

---

# CIS 454 Computer Graphics

## Lecture 11, 10/12/2006

---

Li Shen  
Computer and Information Science  
UMass Dartmouth

---

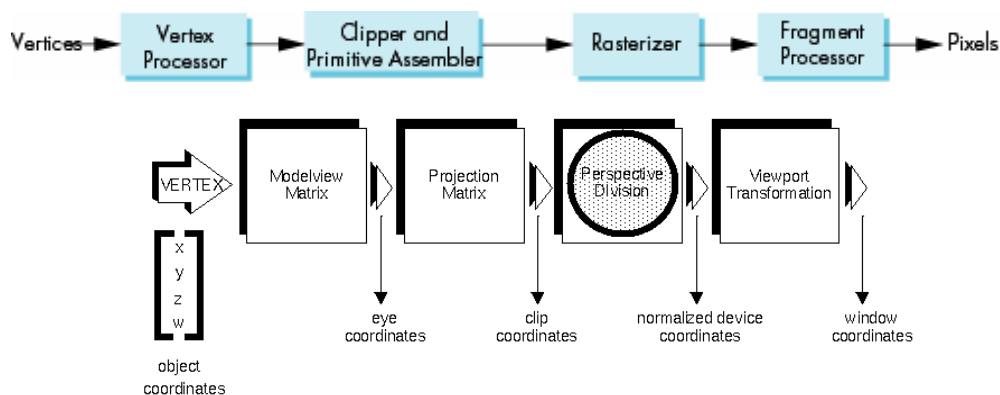
## Notes

- HW2 due next Tuesday
    - Problem 2-5. (10 points) Angel, Exercise 4.8, page 229. Derive the shear transformation from the rotation, translation, and scaling.
-

# Input and Interaction

- Introduction
  - Input devices: physical vs logical
  - Input Modes: Event-driven input
  - Smooth animations: double buffering
  - Programming event input with GLUT
- Working with Callbacks
  - Mouse, keyboard, reshape
  - Menus
- Better interactive programs
  - Picking: select objects from the display
  - Rubberbanding: interactive drawing of lines and rectangles
  - Display Lists: retained mode graphics

## Next Topic: Geometric Objects and Transformations



- Geometry: scalars, points, vectors
- Representation: coordinate systems and frames
- Transformation: rotation, translation, scaling, shear
- OpenGL transformation: `matrix_mode`
- Building models: polygonal models in OpenGL

[gasket3](#)

---

# Geometry

---

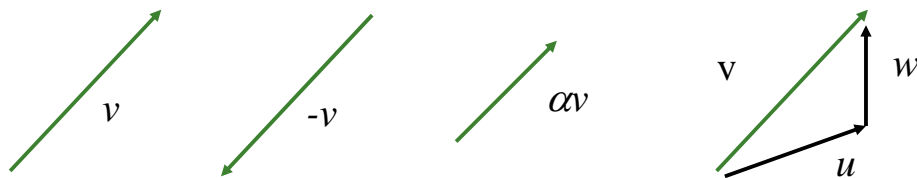
---

## Scalars

- Need three basic elements in geometry
    - Scalars, Vectors, Points
  - Scalars can be defined as members of sets which can be combined by two operations (addition and multiplication) obeying some fundamental axioms (associativity, commutativity, distributivity)
  - Examples include the real and complex number systems under the ordinary rules with which we are familiar
  - Scalars alone have no geometric properties
-

# Vectors

- Vector
  - Quantity with two attributes: direction, magnitude
- Vector operations
  - Inverse: same magnitude but points in opposite direction
  - Every vector can be multiplied by a scalar
  - Zero vector: zero magnitude, undefined orientation
  - Vector sum: use head-to-tail axiom



# Linear Vector Spaces

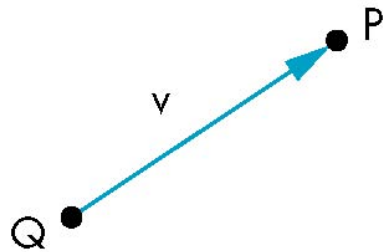
- Mathematical system for manipulating vectors
- Operations
  - Scalar-vector multiplication  $u = \alpha v$
  - Vector-vector addition:  $w = u + v$
- Expressions such as

$$v = u + 2w - 3r$$

Make sense in a vector space

## Points

- 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

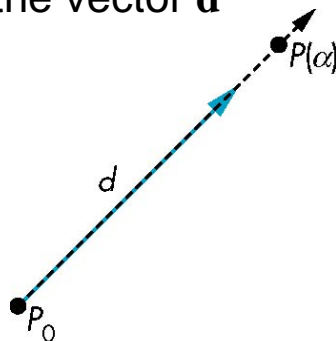
- Point + a vector space
- Operations
  - Vector-vector addition
  - Scalar-vector multiplication
  - Point-vector addition
  - Scalar-scalar operations
- For any point define
  - $1 \cdot P = P$
  - $0 \cdot P = \mathbf{0}$  (zero vector)

## Geometry Outline

- Introduce the elements of geometry
  - Scalars
  - Vectors
  - Points
- Define basic primitives
  - Line segments
  - Polygons

## Lines: Parametric Form

- 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: Parametric Form

- Two-dimensional forms

- Explicit:  $y = mx + h$
- Implicit:  $ax + by + c = 0$
- Parametric:

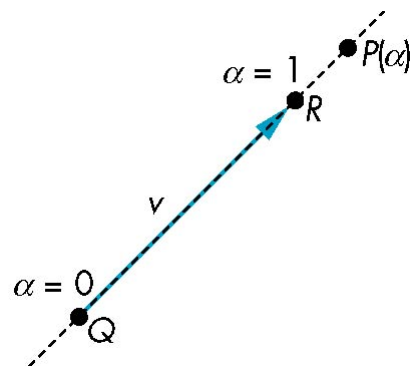
$$x(\alpha) = \alpha x_0 + (1-\alpha)x_1$$

$$y(\alpha) = \alpha y_0 + (1-\alpha)y_1$$

## Rays and Line Segments

- If  $\alpha \geq 0$ , then  $P(\alpha)$  is the *ray* leaving  $P_0$  in the direction  $\mathbf{d}$
- If we use two points to define  $\mathbf{v}$ , then
 
$$P(\alpha) = Q + \alpha \mathbf{v} = Q + \alpha (\mathbf{R} - \mathbf{Q})$$

$$= \alpha \mathbf{R} + (1-\alpha)\mathbf{Q}$$
- For  $0 \leq \alpha \leq 1$  we get all the points on the *line segment* joining  $\mathbf{R}$  and  $\mathbf{Q}$



## Affine Sums

- Consider the “sum”

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

- Can show by induction that this sum makes sense iff

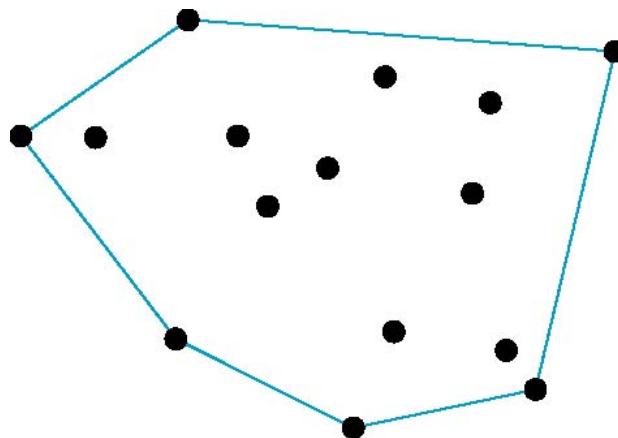
$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$$

in which case we have the *affine sum* of the points  $P_1, P_2, \dots, P_n$

- If, in addition,  $\alpha_i \geq 0$ , we have the *convex hull* of  $P_1, P_2, \dots, P_n$

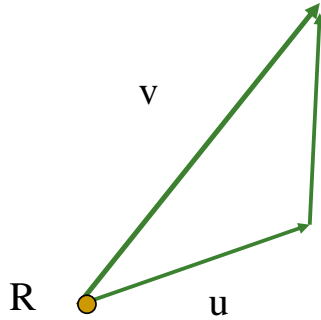
## Convex Hull

- Smallest convex object containing  $P_1, P_2, \dots, P_n$
- Formed by “shrink wrapping” points

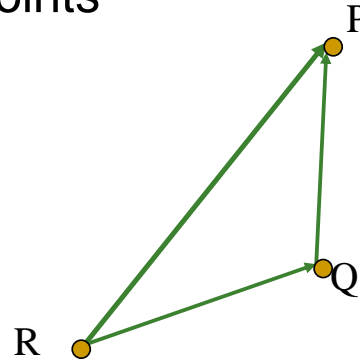


## Planes

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

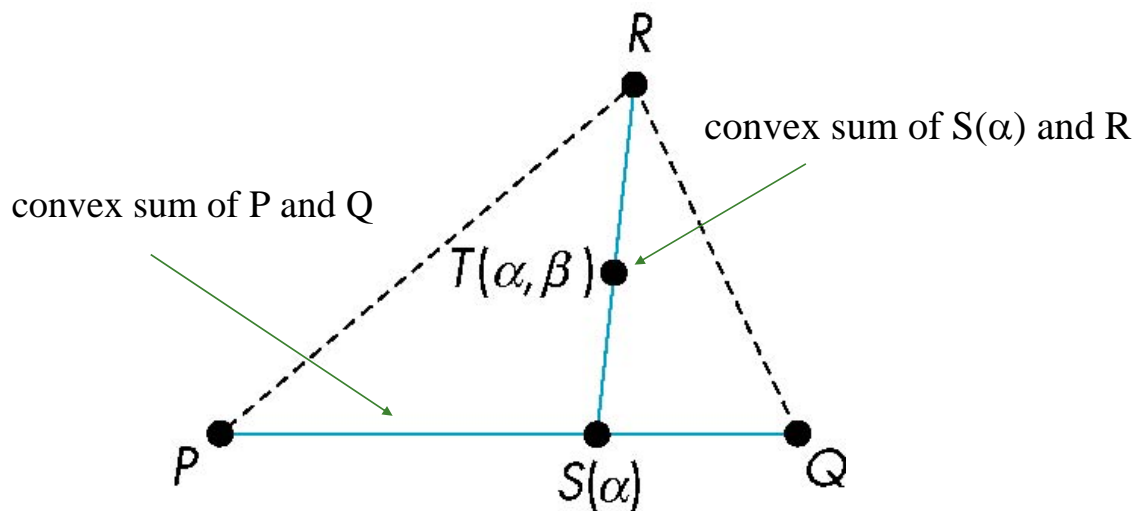


$$P(\alpha, \beta) = R + \alpha u + \beta v$$



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

## Triangles



for  $0 \leq \alpha, \beta \leq 1$ , we get all points in triangle

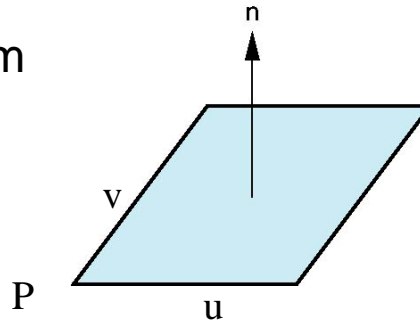
## Normals

- Every plane has a vector  $n$  normal (perpendicular, orthogonal) to it
- From *point - two vector* form  $P(\alpha,\beta)=R+\alpha u+\beta v$ , we know we can use the cross product to find

$$\mathbf{n} = \mathbf{u} \times \mathbf{v}$$

and the equivalent form

$$(P(\alpha,\beta)-P) \cdot \mathbf{n}=0$$



## Summary: Geometry

- Introduce the elements of geometry
  - Scalars
  - Vectors
  - Points
- Develop mathematical operations among them in a coordinate-free manner
- Define basic primitives
  - Line segments
  - Polygons

---

# Linear Algebra Review

---

---

## Overview

- Basic matrix operations (+, -, \*)
  - Cross and dot products
  - Determinants and inverses
-

## What is a Matrix?

- A matrix is a set of elements, organized into rows and columns

$m \times n$  matrix

$$\begin{array}{c} \underbrace{\hspace{10em}}_{n \text{ columns}} \\ \left\{ \begin{array}{l} \left[ \begin{array}{cc} a_{00} & a_{01} \\ a_{10} & a_{11} \end{array} \right] \end{array} \right. \\ m \text{ rows} \end{array}$$

## Basic Operations

- Transpose: Swap rows with columns

$$M = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad M^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

$$V = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad V^T = [x \quad y \quad z]$$

## Basic Operations

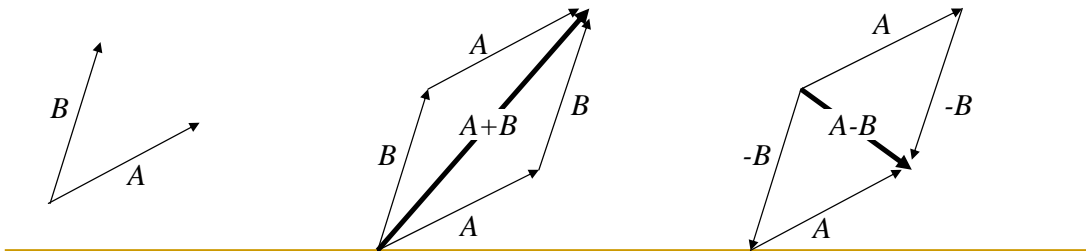
- Addition and Subtraction

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

Just add elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

Just subtract elements



## Basic Operations

- Multiplication

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

Multiply each row  
by each column

An  $m \times n$  can be multiplied by an  $n \times p$  matrix  
to yield an  $m \times p$  result

- Goal: transform column vectors

$$b = M a$$

## Multiplication

- Is  $AB = BA$ ? Maybe, but maybe not!

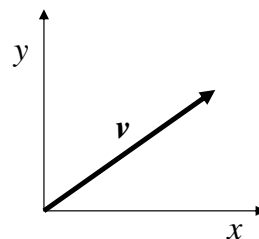
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & \dots \\ \dots & \dots \end{bmatrix} \quad \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ea+fc & \dots \\ \dots & \dots \end{bmatrix}$$

- Heads up: multiplication is NOT commutative!

## Vector Operations

- Vector:  $n \times 1$  matrix
- Interpretation: a point or line in  $n$ -dimensional space
- Dot Product, Cross Product, and Magnitude defined on vectors only

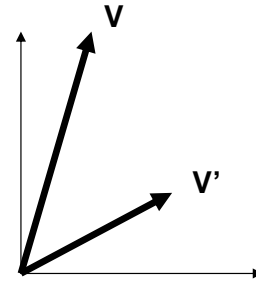
$$\vec{v} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



## Vector Interpretation

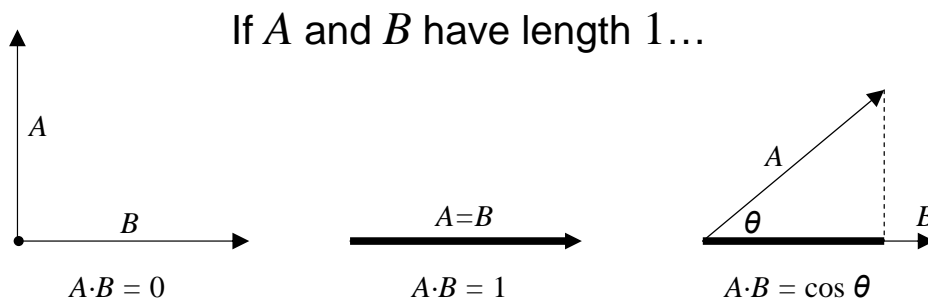
- Think of a vector as a line in 2D or 3D
- Think of a matrix as a transformation on a line or set of lines

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



## Vectors: Dot Product

- Interpretation: the dot product measures to what degree two vectors are aligned



## Vectors: Dot Product

$$A \cdot B = AB^T = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = ad + be + cf$$

Think of the dot product as a matrix multiplication

$$\|A\|^2 = AA^T = aa + bb + cc$$

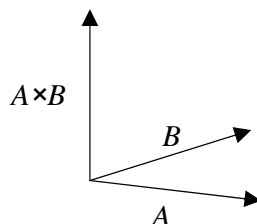
The magnitude is the dot product of a vector with itself

$$A \cdot B = \|A\| \|B\| \cos(\theta)$$

The dot product is also related to the angle between the two vectors

## Vectors: Cross Product

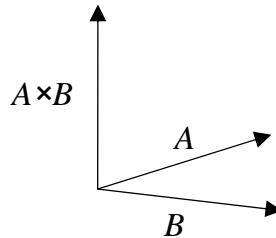
- The cross product of vectors  $A$  and  $B$  is a vector  $C$  which is perpendicular to  $A$  and  $B$
- The magnitude of  $C$  is proportional to the sin of the angle between  $A$  and  $B$
- The direction of  $C$  follows the **right hand rule** if we are working in a right-handed coordinate system



$$\|A \times B\| = \|A\| \|B\| \sin(\theta)$$

# Vectors: Cross Product

The cross-product can be computed as a specially constructed determinant



$$A \times B = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

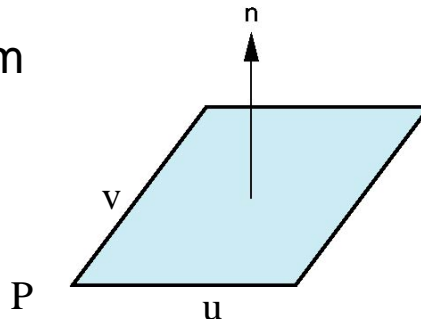
# Normals

- Every plane has a vector  $n$  normal (perpendicular, orthogonal) to it
- From *point - two vector* form  $P(\alpha, \beta) = R + \alpha u + \beta v$ , we know we can use the cross product to find

$$n = u \times v$$

and the equivalent form

$$(P(\alpha, \beta) - P) \cdot n = 0$$



## Inverse of a Matrix

- Identity matrix:  
 $AI = A$
- Some matrices have an inverse, such that:  
 $AA^{-1} = I$
- Inversion is tricky:  
 $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$   
Derived from non-commutativity property

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

## Gauss-Jordan Elimination

1. Append the identity matrix to  $A$
2. Subtract multiples of the other rows from the first row to reduce the diagonal element to 1
3. Transform the identity matrix as you go
4. When the original matrix is the identity, the identity has become the inverse!

$$[A \ I] \equiv \begin{bmatrix} a_{11} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 & b_{11} & \dots & b_{1n} \\ 0 & 1 & \dots & 0 & b_{21} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & b_{n1} & \dots & b_{nn} \end{bmatrix}$$

$$B \equiv \begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix}$$

## Determinant of a Matrix

- Used for inversion
- If  $\det(A) = 0$ , then  $A$  has no inverse
- Can be found using factorials, pivots, and cofactors!
- Lots of interpretations – e.g., signed area or volume

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(A) = ad - bc$$

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

## Definition of Determinant

- Determinant  $|A|$  is a uniquely defined scalar associated with a square matrix  $A$
- $|A|$  defined as the sum of all possible products

$$\sum_t (-1)^t a_{1j_1} a_{2j_2} \dots a_{nj_n},$$

where the series of second subscripts is a permutation of  $(1, \dots, n)$  including the natural order  $(1, \dots, n)$ , and  $t$  is the number of transpositions required to change a permutation back into the original order

## Determinant of a Matrix

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

For a 3x3 matrix:  
Sum from left to right  
Subtract from right to left

**Note:** In the general case, the determinant has  $n!$  terms

## Evaluating a Determinant

$$\begin{vmatrix} + & - \\ - & + \end{vmatrix} \quad \begin{vmatrix} + & - & + \\ - & + & - \\ + & - & + \end{vmatrix} \quad \begin{vmatrix} + & - & + & - \\ - & + & - & + \\ + & - & + & - \\ - & + & - & + \end{vmatrix}$$

$$|A| = \begin{vmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{vmatrix} = 9 \begin{vmatrix} 5 & 4 \\ 2 & 1 \end{vmatrix} - 8 \begin{vmatrix} 6 & 4 \\ 3 & 1 \end{vmatrix} + 7 \begin{vmatrix} 6 & 5 \\ 3 & 2 \end{vmatrix}$$

$$= (9)[(5)-(8)] - 8[6-12] + (7)[12-15] = -27 + 48 - 21 = 0$$

---

# Summary

- Basic matrix operations (+, -, \*)
  - Cross and dot products
  - Determinants and inverses
- 

---

# Representation

---

---

## Objectives

- Introduce concepts such as dimension and basis
  - Introduce coordinate systems for representing vectors spaces and frames for representing affine spaces
  - Discuss change of frames and bases
  - Introduce homogeneous coordinates
- 

---

## Linear Independence

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

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n = 0 \text{ iff } \alpha_1 = \alpha_2 = \dots = 0$$

- 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

- 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  $v_1, v_2, \dots, v_n$ , any vector  $v$  can be written as

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

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

---

---

## Representation

- Until now we have been able to work with geometric entities without using any frame of reference, such as a coordinate system
  - Need a frame of reference to relate points and objects to our physical world.
    - For example, where is a point? Can't answer without a reference system
    - World coordinates
    - Camera coordinates
-

---

# Coordinate Systems

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

$$\mathbf{a} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_n]^T = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \alpha_n \end{bmatrix}$$

---

---

# After Class

- Read
    - Chapter 4
    - Appendices B,C
  - Work on [HW2](#)
-