Texture Mapping & Other Fun Stuff

Last Time?
- Distribution Ray Tracing
- Bounding Boxes
- Spatial Acceleration Data Structures
  - Regular Grid
  - Adaptive Grids
  - Hierarchical Bounding Volumes
- Flattening the Transformation Hierarchy

Regular Grid Discussion
- Advantages?
  - easy to construct
  - easy to traverse
- Disadvantages?
  - may be only sparsely filled
  - geometry may still be clumped

Adaptive Grids
- Subdivide until each cell contains no more than \( n \) elements, or maximum depth \( d \) is reached

Primitives in an Adaptive Grid
- Can live at intermediate levels, or be pushed to lowest level of grid

Adaptive Grid Discussion
- Advantages?
  - grid complexity matches geometric density
- Disadvantages?
  - more expensive to traverse (especially octree)
Bounding Volume Hierarchy
- Find bounding box of objects
- Split objects into two groups
- Recurse

Where to split objects?
- At midpoint  OR
- Sort, and put half of the objects on each side  OR
- Use modeling hierarchy
Intersection with BVH

• Check sub-volume with closer intersection first

Intersection with BVH

• Don't return intersection immediately if the other subvolume may have a closer intersection

Bounding Volume Hierarchy Discussion

• Advantages
  – easy to construct
  – easy to traverse
  – binary

• Disadvantages
  – may be difficult to choose a good split for a node
  – poor split may result in minimal spatial pruning

Questions?

Today

• 2D Texture Mapping
  – Perspective Correct Interpolation
  – Specifying Texture Coordinates
  – Illumination & Reflectance
• Procedural Solid Textures
• Other Mapping Techniques
• Texture Aliasing

The Problem:

• We don't want to represent all this detail with geometry
The Quest for Visual Realism

Texture Mapping
- Increase the apparent complexity of simple geometry
- Like wallpapering or gift-wrapping with stretchy paper
- Curved surfaces require extra stretching or even cutting

Texture Tiling
- Specify a texture coordinate (u,v) at each vertex
- Canonical texture coordinates (0,0) → (1,1)

Texture Interpolation
- Specify a texture coordinate (u,v) at each vertex
- Can we just linearly interpolate the values in screen space?

Interpolation - What Goes Wrong?
- Linear interpolation in screen space:
Specify More Coordinates?

- We can reduce the perceived artifacts by subdividing the model into smaller triangles.
- However, sometimes the errors become obvious
  - At "T" joints
  - Between levels of detail (mipmapping... in a few weeks)

Subdivision

- Notice that uniform steps on the image plane do not correspond to uniform steps along the edge.

Linear Interpolation in Screen Space

\[ p(t) = p_1 + t(p_2 - p_1) = \frac{3}{z_1} + t\left(\frac{3}{z_2} - \frac{3}{z_1}\right) \]

interpolation in 3-space

\[ \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = s \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \left(1 - s\right) \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = x_1 + s(x_2 - x_1) \\ y_1 + s(y_2 - y_1) \\ z_1 + s(z_2 - z_1) \]

Perspective Correct Interpolation

We need a mapping from \( t \) values to \( s \) values:

\[ \frac{z_1 + t(z_2 - z_1)}{z_1 + t(z_2 - z_1)} \]

Solve for \( s \) in terms of \( t \):

\[ s = \frac{t z_1}{z_1 + t(z_2 - z_1)} \]

Unfortunately, at this point in the pipeline (after projection) we no longer have \( z \). However, we do have \( w_1 = 1/z_1 \) and \( w_2 = 1/z_2 \), so:

\[ s = \frac{w_1}{w_1 + t(w_2 - w_1)} \]
Today

- 2D Texture Mapping
  - Perspective Correct Interpolation
  - Specifying Texture Coordinates
  - Illumination & Reflectance
- Procedural Solid Textures
- Other Mapping Techniques
- Texture Aliasing

Texture Mapping Difficulties

- Tedious to specify texture coordinates
- Acquiring textures is surprisingly difficult
  - Photographs have projective distortions
  - Variations in reflectance and illumination
  - Tiling problems

Common Texture Coordinate Mappings

- Orthogonal
- Cylindrical
- Spherical
- Perspective Projection
- Texture Chart

Projective Textures

- Use the texture like a slide projector
- No need to specify texture coordinates explicitly
- A good model for shading variations due to illumination
- A fair model for reflectance (can use pictures)

Projective Texture Example

- Modeling from photographs
- Using input photos as textures

Texture Mapping & Illumination

- Texture mapping can be used to alter some or all of the constants in the illumination equation:
  - pixel color, diffuse color, alter the normal, ….

\[
\mathbf{L}_{\text{color}} = \mathbf{K}_d \mathbf{L}_{\text{ambient}} + \sum_{\text{g}} \left( \mathbf{K}_d \left( \mathbf{N} \cdot \mathbf{L} \right) + \mathbf{K}_s \left( \mathbf{L} \cdot \mathbf{R} \right)^m \right)
\]

Phong’s Illumination Model

Can’t do this!
Texture Chart

• Pack triangles into a single image

Today

• 2D Texture Mapping
• Procedural Solid Textures
• Other Mapping Techniques
• Texture Aliasing

Procedural Textures

• Advantages:
  – easy to implement in ray tracer
  – more compact than texture maps
    (especially for solid textures)
  – infinite resolution

• Disadvantages
  – non intuitive
  – difficult to match existing texture

Questions?

Procedural Textures

f (x,y,z) → color

Questions?
Today

- 2D Texture Mapping
- Procedural Solid Textures
- Other Mapping Techniques:
  - Bump Mapping
  - Displacement Mapping
  - Environment Mapping (for Reflections)
  - Light Maps (for Illumination)
- Texture Aliasing

What's Missing?

- What's the difference between a real brick wall and a photograph of the wall texture mapped onto a plane?
- What happens if we change the lighting or the camera position?

Remember Gouraud Shading?

- Instead of shading with the normal of the triangle, shade the vertices with the average normal and interpolate the color across each face

Phong Normal Interpolation (Not Phong Shading)

- Interpolate the average vertex normals across the face and compute per-pixel shading

Must be renormalized

Bump Mapping

- Use textures to alter the surface normal
  - Does not change the actual shape of the surface
  - Just shaded as if it were a different shape
Another Bump Map Example

What's Missing?
- There are no bumps on the silhouette of a bump-mapped object
- Bump maps don’t allow self-occlusion or self-shadowing

Displacement Mapping
- Use the texture map to actually move the surface point
- The geometry must be displaced before visibility is determined

Displacement Mapping
Image from: Geometry Caching for Ray-Tracing Displacement Maps by Matt Pharr and Pat Hanrahan.

Today
- 2D Texture Mapping
- Procedural Solid Textures
- Other Mapping Techniques:
  - Projective Shadows and Shadow Maps
  - Bump Mapping
  - Displacement Mapping
  - Environment Mapping (for Reflections)
  - Light Maps (for Illumination)
- Texture Aliasing
Environment Maps

- We can simulate reflections by using the direction of the reflected ray to index a spherical texture map at “infinity”.
- Assumes that all reflected rays begin from the same point.

Environment Mapping Example

Texture Maps for Illumination

- Also called "Light Maps"

Questions?

Image by Henrik Wann Jensen
Environment map by Paul Debevec

Today

- 2D Texture Mapping
- Procedural Solid Textures
- Other Mapping Techniques:
  - Texture Aliasing
Textures can Alias

- **Aliasing** is the under-sampling of a signal, and it's especially noticeable during animation.

Next Time:

**Real-Time Shadows**