Inverse Mapping

Reading

Recommended

- A. Grassner. Introduction to Ray Tracing. Chapter on Texture Mapping. (available in reserve section in the Library).

Inverse Mapping

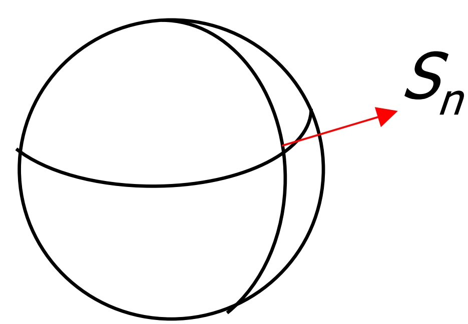
- Consider an object point P visible to a pixel p (given by the ray tracer).
- Texture mapping takes a texture map as input.
- To determine which pixel in the texture map is mapped to P, a mapping function from P to the texture map is needed.
- On hitting P, the ray tracer will invoke an inverse mapping function, which is primitive dependent (just like ray-object intersection) to get the needed color.
- Once the color of 3D point P is known, you can apply Phong shading to determine the exact shade of 2D pixel p.

Inverse Mapping Functions

- In Glassner’s book, it describes the mathematics of inverse mapping of
  - Spheres
  - Convex quadrilaterals
  - Circles
  - Cones
- We shall only describe spherical inverse function.

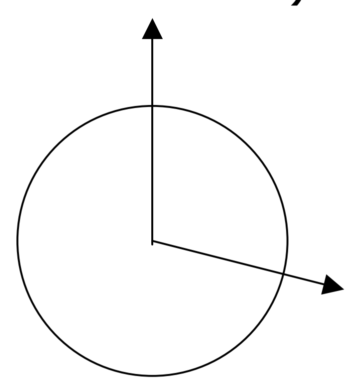
Spherical Inverse Mapping

- The problem is to convert the intersection point into its longitude and latitude value. (Remember the Globe?)
- The inputs
  - S_n (normal to the intersection point P)
  - $S_{pole} = S_p = [X_p\ Y_p\ Z_p]^T$
  - $S_{equator} = S_e = [X_e\ Y_e\ Z_e]^T$
- By definition, the inner product of S_p and S_e is zero (i.e., they are perpendicular)



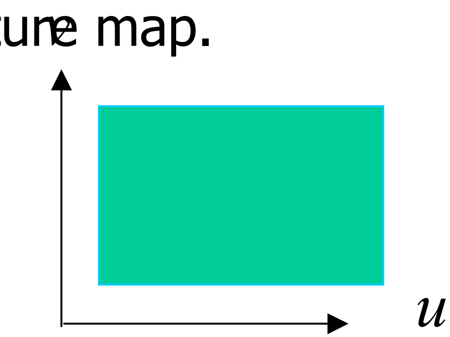
Spherical Inverse Mapping

- S_p is a unit vector which points from the sphere's center to the north pole of the sphere
- S_e is a unit vector which points to a reference point on the equator. (You can choose any convenient point, say $(1\ 0\ 0)^T$, since a sphere is "orientation-less").



U-V space

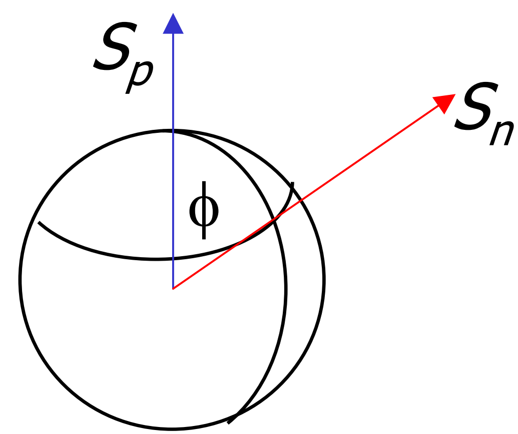
- We sometimes call parameter space ***u-v space***.
- The parameter *u* varies along the *equator* from zero to one. At the pole, *u* is defined as zero.
- The parameter v varies from zero to one, from the south pole to the north.
- S_n, the normal to the sphere at the intersection point *P*, is equal to the unit vector pointing from the sphere's center to the intersection point.
- We use *u* and *v* to parameterize the texture map.



Obtain *v*

- We obtain the latitudinal parameter v. This is equal to:

$$\phi = \cos^{-1} (- S_n \cdot S_p)$$
$$v = \phi / \pi \text{ in } [0,1]$$



- If v is equal to zero or one, then u is set to zero. Else, ...

Obtain *u*

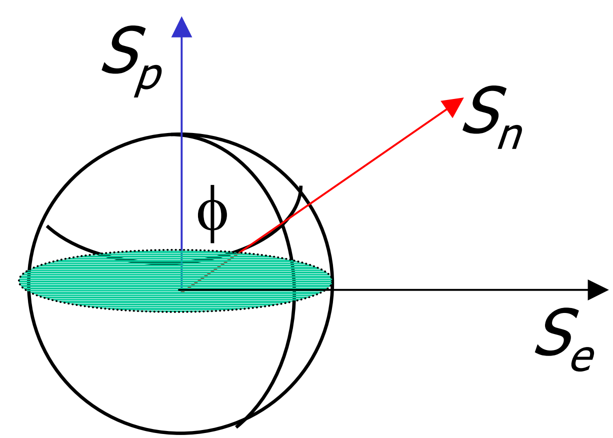
- Else, obtain the longitudinal parameter u by:

  $$\theta = \cos^{-1}((S_e \,.\, S_n) / \sin\phi) / 2\pi$$

  (to understand this equation, let S_n be incident on the equator)

- Now, take the cross product of the two sphere axes and compare this direction with the direction of the normal:

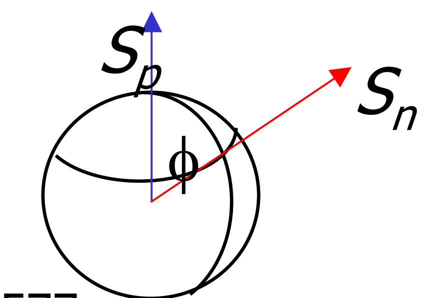
- if $((S_p \times S_e) \,.\, S_n) > 0$
  - then $u = \theta$
  - else $u = 1 - \theta$



Example

- Let $S_n = [0.577\ -0.577\ 0.577]^T$ on a sphere whose axes are
  - $S_p = [0\ 0\ 1]^T$
  - $S_e = [1\ 0\ 0]^T$

$$\phi = \cos^{-1}(-S_n \cdot S_p) = 2.186$$
$$v = \phi / \pi = 2.186 / 3.14159 = \mathbf{0.696}$$
$$\theta = \cos^{-1}((S_e \cdot S_n) / \sin\phi) / 2\pi = 0.125$$



- Testing: $([0\ 0\ 1] \times [1\ 0\ 0]) \cdot [0.577\ -0.577\ 0.577] = -0.577 < 0$
- $u = 1 - 0.125 = \mathbf{0.875}$
- Therefore, the texture coordinates are (0.696, 0.875). If we are using a 640 x 480 texture image, the texture element is

$$639 \times 0.875 = \mathbf{559}$$
$$479 \times 0.696 = \mathbf{334}$$

- You can perform **bilinear resampling** to get an average color.