Texture Mapping

Greg Humphreys
CS445: Intro Graphics
University of Virginia, Fall 2004

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
- Modeling
- Transformations
- Lighting

3D World Coordinates
- Projection
- Transformation
- Clipping
- Camera Coordinates
- Screen Coordinates
- Texture Mapping
- Image

Surface Textures

- Add visual detail to surfaces of 3D objects

Surface Textures

- Add visual detail to surfaces of 3D objects

Parameterization

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

[Daren Horley]
Option: Varieties of projections

Option: unfold the surface

Option: make an atlas

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 \((t,s)\)

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 http://graphics.cs.mtu.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:
  http://graphics.cs.mtu.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

Linear interpolation of texture coordinates
Correct interpolation with perspective divide

Hill Figure 8.42
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?)
    - [Link](http://graphics.cs.utah.edu/asseti/6.837/F98/lecture21/Slides17.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?)
    - [Link](http://graphics.cs.utah.edu/asseti/6.837/F98/lecture21/Slides14.html)

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?
  ```
  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
    - $|PIX| < |TEX|$:
      - Easy: interpolate (bilinear) between texel values
    - $|PIX| > |TEX|$:
      - Hard: average the contribution from multiple texels
    - $|PIX| \sim |TEX|$
      - Still need interpolation!

Texture Filtering

- Size of filter depends on projective warp
  - Can prefilter images
    - Mip maps
    - Summed area tables
  - Angel Figure 9.14
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, do two subtracts and an add
  - Better ability to capture very oblique projections
  - But, cannot store values in a single byte

Overview

- Texture mapping methods
  - Parameterization
  - Mapping
  - Filtering

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

MIP-map Example

- No filtering:
  - AAAAAAAGH 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:

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

\[ I = T_c(\alpha L + \beta F) + \sum (K_a(N \cdot L) + K_s(V \cdot R) S \cdot I_a + K_d F_a + K_v I_v) \]
Texture Mapping Variations

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

Texture as $R,G,B$:

Texture as diffuse lighting coefficients:

Bump Mapping

- Texture = change in surface normal!

More Bump Mapping

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

Displacement Mapping

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 Waiter 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