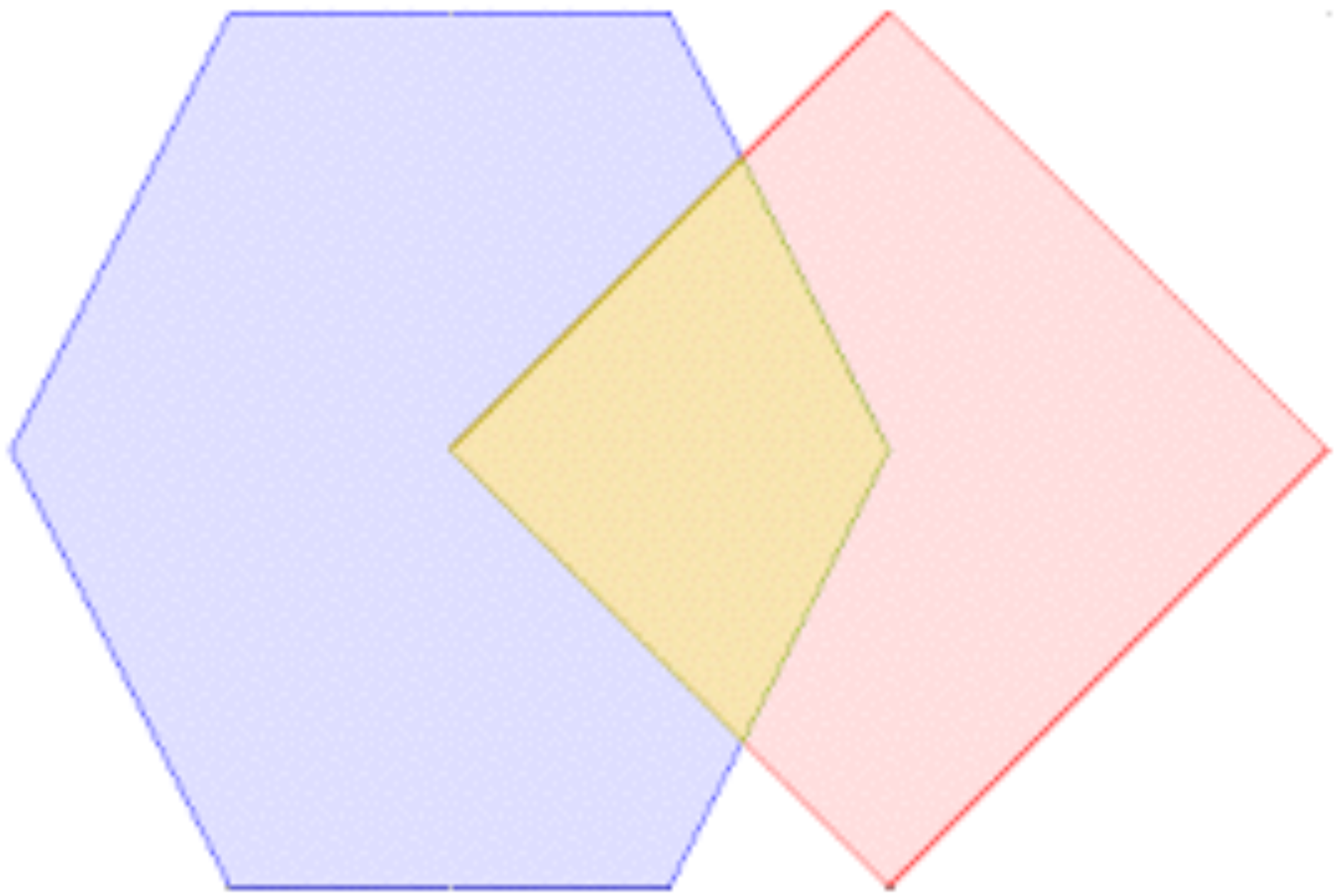
Geometric Tests

CS 355: Introduction to Graphics and Image Processing

Geometric Tests

- Lots of things in graphics involve making geometric tests:
  - Interactive selection
  - Intersections
  - Collisions (e.g., games)
  - Ray intersections
  - Nearest points
  - …



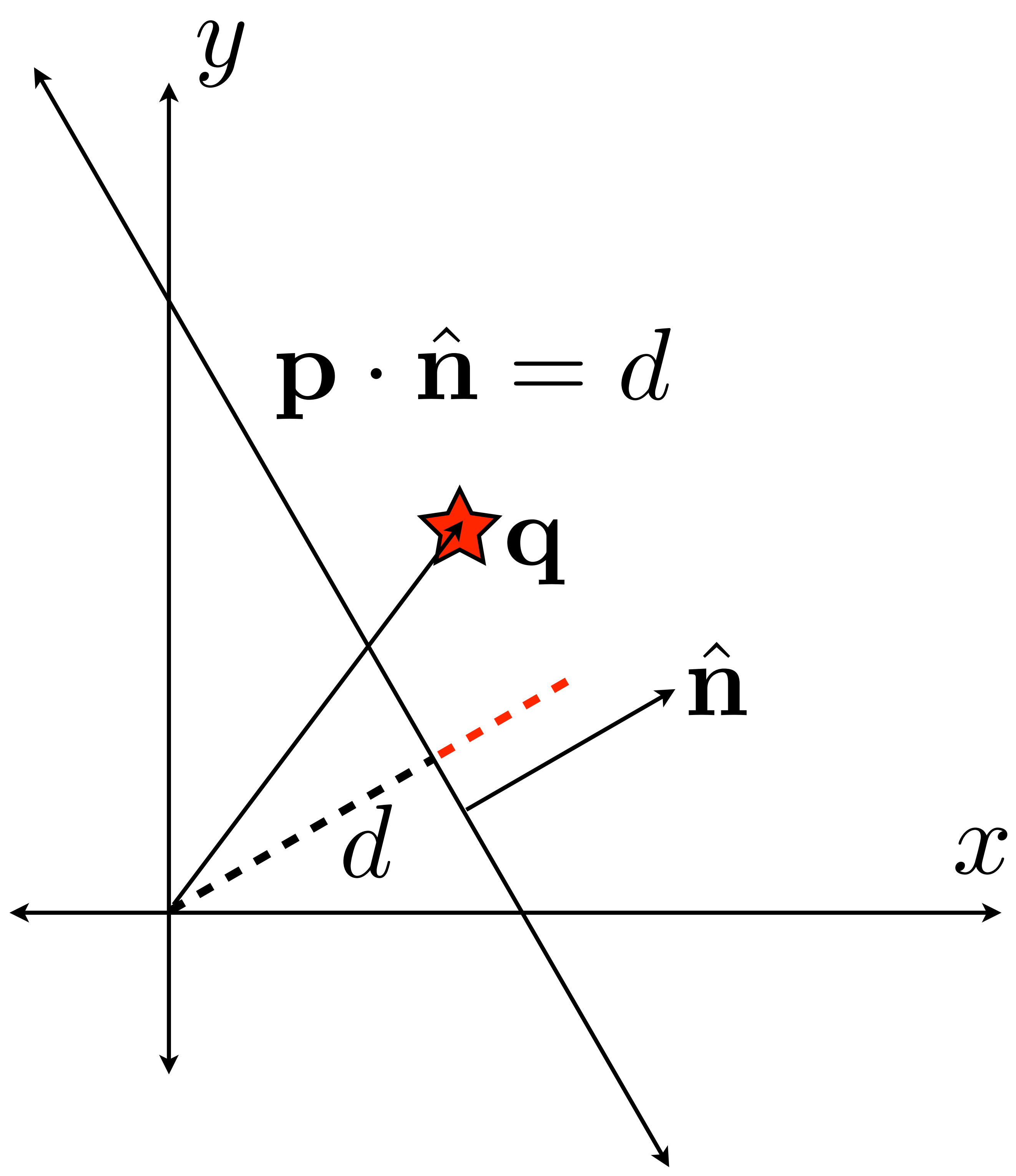
Distance to a Line

- Points **p** on the line **L** satisfy this constraint:

$$\mathbf{p} \cdot \hat{\mathbf{n}} - d = 0$$

- Distance from point **q** to the line **L**:

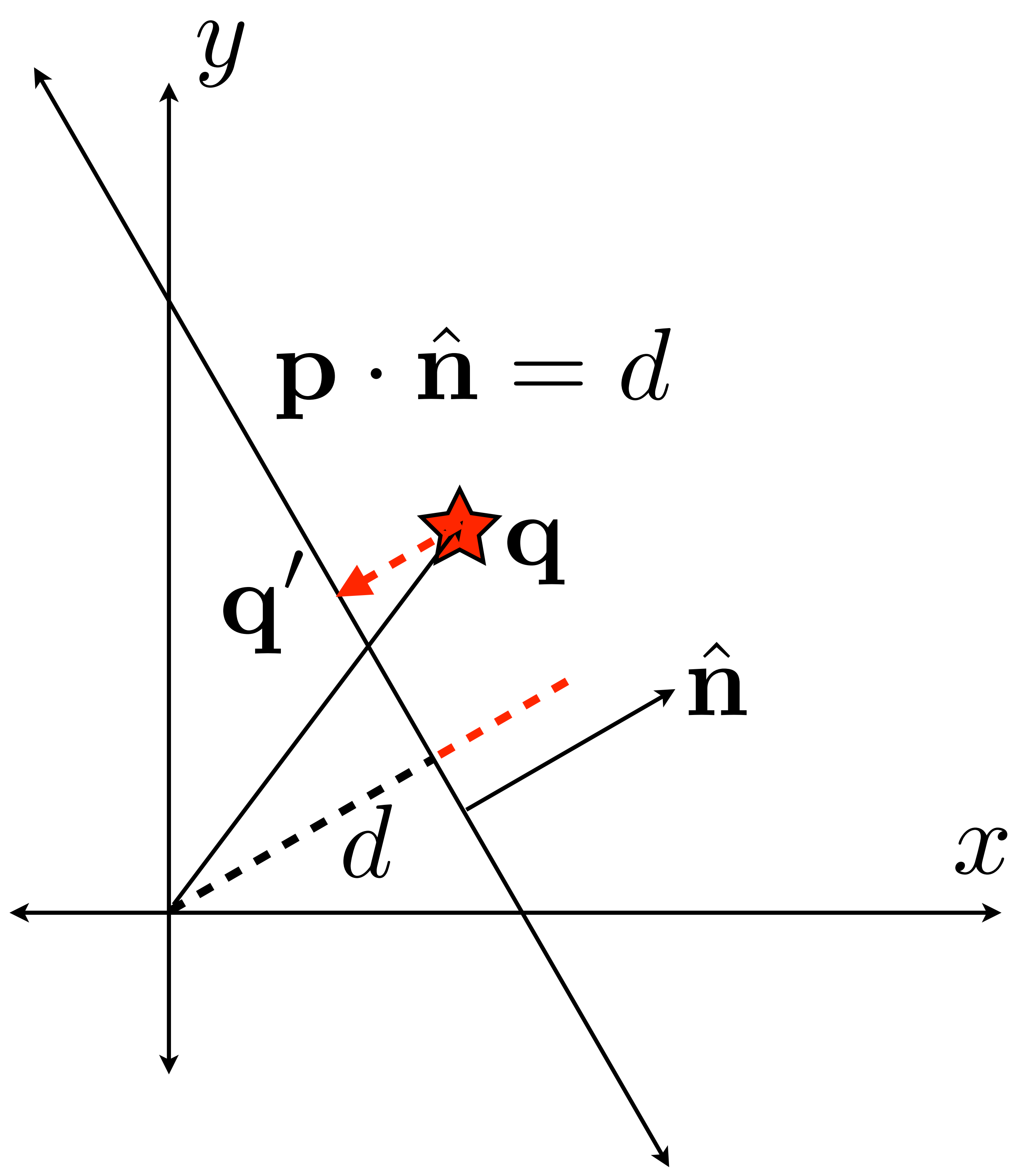
$$|\mathbf{q} \cdot \hat{\mathbf{n}} - d|$$



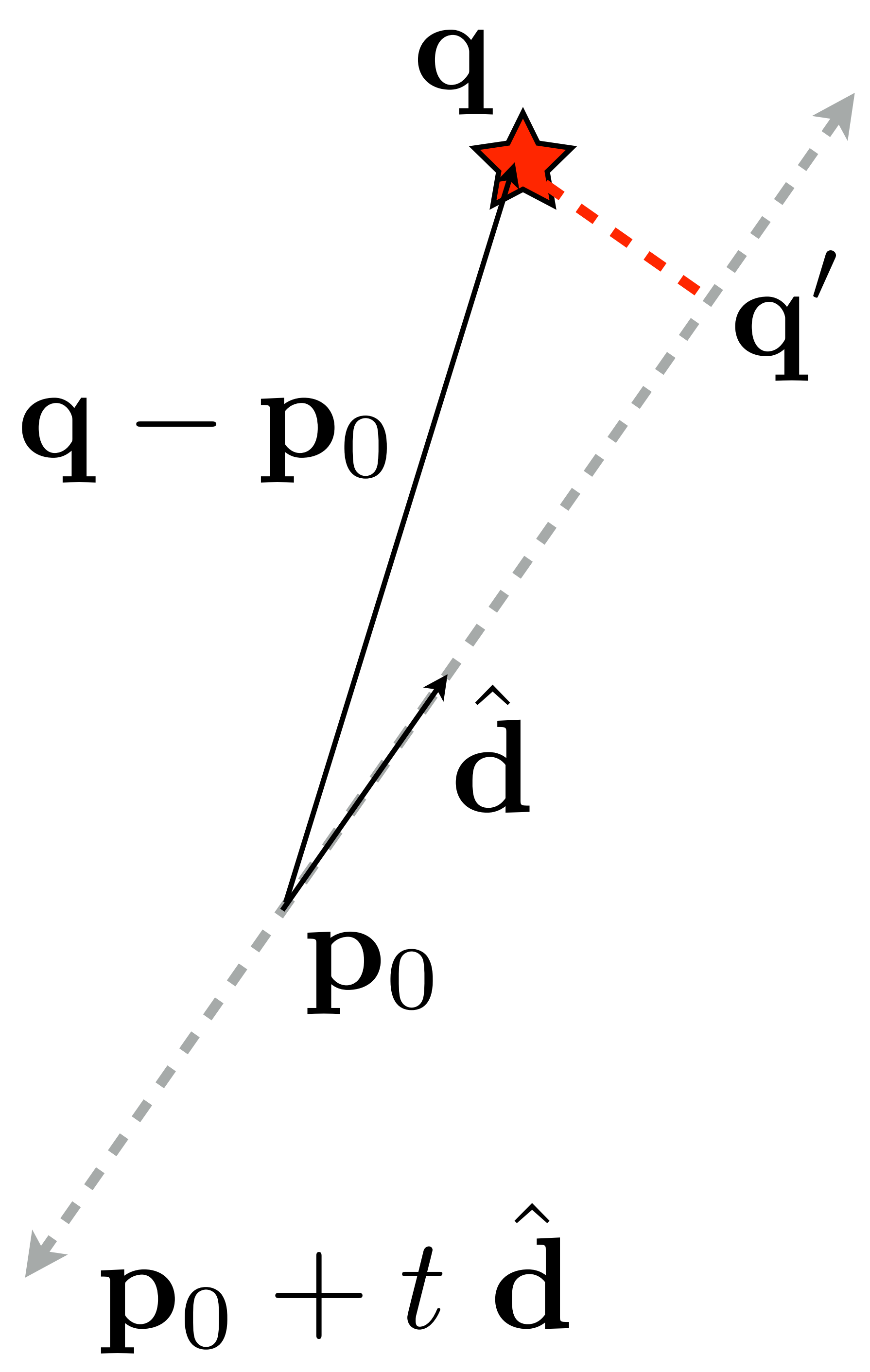
Closest Point to a Line

- To get the closest point on the line **L** to point **q**, go back along the normal direction:

$$\mathbf{q}' = \mathbf{q} - (\mathbf{q} \cdot \hat{\mathbf{n}} - d)\ \hat{\mathbf{n}}$$



Closest Point to a Line



At what value of t is the line closest to q?

$$t = (\mathbf{q} - \mathbf{p}_0) \cdot \hat{\mathbf{d}}$$

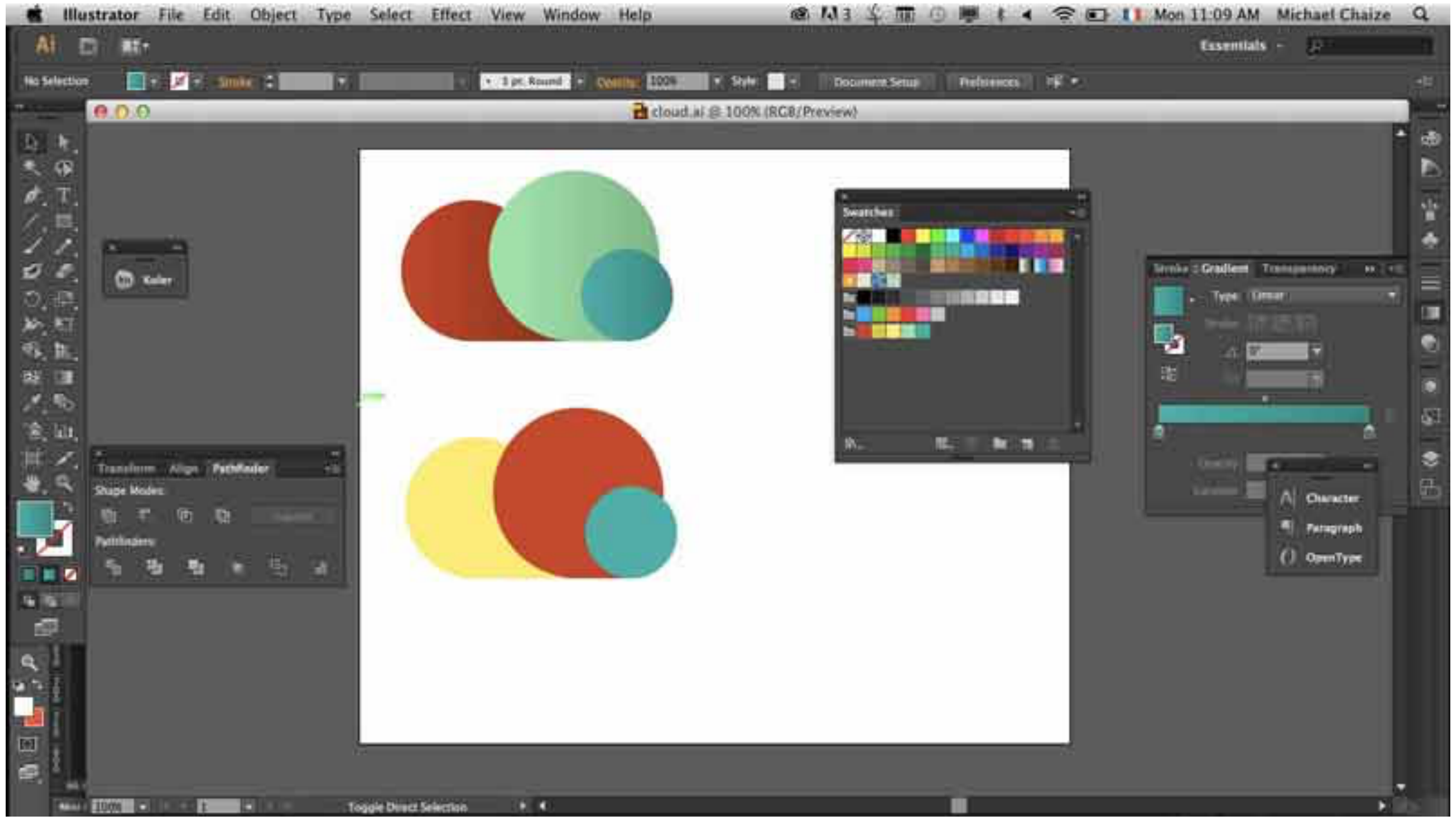
$$\mathbf{q}' = \mathbf{p}_0 + ((\mathbf{q} - \mathbf{p}_0) \cdot \hat{\mathbf{d}})\ \hat{\mathbf{d}}$$

$$\text{distance} = \|\mathbf{q} - \mathbf{q}'\|$$

Works for higher dimensions as well!

Point-In-Shape Tests

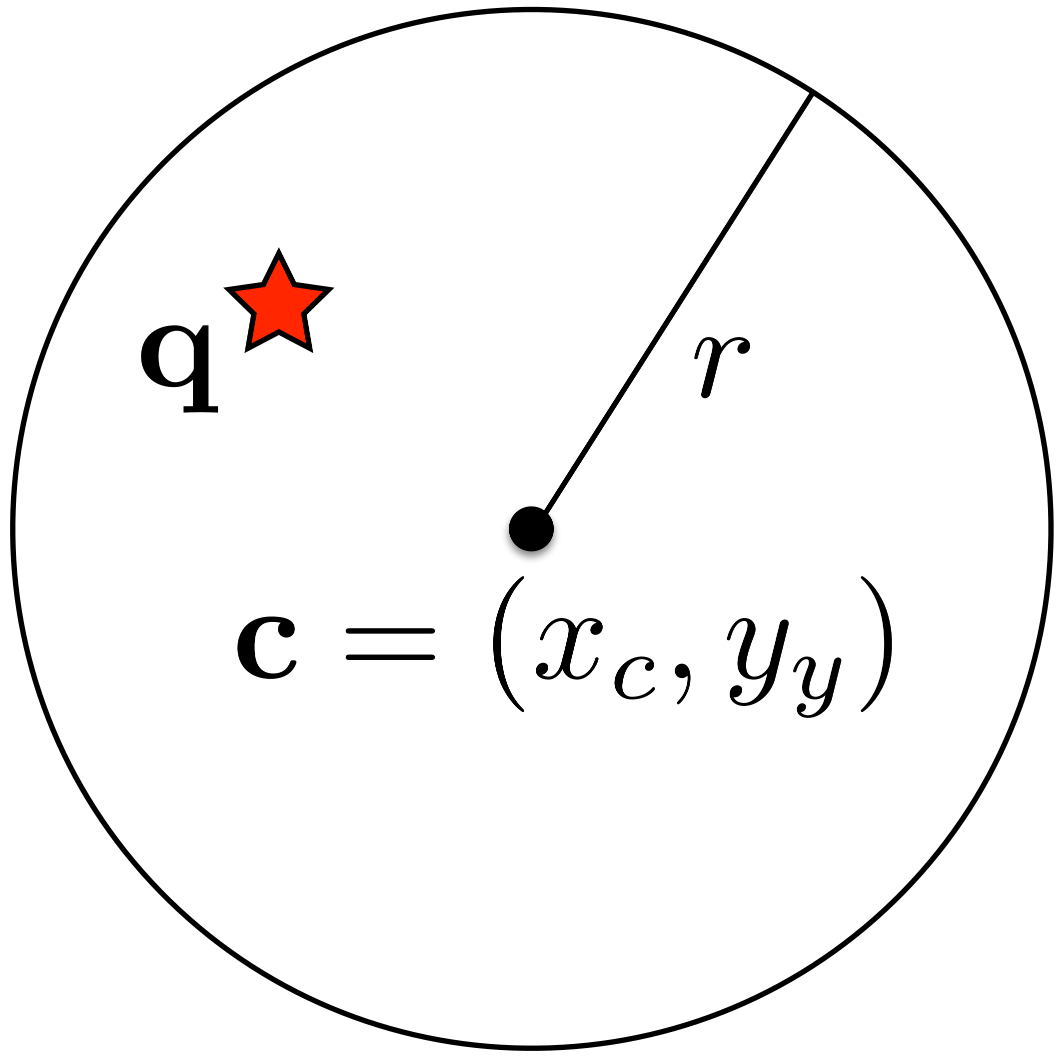
- Useful for selection tests in graphical interaction
- “Did the user click on…”



Point in Circle

Point on the circle:

$$\|\mathbf{q} - \mathbf{c}\| = r$$

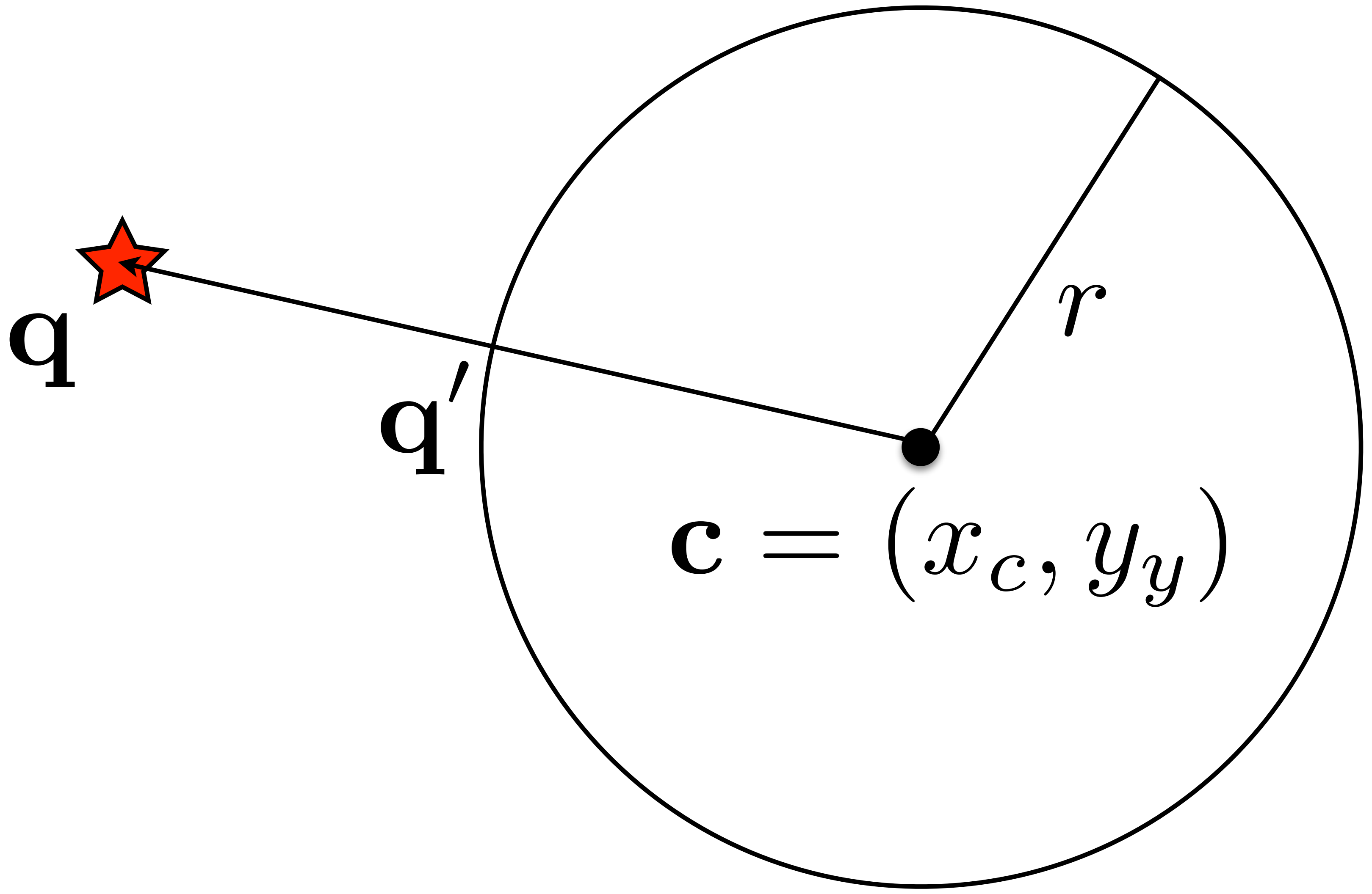


Point in the circle:

$$\|\mathbf{q} - \mathbf{c}\| \leq r$$

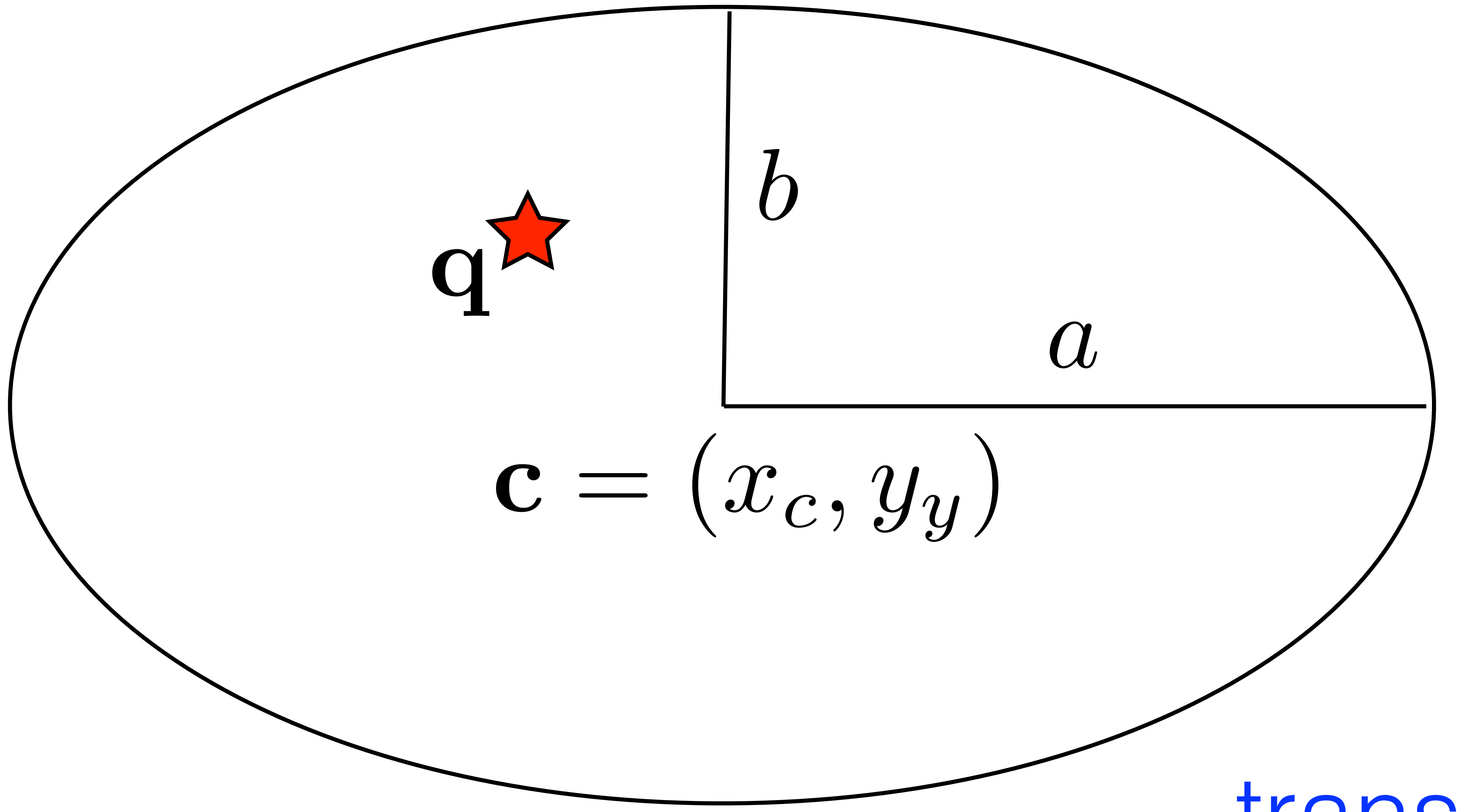
$$(q_x - c_x)^2 + (q_y - c_y)^2 \leq r^2$$

Closest Point on Circle



$$\mathbf{q}' = \mathbf{c} + r\ \frac{\mathbf{q} - \mathbf{c}}{\|\mathbf{q} - \mathbf{c}\|}$$

Point in Ellipse (Oval)



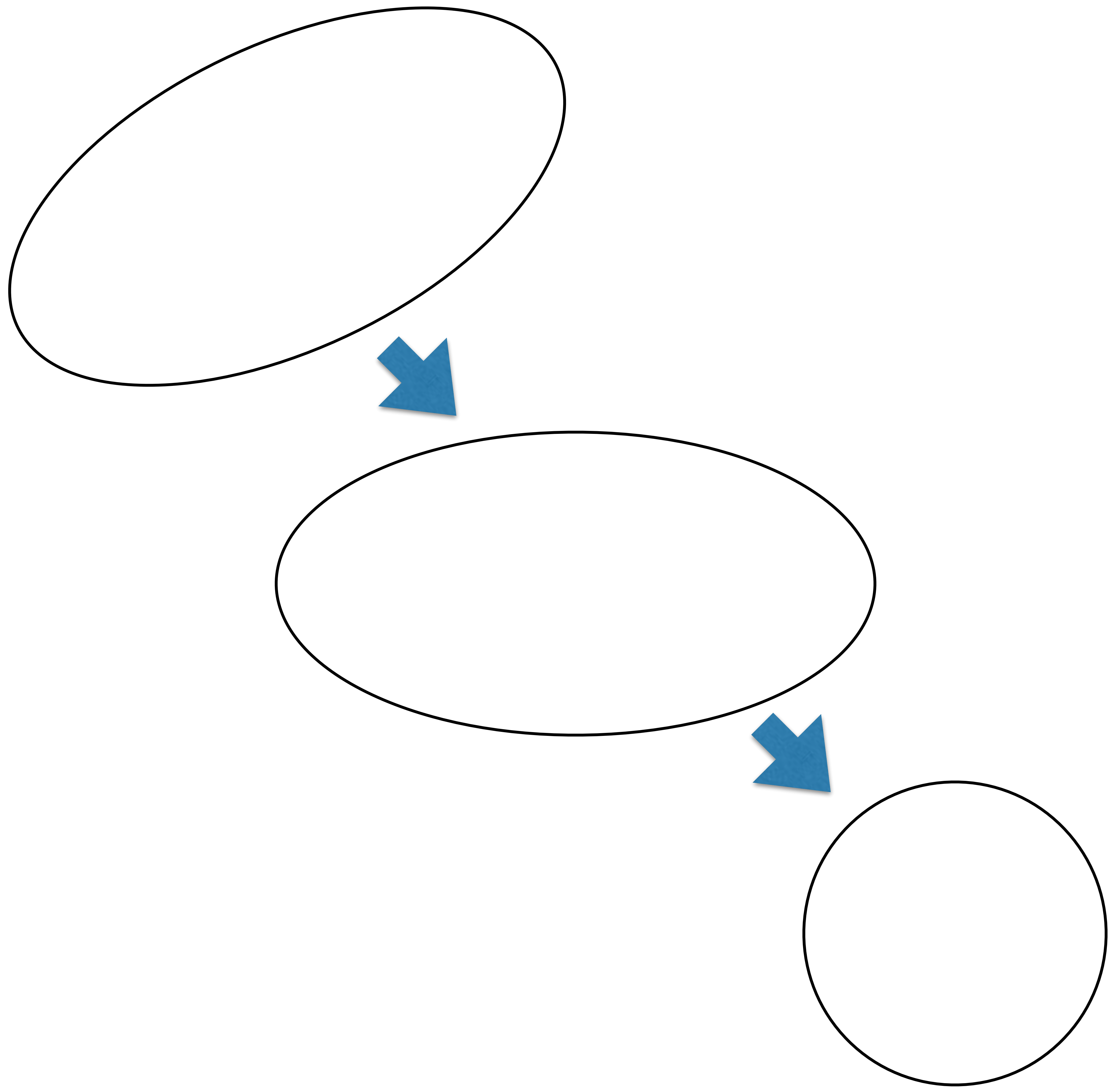
translate!

$$\left(\frac{q_x - c_x}{a}\right)^2 + \left(\frac{q_y - c_y}{b}\right)^2 \leq 1$$

scale!

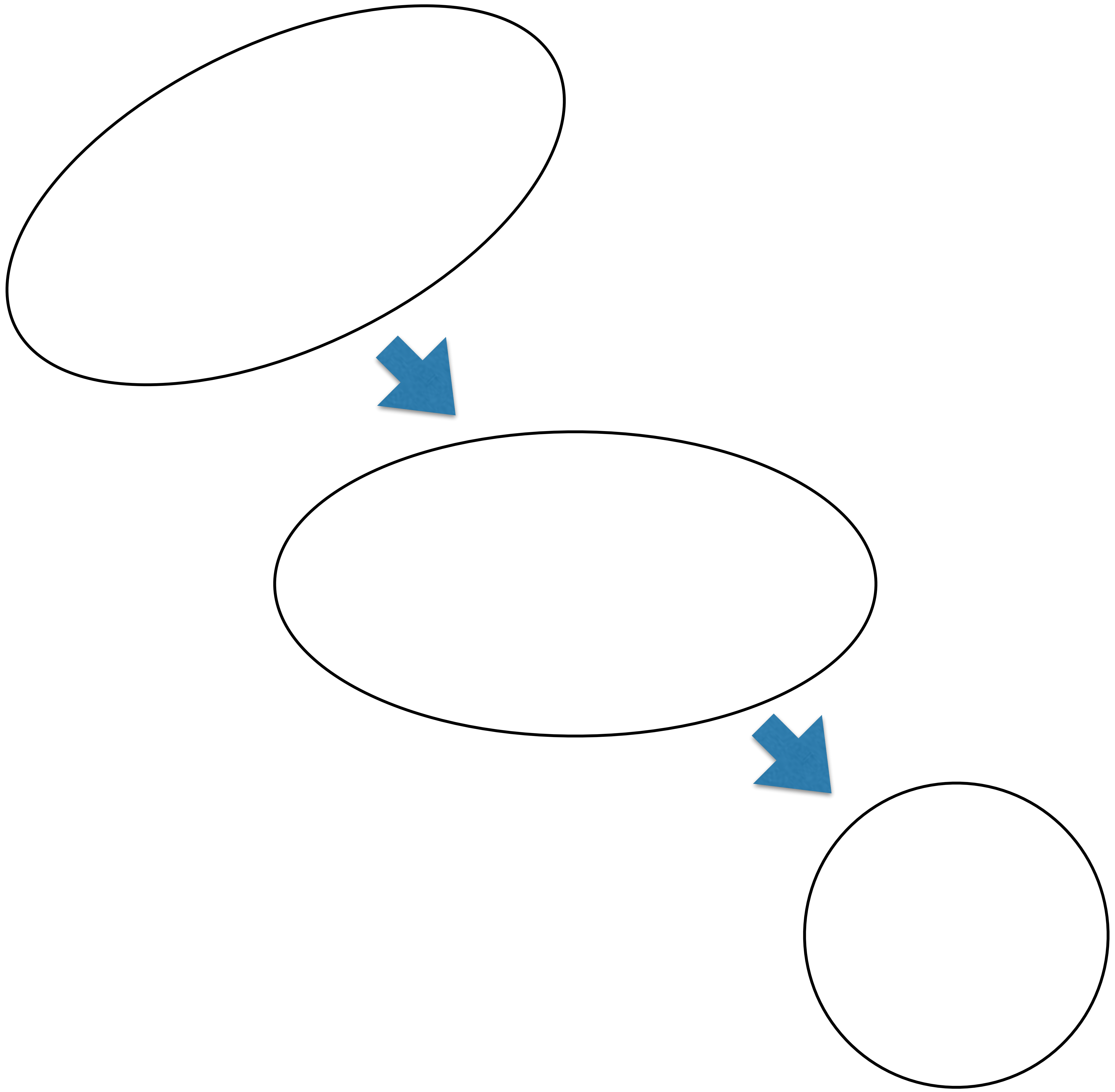
Transformations and Tests

- What seem like complicated tests become easier after transformation
- Center the shape
- Rotate to standard alignment
- Scale if necessary
- Simple tests

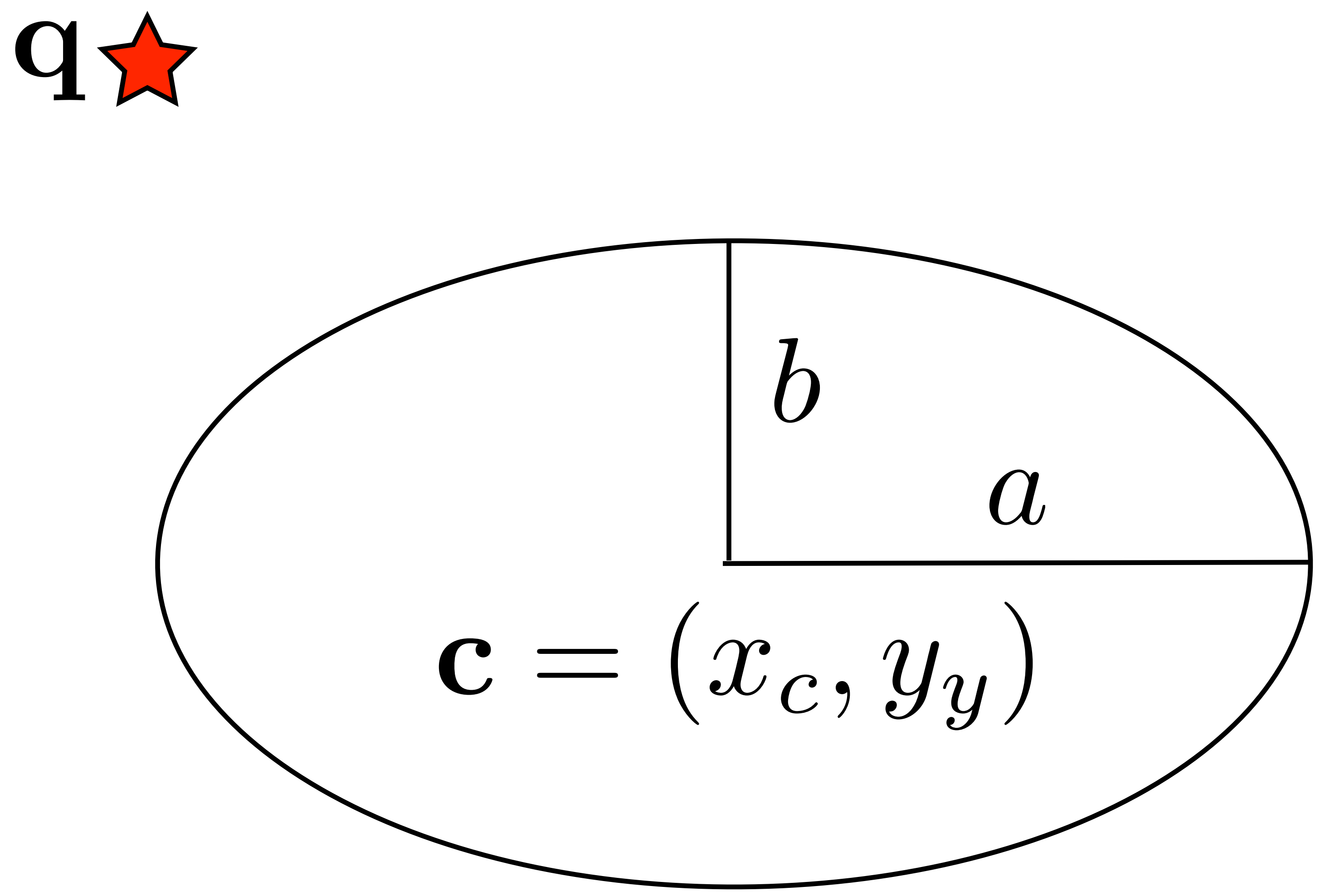


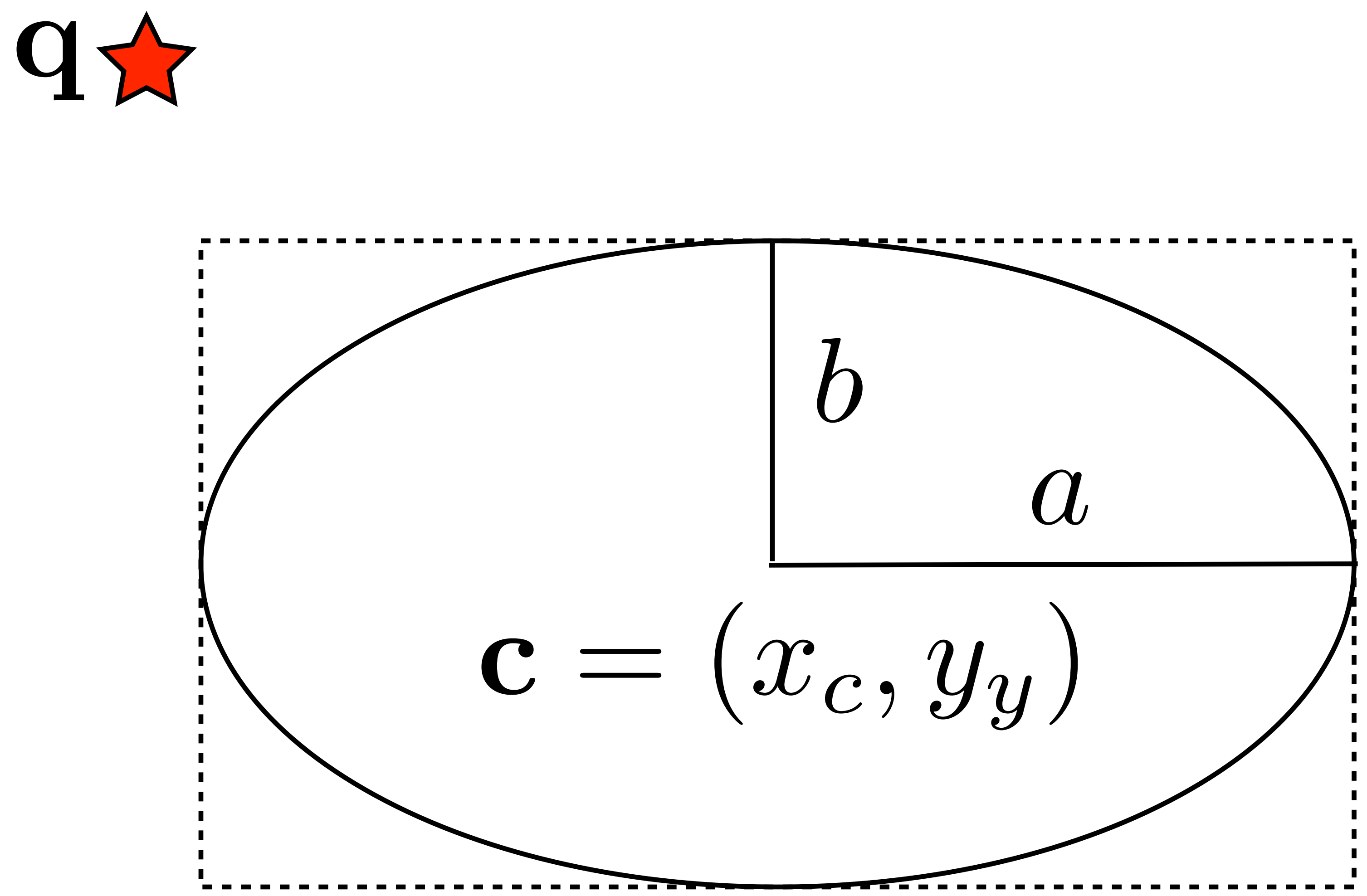
Transformations and Tests

- **Object-to-world** for rendering
- **World-to-object** for selection (geometric tests are often easiest the object's own intrinsic space)



Bounding Boxes





Hard test:

$$\left(\frac{q_x - c_x}{a}\right)^2 + \left(\frac{q_y - c_y}{b}\right)^2 \leq 1$$

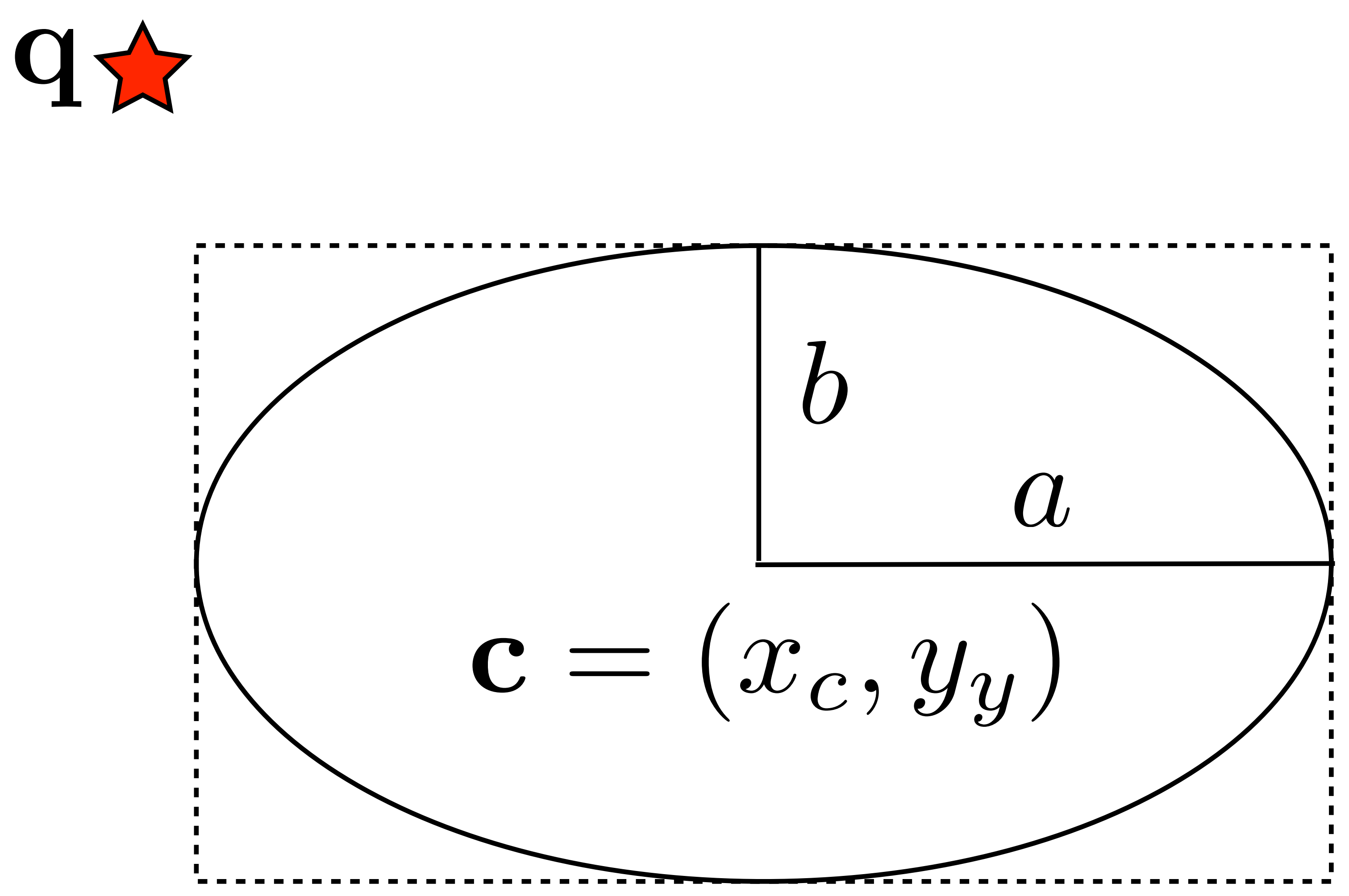
Easy test:

$$|q_x - c_x| \leq a$$

$$|q_y - c_y| \leq b$$

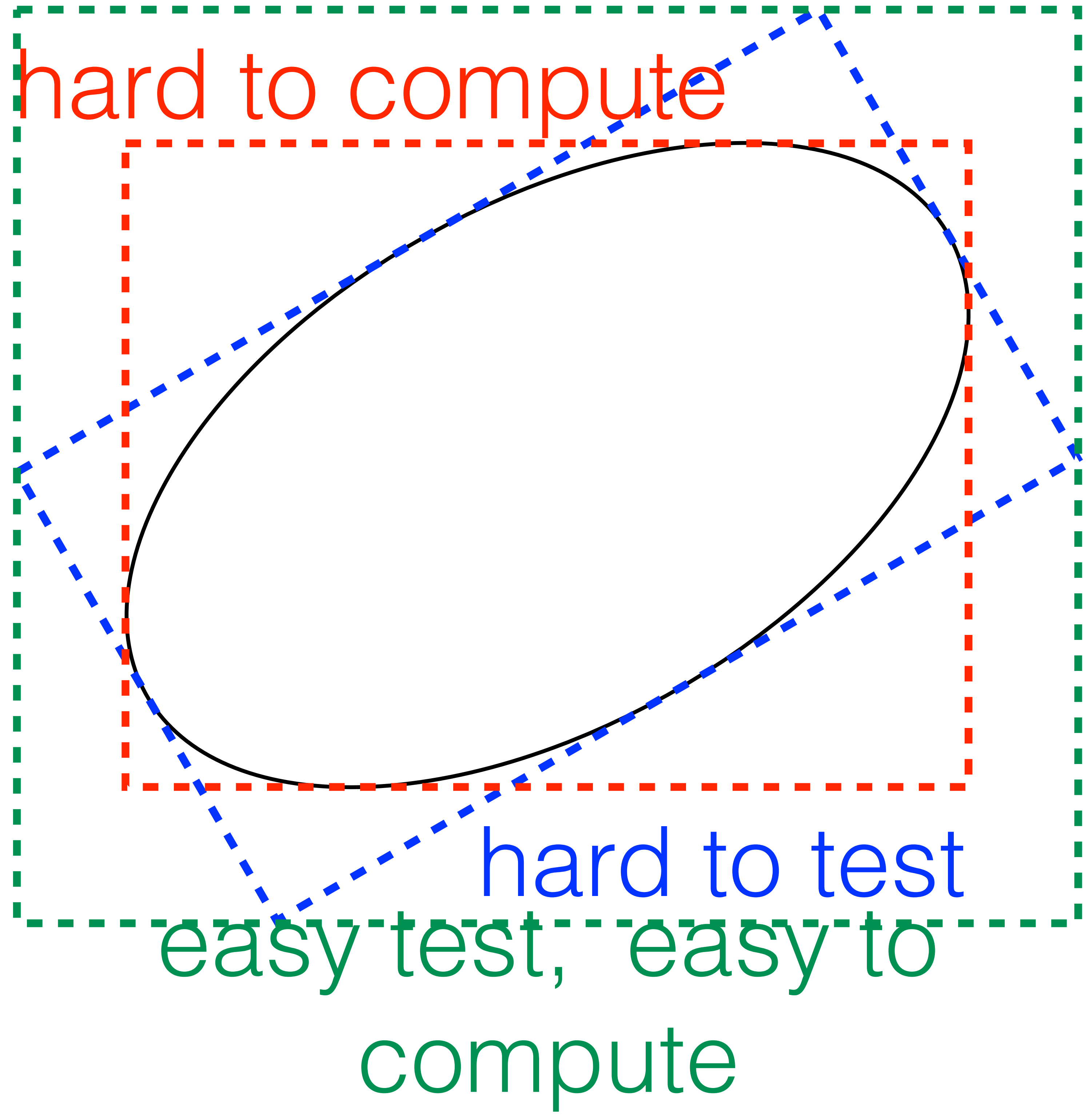
Bounding Boxes

- Idea: use bounding box tests as a "quick reject"
- If passes, then spend time on more complex tests
- The more complex the test, the bigger the win



Bounding Boxes

- Boxes don't necessarily have to be "tight" to be useful
- Looser boxes may be easier to compute and test
- "Axis-Aligned Bounding Box" (AABB)

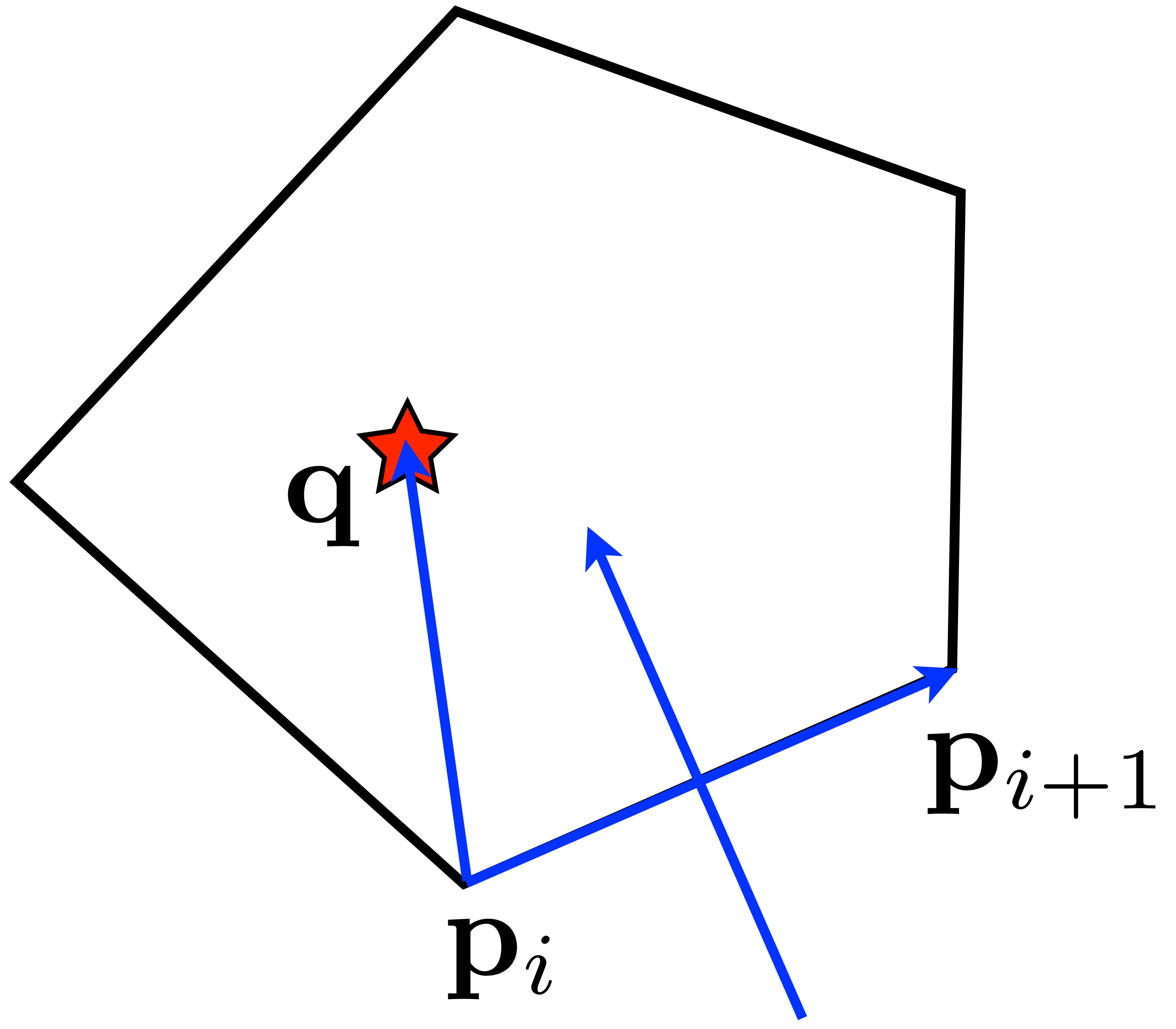


Convex Polygons

- In 2D: for all edges, the point is on the same side of the edge
- Walk around the polygon (in order) and test

$$(\mathbf{q} - \mathbf{p}_i) \cdot (\mathbf{p}_{i+1} - \mathbf{p}_i)_\perp > 0$$

- Be consistent with ordering and perpendiculars

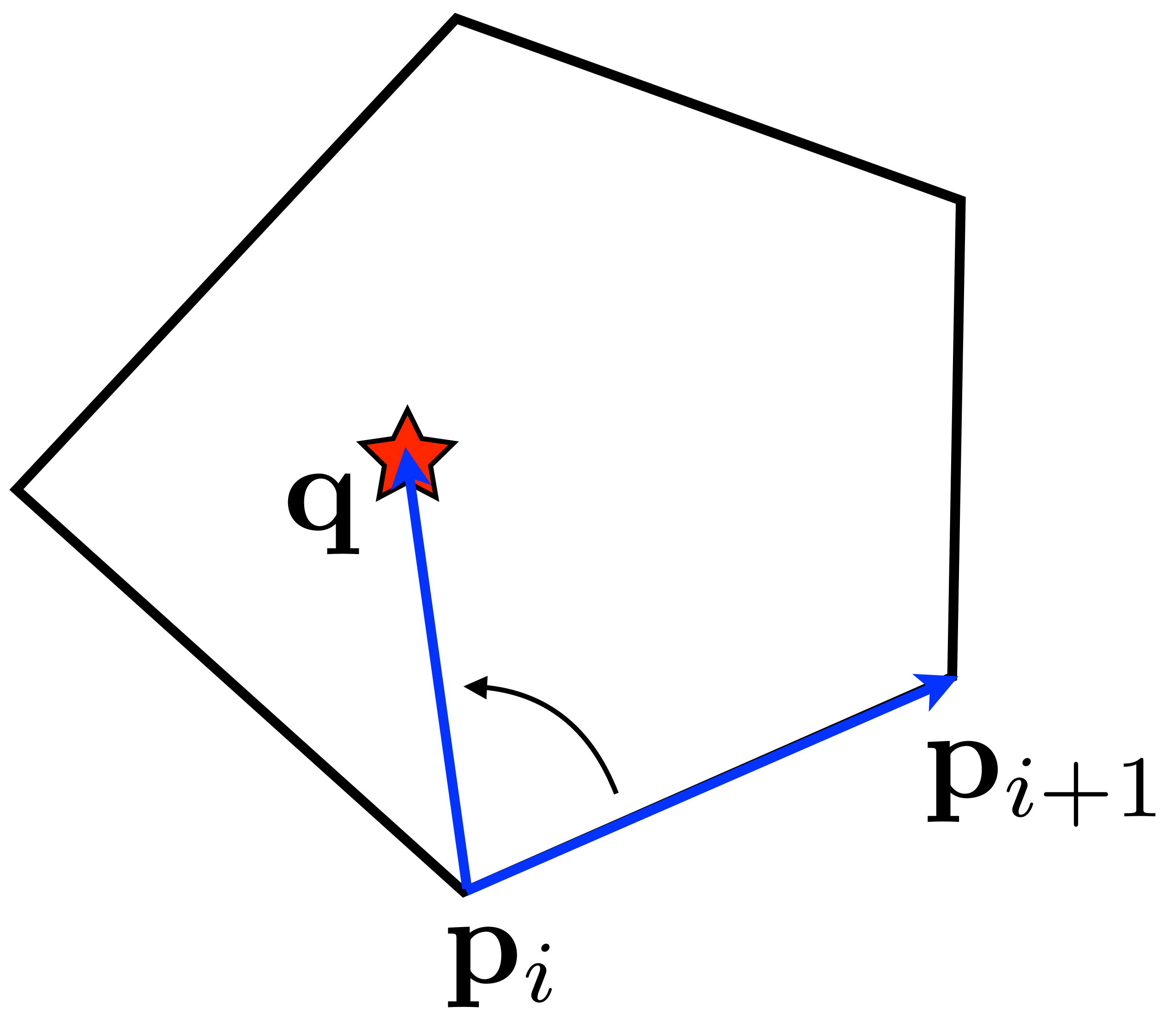


Convex Polygons

- Can also do using cross-products
- Turn all 2-D points (x,y) to 3-D ones (x,y,0)
- Test the z component of this cross-product:

$$(\mathbf{p}_{i+1} - \mathbf{p}_i) \times (\mathbf{q} - \mathbf{p}_i)$$

- If all are positive, the point is in the shape



Polygon Bounding Boxes

- Bounding boxes for polygons are really easy
- Just min / max tests over all of the vertices' x and y coordinates

$$\min(x_i)$$

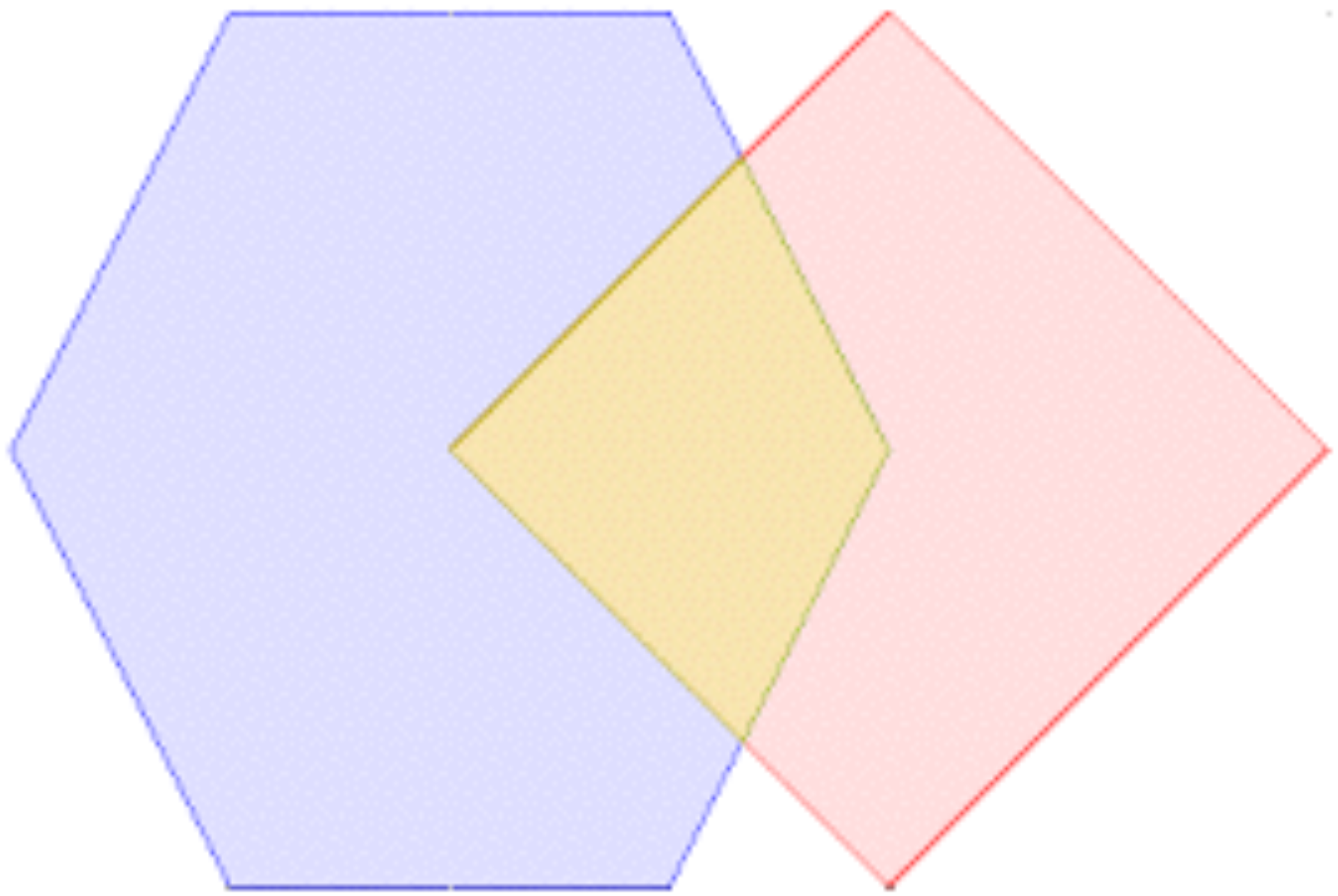
$$\min(y_i)$$

$$\max(x_i)$$

$$\max(y_i)$$

Intersecting Polygons

- To test to see if two convex polygons intersection, see if any of the vertices of one are within the other, and vice versa



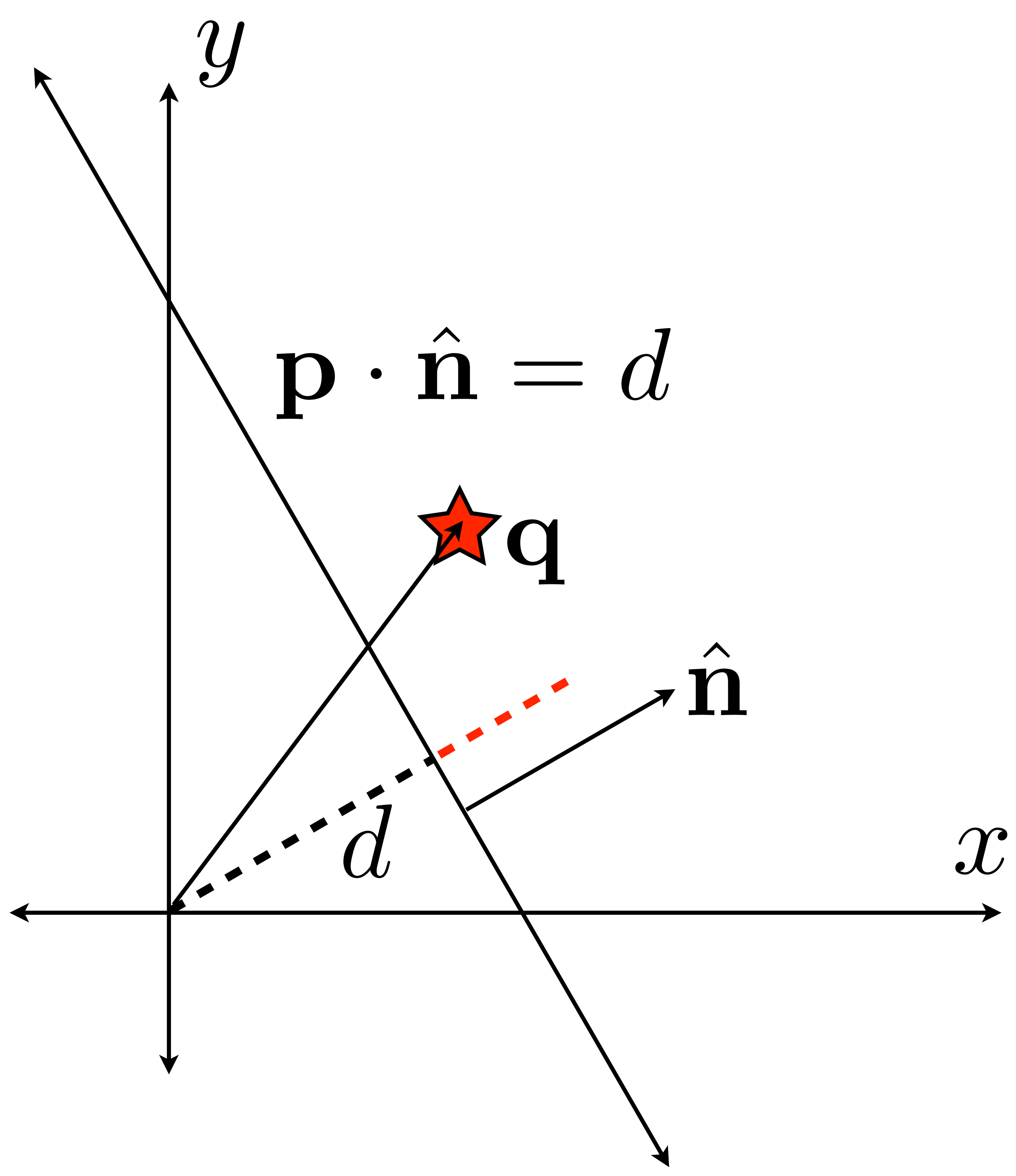
Which Side of a Line?

- Points **p** on the line **L** satisfy this constraint:

$$\mathbf{p} \cdot \hat{\mathbf{n}} - d = 0$$

- Use to test which side of a line a point is on by looking at the sign of

$$\mathbf{q} \cdot \hat{\mathbf{n}} - d$$



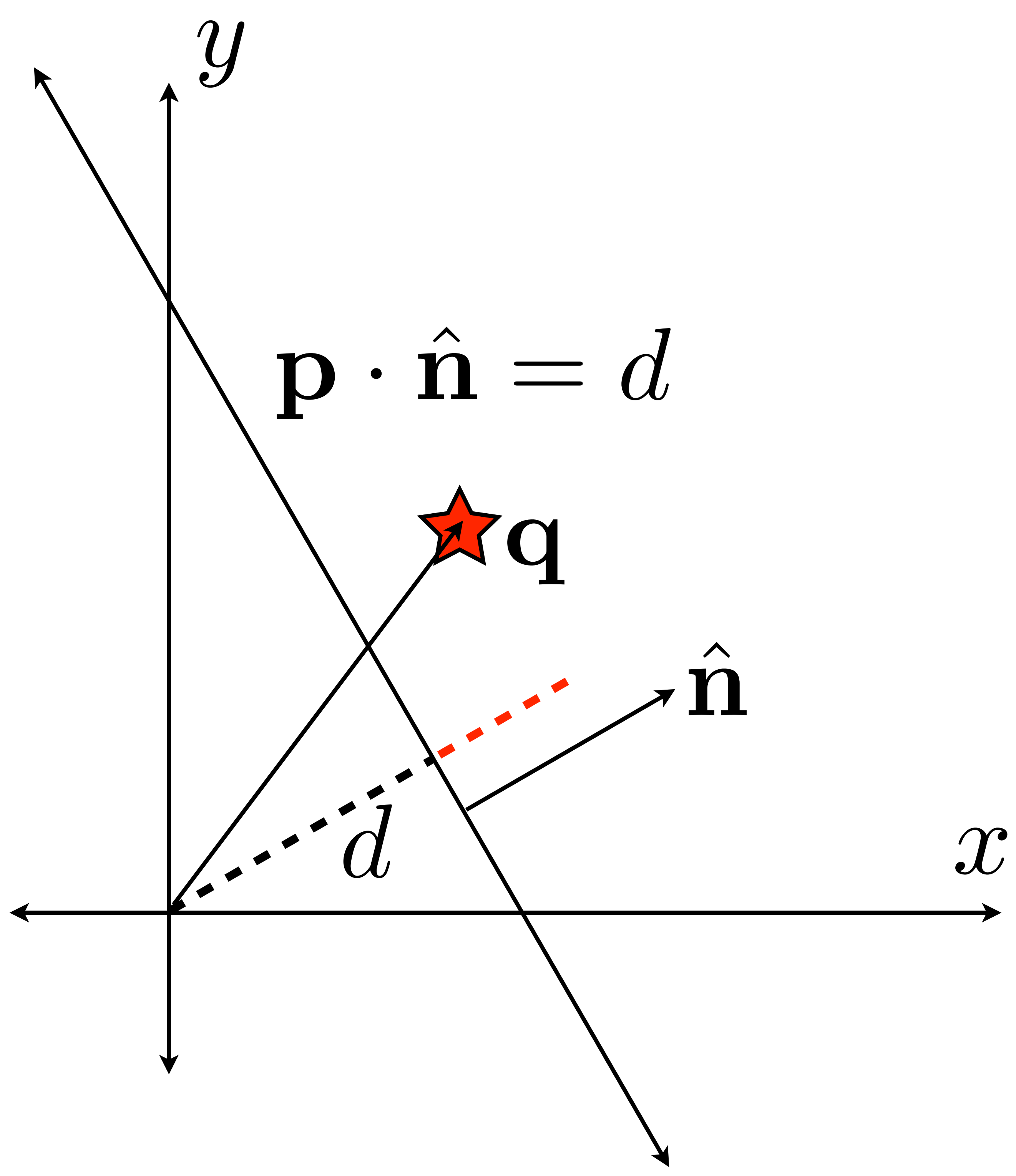
Which Side of a Plane?

- A 2-D plane in 3-D space can be represented the same way:

$$\mathbf{p} \cdot \hat{\mathbf{n}} - d = 0$$

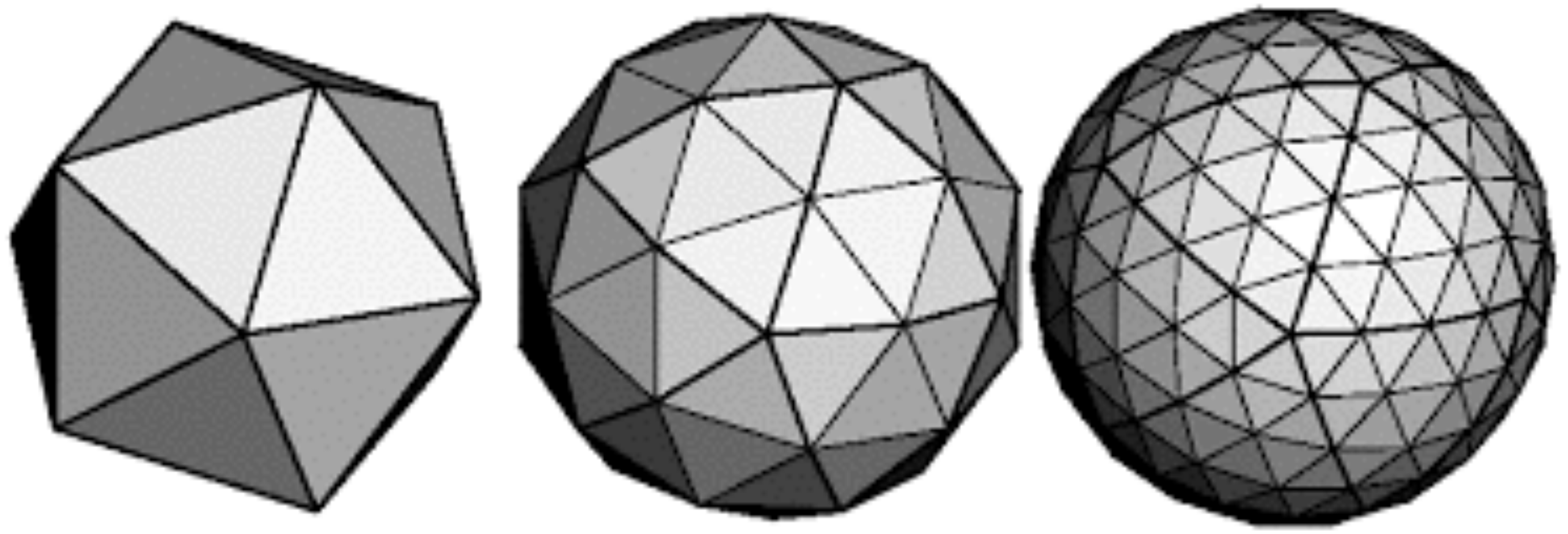
- Use to test which side of a plane a point is on by looking at the sign of

$$\mathbf{q} \cdot \hat{\mathbf{n}} - d$$



Point Inside Polygonal Model

- A point is inside a convex polygonal model if it is on the "inside" side of all of the faces
- Can test for intersection of two convex shapes by seeing if a vertex of another is inside the other, and vice versa



Lab 9

- Don't perform backward mapping for every point in the image
- Compute a bounding box around the four corners of the frame (min/max on the x, y coordinates)
- Test to see if each point is inside the quadrilateral of the frame (point in convex polygon test)

Lab 9

- We give you an implementation of the 4-point algorithm:
  - You give it matching points (match corners of the target frame to corners of the image)
  - It gives you back a homography
- You implement:
  - Point-in-frame geometric tests
    - Iterate over bounding box
    - Point-in-quadrilateral tests
  - Backward warping
  - Bilinear interpolation

Coming up...

- Frequency-domain processing