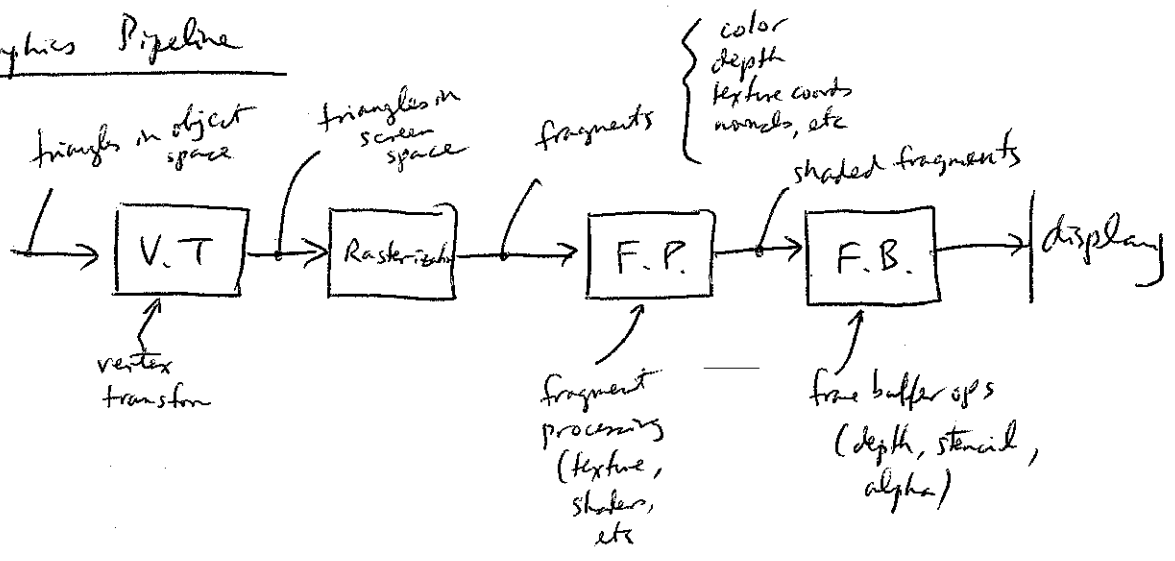


Perspective - correct interpolation

1. Project interpolants by dividing by $w \rightarrow$ this allows you to do the interpolation in screen space
2. Interpolate interpolants as $\frac{1}{w}$
3. Get original interpolants by multiplying by interpolated w .

Graphics Pipeline



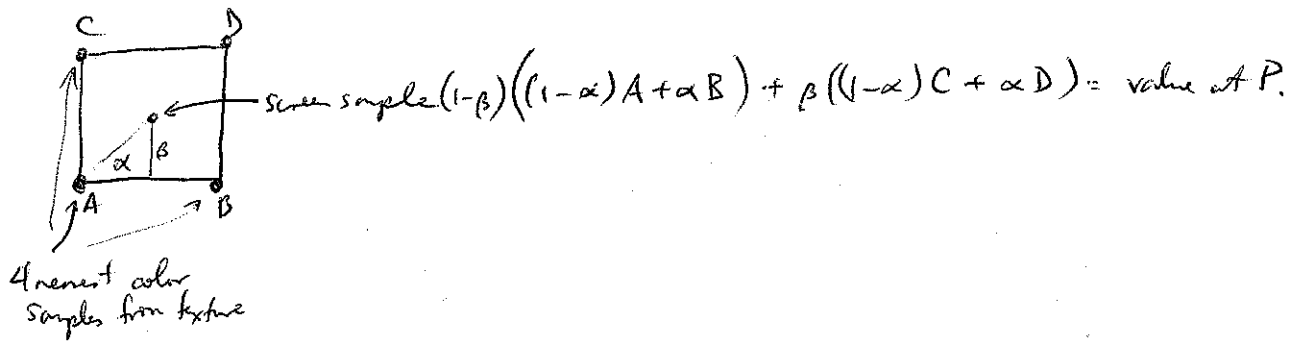
Texture Mapping

- Screen pixel samples may not align with texture samples.
- Need to interpolate!

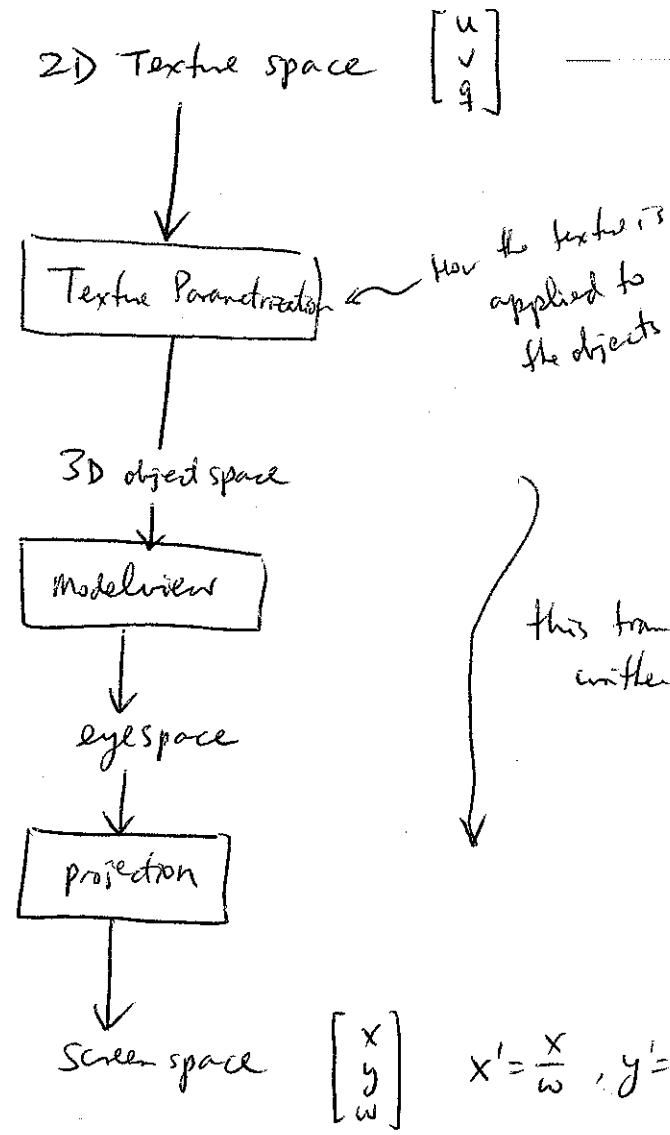
nearest neighbor
bilinear reconstruction

Two kinds of reconstruction available on the hardware: gl-nearest, gl-linear

GL_LINEAR



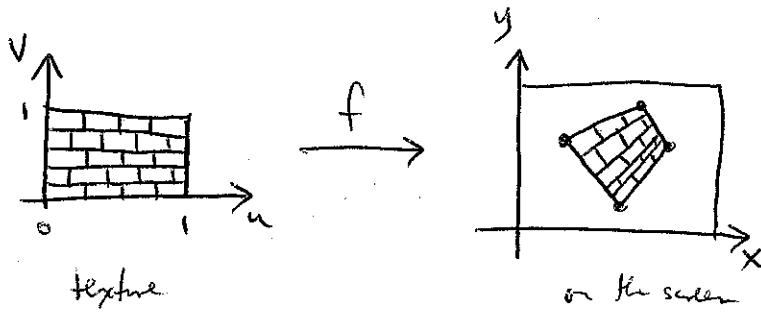
Coordinate Systems:



this transformation can be written as a matrix

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_f \begin{bmatrix} u \\ v \\ q \end{bmatrix}$$

Common Mapping pipelines



Forward Mapping ← Texture order u, v are ints

for all (u, v) in texture {
 x, y are floats → (x, y) = f(u, v)
 Screen(x, y) = texture(u, v) ← put texture sample into screen
 }

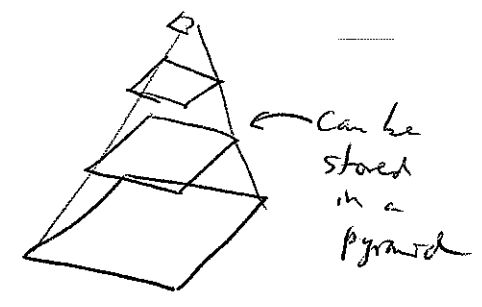
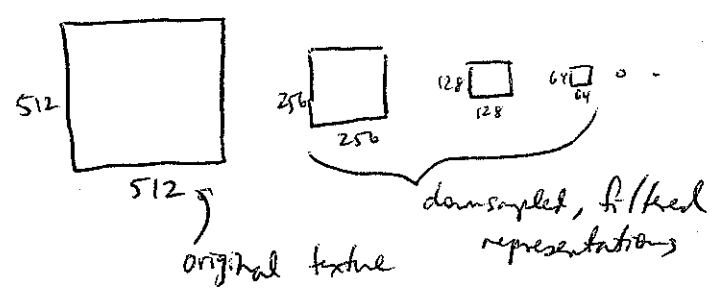
Backward mapping ← screen order integers x, y

for all (x, y) on polys on screen {
 floats(u, v) → (u, v) = f⁻¹(x, y)
 screen(x, y) = texture(u, v)
 }

Mipmapping

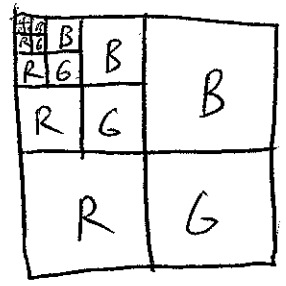
Mip Map - Multum In Parvo - "many things in a small place"
 by Lance Williams

Idea: Build a "pyramid" of images at decreasing resolutions:



when fetching a sample, we decide which "level" of the pyramid to set the colors from.

Compact representation:



only takes $\frac{4}{3}$ x the size of the original image!