Texture Mapping

CS 417 Lecture 25

Texture mapping

• Objects have properties that vary across the surface

Texture Mapping

• So we make the shading parameters vary across the surface

Texture mapping

• Adds visual complexity; makes appealing images

Texture mapping

• Color is not the same everywhere on a surface
  – one solution: multiple primitives
• Want a function that assigns a color to each point
  – the surface is a 2D domain, so that is essentially an image
  – can represent using any image representation
  – raster texture images are very popular

A definition

Texture mapping: a technique of defining surface properties (especially shading parameters) in such a way that they vary as a function of position on the surface.

• This is very simple!
  – but it produces complex-looking effects
Examples

- Wood gym floor with smooth finish
  - diffuse color $k_D$ varies with position
  - specular properties $k_S$, $n$ are constant
- Glazed pot with finger prints
  - diffuse and specular colors $k_D$, $k_S$ are constant
  - specular exponent $n$ varies with position
- Adding dirt to painted surfaces
  - Simulating stone, fabric, …
  - in many cases textures are used to approximate effects of small-scale geometry
  - they look flat but are a lot better than nothing

Mapping textures to surfaces

- The big question of texture mapping: where on the surface does the image go?
  - for a flat rectangle the same shape as the image, there is an obvious answer
  - for other geometry things become more interesting
- “Putting the image on the surface”
  - this means we need a function that tells where each point on the image goes
  - when we use raster texture images the pixels in those images are often called “texels”
  - the mapping function tells us where each texel goes

Texture coordinate functions

- The mapping from texture to surface must be invertible
  - that is, every surface point gets only one color assigned
  - it is OK (and in fact useful) for multiple surface points to be mapped to the same texture point
    - e.g. repeating tiles
- The function we actually use is this inverse:
  $$ \phi : S \rightarrow \mathbb{R}^2 $$
  - this tells us where in the texture we should look to get the color for a particular surface point

Examples of coordinate functions

- A rectangle
  - image can be mapped directly, unchanged

Examples of coordinate functions

- For a sphere: latitude-longitude coordinates
  - $\phi$ maps point to its latitude and longitude
Examples of coordinate functions

- A parametric surface (e.g. spline patch)
  - surface parameterization gives mapping function directly
    (well, the inverse of the parameterization)

Examples of coordinate functions

- For non-parametric surfaces it is trickier
  - directly use world coordinates
    - need to project one out

Examples of coordinate functions

- For non-parametric surfaces it is trickier
  - directly use world coordinates
  - use intermediate parametric object

Examples of coordinate functions

- For non-parametric surfaces it is trickier
  - directly use world coordinates
  - use intermediate parametric object
Examples of coordinate functions

- For non-parametric surfaces it is trickier
  - directly use world coordinates
  - use intermediate parametric object

Examples of coordinate functions

- For non-parametric surfaces it is trickier
  - directly use world coordinates
  - use intermediate parametric object

Texture in the graphics pipeline

- Texture coordinates for a triangle
- Interpolating coordinates over meshes
- Texturing as a fragment operation