Lecture 22: Texture Mapping

November 27, 2018
Now to Texture mapping

Ubiquitous, and sometimes obvious

Image from http://www.minecraftercamp.com
Adding surface detail

- Surfaces in the world have appearance
  - They are seldom one flat color.
  - They have true texture – repeating patterns.
  - They have structured markings.
  - They have tiny changes in surface height.

- Purists view (don’t try this!)
  - Use ever more even smaller uniform triangles.

- Pragmatists view
  - Paint surfaces with images – texture mapping.
Texture Mapping

• Use projective geometry to compute where vertices appear in the image
• Apply shading to determine the color of pixels
  -- or --
• Map an existing texture onto a surface
  -- Textures supercede/augment the specification of surface material
  -- Leaves room for distinction diffuse vs. specular
Mapping

- Guess what? The underlying problem is to apply a geometric transformation

$$t,s \text{ coordinates}$$
Mapping (II)

- Textures are color images
  - Logical texture coords run from (0,0) to (1,1)
  - Coordinates fixed regardless of image size
- Polygons are 2D surfaces in a 3D space
- The transformation from texture coordinates to surface coordinates is expressed as – you guessed it – a matrix
Texture Matrices

• Given vertices and the corresponding texture coordinates...

\[
\begin{bmatrix}
  t \\
  s
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
\]

- How many correspondences are needed?
- Maps from surface to texture
Worked Example - Before

• Consider point correspondences  
  – Pairs of points in texture and 3-D coordinates  
  – Three such pairs of points yield six constraints  
  – Constraints match free variables – six.

• Specifically  
  – Point (0,0) matches point (1,2,3)  
  – Point (1,1) matches point (2,2,2)  
  – Point (0,1) matches point (3,2,2)
Worked Example

Match #1
\[
\begin{bmatrix}
0 \\
0
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f
\end{bmatrix} \begin{bmatrix}
1 \\
2 \\
3
\end{bmatrix}
\]

Match #2
\[
\begin{bmatrix}
1 \\
1
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f
\end{bmatrix} \begin{bmatrix}
2 \\
2 \\
2
\end{bmatrix}
\]

Linear Alg. Setup
\[
\begin{bmatrix}
0 \\
1 \\
0
\end{bmatrix} = \begin{bmatrix}
1 & 2 & 3 \\
2 & 2 & 2 \\
3 & 2 & 3
\end{bmatrix} \begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
\]

Match #3
\[
\begin{bmatrix}
0 \\
1 \\
0
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f
\end{bmatrix} \begin{bmatrix}
3 \\
2 \\
3
\end{bmatrix}
\]

Solved in Maple

\[
\begin{align*}
a &= 0 \\
b &= \frac{3}{2} \\
c &= -1 \\
d &= \frac{1}{2} \\
e &= \frac{1}{2} \\
f &= -\frac{1}{2}
\end{align*}
\]

3 equations for first texture coords
Texture Mapping (II)

• The fragment processor computes a reflectance color for every pixel
• When textures are enabled,
  – The fragment processor also computes a texture value for every pixel
  – Using the pixel to do texture mapping
• These values are multiplied together to produce the final value
Issue #1: Sampling

- The mapping from surface points to texture coordinates produces real values
Sampling

• Nearest-neighbor:
  – pick the closest texture pixel

• Bilinear:
  – linearly interpolate in both dimensions

• Bicubic:
  – fit a 3rd order surface to 16 surrounding points
  – Not as expensive as it sounds
Sampling (III)

• A better solution is for the texture map to be roughly the same size as the surface projection.

• A MipMap is an image pyramid built from a texture map
  – Example: if the texture is 64x64, the pyramid also includes 32x32, 16x16, etc.
Issue #2: Getting Textures

- **WEB!**
  - *millions* of textures – people use them for backgrounds of web pages a lot! You can download them in bulk packages, etc..

- **Build your own**
  - Make them “seamless”
    - When tiled, you cannot see the edges of the tiles.
Just for example ...
Texture Makers

• There are tons of them.. Some examples:
  – http://www.backgroundmagic.com/software/BGM.zip
  – http://216.156.212.112/photoseam.exe
Example in SketchUp - Cube
Import a Texture Image
Place Texture on Face
View the Result
Position/Reset-Position
The Essence of Tiling
View the Result
Support for ‘Painting’ Textures
Final Result – Textured Cube
Alas - .obj support marginal

- Texture vertices do come through.
- File linkage to texture map does not.

```
# Alias OBJ Model File
# Exported from SketchUp, (c) 2000-2012 Trimble Navigation Limited
# File units = meters

mtllib test03.mtl

g Mesh1 Model

usemtl m_024
v 1 0 0
vt 4.02933 0
vn 0 0 -1
v 0 0 0
vt 0 0
v 0 1 0
vt 0 -4.02933
v 1 1 0
vt 4.02933 -4.02933
f 1/1/1 2/2/1 3/3/1 4/4/1
```
More Examples

Surface

Textures
Texture Map from Image
Texture in Blender

• This is the tip of a very useful Iceberg

Credit where credit is due – YouTube video: “TEXTURE MAPPING For Absolute Beginners - Blender Tutorial”