Texture Mapping in Practice

Dinesh K. Pai

Textbook Appendix A4, Chapter 15

Some slides courtesy of M. Kim, KAIST

Today

- Announcements
  - Assignment 2 spotlights

- Texture mapping, continued
Assignment 2 Spotlights

- Soohyun Kim
- Tim Straubinger
- Hoda Hashemi
- Fan Wu
- Sean Vanbergen
- Jean Chen

Texture mapping

- In basic texturing, we simply ‘glue’ part of an image onto a triangle by specifying texture coordinates at the three vertices.
Texture mapping

- A bunch of OpenGL/WebGL functions to load a texture and set various parameters (lin/const, mipmap, wrapping rules).
- A uniform variable is used to point to the desired texture unit.

Texture mapping

- Varying variables are used to store texture coordinates.
- In this simplest incarnation, we just fetch r,g,b values from the texture and send them directly to the frame buffer.
Texture mapping

- Alternatively, the texture data could be interpreted as, say, the diffuse material color of the surface point, which would then be followed by the diffuse material computation described earlier.

Steps for Texture Mapping

1. Create a *texture object* and load texels into it
2. Include *texture coordinates* with your vertices
3. Associate a *texture sampler* with each texture map used in shader
4. Retrieve texel values

(Reference: Red Book)
Recap
Understanding Texture Mapping

- Better view: An efficient way to model surface detail using discrete (sampled) data
- Need to understand two surprisingly subtle concepts
  - “Coordinates” Parameterization of surfaces
  - “Images” Sampled representations of continuous functions
- More details in Chapters 16-18. We'll be covering this at a high level.

An intuitive example
How to model the earth?

![Image of the Earth with longitude and latitude lines]
Earth (texture) Map

Texture coordinates, simple case

Cubli-linear coordinates

\( f(p) \)

\( f(\text{texture map}) \)
Texture Mapping

Construct a map $f$ which covers the higher dimensional object.

$$f(p) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos p \\ \sin p \\ 1 \end{pmatrix}$$

In general, this simple technique doesn't work. (No good "global" function exists.) But can do this locally! $\breve{p}$, $u$, $v$

$f$ can be a simple linear interpolation of vertex values. Using varying variables $p$, $u$, $v$. 

P-nurbs data $\rightarrow$ $\breve{p}$, $u$, $v$