OpenGL Ray-Tracing

> This is a simple ray-tracing program that uses OpenGL to render the scene. The program is written in C++ and uses the
> OpenGL library for rendering. The program uses the ray-tracing algorithm to render the scene. The program uses the
> Phong reflection model to calculate the color of the objects in the scene.

Table of Contents

Installation

This program requires the OpenGL library (`freeglut`) to run. Required libraries are included in the
repository releases
(please unzip mingw.zip to the current directory "./mingw64").

To install the program, you need to clone the repository and compile the program using the following commands:

```
git clone git@github.com:oopb/opengl_ray_tracing.git
cd opengl_ray_tracing
```

To compile the program, you need to run the following command:

```
.\mingw64\bin\g++.exe -o ray_tracing ray_tracing.cpp -lfreeglut -lopengl32 -lglu32
```

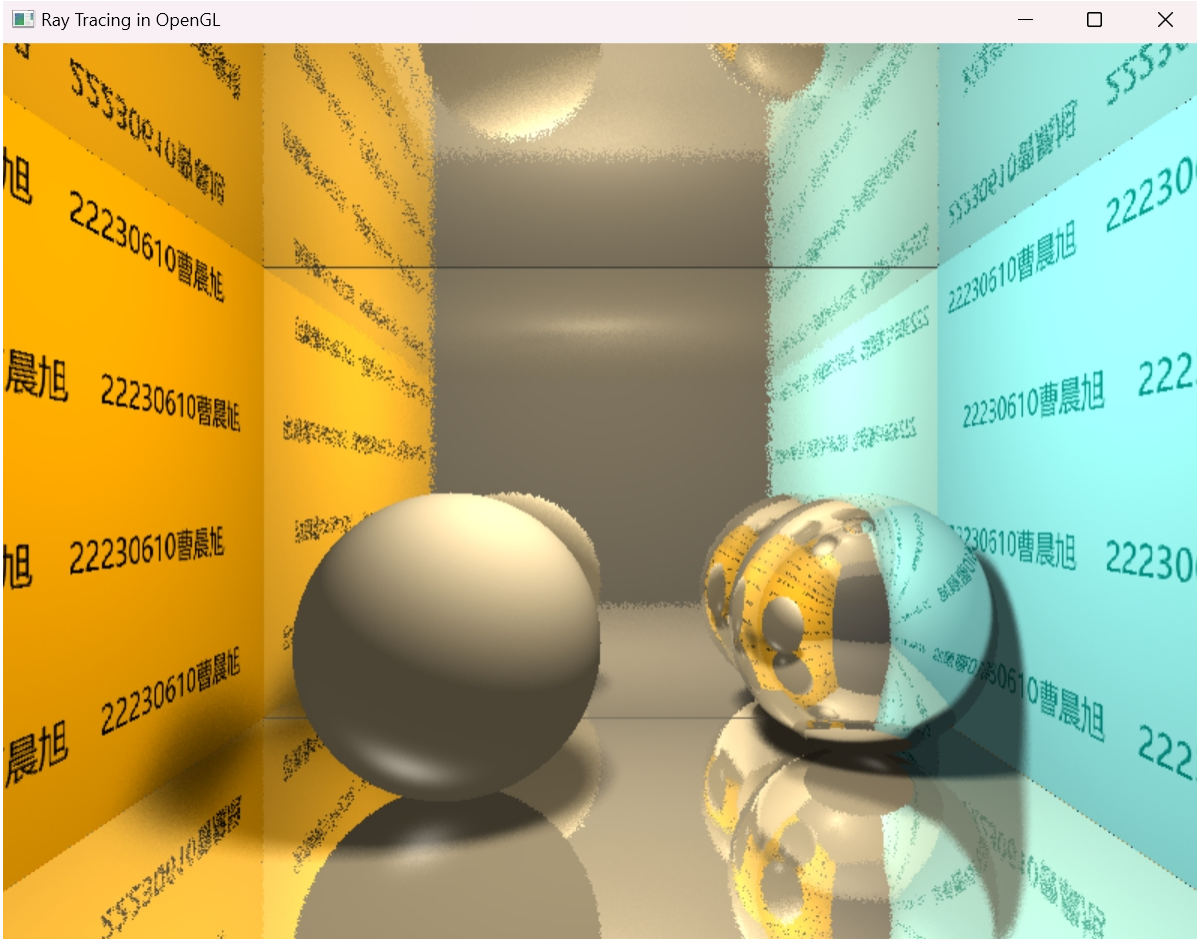
Usage

To run the program, you need to run the following command:

```
./ray_tracing
```

Or you can also double-click on the `ray_tracing.exe` file to run the program.

After running the program (and wait for a few seconds), you should see a window with the rendered scene.



Details

Ray-Tracing Algorithm

The ray-tracing algorithm is a rendering algorithm that uses the ray-tracing technique to render the scene. The
ray-tracing algorithm works by tracing the path of the light rays in the scene. The algorithm works by casting a ray
from the camera through each pixel in the image plane. The ray is then tested for intersection with the objects in the
scene. If the ray intersects with an object, the color of the object is calculated using the Phong reflection model.

Phong Reflection Model

The Phong reflection model is a lighting model that is used to calculate the color of the objects in the scene. The
Phong reflection model works by calculating the color of the object based on the ambient, diffuse, and specular
components of the lighting. The ambient component is the color of the object when there is no light. The diffuse
component is the color of the object when the light is shining directly on the object. The specular component is
the color of the object when the light is shining directly on the object and is reflected off the object.

The Phong reflection model is used to calculate the color of the objects in the scene. The color of the object is
calculated using the following formula:

```
color = ambient + diffuse + specular
```

The ambient component is calculated using the following formula:

```
ambient = ka * Ia
```

The diffuse component is calculated using the following formula:

```
diffuse = kd * Id * max(0, dot(N, L))
```

The specular component is calculated using the following formula:

```
specular = ks * Is * pow(max(0, dot(R, V)), shininess)
```

Where:

- `ka` is the ambient reflection coefficient
- `Ia` is the ambient light intensity
- `kd` is the diffuse reflection coefficient
- `Id` is the diffuse light intensity
- `N` is the normal vector of the object
- `L` is the light vector
- `ks` is the specular reflection coefficient
- `Is` is the specular light intensity
- `R` is the reflection vector
- `V` is the view vector
- `shininess` is the shininess coefficient
- `dot` is the dot product
- `max` is the maximum function
- `pow` is the power function

Natural Soft Shadows

Natural soft shadows are created by simulating the light rays that are blocked by the objects in the scene. The soft
shadows are created by casting multiple rays from the light source to the object. The rays are then tested for
intersection with the objects in the scene. If the ray intersects with an object, the color of the object is calculated
using the Phong reflection model. The color of the object is then multiplied by the transparency of the object to create
the soft shadows.

Here's how to achieve soft shadows:

- **Light source sampling**: simulating an area light source by generating multiple random sample points around the
  light source. Each sample point represents a subarea of the light source.
- **Shadow Ray Casting**: Cast Shadow Rays from the hit point to each light source sample point to check if these rays
  are blocked by other objects in the scene.
- **Calculate the occlusion ratio**: count the number of the sample points of the light source which is occluded, and
  calculate the occlusion ratio (shadow factor) . The higher the occlusion ratio, the deeper the shadow.
- **Applying the shadow factor**: The Shadow Factor is applied to the lighting calculation to achieve a soft shadow
  effect.

Reflection

Reflection is the process of simulating the light rays that are reflected off the objects in the scene. The reflection
is calculated using the reflection vector and the view vector. The reflection vector is the vector that is reflected off
the object. The reflection vector is calculated using the following formula:

```
R = 2 * dot(N, L) * N - L
```

Where:

- `R` is the reflection vector
- `N` is the normal vector of the object
- `L` is the light vector
- `dot` is the dot product
- `2` is the scalar
- `N` is the normal vector of the object

The reflection vector is then used to calculate the color of the object using the Phong reflection model. The color of
the object is then multiplied by the reflection coefficient to create the reflection effect.

Acknowledgements

- OpenGL
- FreeGLUT
- GLU
- Phong Reflection Model
- Ray-Tracing
- Ray-Tracing in One Weekend

- stb_image

Source Code

- GitHub Repository