

情報可視化論

最終課題

学籍番号 1385066T

氏名 林賢悟

実装した機能

今回私が最終課題で実装した可視化アプリケーションは、ロブスターデータに対するボリュームレンダリングである。さらにそのロブスターデータのボリュームレンダリングに加え、マウスコントロール機能、ユーザーインターフェイスとして Phong シェーディングの反射モデルの変換、不透明度に対応する値の変更、伝達関数のカラーマップの変更を追加した。

機能説明

まずアプリケーションを開くと、Phoong シェーディングの Lambertian 反射モデル、不透明度に対応する値(以下 alpha)が 255、伝達関数の colormap が虹色のロブスターデータのボリュームレンダリングが表示される。

シェーディングモデルは Phong シェーディングの Lambertian、Phong、Blinn-Phong、Cook-Torrance 反射モデルが選択でき、ユーザーインターフェイス(以下 UI)の model の部分で 'lambertian'、'phong'、'blinnphong'、'cooktorrance' を選択することで、それぞれの反射モデルに変換することができる。

alpha の値は 0~5000 まで変更することができ、値が大きくなるほど透明度が大きくなる。値の変更は UI の alpha 部分のゲージをクリックする、またはゲージの横に表示されてある alpha の数字を直接入力することで変更できる。

伝達関数の colormap は虹色と赤色の 2 種類用意しており、UI の colormap の部分で rainbow を選択すると w10 の task1 で行ったような虹色 colormap が、red を選択すると w10 の task2 で行ったような赤白 colormap がそれぞれ伝達関数として使用されレンダリングされる。

工夫した点

今回最終課題で最も工夫した部分は UI の部分である。UI は dat.UI を利用して作成した。

まずシェーディングについて、どの反射モデルを使用するかを判断するために

html 内に uniform 変数として int 型の `shader_model` を用意した。反射モデルは 4 つあるので、この値が 0 か 1 か 2 かその他によって反射モデルを判断する。そして UI が操作されると、その選択した反射モデルによって `shader_model` の数値を変えて、ロブスターデータの `material`(ソース内では `raycaster_material`) を変更して再描画する。`alpha` や `colormap` も同様にして、UI が操作された時に `alpha`、`colormap` に対応する部分に変更されロブスターデータの `material` を変更して再描画する。ボリュームレンダリングに関しては坂本先生の実装例を参考にして作成した。