CS-184: Computer Graphics

Lecture #11: Texture and Other Maps

Prof. James O'Brien
University of California, Berkeley

Today

- Texture Mapping
  - 2D
  - 3D
  - Procedural
- Bump and Displacement Maps
- Environment Maps
- Shadow Maps
Surface Detail

- Representing all detail in an image with polygons would be cumbersome

2D Texture Mapping of Images

- Use a 2D image and map it to the surface of an object
2D Texture Mapping of Images

- Example of texture distortion

Texture Coordinates

- Assign coordinates to each vertex
- Within each triangle use linear interpolation
- Correct for distortion!
Procedural Textures

- Generate texture based on some function
  - Well suited for “random” textures
  - Often modulate some noise function

Assigning Texture Coordinates

- Map a simple shape onto object by projection
  - Sphere, cylinder, plane, cube
- Assign by hand
- Use some optimization procedure
Repeating Textures

- Image Tiles allow repeating textures
  - Images must be manipulated to allow tilling
  - Often result in visible artifacts
    - There are methods to get around artifacts....

Repeating Textures

- Image Tiles allow repeating textures
  - Images must be manipulated to allow tilling
  - Often result in visible artifacts
    - Artifacts not an issue for artificial textures
Non-Color Textures

Color Specular Bump

Bump Mapping

No bump mapping  With bump mapping

Images by Paul Baker
www.paulsprojects.net
Bump Mapping

- Add offset to normal
  - Offset is in texture coordinates S,T,N
  - Store normal offsets in RGB image components
  - Should use correctly orthonormal coordinate system

- Normal offsets from gradient of a grayscale image
  - \( \mathbf{b}(u, v) = [s, t, n](u, v) = \nabla i(u, v) \)
  - \( \nabla = \begin{bmatrix} \frac{\partial}{\partial u} & \frac{\partial}{\partial v} \end{bmatrix}^T \)

Bump Map Example

Catherine Bendebury and Jonathan Michaels
CS 184 Spring 2005
Displacement Maps

- Actually move geometry based on texture map
  - Expensive and difficult to implement in many rendering systems
  - Note silhouette

Environment Maps

- Environment maps allow crude reflections
- Treat object as infinitesimal
  - Reflection only based on surface normal
- Errors hard to notice for non-flat objects
Environment Maps

$u = \frac{y + x}{2x}$

$v = \frac{z + x}{2x}$

right face has $x > |y|, x > |z|$
Shadow Maps

- Pre-render scene from perspective of light source
  - Only render Z-Buffer (the shadow buffer)
- Render scene from camera perspective
  - Compare with shadow buffer
  - If nearer light, if further shadow

Note: These images don’t really go together, see the paper...

From Stamminger and Drettakis
SIGGRAPH 2002
Deep Shadow Maps

- Some objects only partially occlude light
  - A single shadow value will not work
  - Similar to transparency in Z-Buffer

From Lokovic and Veach
SIGGRAPH 2000