Computer Graphics

COMP3421/9415
2021 Term 3 Lecture 14

What did we learn last lecture?

A Variety of Advanced Lighting

- Addons to Phong Lighting
- Gamma Correction
- Lightmapping
- HDR
- Blinn Phong

What are we covering today?

Lighting Maps

- Additional Surface Details
- Aiding in lighting effects
- Allowing low complexity geometry to appear to have more features

Maps

Revision on Textures

Textures are a type of Map, stored in a Material

- Usually 2D Colour data
  - More specifically diffuse reflectivity!
- Mapped to an object via texture coordinates
- Texture coordinates are part of vertex attributes
- Allows us to sample texels from the texture in our shaders
- Gives us very high detail of colour on surfaces

What is a map?

A more general form of a Texture

- Maps are some information about the surface of an object
- Usually 2D

- Can store different types of data
- Mapped to vertices via coordinates stored in the verts

- Maps are like textures but they come in different types
- A texture is a Diffuse Map (storing diffuse reflective colour)
- We're going to add some other maps!

Different maps

Going beyond textures

- Diffuse Maps - colour
- Specular Maps - shininess
- Normal Maps - surface features

- Each map stores different information

- Allows us to sample from a different map for different functions in the fragment shader
- Maps don't have to be the same resolution as each other
- We can store multiple maps in the one material for an object

Why use maps?

Finer grained detail on surfaces

- Instead of needing multiple objects
- We can have simpler geometry
- And detailed maps representing surface detail
- We used this justification for textures,
- It makes just as much sense for reflectivity and surface details

Diffuse Maps

Diffuse Maps

(Textures)

- We're renaming a concept, but we're already familiar with it
- Involved in the Ambient and Diffuse steps of lighting

Image credit: learnopengl.com

Using Diffuse Maps

We already know how to use Textures!

- Map texture coordinates from vertices to texels
- Splitting up the Phong lighting equation
- Use Diffuse maps for Ambient and Diffuse
- Split the specular calculation to use a different map

Specular Maps

Specular Maps

A map of shininess

- This map ignores any diffuse colour reflections
- Purely areas of an object that directly reflect light
- Often greyscale (doesn't change the colour of light it reflects)
- But can be coloured to represent materials like gold that tint reflections

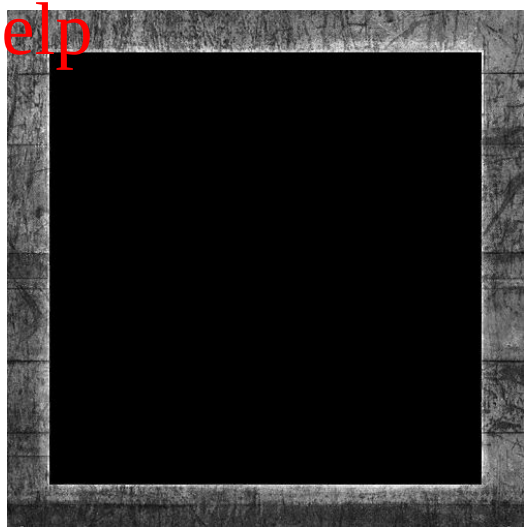


Image credit: learnopengl.com

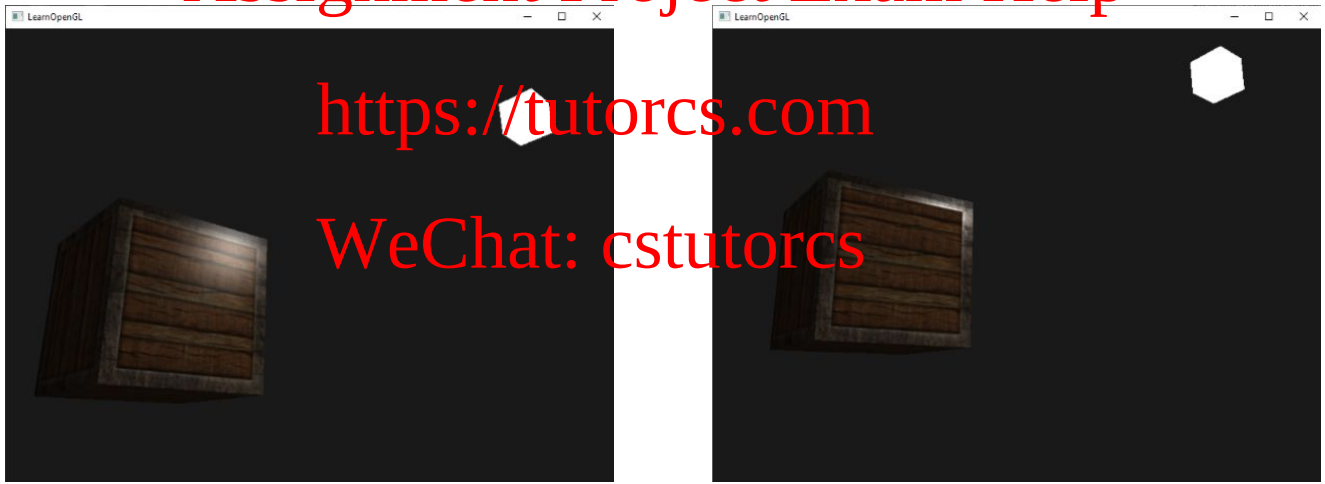
Using Specular Maps

Like having another texture

- Sample from the specular map using texture coordinates
- Specular map is only involved in specular lighting calculations
- Multiply the specular lighting by the sample from the specular map

Comparison of Lighting using maps

A diffuse map vs both diffuse and specular maps



Images credit: learnopengl.com

Blinn-Phong Algorithm with maps

Our algorithm stays the same

$$I_{total} = i_a * k_a + k_d * (L.N) * i_d + k_s * (H.N)^a * i_s$$

- k_a and k_d are sampling from the Diffuse map
- k_s is sampled from the Specular map

Creating Maps

2D Images

- The example specular map is just the diffuse texture modified
  - The metal rim is turned greyscale
  - The wooden centre is blacked out (meaning it will have no specular reflection)
- We will often rely on artists to create different maps
- Since texture coords are 0.0-1.0, we don't need to have our maps at the same resolution

Break Time

Homework (not really)

- Choose a game or film (one with a significant amount of CG or VFX)
- *Pull it to pieces!*

- Can you identify what techniques are in use?
- Are there things you don't understand but are definitely graphics effects?

- Go deep . . . look closely at lighting and shadows
- What changes when you move around?
- What looks like interesting art vs interesting processing?
- Can you tell which tricks are being used to conserve processing time?

Normal Maps

What's a Normal Map?

Definitely not an abnormal map

- We've interpolated colours and reflectivity
- What else do we use to calculate lighting?
- Surface Normals!
- RGB: a 3 float vector
- Normal: a 3 float vector!!!
- This looks too easy

What do Normal Maps do?

A simulation of detailed surface geometry

- If our normals change per fragment
- The surface appears to have more direction changes

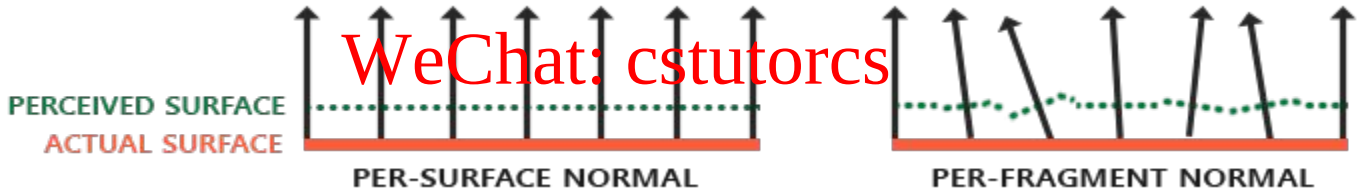


Image credit: learnopengl.com

What do Normal Maps do?

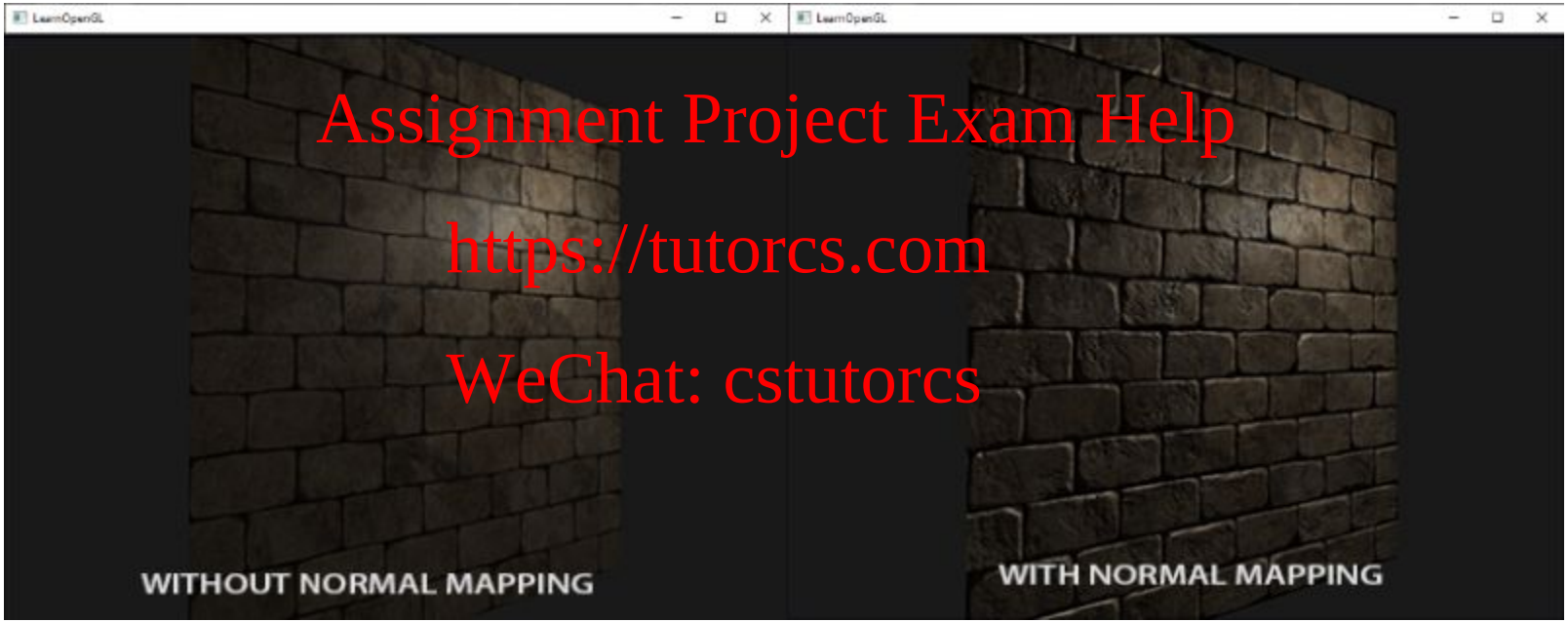


Image credit: learnopengl.com

Effects of Normal Maps

Changing Normal Directions

- Significantly alters lighting calculations
- Especially in Specular
- Enables definition of shapes where there's no geometry

Encoding Normals

3D vectors can be viewed as RGB

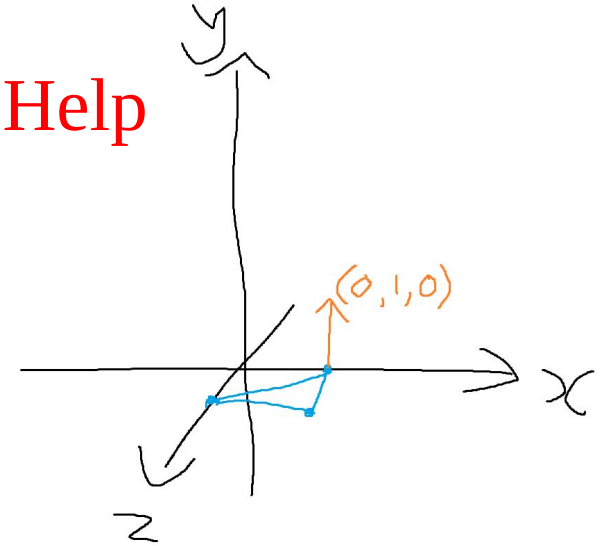
- If we want to view a normal map
- It's majority blue, (0,0,1) pointing towards us along the z axis
- You can see the colour changing on the edges of the bricks as the normals aim further away from the z axis

Image credit: learnopengl.com

Using Normal Maps

Is this the same as Diffuse and Specular maps?

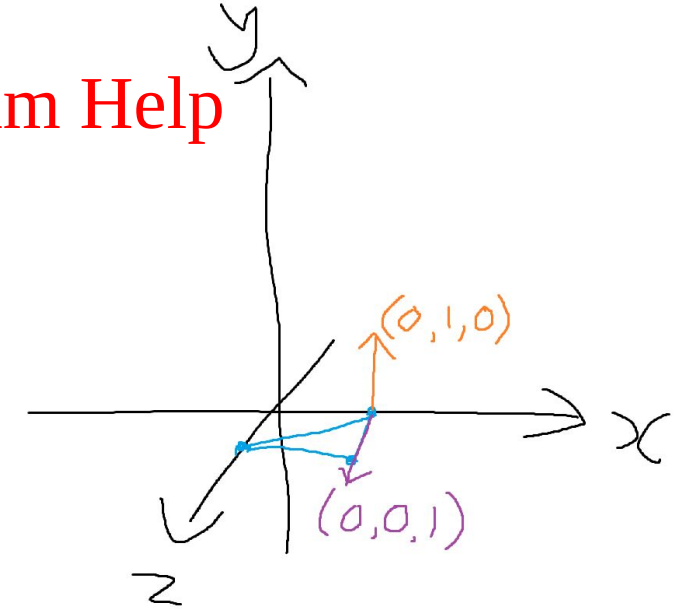
- Intuition says:
  - Sample the normal
  - Replace N
  - Calculate as we would previously
- Here's an example:
  - A triangle is made up of: (1,0,0), (0,0,1),(1,0,1)
  - It's default surface normal would be (0,1,0)



Using Normal Maps

Using the mapped normals

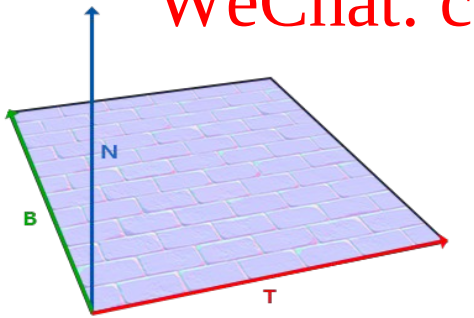
- If the average normal in the map is (0,0,1)
- But the "expected" normal of the triangle is (0,1,0)
- Do we have an issue?
- Normals are not aiming "out" of the surface as expected
- This could be much worse if the triangle is facing away from the Z axis!

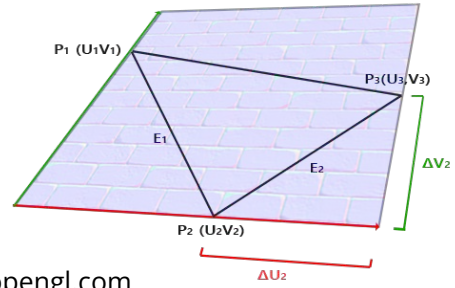


Tangent Space

Transforming our Normal Map to align with the triangle surface

- Map space has its own coordinate system
  - Three axes, the Tangent, Bitangent and Normal
- The triangle exists in the world's coordinate system
- But the triangle's texture coordinates are in map space!





Images credit: learnopengl.com

Transformation Maths (abbreviated)

Finding the Axes of Tangent Space

- N is the surface normal
- We need to find T (tangent) and B (bitangent)

- Direct relationship between texture coords in map space
- . . . and world coords in vertices

- Using rates of change between coordinates
- We can solve an equation for the T and B and build a set of axes in world space
- This allows us to rotate the normals from the map to match the surface

Efficiency and Benefits of Normal Mapping

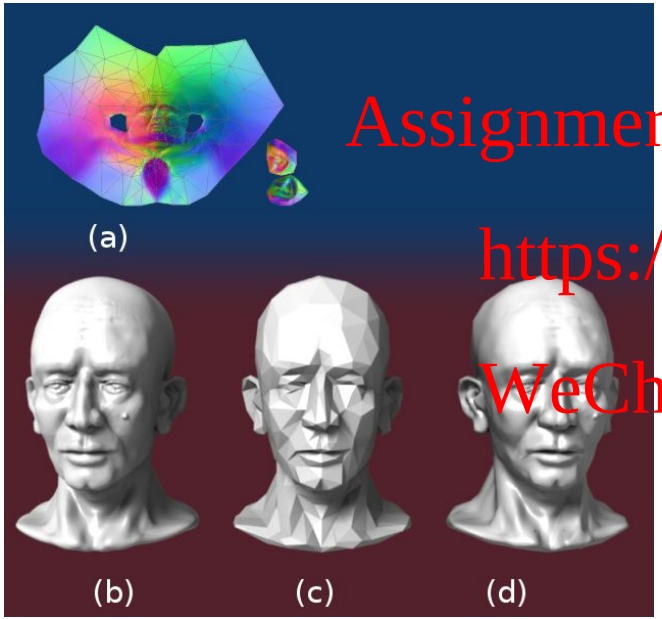
How much does it cost?

- If we have to calculate a set of axes and a transform for every triangle . . .
- Are we getting enough value to justify the expense?
- This is a visual quality decision and is subjective

Benefits?

- Geometry detail requirements are much lower with Normal Mapping
- Potentially much more efficient processing for curved surfaces

Normal Mapping Examples



Creating a normal map (a) from a high polygon model (b), then optimising the model (c) and reapplying the normal map (d)
Image credit: Banlu Kemiyatorn

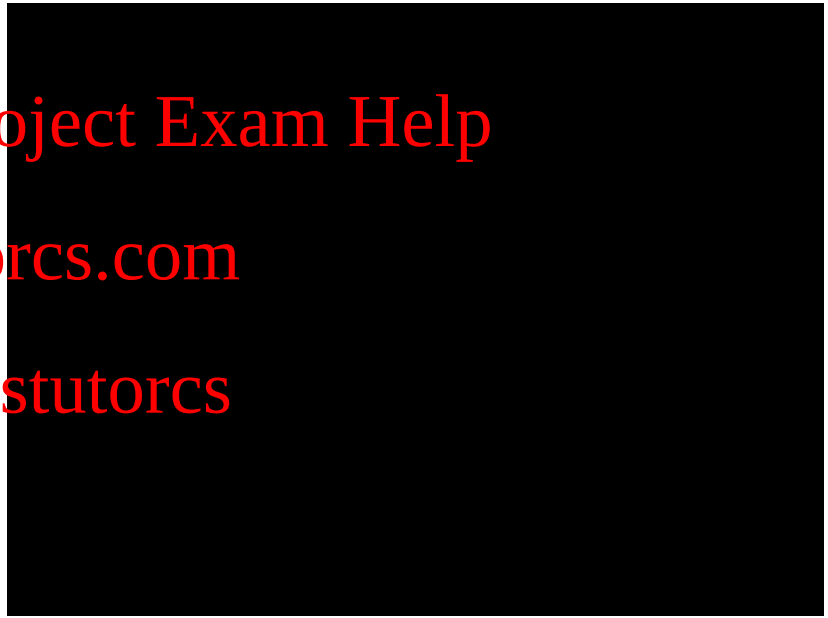


Image credit: Xavier Poon

What did we learn today?

Maps

- Some revision on Diffuse Maps (Textures)
- Adding Specular Maps
  - Light Reflectivity maps
  - Allowing partial reflectivity on an object
- Normal Maps
  - Surface detail mimicking geometry