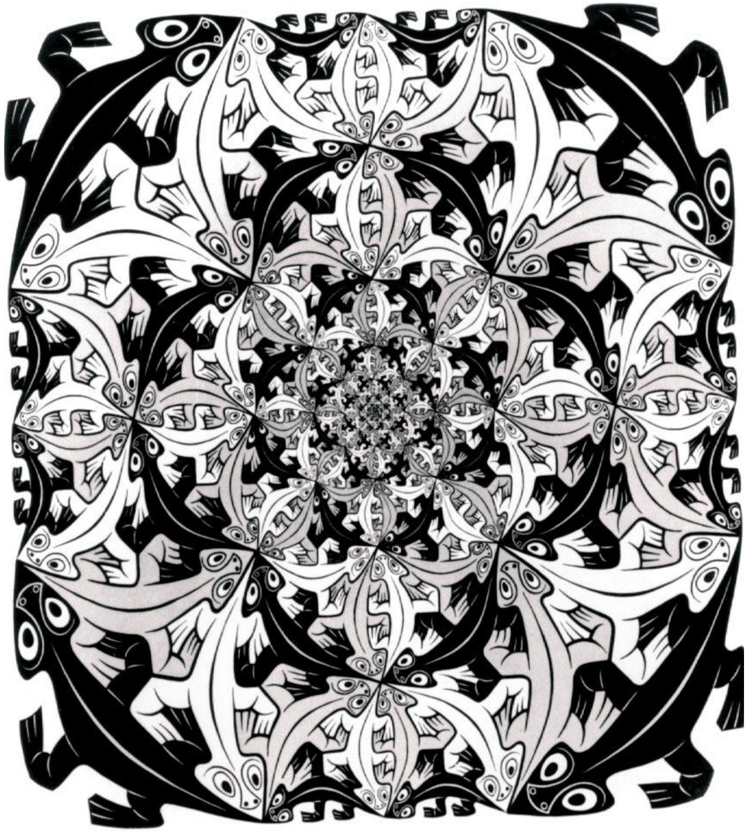


# Geometric primitives and the rendering pipepline

# Rendering geometric primitives

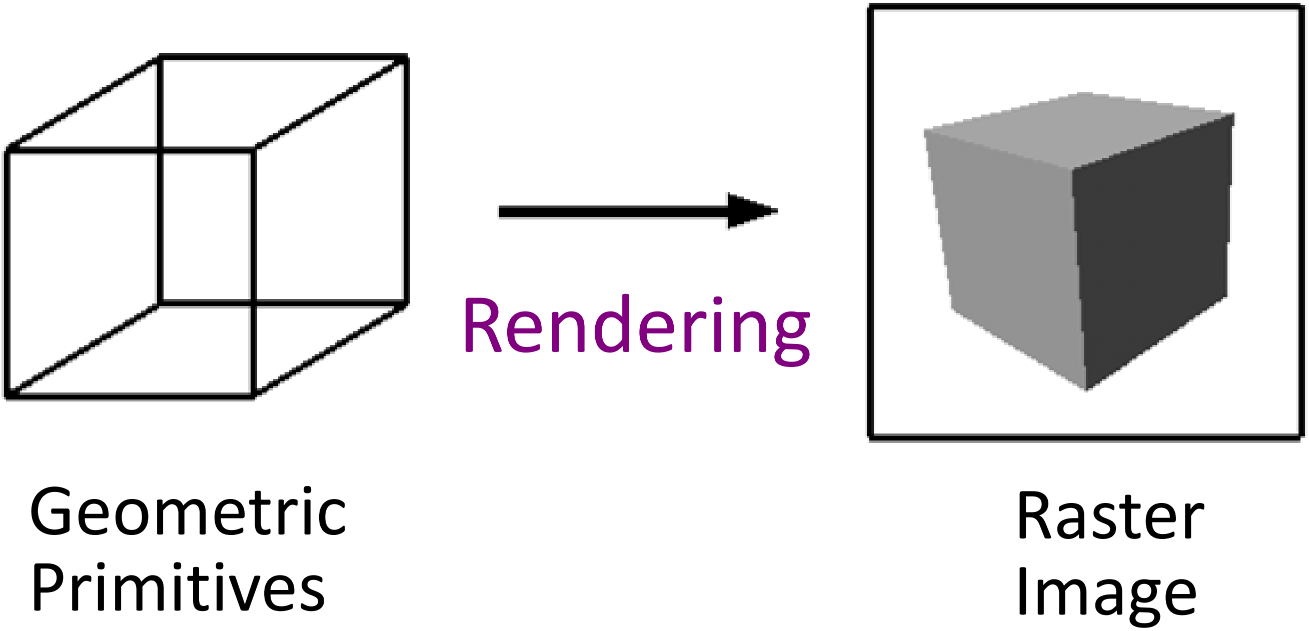
- Describe objects with points, lines, and surfaces
  - Compact mathematical notation
  - Operators to apply to those representations
- Render the objects
  - The rendering pipeline

- Appendix A1-A5

H&B Figure 109

# Rendering

- Generate an image from geometric primitives

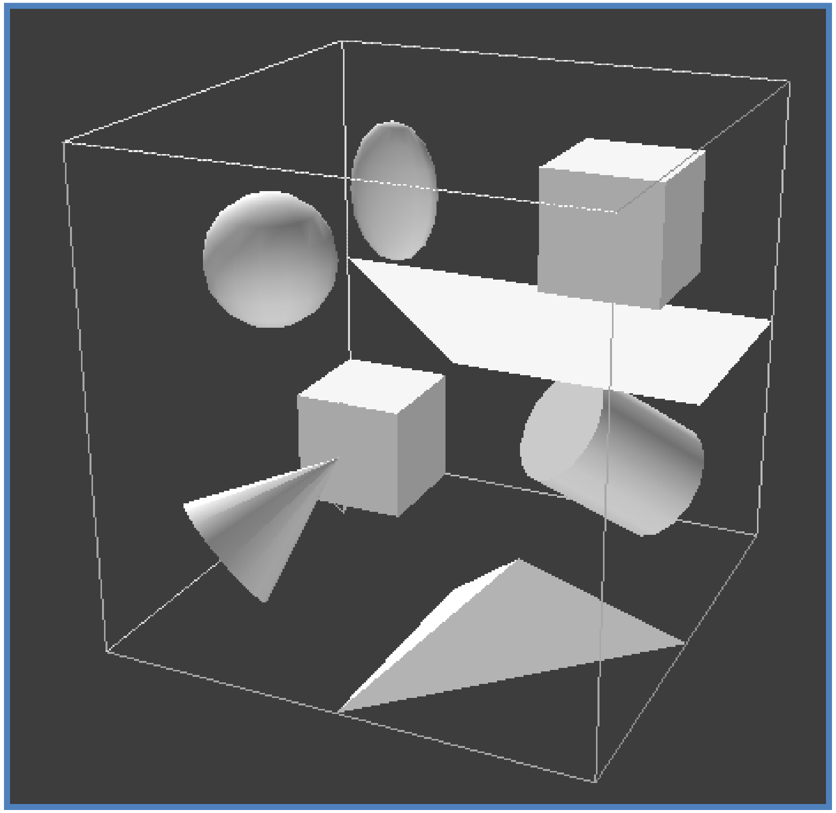


Rendering

Geometric
Primitives

Raster
Image

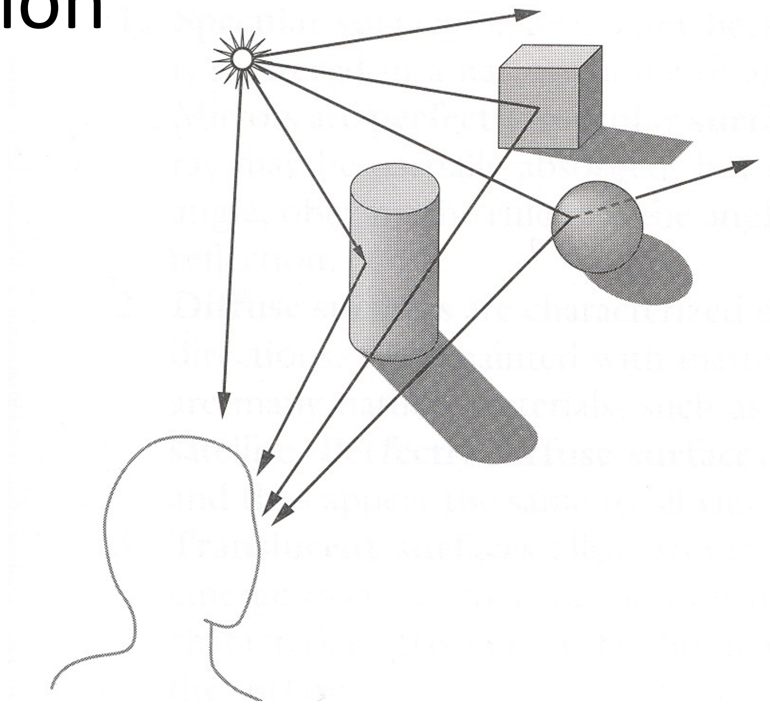
# 3D Rendering Example



What issues must be addressed by a
3D rendering system?

# Overview

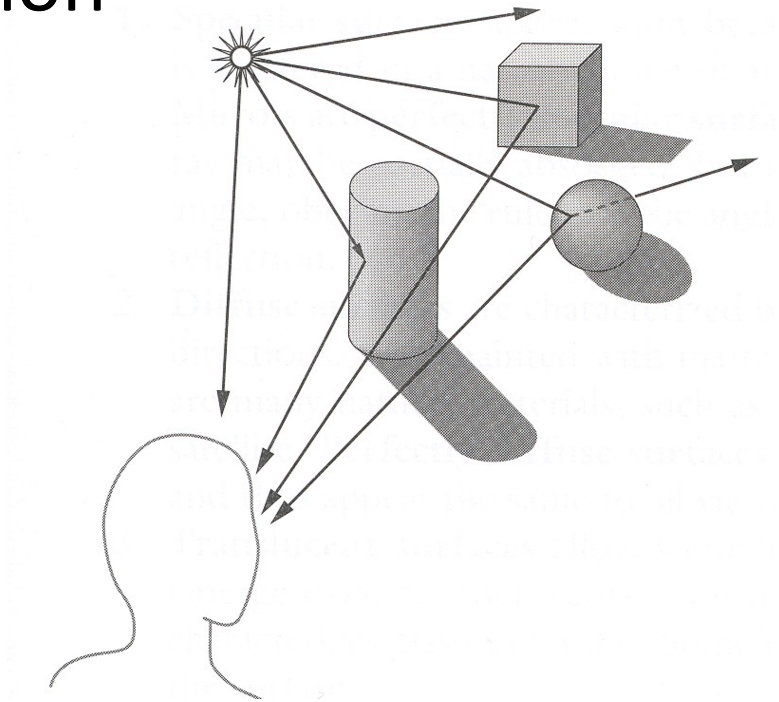
- 3D scene representation
- 3D viewer representation
- Visible surface determination
- Lighting simulation



# Overview

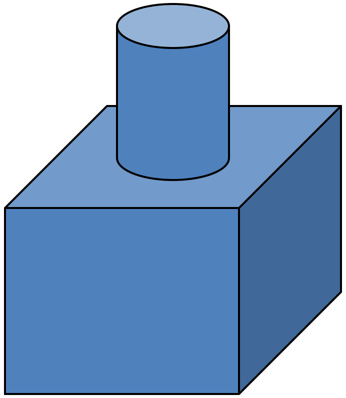
- 3D scene representation
- 3D viewer representation
- Visible surface determination
- Lighting simulation

How is the 3D scene described in a computer?



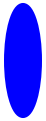
# 3D Scene Representation

- Scene is usually approximated by 3D primitives
  - Point
  - Line segment
  - Polygon
  - Polyhedron
  - Curved surface
  - Solid object
  - etc.



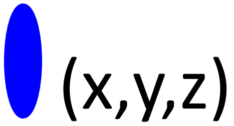
# 3D Point

- Specifies a location



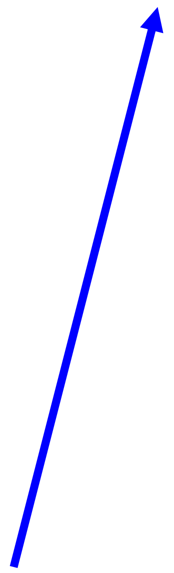
# 3D Point

- Specifies a location
  - Represented by three coordinates
  - Infinitely small



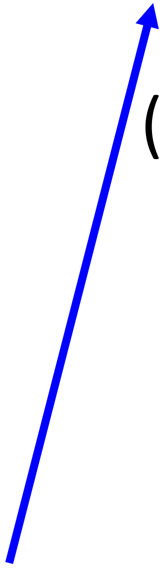
# 3D Vector

- Specifies a direction and a magnitude



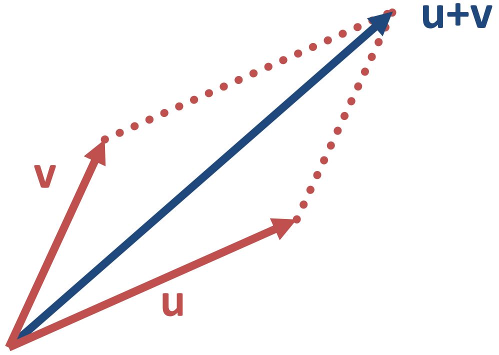
# 3D Vector

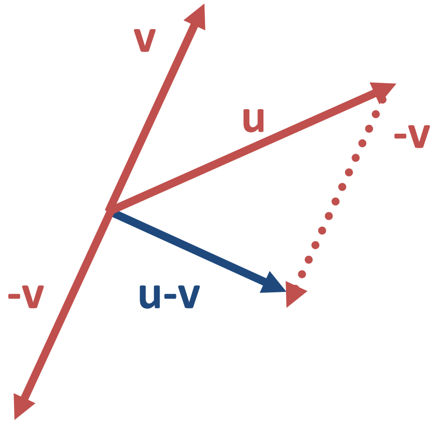
- Specifies a direction and a magnitude
  - Represented by three coordinates
  - Magnitude ||V|| = sqrt(dx dx + dy dy + dz dz)
  - Has no location



# Vector Addition/Subtraction

- operation **u + v**, with:
    - Identity $\mathbf{0}$ : $\mathbf{v} + \mathbf{0} = \mathbf{v}$
    - Inverse **-** : $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- Addition uses the “parallelogram rule”:





# Vector Space

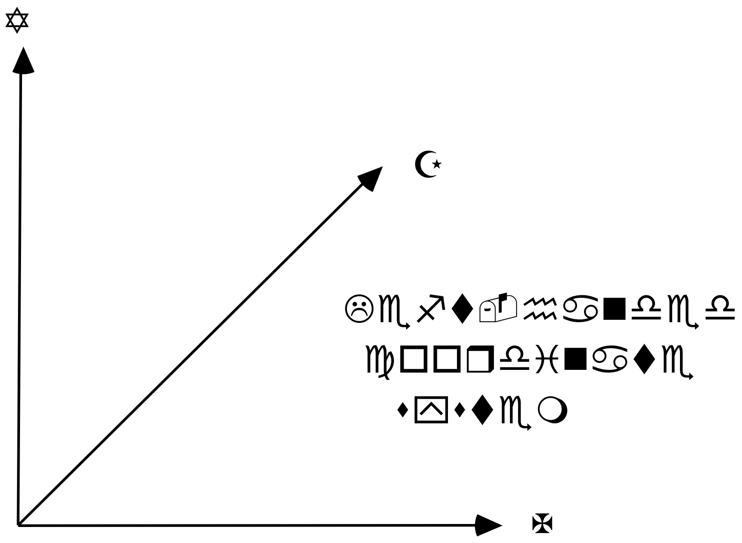
- Vectors define a vector space
  - They support vector addition
    - Commutative and associative
    - Possess identity and inverse
  - They support scalar multiplication
    - Associative, distributive
    - Possess identity

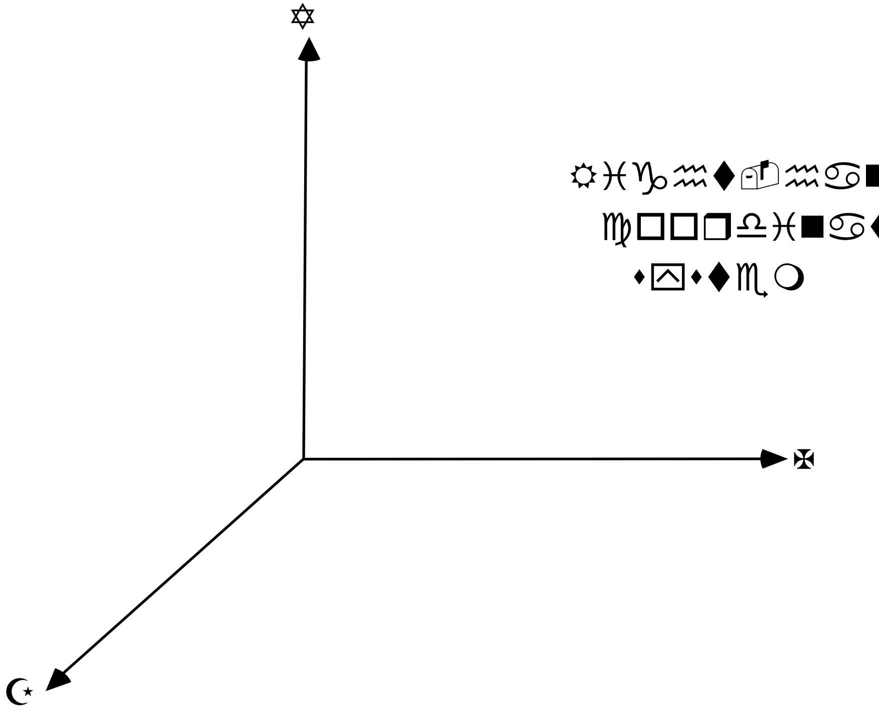
# Affine Spaces

- Vector spaces lack position and distance
  - They have magnitude and direction but no location
- Combine the point and vector primitives
  - Permits describing vectors relative to a common location
- A point and three vectors define a 3-D coordinate system
- Point-point subtraction yields a vector

# Coordinate Systems

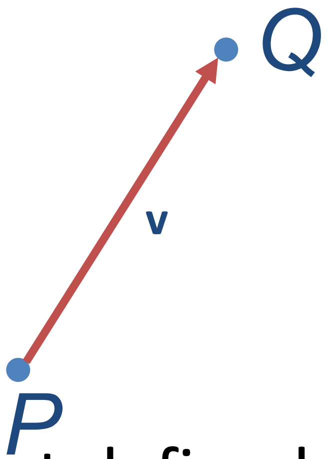
- Grasp z-axis with hand
- Thumb points in direction of z-axis
- Roll fingers from positive x-axis towards positive y-axis





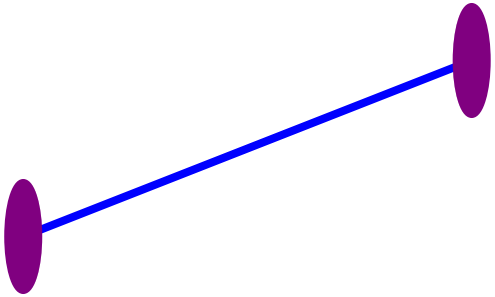
# Points + Vectors

- Points support these operations
  - Point-point subtraction: $Q - P = \mathbf{v}$
    - Result is a vector pointing from P to Q
  - Vector-point addition: $P + \mathbf{v} = Q$
    - Result is a new point
  - Note that the addition of two points is not defined



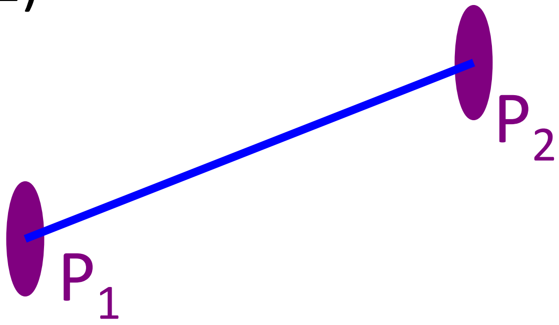
# 3D Line Segment

- Linear path between two points



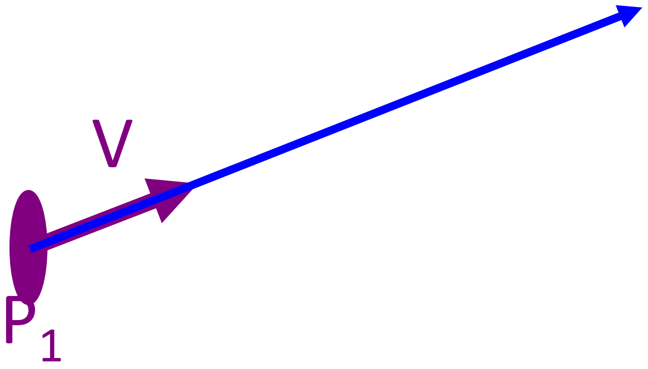
# 3D Line Segment

- Use a linear combination of two points
  - Parametric representation:
    - $P = P_1 + t\,(P_2 - P_1), \quad (0 \leq t \leq 1)$



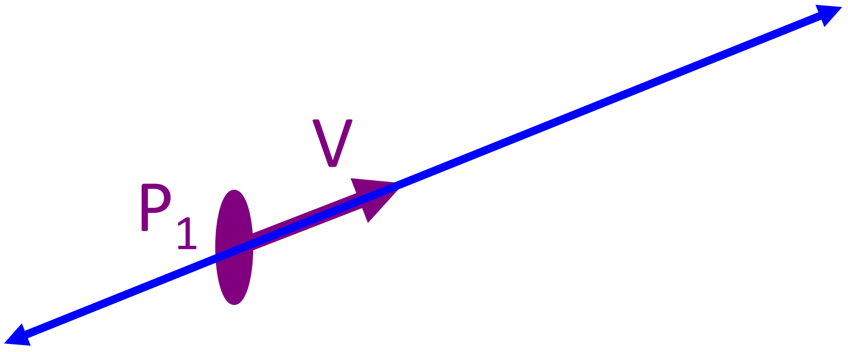
# 3D Ray

- Line segment with one endpoint at infinity
  - Parametric representation:
    - $P = P_1 + t\,V, \quad (0 <= t < \infty)$



# 3D Line

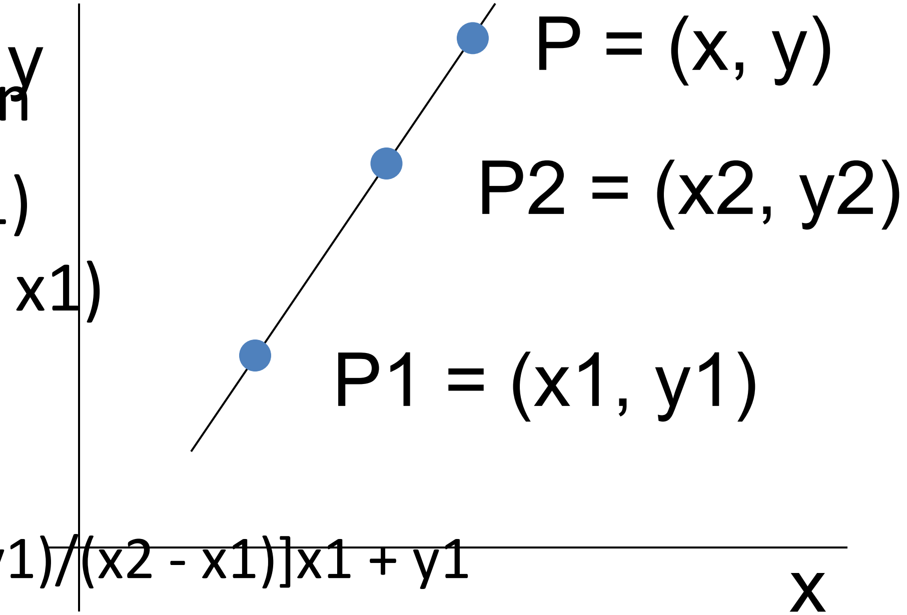
- Line segment with both endpoints at infinity
  - Parametric representation:
    - $P = P_1 + t\,V, \quad (-\infty < t < \infty)$



# 3D Line – Slope Intercept

- Slope $= m$
- $= \text{rise} / \text{run}$
- Slope $= (y - y1) / (x - x1)$
  $= (y2 - y1) / (x2 - x1)$

- Solve for y:
- $y = [(y2 - y1)/(x2 - x1)]x + [-(y2-y1)/(x2 - x1)]x1 + y1$
- or: $y = mx + b$



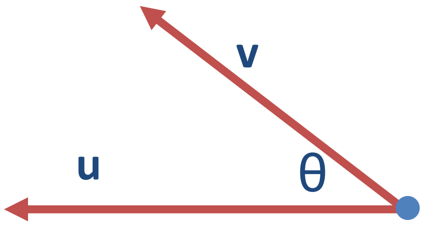
# Euclidean Spaces

- Q: What is the distance function between points and vectors in affine space?
- A: Dot product
  - Euclidean affine space = affine space plus dot product
  - Permits the computation of distance and angles

# Dot Product

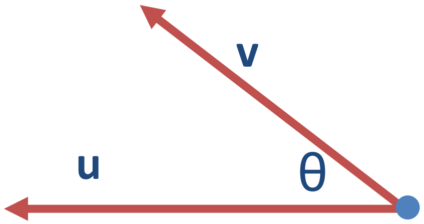
- The dot product or, more generally, inner product of two vectors is a scalar:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = \mathbf{x}_1\mathbf{x}_2 + \mathbf{y}_1\mathbf{y}_2 + \mathbf{z}_1\mathbf{z}_2 \qquad \text{(in 3D)}$$



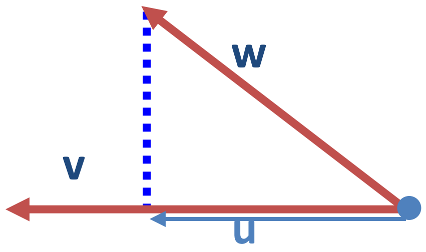
# Dot Product

- Useful for many purposes
  - Computing the length (Euclidean Norm) of a vector:
    - length($\mathbf{v}$) = ||**v**|| = sqrt($\mathbf{v} \bullet \mathbf{v}$)
  - *Normalizing* a vector, making it unit-length: **v = v / ||v||**
  - Computing the angle between two vectors:
    - $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}|\,|\mathbf{v}|\cos(\theta)$
  - Checking two vectors for orthogonality
    - $\mathbf{u} \bullet \mathbf{v} = \mathbf{0.0}$



# Dot Product

- Projecting one vector onto another
  - If **v** is a unit vector and we have another vector, **w**
  - We can project **w** perpendicularly onto **v**



[redacted]

  - And the result, **u**, has length **w** • **v**

# Dot Product

- Is commutative
  - $u \bullet v = v \bullet u$
- Is distributive with respect to addition
  - $u \bullet (v + w) = u \bullet v + u \bullet w$

# Cross Product

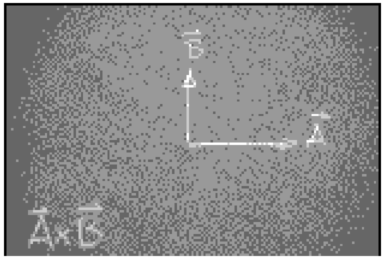
- The cross product or vector product of two vectors is a vector:

[redacted]

- both
- Right-hand rule dictates direction of cross product

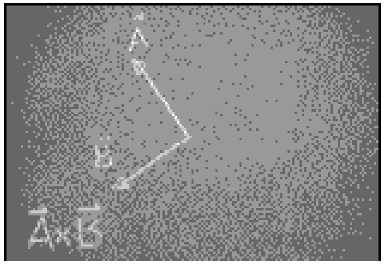
# Cross Product Right Hand Rule

- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product



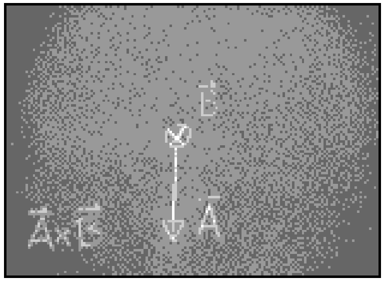
# Cross Product Right Hand Rule

- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product



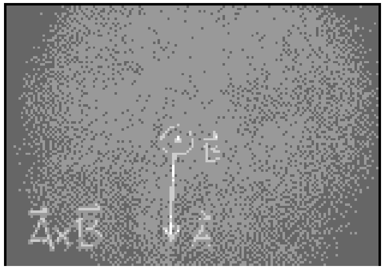
# Cross Product Right Hand Rule

- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product



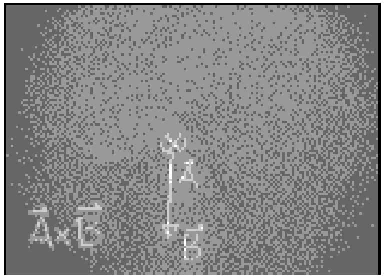
# Cross Product Right Hand Rule

- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product



# Cross Product Right Hand Rule

- See: http://www.phy.syr.edu/courses/video/RightHandRule/index2.html
- Orient your right hand such that your palm is at the beginning of A and your fingers point in the direction of A
- Twist your hand about the A-axis such that B extends perpendicularly from your palm
- As you curl your fingers to make a fist, your thumb will point in the direction of the cross product

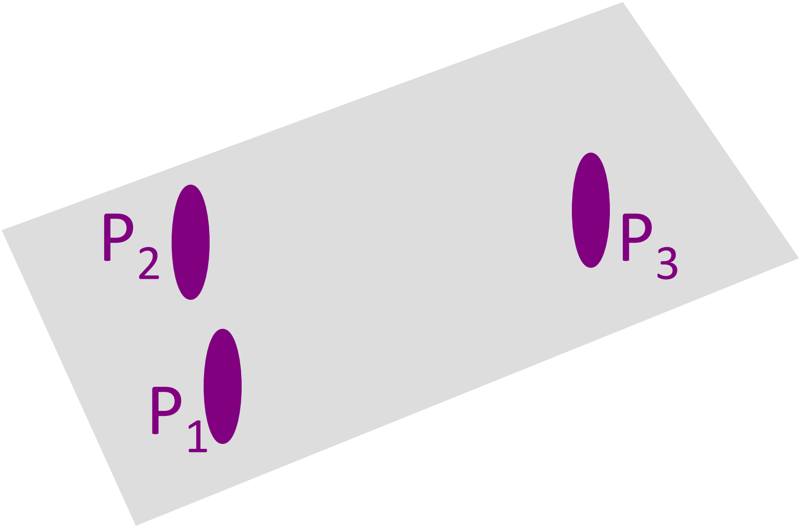


# Other helpful formulas

- Length = sqrt $(x2 - x1)^2 + (y2 - y1)^2$
- Midpoint, p2, between p1 and p3
  - p2 = ((x1 + x3) / 2, (y1 + y3) / 2))
- Two lines are perpendicular if:
  - M1 = -1/M2
  - cosine of the angle between them is 0
  - Dot product = 0

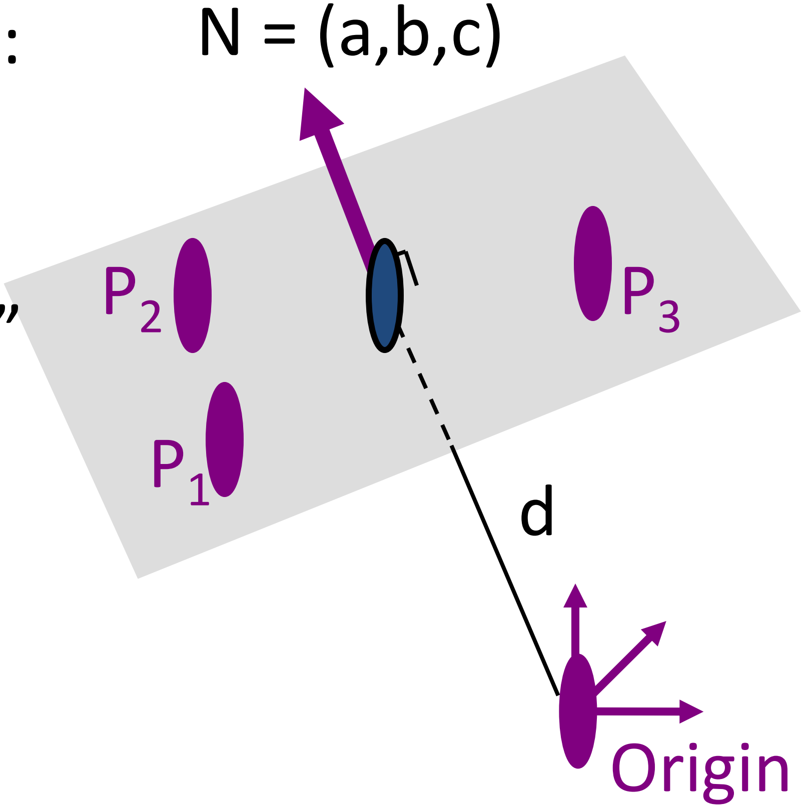
# 3D Plane

- A linear combination of three points



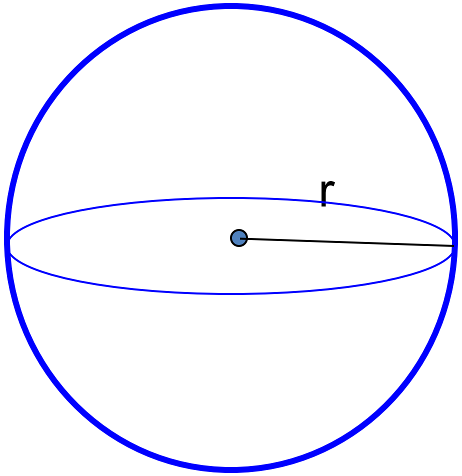
# 3D Plane

- A linear combination of three points
  - Implicit representation:
    - $ax + by + cz + d = 0$, or
    - $P \cdot N + d = 0$
  - N is the plane “normal”
    - Unit-length vector
    - Perpendicular to plane



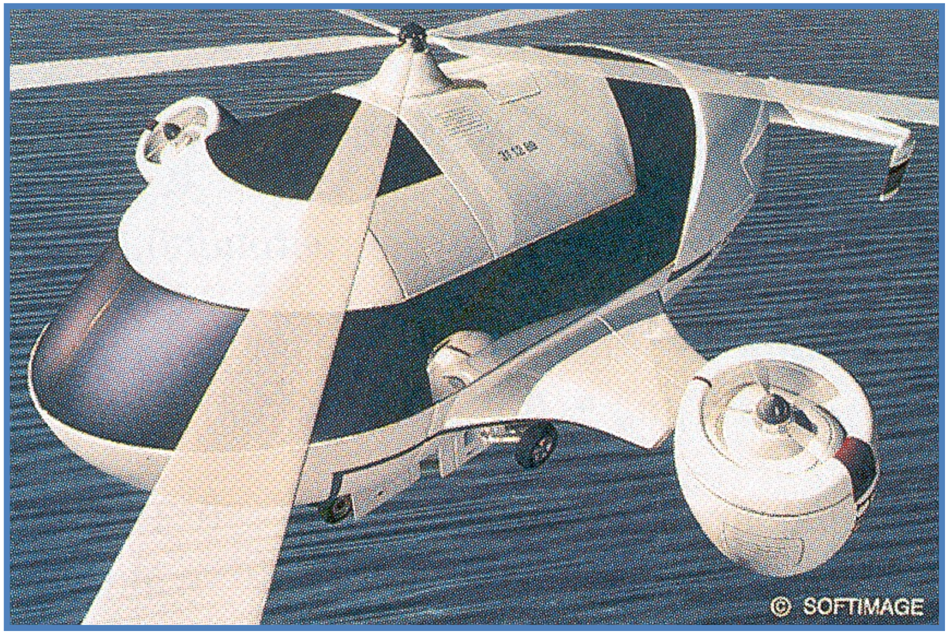
# 3D Sphere

- All points at distance “r” from point “($c_x$, $c_y$, $c_z$)”
  - Implicit representation:
    - $(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2$
  - Parametric representation:
    - $x = r \cos(\phi) \cos(\Theta) + c_x$
    - $y = r \cos(\phi) \sin(\Theta) + c_y$
    - $z = r \sin(\phi) + c_z$



# 3D Geometric Primitives

- More detail on 3D modeling later in course
  - Point
  - Line segment
  - Polygon
  - Polyhedron
  - Curved surface
  - Solid object
  - etc.



H&B Figure 10.46