

ボリューム可視化アプリケーションを開発した。今回のプログラムは、Google Chrome で動作確認を行った。図 1 に、開発したアプリケーションの画面を示す。



図 1 アプリケーションの画面

(初期値設定 Isovalue : 15, Color {R : 180, G : 110, B : 80})

図 1 では、左側の 80%width はボリューム可視化の結果を表示する画面であり、右側の 20%width はユーザ操作画面である。画面の左上に“Visualization Result”を表示し、一番下の左側と右側それぞれに” Rendering screen” , ” UI” と示している。

ユーザ操作画面では、普段の授業内容に基づき、主に三つの機能を注目した。

(図 1 の右上の紫色の区域では、タイトルと学籍番号及び氏名を書いた)

Isovalue : 等値面の変更の初期値を 15 に設定し、0 から 255 までの変化さを作った。直接にそのボタンを移動すれば、その値が大きければ大きいほど、左側に表示した等値面の消えた部分が大きくなってくる。

Color : カラーマップを用いて、R, G, B の三つの変化区間を作った。R, G, B それぞれ初期値を 180, 110, 80 に設定し、0 から 255 まで変化区間を作った。その使い方が Isovalue 機能と同じであるが、そこで数字が大きければ大きいほど、それぞれ Red, Green, Blue の本色に近づいている。

Shading と Reflection : 両方の機能を合わせて、その変化の大きさを検討したいため、両方の選択肢の中からどちらかを一つ選んで、最後に “Apply” というボタンを押せば、実行するという。ここでは、第 10 回の授業内容を参考し、Shading では “Gouraud”, “Phone” という選択肢を設定し、Reflection では “Lambertian”, “Phone”, “Blinn\_Phong”, “Cook Torrance”, “Toon” という選択肢を設定した。“Apply” ボタンでの実行では、Shading と Reflection の両方の中で一つを選ばなければならない。それによって、組み合わせた変化の結果が左側に表示できる。

そして、図 1 の右下の赤色のところに、本アプリケーションの内容を簡潔に説明し、日付と名前を書いた。上述三つの機能を組み合わせて、様々な効果が達成できている。今後では、ボリュームの断面表示も追加して、さらに情報可視化の知識を深めたいと思う。(図 2 は Shading と Reflection を追加して表示した画面である)

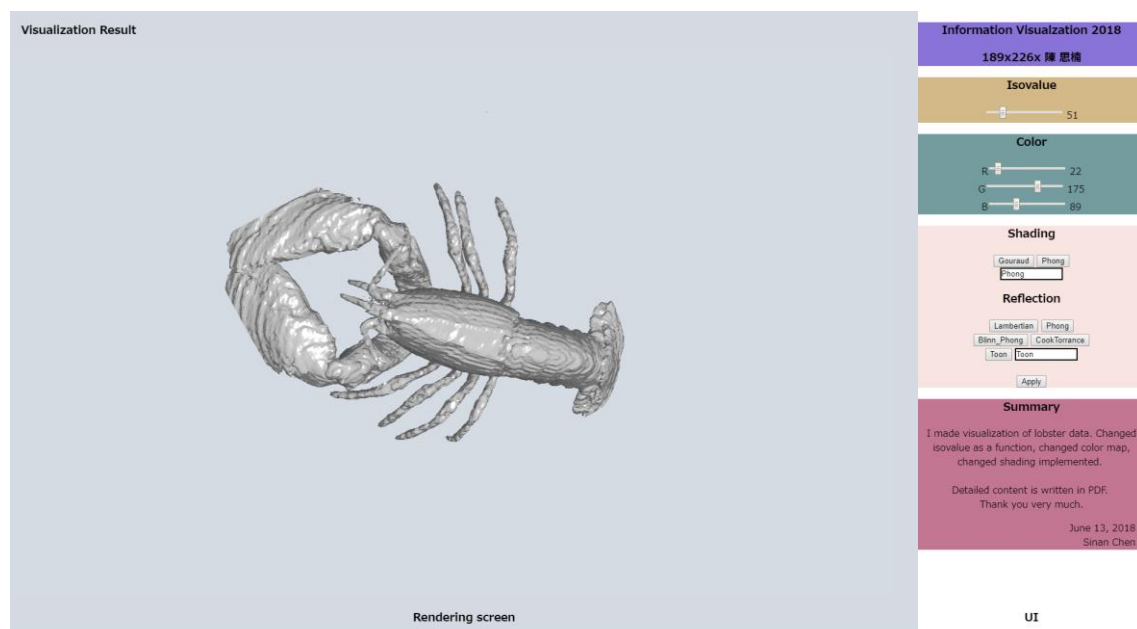


図 2 Shading と Reflection を追加して表示した画面  
(値設定 Isovalue : 51, Color {R : 22, G : 175, B : 89}, Phone, Toon)