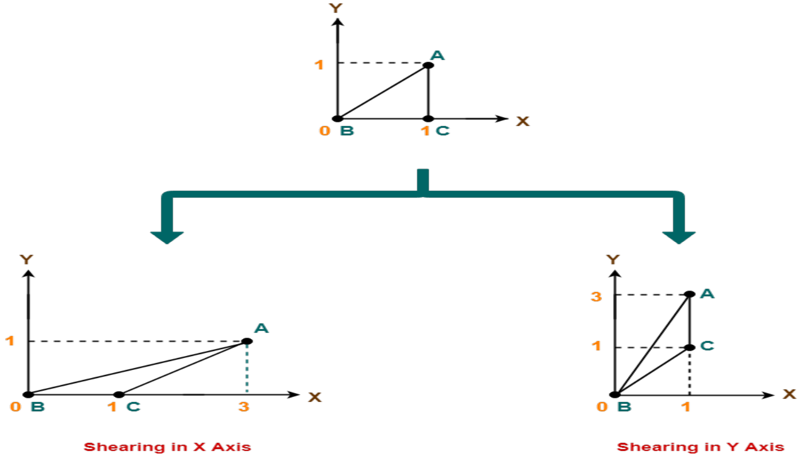
a = d = 1;
and $b \neq 0$, c=0
x' = x
y' = bx + y ;
This is called as Sheering in Y Direction

Shearing in X and Y Directions

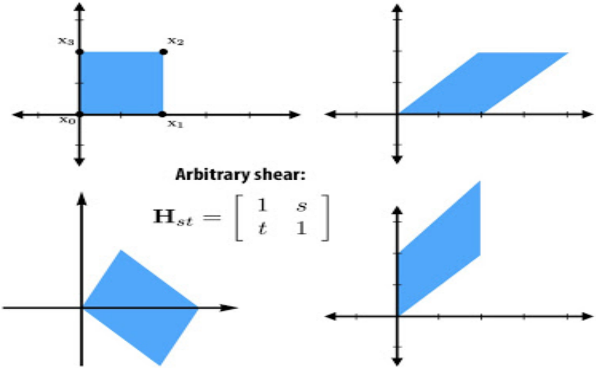


X-Shear: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

Y-Shear: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$


Y
1
A
0 B
1 C
X
Y
1
A
0 B
1 C
3
X
Shearing in X Axis
Y
3
A
1
C
0 B
1
X
Shearing in Y Axis

Shear



Shear in x:

$$\mathbf{H}_{xs} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$

Arbitrary shear:

$$\mathbf{H}_{st} = \begin{bmatrix} 1 & s \\ t & 1 \end{bmatrix}$$

Shear in y:

$$\mathbf{H}_{ys} = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix}$$

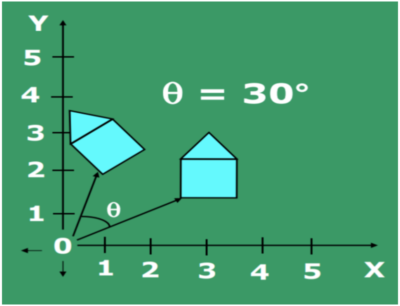
CMU 15-462/662

ROTATION

$$X' = x\cos\theta - y\sin\theta$$
$$Y' = x\sin\theta + y\cos\theta$$

In matrix form, this is :

$$\mathbf{A} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



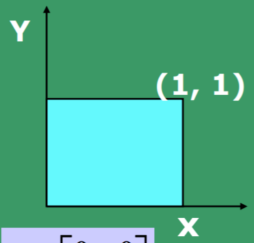
Positive Rotations: counter clockwise about the origin

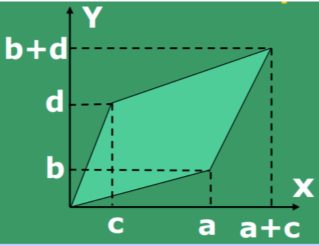
For rotations, $|A| = 1$ and $A^T = A^{-1}$. Rotation matrix is orthogonal.

Special cases of Rotations

Matrix T	θ (in degrees)
$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	90
$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	180
$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	270 or -90
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	360 or 0

Example - Transformation of a Unit Square



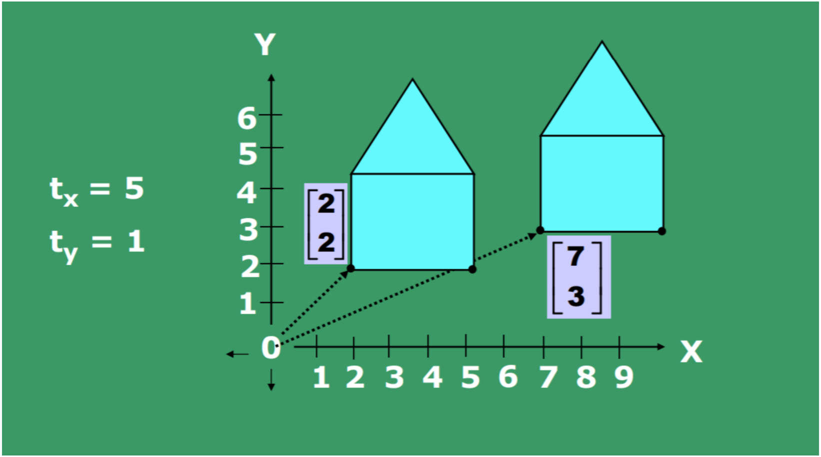


$$S = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$S' = S \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix}$$

Area of the unit square after transformation
$= ad - bc = |T|$**.**
Extend this idea for any arbitrary area.

Translations



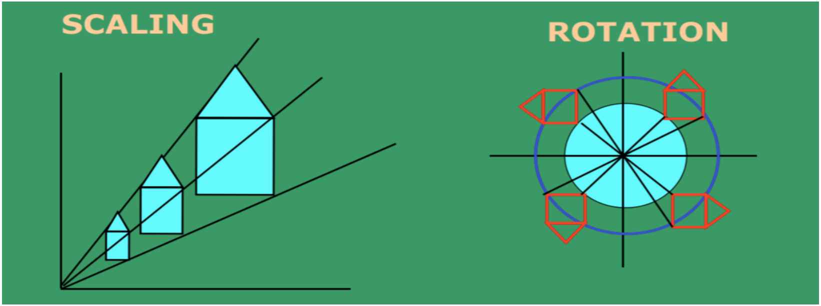
Translation of (x, y) by (t_x, t_y): $T(x, y) = (x, y) + T_d$, where $T_d = (t_x t_y)$

Where else are translations introduced?

- **Rotations** - when objects are not centered at the origin.
- **Scaling** - when objects/lines are not centered at the origin - if line intersects the origin, no translation.

Origin is invariant to Scaling, reflection and Shear – not translation.

Note: Scaling and Rotations are introducing scaling



- Translation is not linear
- Can you make it linear by adding one more dimension
- Yes, by using homogeneous coordinates which is adding one more dimension.

HOMOGENEOUS COORDINATES

Use a 3 x 3 matrix:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & c & t_x \\ b & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

We have:

$x' = ax + cy + t_x$

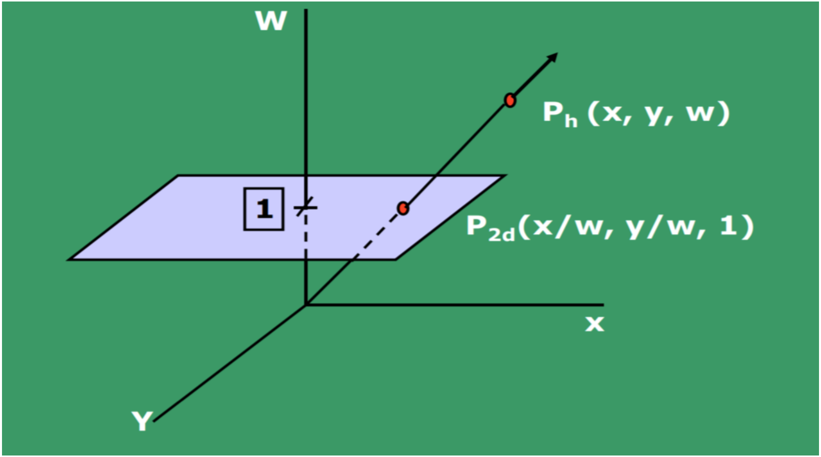
$y' = bx + cy + t_y$

- (X, Y) in Cartesian coordinate is mapped to (wX, wY, w) in the homogeneous coordinate system
- Given (x, y, w) in homogeneous coordinate system, the corresponding (X, Y) in Cartesian coordinate system is (X, Y) = (x/w, y/w)
- The transformation matrix given above is called as affine transform in 2D

Affine Transform

- Alternative definition of 2D affine transform: Y=AX+B, where X is input in 2D, Y is output in 2D, and B is constant point in 2D
- HW: Prove the equivalence of the above two defns.
- Observation: Every 2D-linear transform is 2D affine, but the converse is not true(when b =0 affine and linear are the same)
- Some Properties of Affine:
  - Affine transform preserves **collinearity** of points
  - Affine transform preserves **parallelism** of lines
  - Affine transform preserves **Convexity**
    - Line is mapped to line
    - Triangle is mapped to triangle
- Scaling, Rotation, Translation, Reflection and Shearing are Affine Transforms

Interpretation of Homogeneous Coordinates



Interpretation of Homogeneous Coordinates (cont.)

- Two homogeneous coordinates (x_1, y_1, w_1) (x_2, y_2, w_2) may represent the same point, iff they are multiples of one another: say, (1,2,3) (3,6,9).
- There is no unique homogeneous representation of a point.
- All triples of the form (t.x, t.y, t.W) form a line in x,y,W space.
- For all triplets (t.x, t.y, t.W), $\forall t$, the corresponding Cartesian coordinates is a single point (X, Y) = (tx/tw, ty/tw) = (x/w ,y/w)
- Hence, a single point (X, Y) is uniquely mapped to a line in homogeneous coordinate system
- Cartesian coordinates are just the plane w=1 in this space.
- When W=0, the corresponding points in Cartesian coordinates are the points at infinity

Composite Transformation: Composition of transformations T_1, T_2, T_3 etc. to a set of points,

We can do it in two ways:

- **Method 1:** Calculate $p' = T_1 * p$, $p" = T_2 * p'$, $p''' = T_3 * p"$
- **Method 2:** Calculate $T = T_1 * T_2 * T_3$, then $p''' = T * p$.
- Method 2, saves large number of additions and multiplications
- Therefore, We Multily the matrices into one final transformation matrix, and then apply that to the points

Translations:

Translate the points by (tx_1, ty_1), then by (tx_2, ty_2):

$$\begin{bmatrix} 1 & 0 & (\mathrm{tx}_1 + tx_2) \\ 0 & 1 & (\mathrm{ty}_1 + ty_2) \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling:

Similar to translations: Scaling by $(a_1,\ b_1)$ followed by $(a_2,\ b_2)$ is the same scaling by $(a_1 a_2,\ b_1 b_2)$

Rotations:

To rotate by θ_1, then by θ_2:

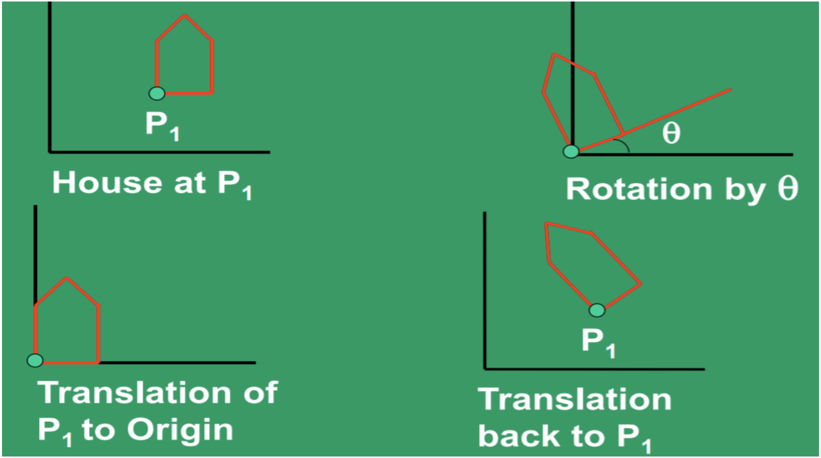
- Substitute $(\theta_1 + \theta_2)$ for θ in rotation matrix, or
- Calculate rotation matrices T_1 for θ_1, then T_2 for θ_2 multiply them.

Exercise: Both gives the same result – work it out

Rotation about an arbitrary point P in space

- The rotation matrix defined before is for rotating any point Q about origin
- To rotate a point Q about any arbitrary point P
  - Translate P to make it coincide with origin, say the translation is $(-P_x, -P_y)$
  - Translate Q by $(-P_x, -P_y)$
  - Rotate Q about origin
  - Translate Q by (P_x, P_y)

Rotation about an arbitrary point P in space (cont.)



Rotation about an arbitrary point P in space (cont.)

$$T = T_3(P_x, P_y) * T_2(\theta) * \mathrm{T}_1(-P_x, -P_y)$$

$$= \begin{bmatrix} 1 & 0 & \mathrm{P}_x \\ 0 & 1 & \mathrm{P}_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -\mathrm{P}_x \\ 0 & 1 & -\mathrm{P}_y \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling about an arbitrary point in Space

Steps:

- Translate P to origin
- Scale
- Translate P back

$$T = T_1(P_x, P_y) * T_2(S_x, S_y) * T_3(-P_x, -P_y)$$

$$\mathrm{T} = \begin{bmatrix} S_x & 0 & \mathrm{P}_x * (1 - S_x) \\ 0 & S_y & \mathrm{P}_y * (1 - S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

- Reflextion matrices are known for reflecting about X, Y axes and also diagonals
- Hence, To transform about an arbitrary line, do the following
  - Translate the arbitrary line to a line passing through origin
  - Rotate the line to align with X-axis
  - Reflect the object about X axis
  - Reverse the rotation (Apply inverse Rotation)
  - Reverse the translation
- The Transformation Matrix: $T_{GenRfl} = T_r^{-1} R^T T_{rfl} R T_r$

Commutativity of Transformations

If we scale, then translate to the origin, and then translate back, is that equivalent to translate to origin, scale, translate back?

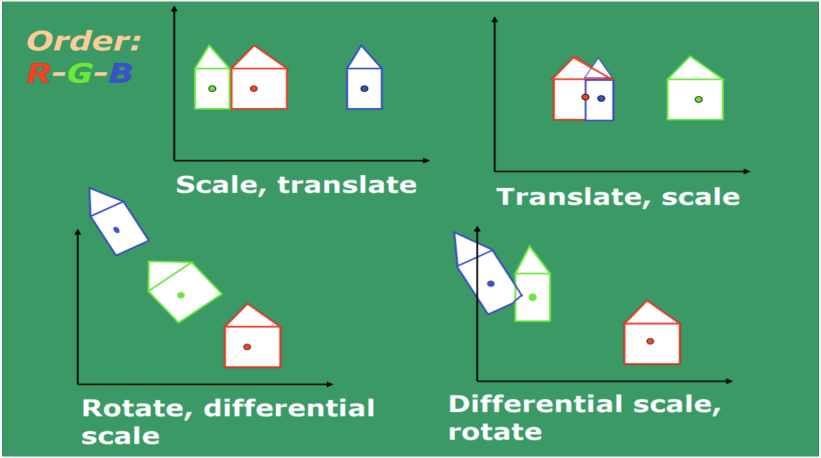
When is the order of matrix multiplication unimportant?

When does T1 * T2 = T2 * T1?

Cases where T1 * T2 = T2 * T1:

T_1	T_2
translation	translation
scale	scale
rotation	rotation
scale(uniform)	rotation

Commutativity of Transformations (cont.)



World Vs Screen Coordinate Systems

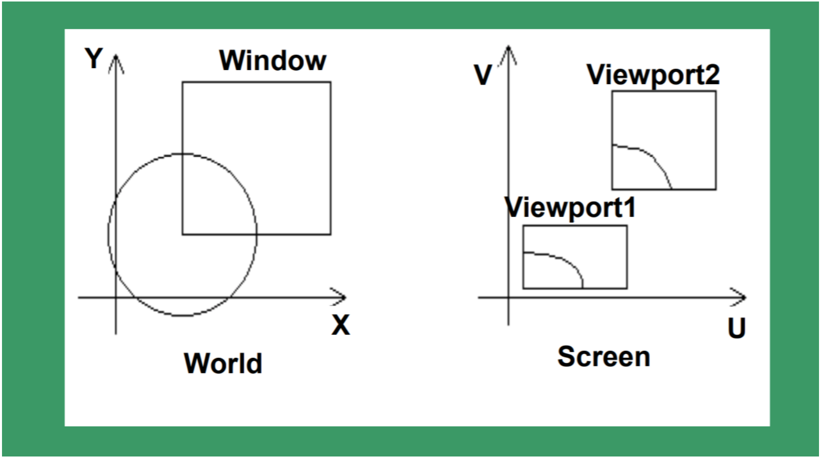
Screen Coordinates: The coordinate system used to address the screen (device coordinates)

World Coordinates: A user-defined application specific coordinate system having its own units of measure, axis, origin, etc.

Window: The rectangular region of the world that is visible.

Viewport: The rectangular region of the screen space that is used to display the window.

World Vs Screen Coordinate Systems (cont.)



WINDOW TO VIEWPORT TRANSFORMATION

Purpose is to find the transformation matrix that maps the window in world coordinates to the viewport in screen coordinates.

Window: (x, y space) denoted by:

$x_{min}, y_{min}, x_{max}, y_{max}$

Viewport: (u, v space) denoted by:

$u_{min}, v_{min}, u_{max}, v_{max}$

WINDOW TO VIEWPORT TRANSFORMATION (cont.)

The overall transformation:

- Translate the window to the origin
- Scale it to the size of the viewport
- Translate it to the viewport location

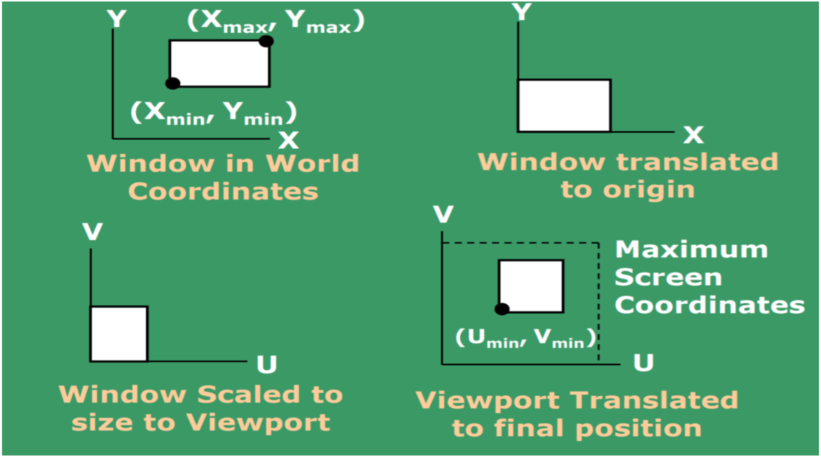
$$M_{WV} = T(U_{min}, V_{min}) * S(S_x, S_y) * T(-x_{min}, -y_{min})$$

$$S_x = (U_{max} - U_{min})/(x_{max} - x_{min})$$

$$S_y = (V_{max} - V_{min})/(y_{max} - y_{min})$$

$$M_{WV} = \begin{bmatrix} S_x & 0 & (\text{-}x_{min} * S_x + U_{min}) \\ 0 & S_y & (\text{-}y_{min} * S_y + V_{min}) \\ 0 & 0 & 1 \end{bmatrix}$$

WINDOW TO VIEWPORT TRANSFORMATION (cont.)



Consider two parallel lines:

- A[X_1, Y_1] to B[X_2, Y_2] and
- C[X_3, Y_3] to B[X_4, Y_4].

Slope of the lines: m $= \frac{Y_2 - Y_1}{X_2 - X_1} = \frac{Y_4 - Y_3}{X_4 - X_3}$

Solve the problem:

If the lines are transformed by a matrix: $T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

The slope of the transformed lines is: m' = (b+dm)/(a+cm)

Acknowledgements

- Some of the slides have been adopted from NPTEL and different internet sources. The due credits are acknowledged.

Thank You! :)