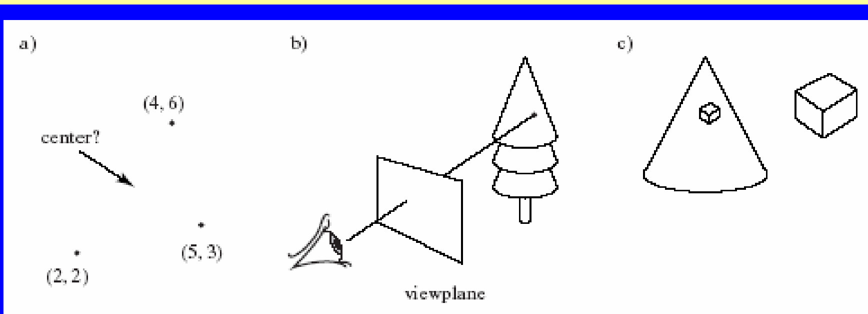
Prof. Reuven Aviv
Department of Computer Science
Tel Hai Academic College
Computer Graphics

Vector Mathematics and Clipping

Slides adapted from F. Hill and S. Kelley, Computer Graphics

Graphics Problems solved by Vectors

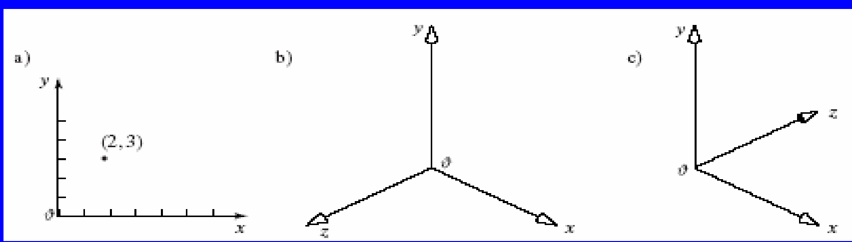
- Where is the center of the circle through 3 points?
- What shape appears on the viewplane, and where?
- Where does the reflection of the cube appear on the shiny cone, and what is its exact shape



Points and vectors

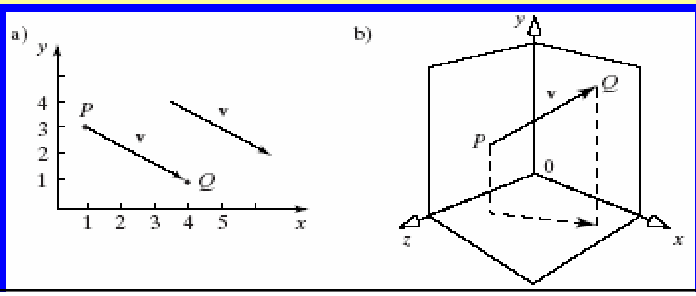
Basics of Points and Vectors

- *What's the diff between vector and point?*
- In both cases, they are described by 2/3 coordinates
  - relative to some coordinate system
- vector has length & direction, but no position
- A point has position but no length or direction
- A scalar has only size (a number)



Displacement Vectors

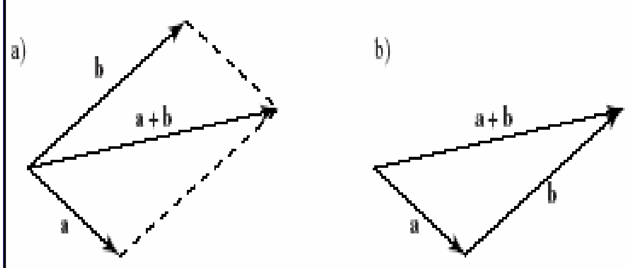
- *What is the displacement vector **v** from **P** = (1, 3) to **Q** = (4, 1)*
  - **v** = (3, -2), calculated by subtracting the coordinates individually (**Q** – **P**).
- **v:** no position, the 2 **v** arrows are same vector.
- Similarly for points and displacement vectors in 3D

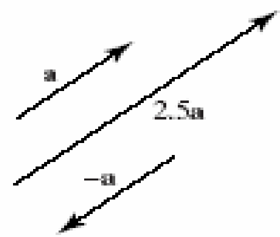


Point & Vector Operations

- The difference between 2 points is a vector:
  - **v** = **Q** – **P**.
- The sum of a point and a vector is a point:
- **P** + **v** = **Q**.
- We represent an n-dimensional point or a vector by an n-tuple of their components
  - **v** = (v_x, v_y, v_z)
  - **P** = (P_x, P_y, P_z)

Vectors: Addition and multiplication by scalar

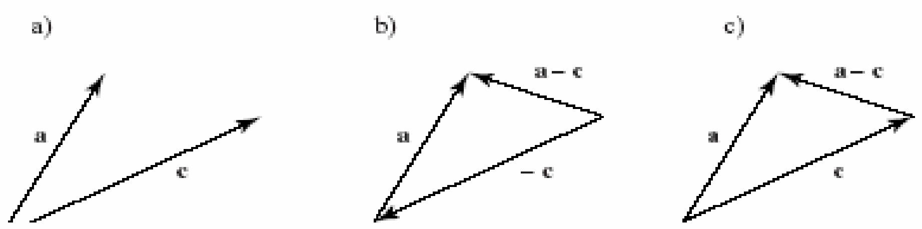




- **a** + **b**
  - **Move tail of b to head of a**
  - **Draw vector a+b from tail of a to head of b**

Subtraction of Vectors

- Subtracting **c** from **a** = adding **a** and (-**c**),
  - Move tail of **a** to the head of **-c**
- draw a vector from the tail of **–c** to head of **a**

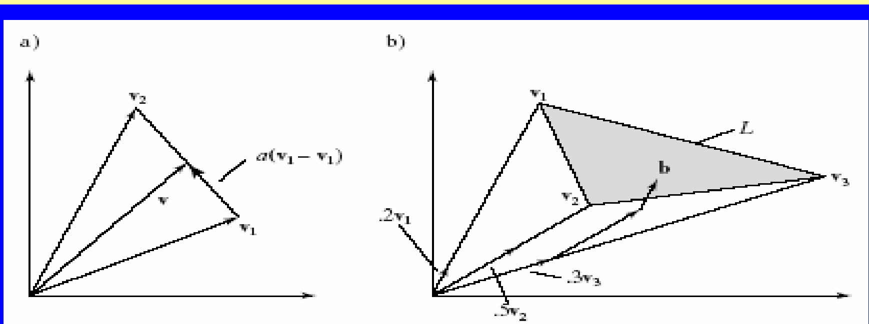


Affine and Convex Combinations of Vectors

- $\Sigma\alpha_i\underline{\mathbf{v}}_i$ is an affine combination if $\Sigma\alpha_i = 1$
  - $3\underline{\mathbf{a}} + 2\underline{\mathbf{b}} - 4\underline{\mathbf{c}}$ is an affine combination of $\underline{\mathbf{a}}$, $\underline{\mathbf{b}}$, and $\underline{\mathbf{c}}$
  - $3\underline{\mathbf{a}} + \underline{\mathbf{b}} - 4\underline{\mathbf{c}}$ is not.
  - $(1-t)\underline{\mathbf{a}} + t\underline{\mathbf{b}}$ is an affine combination of $\underline{\mathbf{a}}$ and $\underline{\mathbf{b}}$.
- An affine combination is a convex combination if $\alpha_i \geq 0$ for $1 \leq i \leq m$.
  - $.3\underline{\mathbf{a}}+.7\underline{\mathbf{b}}$ is a convex combination of **a** and **b**,
  - $1.8\underline{\mathbf{a}} - .8\underline{\mathbf{b}}$ is not.

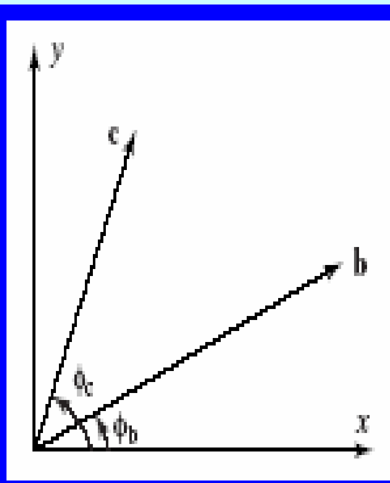
The Set of All Convex Combinations of 2 or 3 Vectors

- $\mathbf{v} = (1-\alpha)\mathbf{v}_1 + \alpha\mathbf{v}_2$, $0 <= \alpha <= 1$, gives the set of all convex combinations of $\mathbf{v}_1$ and $\mathbf{v}_2$.
  - $\underline{\mathbf{v}} = \underline{\mathbf{v}}_1 + \alpha(\underline{\mathbf{v}}_2 - \underline{\mathbf{v}}_1)$
- These are the vectors shown in the figure
- *Why?*



Application of dot (scalar) product: Angle Between 2 Vectors

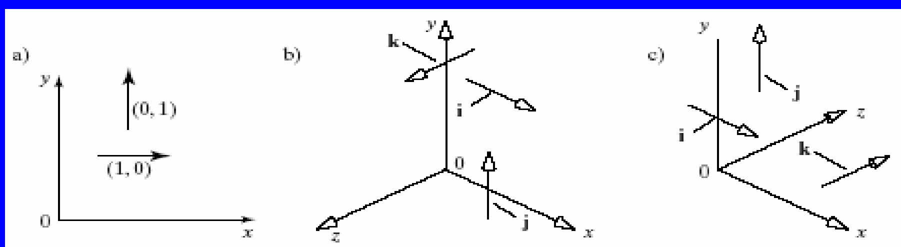
- $\underline{\mathbf{b}} = (|\mathbf{b}| \cos \varphi_b, |\mathbf{b}| \sin \varphi_b)$
- $\underline{\mathbf{c}} = (|\mathbf{c}| \cos \varphi_c, |\mathbf{c}| \sin \varphi_c)$
- $\underline{\mathbf{b}} \cdot \underline{\mathbf{c}} = |\mathbf{b}||\mathbf{c}| \cos \varphi_c \cos \varphi_b + |\mathbf{b}||\mathbf{c}| \sin \varphi_b \sin \varphi_c$
- $= |\mathbf{b}||\mathbf{c}| \cos (\varphi_c - \varphi_b)$
- $|\mathbf{b}||\mathbf{c}| \cos \theta$
- θ is the smaller angle between $\underline{\mathbf{b}}$, $\underline{\mathbf{c}}$



$$\cos(\theta) = \hat{\mathbf{b}} \cdot \hat{\mathbf{c}}$$

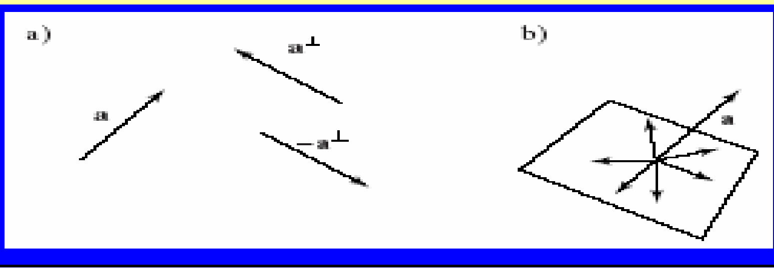
Standard Unit Vectors

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



Finding a 2D "Perp" Vector

- For vector $\underline{\mathbf{a}} = (a_x, a_y)$, define vector $\underline{\mathbf{a}^\perp} = (-a_y, a_x)$
- $\underline{\mathbf{a}^\perp}$ perpendicular to $\underline{\mathbf{a}}$ in the *counterclockwise* sense
- $-\underline{\mathbf{a}^\perp}$ perpendicular to $\underline{\mathbf{a}}$ in the *clockwise* sense
- *Who are the perp vectors in 3D?*
- any vector in the plane perpendicular to $\underline{\mathbf{a}}$ is a "perp" vector.

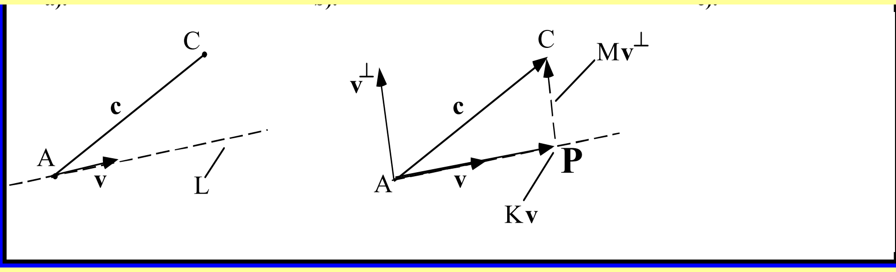


Properties of the perp vectors $\perp$

- *Who is $(\underline{a} \pm \underline{b})^\perp$*
- $\underline{\mathbf{a}^\perp} \pm \underline{\mathbf{b}^\perp}$
- $(s\underline{\mathbf{a}})^\perp = s(\mathbf{a}^\perp)$
- $(\mathbf{a}^\perp)^\perp = -\mathbf{a}$
- *How much is $(a^\perp \cdot b)$*
- $= -a_y b_x + a_x b_y = (-\mathbf{b}^\perp \cdot \mathbf{a})$
- $\mathbf{a}^\perp \cdot \mathbf{a} = a \cdot \mathbf{a}^\perp = 0$;
- $|\mathbf{a}^\perp| = |\mathbf{a}|$;

Vector Decomposition

- Vectors: $\underline{\mathbf{c}} = \mathbf{C} - \mathbf{A}$; $\underline{\mathbf{v}}$ along line L from **A**
- decompose $\underline{\mathbf{c}}$ into a part along $\underline{\mathbf{v}}$ and a part along $\underline{\mathbf{v}}^{\perp}$



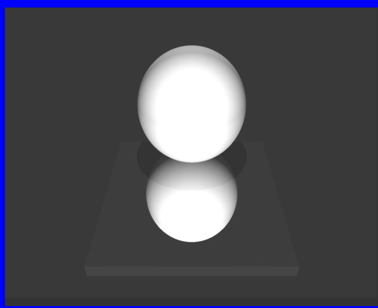
- *What's the distance of C from the line L?*
- *drop a perpendicular onto L from C, where does it hit L (point P)?*

Vector decomposition

- decompose $\underline{\mathbf{c}} = K\underline{\mathbf{v}} + M\underline{\mathbf{v}}^{\perp}$ some K, M
  - Distance **C** from line L is $|M|$
  - The hit point is $\mathbf{P} = \mathbf{A} + K\underline{\mathbf{v}}$
- *How can we find K?*
- Multiply by $\underline{\mathbf{v}}$ → $(\underline{\mathbf{c}} \cdot \underline{\mathbf{v}}) = K(\underline{\mathbf{v}} \cdot \underline{\mathbf{v}}) + M\cancel{(\underline{\mathbf{v}}^{\perp} \cdot \underline{\mathbf{v}})} = K|\mathbf{v}|^2$
  - $K = \underline{\mathbf{c}} \cdot \underline{\mathbf{v}} / |\underline{\mathbf{v}}|^2$.
  - $M = (\underline{\mathbf{c}} \cdot \underline{\mathbf{v}}^{\perp}) / |\underline{\mathbf{v}}|^2$
- $\underline{\mathbf{c}} = ((\underline{\mathbf{c}} \cdot \underline{\mathbf{v}}) / |\underline{\mathbf{v}}|^2)\underline{\mathbf{v}} + ((\underline{\mathbf{c}} \cdot \underline{\mathbf{v}}^{\perp}) / |\underline{\mathbf{v}}|^2)\, \underline{\mathbf{v}}^{\perp}$

Application of Projection: Reflections

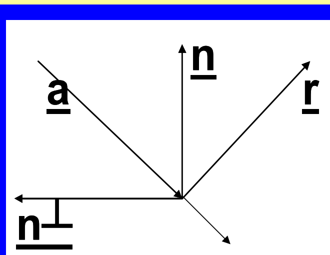
- A reflection occurs when light hits a shiny surface (below) or when a billiard ball hits the wall edge of a table.



Reflections (1)

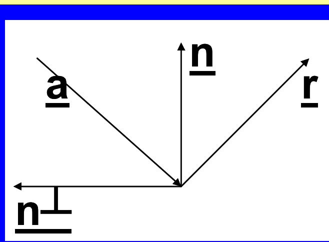
- incident light vector $\underline{\mathbf{a}}$, reflection vector $\underline{\mathbf{r}}$, $|\underline{\mathbf{a}}| = |\underline{\mathbf{r}}| = 1$

  What's the Law of reflection?
- angle of reflection must equal the angle of incidence
  - $r_x = a_x$, $r_y = -a_y$
  - $(\underline{\mathbf{r}}\cdot\underline{\mathbf{n}}) = -(\underline{\mathbf{a}}\cdot\underline{\mathbf{n}})$ $\quad (\underline{\mathbf{r}}\cdot\underline{\mathbf{n}}^{\perp}) = (\underline{\mathbf{a}}\cdot\underline{\mathbf{n}}^{\perp})$
- *How are we going to find $\underline{\mathrm{r}}$ from $\underline{\mathrm{a}}$?*



Reflection (2)

- Decompose $\underline{a}$ and $\underline{r}$ along $\underline{n}$ and $\underline{n}^{\perp}$
  - (1) $\underline{a} = (\underline{a}\cdot\underline{n})\underline{n} + (\underline{a}\cdot\underline{n}^{\perp})\underline{n}^{\perp}$
  - (2) $\underline{r} = (\underline{r}\cdot\underline{n})\underline{n} + (\underline{r}\cdot\underline{n}^{\perp})\underline{n}^{\perp}$
- Subtract (1) from (2) and use the Law of Reflection
  - $\underline{r} = \underline{a} - 2(\underline{a}\cdot\underline{n})\underline{n}$



Vector Cross Product (3D Vectors Only)

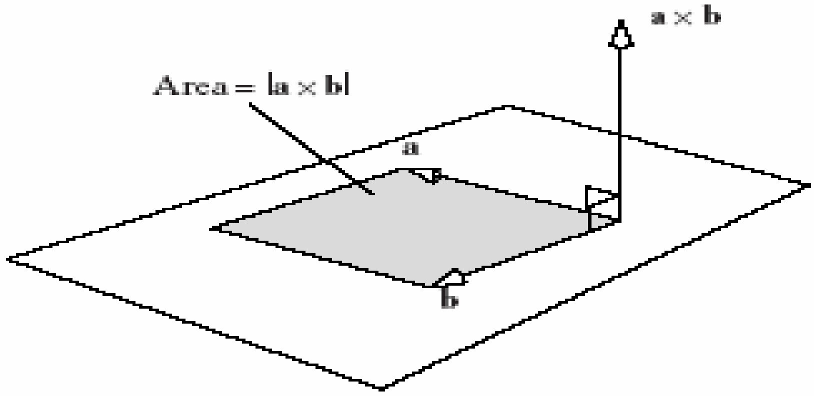
- *What is $\underline{a}$X$\underline{b}$?*
- $\underline{a} \times \underline{b} = (a_y b_z - a_z b_y)\underline{i} + (a_z b_x - a_x b_z)\underline{j} + (a_x b_y - a_y b_x)\underline{k}$.
- The determinant below also gives the result:
- *What is the direction of c?*
- $\underline{c} = \underline{a} \times \underline{b}$ perpendicular to $\underline{a}$ and $\underline{b}$.
  - direction of $\underline{c}$ is given by a right hand rule

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

Properties of cross product

- **a** · (**a** x **b**) = 0 *why?*
- **a** x **b** = |**a**||**b**| sin θ, where θ is the smaller angle between **a** and **b**.
- **a** x **b** = 0 if **a** and **b** point in the same or opposite directions, or if one or both has length 0.

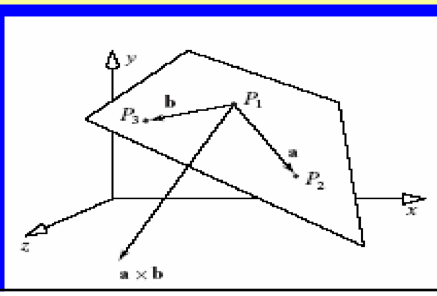
Geometric Interpretation of the Cross Product?



•**a** x **b** the area of the parallelogram formed by **a** and **b**

Application: Finding the Normal to a Plane

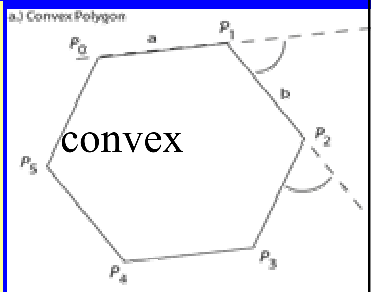
- 3 non-collinear points $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$, define a plane
  - find a normal to the plane
  - *How?*
- **Construct vectors:** $\underline{\mathbf{a}} = \mathbf{P_2} - \mathbf{P_1}$, $\underline{\mathbf{b}} = \mathbf{P_3} - \mathbf{P_1}$,
  - $\underline{\mathbf{n}} = \underline{\mathbf{a}} \times \underline{\mathbf{b}}$.
  - The normal on the other side of the plane is $-\underline{\mathbf{n}}$

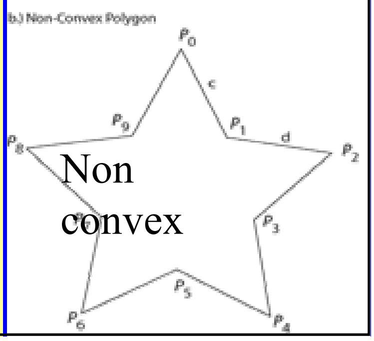


Where is $n^{\perp}$?

Convexity of Polygons

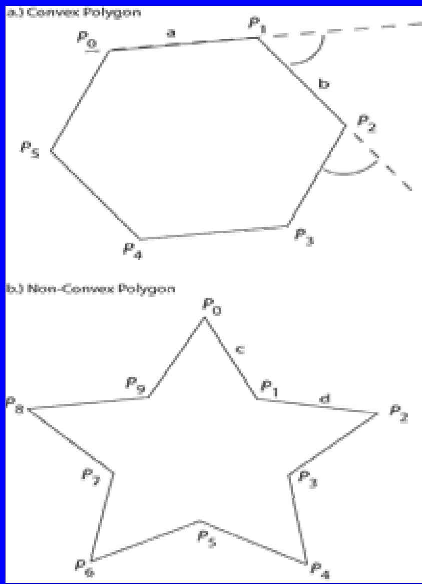
- *What is a convex polygon?*
- Traversing around a <u>convex</u> polygon, ***<u>all turns</u>*** are either left turn or right.
- An *<u>edge vector</u>* points along the edge of the polygon in the direction of travel.
- *How to test if a polygon is convex?*





Testing for convexity of a given polygon?

- Calculate 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.



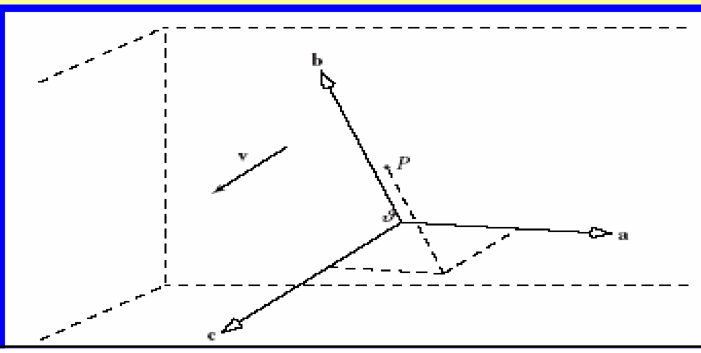
Homogeneous Coordinates

Coordinate Frame

- A vector or point has coordinates in an underlying Coordinate Frame (or Coordinate System)
- *What's the change in vector coordinates if the Coordinate system is translated?*
- no change: vector has no location
- *Is there a change in the coordinates of a point?*
- YES
- *How a Coordinate System is defined?*
- a Coordinate System is defined by a single point (the origin, ***O***) and 3 mutually perpendicular unit vectors: $\underline{\mathbf{a}}$, $\underline{\mathbf{b}}$, and $\underline{\mathbf{c}}$.

Coordinate Frames (2)

- A vector $\underline{\mathbf{v}}$ is represented by (v_1, v_2, v_3)
  - such that $\underline{\mathbf{v}} = v_1\underline{\mathbf{a}} + v_2\underline{\mathbf{b}} + v_3\underline{\mathbf{c}}$.
- A point **P** is represented by (v_1, v_2, v_3)
  - Such that $\mathbf{P} - \mathbf{O} = v_1\underline{\mathbf{a}} + v_2\underline{\mathbf{b}} + v_3\underline{\mathbf{c}}$.



Homogeneous Coordinates

- represent both points and vectors using same set of underlying basic objects, (**a**, **b**, **c**, ***O***).
- A vector has no position, we represent it as
  - $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \boldsymbol{O})(v_1, v_2, v_3, 0)^T = v_1\mathbf{a} + v_2\mathbf{b} + v_3\mathbf{c}$
- A point depends on origin (***O***), so we represent it by
  - $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \boldsymbol{O})(v_1, v_2, v_3, 1)^T = \boldsymbol{O} + v_1\mathbf{a} + v_2\mathbf{b} + v_3\mathbf{c}$
- (v_1, v_2, v_3, x) are the 4D homogeneous coordinates of vector or a point,
  - with $x = 0$ (vector) or $x = 1$ (point)

Changing to and from Homogeneous Coordinates

- To: vector object - put 0 as the 4th coordinate;
  - Point object - put a 1.
- From: remove the 4th coordinate.
- OpenGL uses 4D homogeneous coordinates
  - If you send it a 3D point 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 converts it to $(x, y, 0, 1)$
- All computations are done within OpenGL in 4D homogeneous coordinates.

Combinations (1)

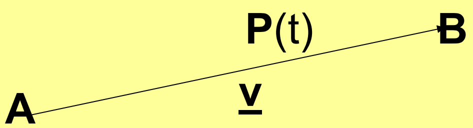
- Linear combinations of vectors and points:
- *What is the difference of* 2 **points**
- vector: the fourth component is 1 – 1 = 0
- *What is the sum of a* **point** *and a* **vector**
- a **point**: the fourth component is 1 + 0 = 1
- *What is the sum of 2* **vectors**
- a **vector**: 0 + 0 = 0
- *a* **vector** *multiplied by a scalar is a vector, s x 0 = 0.*
- All *linear combinations* of **vectors** are **vectors**. *Why?*
- *What is a linear combination of points?*

Combinations of Points

- $\Sigma\alpha_i \mathbf{P}_i$ is a point only if it is an affine combination
  - The fourth component of the combination $\Sigma\alpha_i = 1$
- *What happens if combination is not affine?*
- Consider combination $\mathbf{E} = \alpha\mathbf{P}_1 + \beta\mathbf{P}_2$ of points $\mathbf{P}_1$, $\mathbf{P}_2$
- Suppose the origin **O** moves by vector **u**
  - All points move by **u**; $\mathbf{P}_i \rightarrow \mathbf{P}_i + \mathbf{u}$, $\mathbf{E} \rightarrow \mathbf{E} + \mathbf{u}$
  - $\mathbf{E} \rightarrow \alpha\mathbf{P}_1 + \beta\mathbf{P}_2 + (\alpha + \beta)\mathbf{u} = \mathbf{E} + (\alpha + \beta)\mathbf{u}$ **!!!**

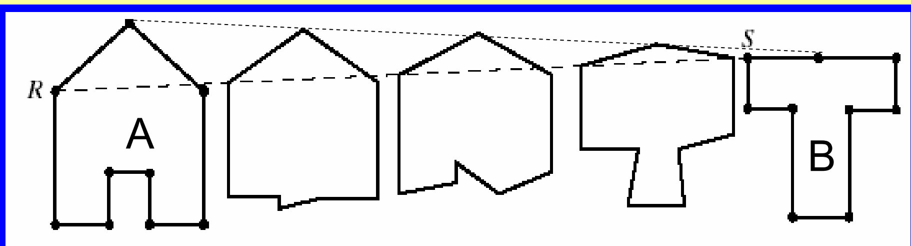
Point + Vector

- 2 points **A**, **B**. Vector $\underline{\mathbf{v}}$ = **B** - **A**
- *What is $P(t) = A + t\underline{v}$ for t between 0 and 1?*
- **P(t)** = **A** + t(**B** – **A**)
  - **P(t)** = (1-t)**A** + **tB** **P(0) = A; P(1) = B**
  - **P(t)** an affine combination of the points
  - **P**(t) Linear interpolation (lerp) of the two points

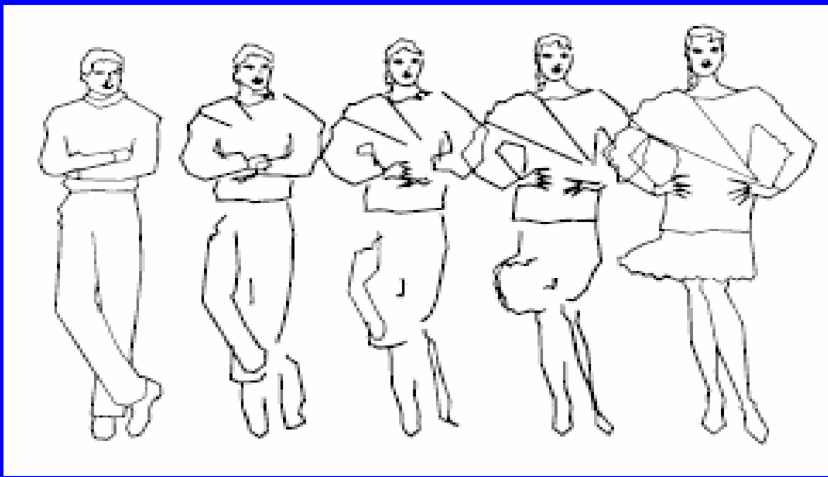


Application: Tweening

- Given polylines A, B
  - each with n vertices $\mathbf{A}_i$, $\mathbf{B}_i$, i = 1 to N
- for t = 0.0, 0.1, …, 1.0
  - Construct points $\mathbf{P}_i(t) = (1\text{-}t)\mathbf{A}_i + t\mathbf{B}_i$,
  - draw the polyline for $\mathbf{P}_i(t)$, one t after another
- *Result?*



Tweening Example

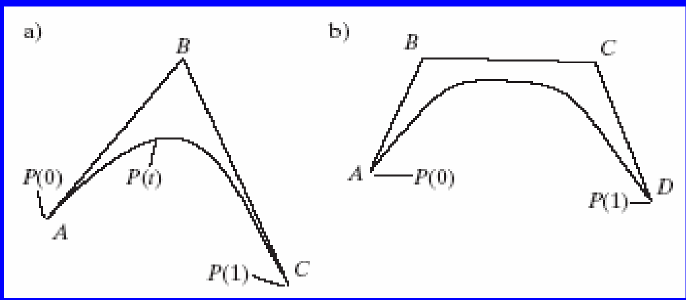


Uses of Tweening

- In films, artists draw only the key frames of an animation sequence (usually first, last).
  - generate in-between frames by tweening
- Preview: construct smooth curve that passes <u>through or near</u> 3 points, **A**, **B**, **C**
- expand $1 = [(1\text{-}t) + t]^2 = (1\text{-}t)^2 + 2(1\text{-}t) + t^2$
  - Partition 1 into three components that add to 1
- Construct affine combination
  - $\mathbf{P}(t) = (1\text{-}t)^2\mathbf{A} + 2t(1\text{-}t)\mathbf{B} + t^2\mathbf{C}$
  - $\mathbf{P(0) = A\ ;\ P(1) = C;}$
  - Bezier Curve with **A**, **B**, **C** <u>Control points</u>

Preview Bezier Curve

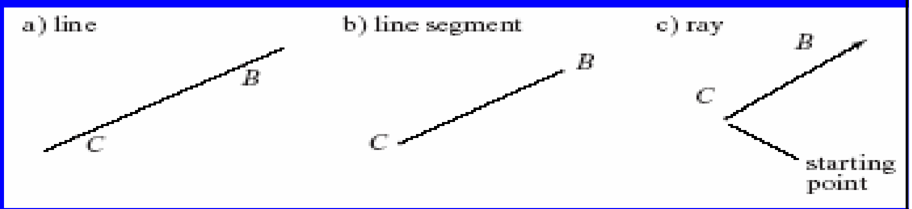
- *How to generalize to 4 points?*
- expanding $((1-t) + t)^3$ to four terms and using each term as the coefficient of a point
- *Generalization to n points…?*



Representations of lines, segments and planes

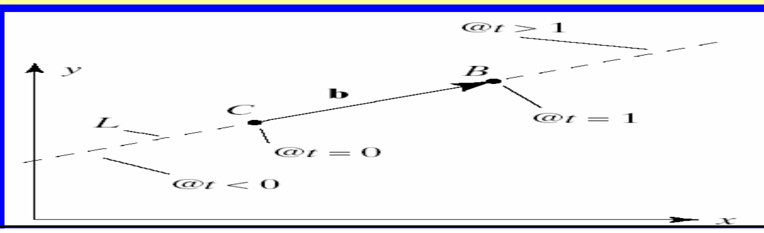
Lines, segments, rays

- A line passes through 2 points and is infinitely long.
- A (line) segment has 2 endpoints.
- *What is a ray?*
- A ray has a single endpoint.



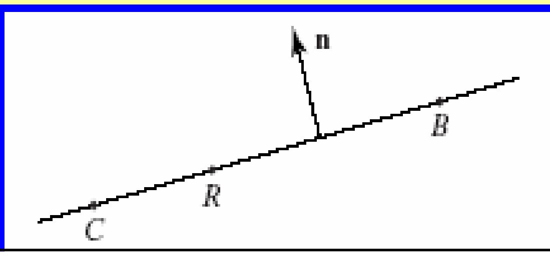
Parametric Representation

- Given 2 points, **B** and **C**, on a line.
- *Write a formula for any point P on line*
- **P**(t) = **C** + **b**t, where **b** = **B** – **C**
  - $-\infty \leq t \leq \infty$: line
  - $0 \leq t \leq 1$: line segment;
  - $-\infty \leq t \leq 0$ or $0 \leq t \leq \infty$: ray
- Note: **B**, **C**, **P**(t), **b** four components each
  - *What are the 4'th components of* **P(t)**, **b** *?*



Point-Normal (implicit) form (2D only)

- **B** and **C** fixed points, **R** = (x,y) any point on the line
  - Need implicit form: linear equation between x, y
- **b** = **B** – **C**
- **b**$^\perp$ = **n**, perpendicular to **R** – **C**
- **n** · (**R** – **C**) = 0 → $n_x(x-C_x) + n_y(y-C_y) = 0$
  - $-b_y(x-C_x) + b_x(y-C_y) = 0$

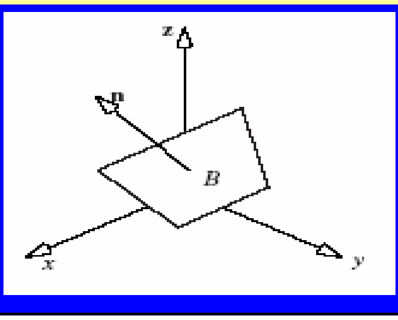


Changing Representations

- From point-normal to the canonical form:
- **n**·**R** - (**n**·**C**) = **0** → $n_x x + n_y y$ = (**n**·**C**)
  - $fx + gy = 1$
- From canonical form to the point-normal form
- $f x + g y = 1$ → $(f,g)\cdot(x,y)^T = 1$
  - $(f,g)\cdot(C_x,C_y) = 1$ where **C** is on the line
  - $(f, g)\cdot(x-C_x, y-C_y)^T = 0$
  - Hence **n** is (f, g) (or any multiple thereof).
- From point normal to parametric form
  - **n**·(**R** - **C**) = 0 → **R**(t) = **C**+ **b**t = **C** + **n**$^\perp$t

Representing Planes: Point-Normal Form

- **B** is a given (4 coordinates) point on the plane
- **P** = $(x, y, z, 1)^T$ any point on the plane
- **n** vector normal to the plane
- *What is the relation between these three?*
- $\mathbf{n}\cdot(\mathbf{P} - \mathbf{B}) = 0$

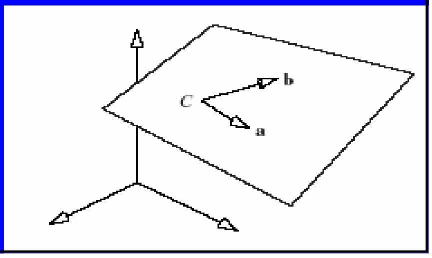


Plane: Parametric Form

- Given 3 non-collinear points on the plane **A**, **B**, and **C**.
  - $\underline{\mathbf{a}} = \mathbf{A} - \mathbf{C}$
  - $\underline{\mathbf{b}} = \mathbf{B} - \mathbf{C}$
- *Express a general point P(s,t) in the plane in terms of C, $\underline{a}$ and $\underline{b}$*

- $\mathbf{P}(s, t) = \mathbf{C} + s\underline{\mathbf{a}} + t\underline{\mathbf{b}}$
- Infinite plane:
  - $-\infty \le s \le \infty$ & $-\infty \le t \le \infty$
- *patch:*

$0 \le s \le 1$ & $0 \le t \le 1$

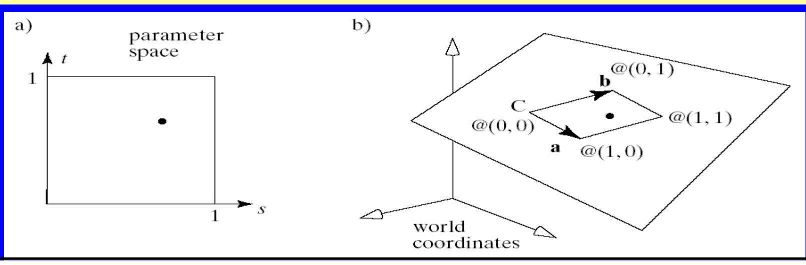


Planes: Parametric Form (2)

- P(s,t) = C + s**a** + t**b**
- **a** = **A** – **C**
- **b** = **B** – **C**
- *Express P(s,t) in terms of A, B, C*
- **P**(s, t) = s **A** + t **B** + (1 - s - t)**C**
- Any point on a plane is an affine combination of three non-collinear points on the plane

Patch: Parameter Space

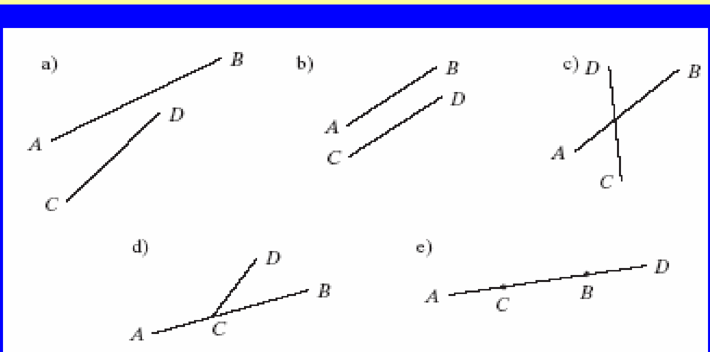
- s, t values from finite parameter space (a square)
- corners of patch at four corners of parameter space
  - **P**(0, 0) = **C** **P**(1, 0) = **C** + **a**
  - **P**(0, 1) = **C** + **b**; **P**(1, 1) = **C** + **a** + **b**
- *What happens if we change* **C** *?*
- patch translated, no change in shape or orientation



Intersection problems

Relations between 2 Line Segments in space

- They can miss each other (a and b), overlap in one point (c and d), or even overlap over some region (e). They may or may not be parallel.



Intersection of 2 Line Segments

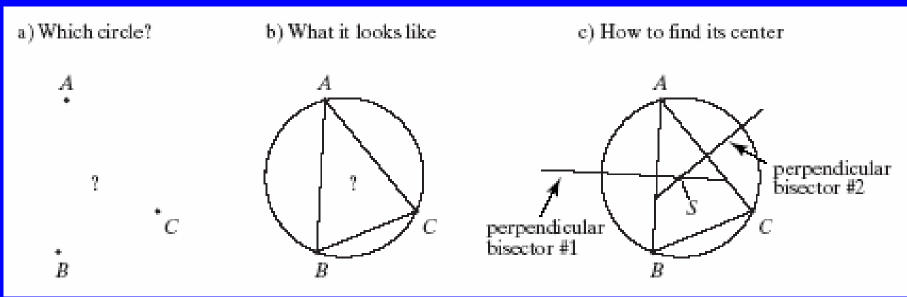
- Line segments (**A**, **B**) (**C**, **D**)
- *Find the intersection point*
- **P**(t) = **A** + $\underline{\mathbf{b}}$t, $\underline{\mathbf{b}}$ = **B** – **A**
- **Q**(u) = **C** + $\underline{\mathbf{d}}$u, $\underline{\mathbf{d}}$ = **C** – **D**
- Intersection: **P**(t*) = **Q**(u*)
  - $\underline{\mathbf{B}}$t* = $\underline{\mathbf{c}}$ + $\underline{\mathbf{d}}$u*, where $\underline{\mathbf{c}}$ = **C** – **A**
- *How we find the values of the t*, u* parameters*
- Taking dot product with $\mathbf{d}^{\perp}$: $(\underline{\mathbf{b}}\cdot\underline{\mathbf{d}}^{\perp})\, t^* = (\underline{\mathbf{c}}\cdot\underline{\mathbf{d}}^{\perp})$
- Taking dot product with $\mathbf{b}^{\perp}$: $-(\underline{\mathbf{c}}\cdot\underline{\mathbf{b}}^{\perp}) = (\underline{\mathbf{d}}\cdot\underline{\mathbf{b}}^{\perp})u^*$

Intersection of 2 Line Segments (2)

- Case1: $(\underline{\mathbf{b}}\cdot\underline{\mathbf{d}}^{\perp}) = 0$ (or $(\underline{\mathbf{d}}\cdot\underline{\mathbf{b}}^{\perp})$) no solution *why?*
- Segments are on Parallel lines or on same line
- Case 2: $(\underline{\mathbf{b}}\cdot\underline{\mathbf{d}}^{\perp}) \neq 0$
  - $t^* = (\underline{\mathbf{c}}\cdot\underline{\mathbf{d}}^{\perp}) / (\underline{\mathbf{b}}\cdot\underline{\mathbf{d}}^{\perp})$; $u^* = (-\,\underline{\mathbf{c}}\cdot\underline{\mathbf{b}}^{\perp}) / (\underline{\mathbf{d}}\cdot\underline{\mathbf{b}}^{\perp})$
- *What do we require from s*, t* ?*
- Real intersection: iff $0 \leq t^* \leq 1$ and $0 \leq u^* \leq 1$,
  - $\mathbf{P}(s^*, t^*) = \mathbf{A} + \underline{\mathbf{b}}((\underline{\mathbf{c}}\cdot\underline{\mathbf{d}}^{\perp}) / (\underline{\mathbf{b}}\cdot\underline{\mathbf{d}}^{\perp}))$

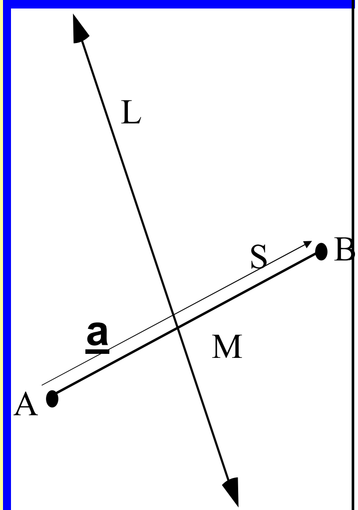
Application: Finding A Circle through 3 Points

- Given 3 points **A**, **B**, **C**
- *Where is the center* S *?*
  - center **S**: meeting point of perpendicular bisectors
  - radius r = |**A** – **S**|.



Parametric form of Perpendicular Bisector

- *Midpoint of segment AB?*
- **M** = ½ (**A** + **B**) = **A** + **(B-A)**/2
- **M** = **A** + **a**/2   **a** = **B-A**
- *direction of the PB?*
- Direction: (**B** – **A**)$^\perp$ = **a**$^\perp$
- *Parametric form of PB?*
- **P(t) = A** + **a**/2 + **a**$^\perp$ t
- **S** = **P**(t*)   intersection point



Circle through 3 Points (3)

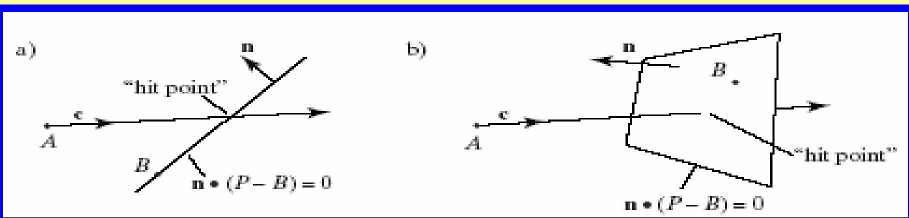
- $\underline{\mathbf{b}} = \mathbf{C} - \mathbf{B}$ $\underline{\mathbf{c}} = \mathbf{A} - \mathbf{C}$ $\underline{\mathbf{a}} + \underline{\mathbf{b}} + \underline{\mathbf{c}} = 0$
- PB of (A,B): $\mathbf{P}(t) = \mathbf{A} + \underline{\mathbf{a}} / 2 + \underline{\mathbf{a}}^{\perp} t$
- PB of (A, C): $\mathbf{Q}(u) = \mathbf{A} - \underline{\mathbf{c}} / 2 + \underline{\mathbf{c}}^{\perp} u$
- Intersection: $\mathbf{a}^{\perp} t^* = \underline{\mathbf{b}}/2 + \underline{\mathbf{c}}^{\perp} u^*$ *How we find t* ?*
- Multiply by $\underline{\mathbf{c}}$
  - $t^* = (½)(\underline{\mathbf{b}} \cdot \underline{\mathbf{c}})/(\underline{\mathbf{a}}^{\perp} \cdot \underline{\mathbf{c}})$;
- *where is S?*
- $\mathbf{S} = \mathbf{P}(t^*) = \mathbf{A} + \underline{\mathbf{a}}/2 + \underline{\mathbf{a}}^{\perp} t^*$

$$radius = |\mathbf{S} - \mathbf{A}|$$

$$radius = \frac{|\mathbf{a}|}{2}\sqrt{\left(\frac{\mathbf{b}\cdot\mathbf{c}}{\mathbf{a}^{\perp}\cdot\mathbf{c}}\right)^2 + 1}$$

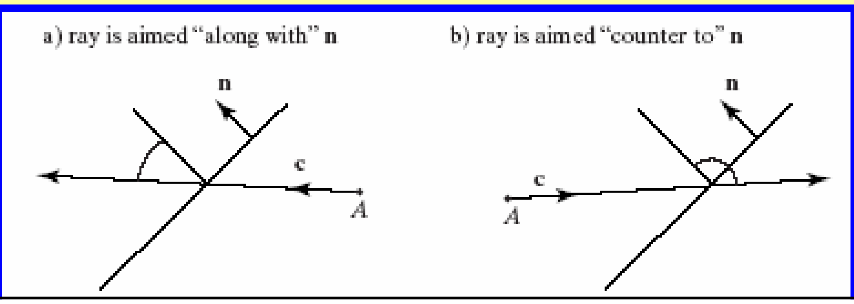
Ray Intersection with Line or Plane

- The hit point on the ray: $\mathbf{P}(t^*) = \mathbf{A} + \underline{\mathbf{c}}t^*$
- $\mathbf{P}$ satisfies the equation of the line/plane
  - $\underline{\mathbf{n}} \cdot (\mathbf{P}(t^*) - \mathbf{B}) = 0$ $\mathbf{B}$ some point on line/plane
  - $\underline{\mathbf{n}} \cdot (\mathbf{A} + \underline{\mathbf{c}}t^* - \mathbf{B}) = 0$ $\underline{\mathbf{n}}$ normal to line or plane
- $t^* = \underline{\mathbf{n}} \cdot (\mathbf{B} - \mathbf{A})/(\underline{\mathbf{n}} \cdot \underline{\mathbf{c}})$, iff $\underline{\mathbf{n}} \cdot \underline{\mathbf{c}} \neq 0$.
- *What if $\underline{\mathbf{n}} \cdot \underline{\mathbf{c}} = 0$?*



Ray Intersection with Line or Plane (2)

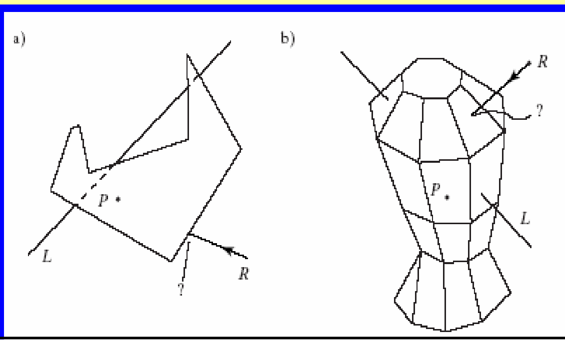
- $P(t^*) = A + \underline{c}t^* = A + \underline{c}(\underline{n} \cdot (B - A) / \underline{n} \cdot \underline{c})$
- If $\underline{n} \cdot \underline{c} > 0$, **c** and **n** make an angle of less than 90° with each other. **n** goes along with **c**
- If $\underline{n} \cdot \underline{c} < 0$, **c** and **n** make an angle of more than 90° with each other. **n** goes away from **c**



Polygon Clipping Problem

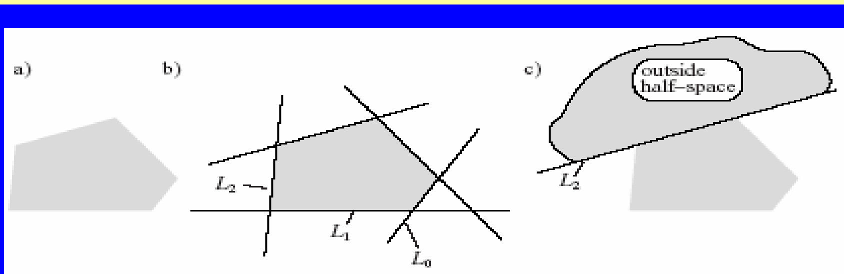
Polygon Clipping Problems

- 1. Where does a given ray line enters polygon?
- 2. Which part of a given ray line L lies inside / outside a polygon?
- 3. Which part of a <u>segment</u> lies inside a polygon



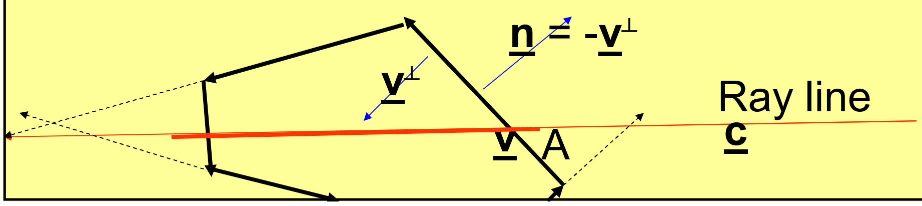
Convex Polygons and Polyhedrons

- Convex Polygons and Polyhedra: a simple case
- each line/plane divides space into two halves:
  - *inside half space*, where the polyhedron lies. *Outside*: shares no points with the polyhedron
- polyhedron: intersection of all the inside half spaces



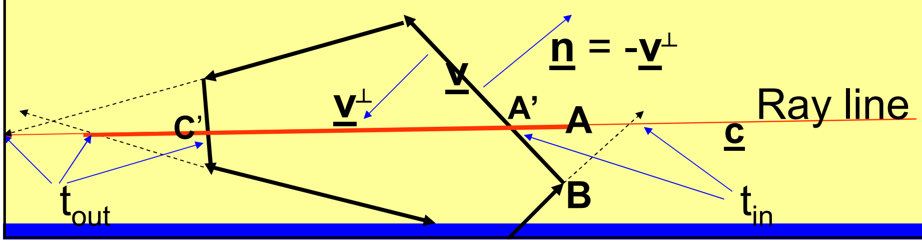
Line Intersection an edge of a Polygon

- Ray **P**(t) = **A** + **c**t   segment $0 <= t <= 1$
- **n** is the *outside normal* to the edge-line.
  - **n** =-**v**$^{\perp}$ = $(v_y, -v_x)$, **v** = (v_x, v_y) the edge vector
- line intersect each edge-line (except parallel)
  - if **n**·**c** < 0 line enter the inside half-space
  - if **n**·**c** > 0 line leave the inside half-space



Ray clipping on Polygon

- Calculate all in-points t_{in} & out-points t_{out}
- Largest t_{in} & smallest t_{out} are the intersection points of line with the actual polygon
- Compare these values with endpoints of ray segment
- The clipped ray is given by the (**A'**, **C'**) segment
- **A'** = **A** + **c***max(0, t_{in})
- **C'** = **A** + **c***min(1, t_{out})

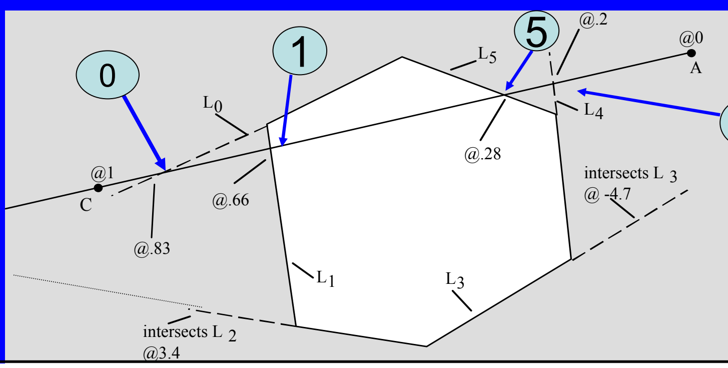


Cyrus Beck Algorithm – Ray Clip on Convex Polygon

- Ray : $\mathbf{A} + \underline{\mathbf{c}}t$ segment: $0 <= t <= 1$
- Initiate a "candidate interval" $[t_{in}, t_{out}] = [0..1]$ (or +/- ∞)
- For each edge-line with vertex **B**, normal $\underline{\mathbf{n}}$
  - calculate intersection point $t^* = \underline{\mathbf{n}} \cdot (\mathbf{B}-\mathbf{A}) / (\underline{\mathbf{n}} \cdot \underline{\mathbf{c}})$
  - in-point ($\underline{\mathbf{n}} \cdot \underline{\mathbf{c}} < 0$) or out-point ($\underline{\mathbf{n}} \cdot \underline{\mathbf{c}} > 0$)
  - If it's in-point set new t_{in} = max(previous t_{in}, t^*)
  - If it's out-point set new t_{out} = min(previous t_{out}, t^*)
  - If $t_{in} > t_{out}$, ray does not intersect the real polygon. Stop
- If the candidate interval is not empty then the segment from $\mathbf{A} + \underline{\mathbf{c}}t_{in}$ to $\mathbf{A} + \underline{\mathbf{c}}t_{out}$ is inside the polygon.

Example of Cyrus Beck Algorithm

- Test against L_0:
  - $t_{out} = 0.83$
- Test against L_1:
  - tout = 0.66
- Test against L2:
  - tout = 0.84 → tout = 0.66
- Test against L3:
  - tin = -4.7
- Test against L4:
  - tin 0.2
- Test against L5:
  - tin = 0.28



Example of Cyrus Beck Algorithm

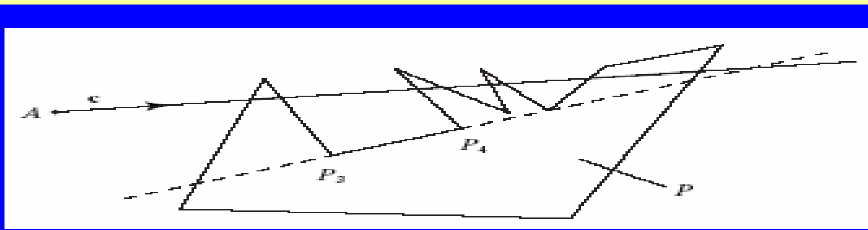
Line tested	T_{in}	T_{out}
0	0	0.83
1	0	0.66
2	0	0.66
3	0	0.66
4	0.2	0.66
5	0.28	0.66

3D Cyrus-Beck Clipping

- The Cyrus Beck clipping algorithm works in three dimensions in exactly the same way
- The polygon is a convex polyhedron
  - the edges are planes
- the ray segment is a line in 3D space.

Ray Clipping Arbitrary Polygons (2D)

- Much harder than for convex polygons.
  - Line may intersect polygon any even number of times, in in/out pairs
- *Which segments are inside? Outside?*
- *Consecutive pairs define in/out segments*
- Intersection at vertex? Counted as 2 intersections

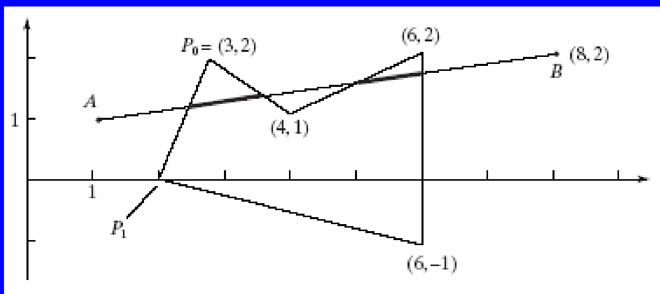


Clipping for Arbitrary Polygons (2)

- Ray $\mathbf{A} + \underline{\mathbf{c}}t$; Edge i: $\mathbf{P}_i + \underline{e}_i u$, $\underline{\mathbf{e}}_i = \mathbf{P}_{i+1} - \mathbf{P}_i$
- Polygon is closed, $P_N = P_0$
- calculate intersection points with edges
  - Intersection with edge i
  - $t_j = (\underline{\mathbf{e}}_i^{\perp} \cdot \underline{\mathbf{b}}_i) / (\underline{\mathbf{e}}_i^{\perp} \cdot \underline{\mathbf{c}})$, where $\underline{\mathbf{b}}_i = \mathbf{P}_i - \mathbf{A}$.
  - $u_j = \underline{\mathbf{c}}^{\perp} \cdot \underline{\mathbf{b}}_i / \underline{\mathbf{e}}_i^{\perp} \cdot \underline{\mathbf{c}}$
- True intersections occur only for $0 \le u \le 1$
  - Ignore all other intersections
- Sort all intersection points by increasing t.

Clipping for Arbitrary Polygons (3)

- Look at the sorted list of t values
- line is *inside* between the first pair t values, the second pair of t values, etc



More Advanced Clipping

Algorithm	OpenGL Yes/No?	Description
Cohen Sutherland	Yes	Line segments against a rectangle or cube (2D: square)
Cyrus-Beck	Tries; poor results	Line against a convex polygon
Sutherland–Hodgman	No	Any polygon (convex or non-convex) against any convex polygon
Weiler–Atherton	No	Any polygon against any polygon