Computer Graphics

Lecture 2

Transformations

Transformations.

What is a transformation?

$P' = T(P)$

What does it do?

Transform the coordinates / normal vectors of objects

Why use them?

Modelling

-Moving the objects to the desired location in the environment

-Multiple instances of a prototype shape

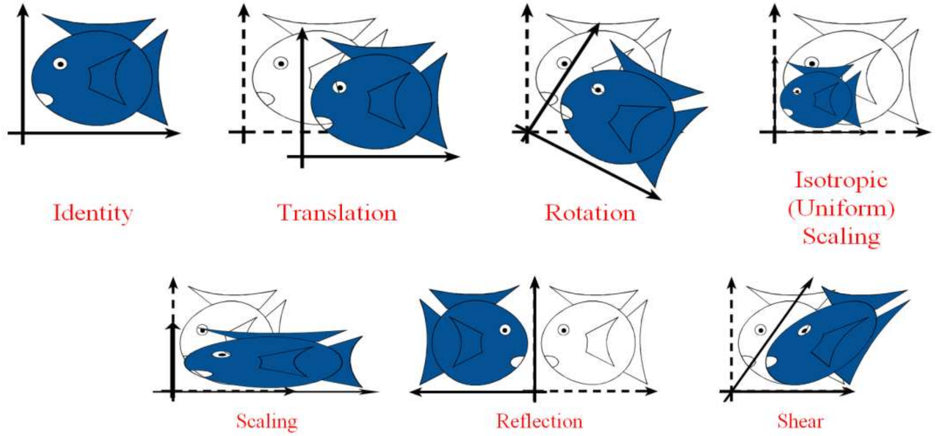
-Kinematics of linkages/skeletons – character animation

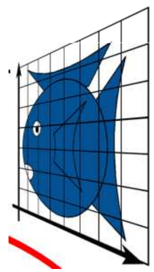
Viewing

Virtual camera: parallel and perspective projections

Types of Transformations

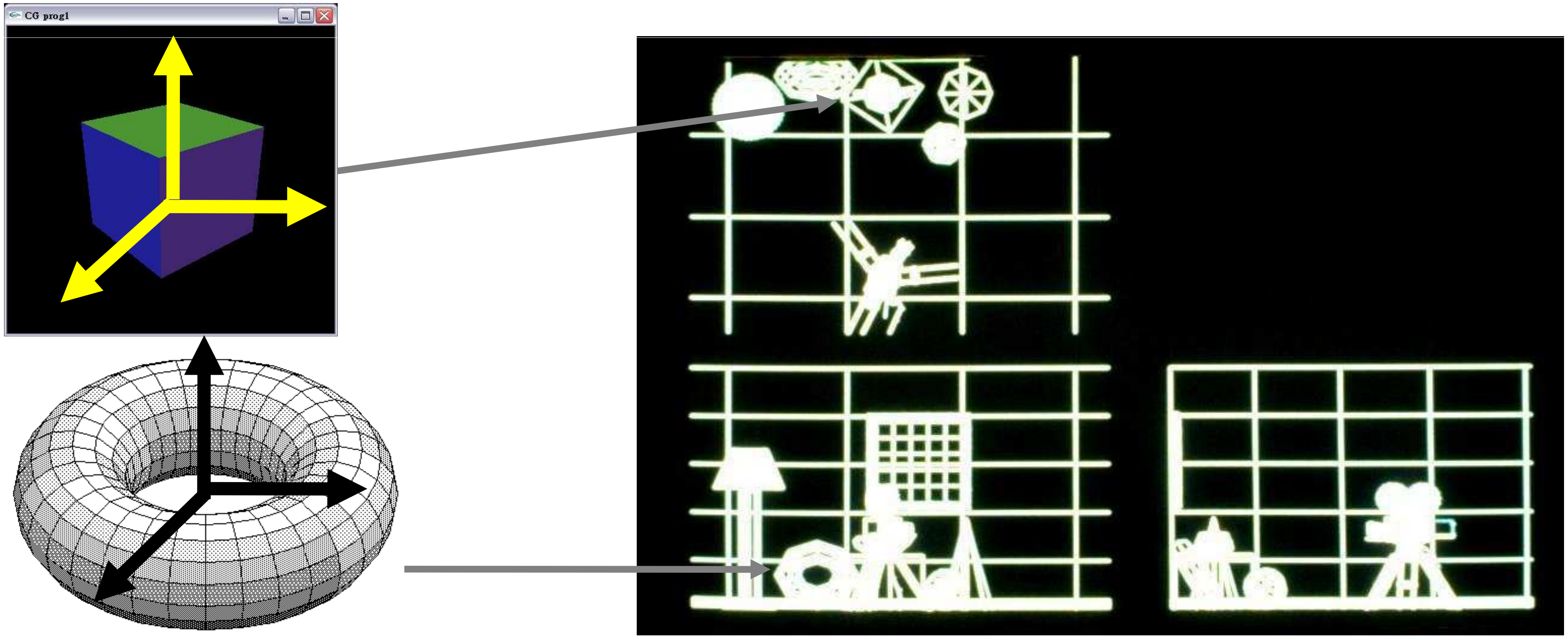
– **Geometric Transformations**
    - Translation
    - Rotation
    - scaling
- **Linear** (preserves parallel lines)
    - Non-uniform scales, shears or skews
– **Projection** (preserves lines)
    - Perspective projection
    - Parallel projection
– **Non-linear** (lines become curves)
    - Twists, bends, warps, morphs,





Geometric Transformation

– Once the models are prepared, we need to place them in the environment
– Objects are defined in their own local coordinate system
– We need to translate, rotate and scale them to put them into the world coordinate system



2D Translations.

Point P defined as $P(x, y)$,

translate to Point $P'(x', y')$ a distance d_x parallel to x axis, d_y parallel to y axis.

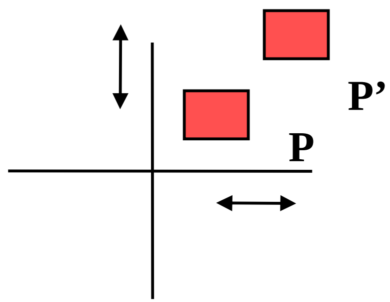
$$x' = x + d_x \quad y' = y + d_y$$

Define the column vectors

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Now

$$P' = P + T$$

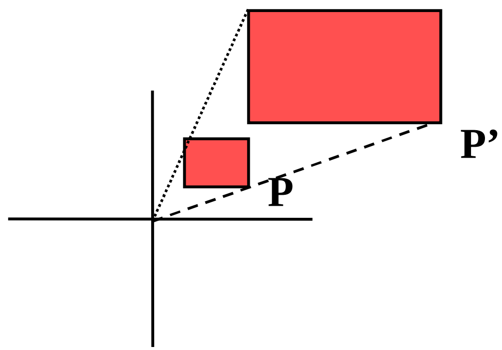


2D Scaling from the origin.

Point P defined as $P(x, y)$,

Perform a scale (stretch) to Point $P'(x', y')$ by a factor s_x along the x axis, and s_y along the y axis.

$$x' = s_x . x, \quad y' = s_y . y$$



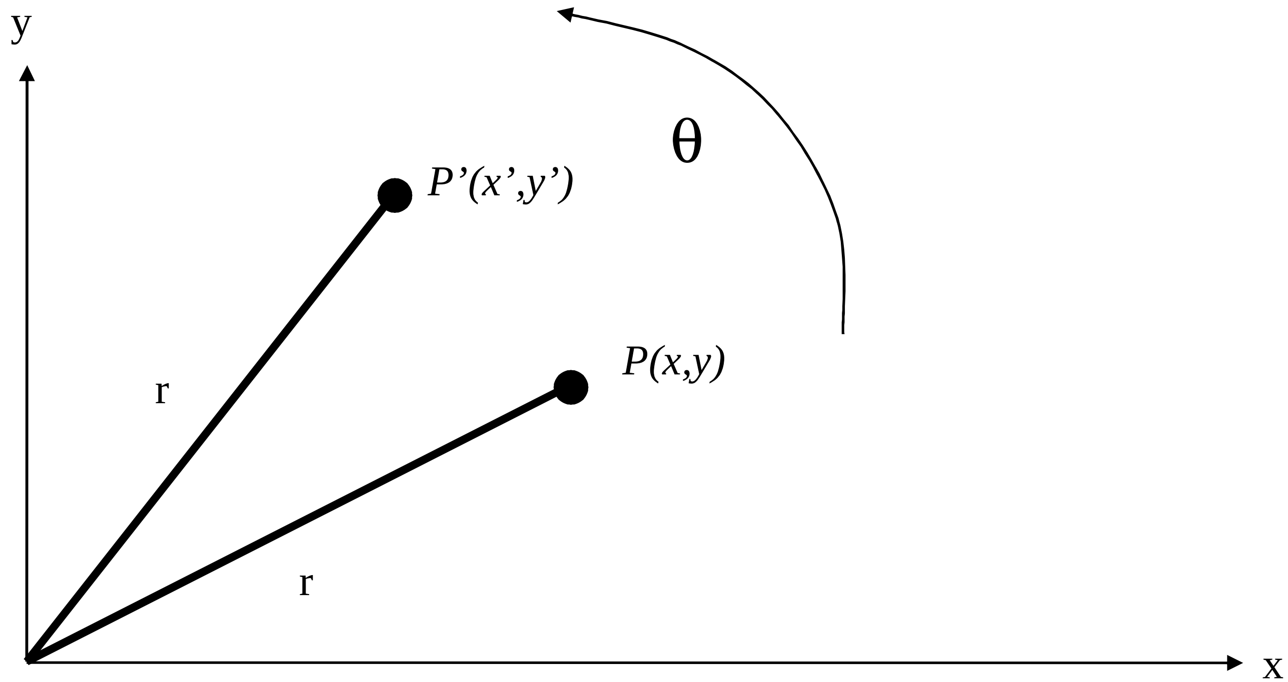
Define the matrix

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

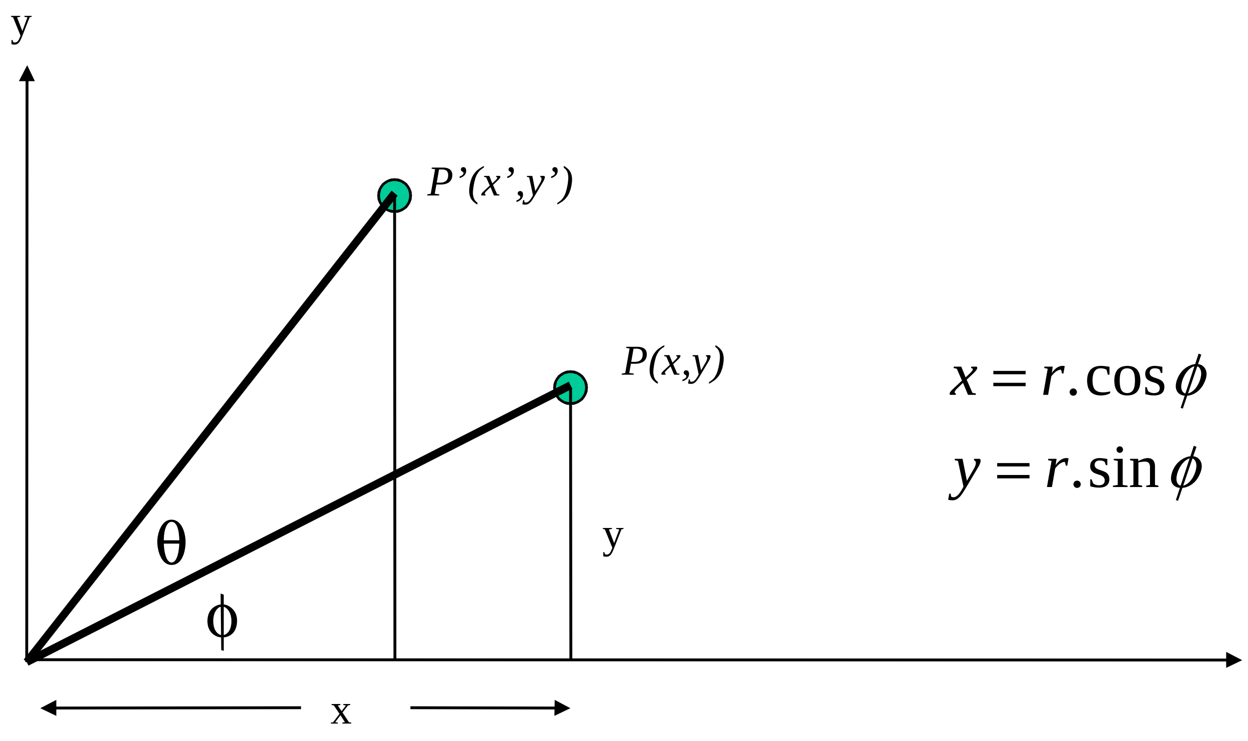
Now

$$P' = S \cdot P \quad \text{or} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotation about the origin.



2D Rotation about the origin.



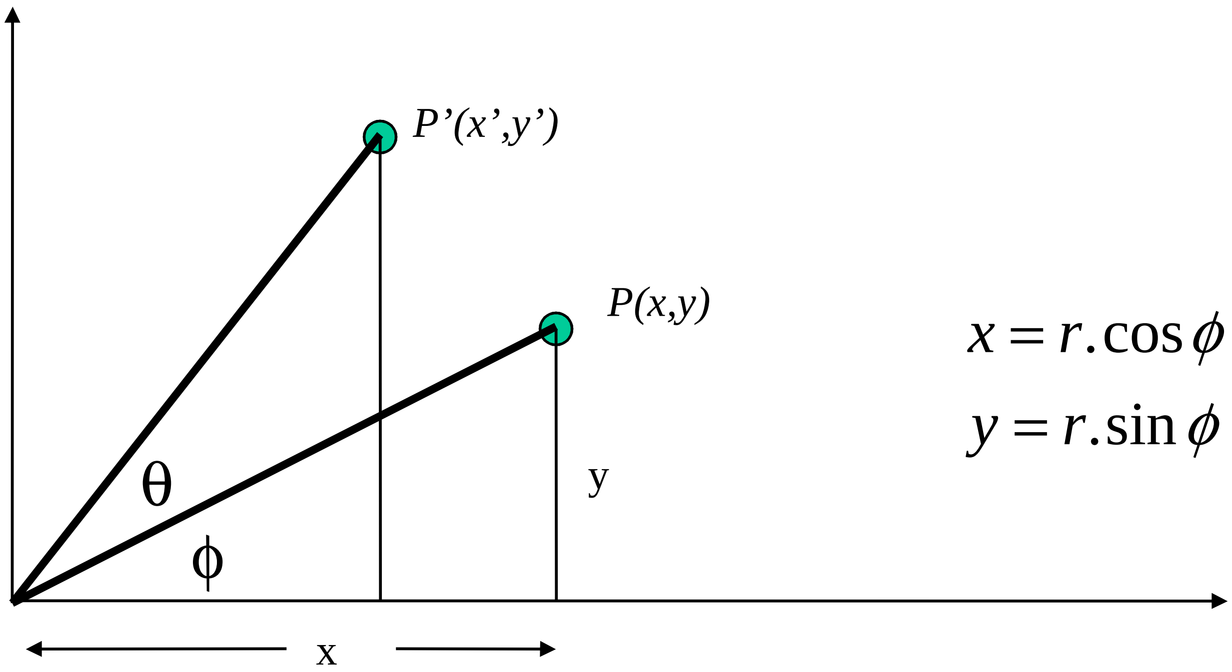
$$x = r.\cos\phi$$

$$y = r.\sin\phi$$

2D Rotation about the origin.

$$x' = r.\cos(\theta + \phi) = r.\cos\phi.\cos\theta - r.\sin\phi.\sin\theta$$

$$y' = r.\sin(\theta + \phi) = r.\cos\phi.\sin\theta + r.\sin\phi.\cos\theta$$



$$x = r.\cos\phi$$

$$y = r.\sin\phi$$

2D Rotation about the origin.

$$x' = r.\cos(\theta + \phi) = r.\cos\phi.\cos\theta - r.\sin\phi.\sin\theta$$

$$y' = r.\sin(\theta + \phi) = r.\cos\phi.\sin\theta + r.\sin\phi.\cos\theta$$

Substituting for r :

$$x = r.\cos\phi$$

$$y = r.\sin\phi$$

Gives us :

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

$$y' = x.\sin\theta + y.\cos\theta$$

2D Rotation about the origin.

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

$$y' = x.\sin\theta + y.\cos\theta$$

Rewriting in matrix form gives us :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\text{Define the matrix } R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad P' = R \cdot P$$

Transformations.

- Translation.
  - $P' = T + P$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

- Scale
  - $P' = S \cdot P$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

- Rotation
  - $P' = R \cdot P$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

- We would like all transformations to be multiplications so we can concatenate them
- $\Rightarrow$ express points in homogenous coordinates.

Homogeneous coordinates

- Add an extra coordinate, W, to a point.
  - *P(x,y,W).*
- Two sets of homogeneous coordinates represent the same point if they are a multiple of each other.
  - (2,5,3) and (4,10,6) represent the same point.
- If $W \neq 0$, divide by it to get Cartesian coordinates of point (x/W,y/W,1).
- If W=0, point is said to be at infinity.

Translations in homogenised coordinates

- Transformation matrices for 2D translation are now 3x3.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x + d_x$$
$$y' = y + d_y$$
$$1 = 1$$

Concatenation.

- When we perform 2 translations on the same point

$$P' = T(d_{x1}, d_{y1}) \cdot P$$

$$P'' = T(d_{x2}, d_{y2}) \cdot P'$$

$$P'' = T(d_{x1}, d_{y1}) \cdot T(d_{x2}, d_{y2}) \cdot P = T(d_{x1} + d_{x2}, d_{y1} + d_{y2}) \cdot P$$

So we expect:

$$T(d_{x1}, d_{y1}) \cdot T(d_{x2}, d_{y2}) = T(d_{x1} + d_{x2}, d_{y1} + d_{y2})$$

Concatenation.

The matrix product $T(d_{x1}, d_{y1}) \cdot T(d_{x2}, d_{y2})$ is:

$$\begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} = ?$$

Matrix product is variously referred to as *compounding*, *concatenation*, or *composition*

Concatenation.

The matrix product $T(d_{x1}, d_{y1}) \cdot T(d_{x2}, d_{y2})$ is:

$$\begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y1} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix product is variously referred to as *compounding*, *concatenation*, or *composition.*

Properties of translations.

1. $T(0,0) = I$
2. $T(s_x, s_y) \cdot T(t_x, t_y) = T(s_x + t_x, s_y + t_y)$
3. $T(s_x, s_y) \cdot T(t_x, t_y) = T(t_x, t_y) \cdot T(s_x, s_y)$
4. $\mathrm{T}^{-1}(s_x, s_y) = T(-s_x, -s_y)$

Note : 3. translation matrices are *commutative.*

Homogeneous form of scale.

Recall the (x,y) form of Scale :

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

In homogeneous coordinates :

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Concatenation of scales.

The matrix product $S(s_{x1}, s_{y1}) \cdot S(s_{x2}, s_{y2})$ is:

$$\begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1} \cdot s_{x2} & 0 & 0 \\ 0 & s_{y1} \cdot s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Only diagonal elements in the matrix - easy to multiply!

Homogeneous form of rotation.

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

For rotation matrices,

$$R^{-1}(\theta) = R(-\theta).$$

Rotation matrices are orthogonal, i.e :

$$R^{-1}(\theta) = R^{T}(\theta)$$

Orthogonality of rotation matrices.

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R^T(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Other properties of rotation.

$$R(0) = I$$

$$R(\theta) \cdot R(\phi) = R(\theta + \phi)$$

and

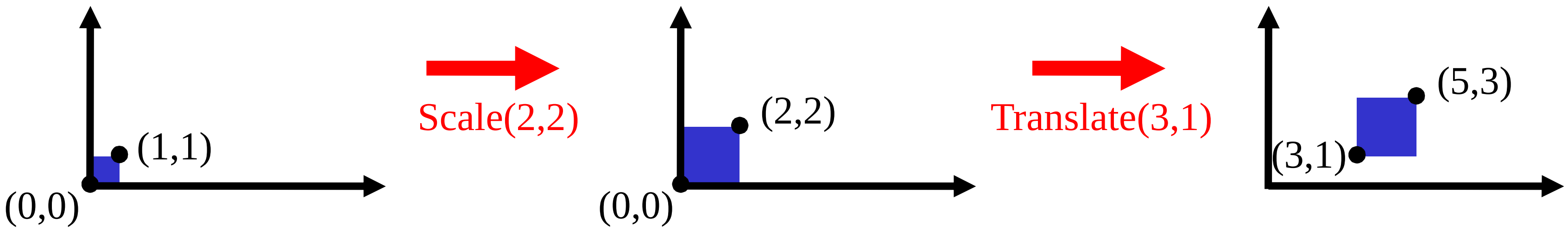
$$R(\theta) \cdot R(\phi) = R(\phi) \cdot R(\theta)$$

But this is only because the axis of rotation

is the same

For 3D rotations, need to be more careful

Scale then Translate

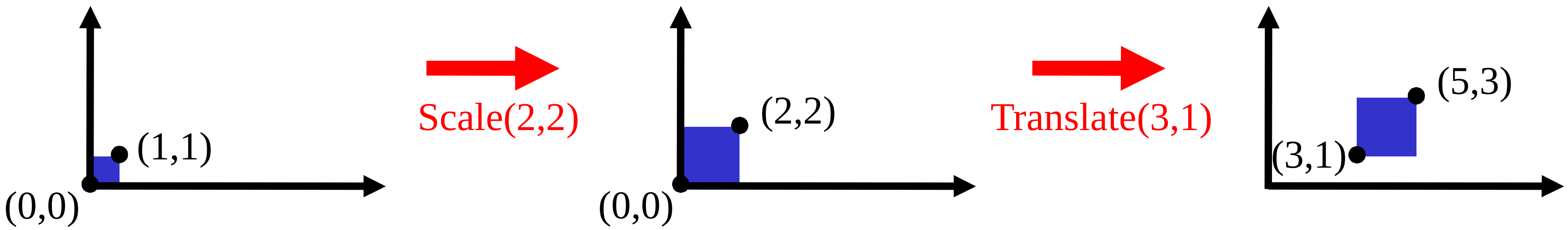


Use matrix multiplication: p' = T (S p) = TS p

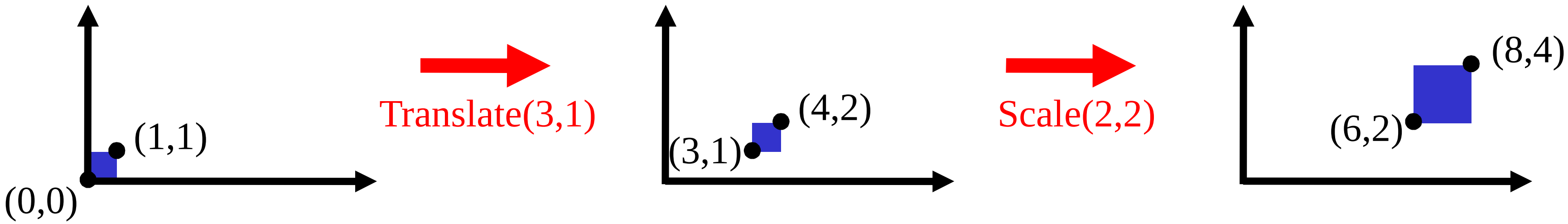
$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Caution: matrix multiplication is NOT commutative!

Scale then Translate: $p' = T(Sp) = TSp$



Translate then Scale: $p' = S(Tp) = STp$



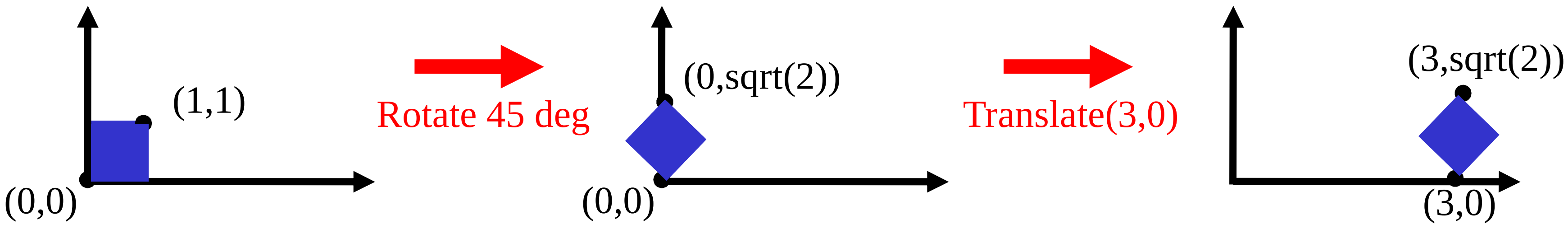
Scale then Translate: p' = T (S p) = TS p

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

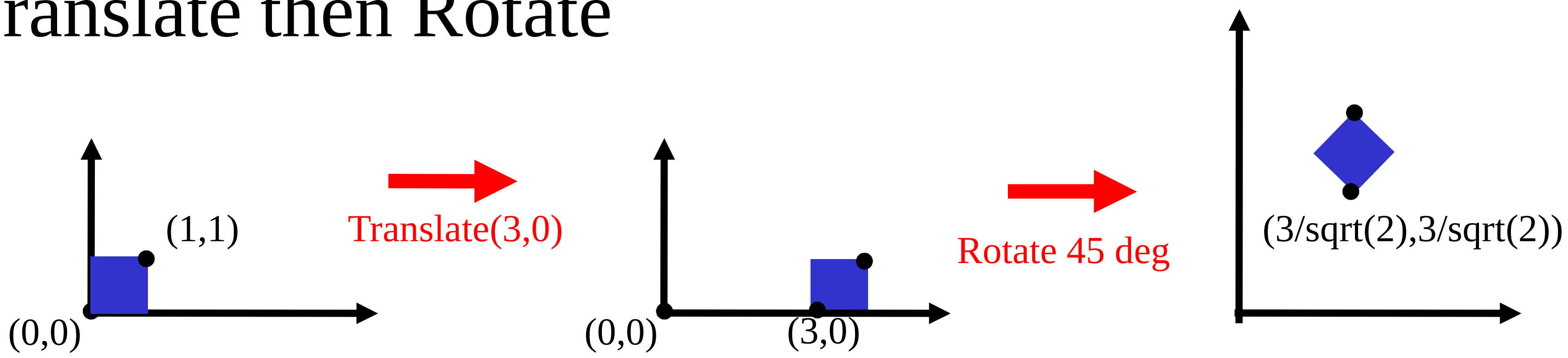
Translate then Scale: p' = S (T p) = ST p

$$ST = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotate then Translate



Translate then Rotate



Caution: matrix multiplication is NOT commutative!

Rotate then Translate: p' = T (R p) = TR p

$$TR = \begin{pmatrix} 1 & 0 & \color{red}{3} \\ 0 & 1 & \color{red}{1} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \color{red}{0} & -1 & \color{red}{3} \\ 0 & \color{red}{0} & \color{red}{1} \\ 0 & 0 & 1 \end{pmatrix}$$

Translate then Rotate: p' = R (T p) = RT p

$$RT = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \color{red}{3} \\ 0 & 1 & \color{red}{1} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & \color{red}{-1} \\ 1 & \color{red}{0} & \color{red}{3} \\ 0 & 0 & \color{red}{1} \end{pmatrix}$$

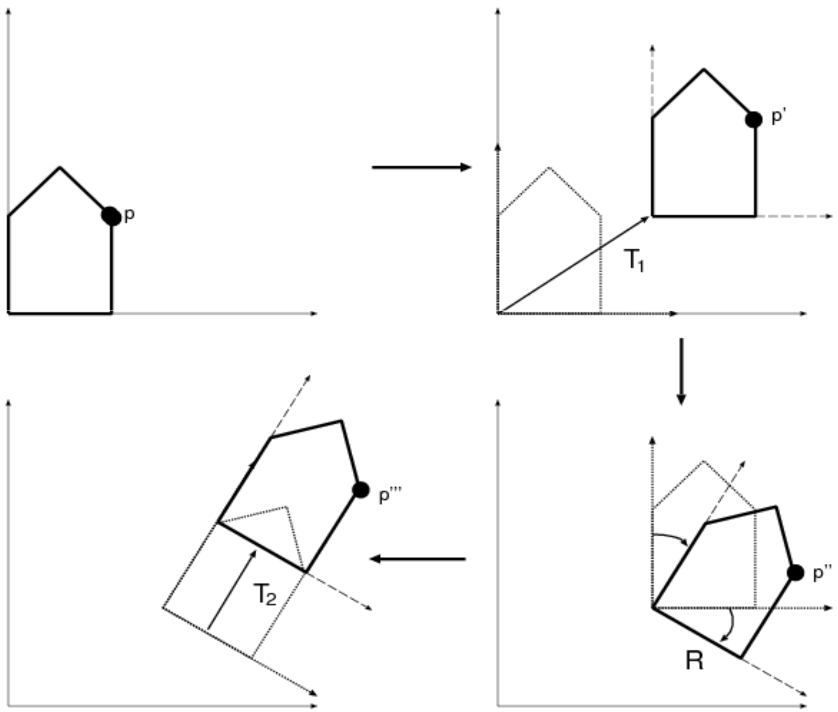
Types of transformations.

- Rotation and translation
  - Angles and distances are preserved
  - Unit cube is always unit cube
  - *Rigid-Body* transformations.
- Rotation, translation and scale.
  - Angles & distances not preserved.
  - But parallel lines are.

Transformations of coordinate systems.

- Have been discussing transformations as transforming points.
- Always need to think the transformation in the world coordinate system
- Sometimes this might not be so convenient
  - i.e. rotating objects at its location

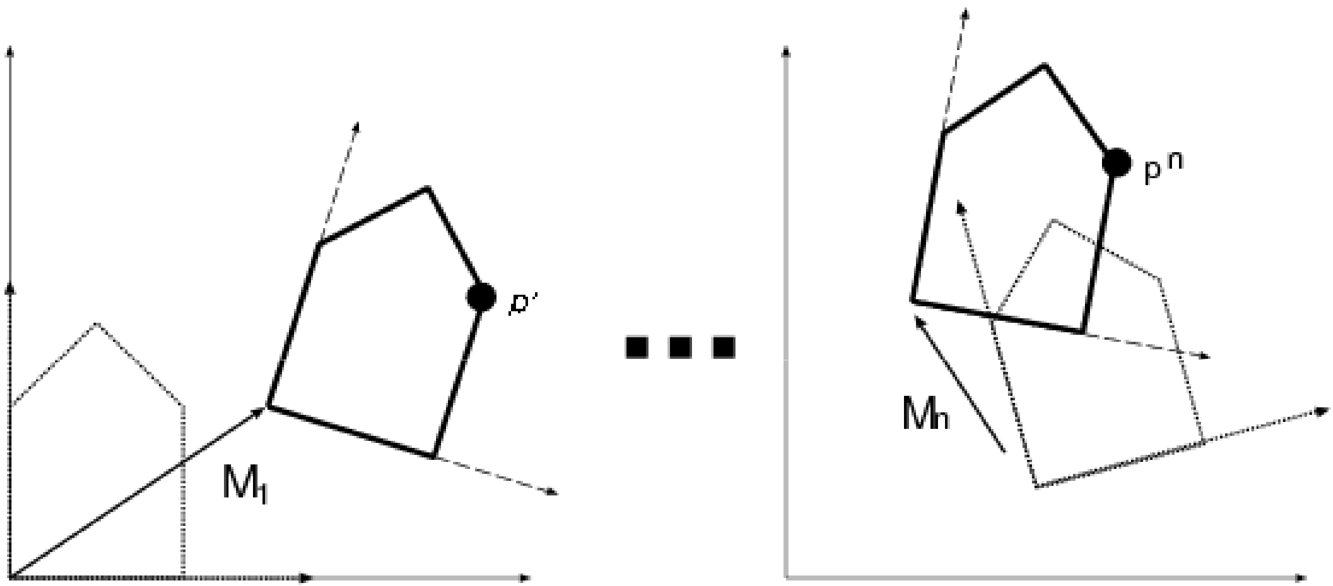
Transformations of coordinate systems - Example



p' = T1 p
p'' = T1 R p
p''' = T1 R T2 p

- Concatenate local transformation matrices from left to right
- Can obtain the local – world transformation matrix
- p',p'',p''' are the world coordinates of p after each transformation

Transformations of coordinate systems - example



$$p^n = M_1 M_2 \cdots M_n \, p$$

p^n is the world coordinate of point p after n transformations

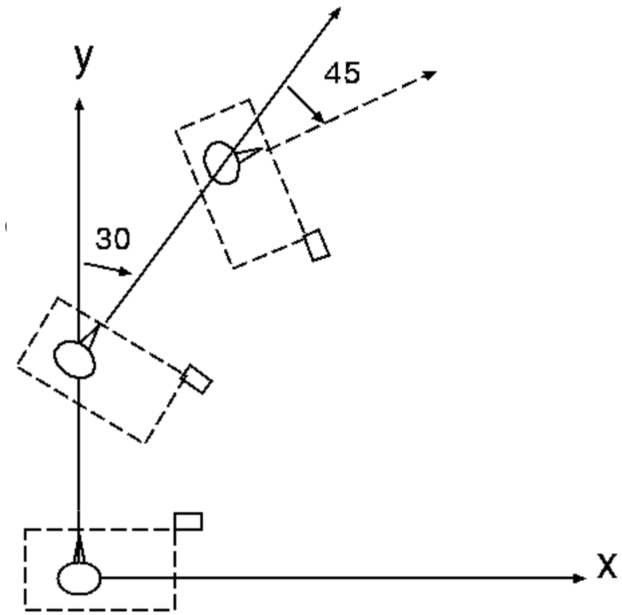
Quiz

I sat in the car, and find the side mirror is 0.4m on my right and 0.3m in my front

I started my car and drove 5m forward, turned 30 degrees to right, moved 5m forward again, and turned 45 degrees to the right, and stopped

What is the position of the side mirror now, relative to where I was sitting in the beginning?

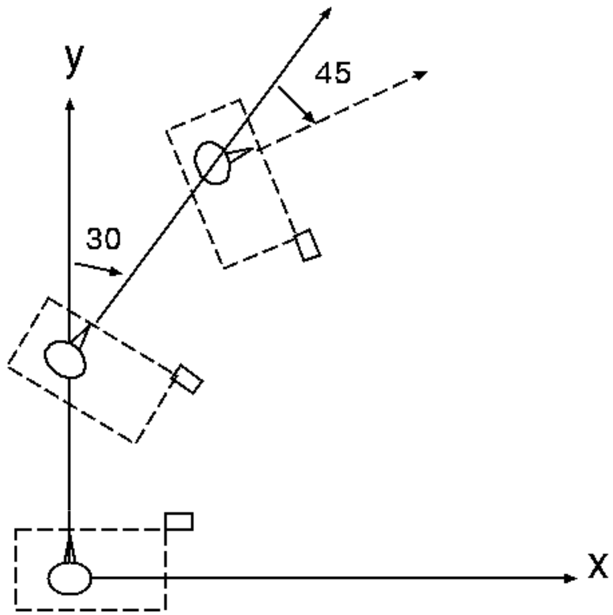
Solution



- The side mirror position is loc
  (0,4,0.3)
- The matrix of first driving
  forward 5m is

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

Solution



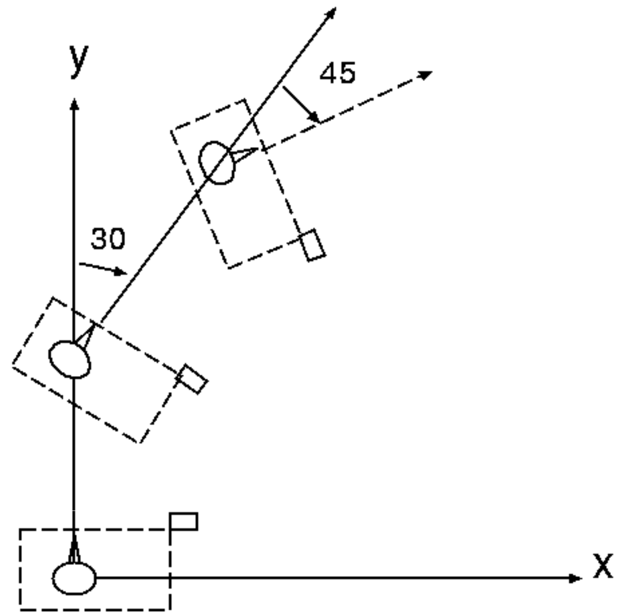
- The matrix to turn to the right 30 and 45 degrees (rotating -30 and -45 degrees around the origin) are

$$R_1 = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{respectively}$$

Solution

The local-to-global transformation matrix at the last configuration of the car is

$$TR_1TR_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

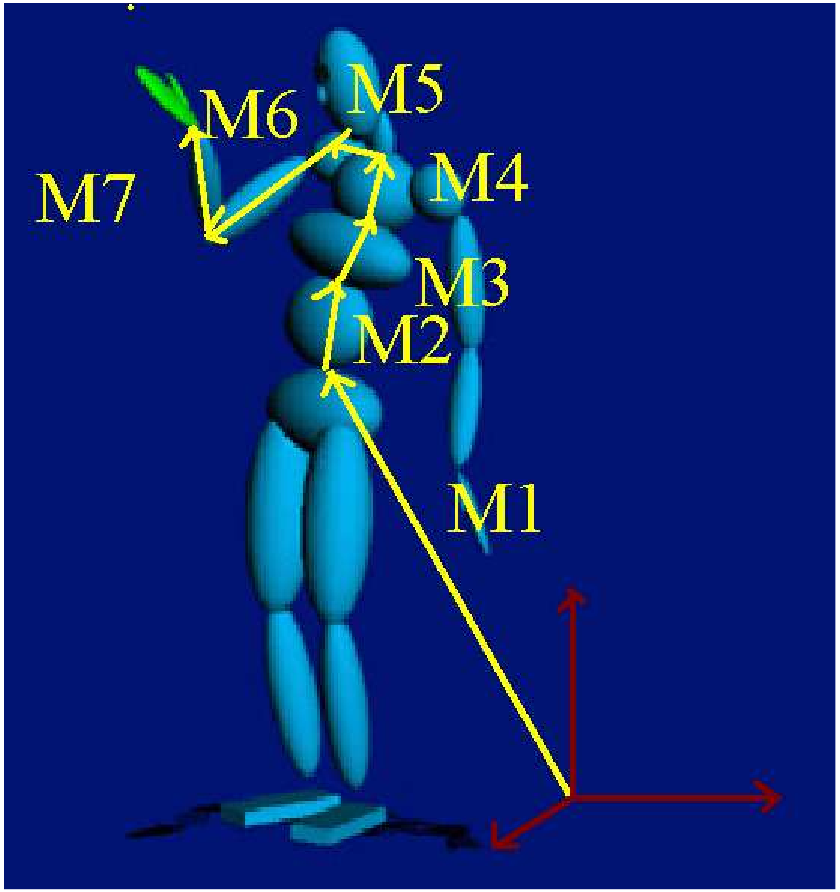


The final position of the side mirror can be computed by $TR_1TR_2\, p$ which is around (2.89331, 9.0214)

This is convenient for character animation / robotics

In robotics / animation, we often want to know what is the current 3D location of the end effectors (like the hand)

Can concatenate matrices from the origin of the body towards the end effecter



Transformations of coordinate systems.

Define $P^{(i)}$ as a point in coordinate system i

Define $M_{i \leftarrow j}$ as the transform that converts a point in system j to a point in system i

$P^{(i)} = M_{i \leftarrow j} \cdot P^{(j)}$

and

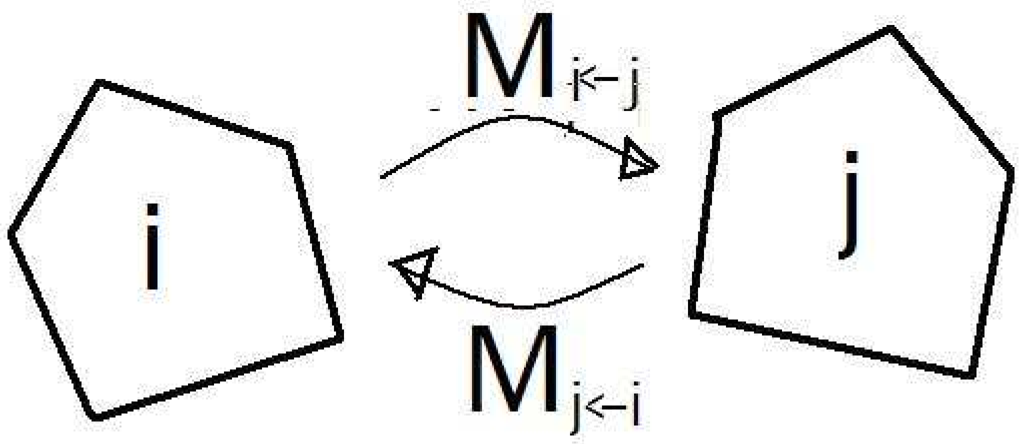
$P^{(j)} = M_{j \leftarrow k} \cdot P^{(k)}$

we obtain by substitution :

$M_{i \leftarrow k} = M_{i \leftarrow j} \cdot M_{j \leftarrow k}$

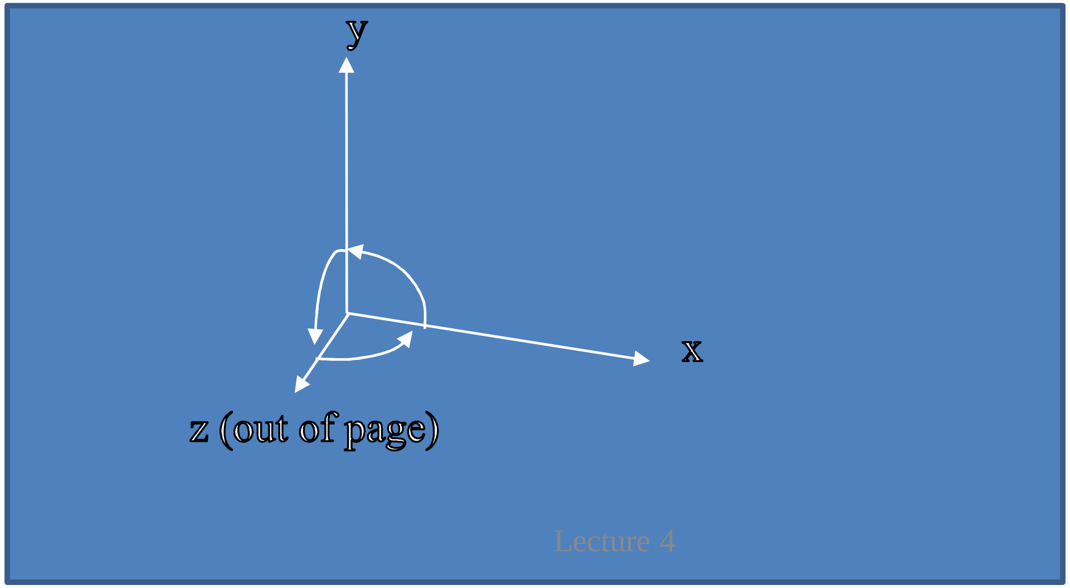
It can also be shown that :

$M_{j \leftarrow i} = M_{i \leftarrow j}^{-1}$



3D Transformations.

- Use homogeneous coordinates, just as in 2D case.
- Transformations are now 4x4 matrices.
- We will use a right-handed (world) coordinate system - (z out of page).



Translation in 3D.

Simple extension to the 3D case:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scale in 3D.

Simple extension to the 3D case:

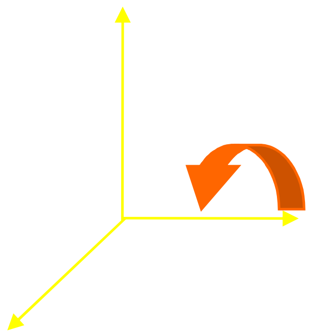
$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation in 3D

- Need to specify which axis the rotation is about.
- z-axis rotation is the same as the 2D case.

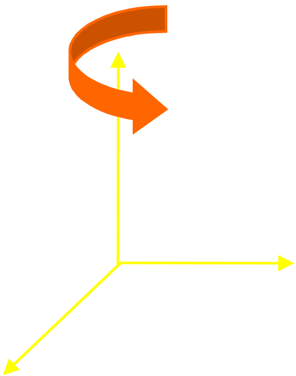
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotating About the x-axis $R_x(\theta)$



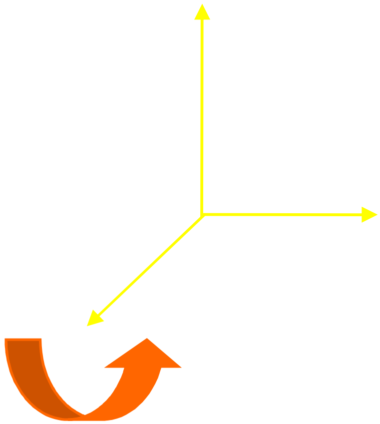
$$\begin{pmatrix} x \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotating About the y-axis
$R_y(\theta)$



$$\begin{pmatrix} x' \\ y \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

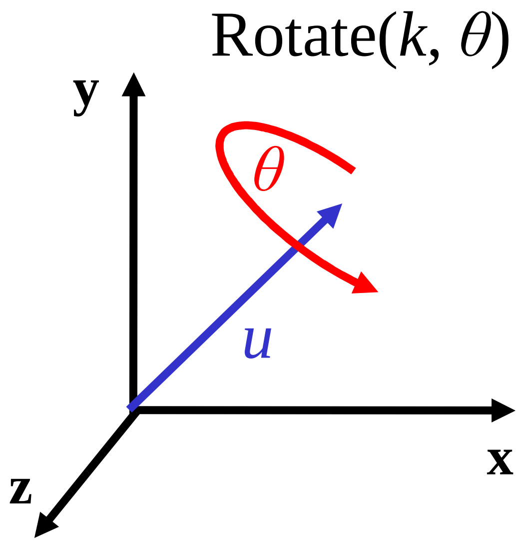
Rotation About the z-axis $R_z(\theta)$



$$\begin{pmatrix} x' \\ y' \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotation about an arbitrary axis

- About (u_x, u_y, u_z), a unit vector on an arbitrary axis



$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} u_x u_x(1-c)+c & u_z u_x(1-c)-u_z s & u_x u_z(1-c)+u_y s & 0 \\ u_y u_x(1-c)+u_z s & u_z u_x(1-c)+c & u_y u_z(1-c)-u_x s & 0 \\ u_z u_x(1-c)-u_y s & u_y u_z(1-c)+u_x s & u_z u_z(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

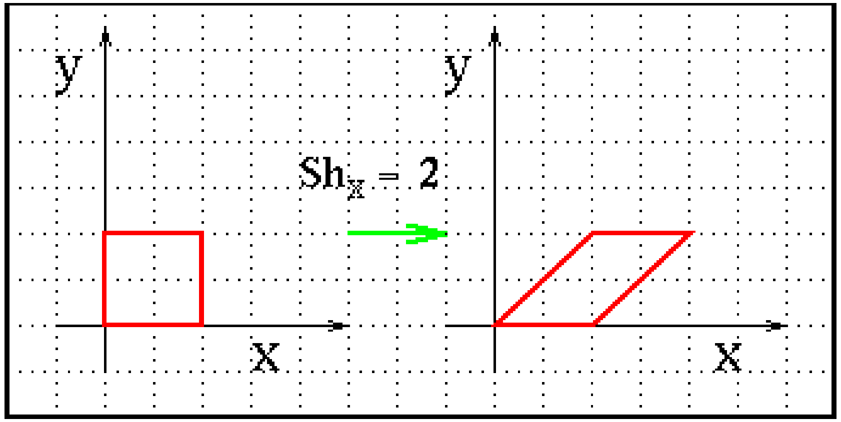
where $c = \cos\theta$ & $s = \sin\theta$

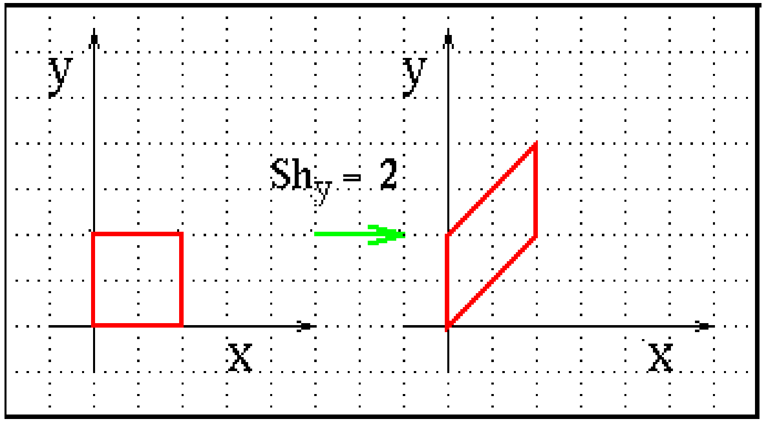
Rotation

- Not commutative if the axis of rotation are not parallel

$$R_x(\alpha)R_y(\beta) \neq R_y(\beta)R_x(\alpha)$$

Shearing



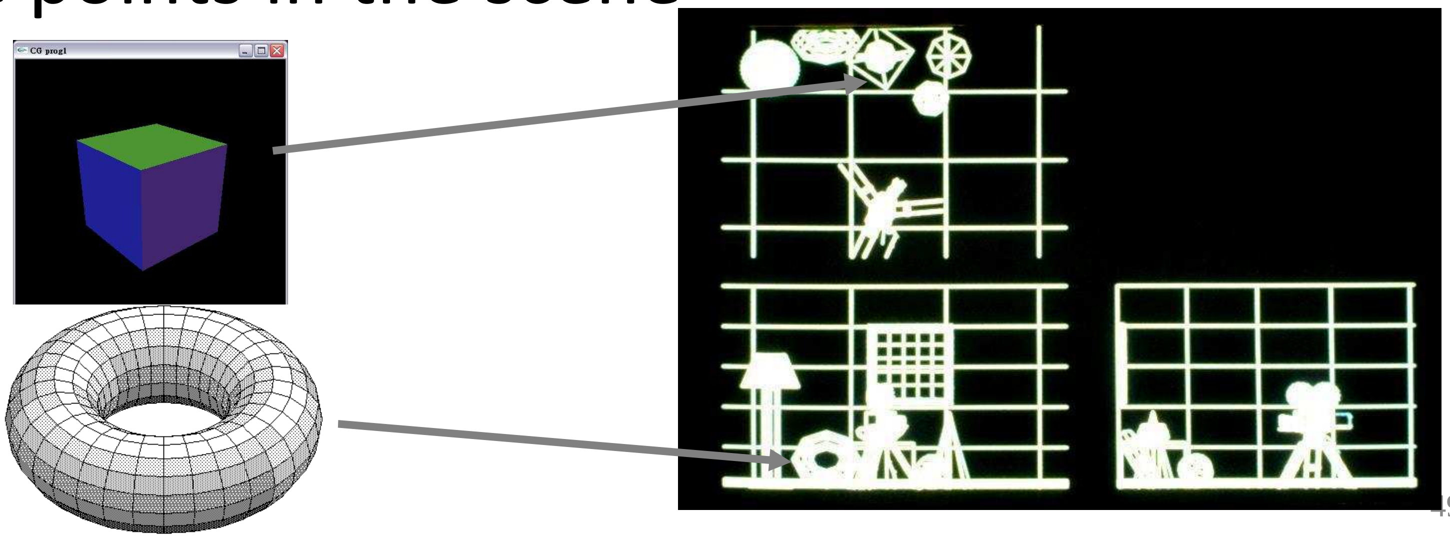


$$\begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Calculating the world coordinates of all vertices

- For each object, there is a local-to-global transformation matrix
- So we apply the transformations to all the vertices of each object
- We now know the world coordinates of all the points in the scene



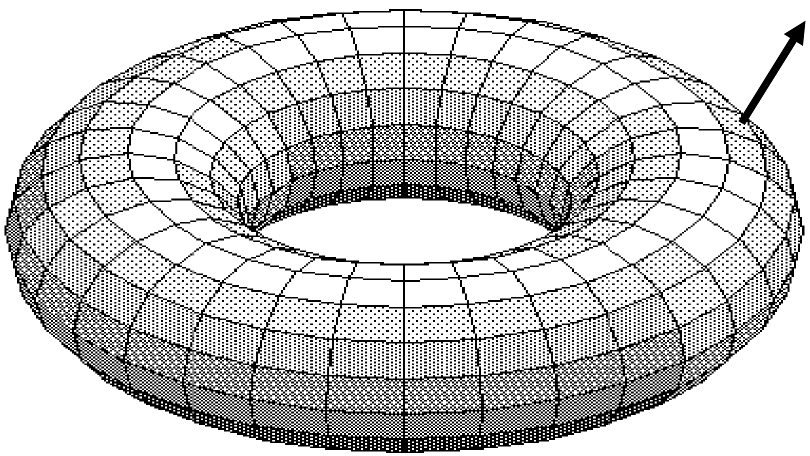
Normal Vectors

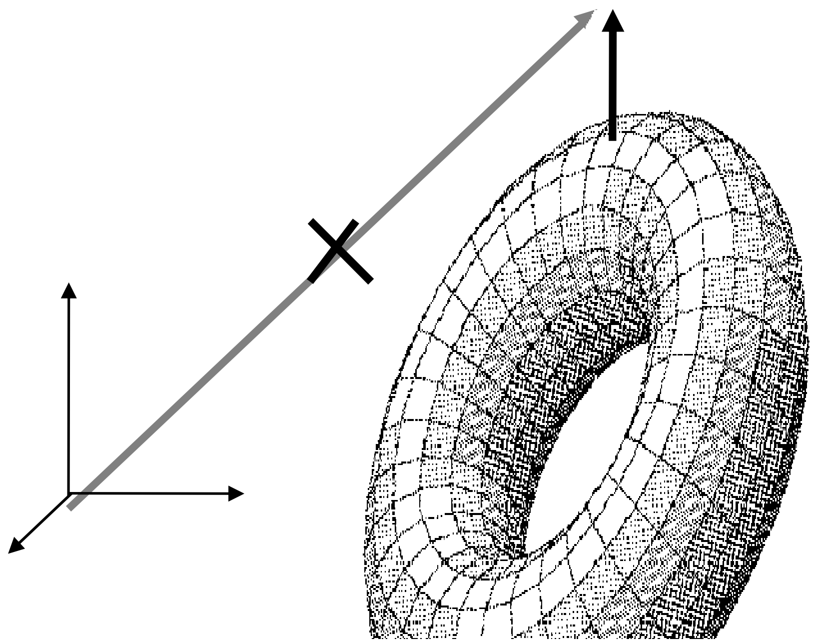
We also need to know the direction of the normal vectors in the world coordinate system

This is going to be used at the shading operation

We only want to rotate the normal vector

Do not want to translate it





Normal Vectors - (2)

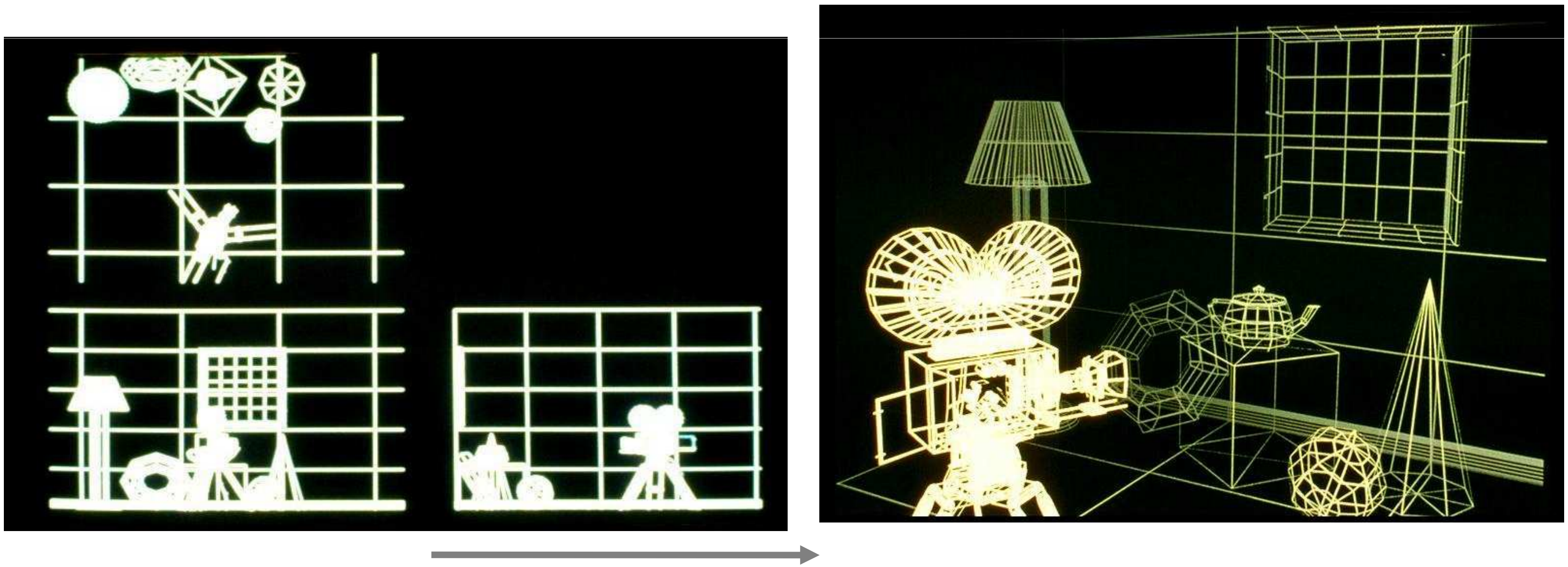
We need to set elements of the translation part to zero

$$\begin{bmatrix} r_{11} & r_{11} & r_{11} & t_x \\ r_{11} & r_{11} & r_{11} & t_y \\ r_{11} & r_{11} & r_{11} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} r_{11} & r_{11} & r_{11} & 0 \\ r_{11} & r_{11} & r_{11} & 0 \\ r_{11} & r_{11} & r_{11} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Viewing

Now we have the world coordinates of all the vertices

Now we want to convert the scene so that it appears in front of the camera

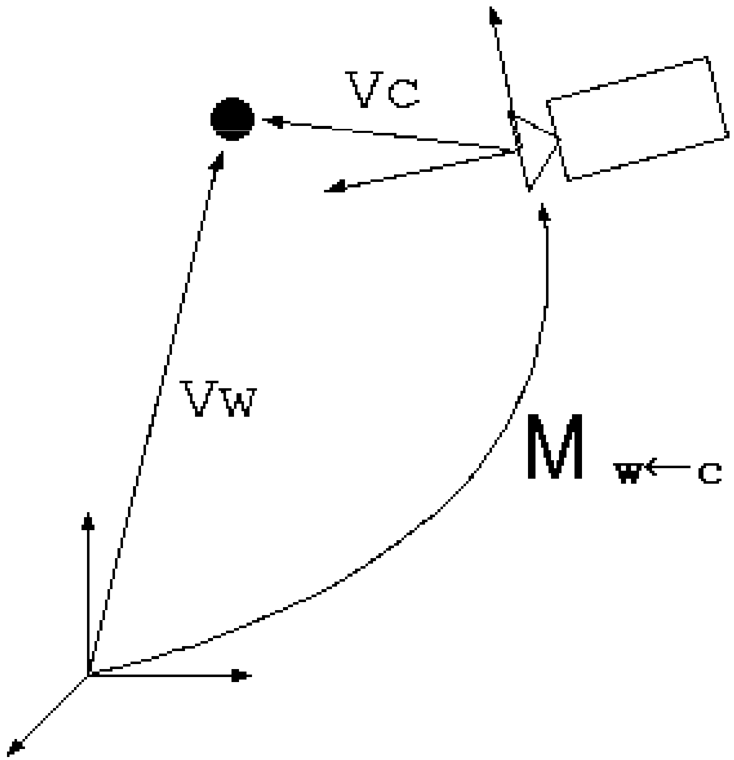


View Transformation

We want to know the positions in the camera coordinate system

We can compute the camera-to-world transformation matrix using the orientation and translation of the camera from the origin of the world coordinate system

$M_{w \leftarrow c}$



View Transformation

We want to know the positions in the camera coordinate system

$$V_w = M_{w \leftarrow c}\ V_c$$

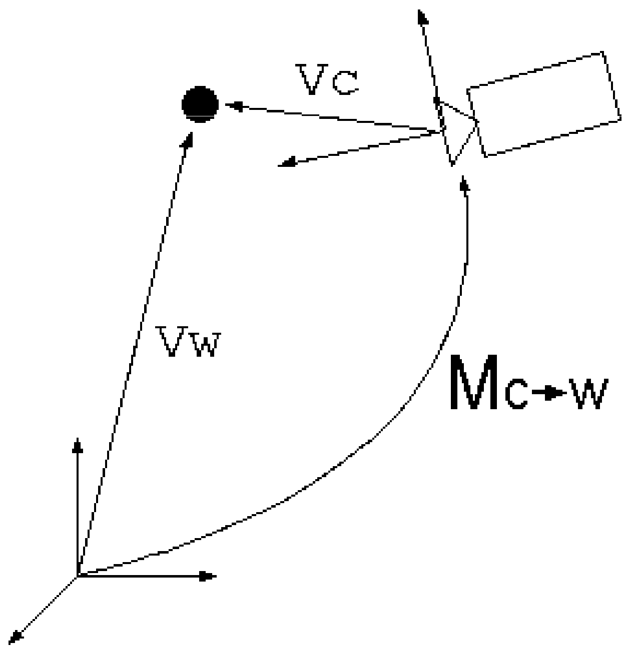
Point in the camera coordinate

Point in the world coordinate

Camera-to-world transformation

$$V_c = M^{-1}_{w \leftarrow c}\ V_w$$

$$= M_{c \leftarrow w}\ V_w$$



Summary.

- Transformations: translation, rotation and scaling
- Using homogeneous transformation, 2D (3D) transformations can be represented by multiplication of a 3x3 (4x4) matrix
- Multiplication from left-to-right can be considered as the transformation of the coordinate system
- Need to multiply the camera matrix from the left at the end
- Reading: Foley et al. Chapter 5, Appendix 2 sections A1 to A5 for revision and further background (Chapter 5)