

情報可視化論 最終課題

2018 年 6 月 15 日提出

181x104x Kawakami Yuta

<概要>

今課題では、講義で扱った技術をいくつかと、講義では扱ってなかったが Slides に載っていた技術を用いて、volume rendering application を作成した。

<実装内容>

今課題では、volume rendering の制御機能を追加するために、dat.gui.js を用いた。これにより、下の図 1 のような制御ウィンドウが画面右上に設置されている。

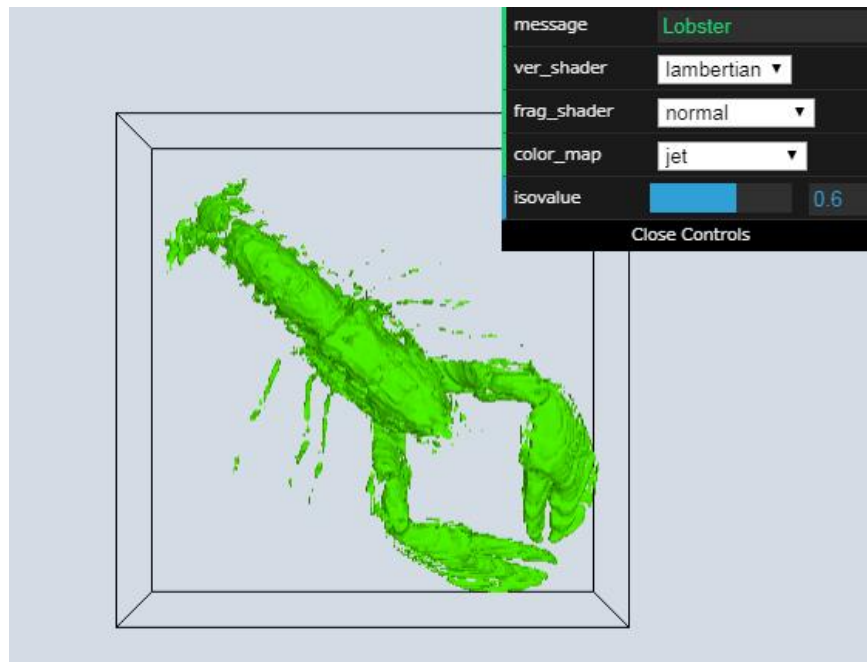


図 1 LobsterData vertex_shader (lambertian)

制御ウィンドウの説明を簡単にしていく。

➤ message

Lobster (図 1) か Hydrogen (図 2) を手入力すると、可視化されている volume data が切り替わる。初期設定は LobsterData となっている。今回は LobsterData のみでなく他の Data も追加しようと考え、KVS.js 内から HydrogenData を選択した。個別の Data ごとに制御ウィンドウを変更できるとなお良かったが、断念した。

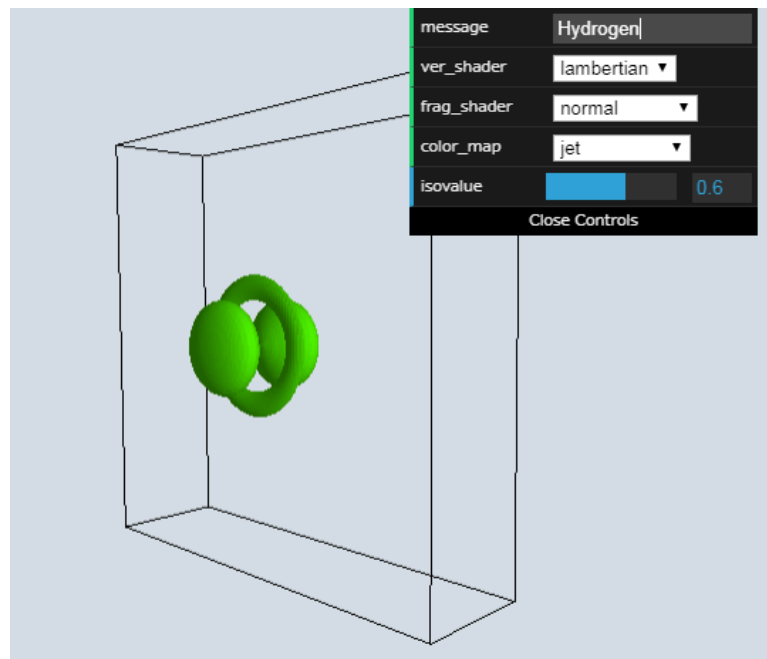


図 2 HydrogenData vertex_shader(lambertian)

➤ ver_shader

これにより, vertex_shader を制御している. normal と lambertian が用意されており, ver_shader と frag_shader を normal に設定すると, 影がまったくない色抜きのデータが可視化される (図 3) .

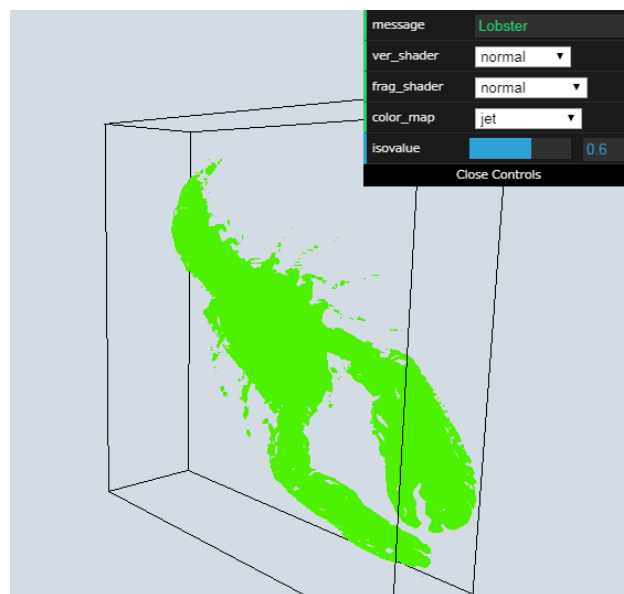


図 3 LobsterData vertex_shader(normal) fragment_shader(normal)

➤ frag_shader

これにより, fragment_shader を制御している. ver_shader より選択肢を増やした. normal, lambertian, phong, blinnphong, cooktrrance を実装してみた.

➤ color_map

これにより、可視化用のカラーマップを制御している。選択肢としては、講義で扱っていた jet, white_red に加え, white_blue, white_green を追加した。以下のソースコードのように, if 文で条件分岐させて cmap を作成した。

```
if(color_map_type == 'jet'){
    var R = Math.max( Math.cos( ( S - 1.0 ) * Math.PI ), 0.0 );
    var G = Math.max( Math.cos( ( S - 0.5 ) * Math.PI ), 0.0 );
    var B = Math.max( Math.cos( S * Math.PI ), 0.0 );
}else if(color_map_type == 'white_blue'){
    var R = Math.max(1 - S, 0.0 );
    var G = Math.max(1 - S, 0.0 );
    var B = Math.cos( 0 );
}else if(color_map_type == 'white_green'){
    var R = Math.max(1 - S, 0.0 );
    var G = Math.cos( 0 );
    var B = Math.max(1 - S, 0.0 );
}else if(color_map_type == 'white_red'){
    var R = Math.cos( 0 );
    var G = Math.max(1 - S, 0.0 );
    var B = Math.max(1 - S, 0.0 );
}
```

➤ isovalue

これにより, isovalue の値を制御している。表示範囲は 0~1 となっている。

これらの機能を用いた、いくつかの例を以下に載せる。

