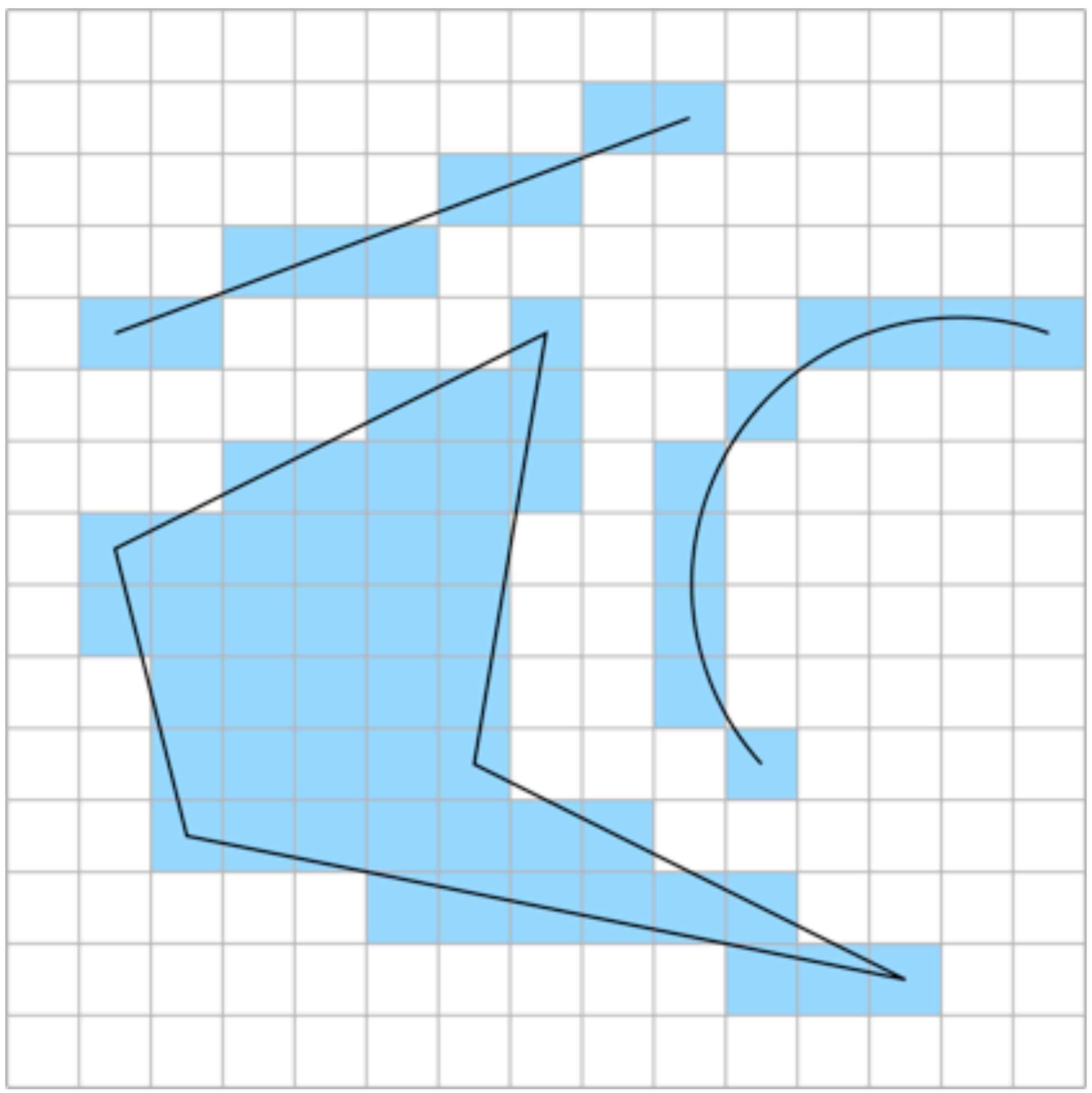
Rasterization

Prof. Vladlen Koltun
Computer Science Department
Stanford University

Rasterization



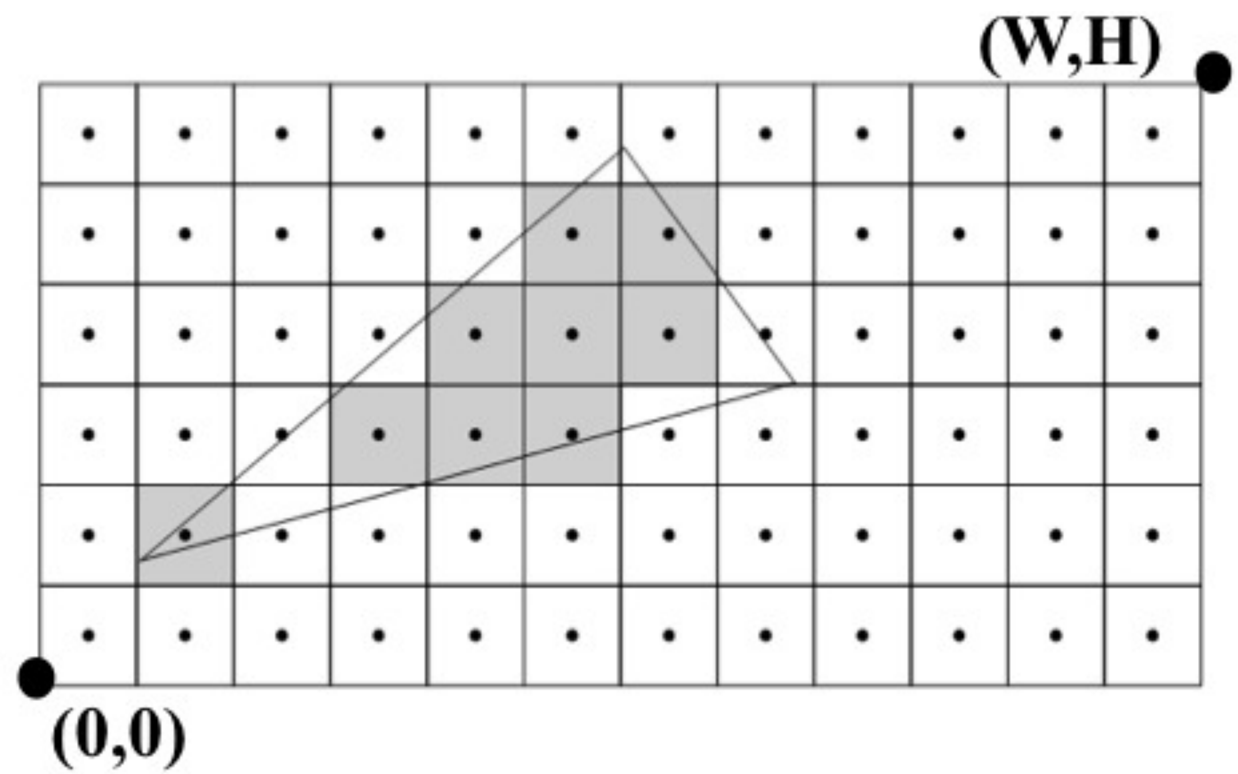
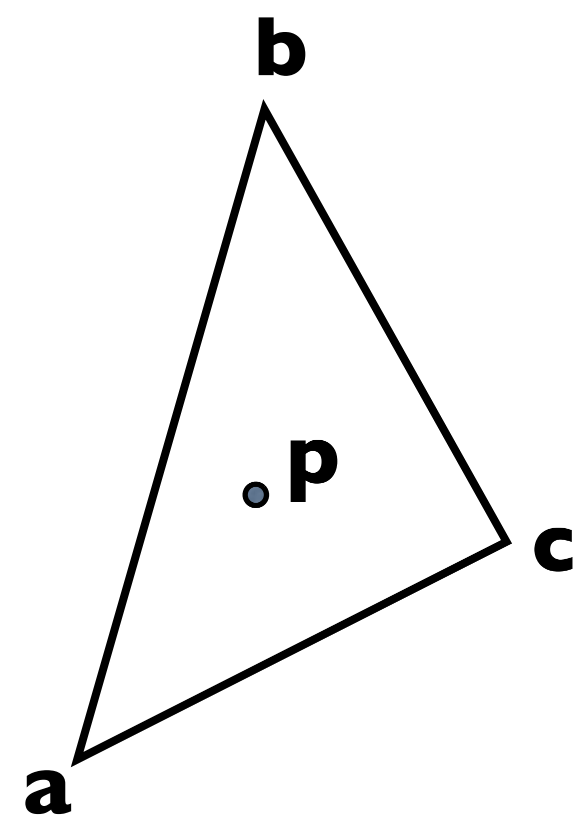
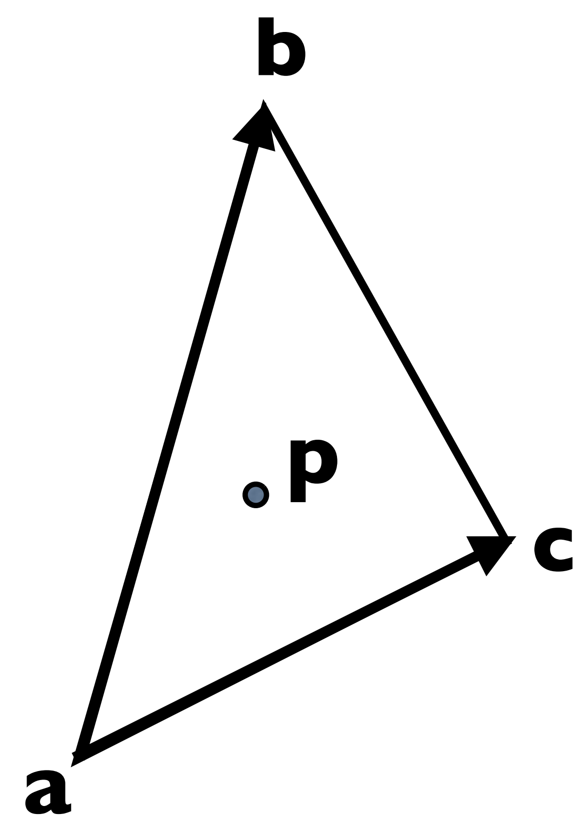


Image from Wikimedia Commons, author Wojciech Mula

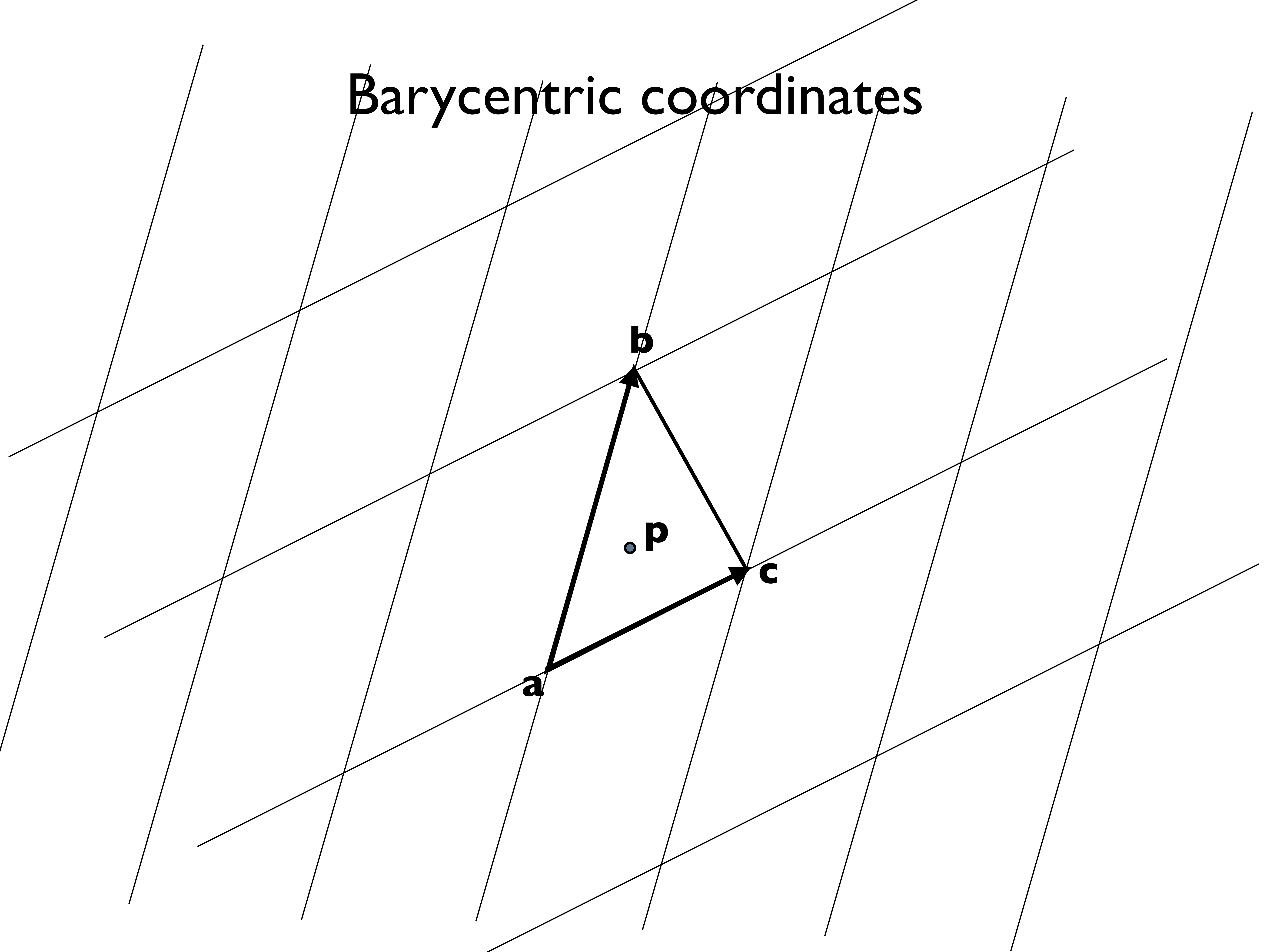
Barycentric coordinates



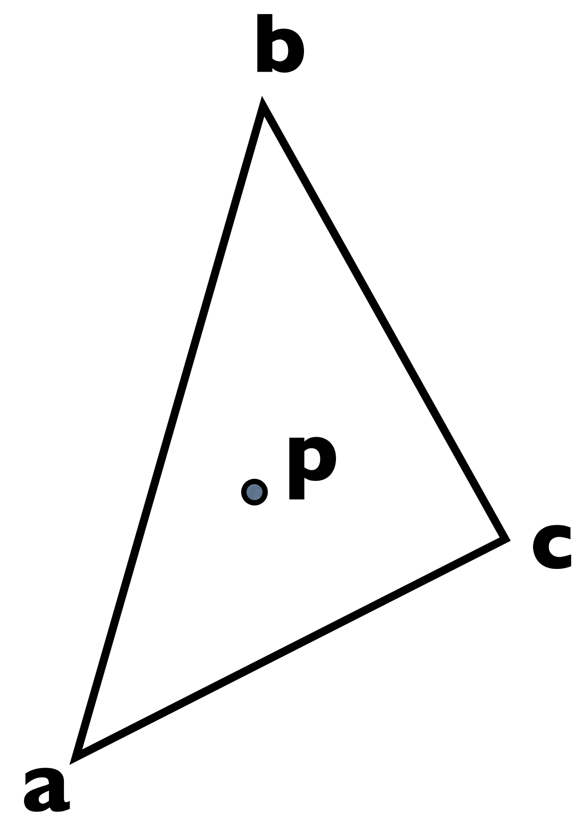
Barycentric coordinates



Barycentric coordinates

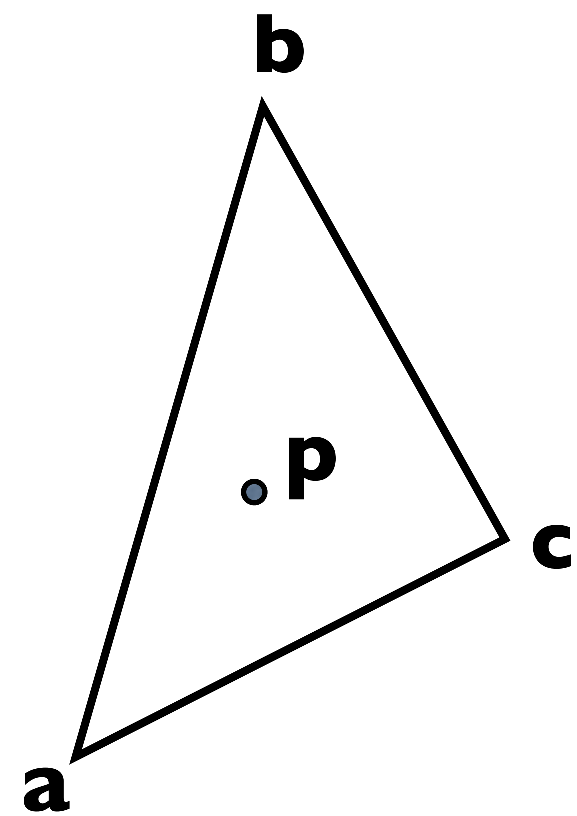


Barycentric coordinates



$$\mathbf{p} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

Barycentric coordinates

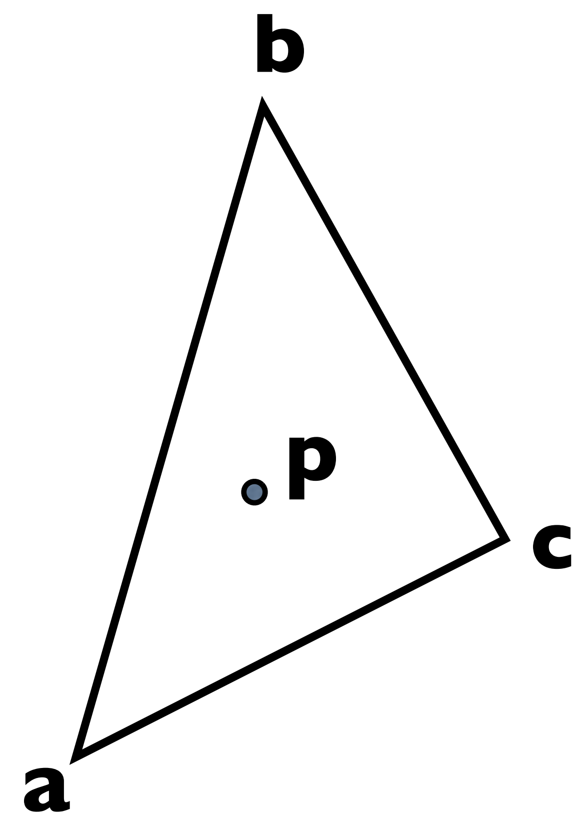


$$\mathbf{p} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

or

$$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} \quad \text{such that} \quad \alpha + \beta + \gamma = 1$$

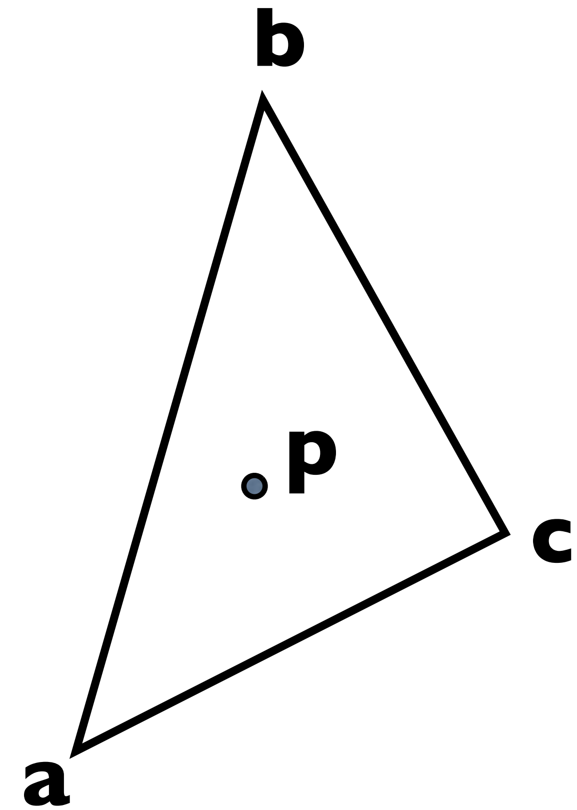
Barycentric coordinates



$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$ such that $\alpha + \beta + \gamma = 1$

p is inside the triangle if and only if $0 \leq \alpha, \beta, \gamma \leq 1$

Barycentric coordinates

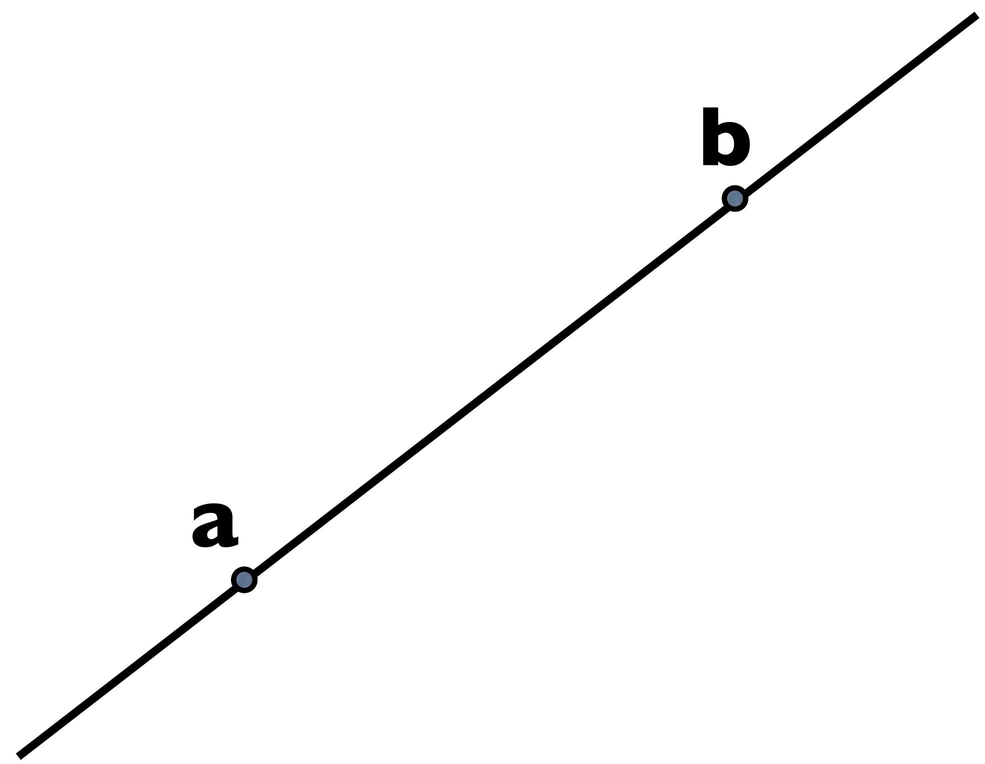


$$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} \quad \text{such that} \quad \alpha + \beta + \gamma = 1$$

p is inside the triangle if and only if $0 \leq \alpha, \beta, \gamma \leq 1$

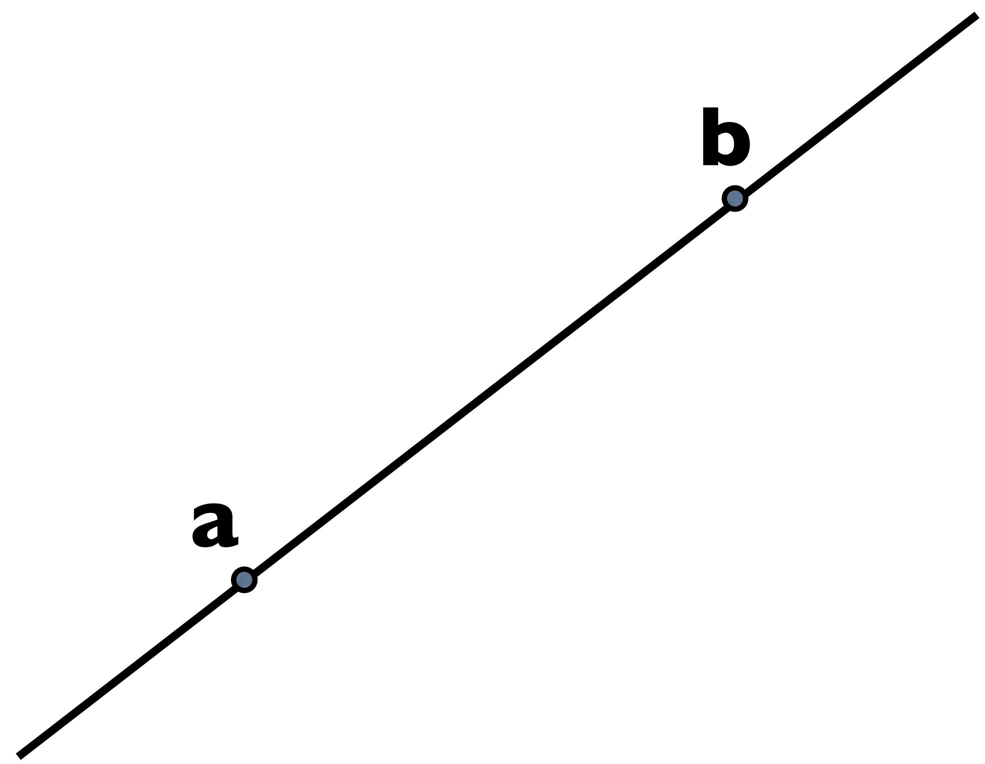
or, equivalently, $\alpha, \beta, \gamma \geq 0$

Computing barycentric coordinates



$$y = \frac{y_b - y_a}{x_b - x_a}(x - x_a) + y_a$$

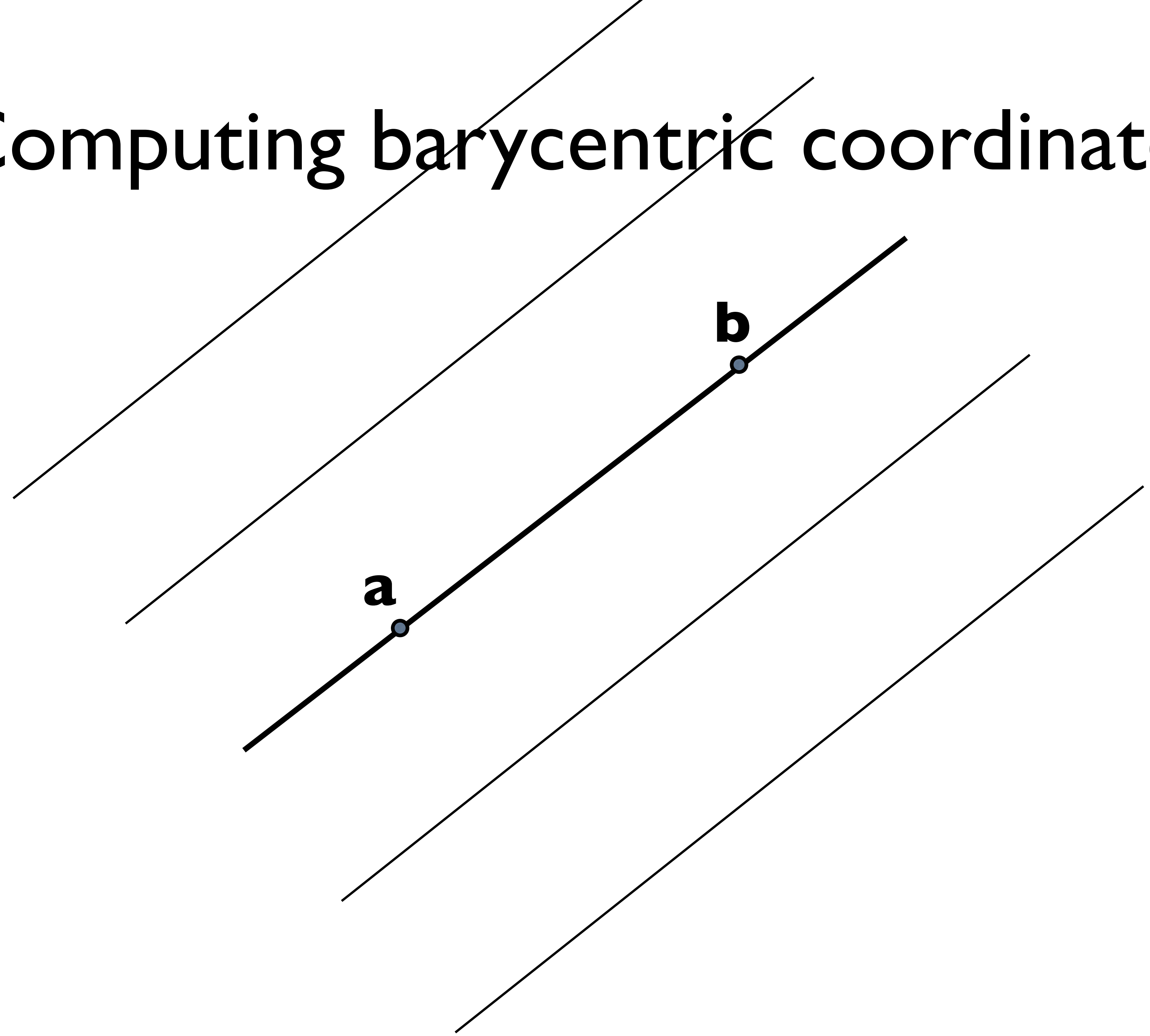
Computing barycentric coordinates



$$y = \frac{y_b - y_a}{x_b - x_a}(x - x_a) + y_a$$

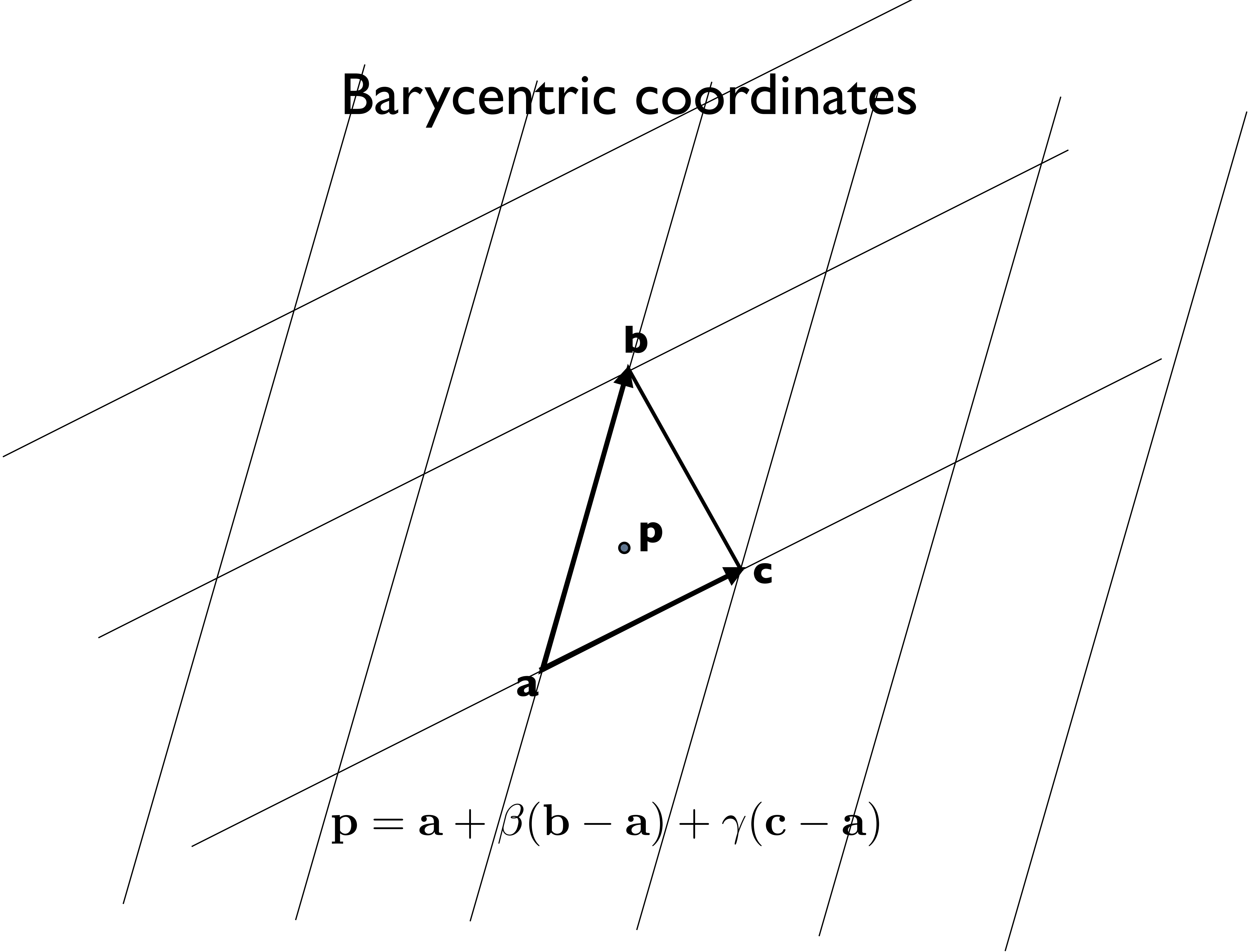
$$f_{ab}(x, y) \equiv (y_a - y_b)x + (x_b - x_a)y + x_a y_b - x_b y_a = 0$$

Computing barycentric coordinates



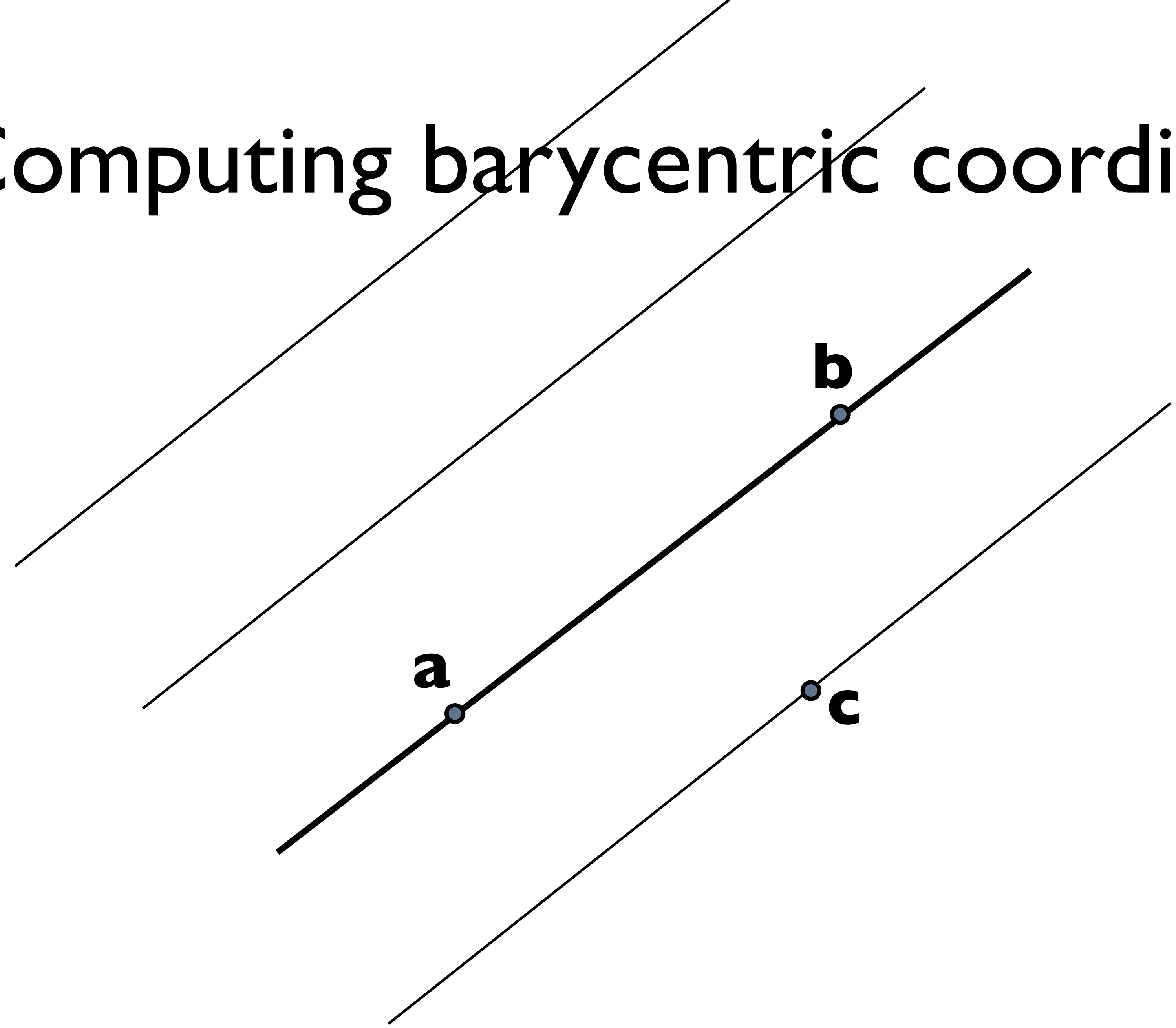
$$f_{ab}(x,y) \equiv (y_a - y_b)x + (x_b - x_a)y + x_a y_b - x_b y_a = 0$$

Barycentric coordinates



$$\mathbf{p} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

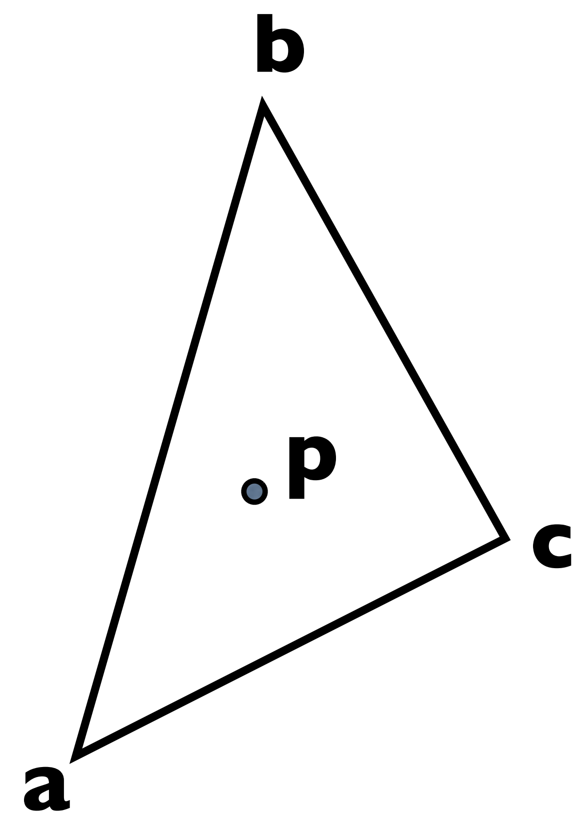
Computing barycentric coordinates



$$\gamma = \frac{f_{ab}(x, y)}{f_{ab}(x_c, y_c)}$$

$$= \frac{(y_a - y_b)x + (x_b - x_a)y + x_a y_b - x_b y_a}{(y_a - y_b)x_c + (x_b - x_a)y_c + x_a y_b - x_b y_a}$$

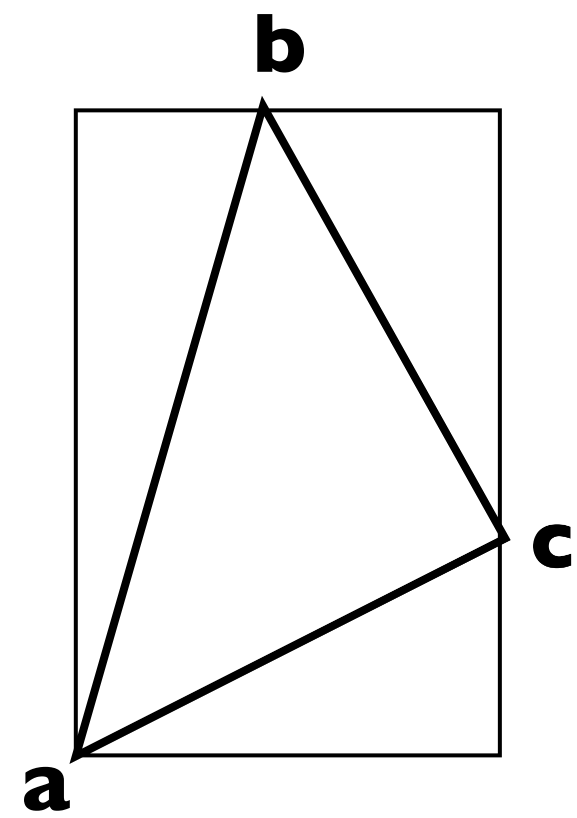
Barycentric coordinates



Can compute γ and β in this way, and α as

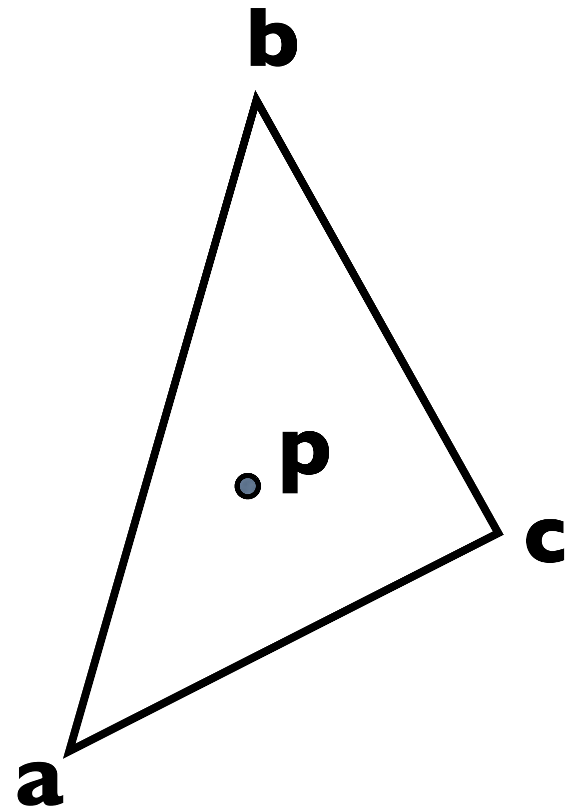
$$\alpha = 1 - \beta - \gamma$$

Overall algorithm



Iterate over bounding rectangle.
For each pixel:
- compute barycentric coordinates for pixel center.
- if they are nonnegative, compute interpolated values and store with generated fragment.

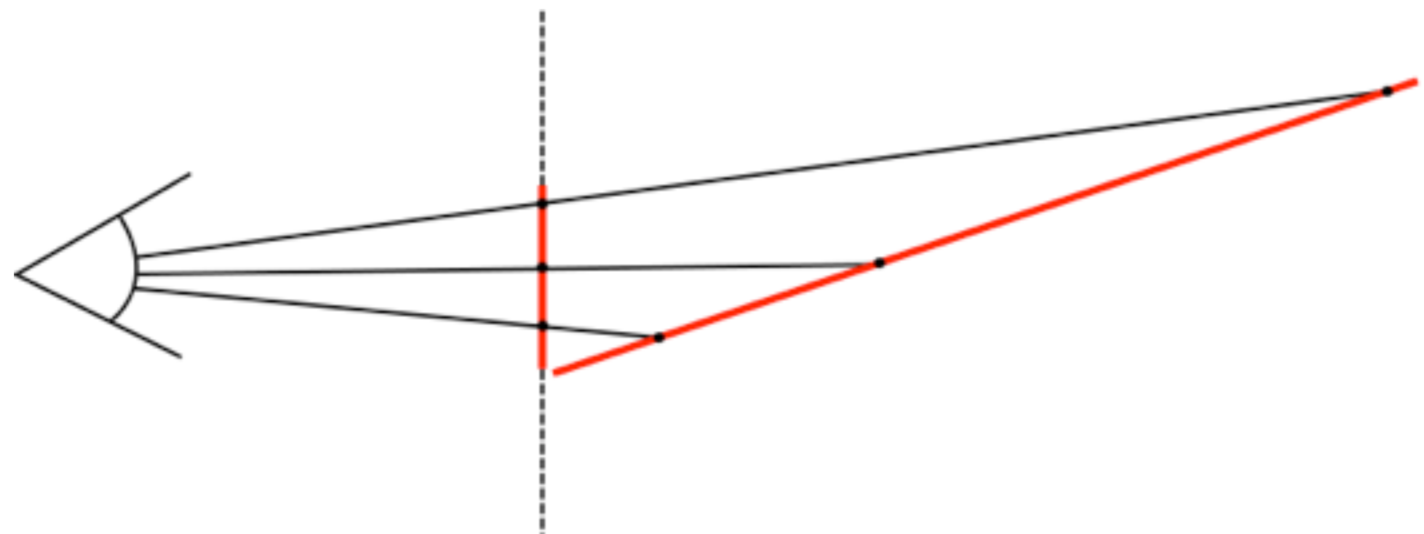
Barycentric coordinates



Can use barycentric coordinates to linearly interpolate values associated with vertices, such as color:

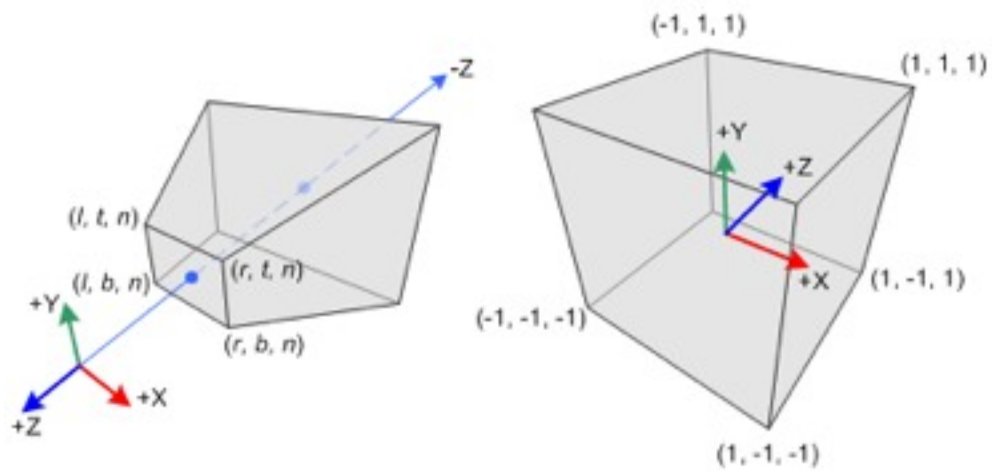
$$v_{\mathbf{p}} = \alpha v_{\mathbf{a}} + \beta v_{\mathbf{b}} + \gamma v_{\mathbf{c}}$$

Perspective-correct interpolation



- Linear interpolation in screen space does not correspond to linear interpolation across the original primitive. Can result is major visual artifacts.

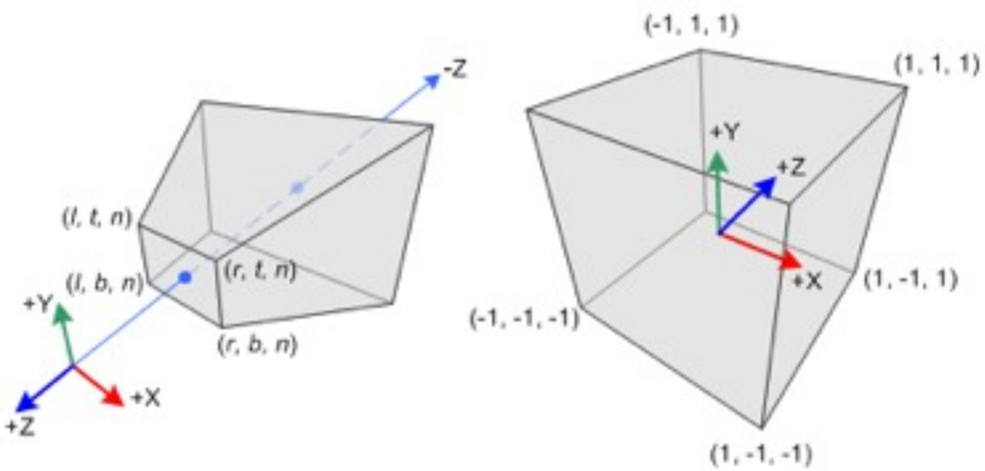
Perspective-correct interpolation



$$M = \begin{pmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- The problem is not the projection transform itself, since the transform is linear in homogenous coordinates.
- The problem is the conversion from clip coordinates to Normalized Device Coordinated, which involves a division by w. This division breaks linearity.

Perspective-correct interpolation



$$M = \begin{pmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- The problem is not the projection transform itself, since the transform is linear in homogenous coordinates.
- The problem is the conversion from clip coordinates to Normalized Device Coordinated, which involves a division by w. This division breaks linearity.
- If we denote a point in clip coordinates by $\hat{a} = (a_x, a_y, a_z, a_w)$, the corresponding point in Cartesian coordinates is $a = \left(\frac{a_x}{a_w}, \frac{a_y}{a_w}, \frac{a_z}{a_w}\right)$

Perspective-correct interpolation

- Consider a linear attribute $\hat{f}$ defined over the homogeneous space and the corresponding (non-linear) attribute f defined over the Cartesian space.

- We wish to compute f(p), where

$$
\begin{aligned}
\mathbf{p} &= \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} \\
&= \alpha\left(\frac{a_x}{a_w}, \frac{a_y}{a_w}, \frac{a_z}{a_w}\right) + \beta\left(\frac{b_x}{b_w}, \frac{b_y}{b_w}, \frac{b_z}{b_w}\right) + \gamma\left(\frac{c_x}{c_w}, \frac{c_y}{c_w}, \frac{c_z}{c_w}\right) \\
&= \frac{\alpha}{a_w}(a_x, a_y, a_z) + \frac{\beta}{b_w}(b_x, b_y, b_z) + \frac{\gamma}{c_w}(c_x, c_y, c_z) \\
&= \frac{\alpha}{a_w}L\hat{a} + \frac{\beta}{b_w}L\hat{b} + \frac{\gamma}{c_w}L\hat{c}
\end{aligned}
$$

where L is a linear mapping

Perspective-correct interpolation

- Since $\hat{f}$ is a linear function, L is a linear mapping, and p is a linear combination of a, b, and c with coefficients α, β, γ,

$$
\begin{aligned}
f(\mathbf{p}) = \hat{f}(\hat{\mathbf{p}}) &= \frac{\alpha}{a_w}\hat{f}(\hat{\mathbf{a}}) + \frac{\beta}{b_w}\hat{f}(\hat{\mathbf{b}}) + \frac{\gamma}{c_w}\hat{f}(\hat{\mathbf{c}}) \\
&= \frac{\alpha}{a_w}f(\mathbf{a}) + \frac{\beta}{b_w}f(\mathbf{b}) + \frac{\gamma}{c_w}f(\mathbf{c})
\end{aligned}
$$

- Thus a different attribute, $f'(\mathbf{p}) = \dfrac{f(\mathbf{p})}{w(\mathbf{p})}$, is linear in the Cartesian space

- We can interpolate f' and then multiply by w to get f.

Perspective-correct interpolation

- Thus a different attribute, $f'(\mathbf{p}) = \dfrac{f(\mathbf{p})}{w(\mathbf{p})}$, is linear in the Cartesian space

- We can interpolate f' and then multiply by w to get f.

- But how do we interpolate w? If we consider the constant attribute g(p)=1, we can see that g'(p) = 1/w is linear in screen space.

- We can thus interpolate f' and g' and take

$$f(\mathbf{p}) = \frac{f'(\mathbf{p})}{g'(\mathbf{p})}$$