

情報メディア実験 III・レイトレーシングによるコンピュータグラフィクス入門
第1回レジュメ

2011/12/07

担当: 金森由博 (kanamori@cs.tsukuba.ac.jp)

※今年度の教材の準備に時間がかかっているため、昨年度のもので実施します。
迷惑をかけてすみません…。なお、途中までは今年度実施予定分と同じ内容です。

1. 雑型ソースコードの入手・コンパイル

金森の実験のページ

<http://kanamori.cs.tsukuba.ac.jp/jikken/>

からダウンロードする。

2. プログラムの動作の説明

レイトレーシングには時間がかかるので、このプログラムではプレビュー用に物体を描画する機能を加えてある。この機能を実装するために、OpenGL というグラフィクスライブラリ、そのユーティリティライブラリである GLUT、GUI を提供する GLUI というライブラリを利用している。基本的には OpenGL/GLUT/GLUI のことを勉強しなくてもよいように作ってあるが、機能を追加・修正したい場合は `OGLDisplayDriver.cpp/.h` を編集してほしい。OpenGL/GLUT については下記[1]を、GLUI については下記[2]や、インターネットの他のサイトを参考にしてほしい。

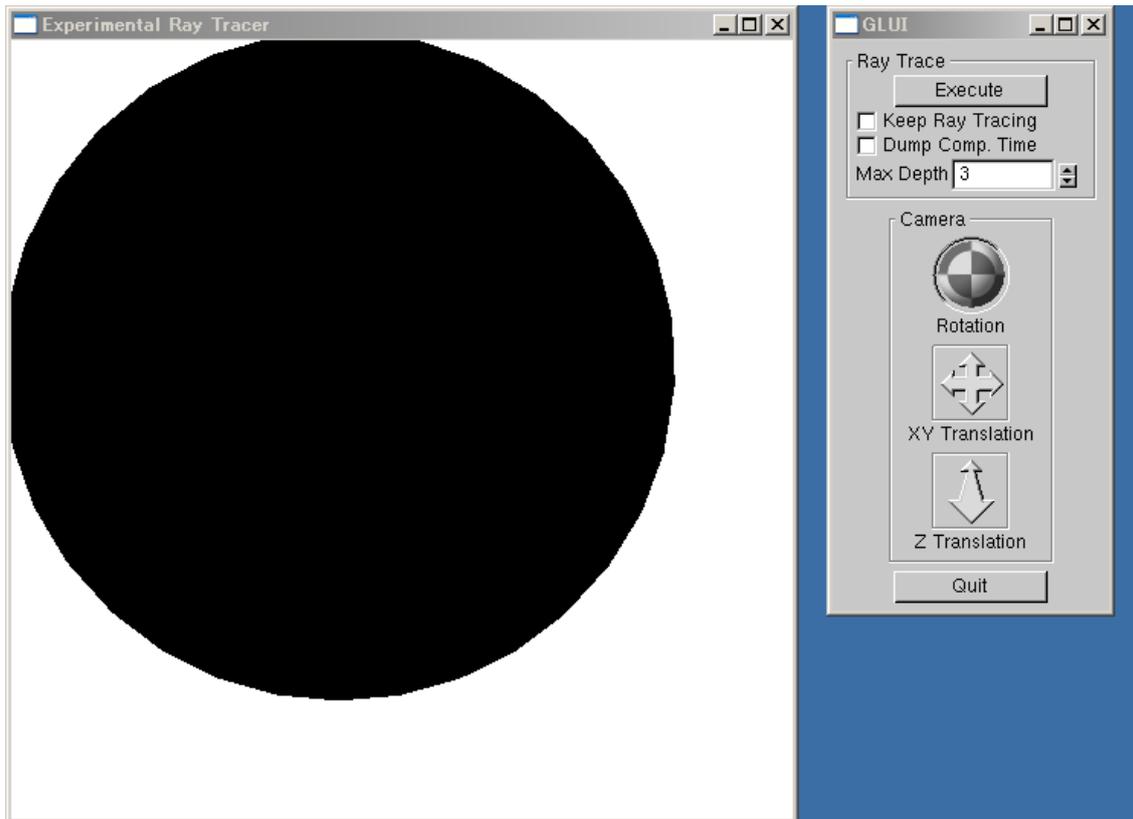
[1] GLUT による「手抜き」OpenGL 入門

<http://www.wakayama-u.ac.jp/~tokoi/opengl/libglut.html>

[2] GLUI 日本語訳

<http://ktm11.eng.shizuoka.ac.jp/glui/glui.html>

プログラムの実行時のスクリーンショットを次に示す。



左ウィンドウ：

OpenGL によるプレビューと、レイトレーシングの結果を表示する。キーボードの "q" やエスケープキーを押すとウィンドウが閉じる (OGLDisplayDriver.cpp の Keyboard 関数で動作を修正できる)。

右ウィンドウ：

画面操作やパラメータを調整するための GUI。

- [Ray Trace] タブ

- [Execute] ボタン

- レイトレーシングを実行する。

- [Keep Ray Tracing] チェックボックス

- チェックが入っている場合、[Camera]のところで視点を移動するたびに、レイトレーシングを実行して表示する。描画に時間がかかるシーンの場合は、これを使わず、[Execute]ボタンで描画する方がよい。

- [Dump Comp. Time] チェックボックス

- チェックが入っている場合、レイトレーシングが実行されるたび、かかった時間をコンソールに表示する。

- [Camera] タブ

- [Rotation]

マウスでドラッグすると、カメラ（視点）を回転させる。

–[XY Translation]

マウスでドラッグすると、カメラを XY 方向に平行移動させる。

–[Z Translation]

マウスでドラッグすると、カメラを Z 方向に平行移動させる。

3. 雑型ソースコードの説明

ソースコードは C++ 言語で書かれている。C++ の知識がなくても、それほど支障がないよう配慮したつもりだが、必要に応じてネットなどの資料で勉強してほしい。

ソースコード中には、課題のためにわざと機能を削っている部分（TODO という文字を探してほしい）がある。各自、その部分を自分で考えて埋めてほしい。

プログラムは以下のファイルからなる。

[修正が必要なファイル]

- main.cpp

レイトレーシングのコア部分がかかれている。

- Triangle.cpp / .h

視線（レイ）と三角形の交差判定などを記述。

- DiffuseMaterial.h

拡散反射する材質の反射光の計算を記述。

- PhongMaterial.h

光沢反射（Phong モデル）する材質の反射光の計算を記述。

- EnvironmentMap.cpp

環境マップの入力や、色の取得関数を記述。

[場合によっては修正の必要のあるファイル]

- SceneLoader.cpp / .h

設定ファイルを読み込んで、シーンを設定。

- Scene.cpp / .h

レイトレーシングするシーンの物体や光源を保持。

- Ray.h

視線（レイ）を定義。

- Camera.cpp/.h

カメラのパラメータや機能を実装。

- ImageRect.h

矩形画像のサイズやデータを保持。

- DebugIntersectionLog.h

デバッグ用の情報を記録するためのクラスを記述。

- **GeometricObject.h**

視線と交差判定する対象となる幾何的対象物を定義。

- **Sphere.cpp / .h**

視線（レイ）と球の交差判定などを記述。

- **Plane.cpp / .h**

視線（レイ）と平面の交差判定などを記述。

- **LightSource.h**

光源を定義。

- **PointLight.h**

点光源を定義。

- **DirectionalLight.h**

平行光源を定義。

- **Material.h**

材質の光を反射する性質を定義。

- **PerfectSpecularMaterial.h**

鏡面反射する材質を定義。

- **SpecularRefractionMaterial.h**

屈折を伴う材質を定義。

- **my_algebra.h**

ベクトル演算・行列演算を定義。

- **OGLDisplayDriver.cpp / .h**

OpenGL / GLUT / GLUI を用いた描画処理を記述。

その他のファイルは、特に修正する必要はないと思われる。

4. 最初の課題

手始めに、視線と三角形の交差判定のコードを完成させてほしい。視線と球、平面の交差判定が参考になると思われる。