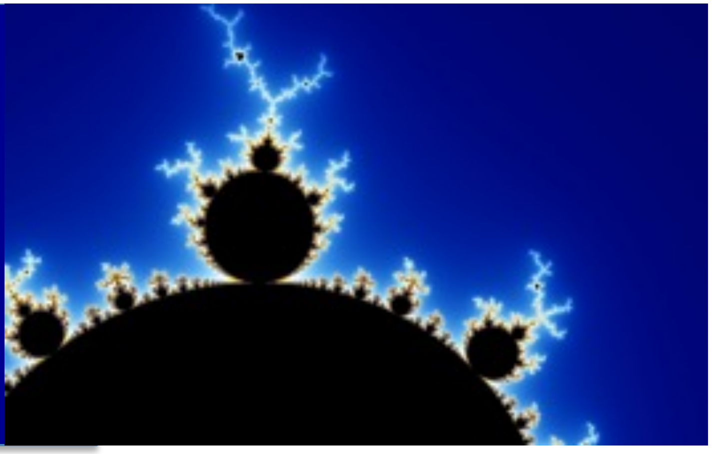
Computer Graphics



Vector Tools for Graphics

Time for some math

- We’re going to review some of the basic mathematical constructs used in computer graphics
  - Scalars
  - Points
  - Vectors
  - Matrices
  - Other stuff (rays, planes, etc.)
- Why?
  - Most of computer graphics is defined in 3D
    - 2D is only a special case
  - Vector analysis and transformations are crucial to 3D graphics

Scalars

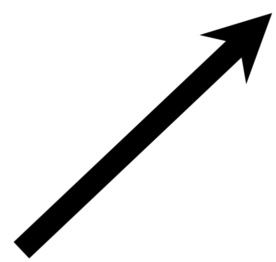
- A <u>scalar</u> is a quantity that does not depend on direction
  – In other words, it's just a regular number
    - *i.e.* 7 is a scalar
    - so is 13.5
    - or -4

Points •

- A <u>point</u> is a list of n numbers referring to a location in n-Dimension
- The individual components of a point are often referred to as <u>coordinates</u>
  – *i.e.* (2, 3, 4) is a point in 3-D space
    - This point's x-coordinate is 2, it's y-coordinate is 3, and it's z-coordinate is 4

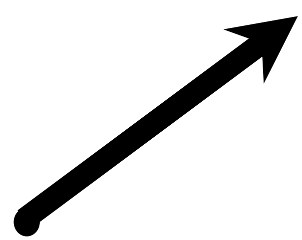
Vectors

- A <u>vector</u> is a list of *n* numbers referring to a direction and magnitude in n-D
- From a data structures perspective, a vector looks exactly the same as a point
  - *i.e.* (2, 3, 4) is a vector in 3-D space
    - Vector does not have a fixed position



Rays

- A <u>ray</u> is just a vector with a starting point
  - Ray = (Point, Vector)
- Let a ray be defined by point **p** and vector $\vec{\mathbf{d}}$
- The <u>parametric</u> form of a ray expresses it as a function as some scalar t, giving the set of all points the ray passes through:
  - $r(t) = \mathbf{p} + t\vec{\mathbf{d}}$, $0 \leq t \leq \infty$



Vectors

- We said that a vector encodes a direction and a magnitude in n-D
  - How does it do this?
- Here are two ways to denote a vector in 2-D:

$$\vec{\mathbf{V}} = \langle V_x, V_y \rangle$$

$$\vec{\mathbf{V}} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$$

Vector Magnitude

- Geometrically, the magnitude of a vector is the Euclidean distance between its start and end points, or more simply, it's length

- Vector magnitude in n-D: $$||\vec{\mathbf{V}}|| = \sqrt{\sum_{i=1}^{n} V_i^2}$$

- Vector magnitude in 2-D: $$||\vec{\mathbf{V}}|| = \sqrt{V_x^2 + V_y^2}$$

Normalized Vectors

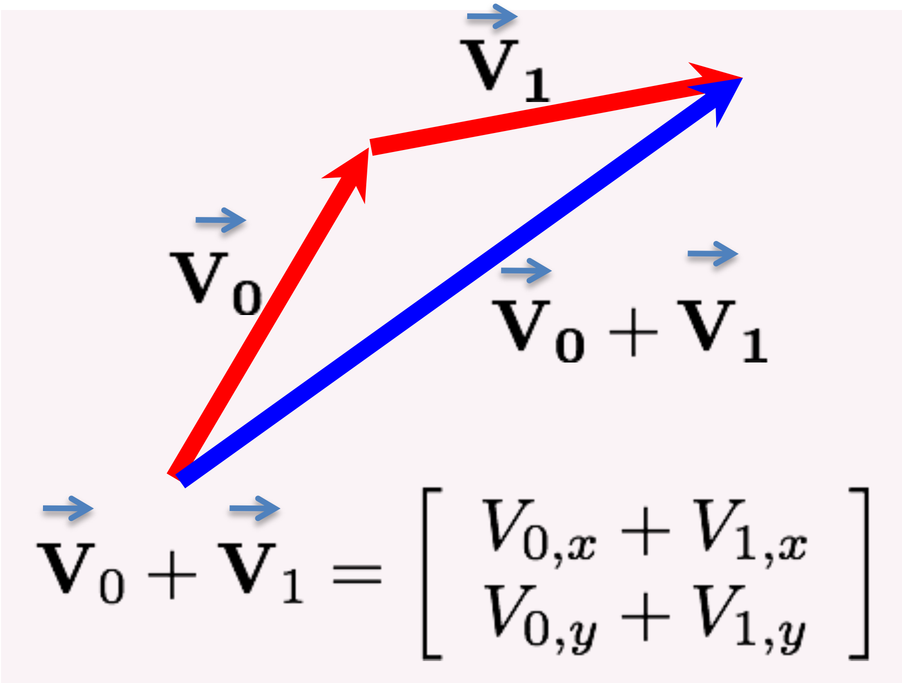
- Most of the time, we want to deal with normalized, or unit, vectors
- This means that the magnitude of the vector is 1: $||\vec{\mathbf{V}}|| = 1$
- We can normalize a vector by dividing the vector by its magnitude:

$$\hat{V} = \frac{\vec{V}}{||\vec{\mathbf{V}}||}$$

Vector Addition

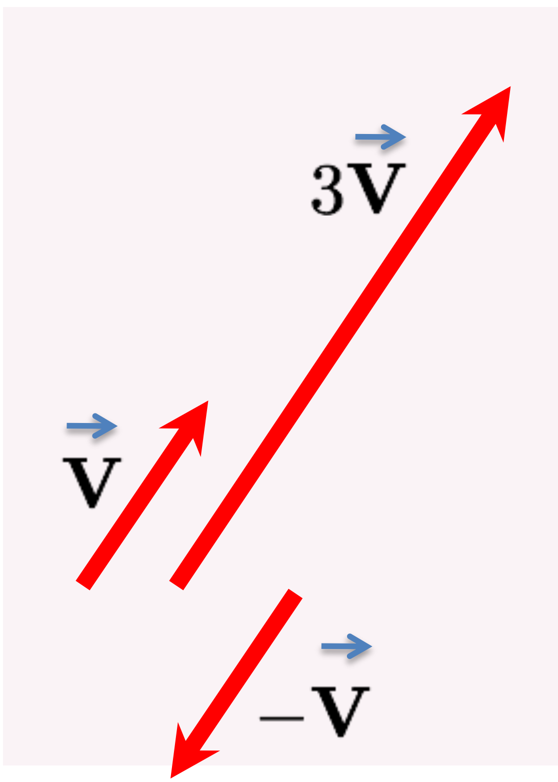
- Vectors are closed under addition
  - Head to tail
  - Vector + Vector = Vector

Vector Addition



Vector Scaling

- Vectors are closed under multiplication with a scalar
  - Scalar * Vector = Vector



Vector Scaling

$$a\vec{\mathbf{V}} = \begin{bmatrix} aV_x \\ aV_y \end{bmatrix}$$

Properties of Vector Addition & Scaling

Addition is Commutative

$$\vec{\mathbf{P}} + \vec{\mathbf{Q}} = \vec{\mathbf{Q}} + \vec{\mathbf{P}}$$

Addition is Associative

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

Scaling is Commutative and Associative

$$(ab)\vec{\mathbf{P}} = a(b\vec{\mathbf{P}})$$

Scaling and Addition are Distributive

$$a(\vec{\mathbf{P}} + \vec{\mathbf{Q}}) = a\vec{\mathbf{P}} + a\vec{\mathbf{Q}}$$

$$(a + b)\vec{\mathbf{P}} = a\vec{\mathbf{P}} + b\vec{\mathbf{P}}$$

Points and Vectors

- Can define a vector by 2 points
  - Point - Point = Vector
- Can define a new point by a point and a vector
  - Point + Vector = Point

Vector Multiplication?

- What does it mean to multiply two vectors?
  - Not uniquely defined
- Two product operations are commonly used:
  - Dot (scalar, inner) product
    - Result is a scalar
  - Cross (vector, outer) product
    - Result is a new vector

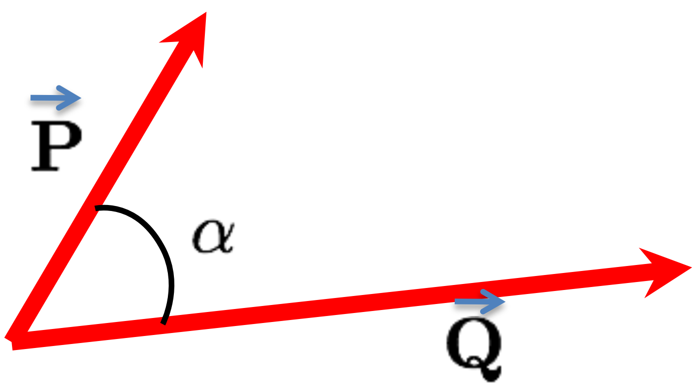
Dot Product Application: Lighting

- $\vec{P} \bullet \vec{Q} = ||\vec{P}|| \bullet ||\vec{Q}|| \cos(\alpha)$
- So what does this mean if $\vec{P}$ and $\vec{Q}$ are normalized?
  - Can get $\cos(\alpha)$ for just 3 multiplies and 2 adds (in 3D)
    - Very useful in lighting and shading calculations
    - Example: Lambert's cosine law

Dot Product

$$\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}} = \sum_{i=1}^{n} P_i Q_i = \begin{bmatrix} P_1 & P_2 & \dots & P_n \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ \dots \\ Q_n \end{bmatrix}$$

$$\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}} = ||\mathbf{P}||\ ||\mathbf{Q}||\ \cos\alpha$$



$$\alpha = cos^{-1}\left(\frac{\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}}}{||\mathbf{P}||\ ||\mathbf{Q}||}\right)$$

Properties of Vector Dot Products

Commutative

$$\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}} = \vec{\mathbf{Q}} \cdot \vec{\mathbf{P}}$$

Associative with Scaling

$$(a\vec{\mathbf{P}}) \cdot \vec{\mathbf{Q}} = a(\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}})$$

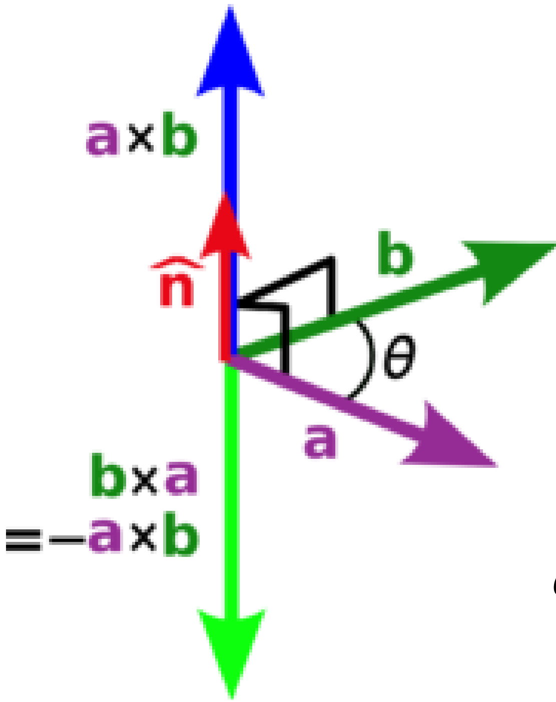
Distributive with Addition

$$\vec{\mathbf{P}} \cdot (\vec{\mathbf{Q}} + \vec{\mathbf{R}}) = \vec{\mathbf{P}} \cdot \vec{\mathbf{Q}} + \vec{\mathbf{P}} \cdot \vec{\mathbf{R}}$$

$$\vec{\mathbf{P}} \cdot \vec{\mathbf{P}} = ||\vec{\mathbf{P}}||^2$$

$$|\vec{\mathbf{P}} \cdot \vec{\mathbf{Q}}| \leq ||\vec{\mathbf{P}}|| \; ||\vec{\mathbf{Q}}||$$

Cross Product



$$\vec{\mathbf{a}} \times \vec{\mathbf{b}} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}.$$

$$\vec{a} \times \vec{b} = i \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} - j \begin{bmatrix} a_1 & a_3 \\ b_1 & b_3 \end{bmatrix} + k \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$$

$$= i(a_2 b_3 - a_3 b_2) - j(a_1 b_3 - a_3 b_1) + k(a_1 b_2 - a_2 b_1)$$

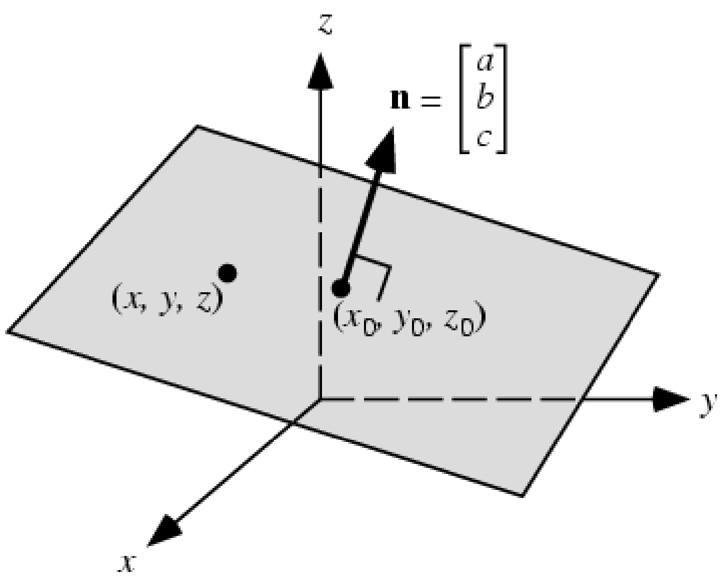
Cross Product Application: Normals

- A <u>normal</u> (or <u>surface normal</u>) is a vector that is perpendicular to a surface at a given point
  - This is often used in lighting calculations
- The cross product of 2 orthogonal vectors on the surface is a vector perpendicular to the surface
  - Can use the cross product to compute the normal

Planes

- How can we define a plane?
  - 3 non-linear points
  - A perpendicular vector and an incident point
    - $n \bullet (x-x_0) = 0$ (why?)

    $ax + by + cz + d = 0$



Columns and Rows

- In this class, we will generally assume that a list forms a column vector:

$$(a, b, c, d) \Longrightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- The reason for this will become clear when we talk about matrices

Matrices

- Reminder: A matrix is a rectangular array of numbers
  - An m x n matrix has m rows and n columns
- M_{ij} denotes the entry in the i-th row and j-th column of matrix M
  - These are generally thought of as 1-indexed
    - instead of 0-indexed
- Here, M is a 2x5 matrix:

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} \end{bmatrix}$$

Matrix Transposes

- The transpose of an m x n matrix is an n x m matrix
  - Denoted M^T
  - $M^T_{ij} = M_{ji}$

$$\mathbf{M}^T = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} \end{bmatrix}^T = \begin{bmatrix} M_{11} & M_{21} \\ M_{12} & M_{22} \\ M_{13} & M_{23} \\ M_{14} & M_{24} \\ M_{15} & M_{25} \end{bmatrix}$$

Matrix Addition

- Only well defined if the dimensions of the 2 matrices are the same
  - That is, $m_1 = m_2$ and $n_1 = n_2$
  - Here, M and G are both 2 x 5

$$(\mathbf{M}+\mathbf{G})_{ij} = M_{ij} + G_{ij}$$

$$\mathbf{M}+\mathbf{G} = \left[\begin{array}{ccccc} M_{11}+G_{11} & M_{12}+G_{12} & M_{13}+G_{13} & M_{14}+G_{14} & M_{15}+G_{15} \\ M_{21}+G_{21} & M_{22}+G_{22} & M_{23}+G_{23} & M_{24}+G_{24} & M_{25}+G_{25} \end{array}\right]$$

Matrix Scaling

– Just like vector scaling
  - Matrix * Scalar = Matrix

$$(a\mathbf{M})_{ij} = aM_{ij}$$

$$a\mathbf{M} = \begin{bmatrix} aM_{11} & aM_{12} & aM_{13} & aM_{14} & aM_{15} \\ aM_{21} & aM_{22} & aM_{23} & aM_{24} & aM_{25} \end{bmatrix}$$

Properties of Matrix Addition and Scaling

Addition is Commutative $\qquad \mathbf{F} + \mathbf{G} = \mathbf{G} + \mathbf{F}$

Addition is Associative $\qquad (\mathbf{F} + \mathbf{G}) + \mathbf{H} = \mathbf{F} + (\mathbf{G} + \mathbf{H})$

Scaling is Associative $\qquad a(b\mathbf{F}) = (ab)\mathbf{F}$

Scaling and Addition are Distributive $\qquad a(\mathbf{F} + \mathbf{G}) = a\mathbf{F} + a\mathbf{G}$

$$(a + b)\mathbf{F} = a\mathbf{F} + b\mathbf{F}$$

Matrix Multiplication

- Only well defined if the number of columns of the first matrix and the number of rows of the second matrix are the same
  - Matrix * Matrix = Matrix
  - *i.e.* if F is m x n, and G is n x p, then F*G is m x p
- Let's do an example

$$(\mathbf{FG})_{ij} = \sum_{k=1}^{m} F_{ik} G_{kj}$$

The Identity Matrix

- Defined such that the product of any matrix M and the identity matrix I is M
  - **IM = MI = M**
- The identity matrix is a square matrix with ones on the diagonal and zeros elsewhere

$$(\mathbf{I}_n)_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \qquad \mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matrix Point Multiplication

- Given a Transformation matrix, e.g., translation matrix with translation along vector [3, 4, 1, 0]
- Where will the point [1, 2, 2, 1] be after the translation?

Applied to Graphics…

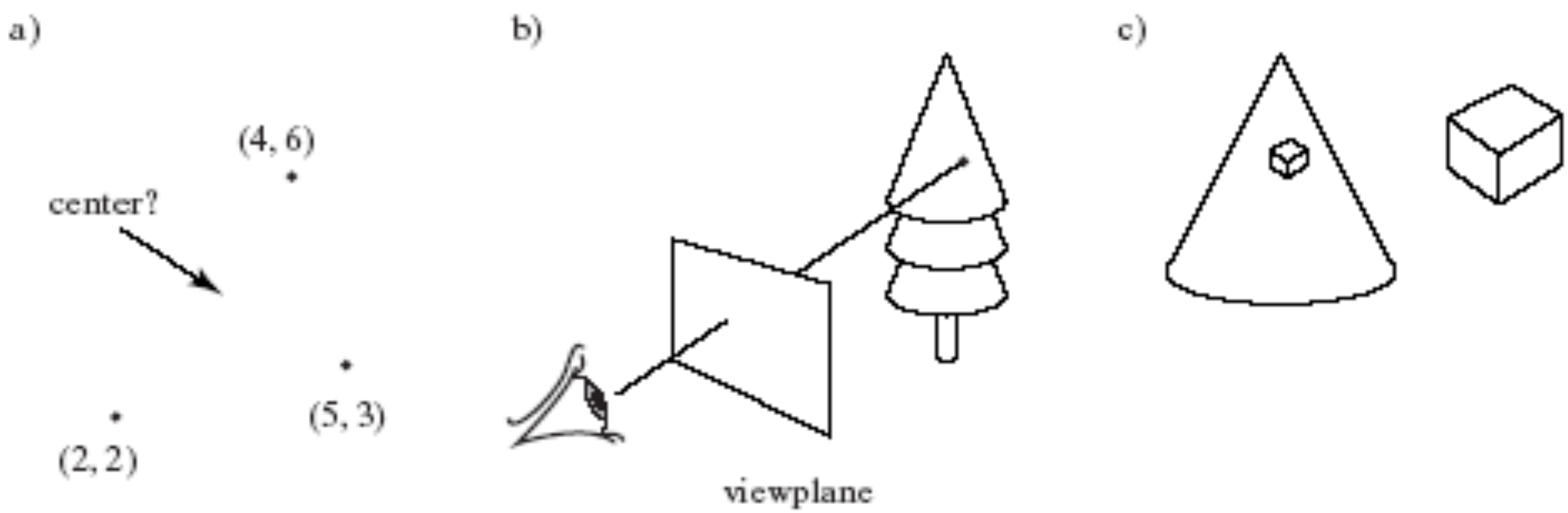
- In computer graphics, we work with objects defined in a three dimensional world
- All objects to be drawn, and the cameras used to draw them, have shape, position, and orientation.
- We must write computer programs that somehow
  - describe these objects
  - describe how light bounces around illuminating them
  - so that the final pixel values on the display can be computed.

Back to Graphics…

- The two fundamental sets of tools that come to our aid in graphics are *vector analysis* and *transformations*
- We develop methods to describe various geometric objects, and we learn how to convert geometric ideas to numbers.
- This provides a collection of crucial algorithms that we can use in graphics programs.

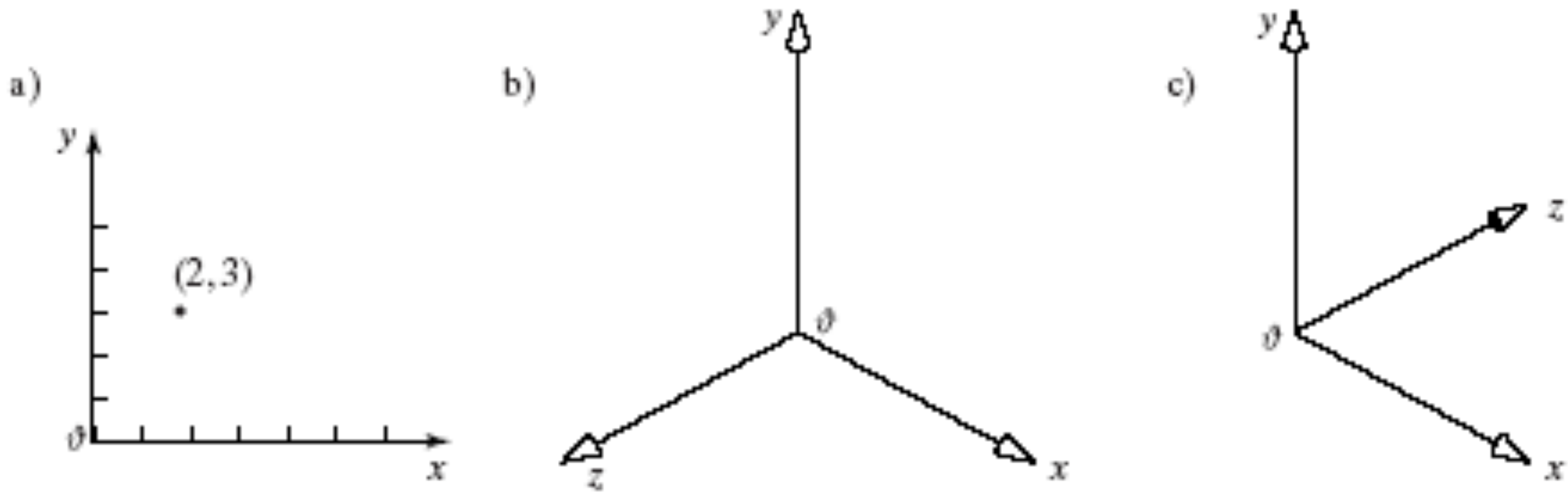
Easy Problems for Vectors

- Where is the center of the circle through the 3 points?
- What image shape appears on the viewplane, and where?
- Where does the reflection of the cube appear on the shiny cone, and what is the exact shape of the reflection?



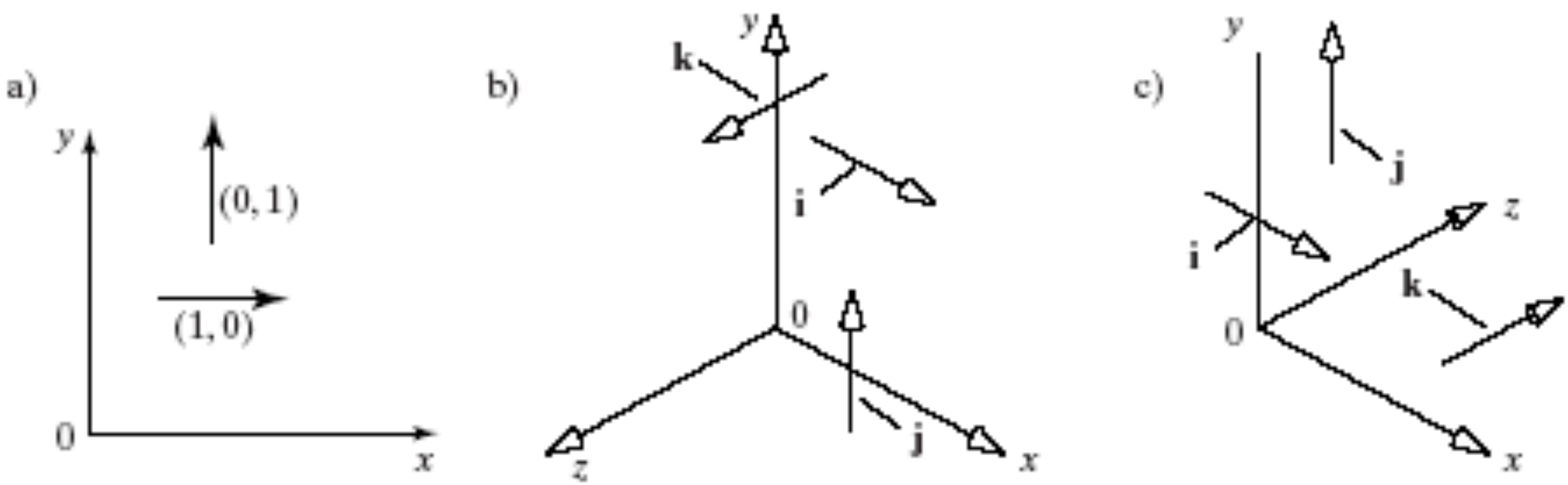
Basics of Points and Vectors

- All points and vectors are defined relative to some coordinate system. Shown below are a 2D coordinate system and a right- and a left-handed 3-D coordinate system.



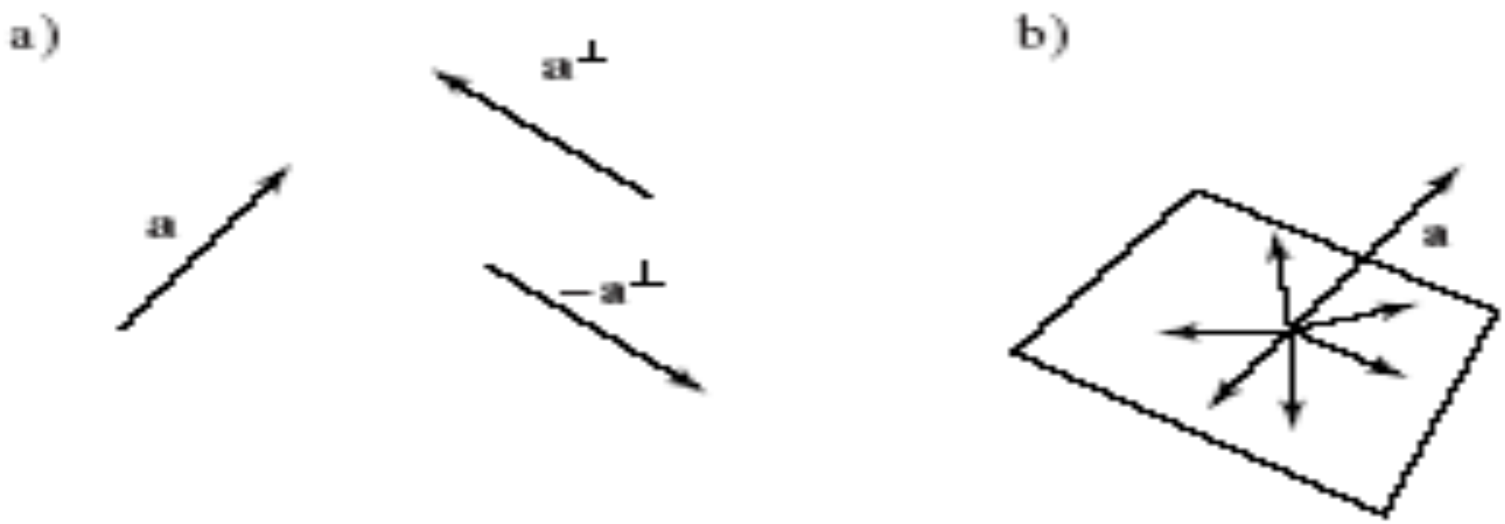
Standard Unit Vectors

- The standard unit vectors in 3D are **i** = (1,0,0,0), **j** = (0, 1, 0, 0), and **k** = (0, 0, 1, 0). **k** always points in the positive z direction
- In 2D, **i** = (1,0) and **j** = (0, 1).
- The unit vectors are orthogonal.



Finding a 2D "Perp" Vector

- If vector **a** = (a_x, a_y), then the vector perpendicular to **a** in the *counterclockwise* sense is $\mathbf{a}^{\perp} = (-a_y, a_x)$, and in the *clockwise* sense it is $-\mathbf{a}^{\perp}$.
- In 3D, any vector in the plane perpendicular to **a** is a "perp" vector.



Matrix Inversion

- After performing a Transformation (Translate, Rotate, or Scale), how to undo this Transformation?
  - Suppose the transformation matrix is A,
  - Apply the inversion, A^{-1}, will undo the transformation

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

https://ncalculators.com/matrix/inverse-matrix.htm

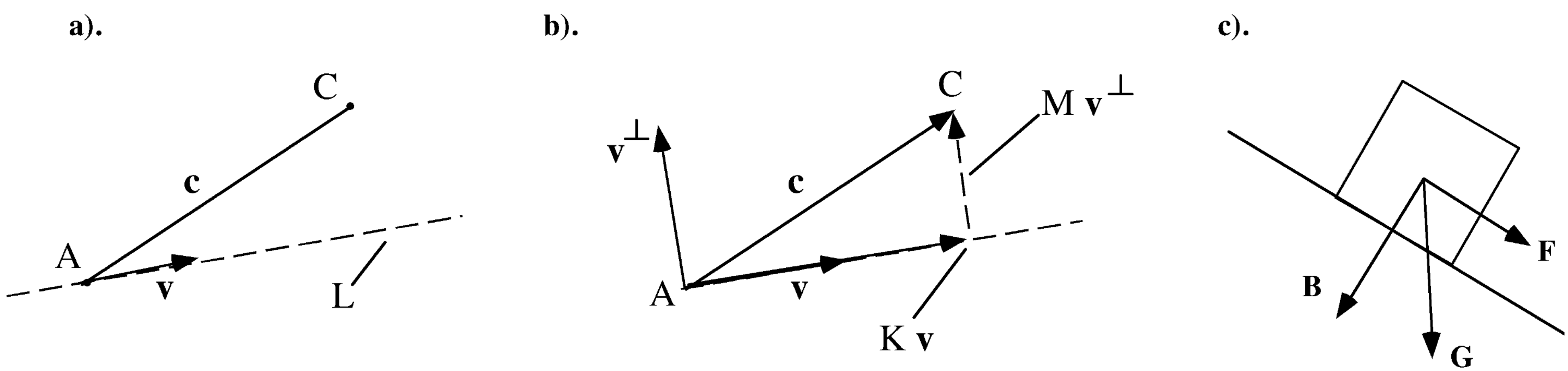
$$A^{-1} = \frac{1}{det(A)} adj(A)$$

$$adj(A) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

$$\begin{aligned}
a_{11} &= -hkn + gln + hjo - flo - gjp + fkp \\
a_{12} &= dkn - cln - djo + blo + cjp - bkp \\
a_{13} &= -dgn + chn + dfo - bho - cfp + bgp \\
a_{14} &= dgj - chj - dfk + bhk + cfl - bgl \\
a_{21} &= hkm - glm - hio + elo + gip - ekp \\
a_{22} &= -dkm + clm + dio - alo - cip + akp \\
a_{23} &= dgm - chm - deo + aho + cep - agp \\
a_{24} &= -dgi + chi + dek - ahk - cel + agl \\
a_{31} &= -hjm + flm + hin - eln - fip + ejp \\
a_{32} &= djm - blm - din + aln + bip - ajp \\
a_{33} &= -dfm + bhm + den - ahn - bep + afp \\
a_{34} &= dfi - bhi - dej + ahj + bel - afl \\
a_{41} &= gjm - fkm - gin + ekn + fio - ejo \\
a_{42} &= -cjm + bkm + cin - akn - bio + ajo \\
a_{43} &= cfm - bgm - cen + agn + beo - afo \\
a_{44} &= -cfi + bgi + cej - agj - bek + afk
\end{aligned}$$

Orthogonal Projections and Distance from a Line

- We are given 2 points A and C and a vector $\vec{\mathbf{v}}$. The following questions arise:
  - How far is C from the line L that passes through A in direction $\vec{\mathbf{v}}$?
  - If we drop a perpendicular onto L from C, where does it hit L?
  - How do we decompose a vector $\vec{\mathbf{c}}$ = C – A into a part along L and a part perpendicular to L?



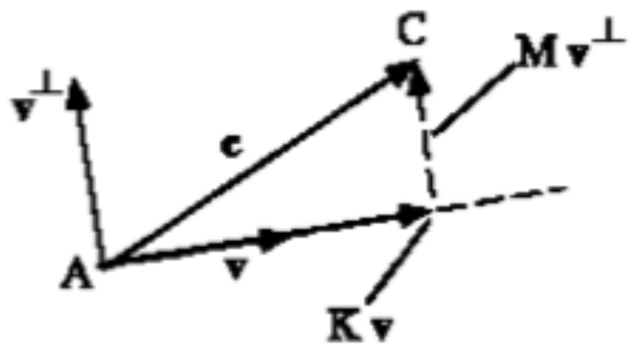
Answering the Questions

- We may write $\mathbf{c} = K\mathbf{v} + M\mathbf{v}^{\perp}$.
- If we take the dot product of each side with $\mathbf{v}$,
  - we obtain $\mathbf{c}\cdot\mathbf{v} = K\mathbf{v}\cdot\mathbf{v} + M\mathbf{v}^{\perp}\cdot v = K|\mathbf{v}|^2$,
  - or $K = \mathbf{c}\cdot\mathbf{v}/|\mathbf{v}|^2 = c \cdot v \cdot v$. (v is the unit vector)
- Likewise, taking the dot product with $\mathbf{v}^{\perp}$
  - we obtain $\mathbf{c}\cdot\mathbf{v}^{\perp} = K\mathbf{v}\cdot\mathbf{v}^{\perp} + M\mathbf{v}^{\perp}\cdot v^{\perp} = M|\mathbf{v}^{\perp}|^2$
  - gives $M = \mathbf{c}\cdot\mathbf{v}^{\perp}/|\mathbf{v}^{\perp}|^2$.
- Answers to the original questions: $M\mathbf{v}^{\perp}$, $K\mathbf{v}$, and both.

Practice Question

- Find the projection of the vector c=<6, 4> onto v=<1,2>
- How far is the point C=(6, 4) from the line that passes through A=(1, 1) and B=(4, 9)?

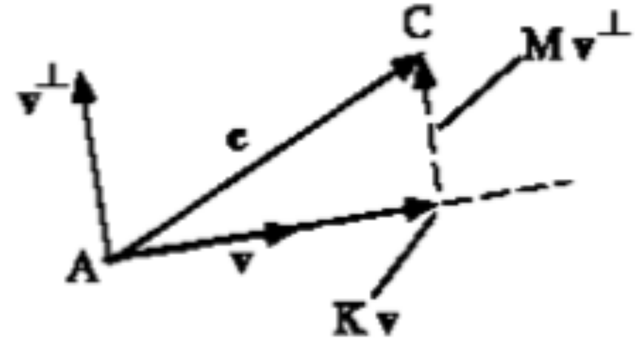
$K = \mathbf{c}\cdot\mathbf{v}/|\mathbf{v}|^2$
$M = \mathbf{c}\cdot\mathbf{v}^{\perp}/|\mathbf{v}^{\perp}|^2$



Practice Question

- Find the projection of the vector c=<6, 4> onto v=<1,2>
- How far is the point C=(6, 4) from the line that passes through A=(1, 1) and B=(4, 9)?

$K = \mathbf{c \cdot v}/|\mathbf{v}|^2$
$M = \mathbf{c \cdot v^{\perp}}/|\mathbf{v^{\perp}}|^2$



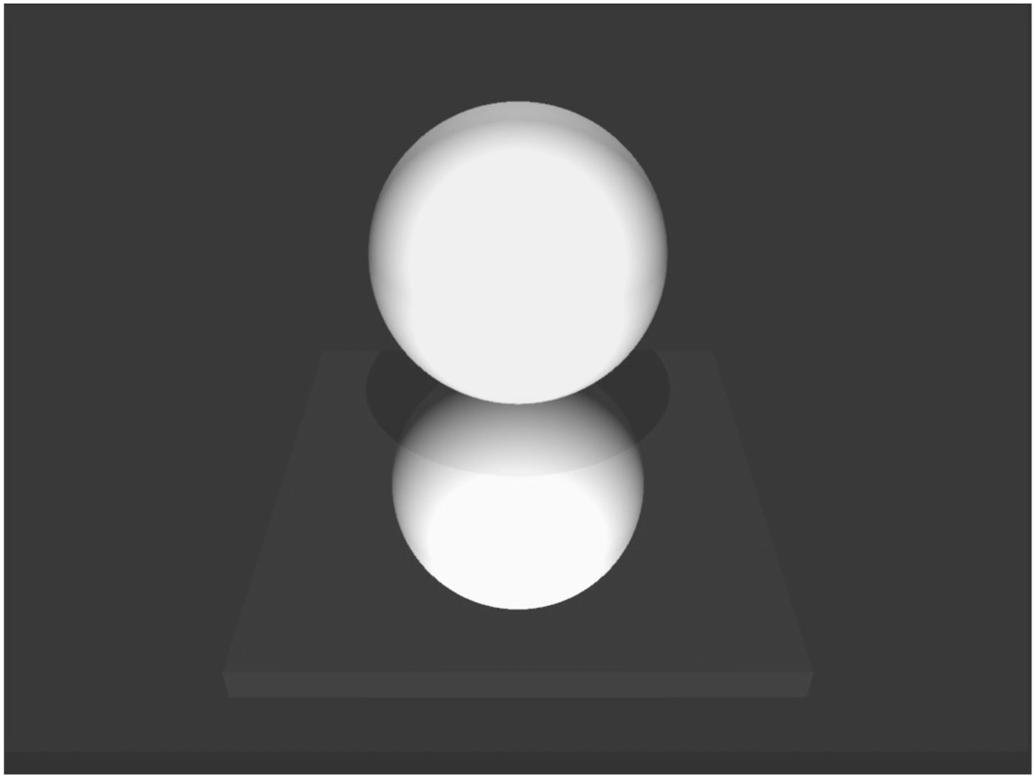
1) $K = \mathbf{c \cdot v}/|\mathbf{v}|^2$

$Kv = \mathbf{c} \cdot \mathbf{v} / |\mathbf{v}|^2 \cdot \mathbf{v} = \langle 6, 4 \rangle \cdot \mathbf{\langle 1, 2 \rangle / 5} \cdot \mathbf{\langle 1, 2 \rangle} = 14/5 \cdot \mathbf{\langle 1, 2 \rangle} = \langle 14/5, 28/5 \rangle$

2) $c = C - A = (6, 4) - (1, 1) = \langle 5, 3 \rangle$
$v = B - A = (4, 9) - (1, 1) = \langle 3, 8 \rangle$
$v^{\perp} = \langle -8, 3 \rangle$
$M \cdot v^{\perp} = c \cdot v^{\perp} / (v^{\perp})^2 \cdot v^{\perp} = \langle 5, 3 \rangle \cdot \langle -8, 3 \rangle / 73 \cdot \langle -8, 3 \rangle$
$= -31/73 \cdot \langle -8, 3 \rangle = \langle 248/73, -93/73 \rangle$

Reflections

- When a billiard ball hits the wall edge of a table.
- A reflection occurs when light hits a shiny surface (below)

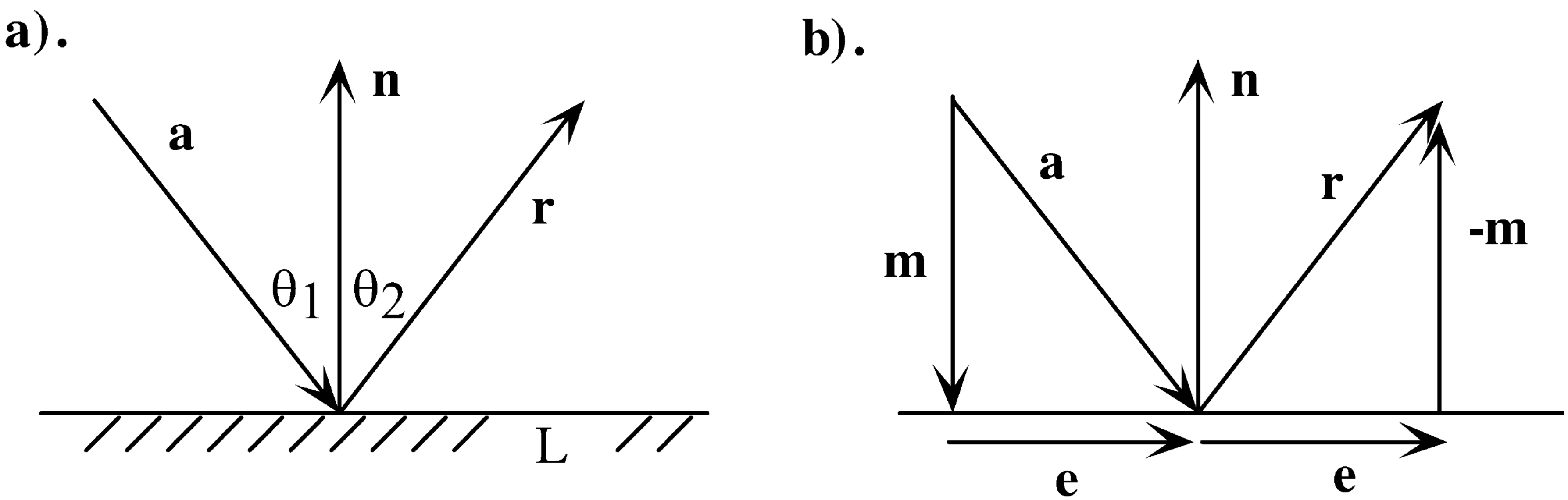


Reflections

- When light reflects from a mirror, the angle of reflection must equal the angle of incidence:

  $\theta_1 = \theta_2.$

- Vectors and projections allow us to compute the new direction r, in either two-dimensions or three dimensions.



Reflection – dot product

- The illustration shows

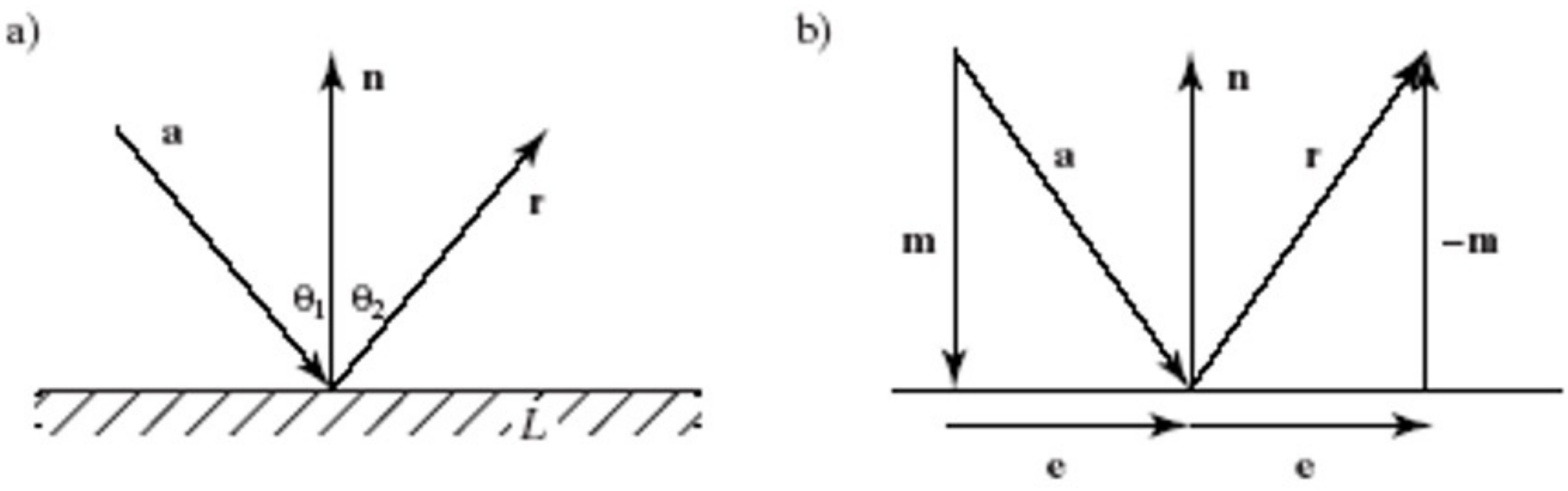
  e = a – m

  r = e – m = a – 2m

  m = [(a·n)/|n|²]n = $(a\cdot \hat{n})\cdot \ \hat{n}$

$$r = a - 2\left(\frac{a\cdot n}{|n|^2}\right)n$$

$$= a - 2(a\cdot \hat{n})\cdot \hat{n}$$



Practice Question

- Given: $a=(4, -2)$ and surface normal $n=(0, 3)$ what is a's reflected light about n?

$$\vec{a} = < 4, -2 >, \vec{n} = < 0, 3 >$$

$$\hat{n} = < 0, 1 >$$

$$\vec{r} = \vec{a} - 2(\vec{a} \cdot \hat{n}) \cdot \hat{n}$$

$$= < 4, -2 > -2(4 * 0 + (-2) * 1) \bullet < 0, 1 >$$

$= <4, -2> + 4*<0, 1>$

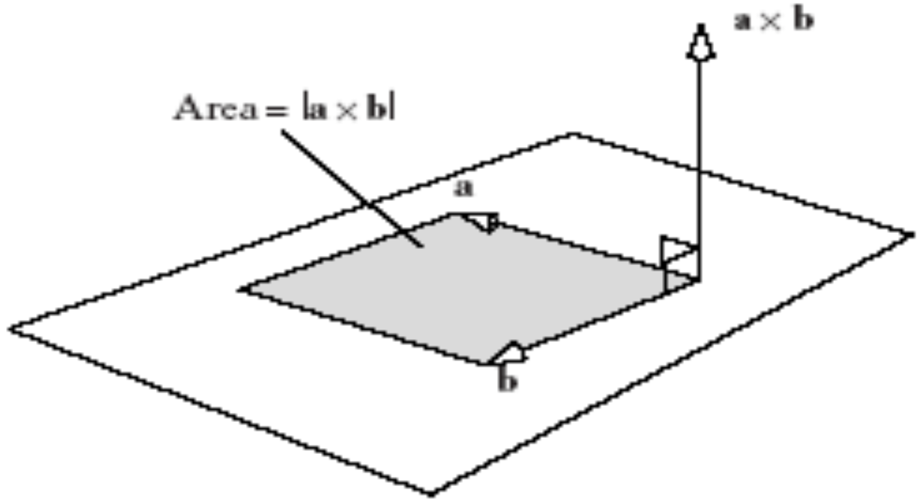
$=<4, -2> + <0, 4>$

$=<4, 2>$

Vector Cross Product (reminder)

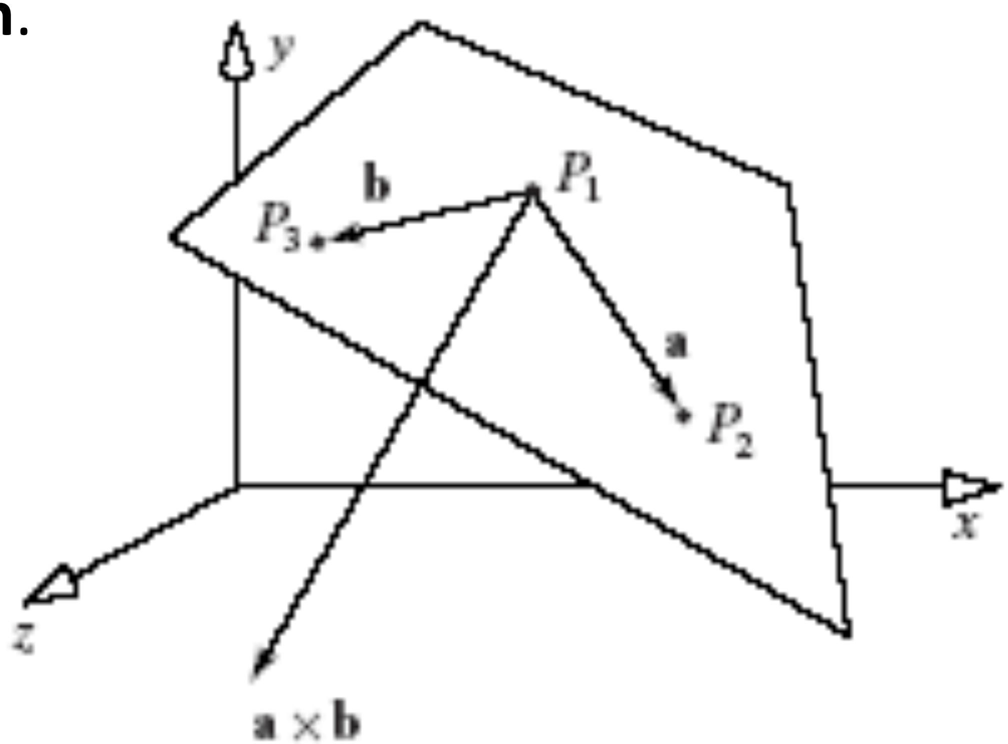
- **a** x **b** = $(a_y b_z - a_z b_y)$**i** + $(a_z b_x - a_x b_z)$**j** + $(a_x b_y - a_y b_x)$**k**.
- 3D Vectors Only
- The determinant below also gives the result:

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$



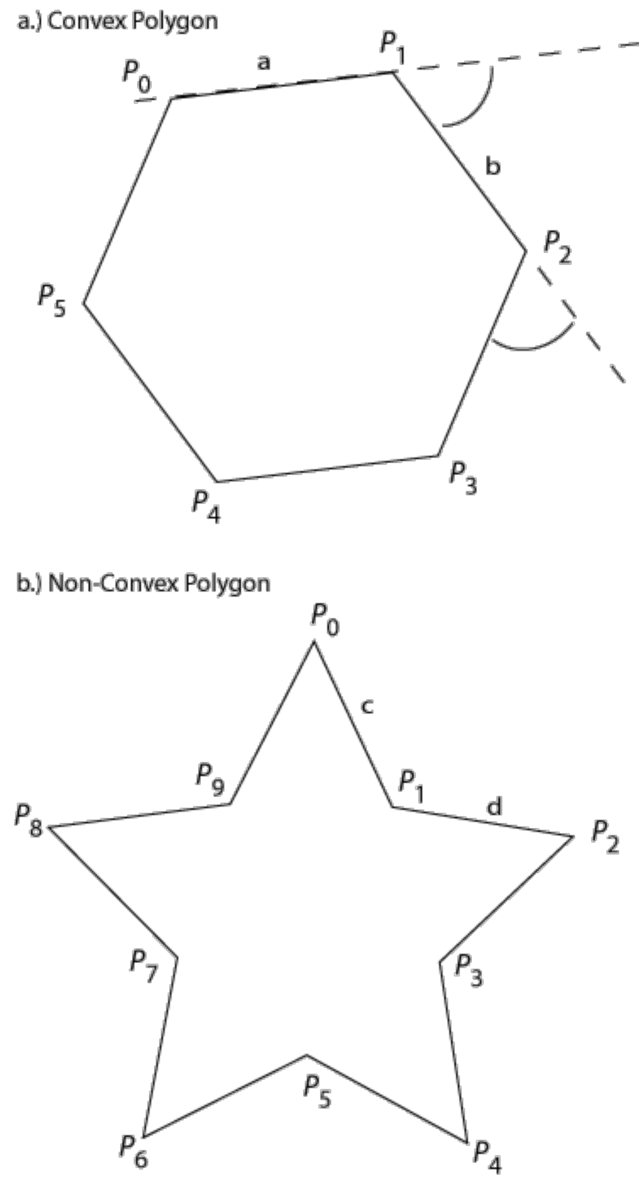
Application: Finding the Normal to a Plane

- Given any 3 non-collinear points P1, P2, and P3 in a plane, we can find a normal to the plane:
  - **a** = P2 – P1, **b** = P3 – P1, **n** = **a** x **b**. The normal on the other side of the plane is –**n**.



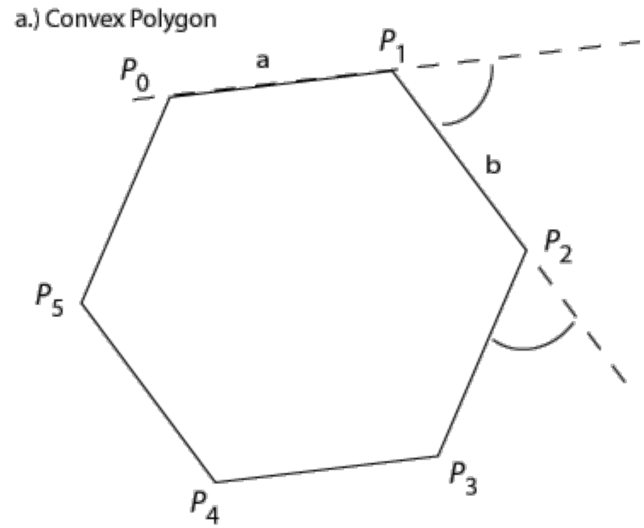
Convexity of Polygons

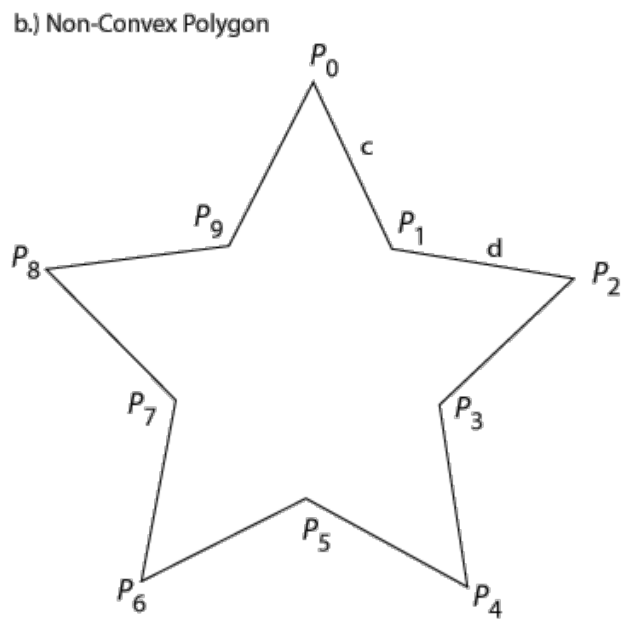
- Traversing around a <u>convex</u> polygon from one edge to the next
  - either a left turn or a right turn is taken
  - they all must be the same kind of turn
    - all left or all right
- An edge vector points along the edge of the polygon in the direction of travel.



Convexity of Polygons

- Take the cross product of each edge vector with the next forward edge vector.
- If all the cross products point into (or all point out of) the plane, the polygon is convex; otherwise it is not.





Representations of Key Geometric Objects

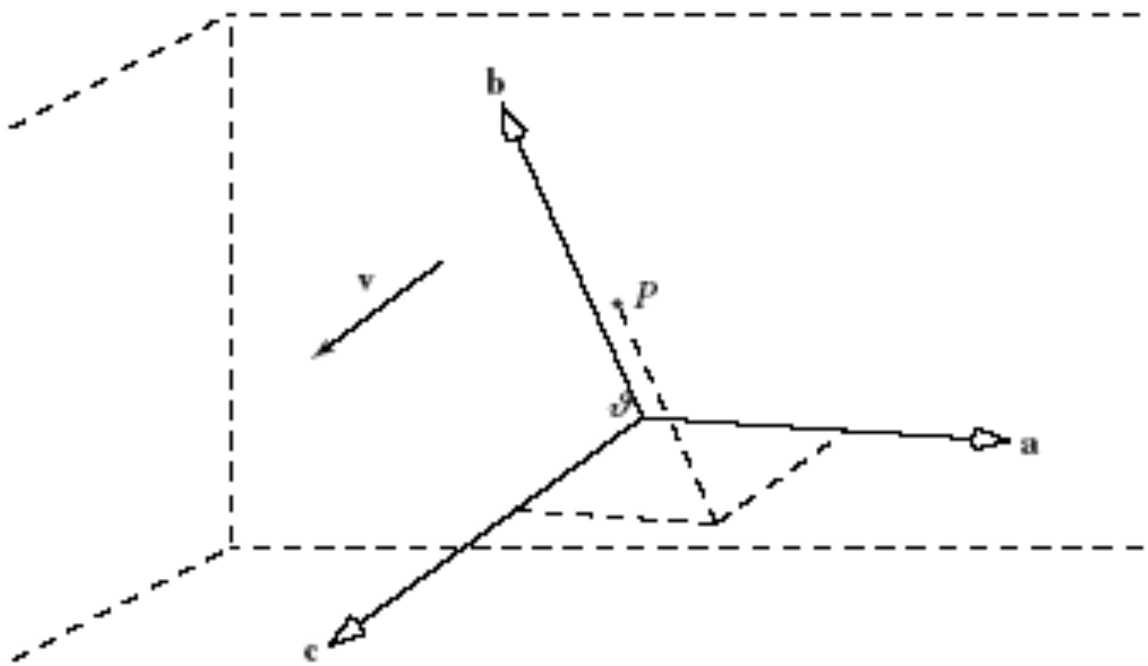
- Lines and planes are essential to graphics, and we must learn how to represent them
  - i.e., how to find an equation or function that distinguishes points on the line or plane from points off the line or plane.

Coordinate Systems and Frames

- A vector or point has coordinates in an underlying coordinate system.
- In graphics, we may have multiple coordinate systems
  - with origins located anywhere in space.
- We define a coordinate frame as a single point (the origin, $\mathcal{O}$) with 3 mutually perpendicular unit vectors: **a**, **b**, and **c**.

Coordinate Frames

- A vector $\mathbf{v}$ is represented by $(v_1, v_2, v_{3,})$ such that $\mathbf{v} = v_1\mathbf{a} + v_2\mathbf{b} + v_3\mathbf{c}$.
- A point P is represented by $(p_1, p_2, p_{3,})$, $P - \mathcal{O} = p_1\mathbf{a} + p_2\mathbf{b} + p_3\mathbf{c}$.



Homogeneous Coordinates

- It is useful to represent both points and vectors by the same set of underlying objects, ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathcal{O}$).
- A vector has no position, so we represent it as ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathcal{O}$)$(v_1, v_2, v_3, 0)^T$.
- A point has an origin ($\mathcal{O}$), so we represent it by ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathcal{O}$)$(p_1, p_2, p_3, 1)^T$.

Changing to and from Homogeneous Coordinates

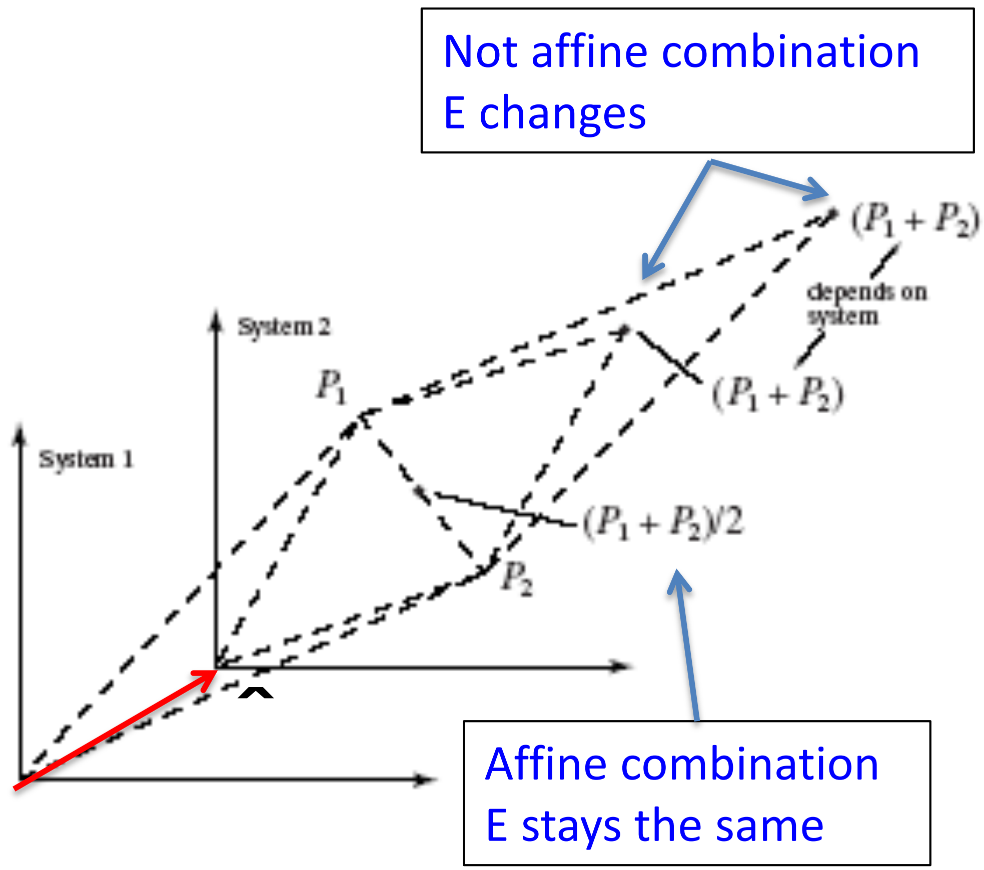
- To: if the object is a vector, add a 0 as the 4th coordinate;
  - if it is a point, add a 1.
- From: simply remove the 4th coordinate.
- OpenGL/WebGL uses 4D homogeneous coordinates for all its vertices.
  - If you send it a 3-tuple in the form (x, y, z), it converts it immediately to $(x, y, z, 1)$.
  - If you send it a 2D point (x, y), it first appends a 0 for the z-component and then a 1, to form $(x, y, 0, 1)$.
- All computations are done within OpenGL/WebGL in 4D homogeneous coordinates.

Combinations

- Why? Easy math
- Linear combinations of vectors and points:
  - The difference of 2 points is a vector: the fourth component is $1 - 1 = 0$
  - The sum of a point and a vector is a point: the fourth component is $1 + 0 = 1$
  - The sum of 2 vectors is a vector: $0 + 0 = 0$
  - A vector multiplied by a scalar is still a vector: $a \times 0 = 0$.
  - Linear combinations of vectors are vectors.

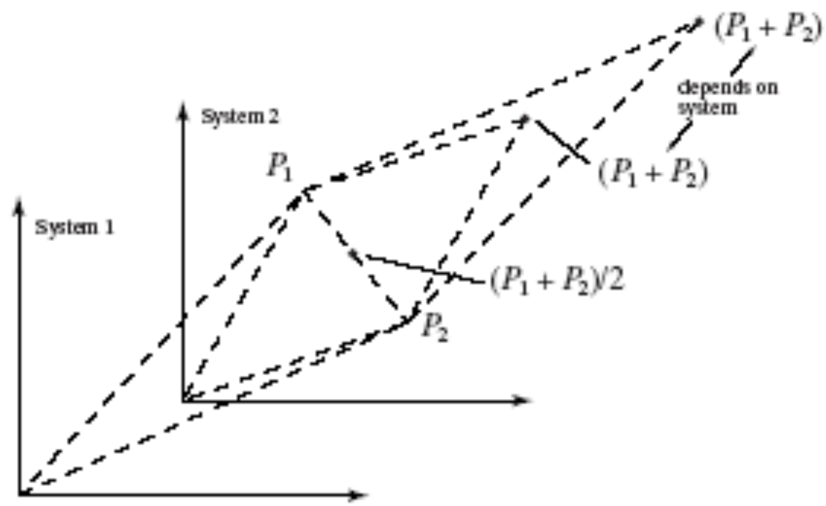
Combinations (2)

- The sum of 2 points: $E = a_1 \cdot P_1 + a_2 \cdot P_2$ is a point only if the points are part of an affine combination, so that $a_1 + a_2 = 1$. The sum is a vector only if $a_1 + a_2 = 0$.
- We require this rule to remedy the problem shown at right:



Combinations (3)

- If we form *any* linear combination of two points, say $E = fP + gR$, when $f + g$ is different from 1, a problem arises if we translate the origin of the coordinate system.
- Suppose the origin is translated by vector **u**, so that P is altered to $P + \mathbf{u}$ and R is translated to $R + \mathbf{u}$.
- If E is a point, it must be translated to $E' = E + \mathbf{u}$.
- But we have $E' = fP + gR + (f + g)\mathbf{u}$, which is *not* $E + \mathbf{u}$ unless $f + g = 1$.



Point + Vector

- Suppose we add a point A and a vector that has been scaled by a factor t:
  - The result is a point, P = A + t**v**.
- Now suppose **v** = B – A, the difference of 2 points, then: P = tB + (1-t)A,
  - P is an affine combination of two points, A and B
  - P is always on the line connecting A and B
  - The position of P on line AB is proportional to t

Linear Interpolation of 2 Points

- P = (1-t)A + tB is a linear interpolation (lerp or tween) of 2 points. This is very useful in graphics in many applications,
  - P_x (t) provides an x value that is fraction t of the way between A_x and B_x. (Likewise P_y, P_z).

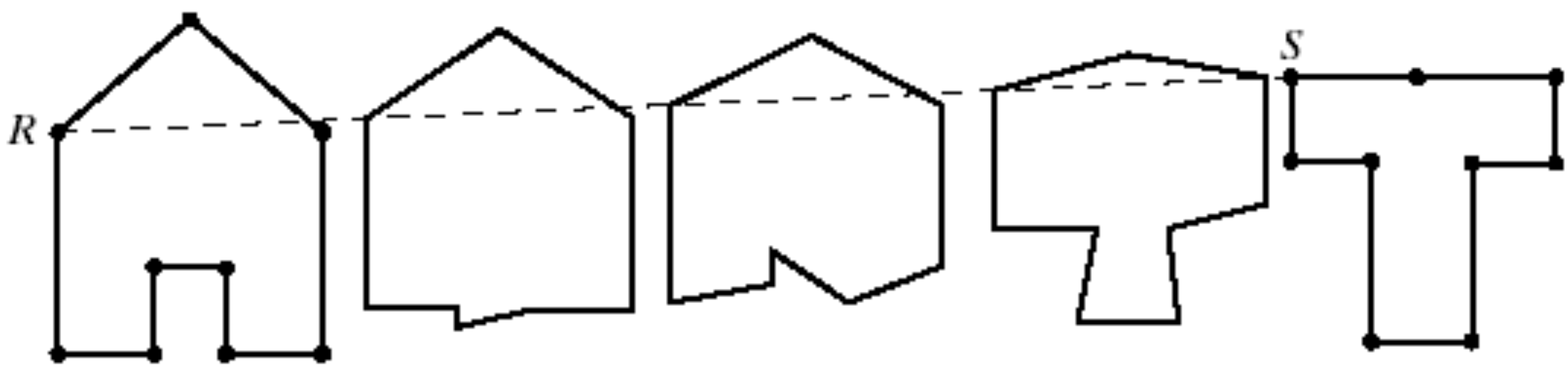
```
float Tween (float A, float B, float t)
{
   return   A + (B – A) * t;  // return float
}
```

Tweening and Animation

- Tweening takes 2 polylines and interpolates between them (using lerp) to make one turn into another (or vice versa).
- We are finding here the point P(t) that is a fraction t of the way along the straight line (not to be drawn) from point A to point B.
- To start, it is easiest if you use 2 polylines with the same number of lines.

Tweening

- We use polylines A and B, each with n points numbered 0, 1, …, n-1.
- We form the points $P_i(t) = (1-t)A_i + tB_i$, for t = 0.0, 0.1, …, 1.0 (or any other set of t in [0, 1]), and draw the polyline for P_i.



Use of Tweening in animation

- In films, artists draw only the key frames of an animation sequence (usually the first and last).
  - Tweening is used to generate the in-between frames.

  - Tweening demo

Practice Questions

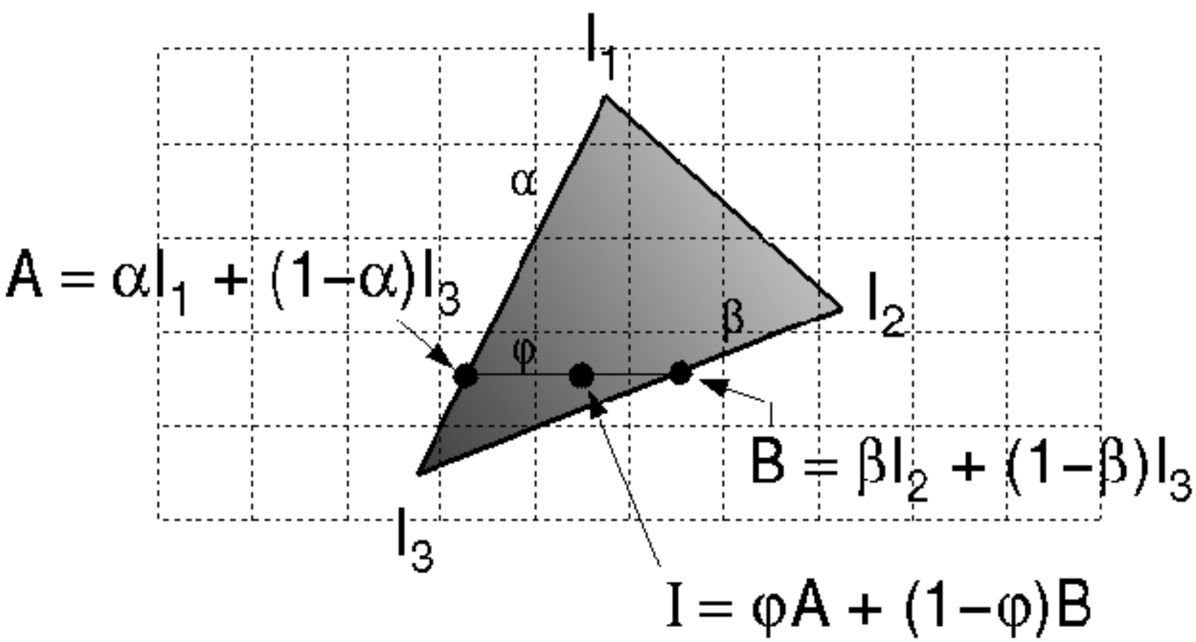
- What is the effect of tweening when all of the points A_i in polyline A are the same? How is polyline B distorted in its appearance in each tween?
- Polyline A is a square with vertices (1, 1), (-1, 1), (-1, -1), (1, -1) and polyline B is a wedge with vertices (4, 3), (5, -2), (4, 0), (3, -2). Sketch the shape P(t) for t=-1, -0.5, 0.5, and 1.5.

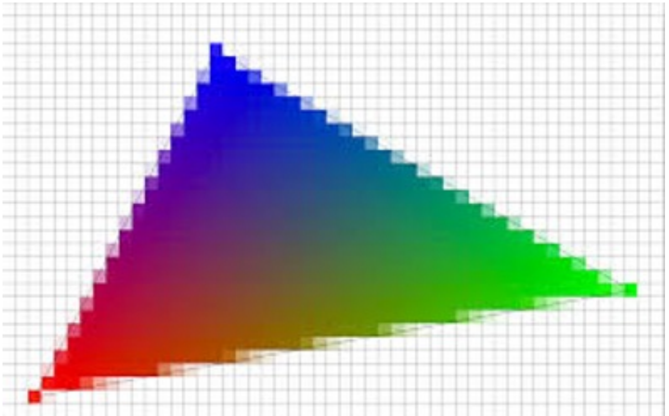
Other uses of Interpolation

Gouraud Shading

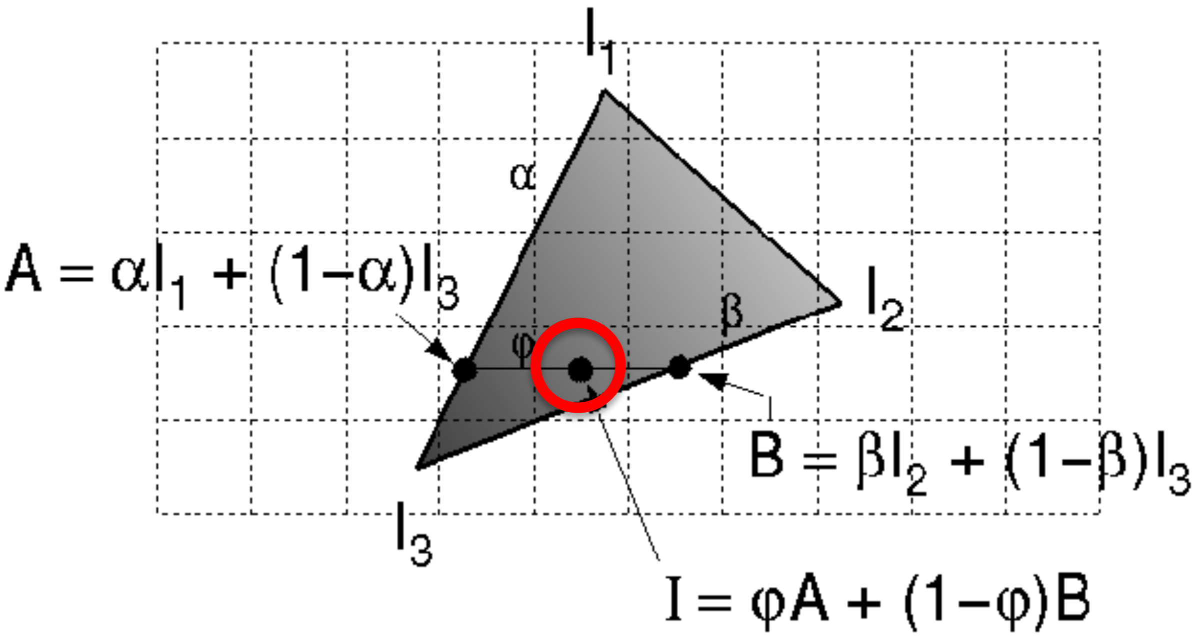
Bilinearly interpolate colors at vertices

down and across scan lines





Practice Question



I_1 color [1, 0, 0, 1) red
I_2 color (0, 1, 0, 1) green
I_3 color (1, 0, 1, 1) yellow

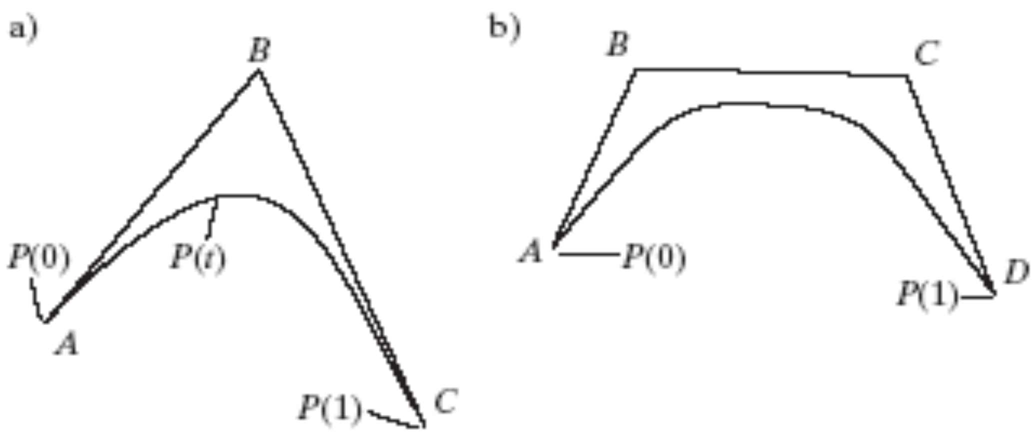
α = 0.8, β = 0.4, ϕ=0.6
What is the color of at point I, circled in red?

Answer: (0.84, 0.16, 0.36, 1)

Other uses of Tweening

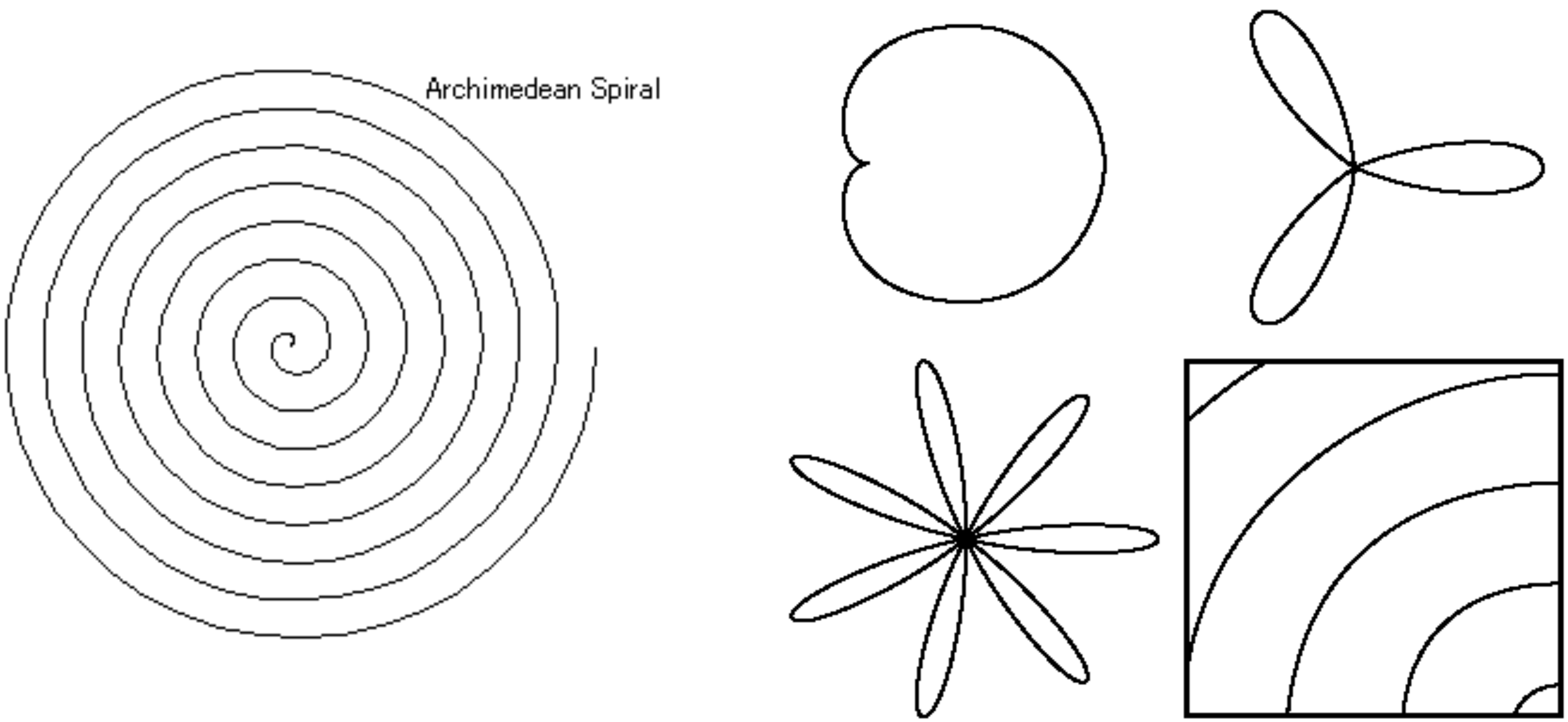
- We want a smooth curve that passes through or near 3 points (A, B, and C). We expand $((1\text{-}t) + t)^2$ and write: $P(t) = (1\text{-}t)^2A + 2t(1\text{-}t)B + t^2C$
  - This is called the Bezier curve for points A, B, and C.
  - It can be extended to 4 points by expanding

  $((1\text{-}t) + t)^3$ and using each term as the coefficient of a point.



Shapes as Parametric Curves

- For drawing purposes, parametric forms circumvent all of the difficulties of implicit and explicit forms.
- Cardioid, 2 rose curves, Archimedean spiral



Polar Coordinates Parametric Form

- General form: $x = f(\theta)*\cos(\theta)$

  $y = f(\theta)*\sin\theta$
- cardioid: $f(\theta) = K*(1 + \cos(\theta))$, $0 \le \theta \le 2\pi$
- rose: $f(\theta) = K* \cos(n*\theta)$, $0 \le \theta \le 2n\pi$, where n is number of petals (n odd) or twice the number of petals (n even)
- spirals: Archimedean, $f(\theta) = K\theta$

  Logarithmic, $f(\theta) = Ke^{a\theta}$
- K is a scale factor for the curves.

Polar coordinates Parametric Form

- conic sections (ellipse, hyperbola, circle, parabola): $f(\theta) = (1 \pm e \cos \theta)^{-1}$
  - e is eccentricity:

    1 : parabola;

    0 : circle;

    between 0 and 1, ellipse;

    greater than 1, hyperbola