Texture Mapping

Reading

Required
♦ Angel, 8.6, 8.7, 8.9, 8.10, 9.13–9.13.2

Recommended

Optional
♦ Woo, Neider, & Davis, Chapter 9

Texture mapping

Texture mapping allows you to take a simple polygon and give it the appearance of something much more complex.
♦ Due to Ed Catmull PhD thesis, 1974
♦ Refined by Blinn & Newell, 1976

Texture mapping ensures that “all the right things” happen as a textured polygon is transformed and rendered.

Non-parametric texture mapping

With “non-parametric texture mapping”:
♦ Texture size and orientation are fixed
♦ They are unrelated to size and orientation of polygon
♦ Gives cookie-cutter effect
Parametric texture mapping

With "parametric texture mapping," texture size and orientation are tied to the polygon.

Ideas:
- Separate "texture space" and "screen space"
- Texture the polygon as before, but in texture space
- Deform (render) the textured polygon into screen space

A texture can modulate just about any parameter—diffuse color, specular color, specular exponent ...

Implementing texture mapping

A texture lives in its own abstract image coordinates parameterized by \((u,v)\) in the range \([0..1],[0..1]\):

It can be wrapped around many different surfaces:

\[ Q = a_1 v + b_1 + c_1 \]

Completing \((u,v)\) texture coordinates in array tracer is fairly straightforward.

Note: if the surface moves/deforms, the texture goes with it.

Mapping to texture image coords

The texture is usually stored as an image. Thus, we need to convert from abstract texture coordinate:

\((u,v)\) in the range \([0..1],[0..1]\)

to texture image coordinates:

\((i_{text}, j_{text})\) in the range \([0..w_{text}],[0..h_{text}]\)

Mapping to abstract texture coords

Mapping to texture pixel coords

Q: What do you do when the texture sample you need lands between texture pixels?

Texture resampling

We need to resample the texture:

A common choice is \textbf{bilinear interpolation}:

\[ T(i,j) = T(i+\Delta_x,j+\Delta_y) \]

\[ \Delta_x = \frac{i+\Delta_x}{w_{text}} \]

\[ \Delta_y = \frac{j+\Delta_y}{h_{text}} \]
Displacement mapping

Textures can be used for more than just color.

In displacement mapping, a texture is used to perturb the surface geometry itself.

\[ \hat{\mathbf{Q}}(u) = \mathbf{Q}(u) + d(u) \mathbf{N}(u) \]

- These displacements "animate" with the surface

Q: Do you have to do hidden surface calculations on \( \hat{\mathbf{Q}} \)?

Yes

Bump mapping

In bump mapping, a texture is used to perturb the normal:

- Use the original, simpler geometry, \( \mathbf{Q}(u) \), for hidden surfaces
- Use the normal from the displacement map for shading:

\[ \mathbf{N} = \text{normal}[\mathbf{Q}(u)] \]

Q: What artifacts in the images would reveal that bump mapping is a failure?

Silhouettes, no self-shadowing, incorrect shading, casting and other objects

Displacement vs. bump mapping

Input texture

Randomized displacement map over a rectangular surface

Original rendering

Rendering with bump map wrapped around cylinder

Bump map and rendering by Wyvem Aldinger
Solid textures

Q: What kinds of artifacts might you see from using a marble veneer instead of real marble?

One solution is to use solid textures:

- Use model-space coordinates to index into a 3D texture
- Like "carving" the object from the material

One difficulty of solid texturing is coming up with the textures.

Solid textures (cont’d)

Here’s an example of a vase cut from a solid marble texture:

Solid marble texture by Ken Radin, (Friday, M-21)

Solid textures (cont’d)

\[
\text{in}(x,y,z) = \begin{cases} \text{stripes}(x) & \text{if } K > \text{noise}(x,y,z) \\ \text{stripes}(x + \text{shift}(x,y,z)) & \text{otherwise} \end{cases}
\]

Increasing \(K\)

Environment mapping

In environment mapping (also known as reflection mapping), a texture is used to model an object’s environment:

- Rays are bounced off objects into environment
- Color of the environment used to determine color of the illumination
- Really, a simplified form of ray tracing
- Environment mapping works well when there is just a single object — or in conjunction with ray tracing

Under simplifying assumptions, environment mapping can be implemented in hardware.

With a ray tracer, the concept is easily extended to handle refraction as well as reflection.
Summary

What to take home from this lecture:

1. The meaning of the boldfaced terms.

2. Familiarity with the various kinds of texture mapping, including their strengths and limitations.