Assignment 6 - Material

In this assignment, you are going to practice materials and appearance modeling in computer graphics, including texturing, shading, and the Bidirectional Reflectance Distribution Function (BRDF). For a deeper understanding, it also covers a few related topics that were mentioned but not detailly discussed in the lecture. You should use any resources (e.g., books, search engines, calculators, and etc.) that may help you accomplish it. Code skeletons can be found in github.com/mimuc/cg1-ss20.

Task 1: Texturing

Texture mapping is often known as a process in which an artist manually creates texture in a baking process, and then UV coordinates are saved for each given vertex. Thus, a given texture can be mapped to a given geometry directly.

a) Recall the spherical coordinate system from Assignment 1. Given a point (x, y, z) on the spherical surface, show the corresponding UV coordinates of the sphere.

Given a triangle ABC where the coordinates of each vertex are $A(x_a, y_a, z_a)$, $B(x_b, y_b, z_b)$, $C(x_c, y_c, z_c)$, while the point $P(x_p, y_p, z_p)$ is located inside the triangle. Assume $w_1 + w_2 + w_3 = 1$ such that $w_1 A + w_2 B + w_3 C = P$.

b) Try to use different values for w_1, w_2, w_3, what can you conclude regarding the position of P?

c) Calculate and represent w_1, w_2, w_3 using the coordinates of A, B, C, and P.

In the lecture, you also learned the idea of using a Mipmap for a range query of a texture. Such a query is used to create a Mip hierarchy for a given texture.

d) Assume a texture with a size of $d \times d$, what is the storage overhead of the corresponding Mipmap?

Now, your task is to create a model of the earth using texture mapping that is similar to Figure 1. An interactive demo can be found online[1].

e) Look for `// TODO:` in `main.js` and implement the `setup` method that creates the earth with a starry sky, then animate the rotation of the earth that you have created in the `update` method.

f) What are the main limitations of the bump map and the environment map? Name one limitation for each.

g) Look closer to the Antarctic and Arctic, what issue can you observe regaring texture mapping?

Answer text questions in `README.md` that comes with the provided code skeleton, then include your answers and implementation in a folder called "task01". ***Exclude*** the installed dependencies (folder `node_modules`) in your submission.

[1] https://www.medien.ifi.lmu.de/lehre/ss20/cg1/demo/6-material/earth/

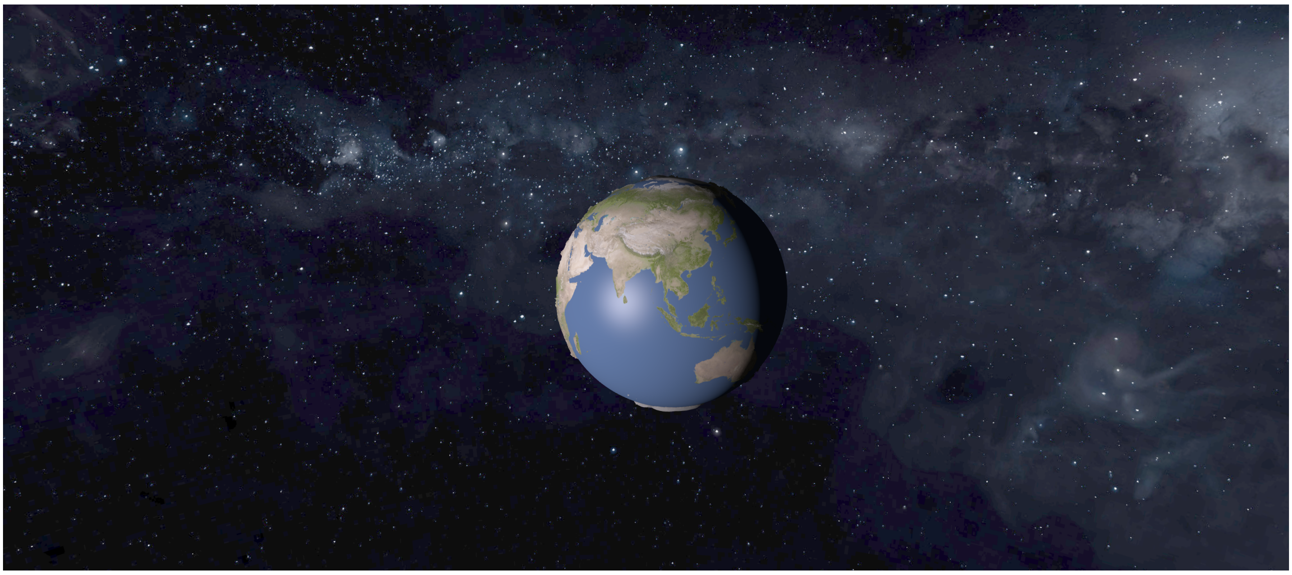


Figure 1: The Earth.

Task 2: Shading and Shadowing

In the lecture, you have learned Phong's illumination model which consists of the three terms: ambient L_a, diffuse L_d, and specular L_s:

$$L_{\text{Phong}} = L_a + L_d + L_s = k_a I_a + k_d I_d \max(0, \mathbf{N} \cdot \mathbf{L}) + k_s I_s \max(0, \mathbf{R} \cdot \mathbf{V})^p$$

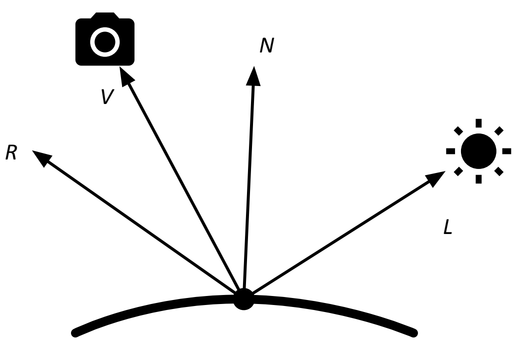
where

- k_a is the ambient coefficient and I_a is the intensity on the surface
- k_d is the diffuse coefficient, I_d is the intensity on the surface, and $\mathbf{N}$ is the surface normal, $\mathbf{L}$ is a unit vector from the surface towards the light source
- k_s is the specular coefficient, I_s is the intensity on the surface, $\mathbf{R}$ is a unit vector of the reflected beam, $\mathbf{V}$ is a unit vector from the surface towards the camera, and p is the shininess.

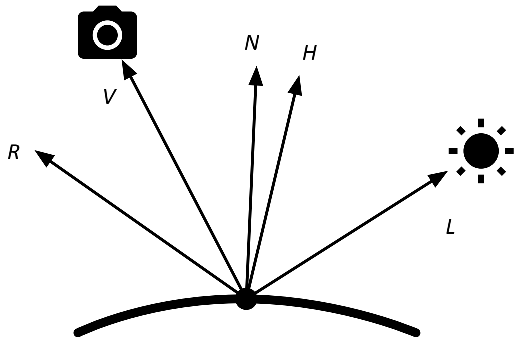
The specular reflection term is computed based on the angle of the reflected beam on the surface and the camera's direction, as shown in Figure 2a.

a) What is the assumption made for the ambient term regarding local illumination?

b) What is the Phong reflection model if there are m light sources based on the assumption in a)? Give the corresponding formula.



(a) The Phong reflection model.



(b) The Blinn-Phong reflection model.

Figure 2: Comparision of the Phong and the Blinn-Phong reflection models.

The Blinn-Phong reflection model is an improved version of Phong's reflection model in terms of computation efficiency. The difference is that the specular term L_s' is computed based on the surface normal and a *half vector* of the camera and light direction, as shown in Figure 2b.

$$L_{\text{Blinn-Phong}} = L_a + L_d + L_s' = k_a I_a + k_d I_d \max\left(0, \mathbf{N} \cdot \mathbf{L}\right) + k_s I_s \max(0, \mathbf{N} \cdot \mathbf{H})^p$$

where the half vector $\mathbf{H} = \frac{\mathbf{V}+\mathbf{L}}{||\mathbf{V}+\mathbf{L}||}$. Since there is no need to compute the reflected beam it is therefore more efficient compared to the Phong reflection model.

In the code skeleton, the scene creates three bunnies, a ground plane and a point light. Your task is to implement a scene similar to Figure 3, and an interactive demo can be found online[2].

Figure 3: The Blinn-Phong reflection model with different shading frequency.

If you look closely at the three bunnies, you will see that they are slightly different from each other. This is because they applied the Blinn-Phong model using different *shading frequencies*: per-face, per-vertex, and per-pixel.

[2] https://www.medien.ifi.lmu.de/lehre/ss20/cg1/demo/6-material/blinn-phong/

Hence, in your implementation: for the bunny on the left, you should implement the Blinn-Phong reflection model using *flat shading*; for the bunny in the middle, you should implement the Blinn-Phong reflection model using *Gouraud shading*; for the bunny on the right, you should implement the Blinn-Phong reflection model using *Phong shading*.

c) Implement the `setupBunny` method in `main.js`, then look for `// TODO:` in the `shaders/` folder and implement the Blinn-Phong reflection model for three different shading frequencies.

d) Zoom closer at each bunny, take a screenshot and highlight the different shading behaviors you observed, then describe the characteristics of each behaviour.

e) Why can you not tell the difference of flat shading and the other two shadings unless you zoom in and what can you conclude from this?

f) Try different values for the shininess (i.e. `this.params.light.Shininess`), what can you observe for the shading behavior?

g) Implement the `setupShadow` method in `main.js`, and create shadows that are casted from the three bunnies.

h) Comparing your implementation with/without shadowing, what can you conclude from your result regarding the position of the bunny?

i) What's wrong about the shadows in comparison to the real world?

Hints for the implementation:

- Recall the `Shadermaterial` from Assignment 5 that provides several built-in attributes and uniforms by the `WebGLProgram`[3].
- Transform (e.g. `projectionMatrix` and `modelViewMatrix`) and normalize (`normalize()`) your vectors and think where they should be computed (in the vertex or the fragment shader).
- Check section the "Built-In Functions" in the OpenGL ES 3.0 API Reference Card [4], page 5-6 and look for the functions you need for vector and matrix operations.
- Check the example provided by three.js document[5] and see how to implement shadow effects.

Answer text questions in the `README.md` that comes with the provided code skeleton, then include your answers and implementation in a folder called "task02". ***Exclude*** the installed dependencies (folder `node_modules`) in your submission.

[3] https://threejs.org/docs/#api/en/renderers/webgl/WebGLProgram
[4] https://www.khronos.org/files/opengles3-quick-reference-card.pdf
[5] https://threejs.org/docs/index.html#api/en/lights/shadows/PointLightShadow

Task 3: BRDF

The rendering equation is an integral equation that described the equilibrium radiance leaving a point is given as the sum of emitted plus reflected radiance:

$$L_o(x, w_o) = L_e(x, w_o) + \int_\Omega L_i(x, w_i) f_r(x, w_i, w_o) \cos\theta_i \, \mathrm{d}w_i$$

where

- $L_o(x, w_o)$ is the outgoing light radiance at the position x on the direction of w_o
- $L_e(x, w_o)$ is the direct emission radiance at the position x on the direction of w_o
- $L_i(x, w_i)$ is the incoming light radiance to the position x on the direction of w_i.
- θ_i is the incident angle of incoming light, and Ω is the hemisphere centered around x containing all possible values for w_i
- $f_r(x, w_i, w_o)$ is the so called *bidirectional reflectance distribution function (BRDF).*

With different forms of the BRDF, we are able to describe different reflectance behavior and therefore create different materials.

a) Assume a light is equally reflected in each output direction. Based on the rendering equation and law of conservation of energy, deduce $f_r(x, w_i, w_o)$.

b) What is the implicit assumption made by the rendering equation that can be a limitation for describing a material?

c) Describe a material that cannot be correctly rendered by the rendering equation.

Include your answers in a Markdown file called “task03.md”.

Submission

- Participation in the exercises and submission of the weekly exercise sheets is voluntary and not a prerequisite for participation in the exam. However, participation in an exercise is a good preparation for the exam (the content is the content of the lecture and the exercise).
- For non-coding tasks, write your answers in a Markdown file. Markdown is a simple mark-up language that can be learned within a minute. A recommended the Markdown GUI parser is typora (`https://typora.io/`), it supports parsing embedded formula in a Markdown file. You can find the syntax reference in its Help menu.
- **Please submit your solution as a ZIP file** via Uni2Work (`https://uni2work.ifi.lmu.de/`) before the deadline. We do not accept group submissions.
- Your solution will be corrected before the discussion. Comment your code properly, organize the code well, and make sure your submission is clear because this helps us to provide the best possible feedback.
- If we discover cheating behavior or any kind of fraud in solving the assignments, you will be withdrawn for the entire course! If that happens, you can only rejoin the course next year.
- If you have any questions, please discuss them with your fellow students first. If the problem cannot be resolved, please contact your tutorial tutor or discuss it in our Slack channel.