Texture Mapping

• Technique to add detail to an image without requiring modeling detail
  • Enhances visual richness while entailing only a relatively small increase in computation

• Originally developed to add the appearance of surface complexity when rendering geometric surfaces

• Texture mapping capabilities for volume rendering are becoming more common

Texture Mapping

• Analogy is pasting a picture to the surface of an object

• Must have a texture map (picture to be pasted) and texture coordinates

• Can think of it as a custom lookup table for color, intensity and/or transparency that is applied to an object as it is rendered

Texture Mapping

• When creating image detail, it is cheaper to employ mapping techniques than to use numerous tiny polygons.

• The image on the right portrays a brick wall, a lawn and the sky. In fact, the wall was modeled as a rectangular solid, and the lawn and the sky were created from rectangles.

• The entire image contains eight polygons.

• Imagine the number of polygons it would require to model the blades of grass in the lawn! Texture mapping creates the appearance of grass without the cost of rendering thousands of polygons.

Texture Mapping Process

• Typically associated with each vertex in the geometry is position, normal, color, and other point attributes (can add texture map values too)

• Alternatively, define a procedure to define texture map values (does not take advantage of dedicated hardware)

• Can also move a texture map across a surface to generate animation
  • Choose a texture map whose intensity varies monotonically from dark to light, and then move the texture along an object gives appearance of animation
Texture Map

- Texel (texture element) is a pixel in the texture map
- Can have 3-component texture maps with each texel representing a RGB
- Can add alpha values resulting in 4-component texture map (RGBA)
- Apply texture map onto surface of underlying geometry
  - One approach is to ignore the original color of the surface
  - Another approach is blend with the surface color

Texture Mapping

- Texture mapping can be divided into 2D and 3D techniques
  - 2D techniques place a two-dimensional (flat) image onto an object using methods similar to pasting wallpaper onto an object
  - 3D techniques are analogous to carving the object from a block of marble
- The source image (texture) is mapped onto a surface in the 3D object space, which is then mapped to the image (screen) by the viewing projection. Texture space is labeled \((u, v)\) and object space is labeled \((x_0, y_0, z_0)\) while screen space is labeled \((x, y)\).

Photo Textures

- To this point we have seen textures used as labels over geometry. Their applications are much broader. **Simple extensions include:**
  - Spatially varying surface properties:
    - The diffuse shading coefficient, \(k_d\), could be stored as a texture.
    - Likewise for the intrinsic color, \(k_a, k_s\), and shininess.
  - Textures can also be used in a lot of other ways
Texture mapping can be used to alter some or all of the constants in the illumination equation. We can simply use the texture as the final color for the pixel, or we can just use it as diffuse color, or we can use the texture to alter the normal, or... the possibilities are endless!

\[ L = k_d \cdot \sum L_i + k_r \cdot \sum (L_i \cdot n_i) \cdot \max(0, |L_i \cdot n|) \]

Textures can be used to alter the surface normal of an object. This does not actually change shape of the surface—we are only shading it as if it were a different shape! This technique is called bump mapping. The texture map is treated as a single-valued height function. The value of the function is not actually used, just its partial derivatives. The partial derivatives tell how to alter the true surface normal at each point on the surface to make the object appear as if it were deformed by the height function.

Since the actual shape of the object does not change, the silhouette edge of the object will not change. Bump Mapping also assumes that the illumination model is applied at every pixel (as in ray tracing).

More Bump Map Examples

Quake uses light maps in addition to texture maps. Texture maps are used to add detail to surfaces, and light maps are used to store pre-computed illumination. The two are combined together at run-time, and cached for efficiency.

Quakes w/Diffuse Texture & Bump Map

Textures Only

Textures + Light Maps

Simple VTK Example with Geometric Modeling
Texture-based Volume Rendering

- Volume rendering by ray casting is time-consuming
  - one ray per pixel
  - each ray involves tracking through volume calculating samples, and then compositing
  - different for each viewpoint
- Alternative approach - using texture maps which can exploit graphics hardware if available

Volume Rendering with Textures

- 2D exploits existing hardware
- 3D is still expensive due to limited hardware availability

Texture-Mapped Volume Rendering

1. Data samples are extracted from volume using some form of interpolation if necessary (can be performed in hardware)
2. Blending is performed where the samples are combined with the current image in the frame buffer (e.g., compositing or maximum intensity)

2D Texture-based Volume Rendering

- Decompose dataset into orthographic slides along the axis of the volume most parallel to the viewing direction
- Each image mapped to a slice of the data
2D Texture Mapping

- As an alternative to ray casting, texture map the volume onto polygons
- Draw polygons in back-to-front order, blend using transparency function

Performance Analysis

1. Software sampling of the volume
   - required to create texture image
   - dependent on the view direction
2. Texture download to hardware
   - transfer rate from main memory to texture hardware memory
3. Texture-mapped polygon scan conversion rate
   - rate at which hardware can process pixels in the image

2D Texture-based Volume Rendering

- View volume as set of slices parallel to coordinate planes
- Pre-compute the textured planes for each of the 3 directions and save in memory to minimize software sampling cost
- Choose the orientation best suited to viewing direction

3D Texture-based Volume Rendering

- For a given viewing direction, select slices perpendicular to this direction
- This requires interpolation to get the values on the slices
- 3D texture hardware can load and interpolate between multiple slices in the volume; while 2D hardware cannot
- If 3D hardware has extensive memory then entire volume can be downloaded

Texture-based Volume Rendering

- Set of equally spaced planes along the viewing direction are clipped against the volume
- Draw from back-to-front the set of polygons
  - first polygon drawn as an area of colored pixels, with associated opacity, as determined by transfer function and interpolation
  - and merged with background in a compositing operation (supported by 3D texture hardware)
  - successive polygons drawn on top

2D versus 3D Hardware

- 2D much cheaper and more common
- 3D hardware is superior in its ability to sample the volume which generally results in higher quality images
3D Texture-Mapping Versus Ray Casting

- In principle, the two techniques are equivalent:
  - Both sample the entire volume using either nearest neighbor or trilinear interpolation
  - Both combine samples to form a pixel value using compositing or max value
- Texture mapping can take advantage of 3D hardware for sampling and blending (faster than software only approach for same generation platform)
- Ray casting typically produces less artifacts since it uses main memory with more precision for intermediate computation than dedicated hardware, which has a limited precision frame buffer
- Ray casting typically has more ray function flexibility and support for advanced techniques that include shading as compared to dedicated hardware
- 3D Texture-Mapping hardware engineers are working to eliminate its shortcomings

Summary of Volume Rendering Techniques (discussed)

- Ray casting
  - high quality, computationally expensive
- Texture-based & Shear-warp
  - Both lower quality, but make use of hardware accelerators