Computer Graphics Seminar

MTAT.03.305

Spring 2020

Raimond Tunnel
Previously...

Data

Vertex transformations

Culling & Clipping

Vertex shader

Rasterization

Fragment shading

Visibility tests & Blending

\[ M = M_1 \cdot M_2 \cdot M_3 \cdot \ldots \]

\[ P \cdot V \cdot M \cdot v = \left( \frac{v_x}{v_w}, \frac{v_y}{v_w}, \frac{v_z}{v_w} \right) \]

\[ v_0 \]

\[ v_1 \]

\[ v_2 \]
Previously...

- We define our geometry (points, lines, triangles)
- We apply transformations (matrices)

\[
\begin{pmatrix}
\cos(45^\circ) & -\sin(45^\circ) \\
\sin(45^\circ) & \cos(45^\circ)
\end{pmatrix}
\]

When is this true?
Now We Add Color?

What exactly is here?

Adding color...?
Material Properties

- We want GPU to take into account a color property when rendering some geometry.

What is depicted here?

http://cgdemos.tume-maailm.pri.ee/
Material Properties

● We want GPU to take into account a color property when rendering some geometry.

Red cube?
Two red trapezoids?
Flat red polygon?

http://cgdemos.tume-maailm.pri.ee/
What is color?
What Is Color?

- Spectrum of the light reflected off a surface.
What Is Color?

- Spectrum of the light reflected off a surface.
- In 3D it is not enough to just say that a thing is red.
What Is Color?

- Spectrum of the light reflected off a surface.
- In 3D it is not enough to just say that a thing is red.
- We need to say that somewhere we have some kind of light source.
Directional Light

- Ok, we define a light direction and our surface
Directional Light

- Light reflects by the incident angle, right?
Directional Light

- The viewer does not see the surface at x=5?
Diffuse Reflection

- Reality – our surfaces are diffusely reflective!
Diffuse Reflection

- Light entering at a specific angle
Diffuse Reflection

- Photon excites an atom
Diffuse Reflection

• The energy is transferred to the next atom
Diffuse Reflection

- The energy is transferred to the next atom
- Some energy is absorbed
Diffuse Reflection

- Excited atoms vibrate, giving off heat
Diffuse Reflection

- Finally photon exits the surface
Diffuse Reflection

- In a quite random direction
Diffuse Reflection

• This is *generally* how pigments work

Nice post: https://physics.stackexchange.com/a/240848
Diffuse Reflection

• Can be caused by other reasons too!
Diffuse Reflection

- Can be caused by other reasons too!
- For example **structural coloration** in nature.

All of these feathers are actually brown.

https://en.wikipedia.org/wiki/Pollia_condensata
Diffuse Reflection

- Can be caused by other reasons too!
- For example *structural coloration* in nature.
Diffuse Reflection

- We assume **diffuse light scatters uniformly**
Diffuse Reflection

- So all we need now is the angle between the surface normal and the Light Direction.
  
  More correct is: direction towards the light

- Why this angle?
Diffuse Reflection

Hint?
Diffuse Reflection

- The actual light energy per surface unit depends on the angle.

\[ \frac{1}{\cos(26.57^\circ)} \approx 1.12 \]

\[ \frac{1}{\cos(55.82^\circ)} \approx 1.72 \]
Diffuse Reflection

- In CG we are interested in how much light reaches 1 surface unit (pixel).

\[
\cos(26.57^\circ) \approx 0.89 \\
\cos(55.82^\circ) \approx 1.72
\]
Diffuse Reflection & Directional Light

- Given a surface point and a light source, we can calculate the color of that surface point.
- We use a \textit{cosine} between the \textit{surface normal} and a \textit{vector} \( l \) going towards the light source.
The Dot Product

- To find the cosine of the angle, we can use a scalar / **dot product** operation.

\[ v \cdot u = \|v\| \cdot \|u\| \cdot \cos(\text{angle}(u, v)) \]

Geometric definition
The Dot Product

- To find the cosine of the angle, we can use a scalar / dot product operation.

\[ v \cdot u = \| v \| \cdot \| u \| \cdot \cos(\text{angle}(u, v)) \]

\[ v \cdot u = v_1 \cdot u_1 + v_2 \cdot u_2 + v_3 \cdot u_3 \]

Algebraic definition
The Dot Product

- To find the cosine of the angle, we can use a scalar / **dot product** operation.

\[ \mathbf{v} \cdot \mathbf{u} = \|\mathbf{v}\| \|\mathbf{u}\| \cdot \cos(\text{angle} \,(\mathbf{u}, \mathbf{v})) \]  

Geometric definition

\[ \mathbf{v} \cdot \mathbf{u} = v_1 \cdot u_1 + v_2 \cdot u_2 + v_3 \cdot u_3 \]  

Algebraic definition

- When we have normalized (unit) vectors, geometric definition simplifies to:

\[ \mathbf{v} \cdot \mathbf{u} = \|\mathbf{v}\| \|\mathbf{u}\| \cdot \cos(\alpha) = 1 \cdot 1 \cdot \cos(\alpha) = \cos(\alpha) \]
The Dot Product

- So if we put those two definitions together:

\[ \mathbf{v} \cdot \mathbf{u} = v_1 \cdot u_1 + v_2 \cdot u_2 + v_3 \cdot u_3 = \cos(\alpha) \]

This should be quite easy for the computer to calculate...
The Dot Product

- The dot product and the cosine between two vectors are used quite often in CG.
The Dot Product

- Dot product of two vectors $u$ and $v$ is the same as vector multiplication.

$$v \cdot u = v_1 \cdot u_1 + v_2 \cdot u_2 + v_3 \cdot u_3 = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = v^T u$$

- So for our surface point we get:

$$\text{Intensity} = \text{directionTowardsLight}^T \cdot \text{surfaceNormal}$$

$$I = l^T \cdot n$$

$$I \in [0, 1]$$
Diffuse Reflection & Directional Light

- Two missing things:
  - Intensity of the light source $L \in [0, 1]$
  - Reflectivity of our material $M \in [0, 1]$
Diffuse Reflection & Directional Light

• Also the color!
• We apply to each of 3 RGB channels.

\[ I_R = n^T \cdot l \cdot L_R \cdot M_R \]
\[ I_G = n^T \cdot l \cdot L_G \cdot M_G \]
\[ I_B = n^T \cdot l \cdot L_B \cdot M_B \]

Light that light source emits
Light that material reflects
$L_{diff} = 1.0$
$L_{diff} = 0.7$
\[ L_{\text{diff}} = 0.7 \]

\[ M_{\text{diff}} = 1.0 \]

\[ n \cdot l \approx 0.9 \]
$L_{diff} = 0.7$

$M_{diff} = 0.5$

$n \cdot l \approx 0.9$
Diffuse Reflection & Directional Light

What color are the apples if red light shines upon them?

What is wrong with this example? (2+ things)
Point Light

- Point lights work the same way, but the light source is a point.
Point Light

• Sometimes distance *attenuation* parameters are added.
Point Light

• Distance **attenuation** parameters:
  
  • In OpenGL:
    
    \[
    \text{attenuation} = \frac{1}{k_c + k_l \cdot d + k_q \cdot d^2}
    \]

  • In Three.js:
    
    `PointLight(hex, intensity, distance)`

*Distance - If non-zero, light will attenuate linearly from maximum intensity at light position down to zero at distance.*

https://threejs.org/docs/#api/en/lights/PointLight
Ambient Light

- So, now we have 2 lights and a diffuse surface.
- Are we OK?
Ambient Light

- World contains much more than 1 cube and a light source.
- Do you know what scene this is?
- Calculating every reflection from every other object is time-consuming.
- What can we do?
Ambient Light

• Ambient light source – estimates the light reflected off of other objects in the scene
Ambient Light

- Ambient light source – estimates the light reflected off of other objects in the scene
- Ambient material property – how much object reflects that light (usually same as diffuse)
Ambient Light

- Ambient light source – estimates the light reflected off of other objects in the scene
- Ambient material property – how much object reflects that light (usually same as diffuse)
• So together with diffuse lighting we get:

\[ I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} \]

\[ I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} \]

\[ I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} \]

Red channel

Green channel

Blue channel

What could go wrong?
Is this it?

• Well, we have already made a very rough approximation of reality with the ambient term.
• Is there anything else that we have forgotten?
Specular Reflection

- Materials also do reflect light specularly.
Specular Reflection

- Materials also reflect light specularly.
- Especially varnished materials and metals!
Specular Reflection

- Materials also reflect light specularly.
- Especially varnished materials and metals!
- Specular reflection is the direct reflection of the light from the environment.
Specular Reflection

- Materials also reflect light specularly.
- Especially varnished materials and metals!
- Specular reflection is the direct reflection of the light from the environment.
- Often we want just a **specular highlight** –
  - that is the **reflection** of the light source!
Specular Highlight

- Depends on the viewer's position.
Specular Highlight

- At point 4, which viewer direction should produce more intense specular highlight?
Specular Highlight

- How to calculate that based on $\beta$?
Specular Highlights

- Ok, so add a specular term based on the actual reflection direction \( (r) \) and viewer direction \( (v) \).

\[
I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} + r^T \cdot v \cdot L_{S_R} \cdot M_{S_R}
\]

\[
I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} + r^T \cdot v \cdot L_{S_G} \cdot M_{S_G}
\]

\[
I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} + v^T \cdot r \cdot L_{S_B} \cdot M_{S_B}
\]
Specular Highlights

\[ I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} + r^T \cdot v \cdot L_{S_R} \cdot M_{S_R} \]

\[ I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} + r^T \cdot v \cdot L_{S_G} \cdot M_{S_G} \]

\[ I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} + v^T \cdot r \cdot L_{S_B} \cdot M_{S_B} \]

Some properties are usually the same in the same channel.

Any errors on the slide?

Is there something missing?
Specular Highlights

- Specular highlight values for different angles:

<table>
<thead>
<tr>
<th>$M_S$</th>
<th>$L_S$</th>
<th>$\beta$</th>
<th>$\sim \cos(\beta)$</th>
<th>$\sim I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>1</td>
<td>10°</td>
<td>0.98</td>
<td>0.25</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>20°</td>
<td>0.94</td>
<td>0.24</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>30°</td>
<td>0.87</td>
<td>0.22</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>40°</td>
<td>0.77</td>
<td>0.19</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>50°</td>
<td>0.64</td>
<td>0.16</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>60°</td>
<td>0.5</td>
<td>0.12</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>70°</td>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>80°</td>
<td>0.17</td>
<td>0.04</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>90°</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Assume we are dealing with one channel (e.g. red)
Assume the channel values are between $[0, 1]$ (mapped later to $[0, 255]$)

This is actually too little change in the result for such a big change from 10° to 20°.

This is too much for such big angles.
Specular Highlights

- How to increase the contrast? Use a power.

<table>
<thead>
<tr>
<th>β</th>
<th>$\cos^2(\beta)$</th>
<th>$I$</th>
<th>$\cos^3(\beta)$</th>
<th>$I$</th>
<th>$\cos^4(\beta)$</th>
<th>$I$</th>
<th>$\cos^5(\beta)$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10°</td>
<td>0.97</td>
<td>0.24</td>
<td>0.96</td>
<td>0.24</td>
<td>0.94</td>
<td>0.23</td>
<td>0.92</td>
<td>0.23</td>
</tr>
<tr>
<td>20°</td>
<td>0.88</td>
<td>0.22</td>
<td>0.83</td>
<td>0.21</td>
<td>0.78</td>
<td>0.20</td>
<td>0.73</td>
<td>0.18</td>
</tr>
<tr>
<td>30°</td>
<td>0.75</td>
<td>0.19</td>
<td>0.65</td>
<td>0.16</td>
<td>0.56</td>
<td>0.14</td>
<td>0.49</td>
<td>0.12</td>
</tr>
<tr>
<td>40°</td>
<td>0.59</td>
<td>0.15</td>
<td>0.45</td>
<td>0.11</td>
<td>0.34</td>
<td>0.09</td>
<td>0.26</td>
<td>0.07</td>
</tr>
<tr>
<td>50°</td>
<td>0.41</td>
<td>0.10</td>
<td>0.27</td>
<td>0.07</td>
<td>0.17</td>
<td>0.04</td>
<td>0.11</td>
<td>0.03</td>
</tr>
<tr>
<td>60°</td>
<td>0.25</td>
<td>0.06</td>
<td>0.13</td>
<td>0.03</td>
<td>0.06</td>
<td>0.02</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>70°</td>
<td>0.12</td>
<td>0.04</td>
<td>0.04</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>80°</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>90°</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Values above 0.25 are in red
Specular Highlights

- Specify some **shininess** value $c$ for the material

\[
I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} + (r^T \cdot v)^c \cdot L_{S_R} \cdot M_{S_R}
\]

\[
I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} + (r^T \cdot v)^c \cdot L_{S_G} \cdot M_{S_G}
\]

\[
I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} + (r^T \cdot v)^c \cdot L_{S_B} \cdot M_{S_B}
\]
Phong's Lighting Model

\[
I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} + (r^T \cdot v)^c \cdot L_{S_R} \cdot M_{S_R}
\]

\[
I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} + (r^T \cdot v)^c \cdot L_{S_G} \cdot M_{S_G}
\]

\[
I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} + (r^T \cdot v)^c \cdot L_{S_B} \cdot M_{S_B}
\]

Ambient light approximation.
Phong's Lighting Model

\[ I_R = L_{AR} \cdot M_{AR} + n^T \cdot l \cdot L_{DR} \cdot M_{DR} + (r^T \cdot v)^c \cdot L_{SR} \cdot M_{SR} \]

\[ I_G = L_{AG} \cdot M_{AG} + n^T \cdot l \cdot L_{DG} \cdot M_{DG} + (r^T \cdot v)^c \cdot L_{SG} \cdot M_{SG} \]

\[ I_B = L_{AB} \cdot M_{AB} + n^T \cdot l \cdot L_{DB} \cdot M_{DB} + (r^T \cdot v)^c \cdot L_{SB} \cdot M_{SB} \]

Lambertian / diffuse reflectance
Phong's Lighting Model

\[ I_R = L_{AR} \cdot M_{AR} + n^T \cdot l \cdot L_{DR} \cdot M_{DR} + (r^T \cdot v)^c \cdot L_{SR} \cdot M_{SR} \]

\[ I_G = L_{AG} \cdot M_{AG} + n^T \cdot l \cdot L_{DG} \cdot M_{DG} + (r^T \cdot v)^c \cdot L_{SG} \cdot M_{SG} \]

\[ I_B = L_{AB} \cdot M_{AB} + n^T \cdot l \cdot L_{DB} \cdot M_{DB} + (r^T \cdot v)^c \cdot L_{SB} \cdot M_{SB} \]

Phong's specular reflectance term
Phong's Lighting Model

\[ I_R = L_{A_R} \cdot M_{A_R} + n^T \cdot l \cdot L_{D_R} \cdot M_{D_R} + (r^T \cdot v)^c \cdot L_{S_R} \cdot M_{S_R} \]

\[ I_G = L_{A_G} \cdot M_{A_G} + n^T \cdot l \cdot L_{D_G} \cdot M_{D_G} + (r^T \cdot v)^c \cdot L_{S_G} \cdot M_{S_G} \]

\[ I_B = L_{A_B} \cdot M_{A_B} + n^T \cdot l \cdot L_{D_B} \cdot M_{D_B} + (r^T \cdot v)^c \cdot L_{S_B} \cdot M_{S_B} \]

Something still missing?
Blinn-Phong Model

• Sometimes Phong's specular term is replaced with Blinn-Phong's specular term.
Blinn-Phong Model

- Sometimes Phong's specular term is replaced with Blinn-Phong's specular term.
- Instead of viewer direction and reflected light's direction, we use the surface normal ($n$) and a half angle vector ($h$) between the light source and the viewer.
Blinn-Phong Model

- There are some differences
- These are not the only two possibilities

DEMO 2: http://cgdemos.tume-maailm.pri.ee/
Now You Know

Data

Vertex transformations

Culling & Clipping

Rasterization

Fragment shading

Visibility tests & Blending

Light Source

Vertex shader

Fragment shader

$M_{amb}$

$M_{diff}$

$M_{spec}$

$L_{amb}$

$L_{diff}$

$L_{spec}$

$n$
The Standard Graphics Pipeline

Data

Vertex transformations

Culling & Clipping

Rasterization

Fragment shading

Fragment shader

Visibility tests & Blending

Vertex shader

\[ M = M_1 \cdot M_2 \cdot \ldots \]

\[ P \cdot V \cdot M \cdot v \]

\[ \left( \frac{v_x}{v_w}, \frac{v_y}{v_w}, \frac{v_z}{v_w} \right) \]

Light Source

\[ L_{\text{amb}}, L_{\text{diff}}, L_{\text{spec}} \]

\[ M_{\text{amb}}, M_{\text{diff}}, M_{\text{spec}} \]
Conclusion

- Computer graphics can be used to create an illusion of reality

  Reality \(\rightarrow\) Approximation \(\rightarrow\) Mathematical description \(\rightarrow\) Approximation \(\rightarrow\) Replication

- First approximation is of the shape – geometry
- GPU wants those **triangles**
- **Vertices** and **transformation matrices**
Conclusion

- Many ways to approximate lighting (Lambert, Phong, Blinn), reflections, refractions, shadows...

- **Ambient, diffuse, specular terms**

\[ I = L_A \cdot M_A + n^T \cdot l \cdot L_D \cdot M_D + \left( r^T \cdot v \right)^c \cdot L_S \cdot M_S \]
Thanks for Listening!
Next Time

- Shaders in WebGL
- Shader Programming Workshop (all of this in practice)
- Bring your laptop!

three.js

Notepad++

WebGL ™