Chapter 4
Geometric Objects and Transformations

Geometric transformations of mathematics

Key Contents

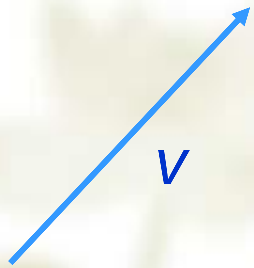
1. Space: linear vector, affine, Euclidean
2. Vector, Dimension and basis
3. Point and Coordinate Systems
4. Vector Operations and Application
5. Line, Plane and Parametric equation
6. Parametric equation of curve, surface
7. Geometry Relation of Line and Plane
8. Triangle-Based Collision Detection
9. Transformations

Space: linear vector, affine, Euclidean

- Linear Vector Space: Direction, Magnitude
- Affine Space: Point, Vector
- Euclidean Space: Geometry with Distance and Angle

Vectors

- Physical definition: a vector is a quantity with two attributes
  - Direction
  - Magnitude
- Examples include
  - Force
  - Velocity
  - Directed line segments
    - Most important example for graphics
    - Can map to other types



Linear Independence

- A set of vectors $v_1, v_2, \ldots, v_n$ is *linearly independent* if

  $$\alpha_1 v_1+\alpha_2 v_2+.. \alpha_n v_n=0 \quad \text{iff} \quad \alpha_1=\alpha_2=\ldots=0$$

- If a set of vectors is linearly independent, we cannot represent one in terms of the others
- If a set of vectors is linearly dependent, at least one can be written in terms of the others

Dimension, basis

- In a vector space, the maximum number of linearly independent vectors is fixed and is called the *dimension* of the space
- In an *n*-dimensional space, any set of n linearly independent vectors form a *basis* for the space
- Given a basis $v_1, v_2, \ldots, v_n$, any vector v can be written as

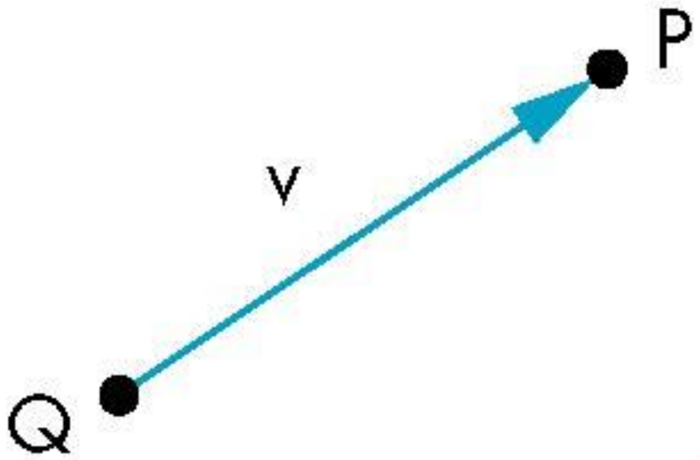
  $$v=\alpha_1 v_1+\alpha_2 v_2+\ldots+\alpha_n v_n$$

  where the $\{\alpha_i\}$ are unique

- Vectors spaces insufficient for geometry
  - Need points

Points and Vectors

- Location in space
- Operations allowed between points and vectors
  - Point-point subtraction yields a vector
  - Equivalent to point-vector addition

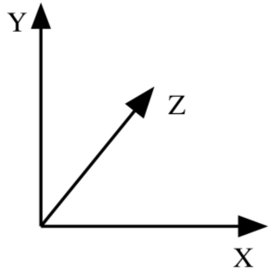


v=P-Q

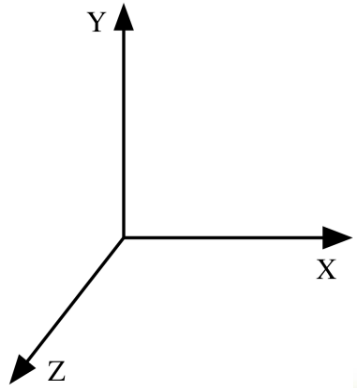
P=v+Q

Affine Spaces

- Point + basis is affine space



left-hand



right-hand

- Operations
  - Vector-vector addition $v = v_1 + v_2$
  - Scalar-vector multiplication $v = \mathrm{a}\, v_1$
  - Point-vector addition $Q = \mathrm{P} + v$
  - Point-Point subtraction $v = \mathrm{P} - \mathrm{Q}$

Coordinate Systems

- Consider a basis $v_1, v_2, \ldots, v_n$
- A vector is written $v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$
- The list of scalars $\{\alpha_1, \alpha_2, \ldots \alpha_n\}$ is the *representation* of v with respect to the given basis
- We can write the representation as a row or column array of scalars

$$\mathbf{a} = [\alpha_1 \;\; \alpha_2 \;\; \ldots \;\; \alpha_n]^T = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ \alpha_n \end{bmatrix}$$

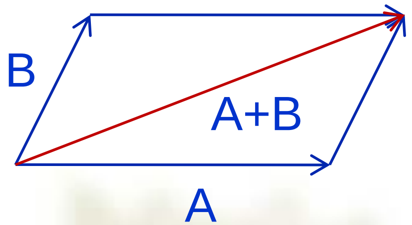
a is a N-Triple

Vector Operations

- Vectors have four primary operations

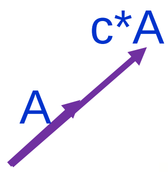
1. Addition

$$(a_1, a_2, a_3) + (b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$$



2. Scalar multiplication

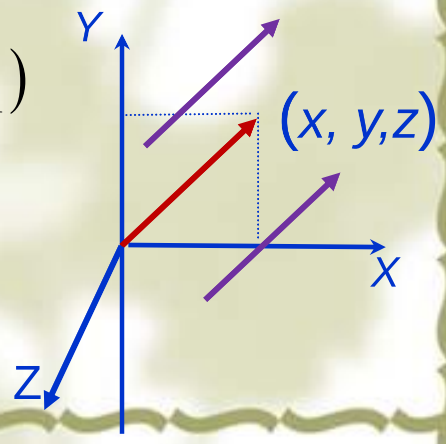
$$c * (a_1, a_2, a_3) = (c * a_1, c * a_2, c * a_3)$$



3. Dot product $(a_1, a_2, a_3) \bullet (b_1, b_2, b_3) = a_1 b_1 + a_2 b_2 + a_3 b_3$

4. Cross product

$$(a_1, a_2, a_3) \times (b_1, b_2, b_3) = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$



- Vectors have length, given by

$$\|v\| = \sqrt{(x^2 + y^2 + z^2)}$$

A *unit* vector has length 1

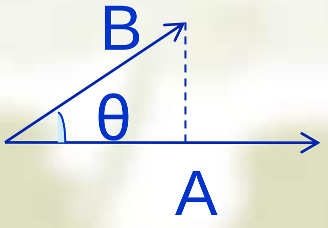
Dot Product （点积）

- Operates by adding the componentwise products and returns a scalar

$$(a_1, a_2, a_3) \bullet (b_1, b_2, b_3) = a_1 b_1 + a_2 b_2 + a_3 b_3$$

- For any two vectors A and B, if theta is the angle between the vectors,

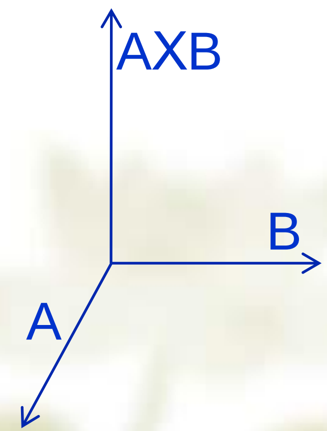
$$A \bullet B = \|A\| * \|B\| * \cos(\theta)$$



If two vectors have a dot product of zero, they are orthogonal (perpendicular to each other)

Cross Product（叉积）

- The cross product of two vectors yields another vector
- This vector is perpendicular to both of the original vectors
- The computation is given by a determinant calculation



$$(a_1, a_2, a_3) \times (b_1, b_2, b_3) = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$

Cross Product（叉积）

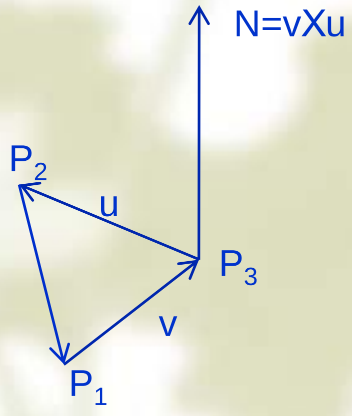
- Cross products are not commutative:

$$A \times B = -(B \times A)$$

- Cross products are very handy when you want to compute a vector normal to two given vectors (e.g. vertex normals)

$v = P_3 - P_1$, $u = P_2 - P_3$

$$N = v \times u$$

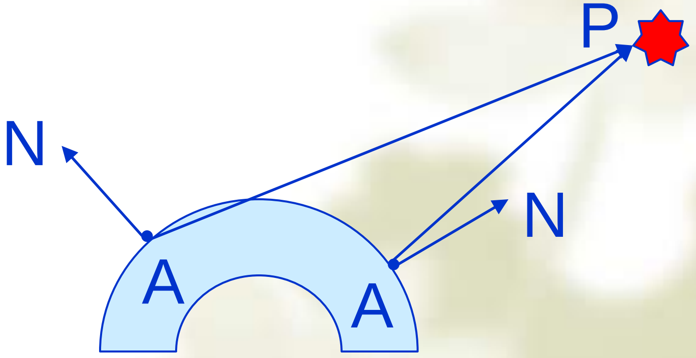


点积应用：light or not

- If you have a normal vector *N* at a point *A* and an incoming vector *P* (e.g. a light vector), how to decide a point A is light or not?

$N \cdot P = |N|\,|P| \cos\theta = \cos\theta$

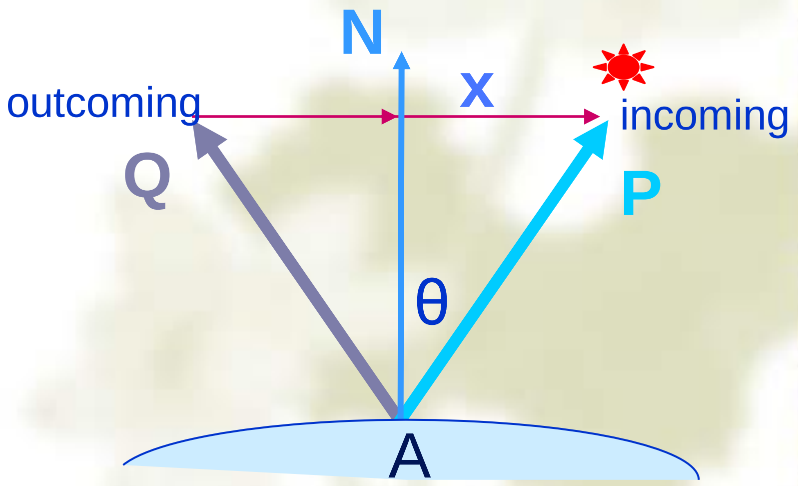
If $\cos\theta > 0$, light
If $\cos\theta < 0$, dark



点积应用：Reflection Vectors

- If you have a normal vector *N* at a point *A* and an incoming vector *P* (e.g. a light vector), you may need to compute the reflection vector *Q* of the incoming
- This is given by

$$Q = 2N\left(N \bullet P\right) - P$$



Calculating The *Q* Vector

∵ $\mathbf{N} \cdot \mathbf{P} = \|\mathrm{N}\|\,\|\mathrm{P}\| \cos\theta = \cos\theta$

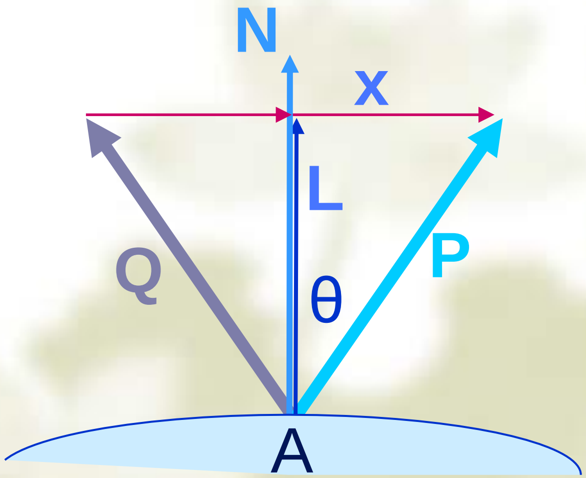
$\mathbf{L} = \mathbf{N} \cos\theta = \mathbf{N}(\mathbf{N} \cdot \mathbf{P})$

$\mathbf{X} = \mathbf{P} - \mathbf{L} = \mathbf{P} - \mathbf{N}(\mathbf{N} \cdot \mathbf{P})$

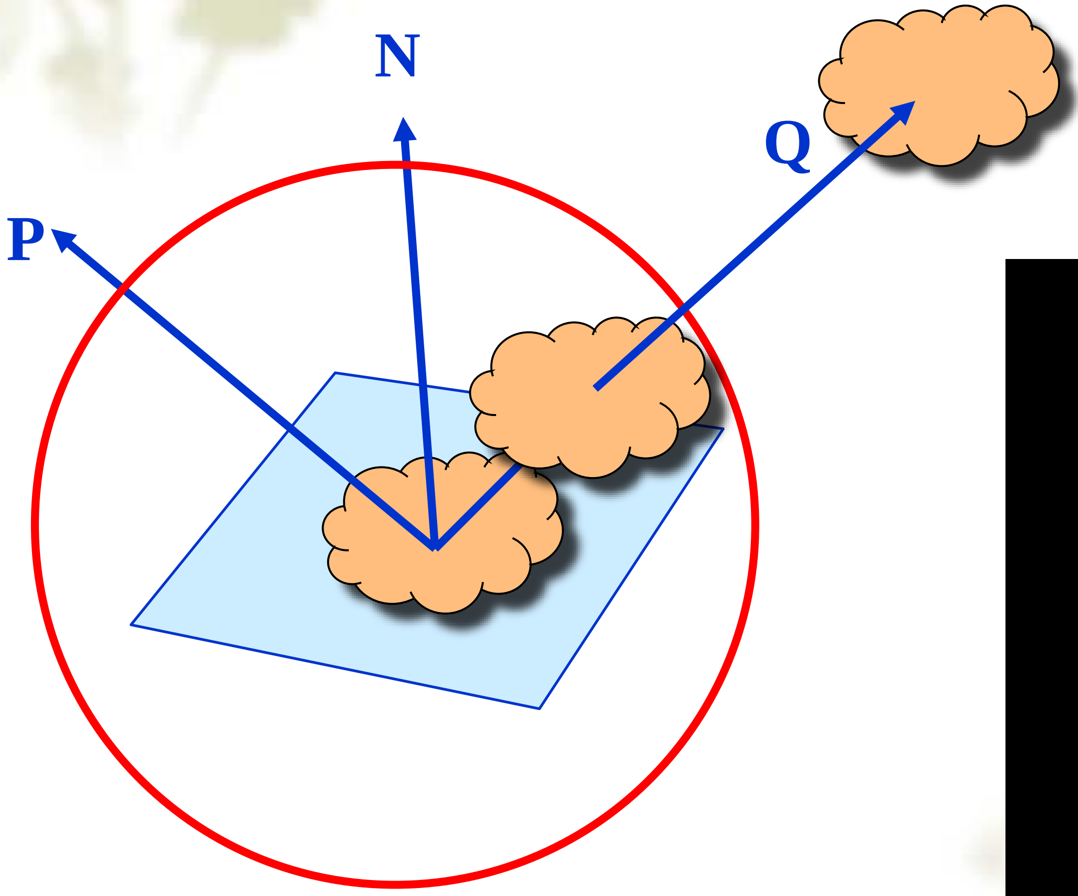
又∵ $\mathbf{Q} + 2\mathbf{X} = \mathbf{P}$

∴ $\mathbf{Q} = \mathbf{P} - 2\mathbf{X} = \mathbf{P} - 2\mathbf{P} + 2\mathbf{N}(\mathbf{N} \cdot \mathbf{P})$

$= 2\mathbf{N}(\mathbf{N} \cdot \mathbf{P}) - \mathbf{P}$



Mapping to a Sphere



Confusing Points and Vectors

Consider the point and the vector

$$P = P_0 + \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_n v_n$$

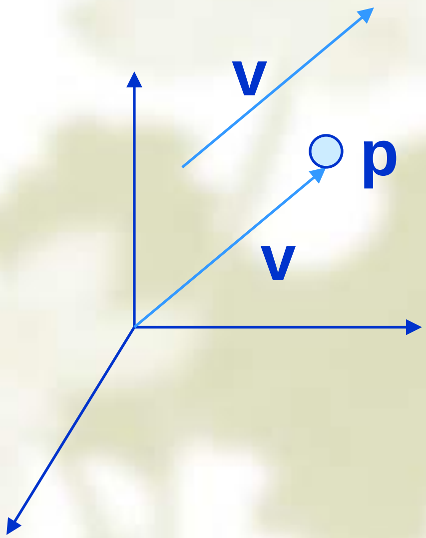
$$v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$$

They appear to have the similar representations

$\mathbf{p} = [\beta_1\ \beta_2\ \beta_3]$ $\qquad$ $\mathbf{v} = [\alpha_1\ \alpha_2\ \alpha_3]$

which confuses the point with the vector

A vector has no position



A Single Representation

If we define $0 \cdot P = \mathbf{0}$ and $1 \cdot P = P$ then we can write

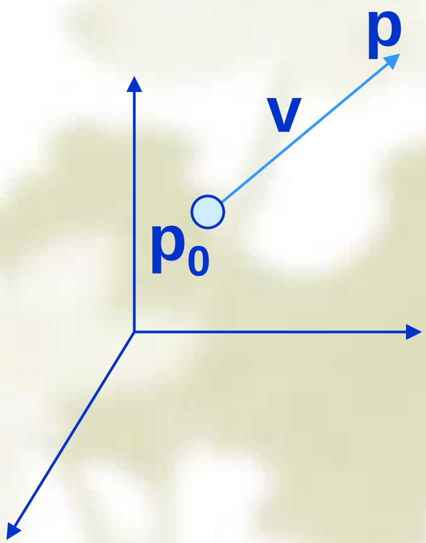
$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = [\alpha_1\ \alpha_2\ \alpha_3\ 0]\ [v_1\ v_2\ v_3\ P_0]^T$

$P = P_0 + \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 = [\beta_1\ \beta_2\ \beta_3\ 1]\ [v_1\ v_2\ v_3\ P_0]^T$

Thus we obtain the four-dimensional *homogeneous coordinate* representation

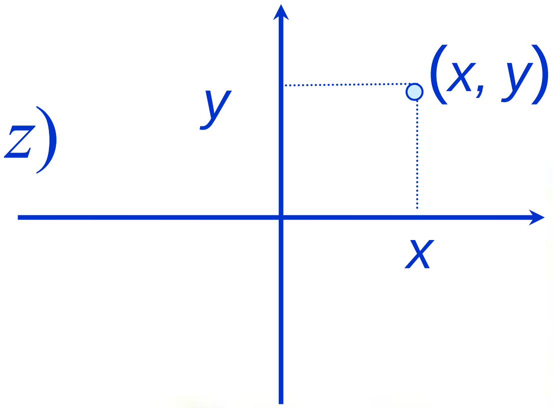
$\mathbf{v} = [\alpha_1\ \alpha_2\ \alpha_3\ 0]^T$

$\mathbf{p} = [\beta_1\ \beta_2\ \beta_3\ 1]^T$

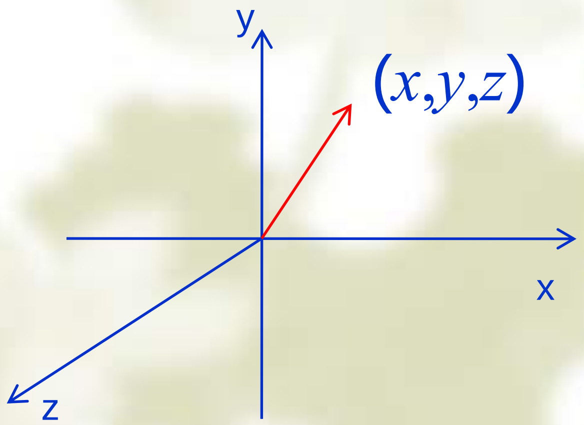


Points and Vectors

- Algebraically, a point is (x,y) or (x,y,z) as a position in affine space

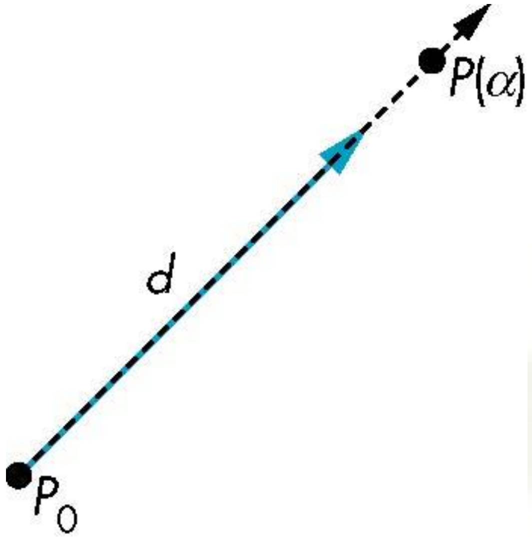


- However, a triple (x,y,z) or a quadruple (x,y,z,w) will sometimes have another meaning, such as
  - a normal vector(A,B,C)
  - a light direction(X,Y,Z)
  - a eye direction(X,Y,Z)
  - a color(R,G,B,A)



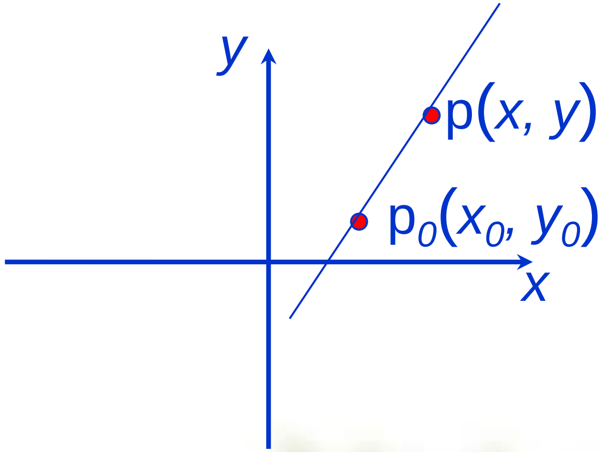
Lines

- Consider all points of the form
  - $P(\alpha)=P_0 + \alpha\,\mathbf{d}$
  - Set of all points that pass through P_0 in the direction of the vector $\mathbf{d}$



Lines and Parametric Equation

- Line functions:
  - $Ax+By+C=0$
  - $y=y_0+k(x-x_0)$
  - $y=kx+b$
  - $(y-y_0)/(y_1-y_0)=(x-x_0)/(x_1-x_0)$



- The line is through a given point P_0 offset by a fraction of the direction vector P_1-P_0(or the line's vector) with a scalar multiple

$$P = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1 \qquad (-\infty \leq t \leq \infty)$$

Or we say the *parametric equation* of the line through $P_0 P_1$:

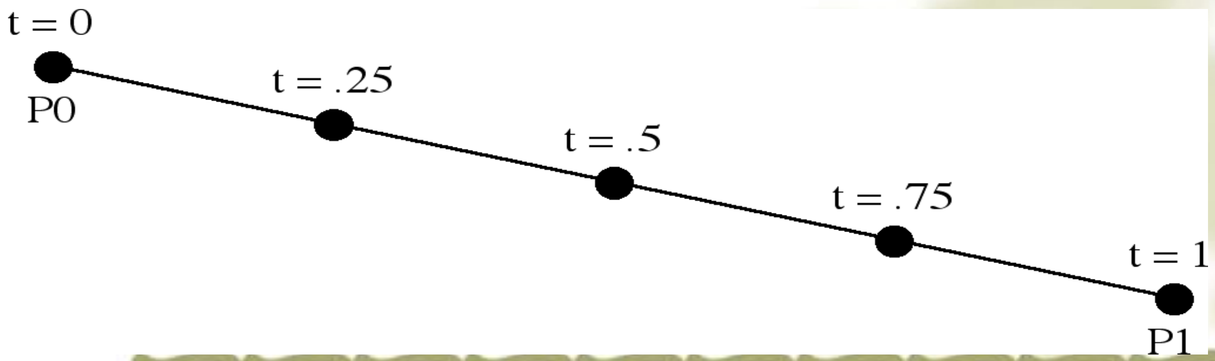
$$\begin{cases} x = x_0 + t(x_1 - x_0) \\ y = y_0 + t(y_1 - y_0) \\ z = z_0 + t(z_1 - z_0) \end{cases} \qquad -\infty \leq t \leq \infty$$

Rays and Line Segments

A ray is defined only points that lie in the same direction as the line's vector are included and the value of the parameter is $t \geq 0$. $$P = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1$$

A line segment is the all points between two points P_0 (x_0,y_0,z_0) and $P_1(x_1,y_1,z_1)$ and the value of the *parameter* is $0 \leq t \leq 1$ $$P = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1$$

When you define two adjacent vertices of a graphics object, the edge between them is a line segment



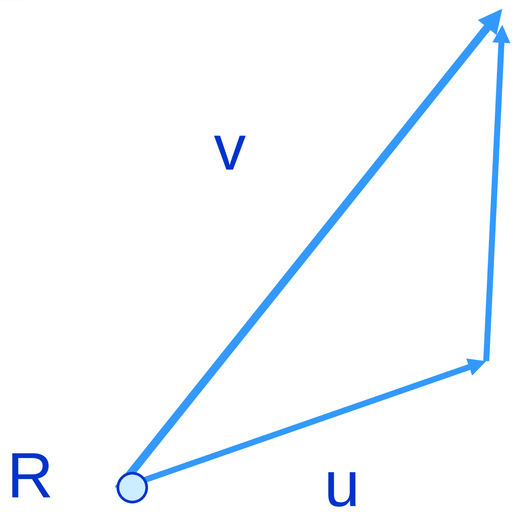
Quesion

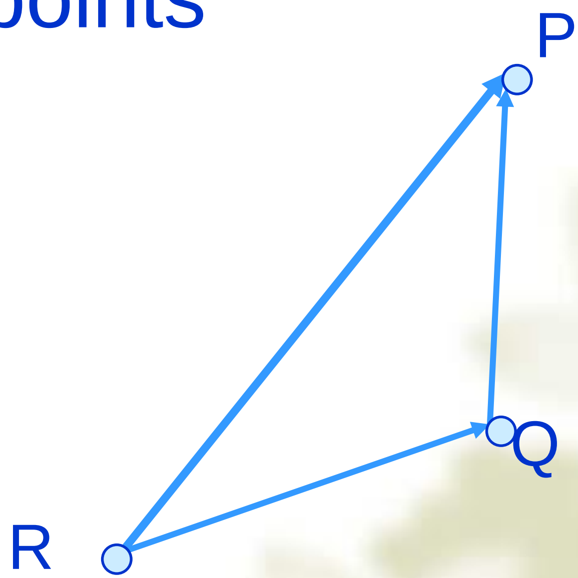
Why use parametric equation to represent a line or a plane in the Computer Graphics

- Coordinate systems independently
- More robust and general than other forms
- Extends to curves and surfaces

Planes and Parametric Equation

- A plane can be defined by a point and two vectors or by three points

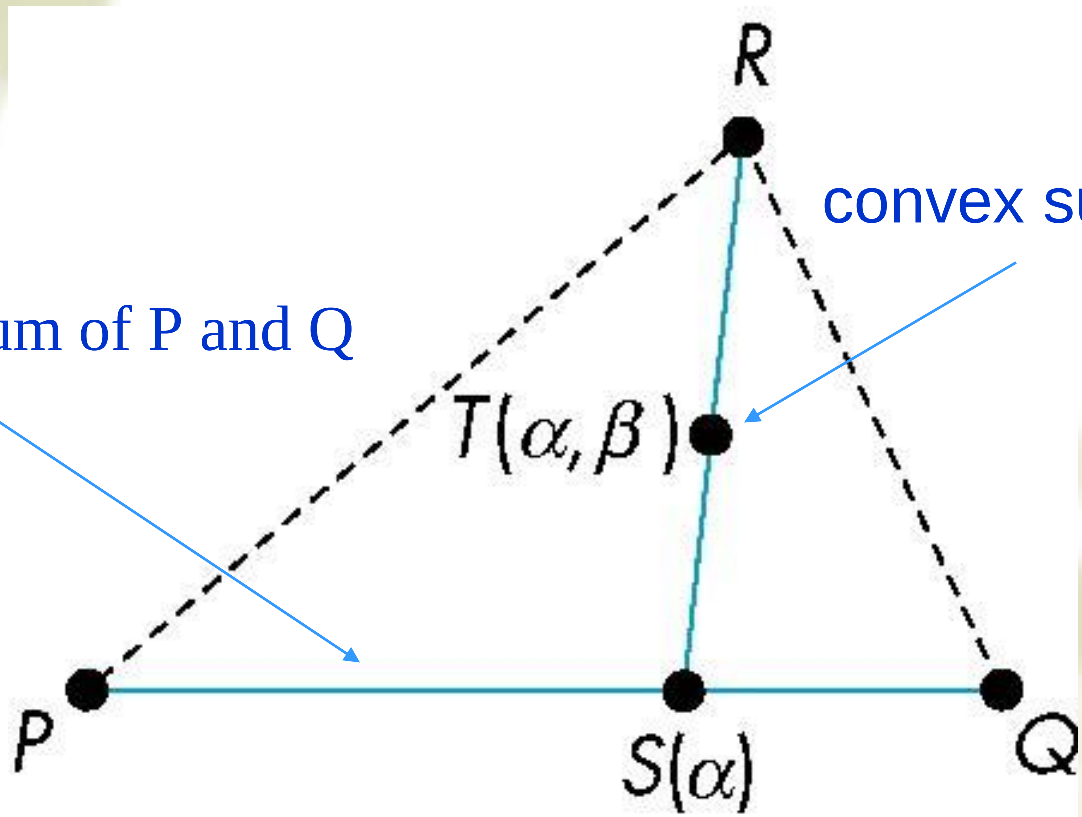




$P(\alpha,\beta)=R+\alpha u+\beta v$

$P(\alpha,\beta)=R+\alpha(Q-R)+\beta(P-R)$

Triangles



for 0<=α,β<=1, we get all points in triangle

Plane Equations

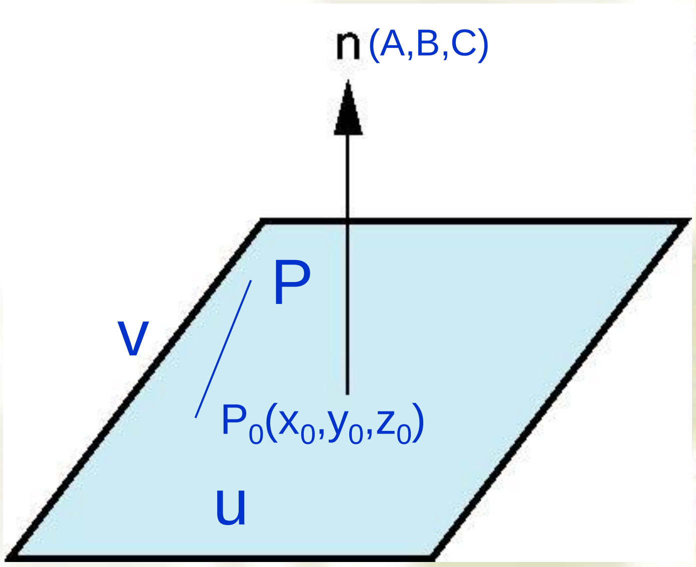
- Implicit equation for a plane:

$$Ax + By + Cz + D = 0$$

- Equation of a plane can be derived from the second definition

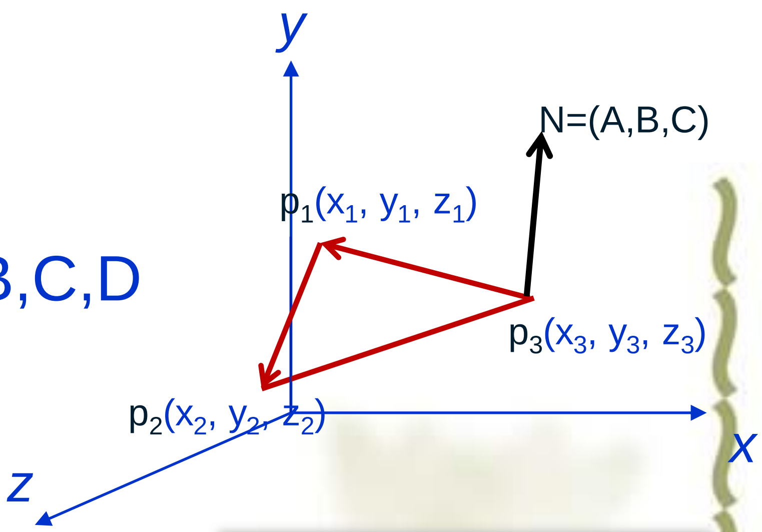
$$(A,B,C)\bullet(x-x_0,y-y_0,z-z_0)=0$$

Here the coefficients A, B, C are the components of the normal vector



Plane Equation

- Plane Equations
  - $Ax+By+Cz+D=0$
  - Three noncollinear points for A,B,C,D

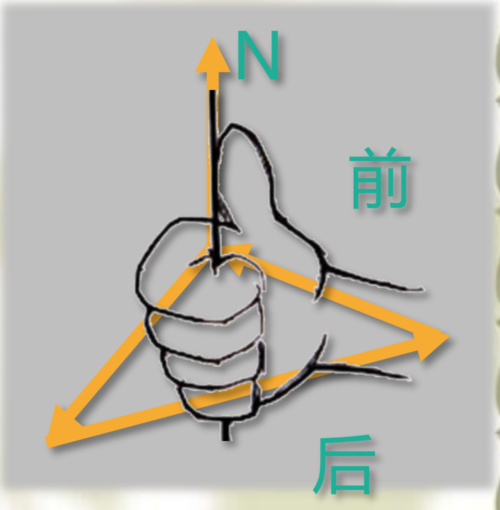


$\because \quad \mathbf{V} = p_1 - p_3 \qquad \mathbf{W} = p_2 - p_1$

$\therefore$ normal vector $\mathbf{N} = \mathbf{V} \times \mathbf{W}$

$A=v_2w_3 - v_3w_2, \quad B=v_3w_1 - v_1w_3, \quad C=v_1w_2 - v_2w_1$

$\therefore \quad D= -(Ax + By + Cz)= -(Ax_1 + By_1 + Cz_1)$



$$
\begin{aligned}
A &= y_1(z_2 - z_3) - y_2(z_1 - z_3) + y_3(z_1 - z_2) \\
B &= -x_1(z_2 - z_3) + x_2(z_1 - z_3) - x_3(z_1 - z_2) \\
C &= x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2) \\
D &= -x_1(y_2z_3 - y_3z_2) + x_2(y_1z_3 - y_3z_1) - x_3(y_1z_2 - y_2z_1)
\end{aligned}
$$

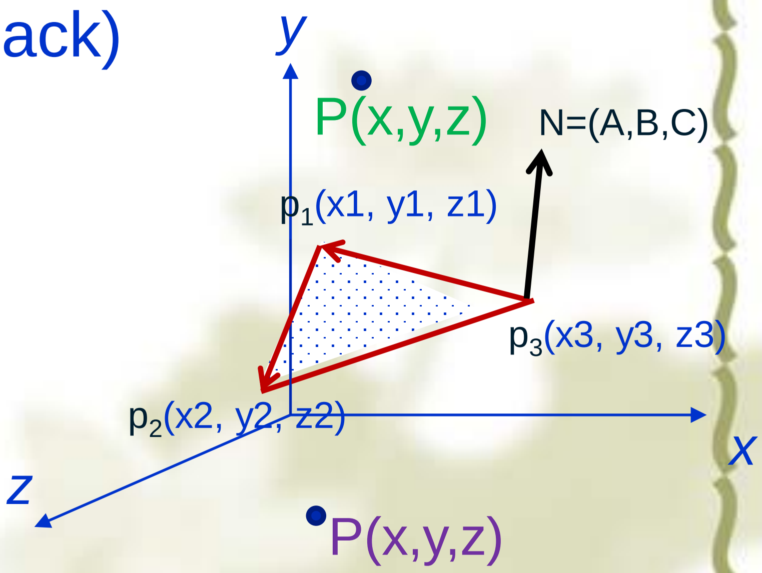
the Relations of a Plane and a Point

- All points(x,y,z) on positive side of the plane

  if $Ax + By + Cz > -D$ (in front)

- All points(x,y,z) on negative side of the plane

  if $Ax + By + Cz < -D$ (in back)



Polygon Tables

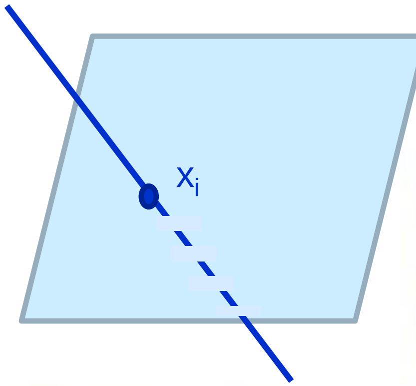
- Geometric tables : vertex table, edge table, surface-facet tables
- Attribute tables : color, normal, texture

The line segment intersects the plane

Line equation: $x = x_0 + t(x_1 - x_0)$

$y = y_0 + t(y_1 - y_0)$ $\quad 0 \le t \le 1$

$z = z_0 + t(z_1 - z_0)$



Plane equation: $Ax + By + Cz + D = 0$

So, $A(x_0 + t(x_1 - x_0)) + B(y_0 + t(y_1 - y_0)) + C(z_0 + t(z_1 - z_0)) + D = 0$

$t = -(Ax_0 + By_0 + Cz_0 + D)/(A(x_1 - x_0) + B(y_1 - y_0) + C(z_1 - z_0))$

If $0 \le t \le 1$,The point of intersection :

$x_i = x_0 + t(x_1 - x_0)$

$y_i = y_0 + t(y_1 - y_0)$

$z_i = z_0 + t(z_1 - z_0)$

If $t \le 0$ or $1 \le t$,no intersection of the line segment and the plane

Distance from a Point to a Plane

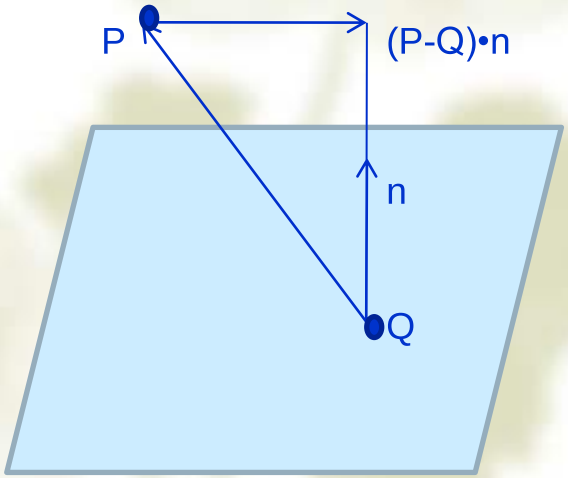
Plane Ax + By + Cz + D = 0 with normal vector(A, B, C)

Unit normal vector is n$(a$, b, c$)$

Any point *Q* is on the plane

So, the distance from any point *P* to the plane

is given by



$$D = |(P - Q) \bullet n|$$

$= a_1*A + a_2*B + a_3*C$

$= |P – Q|*|n|*\cos\theta$

$= |P-Q|*\cos\theta$

A Point inside a Triangle

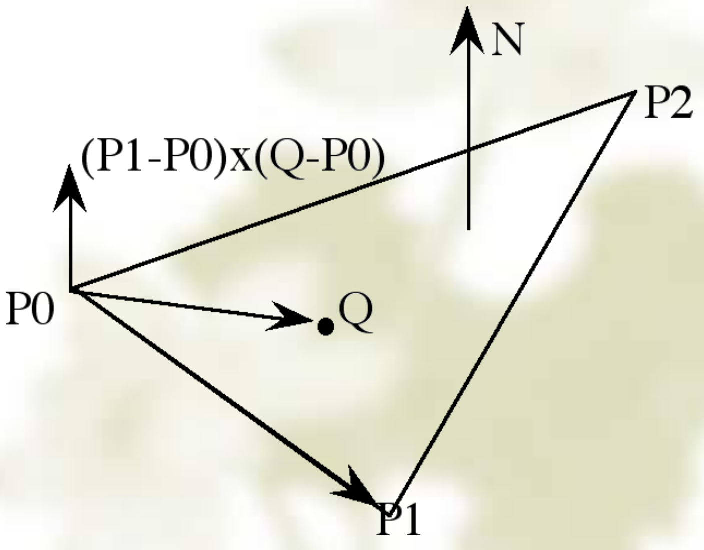
A point Q is inside a triangle $P_0P_1P_2$， iff

$$N\cdot((P1-P0)\times(Q-P0)) > 0$$

$$N\cdot((P2-P1)\times(Q-P1)) > 0$$

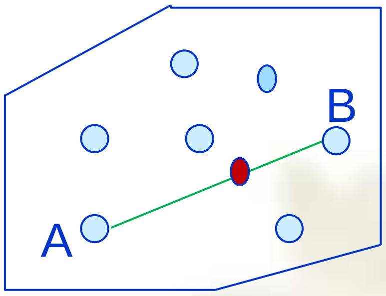
$$N\cdot((P0-P2)\times(Q-P2)) > 0$$

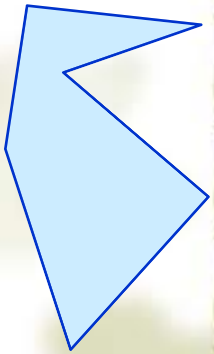
satisfy at the same time



Convexity

- The object is convex iff for any two points A and B in the object, all points between them is all in the object.



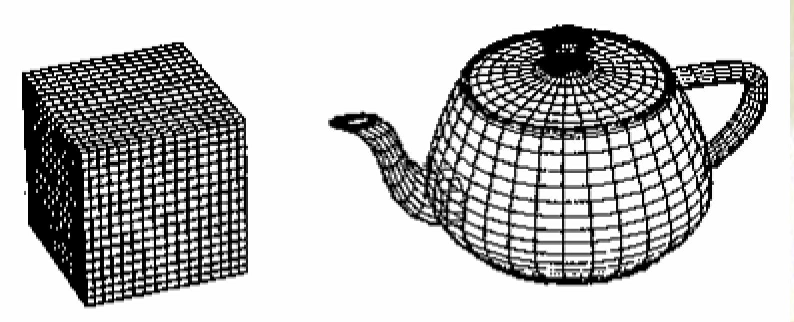


- Convex hull:

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_n P_n$$

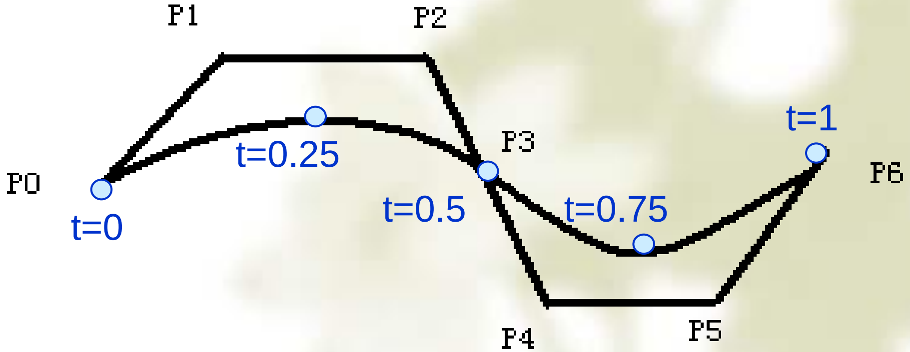
therein $$\sum_{i=1\ldots n} \alpha_i = 1$$

and $$\alpha_i \geq= 0, \quad i = 1,2,\ldots,n$$



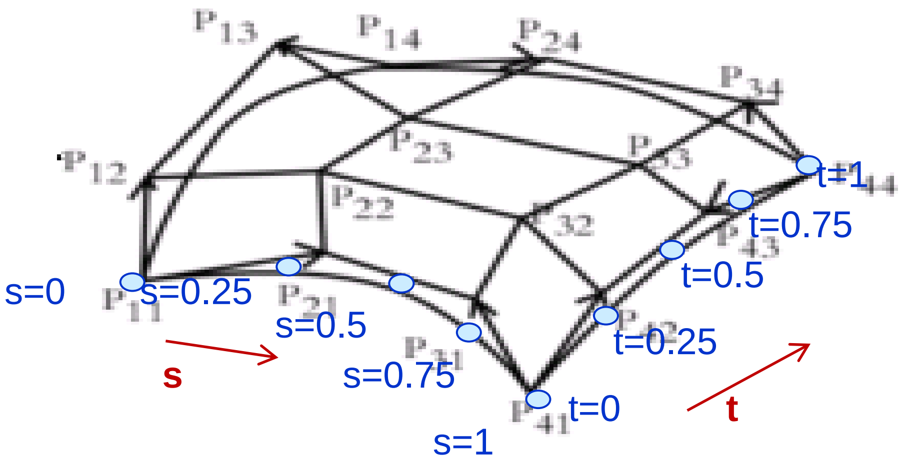
Parametric Curves

- A parametric curve is defined by three functions of one parameter $x(t)$, $y(t)$, $z(t)$ where the function is nonlinear
- For any value of t, the point $(x(t), y(t), z(t))$ is on the curve
- The parameter is often limited to t=[0,1]



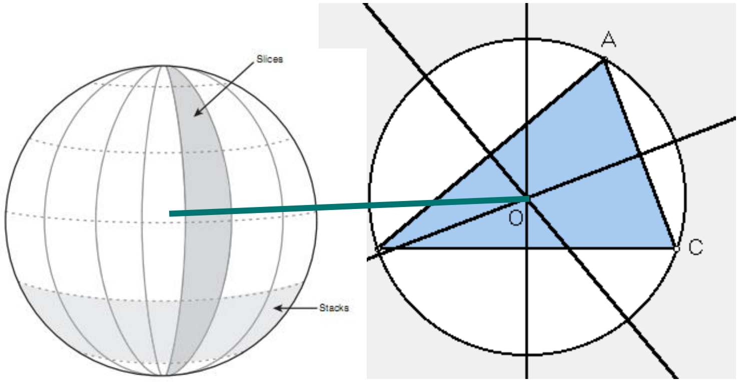
Parametric Surfaces

- A parametric surface is defined by three functions of two parameters $x(s,t)$, $y(s,t)$, $z(s,t)$ where the function is nonlinear
- For any values of the parameters (s, t), the point $(x(s,t),y(s,t),z(s,t))$ is on the surface
- The parameters are often limited to [0,1]

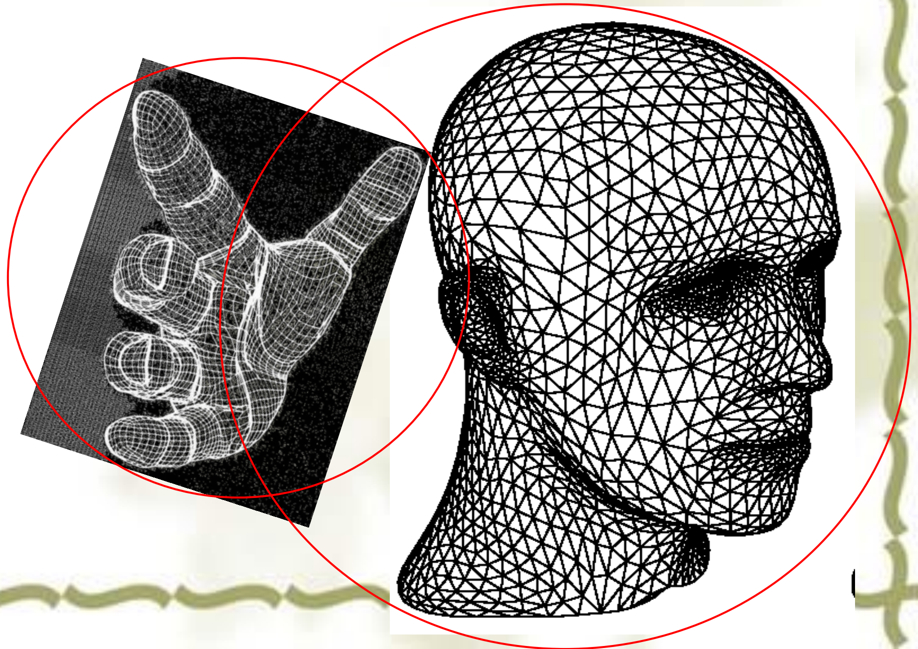


Triangle-Based Collision Detection

1. Detecting two objects possible collisions instead of actual collisions using bounding sphere or bounding box which contains the object
2. If having a possible collisions, we test whether or not a triangle on one object collides the bounding object
3. Testing if or not two triangles collide



circumcircle



Collision Detection Algorithm

1.比较两个包围球或包围盒：两个球心距离小于两个半径的和，

即$\sqrt{P1^2+P2^2} < r1+r2$，则可能碰撞；

2.可能碰撞时，则一个物体的每个三角形与另一个包围体比较，假设是包围球：

比较三角形的每个顶点：如果顶点与球心距离小于三角形最长边+球的半径，则可能碰撞；或 比较三角形外接圆（**circumcircle**）与包围球：如果外接圆心与球心距离 小于 外接圆半径+球的半径，则可能碰撞；

3.比较两个三角形是否碰撞：一个三角形的每个顶点 是否 在另一个三角形*平面*的一边,即$Ax_i+By_i+Cz_i+D>0$ (i=1,2,3) 或 $Ax_i+By_i+Cz_i+D<0$ (i=1,2,3)

如果在一边，则不会碰撞，否则可能碰撞（三角形与另一三角形相交）；

4.判断一个三角形是否与另一个三角形相交：

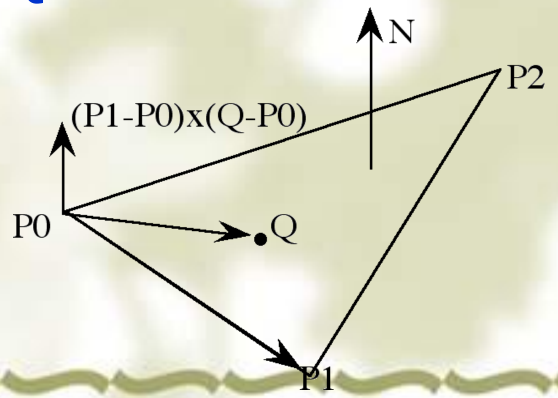
一个三角形的边Q_0Q_1与另一个三角形平面的交点Q是否 在另一个三角形$P_0P_1P_2$里面,即，若以下三个不等式同时成立：

$N\cdot((P1-P0)\times(Q-P0)) > 0$

$N\cdot((P2-P1)\times(Q-P1)) > 0$

$N\cdot((P0-P2)\times(Q-P2)) > 0$

则Q在三角形$P_0P_1P_2$里面

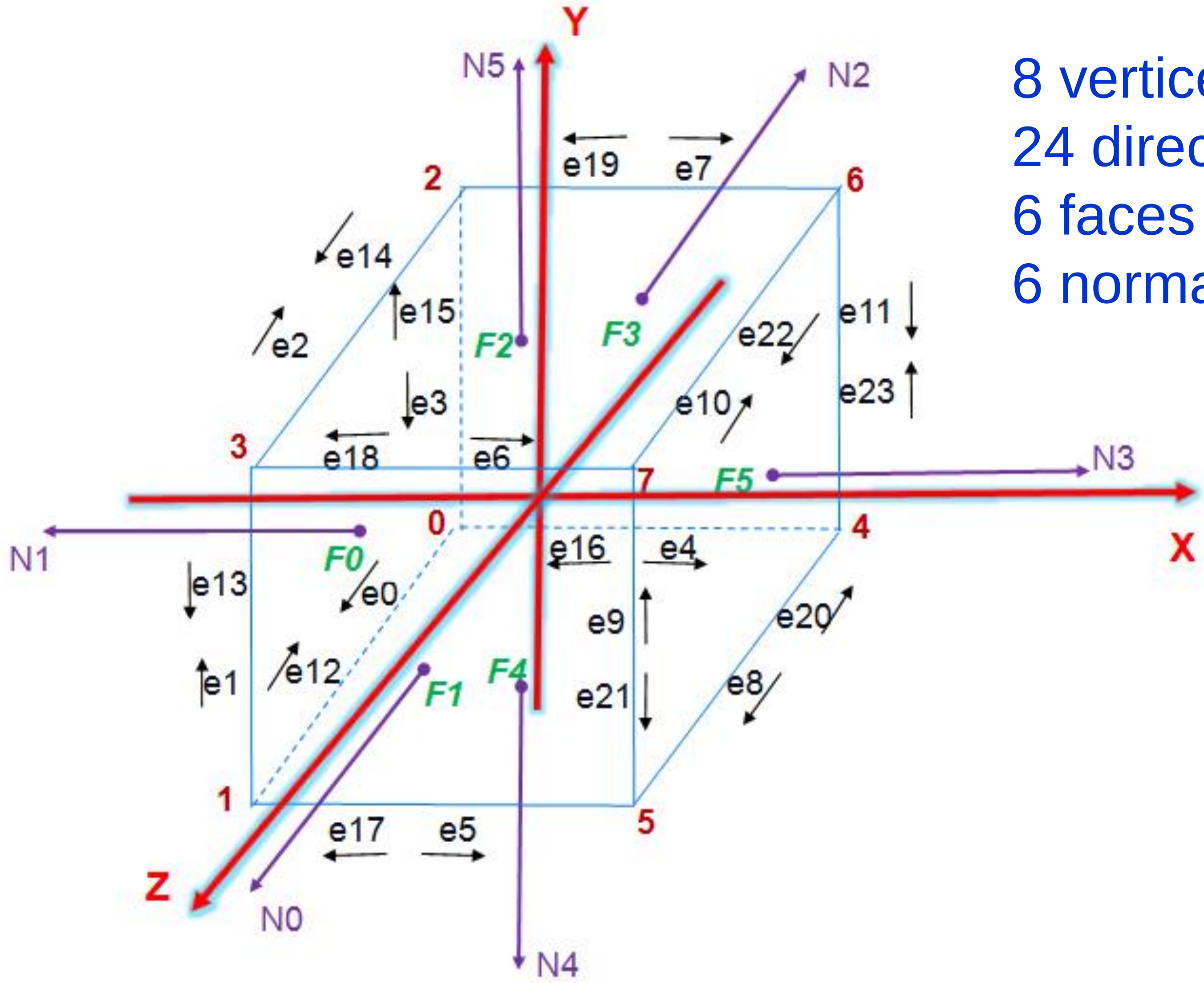


Quesion

How to represent a cube with geometry data?

Example: Cube model

OpenGL Vertex Arrays



8 vertices
24 directional edges
6 faces
6 normals

Example: Cube model

```
vec3   vertices[8]={{-1.0, -1.0, -1.0},     0
                    {-1.0, -1.0,  1.0},     1
                    {-1.0,  1.0, -1.0},     2
                    {-1.0,  1.0,  1.0},     3
                    { 1.0, -1.0, -1.0},     4
                    { 1.0, -1.0,  1.0},     5
                    { 1.0,  1.0, -1.0},     6
                    { 1.0,  1.0,  1.0} };   7
```

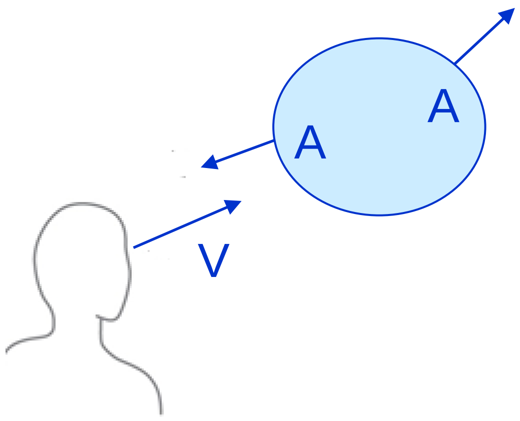
```
face cube[6]={{ 0, 1, 2, 3},        F0
              { 5, 9,  18, 13},     F1
              {14, 6, 10, 19},      F2
              { 7, 11, 16, 15},     F3
              { 4, 8, 17, 12},      F4
              { 22,21,20,23} }      F5
```

```
edge edges[24]={{ 0, 1}, { 1, 3},       e0, e1
                { 3, 2}, { 2, 0},       e2, e3
                { 0, 4}, { 1, 5},       e4, e5
                { 3, 7}, { 2, 6},       e6, e7
                { 4, 5}, { 5, 7},       e8, e9
                { 7, 6}, { 6, 4},       e10, e11
                { 1, 0}, { 3, 1},       e12, e13
                { 2, 3}, { 0, 2},       e14, e15
                { 4, 0}, { 5, 1},       e16, e17
                { 7, 3}, { 6, 2},       e18, e19
                { 5, 4}, { 7, 5},       e20, e21
                { 6, 7}, { 4, 6} };     e22, e23
```

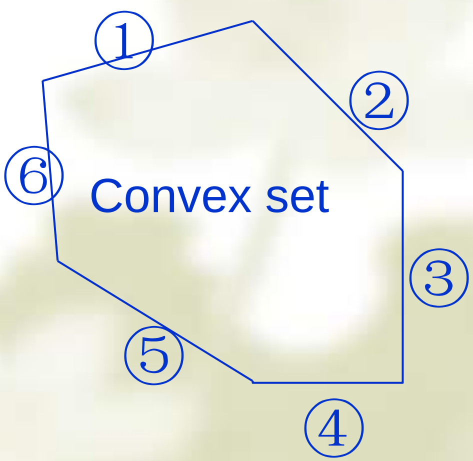
```
vec3   normals[6]={{ 0.0, 0.0, 1.0},    N0
                   {-1.0, 0.0, 0.0},    N1
                   { 0.0, 0.0,-1.0},    N2
                   { 1.0, 0.0, 0.0},    N3
                   { 0.0,-1.0, 0.0},    N4
                   { 0.0, 1.0, 0.0} };  N5
```

作业3

1. 给定视线方向V和物体A点的法向量，如何判断A点是否能被看到？
2. 如何判断一个多面体是一个凸多面体？



第1题



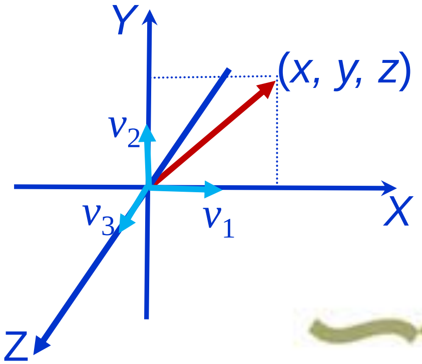
第2题

Transformations

1. Coordinate Systems
2. Space, Matrix and operations
3. Affine Transformations(3x3)
4. Homogeneous Coordinates Transformations(4x4)
5. Inverse Transformations
6. Composite Transformations
7. Reflection and Shear Transformations
8. Projection Transformations: Orthographic and Perspective

Cartesian Coordinate System

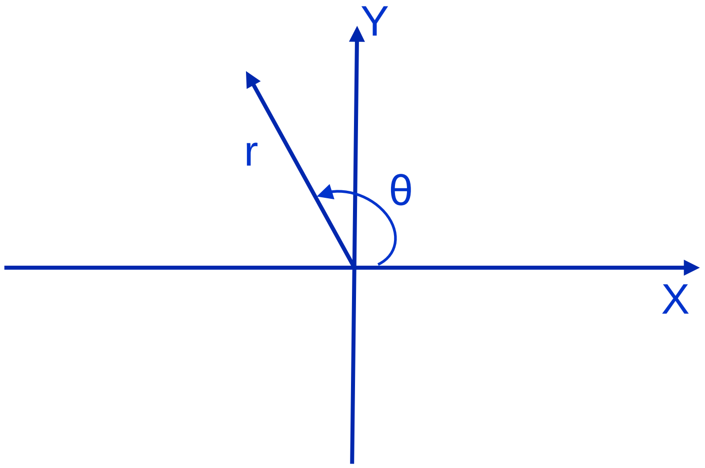
- Affine Space as Point and *n*-dimension
- If v vector is in a 3-dimensional space, v is written $v=\alpha_1 v_1+\alpha_2 v_2+\alpha_3 v_3$
- Cartesian coordinates are a very special kind of affine space that correspond to the case where $v_1=(1,0,0), v_2=(0,1,0), v_3=(0,0,1)$
- v_1, v_2, v_3 as the unit vectors of the x-axis, the y-axis, and the z-axis, respectively



Polar coordinate system

- The polar coordinates r (the radial coordinate) and α (the polar angle)
- Related to the Cartesian coordinates by

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \qquad \begin{matrix} 0 \leqslant r < \infty \\ 0 \leqslant \theta < 2\pi \end{matrix}$$



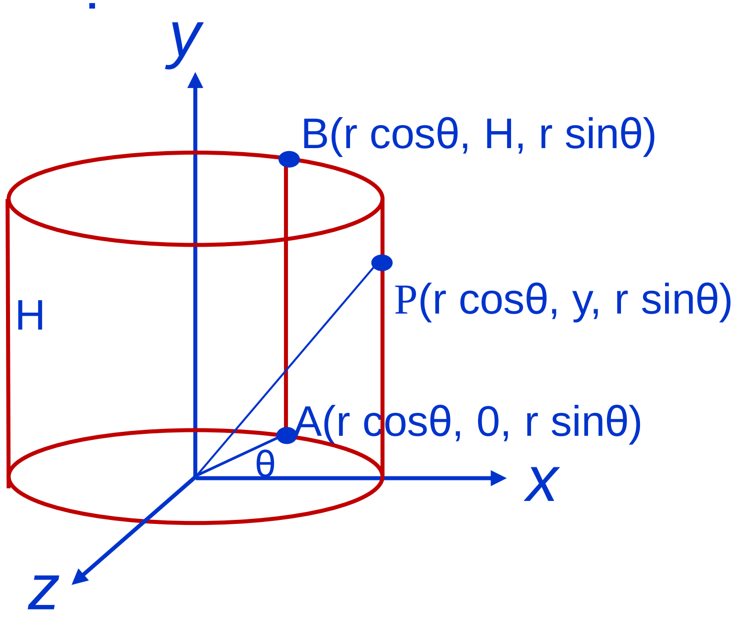
Cylindrical coordinate system

2-dimensional polar coordinates to three dimensions by superposing a height (y) axis

Related to the Cartesian coordinates by :

$$\begin{cases} x = r\cos\theta \\ y = y \\ z = r\sin\theta \end{cases}$$

$0 \leqslant r < \infty$
$0 \leqslant \theta < 2\pi$
$-\infty < y < \infty$

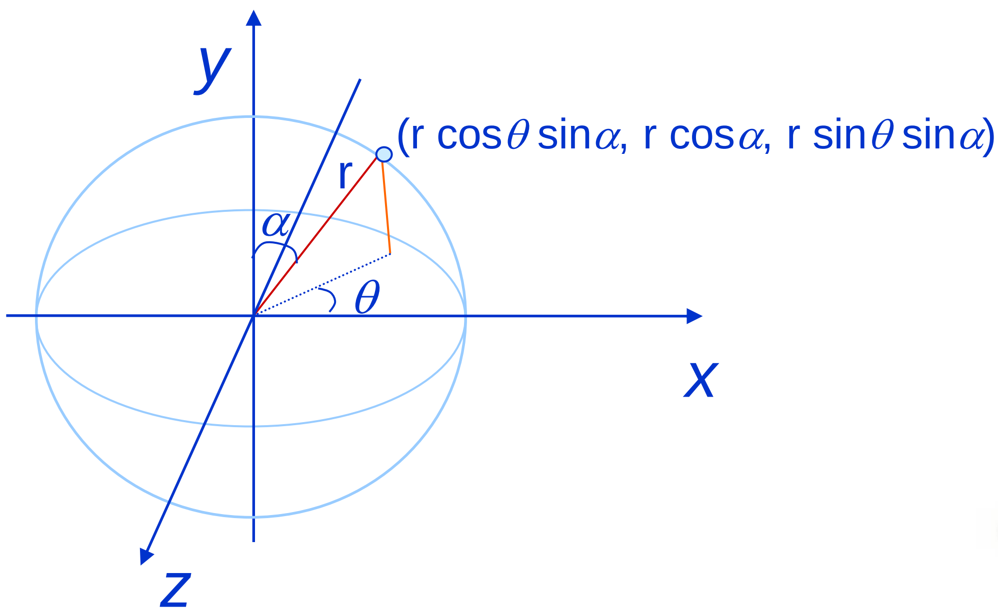


Spherical coordinate system

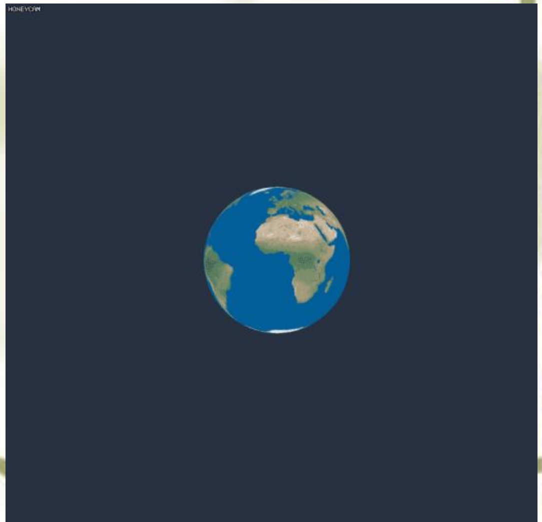
Spherical coordinates are related to the Cartesian coordinates by

$$\begin{cases} x = r\cos\theta\sin\alpha \\ y = r\cos\alpha \\ z = r\sin\theta\sin\alpha \end{cases}$$

$$0 \leqslant r < \infty$$
$$0 \leqslant \theta < 2\pi$$
$$0 \leqslant \alpha \leqslant \pi$$

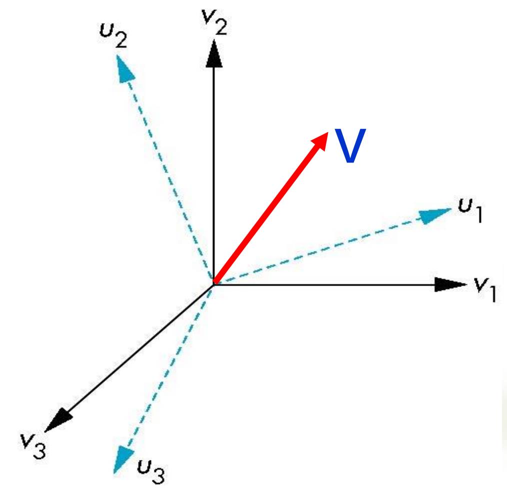


墨卡托（Mercator）投影：
地球投影到柱平面



Change of Coordinate Systems

- Consider two representations of the same vector v with respect to two different bases
- The representations are



$$\mathbf{a}=[\alpha_1\ \alpha_2\ \ \alpha_3\,]$$

$$\mathbf{b}=[\beta_1\ \ \beta_2\ \ \beta_3\,]$$

where

$$\mathrm{v}=\alpha_1 v_1+\alpha_2 v_2+\alpha_3 v_3=[\alpha_1\ \alpha_2\ \alpha_3]\,[v_1\ v_2\ v_3]^T$$

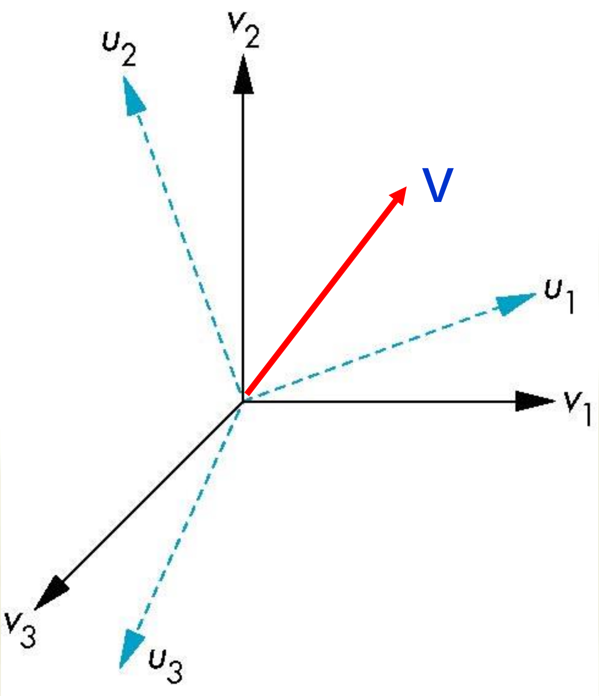
$$=\beta_1 u_1+\beta_2 u_2+\beta_3 u_3=[\beta_1\ \beta_2\ \beta_3]\,[u_1\ u_2\ u_3]^T$$

线性代数表示！

Two bases by Matrix Form

Each of the basis vectors u1,u2, u3, are vectors that can be represented in terms of another basis vectors v1,v2,v3

$$u_1 = \gamma_{11}v_1+\gamma_{12}v_2+\gamma_{13}v_3$$
$$u_2 = \gamma_{21}v_1+\gamma_{22}v_2+\gamma_{23}v_3$$
$$u_3 = \gamma_{31}v_1+\gamma_{32}v_2+\gamma_{33}v_3$$



The coefficients define a 3 x 3 matrix

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

Matrix Operations

$$\begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix}$$

$u_1 = \gamma_{11}v_1+\gamma_{12}v_2+\gamma_{13}v_3$
$u_2 = \gamma_{21}v_1+\gamma_{22}v_2+\gamma_{23}v_3$
$u_3 = \gamma_{31}v_1+\gamma_{32}v_2+\gamma_{33}v_3$

1xn row matrices or nx1 column matrices

$v = [a1, a2, a3]$ $\quad$ $p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

The transpose of v and p is the column matrix

$v^T = \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix}$ $\quad$ $p^T = [x, y, z]$

Matrix Operations

Let Matrices $A=(a_{ij})_{m\times s}$, $B=(b_{ij})_{s\times n}$, the product C of A and B is

$$C=A_{m\times s}\,B_{s\times n}=(c_{ij})_{m\times n}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Longrightarrow$$

$p' = M\,p$ √

$p'^{T} = p^{T} M^{T}$ X

Addition

commutative law: A+B = B+A

associative law: (A+B)+C = A+(B+C)

Multiplication

NO commutative law : AB ≠ BA

associative law: (AB)C = A(BC)

distributive law： A(B+C) = AB+AC

Inverse Matrix

- A square matrix A is called nonsingular if $\det A \neq 0$
- If A is nonsingular, there is an existence of n x n matrix A^{-1}, which is called the inverse matrix A such that it satisfies the property: $AA^{-1} = A^{-1}A = I$, where I is the Identity matrix

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

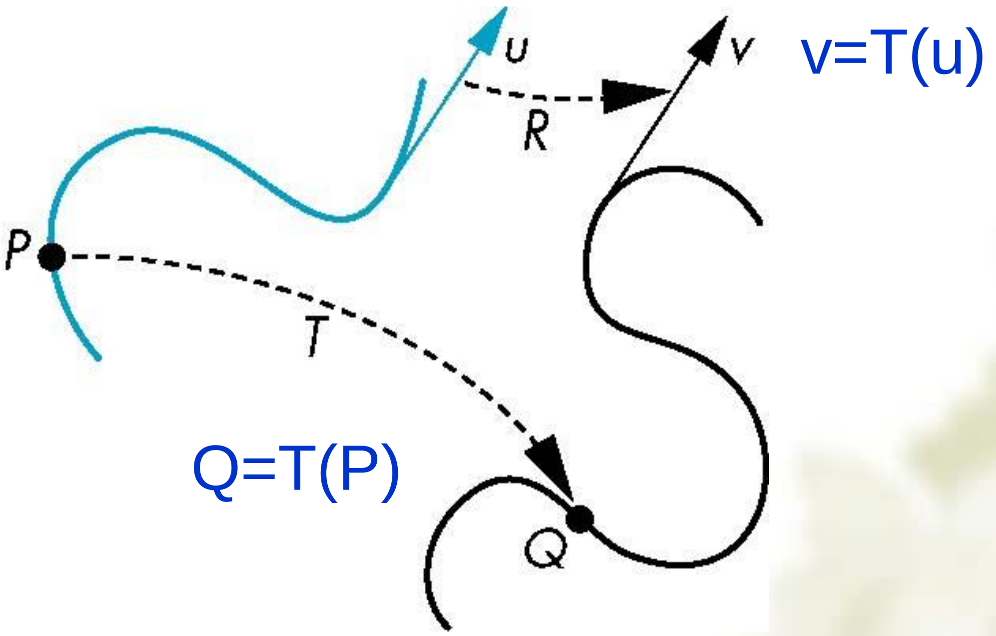
If $A_{nxn}B_{nxn} = B_{nxn}A_{nxn} = I_n$

Then, A and B are the inverses of each other

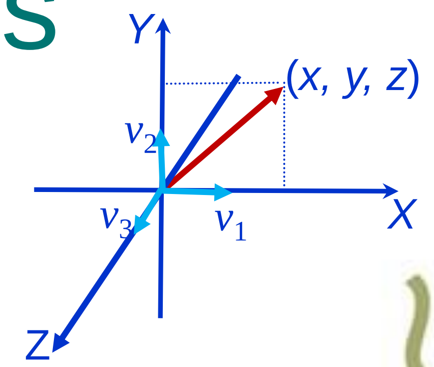
$A = B^{-1}$ or $B = A^{-1}$

General Transformations

A transformation maps points to other points and/or vectors to other vectors



Affine Transformations



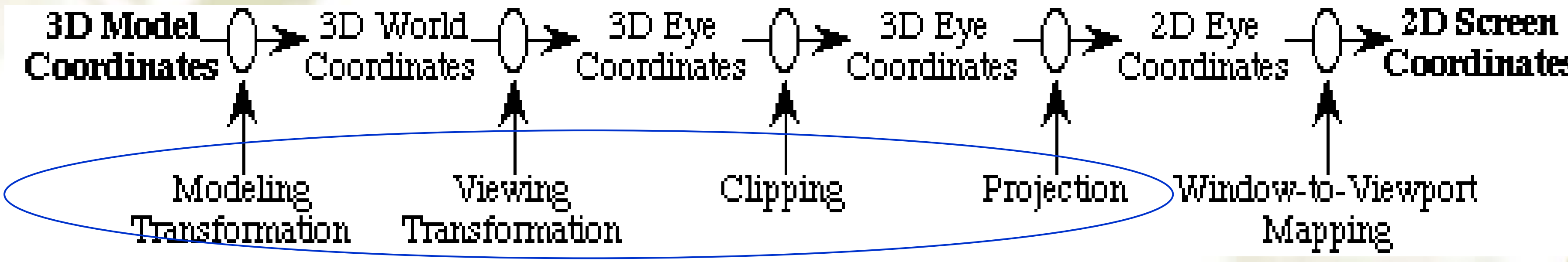
- Transformations in Affine Space
- Characteristic of many physically
  - Line preserving--only transform endpoints of line segments
  - Rigid body transformations: rotation, translation, reflection
  - Scaling, Shear--ratios of distances preserving
- Transformed vertex p' can get from p with transformation matrix M:

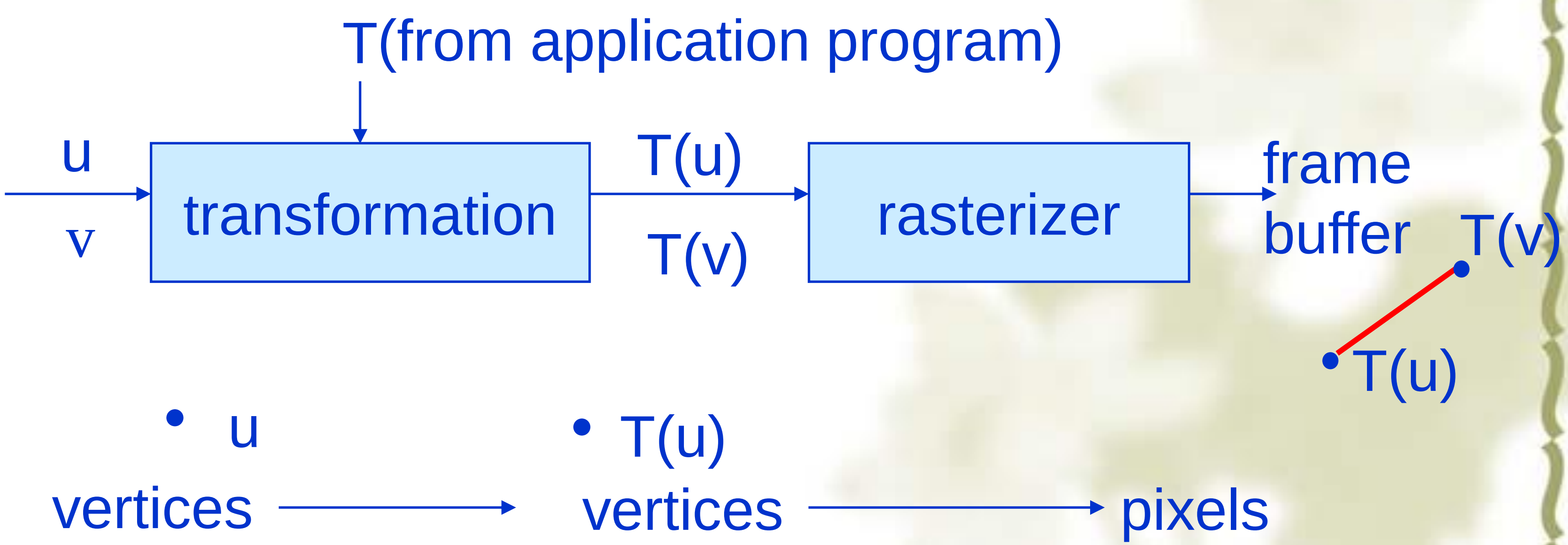
$$p' = M\, p$$

- Affine transformations: translation, scaling, rotation, reflection, shear

Pipeline Implementation

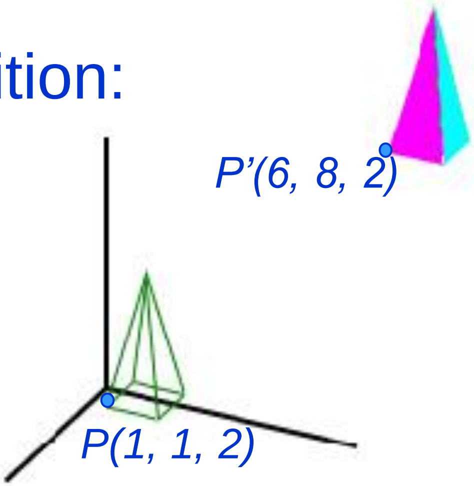
Graphics geometry pipeline





Translation Transformations

- To move a vertex of an object to a new position:
  A vertex of an object is $P(1, 1, 2)$
  The distances of translation $d(d_x, d_y, d_z)$ are
  $d_x = 5$
  $d_y = 7$
  $d_z = 0$



- In general, $x' = x + d_x, \quad y' = y + d_y, \quad z' = z + d_z$
- *So,*

$$x' = x + d_x = 1 + 5 = 6$$
$$y' = y + d_y = 1 + 7 = 8$$
$$z' = z + d_z = 2 + 0 = 2$$

- A new position of the vertex is $P'(6, 8, 2)$

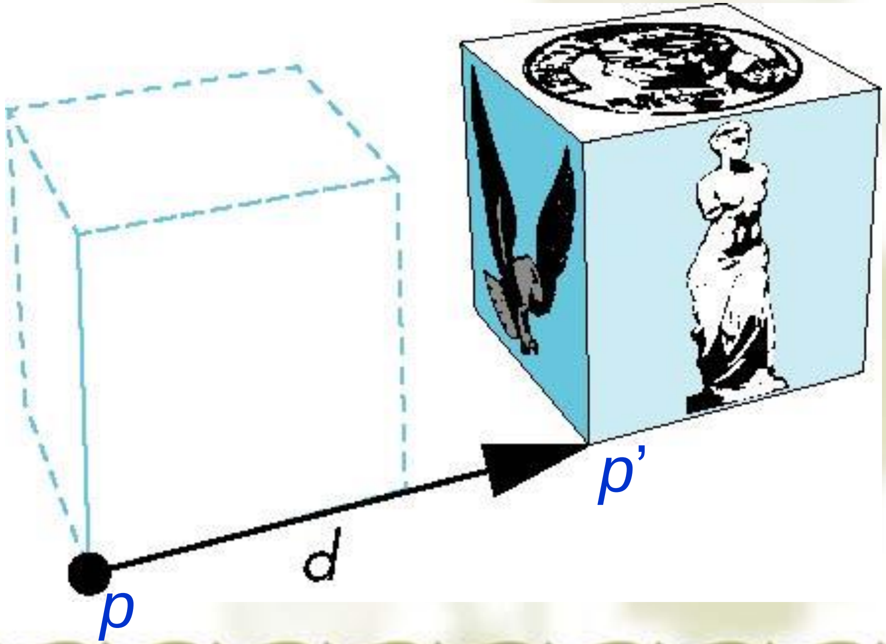
Translation Transformations

- Translate (move, displace) a point to a new location

- Displacement determined by a vector d
  $p' = p + d$

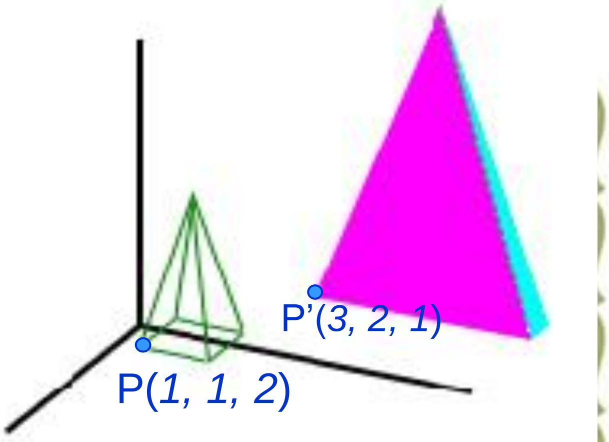
Or
$$\begin{cases} x' = x + d_x \\ y' = y + d_y \\ z' = z + d_z \end{cases}$$

$$d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$



Scaling Transformations

- To scale an object **relative to the origin** in **axis direction** :
  A vertex of an object is *P*(1, 1, 2)
  The scaling values are
  $s_x = 3$
  $s_y = 2$
  $s_z = 1/2$



- In general, $x' = x * s_x$, $y' = y * s_y$, $z' = z * s_z$
- *So,*

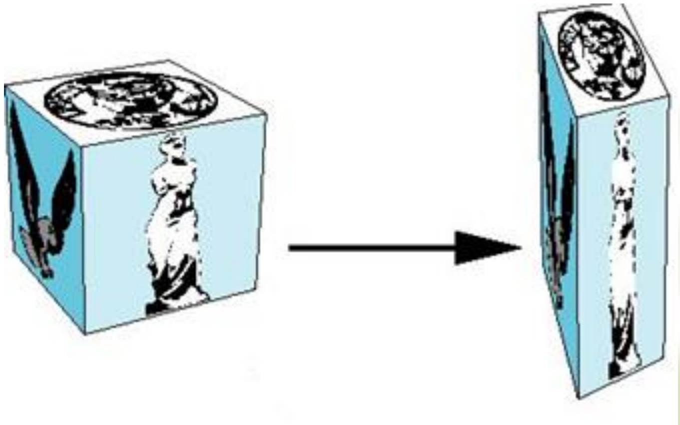
$$x' = x * s_x = 1 * 3 = 3$$
$$y' = y * s_y = 1 * 2 = 2$$
$$z' = z * s_z = 2 * 1/2 = 1$$

- A new position of the vertex is P'(3, 2, 1)

Scaling Transformations

- Expand or contract along each axis (fixed point of origin)
- Displacement determined by a matrix *S*

$$p' = S\, p$$



Or

$$\begin{cases} x' = s_x x \\ y' = s_y x \\ z' = s_z x \end{cases}$$

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

Rotation Transformations(2D)

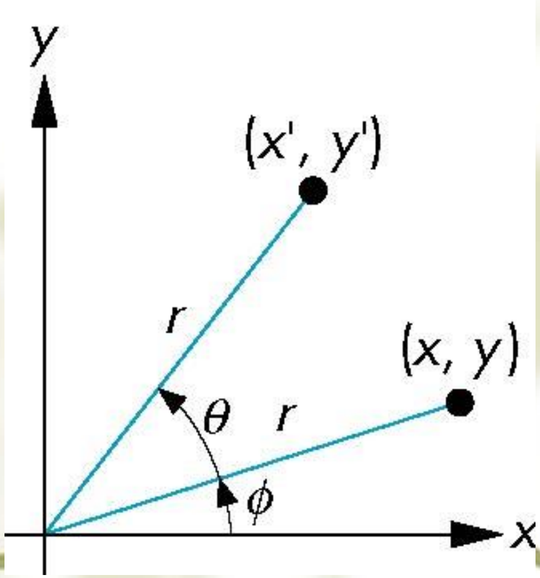
- To rotate an object **with respect to z-axis** **relative to the origin** in **counterclockwise**:
  A vertex of an object is *P*(3, 1, 0)
  The rotation angle is $\theta = 30^0$

- In general, $x' = x^*cos\theta - y^*sin\theta$, $y' = x^* sin\theta + y^*cos\theta$, $z' = z$
- *So,*

  $x' = x^*cos\theta - y^*sin\theta = 2.1$

  $y' = x^*sin\theta + y^*cos\theta = 2.4$

  $z' = z = 0$



- A new position of the vertex is *P'*(2.1, 2.4, 0)

Rotation Transformations(2D)

To rotate a point with respect to z-axis relative to the origin in counterclockwise

$\because \quad x = r \cos\phi$

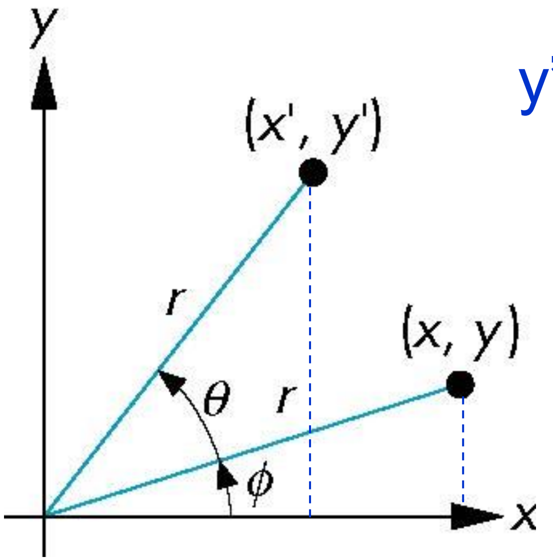
$\quad\quad y = r \sin\phi$

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

$\quad\quad = x*\cos\theta - y*\sin\theta$

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

$\quad\quad = x*\sin\theta + y*\cos\theta$



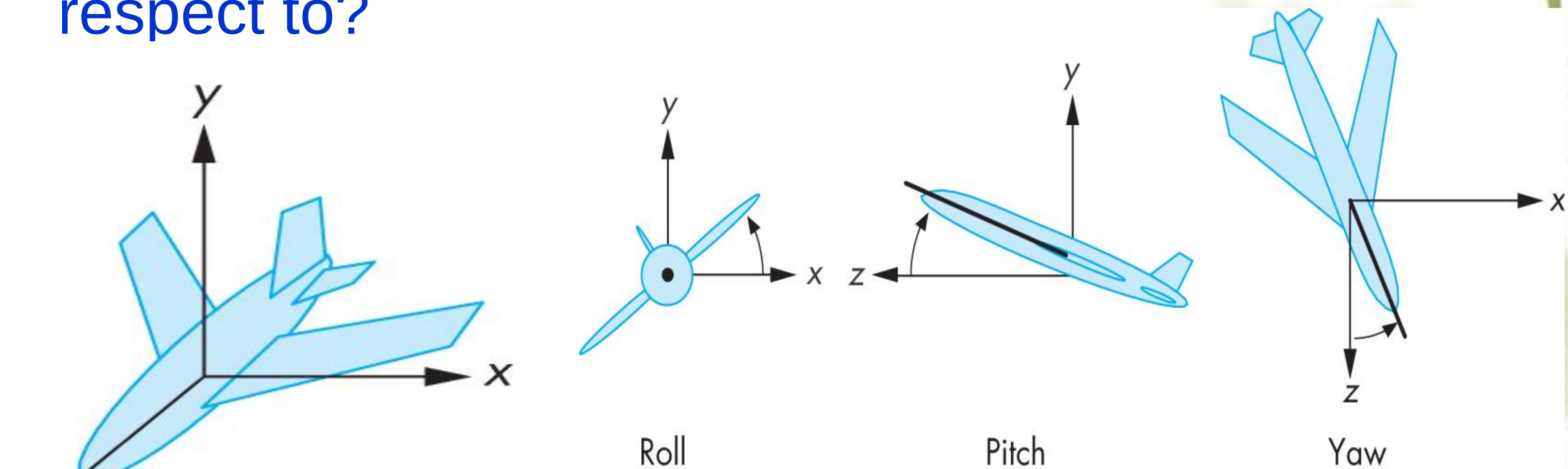
Displacement determined by a matrix *R*

$$p' = R_z(\theta)\, p$$

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

Question1:

The position of an airplane is specified by the roll, pitch, yaw. When the airplane is flying with each operation, which axis is the plane rotation with respect to?



Roll	Pitch	Yaw
z-axis	x-axis	y-axis
滚转角	俯仰角	偏航角

Three Basic Transformations

The basic linear transformations are:

- translation: $p' = p + d$, where d is translation vector
- scaling: $p' = S*p$, where S is a scaling matrix
- rotation: $p' = R*p$, where R is a rotation matrix

$$d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \qquad S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \qquad R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite Transformations

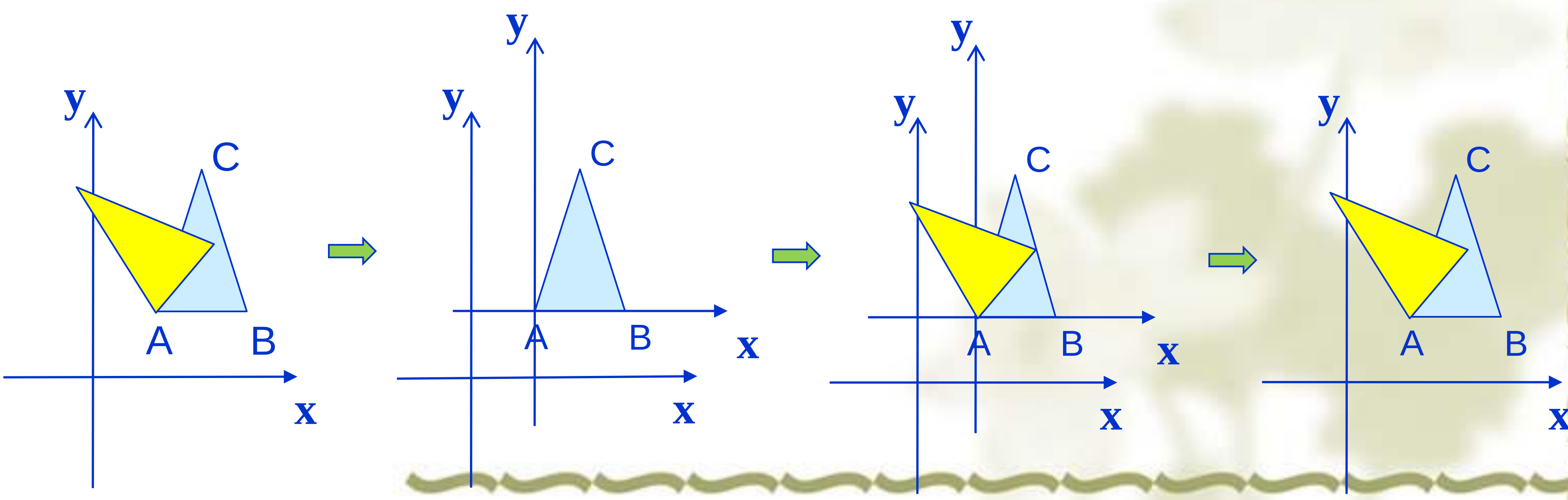
Rotation with respect to a fixed position A(a1,a2,a3)

1. translate fixed point to origin d(-a1,-a2,-a3) = d_1
2. Rotation $R_z(\theta)$
3. translate fixed point back to its starting position d(a1,a2,a3) = d_2

So: $p' = R\,(p + d_1) + d_2$

?

$$p' = (D_2\,(R\,(D_1\,p))) = (D_2\,R\,D_1\,)p \quad = Mp$$



Homogeneous Coordinates

We use homogeneous coordinates in order to express all transformations as matrices and allow them to be combined easily:

$$p' = (M_3\,(M_2\,(M_1\,p))) = (M_3\,M_2\,M_1)\,p = M\,p$$

We can express translation using a 4 x 4 matrix $\mathbf{T}$ in homogeneous coordinates

$p' = \mathbf{T}p$

Homogeneous Coordinates

$$(x, y, w) \Leftrightarrow \left(\frac{x}{w}, \frac{y}{w}\right)$$

Homogeneous Cartesian

A point in homogeneous coordinates (x, y, w), $w \neq 0$, corresponds to the 2-D vertex (x/w, y/w) in Cartesian coordinates

$$(1,2,3) \Rightarrow \left(\frac{1}{3}, \frac{2}{3}\right)$$

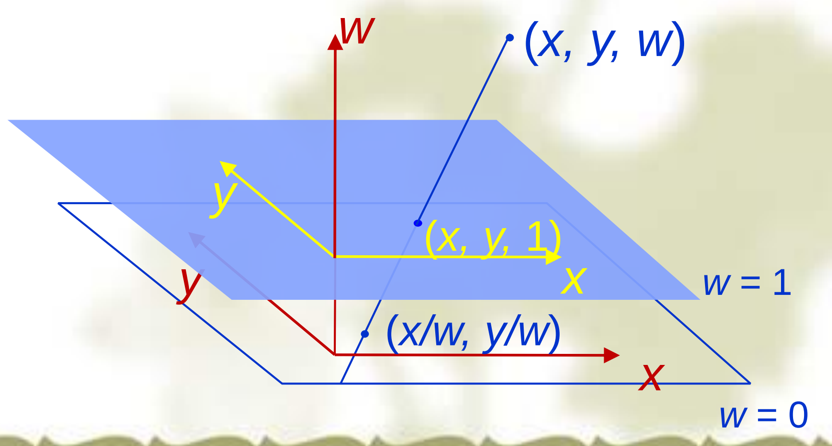
$$(2,4,6) \Rightarrow \left(\frac{2}{6}, \frac{4}{6}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$$

$$(4,8,12) \Rightarrow \left(\frac{4}{12}, \frac{8}{12}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$$

$$\vdots \qquad \vdots$$

$$(1a,2a,3a) \Rightarrow \left(\frac{1a}{3a}, \frac{2a}{3a}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$$

The points (1a, 2a, 3a) are homogeneous, because they correspond to the point (1/3, 2/3)



Homogeneous Coordinates and Computer Graphics

Homogeneous coordinates are key to all computer graphics systems

- ✓ All standard transformations (rotation, translation, scaling) can be implemented with matrix multiplications using 4 x 4 matrices
- ✓ Hardware pipeline works with 4 dimensional representations
- ✓ For orthographic viewing, we can maintain $w=0$ for vectors and $w=1$ for points
- ✓ For perspective we need a *perspective division*

Representing One Space in Terms of the Other

Extending what we did with change of bases

$$
\begin{aligned}
u_1 &= \gamma_{11}v_1+\gamma_{12}v_2+\gamma_{13}v_3 \\
u_2 &= \gamma_{21}v_1+\gamma_{22}v_2+\gamma_{23}v_3 \\
u_3 &= \gamma_{31}v_1+\gamma_{32}v_2+\gamma_{33}v_3 \\
Q_0 &= \gamma_{41}v_1+\gamma_{42}v_2+\gamma_{43}v_3 +P_0
\end{aligned}
$$

defining a 4 x 4 matrix

$$
\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}
$$

Homogeneous Coordinates Representations

Within the two spaces any point or vector has a representation of the same form

$\mathbf{a}=[\alpha_1\ \alpha_2\ \ \alpha_3\ \alpha_4\ ]^T$ in the first space
$\mathbf{b}=[\beta_1\ \beta_2\ \ \beta_3\ \beta_4\ ]^T$ in the second space

where $\boldsymbol{\alpha_4=\beta_4=1}$ **for points** and $\boldsymbol{\alpha_4=\beta_4=0}$ **for vectors** and

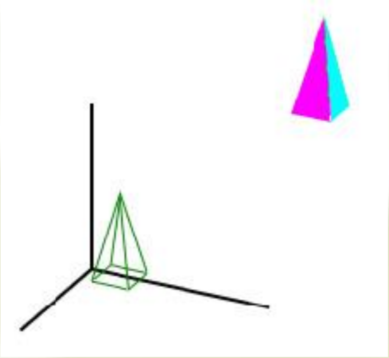
$$a = Mb$$

The matrix $\mathbf{M}$ is 4 x 4 and specifies an affine transformation in homogeneous coordinates

Translation on Homogeneous Coordinates

- Translation $p(x,y,z)^T$ by a distance $\mathbf{T}(d_x, d_y, d_z)^T$
- We express p as $(x, y, z, 1)^T$ and form a translation matrix T in homogeneous coordinates
- The translated point is p'

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + dx \\ y + d_y \\ z + d_z \\ 1 \end{bmatrix}$$

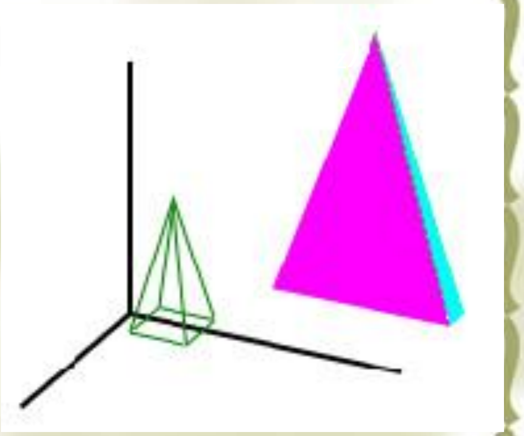


$p' = T\,p$

Scaling on Homogeneous Coordinates

- Scaling by s_x, s_y, s_z relative to the origin in axis direction
- Scaling matrix in homogeneous coordinates is S

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \\ 1 \end{bmatrix}$$



$p' = S\,p$

Rotation on Homogeneous Coordinates (2D)

- Rotation is specified with respect to an axis - easiest to start with coordinate axes
- To rotate about the z-axis, matrix in homogeneous coordinates is $R_z(\theta)$

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

$p' = R_z(\theta)\, p$

Inverses Matrices

Inverse matrices of transformation by simple geometric observations

- Translation: $\mathbf{T}^{-1}(d_x, d_y, d_z) = \mathbf{T}(-d_x, -d_y, -d_z)$
- Scaling: $\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$
- Rotation: $\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta)$

  Note that since $\cos(-\theta) = \cos(\theta)$ and $\sin(-\theta) = -\sin(\theta)$

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

Translation Inverse Matrix

$\mathbf{T}^{-1}(d_x, d_y, d_z) = \mathbf{T}(-d_x, -d_y, -d_z)$

if'

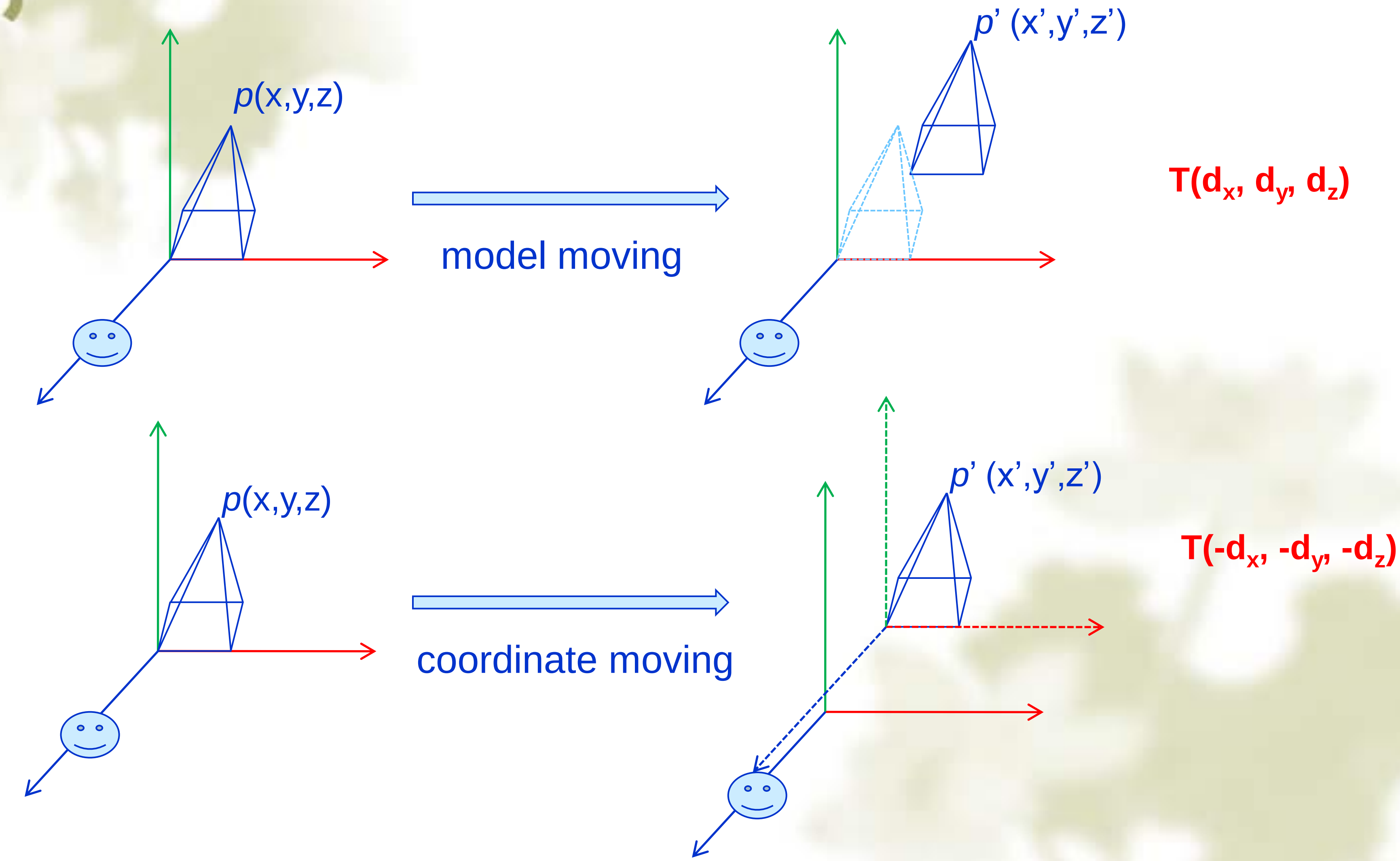
$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

then'

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\because \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation Inverse transformation



Model moving and view moving are the same, but inverse each other

Scaling Inverse Matrix

$$\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$$

if

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

then

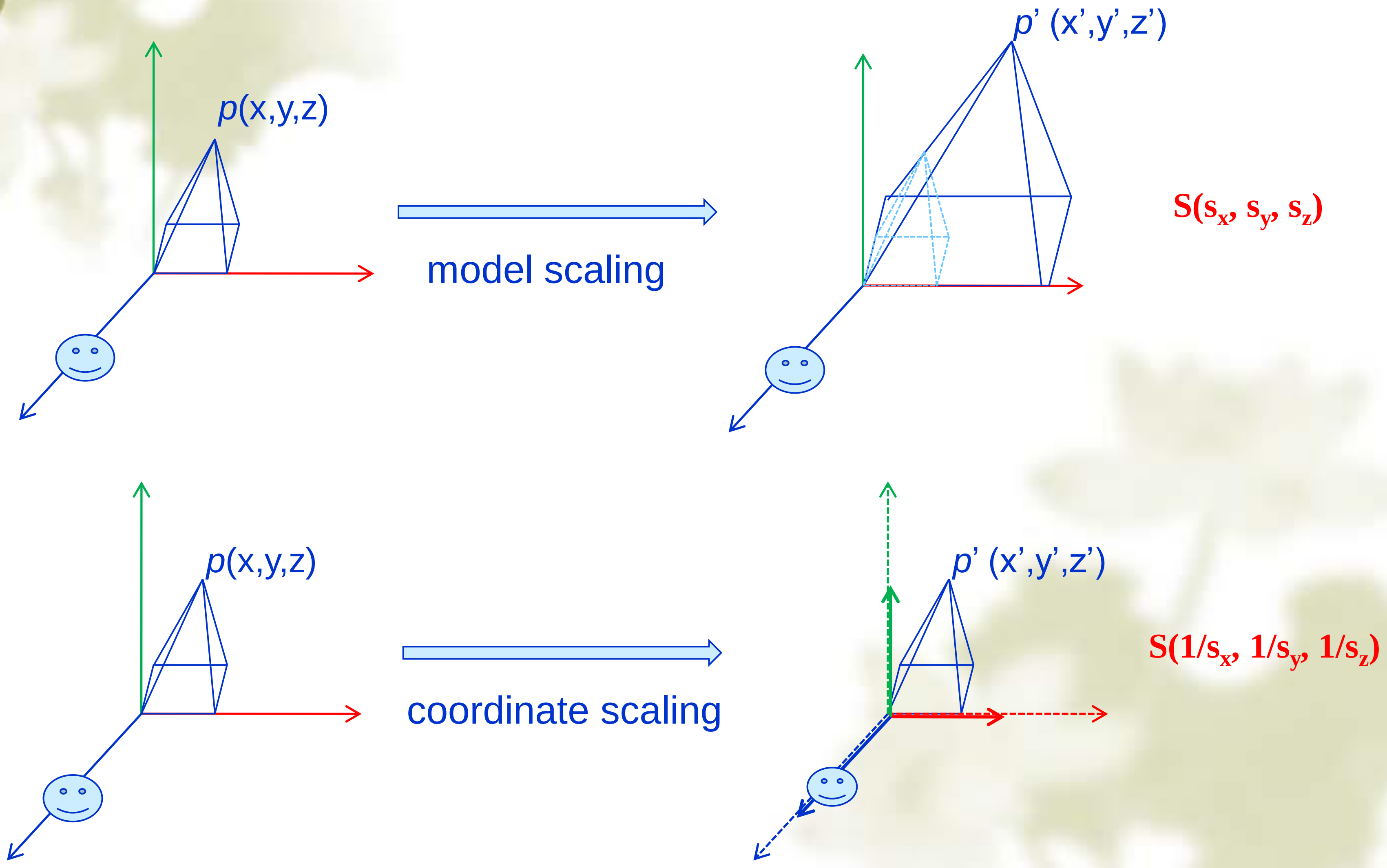
$$S^{-1} = \begin{bmatrix} 1/S_x & 0 & 0 & 0 \\ 0 & 1/S_y & 0 & 0 \\ 0 & 0 & 1/S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\because$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/S_x & 0 & 0 & 0 \\ 0 & 1/S_y & 0 & 0 \\ 0 & 0 & 1/S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/S_x & 0 & 0 & 0 \\ 0 & 1/S_y & 0 & 0 \\ 0 & 0 & 1/S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling Inverse transformation



Rotation Inverse Matrix (2D)

$$\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta)\,,\quad \cos(-\theta) = \cos(\theta) \;\text{ and }\; \sin(-\theta) = -\sin(\theta)$$

$$\sin^2(\theta) + \cos^2(\theta) = 1$$

if

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

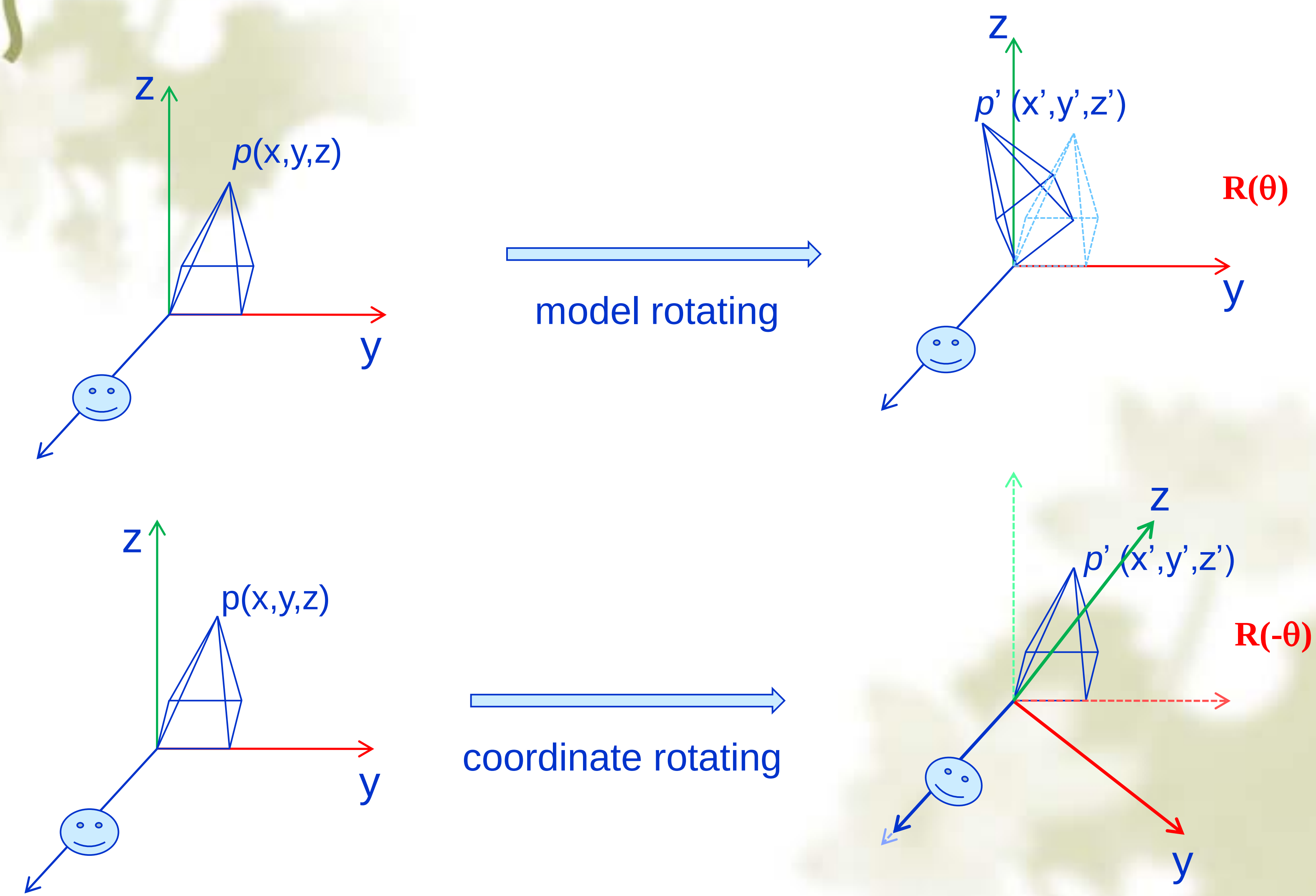
then

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

$$\because \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Rotation Inverse transformation(2D)

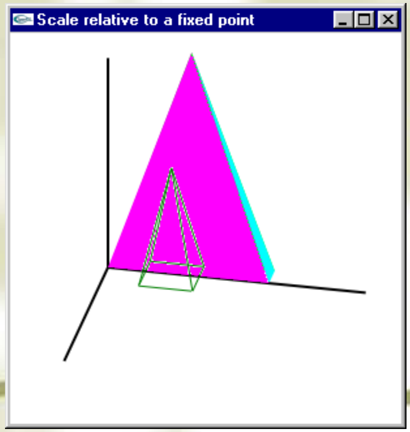


Composite Transformations

- The attraction of homogeneous coordinates is that a sequence of transformations may be encapsulated as a single matrix by associative law, called **concatenation**
- First example, *scaling with respect to a fixed position (a1,a2,a3)*
  1. translate fixed point to origin T(-a1,-a2,-a3) = T_1
  2. Scale S
  3. translate fixed point back to its starting position T(a1,a2,a3)=T_2
- Thus: $p' = (T_2\,(S\,(T_1\,p)))$

$$= (T_2\,S\,T_1\,)p$$

$$M = T_2\,S\,T_1$$

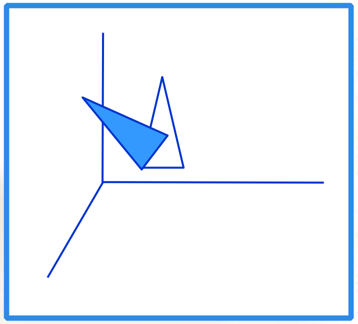


Composite Transformations

- Second example, ***rotation with respect to a fixed position (a1,a2,a3)***
  1. translate fixed point to origin T(-a1,-a2,-a3) = T_1
  2. Rotation R
  3. translate fixed point back to its starting position T(a1,a2,a3) = T_2
- Thus: $p' = (T_2\,(R\,(T_1\,p)))$

  $= (T_2\,R\,T_1)\,p$

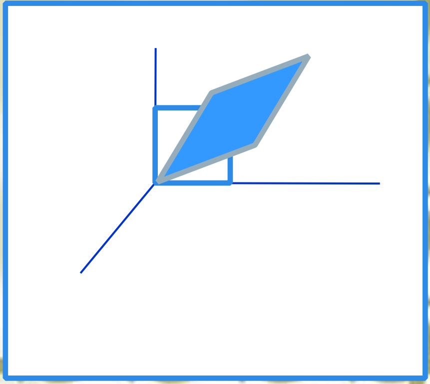
  $M = T_2\,R\,T_1$



- Third example, ***Scaling with respect to a fixed direction (ɵ)***
  1. rotation a fixed angle counterclockwise $R_z(\theta) = R_1$
  2. scaling S
  3. rotation a fixed angle clockwise $R_z(-\theta) = R_2$
- Thus: $p' = (R_2\,(S\,(R_1\,p)))$

  $= (R_2\,S\,R_1)\,p$

  $M = R_2\,S\,R_1$

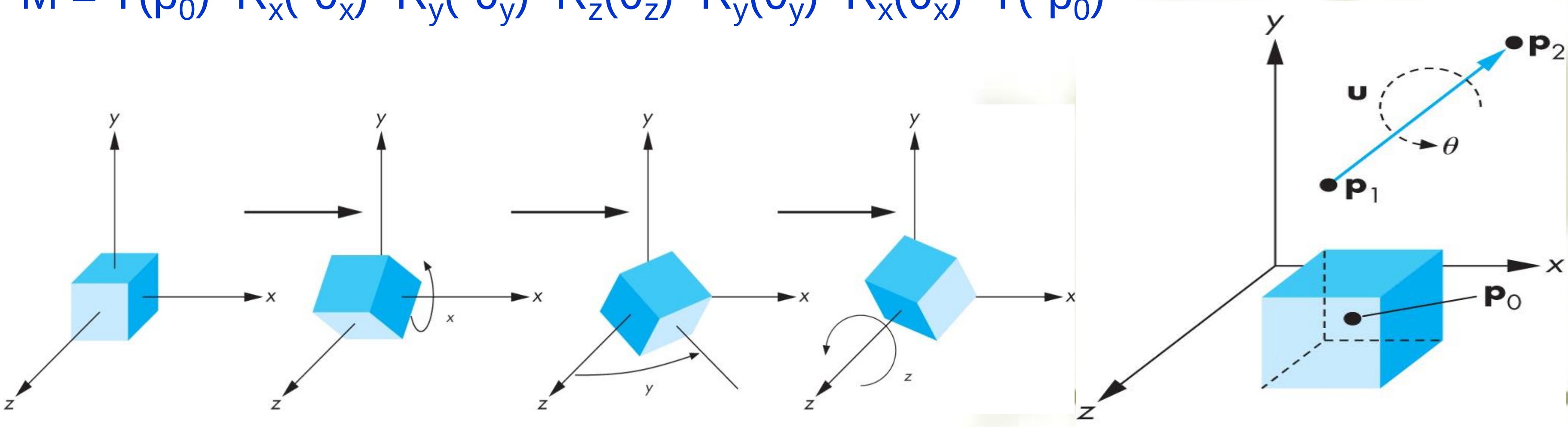


Composite Transformations

- Forth example, ***Rotation with respect to an arbitrary line***
  1. translate fixed point to origin $T(-a1,-a2,-a3) = T(-p_0)$
  2. rotation a fixed angle counterclockwise about X-axis $R_x(\theta_x)$
  3. rotation a fixed angle counterclockwise about Y-axis $R_y(\theta_y)$
  4. rotation a fixed angle counterclockwise about Z-axis $R_z(\theta_z)$
  5. rotation a fixed angle clockwise back about Y-axis $R_y(-\theta_y)$
  6. rotation a fixed angle clockwise back about X-axis $R_x(-\theta_x)$
  7. translate fixed point back to its starting position $T(a1,a2,a3) = T(p_0)$

Thus: $p' = T(p_0)\ R_x(-\theta_x)\ R_y(-\theta_y)\ R_z(\theta_z)\ R_y(\theta_y)\ R_x(\theta_x)\ T(-p_0)\ p$

$M = T(p_0)\ R_x(-\theta_x)\ R_y(-\theta_y)\ R_z(\theta_z)\ R_y(\theta_y)\ R_x(\theta_x)\ T(-p_0)$



Euler Rotation Matrix

Euler rotation matrix R in an affine space is represented by 3 orthogonal normalized vectors, *u,v and n*.

A vector V_{uvn} in *uvn* is transformed by R into V_{xyz} in *xyz*.

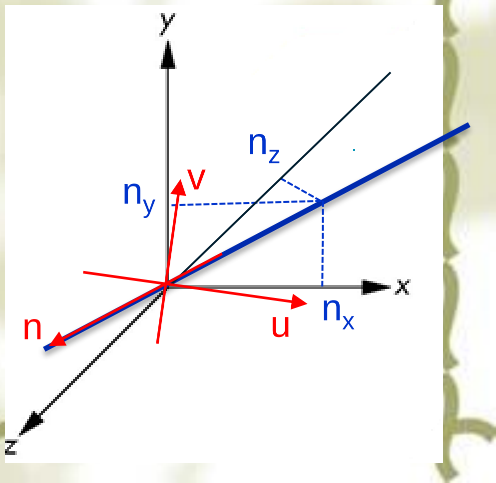
$$R=\begin{bmatrix} u_x & v_x & n_x & 0 \\ u_y & v_y & n_y & 0 \\ u_z & v_z & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_{xyz} = R\ V_{uvn}$$

A vector V_{xyz} in *xyz* is transformed by R^T into V_{uvn} in *uvn.*

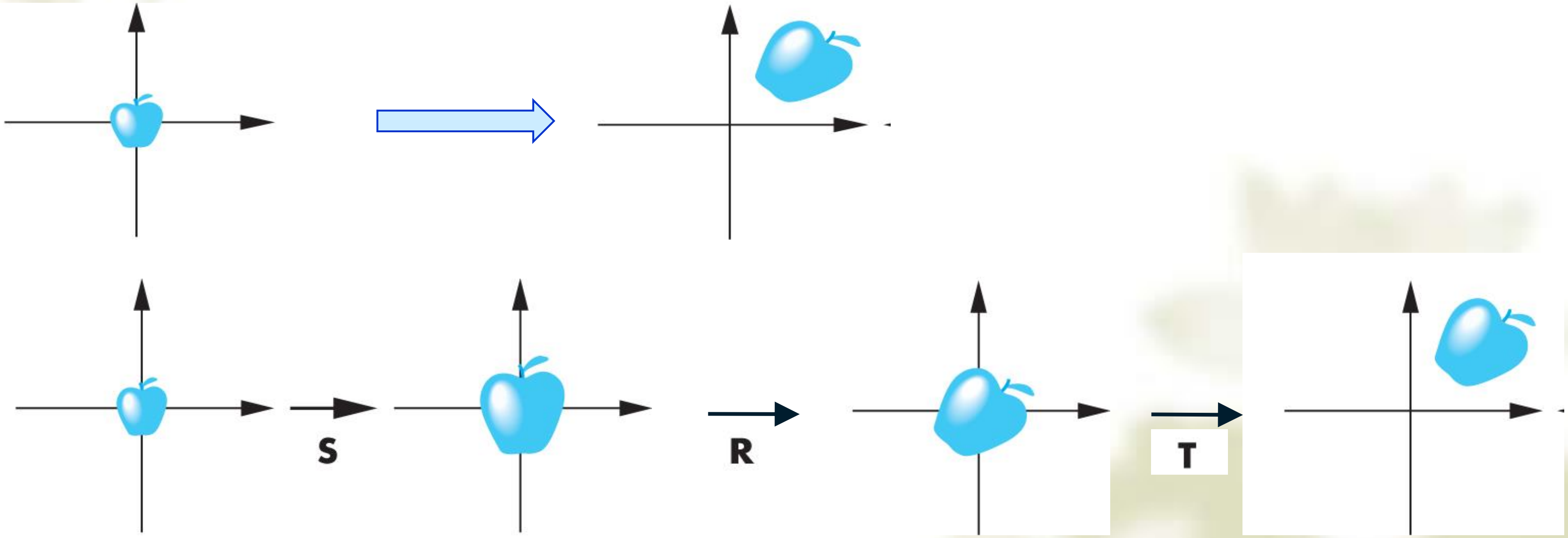
$$R' = R^T = R^{-1} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_{uvn} = R'\ V_{xyz}$$



Models in Scene

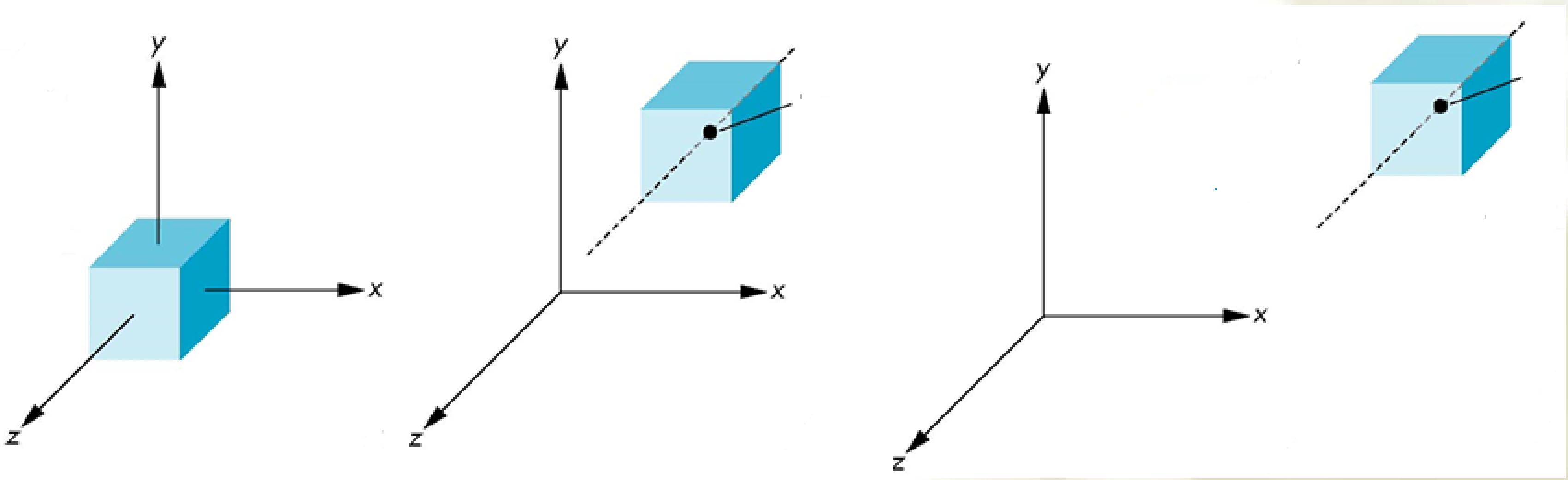
Put a model into the scene, an *instance transformation* to its vertices to Scale, and Orient, and Locate



M=TRS

Question2:

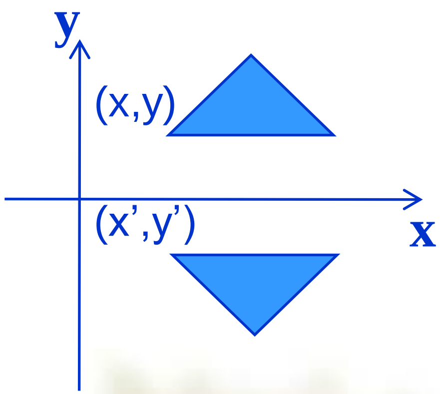
If we do the two consecutive translations, the two translations can commute?



Reflection Transformations

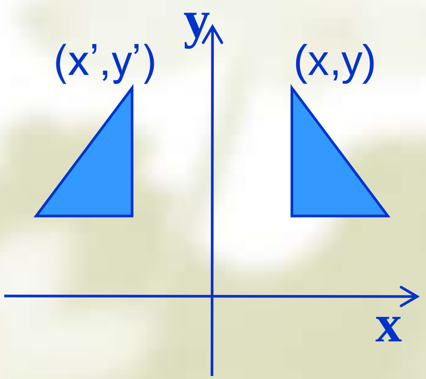
x-axis as a reflective axis, the matrix M:

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



y-axis as a reflective axis, the matrix M:

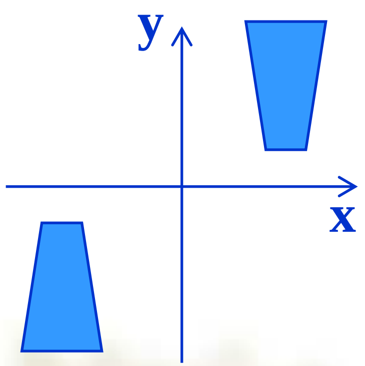
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Reflection Transformations

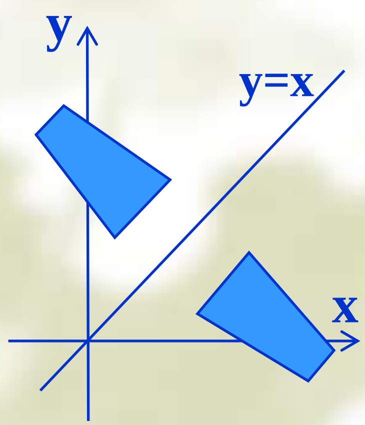
z-axis as a reflective axis (origin reflection), the matrix M:

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



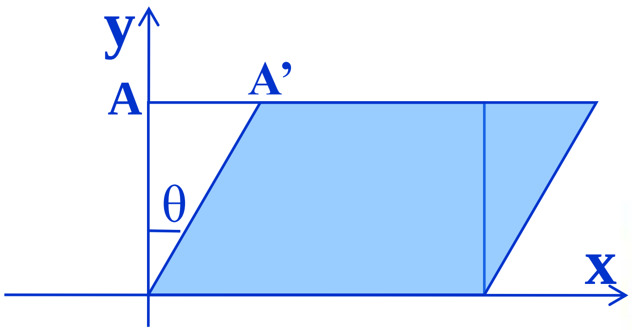
Line y=x as a reflective axis, the matrix M:

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



Shear Transformations

Shear along x-axis, the matrix M：$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

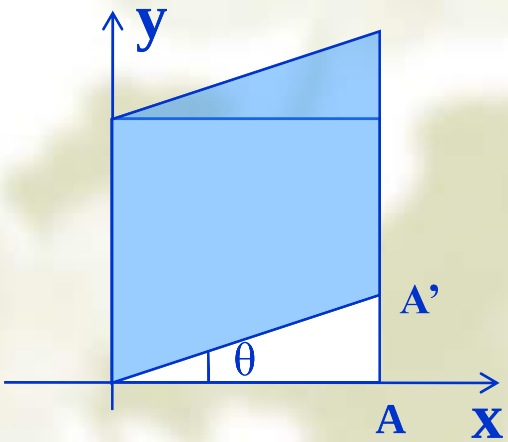


$$\begin{cases} x = x + sh_x * y \\ y = y \end{cases}$$

Shear parameter $sh_x = tg\,\theta$， represents the shear level along x-axis

So $|AA'| = y * sh_x$

Shear along y-axis, the matrix M：$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$



$$\begin{cases} x = x \\ y = y + sh_y * x \end{cases}$$

Twisting (differential rotation)

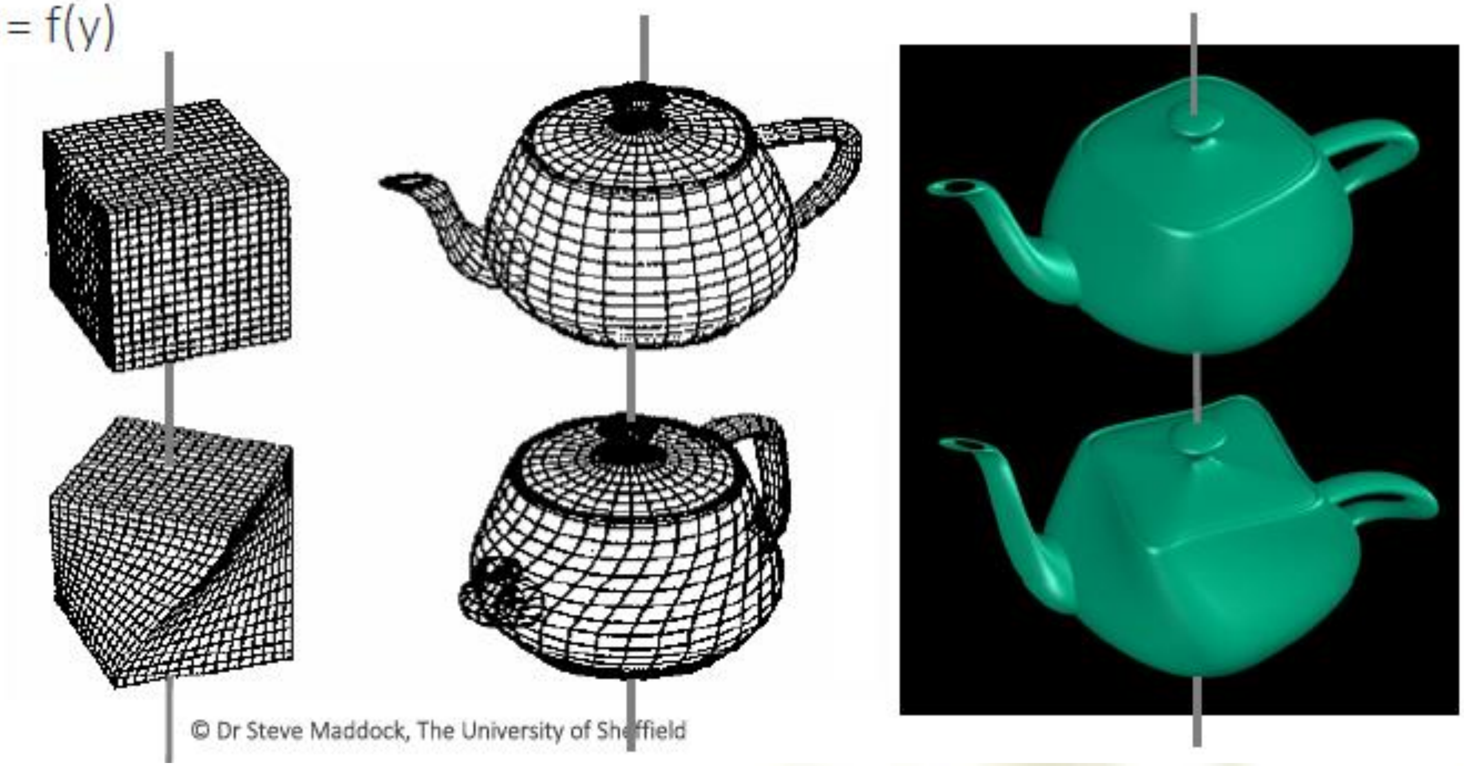
- Example: twist an object about its y axis:

  $x' = x\cos\theta + z\sin\theta$

  $y' = y$

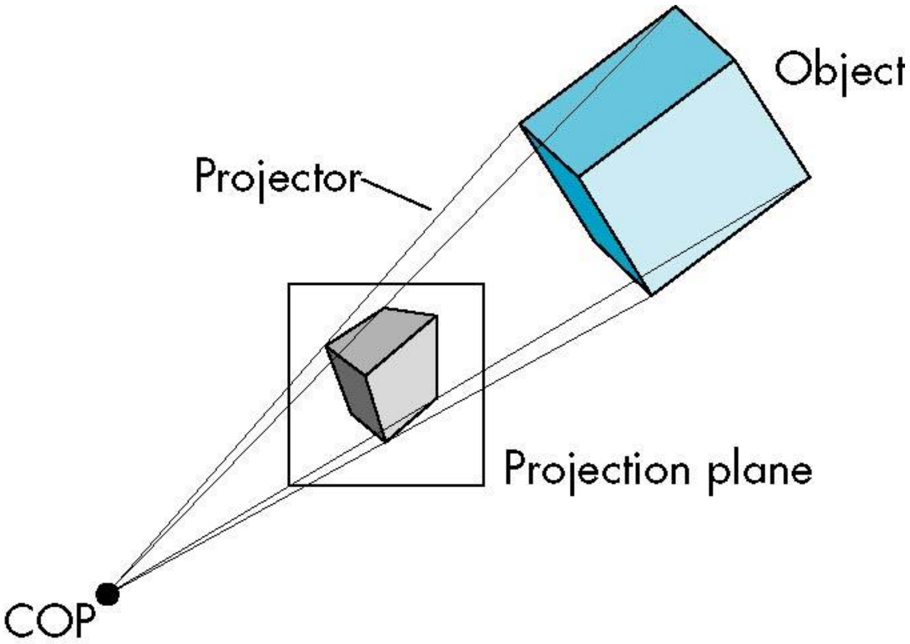
  $z' = -x\sin\theta + z\cos\theta$

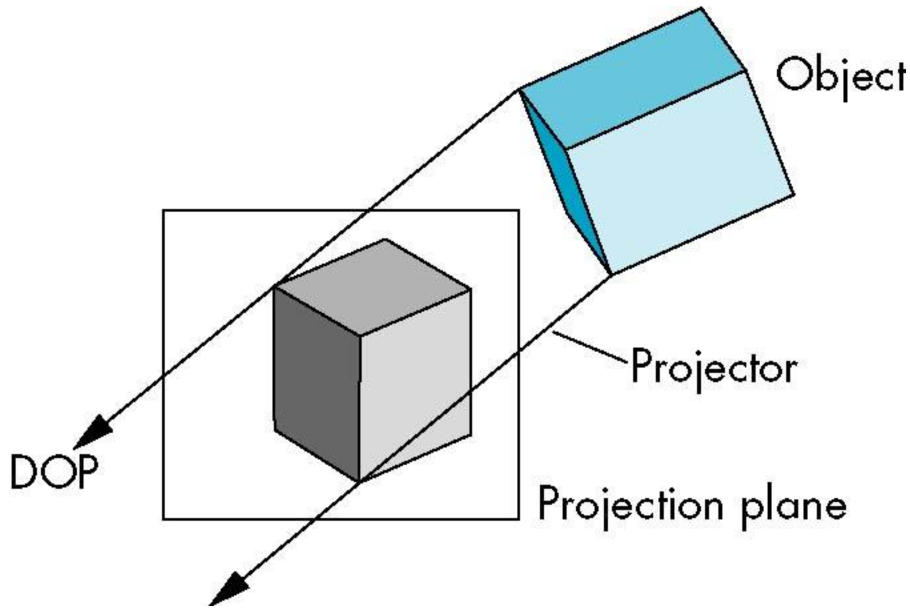
  where $\theta = f(y)$



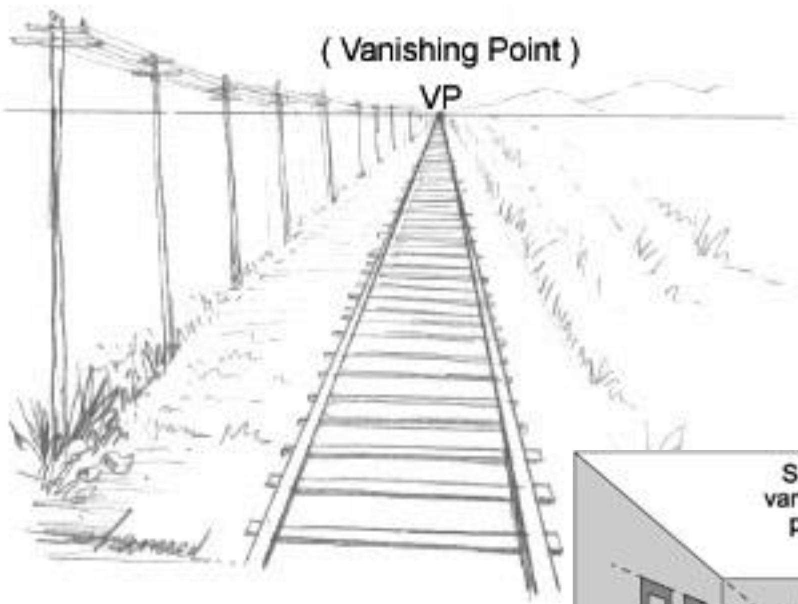
Projection Transformations

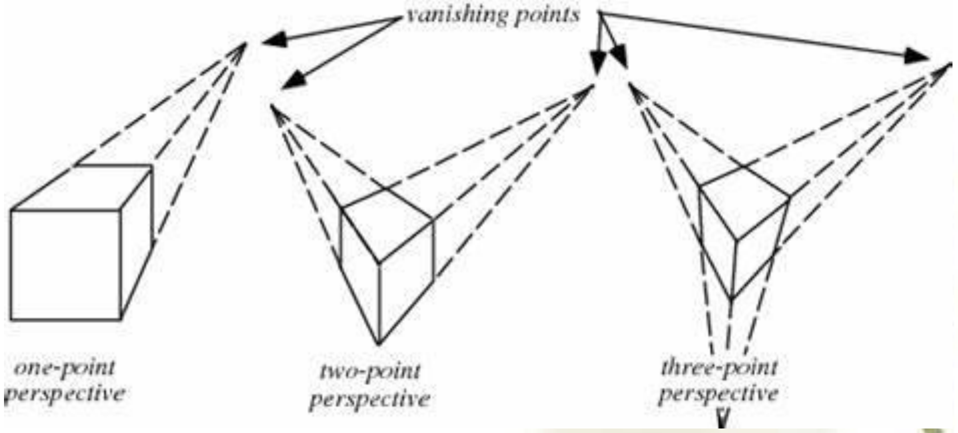
There are two types of projection transformation: perspective (left) or parallel (right)

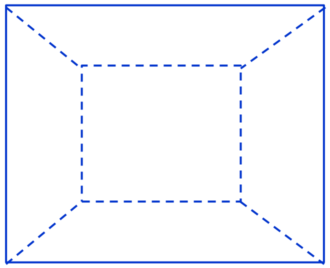


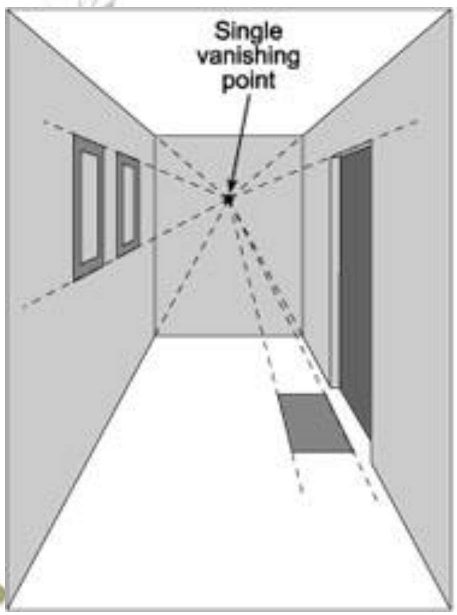


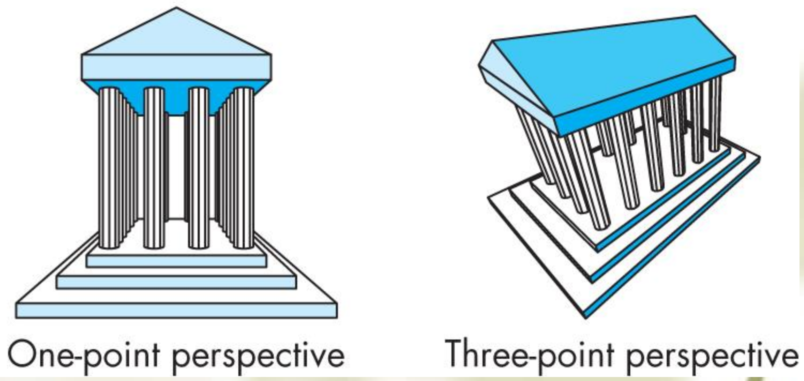
Vanishing Point in Perspective





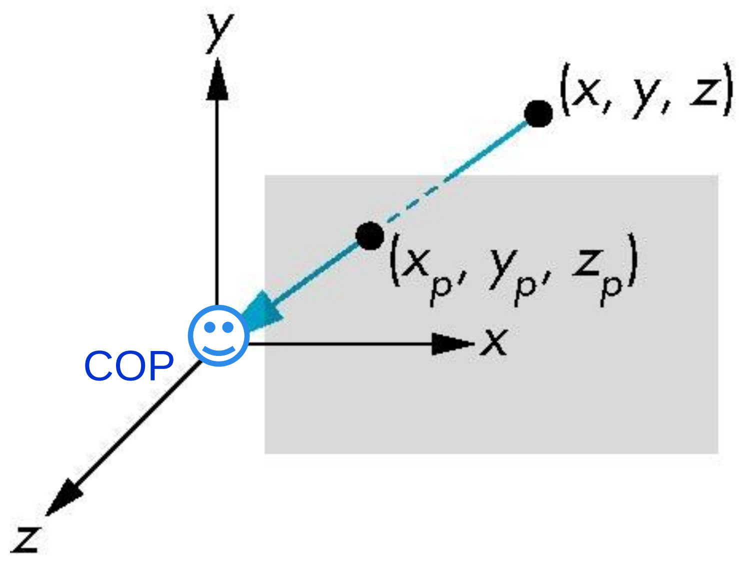






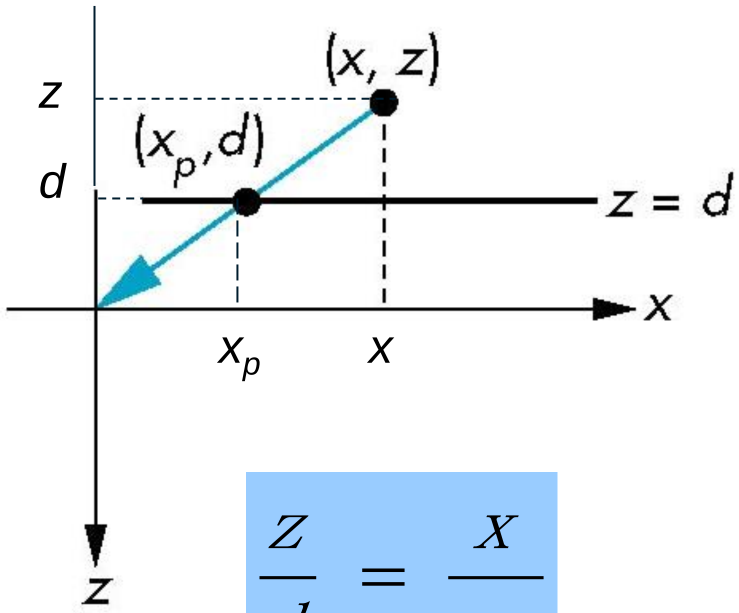
Simple Perspective

- Center of projection (COP, eyes) at the origin
- Projection plane $z = d$, ($d < 0$, equation）
- Vertex (x,y,z) on the object
- Vertex (x_p,y_p,z_p) on the projection plane

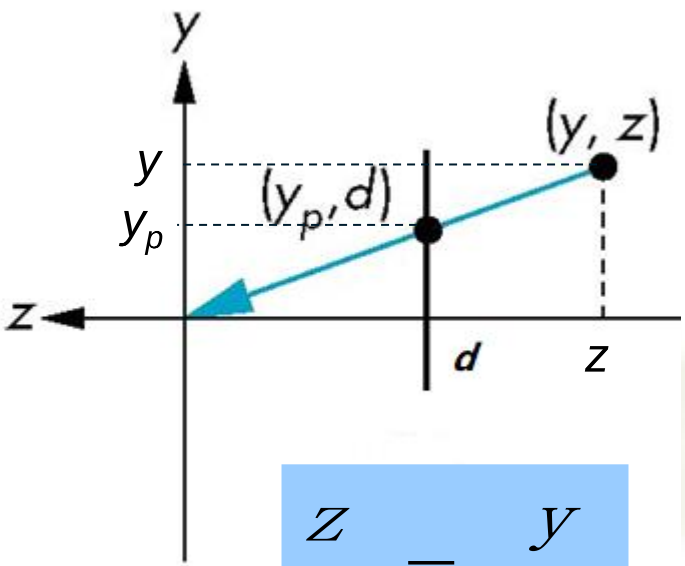


Perspective Equations

top view



side view



$$\frac{z}{d} = \frac{x}{x_p}$$

$$\frac{z}{d} = \frac{y}{y_p}$$

So,

$$x_p = \frac{x}{z/d} \qquad y_p = \frac{y}{z/d} \qquad z_p = d$$

perspective division

Perspective Matrix

Set, $h = \frac{z}{d}$

and, $x_h = h \cdot x_p \qquad y_h = h \cdot y_p \qquad z_h = h \cdot z_p = z$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

M_p

So, $\begin{bmatrix} x_p \\ y_p \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} x_h / h \\ y_h / h \\ z_h / h \\ 1 \end{bmatrix}$

(x_h,y_h,z_h,h) is the homogeneous point

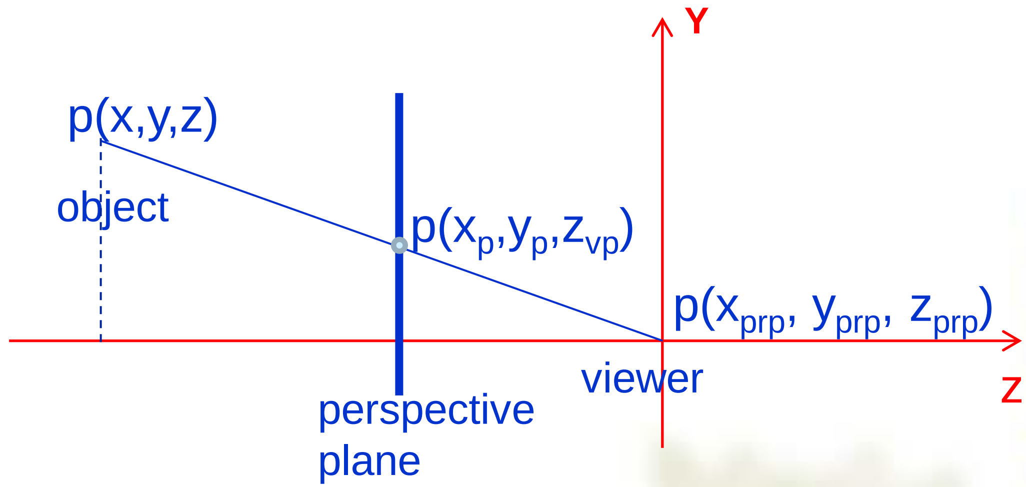
General Perspective Matrix

line equation:

$x' = (1-u)x + u * x_{prp}$

$y' = (1-u)y + u * y_{prp} \qquad 0 \le u \le 1$

$z' = (1-u)z + u * z_{prp}$



on the perspective plane:

$$\because \quad u = \frac{z_{vp} - z}{z_{prp} - z}$$

$$1 - u = 1 - \frac{z_{vp} - z}{z_{prp} - z} = \frac{z_{prp} - z_{vp}}{z_{prp} - z}$$

$$\therefore \begin{cases} x_p = x\left(\dfrac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + x_{prp}\left(\dfrac{z_{vp} - z}{z_{prp} - z}\right) \\ y_p = y\left(\dfrac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + y_{prp}\left(\dfrac{z_{vp} - z}{z_{prp} - z}\right) \end{cases}$$

Set: $h = z_{prp} - z$

So: $$\begin{cases} x_p = x\left(\dfrac{z_{prp} - z_{vp}}{h}\right) + x_{prp}\left(\dfrac{z_{vp} - z}{h}\right) \\ y_p = y\left(\dfrac{z_{prp} - z_{vp}}{h}\right) + y_{prp}\left(\dfrac{z_{vp} - z}{h}\right) \end{cases}$$

And set:

$$x_h = x_p \cdot h$$
$$y_h = y_p \cdot h$$

So: $$\begin{cases} x_h = x(z_{prp} - z_{vp}) + x_{prp}(z_{vp} - z) \\ y_h = y(z_{prp} - z_{vp}) + y_{prp}(z_{vp} - z) \end{cases}$$

here, (x_h, y_h, z_h, h) is the homogeneous point, corresponds to the point $(x_h/h, y_h/h, z_h/h, 1)$

$$p_h = M_{pers} \cdot p$$

$$\begin{pmatrix} x_h \\ y_h \\ z_h \\ h \end{pmatrix} = \begin{bmatrix} z_{prp} - z_{vp} & 0 & -x_{prp} & x_{prp} z_{vp} \\ 0 & z_{prp} - z_{vp} & -y_{prp} & y_{prp} z_{vp} \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & z_{prp} \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

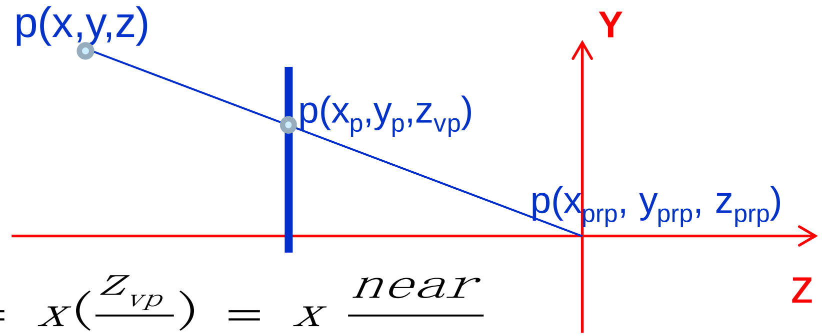
where,

$$s_z = -\frac{near + far}{near - far}$$

$$t_z = -\frac{2 * near * far}{near - far}$$

$$\begin{pmatrix} x_p \\ y_p \\ z_{vp} \\ 1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} x_h / h \\ y_h / h \\ d \\ 1 \end{pmatrix}$$

General Cases:



1. $(x_{prp},y_{prp},z_{prp})=(0,0,0)$

   视点在原点

$$\begin{cases} x_p = x\left(\dfrac{z_{vp}}{z}\right) = x\dfrac{near}{z} \\ y_p = y\left(\dfrac{z_{vp}}{z}\right) = y\dfrac{near}{z} \end{cases}$$

2. $x_{prp}=y_{prp}=z_{vp}=0$

   投影面在原点

$$\begin{cases} x_p = x\left(\dfrac{z_{prp}}{z_{prp}-z}\right) \\ y_p = y\left(\dfrac{z_{prp}}{z_{prp}-z}\right) \end{cases}$$

3. $x_{prp}=y_{prp}=0$

   视点、投影面都不在原点
   但视点在z轴上

$$\begin{cases} x_p = x\left(\dfrac{z_{prp}-z_{vp}}{z_{prp}-z}\right) \\ y_p = y\left(\dfrac{z_{prp}-z_{vp}}{z_{prp}-z}\right) \end{cases}$$

4. $z_{vp}=0$

   投影面在原点
   视点不在z轴上

$$\begin{cases} x_p = x\left(\dfrac{z_{prp}}{z_{prp}-z}\right) - x_{prp}\left(\dfrac{z}{z_{prp}-z}\right) \\ y_p = y\left(\dfrac{z_{prp}}{z_{prp}-z}\right) - y_{prp}\left(\dfrac{z}{z_{prp}-z}\right) \end{cases}$$

Simple Parallel: Orthographic Matrix

Homogeneous coordinate represents orthographic projection as 4X4

$$\begin{cases} x_p = x \\ y_p = y \\ z_p = 0 \\ w_p = 1 \end{cases}$$

$$p_p = \mathbf{M}\, p$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In practice, we can let **M = I** and set the *z* term to zero later

作业4

1. write down the matrices for rotation about y and z axes

2. Page 161. #4.1

3. Page 161. #4.23

Question:

1. why can not commute the matrixes?
   Please give a sample to explain the reason
2. If we do the two consecutive translations, the two translations can commute?