Texture Mapping
Limits of Geometric Modeling

Although graphics cards can render over 133 million polygons per second (top of the line NVIDIA card), 2-billion triangles per second, that number is insufficient for many phenomena at real-time rates:

- Clouds
- Grass
- Terrain
- Skin
Modeling an Orange

Start with an orange-colored sphere?

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

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
Bump-Mapped orange

http://en.wikipedia.org/wiki/Bump_mapping
Half-Life 2

GTA 5
Several Forms of Mapping

- Texture
- Bump
- Environment
Texture Mapping

Most General:

Texture is a table of values that you can query in a shader and use for any purpose you desire!

Texture Creation:

Drawn
Procedurally generated
Rendered and re-read
...

Oregon State University
Although the idea is simple---map an image to a surface---there are potentially several coordinate systems involved.
Texture mapping coordinate systems

World Coordinates

  Conceptually, where the mapping takes place

Texture coordinates

  Used to identify points in the image to be mapped

Screen Coordinates

  Where the final image is really produced
Texture Mapping Process

3D Polygon with Texture coordinate values

t
s

y

x

z

texture coordinates

world coordinates

screen coordinates

Angel Figure 7.9
Basic problem is how to find the mappings

Consider mapping from texture coordinates to a point on a surface. Appears to require three functions

\[ x = f_x(s,t) \]
\[ y = f_y(s,t) \]
\[ z = f_z(s,t) \]

But we really want to go the other way
Backward Mapping

We really want to go backwards (twice)

1. Given a fragment, we want to know to which point on an object it corresponds to (Screen to World/Object)

2. Given a point on an object, we want to know to which point in the texture it corresponds to (World/Object to texture)

For 2, Need a map of the form

\[ s = f_s(x,y,z) \]

\[ t = f_t(x,y,z) \]
One solution to the mapping problem is to first map the texture to a simple intermediate surface.
Example: map to cylinder.
Object to Texture

Map from intermediate object to actual object
- Normals from intermediate to actual
- Normals from actual to intermediate
- Vectors from center of intermediate
3D Scanner
Other Methods for Object to Texture

3D Paint

Paint directly on the geometry, that is already parameterized, unfold the geometry onto a plane.
Texture Mapping Process

Assuming you’ve already assigned a texture coordinate to each vertex...When rasterizing fragment x,y

1. Compute *interpolated* texture coordinate s,t for the fragment
2. Look up s,t in the corresponding texture for that primitive
3. Use the color at texture location s,t as the color of the material at that fragment ...then compute illumination!
Point sampling (pixels) to map to the texture \((s,t)\) can lead to aliasing errors.
A better but slower option is to use *area averaging*

Note that *preimage* of pixel is curved
Angel Figure 7.22
Alternative Approach: Mip-Mapping

Idea: Pre-filtering

• Create a pyramid of filtered versions of the image
• Sample the appropriate level or interpolated level
Texture Demo