Computer Graphics

COMP3421/9415

2021 Term 3 Lecture 13

What did we learn last lecture?

Phong Lighting

- A complete algorithm
- Ambient, Diffuse and Specular Lighting
- Working with different light types
- And with multiple lights

What are we covering today?

Advanced Lighting

- What can we add to Phong Lighting?
- Colour Perception and Gamma Correction
- Lightmapping
- HDR
- Blinn-Phong

Advanced Lighting

Phong Lighting

Let's look at some places we could improve it

- Multiple Lights are inefficient
  - Is there a way to render some of our lights beforehand, so they're not taking up frame time?

- Light values are clamped to 0.0-1.0
  - Phong Lighting can add more than 1.0 light, what do we do with that?

- Dot Products are clamped to 0.0-1.0
  - What happens when a dot product could be negative like in reflected light?
- There are no shadows
  - We're going to need to learn a few other techniques before we can take this on!

Gamma Correction

The Perception of Colour and Light

Human Perception vs Numerical Output

- A scale of brightness from 0.0 to 1.0
- Unfortunately, we perceive brightness differently to how our monitors display it

- Old CRT monitors used to display colour similarly to our perception

    - An exponential rather than linear scale of light per voltage
    - Known as the monitor's Gamma
- Current monitors replicate the same Gamma curve

Image credit: learnopengl.com

RGB Values vs Actual Output

Our basic lighting is all 0.0 - 1.0 scale

- We've assumed linear colour and light scale
- But our output is not linear!

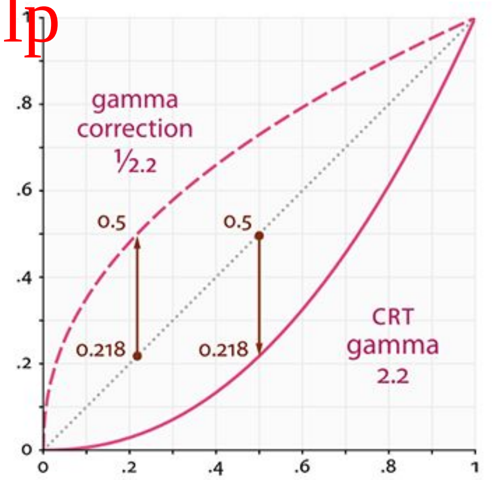
- RGB (0.2,0.2,0.2) does not output twice as many photons as (0.1,0.1,0.1)

- We need to correct our Gamma!

Gamma Correction

Counteracting the gamma curve

- If we intended our RGB values to be linear . . .
- And our monitor's output is non-linear
- We should change our RGB values
- Gamma corrected values negate the monitor's gamma curve
- The most common gamma curve is 2.2 (based on CRT monitors)



A graph of RGB values to actual output
Image credit: learnopengl.com

OpenGL Gamma Correction

sRGB

- sRGB is a colour space that automatically corrects a gamma of 2.2
- We can use a sRGB framebuffer that converts RGB values as we write to it

- or
- We can adjust values as we finalise fragment colours

- Games often allow us to manually adjust gamma in case we're not exactly on a 2.2 gamma

Gamma complications

If all of our monitor output is skewed . . .

- Is it ok if we make art (like textures) with the gamma curve
- Then display it with roughly the same curve?
- Sadly not!
- Our lighting calculations are all linear
- So we're going to correct gamma in our output
- Which means we might need our artists to create content using corrected gamma also!

Differences with Gamma Correction

Image credit: learnopengl.com, taken from Wolfire Games Blog

Lightmapping

Ambient Phong Lighting

Ambient lighting is a very rough guess

- Phong Lighting just assumes ambient lighting
- It's consistent and reaches every surface equally
- We touched on Ray Tracing, a very accurate global illumination technique
- But very time/performance consuming
- What if we could "pre-bake" our global illumination?

Lightmapping

Difficult and Complex lighting is calculated in advance

- Not just Ray Tracing
- Any illumination technique can be calculated in advance
- Lighting is calculated then "baked" into a lightmap
- A lightmap is a buffer of all the lighting in a scene
- Works ok for static objects and static lights
- Doesn't work for dynamic objects and lights

Lightmap Example

An example of the lighting in a scene being baked into a 2D buffer

Image credit: Wikipedia user Narpas

Creating Lightmaps

Creation

- Any lighting can be pre-baked
- Historically used for unchanging lights like the Sun
- and unchanging environments like buildings etc
- Lighting is calculated in advance and saved to a buffer
- Usually this is a specific light buffer that vertices map to (just like a texture)

Using Light maps

What the realtime lighting shaders do

- When a light map exists:
- Instead of using the Ambient light equation
- We'd read lighting from the light map just like a texture
- This means the vertices of any static geometry would have a light map coordinate mapping to "lumels" in the light map

This method is as old as Quake (1996)

- They also used multiple light maps for the same area to simulate things like flickering lights by switching light maps

Break Time

We're getting closer to actual games now

- Some things, like lightmapping are used extensively (especially in older games)

- You may have heard of Ambient Occlusion (calculation of micro shadows in realtime)

- Gamma Correction is something that you will have probably seen!
- The more we learn in this course, the more you will understand how the visuals in the games we play are made

HDR

Clamping Light values

What's the range of our light values?

- RGB from 0.0-1.0 in our previous lighting calculations
- What happens in regions of bright light?
  - > 1.0 is clamped to 1.0
- Loss of contrast between colours
  - Leads to a loss of detail

Image credit: learnopengl.com

High Dynamic Range

Originally a concept from Photography

- Multiple photographs taken at different exposure levels
- High Exposure to capture detail in dark areas

- Low Exposure to capture detail in brighter areas
- Combined into a single photo

- Captures detail in both bright and dark areas

HDR Photography

Images credit: Kevin McCoy

Result of Processing multiple HDR images
with "Natural Tone Mapping"
Image credit: Sebastian Nibisz

HDR in OpenGL

How do we implement HDR in realtime graphics?

- Remove the clamping on light values
- Transform the new unclamped light values back into 0.0-1.0
- Write the final information into the framebuffer

In OpenGL

- Create an intermediate frame buffer that stores floating point RGB values
- Use a Tone Mapping algorithm to convert those values to 0.0-1.0

Frame Buffers

A new concept . . . a buffer that stores "pixels"

- We can create buffers in OpenGL
- We can create a buffer that is exactly the same size as the screen
- Each element of the buffer maps to a screen/window pixel
- Each element can contain RGB values for example
    - In HDR, we'd use floats instead of 0.0-1.0 clamped values
- A buffer like this is very similar to a texture!
- This is not the last time we'll be using these!

Tone Mapping

An algorithm to convert exposure levels to Low Dynamic Range

- A simple idea would be to divide HDR values by their maximum
  - Resulting in 0.0-1.0

- But much more complex algorithms exist that:
  - Maintain contrast in local areas
  - Treat bright and dark areas differently

- We can also dynamically change our HDR algorithm
  - Like indoors vs outdoors as we walk in between areas

Blinn-Phong

Measuring our Angles

We've been using dot products to work with angles

- Dot products allow us to go from vectors to angles easily
- Also means we have simple maths for our GPUs
- But we've been clamping them between 0.0 and 1.0
- What issues can this cause?

Angles above 90°

Possible issues at very low Phong Exponents

- Left shows intended Phong lighting
- Right shows a V to R angle higher than 90°
- Our dot products will reduce our lighting to zero at angles > 90°

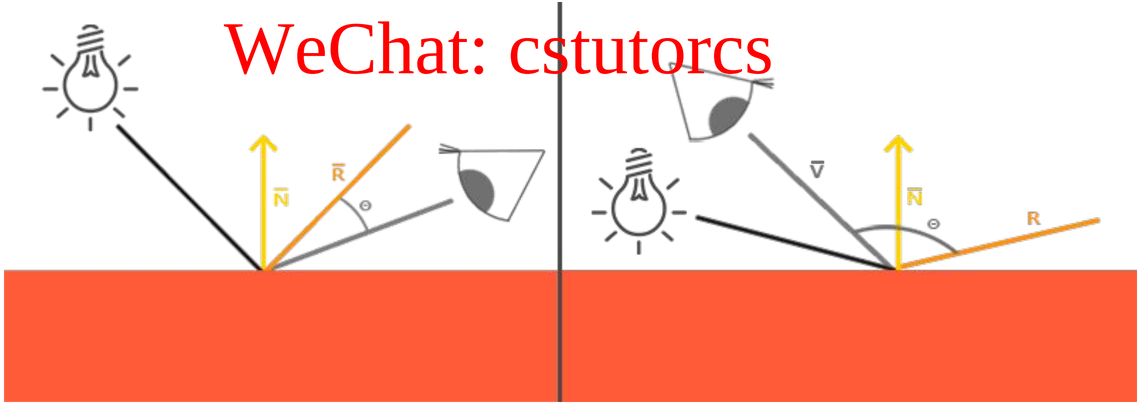


Image credit: learnopengl.com

Dot Product based artifacts

A hard cutoff edge in situations with low Phong exponents

Image credit: learnopengl.com

Blinn-Phong

Can we make sure we never have a > 90° angle?

- Blinn-Phong is a modification of specular Phong lighting
- Phong lighting will reflect the light direction

- Then dot product with the viewer direction
- Blinn-Phong will instead create a "Halfway Vector"

- Halfway between the light and the viewer
- Then test the dot product of the Halfway and the Normal
- It is much harder for the angle between Halfway and Normal to be > 90°

Blinn-Phong

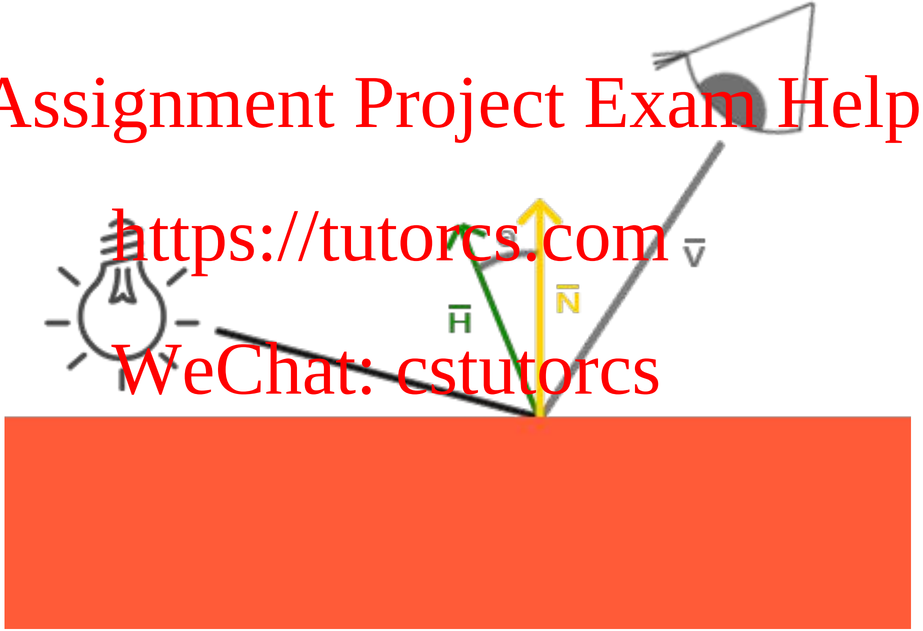


Image credit: learnopengl.com

More Advanced Lighting

Where can we go from here?

Things we haven't talked about yet:

- Shadows
- Direct Reflections (mirrors etc)
- Techniques for hundreds of lights at once
- Screen space effects
    - Bloom
    - Blur
    - Motion Blur
    - Anti-Aliasing
    - etc

What did we learn today?

Some addons to Phong Lighting

- Some corrections, some addons
- Gamma Correction
- Lightmapping
- HDR
- Blinn-Phong
- There are a lot more that we haven't seen yet!