Computer Graphics

- Volume Rendering -

Philippe Weier
Alexander Rath
Philipp Slusallek
Overview

- Motivation
- Volume Representation
- Indirect Volume Rendering
- Volume Classification
- Direct Volume Rendering
Applications: Bioinformatics
Applications: Entertainment
Applications: Industrial
Applications: Medical
Applications: Simulations

Image by [RTVG 08]
Volume Processing Pipeline

- **Acquisition**
  - Measurement or computation of the data

- **Filtering**
  - Picking desired features, cleaning, noise-reduction, re-sampling, reconstruction, classification, ...

- **Mapping**
  - Map N-dimensional data to visual primitives

- **Rendering**
  - Generate the image

- **Post-processing**
  - Enhancements (gamma correction, tone mapping)
Volume Acquisition

- **Measurements**
  - Computer Tomography (CT, X-Ray),
  - Magnetic Resonance Imaging (MRI, e-spin)
  - Positron-Emission Tomography (PET)
  - Ultrasound, sonar
  - Electron microscopy
  - Confocal microscopy
  - Cryo-EM/Light-Tomography
  - Seismic exploration
  - ...

- **Simulations**
  - Essentially everything > 2D

- **Visualization of mathematical objects**
Filtering

• **Raw data usually unsuitable**
  – Cleaning & repairing
  – Noise reduction and removal
  – Correcting incomplete, out-of-scale values
  – Selection of relevant aspects
    • Lots of information and features in a 3D volume
    • Potentially hiding/obscuring each other
  – Classification

• **Adaptation of format**
  – Re-sampling (often to Cartesian grids)

• **Transformations**
  – Volume reconstructing of 3D data from projection
Mapping

• Create something visible
  – Interpretation of measurement values
  – Mapping to geometric primitives
  – Mapping to parameters (colors, absorption coefficients, ...)

• Rendering
  – Surface extraction vs. direct volume rendering
  – Single volume vs multiple (possibly overlapping)
  – Object-based vs. image-based rendering
    • Forward- or backward mappings (rasterization/RT)
Volume Rendering

• **Our input?**
  – Representation of volume

• **Our output?**
  – Colors for given samples (pixels)

• **Our tasks?**
  – Map “weird values” to optical properties
  – “Project 1D/2D/3D/nD data values within a 3D context to 2D image plane”
VOLUME ACQUISITION AND REPRESENTATION
Data Acquisition

- **Simulated Data**
  - Fluid dynamics
  - Heat transfer
  - etc...
  - Generally: “Scientific Visualization”

- **Measured Data**
  - CT (Computed Tomography) scanner
    - Reconstructed from rotated series of two-dimensional X-ray images
    - Good contrast between high and low density media (e.g., fat and bones)
  - MRI (Magnetic Resonance Imaging)
    - Based on magnetic/spin response of hydrogen atoms in water
    - Better contrast between different soft tissues (e.g., brain, muscles, heart)
  - PET (Positron Emission Tomography)
  - And many others (also here on campus, e.g., material science)
Data Acquisition

- CT vs. MRI
Volume Representations

• **Definition**
  – 3D field of values: Essentially a 3D scalar or color texture
  – Sometimes higher dimensional data (e.g., vector/tensor fields)

• **Sampled representation**
  – 3D lattice of sample points (akin to an image but in 3D)
    • Typically, equal-distance in each directions
  – Generally, point cloud in space
  – Ideally, neighborhood information (topology)
  – Data values at these locations

• **Procedural**
  – Mathematical description of values in space
  – Sum of Gaussians (e.g., in quantum mechanics)
  – Perlin noise (e.g., for non-homogeneous fog)
  – Always convertible to sampled representation
    • But with loss of information
Volume Organization

• **Rectilinear Grids**
  – Common for scanned data
  – May have different spacing

• **Curvilinear Grids**
  – Warped rectilinear grids

• **Unstructured Meshes**
  – Common for simulated data
  – E.g., tetrahedral meshes

• **Point clouds**
  – No topological/connection information
    • Neighborhood computed on the fly
Reconstruction Filter

- **Nearest Neighbor**
  - Cell-centered sample values

- **Tri-Linear Interpolation**
  - Node-centered sample values
Tri-Linear Interpolation

- **Compute Coefficients**
  - \( \text{wx} = (x - x_0) / (x_1 - x_0) \)
  - \( \text{wy} = (y - y_0) / (y_1 - y_0) \)
  - \( \text{wz} = (z - z_0) / (z_1 - z_0) \)

- **3-D Scalar Field per Voxel**
  - \( f(x, y, z) = (1 - \text{wz}) (1 - \text{wy}) (1 - \text{wx}) \cdot c_{000} \)
  - \( + (1 - \text{wz}) (1 - \text{wy}) \cdot \text{wx} \cdot c_{100} \)
  - \( + (1 - \text{wz}) \cdot \text{wy} (1 - \text{wx}) \cdot c_{010} \)
  - \( + (1 - \text{wz}) \cdot \text{wy} \cdot \text{wx} \cdot c_{110} \)
  - \( + \text{wz} (1 - \text{wy}) (1 - \text{wx}) \cdot c_{001} \)
  - \( + \text{wz} (1 - \text{wy}) \cdot \text{wx} \cdot c_{101} \)
  - \( + \text{wz} \cdot \text{wy} (1 - \text{wx}) \cdot c_{011} \)
  - \( + \text{wz} \cdot \text{wy} \cdot \text{wx} \cdot c_{111} \)
Tri-Linear Interpolation

- **Successive Linear Interpolations**
  - Along X
    - $c_{00} = (1 - wx) c_{000} + wx c_{100}$
    - $c_{01} = (1 - wx) c_{001} + wx c_{101}$
    - $c_{10} = (1 - wx) c_{010} + wx c_{110}$
    - $c_{11} = (1 - wx) c_{011} + wx c_{111}$
  - Along Y
    - $c_{0} = (1 - wy) c_{00} + wy c_{10}$
    - $c_{1} = (1 - wy) c_{01} + wy c_{11}$
  - Along Z
    - $c = (1 - wz) c_{0} + wz c_{1}$

- **Order of dimensions does not matter**
VOLUME MAPPING
Mapping / Classification

• **Definition**
  - Map scalar data values to optical properties
  - E.g.
    • Optical density
    • Albedo
    • Emission

• **Instances**
  - Analytical function
  - Discrete representation
    • Array of sample colors corresponding to sample data values
    • Interpolate colors for data values in between given data points
Mapping / Classification

• Physical Mapping
  – Physically-based mapping via optical properties of material
    • Concentration of soot to optical density, albedo, etc…
    • Temperature to emitted blackbody radiation
  – Allows for realistic rendering, often intuitively interpretable by us
Mapping / Classification

- **Empirical or task-specific mapping (Transfer Function)**
  - User-defined mapping from data to colors
    - Typically stored as an array of sample correspondences (color map transfer function)
  - Mapping may have no physical interpretation
    - Assigning color to pressure, electrostatic potential, electron density, …
    - Use of ideas from *visualization*
  - Highlight specific features of the data
    - Isolate bones from fat
Pre/Post-Classification

• **Pre-Classification**
  - First classify data values in sample cells
  - Then interpolate classified optical properties

• **Post-Classification**
  - First interpolate data values, then classify interpolated values
Cinematic Rendering

- Deutsche Zukunftspreis 2017
  - Klaus Engel & Robert Schneider, Siemens Healthineers
DIRECT VOLUME RENDERING
Direct Volume Rendering

- **Definition**
  - Directly render the volumetric data (only) as translucent material
Scattering in a Volume

\[ \omega \]

out-scattering

in-scattering

emission

absorption
Beer’s Law

- **Volumetric Attenuation**
  - Assume constant optical density $\kappa_{01}$
  - Transmittance:
    - $T(x_0, x_1) = e^{-\kappa_{01}(x_1-x_0)}$
  - Transmitted radiance:
    - $L_0(x_0, \omega) = T(x_0, x_1) L_0(x_1, \omega)$
Analytical Form

- **Volumetric Attenuation**
  - Assume constant optical density $\kappa_{01}$ (extinction coefficient)
  - Transmittance: $T(x_0, x_1) = e^{-\kappa_{01}(x_1-x_0)}$
  - Transmitted radiance: $T(x_0, x_1) L_0(x_1, \omega)$

- **Volumetric Contribution/Emission**
  - Also assume (constant) volume radiance $L_v(x, \omega)$ [Watt/(sr m^3)]
  - Contributed radiance: $(1 - T(x_0, x_1))L_v(x_{01}, \omega)$

- **Volumetric Equation**
  - Radiance reaching the observer
    - Emission within segment + transmitted background radiance
  - $L_0(x_0, \omega) = (1 - T(x_0, x_1))L_v(x_{01}, \omega) + T(x_0, x_1)L_0(x_1, \omega)$
Ambient Homogenous Fog

- **Constant-Optical Density**
- **Volumetric Contributions**
  - Assume constant volumetric albedo $\rho_v(x)$
  - Assume constant ambient lighting $L_a$ (everywhere, no shadowing)
  - Leads to constant volume radiance $L_v(x, \omega) = L_a \rho_v$
- **Pervasive Fog**
  - Entry at camera, exit at intersection, or inf.
- **Algorithm**
  - Compute surface illumination $L_o(x_1, \omega)$
    - Modulate shadow visibility by transmittance between surface and light source
  - Compute volume transmittance $T(x_0, x_1)$ and attenuate surface radiance
  - Add contributions from volume radiance
Ambient Homogeneous Fog

- **Pros**
  - Simple
  - Efficient

- **Cons**
  - No true light contributions
  - No volumetric shadows
Ray-Marching

- Riemann Summation
  - Non-constant optical density / non-constant volume radiance
  - Sample volume at discrete locations
  - Assume constant density and volume radiance in each interval
Ray-Marching

- **Homogeneous Segments**
  - \( L_0(x_0, \omega) = (1 - e^{-\kappa_{01}\Delta x})L_v(x_{01}, \omega) + e^{-\kappa_{01}\Delta x}L_0(x_1, \omega) \)
  - \( L_0(x_1, \omega) = (1 - e^{-\kappa_{12}\Delta x})L_v(x_{12}, \omega) + e^{-\kappa_{12}\Delta x}L_0(x_2, \omega) \)
  - \( L_0(x_2, \omega) = ... \)

- **Recursive Substitution**

\[
L_0(x_0, \omega) = (1 - e^{-\kappa_{01}\Delta x})L_v(x_{01}, \omega) + e^{-\kappa_{01}\Delta x}\left( (1 - e^{-\kappa_{12}\Delta x})L_v(x_{12}, \omega) + e^{-\kappa_{12}\Delta x}(...) \right)
\]

\[
= (1 - e^{-\kappa_{01}\Delta x})L_v(x_{01}, \omega) + e^{-\kappa_{01}\Delta x}(1 - e^{-\kappa_{12}\Delta x})L_v(x_{12}, \omega) + e^{-\kappa_{01}\Delta x}e^{-\kappa_{12}\Delta x}(...)
\]

\[
= \sum_{i=0}^{n-1} \left( \prod_{j=0}^{i-1} e^{-\kappa_{j,j+1}\Delta x} \right) (1 - e^{-\kappa_{i,i+1}\Delta x})L_v(x_{i,i+1}, \omega) + \left( \prod_{j=0}^{n-1} e^{-\kappa_{j,j+1}\Delta x} \right) L_0(x_n, \omega)
\]
Ray-Marching (front to back)

- \( L = 0; \)
- \( T = 1; \)
- \( t = 0; // t_{enter}; \)
- \( \text{while}(t < t_{exit}) \)
  - \( \text{dt} = \min(t_{step}, t_{exit} - t); \)
  - \( P = \text{ray.origin} + (t + \text{dt}/2) \times \text{ray.direction}; \)
  - \( b = \exp(-\text{volume.density}(P) \times \text{dt}); \)
  - \( L += T \times (1 - b) \times \text{Lv}(P); \)
  - \( T *= b; \)
  - // Optional early termination
  - \( t += t_{step}; \)
- \( L += T \times \text{trace}(\text{ray.origin} + t_{exit} \times \text{ray.direction}, \text{ray.direction}); \)
- \( \text{return} L; \)
Homogeneous Fog

- Constant-optical density
- Non-constant volume radiance
  - Similar to surface reflected radiance (i.e., rendering equation)
  - Use phase function $\rho(x, \Delta \omega)$, (e.g., $\frac{\rho v}{4\pi}$) instead of $\text{BRDF} \times \text{cosine}$
  - Modulate shadow visibility by transmittance
Homogeneous Fog

- **E.g., Anisotropic Point Light**
  - Modulate visibility at surfaces by transmittance

\[
L_{rl}(x, \omega_o) = \frac{I(-\omega)}{||x - y||^2} V(x, y) T(x, y) f_r(\omega(x, y), x, \omega_o) \cos \theta_i
\]

- Modulate visibility at each volume sample by transmittance

\[
L_v(x, \omega_o) = \frac{I(-\omega)}{||x - y||^2} V(x, y) T(x, y) \frac{\rho_v}{4 \pi}
\]
Homogeneous Fog

- Inverse Square Law
- Volumetric Shadows
- Projective Light
Heterogeneous Fog

- **Assumptions**
  - Non-constant-optical density
  - Non-constant volume radiance

- **Shadow visibility modulated by transmittance**
  - Ray-marched shadow rays at surface
  - Ray-marched shadow rays at each volume sample

\[
T(x_0, x_n) = \prod_{j=0}^{n-1} e^{-\kappa_{j,j+1}\Delta x}
\]
Heterogeneous Fog
Ray-Casting

• Early Ray Termination
  – Abort ray-marching when subsequent contributions are negligible
  – if (T < epsilon) return L;
  – Very effective in dense volumes
  – Also avoids ray-marching to infinity

• Grid Traversal
  – 3-D DDA
  – Ray-marching

• Adaptive Marching
  – Bulk integration over homogeneous regions (e.g., octree, bricks)
  – Pre-compute and store maximum step size separately
  – Increasing step size with decreasing accumulated transmittance
  – Vertex Connection and Merging & Joint Path Sampling [Siggraph’14]
Full Volumetric Light Simulation

- Taking into account multiple scattering in the volume
Full Volumetric Light Simulation

- Including Shadows, Caustics, etc.