Visualization

"a picture says more than a thousand numbers"

Visualization can facilitate people to better understand the information embedded in the given dataset.

The merge of data with the display geometric objects through computer graphics.

Data

2D dataset
- Bar chart, pie chart, graph, stocks.
- Information visualization

3D dataset
- Scalar data
- Vector data
- Tensor data
Data

- 2D dataset
  - Bar chart, pie chart, graph, stocks.
  - Information visualization (higher dimensions)
- We focus on 3D datasets.

Volume Visualization

- Iso-surface extraction (Polygonization)
  - Marching cubes
- Direct volume rendering
  - Ray casting
  - splatting

Iso-contour extraction

- $f(x, y) = 0$
- $f(x, y) > 0$
- $f(x, y) < 0$

Four unique cases (after considering symmetry)

Marching Squares

- $f(x, y) = 0$
- $f(x, y) > 0$
- $f(x, y) < 0$
Principle of Occam’s razor:
If there are multiple possible explanations of a phenomenon that are consistent with the data, choose the simplest one.

Linear interpolation
- \( f(x,y) = 0 \)
- \( f(x,y) > 0 \)
- \( f(x,y) < 0 \)

\[
\begin{align*}
fi, j+1 &> 0 \\
fi, j &< 0 \\
fi+1, j+1 &> 0 \\
fi+1, j &> 0
\end{align*}
\]

\[
\Delta x, j=0
\]

\[
\Delta x / h = (b-a) / (-a)
\]

\[
\Delta x = (a-b) h / a
\]
Marching Squares-ambiguity

More information are needed to resolve ambiguity

Iso-contour extraction

- \( f(x,y) = 0 \)
- \( f(x,y) > 0 \)
- \( f(x,y) < 0 \)

Surface (comparison)

- A mesh of polygons:
  - 200 polys
  - 1,000 polys
  - 15,000 polys

as "empty" foot

Surface - Pros and Cons

• Pros:
  - fast rendering algorithms are available
  - acceleration in special hardware is relatively easy and cheap (many $200 game boards)
  - use OpenGL to specify rendering parameters
  - surface realism can be added via texture mapping

• Cons:
  - discards the interior of the object and just maintains the object’s shell
  - does not facilitate real-world operations such as cutting, slicing, direction
  - does not enable artificial viewing modes such as semi-transparencies, X-ray
  - surface-less phenomena such as clouds, fog, gas are hard to model and represent

Volumes with direct rendering

- Maintains a representation that is close to the underlying fully-3D object (but discrete)
- Models the object as a magic gel that can change its properties at any time.
- Different aspects of the dataset can be emphasized via changes in the functions that translate raw densities into colors and transparencies
- Volume rendering is a formidable technique for the exploration of datasets, since when the nature of the data is not known, it is difficult to create the right polygonal mesh
Direct Volume Rendering

Volumes with direct rendering
- Pros:
  - maintains a representation that is close to the underlying fully-3D object (but discrete)
  - can achieve a level of realism (and ‘hyper-realism’) that is unmatched by surface graphics.
  - allows easy and natural exploration of volumetric datasets
- Cons:
  - has high rendering complexity
  - hardware acceleration is complex and expensive (a commodity board costs over $3,000)

Volume Rendering
- The process of generating a 2D image from the 3D volume is called volume rendering
- Sampling: MRI, CT, Ultrasound
- Numerical simulations: Computational Fluid Dynamics, Finite Element Method
- Discretization: voxelization

Volume Grid Types
- Cubic
- Anisotropic rectilinear
- Rectilinear
- Curved
- Structured

Volume Rendering Modes
- For each pixel in the image, a ray is cast into the volume.
- Four main volume rendering modes exist:
  - X-ray: rays sum volume contributions along their linear paths
  - Isosurface: rays look for the object surfaces, defined by a certain volume value
  - Maximum Intensity Projection (MIP): a pixel value stores the largest volume value along its ray
  - Full volume rendering: rays composite volume contributions along their linear paths

Volume Rendering Pipeline
Full Volume Rendering

• It is the accumulation of colors weighted by opacities
• Colors and opacities of back pixels are attenuated by opacities of front pixels:
  \[
  r_{\text{rgb}} = R_{\text{back}} \cdot \alpha_{\text{back}} (1 - \alpha_{\text{front}}) + R_{\text{front}} \cdot \alpha_{\text{front}}
  \]
• Volume rendering uses this recursive expression to combine (=composite) the samples taken along the ray

Raycasting

• Consider a volume consisting of particles:
  - each has color \( C \) and light attenuating density \( \mu \)
• A rendering ray accumulates attenuated colors:
• We write the continuous volume rendering integral:
  \[
  f(x) = \int \frac{C(x) \mu(x)}{\mu(x)} \, dx \quad (\text{this is generally not solvable analytically})
  \]
• We can approximate it by discretizing it into sampling intervals of width \( \Delta s \):
  \[
  f(x) = \sum_{k} \frac{C(x) \mu(x)}{\mu(x)} \Delta s \quad \prod_{j=0}^{k-1} \int_{\mu(x)}^{\mu(x) + \mu} \mu(x)
  \]

Scientific Visualization Examples

• Medical data visualization
  – CT, MRI, PET, …
• Scientific examples:
  - Diffusion of heat through a metal bar
  - Interactive display of molecule from molecule description file
Computational Fluid Dynamics

- Flow around an airplane wing
- Shock wave
- Vortex flow

Display of Natural Phenomena

- Global wind and weather modeling