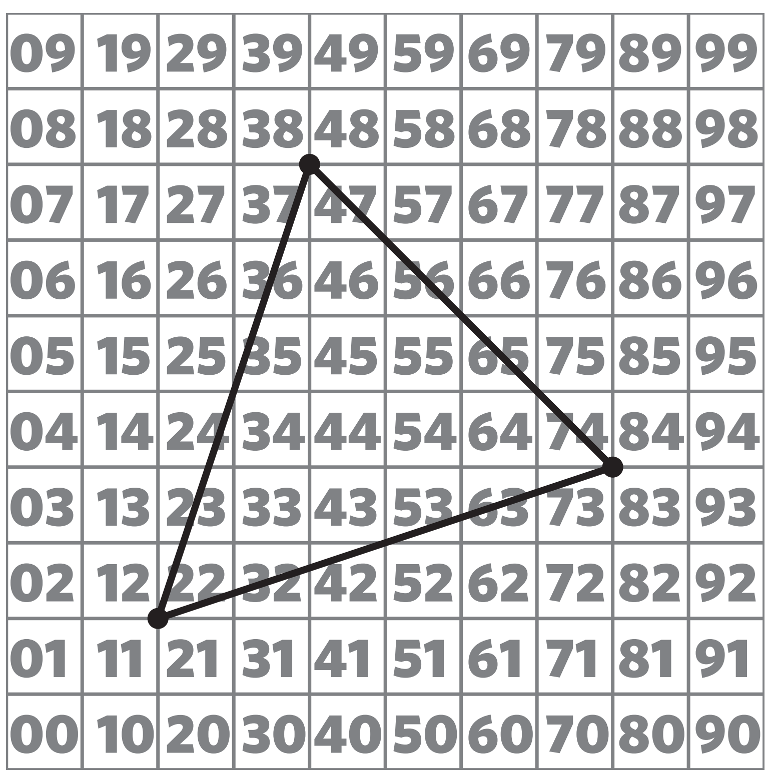
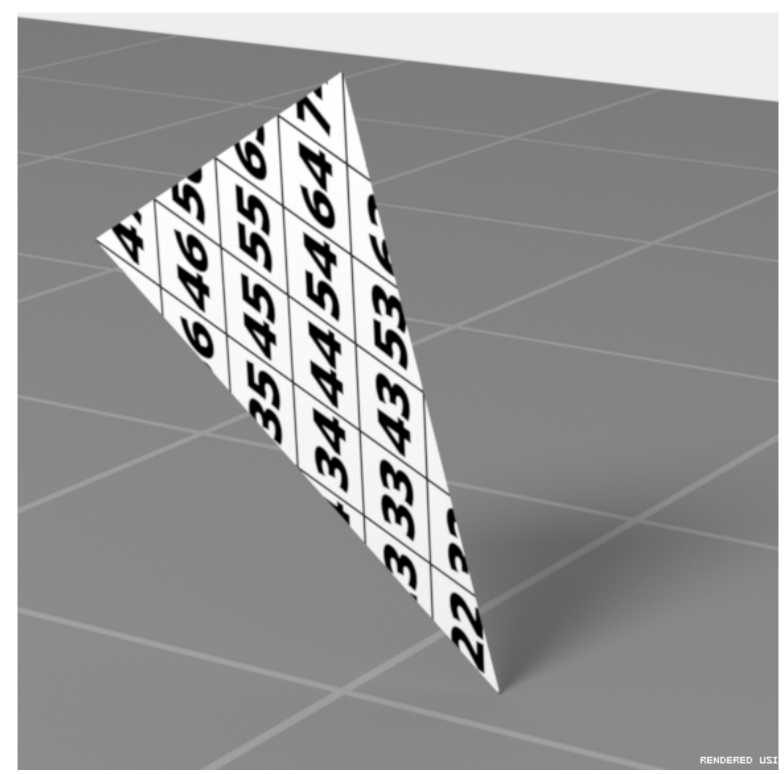
Interpolated values in ray tracing

CS 4620 Lecture 6.5

Texture coordinates on meshes

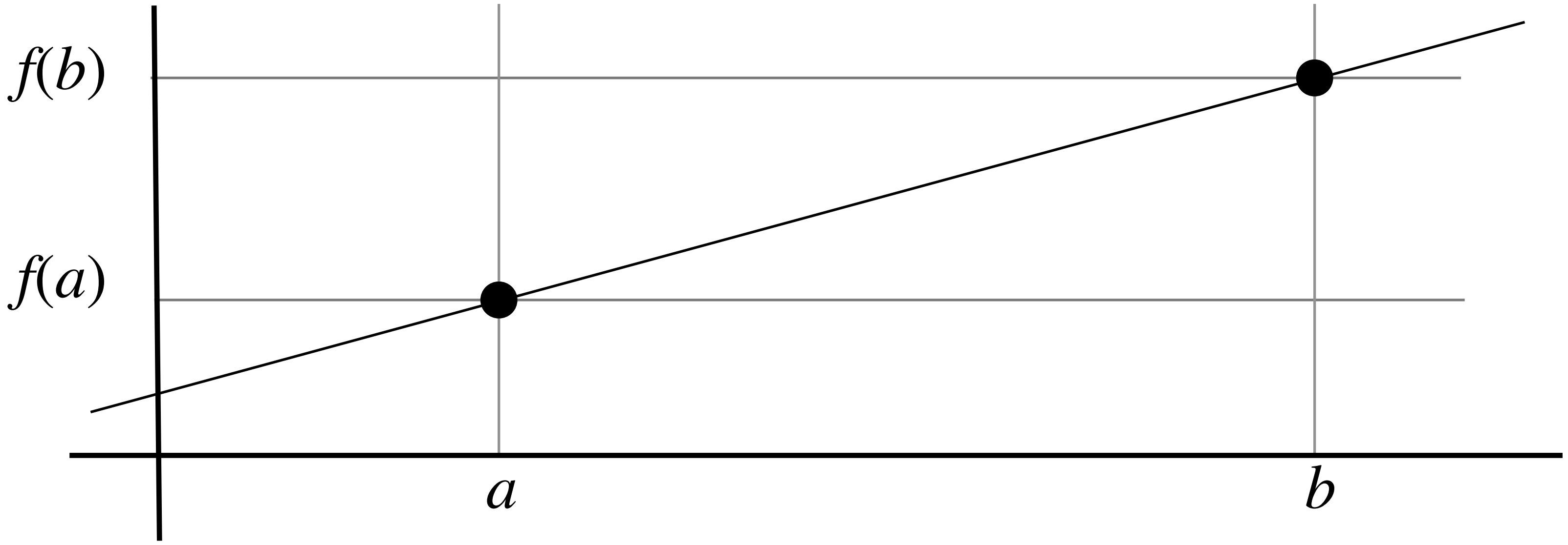
- **Texture coordinates are per-vertex data like vertex positions**
  - can think of them as a second position: each vertex has a position in 3D space and in 2D texture space
- **How to come up with (*u*,*v*)s for points inside triangles?**





Linear interpolation, 1D domain

- **Given values of a function f(x) for two values of x, you can define in-between values by drawing a line**



See textbook Sec. 2.6

- there is a unique line through the two points
- can write down using slopes, intercepts
- …or as a value added to f(a)
- …or as a convex combination of f(a) and f(b)

$$
\begin{aligned}
f(x) &= f(a) + \frac{x-a}{b-a}(f(b) - f(a)) \\
&= (1-\beta)f(a) + \beta f(b) \\
&= \alpha f(a) + \beta f(b)
\end{aligned}
$$

Linear interpolation in 1D

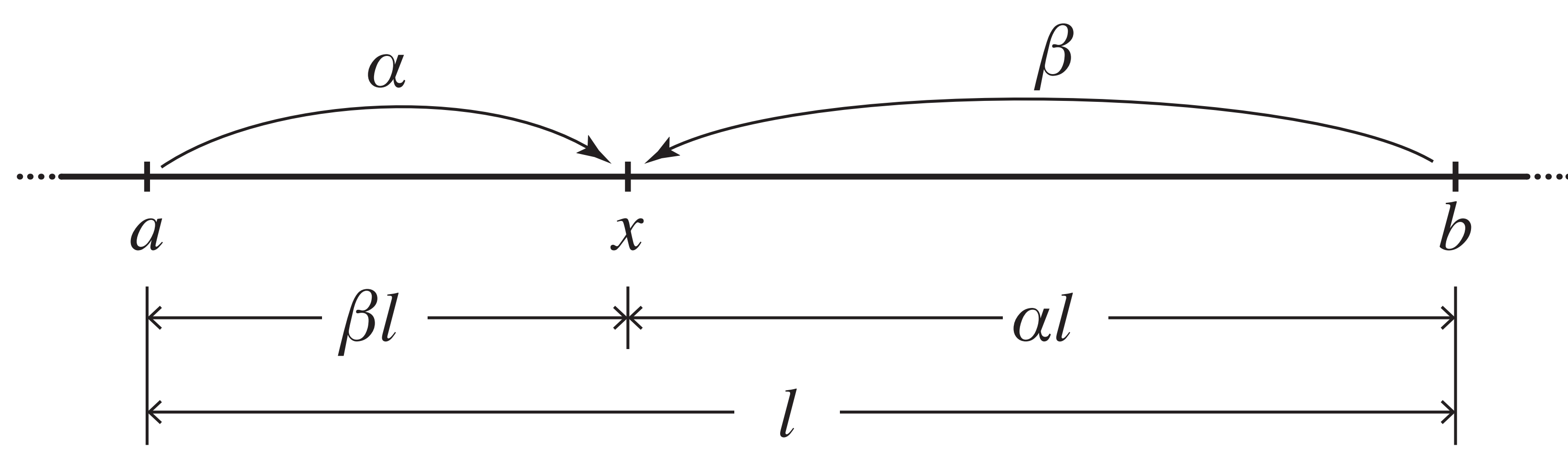
- **Alternate story**
  1. write *x* as convex combination of a and b

$$x = \alpha a + \beta b \quad \text{where } \alpha + \beta = 1$$

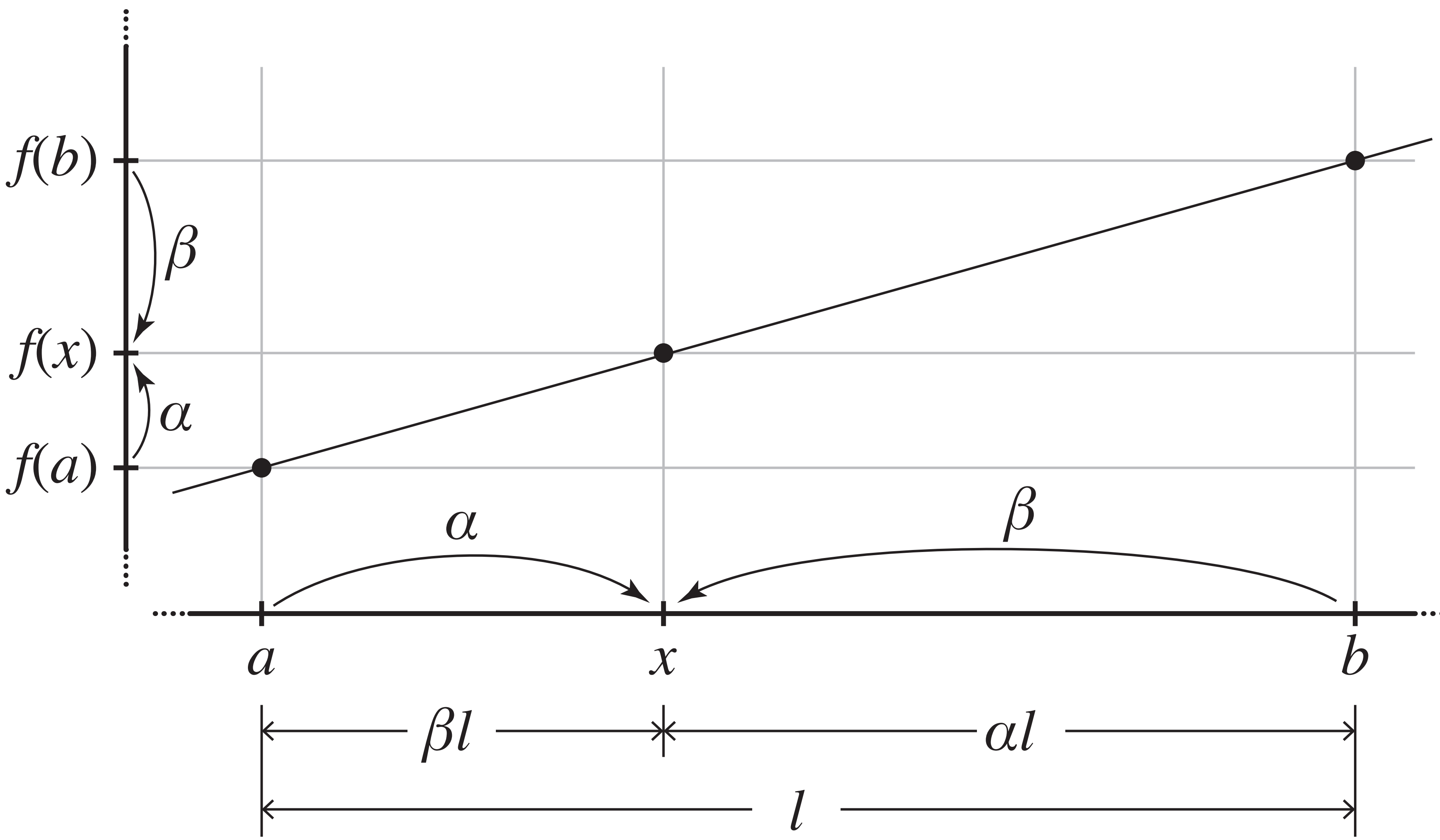
  2. use the same weights to compute ***f***(***x***) as a convex combination of ***f***(***a***) and ***f***(***b***)

$$f(x) = \alpha f(a) + \beta f(b)$$

Linear interpolation in 1D



Linear interpolation in 1D



Linear interpolation in 2D

- **Use the alternate story:**
  1. Write **x**, the point where you want a value, as a convex linear combination of the vertices

     $$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} \quad \text{where } \alpha + \beta + \gamma = 1$$

  2. Use the same weights to compute the interpolated value *f*(**x**) from the values at the vertices, *f*(**a**), *f*(**b**), and *f*(**c**)

     $$f(\mathbf{x}) = \alpha f(\mathbf{a}) + \beta f(\mathbf{b}) + \gamma f(\mathbf{c})$$

See textbook Sec. 2.7

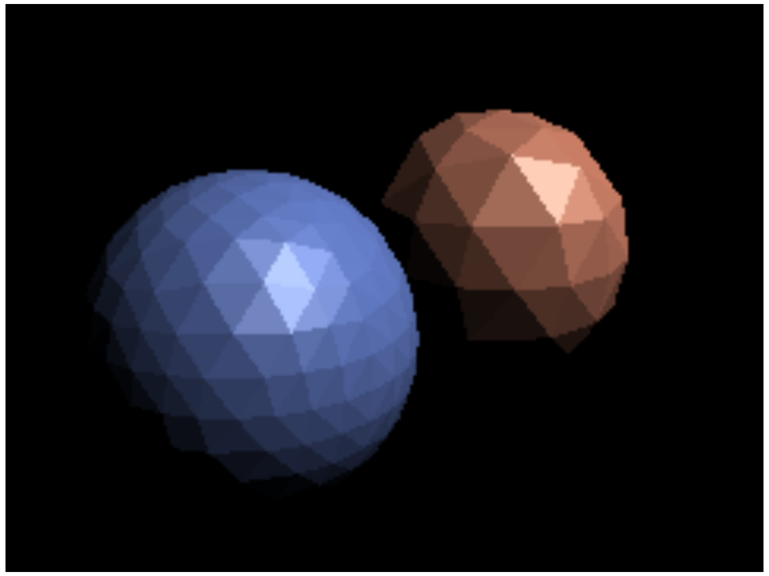
Interpolation in ray tracing

- **When values are stored at vertices, use linear (barycentric) interpolation to define values across the whole surface that:**
  1. …match the values at the vertices
  2. …are continuous across edges
  3. …are piecewise linear (linear over each triangle)

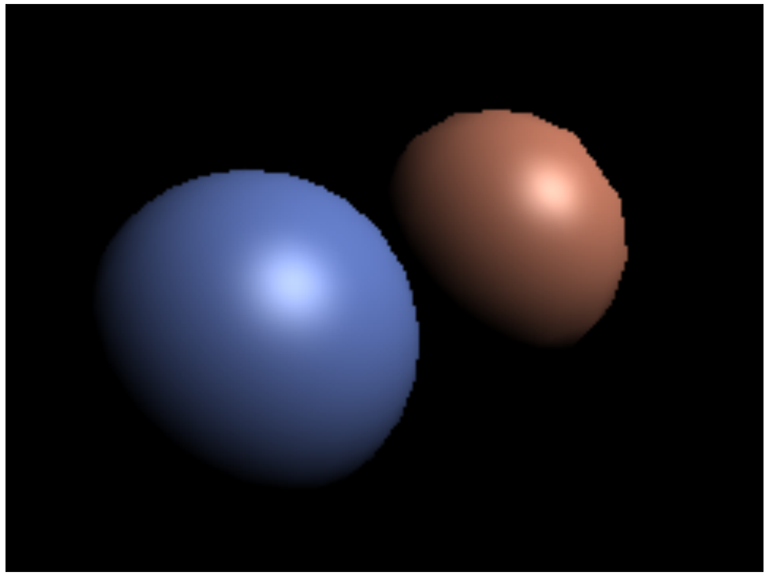
     as a function of 3D position, not screen position—more later
- **How to compute interpolated values**
  4. during triangle intersection compute barycentric coords
  5. use barycentric coords to average attributes given at vertices

What to interpolate?

- **Texture coordinates**
  – without interpolating there can't really be textures
- **Surface normals**
  – for smooth surfaces approximated with meshes
  – use interpolated normal for shading in place of actual normal
  – "shading normal" vs. "geometric normal"



geometric normals



interpolated normals

