CMPE360

Project 4

Ray Tracing II

Section 02

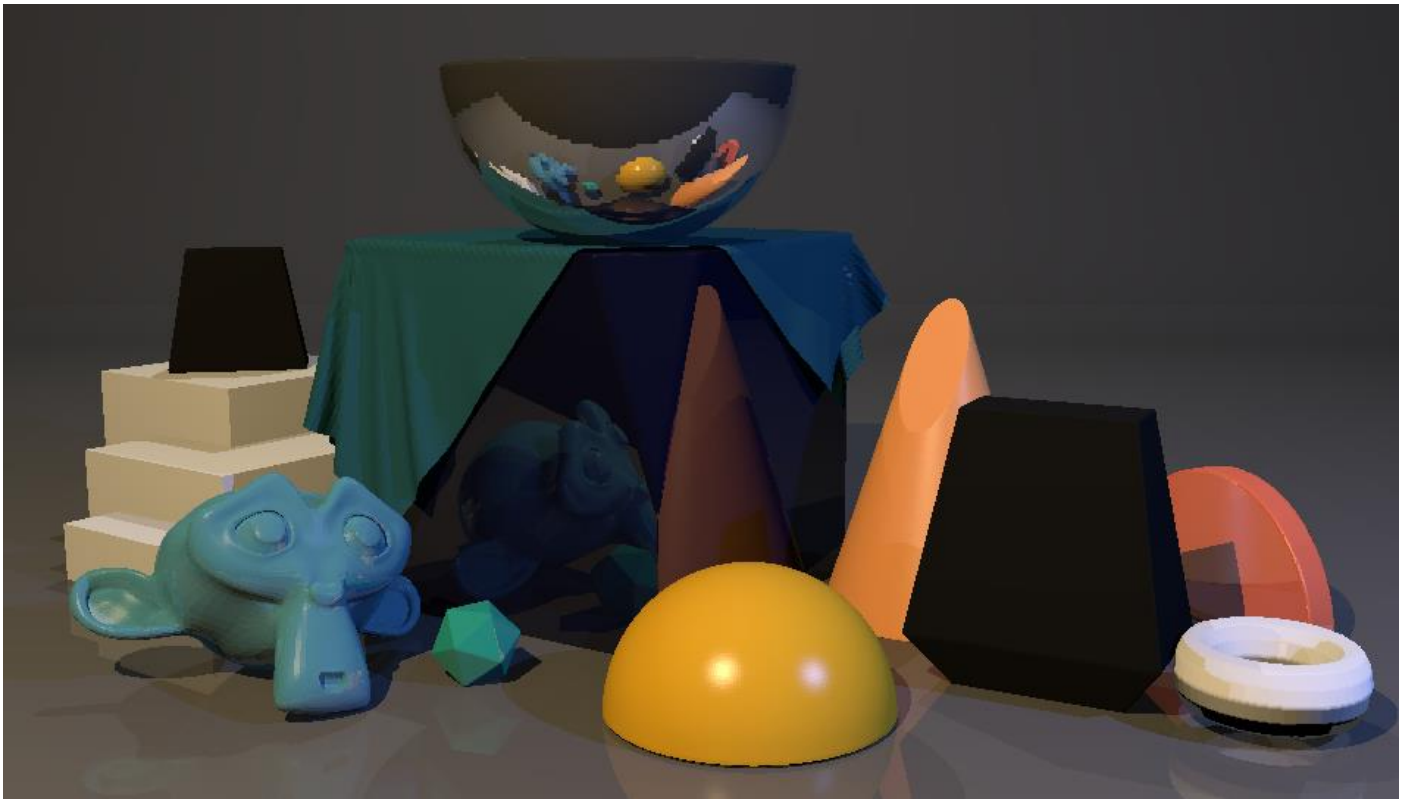
Erkan Sancak - Elif Aysu Kürşad

442935667 - 14162766452

Table of Contents

Part I

A) Reflection



B) Explanation

Reflection is obtained by transforming the object in the X or Y axis. With the addition of refleciton objects now reflect their surroundings. The overall visual appeal of the scene is significantly improved with the addition of reflection.

- `D_reflect` represents the direction vector of the reflected ray in the ray tracing process. Then it is used to trace the reflected ray recursively to simulate multiple reflections in the scene. Also reflection direction is normalized to ensure it has unit length.

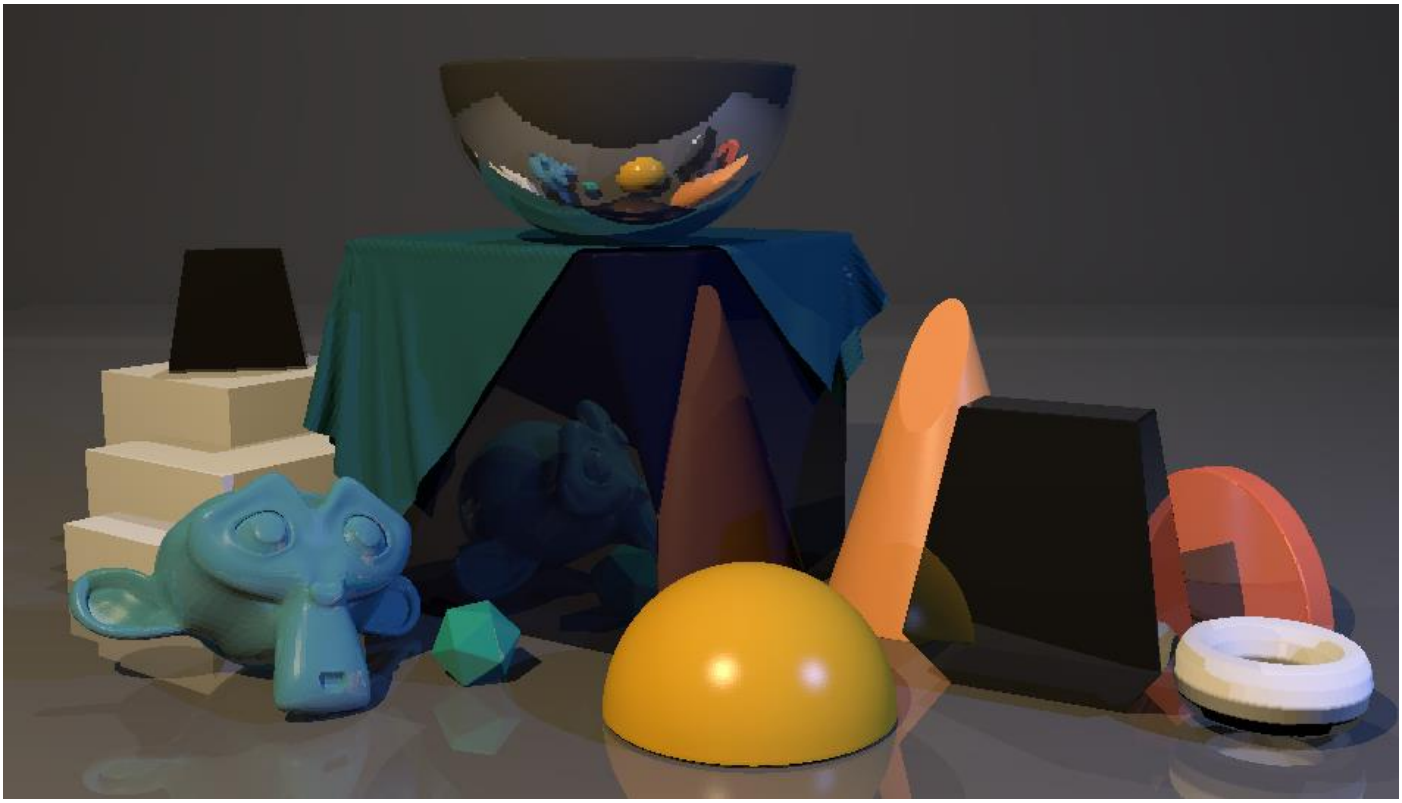
- `RT_trace_ray` helps to recursively trace the reflected ray until it becomes zero.

- `color += reflectivity * reflect_color` combines reflection with pixel color.

Overall, this implementation captured the way light interacts with surfaces and contributes to achieving realistic reflections in the rendered scene, enhancing the visual quality and realism.

Part II

A) Fresnel



B) Explanation

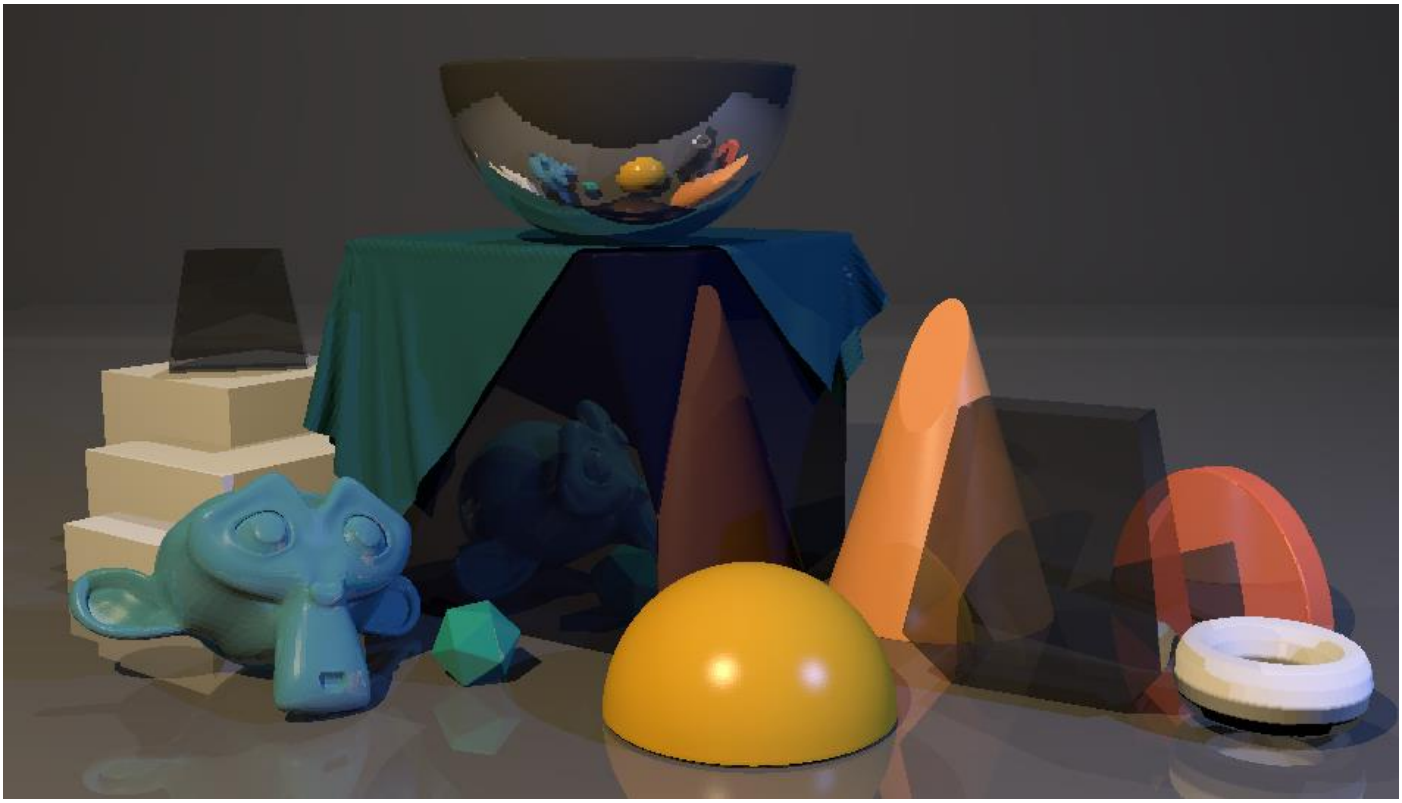
The Fresnel effect describes how the reflectivity of a surface changes with the viewing angle. At normal incidence less ligt is reflected however, while at grazing angles, more light is reflected.

- `R_0 = ((1 - mat.ior) / (1 + mat.ior))**2 means` $R_0 = (\frac{n_1 - n_2}{n_1 + n_2})^2$ it is formula for Fresnel reflection at normal incidence. R_0 is the reflection coefficient for transitioning.
- `cos_theta = -ray_dir.dot(hit_norm)` calculates Θ between ray direction `(ray_dir)` and surface normal `(hit_norm)`
- `fresnel_factor = R_0 + (1 - R_0) * (1 - cos_theta)**5` Schlick's approximation for Fresnel reflection. Combines R_0 with $\cos(\theta)$.

Overall, this implementation determine the reflectivity of a surface based on Fresnel effect with Schlick's approximation. Calculates the reflectivity based on the angle of incidence and the material properties with contributing to the overall visual fidelity of the scene.

Part III

A) Transmission



B) Explanation

Transmission effect simulates the behavior of light passing through transparent materials. This effect achived by calculation of color contribution of transmitted ray. So, the black block turned into transparent material, showing what is through it.

- `n1` and `n2 are the` refractive indices and checks if the ray is inside of object or not.
- `cos_theta_i` calculates incident angle and sin_theta_t calculates transmission angle
- `if sin_theta_t > 1.0`
    - internal reflection occurs and calculates the `D_transmit` (transmitted direction) as reflected direction.
    - `else` calculates the transmitted direction `(D_transmit)` based on Snell's Law.
- The color contribution of the transmitted ray `(transmit_color)` is multiplied by `(1 - reflectivity) * mat.transmission` and added to the pixel color

Overall, this implementation simulates the transmission of light through transparent materials, and the effect is observed in the final rendered image.