Proxy Geometry

Slices intersected with the volume's proxy cuboid
Object(Axis)-Aligned Slicing
Object(Axis)-Aligned Slicing

Drawbacks

- We need to create 3 sets of stacked textured quads, one for each axis.
Viewport-Aligned Slicing
Viewport-Aligned Slicing

Utilizes 3D textures
Viewport-Aligned Slicing

Drawbacks

- Sampling from 3D textures are slower than 2D textures.
Wood-Grain Artifact
Ray Casting
Ray Casting

Ray for each pixel in your viewport
Ray Casting

1. Ray Casting
2. Sampling
3. Shading
4. Compositing
Ray Casting

- Parallel.
- Ray for each fragment
- Fragment shader pseudocode:

  Determine volume entry and exit positions.
  Compute ray direction.
  While (ray position in volume || other ray-termination condition)
    Sample from 3D texture at current position.
    Compositing of color and opacity.
    Advance position along ray.
  End while
Better lookup table, lighting and interpolation
Illustrative Rendering
Illustrative Rendering
Visualizing our data

- LookUp Tables
- 2D Viewing
  - MPR
  - Panorama
- 3D Viewing
  - Volume Rendering
  - ISO-Surface Rendering
3D Surfaces

- Explicit Surface Representation
- Defined as triangle mesh at the boundary of the surface.
Surface Reconstruction

Point Cloud

Surface Reconstruction

Triangle Mesh
ISO Surface
Optical Scanned Cast
User Interaction & Feedback
Scenegraphs
Organizing the world
Scenegraphs

Four Wheels

Abstract Humanoid
Scenegraphs
Scenegraphs
Multiple Views
Real-time loop
Collision Detection
Intersection Test
Intersection Test
Collision Example
Computer Vision
Image Segmentation
Segmentation on Meshes
Automatic Segmentation on a 3D Surface (Triangle Mesh)
Image Registration
Image Registration
Image Registration Approach

1. (Find feature points on the two scans).
2. Establish correspondences.
3. Compute the aligning transformation.

Partially Overlapping Scans → Aligned Scans
Implementation/Code

- Code for design
- Use Open-Source
  - QT
  - VTK (Visualization Toolkit)
  - ITK (Insight Segmentation and Registration Toolkit)
  - Open Inventor
- Code Native
  - C++
  - OpenMP
  - OpenGL / GLSL
  - CUDA / OpenCL
Targets are many,
Choose your target before aiming.