Textures

CS 4620 Lecture 22
Announcements

• Prelim grading this Friday
  – Will discuss it on Monday
  – Still few stragglers on taking the prelim
Bump mapping

- Perturbs normal based on input grayscale height field
Normal mapping

- Stores normals as texture map over coarse geometry

original mesh
4M triangles

simplified mesh
500 triangles

simplified mesh and normal mapping
500 triangles

[Paolo Cignoni]
Displacement mapping

- Base subdivision surface
- Hand-painted displacement map (detail)
- Displaced surface

Paweł Filip
tolas.wordpress.com
Bump mapping

- “Simulation of Wrinkled Surfaces” Blinn 78

- Blinn: keep surface, use new normals
Bump Mapping

• Alter normals of surface
  – Only affects shading normals

• Also, mimics effect of small scale geometry (detail)
  – Except at silhouette
  – Adds perceived bumps, wrinkles
Blinn’s Original Method

- Look up $b_u$ and $b_v$
- $N'$ is not normalized
Bump Mapping
Bump Mapping
Before Bump Mapping

• First, need some frame of reference
  – Normal is modified with respect to that
  – Have tangent space basis: t and b
  – Normal, tangent and bitangent vectors
Tangent Space Basis Vectors for an Arbitrary Mesh


- Find T and B for a triangle (a, b, c)
  - so that $Q - a = (u - u_0)T + (v - v_0)B$
  - T and B are tangent vectors aligned to the TM
Tangent Space Basis Vectors for an Arbitrary Mesh

- Find T and B for a triangle \((a,b,c)\)
  - so that \(Q - a = (u - u_0)T + (v - v_0)B\)
  - \(T\) and \(B\) are tangent vectors aligned to the TM

\[
egin{align*}
Q_1 &= (c - a), \quad Q_2 = (b - a) \\
(s_1, t_1) &= (u_1 - u_0, v_1 - v_0) \\
(s_2, t_2) &= (u_2 - u_0, v_2 - v_0)
\end{align*}
\]
Tangent Space Basis Vectors for an Arbitrary Mesh

\[ Q_1 = s_1 T + t_1 B \]
\[ Q_2 = s_2 T + t_2 B \]

\[
\begin{bmatrix}
Q^x_1 & Q^y_1 & Q^z_1 \\
Q^x_2 & Q^y_2 & Q^z_2
\end{bmatrix}
= \begin{bmatrix}
s_1 & t_1 \\
s_2 & t_2
\end{bmatrix}
\begin{bmatrix}
T^x & T^y & T^z \\
B^x & B^y & B^z
\end{bmatrix}
\]
Tangent Space Basis Vectors for an Arbitrary Mesh

• For tangent vectors of a single vertex
  – Average tangents for tris sharing the vertex
  – Same as normal

• \( B = N \times T \)

• \( N' = N + bu \ T + bv \ B = N + bu \ U + bv \ V \)
Blinn’s Original Method

- Look up $b_u$ and $b_v$
- $N'$ is not normalized
Rendering with Bump Maps

- $N'.L$
- Perturb $N$ to get $N'$ using bump map
- Transform $L$ to tangent space of surface
  - Have $N$, $T$ (tangent), bitangent $B = T \times N$
Transforming into this space

• Transform light vector into tangent space using following basis matrix

\[
\begin{bmatrix}
T_x & T_y & T_z & 0 \\
B_x & B_y & B_z & 0 \\
N_x & N_y & N_z & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Normal Maps

- Preferred technique for bump mapping for modern graphics cards
- Store new normals in texture map
  - Encodes \((x, y, z)\) mapped to \([-1, 1]\)
- More memory but lower computation

Normal Map

Height Map
• Store

\[ \text{colorComponent} = 0.5 \times \text{normalComponent} + 0.5 \]

• Use

\[ \text{normalComponent} = 2 \times \text{colorComponent} - 1 \]
Creating Normal Maps

Low-poly version

High-poly version

normals recorded at ray intersections

rays cast from low to high version

© Kavita Bala, Computer Science, Cornell University
Creating Normal Maps

• First create complex geometry

• Simplify (in modeling time) to simple mesh with normal map
Normal Map
Which space is normal map in?

- **World space**
  - Easy computation
    - Get normal
    - Get light vector
    - Compute shading
  - Can we use the same normal map for…
    - two walls
    - A rotating object

- **Object space**
  - Better, but cannot be reused for symmetric parts of object
Which space is normal?

• Tangent space normals
  – Can reuse for deforming surfaces
  – Transform lighting to this space and shade
Displacement and Bump/Normal Mapping

• Mimic effect of geometric detail/meso geometry
  – Also detail mapping
Displacement Mapping

\[ P_{new} = P_{old} + DM(u) \times \hat{N} \]
Displacement Maps
Displacement Maps: where?

Without Vertex Textures

With Vertex Textures

Images used with permission from Pacific Fighters. © 2004 Developed by 1C:Maddox Games. All rights reserved. © 2004 Ubi Soft Entertainment.
Texture Maps

• Most flexible part of graphics hardware
• Textures can modulate
  – Material
    ▪ Diffuse, Specular/roughness (gloss maps)
  – Geometry
    ▪ Positions
      • displacement mapping
    ▪ Normals
      • bump mapping, normal mapping
  – Lighting
    ▪ Environment mapping
    ▪ Reflection mapping
    ▪ Shadow mapping