CMPE360

Project 4

Ray Tracing I

Section 02

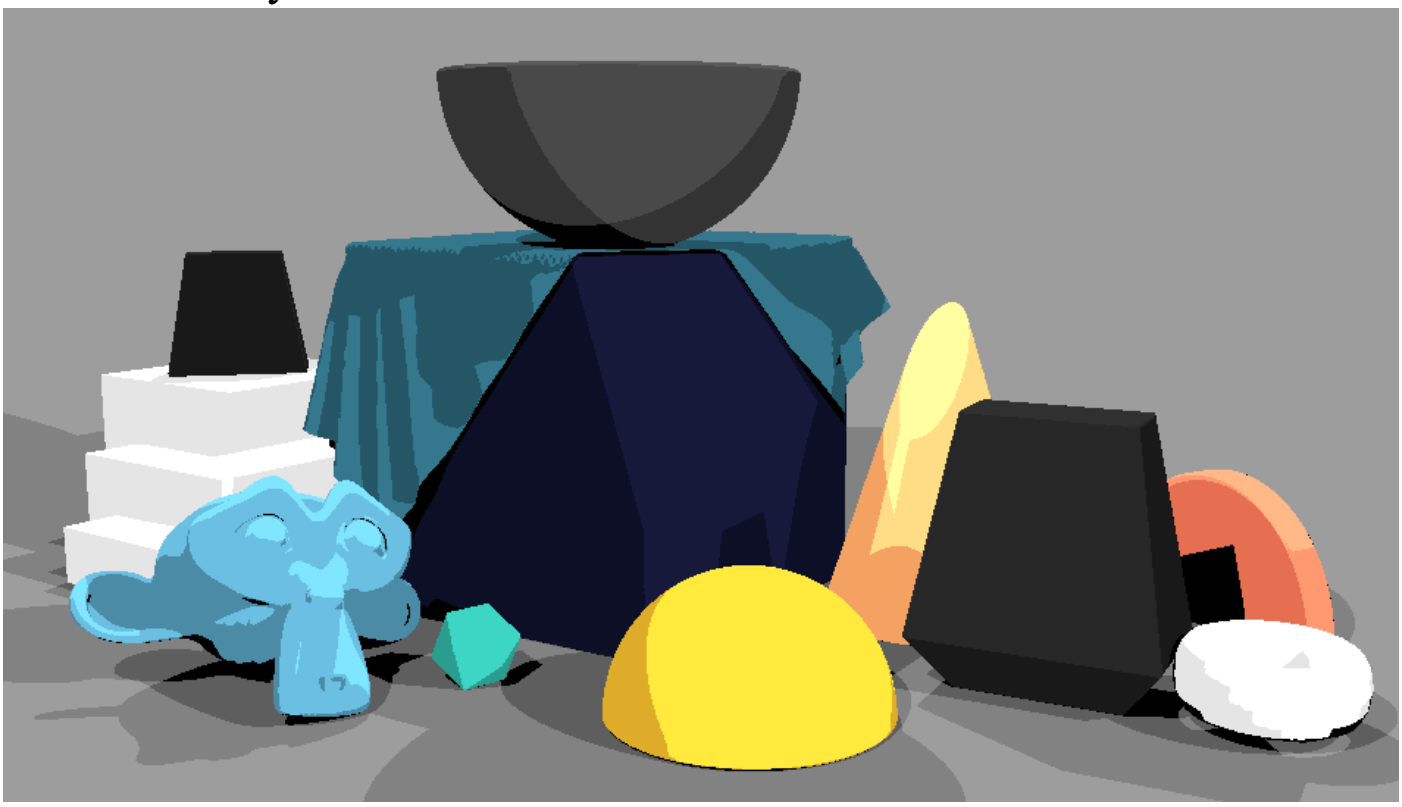
Erkan Sancak

44293566706

Table of Contents

Part I

I. Shadow-Ray



Shadow ray is a crucial element in rendering and lighting calculations, helping simulate realistic shadows and enhancing the overall visual quality of a rendered scene. It is commonly used in shadow mapping and ray tracing.

- `light_vec = light.location - hit_loc` represents the direction from the point of intersection to the light source.
- `light_dir = light_vec.normalized()` represents the normalized direction from the hit location to the light source.
- `new_orig = hit_loc + eps * hit_norm` helps prevent self-occlusion issues.
- `has_light_hit, _, _, _, _, _ = ray_cast(scene, new_orig, light_dir)` checks whether the shadow ray hits any object between the hit location and the light source or not.

First, shadow ray is cast from the hit location towards the light source using `ray_cast` function.After that, it is checked whether shadow ray intersects any objects in the path or not. Finally, that point considered as a shadow location.

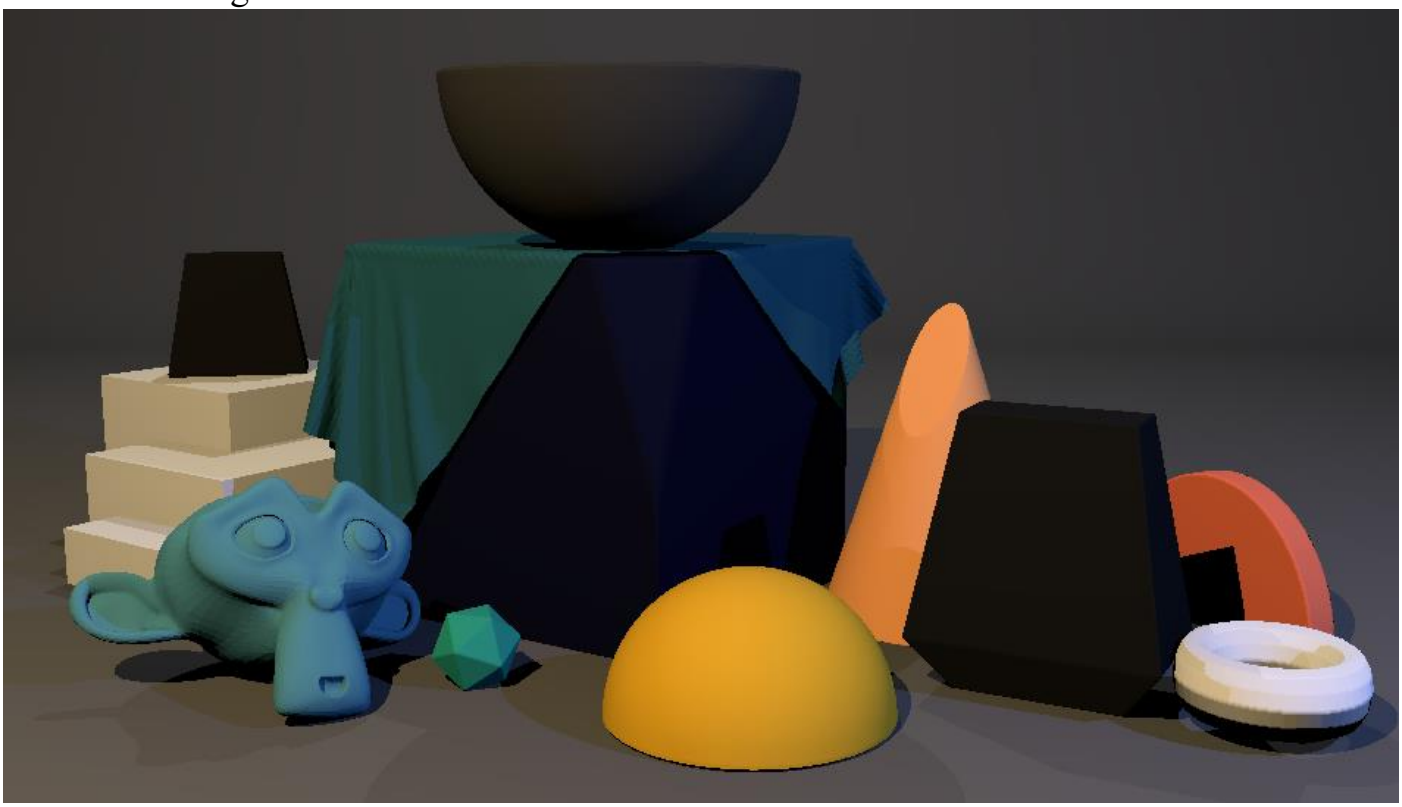
Overall, the shadow ray enhances the rendering by introducing shadows where convinent and more important, creating a more realistic scene.

Part II

II.Bling-Phong Effect

The Blinn-Phong model is a shading technique in that combines diffuse and specular reflection components to simulate the way light interacts with surfaces.

Diffuse-Shading



Diffuse shading is based on Lambertian reflectance, which describes the way light is reflected equally in all directions from a perfectly diffuse surface. Since the light is reflected evenly, the viewpoint does not matter. It is mainly used for render the surface or objects that are not shinny.

- `I_light = light_color / light_vec.length_squared` calculates intensity of the light, and scales it inversely by the distance.
- `diffuse_intensity = np.maximum(hit_norm.dot(light_dir), 0)` calculates diffuse component using the dot product between the surface normal `hit_norm` and the direction from the surface point to the light `light_dir.`
- `color += diffuse_intensity * diffuse_color * I_light` result is added to the pixel color.

Overall, diffuse shading accounts for the even scattering of light across a surface so it is ideal for modelling matte surfaces.

Specular Shading

Specular shading simulates the reflection of light sources by creating bright, glossy highligts on the surfaces. Reflection is highly directional and depends on the position of the light source and the viewer (viewpoint). It is mainly used for render the surface or objects are glossy or shiny.

- `diffuse_intensity = np.maximum(hit_norm.dot(light_dir), 0)` calculates the diffuse reflection intensity based on the dot product of surface normal and light direction, ensuring no negative contributions.

- `color += diffuse_intensity * diffuse_color * I_light` adds the diffuse reflection contribution to the overall pixel color.

- `half_vector = (light_dir + (-ray_dir)).normalized()` calculates the normalized half vector between light direction and view direction.

- `specular_intensity = np.maximum(hit_norm.dot(half_vector), 0)`
  `specular_intensity = specular_intensity ** specular_hardness` calculates the specular reflection intensity, ensuring no negative contributions and applying a specular hardness factor.

- `color += specular_intensity * specular_color * I_light` adds the specular reflection contribution to the overall pixel color.

Overall, specular shading accounts for the bright reflections that occur on glossy or shiny surfaces and materials when illuminated. The specular component of the reflection is calculated based on the viewer's position, the light source direction, and the surface normal. Unlike in diffuse shading, specular shading is highly depends on viewpoint.

Part III

III.Ambient Light

Ambient light is a type of indirect illumination comes from all directions in a scene. It adds atmosphere and depth to a scene. Furthermore, ambient lighting is used for global illumination, simulate the natural light in a scene, and to create a more realistic, immersive experience.

- `if no_light_hit:`
  `ambient_color = scene.simpleRT.ambient_color` checks if there's no direct light hitting the surface, indicating that the point is in shadow and retrieves the ambient color from the scene settings.

- `color += Vector(ambient_color) * diffuse_color` adds the ambient component to the overall pixel color.

Overall, ambient light helps to reveal details in shadowed regions, creating a more balanced, visually pleasing and more realistic scenes.