CSE 167:
Introduction to Computer Graphics
Lecture #16: Shadow Volumes

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2010

Announcements

- Second midterm grades will be on-line by December 2nd
- Please check Gradesource for accuracy. Homework assignments 1-6 and midterm #1 should be complete.
- Final project to be presented on **Friday, Dec 3rd, between 2 and 4pm in room 4140**
  - No late submissions accepted

Lecture Overview

- Shadow Volumes
- Volume Rendering

Shadow Volumes

NVIDIA md2shader demo

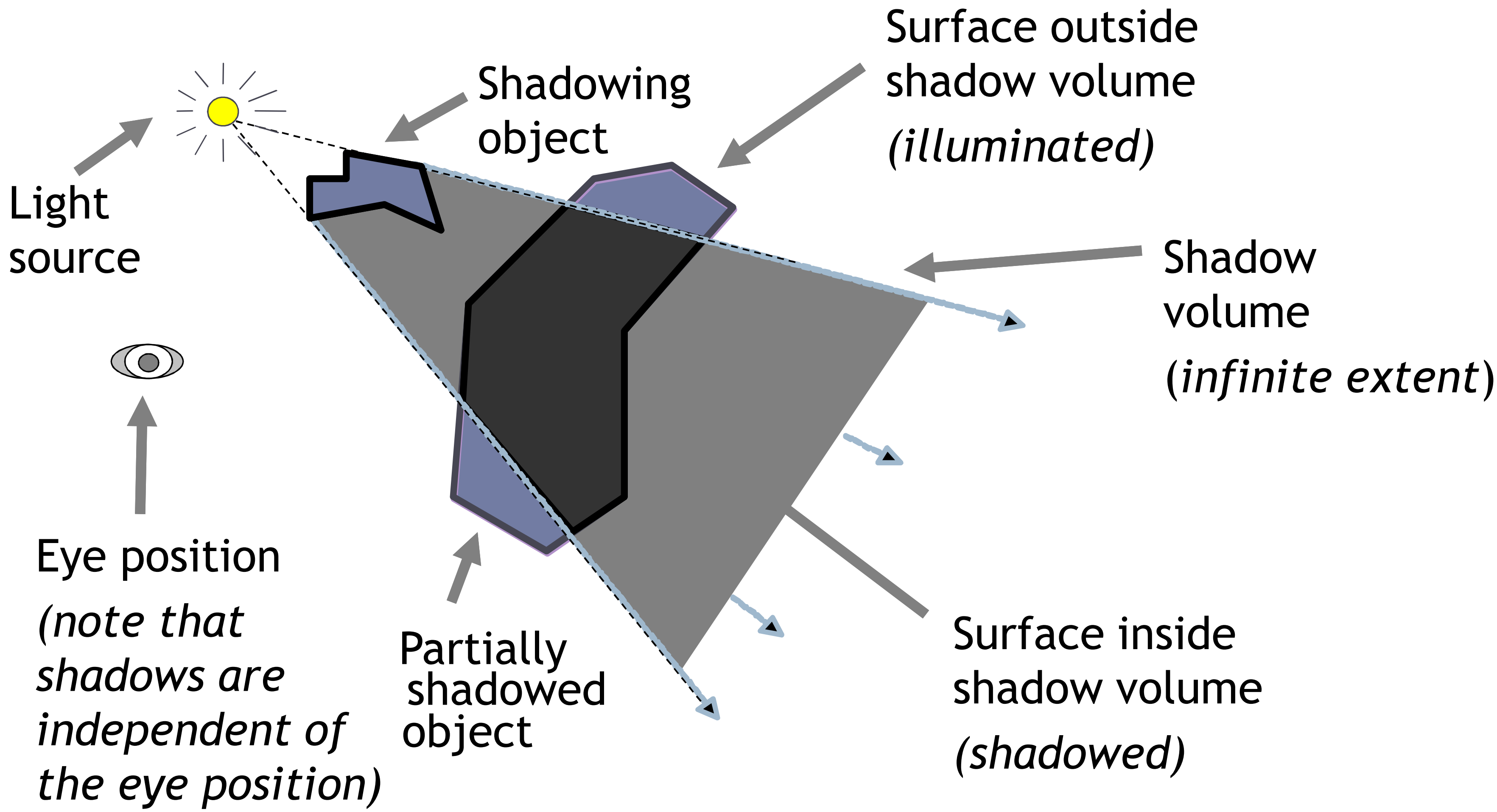
Shadow Volumes

- A single point light source splits the world in two
  - Shadowed regions
  - Unshadowed regions
  - Volumetric shadow technique
- A shadow volume is the boundary between these shadowed and unshadowed regions
  - Determine if an object is inside the boundary of the shadowed region and know the object is shadowed

Shadow Volumes

- Many variations of the algorithm exist
- Most popular ones use the stencil buffer
  - Depth Pass
  - Depth Fail (a.k.a. Carmack's Reverse, developed for Doom 3)
  - Exclusive-Or (limited to non-overlapping shadows)
- Most algorithms designed for hard shadows
- Algorithms for soft shadows exist

Shadow Volumes

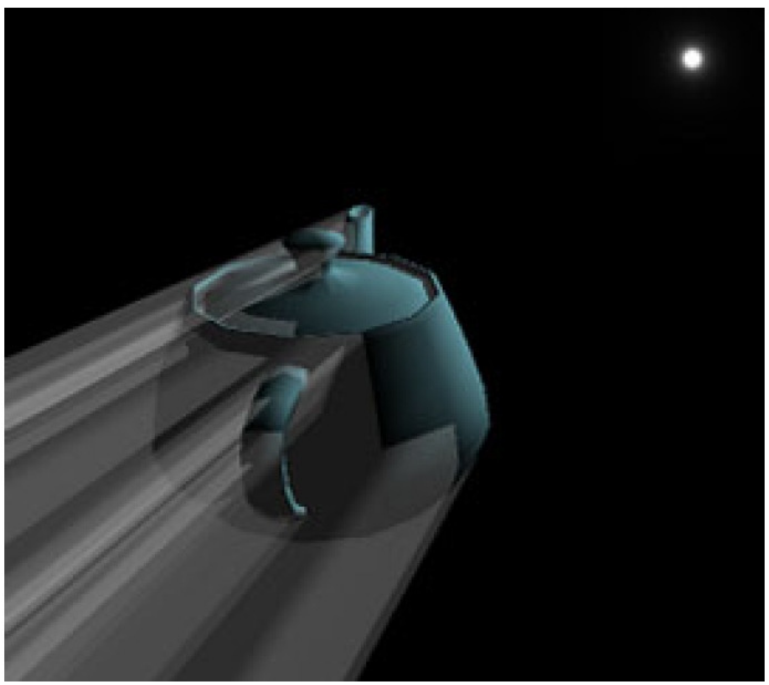


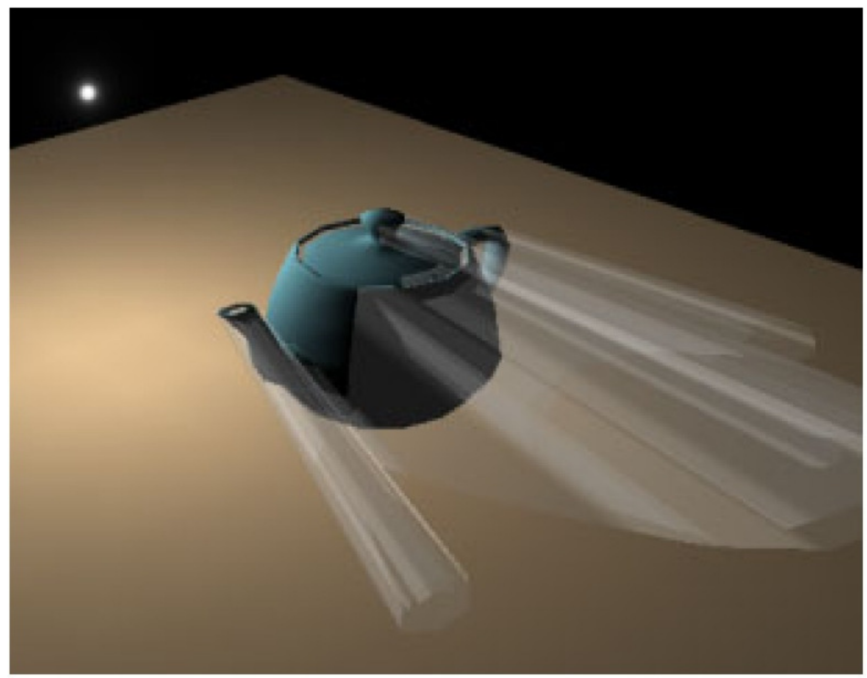
Shadow Volume Algorithm

- High-level view of the algorithm
  - Given the scene and a light source position, determine the geometry of the shadow volume
  - Render the scene in two passes
    - Draw scene with the light *enabled*,
      updating *only* fragments in *unshadowed* region
    - Draw scene with the light *disabled*,
      updated *only* fragments in *shadowed* region

Shadow Volume Construction

- Need to generate shadow polygons to bound shadow volume
- Extrude silhouette edges from light source

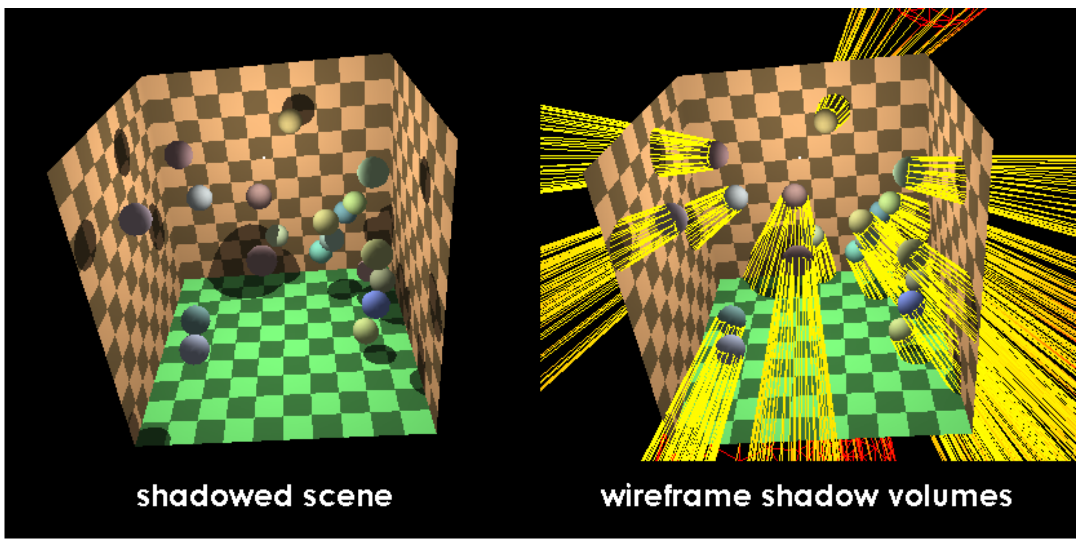




Extruded shadow volumes

Shadow Volume Construction

- Done on the CPU
- Silhouette edge detection
  - An edge is a silhouette if one adjacent triangle is front facing, the other back facing with respect to the light
- Extrude polygons from silhouette edges



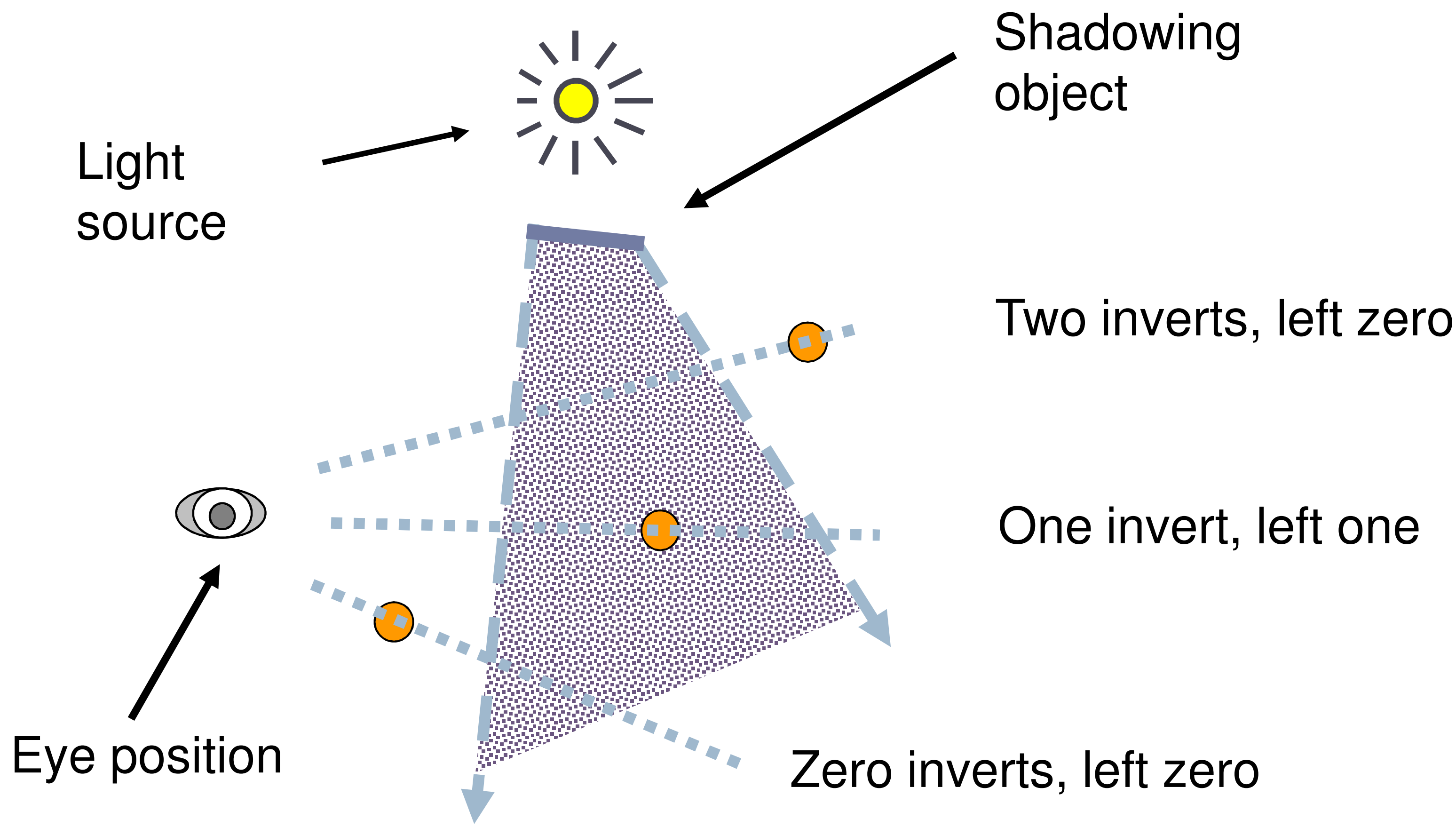
Stenciled Shadow Volumes

- Advantages
  - Support omnidirectional lights
  - Exact shadow boundaries
- Disadvantages
  - Fill-rate intensive
  - Expensive to compute shadow volume geometry
  - Hard shadow boundaries, not soft shadows
  - Difficult to implement robustly

Tagging Pixels as Shadowed or Unshadowed

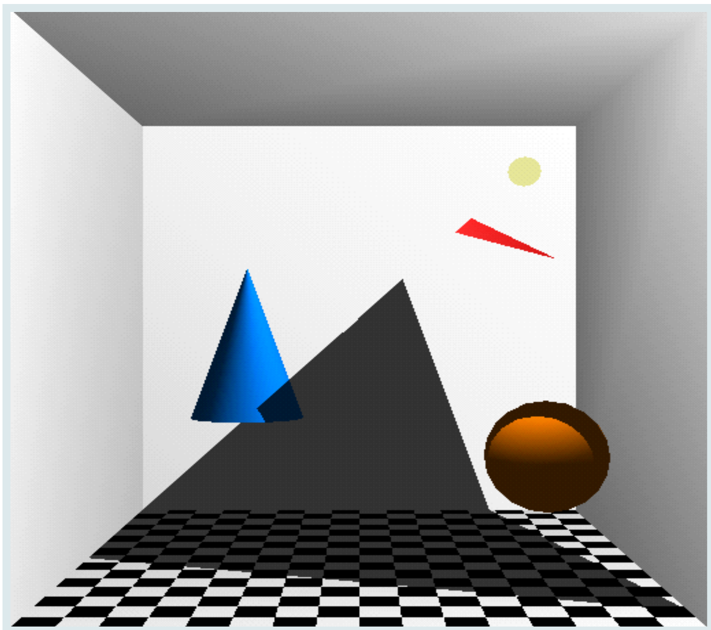
- The stenciling approach
  - Clear stencil buffer to zero and depth buffer to 1.0
  - Render scene to leave depth buffer with closest Z values
  - Render shadow volume into frame buffer with depth testing but *without* updating color and depth, but *inverting* a stencil bit (Exclusive-Or method)
  - This leaves stencil bit set within shadow

Stencil Inverting of Shadow Volume

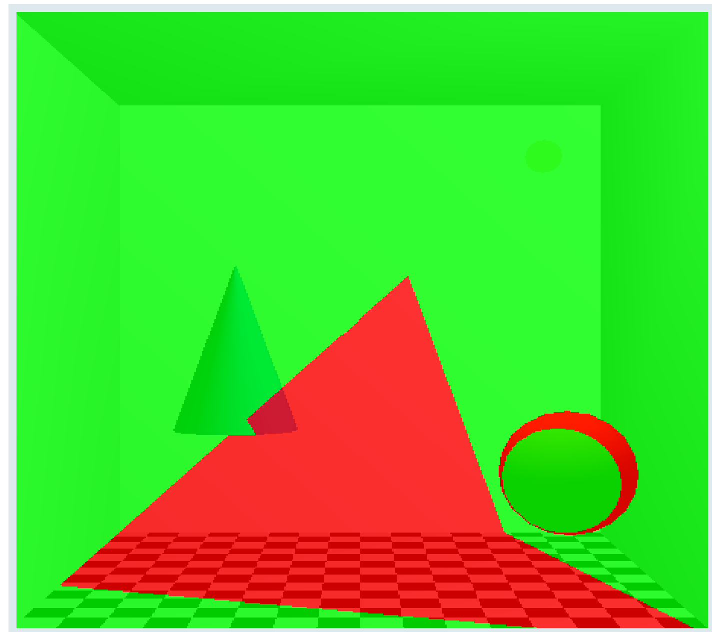


Visualizing Stenciled Shadow Volume Tagging

Shadowed scene



Stencil buffer contents



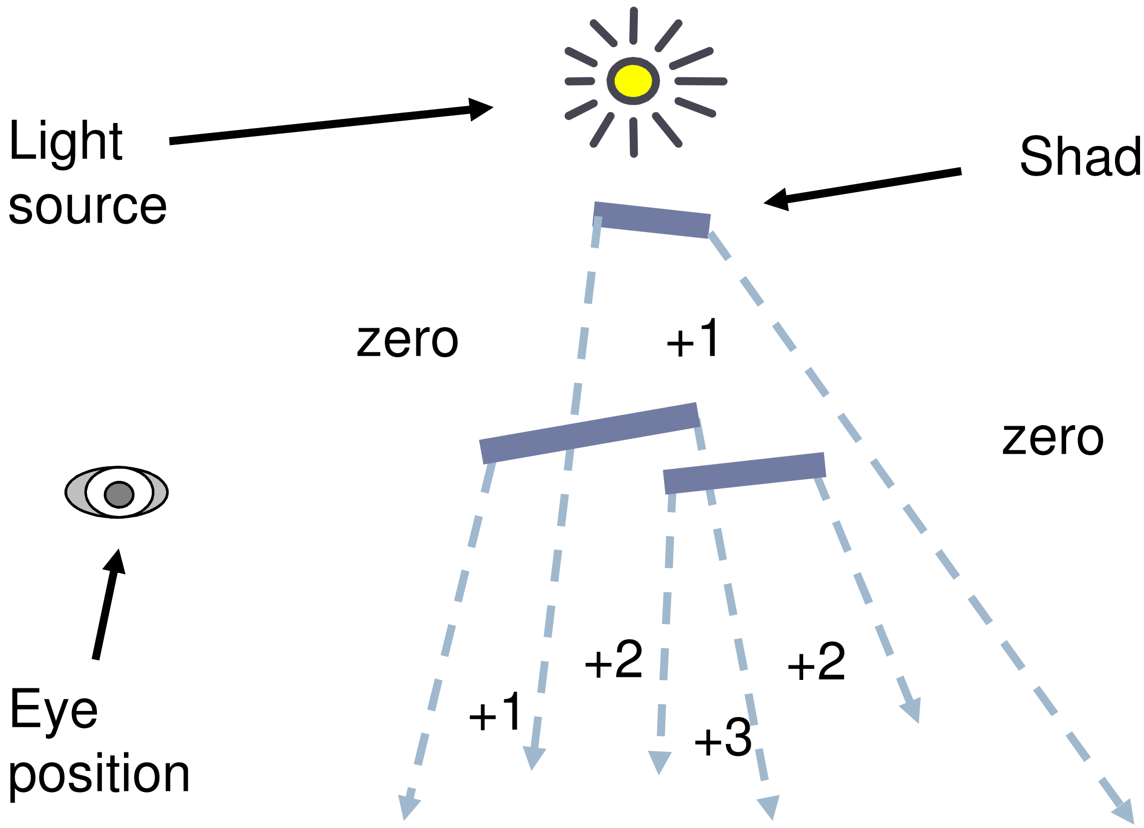
red = stencil value of 1
green = stencil value of 0

GLUT *shadowvol* example credit: Tom McReynolds

For Shadow Volumes With Intersecting Polygons

- Use a stencil enter/leave counting approach
  - Draw shadow volume twice using face culling
    - 1st pass: render *front* faces and *increment* when depth test passes
    - 2nd pass: render *back* faces and *decrement* when depth test passes
  - This two-pass way is more expensive than invert
  - Inverting is better if all shadow volumes have no polygon intersections

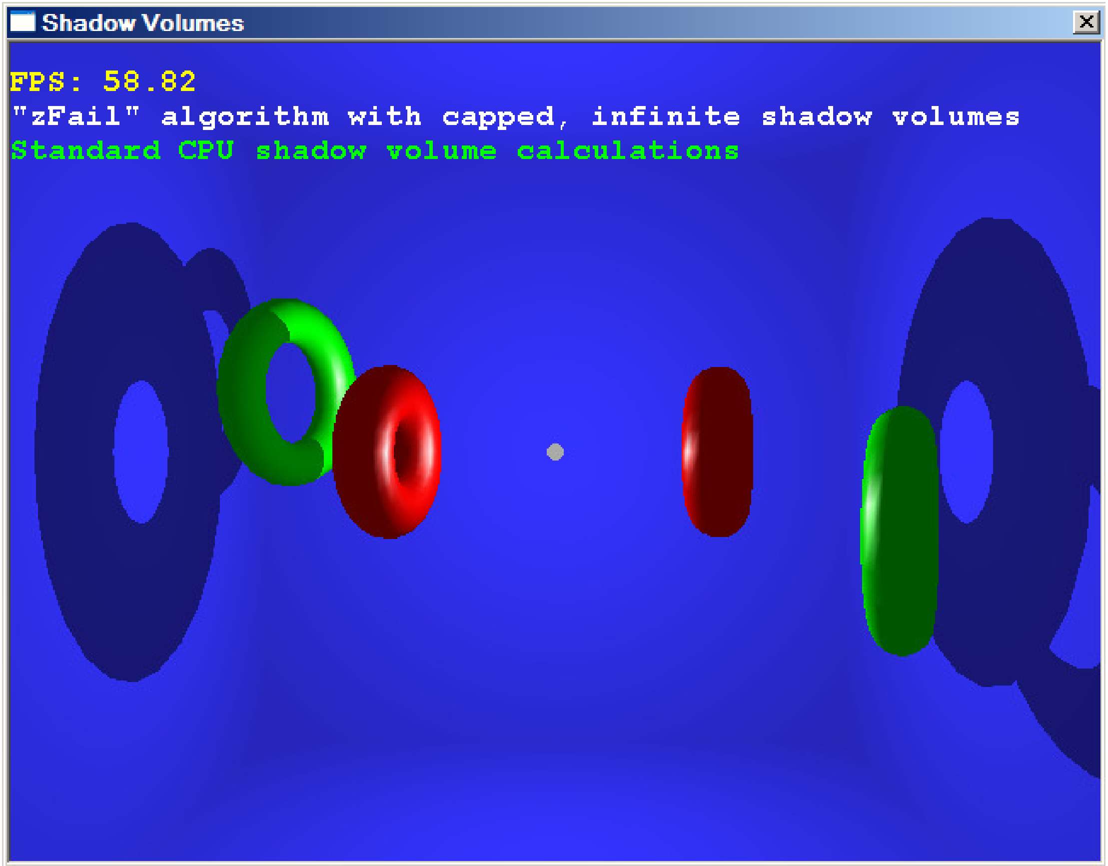
Increment/Decrement Stencil Volumes



Shadow Volume Demo

- URL:

http://www.paulsprojects.net/opengl/shadvol/shadvol.html



Resources

- Overview, lots of links
  http://www.realtimerendering.com/
- Basic shadow maps
  http://en.wikipedia.org/wiki/Shadow_mapping
- Avoiding sampling problems in shadow maps
  http://www.comp.nus.edu.sg/~tants/tsm/tsm.pdf
  http://www.cg.tuwien.ac.at/research/vr/lispsm/
- Faking soft shadows with shadow maps
  http://people.csail.mit.edu/ericchan/papers/smoothie/
- Alternative: shadow volumes
  http://en.wikipedia.org/wiki/Shadow_volume
  http://developer.nvidia.com/object/robust_shadow_volumes.html
  http://www.gamedev.net/reference/articles/article1873.asp

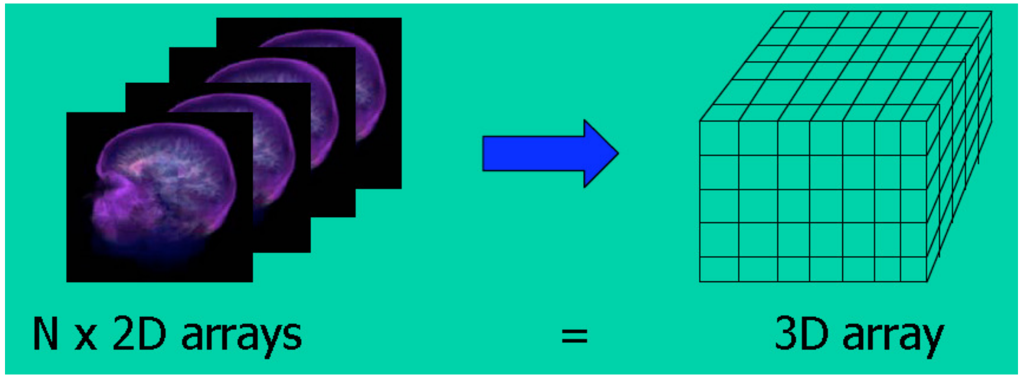
Lecture Overview

- Shadow Volumes
- Volume Rendering

What is Volume Rendering

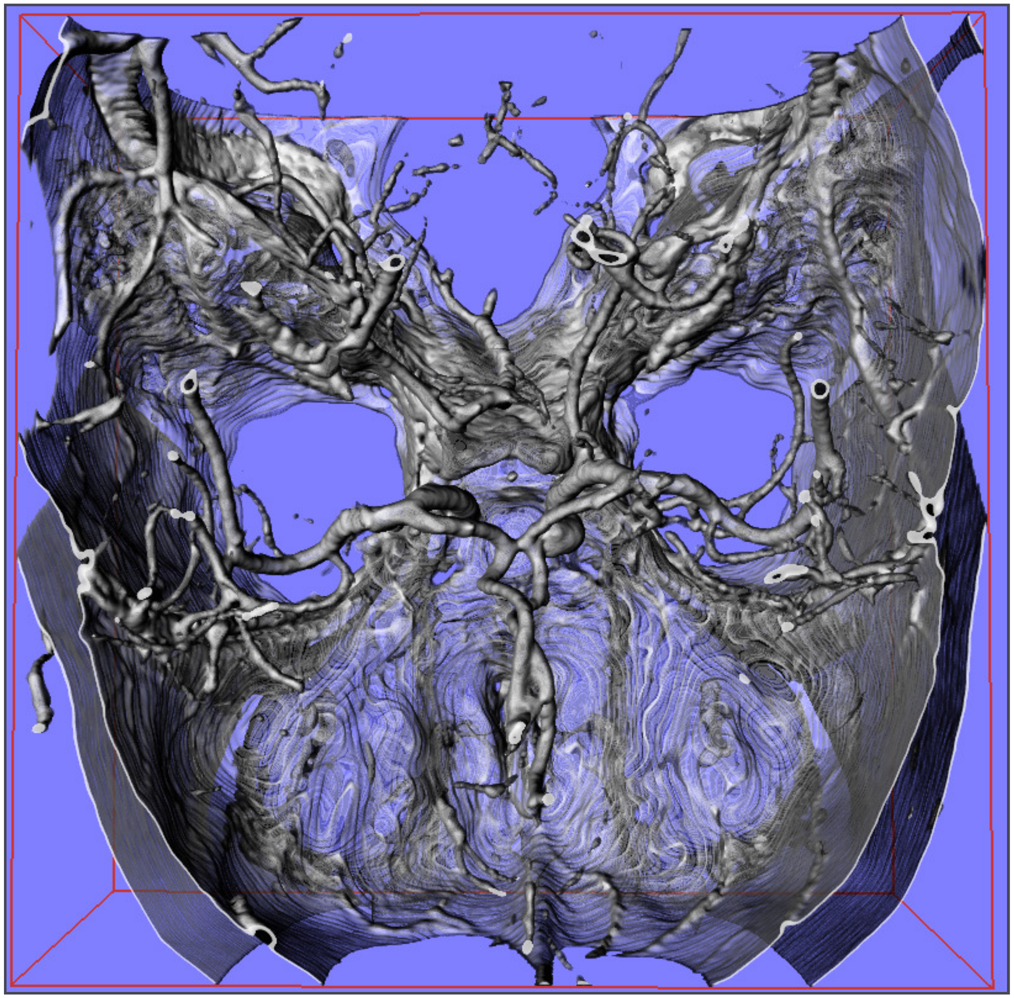
- A *Volume* is a 3D array of voxels (volume elements, 3D equivalent of pixels)
- 3D images produced by CT, MRI, 3D mesh-based simulations are easily represented as volumes
- The *Voxel* is the basic element of the volume
  Typical volume size may be 128^3 voxels, but any other size is acceptable.
- *Volume Rendering* means rendering the voxel-based data into a viewable 2D image.

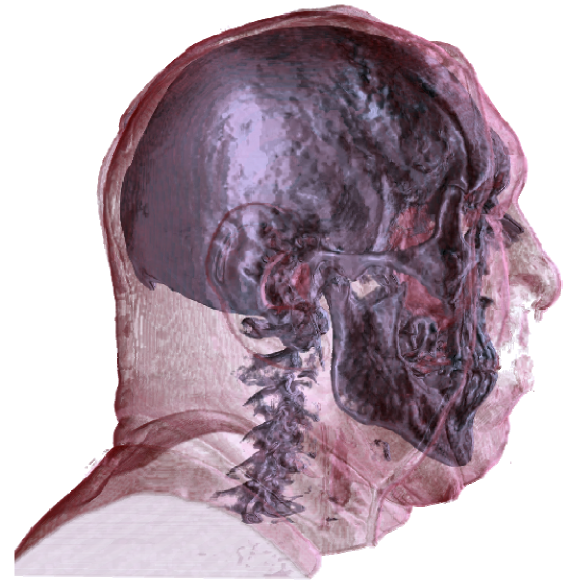
Volume Data Types



- 3D volume data are represented by a finite number of cross-sectional slices (3D grid)
- Each voxel stores a data value
  - Single bit: binary data set
  - Typical: 8 or 16 bit integers
  - Simulations often generate floating point
  - Sometimes multi-valued (multiple data values per voxel), for instance RGB, multi-channel confocal microscopy

Applications: Medicine





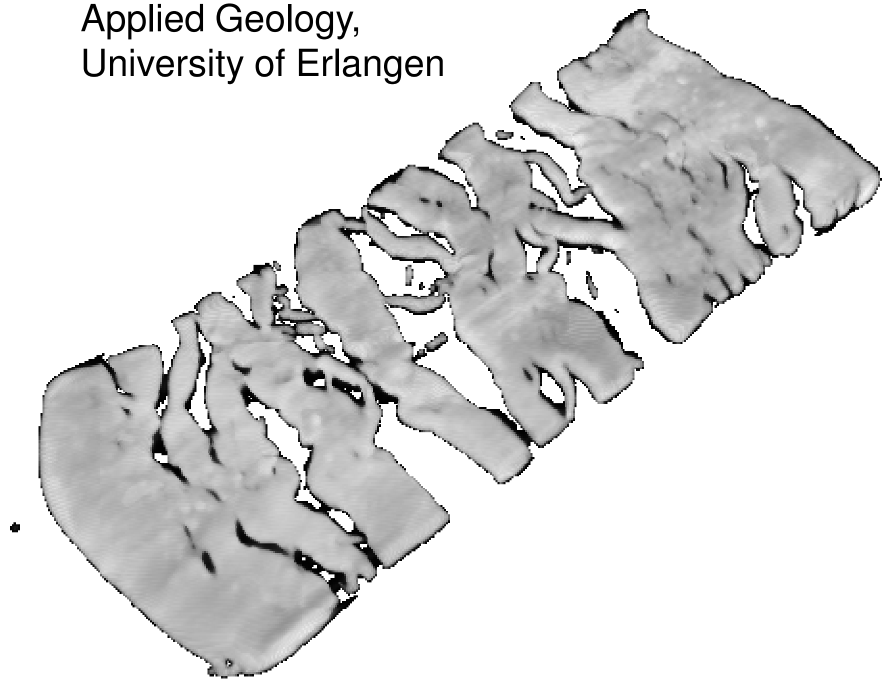
CT Human Head:
Visible Human Project, US National Library of Medicine, Maryland, USA

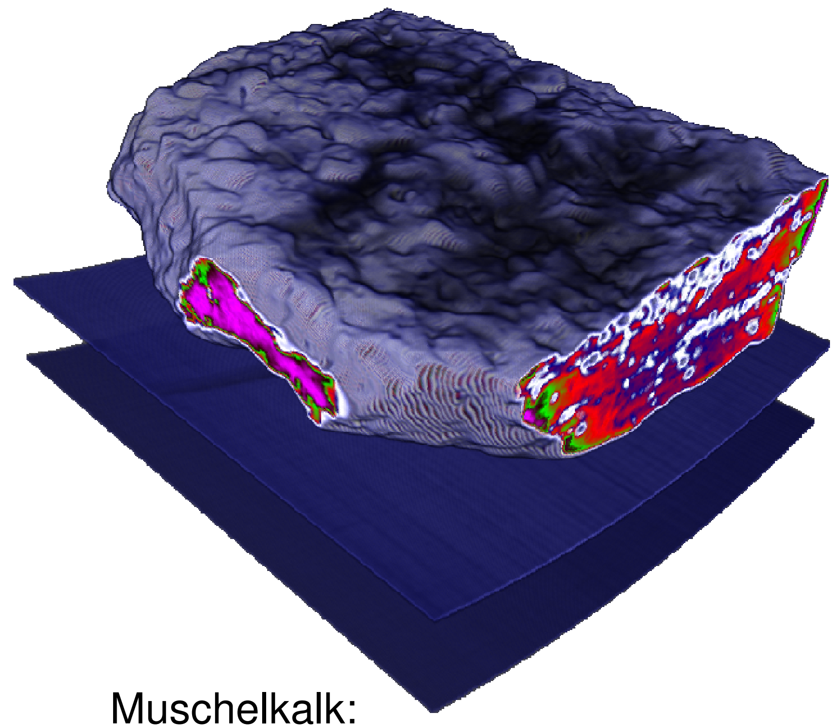
CT Angiography:
Dept. of Neuroradiology University of Erlangen, Germany

This and some of the following slides are from a Eurographics 2006 course by Dr. Christof Rezk Salama, Computer Graphics and Multimedia Group, University of Siegen, Germany

Applications: Geology

Deformed Plasticine Model,
Applied Geology,
University of Erlangen





Muschelkalk:
Paläontologie,
Virtual Reality Group,
University of Erlangen

Applications: Archaeology

Hellenic Statue of Isis

3rd century B.C.

ARTIS, University of Erlangen-Nuremberg, Germany

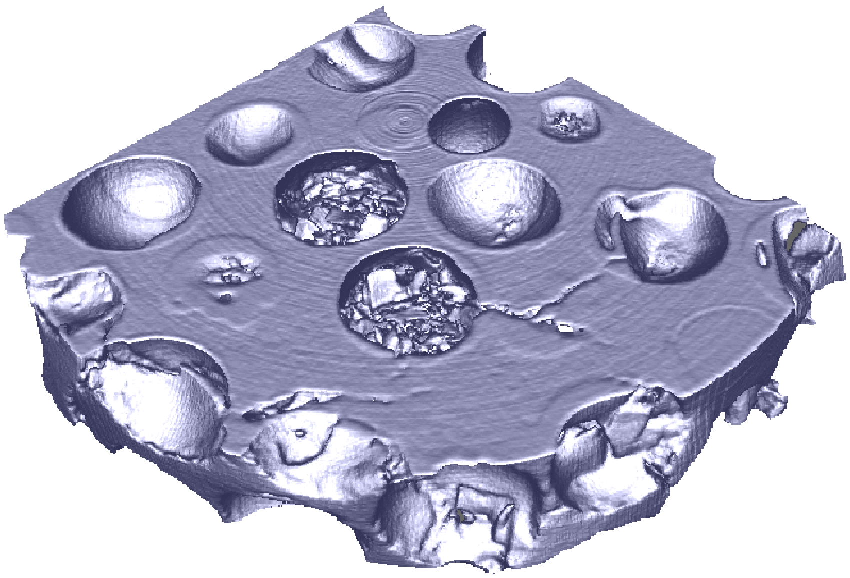
Sotades Pygmaios Statue

5th century B.C

ARTIS, University of Erlangen-Nuremberg, Germany

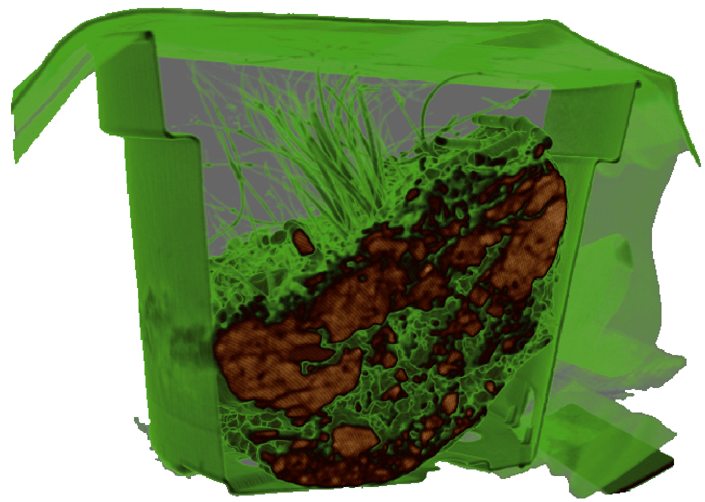
Applications

Material Science, Quality Control



Micro CT, Compound Material
Material Science Department, University of Erlangen

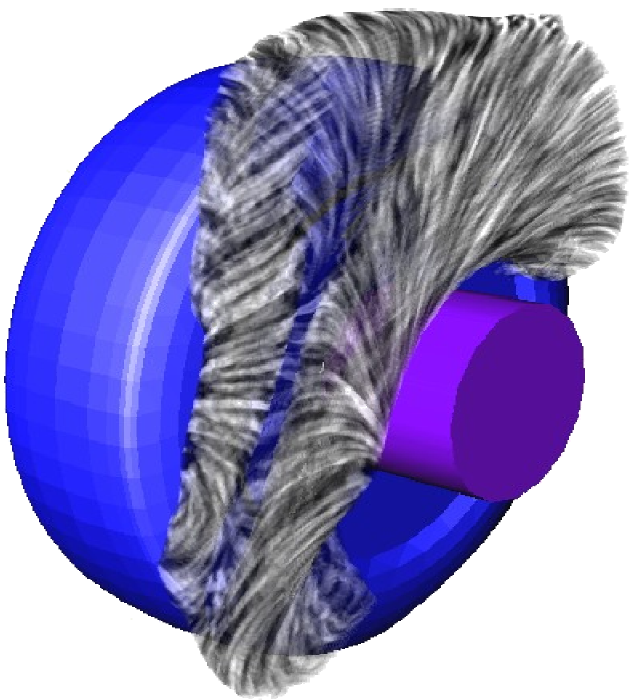
Biology

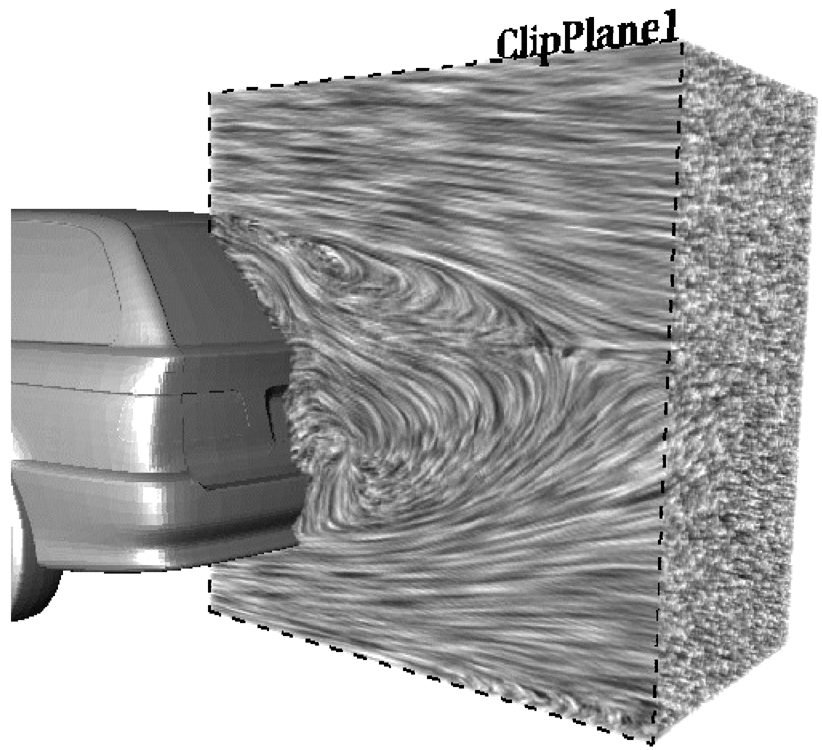


Biological sample of soil, CT
Virtual Reality Group, University if Erlangen

Applications

Computational Science and Engineering





Methods of Representation

- Polygonal - Triangle Mesh
- Freeforms - parametric curves, patches…
- Solid Modelling - CGS (Constructive Solid Geometry)
- Direct Volume Rendering

Why Direct Volume Rendering?

Pros

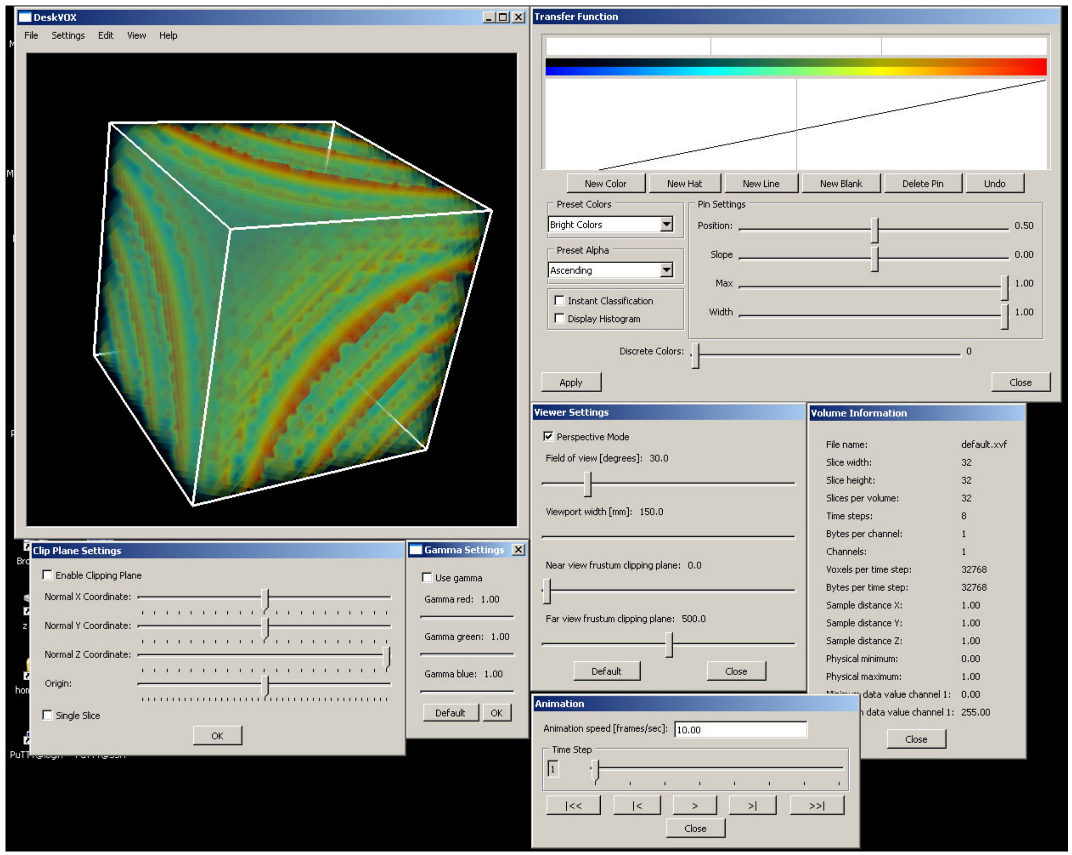
- Natural representation of CT/MRI images
- Transparency effects (Fire, Smoke…)
- High quality

Cons

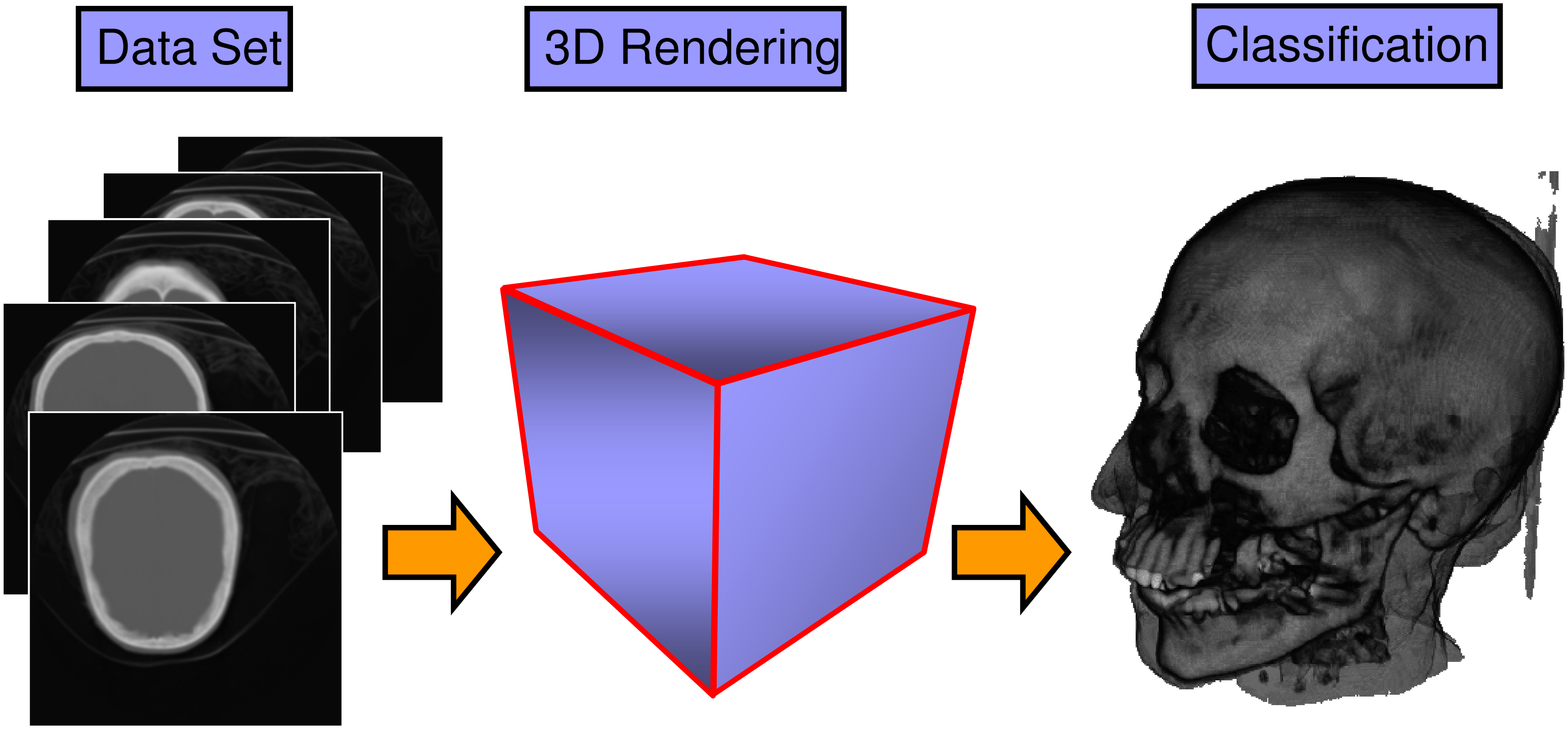
- Huge data sets
- Computationally expensive
- Cannot be embedded easily into polygonal scene

Volume Rendering: Demo

- Virvo URL: http://www.calit2.net/~jschulze/projects/vox/



Volume Rendering Outline



- in real-time on
  commodity graphics hardware

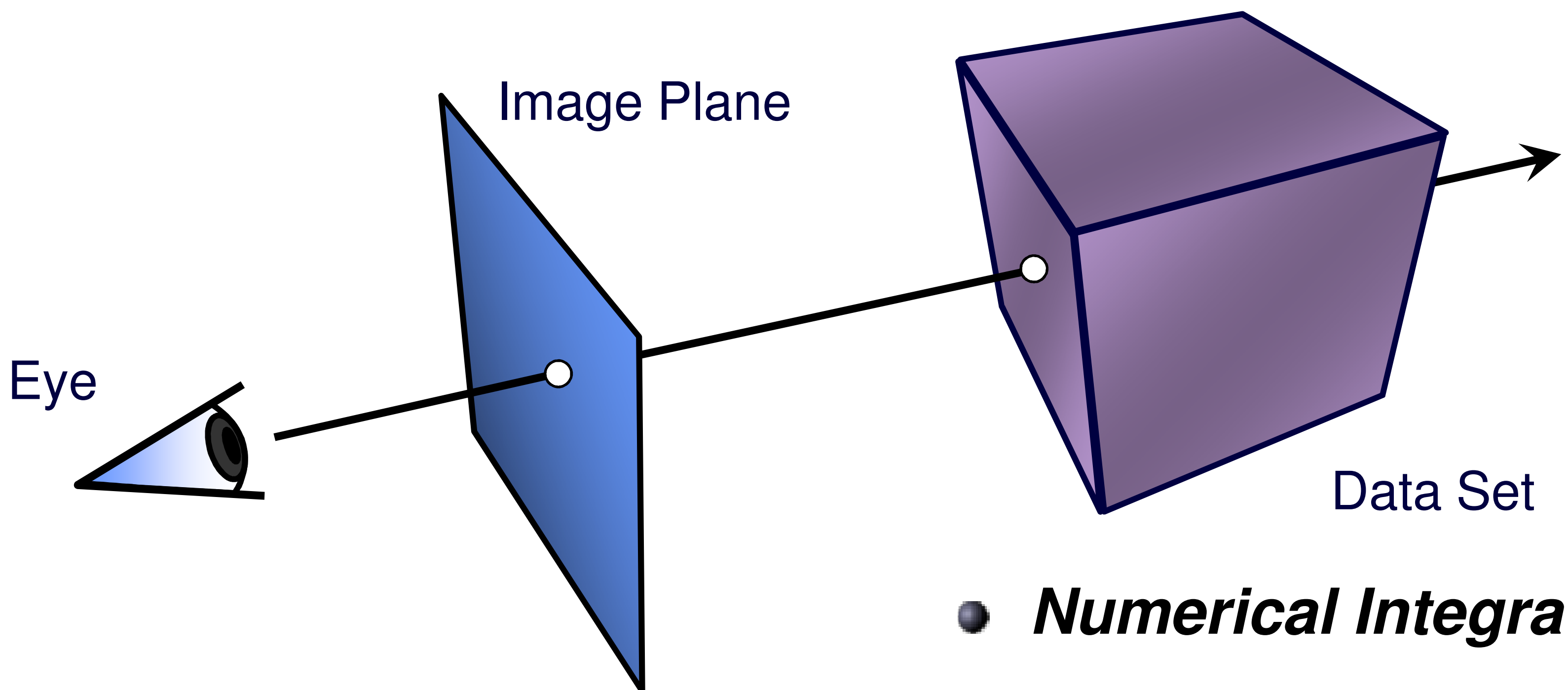
Rendering Methods

There are two categories of volume rendering algorithms:

1. Ray casting algorithms (Object Order)
   - Basic ray-casting
   - Using octrees
2. Plane Composing (Image Order)
   - Basic slicing with 2D textures
   - Shear-Warp factorization
   - Translucent textures with image-aligned 3D textures

Ray Casting

- Software Solution



- ***Numerical Integration***
- ***Resampling***

➡ ***High Computational Load***

Next Lecture

- Midterm review
- Final project Q&A
- Volume Rendering Part 2