Homework Assignment #1

- Will be graded!

- 'Snowflake' 2D animation
  - Understanding polygon draw.
  - Data handling
  - Vertex Shader (transformations)
  - Creativity!!

- Due: **March 30 (Wednesday)** before midnight

Lab Session Tomorrow

- Applying transformation with OpenGL
- Homework #1 help

- **Optional** but recommended to attend
- Will use both rooms (307 & 306)
- Just in case, bring your notebook computer (and a power cord)!

I. Getting Started

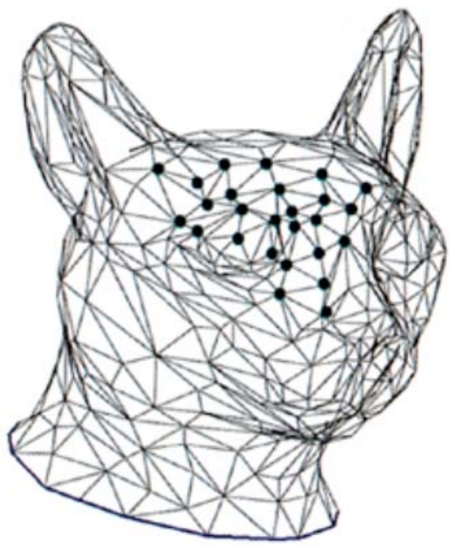
Linear and Affine Transformation

Chapters 2 & 3

Chapter 1

- Geometric object
- Triangles
- Vertices
  - Attributes for each vertex
    - Location
    - Color
    - Material property (e.g. shininess)
    - …

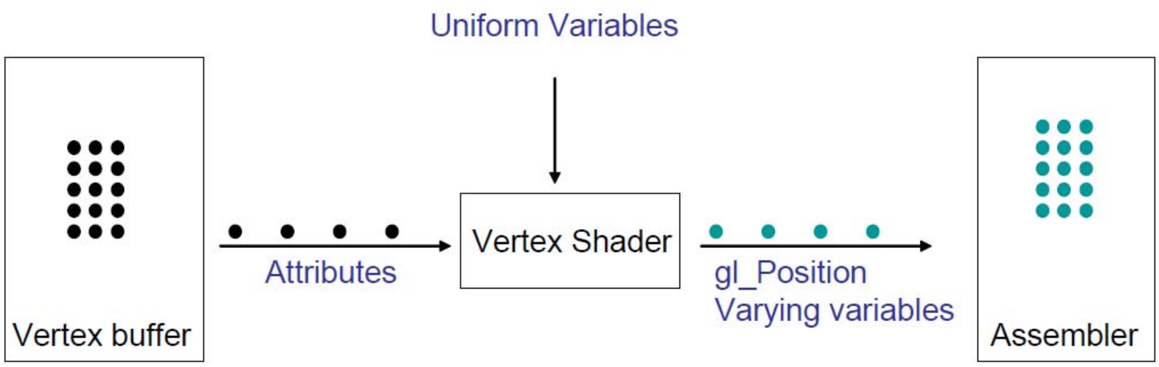
Vertex Buffer



Vertex Shader

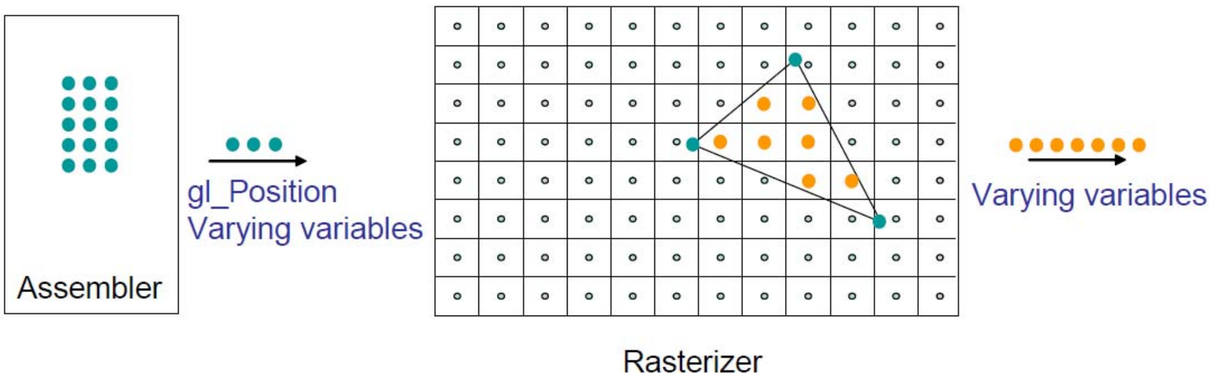
- Vertices are stored in a vertex buffer.
- When a draw call is issued, each of the vertices passes through the vertex shader.
- On input to the vertex shader, each vertex (black) has associated attributes.
- On output, each vertex (cyan) has a value for gl_Position and for its varying variables.

Uniform variables are set by your program, but you can only set them in between OpenGL draw calls and not per vertex.
(e.g., virtual camera parameters)



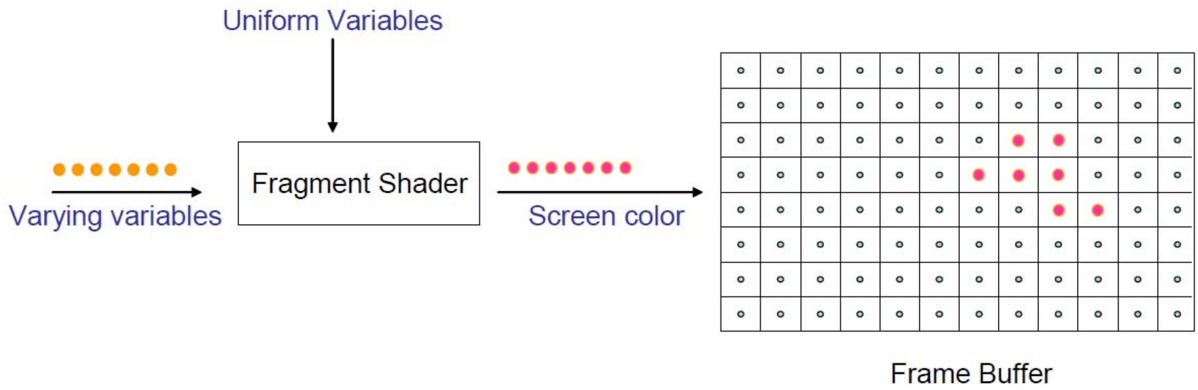
In the pipeline: Rasterization

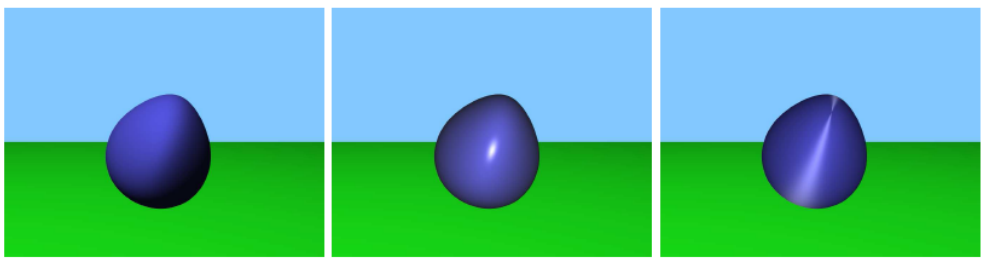
- The data in gl_Position is used to place the three vertices of the triangle on a virtual screen.
- The rasterizer figures out which pixels (orange) are *inside* the triangle and interpolates the varying variables from the vertices to each of three pixels.



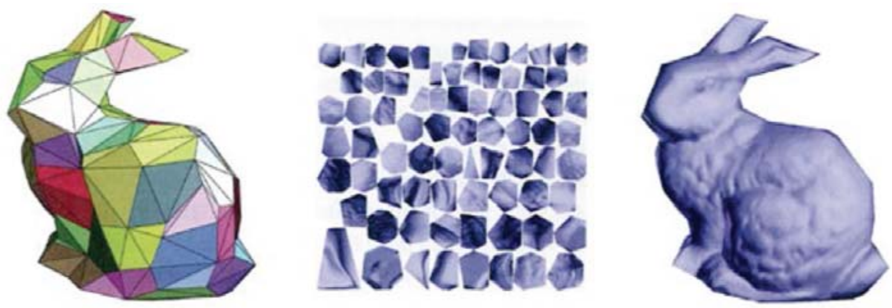
Fragment Shader

- Each pixel (orange) is passed through the fragment shader, which computes the final color of the pixel (pink).
- The pixel is then placed in the frame buffer for display.





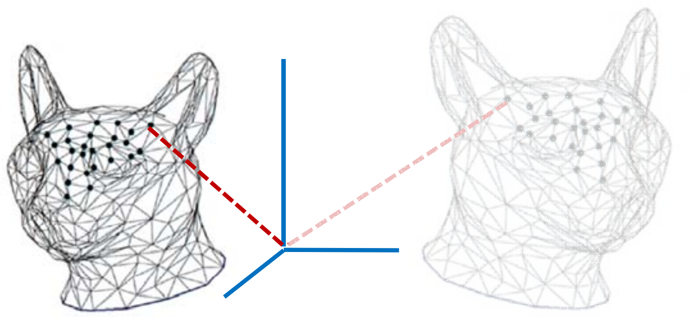
- By changing the fragment shader, we can simulate light reflecting off of different kinds of materials.



- A simple geometric object described by a small number of triangles.
- An auxiliary image called a texture.
- Parts of the texture are glued onto each triangle giving a more complicated appearance.

Chap 2. Linear
Chap 3. Affine Transformation

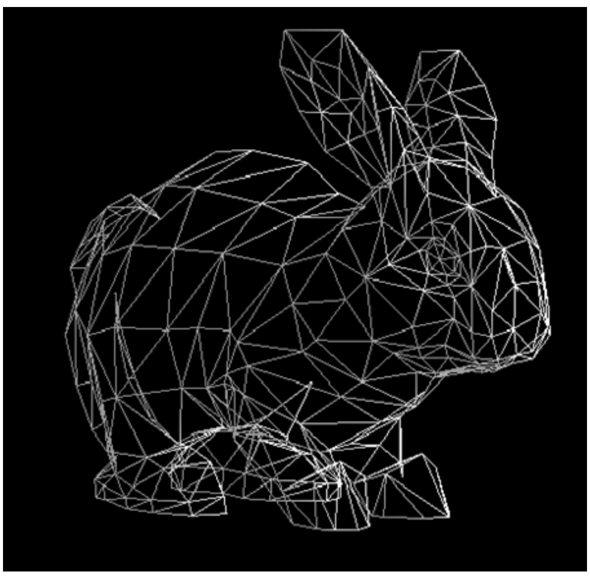
- How to represent points using coordinates
- How to perform useful geometric transformations to these points.



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

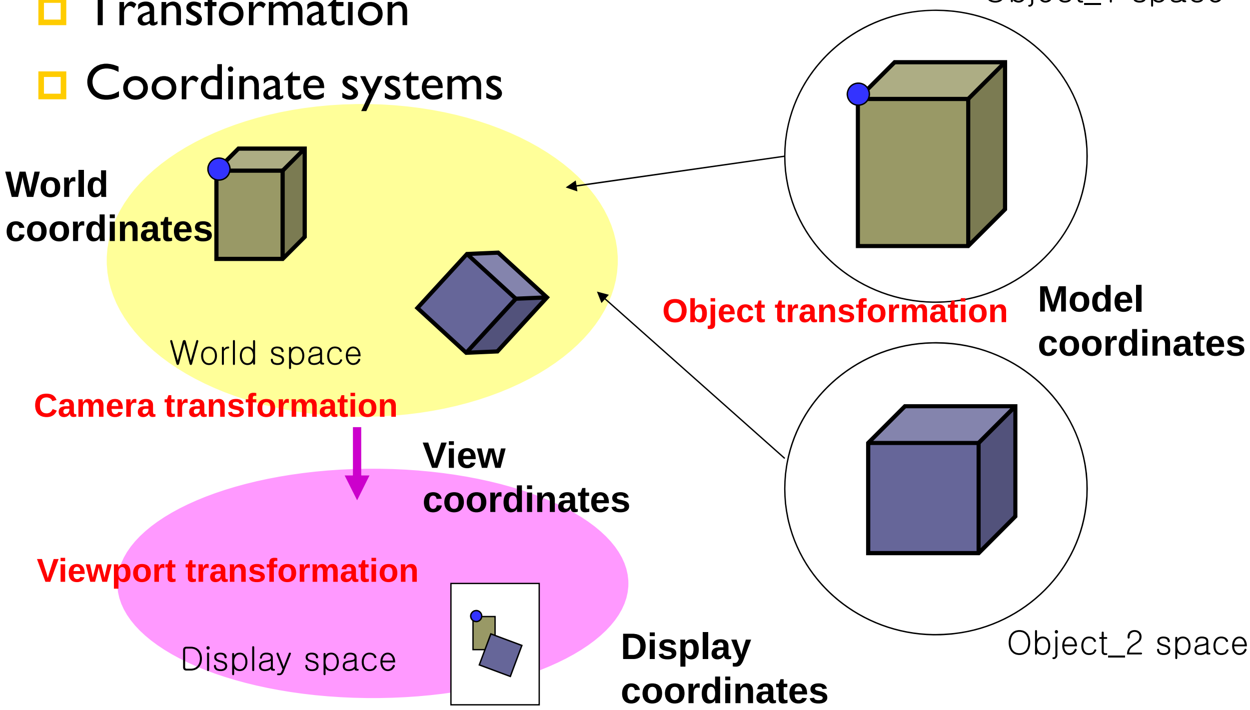
Geometric Objects

- Made of (defined by) a set of vertices (faces)
  - Usually triangles
  - Flat polygons
- Vertex has **position** information (coordinates)
- Objects
  - Point : 1 vertex
  - Triangle : 3 vertices
  - Rabbit : 251 triangles
  - …



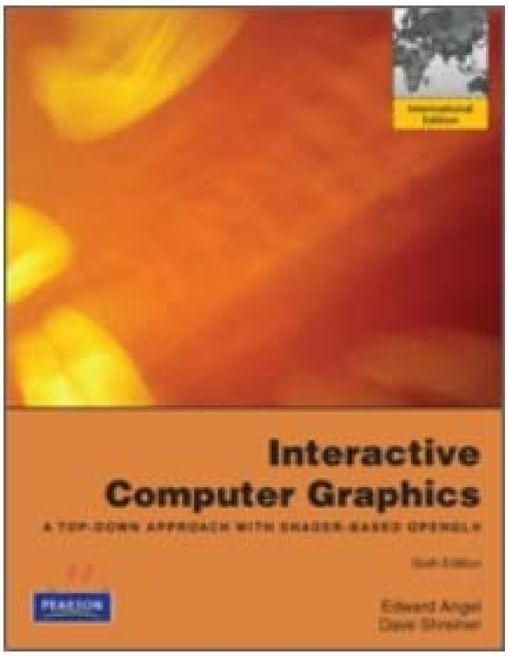
Most Essential Part

- Transformation
- Coordinate systems



Angel book: Chapter 3, Appendix

- Reference book
  - Edward Angel and Dave Shreiner
  - Interactive Computer Graphics: A Top-Down Approach with Shader-based OpenGL



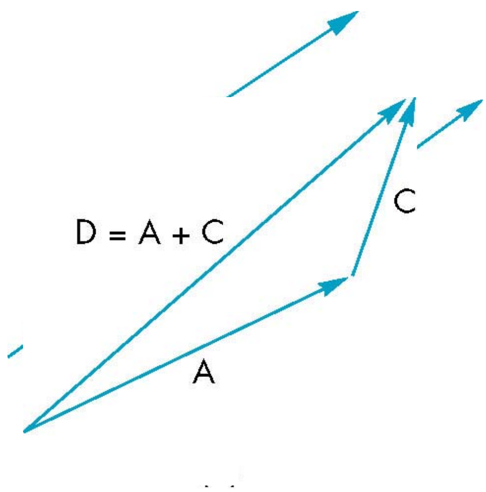
- available from KAIST library

Spaces (Mathematical view)

- (Linear) Vector space
  - Scalars (real numbers)
  - Vectors
- Affine space
  - Points
- Euclidean space
  - Concept of distance

Scalars, Points, and Vectors

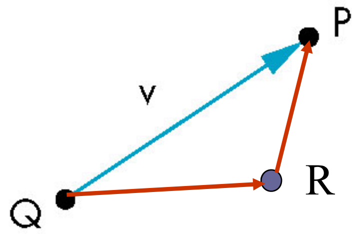
- 3 basic types needed to describe the geometric objects and their relations
  - Point: P, Q, R, …
    - a location in the space
  - Scalar: α, β, δ, ...
    - real numbers to specify the quantities
    - operations on the real numbers (addition, multiplication)
  - Vector: *u, v, w*, …
    - any quantity with direction and magnitude
    - scalar-vector multiplication
    - vector-vector addition
    - head-to-tail rule
  - **Vector space**
    - scalars & vectors



Affine Space

- Extension of the vector space that includes an additional type of object: a point
  - Addition & multiplication
  - Vector-vector addition & Scalar-vector multiplication
  - Vector-point addition (produces a new **point**)
  - Point-point subtraction (produces a **vector**)

$$v = \mathbf{P} - \mathbf{Q}$$
$$\mathbf{P} = v + \mathbf{Q}$$
$$(P - R) + (R - Q) = P - Q$$



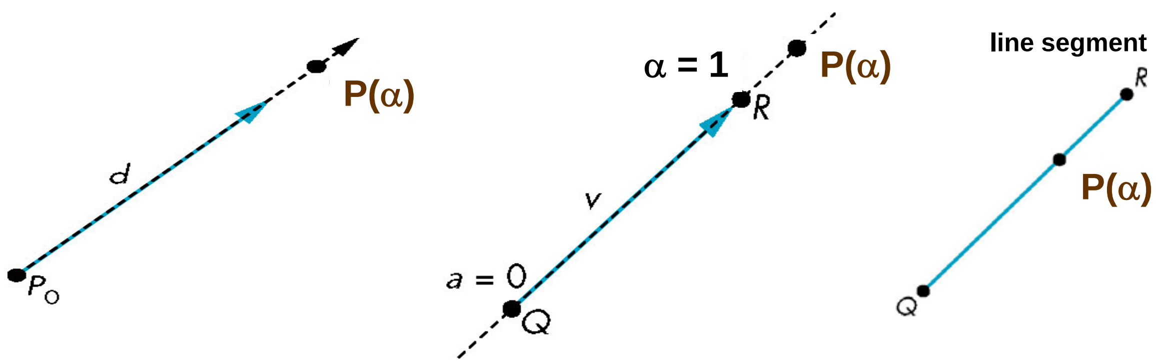
Line

- The sum of a point and a vector
- All points of the form: $P(\alpha) = P_0 + \alpha d$

a scalar that can vary over some range of value

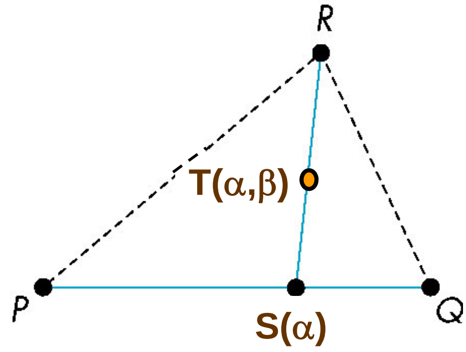
parametric form of the line

- $v = R - Q$
  $P = Q + \alpha v = Q + \alpha (R - Q) = \alpha R + (1 - \alpha)Q$
- $P = \alpha_1 R + \alpha_2 Q$ where $\alpha_1 + \alpha_2 = 1$



Plane

- Extension of the parametric line
- 3 points P, Q, R
- Line segment PQ
  - $S(\alpha) = \alpha P + (1-\alpha)Q$



- Another line segment $S_\alpha R$
  - $T(\beta) = \beta S + (1-\beta)\,R$
  - $T(\alpha,\beta) = \beta\,[\alpha P + (1-\alpha)Q] + (1-\beta)\,R$

$$+ (P - P + \beta P - \beta P)$$

$$= P + \beta\,(1-\alpha)\,(Q - P) + (1-\beta)\,(R - P)$$

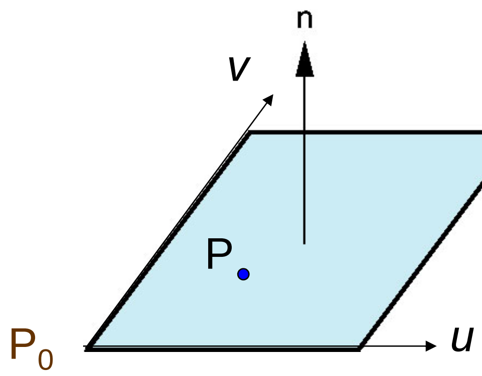
- All points $T(\alpha,\beta)$ lie in a plane

Plane

- A point and two nonparallel vectors
  - $T(\alpha,\beta) = P_0 + \alpha u + \beta v$
  - $P - P_0 = \alpha u + \beta v$

- $n \cdot (P - P_0) = 0$ where $n = u \times v$

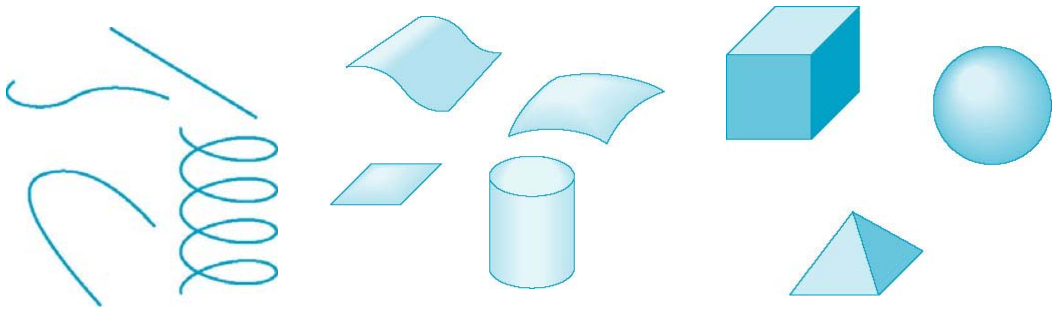
normal vector



3D Primitives

- 2D
  - Simple curves (line segments)
  - Objects with interior (polygon)
- 3D
  - Curves in space
  - Surfaces
  - Volumetric objects

Mathematical definition?
Efficient implementations?



3D Primitives

- To be a good match with the current graphics hardware:
  - The objects are described by their surfaces (hollow)
  - The objects can be specified by vertices
  - The objects are composed of flat convex polygons

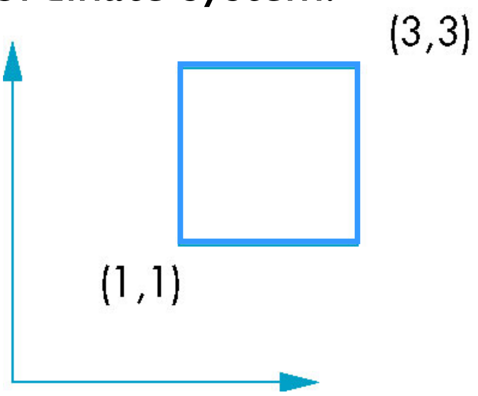
  - Arbitrary polygons are tessellated into triangular polygons

- All the primitives with which we work can be specified through a set of vertices.

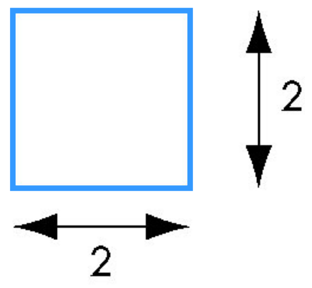
    - Exception: Constructive solid geometry (GSG)
    - New approach: Voxel-based

Coordinate-free geometry

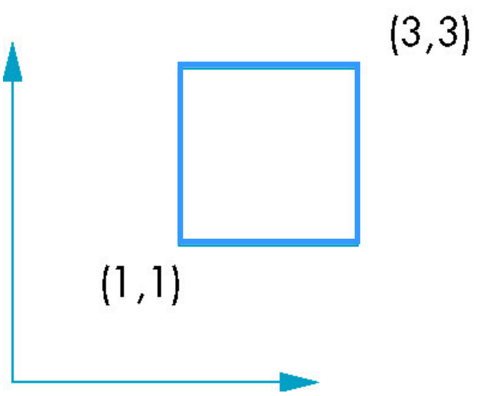
- Points exist in space regardless of any reference or coordinate system.

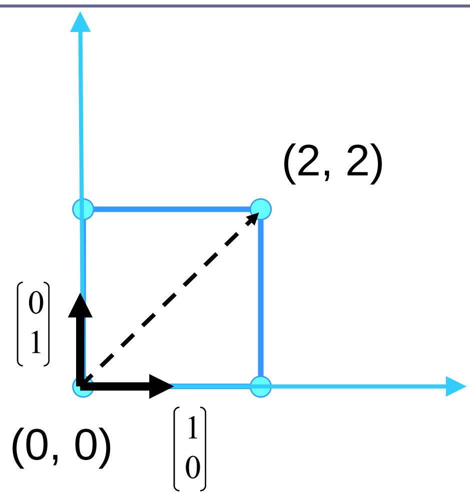


Object and
coordinate system



Object without
coordinate system

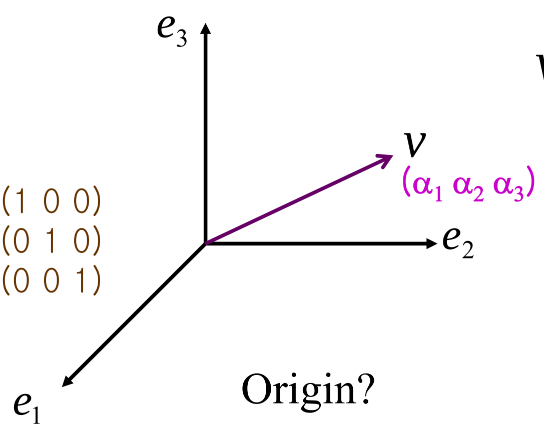




Coordinate Systems …

- Coordinate system
  - defined by the basis vectors
- Representation of a vector in 3D space
  - with any three linearly independent vectors

$e_1^T = (1\ 0\ 0)$
$e_2^T = (0\ 1\ 0)$
$e_3^T = (0\ 0\ 1)$



$$v = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3$$

Basis (vector)
Components (scalar)

… and Frames

- Representation requires both the reference point and the basis vectors:
  <u>a frame</u>
- Vector: $w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$
  - Need to know 3 scalars
- Point: $P = P_0 + \eta_1 v_1 + \eta_2 v_2 + \eta_3 v_3$
  - Need to know 3 scalars and the location of the origin

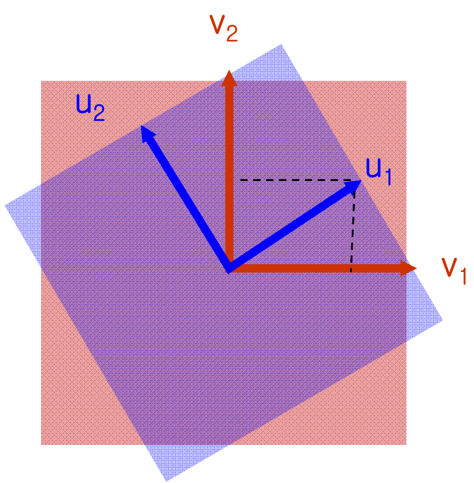
Change of coordinate systems

- Two bases $\{v_1, v_2, v_3\}$, $\{u_1, u_2, u_3\}$
- Each basis vector (u_i) can be represented in terms of the other basis (v_j)

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

Change of coordinate systems

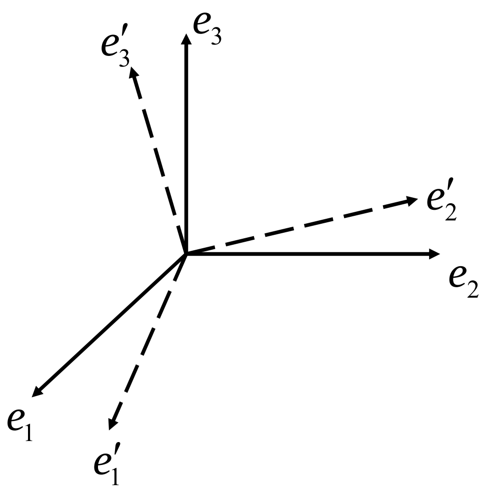
- Two bases $\{v_1, v_2, v_3\}$, $\{u_1, u_2, u_3\}$



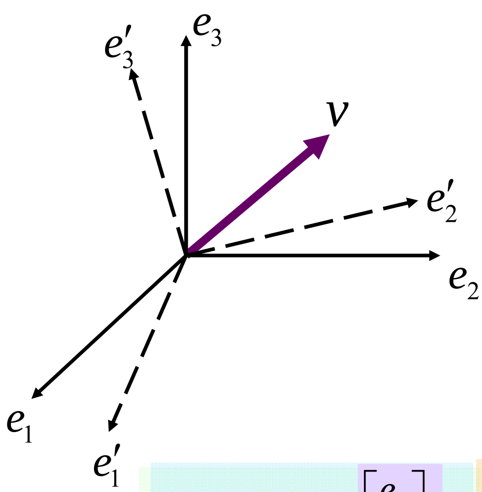
$$u_1 = \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3,$$
$$u_2 = \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3,$$
$$u_3 = \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3.$$

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

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



$$\begin{array}{ccc} \text{U} & & \text{V} \\ \begin{bmatrix} e_1' \\ e_2' \\ e_3' \end{bmatrix} & = \mathrm{M} & \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \end{array}$$



Arbitrary vector, v

$$v = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

same vector v with respect to $e_1' - e_2' - e_3'$;

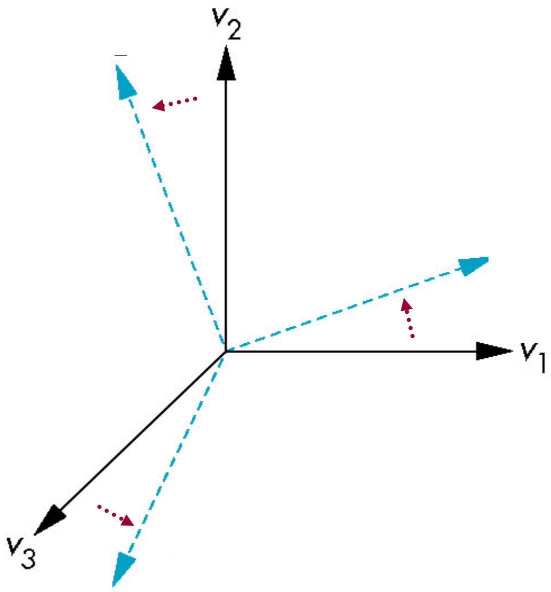
$$v = \begin{bmatrix} \alpha_1' & \alpha_2' & \alpha_3' \end{bmatrix} \begin{bmatrix} e_1' \\ e_2' \\ e_3' \end{bmatrix} \qquad \begin{bmatrix} e_1' \\ e_2' \\ e_3' \end{bmatrix} = \mathrm{M} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \alpha_1' & \alpha_2' & \alpha_3' \end{bmatrix} \begin{bmatrix} e_1' \\ e_2' \\ e_3' \end{bmatrix} = \begin{bmatrix} \alpha_1' & \alpha_2' & \alpha_3' \end{bmatrix} M \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_1' \\ \alpha_2' \\ \alpha_3' \end{bmatrix} = \left(M^T\right)^{-1} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

What is M?

Rotation and Scaling



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

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

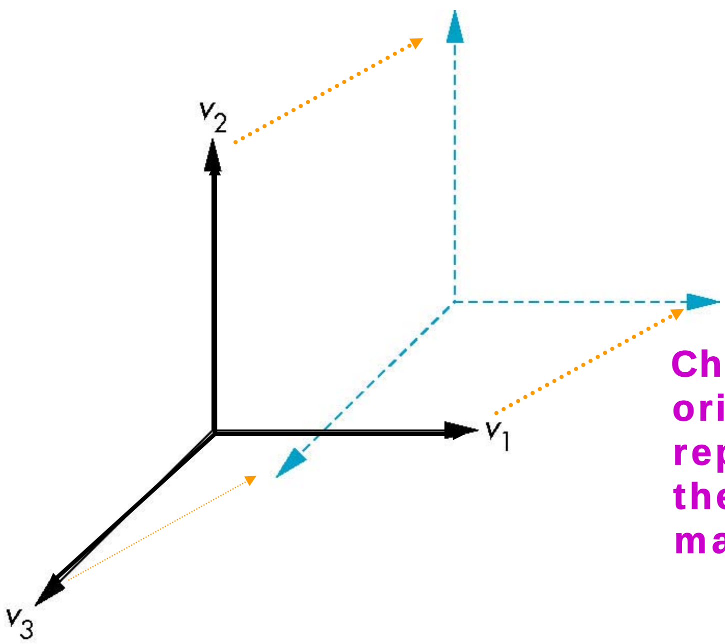
$$\begin{bmatrix} \alpha_1' \\ \alpha_2' \\ \alpha_3' \end{bmatrix} = \left(M^T\right)^{-1} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

U V

Translation

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

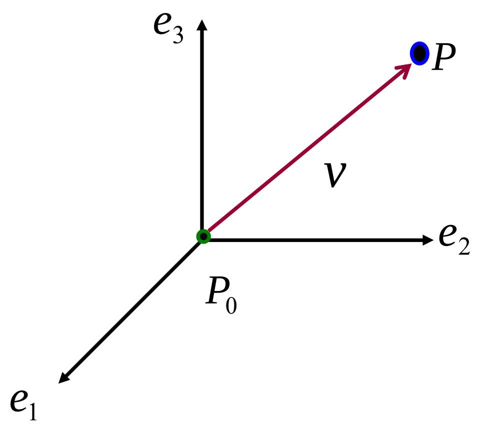
$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$



Change of the origin cannot be represented in the previous 3x3 matrix

Homogenous Coordinates

Frame



$$\text{vector} \quad v = \sum \alpha_i e_i$$

$$\text{point} \quad P = P_0 + \sum \alpha_i e_i$$

Homogenous Coordinates

$$\text{point P} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \quad \text{vector} \quad v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

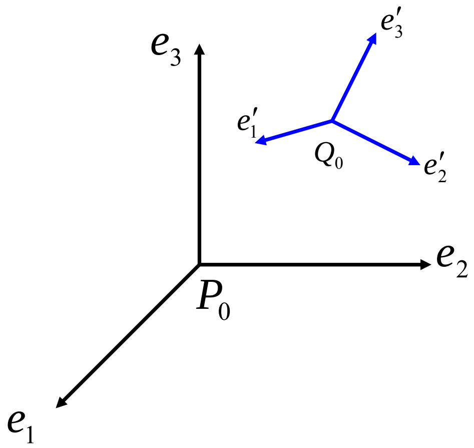
homogenous-coordinate representation of the point

$$\text{point} \quad P = P_0 + \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 = \boxed{\begin{bmatrix} \alpha_1 & \alpha_2 & \boldsymbol{\alpha}_3 & 1 \end{bmatrix}} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ P_0 \end{bmatrix}$$

$$\text{equivalently}, \quad P = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 1 \end{bmatrix}, \quad v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ 0 \end{bmatrix}$$

$$u_1 = \gamma_{11} v_1 + \gamma_{12} v_2 + \gamma_{13} v_3 + 0$$
$$u_2 = \gamma_{21} v_1 + \gamma_{22} v_2 + \gamma_{23} v_3 + 0$$
$$u_3 = \gamma_{31} v_1 + \gamma_{32} v_2 + \gamma_{33} v_3 + 0$$
$$Q_0 = \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 + P_0$$

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



M

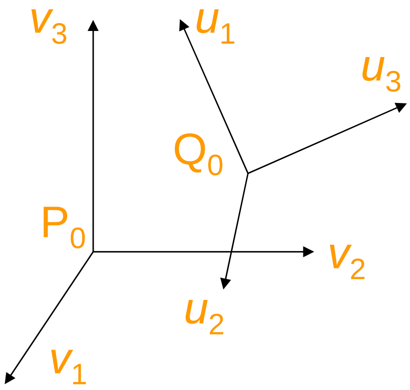
$$\begin{bmatrix} e'_1 \\ e'_2 \\ e'_3 \\ Q_0 \end{bmatrix} = \begin{bmatrix} & \gamma_{ij} & & 0 \\ & & & 0 \\ & & & 0 \\ \beta_1 & \beta_2 & \beta_3 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ P_0 \end{bmatrix}$$

Coordinate Systems & Frames

- Coordinate system:
  defined by the basis vectors
- Frame: basis vectors + reference point
  - Vector: $w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$
  - Point: $P = P_0 + \eta_1 v_1 + \eta_2 v_2 + \eta_3 v_3$
- Homogenous coordinates
  - Vector: $[\ \alpha_1, \alpha_2, \alpha_3, 0\ ]^T$
  - Point: $[\ \eta_1, \eta_2, \eta_3, 1\ ]^T$

Change of Frames

- Two frames $\{v_1, v_2, v_3, P_0\}$, $\{u_1, u_2, u_3, Q_0\}$



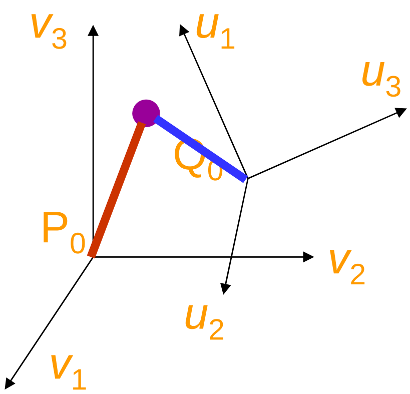
$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

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

Change of Frames

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

- Two frames $\{v_1, v_2, v_3, P_0\}$, $\{u_1, u_2, u_3, Q_0\}$



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

$$\mathbf{b}^T \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{b}^T \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = \mathbf{a}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

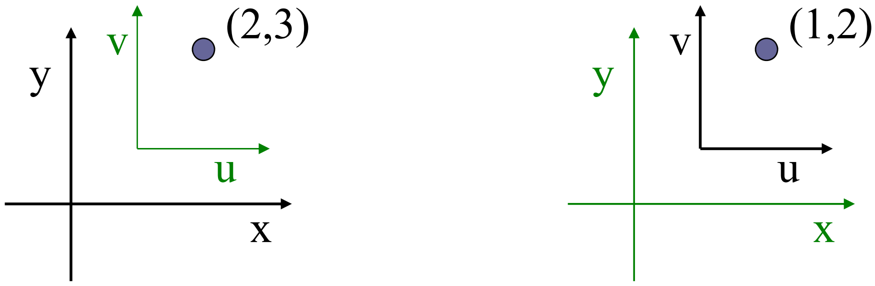
$$\boxed{\mathbf{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$\mathbf{a} = \mathbf{M}^T \mathbf{b}$$

$$\mathbf{b} = \underline{\mathbf{A}}\mathbf{a} = (\mathbf{M}^T)^{-1} \mathbf{a}$$

Coordinate Systems

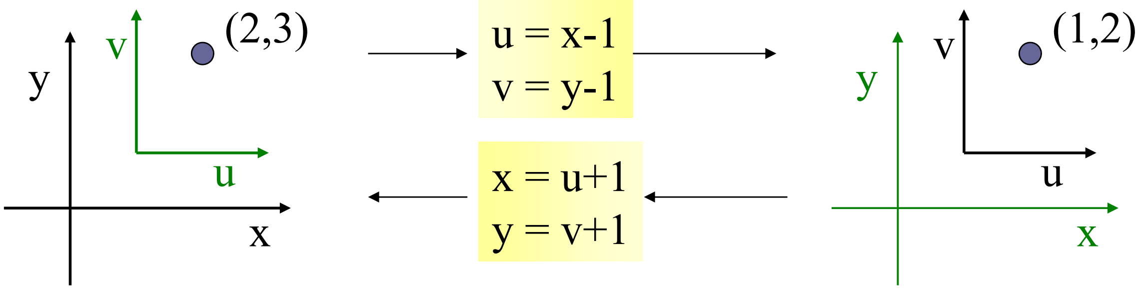
- Different coordinate systems represent the same point in different ways



  - Some operations are easier in one coordinate system than in another

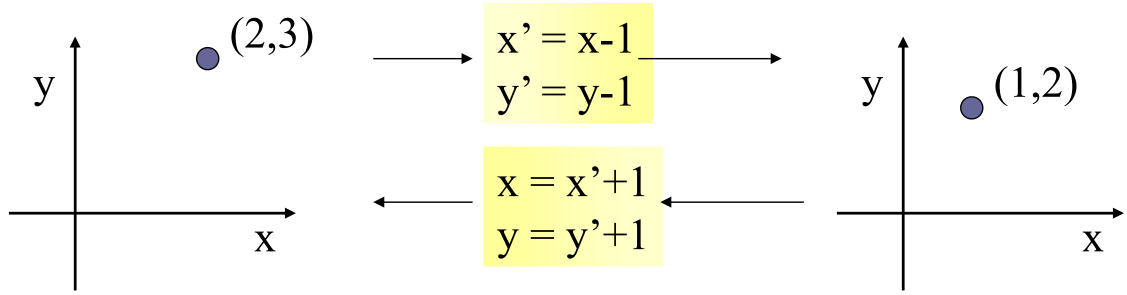
Transformation

- Transformations convert points between coordinate systems



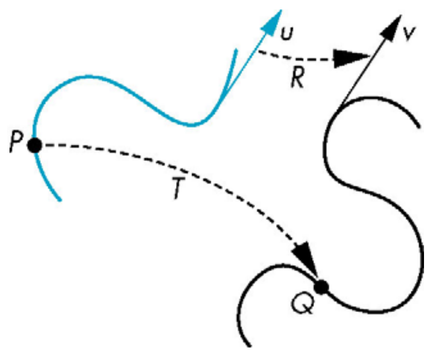
Transformation (alternative interpretation)

- Transformations modify an object's shape and location in one coordinate system



Transformation

- A transformation is a function that takes a point (or vector) and maps that point (or vector) into another point (or vector).
  - Q = T (P) ; *v* = R (*u*)



- With homogenous coord representations
  - **q** = f(**p**); **v** = f(**u**)
  - Single function f
  - Transforms points/vectors **in a given frame**

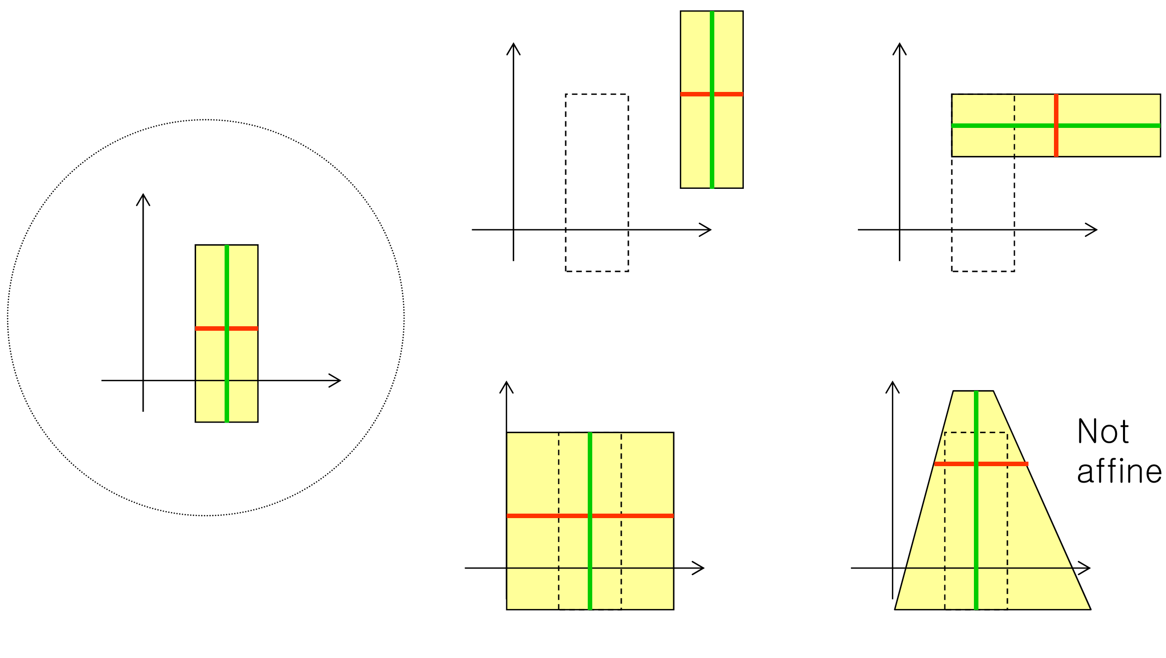
Affine Transformation

- Any transformation preserving
  - Colinearity
    - All points lying on a line initially still lie on a line after transformation
  - Ratio of distances
    - The midpoint of a line segment remains the midpoint after transformation

$$P' = f(P)$$

$$P' = f\underbrace{(\alpha P_1 + \beta P_2)}_{\text{linear combination of vertices}} = \underbrace{\alpha f(P_1) + \beta f(P_2)}_{\text{linear combination of transformed vertices}}$$

Affine Transformation



Affine Transformation

- Matrix representation

$P' = f(P) \Rightarrow P' = T\,P$

$v' = f(v) \Rightarrow v' = T\,v$

where,

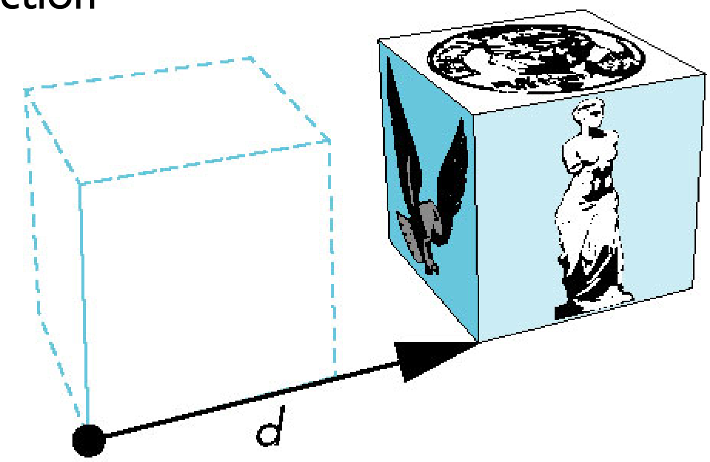
$$T = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation, Translation, Scaling

- Object transformation
  - Moving to new positions a group of points that describes one or more geometric objects
  - With a single transformation
  - Preserving the relationships among the vertices of the object

Translation

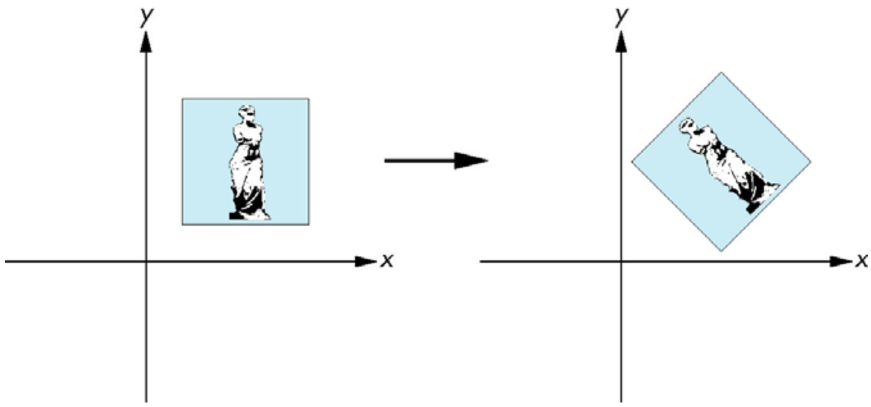
- An operation that displaces points by a fixed distance in a given direction
- P' = P + d



- 3 degrees of freedom

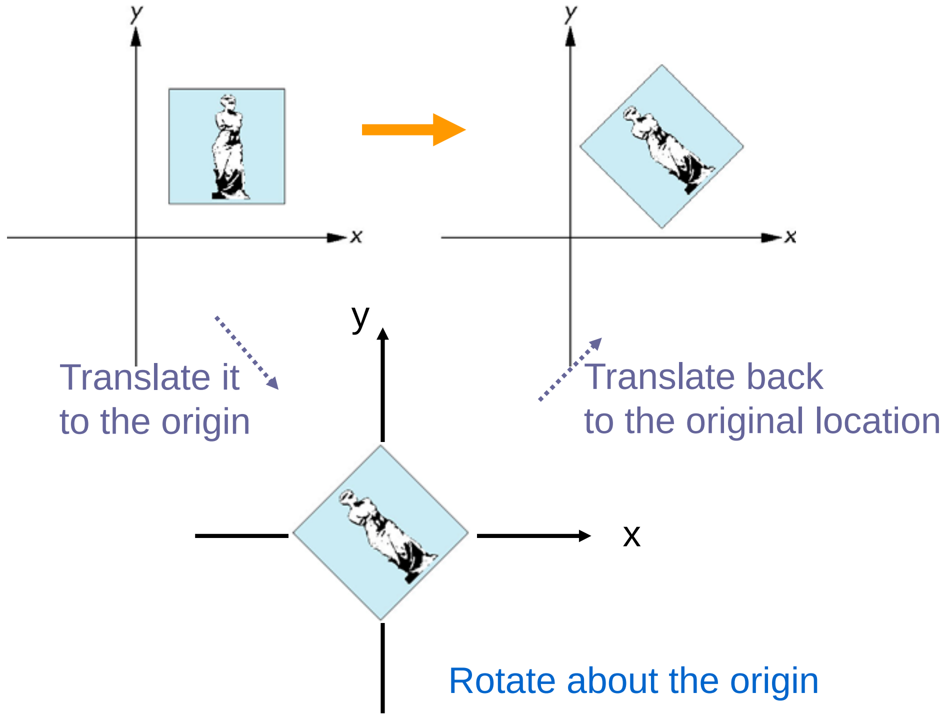
Rotation

- 2D Rotation
- Rotation about a fixed point



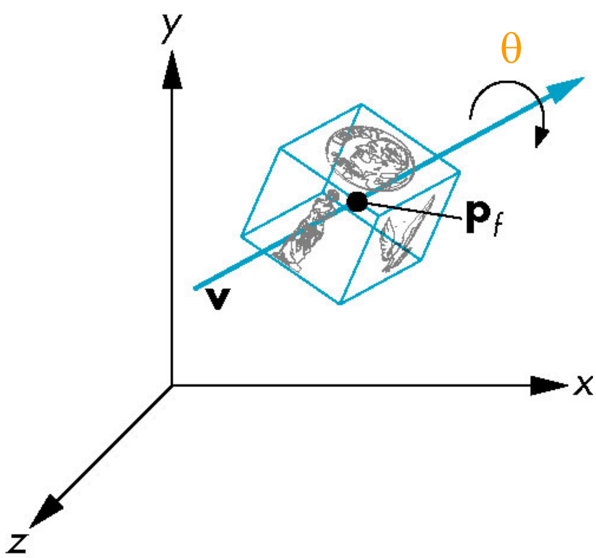
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotation



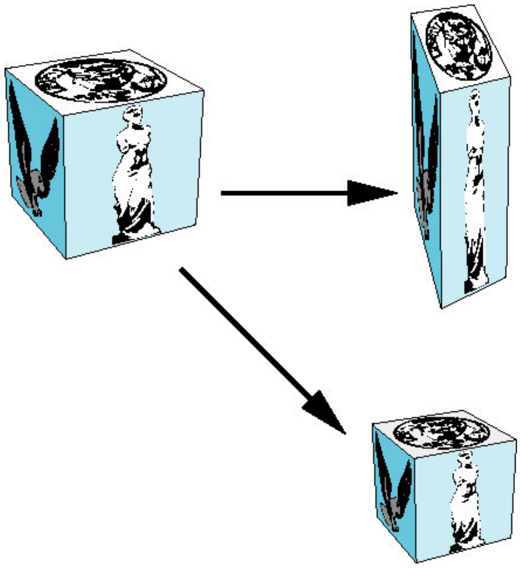
Rotation

- 3D rotation
  - a fixed point P_f
  - a rotation angle θ
  - a vector **v** about which to rotate
  - For given fixed point, there are 3 degrees of freedom



Scaling

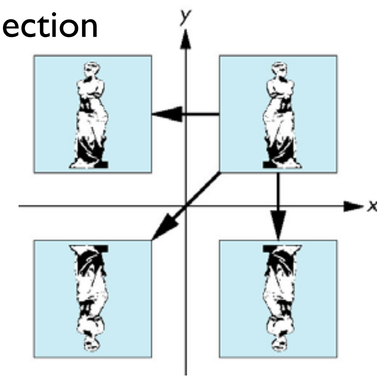
- Affine non-rigid-body transformation
  - Uniform
  - Nonuniform

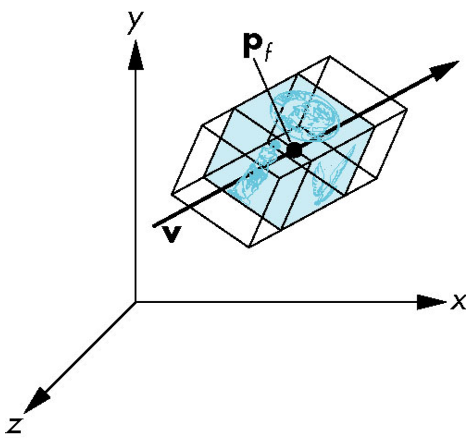


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \mathrm{Sx} & 0 \\ 0 & \mathrm{Sy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scaling

- A fixed point
- A direction in which we wish to scale
- A scale factor
  - >1 : longer
  - Between 0 and 1: smaller
  - <0 : reflection





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

Transformations in Homogeneous Coordinates

- 4x4 matrix

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation

$$p' = p + d \qquad p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \quad d = \begin{bmatrix} dx \\ dy \\ dz \\ 0 \end{bmatrix}$$

$$p' = Tp \qquad T = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -dx \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

- **p' = S p**

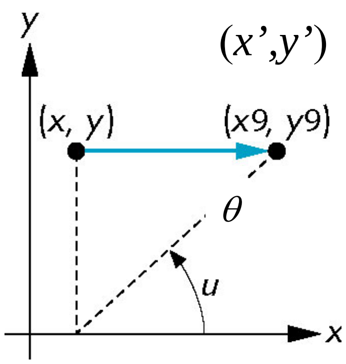
$$S = S(Sx, Sy, Sz) = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

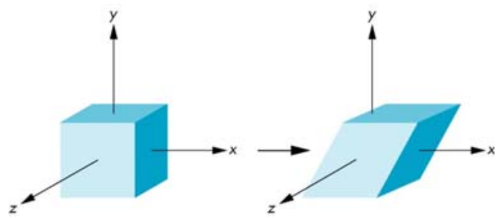
- **Inverse**

$$S^{-1} = (Sx, Sy, Sz) = S(\tfrac{1}{Sx}, \tfrac{1}{Sy}, \tfrac{1}{Sz})$$

Shear

(x', y')





$$x' = x + y\cot\theta$$
$$y' = y$$
$$z' = z$$

$$H_{xy}(\theta) = \begin{bmatrix} 1 & \cot\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation

- With a fixed point at the origin
- About the coordinate axes

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

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

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

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

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

orthogonal matrix

Concatenation of Transformations

- Arbitrary rotation at an origin
  - $R = R_z R_y R_x$
- $q = C\,B\,A\,p = C\,(B\,(A\,p))$

$$p \longrightarrow \boxed{A} \longrightarrow \boxed{B} \longrightarrow \boxed{C} \longrightarrow q$$

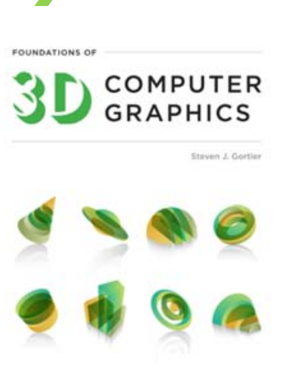
- $q = M\,p$, where $M = C\,B\,A$

$$p \longrightarrow \boxed{M\,(= C\,B\,A)} \longrightarrow q$$

(so far …) Transformation

- Coordinate systems
- Change of frames
- Homogenous coordinates
- Affine Transformation
  - Translation
  - Scaling
  - Shearing
  - Rotation
- Concatenation of transformation

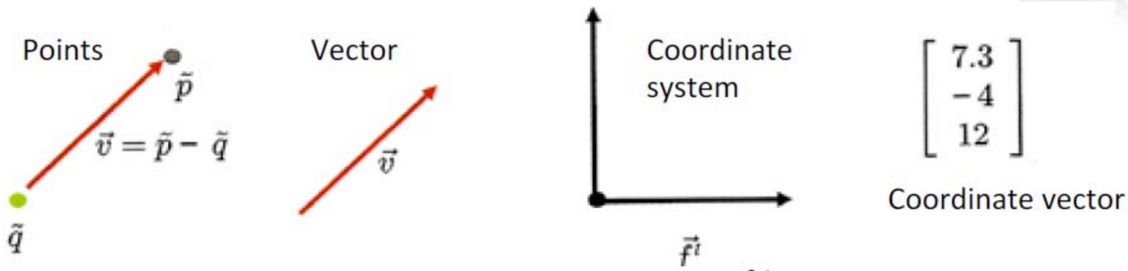
- 3D rotation
  - about a fixed point
  - about an arbitrary axis



Let's get to our textbook

Point vs. Coordinate Vector

Slide from Prof. MH Kim



1. Point (geometric object): notated as $\tilde{p}$(tilde above the letter), non-numerical object.
2. Vector (motion): notated as $\vec{v}$(arrow ab non-numerical object.

Geometric entity represents motion between two points in the world

3. Coordinate system: denoted as $\vec{\mathbf{f}}^t$ (bold: column vector, t makes it transpose), non-numerical object basis for vector; frame for point
4. Coordinate vector: noted as $\mathbf{c}$ (bold letter), numerical object