Real-Time Medical Visualization with FireGL

Natalya Tatarchuk
Jeremy Shopf
Game Computing Applications Group
AMD GPG
Real-Life Applications for Medical Visualization

- CT / MRI / PET scans
- Virtual surgeries
  - Endoscopy
  - Laparoscopy
  - Angioplasty
  - And more
- Examinations of the vascular systems
  - Ex: cerebral arteries
- Medical training

Virtual endoscopy has been shown to help planning a complex endoscopic procedure to remove pituitary tumors

Viewing the volume along with the surfaces in pre-operative planning gives additional insight on the position and structure of objects of interest.

Image courtesy of S. Wolfsberger, Department of Neurosurgery, Medical University Vienna.
Virtual Endoscopy

- **Minimally invasive procedures**
- **Particularly useful in surgery**
  - Neurosurgery, radiology and many other fields
  - Low cost and risk
  - Faster patient recovery times
  - Performed using an endoscope
- **Requirements**
  - High interactivity and feedback
  - Ability to support deformable surfaces
  - Polygonal mesh for force-feedback / collision response
  - Direct volume rendering provides navigation context near the operated surface

This image demonstrates a typical endoscopic view that can be retrieved from the inside of regions that would otherwise be difficult to reach.

*Image taken from [Neubauer et al., 2004].*
Interactive Medical Visualization

- Real-time ray-casting on the GPU for direct volume rendering
- Real-time isosurface extraction on the GPU
- Real-time classification with transfer function design on the GPU
- Essential elements for virtual surgeries
- Combines direct volume rendering with isosurface for best results
Medical Visualization Demo
Medical Visualization Process Outline

Data collection → Preprocessing and classification → Interactive visualization
Isosurface Generation

Extract a polygonal surface from scalar field
- Medical data
- Implicit surfaces and level sets
- Fluids (including CFD)
- Deformable objects
- Scientific visualization

Compute-intensive process
- Until recently GPU architecture wasn’t flexible enough for a straightforward implementation

**DirectX® 10 and the computational power of the ATI FireGL Series enable efficient, fast real-time isosurface extraction**
- Take advantage of tremendous compute power and memory bandwidth of this hardware architecture
- Flexible and powerful geometry shaders are used to build tetrahedral meshes on the fly from implicit voxel grids and extract surface triangles
Isosurface Extraction – Applications

Fluid dynamics
Isosurface Extraction – Applications

Meteorology

Chemical Visualizations
Isosurface Extraction Process

1. Generate voxel grid
2. Convert to tetrahedra
3. Run marching tetrahedra and shade
GPU Isosurface Extraction Overview

Tessellate volume into tetrahedra
- A cube can be divided into 5 or 6 tetrahedra

Check if surface intersects each edge
- Test if one vertex is inside, the other outside
- Then solve for best intersection

2 Pass algorithm
Dynamic Domain Voxelization

• Pass 1: Generate tetrahedra for voxels for a given resolution

• Dynamically voxelize the domain
  – Tessellate cubes into tetrahedra near the surface
  – Output tetrahedra as points to stream out buffer

• Only generate tetrahedra for voxels containing isosurface
  – Check whether isosurface passes through a given tetrahedron before generating its triangles
  – Reduces output memory requirements
  – Common optimization: can skip running marching tetrahedra algorithm for voxels which do not contain the surface
Extracting the Surface with the Marching Tetrahedra Pass

Perform marching tetrahedra on generated tetrahedra

- Identify edges intersecting surface
- Find surface-edge intersection
  - Fit a parabola on the edge and find root, by either:
    - Perform third function evaluation along edge
    - Use function gradients to estimate a parabola

- Output each isosurface triangle to a stream-out buffer
  - For later re-use and rendering or straight to rasterization for immediate results
Marching Tetrahedra Benefits

Emit 0, 1 or 2 tris per tetra based on edges
- Use a lookup table to consider all cases
- Solve for best intersection

Why not perform on cube?
- Smaller lookup table
- Fewer special cases
- Watertight and not ambiguous
- Allows for adaptive subdivision
Surface Quality Improvement

- Need to balance grid resolution with resulting isosurface quality
- Higher grids require significant amounts of memory
  - We cannot dynamically allocate memory on GPU!
- We develop a higher order interpolation scheme
  - Improves resulting surface quality
  - Using quadratic surface interpolation for every intersection computation

64^3 grid

Linear interpolation

Quadratic interpolation
Surface Quality Improvement: A Close Up

- Linear interpolation
- Quadratic interpolation

64³ grid
Direct Volume Rendering Overview

Cast rays from the eye into the volume and accumulate opacity

- Use dynamic flow control for early ray termination
- Terminate a ray when it accumulates an opacity value greater than or equal to one or when it leaves the volume
Direct Volume Rendering Applications

Volumetric effects
Direct Volume Rendering Applications

Deep shadows
And many more!
DVR Setup slide

- Back faces (ray exit)
- Front faces (ray entrance)
- Subtract the two to get the view vector

[Kruger 2003]
DVR Setup slide

- Back faces (ray exit)
- Front faces (ray entrance)
- Subtract the two to get the view vector

[Kruger 2003]
Rendering from Within the Volume

Moving inside the dataset causes problems
- Near clipping plane intersects the bounding geometry
- This intersection can cause visible holes
- Ray start position is undefined

Use camera position as ray start

Incorrect  Correct
Integrating Extracted Isosurface with Volume Rendering

Use isosurface as tighter bounds for ray casting

- Optimization
Direct Volume Rendering

- Volume data contains density value and 3D gradient
- Use density and gradient to parameterize transfer function
- Transfer function maps density and gradient to a color and opacity
- Use gradient for diffuse and specular lighting
Addressing Aliasing Artifacts with Jittering

- Volume visualization is typically texture fetch bound
- The less texture fetches, the better
- Reducing the number of ray steps will reduce the number of fetches, ergo performance
- Less ray steps (undersampling) will lead to aliasing
- Any technique that reduces sampling while minimizing perceived aliasing is a win
- One technique: jittering ray start positions by a random fraction of a step.
Comparison

Jittering shifts aliasing from bands to high frequency noise

Aliasing is still there, just less distracting
Interactive Transfer Function Design
Combining DVR and Isosurface

- Combine direct volume rendering and extracted isosurface
- March ray until you hit the isosurface or volume bounding box
- Difficulty: isosurface won’t be aligned to exact ray sampling positions
- Solution: Weight last volume sample density by the fraction of the step size before intersection isosurface (Dist / Step size)
Direct Volume Rendering with Isosurface
Data Sources

- CT scan of human body
  - Visible Human Project, US National Library of Medicine, Maryland, USA

- Visible Human Male dataset as used in our demo:
  - 576 MB for density and gradient data in 7 3D textures during rendering
  - 120 MB used for render targets for compositing and other tasks
  - 85 MB used for stream out buffers for isosurface extraction
Real-Time Medical Analysis and Visualization on GPU

- Combine direct volume rendering with isosurface for best results
  - **Essential elements for virtual surgeries**
  - **Extremely useful techniques for interactive games**
    - Fluids, clouds, smoke, scattering effects, and many more!
- Powerful DirectX® 10 generation ATI FireGL Series hardware provides **top-of-the-line performance** for compute-intensive techniques
  - Dynamic extraction of surface features and statistics computation
  - Ray casting for volumetric effects
  - Interactive dataset classification
Thanks!

• Daniel Szecket

• The other great folks at Game Computing Applications Group (formerly known as 3D Application Research Group) at AMD Graphics Products Group

• Christopher DeCoro (Princeton University)

• See the new AMD technical report post-Siggraph for more details
  – “Real-Time Isosurface Extraction Using the GPU Programmable Geometry Pipeline” (C. DeCoro, N. Tatarchuk, J. Shopf)
Questions?