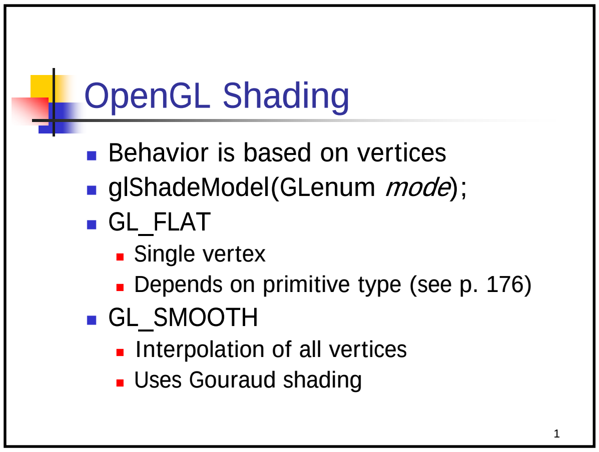
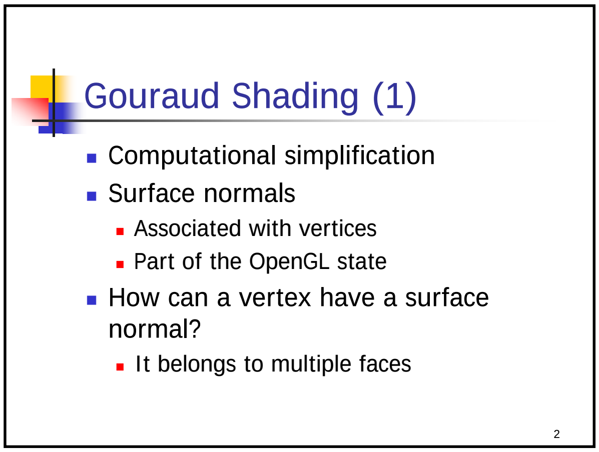


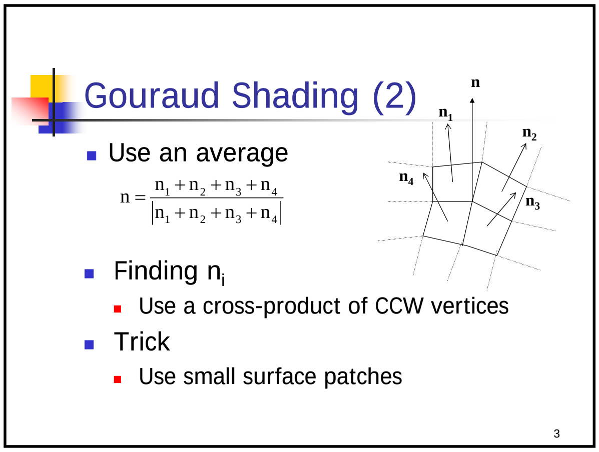
OpenGL Shading

- Behavior is based on vertices
- glShadeModel(GLenum *mode*);
- GL_FLAT
  - Single vertex
  - Depends on primitive type (see p. 176)
- GL_SMOOTH
  - Interpolation of all vertices
  - Uses Gouraud shading



Gouraud Shading (1)

- Computational simplification
- Surface normals
  - Associated with vertices
  - Part of the OpenGL state
- How can a vertex have a surface normal?
  - It belongs to multiple faces

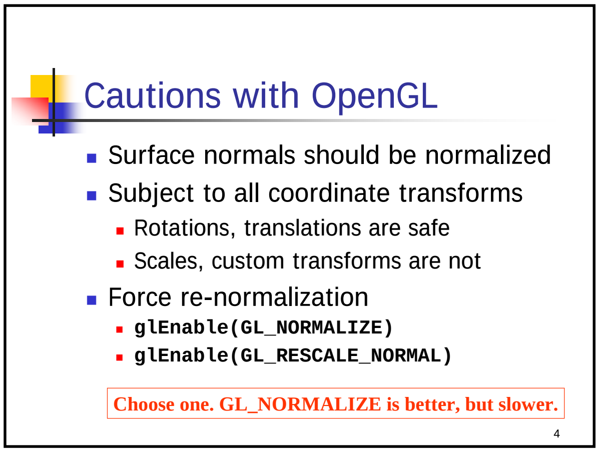


Gouraud Shading (2)

- Use an average

$$n = \frac{n_1 + n_2 + n_3 + n_4}{|n_1 + n_2 + n_3 + n_4|}$$

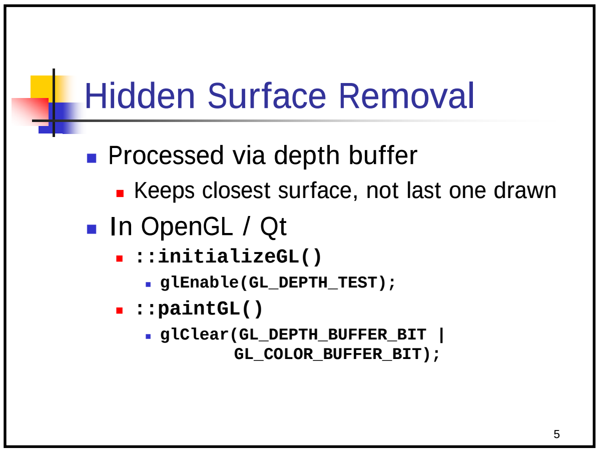
- Finding n_i
  - Use a cross-product of CCW vertices
- Trick
  - Use small surface patches



Cautions with OpenGL

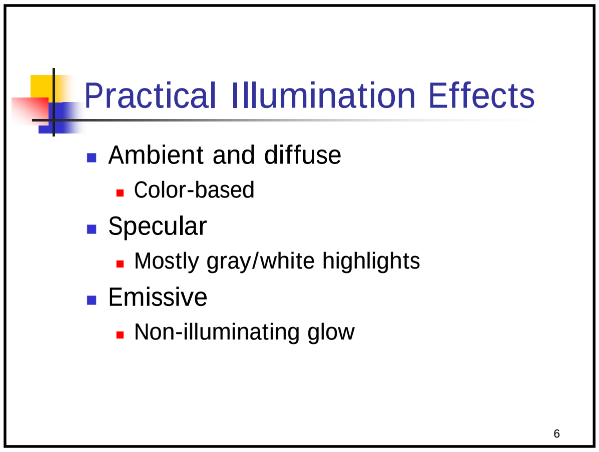
- Surface normals should be normalized
- Subject to all coordinate transforms
  - Rotations, translations are safe
  - Scales, custom transforms are not
- Force re-normalization
  - `glEnable(GL_NORMALIZE)`
  - `glEnable(GL_RESCALE_NORMAL)`

Choose one. GL_NORMALIZE is better, but slower.



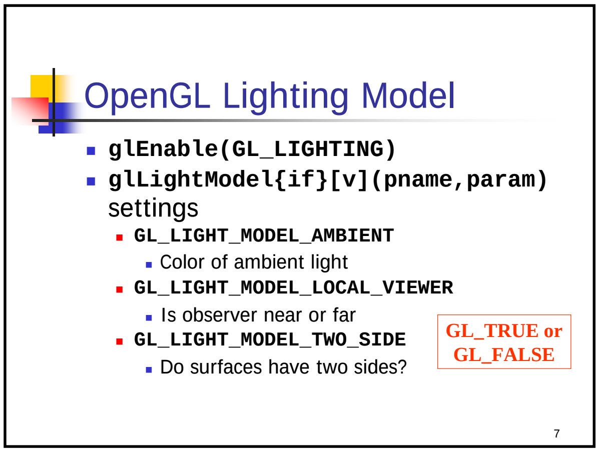
Hidden Surface Removal

- Processed via depth buffer
  - Keeps closest surface, not last one drawn
- In OpenGL / Qt
  - `::initializeGL()`
    - `glEnable(GL_DEPTH_TEST);`
  - `::paintGL()`
    - `glClear(GL_DEPTH_BUFFER_BIT | GL_COLOR_BUFFER_BIT);`



Practical Illumination Effects

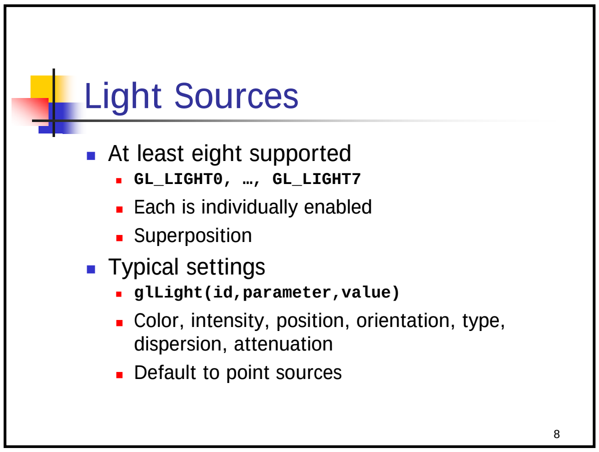
- Ambient and diffuse
  - Color-based
- Specular
  - Mostly gray/white highlights
- Emissive
  - Non-illuminating glow



OpenGL Lighting Model

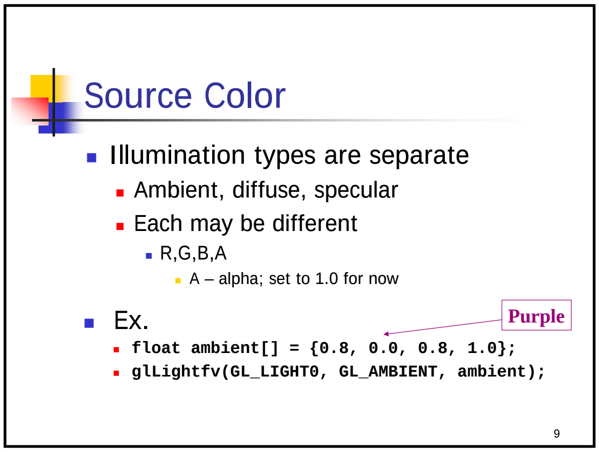
- **`glEnable(GL_LIGHTING)`**
- **`glLightModel{if}[v](pname,param)`** settings
  - **`GL_LIGHT_MODEL_AMBIENT`**
    - Color of ambient light
  - **`GL_LIGHT_MODEL_LOCAL_VIEWER`**
    - Is observer near or far
  - **`GL_LIGHT_MODEL_TWO_SIDE`**
    - Do surfaces have two sides?

GL_TRUE or GL_FALSE



Light Sources

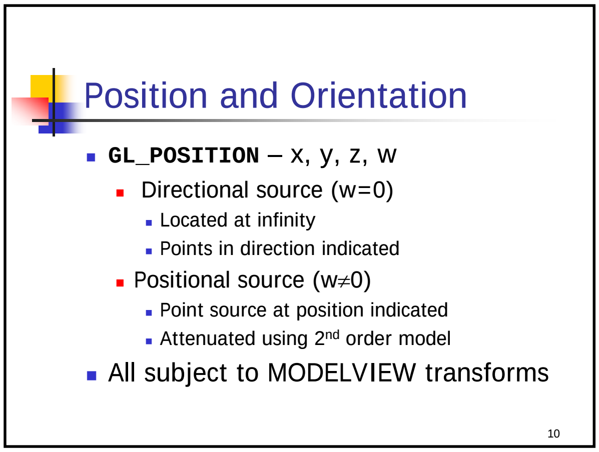
- At least eight supported
  - **`GL_LIGHT0, …, GL_LIGHT7`**
  - Each is individually enabled
  - Superposition
- Typical settings
  - **`glLight(id,parameter,value)`**
  - Color, intensity, position, orientation, type, dispersion, attenuation
  - Default to point sources



Source Color

- Illumination types are separate
  - Ambient, diffuse, specular
  - Each may be different
    - R,G,B,A
      - A – alpha; set to 1.0 for now
- Ex.
  - **`float ambient[] = {0.8, 0.0, 0.8, 1.0};`**
  - **`glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);`**

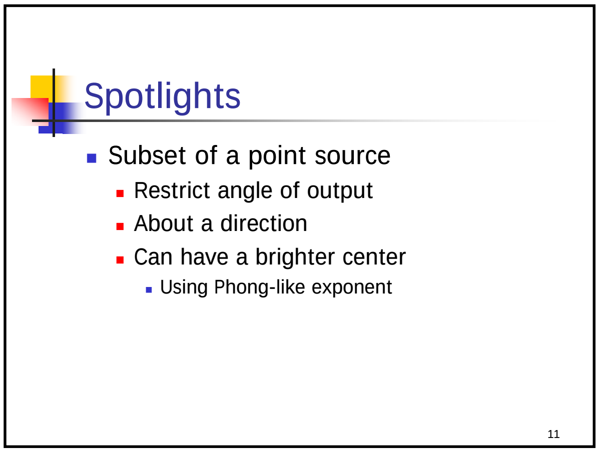
Purple



Position and Orientation

- **GL_POSITION** – x, y, z, w
  - Directional source (w=0)
    - Located at infinity
    - Points in direction indicated
  - Positional source (w≠0)
    - Point source at position indicated
    - Attenuated using 2nd order model
- All subject to MODELVIEW transforms

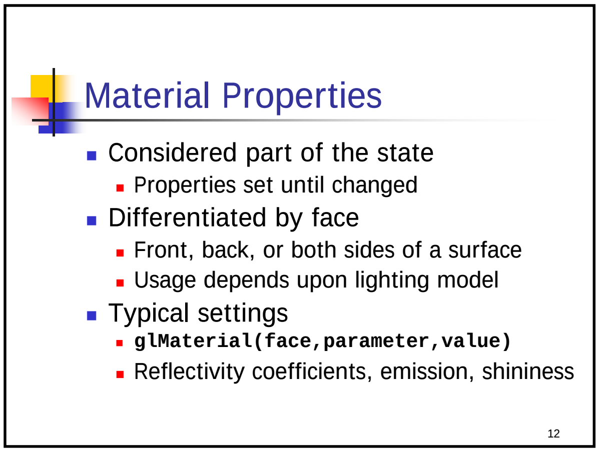
10



Spotlights

- Subset of a point source
  - Restrict angle of output
  - About a direction
  - Can have a brighter center
    - Using Phong-like exponent

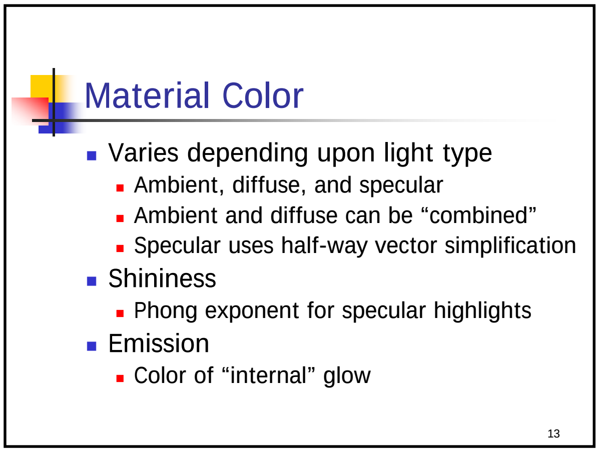
11



Material Properties

- Considered part of the state
  - Properties set until changed
- Differentiated by face
  - Front, back, or both sides of a surface
  - Usage depends upon lighting model
- Typical settings
  - `glMaterial(face,parameter,value)`
  - Reflectivity coefficients, emission, shininess

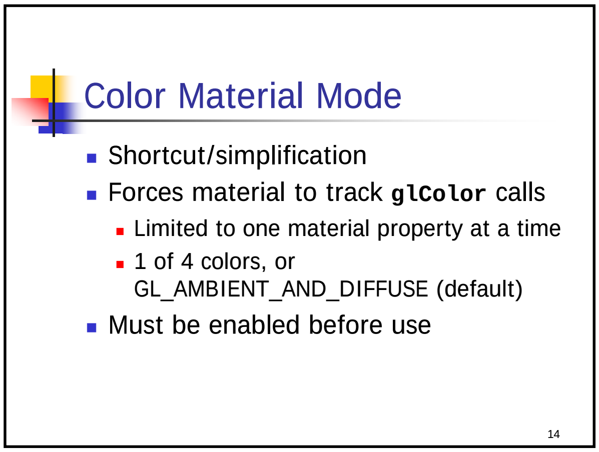
12



Material Color

- Varies depending upon light type
  - Ambient, diffuse, and specular
  - Ambient and diffuse can be “combined”
  - Specular uses half-way vector simplification
- Shininess
  - Phong exponent for specular highlights
- Emission
  - Color of “internal” glow

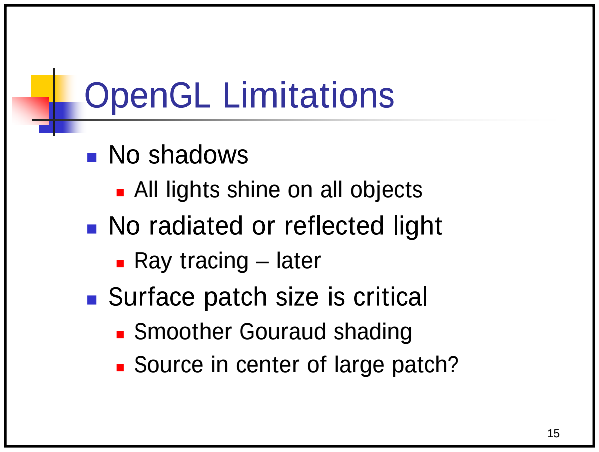
13



Color Material Mode

- Shortcut/simplification
- Forces material to track **`glColor`** calls
  - Limited to one material property at a time
  - 1 of 4 colors, or GL_AMBIENT_AND_DIFFUSE (default)
- Must be enabled before use

14



OpenGL Limitations

- No shadows
  - All lights shine on all objects
- No radiated or reflected light
  - Ray tracing – later
- Surface patch size is critical
  - Smoother Gouraud shading
  - Source in center of large patch?

15