## CT可视化说明

**模块**

Source files内demo.cpp 是main function， shaders是着色器部分，common是用OpenGL和C++实现的可视化功能 ，DcmToModel包含万里的读取dcm文件、外部获得的图片转模型算法

**Demo内部变量**

开头全局变量：

const GLuint WIDTH = 1024;

const GLuint HEIGHT = 768;

Width & height 是窗口的高和宽

const char\* PATH // DCM path

= "D:\\VS\\Project\\DJ\_medical\\CT\_img\\Recon\_4";

Path 是需要读取的dcm文件的文件夹

const uint8\_t ISO = 204; // Isosurface

Iso 是生成模型的过滤值，e.g. iso为0时，为整个胸部，包含内脏、肌肉等等，iso为204时只剩胸骨

const uint8\_t THRESHOLD = 225; // Threshold

Threshold 是阈值，如果该顶点的颜色小于阈值，用阈值替换

// MVP variables

mat4 RotationMatrix = mat4(1);

glm::vec3 position = glm::vec3(0, 0, -15);

glm::vec3 up = glm::vec3(0, 2, -15);

glm::vec3 rotX = glm::vec3(1, 0, 0);

glm::vec3 rotY = glm::vec3(0, 1, 0);

这部分是MVP矩阵，因为需要记忆旋转向量X和Y，放在的全局变量。

视角旋转公式用的是罗德里格斯公式，标在旋转功能里（controlsForFOV）

Function：

void dcmFileToModel(

const char\* path,

const uint8\_t iso,

const uint8\_t threshold,

std::vector<glm::vec3> & objVertices,

std::vector<unsigned int> & objFaces,

std::vector<int> & colors,

int &rescale\_intercept,

unsigned short &rescale\_slope

)

dcmFileToModel 用于DcmToModel模块，读取dcm文件并生成类似于obj文件格式的点面和每个点的颜色

Main:

// Set vertex, color and normal

std::vector<glm::vec3> vertices;

顶点

std::vector<unsigned int> faces;

面

std::vector<glm::vec2> uvs;

UV坐标

std::vector<glm::vec3> normals;

法向量

std::vector<int> colors;

每个点对应的8bit颜色

// Get vertex via loading raw file

int rescale\_intercept;

unsigned short rescale\_slope;

用于把0-255颜色转换为CT值（HU）