Texture Mapping

From: Ed Angel
University of New Mexico

Objectives

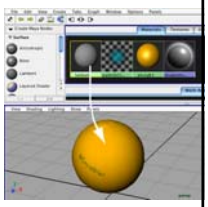
- Introduce Mapping Methods
  - Texture Mapping
  - Environment Mapping
  - Bump Mapping
- Consider basic strategies
  - Forward vs backward mapping
  - Point sampling vs area averaging

The Limits of Geometric Modeling

- Although graphics cards can render over 10 million polygons per second, that number is insufficient for many phenomena
  - Clouds
  - Grass
  - Terrain
  - Skin

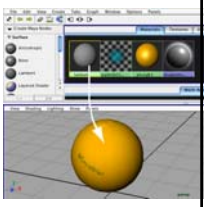
Modeling an Orange

- Consider the problem of modeling an orange (the fruit)
- Start with an orange-colored sphere
  - Too simple
- Replace sphere with a more complex shape
  - Does not capture surface characteristics (small dimples)
  - Takes too many polygons to model all the dimples



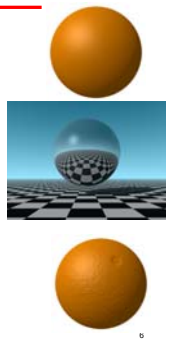
Modeling an Orange (2)

- Take a picture of a real orange, scan it, and "paste" onto simple geometric model
  - This process is known as texture mapping
- Still might not be sufficient because resulting surface will be smooth
  - Need to change local shape
  - Bump mapping



Three Types of Mapping

- Texture Mapping
  - Uses images to fill inside of polygons
- Environment (reflection mapping)
  - Uses a picture of the environment for texture maps
  - Allows simulation of highly specular surfaces
- Bump mapping
  - Emulates altering normal vectors during the rendering process



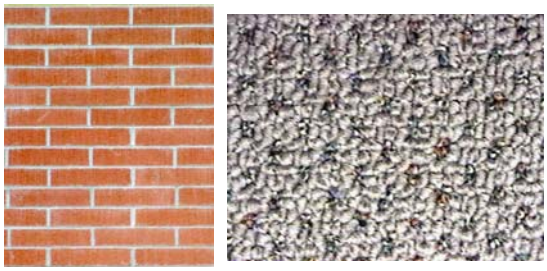
Texture mapping

- Texture mapping: adding surface detail by mapping texture patterns to the surface
- Developed by Catmull (1974), Blinn and Newell (1976), and others

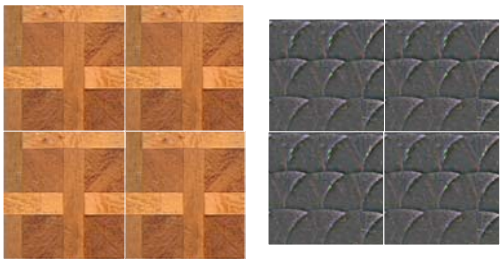
Texture Mapping

- Maps a pattern (texture) onto a surface
- *Texels* fill each pixel
- Texels selected from sample pattern (*texture map*)
- Pattern is repeated

Texture Maps

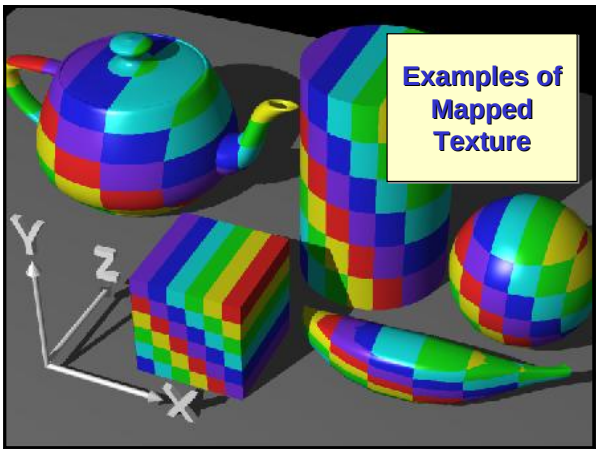


Texture Maps

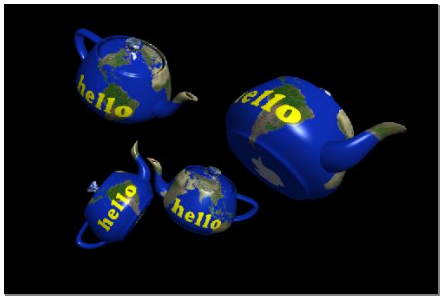


Wallpaper, Analogue Texture Map

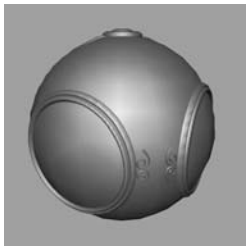
Flooring, Tiling, etc



Examples



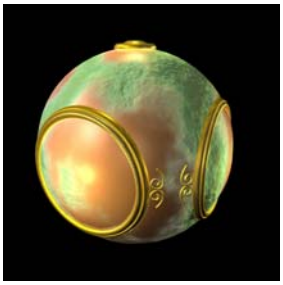
Texture Mapping



geometric model texture mapped

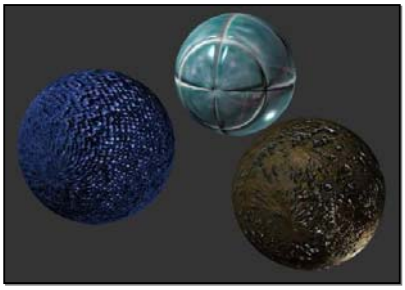
Environment Mapping

Bump Mapping

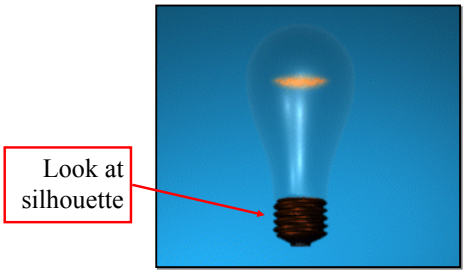


Bump Mapping

Look at smooth silhouettes

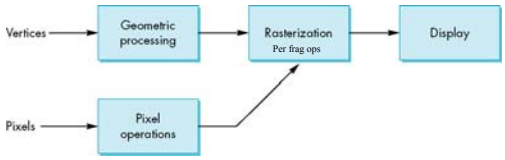


Displacement Mapping



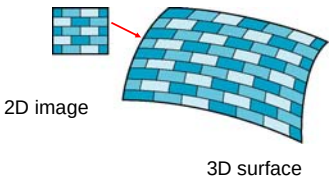
Where does mapping take place?

- Mapping techniques are implemented at the end of the rendering pipeline
  - Very efficient because few polygons make it past the clipper



Is it simple?

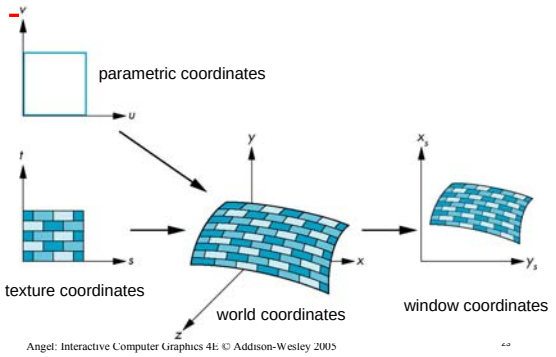
- Although the idea is simple---map an image to a surface---there are 3 or 4 coordinate systems involved



Coordinate Systems

- Parametric coordinates
  - May be used to model curves and surfaces
- Texture coordinates
  - Used to identify points in the image to be mapped
- Object or World Coordinates
  - Conceptually, where the mapping takes place
- Window Coordinates
  - Where the final image is really produced

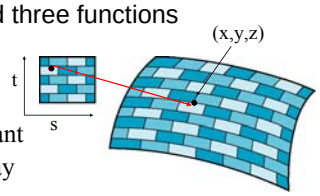
Texture Mapping



Mapping Functions

- Basic problem is how to find the maps
- Consider mapping from texture coordinates to a point a surface
- Appear to need three functions

  x = x(s,t)
  y = y(s,t)
  z = z(s,t)



- But we really want to go the other way

Backward Mapping

- We really want to go backwards
  - Given a pixel, we want to know to which point on an object it corresponds
  - Given a point on an object, we want to know to which point in the texture it corresponds
- Need a map of the form

  s = s(x,y,z)

  t = t(x,y,z)
- Such functions are difficult to find in general

Basic Concept

- Relate a $2D$ image to a $3D$ model
- *Texture coordinates*
  - Texture coordinate is a $2D$ coordinate (u,v) which maps to a location on a texture map
- Texture coordinates are over the interval $[0,1]$, typically

Elements of Texture Mapping

- Texture source function (can be $3D$)
- Inverse map:

  $Texture\ (u,v) \leftarrow Surface\ (x(s,t), y(s,t), z(s,t))$
- Typical texture sources
  - Procedure
  - Tabular data (texture map)

Texture Mapping Techniques

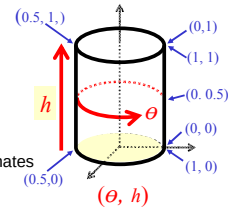
- $2D$ texture mapping: *paint* $2D$ pattern onto the surface
- *Environmental* (reflection) *mapping*
- *Bump mapping*: perturb surface normals to fool shading algorithms
- Procedural texture mapping, $3D$ texture

Cylindrical Mapping

parametric cylinder

x = r cos 2π u
y = r sin 2πu
z = v*h

maps rectangle in u,v space to cylinder of radius θ and height h in world coordinates



s = u
t = v

maps from texture space

Spherical Map

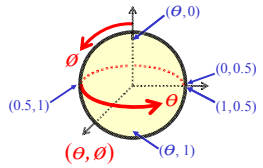
We can use a parametric sphere

x = r cos 2πu
y = r sin 2πu cos 2πv
z = r sin 2πu sin 2πv

in a similar manner to the cylinder but have to decide where to put the distortion

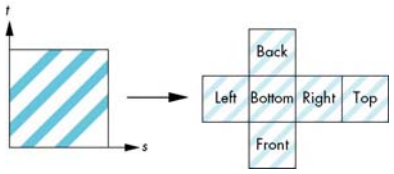
Spheres are used in environmental maps

Spherical Map



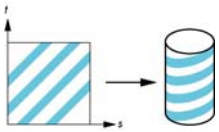
Box Mapping

- Easy to use with simple orthographic projection
- Also used in environment maps



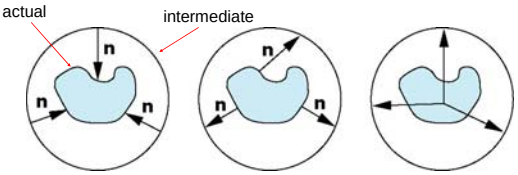
Two-part mapping

- One solution to the mapping problem is to first map the texture to a simple intermediate surface
- Example: map to cylinder

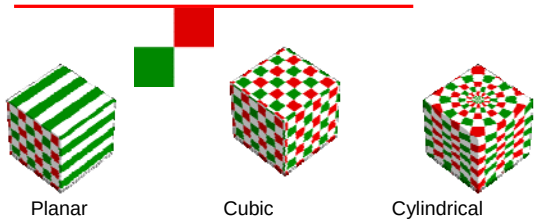


Second Mapping

- Map from intermediate object to actual object
  - Normals from intermediate to actual
  - Normals from actual to intermediate
  - Vectors from center of intermediate



More Examples



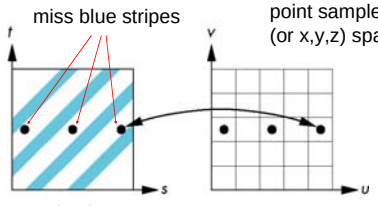
Planar Cubic Cylindrical

Texture Mapping, Paul Bourke (1987)

http://astronomy.swin.edu.au/~pbourke/texture/texturemapping

Aliasing

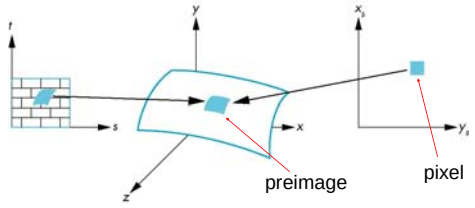
- Point sampling of the texture can lead to aliasing errors



point samples in texture space

Area Averaging

A better but slower option is to use *area averaging*



Note that *preimage* of pixel is curved