

Introduction to Graphics - Supervision 2 exercises

Warm up questions

1. *What is the geometric interpretation of the barycentric coordinates of a triangle?*

α , β , and γ represent the perpendicular distance from each of the sides of the triangle, as a proportion of the triangle's width from that side. The points where one of α , β , or γ is **1** and the other two coordinates are **0** are the corners of the triangle.

2. *Which component(s) of the Phong reflection model require interpolating surface normal vector between vertices?*

The surface normal vector is required for the diffuse and specular components.

3. *What are the major differences between OpenGL, DirectX and Vulkan?*

OpenGL and Vulkan are multiplatform, and open standard, whereas DirectX is only for Windows/Xbox, and is proprietary. OpenGL is for general 3D applications, whereas Vulkan and DirectX are focussed on games.

4. *What is GPGPU and what APIs could be used for that?*

The use of the GPU for general purposes other than just graphics. CUDA and OpenCL are APIs for this.

5. *What are "in", "out" and "uniform" variables in GLSL? How are the values of these variables set?*

`in` variables are inputs to the shader, `out` variables are outputs from the shader, and `uniform` variables are parameters that are constant for the primitive which are passed from the application.

6. *What kind of artefacts would you expect to see when rendering using a texture without a Mipmap?*

Jagged edges, and moire-like patterns.

7. *Give an example in which normal (bump) mapping and displacement mapping will produce very different results.*

When applied to a sphere, bump mapping will maintain the perfectly round silhouette of the object, despite altering the appearance of the surface, which is unrealistic. This problem is overcome by displacement mapping, which will actually alter the shape of the object, rather than just the surface normal.

8. *What colours do we call metamers?*

Colours with different light spectra, but which appear to be the same to human eyes.

9. *Why do we need to encode colours for displays?*

Human vision does not perceive brightness linearly, so we have to encode the colours to give more subtle variations in the darker tones.

10. *Which colour spaces are suitable for*

- *Efficiently encoding colours for displays, using possibly few bits;*
- *User interfaces, such as a colour palette tool;*
- *Calculating the perceived difference between colours.*

RGB encoding is suitable for encoding colours for displays, since most displays have red, green, and blue pixels anyway.

Munsell colour system is much more intuitive for humans to navigate than RGB, so it is suited to use in user interfaces.

The CIE colour space is roughly perceptually uniform, so it can be used to calculate perceived differences in colour.

11. *What is the difference between luma and luminance?*

Luma is the brightness of a pixel in units that are gamma-corrected, so luma appears to increase linearly to the human eye.

Luminance is the objective intensity of a pixel in standard units, so it does not appear to increase linearly.

12. *What is the rationale behind sigmoidal tone-curves?*

Tone curves are necessary to map the broad range of possible colours in an image into the more limited range which can be represented digitally. The sigmoidal shape in particular is chosen to mimic the response of analog film, which has been engineered over many years to give good tone-reproduction.

13. *Why do we need/want to simulate glare due to the lens or eye's optics in rendering?*

To make the image seem more realistic.

Longer questions

1. *What is the worst case scenario, in terms of a number of times a pixel colour is computed, when rendering N triangles using the Z-buffer algorithm? How could we avoid such a worst-case scenario?*

The worst case scenario is for the pixel colour to be computed N times, if each successive triangle is closer than the last one. This can be avoided by first computing the Z-buffer for all of the triangles in

the scene, noting which ones are closest, and then only calculating the colour at each pixel for the closest triangle.

2. *Put the following stages of the OpenGL rendering pipeline in the correct order. Very briefly explain what each stage does.*

- *Rasterization*
- *Vertex shader*
- *Fragment shader*
- *Primitive assembly*
- *Clipping*

Vertex shader - Processes each vertex in the scene, transforming its location in 3D space onto a position on the 2D screen. Also handles transformation of the normals for each vertex.

Primitive assembly - Organises the vertices into polygons.

Clipping - Removes or modifies the vertices in the scene so that they are all within the visual volume.

Rasterization - Generates the pixels which must be drawn for each polygon in the scene. Also interpolates attributes between vertices.

Fragment shader - Handles the colouring of each pixel.

3. *Explain the following OpenGL concepts:*

- *Array Buffer (Vertex Buffer)*
- *Element Array Buffer (Index buffer)*
- *Vertex Array (object)*

The vertex buffer contains positions and normals of all the vertices in the scene, with each vertex being stored once for each polygon it is in.

The index buffer contains all of the vertices in the scene, where each vertex has been assigned an index, so that it can be used in multiple polygons, rather than just one.

The vertex array object encapsulates all of the information on vertices, so it contains the buffer objects, and also the format of the vertex data.

4. *How many vertices do you need to model a cube with normals as a triangle mesh, with and without an index buffer?*

Without an index buffer, it requires 36 vertices, since there are 3 per polygon, 2 polygons per face, and 6 faces in to form a full cube.

With an index buffer, you will still need to store some of the vertices more than once, since the normals for each face are different, however it only takes 4 vertices per face this time, since the two triangles on the face share 2 vertices. This gives a total of 24 vertices for a full cube.

5. *How could you use the following texture types to texture a sphere in OpenGL?*

- *2D*

- 3D
- CUBE_MAP

2D texture could be mapped onto the surface of the sphere, producing a pattern.

3D texture could be used to produce a short animation on the surface of the sphere, by mapping different 2D slices of the 3D sphere onto the surface over time.

A cube map texture could be used to create an environment box that would be reflected in the surface of the sphere.

6. *Explain how can OpenGL map a texture to the area that*

- *contains more pixels than the texture;*
- *contains fewer pixels than the texture.*

If the surface contains more pixels than the texture, the values in the texture must be interpolated, probably using bilinear interpolation.

If the surface contains fewer pixels than the texture, the texture must be downsampled, which can be achieved by using a mipmap, or using anisotropic filtering.

7. *Discuss strategies for avoiding tearing artefacts when rendering an animation.*

Tearing artefacts are caused by switching the colour buffers while the front buffer is being copied to the monitor.

One way to avoid this is to use V-Sync, which means that the buffers are only swapped after the last row in the front buffer is copied to the display. This means that the rendering of the next frame is delayed so that it lings up with the refresh rate of the screen.

Another way is to use adaptive sync (FreeSync or G-Sync), which controls the timing of the frames on the display, so that they are rendered as soon as the fram is ready, and the previous display scan has finished.

8. *What is the relation between LMS cone sensitivities and XYZ colour matching functions?*

The XYZ colour space was derived from the LMS colour space. There is a linear transformation that relates the two colour spaces.

9. *Explain the difference between linear and gamma-corrected (display encoded) colour values.*

Gamma-corrected colour values appear to increase linearly in brightness when viewed by the human eye. Linear colour values increase linearly in intensity, but they do not look like they do to humans.

10. *What do the ITU recommendations 709 and 2020 specify?*

They specify two different colour spaces, 2020 with high dynamic range, and 709 with standard dynamic range. The 709 specification uses the same colour space as sRGB, with gamma corrected values.

11. *Explain the purpose of tone-mapping and display-encoding steps in a rendering pipeline.*

Tone-mapping is used to reduce the dynamic range of the image, so that it can be displayed accurately on a screen. This will make the images seem more realistic, by simulating the effects of human vision.

Display encoding adjusts the colours in the image to account for the contrast in the display, and environmental factors.