Web Based Graphics & Virtual Reality Systems

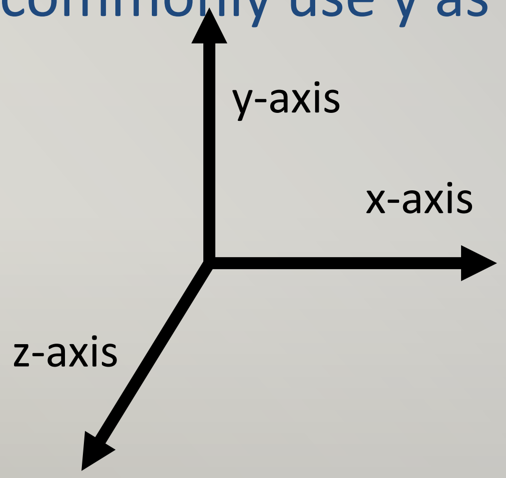
3D Graphics: Vector, Matrix and Transformation

Recap

- In the last lecture, we have already studied some of the basics of 2D graphics, including
  - Representation of shapes
  - 2D Vector
  - 2D Matrix
  - 2D Transformation

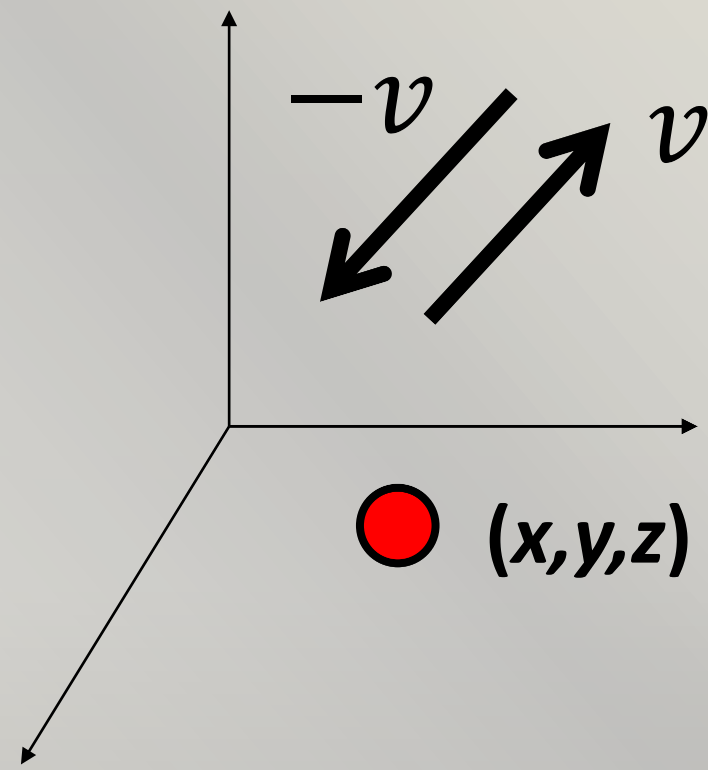
3D Graphics

- Now, we are going to extend from 2D to 3D
- In 3D coordinate system, we have one more axis in the frame, usually we call it z-axis
  - x,y and z axis are perpendicular to each other
  - In graphics, we commonly use y as the up-direction (although it is not a MUST)



3D Vertex and Vector

- Similar to case in 2D, the most basic object is a point (or vertex)
  - *Its coordinate will be (x,y,z)*



- A 3D vector contains 3 elements v_x, vy, v_z
  - $v = < v_x, vy, v_z >$
  - It is directional,
    so $-v$ is in reverse direction to v

3D Vertex and Vector

- Vertex : a position in 3D
- Vector : a direction with magnitude in 3D
- The magnitude is the length of vector

$$|v| = \sqrt{(v_x \times v_x)+(v_y \times v_y)+(v_z \times v_z)}$$

e.g. $v = \langle 1,2,4 \rangle$, $|v| = \sqrt{(1 \times 1)+(2 \times 2)+(4 \times 4)}$

$$= \sqrt{21}$$

A vector with magnitude equals to 1 is called a <u>unit vector</u>

Adding 3D Vectors

- vector v add to vector u form a new vector

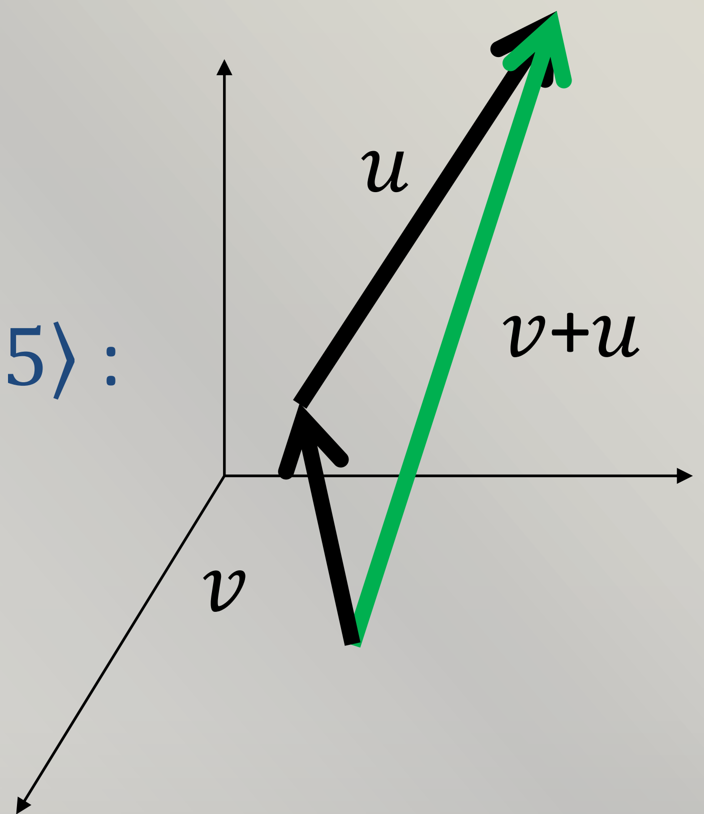
  v+u

Example v = $\langle 1,2,4 \rangle$, u = $\langle 4, 3,5 \rangle$:

v+u = $\langle 1,2,4 \rangle$ + $\langle 4, 3,5 \rangle$

= $\langle 1+4, 2+3, 4+5 \rangle$

= $\langle 5,5,9 \rangle$



Multiplying 3D Vectors with Scalar

- Usually, we refers to a single value as scalar
- When a vector v is multiplying with a scalar s

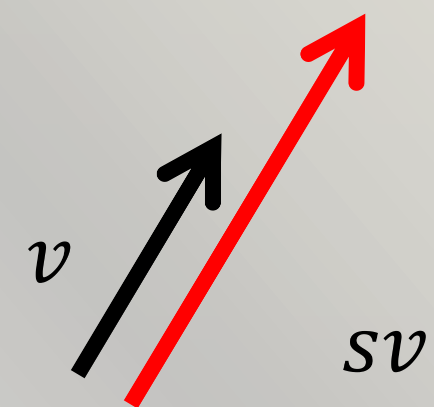
$sv = <sv_x, sv_y, sv_z>$

Physically, we lengthen or shorten the vector

Increase/decrease its magnitude

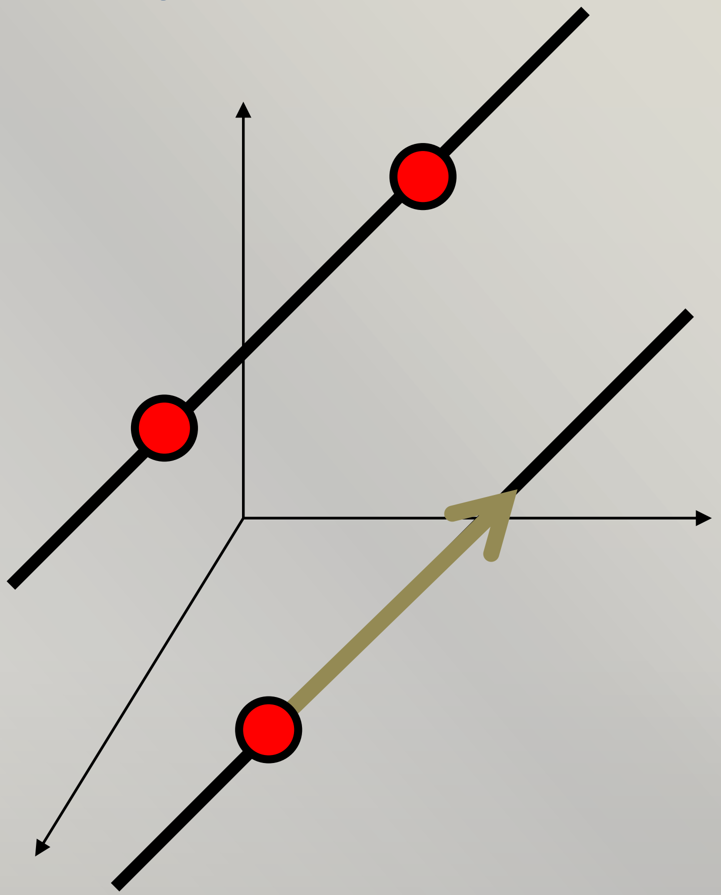
E.g. $s = 2.1$, $v = <1,3,4>$

$sv = = < 2.1,\ 6.3,\ 8.4 >$



Representing a Line

- To form a line, we need at least 2 points / vertices
- Or 1 point and a vector



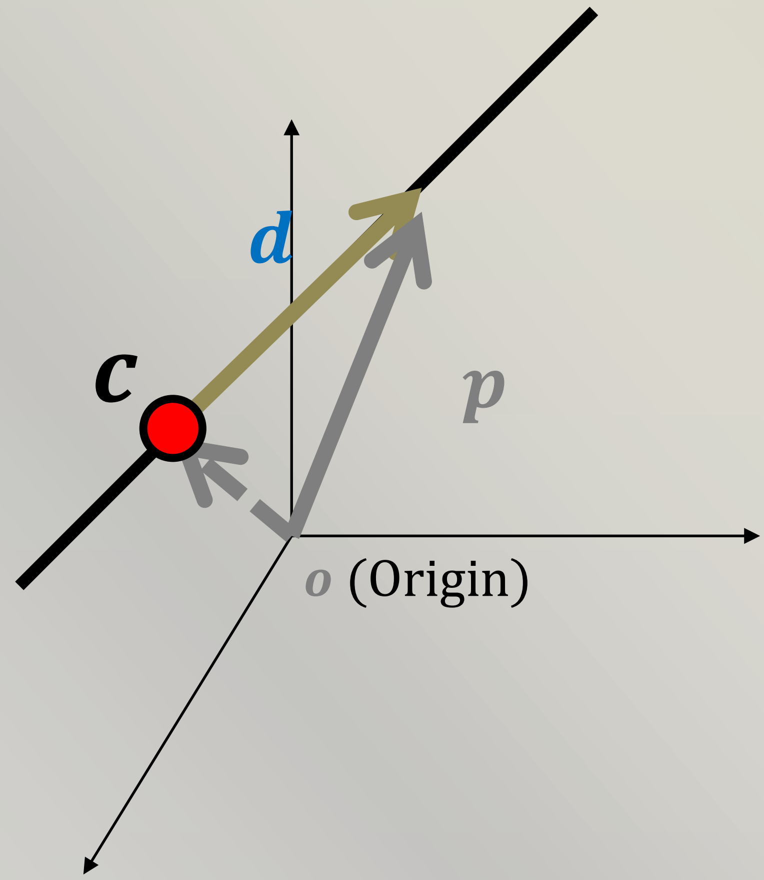
Representing a Line

- In vector form, we have

$$\boldsymbol{p} = \boldsymbol{c} + t\,\boldsymbol{d}$$

vector

  - notice t is a scalar
  - c is any point on the line
  - d is the direction of the line
- So, vertex $\boldsymbol{p}$ is always on the line



Representing a Line

$p = c + t\, d$

- E.g. C = <3,2, 2> , d = <0.5, 0.4, 0.1>
- Then, the line equation will be

p = <3,2, 2> + t <0.5, 0.4, 0.1>

By putting any value of t, we will get n which is always on the line. E.g. t = 0.1

p = <3,2, 2> + 0.1 * <0.5, 0.4, 0.1>

= (3.05,2.04, 2.01)

Dot Product

- The dot product (·)

$$v \cdot u = (v_x \times u_x) + (v_y \times u_y) + (v_z \times u_z)$$

e.g. $v = \langle 1,2,4 \rangle$, $u = \langle 4,3,5 \rangle$

$v \cdot u = 1\times4 + 2\times3 + 4\times5$

$= 5 + 6 + 20 = 31$

  - Notice the result is a scalar (single value) but not vector

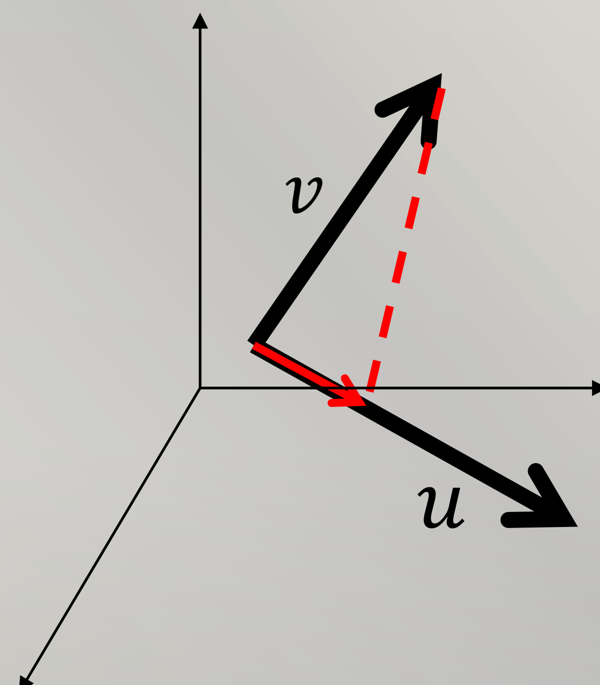
Dot Product

- One physical meaning of dot product is the length of the *projection* of v onto u multiplied by the length of u
- Also, we can use dot product to check if two vectors are perpendicular to each other
  - If the projection has 0 length, then, the two vectors are perpendicular
  - A typical example may be the a unit vector on x-axis and y-axis

$\langle 1,0,0 \rangle \cdot \langle 0,1,0 \rangle$

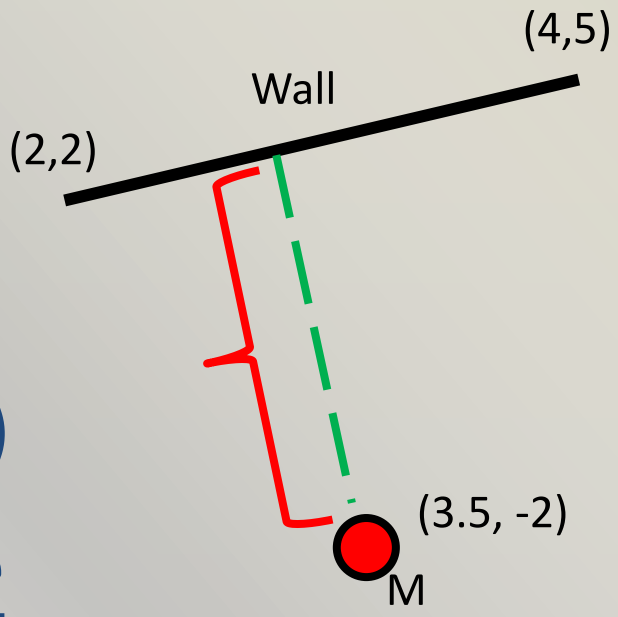
$= 1*0 + 0*1 + 0*0$

$= 0$



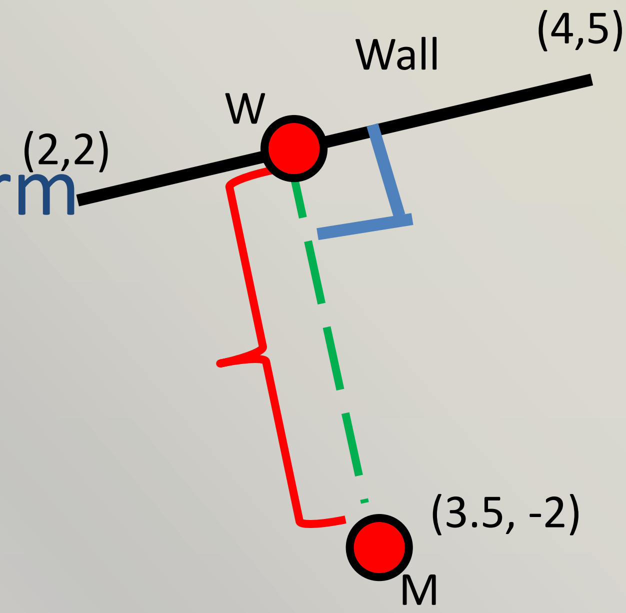
Example

- Given a 2D wall defined by its two end points at (2,2), (4,5)
- if a man is positioned at (3.5,-2)
- What is the **<u>minimum distance</u>** of this man to the wall ? (i.e. minimum distance the man will hit the wall)



Example (Cont’)

- Reminded that the min. dist form at a point W, and MW always perpendicular to the wall !



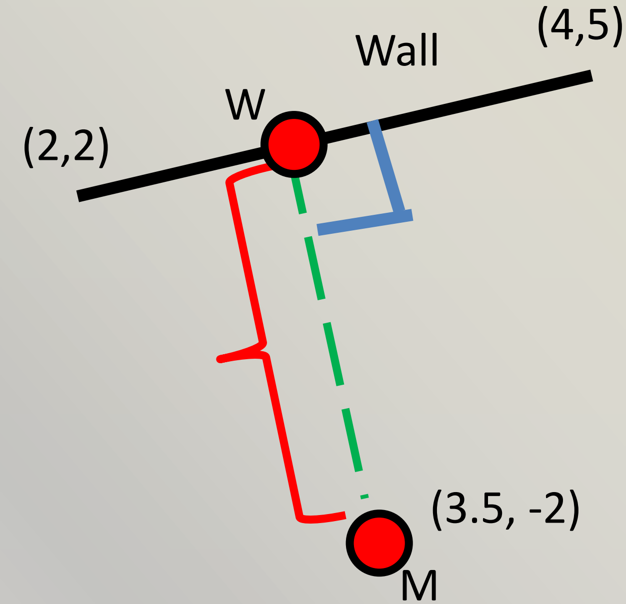
Solution:

1. To find the line equation of the wall :

2. $\mathbf{w} = \langle 2, 2\rangle + t * \langle 4-2, 5-2\rangle$

   $= \langle 2, 2\rangle + t * \langle 2, 3\rangle$

Example (Cont’)



Solution (cont’):

Therefore, MW will form a vector, such that

mw= <2, 2> + t * <4-2,5-2>
- (3.5, -2)

= <-1.5,4> + t * <2,3>

As MW is perpendicular to the Wall

mw dot <2,3> = 0

(<-1.5,4> + t * <2,3>) dot <2,3> = 0

(<-1.5+2t,4+3t>) dot <2,3> = 0

Example (Cont')

Solution (cont'):

<-1.5+2t,4+3t> dot <2,3> = 0

(-1.5+2t)*2 + (4+3t)*3 = 0

-3 + 4t + 12 + 9t = 0

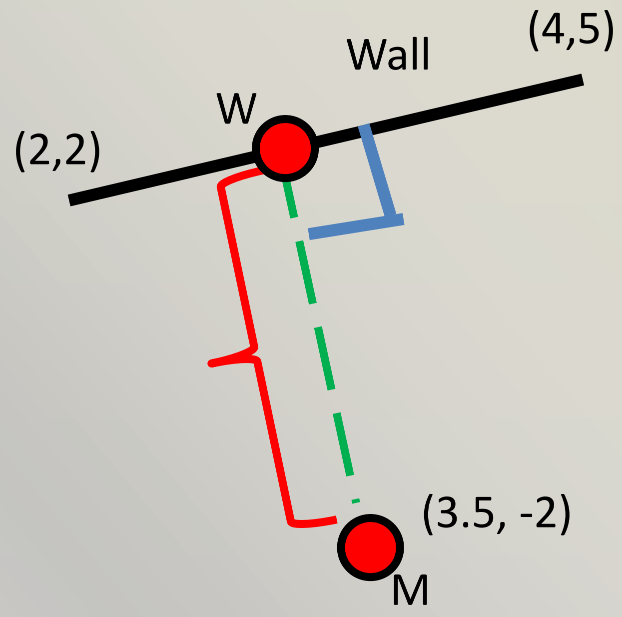
13t = 9

t = 9/13 = 0.6923

Substitute this to compute MW

mw = <-1.5,4> + 0.6923 * <2,3>

= <-0.1154 , 6.0769>



Example (Cont')

Solution (cont'):

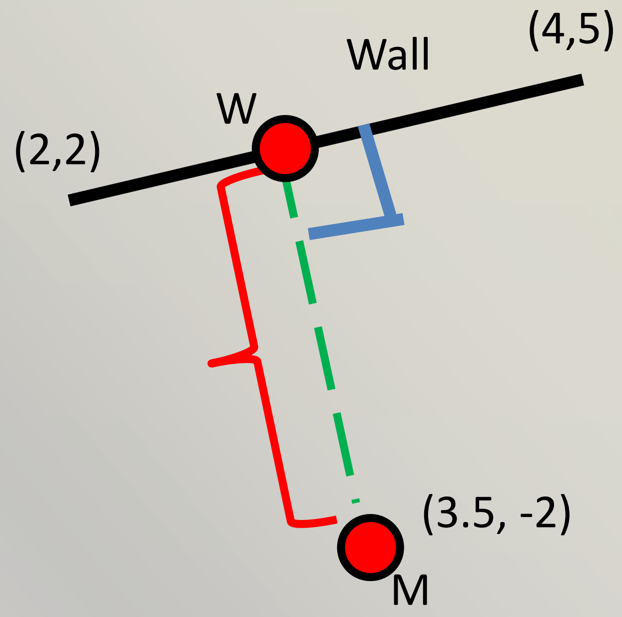
Finally, compute length of MW

mw = <-0.1154 , 6.0769>

= √ -0.1154^2 + 6.0769^2

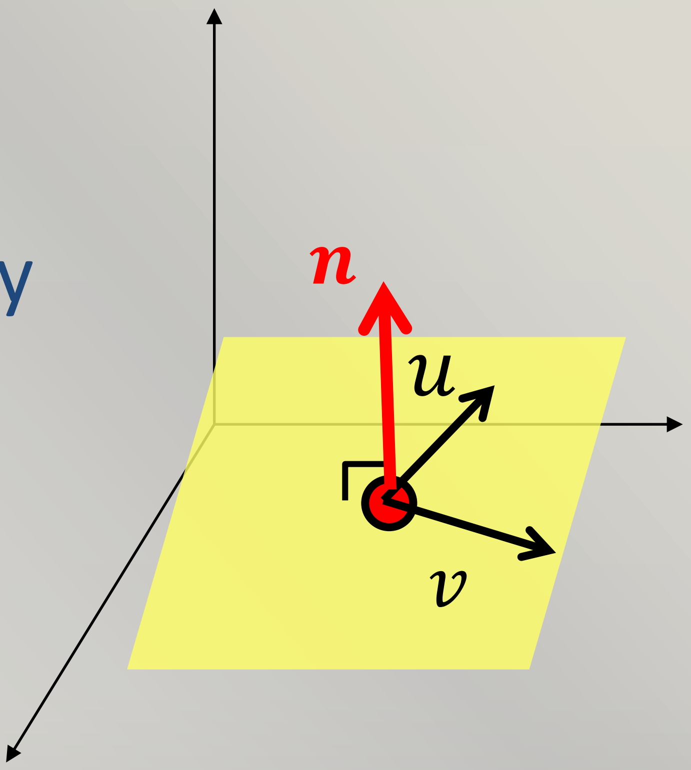
= √ 36.942

= 6.078



Representing a Plane

- A plane can be defined by a point and two vectors or by three points
  - All of them lies on the plane
- But the normal n is commonly used to represent a plane
  - A normal is the vector which perpendicular to the plane



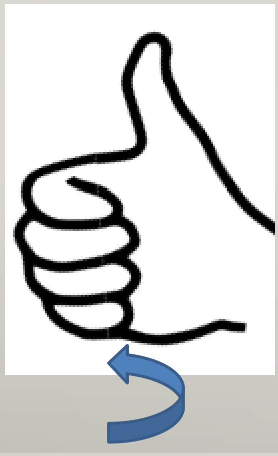
Cross Product

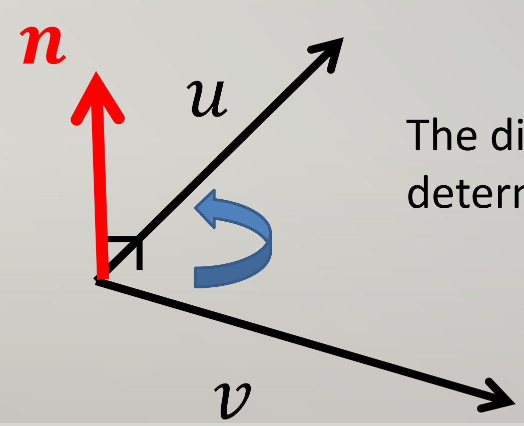
- Actually, we can deduce the normal based on two vectors on the plane by cross product
- Cross product in 3D

$$n = v \times u$$
$$= (v_y u_z - v_z u_y)i + (v_z u_x - v_x u_z)j + (v_x u_y - v_y u_x)k$$

Where i , j, k are standard unit vectors:

i.e. i= <1,0,0> , j= <0,1,0> , k = <0,0,1>





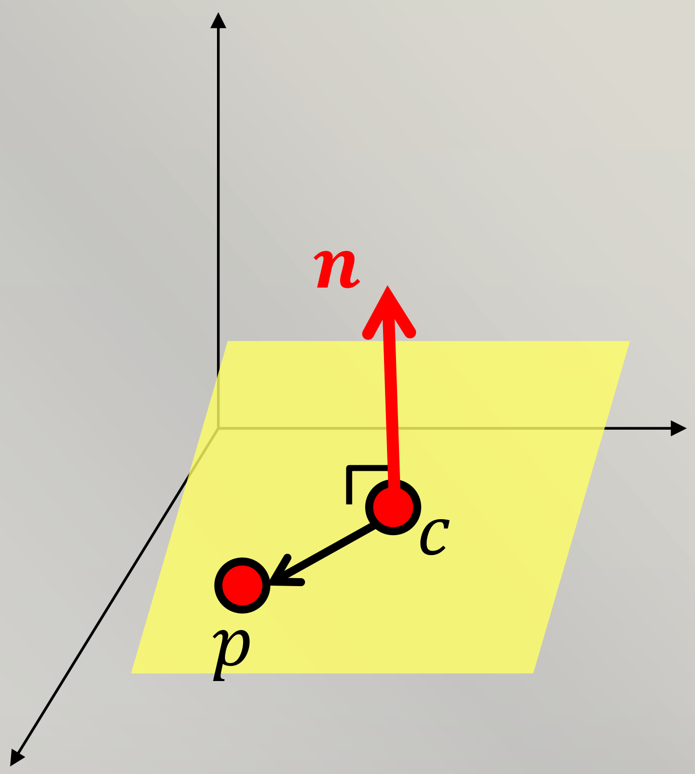
The direction of the vector can be determined by right hand rule

Representing a Plane

- The vector form of a plane:

$$(p - c) \cdot n = 0$$

- Because for any vertex p , if it can from a vector with another vertex c on the plane; this vector is suppose to be perpendicular to the normal n of the plane



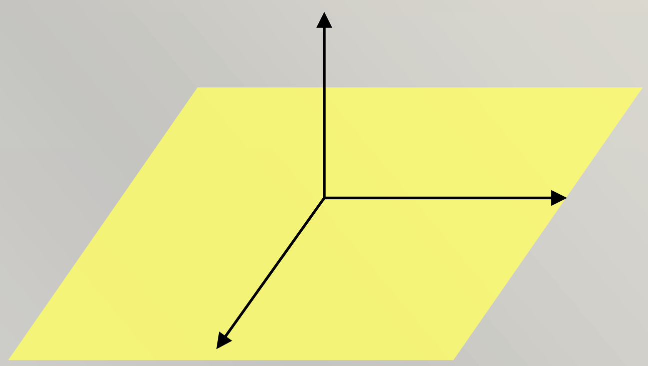
Representing a Plane

For a plane with $c = (0,0,0)$, $n = <0,1,0>$

$(p - (0,0,0)) \cdot <0,1,0> = 0$

will reduce to

$p \cdot <0,1,0> = 0$



Basically, p can be of form <a,0,b> for arbitrary a and b;

this also mean the any point on the x-z plane

Transformation in 3D

- Similar to 2D, common operations on 3D objects include
  - Translation
  - Scaling
  - Rotation

Homogeneous coordinates

- To recap in last lecture, when we discuss about translation
- The translation can not base on matrix multiplication again but require an addition

$$\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} x+3 \\ y+4 \end{pmatrix}$$

- Homogeneous coordinates are able to solve the problem in performing **<u>translation</u>** with also matrix multiplication

Homogeneous coordinates

- We introduce one more dimension w, so that a vertex with homogeneous coordinates
  $<x, y, z, w>$
  can be changed to Cartesian coordinates by

$$<v_x, v_y, v_z> = <x/w, y/w, z/w>$$

where w is not 0

Homogeneous coordinates

- One commonly used w is 1
- Homogeneous coordinates are a standard in all computer graphics systems including current hardware pipeline
- Common transformations (rotation, translation, scaling) can be done with matrix multiplications using 4 x 4 matrices

Common Matrix Operations in Graphics

- A recall of what had learnt about matrix multiplication
- It's general form

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ \boxed{a_{i1} \quad \cdots \quad a_{im}} \\ \vdots & & \vdots \\ a_{r1} & \cdots & a_{rm} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & \boxed{\begin{matrix} b_{1j} \\ \vdots \\ b_{mj} \end{matrix}} & \cdots & b_{1c} \\ \vdots & & & & \vdots \\ b_{m1} & \cdots & & \cdots & b_{mc} \end{bmatrix} = \begin{bmatrix} p_{11} & \cdots & p_{1j} & \cdots & p_{1c} \\ \vdots & & \vdots & & \vdots \\ p_{i1} & \cdots & \boxed{p_{ij}} & \cdots & p_{ic} \\ \vdots & & \vdots & & \vdots \\ p_{r1} & \cdots & p_{rj} & \cdots & p_{rc} \end{bmatrix}$$

$$p_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj}.$$

- An example

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 12 & 17 & 22 & 27 \\ 24 & 33 & 42 & 51 \end{bmatrix}$$

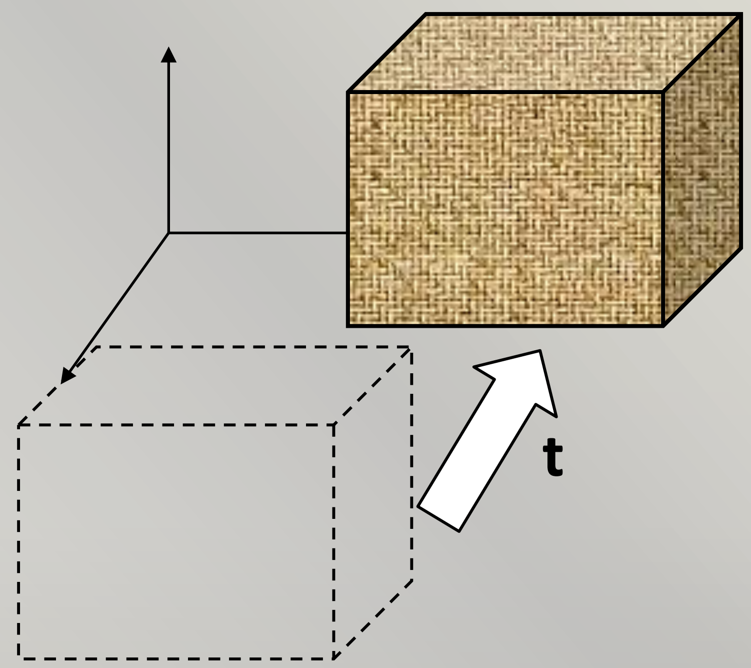
3D Translation

- The translation can be represented in 4 x 4 matrix T in homogeneous coordinates:

Identity Matrix

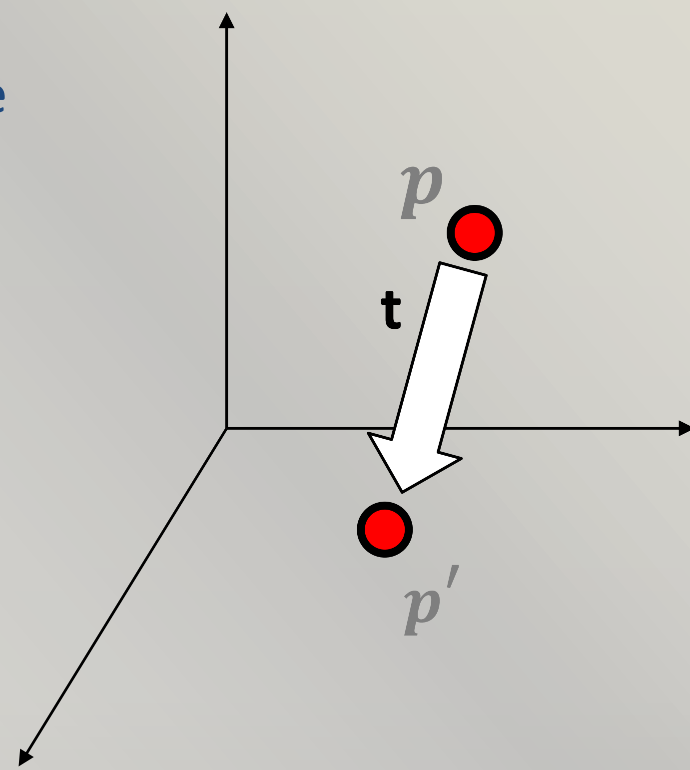
$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Here $t = \langle t_x, t_y, t_z \rangle$ is the direction and magnitude of move



3D Translation

- E.g. t = <2,-1,3>, p = (4,2,0,1)
  - Notice p is in homogenous coordinate
- $\boldsymbol{p}' = \mathbf{T}\boldsymbol{p}$



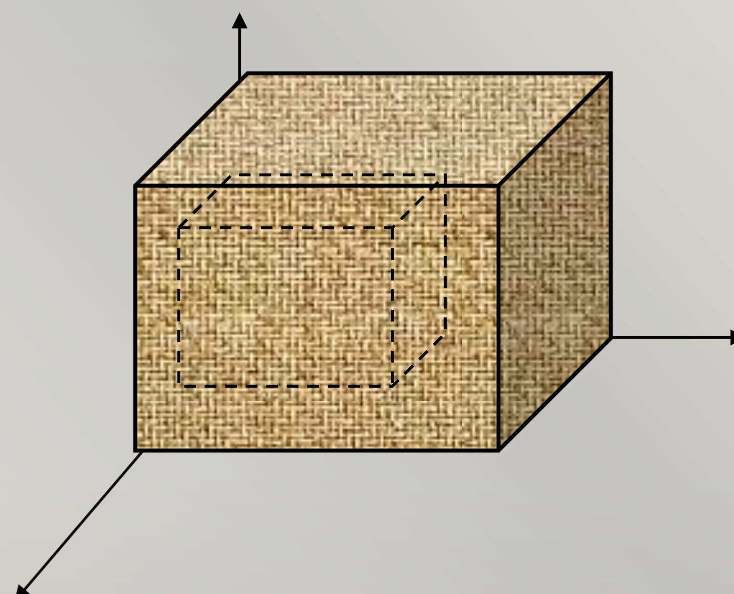
$$\boldsymbol{p}' = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{p}' = \begin{bmatrix} 4+2 \\ 2-1 \\ 0+3 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 3 \\ 1 \end{bmatrix}$$

3D Scaling

- Scale Matrix is a direct extension to its 2D version,
- Now, we have s_x, s_y and s_z as the scaling factor in x, y and z directions

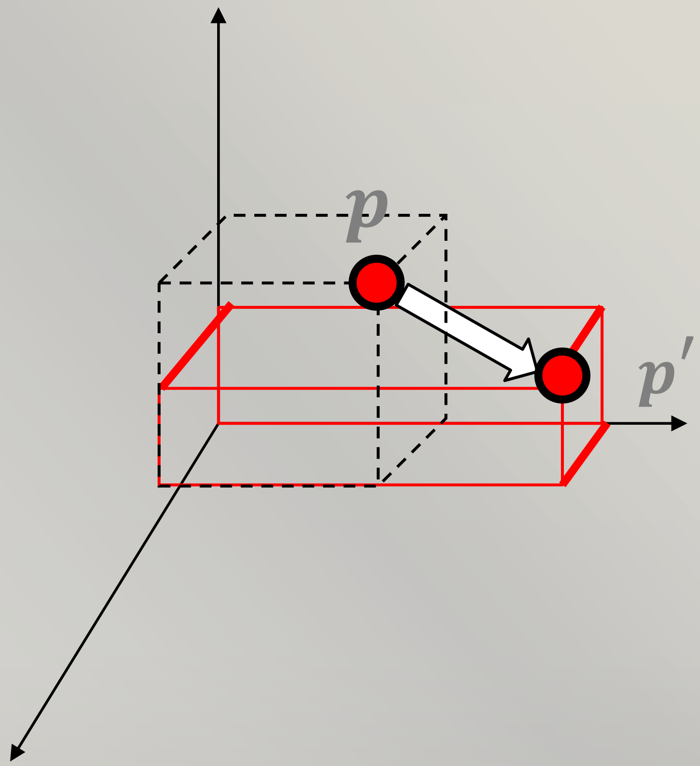
$$\text{scale}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$



3D Scaling

- E.g. if we have scale factors = (2.0, 0.5, 1.0)
- For a vertex p at (1,1,1)

$$\begin{bmatrix} 2.0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.5 \\ 1 \\ 1 \end{bmatrix}$$



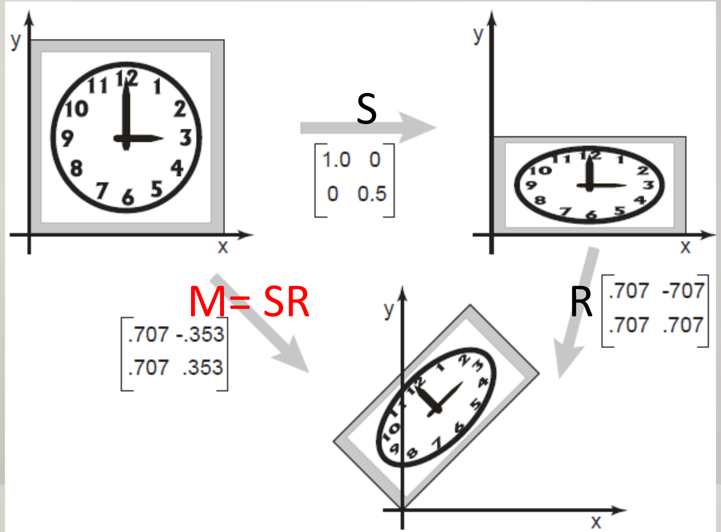
- For an object, we will multiply with all vertices of the object

3D Scaling

- Notice that the scaling is in reference to the origin (0,0,0)
- If object's center is not at origin, a direct multiply with the scale matrix will look like moving the object at the same time
- A simple solution is to first translate the object to the origin, scale and then translate back to the original center

Multiple 3D Transformations

- Before discussing the solution, recall the topic about multiple transformation in last lesson
- To apply more than one transformation, e.g.
  - First, shrink in Y direction for 0.5 (Matrix S)
  - Then, rotate 45 degree in anticlockwise (Matrix R)
- The standard way is do it stepwise, but it is the same as we multiply RS to the vertex

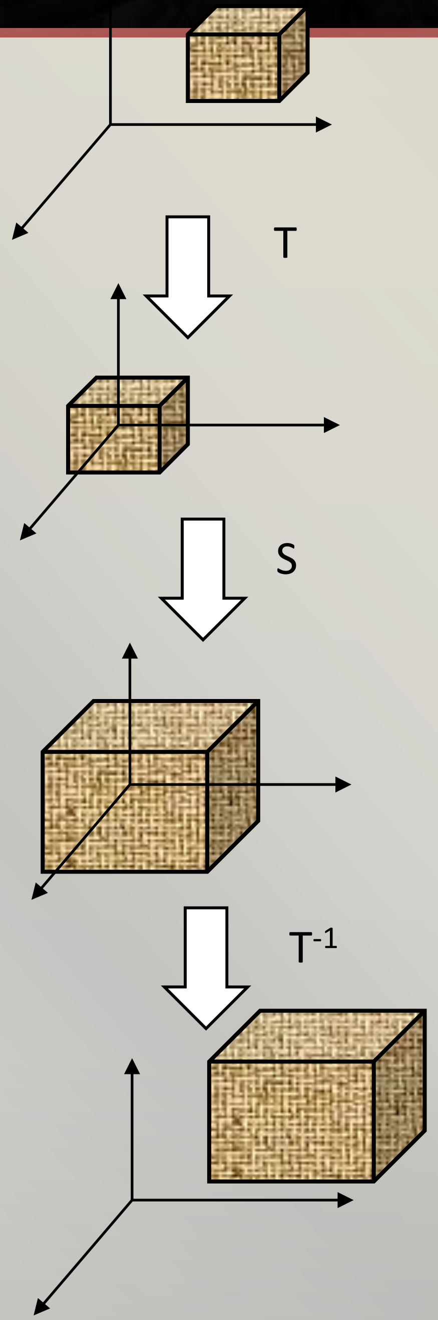


3D Scaling

- To scale according to object center, we need to multiply with translation matrix T, then scale translation matrix S and finally inverse of T (i.e. T^{-1})

$$T^{-1}ST$$

Note the translation is to move the object center to origin



Inverse Matrix

- By definition, inverse matrix A^{-1} of a matrix A should satisfy:

$$A^{-1} A = I$$

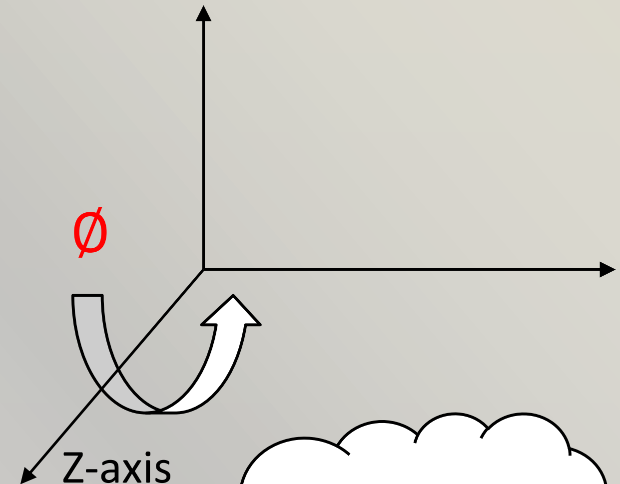
  - I is an identity matrix
  - that is they cancel out each other's effect
- The computation of inverse matrix is a bit involving, and had several ways to do, please check your textbook or books of linear algebra for details

Inverse Matrix

- However, for the 3 common rigid transformation, we have more simpler form:
  - Translation: $\mathbf{T}^{-1}(t_x, t_y, t_z) = \mathbf{T}(-t_x, -t_y, -t_z)$
  - Rotation: $\mathbf{R}^{-1}(\emptyset) = \mathbf{R}(-\emptyset)$
  - Scaling: $\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

3D Rotation

- To rotate in 3D, we need
  - Reference axis
  - Angle of rotation Ø
- A convenience choice of axis is the z-axis

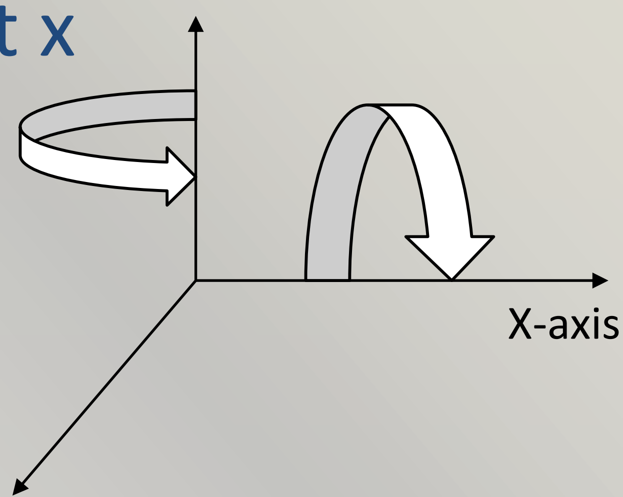


Same as 2D rotation

$$\mathbf{R} = \mathbf{R}_Z(\emptyset) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Rotation

- Similarly we can rotate about x and y axis



$$\mathbf{R}_x(\emptyset) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\emptyset) = \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

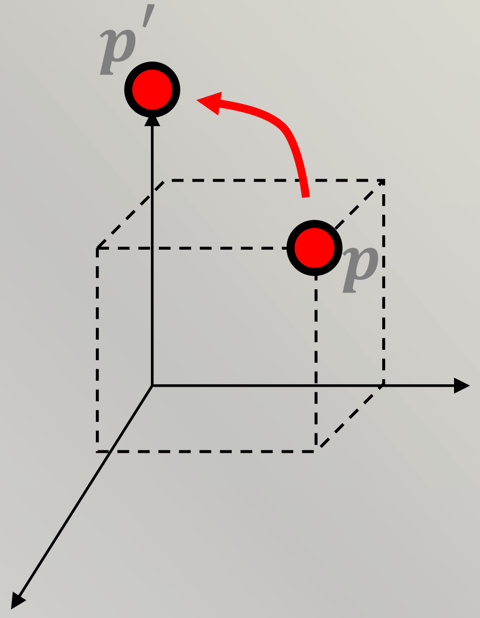
3D Rotation

- E.g. A rotation with 45 degrees (PI/4) in anticlockwise, a point p =(1,1,1)

$$\boldsymbol{p}' = \mathbf{R}_z \boldsymbol{p} = \begin{bmatrix} \cos\pi/4 & -\sin\pi/4 & 0 & 0 \\ \sin\pi/4 & \cos\pi/4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.707 - 0.707 \\ 0.707 + 0.707 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.414 \\ 1 \\ 1 \end{bmatrix}$$



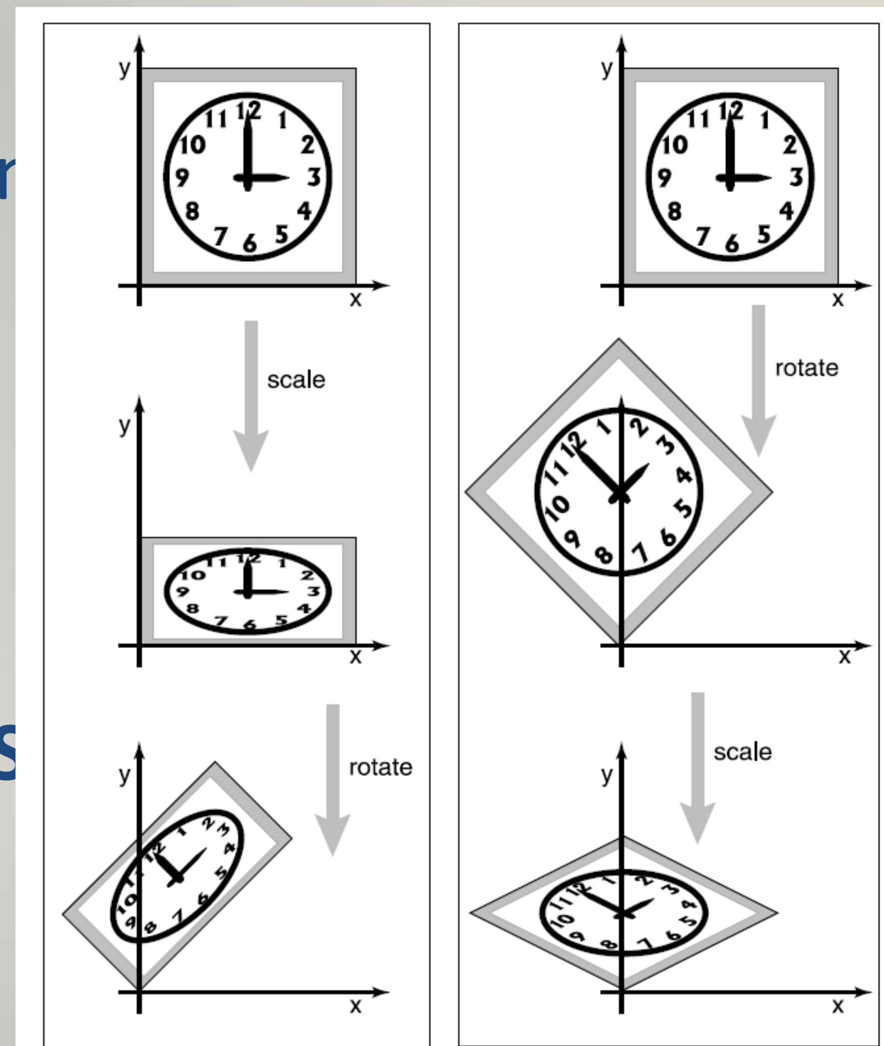
Order of Applying Transformations

- Similar to 2D, order of applying transformation in 3D DOES Matter

- E.g. we know that

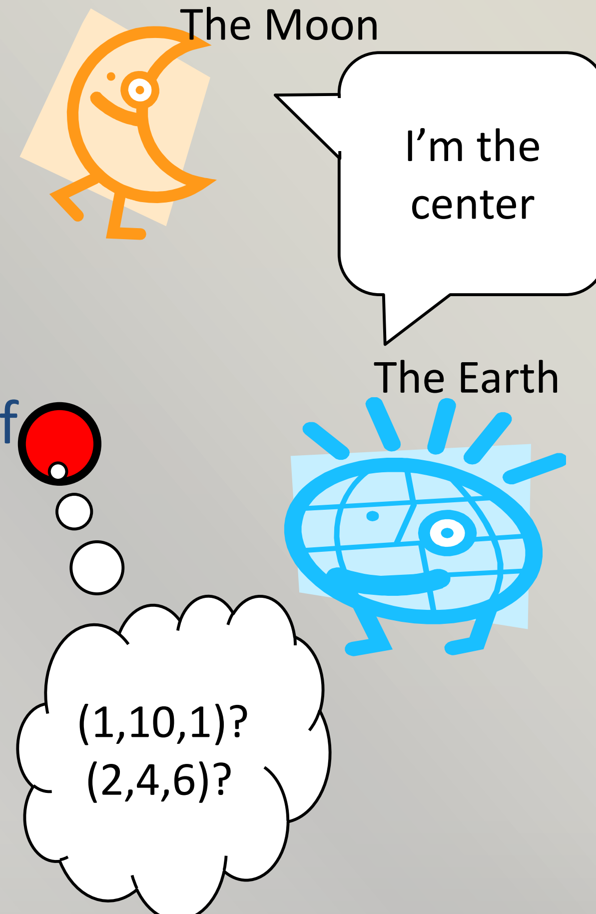
  SR ~~=~~ RS

- So be-careful, Order **DOES** Matter!!



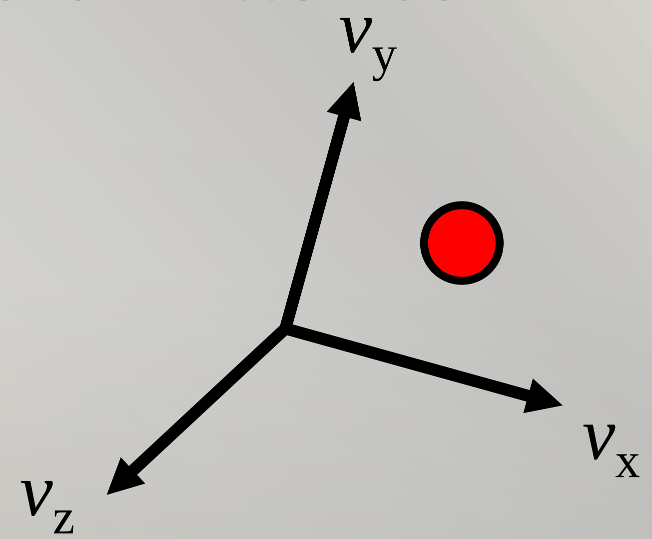
Coordinate System

- When talking about coordinate system, we will need to know our reference
  - Any point will have different coordinates if our coordinate system are different
- E.g. we see a certain position on the Earth or on Moon
  - They will both think they are the origin!



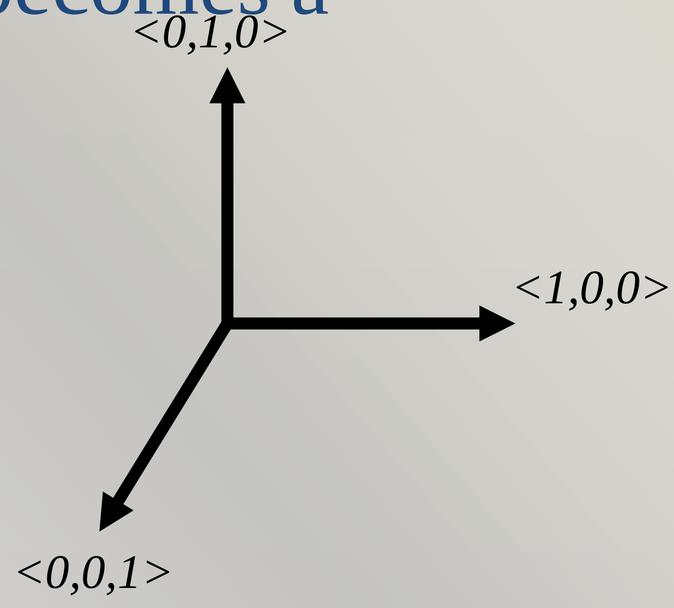
Coordinate System

- In 3D, the coordinate frame are formed by 3 basis vectors (unit vector is used)
  - They are suppose to be perpendicular to each other
  - E.g. v_{x}, v_{y}, v_{z}
- A vector defined by this basis is written as
  $v = a\, v_x + b v_y + c v_z$
- So, the coordinate becomes
  (a, b, c) in this frame



Coordinate System

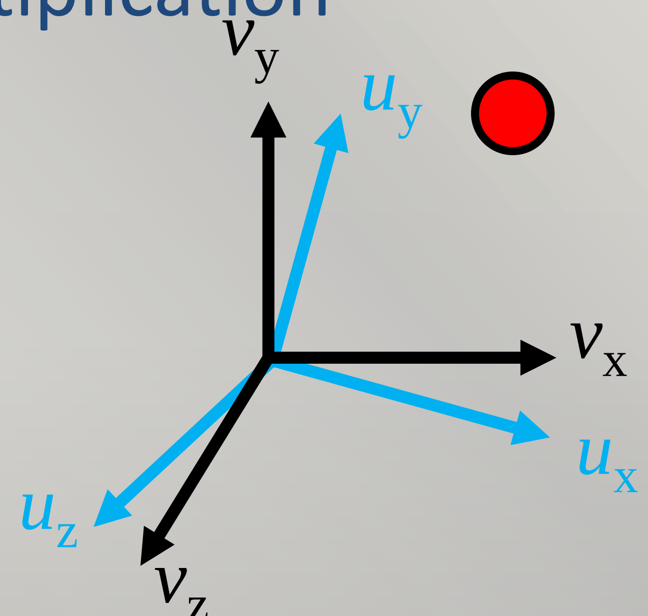
- So the conventional x,y,z axis becomes a particular case in which
  - $v_x = \langle 1,0,0 \rangle$
  - $v_y = \langle 0,1,0 \rangle$
  - $v_z = \langle 0,0,1 \rangle$



  - *Notice that all of them are unit vectors*

Transforming between Spaces

- Our interest here is how to transform a vertex's coordinate from one frame/space to another frame/space
- Again, we can use matrix multiplication



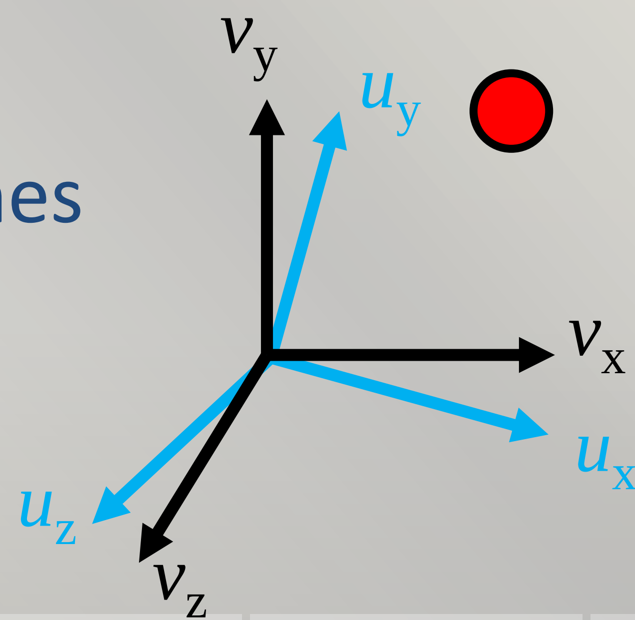
Transforming between Spaces

- First, we represent the second basis *u* in terms of the first basis *v*

$$u_x = \gamma_{11} v_x + \gamma_{12} v_y + \gamma_{13} v_z$$
$$u_y = \gamma_{21} v_x + \gamma_{22} v_y + \gamma_{23} v_z$$
$$u_z = \gamma_{31} v_x + \gamma_{32} v_y + \gamma_{33} v_z$$

- So, coordinates of basis *u* becomes

$$u_x = (\gamma_{11}, \gamma_{12}, \gamma_{13})$$
$$u_y = (\gamma_{21}, \gamma_{22}, \gamma_{23})$$
$$u_z = (\gamma_{31}, \gamma_{32}, \gamma_{33})$$

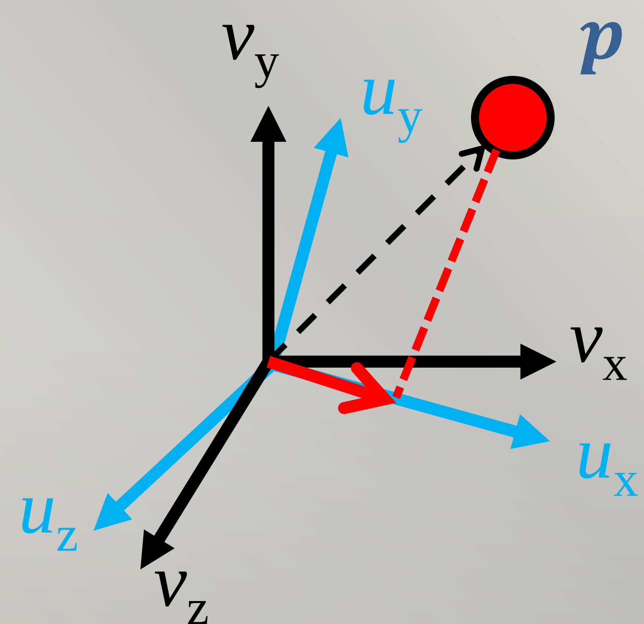


Transforming between Spaces

- Let's consider a vertex $\boldsymbol{p}$ in basis v is going to transform to basis u
- A dot product between $\boldsymbol{p}$ and u_{x}

$$\boldsymbol{u_x} \cdot \boldsymbol{p}$$

  - We obtain the length of $\boldsymbol{p}$ in this axis in new basis
  - This is also the coordinate of $\boldsymbol{p}$ in this axis!!
  - Note u_{x} is a unit vector and all computation is in basis v



- The same can be apply to other axis in the new basis

Transforming between Spaces

$$u_x = (\gamma_{11}, \gamma_{12}, \gamma_{13})$$
$$u_y = (\gamma_{21}, \gamma_{22}, \gamma_{23})$$
$$u_z = (\gamma_{31}, \gamma_{32}, \gamma_{33})$$

- $\mathbf{u}_x \cdot \mathbf{p} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$

- We have similar formulas for u_y and u_z , and combining 3 of them in Matrix format:

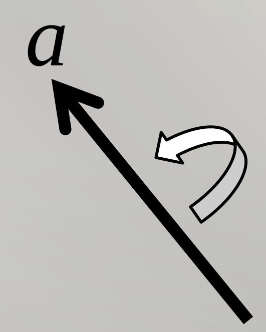
$$\begin{bmatrix} p_{ux} \\ p_{uy} \\ p_{uz} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

M (we use M to represent this transformation matrix)

The transformed coordinate of *p* in new basis *u*

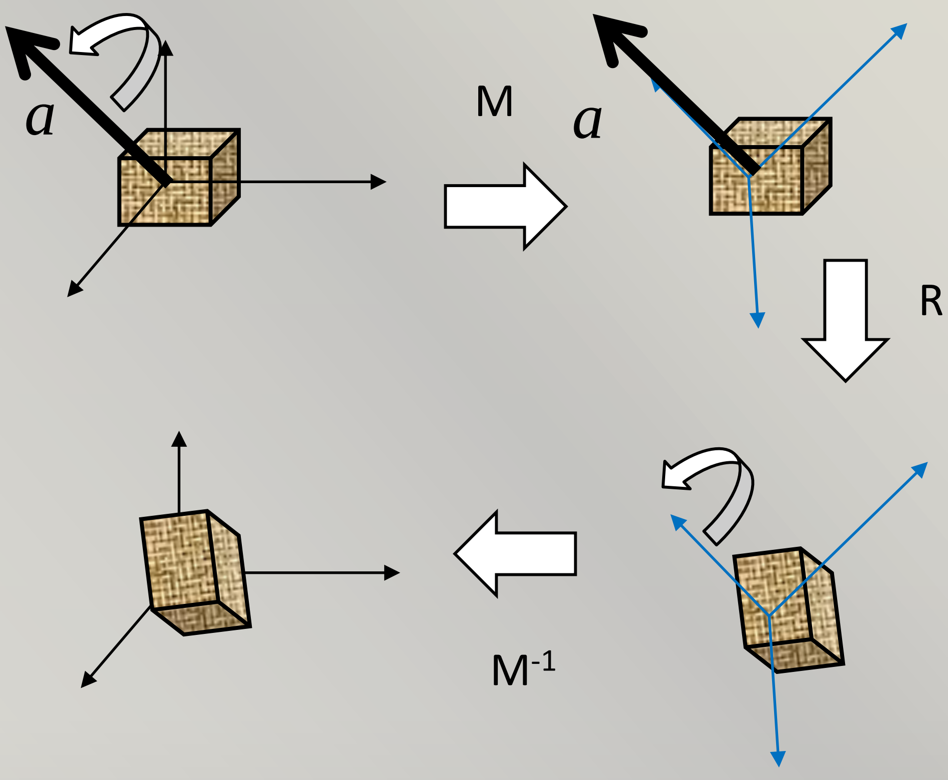
3D Rotation about arbitrary Axis

- To illustrate an application, we will discuss about the rotation with arbitrary axis
  - Suppose we want rotate about axis a in $\emptyset$ *degree*
- The idea is to make the axis coincident with one of the coordinate axes (e.g. y axis), rotate by $\emptyset$, and then transform back



3D Rotation about arbitrary Axis

- So, the method has 3 steps
  - Transform the vertex to the new coordinate frame
  - Rotate
  - Transform back to original coordinate frame



3D Rotation about arbitrary Axis

- E.g. To rotate about the vector <0.577,0.577,0.577> about 45 degree anticlockwise

$$M=\begin{bmatrix} 0.577 & 0 & -0.577 \\ 0.577 & 0.577 & 0.577 \\ 0.3329 & -0.6659 & 0.3329 \end{bmatrix}$$

Obtained by cross product with y-axis

Obtained by cross product between the above 2 vectors

$$R=\begin{bmatrix} \cos\pi/4 & -\sin\pi/4 & 0 \\ \sin\pi/4 & \cos\pi/4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M^{-1}=\begin{bmatrix} 0.577 & 0.577 & 0.3329 \\ 0 & 0.577 & -0.6659 \\ -0.577 & 0.577 & 0.3329 \end{bmatrix}$$

Transforming Space with homogenous Coordinate

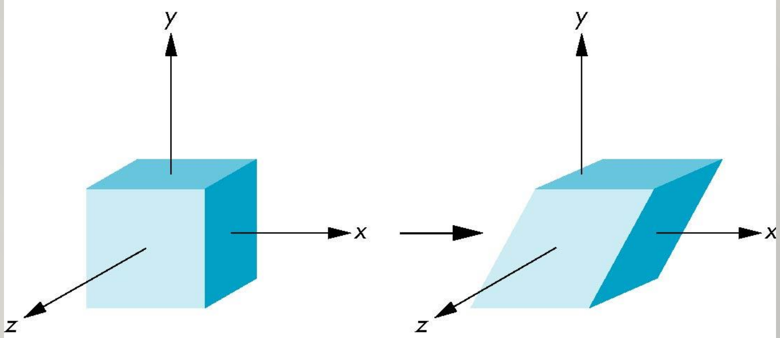
- In the last 3x3 matrix, the transformation between coordinate frame does not allow translations to happen
- Now, we have M as a 4x4 matrix:

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

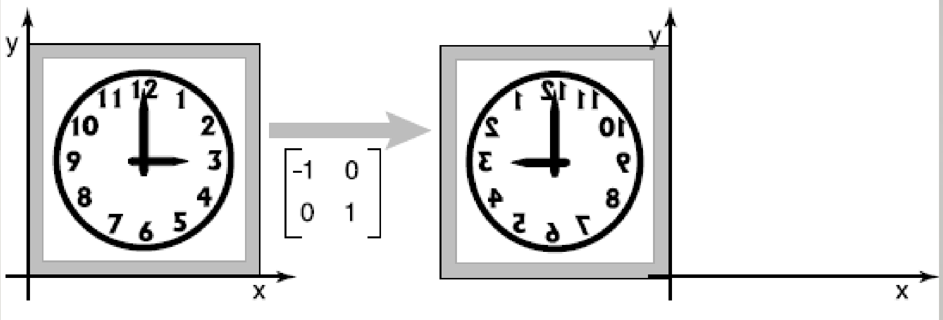
- We are not discuss in full detail here, please refer to your textbook

Other Common Transformations

- Shear: equivalent to pulling faces in opposite directions



- Reflection: inverted in horizontal or vertical direction



Summary

- Studied the 3D coordinate system and related mathematics
  - 3D vector, matrix and transformation
- Homogeneous coordinate is commonly used
- The change of coordinate system can also be achieved by matrix multiplication