CS451
Ray Casting

Jyh-Ming Lien

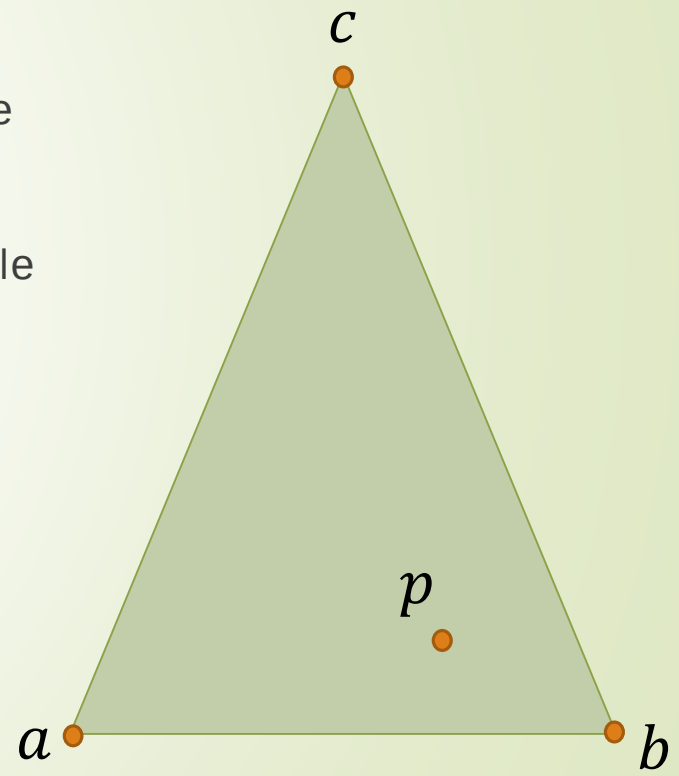
Department of Computer SCience

George Mason University

Based on notes from http://stellar.mit.edu/

Barycentric Coordinates

- Barycentric coordinate
  - Representing the point $\boldsymbol{p}$ using the vertices of a triangle
  - $\boldsymbol{p} = \alpha\boldsymbol{a} + \beta\boldsymbol{b} + \gamma\boldsymbol{c}$ and $\alpha + \beta + \gamma = 1$
  - If we require $1 \geq \alpha, \beta, \gamma \geq 0$ then $\boldsymbol{p}$ spans over the triangle
    - $\alpha = \frac{\Delta_{pbc}}{\Delta_{abc}}, \beta = \frac{\Delta_{pca}}{\Delta_{abc}}, \gamma = \frac{\Delta_{pab}}{\Delta_{abc}}$



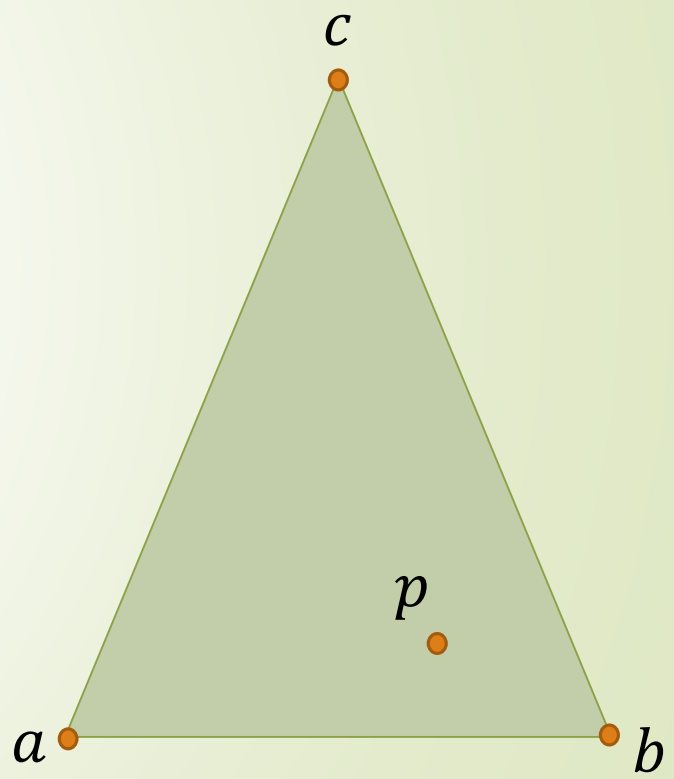
Barycentric Coordinates

- Applications: Deformation

Complex Barycentric Coordinates with Applications to Planar Shape Deformation, Ofir Weber et al.

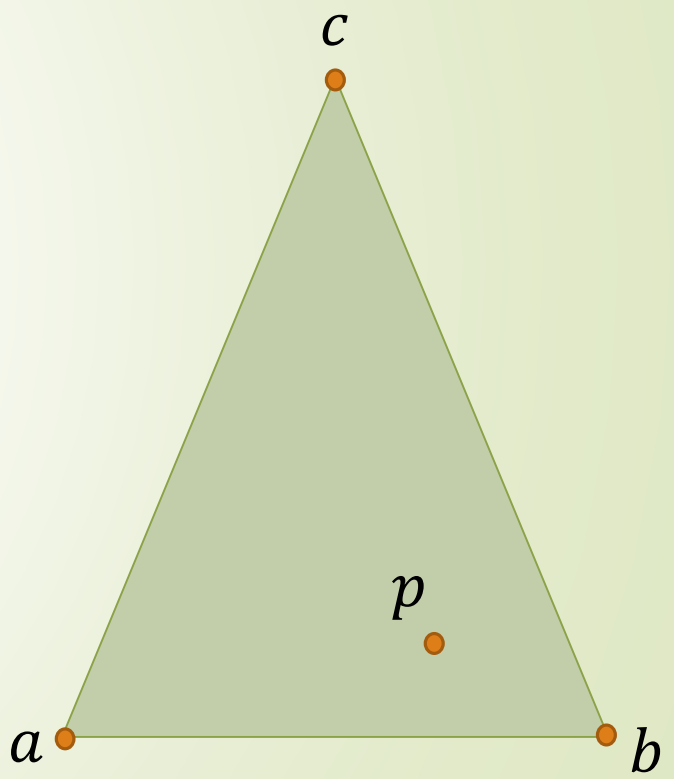
Ray-Triangle Barycentric Intersection

- Barycentric coordinate in a triangle
  - Two degrees of freedom (2D coordinates)
  - $\boldsymbol{p} = \alpha\boldsymbol{a} + \beta\boldsymbol{b} + \gamma\boldsymbol{c} = \boldsymbol{a} + \beta(\boldsymbol{b} - \boldsymbol{a}) + \gamma(\boldsymbol{c} - \boldsymbol{a})$
  - Solve $\boldsymbol{a} + \beta(\boldsymbol{b} - \boldsymbol{a}) + \gamma(\boldsymbol{c} - \boldsymbol{a}) - \boldsymbol{p} = \boldsymbol{0}$ to get β and γ
    - How?



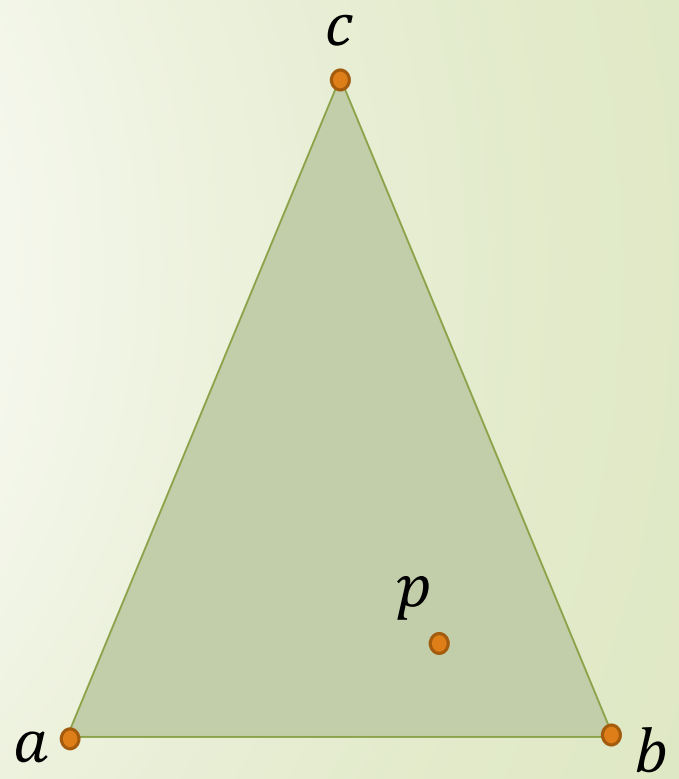
Ray-Triangle Barycentric Intersection

- Barycentric coordinate intersection
  - $R_o + tR_d = a + \beta(\boldsymbol{b} - \boldsymbol{a}) + \gamma(\boldsymbol{c} - \boldsymbol{a})$
  - 3 unknowns, 3 equations!



Benefit of Performing intersections using Barycentric coordinate

- No plane equation is required
- Use Barycentric coordinate for interpolations
  - Color
  - Texture
  - Normal direction
  - Etc…



Midterm Exam on Nov 4th

- From all slides on and before Oct 16 (i.e., no Ray tracing stuff)
- Textbook: Chapters 2, 3, 4, 5.1, 5.2, 5.3, 5.5, 6
- Important topics:
  - Graphics pipelines, vertex shader, pixel/fragment shader
  - Transforms
    - Projection, rotation (rotation matrix, quaternions), scaling, shearing, etc
    - many different coordinate spaces
      - Homogenous coordinates
      - Camera/eye space, world space, object space
      - UV texture space
      - Bernstein basis function
      - bone subspace
      - Tangent space
  - Deformation (Free-form and skeleton deformation)
  - Texturing (texture coordinates, projection functions, texture pipeline, interpolation, mipmaps, environmental mapping, cube mapping)
  - Bump mapping (normal mapping, normal mapping in tangent space, parallax mapping, reflief mapping)