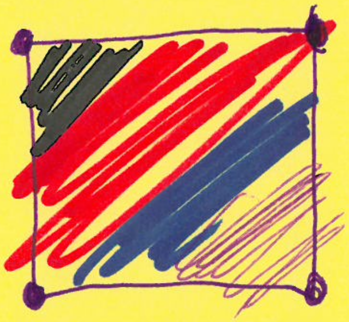
Texturing

2/27



Where to implement texturing:

- CPU
- Vertex Shader
- Fragment Shader

def texture lookup: in coordinate system of the texture find the location of the "Shading" point read the color from the image

```
Color texture_lookup(Texture t, float u, float v){

    i = round(u * t.width)
    j = round(v* t.height)
    return t.get-pixel(i,j)

}
```

```
Color shade_surface_point(Surface s, Point p, Texture t){
    normal = s.get_normal(p)
    (u,v) = s.get_texture_coord(p)
    texture_sample = texture_lookup(t,u,v)
      .
      .
      .
      // shade using favorite shading algo
      .
      .
      .
      return color_from_shading
}
```

$(u, v) = S.\mathrm{get\_texture\_coord}(p)$

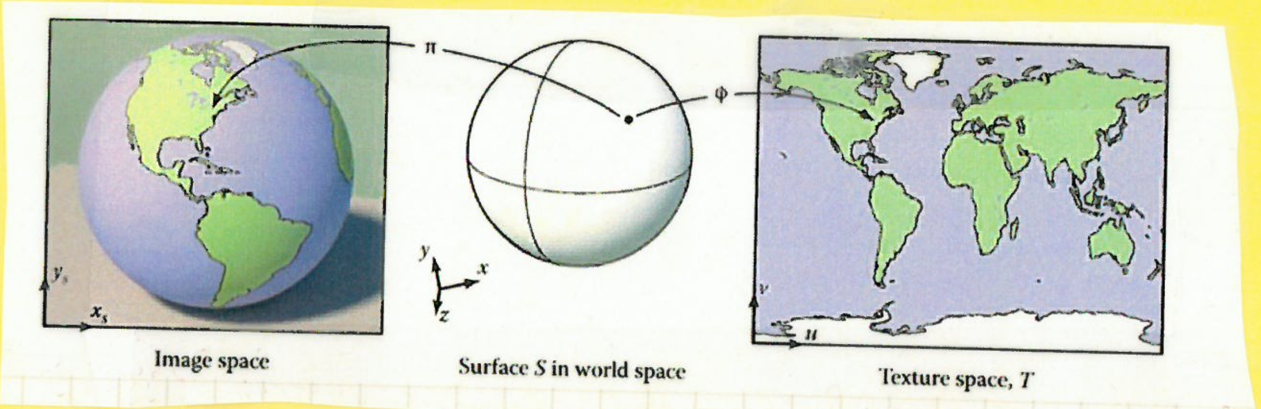


Image space

Surface *S* in world space

Texture space, *T*

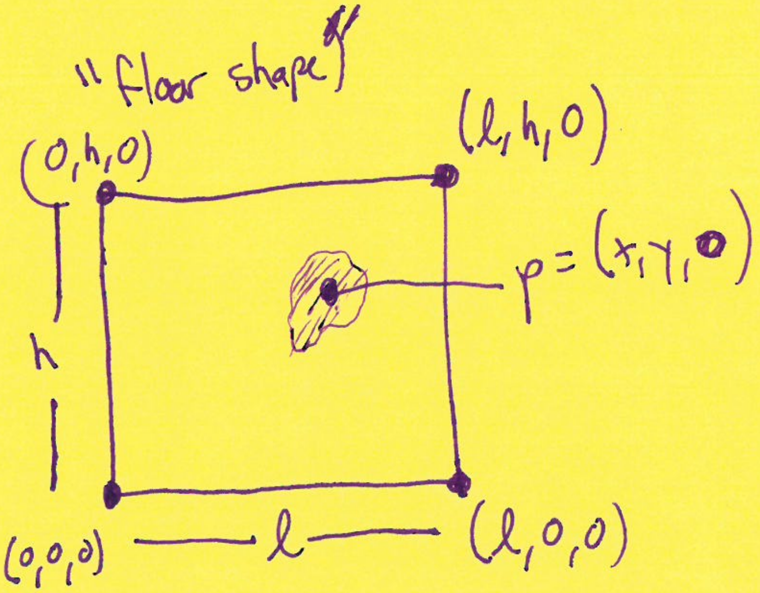
"View map" π takes a point from surface to image space

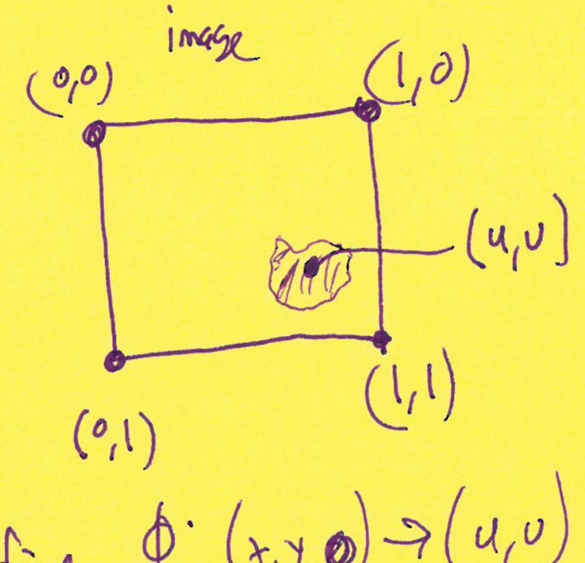
$$\phi : S \to T$$

surface

texture space

$$(x, y, z) \longmapsto (u, v)$$





define $\phi : (x, y, 0) \to (u, v)$

$$(x, y, 0) \to \left(\frac{x}{\ell}, \frac{y}{h}\right)$$

φ ↗