Computer Graphics 2023
U3087 / M5222
National Taipei University

Final Exam

1. What is the difference between a **local** and a **distant** light in terms of lighting direction? (4%)
   local light: lighting direction is the direction from the shading point to the light; distant light: lighting direction is fixed for the entire scene

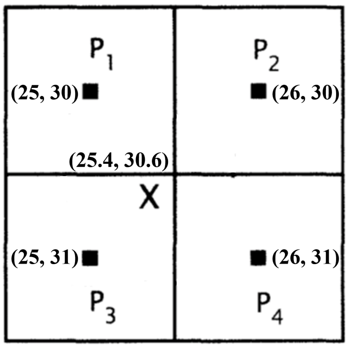
2. Please describe the meaning of **Lambertian term** and how to compute it (4%)
   Illumination on an oblique surface is less than on a normal one. The contribution falls off as $\cos\theta$ where θ is the angle between the lighting direction and the surface normal.

3. Please answer the following questions about **Phong lighting model**:
   (a) The formulation of **diffuse term** does not include the eye position. Why? (4%)
   Diffuse term is used to model rough surface. For this type of surface, the light coming from a direction will scatter in all directions. Thereby, it is view-independent.
   (b) The **specular term** is computed by: $L_s = k_s \cdot I \cdot \max(0, vE \cdot vR)^n$, where vE is the viewing direction and vR is the perfect reflected direction. Please describe the physical meanings of $vE \cdot vR$ and the exponent n, respectively (6%)
   The dot product of vE and vR measures the angle deviated from the perfect reflected direction (vR). Specular lighting gets weaker as the deviation becomes larger; The exponent n controls the speed of declining according to the angle deviated from the perfect reflected direction.

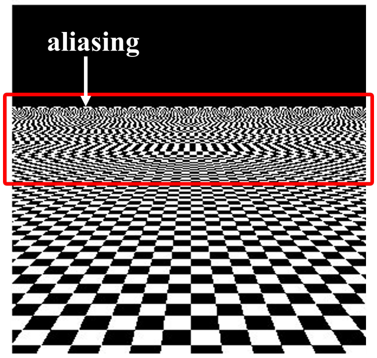
4. Please discuss the **advantages** and **disadvantages** of the phenomenological models (e.g., Phong lighting model) and the geometric optics model (e.g., microfacet model) (4%)
   Phenomenological model is easy to control with intuitive parameters but not accurate; geometric optics is accurate but difficult to use by artists.

5. Please answer the following questions about **texturing**:
   (a) Why do we need textures? (4%)
   Textures can be used to represent spatially varying data and decouple materials from the geometry
   (b) As shown in Fig. 1a, What is the value at coordinate **X** (25.4, 30.6) when **bilinear filtering** is used? (4%)
   The texture data you need are:
   P1 centered at (25, 30) with value **100**; **P2** centered at (26, 30) with value **60**;
   P3 centered at (25, 31) with value **60**; **P4** centered at (26, 31) with value **20**
   The result is $100 * 0.6 * 0.4 + 60 * 0.4 * 0.4 + 60 * 0.6 * 0.6 + 20 * 0.4 * 0.6 = 60$

6. Please answer the following questions about **mipmap**:



(a) bilinear filtering



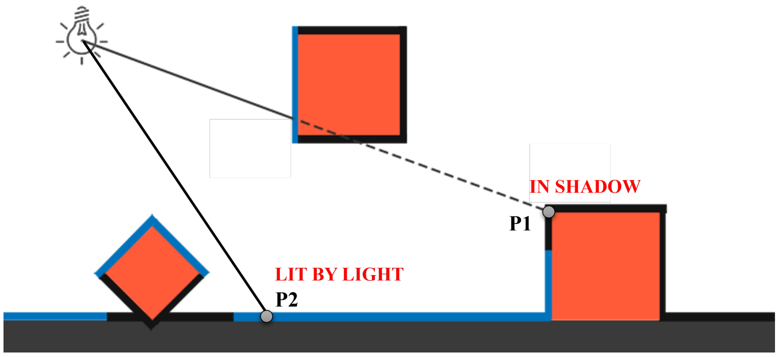
(b) texture aliasing

Figure 1: (a) Figure for question 4. (b) Figure for question 5.

(a) Please describe the root cause of texture aliasing in Fig. 1b (4%)
In that area, one pixel in the screen covers a surface that is larger than one texel in the texture. As a result, using a single texture sampling will undersample the texture signal and produce aliasing.

(b) Mipmap is a popular technique to solve the problem in (a). Please describe how it works. Your answer should include how to build the mipmap for a texture and how to determine the texture color (8%)
The correct way is to average the texel colors inside the footprint covered by a screen pixel. Unfortunately, it is too expensive in the runtime.
To reduce cost, mipmap pre-filters the texture image and builds a hierarchical representation (image pyramid) in the pre-process. Each level in the hierarchy has a half resolution of its previous level. During rendering, the screen-space texture coordinate is first used to estimate the footprint in the texture space, then two successive levels that closest to the footprint are found. Finally, a trilinear filtering process (bilinear in levels and linear between levels) is applied to determine the color

7. Please use the above figure to explain how the **shadow map** algorithm works. Use **P1** and **P2** as examples to explain (10%)



It uses a two-pass algorithm.
In the first pass, the depth map of a scene is rendered from the light view to obtain the closest surface from the light.
In the second pass, the scene is rendered from the camera view. At each shading point, shadows are simulated by projecting the fragment onto the shadow map and

comparing the depth values: if its distance to the camera is larger than the stored distance in the shadow map (such as P1), it is in shadow; otherwise, it is lit by the light (such as P2)

8. Please describe how **OpenGL** renders transparent objects (6%)
Render opaque objects first in any order, then render transparent objects in an order w.r.t their distance to the camera, each time blending with the color buffer

9. Please answer the following questions about **ambient lighting** and **ambient occlusion**:

   (a) Please describe the reason why **Phong lighting model** includes an ambient term (4%)
   Indirect illumination is too expensive to simulate. However, without indirect illumination, we will get a black color if the point is not directly lit by a light. To avoid this, phong lighting model adds a constant value to simulate indirect illumination.

   (b) **Ambient occlusion** is a common technique to improve the ambient term. Please describe how it works (4%)
   Ambient occlusion uses the surface's accessibility to modulate the ambient term. If a point is difficult to reach (e.g., concave regions), it will get darker.

10. Please answer the following questions about **deferred shading**:

    (a) Deferred shading aims to solve an efficiency problem in **GPU graphics pipeline**. Please describe the problem (4%)
    If the scenes have many lights and many occlusions, lots of computation resources are wasted on shading the occluded surfaces that will finally be discarded due to Z-buffer tests

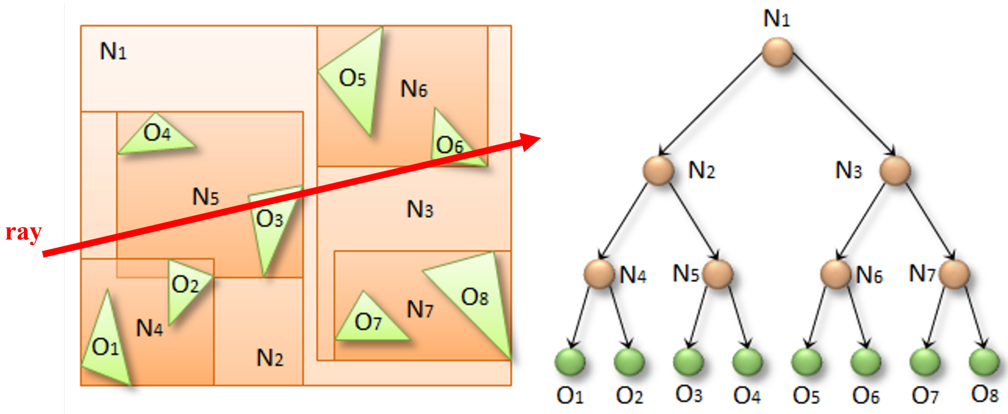
    (b) Following the previous question, please describe how **deferred shading** works (6%)
    Deferred shading uses a two-pass rendering algorithm. In the first pass, it recognizes all visible surfaces from the camera by rendering their geometry and material properties in geometry buffers (G-buffers). In the second pass, lighting is computed only on the visible surfaces based on the G-buffers

    (c) What are the drawbacks of deferred shading? (4%)
    Its drawbacks include larger memory usage, difficult to handle anti-aliasing, and cannot handle transparency

11. Please describe what is **Whitted ray tracing** (8%)
Besides the camera rays, Whitted ray tracing recursively traces three types of rays: shadow ray for determining whether a light can contribute to a shading point (originate from the shading point to the light); reflected ray for computing perfect specular reflection (originate from the shading point along the perfect reflected direction); refracted ray for computing energy from perfect refracted direction (originated from the shading point along the direction determined by Snell's law)

12. Please discuss the **strength** and **weakness** of rasterization compared to ray tracing (6%)
Rasterization has higher parallelism and is better tailored to the GPU architecture; however, it is difficult to simulate inter-object effects such as shadows, indirect lighting, and transparency

13. Using the following figure to describe why a **bounding volume hierarchy (BVH)** can accelerate ray tracing? You should write down the traversal sequence (6%)
We first do intersection test with the bounding box of the scene N1. Since they have intersections, we test with its two children N2 and N3. Again we have intersections



with both bounds, so we keep testing with N4, N5, N6, and N7. This time, we only have intersections with N5 and N6. Thus we only have to test with primitives O3, O4, O5, and O6