TRIANGLES

the last primitive

BASED ON MIT 6.837

slides adapted & project started code translated to Swift by Dion Larson
adapted course materials available for free here
original course materials available for free here

Mathematical Toolbox

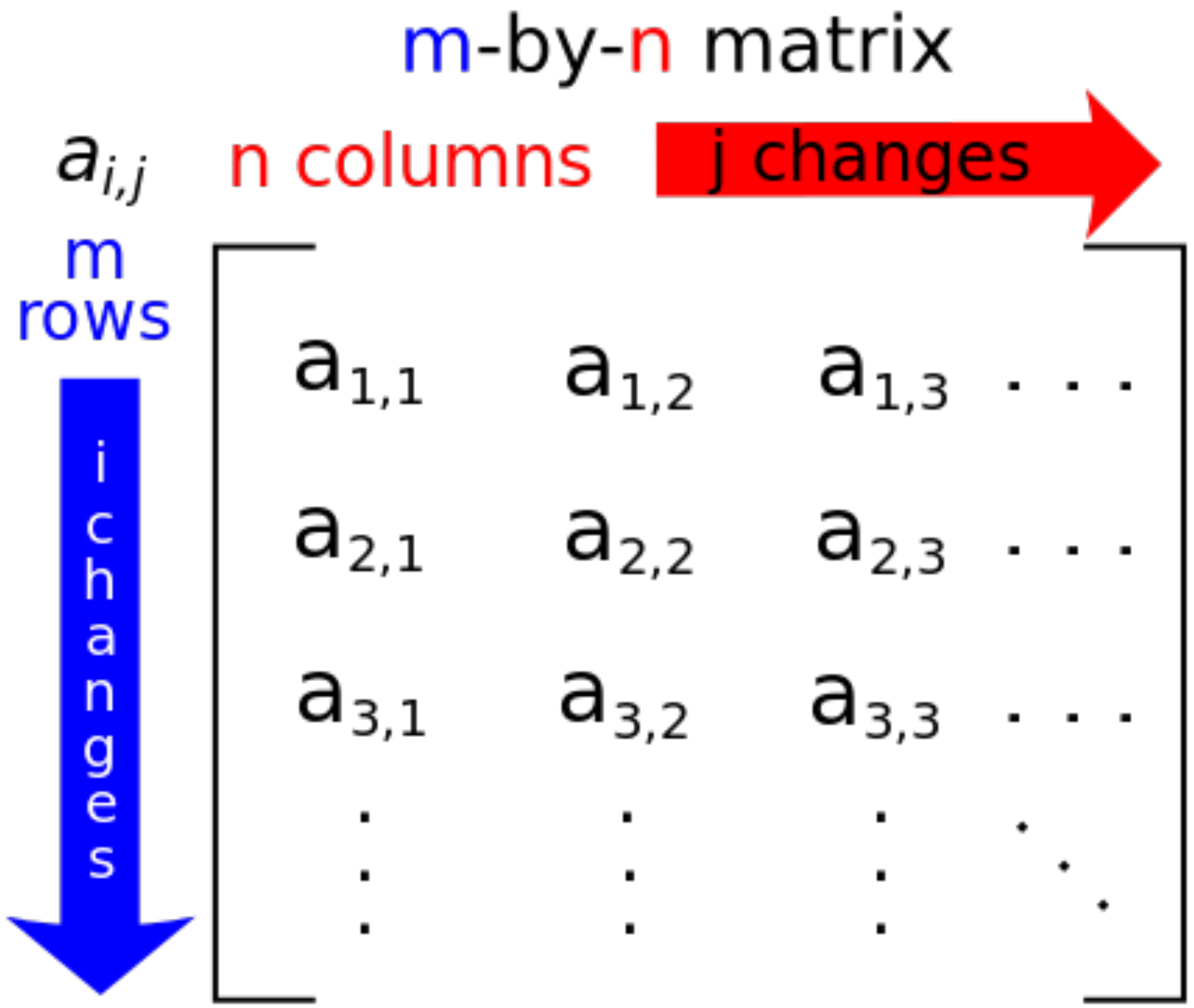
Ray-triangle intersection

Next week

Recap of transforms

Mob programming (shading & transforms)

MATRICES



https://en.wikipedia.org/wiki/Matrix_(mathematics)

https://www.khanacademy.org/math/precalculus/precalc-matrices

DETERMINANT

Value computed from square matrix

Useful in triangle intersection calculations

See MathHelper.swift for implementation

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

$$= aei + bfg + cdh - ceg - bdi - afh.$$

DETERMINANT

Value computed from square matrix

Useful in triangle intersection calculations

See MathHelper.swift for implementation

$$\begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix} = a \begin{vmatrix} f & g & h \\ j & k & l \\ n & o & p \end{vmatrix} - b \begin{vmatrix} e & g & h \\ i & k & l \\ m & o & p \end{vmatrix} + c \begin{vmatrix} e & f & h \\ i & j & l \\ m & n & p \end{vmatrix} - d \begin{vmatrix} e & f & g \\ i & j & k \\ m & n & o \end{vmatrix}.$$

Mathematical Toolbox

Ray-triangle intersection

Next week

Recap of transforms

Mob programming (shading & transforms)

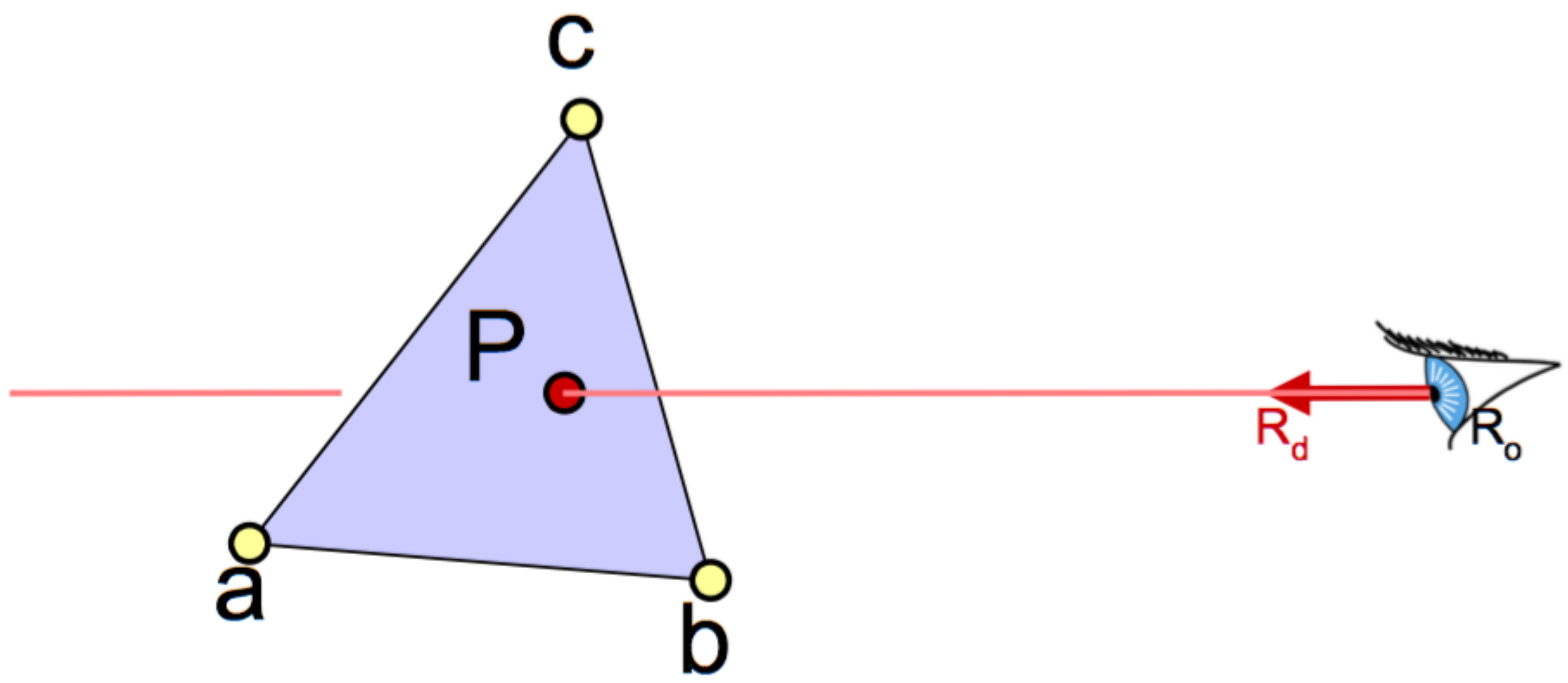
TRIANGLES

Ray-triangle intersection
Meshes of triangles to represent complex geometry

TWO POSSIBLE APPROACHES

Ray-plane intersection + in-triangle test

Barycentric coordinates

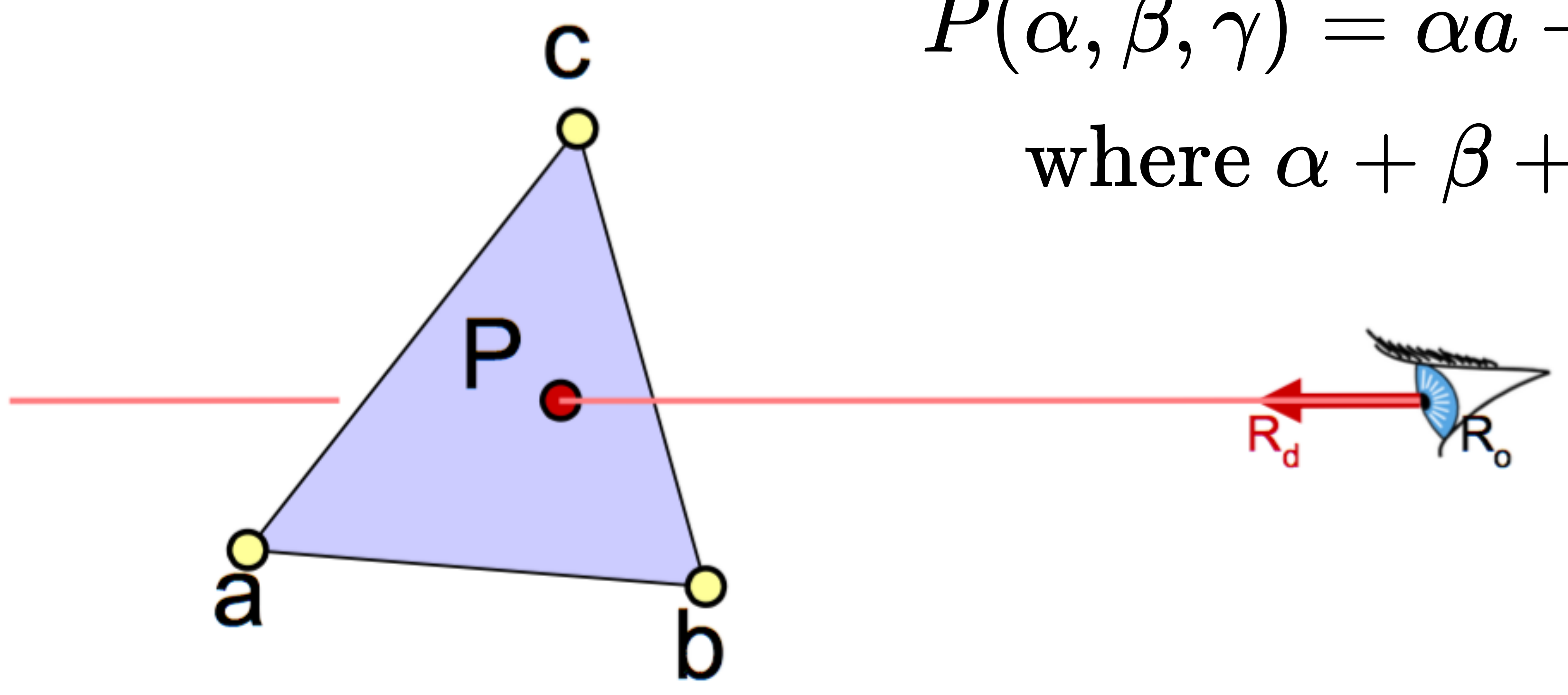


BARYCENTRIC DEFINITION OF A PLANE

A triangle (a, b, c) defines a plane

Points on plane can be written as:

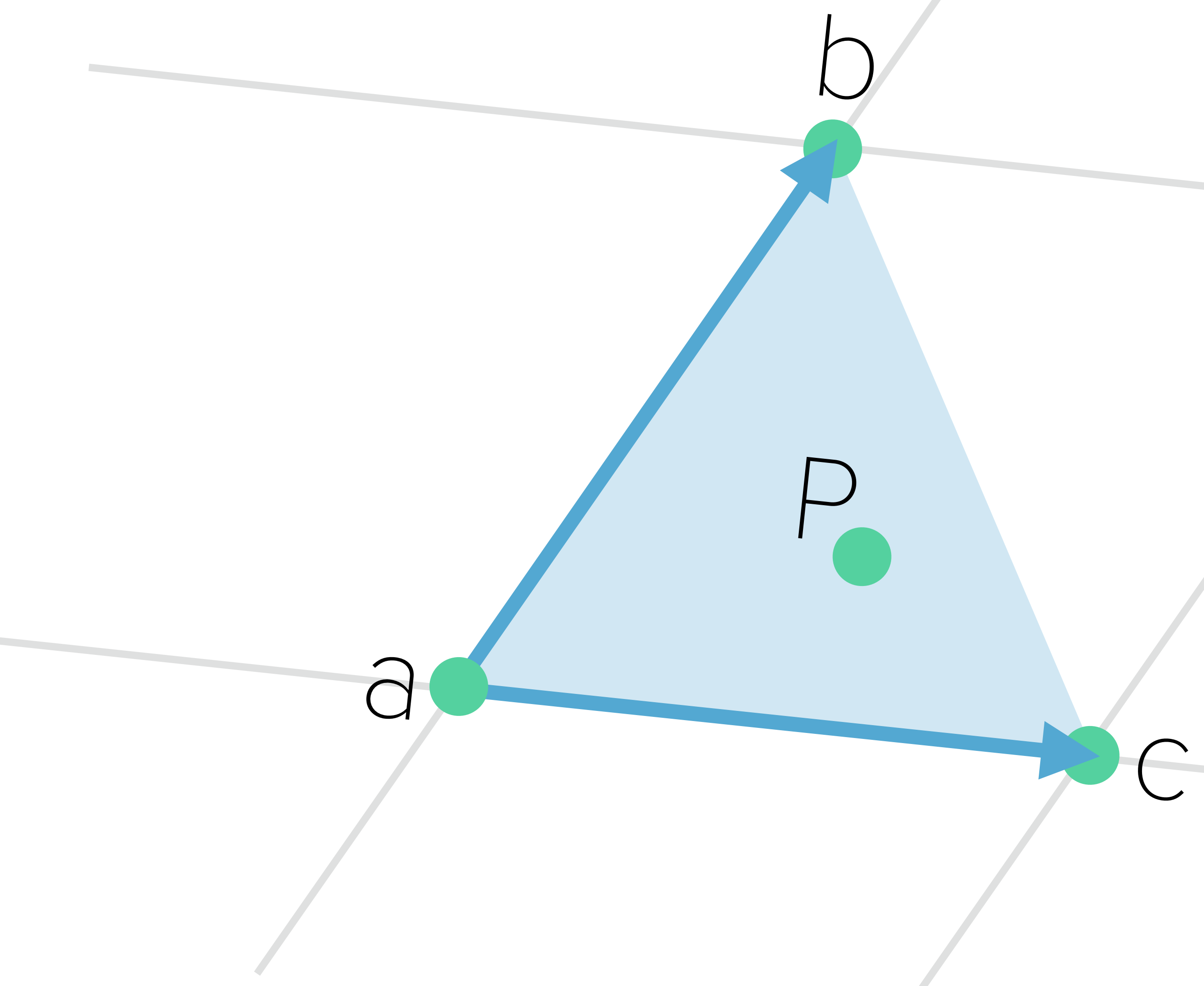
$$P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c$$
$$\text{where } \alpha + \beta + \gamma = 1$$



BARYCENTRIC COORDINATES

Since $\alpha + \beta + \gamma = 1$, we can write $\alpha = 1 - \beta - \gamma$

$$\begin{aligned} P(\alpha, \beta, \gamma) &= \alpha a + \beta b + \gamma c \\ P(\beta, \gamma) &= (1 - \beta - \gamma) a + \beta b + \gamma c \\ &= a + \beta(b - a) + \gamma(c - a) \end{aligned}$$



Vectors that lie on the triangle plane

Non-orthogonal coordinate system on the plane!

BARYCENTRIC DEFINITION OF A TRIANGLE

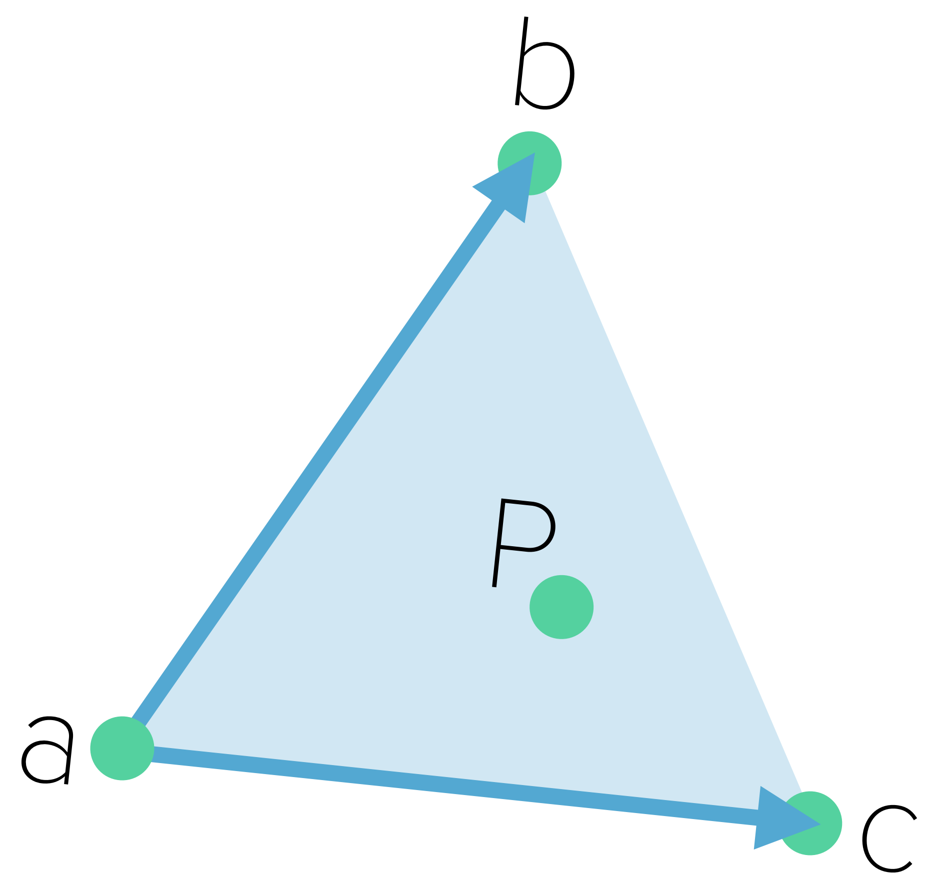
$P(\alpha, \beta, \gamma) = \alpha \boldsymbol{a} + \beta \boldsymbol{b} + \gamma \boldsymbol{c}$ with $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} = 1$

parameterizes the entire plane!

Get just the triangle when $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \geq 0$

Since $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} = \mathbf{1}$ it's implied that:

$$\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}, \mathbf{0} \leq \boldsymbol{\beta} \leq \mathbf{1}, \mathbf{0} \leq \boldsymbol{\gamma} \leq \mathbf{1}$$

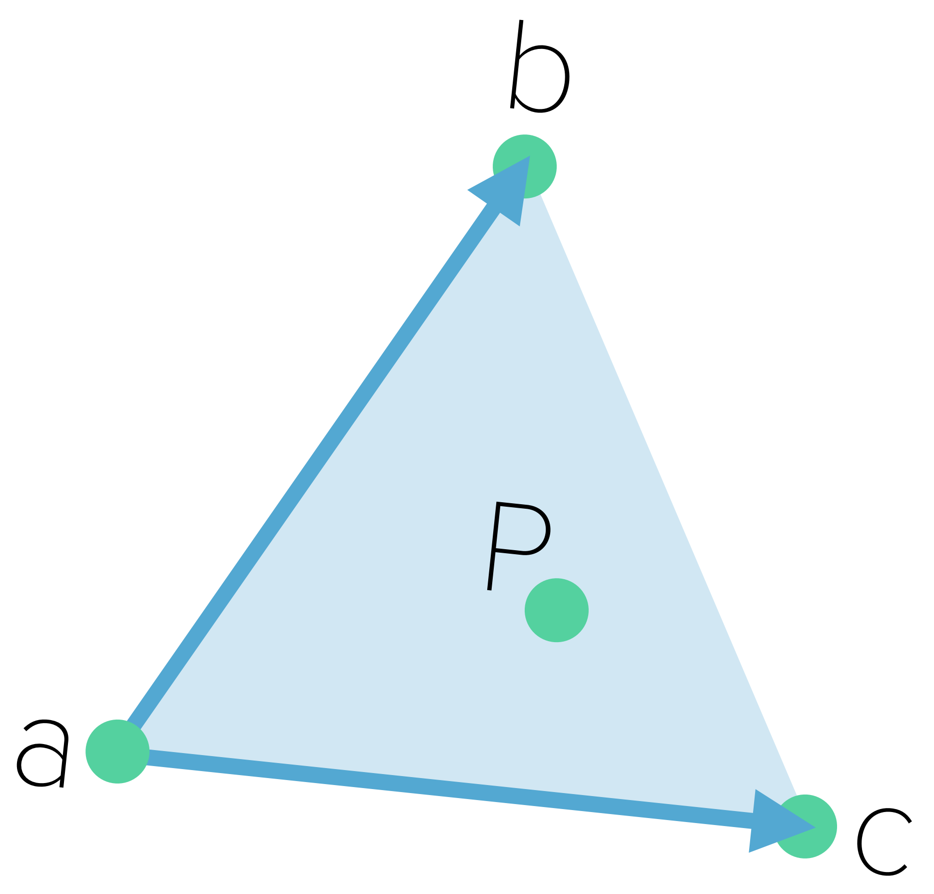


HOW TO COMPUTE α, β, γ ?

Write it out as a 2x2 linear system

$$P(\beta, \gamma) = \boldsymbol{a} + \beta \boldsymbol{e}_1 + \gamma \boldsymbol{e}_2$$

$$\boldsymbol{e}_1 = (\boldsymbol{b} - \boldsymbol{a}), \boldsymbol{e}_2 = (\boldsymbol{c} - \boldsymbol{a})$$



$$\boldsymbol{a} + \beta \boldsymbol{e}_1 + \gamma \boldsymbol{e}_2 - \boldsymbol{P} = 0$$

Linear system of 3 equations and two unknowns!

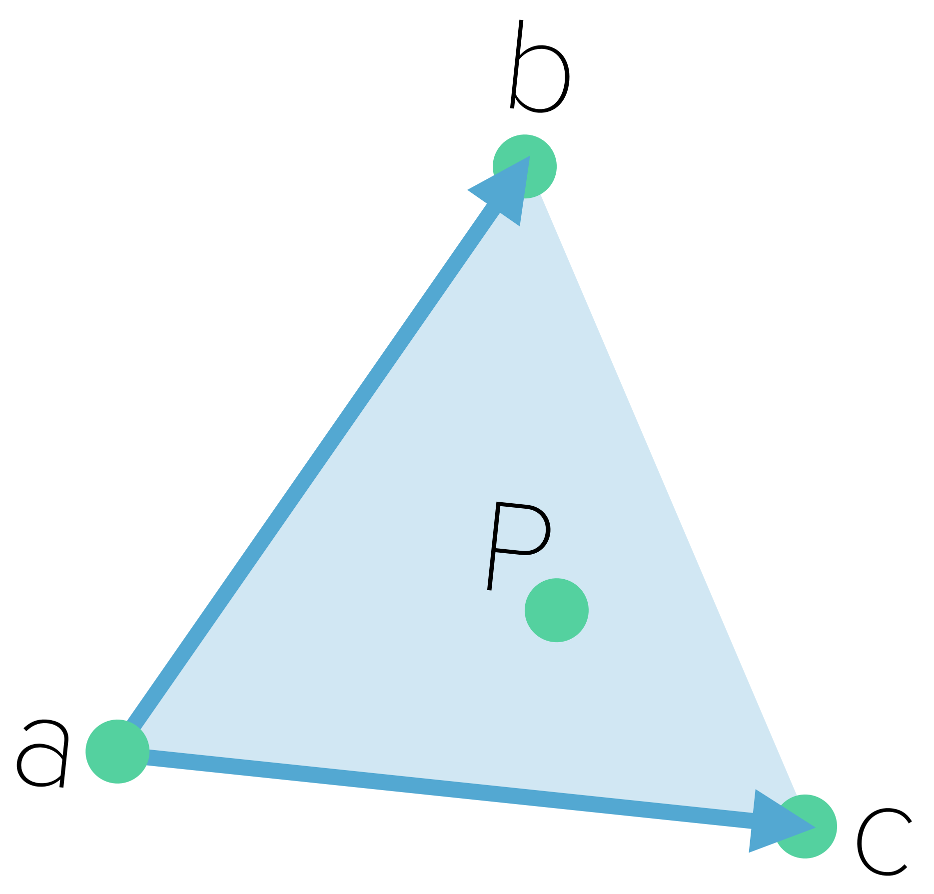
Take the inner products of this equation and $\boldsymbol{e}_1, \boldsymbol{e}_2$

HOW TO COMPUTE α, β, γ ?

Write it out as a 2x2 linear system

$$P(\beta, \gamma) = a + \beta e_1 + \gamma e_2$$

$$e_1 = (b - a), e_2 = (c - a)$$



$$\langle \boldsymbol{e}_1, \ \boldsymbol{a} + \beta \boldsymbol{e}_1 + \gamma \boldsymbol{e}_2 - \boldsymbol{P} \rangle = 0$$

$$\langle \boldsymbol{e}_2, \ \boldsymbol{a} + \beta \boldsymbol{e}_1 + \gamma \boldsymbol{e}_2 - \boldsymbol{P} \rangle = 0$$

$$\begin{pmatrix} \langle \boldsymbol{e}_1, \boldsymbol{e}_1 \rangle & \langle \boldsymbol{e}_1, \boldsymbol{e}_2 \rangle \\ \langle \boldsymbol{e}_2, \boldsymbol{e}_1 \rangle & \langle \boldsymbol{e}_2, \boldsymbol{e}_2 \rangle \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

$$\text{where} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \langle (\boldsymbol{P} - \boldsymbol{a}), \boldsymbol{e}_1 \rangle \\ \langle (\boldsymbol{P} - \boldsymbol{a}), \boldsymbol{e}_2 \rangle \end{pmatrix}$$

$< a, b >$ is the dot product

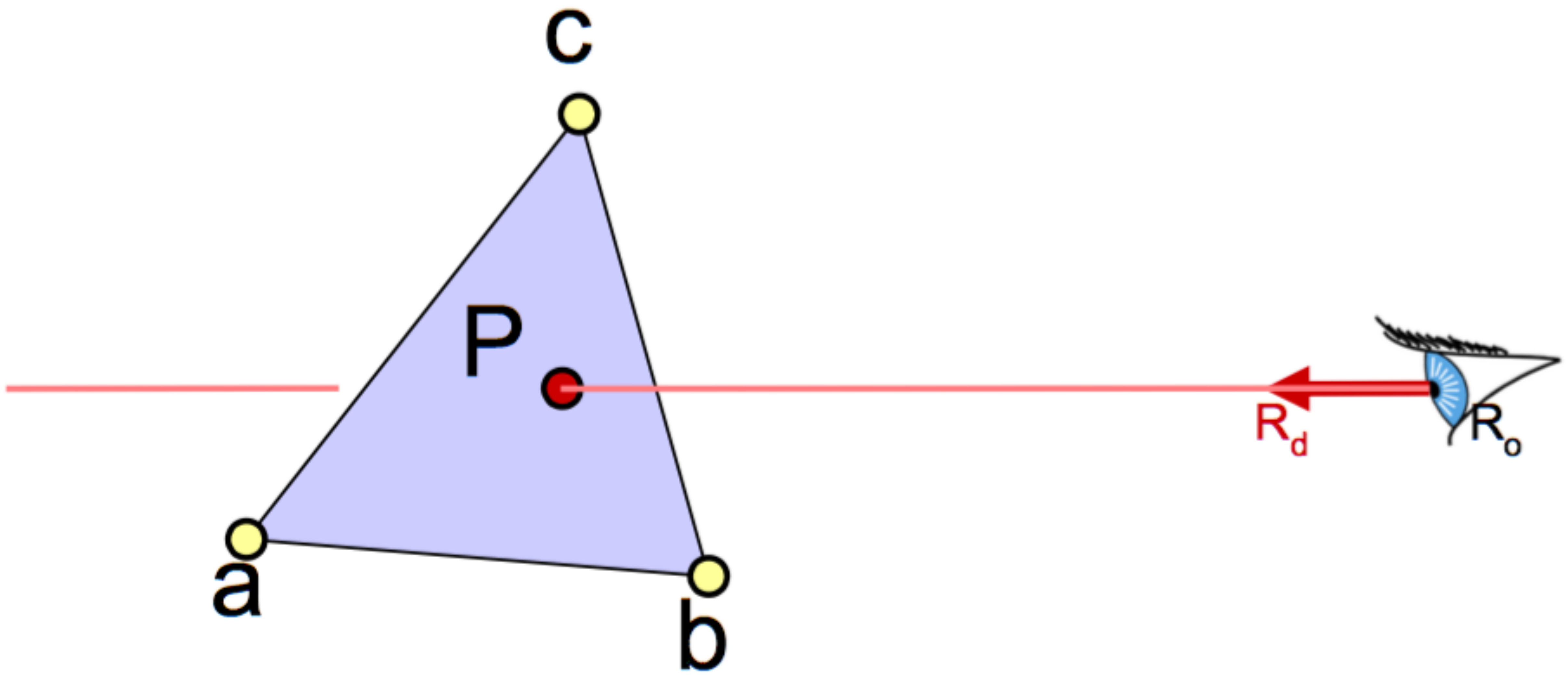
INTERSECTION WITH BARYCENTRIC TRIANGLE

Set ray equation equal to barycentric equation

$$P(t) = P(\beta, \gamma)$$

$$R_o + tR_d = a + \beta(b - a) + \gamma(c - a)$$

Intersection if $\beta + \gamma \leq 1, \beta \geq 0, \gamma \geq 0, t > t_{min}$



INTERSECTION WITH BARYCENTRIC TRIANGLE

$$R_o + tR_d = a + \beta(b - a) + \gamma(c - a)$$

$$R_{ox} + tR_{dx} = a_x + \beta(b_x - a_x) + \gamma(c_x - a_x)$$
$$R_{oy} + tR_{dy} = a_y + \beta(b_y - a_y) + \gamma(c_y - a_y)$$
$$R_{oz} + tR_{dz} = a_z + \beta(b_z - a_z) + \gamma(c_z - a_z)$$

Regroup & write in matrix form **Ax = b**

$$\begin{bmatrix} a_x - b_x & a_x - c_x & R_{dx} \\ a_y - b_y & a_y - c_y & R_{dy} \\ a_z - b_z & a_z - c_z & R_{dz} \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} a_x - R_{ox} \\ a_y - R_{oy} \\ a_z - R_{oz} \end{bmatrix}$$

CRAMER'S RULE

Used to solve for one variable at a time in system of equations

$$\beta = \frac{\begin{vmatrix} a_x - R_{ox} & a_x - c_x & R_{dx} \\ a_y - R_{oy} & a_y - c_y & R_{dy} \\ a_z - R_{oz} & a_z - c_z & R_{dz} \end{vmatrix}}{|A|}$$

$$\gamma = \frac{\begin{vmatrix} a_x - b_x & a_x - R_{ox} & R_{dx} \\ a_y - b_y & a_y - R_{oy} & R_{dy} \\ a_z - b_z & a_z - R_{oz} & R_{dz} \end{vmatrix}}{|A|}$$

$$t = \frac{\begin{vmatrix} a_x - b_x & a_x - c_x & a_x - R_{ox} \\ a_y - b_y & a_y - c_y & a_y - R_{oy} \\ a_z - b_z & a_z - c_z & a_z - R_{oz} \end{vmatrix}}{|A|}$$

| | denotes the determinant

Can be copied mechanically into code

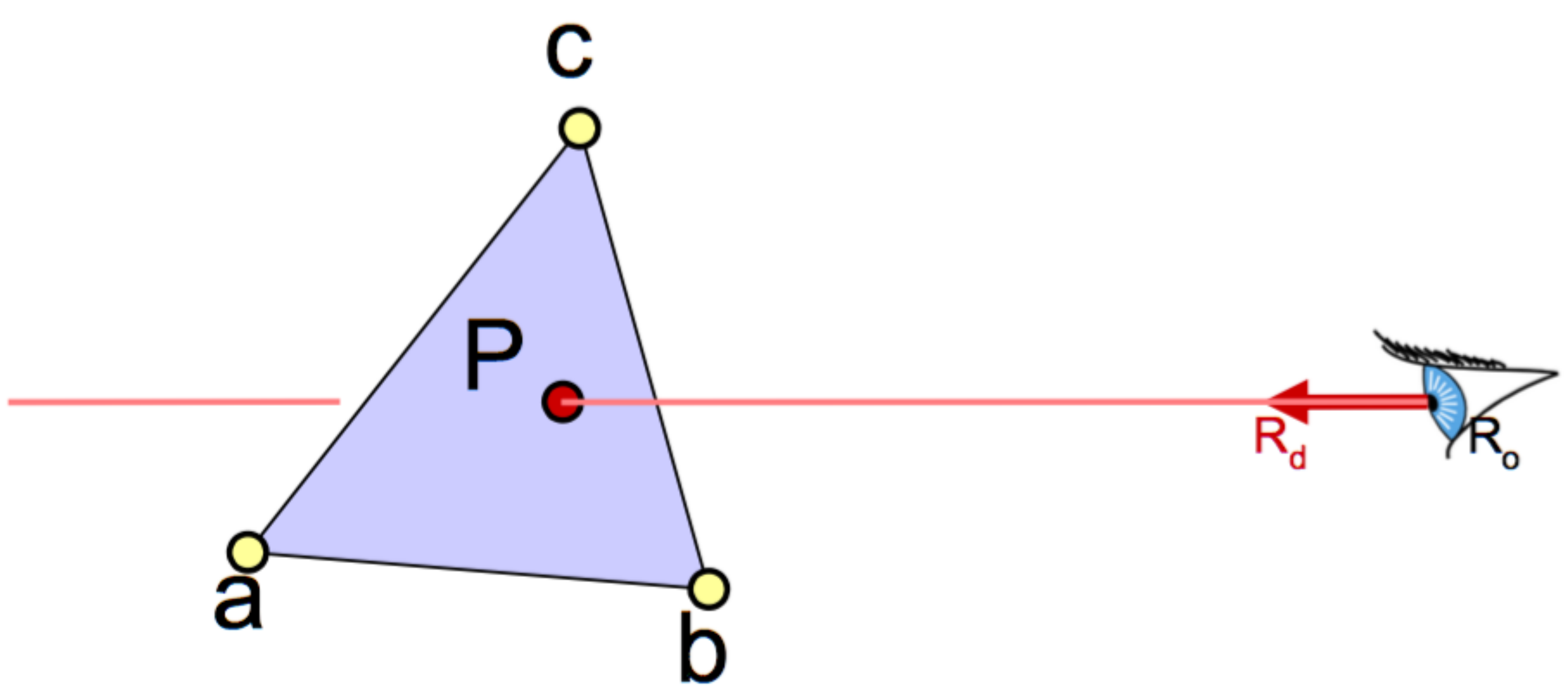
PROS OF THIS APPROACH

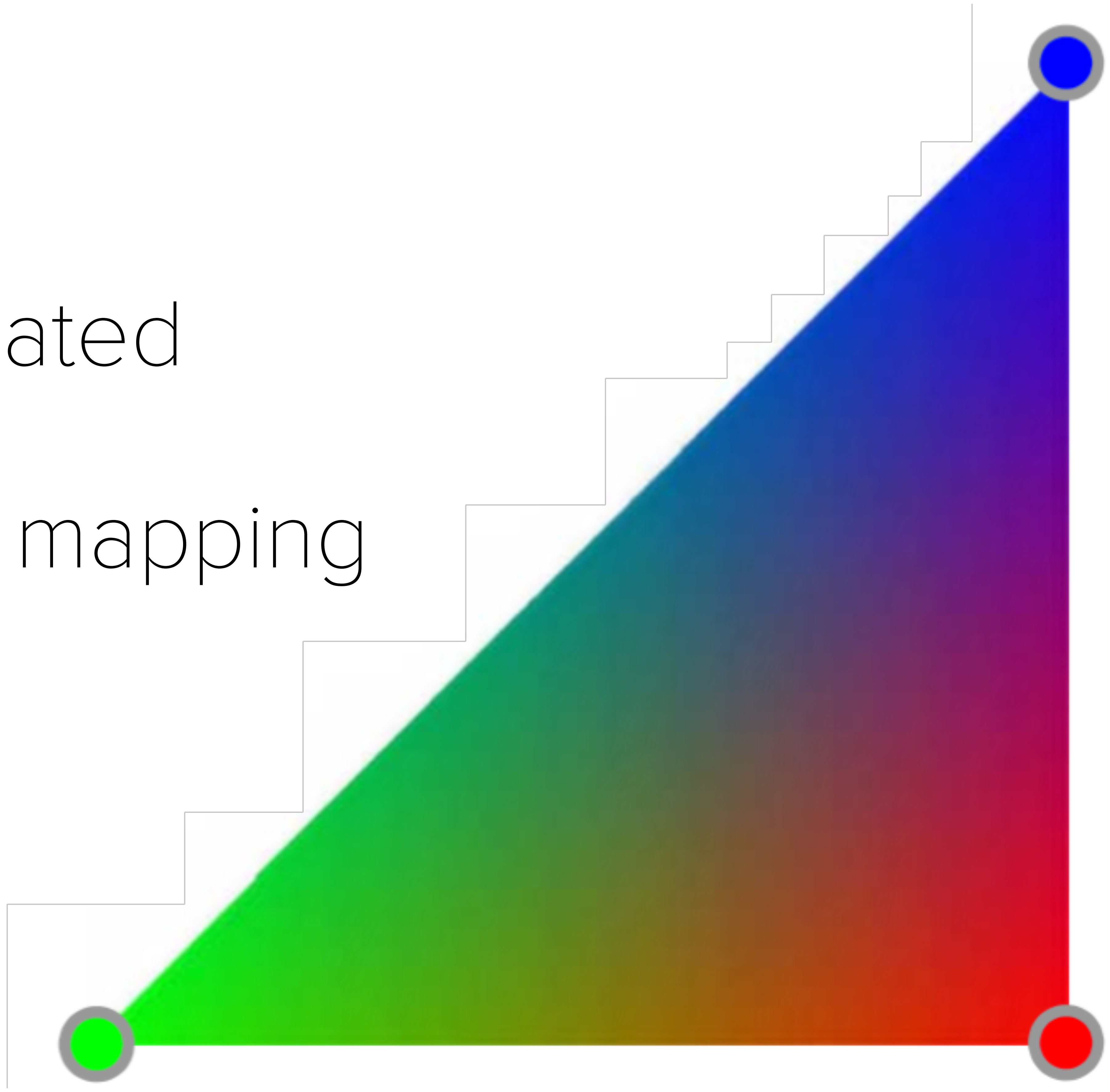
Efficient

No need to store plane equation

Barycentric coordinates are calculated

- Useful for interpolation, texture mapping





BARYCENTRIC INTERPOLATION

Values v_1, v_2, v_3 defined at $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$

Colors, normals, texture coordinates, etc

$$P(\alpha, \beta, \gamma) = \alpha \boldsymbol{a} + \beta \boldsymbol{b} + \gamma \boldsymbol{c}$$

$$v(\alpha, \beta, \gamma) = \alpha v_1 + \beta v_2 + \gamma v_3$$

$$v(1, 0, 0) = v_1, etc$$

Once you know α, β, γ you can interpolate values using the same weights!

Mathematical Toolbox

Ray-triangle intersection

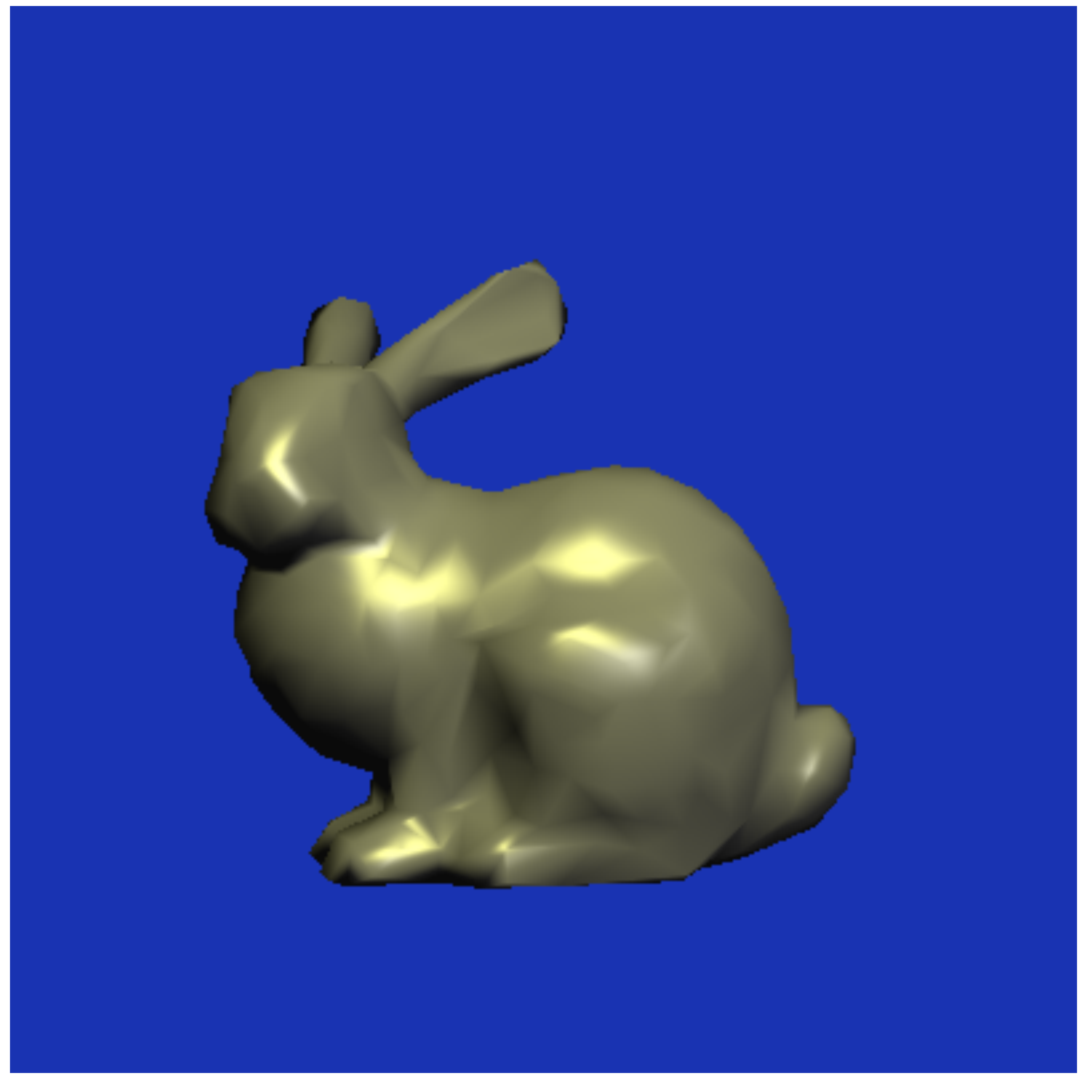
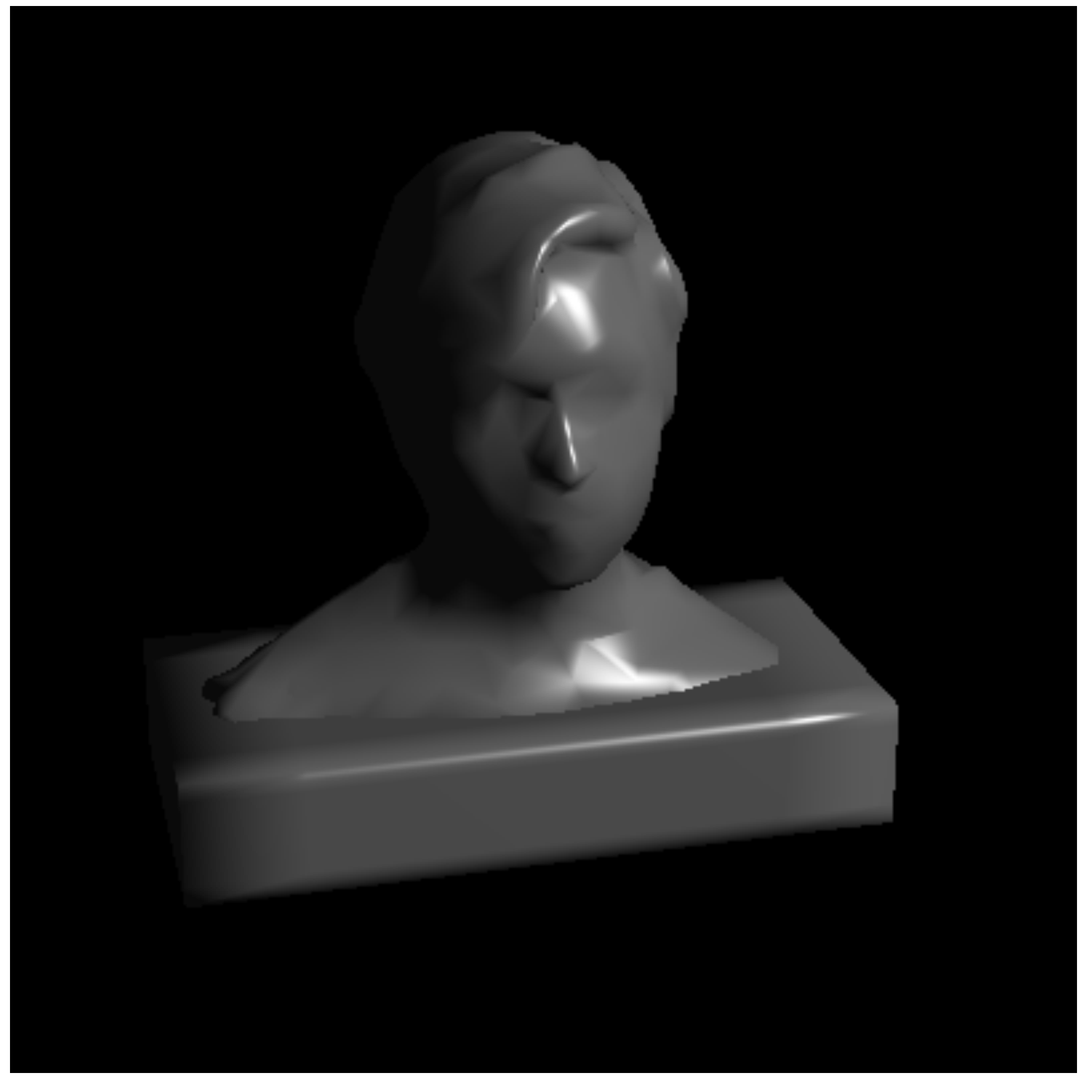
Next week

Recap of transforms

Mob programming (shading & transforms)

TEXTURES

wrapping images around meshes

DUE NEXT SESSION

ray-triangle intersections

Mathematical Toolbox

Ray-triangle intersection

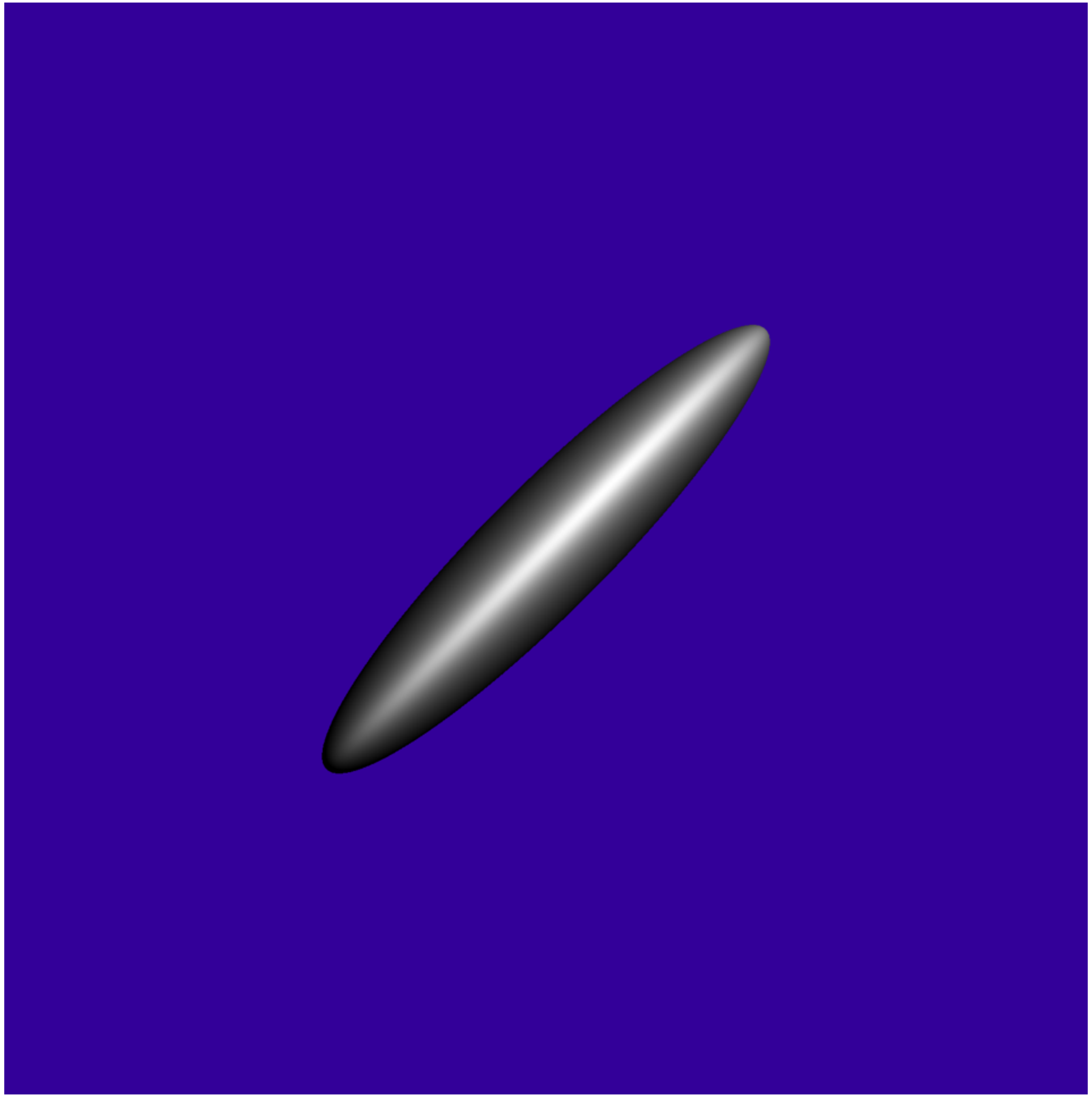
Next week

Recap of transforms

Mob programming (shading & transforms)

TRANSFORMS

Manipulate primitives with transformation matrices



TRANSFORMING RAYS

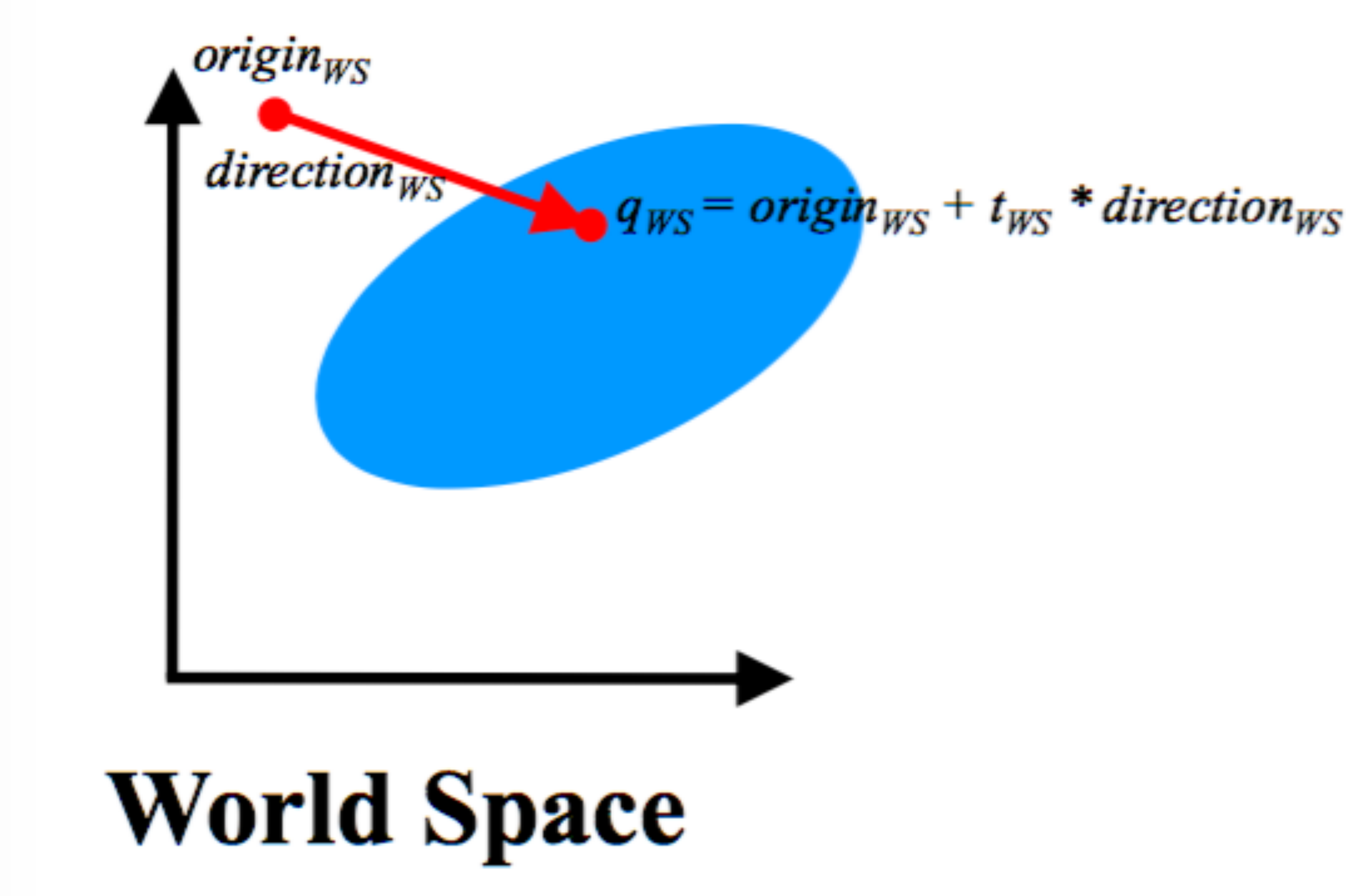
Transform origin

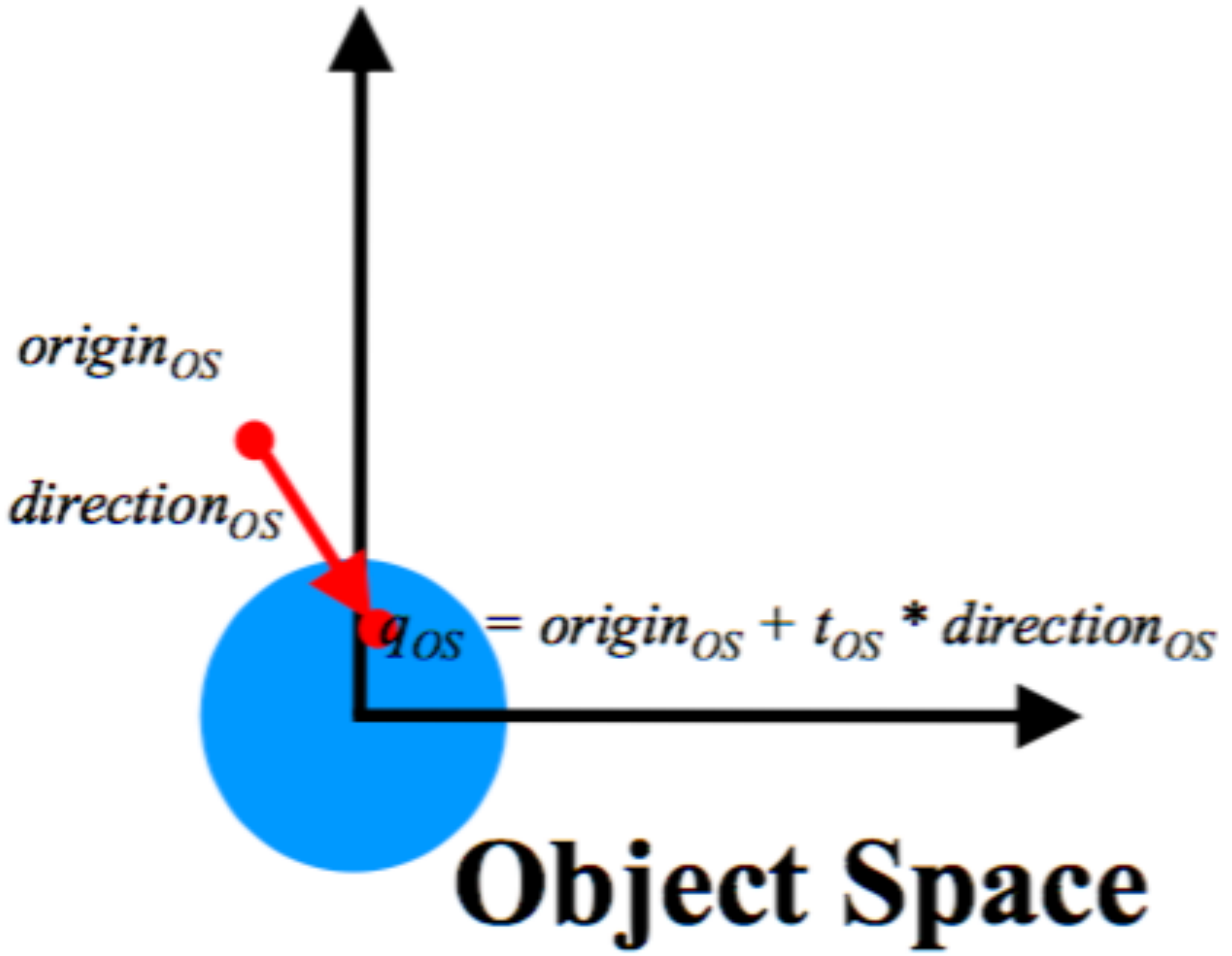
$$origin_{OS} = \mathbf{M}^{-1}\ origin_{WS}$$

Transform direction

$$direction_{OS} = \mathbf{M}^{-1}\ (origin_{WS} + 1 * direction_{WS}) \ - \ \mathbf{M}^{-1}\ origin_{WS}$$

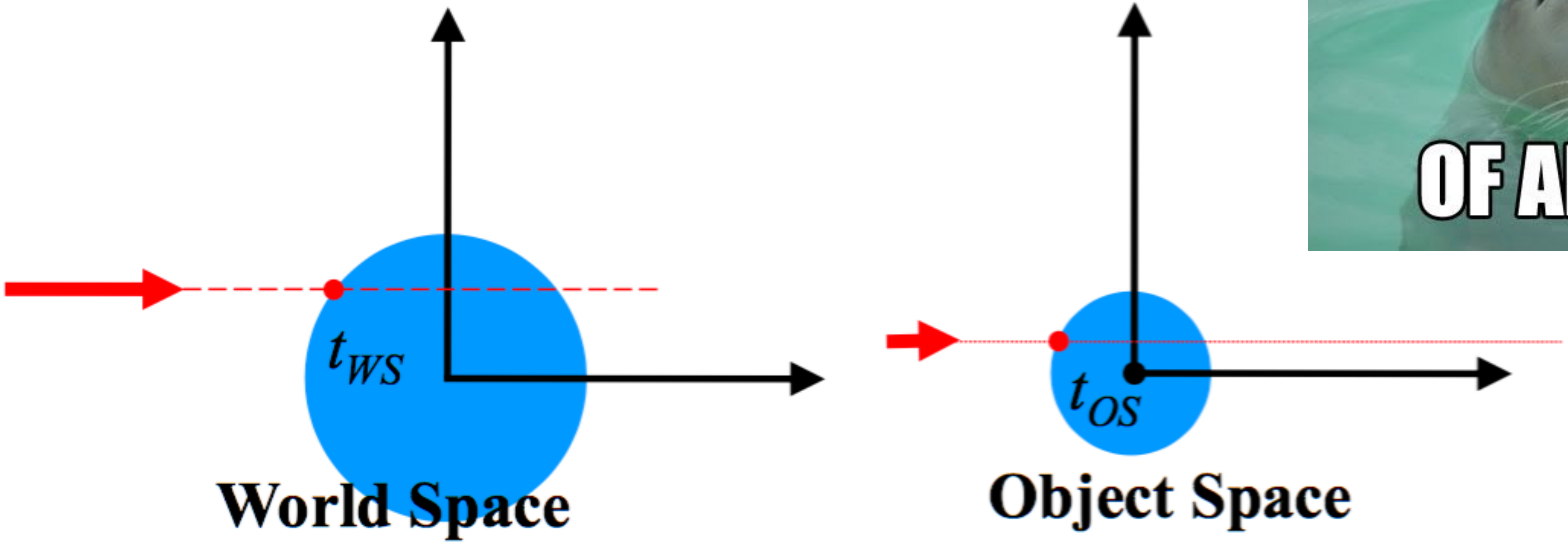
$$direction_{OS} = \mathbf{M}^{-1}\ direction_{WS}$$





DO NOT NORMALIZE DIRECTION

$t_{os} = t_{ws}$ but you cannot rely on t_{os} being the true distance in your intersection equations



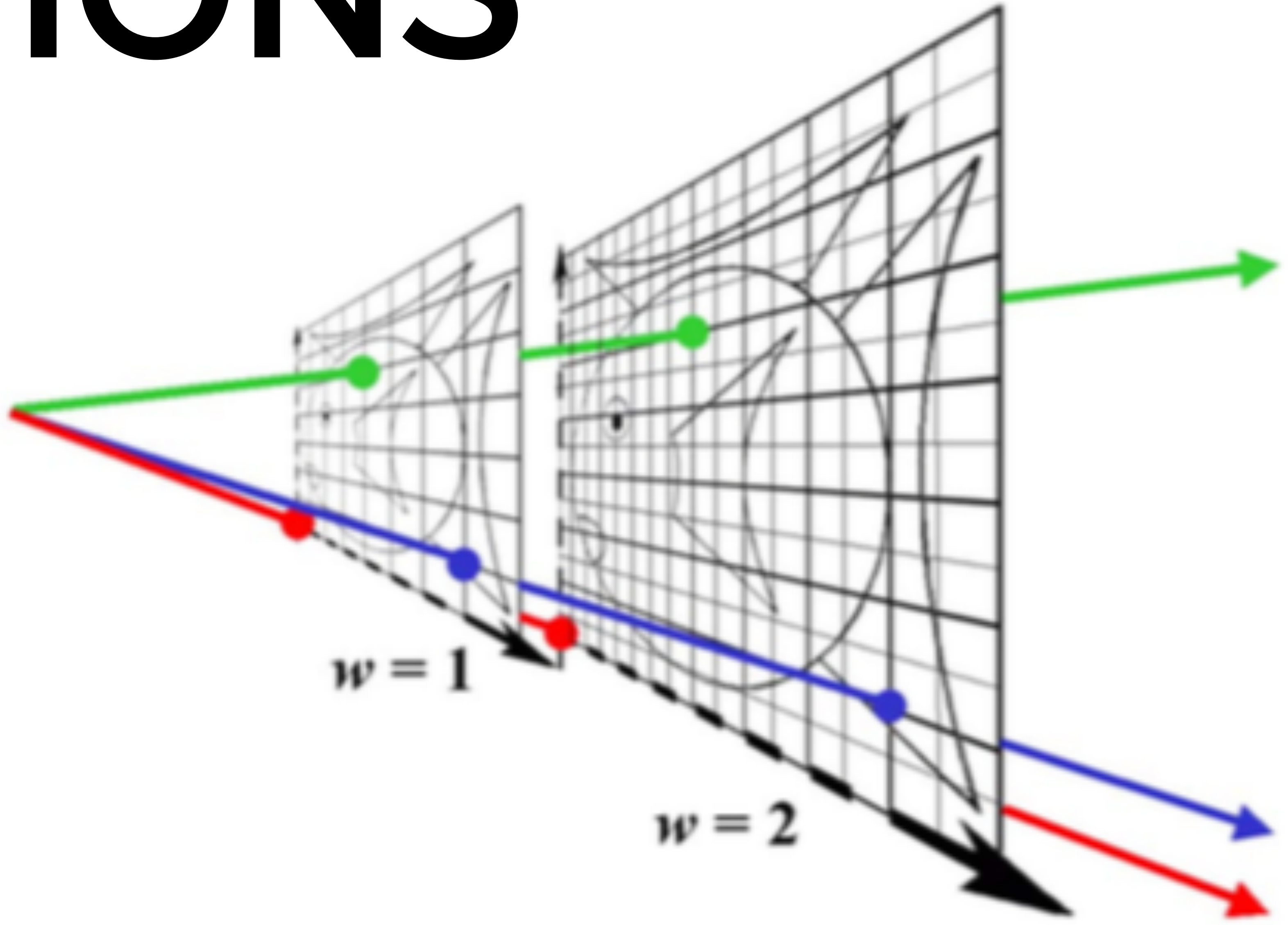
TRANSFORMS ON POINTS VS DIRECTIONS

Transform point

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} ax+by+cz+d \\ ex+fy+gz+h \\ ix+jy+kz+l \\ 1 \end{pmatrix}$$

Transform direction

$$\begin{pmatrix} x' \\ y' \\ z' \\ 0 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix} = \begin{pmatrix} ax+by+cz \\ ex+fy+gz \\ ix+jy+kz \\ 0 \end{pmatrix}$$



Homogeneous coordinates (x, y, z, w)

w = 0 is a point at infinity (direction)

We'll apply all transforms in 4D, logic for point and direction is already written for you. See MathHelper.swift!

TRANSFORM TANGENT VECTOR

v is perpendicular to normal n:

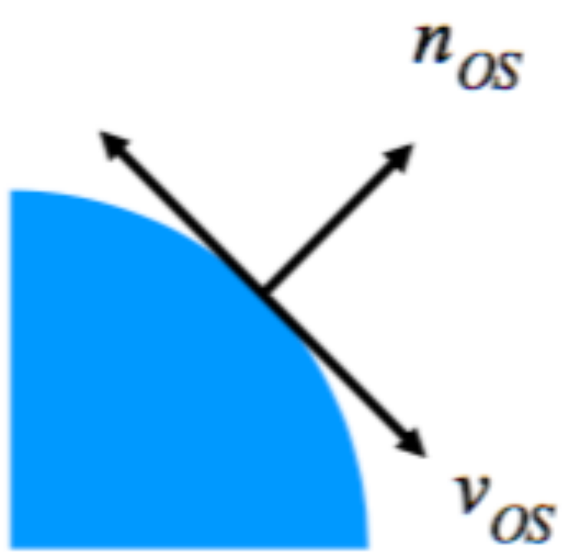
Dot product

$$n_{OS}^{\mathbf{T}}\, v_{OS} = 0$$

$$n_{OS}^{\mathbf{T}}\ (\mathbf{M}^{-1}\ \mathbf{M})\ v_{OS} = 0$$

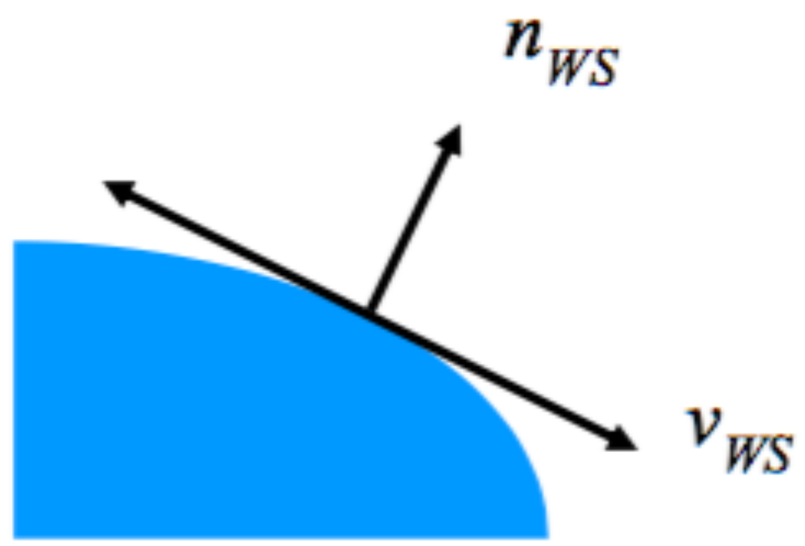
$$(n_{OS}^{\mathbf{T}}\ \mathbf{M}^{-1})\ (\mathbf{M}\ \ v_{OS}) = 0$$

$$(n_{OS}^{\mathbf{T}}\ \mathbf{M}^{-1})\ v_{WS} = 0$$



v_{WS} is perpendicular to normal n_{WS}:

$$n_{WS}^{\mathbf{T}}\, v_{WS} = 0$$



$$n_{WS}^{\mathbf{T}} = n_{OS}^{\mathbf{T}}\, (\mathbf{M}^{-1})$$

$$\boxed{n_{WS} = (\mathbf{M}^{-1})^{\mathbf{T}}\, n_{OS}}$$

Mathematical Toolbox

Ray-triangle intersection

Next week

Recap of transforms

Mob programming (shading & transforms)