Texture Mapping

Greg Humphreys
University of Virginia
CS 445, Fall 2003

Textures

- Describe color variation in interior of 3D polygon
  - When scan converting a polygon, vary pixel colors according to values fetched from a texture

3D Rendering Pipeline (for direct illumination)

3D Primitives → 3D Modeling Coordinates

LIGHTING

Viewing Transformation → 3D World Coordinates

3D World Coordinates → 3D Camera Coordinates

Polygonal Transformation → 2D Screen Coordinates

Clipping

Clipping → 2D Screen Coordinates

Viewing Transformation → 2D Image Coordinates

Texture mapping

Polygonal model

Surface Textures

- Add visual detail to surfaces of 3D objects

With surface texture
Surface Textures

• Add visual detail to surfaces of 3D objects

Parameterization

\[ \text{geometry} + \text{image} = \text{texture map} \]

• Q: How do we decide where on the geometry each color from the image should go?

Option: Varieties of projections

Option: unfold the surface

[Paul Bourke]

[Piponi2000]
Option: make an atlas

charts atlas surface
[Sander2001]

Option: it's the artist's problem

Overview

• Texture mapping methods
  ◦ Parameterization
  ◦ Mapping
  ◦ Filtering

• Texture mapping applications
  ◦ Modulation textures
  ◦ Illumination mapping
  ◦ Bump mapping
  ◦ Environment mapping
  ◦ Image-based rendering
  ◦ Volume textures
  ◦ Non-photorealistic rendering

Texture Mapping

• Steps:
  ◦ Define texture
  ◦ Specify mapping from texture to surface
  ◦ Lookup texture values during scan conversion
Texture Mapping

- When scan convert, map from …
  - image coordinate system (x,y) to
  - modeling coordinate system (u,v) to
  - texture image (t,s)

![Diagram of texture mapping coordinates](image)

Naïve Texture Mapping

- A first cut at a texture-mapping rasterizer:
  - For each pixel:
    - Interpolate u & v down edges and across spans
    - Look up nearest texel in texture map
    - Color pixel according to texel color (possibly modulated by lighting calculations)
  - McMillan's demo of this is at [link](http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide05.html)
  - What artifacts do you see in this demo?

Naïve Texturing Artifacts

- Warping at edges of triangles
- A more obvious example: [link](http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide06.html)
- Consider the geometry of interpolating parameters more carefully
Interpolating Parameters

- The problem turns out to be fundamental to interpolating parameters in screen-space
  - Uniform steps in screen space ≠ uniform steps in world space

Texture Mapping

- Linear interpolation of texture coordinates
- Correct interpolation with perspective divide

Interpolating Parameters

- Perspective foreshortening is not getting applied to our interpolated parameters
  - Parameters should be compressed with distance
  - Linearly interpolating them in screen-space doesn’t do this

- Is this a problem with Gouraud shading?
  - A: It can be, but we usually don’t notice (why?)
    - http://graphics.lcs.mit.edu/classes/6.837/P06/Lecture21/Side17.html

Perspective-Correct Interpolation

- Skipping a bit of math to make a long story short…
  - Rather than interpolating u and v directly, interpolate u/z and v/z
    - These do interpolate correctly in screen space
  - Also need to interpolate Z and multiply per-pixel
  - Problem: we don’t know Z anymore
  - Solution: we do know w ∝ 1/z
  - So…interpolate uw and vw and w, and compute
    - $u = uw/w$ and $v = vw/w$ for each pixel
    - This unfortunately involves a divide per pixel (Just 1?)
### Overview

- **Texture mapping methods**
  - Parameterization
  - Mapping
  - Filtering

- **Texture mapping applications**
  - Modulation textures
  - Illumination mapping
  - Bump mapping
  - Environment mapping
  - Image-based rendering
  - Non-photorealistic rendering

### Texture Filtering

- Must sample texture to determine color at each pixel in image

#### Texture Map Aliasing

- Naive texture mapping aliases badly
- Look familiar?

  ```cpp
  int uval = (int) (u * denom + 0.5f);
  int vval = (int) (v * denom + 0.5f);
  int pix = texture.getPixel(uval, vval);
  ```

- Actually, each pixel maps to a region in texture
  - $|\text{PIX}| < |\text{TEX}|$
    - Easy: interpolate (bilinear) between texel values
  - $|\text{PIX}| > |\text{TEX}|$
    - Hard: average the contribution from multiple texels
  - $|\text{PIX}| \sim |\text{TEX}|$
    - Still need interpolation!

#### Texture Filtering

- Size of filter depends on projective warp
  - Can prefilter images
    - Mip maps
    - Summed area tables

![Angel Figure 9.4](image-url)

![Angel Figure 9.14](image-url)
### Mip Maps

- Keep textures prefiltered at multiple resolutions
  - For each pixel, linearly interpolate between two closest levels (e.g., trilinear filtering)
  - Fast, easy for hardware
- Why “Mip” maps?

---

### Summed-area tables

- At each texel keep sum of all values down & right
  - To compute sum of all values within a rectangle, simply subtract two entries
  - Better ability to capture very oblique projections
  - But, cannot store values in a single byte

---

### MIP-map Example

- No filtering:
  - AAAAAAGH
  - MY EYES ARE BURNING
- MIP-map texturing:
  - Where are my glasses?

---

### Summed-Area Tables

- Mipmaps assume that each pixel projects to a square in the texture (which is a lie)
- SAT can integrate texels covered by the pixel more exactly (but still quickly)
- Example:
  - MIP-map texturing
  - Summed-area table texturing
Overview

- Texture mapping methods
  - Parameterization
  - Mapping
  - Filtering
- Texture mapping applications
  - Modulation textures
  - Illumination mapping
  - Bump mapping
  - Environment mapping
  - Image-based rendering
  - Non-photorealistic rendering

Modulation textures

Map texture values to scale factor

\[ I = T(x)(I_d + K_d I_r + \sum_i (K_a (N \cdot L) + K_s (V \cdot R)^s) S_i I_s + K_t I_t + K_f I_f) \]

Texture Mapping Variations

- A texture can modulate any parameter in the Phong lighting equation

Bump Mapping

- Texture = change in surface normal!

Texture as R,G,B:

Texture as diffuse lighting coefficients:

Texture value

Wood texture

Texture value

Texture value

Sphere w/ diffuse texture

Sweaty bump map

Sphere w/ diffuse texture and sweaty bump map
More Bump Mapping

Displacement Mapping

• How can you tell a bumped-mapped object from an object in which the geometry is explicitly modeled?

Illumination Maps

• Quake introduced illumination maps or light maps to capture lighting effects in video games

Texture map:

Light map

Texture map + light map:

Environment Maps

Images from Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments
Gene Miller and C. Robert Hoffman
SIGGRAPH 1984 “Advanced Computer Graphics Animation” Course Notes
Solid textures

Texture values indexed by 3D location (x,y,z)
- Expensive storage, or
- Compute on the fly, e.g. Perlin noise

Procedural Texture

Procedural Texture Gallery