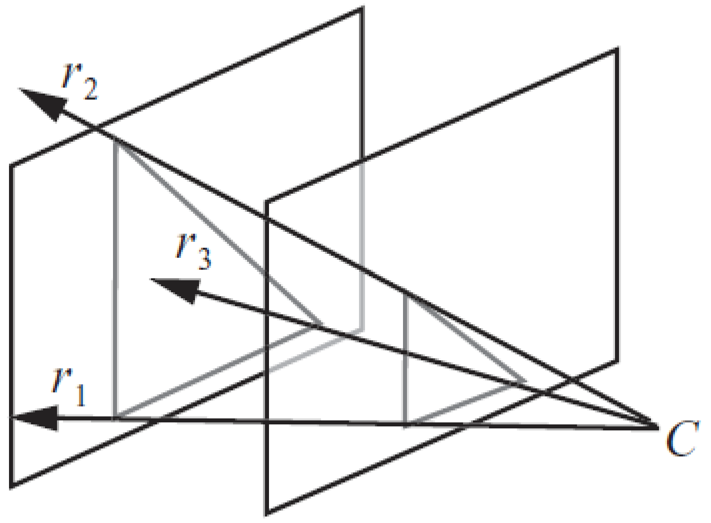
GLC (Briefly)

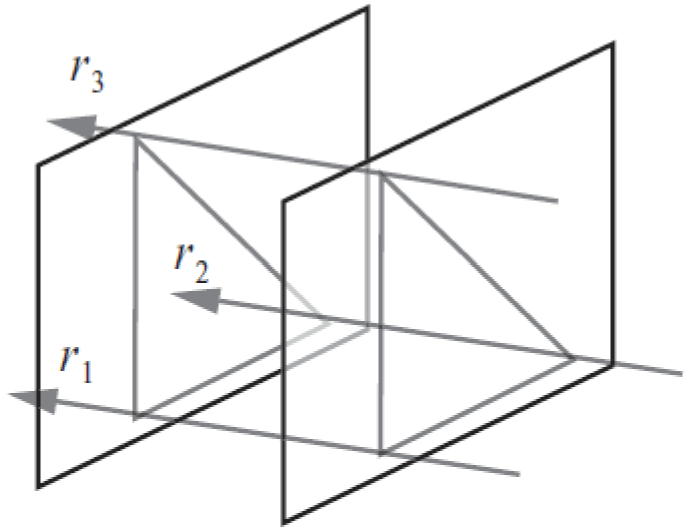
CS535

Daniel G. Aliaga
Department of Computer Science
Purdue University

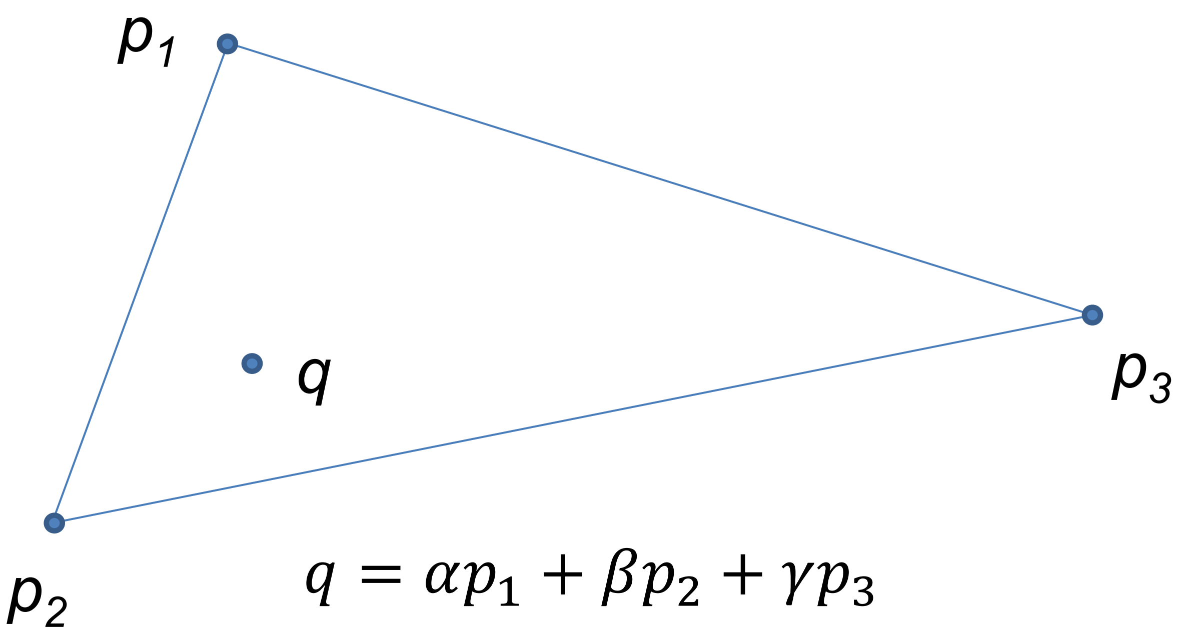
General Linear Cameras

- General Linear Camera (GLC) model that unifies many previous camera models into a single representation.
- GLC model is both general and linear in the sense that, given any vector space where rays are represented as points, it describes all 2D affine subspaces (planes) that can be formed by affine combinations of 3 rays.





Barycentric coordinates

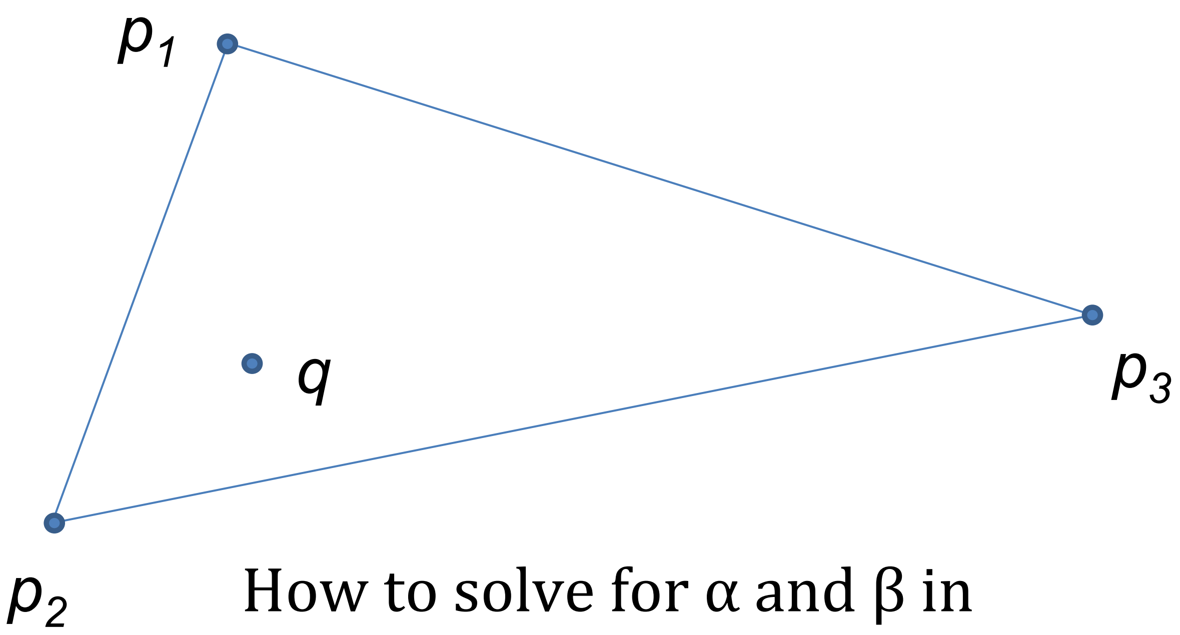


$$q = \alpha p_1 + \beta p_2 + \gamma p_3$$

If $[\alpha + \beta + \gamma = 1 \;\; and \;\; \{\alpha, \beta, \gamma\} \geq 0]$,
then q inside triangle (p_1, p_2, p_3)

Can also write:
$q = \alpha p_1 + \beta p_2 + (1 - \alpha - \beta) p_3$

Barycentric coordinates



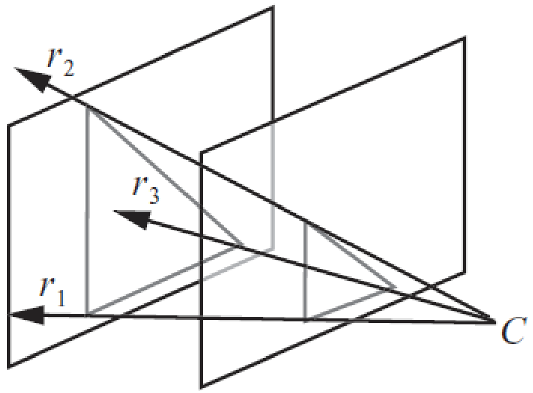
How to solve for α and β in
$q = \alpha p_1 + \beta p_2 + (1 - \alpha - \beta) p_3$?

Two equations, two unknowns:
use 2x2 matrix inversion…

GLC Ray Combination

- (s_i, t_i, u_i, v_i) defines a two-plane parameterization
  - i.e., $(s_i, t_i, 0)$ and $(u_i, v_i, 1)$
- r_i defined by (s_i, t_i, u_i, v_i)
- Using 3 rays, r is an "interpolated ray":

$$r = \alpha \cdot (s_1, t_1, u_1, v_1) + \beta \cdot (s_2, t_2, u_2, v_2) + (1 - \alpha - \beta) \cdot (s_3, t_3, u_3, v_3) \tag{1}$$



Cameras

- Perspective Camera:
  - $(s, t, 0)$ reduces to a point (e.g., $s = t = 0$)
- Orthographic Camera:
  - The 3 rays are parallel…
- Another Camera;
  - You pick one…

Rendering Pipeline

- Algorithm:
  - Define a scene (e.g., colored triangles)
  - Define rays
  - Cast rays into the scene
  - Intersect with triangles (e.g., Barycentric coords)
  - Fill-out framebuffer with color upon intersection
- OpenGL use:
  - Minimally used
  - Just draw “framebuffer” using glTexSubImage2D, or glBlitFrameBuffer