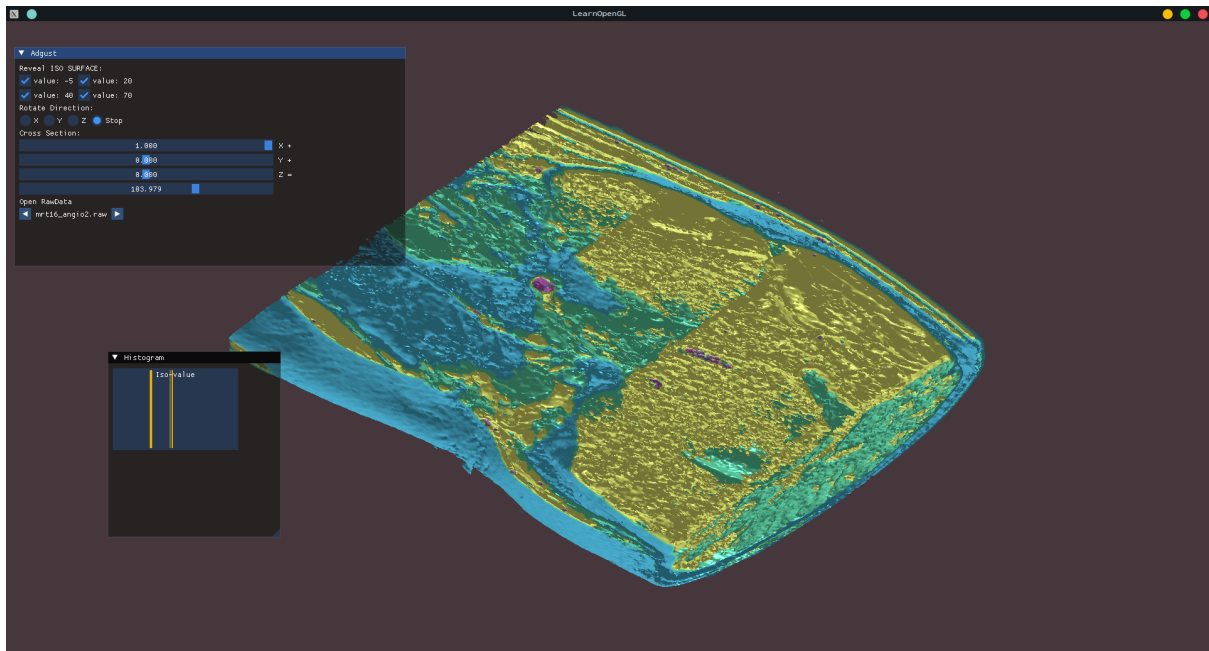


科學試算_HW1:ISO-Surfaces

00957117 黃菁蕙



完成項目：

- 使用 marching-cube 創造 iso-surfaces

```
array<short, 16> kTri = cube::kCubeTriangles[triindex];
```

- 利用梯度找到點的法向量

```
for (int k = 0; k < 256; ++k){
    if(i == 0) grid.x = raw_data[i + 1][j][k] - raw_data[i][j][k];
    else if(i == 256 - 1) grid.x = raw_data[i][j][k] - raw_data[i - 1][j][k];
    else grid.x = (raw_data[i + 1][j][k] - raw_data[i - 1][j][k]) / 2.0;
    if(j == 0) grid.y = raw_data[i][j + 1][k] - raw_data[i][j][k];
    else if(j == 256 - 1) grid.y = raw_data[i][j][k] - raw_data[i][j - 1][k];
    else grid.y = (raw_data[i][j + 1][k] - raw_data[i][j - 1][k]) / 2.0;
    if(k == 0) grid.z = raw_data[i][j][k + 1] - raw_data[i][j][k];
    else if(k == 256 - 1) grid.z = raw_data[i][j][k] - raw_data[i][j][k - 1];
    else grid.z = (raw_data[i][j][k + 1] - raw_data[i][j][k - 1]) / 2.0;
    gri_data[i][j][k] = glm::normalize(grid);
}
```

- 撰寫 VAO & VBO

```
// specular
vec3 viewDir = normalize(viewPos - GLOBAL);
vec3 reflectDir = reflect(-lightDir, norm);
float shininess = 32.0f;
float spec = pow(max(dot(viewDir, reflectDir), 0.0), shininess);
vec3 specular = light.specular * spec;

vec3 result = (ambient + diffuse + specular) * objColor;
Draw = vec4(result, 1.0);
```

```
GLOBAL = (model * vec4(vertexCor, 1)).xyz;
NORM = mat3(transpose(inverse(model))) * normalCor;
//gl_Position = vec4(vertexCor, 1.0);
gl_Position = projection * view * model * vec4(vertexCor, 1.0);
```

- 切出橫截面

```
if(dot(vertexCor + vec3(126.0f, 126.0f, 126.0f), coeff.xyz) > coeff.w){
    CHECK = vec3(1.0f, 1.0f, 1.0f);
}
```

- 畫出iso-value 的值方圖

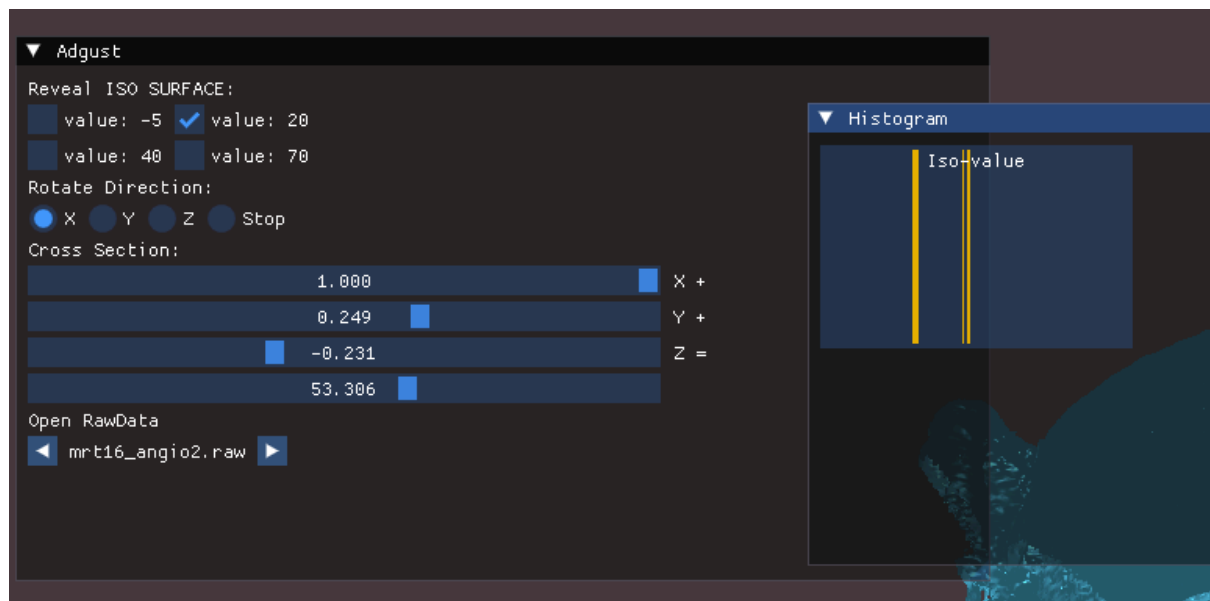
```
ImGui::PlotHistogram("#iso_value", MyIso -> data[dataInd], MyIso->data0
```

- 混和多個 iso-surface

```
if(openSurf[i]){
    MyShader -> setVec3("objColor", gridcolor[i]);
    MyIso -> draw(dataInd * 4 + i);
}
```

功能(可互動項目):

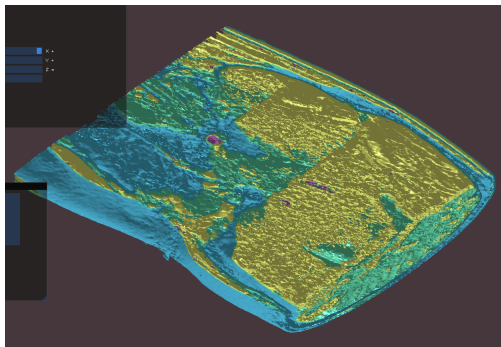
本次作業操控皆由滑鼠點擊交互完成



- (複選)選擇不同的 iso-level
- (單選)切換旋轉方向
- 挑整以切除 $aX + bY + cZ < w$ 的點
- (real time)切換不同檔案

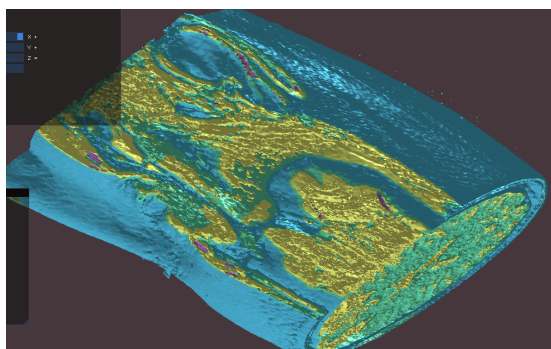
成果展示：

對照組(封面那張)：

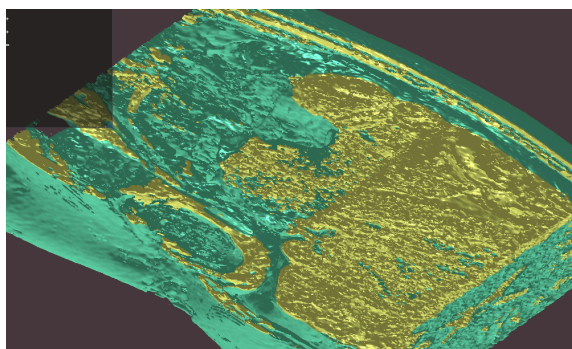


實驗組：

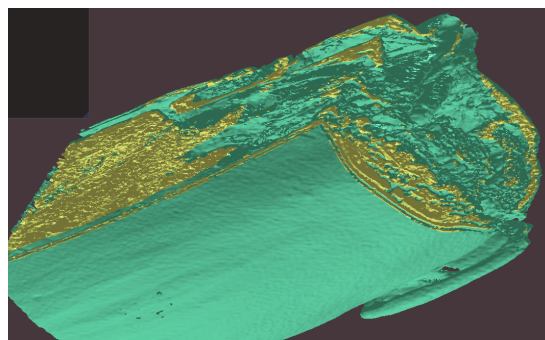
▪ 不同橫截面位置



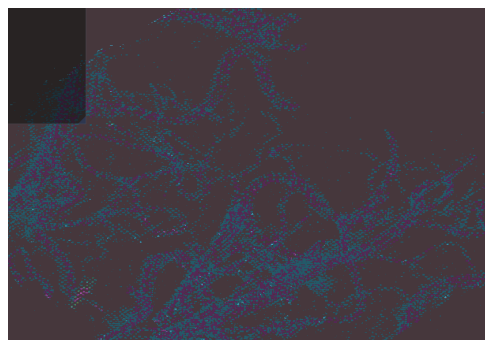
▪ 不同混合iso-surface



▪ 不同角度



▪ 不同物體



心得：

一開始先複習了 GLSL, 參考了作業提供的學習資料, 我才能把它完成。原本我再做切換檔案的時候, 是做要換了在載入另一個, 但我自己測試覺得一開始先等久一點載入, 然後切換real time體感比較好, 所以就變成現在這個版本。