TRANSFORMATION OF POINTS & LINES

Presented by, VINEETH.V
Asst. Professor CCET

Downloaded from Ktunotes.in
<table>
<thead>
<tr>
<th>Transformation of points and line, 2-D rotation, reflection, scaling and combined transformation, homogeneous coordinates, 3-D scaling.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shearing, rotation, reflection and translation, combined transformations, orthographic and perspective projections, reconstruction of 3-D objects.</td>
</tr>
</tbody>
</table>
CO-ORDINATE TRANSFORMATION
CO-ORDINATE TRANSFORMATION

• It means changing of an image from current position (state) to a new position (state) by applying certain rules.

  Current position (state)  New position (state)

• Geometric transformation are the transformations or changes in size, shape, location etc are accomplished by altering the coordinate descriptions of an object.
CO-ORDINATE TRANSFORMATION

Types of transformations are:
1. 2D Transformations
2. 3D transformations

Basic Geometric transformations are:
3. Translation/ Move
4. Scaling
5. Rotation
6. Mirroring/ Reflection/ Flip
7. Shearing
2D TRANSFORMATION
When transformation of coordinates takes place on 2D plane or XY plane, it is called as 2D transformation.

\[(x_1, y_1) \rightarrow (x_2, y_2)\]
Basic 2D Geometric transformations are:

1. 2D Translation/ Move
2. 2D Scaling
3. 2D Rotation
4. 2D Mirroring/ Reflection/ Flip
5. 2D Shearing
1. **2D Translation/Move/Shift**

- It is the **repositioning or shifting an object along a straight-line path** (translation distances- $t_x, t_y$) from one coordinate location to another without deformation.
- Also called as **shift/move/translation**.

\[(x, y) \rightarrow (x', y')\]
2D Translation/ Move

• We translate a 2D point by adding a translation distance $t_x$ and $t_y$, to the original coordinate position $(x,y)$ to move the point to a new position $(x',y')$

New position of $x$, 

$x' = x + t_x$

New position of $y$, 

$y' = y + t_y$

Where, $t_x$ and $t_y$ are translation vector or shift vector
2D Translation/ Move
2D Translations in Homogenised coordinates

• Transformation matrices for 2D translation in 3x3 column matrix:

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

\[
x' = x + t_x \\
y' = y + t_y \\
1 = 1
\]
2. **2D Scaling**

- It alters the size of an object (either reduced or enlarged size)
2D Scaling

• It is transformed **by multiplying** the current coordinate values \((x,y)\) of each vertex by **Scaling factors** \(S_x & S_y\) to produce the new transformed coordinates \((x',y')\)

New position of \(x\),
\[
x' = x \cdot S_x
\]
New position of \(y\),
\[
y' = y \cdot S_y
\]

**Where,** \(S_x & S_y\) **are scaling factors**
2D Scaling - Scale factor [S]

Scale factor [S] value has only positive values:

• Value less than 1 (<1) → Reduce the size of object
• Value greater than 1 (>1) → Enlarge the size of object
• Same value (=1) → Uniform scaling
• Unequal value (≠1) → Differential scaling
2D Scaling

• In matrix form,

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

\[
x' = x.s_x \\
y' = y.s_y
\]
3. **2D Rotation**

- It is the repositioning of an object along a circular path in the xy plane.
2D Rotation

• To generate a rotation, we specify a rotation angle $\theta$ and the position of the rotation point (pivot point) about which the object is to be rotated.

• **Positive value** for $\theta$ → **counter clockwise** rotation

• **Negative value** for $\theta$ → **clockwise** rotation
2D Rotation

From the figure,
---we have to find the new position \((x', y')\)
2D Rotation

Considering a triangle OA:

\[ \cos(\theta + \phi) = \frac{x'}{OP'} \]

\[ x' = OP' \cdot \cos(\theta + \phi) \]

Similarly

\[ \sin(\theta + \phi) = \frac{y'}{OP'} \]

\[ y' = OP' \cdot \sin(\theta + \phi) \]

\[ = OP' \cdot (\sin\theta \cdot \cos\phi + \cos\theta \cdot \sin\phi) \]
2D Rotation

Considering a triangle $OA$

$\cos \Phi = \frac{x}{OP}$

$x = OP \cdot \cos \Phi$

Similarly

$\sin \Phi = \frac{y}{OP}$

$y = OP \cdot \sin \Phi$

we know $OP = OP'$

So,

$x' = x \cos \theta - y \sin \theta$

$y' = x \sin \theta + y \cos \theta$
2D Rotation

- In matrix form,

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix}
= \begin{bmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

\[ [P'] = [R(\theta)] \cdot [P] \]

ie...

\[ x' = x \cos \theta - y \sin \theta \]
\[ y' = x \sin \theta + y \cos \theta \]

Where, \( R(\theta) = \text{Rotation transformation operator} \)
### 4. 2D REFLECTION/FLIP/MIRRORING

#### x-axis (y = 0)

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

Original

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

Reflected

#### y-axis (x = 0)

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

#### xy-plane

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

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

Downloaded from Ktunotes.in
Reflection

- A transformation produces a mirror image of an object.

- Axis of reflection
  - A line in the xy plane
  - A line perpendicular to the xy plane
  - The mirror image is obtained by rotating the object 180° about the reflection axis.

- Rotation path
  - Axis in xy plane: in a plane perpendicular to the xy plane.
  - Axis perpendicular to xy plane: in the xy plane.
5. 2D SHEARING

- Shearing transformation is the transformation which alters the shape of an object.
- Deformation of shape of object takes place in x and y direction.
X shear

- Preserve Y coordinates but change the X coordinates values

\[ x' = x + s h_x \cdot y \]
\[ y' = y \]

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

\[(0,0)\] \( (0,1) \) \( (1,0) \) \( (1,1) \)

\[(0,0)\] \( (1,0) \) \( (2,1) \) \( (3,1) \)

\[ s h_x = 2 \]
Y shear

- Preserve X coordinates but change the Y coordinates values

\[ x' = x \]
\[ y' = y + S_{hy} \cdot x \]

\[
\begin{bmatrix}
1 & 0 & 0 \\
sh_y & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
\[
SH_x = \begin{bmatrix}
1 & sh_x & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \quad \quad SH_y = \begin{bmatrix}
1 & 0 & 0 \\
sh_y & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Various 2D TRANSFORMATIONs

**Translation**

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

**Scaling**

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

**Rotation**

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} =
\begin{bmatrix}
    \cos \theta & -\sin \theta & 0 \\
    \sin \theta & \cos \theta & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

**Shearing**

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

Downloaded from Ktunotes.in
2D Translation

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & t_x \\
  0 & 1 & t_y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}, \quad P' = T(t_x, t_y)
\]

2D Rotation

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}, \quad P' = R(\theta)
\]

2D Scaling

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  S_x & 0 & 0 \\
  0 & S_y & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}, \quad P' = S(S_x, S_y)
\]
\[ R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \]

\[ R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \]

\[ R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \]
3D TRANSFORMATION
3D TRANSFORMATION

• When transformation of coordinates takes place on 3D plane or XYZ plane, it is called as 3D transformation.

\[(x_1, y_1, z_1)\] \rightarrow \[(x_2, y_2, z_2)\]

line
3D TRANSFORMATION

Basic 3D Geometric transformations are:
1. 3D Translation/ Move
2. 3D Scaling
3. 3D Rotation
4. 3D Mirroring/ Reflection/ Flip
5. 3D Shearing
1. 3D Translation

- Moving of object in x,y,z direction as translation vector tx,ty,tz respectively

\[
\begin{align*}
x' &= x + t_x \\
y' &= y + t_y \\
z' &= z + t_z
\end{align*}
\]

\[
\begin{pmatrix}
x' \\
y' \\
z'
\end{pmatrix} = 
\begin{bmatrix}
1 & 0 & 0 & t_x \\
0 & 1 & 0 & t_y \\
0 & 0 & 1 & t_z
\end{bmatrix}
\begin{pmatrix}
x \\
y \\
z \\
1
\end{pmatrix}
\]
3D TRANSLATION

- The matrix representation is equivalent to the three equations:

\[ x' = x + t_x, \quad y' = y + t_y, \quad z' = z + t_z \]

Where parameter \( t_x, t_y, t_z \) are specifying translation distance for the coordinate direction \( x, y, z \) are assigned any real value.
2. 3D Scaling

Enlarging object also moves it from origin

\[
P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{S} \begin{bmatrix} x \\ y \\ z \end{bmatrix}
\]

\[
x' = x \times S_x \\
y' = y \times S_y \\
z' = z \times S_z
\]
Scaling

Scaling size of object in x, y, z direction as scaling vector \( s_x, s_y, s_z \) respectively.

\[
x' = s_x x \\
y' = s_y y \\
z' = s_z z
\]

\[
p' = Sp
\]

\[
S = S(s_x, s_y, s_z) = \begin{bmatrix}
  s_x & 0 & 0 & 0 \\
  0 & s_y & 0 & 0 \\
  0 & 0 & s_z & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]
3. 3D Rotation

- ROTATION at x, y, z direction at rotating angle about a fixed pivot point.
- Need to specify which axis the rotation is about.

Rotation about z-axis

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
Rotating About the $x$-axis $R_x(\theta)$

$$
\begin{bmatrix}
    x' \\
    y' \\
    z'
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 & 0 & 0 \\
    0 & \cos \theta & -\sin \theta & 0 & 0 \\
    0 & \sin \theta & \cos \theta & 0 & 0 \\
    0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
$$
X-AXIS ROTATION

The equation for X-axis rotation

\[ x' = x \]
\[ y' = y \cos\theta - z \sin\theta \]
\[ z' = y \sin\theta + z \cos\theta \]

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & \cos\theta & -\sin\theta & 0 & 0 \\
  0 & \sin\theta & \cos\theta & 0 & 0 \\
  0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
Rotating About the y-axis

\[ R_y(\theta) \]

\[
\begin{pmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{pmatrix} =
\begin{pmatrix}
  \cos \theta & 0 & \sin \theta & 0 \\
  0 & 1 & 0 & 0 \\
  -\sin \theta & 0 & \cos \theta & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  x \\
  y \\
  z \\
  1
\end{pmatrix}
\]
Y-AXIS ROTATION

The equation for Y-axis rotation

\[ x' = x \cos \theta + z \sin \theta \]
\[ y' = y \]
\[ z' = z \cos \theta - x \sin \theta \]

\[
\begin{bmatrix}
    x' \\
    y' \\
    z' \\
    1
\end{bmatrix}
= 
\begin{bmatrix}
    \cos \theta & 0 & \sin \theta & 0 \\
    0 & 1 & 0 & 0 \\
    -\sin \theta & 0 & \cos \theta & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix} \cdot 
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]
Rotation About the $z$-axis

$R_z(\theta)$

$$
\begin{pmatrix}
  x' \\
  y' \\
  z \\
  1
\end{pmatrix} =
\begin{pmatrix}
  \cos \theta & -\sin \theta & 0 & 0 \\
  \sin \theta & \cos \theta & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  x \\
  y \\
  z \\
  1
\end{pmatrix}
$$
Rotation in 3D

• For rotation about the x and y axes:

\[
R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \theta & -\sin \theta & 0 \\
0 & \sin \theta & \cos \theta & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & \sin \theta & 0 \\
0 & 1 & 0 & 0 \\
-\sin \theta & 0 & \cos \theta & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
Rotating About An Arbitrary Point

• What happens when you apply a rotation transformation to an object that is not at the origin?

• Solution:
  – Translate the center of rotation to the origin
  – Rotate the object
  – Translate back to the original location
Rotating About An Arbitrary Point
Rotation about $x$ and $y$ axes

- Same argument as for rotation about $z$ axis
  - For rotation about $x$ axis, $x$ is unchanged
  - For rotation about $y$ axis, $y$ is unchanged

\[
R = R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \theta & -\sin \theta & 0 \\
0 & \sin \theta & \cos \theta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
R = R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & \sin \theta & 0 \\
0 & 1 & 0 & 0 \\
-\sin \theta & 0 & \cos \theta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
4. **3D Reflection**

- Mirroring of object along x, y or z axis

\[
s_x = -1 \quad s_y = 1
\]

\[
s_x = -1 \quad s_y = -1
\]

\[
s_x = 1 \quad s_y = -1
\]
3D Reflection

- Reflection or mirror matrix: a scaling matrix where one scaling factor is \(-1\) and two others are \(1\) or all of the three scaling factors are \(-1\).
- If two are \(-1\), it’s a \(180^\circ\) rotation.

\[
\begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad \text{reflection wrt yz-plane}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad \text{reflection wrt xz-plane}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad \text{reflection wrt xy-plane}
\]

\[
\begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \quad \text{reflection wrt the origin}
\]
Reflection

- Reflection over planes, lines, or points

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

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

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

\[
\begin{pmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
3D REFLECTION

- Reflection about x-axis:
  \[ x' = x \quad y' = -y \quad z' = -z \]
  
  \[
  \begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & -1 & 0 & 0 \\
  0 & 0 & -1 & 0 \\
  0 & 0 & 0 & 1
  \end{pmatrix}
  
- Reflection about y-axis:
  \[ y' = y \quad x' = -x \quad z' = -z \]
5. **3D Shearing**

- Shearing transformation is the transformation which alters the shape of an object.
- Deformation of shape of object takes place in x, y, and z direction.
3D SHEARING

E.g. draw a cube (3D) on a screen (2D). Alter the values for x and y by an amount proportional to the distance from $z_{ref}$. 
Shear

- Let pull in top right edge and bottom left edge
  - Neither $y$ nor $z$ are changed
  - Call $x$ shear direction
3D SHEARING

- Matrix for 3d shearing
- Where a and b can be assigned any real value.

\[
\begin{bmatrix}
x' \\
y' \\
z' \\
1
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & a & 0 \\
0 & 1 & b & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
Shear along Z-axis

\[ \mathbf{SH}_{xy}(sh_x, sh_y) \ * \ P = \ P' \]

\[
\begin{bmatrix}
1 & 0 & sh_x & 0 \\
0 & 1 & sh_y & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \ *
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix} =
\begin{bmatrix}
x + z \cdot sh_x \\
y + z \cdot sh_y \\
z \\
1
\end{bmatrix}
\]
Other Transformations: SHEARING

X-axis 3-D Shear transformation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Y-axis 3-D Shear transformation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
Combined transformation

• It is the transformation of different types of geometric transformation like:
  1. Translate or move object by 2 units in x, y and z direction
  2. Scaling it into 2 units in x, y and z direction
  3. Rotating the points at rotating angle.
Combined transformation

• 2 methods are:

1. Direct method
   • Step by step method of combined transformation
   1. $[\bar{x}] = [T]. [x]$
   2. $[\bar{x}] = [S]. [x]$
   3. $[\bar{x}] = [R(\theta)]. [x]$

2. Concatenation method
2. Concatenation transformation

- We can form arbitrary affine transformation matrices by multiplying together translation, scaling & rotation matrices

\[
[\tilde{X}] = [T_c]. [X]
\]

Where, \( T_c \) = concatenation transformation = \([T]. [S]. [R(\theta)].\)
Matrix Composition

- Transformations can be combined by matrix multiplication

\[
\begin{bmatrix}
x' \\
y' \\
w'
\end{bmatrix} = \begin{bmatrix}
1 & 0 & tx \\
0 & 1 & ty \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\cos\theta & -\sin\theta & 0 \\
\sin\theta & \cos\theta & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
sx & 0 & 0 \\
0 & sy & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
w
\end{bmatrix}
\]

\[p' = T(tx, ty) \quad R(\theta) \quad S(sx, sy) \quad p\]

- Matrix multiplication is associative

\[p' = (T \times (R \times (S \times p))) \quad \rightarrow \quad p' = (T \times R \times S) \times p\]
Matrix concatenation properties

- Multiplication is associative

\[ M_3 \cdot M_2 \cdot M_1 = (M_3 \cdot M_2) \cdot M_1 = M_3 \cdot (M_2 \cdot M_1) \]

- Multiplication is NOT commutative
  - Unless the sequence of transformations are all of the same kind
  - \( M_2M_1 \) is not equal to \( M_1M_2 \) in general
Homogeneous transformation
(w or h)

2D
• It is the conversion of 2x2 matrices of points $P(x,y)$ to 3x3 matrices of point $P(x,y,1)$ in 2D.

3D
• It is the conversion of 3x3 matrices of points $P(x,y,z)$ to 4x4 matrices of point $P(x,y,z,1)$.
2D Translations in Homogenised coordinates

- Transformation matrices for 2D translation in $3 \times 3$ column matrix:

$$
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
$$

$$
x' = x + t_x \\
y' = y + t_y \\
1 = 1
$$

$$
[P'] = [T] + [P]
$$
Matrix representations and homogeneous coordinates

- Multiplicative and translational terms for a 2D transformation can be combined into a single matrix.
- This expands representations to 3x3 matrices.
  - Third column is used for translation terms.
- Result: All transformation equations can be expressed as matrix multiplications.

Homogeneous coordinates: \((x_h, y_h, h)\)

- Carry out operations on points and vectors “homogeneously.”
- \(h\): Non-zero homogeneous parameter such that

\[
\begin{align*}
x &= \frac{x_h}{h}, \\
y &= \frac{y_h}{h}
\end{align*}
\]

- We can also write: \((hx, hy, h)\)
- \(h=1\) is a convenient choice so that we have \((x, y, 1)\)
- Other values of \(h\) are useful in 3D viewing transformations.
Homogeneous Coordinates

The homogeneous coordinates form for a three dimensional point \([x \ y \ z]\) is given as
\[
p = [x' \ y' \ z' \ w]^T = [wx \ wy \ wz \ w]^T
\]
We return to a three dimensional point (for \(w \neq 0\)) by
\[
x \leftarrow x'/w
\]
\[
y \leftarrow y'/w
\]
\[
z \leftarrow z'/w
\]
If \(w = 0\), the representation is that of a vector
Note that homogeneous coordinates replaces points in three dimensions by lines through the origin in four dimensions
For \(w = 1\), the representation of a point is \([x \ y \ z \ 1]\)
Advantages of Homogeneous Coords

- Unified framework for translation, viewing, rot...
- Can concatenate any set of transforms to 4x4 matrix
- No division (as for perspective viewing) till end
- Simpler formulas, no special cases
- Standard in graphics software, hardware
PROJECTIONS
PROJECTIONS

• Once the world coordinate description of the object in a scene are converted to viewing coordinates, we can project 3D object onto 2D view plane.

3D object > 2D view plane
TYPES OF PROJECTIONS

Planar geometric projections

Parallel projections

Orthographic projections

Axonometric projections

Isometric projections

Side projections

Top projections

Front projections

Oblique projections

Perspective projections

One point projections

Two point projections

Three point projections

Cabinet projections

Cavalier projections

Other projections

Other projections
TYPES OF PROJECTIONS

PARALLEL
(parallel projectors)
- Orthographic
  (projectors perpendicular to view plane)
  - Multiview
    (view plane parallel to principal planes)
  - Axonometric
    (view plane not parallel to principal planes)
    - Isometric
    - Dimetric
    - Trimetric

PERTSPECTIVE
(converging projectors)
- Oblique
  (projectors not perpendicular to view plane)
  - General
    - Cavalier
    - Cabinet
  - One point
    (one principal vanishing point)
  - Two point
    (Two principal vanishing point)
  - Three point
    (Three principal vanishing point)
Types Of Projections

There are two broad classes of projection:

– Parallel: Typically used for architectural and engineering drawings

– Perspective: Realistic-looking and used in computer graphics
PARALLEL PROJECTION VS PERSPECTIVE PROJECTION

PARALLEL PROJECTION

PERSPECTIVE PROJECTION

DOP

Object

Projector

Projection plane

COP

Object

Projector

Projection plane

Downloaded from Ktunotes.in
PARALLEL PROJECTION

- In this, coordinate positions are transformed to the view plane along parallel lines.
- Projection lines are parallel to each other.
- Projection lines are extended from the object and intersect the view plane.
Parallel Projection

- Projectors
- Center of projection at infinity
- Projection plane
PARALLEL PROJECTION

ADVANTAGE

• Accurate views of various sides of object are obtained.

DISADVANTAGE

• Does not give a realistic representation of appearance of 3D object.
TYPES OF PARALLEL PROJECTION

- Orthographic
  - Parallel Projection
    - Top
    - Front
    - Side
  - Oblique
    - Axonometric
    - Cavalier
    - Cabinet
Types of Parallel Projection

There are two types of parallel projection:

1. ORTHOGRAPHIC:
   ✓ Projection lines are parallel to each other & also perpendicular to the plane.
   ✓ It used to create different views of given object.
   ✓ There are 3 views:
     i. Front view
     ii. Side view
     iii. Top view
Orthogonal projections:
1. MULTIPLE VIEWS

3D Representation

2D Orthographic Projection
Axonometric orthographic projections

- Orthographic projections that *show more than one face of an object* are called *axonometric* orthographic projections.

- The most common axonometric projection is an *isometric* projection where the projection plane intersects each coordinate axis in the model coordinate system at an equal distance.
ORTHOGRAPHIC PROJECTION- Advantage & Application

• Engineering and architectural drawing commonly employ these orthographic projections, because length and angles are accurately depicted and can measure from drawing itself.
OBLIQUE PROJECTION

- It is obtained by projecting points along the parallel lines that are not perpendicular to projection plane.

TYPES:
1. CAVALIER PROJECTION
2. CABINET PROJECTION
CAVALIER PROJECTION

• It makes 45° with projection plane, $\alpha = 45^\circ$, the view obtained are called cavalier projections.

• All the lines perpendicular to projection plane are projected with no change in length.

➢ Length $L_1$ depends on angle and $z$ coordinate of point.
• 2 common oblique parallel projections: *Cavalier* and *Cabinet*

**Cavalier projection:**
All lines perpendicular to the projection plane are projected with no change in length.
CABINET PROJECTION

- It makes $63.4^\circ$ angle with projection plane, $\alpha = 63.4^\circ$, the view obtained are called **cavalier projections**.
- The lines perpendicular to projection plane are projected with $\frac{1}{2}$ of the length.

**ADVANTAGE**

- It appears more realistic than cavalier, because of reduction in length of perpendiculars.
**Cabinet projection:**

- Lines which are perpendicular to the projection plane (viewing surface) are projected at $\frac{1}{2}$ the length.
- This results in foreshortening of the z axis, and provides a more “realistic” view.
Perspective Projections

- Perspective projections are described by
  - **Centre of projection**: Eye of artists or lens of camera
  - **View Plane**: Plane containing canvas or film strip or frame buffer

- A ray called **projector** is drawn from COP to object point, its intersection with view plane determines the projected image point on view plane.
Perspective Projection

- Center of projection
- Projection plane
- Projectors
- View Plane
- Center of Projection
Types of Perspective

- Perpective Projection
  - One Point
  - Two Point
  - Three Point
Classes of Perspective Projection

- One-Point Perspective
- Two-Point Perspective
- Three-Point Perspective
1. **ONE POINT PERSPECTIVE PROJECTION**
Two-point perspective projection:

- This is often used in architectural, engineering and industrial design drawings.
Three-point perspective projection

- Three-point perspective projection is used less frequently as it adds little extra realism to that offered by two-point perspective projection.
Perspective Projection

ADVANTAGE:
• Looks realistic
  - because size varies inversely with distance

DISADVANTAGE:
• We **cannot** judge distance as parallel projection
Perspective v Parallel

• Perspective:
  – visual effect is similar to human visual system...
  – has 'perspective foreshortening'
    • size of object varies inversely with distance from the center of projection. Projection of a distant object are smaller than the projection of objects of the same size that are closer to the projection plane.

• Parallel:
  It preserves relative proportion of object.
  – less realistic view because of no foreshortening
  – however, parallel lines remain parallel.
Perspective Transformation

• Using 3D homogeneous coordinate representation, perspective projection transformation is shown.
Reconstruction of 3D object
Reconstruction of 3D object

• It is the process of capturing shape and appearance of real object.
• If the model is allowed to change its shape in time, this is referred to as non-rigid or spatio-temporal reconstruction.
Reconstruction of 3D object

• This process can be accomplished either by:
  – Active method
  – Passive methods

1. Active method

• Reconstruct the 3D profile by numerical approximation approach and build the object in scenario based on model using Range finders.
Reconstruction of 3D object

2. Passive method

- Passive methods of 3D reconstruction do not interfere with the reconstructed object;
- they only use a sensor to measure the radiance reflected or emitted by the object's surface to infer its 3D structure through image understanding.
Application of Reconstruction of 3D object

3D reconstruction system finds its application in a variety of field they are:

• Medicine
• Film industry
• Robotics
• City planning
• Gaming
• Virtual environment
• Earth observation
• Archaeology
• Augmented reality
• Reverse engineering
• Animation
• Human computer interaction