Lecture 17: Texture Mapping I

Rui Wang
Topics

- What is texture mapping?
- Defining texture coordinates
- Interpolating texture coordinates
  - Perspective Correct Texture Mapping
- Texture filtering
- Texture applications
Textures: adding visual details

- Add surface details using images (as opposed to polygons).
Textures: adding visual details

- The idea is akin to apply a gift wrapping paper around the object (note: this involves distorting the image)
Texture Mapping

How do we decide where on the geometry each color from the image should go?
Texture Mapping

- Many different possibilities.
How to map textures to objects?

- Each vertex on the object is associated with a 2D texture coordinate \([u,v]\).
- Texture coords should preserve spatial locality.
- They are defined at vertices and interpolated at pixels after rasterization.
Options: cut, unfold, and flatten

[Piponi2000]
Options: make an atlas

charts

atlas

surface

[Sander2001]
Examples of Texture Mapping

- Texture image

(0,0)  (1,0)
(0,1)  (1,1)
Defining Texture Coordinates

- Simple geometry: quad

![Diagram showing texture coordinates on a 2D object, with texture coordinates (0,0), (1,0), (0,1), and (1,1) labeled.]
Defining Texture Coordinates

- Simple geometry: quad

It’s possible that they do not have the same aspect ratio!
Manipulate Texture Coordinates

- Texture extension
  - Texture coords can go beyond [0,1].
Manipulate Texture Coordinates

- Texture extension
  - Texture coords can go beyond \([0,1]\).
Manipulate Texture Coordinates

- What about this?
OpenGL Texture Mapping

- Use `glTexCoord2f` to specify texture coords

```c
glBegin( GL_QUADS );
    glTexCoord2f( 0, 0 );
    glVertex3f(...);
    glTexCoord2f( 0, 1 );
    glVertex3f(...);
    glTexCoord2f( 1, 1 );
    glVertex3f(...);
    glTexCoord2f( 1, 0 );
    glVertex3f( ... );
glEnd();
```

- **Nate Robins Demo**
Defining Texture Coordinates

- Parametric Surfaces
  - Use \([u,v]\) parameters to define texture coords.
Defining Texture Coordinates

- Parametric Surfaces
  - Use \([u,v]\) parameters to define texture coords.

\[
\begin{bmatrix}
\frac{\theta}{\pi}, \frac{\phi}{2\pi}
\end{bmatrix}
\]
Defining Texture Coordinates

- What about arbitrary shapes?
- Idea: use proxy shapes \(\rightarrow\) typically simple parametric surfaces like sphere, cylinder, cube etc.

- First map objects to proxy shape, then map proxy shape to texture.
Spherical Mapping

- Project to sphere
  \[ (x, y, z) \rightarrow (\text{theta, phi}) = (u, v) \]
Cylindrical Mapping

- Project to cylinder
  \((x,y,z) \rightarrow (\theta, z) \rightarrow (u,v)\)
Cubical Mapping
Interpolating Texture Coords

1st try: linear interpolation

\[ I_a = \text{Linear \_ Interpo}(I_1, I_2) \]
\[ I_b = \text{Linear \_ Interpo}(I_1, I_3) \]
\[ I_p = \text{Linear \_ Interpo}(I_a, I_b) \]

Interpolate texture coords in exactly the same way as other vertex attributes (colors, normals etc.)
Artifacts

☐ McMillan’s Demo
  - http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide05.html

☐ Another Example
  - http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide06.html

☐ What artifacts do you see?
☐ What’s the problem?
The problem turns out to be fundamental to interpolating parameters in screen-space.

Uniform steps in screen space ≠ uniform steps in world space.
Interpolating Texture Coords

Linear interpolation of texture coordinates

Correct interpolation with perspective divide
Interpolating Texture Coords

- Perspective effects are not getting applied to interpolated parameters
  - Parameters should be compressed with distance; screen space interpolation does not do this
Interpolating Texture Coords

- Perspective effects are not getting applied to interpolated parameters
  - Parameters should be compressed with distance; screen space interpolation does not do this

- Is this a problem with color interpolation?
Interpolating Texture Coords

- Perspective effects are not getting applied to interpolated parameters
  - Parameters should be compressed with distance; screen space interpolation does not do this

- *Is this a problem with color interpolation?*
  A: It can be, but we usually don’t notice

- [http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide17.html](http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide17.html)
Interpolating Texture Coords

- Perspective effects are not getting applied to interpolated parameters
  - Parameters should be compressed with distance; screen space interpolation does not do this

- Is this a problem with color interpolation?
  A: It can be, but we usually don’t notice

- Is this a problem with ray tracing?

http://graphics.lcs.mit.edu/classes/6.837/F98/Lecture21/Slide17.html
Perspective-Correct Interpolation

- Skipping a bit of math to make a long story short
  - Rather than linearly interpolating u and v, we interpolate u/z and v/z;
  - At the same time, linearly interpolate z;
  - Finally, multiply z back to get the new u, v
- These do interpolate correctly in screen space.