CMSC427 Geometry and Vectors

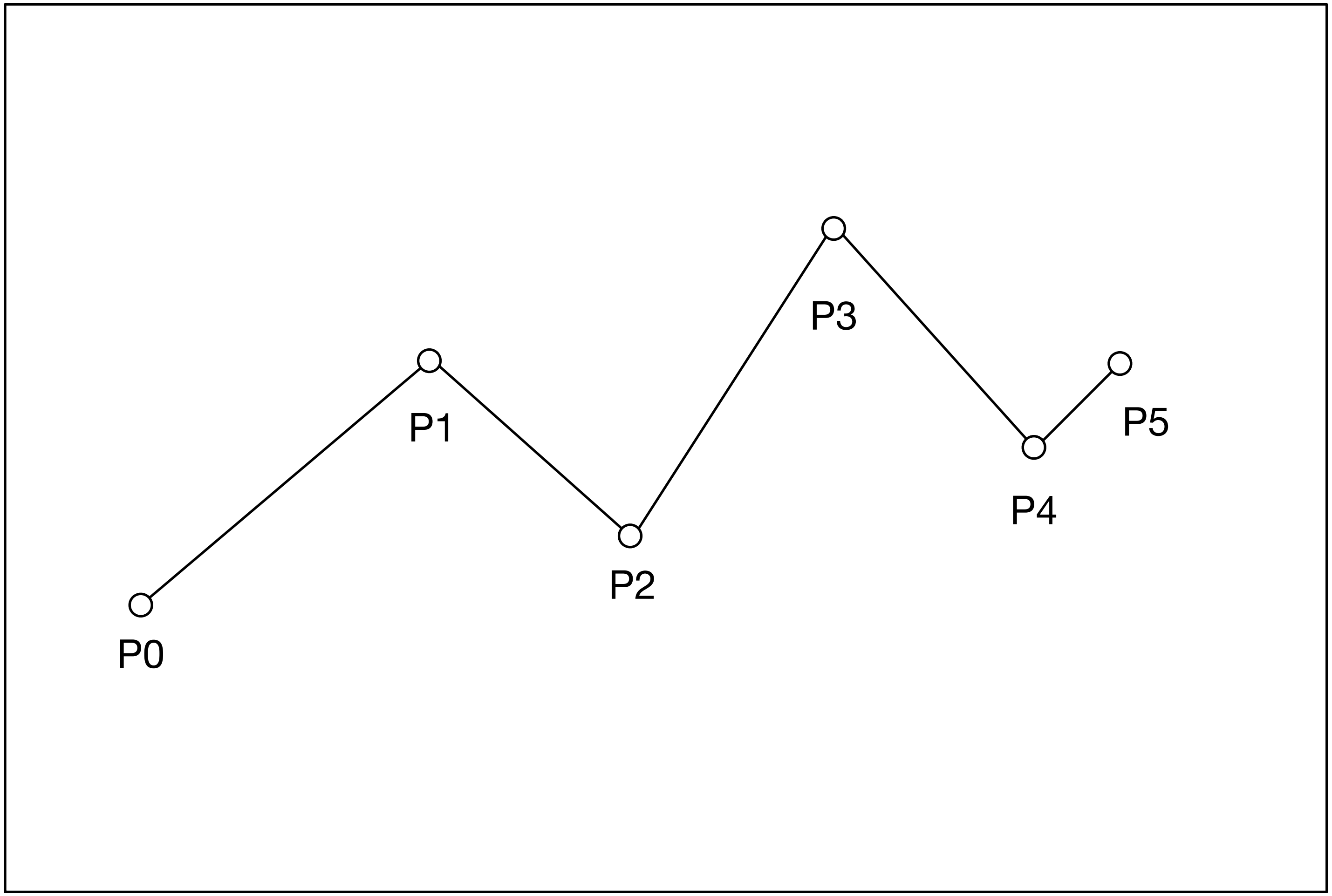
Review: where are we?

- Parametric curves and Hw1?
  - Going beyond the course: generative art
  - https://www.openprocessing.org
  - Brandon Morse, Art Dept, ART370
- Polylines, Processing and Lab0?
  - Questions on Processing?
  - https://processing.org
  - Learning Processing – Dan Shiffman

How can we …

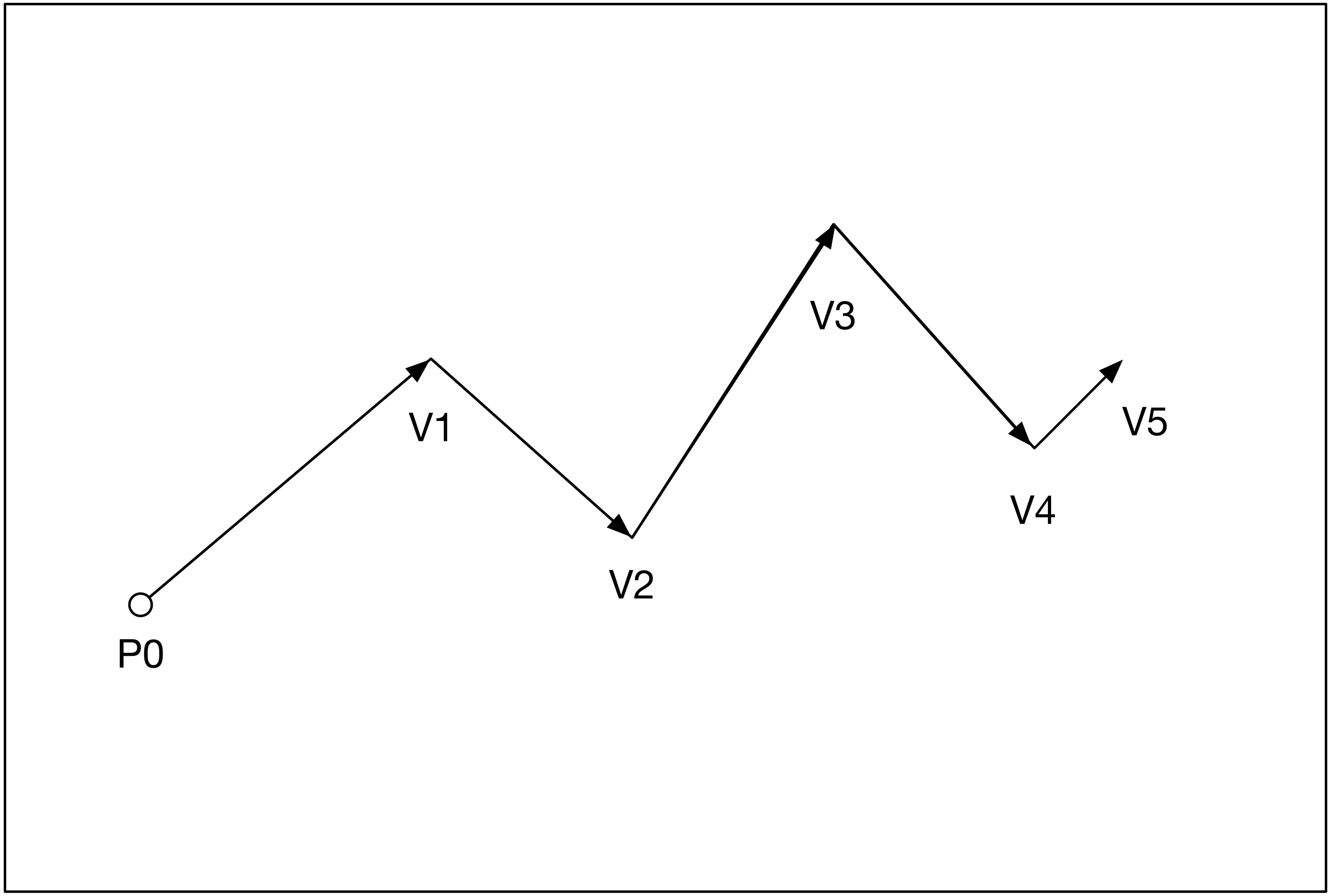
- Find a line perpendicular to another?
- Find symmetric reflecting vector on a mirror?
- Find the distance from a point to a line?
- Figure out if a facet is facing the camera?

Polyline and points



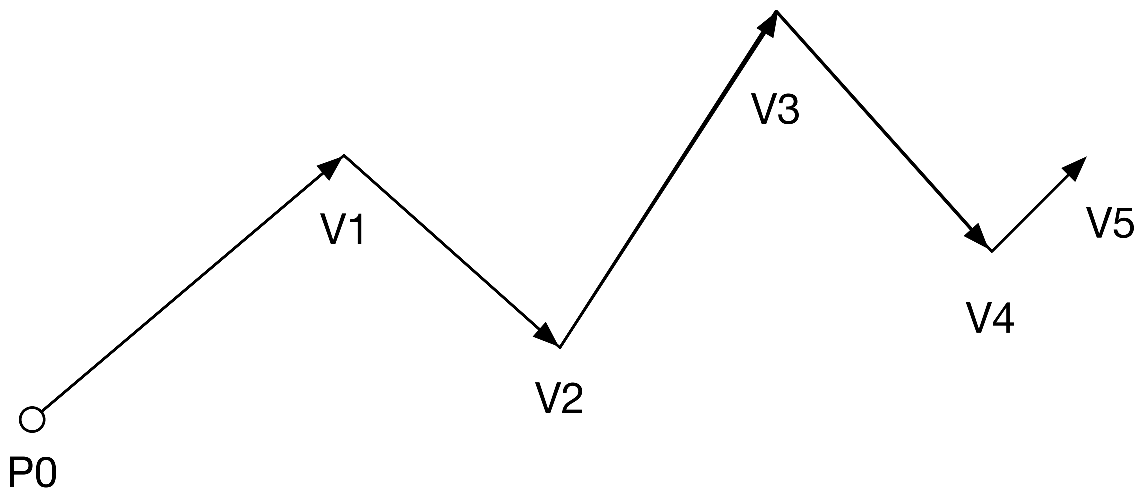
- Polyline as sequence of locations (points)

Polyline and vectors



- Polyline as point plus sequence of vectors
- Assumption: displacements << locations, fewer bits

Polyline as vectors



P1 = P0+V1

P2 = P1+V1 = P0+V1+V2

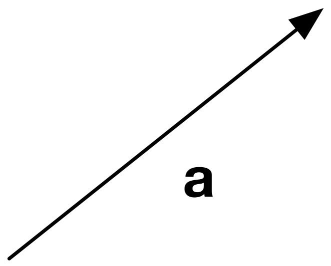
- Polyline as point plus sequence of vectors
- Assumption: displacements << locations, fewer bits

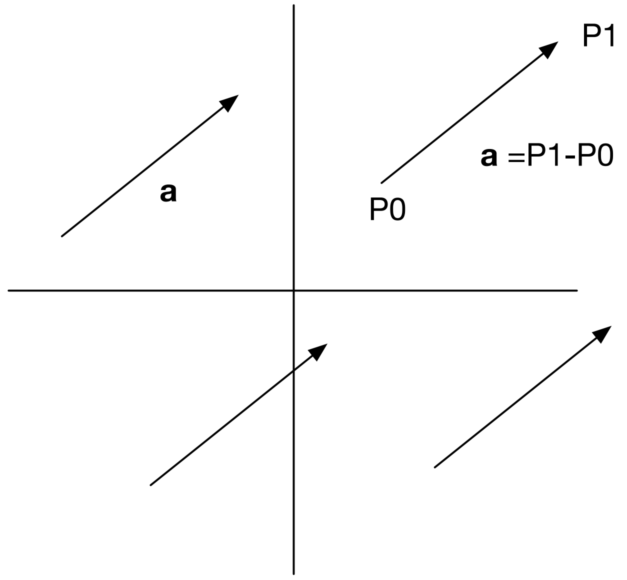
Vector and vector operations

- Objects:
  - Points (x,y) – represent position
  - Vector <x,y> – represent displacement, direction
- Operations:
  - Vector addition, subtraction, scaling
  - Vector magnitude
  - Vector dot product
  - Vector cross product
  - Linear, affine and convex combinations
- Applications:
  - Representations: using vectors for lines, planes, others
  - Metrics: angle between lines, distances between objects
  - Tests: are two lines perpendicular, is a facet visible?

Vectors

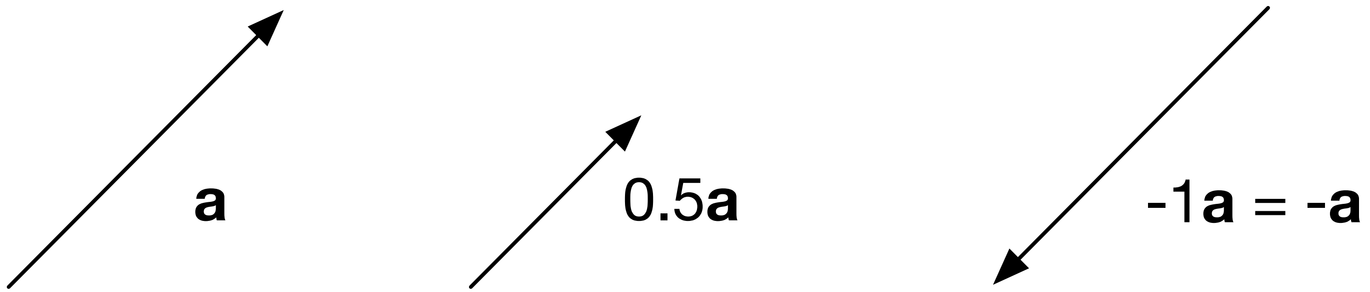
- Direction and distance
- Describes
  - Difference between points
  - Speed, translation, axes
- Notation
  - In bold **a**
  - Angle brackets **a** = <x,y>
- Free vectors
  - No anchor point
  - Displacement, not location



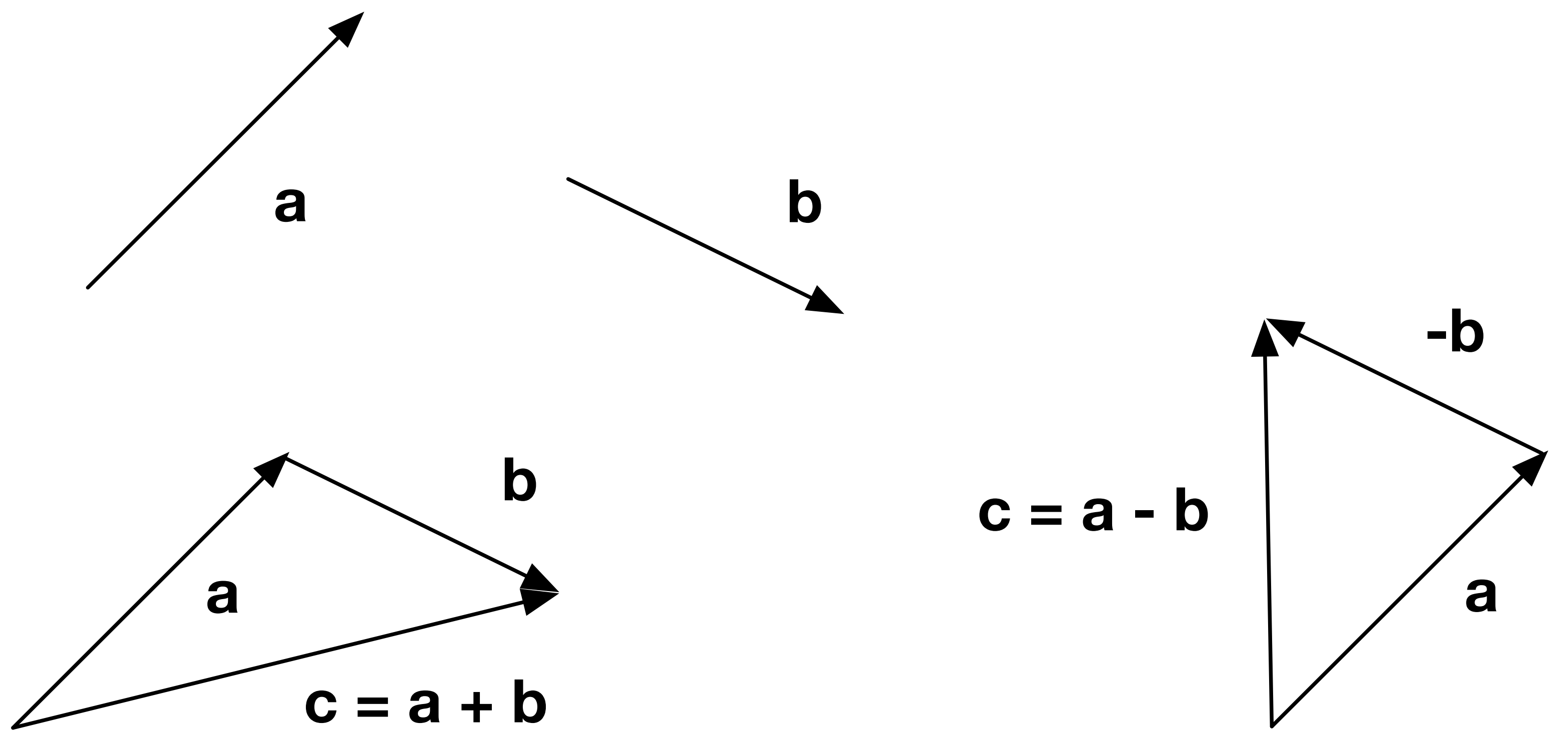


Vector scaling

Multiplication by scalar $s\mathbf{a}$

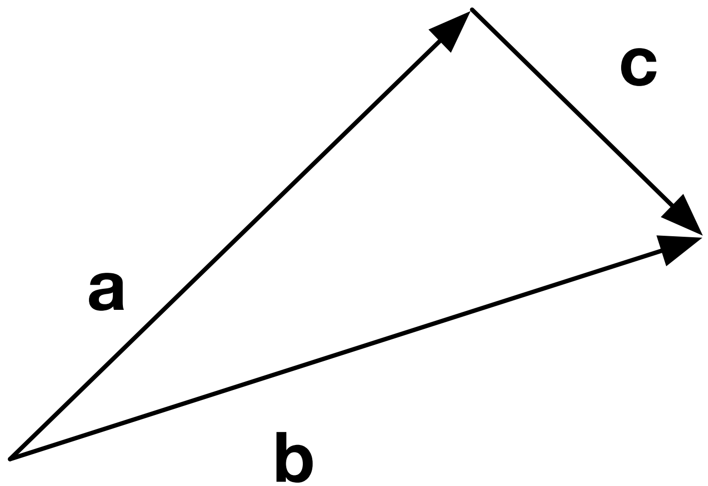


Vector addition and subtraction



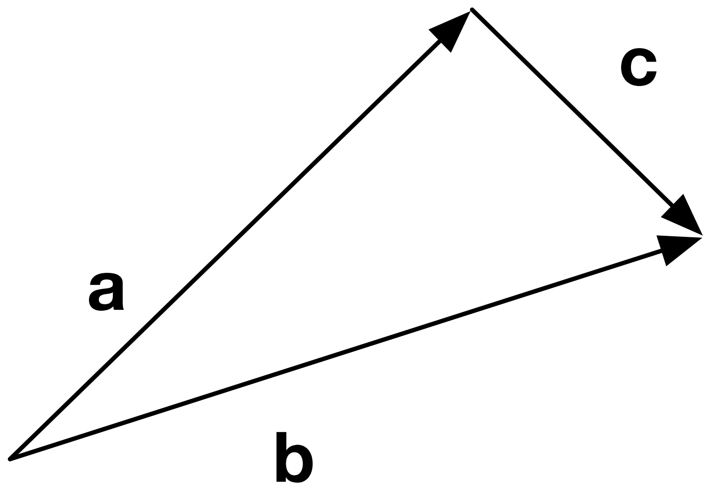
Vector addition and subtraction

What is **c** in terms of **a** and **b**?



Vector addition and subtraction

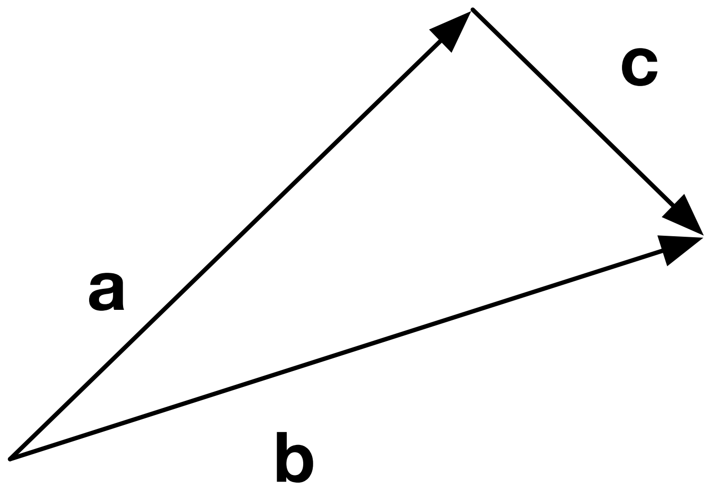
What is **c** in terms of **a** and **b**?



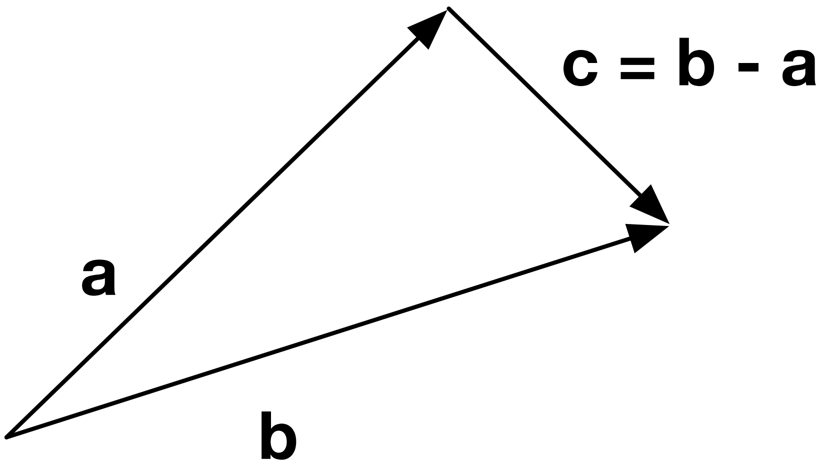
a + c = b

Vector addition and subtraction

What is **c** in terms of **a** and **b**?



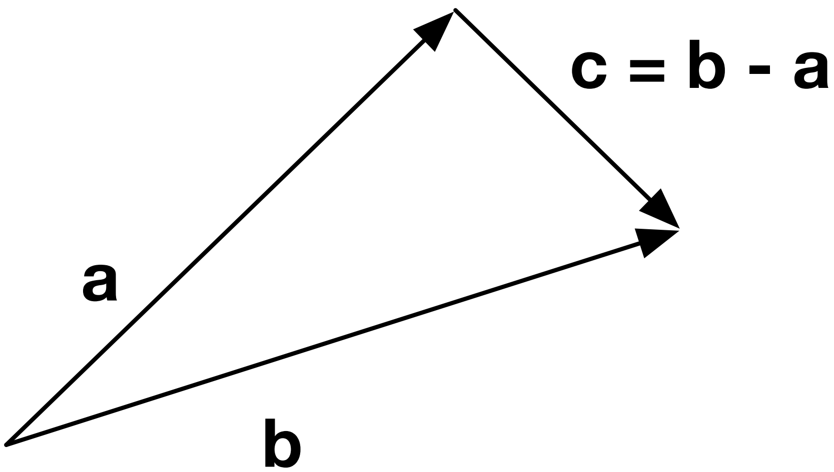
a + c = b



Coordinate vs. coordinate-free representation

Coordinate-free equation
Valid for 2D and 3D

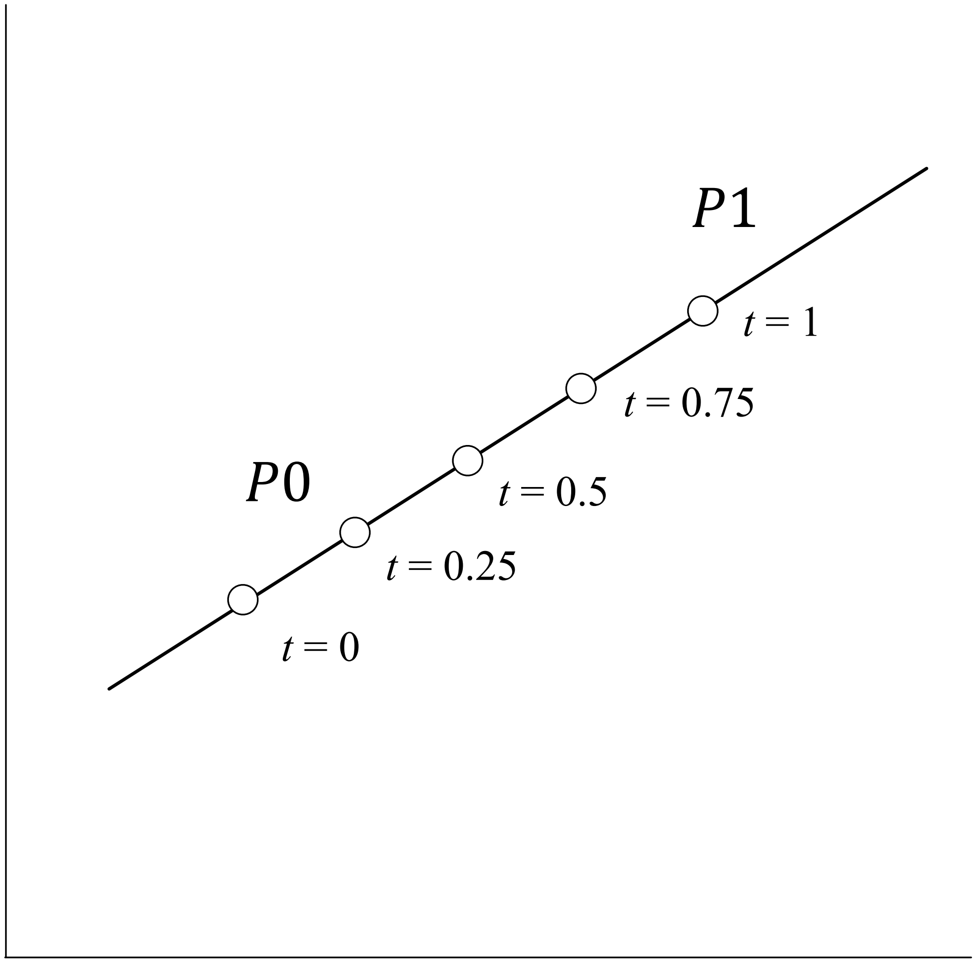
Prefer when possible



Coordinate equation
a = <3,3>
b = <4,2>

c = b - a = <4,2> - <3,3> = <1,-1>

Parametric line in coordinate-free vector format



$$x = t\,dx + px$$
$$y = t\,dy + py$$

Set

$$\boldsymbol{V} = P1 - P0$$
$$= \langle dx, dy \rangle$$

Then

$$P(t) = t\,\boldsymbol{V} + P0$$

Good in 2D and 3D

Vector magnitude

The magnitude (length) of a vector is

$$|\mathbf{v}|^2 = {v_x}^2 + {v_y}^2$$

$$|\mathbf{v}| = \sqrt{{v_x}^2 + {v_y}^2}$$

A vector of length 1.0 is called a *unit vector*

To *normalize* a vector is to rescale it to unit length

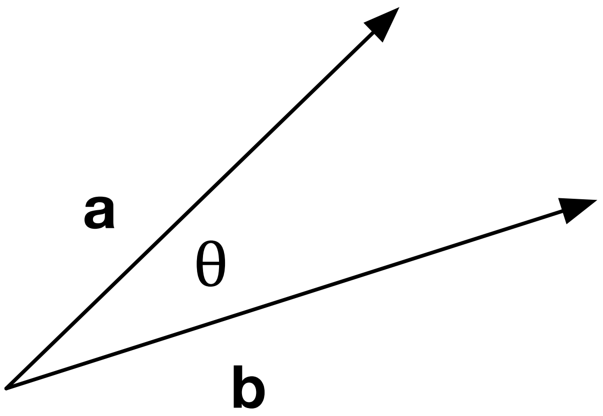
$$\mathrm{n}_v = \frac{\mathbf{v}}{|\mathbf{v}|}$$

Normal vectors represent direction and are used for:
light direction, surface normals, rotation axes

Dot product

- Inner product between two vectors
- Defined using coordinate-free cosine rule

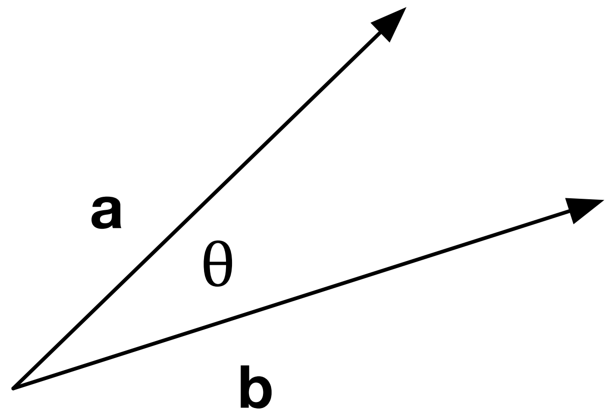
$$\mathbf{a} \bullet \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos(\theta)$$



Dot product

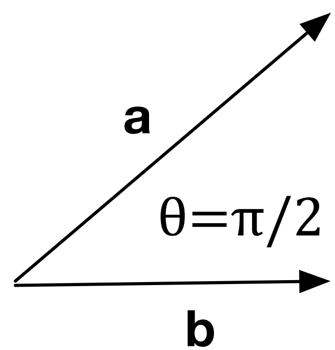
- Inner product between two vectors
- Defined using coordinate-free cosine rule

$$\mathbf{a} \bullet \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos(\theta)$$



- Example:

$$\mathbf{a} \;=< 1,1 >$$
$$\mathbf{b} \;=< 1,0 >$$
$$\theta = \pi/2$$
$$\mathbf{a} \bullet \mathbf{b} = 1.4142 * 1 * 0.70711 = 1$$



Dot product – coordinate version

- Given

$$\mathbf{a} = \langle \boldsymbol{a_x}, \boldsymbol{a_y} \rangle \qquad \mathbf{b} = \langle \boldsymbol{b_x}, \boldsymbol{b_y} \rangle$$

- Then

$$\mathbf{a} \bullet \mathbf{b} = \boldsymbol{a_x b_x} + \boldsymbol{a_y b_y}$$

- In n dimensions

$$\mathbf{a} \bullet \mathbf{b} = \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{a_i b_i}$$

Dot product – coordinate version

- Given

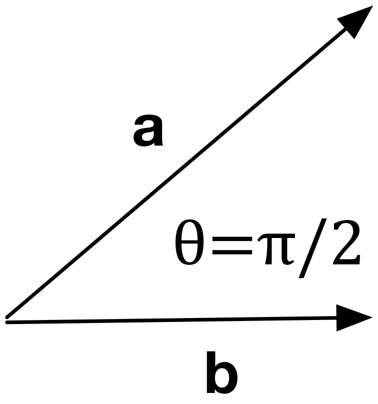
$$\mathbf{a} = \langle \boldsymbol{a_x}, \boldsymbol{a_y} \rangle \qquad \mathbf{b} = \langle \boldsymbol{b_x}, \boldsymbol{b_y} \rangle$$

- Then

$$\mathbf{a} \bullet \mathbf{b} = \boldsymbol{a_x b_x} + \boldsymbol{a_y b_y}$$

- Example revised:

$$\begin{aligned} \mathbf{a} &=< 1,1 > \\ \mathbf{b} &=< 1,0 > \\ \mathbf{a} \bullet \mathbf{b} &= 1 * 1 + 1 * 0 = 1 \end{aligned}$$



Dot product: computing angle (and cosine)

- Given

$$\mathbf{a} \bullet \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos(\theta)$$

- We have

$$\cos(\theta) = \frac{\mathbf{a} \bullet \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

$$\theta = \cos^{-1}\left(\frac{\mathbf{a} \bullet \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right)$$

- Example

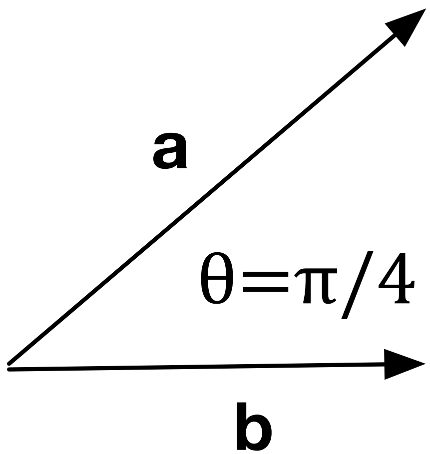
$$\mathbf{a} \; =< 1,1 >, \mathbf{b} \; =< 1,0 >$$

$$\mathbf{a} \bullet \mathbf{b} = 1 * 1 + 1 * 0 = 1$$

$$|\mathbf{a}| = \mathbf{1.4142}, |\mathbf{b}| = \mathbf{1.0}$$

$$\frac{\mathbf{a} \bullet \mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = \frac{\mathbf{1}}{\mathbf{1.4142}} = \mathbf{0.70711}$$

$$\cos^{-1}\left(\frac{\mathbf{a} \bullet \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right) = \mathbf{0.78540} = \boldsymbol{\pi}/4$$



Dot product: testing perpendicularity

- Since

$$\cos(90°) = \cos\left(\frac{\pi}{2}\right) = \mathbf{0}$$

- Then a perpendicular to b gives

$$\mathbf{a} \perp \mathbf{b} \Rightarrow \mathbf{a} \bullet \mathbf{b} = 0$$

- Examples

$$\mathbf{a} =< 1,0 >, \mathbf{b} =< 0,1 >$$
$$\mathbf{a} \bullet \mathbf{b} = 1 * 0 + 1 * 0 = 0$$

$$\mathbf{a} =< 1,1 >, \mathbf{b} =< -1,1 >$$
$$\mathbf{a} \bullet \mathbf{b} = 1 * -1 + 1 * 1 = 0$$

Perp vector and lines

- Given vector

$$\mathbf{v} \; =< v_x, v_y >$$

- The perp vector is

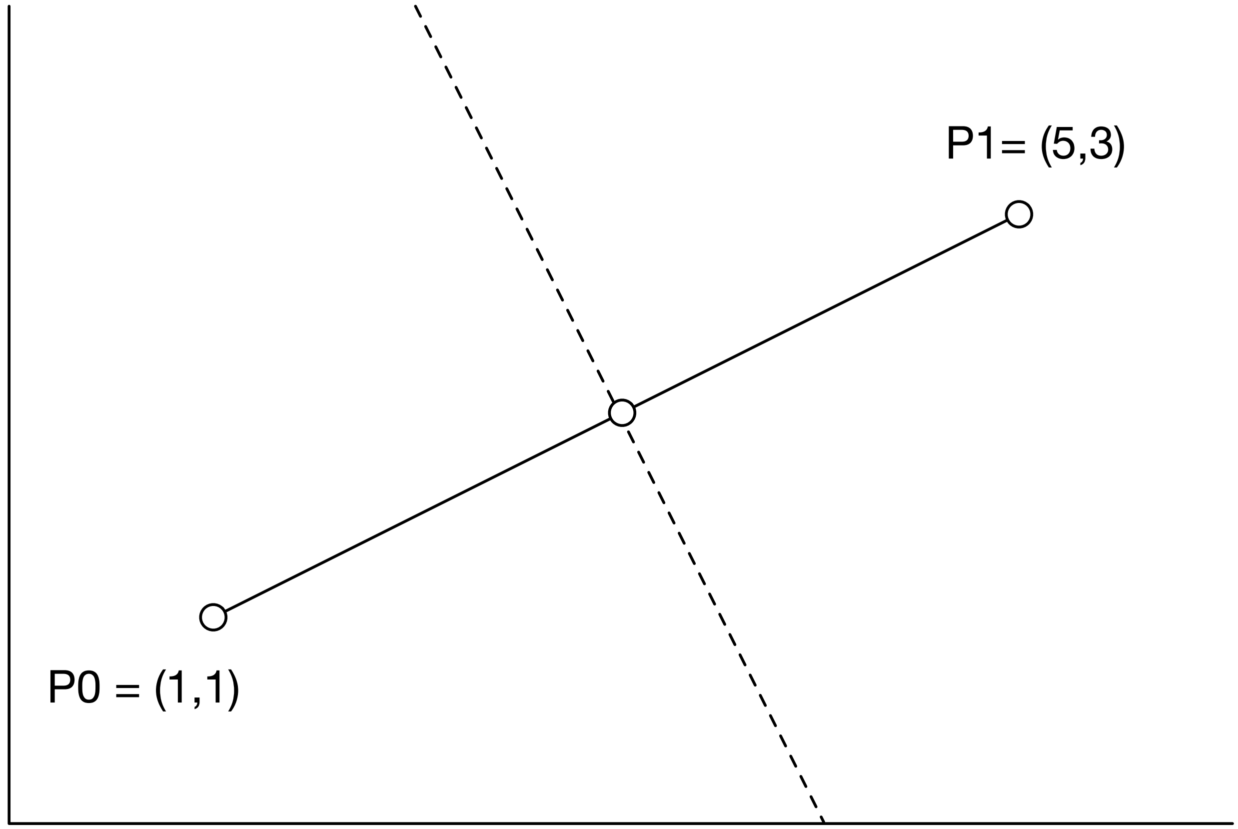
$$=< -v_y, v_x >$$

- So

$$\mathbf{v} \bullet \mathbf{v}^{\perp} = 0$$

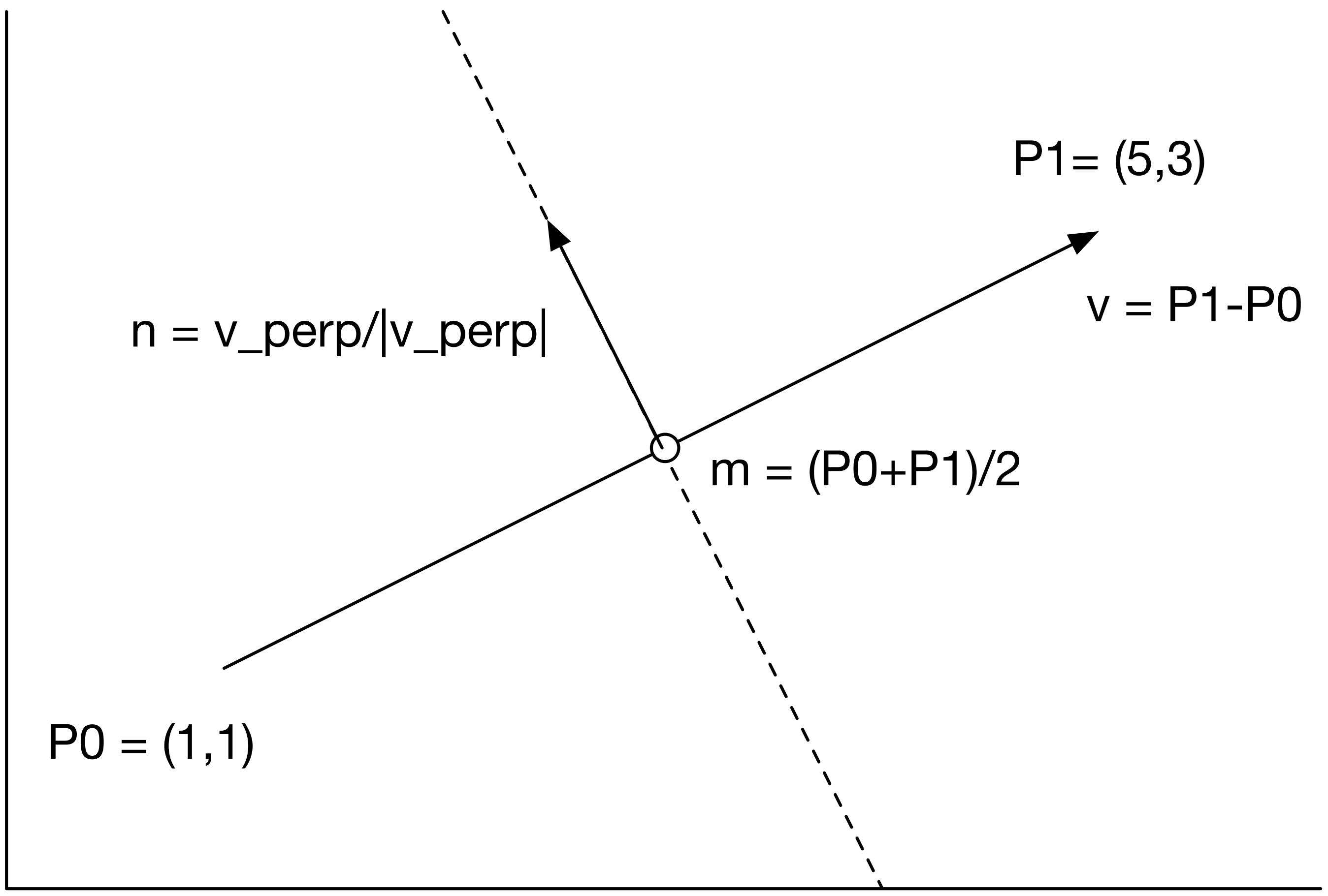
Midpoint bisector

- Given line segment P0 to P1, what line is perpendicular through the midpoint?



Midpoint bisector

- Given line segment P0 to P1, what line is perpendicular through the midpoint?

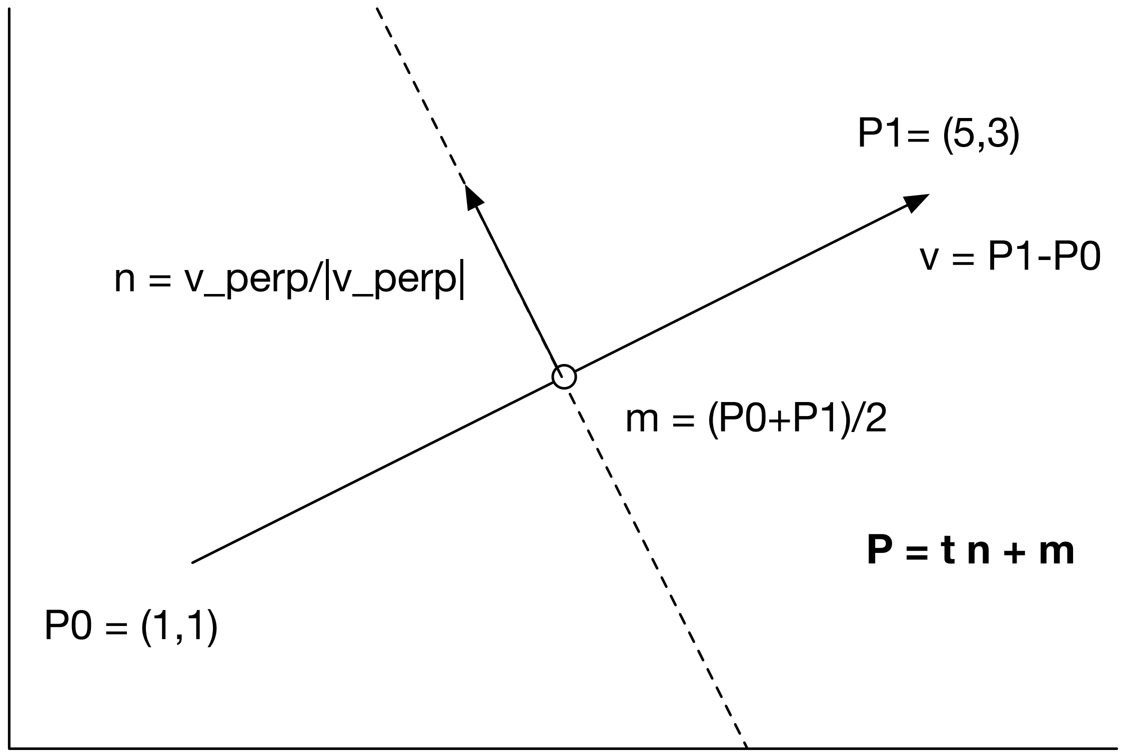


Midpoint bisector

- Result: bisector is

$$P = t\mathbf{n} + m$$

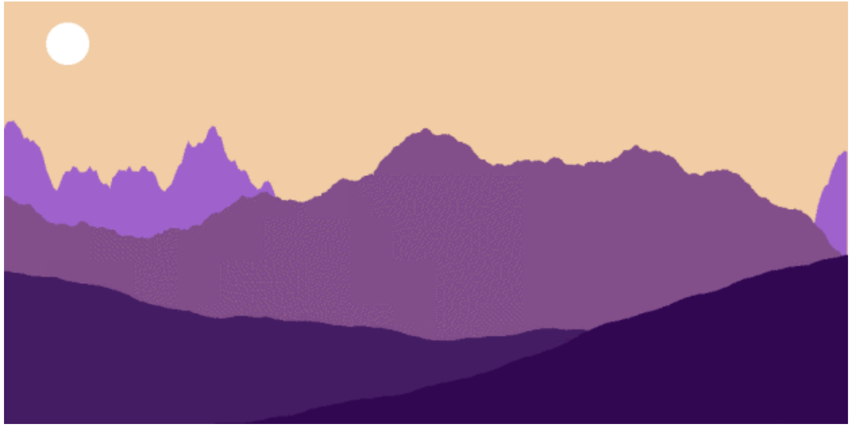
- With n normalized perp vector, m midpoint
- Appropriate range of t?

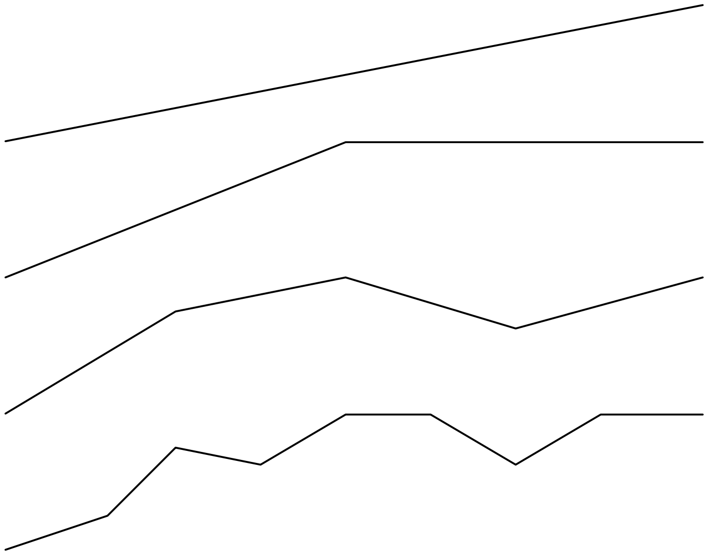


Midpoint displacement terrain (2D recursive algorithm)

- Given two points P0,P1
  - Compute midpoint m
  - Compute bisector line P = tn + m
  - Pick random t, generate P’
  - Call recursively on segments P0-P’, P’-P1

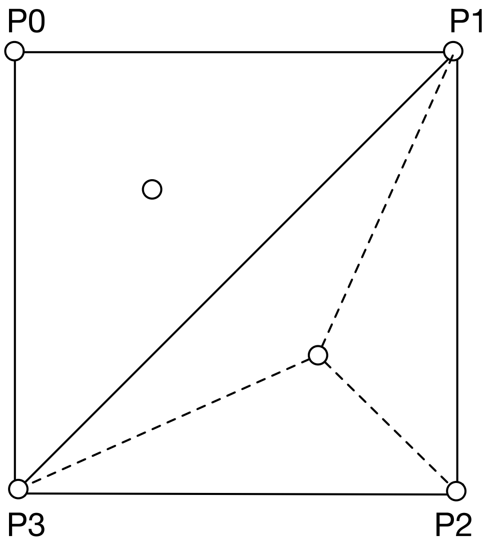
- Tuning needed on magnitude of displacement

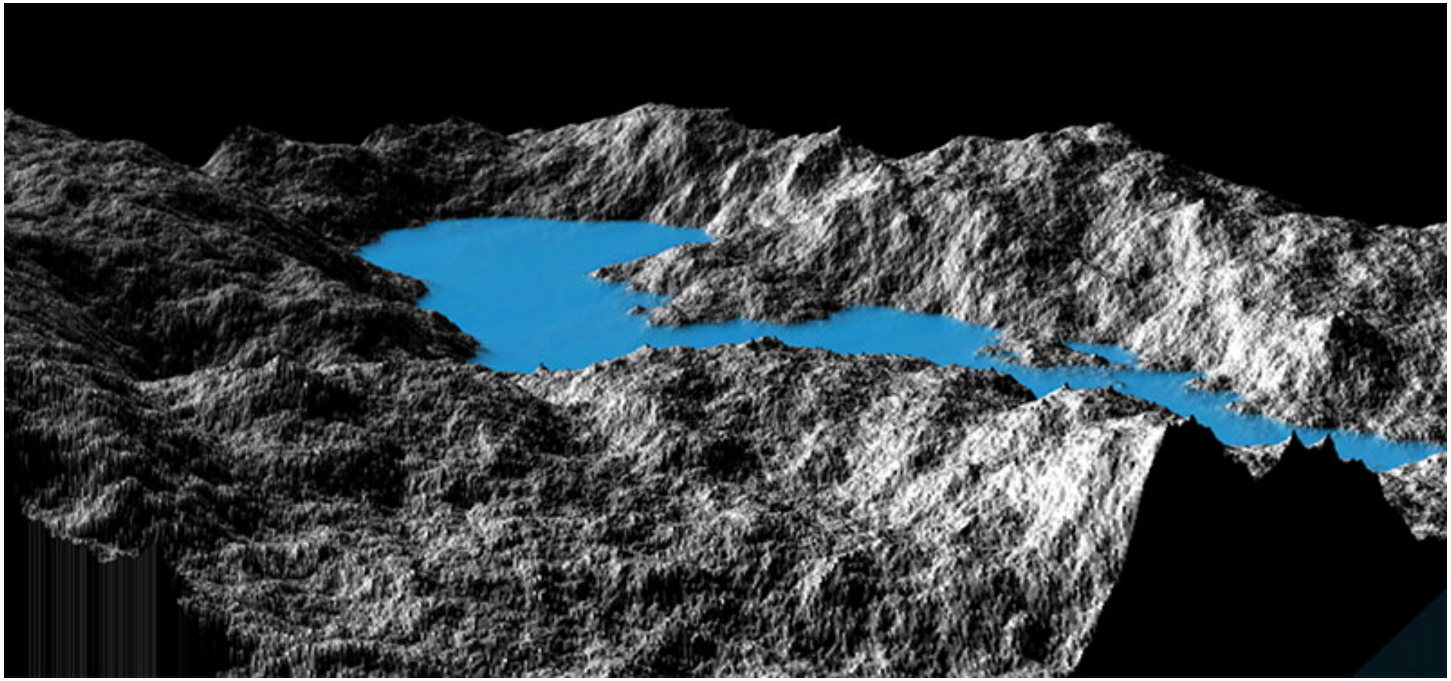




Midpoint displacement terrain (3D recursive algorithm)

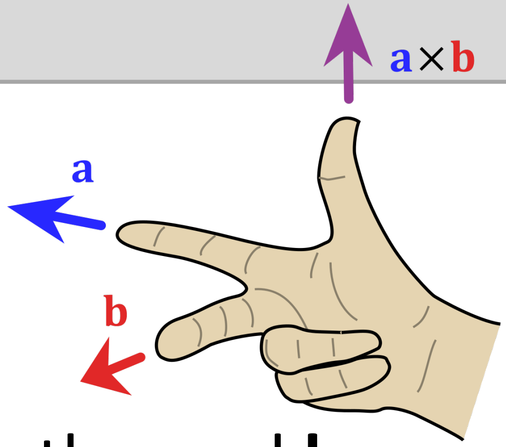
- Start with four points P0,P1,P2,P3
- Divide (quads or triangles?)
- Compute midpoint bisector (how?)
- Displace, repeat





Hunter Loftis

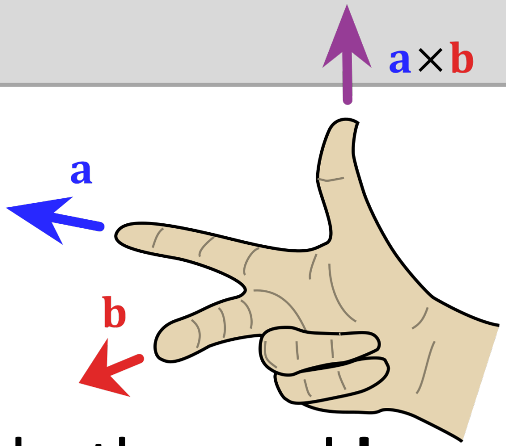
Cross product



$\mathbf{a} \times \mathbf{b}$

- Read as “a cross b”
- **a x b** is a vector perpendicular to both **a** and **b**, in the direction defined by the right hand rule

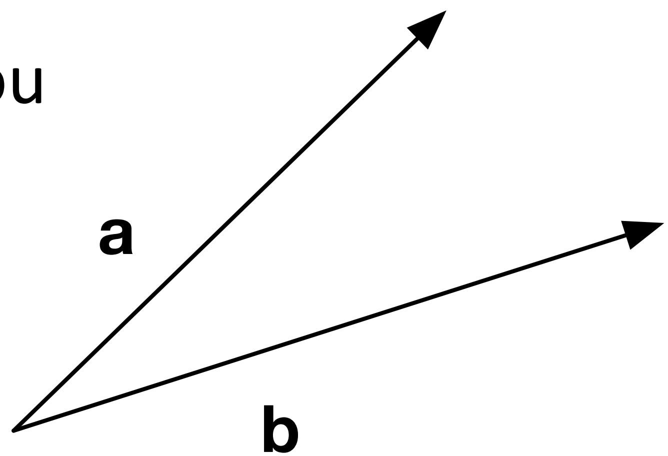
Cross product



a × **b**

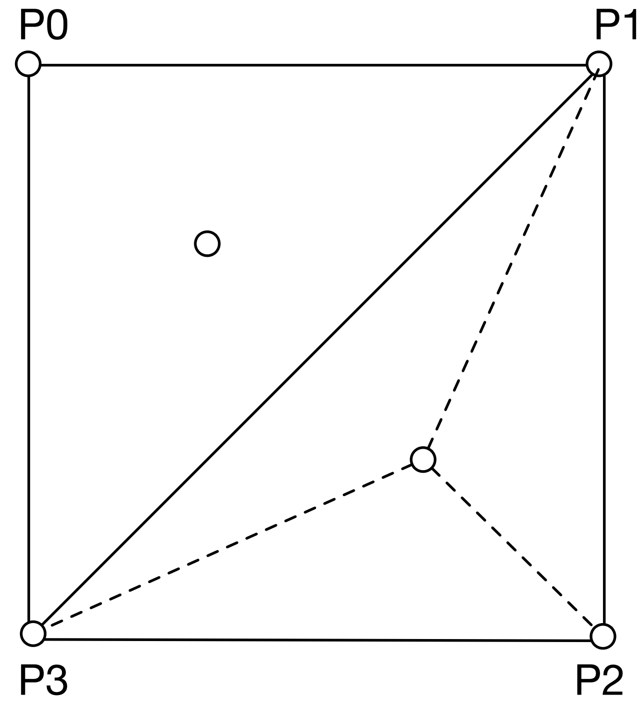
- Read as "a cross b"
- **a x b** is a vector perpendicular to both **a** and **b**, in the direction defined by the right hand rule
- Vector a and b lie in the plane of the projection screen.
- Does **a x b** point towards you or away from you?

  What about **b x a**?



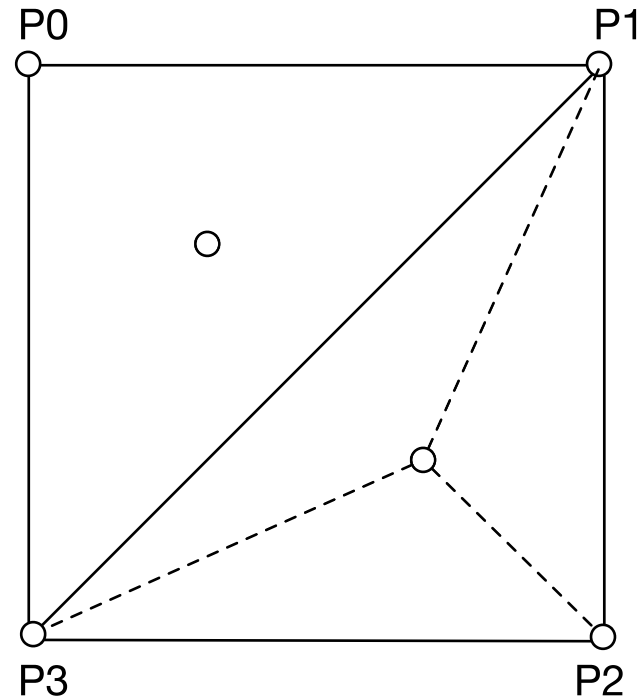
Cross product: normal to plane

- How compute normal vector to triangle P0, P1, P3?
- Assume up is out of page



Cross product: normal to plane

- How compute normal vector to triangle P0, P1, P3?
- Assume up is out of page
- One answer:
  n = (P1-P3) x (P0-P1)



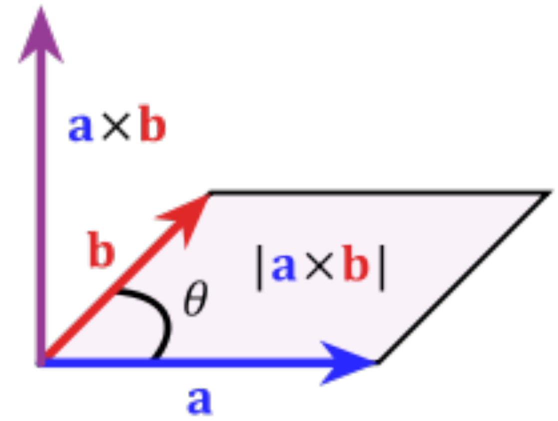
Cross product: length of a x b

- Parallelogram rule

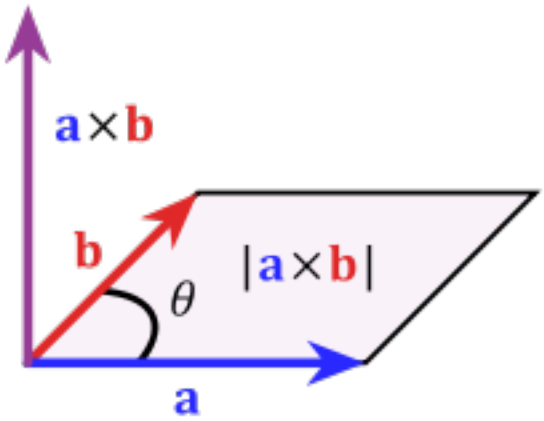
$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}||\mathbf{b}|\sin(\theta)$$

- Area of mesh triangle?
- When would cross product be zero?

$$|\mathbf{a} \times \mathbf{b}| = 0$$



Cross product: length of a x b



- Parallelogram rule

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}||\mathbf{b}|\sin(\theta)$$

- Area of mesh triangle?
- When would cross product be zero?

$$|\mathbf{a} \times \mathbf{b}| = 0$$

- Either a,b parallel, or either degenerate

Cross product: computing, vector approach

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

Cross product: computing, matrix approach

- Determinant of

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

- Computed with 3 lower minors

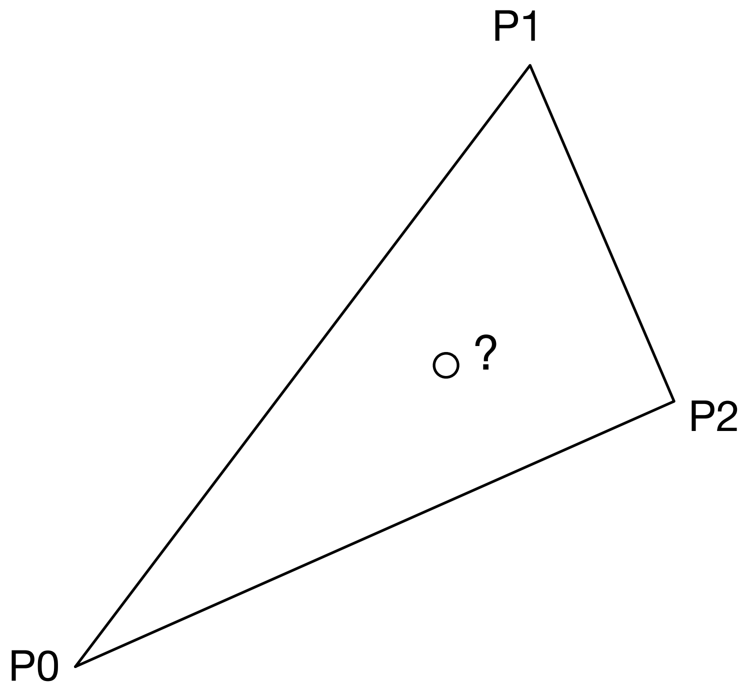
$$\mathbf{a} \times \mathbf{b} = i \begin{bmatrix} a_y & a_z \\ b_y & b_z \end{bmatrix} - j \begin{bmatrix} a_x & a_z \\ b_x & b_z \end{bmatrix} + k \begin{bmatrix} a_x & a_y \\ b_x & b_y \end{bmatrix}$$

- with

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - cb$$

- i, j, k are unit vectors in directions of x, y and z axes
- i=<1,0,0>   j=<0,1,0>   k=<0,0,1>

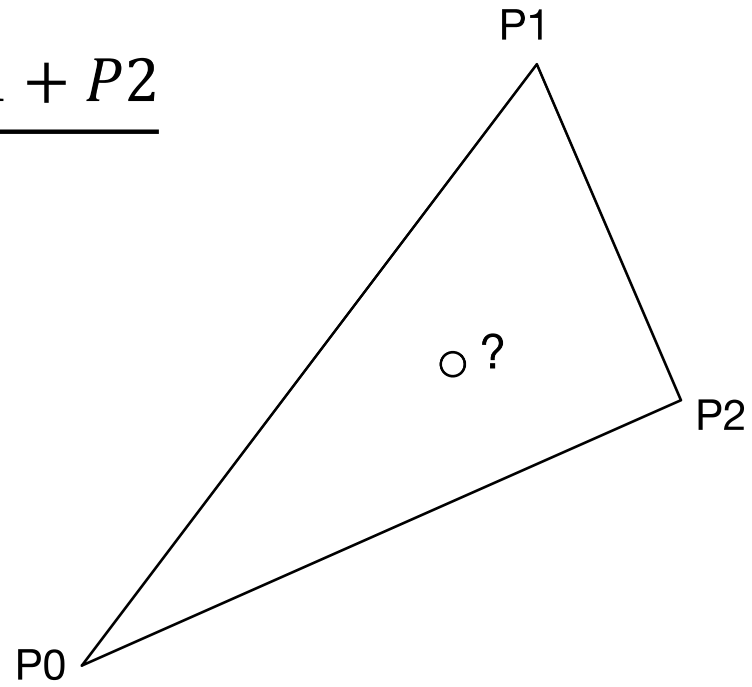
Midpoint of triangle?



Midpoint of triangle?

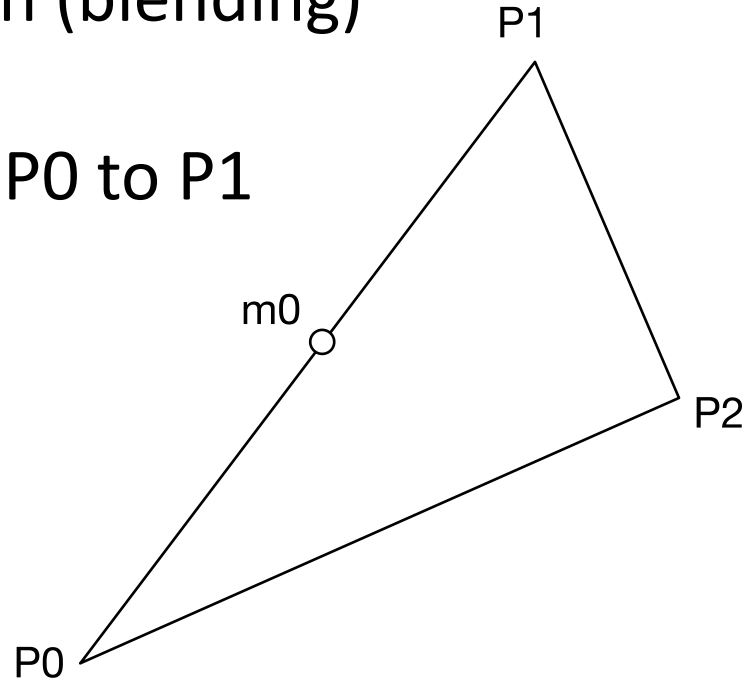
Answer 1:

$$m = \frac{P0 + P1 + P2}{3}$$



Midpoint of triangle?

Answer 2: Double interpolation (blending)



First interpolation: vector line P0 to P1

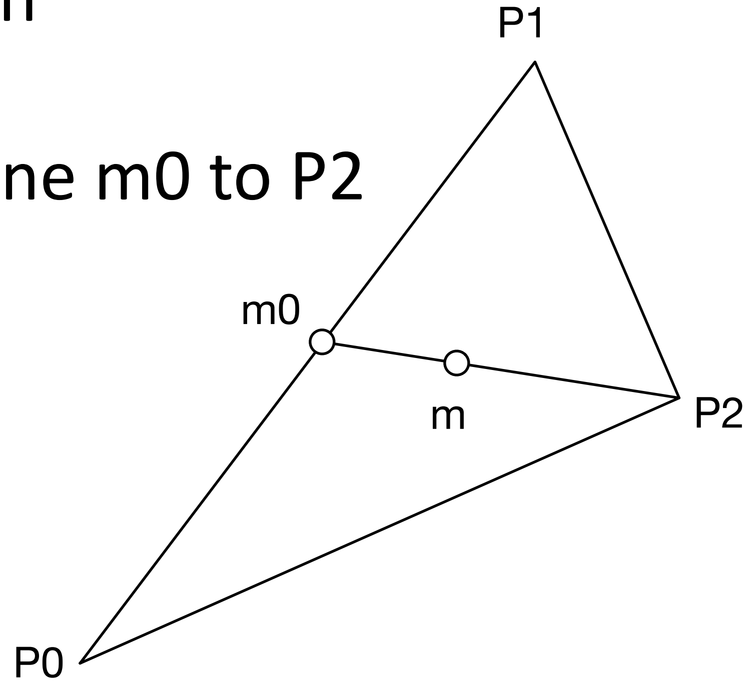
$$P = tV + P0$$
$$= t(P1 - P0) + P0$$
$$= tP1 + (1 - t)P0$$

Midpoint at t = 0.5

$$m0 = 0.5P1 + (1 - 0.5)P0$$
$$= 0.5P1 + 0.5P0$$
$$= \frac{P1 + P0}{2}$$

Midpoint of triangle?

Answer 2: Double interpolation



Second interpolation: vector line m0 to P2

$$
\begin{aligned}
P &= sV' + m0 \\
&= s(P2 - m0) + m0 \\
&= sP2 + (1 - s)m0
\end{aligned}
$$

Midpoint at s = 1/3

$$
\begin{aligned}
m0 &= \frac{1}{3}P2 + \left(1 - \frac{1}{3}\right)m0 \\
&= \frac{1}{3}P2 + \frac{2}{3}m0
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{3}P2 + \frac{2}{3}(0.5P1 + 0.5P0) \\
&= \frac{P2 + P1 + P0}{3}
\end{aligned}
$$

Generalizing: convex combinations of points

A convex combination of a set of points S is a linear combination such that the non-negative coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \le \alpha_i \le 1$$

Generalizing: convex combinations of points

A convex combination of a set of points S is a linear combination such that the non-negative coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq 1$$

Is this equation for a line *segment* a convex combination?

$$P = tP1 + (1 - t)P0$$

Generalizing: convex combinations of points

A convex combination of a set of points S is a linear combination such that the non-negative coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq 1$$

Is this equation for a line *segment* a convex combination?

$$P = tP1 + (1 - t)P0$$

Yes. With t in [0,1], t and (1-t) >= 0, and t+(1-t) = 1

Generalizing: convex combinations of points

A convex combination of a set of points S is a linear combination such that the non-negative coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \le \alpha_i \le 1$$

Is this equation for a triangle a convex combination, assuming s and t are in [0,1]?

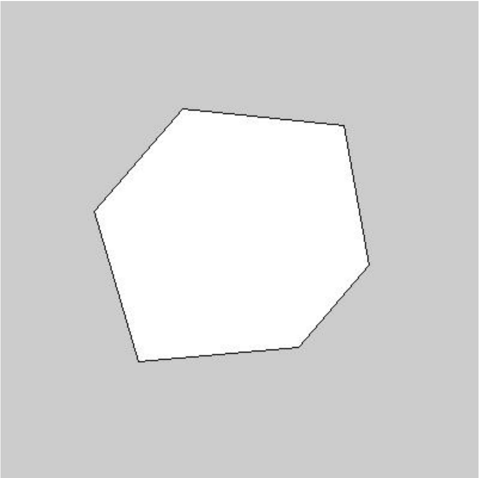
$$P = sP2 + (1-s)(tP1 + (1-t)P0)$$
$$= sP2 + (1-s)(tP1) + (1-s)(1-t)P0)$$
$$= sP2 + (t-st)P1 + (1-s-t+st)\ P0$$

Generalizing: convex combinations of points

A convex combination of a set of points S is a linear combination such that the non-negative coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq 1$$

For general polygon, all convex combinations of vertices yields *convex hull*



Linear, affine and convex combinations

Linear: No constraints on coefficients

$$C = \sum_{P\ in\ S} \alpha_i P_i$$

Affine: Coefficients sum to 1

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1$$

Convex: Coefficients sum to 1, each in [0,1]

$$C = \sum_{P\ in\ S} \alpha_i P_i \quad \text{with} \quad \sum \alpha_i = 1 \quad \text{and} \quad 0 \le \alpha_i \le 1$$

Linear combinations of points vs. vectors

Point – point yields a ….

Vector – vector yields a …

Point + vector yields a …

Point + point yields a …

Linear combinations of points vs. vectors

Point – point yields a …. vector

Vector – vector yields a … vector

Point + vector yields a … point

Point + point yields a … ???? Not defined

Vectors are closed under addition and subtraction
- Any linear combination valid

Points are not
- Affine combination that sums to 0 yields vector
- Affine combination that sums to 1 yields point
- Convex combination yields point in convex hull

Moral: When programming w/ pts&vtrs, know the output type

Applying vectors operations to polygons

Is a polygon winding clockwise?

Is a polygon convex?

Is a polygon simple?

Lab 1: add these methods to Polyline class

If time allows …

1. Normal form of line
2. Normal form of plane
3. Programming with points and vectors
4. Sign of dot product
5. Tweening
6. Distance from point to line

What you should know after today

1. Notation for vectors <x,y> and pts (x,y)
2. Vector math: scaling, addition, subtraction
3. Coordinate and coordinate-free formulas
4. Vector magnitude and normalization
5. Dot products and cosine rule
6. Using Octave Online as vector calculator
7. Dot product, perpendicularity and perp vector
8. Midpoint bisector
9. Midpoint displacement algorithm
10. Cross product, right hand rule and sine rule
11. Computing cross product with determinant
12. Linear, affine and convex combinations
13. Triangle midpoint