Texture-based Volume Rendering

- Volume rendering by **2D texture** mapping:
  - use planes parallel to **base plane** (front face of volume which is "most orthogonal" to view ray). **This is an axis-aligned approach**!
  - draw textured rectangles, using **bilinear** interpolation filter
  - render back-to-front, using α-blending for the α-compositing

Image credit: H.W. Shen, Ohio State U.
Texture in Graphics

Texture mapping can large enhance the reality of the 3D objects.

How does it work?

For each fragment: interpolate the texture coordinates (barycentric)

Texture-Lookup: interpolate the texture color (bilinear)

Image source: google image

[EuroGraphics 2006 Tutorial]
In OpenGL

Setting texture environment in the initialization **before** drawing

```c
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP );

``` Wrapping

```c
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, filter );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, filter );

``` Filter

```c
glTexEnvf( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE );

``` Environment

Put this setting before the code you do the texture mapping!
OpenGL Texturing

Define the texture wrapping parameters. This will control what happens when a texture coordinate greater than 1.0 or less than 0.0 is encountered:

```c
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, wrap );
glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, wrap );
```

where `wrap` is `GL_REPEAT` or `GL_CLAMP`
OpenGL Texturing

Define the texture filter parameters. This will control what happens when a texture is scaled up or down:

```c
    glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, filter);
    glTexParameterf( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, filter);
```

where `wrap` is either `GL_NEAREST` or `GL_LINEAR`
OpenGL Texturing

Define the texture environment properties.

\[ \text{glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, mode);} \]

where \text{mode} is either

\text{GL_REPLACE} specifies that the 3-component texture will be applied as an opaque image on top of the polygon, replacing the polygon’s specified color

\text{GL_MODULATE} specifies that the 3-component texture will be applied as piece of colored plastic on top of the polygon. The polygon’s specified color “shines” through the plastic texture. This is very useful for applying lighting to textures: paint the polygon white with lighting and let it shine up through a texture.
OpenGL Texturing

In your Display() function:

1. If you are using multiple textures, set the current texture using `glTexImage2D()` or by binding the correct one.

2. Enable texture mapping
   
   ```c
   glEnable(GL_TEXTURE_2D);
   ```

3. Draw your polygons, specifying texture coordinate \((s, t)\) before each vertex.
   
   ```c
   glBegin(GL_POLYGON);
   glTexCoord2f(s0, t0);
   glNormal3f (nx0, ny0, nz0);
   glVertex3f (x0, y0, z0);
   
   glEnd();
   ```

4. Disable texture mapping
   
   ```c
   glDisable(GL_TEXTURE_2D);
   ```
Texture-based Volume Rendering

- Volume rendering by **3D texture** mapping:
  - use the voxel data as the 3D texture
  - render an arbitrary number of slices (eg. 100 or 1000) **parallel to image plane** (3- to 6-sided polygons)
  - back-to-front compositing as in 2D texture method

Limited by size of texture memory.

Image credit: H.W.Shen, Ohio State U.
Slicing

object (color, opacity)

Similar to ray-casting with simultaneous rays
Effect of the Sample Rate

1 slice

5 slices

20 slices

45 slices

85 slices

170 slices

View direction
Slice Based Problems?

- Does not perform correct
  - Illumination
  - Accumulation - but can get close

- Can not easily add correct illumination and shadowing
  - See the Van Gelder paper for their addition for illumination
    - Stored in LUT quantized normal vector directions
Computational Strategies

• How can the basic ingredients be combined:
  • Image Order (in screen coordinate)
    • Ray casting (many options)
  • Object Order (in world coordinate)
    • splatting, texture-mapping
  • Combination (neither)
    • Shear-warp, Fourier
Additional Reading

For Ray casting


• *Data Visualization, Principles and Practice, Chapter 10 Volume Visualization*, by A. Telea, AK Peters, 2008

For splatting, please see,

• *Data Visualization, Principles and Practice, Chapter 9, Image Visualization*, by A Telea, AK Peters 2008


For shear-warp factorization, please see,

Acknowledgment

Thanks for materials from
• Prof. Charles D. Hansen, SCI, University of Utah
• Prof. Ronald Peikert (ETH)
• Prof. Robert Laramee, Swansea University
• Prof. Markus Hadwiger, KAUST
• Prof. Jian Huang, University of Tennessee
• Prof. Mike Bailey, Oregon State University