Volume Rendering

Shading and
Transfer Function
Shading Model with Phong

- Simple model used in volume rendering
- Three components
  - Diffuse
  - Specular
  - Ambient
- Uses four vectors
  - To source
  - To viewer
  - Normal
  - Perfect reflector
Material Properties

• 9 coefficients for each light source
  – $I_{dr}$, $I_{dg}$, $I_{db}$, $I_{sr}$, $I_{sg}$, $I_{sb}$, $I_{ar}$, $I_{ag}$, $I_{ab}$

• Material properties match light source properties
  – Nine absorption coefficients
    • $k_{dr}$, $k_{dg}$, $k_{db}$, $k_{sr}$, $k_{sg}$, $k_{sb}$, $k_{ar}$, $k_{ag}$, $k_{ab}$
    – Shininess coefficient $\alpha$
Total Reflection Model

For each light source and each color component, the Phong model can be written as

\[ I = k_d I_d \mathbf{l} \cdot \mathbf{n} + k_s I_s (\mathbf{v} \cdot \mathbf{r})^\alpha + k_a I_a \]

For each color component we add contributions from all sources.
The Halfway Vector

- \( \mathbf{h} \) is normalized vector halfway between \( \mathbf{l} \) and \( \mathbf{v} \)

\[
\mathbf{h} = \frac{(\mathbf{l} + \mathbf{v})}{|\mathbf{l} + \mathbf{v}|}
\]
Using the halfway angle

• Replace \((\mathbf{v} \cdot \mathbf{r})^\alpha\) by \((\mathbf{n} \cdot \mathbf{h})^\beta\)

• \(\beta\) is chosen to match shineness

• Note that halfway angle is half of angle between \(\mathbf{r}\) and \(\mathbf{v}\) if vectors are coplanar

• Resulting model is known as the modified Phong or Blinn lighting model
  – Specified in OpenGL standard
  – Generally used in volume rendering
How to compute normal?

• Not explicitly defined as in surface models
• Use gradient vector in volume data sets to replace normal
  – Compute gradient vector from volume densities
  – Need to compute for each sampling point
Gradient Vector

- The gradient vector \((g_x, g_y, g_z)\) at each sampling point \((x, y, z)\) usually computed by *central difference* operation

\[
g_x = \frac{f(x-1, y, z) - f(x + 1, y, z)}{2} \quad g_y = \frac{f(x, y - 1, z) - f(x, y + 1, z)}{2} \quad g_z = \frac{f(x, y, z - 1) - f(x, y, z + 1)}{2}
\]

- The gradient has to point from higher density to lower density, follow the above equation
Gradient Computation

- Each \( f() \) will generally need to interpolate
- An alternative way:
  - Compute gradient vector for each grid point
  - Then, generate a *gradient volume* in preprocessing, each voxel now stores the gradient vector
  - Interpolate gradient vector for each sampling point from *gradient volume*
Voxel Meaning

• Each voxel stores density
• What does the density represent?
  – Depends on particular applications
  – Real density, temperature, stress...
  – X-ray absorption of material (CT)
  – Magnetic property of material (MRI)
  – Artificial material tag (voxelization from surface models)
• How to visualize? Density → Color?
Transfer Function

• Give meaningful visual attributes to voxel density
  – Color
  – Transparency
• Use the visual attributes to visualize the volume data sets
• Such mapping from voxel density to color (transparency) is defined by transfer function
Define Transfer Function

- An example: for MRI head data set
  - Brain to pink and opaque
  - Skin and skull to grey and semi-transparent, so we can see through
Transfer Functions

- Such mapping need to interact with end users
- Generally, transfer function is an important topic in Human-Computer Interface (HCI) design
General Transfer Functions

- Mapping density to R,G,B,A
  - Red, green, blue, alpha (opacity)
- We will define it as mouse controlled curves
Advance Transfer Functions

• An example: map density to importance for particular applications

• Ivan Viola PhD Thesis: Importance-Driven Expressive Visualization