Texture Mapping, Bump Mapping, OpenGL

CS116B

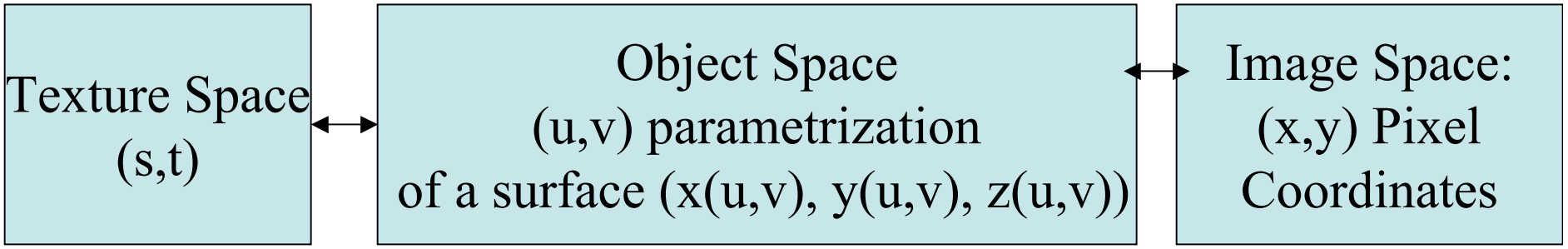
Chris Pollett

May 2, 2004.

Outline

- Surface Texture Mapping
- Texture Reduction Patterns
- Procedural Texturing Methods
- Bump Mapping
- Frame Mapping
- OpenGL

Surface Texture Mapping



- Typically, the texture space and the object space are rectangles so can map using the equations:

  $u(s,t) = a_u*s+b_u*t +c_u$

  $v(s,t) = a_v*s =b_v*t +c_u.$

- The object space to image space mapping depends on the surface we are parametrizing. For example, for a cylinder we might use x=r*cos u, y=r*cos u, z=v.
- We can map either from texture space to image space (**texture scanning**) by composition or can map reverse direction (**pixel-order scanning**).
- The latter is useful to avoid pixel round off errors.

Volume Texturing

- Similar to surface texturing except now texture is 3D. So given by 3 coordinates (s,t,r).
- Might want to do for cut-away displays, scenes like inside a fish-tank,etc.

Texture Reduction Patterns

- As objects get far away, it doesn't make sense to do lots of calculations to apply a texture to them.
- It also can cause distortion in how the texture looks.
- To avoid this we can create different textures of different levels of detail to use depending on the scale of the object.
- These texture reduction patterns are often called **MIP maps** (multum in parvo).

Procedural Texturing Methods

- Another technique for adding a texture pattern to an object is to use a procedural definition for the textures that are to be applied.
- That is have a little program that calculates something that looks like wood graining, marble, etc.

Bump Mapping

- Texture are not very effective when trying to model rough surfaces such as oranges, strawberries, or raisins.
- The problem is the light intensity given in a texture for such an object does not depend on the light in the scene. But these objects change a lot according to the lighting.
- **Bump mapping** is a technique to make realistic bumpy surfaces that can be used instead.

More Bump Mapping

- Let **P**(u,v) be a point on a surface.
- Then $\mathbf{N} = \mathbf{P}_u \times \mathbf{P}_v$ is the normal at (u,v). Let $\mathbf{n} = \mathbf{N}/|\mathbf{N}|$.
- We can add a bump to the surface using an equation: $\mathbf{P}'(u,v) = \mathbf{P}(u,v) + b(u,v)\mathbf{n}$.
- Here b(u,v) is a bump function.
- Can show the perturbed normal is now approximately:

  $\mathbf{N}' = \mathbf{N} + b_v(\mathbf{P}_u \times \mathbf{n}) + b_u(\mathbf{P}_v \times \mathbf{n})$
- We now use this normal to do our lighting calculations.

Frame Mapping

- This is an extension to bump mapping.
- We not only perturb the surface normal, we also perturb the local coordinate system at the point.
- To do this we tweak the tangent vector **T** and calculate a binormal as **B= T** x **N**.
- This is useful in modeling anisotropic surfaces, such as wood grains, cross threading in clothing, and streaks on marbles.