Volume Visualization

Part 1 (out of 3)

Overview: Volume Visualization

- Introduction to volume visualization
  - On volume data
  - Surface vs. volume rendering
  - Overview: Techniques
- Simple methods
  - Slicing, cuberille
- Direct volume visualization
  - Introduction, types of combinations
  - Transfer functions

Volume Data

- Where do the data come from?
  - Medical Application
    - Computed Tomography (CT)
    - Magnetic Resonance Imaging (MR)
  - Materials testing
    - Industrial-CT
  - Simulation
    - Finite element methods (FEM)
    - Computational fluid dynamics (CFD)
  - etc.

3D Data Space

- How are volume data organized?
  - Cartesian resp. regular grid:
    - CT/MR: often \( dx=dy<dz \), e.g. 135 slices (z) \( \cong 512^2 \) values (as x & y pixels in a slice)
  - Data enhancement: iso-stack-calculation
    - Interpolation of additional slices, so that \( dx=dy=dz \Rightarrow 512^3 \) Voxel
  - Data: Cells (cuboid), Corner: Voxel
  - Curvi-linear grid resp. unstructured:
    - Data organized as tetrahedra or hexahedra
    - Often: conversion to tetrahedra

Volume Visualization

- Introduction:
  - VolVis = visualization of volume data
    - Mapping 3D \( \rightarrow \) 2D
    - Projection (e.g., MIP), slicing, vol. rendering, …
  - Volume data =
    - 3D \( \times \) 1D data
    - Scalar data, 3D data space, space filling
  - User goals:
    - Gain insight in 3D data
    - Structures of special interest + context

VolVis – Challenges

- Rendering projection, so much information and so few pixels!
- Large data sizes, e.g. \( 512 \times 512 \times 1024 \) voxel \( \cong 16 \) bit = 512 Mbytes
- Speed,
  - Interaction is very important, >10 fps!
Voxels vs. Cells

- Two ways to interpret the data:
  - Data: set of voxel
    - voxel = abbreviation for volume element (cf. pixel = "picture elem.")
    - voxel = point sample in 3D
    - Not necessarily interpolated
  - Data: set of cells
    - cell = cube primitive (3D)
    - Corners: 8 voxel (see above)
    - Values in cell: interpolation used

Gradients as Normal Vector Replacement

- Gradient \( \nabla f = (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z) \)
- \( \nabla f(x_0) \) normal vector to iso-surface \( f(x_0)=f_0 \)
- Central difference in \( x-, y- \& z \)-direction (in voxel):
  \[
  \nabla f(x,y,z) = \frac{1}{2} \left( \frac{f(y+1)-f(y-1)}{2} \right) \left( \frac{f(z+1)-f(z-1)}{2} \right)
  \]
- Then tri-linear interpolation within a cell

Interpolation

\[
\nabla f(x,y,z) = S(m(x), m(y), m(z))
\]

Nearest Neighbor

Trilinear

Concepts and Terms

- Sampled data (measurement)
- Analytical data (modelling)
- Voxel space (discrete)
- Geometric surfaces (analytic)
- Pixel space (discrete)
- Surface rendering

Example 1:
- CT measurement
- Iso-stack-conversion
- Iso-surface-calculation (marching cubes)
- Surface rendering (OpenGL)
### Concepts and Terms

#### Example 2:
- **MR measurement**
- **Iso-stack-conversion**
- **MIP** (maximum intensity projection)
- **Image:** blood-vessels in hand

#### Example 3:
- **Potential function** \( \rho(x,y,z) \)
- **Iso-surface** \( \rho(x,y,z) = \rho_0 \)
- **Surface:** ray tracing

### Surfaces vs. Volume Rendering

#### Surface rendering:
- **Indirect** volume visualization
- Intermediate representation: iso-surface, “3D”
- **Pros:** Shading→Shape!, HW-rendering

#### Volume rendering:
- **Direct** volume visualization
- Usage of transfer functions
- **Pros:** Illustrate the interior, semi-transparency

### VolVis-Techniques – Overview

#### Simple methods:
- Slicing, MPR (multi-planar reconstruction)
- **Direct volume visualization:**
  - Ray casting
  - Shear-warp factorization
  - Splatting
  - 3D texture mapping
  - Fourier volume rendering
- **Surface-fitting methods:**
  - Marching cubes (marching tetrahedra)
**Image-Order vs. Object-Order**

- **Image-order:**
  - FOR every pixel DO: …
  - Cost, complexity \(\approx\) image size
  - Example: ray casting (tracing viewing rays)

- **Object-order:**
  - FOR every object (voxel) DO: …
  - Cost, complexity \(\approx\) object size (# of voxels)
  - Examples: splatting ("throwing snow balls")

---

**Image-Order Approach**

**Image-Order Approach:** Traverse the image pixel-by-pixel and sample the volume.

Ray Casting

---

**Object-order approach**

**Object-Order Approach:** Traverse the volume, and project to the image plane.

Splatting cell-by-cell

Texture Mapping plane-by-plane

---

**Simple Methods**

Slicing, etc.

---

**Simple Methods**

Slicing:
- Axes-parallel slices
- regular grids: simple
- without transfer function no color
- Windowing:
  - adjust contrast

Windowing:
- white
- black
- data values

---

**Simple Methods**

Slicing:
Slicing

- Not so simple:
  - Slicing through general grid
  - Interpolation necessary
- Slicing:
  - well combinable with 3D-visualization
- Multi-planar reformation (MPR):
  - arbitrary axes, 3D

Direct Volume Visualization

- Overview:
  - No intermediate representation
  - "real 3D"
  - Integration of so much information difficult
  - Object-order vs. image-order rendering
  - Various techniques (ray casting, splatting, shear-warp, texture mapping, Fourier volume rendering, etc.)
  - Various types of combinations (compositing, MIP, first-hit, average, etc.)

Types of Combinations

- Overview:
  - MIP
  - Compositing
  - X-Ray
  - First hit

First Hit: Iso-Surface Extraction

- First: Extracts iso-surfaces (again!), done by Tuy&Tuy '84
**Average: as X-Ray Images**

- Average: Produces basically an X-ray picture

**Types of Combination**

- Possibilities:
  - \( \alpha \)-compositing
  - Shaded surface display
  - Maximum-intensity projection
  - X-ray simulation
  - Contour rendering

**MIP: Maximum-Intensity Projection**

- Max: Maximum Intensity Projection used for Magnetic Resonance Angiograms, for example

**Classification**

- Assignment data \( \Rightarrow \) semantics:
  - Assignment to objects, e.g., bone, skin, muscle, etc.
  - Usage of data values, gradient, curvature
  - Goal: segmentation
  - Often: semi-automatic resp. manual
  - Automatic approximation: transfer functions (TF)

**Compositing: Semi-Transparency**

- Accumulate: Make transparent layers visible!
  - Levoy '88

**Transfer Functions (TF)**

- Mapping data \( \rightarrow \) "renderable quantities":
  1.) data \( \rightarrow \) color
  2.) data \( \rightarrow \) opacity (non-transparency)
Different Transfer Functions

- Image results:
  - Strong dependence on transfer functions
  - Non-trivial specification
  - Limited segmentation possibilities

Gradient-Based Transfer Functions

- 2D-Transfer function:
  - Levoy ‘88
  - Specific opacity at certain threshold
  - but: close-by variation according gradient magnitude
  - highlights transitions (large gradients)
  - dampens homogeneous areas

Lobster – Different Transfer Functions

- Three objects: media, shell, flesh

Inclusion of the Gradient

- Emphasis of changes:
  - Special interest often in transitional areas
  - Gradients: measure degree of change (like surface normal)
  - Larger gradient magnitude ⇒ larger opacity

Multi-Dimensional Transfer Functions (1)

- \( f, f', f'' \) histograms to depict material boundaries

Multi-Dimensional Transfer Functions (2)

- Direct manipulation widgets [Kniss et al. 2002]
Acknowledgments

- For material for this lecture unit
- Roberto Scopigno, Claudio Montani (CNR, Pisa)
- Hans-Georg Pagendarm (DLR, Göttingen)
- Michael Meißner (GRIS, Tübingen)
- Torsten Möller
- Gordon Kindlmann
- Joe Kniss
- etc.