Lecture 6
Texture Mapping

Dr. Lin ZHANG
School of Software Engineering
Tongji University
Spring 2012
Outline

• Overview
• Texture representations
• Texture mapping for mesh model
• Texture mapping for parametric surface
• Bump mapping
Overview

• Texture mapping
  • It is a method for adding detail, surface texture (a bitmap or raster image), or color to a computer-generated 3D model
  • Its application to 3D graphics was pioneered by Dr. Edwin Catmull in his Ph.D. thesis of 1974

Textured objects
Overview

• Effects of texture mapping

For more info on the computer artwork of Jeremy Birn see [http://www.3drender.com/jbirm/productions.html](http://www.3drender.com/jbirm/productions.html)
Overview

- Effects of texture mapping
Overview

3D model                Texture mapped model

Lin ZHANG, SSE, 2012
Overview

Texture mapped model

We need a function that associates each surface point with a 2D coordinate in the texture map

Texture map (2D image)

Lin ZHANG, SSE, 2012
Overview

Texture mapped model

For each point rendered, look up color in texture map

Texture map (2D image)
Outline

• Overview
• Texture representations
• Texture mapping for mesh model
• Texture mapping for parametric surface
What is texture?

• Easy to recognize, hard to define
What is texture?

- Provides information in the spatial arrangement of colors or intensities in an image
- Characterized by the spatial distribution of intensity levels in a neighborhood
- Repeating pattern of local variations in image intensity
- Often has some degree of randomness
- Cannot be defined for a point
Texture representations

- Some well defined mathematical functions can generate kinds of textures

\[ g(u, v) = \begin{cases} 
0 & |u \times c| + |v \times c| \text{ is odd} \\
1 & \text{otherwise}
\end{cases} \]

where \( c \) is a constant integer
Texture representations

• Some well defined mathematical functions can generate kinds of textures

E.g. 2, \[ f(u, v) = A \left( \cos(pu) + \cos(qv) \right) \]
where \( A \) is a random variable in \([0, 1]\), \( p \) and \( q \) are constants
Texture representations

• In most cases, textures are represented as 2D images
Outline

• Overview
• Texture representations
• Texture mapping for a mesh model
• Texture mapping for a parametric surface
Texture mapping

- Mapping a texture
  - Take points on the surface of an object
  - Return an entry in the texture
- In ray-casting
  - Ray cast pixel \((x, y)\), get visible point
  - Get texture coordinates \((u, v)\) at that point
  - Look up texture color using \(UV\) coordinates
Texture mapping

- Mapping a texture
- Take points on the surface of an object
- Return an entry in the texture
- In ray-casting
- Ray cast pixel \((x, y)\)
- Get visible point
- Get texture coordinates \((u, v)\) at that point
- Look up texture color using UV coordinates
Texture mapping for a mesh model

- Each vertex $P$ stores 2D $(u, v)$ “texture coordinates”
  - $UV$s determine the 2D location in the texture for the vertex
- Then we interpolate using barycentrics

$$(u_0, v_0)$$

$$(\alpha u_0 + \beta u_1 + \gamma u_2, \alpha v_0 + \beta v_1 + \gamma v_2)$$

$$(u_1, v_1)$$

$$(u_2, v_2)$$
Texture mapping for a mesh model

- Each vertex $P$ stores 2D $(u, v)$ “texture coordinates”
  - $UV$s determine the 2D location in the texture for the vertex
- Then we interpolate using barycentrics
Texture mapping for a mesh model

• Per-vertex \((u, v)\) “texture coordinates” are specified:
  • Manually, provided by user (tedious!)
  • Automatically using parameterization optimization
Texture mapping for a mesh model

- **Texture \((u, v)\) optimization**
  - **Goal**: “flatten” 3D object onto 2D UV coordinates
  - **For each vertex**, find coordinates \(u,v\) such that distortion is minimized
    - distances in \(UV\) correspond to distances on mesh
    - angle of 3D triangle same as angle of triangle in \(UV\) plane
  - **Cuts are usually required** (discontinuities)
Texture mapping for a mesh model

• To learn more
  • For this course, assume UV given per vertex
  • “Mesh Parameterization: Theory and Practice”, Kai Hormann, Bruno Lévy and Alla Sheffer, *ACM SIGGRAPH Course Notes, 2007* (available on the course website)

How about non-polygonal geometry?

• No vertices, can’t specify uv$s that way!
• Solution: parametric texturing
  • Deduce (u, v) from (x, y, z)
  • Various mappings are possible
Outline

• Overview
• Texture representations
• Texture mapping for a mesh model
• Texture mapping for a parametric surface
Texture mapping for a parametric surface

- It is easy and straightforward for texture mapping for parametric surfaces $S(u, v)$
  - Plane
  - Cylinder
  - Cone
  - Sphere
  - Torus
Texture mapping for a parametric surface

- It is easy and straightforward for texture mapping for parametric surfaces $S(u, v)$

E.g. 1,

$$\begin{cases} 
  x = r \cos u \\
  y = r \sin u, \, 0 \leq u < 2\pi, 0 \leq v \leq height \\
  z = v 
\end{cases}$$

Object space

$$\begin{cases} 
  u = \arctan \frac{2(y, x)}{2\pi} \\
  v = z / height 
\end{cases}$$

Texture space
Texture mapping for a parametric surface

- It is easy and straightforward for texture mapping for parametric surfaces $S(u, v)$

E.g. 1,
Texture mapping for a parametric surface

- It is easy and straightforward for texture mapping for parametric surfaces $S(u, v)$

Note: texture mapping can be combined with lighting models
Outline

• Overview
• Texture representations
• Texture mapping for a mesh model
• Texture mapping for a parametric surface
• Bump mapping
Bump mapping

- Bump mapping is a technique in computer graphics for simulating bumps and wrinkles on the surface of an object.
- This is achieved by perturbing the surface normals of the object and using the perturbed normal during lighting calculations.
- The result is an apparently bumpy surface rather than a smooth surface although the surface of the underlying object is not actually changed.
- Bump mapping was introduced by Blinn in 1978\textsuperscript{[1]}

Bump mapping

- Kinds of bump mapping schemes exist in the literature
- Normal mapping is the most popular one
  - It is a technique to light a 3D model with a low polygon count as if it was a more detailed model
  - It does not actually add any detail to the geometry, so the edges of the model will still look the same, however the interior will look a lot like the high-res model used to generate the normal map
  - The RGB values of each texel in the normal map represent the x,y,z components of the normalized mesh normal at that texel
  - Instead of using interpolated vertex normals to compute the lighting, the normals from the normal map texture are used
Bump mapping

• Normal mapping is the most popular one
• The procedure is similar to texture mapping
  • For each object point \( p = (x, y, z) \), find its texture correspondence point \((u, v)\)
  • Normalize the value \((R, G, B)\) at the position \((u, v)\) in the texture as \(n\), and take \(n\) as the normal of \(p\) when calculating lighting
• There are many variants to the above general steps
Bump mapping

- Normal mapping is the most popular one
Bump mapping

• Normal mapping is the most popular one

No bump mapping

With bump mapping
Bump mapping

- Normal mapping is the most popular one

No bump mapping  With bump mapping
Thanks for your attention