



---

# Accelerating Volume Rendering

---

Johns Hopkins Department of Computer Science  
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen



# Accelerating Volume Rendering

---

## Ray casting

- Alpha-cutoff
- Space leaping

## Splatting

- Hierarchical splatting
- Texture splats

## Parallel algorithms

- Screen-space subdivision
- Object-space subdivision

## 3D texture mapping

- Transparent textures
- Shaded isosurfaces



## Video

---

**State et al., “Interactive Volume Visualization on a Heterogeneous Message-Passing Multicomputer,” *Proceedings of the 1995 Symposium on Interactive 3D Graphics.***



# Texture Splats

---

**Uses texture hardware to apply contribution of each voxel to pixels**

**Precompute splat kernel texture**

**For each voxel plane (front-to-back)**

**For each voxel**

**Translate textured polygon**

**Set rgba according to voxel**

**Render textured polygon**



# 3D Texture Map Approach

---

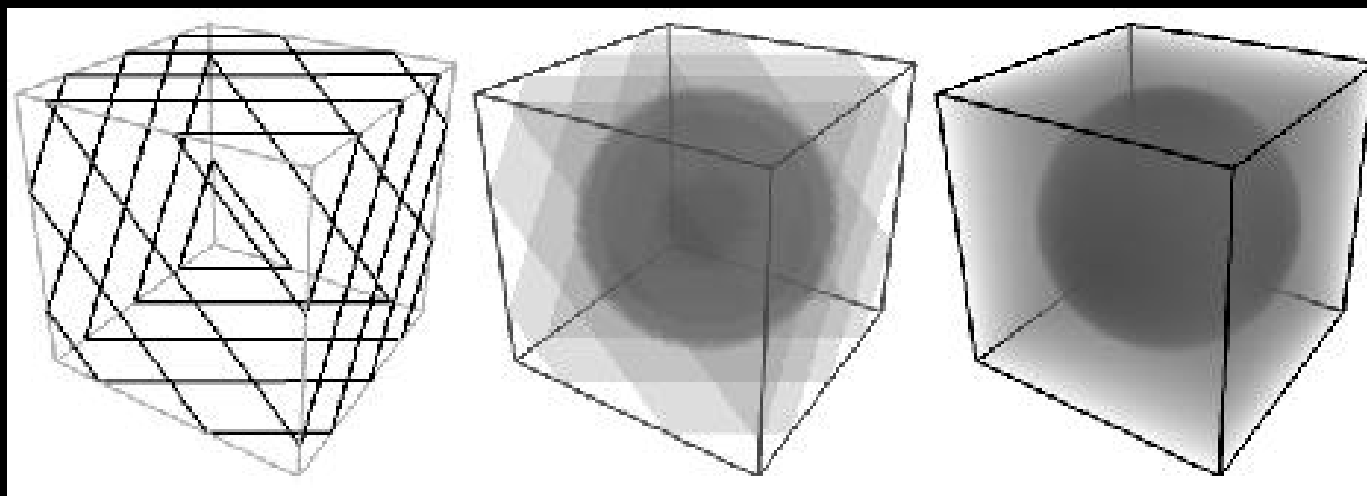
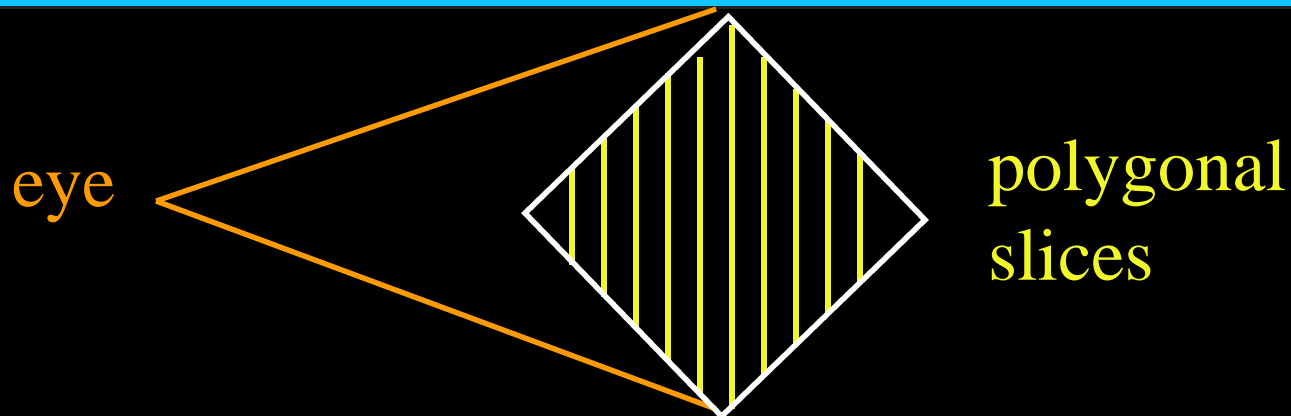
**Load volume data into 3D texture hardware**

**Generate set of slice polygons parallel to viewing plane**

**Render slices with 3D texture coordinates in front-to-back order**



# Transparent Textures Illustration



from Westermann and Ertl, "Efficiently Using Graphics Hardware in Volume Rendering Applications, *Proceedings of SIGGRAPH 98*, page 170.



# Clipping Volumes

---

**Useful for getting clear view of interior structures**

**More general than just clipping planes**

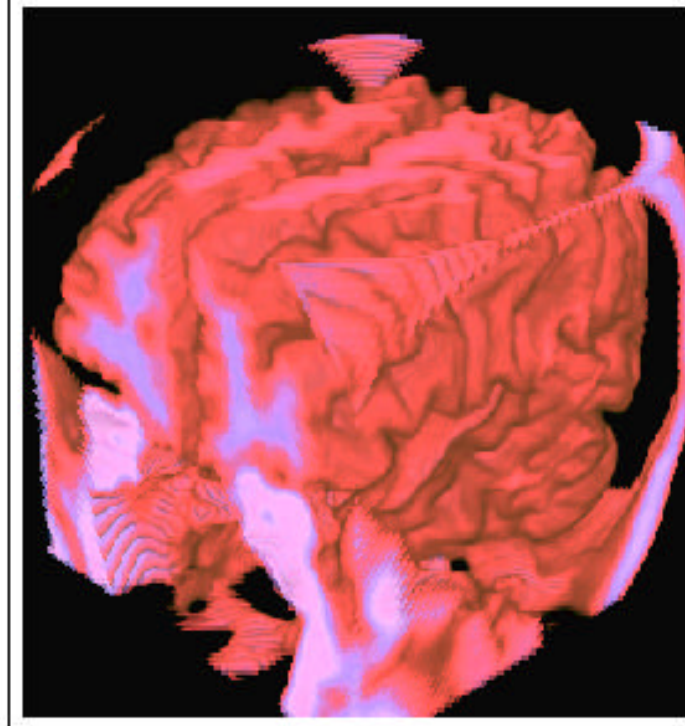
**Possible using pixel-operations on each slice**



# Clipping Box Examples



(a) Box clipping performed with the stencil buffer.



(b) Inverse box clipping of the brain.

from Westermann and Ertl, "Efficiently Using Graphics Hardware in Volume Rendering Applications, *Proceedings of SIGGRAPH 98*, page 177.





# Clipping with the Stencil Buffer

---

**Allow any closed, polygonal clip volume**

**Slice needs to be culled to pixels inside clip volume**

**Align clipping plane (near plane) with slice**

**Render clip volume into z- and stencil-buffers**

- **Back-facing polygons set stencil buffer**
- **Front-facing polygons clear stencil buffer**

**Render slice with stencil-buffer test enabled**



# Clipping Volume Diagram

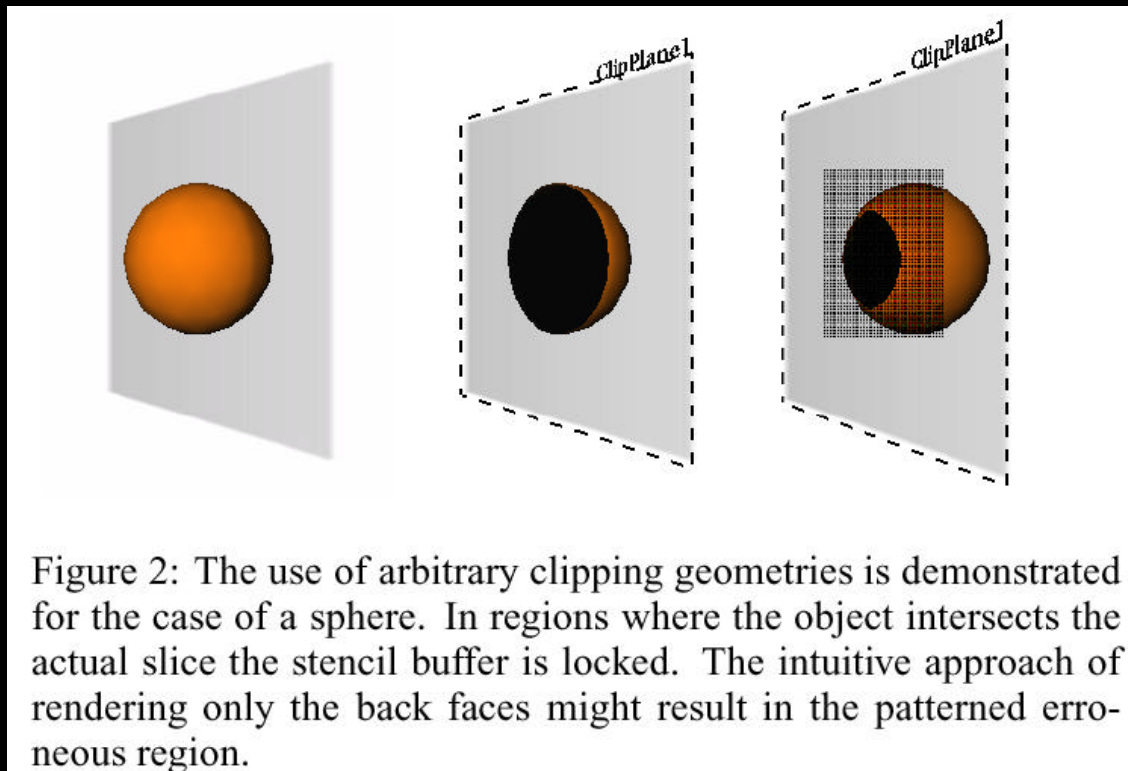


Figure 2: The use of arbitrary clipping geometries is demonstrated for the case of a sphere. In regions where the object intersects the actual slice the stencil buffer is locked. The intuitive approach of rendering only the back faces might result in the patterned erroneous region.

from Westermann and Ertl, “Efficiently Using Graphics Hardware in Volume Rendering Applications,” *Proceedings of SIGGRAPH 98*, page 177.

On the plane, pixels where we see the interior of the clip volume should leave the stencil buffer set, so those pixels of the volume are rendered.



# Direct Iso-surface Rendering

---

## Advantages

- Avoids generation of polygonal representation
- Allows interactive setting of threshold value

**Store volume values in alpha component**

**Enable alpha test to only render slice pixels with  $\alpha \geq \text{threshold}$**

**Also enable z test so only first such value is rendered**



# Iso-surface Shading

---

**Store gradient + material value in another  
3D rgba texture**

**Render iso-surface from gradient texture**

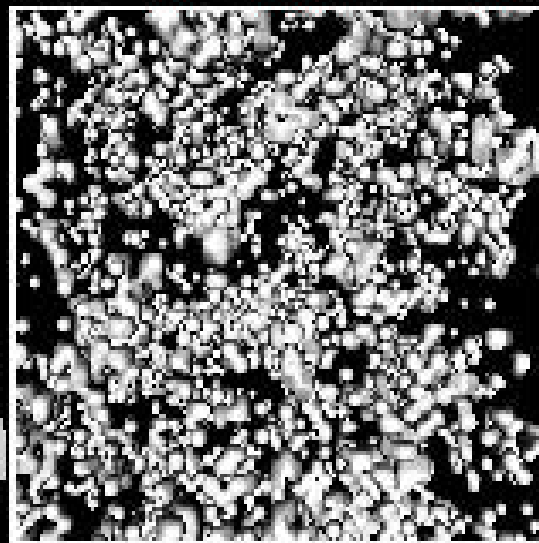
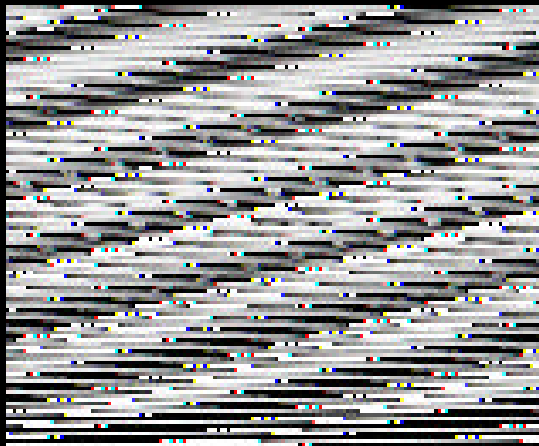
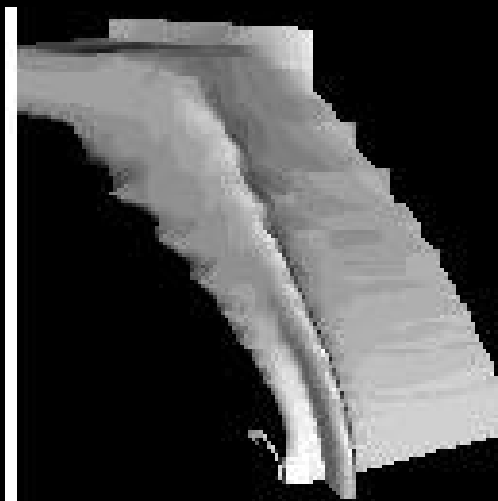
**Apply directional light using color matrix  
multiplication (copy buffer onto itself)**

**Build normal transformation into the same  
matrix**



# Shaded Iso-surface Examples

---



from Westermann and Ertl,  
“Efficiently Using Graphics  
Hardware in Volume  
Rendering Applications,  
*Proceedings of SIGGRAPH  
98*, page 177.