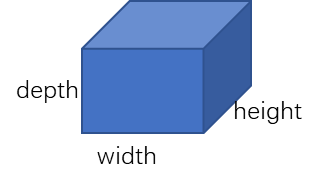
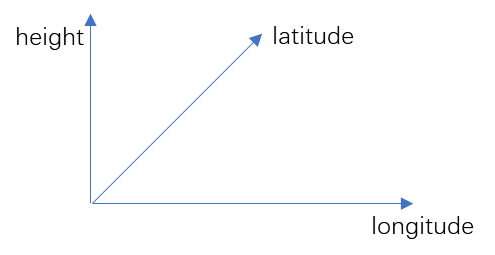
**海洋数据数据读入**

主要是OceanData.hpp文件里

体绘制需要一个规则的三维纹理，为了转换得到，需要海洋数据点自带位置和标量两个属性。



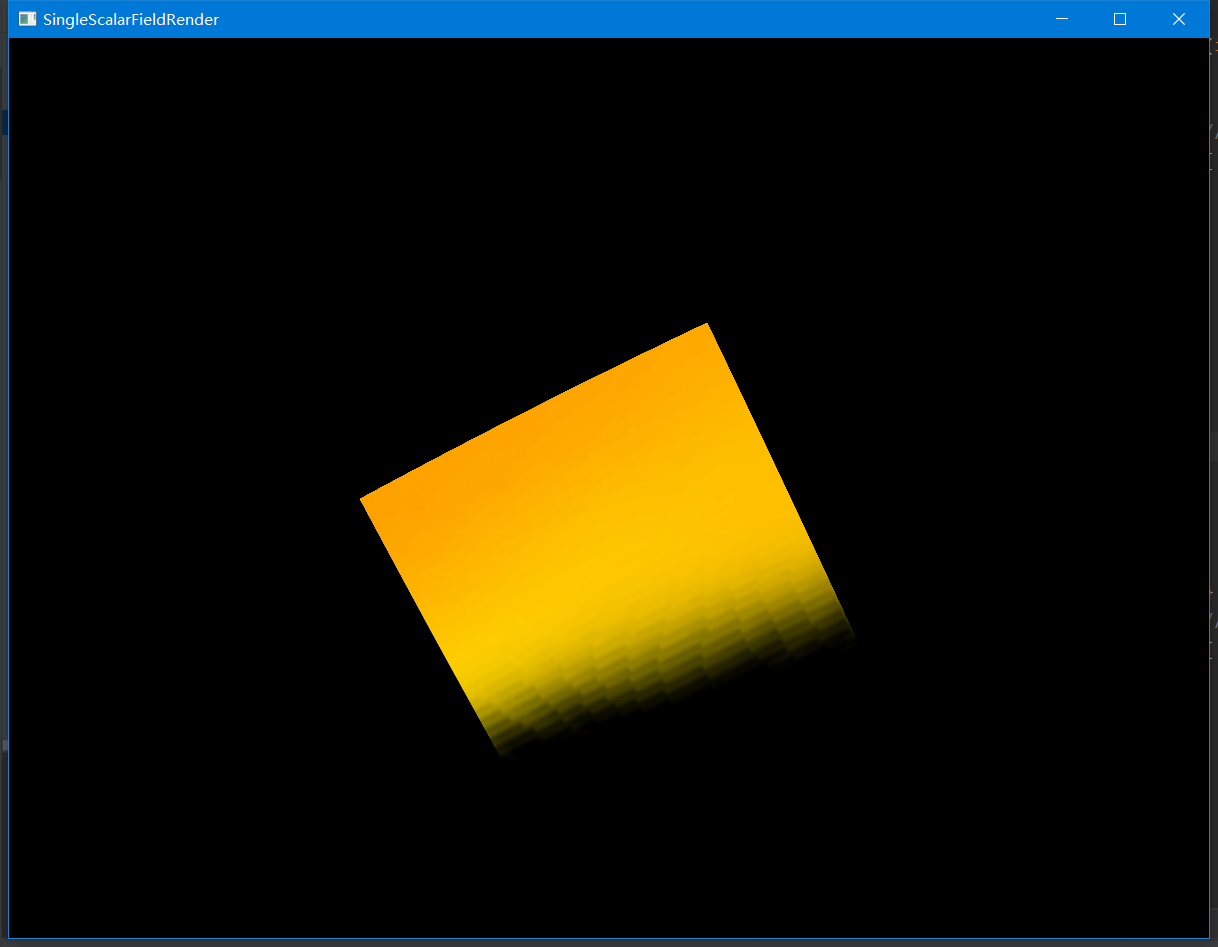
对于海洋数据点，同一经度的视为同一行，同一纬度的视为同一列，同一高度的视为同一层。所以要求海洋数据点每一个高度对应的点都是同样的经纬度坐标，同时还要求每一层的经纬度范围一致。满足这些要求下，可以提供四个文件：经度坐标、纬度坐标、高度坐标、标量数据（每个数据文件末尾不要有多余的回车）。设每个文件里数据的个数为：NLong，NLat，NHeight，NScalar，则需要满足 NLong \* NLat \* NHeight = NScalar。



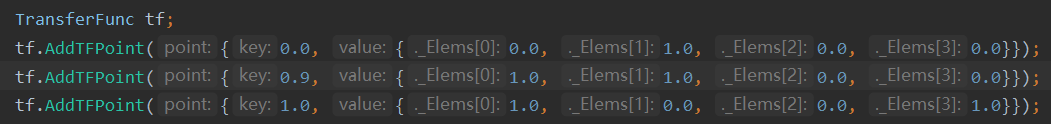
提供了该函数用于读取海洋数据点，输入是上面四个文件的路径（顺序要对），目前读取文本文件是按照float读取的，如果要改成double可以在函数里替换。因为绘制要求的纹理是uint8，所以对于读取的数据进行了拉伸，按最小值和最大值拉伸到了0-255。

其余的一些函数主要是涉及到绘制原理，计算体数据的包围盒mesh和直角坐标范围之类。

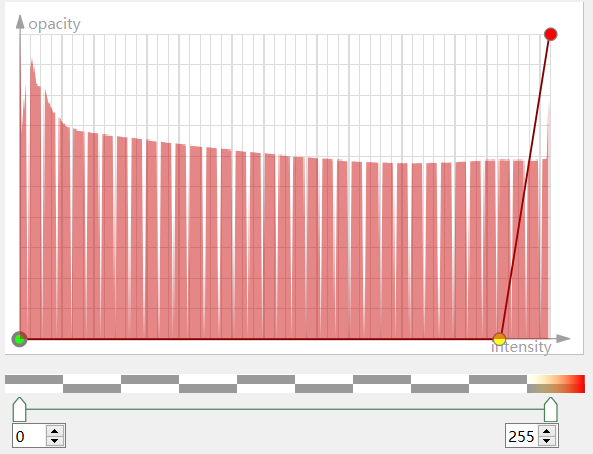
**单个标量场的渲染**



对应的传输函数为



传输的函数意思是标量值(0-255)对应发射的颜色RGBA，体绘制会模拟光线从眼睛打出，光线会一定间隔前进，光线会计算累计的吸收的颜色，当透明度达到0.99就停止，得到最终呈现的颜色。传输函数由几个点定义，点与点之间的颜色和透明度通过线性插值得到，具体如下图。横坐标是0-255，纵坐标是透明度，每个点定义颜色和透明度。调o整合适的传输函数，可以看到感兴趣的体数据内容，包括一些三维结构。



所以根据这个传输函数定义，上图中的渲染结果可以解释为

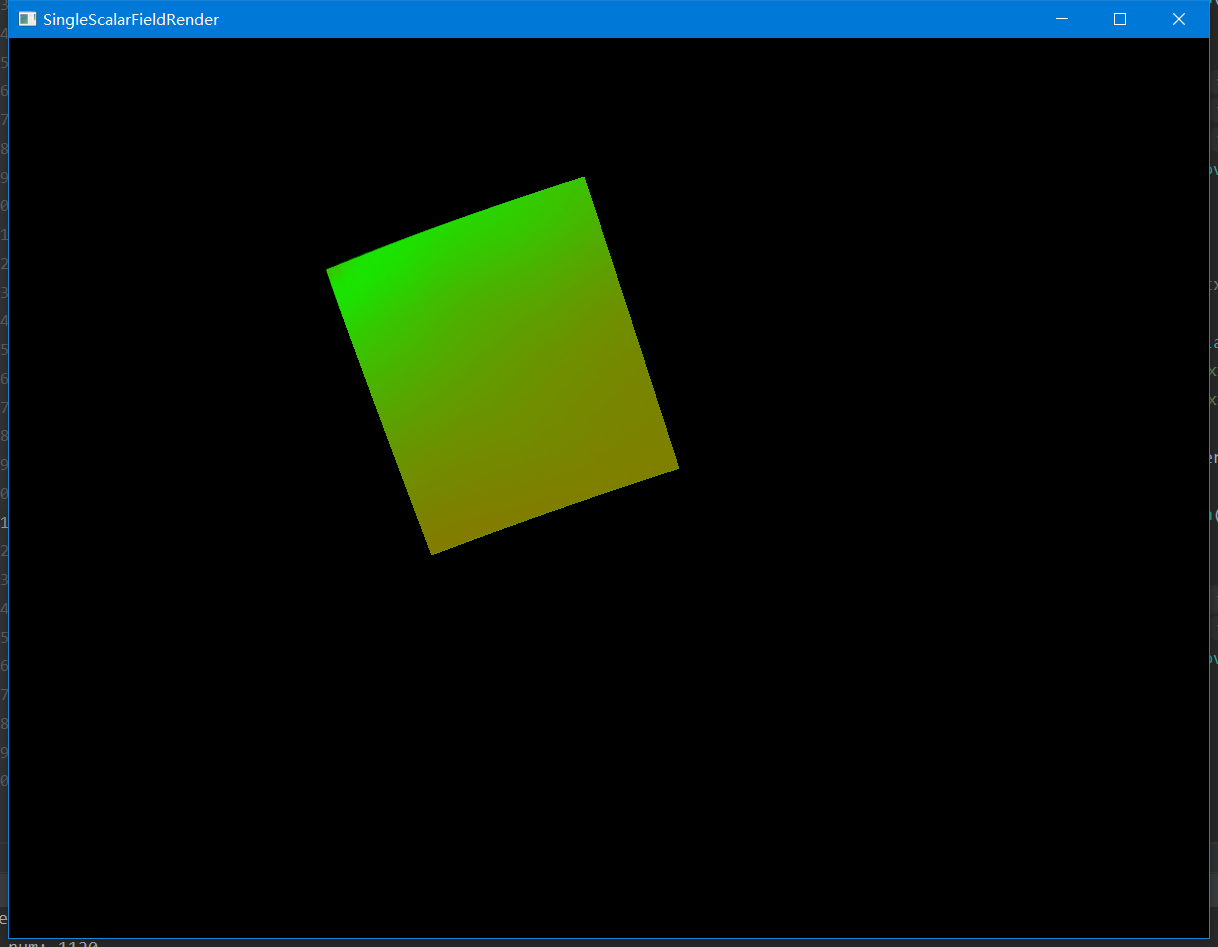


这部分归一化后的数据比较小

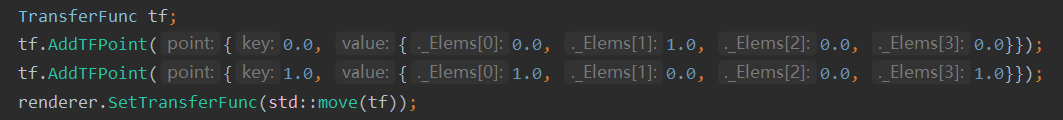


这部分归一化后的数据比较大

**相似性比较后的单个体渲染**



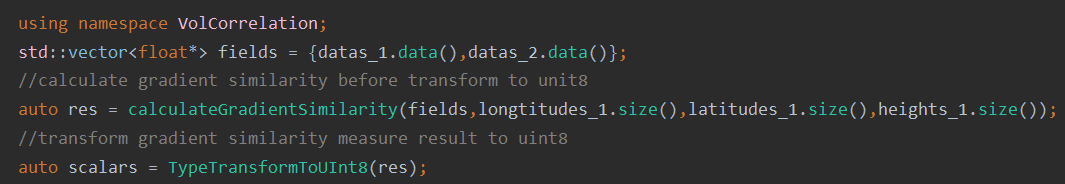
就是比较范围相同的两个数据之间的相似性，数值越小越相似，上图渲染用的传输函数为



也就是说值越小越接近绿色，值越大越接近红色，相似性越小。

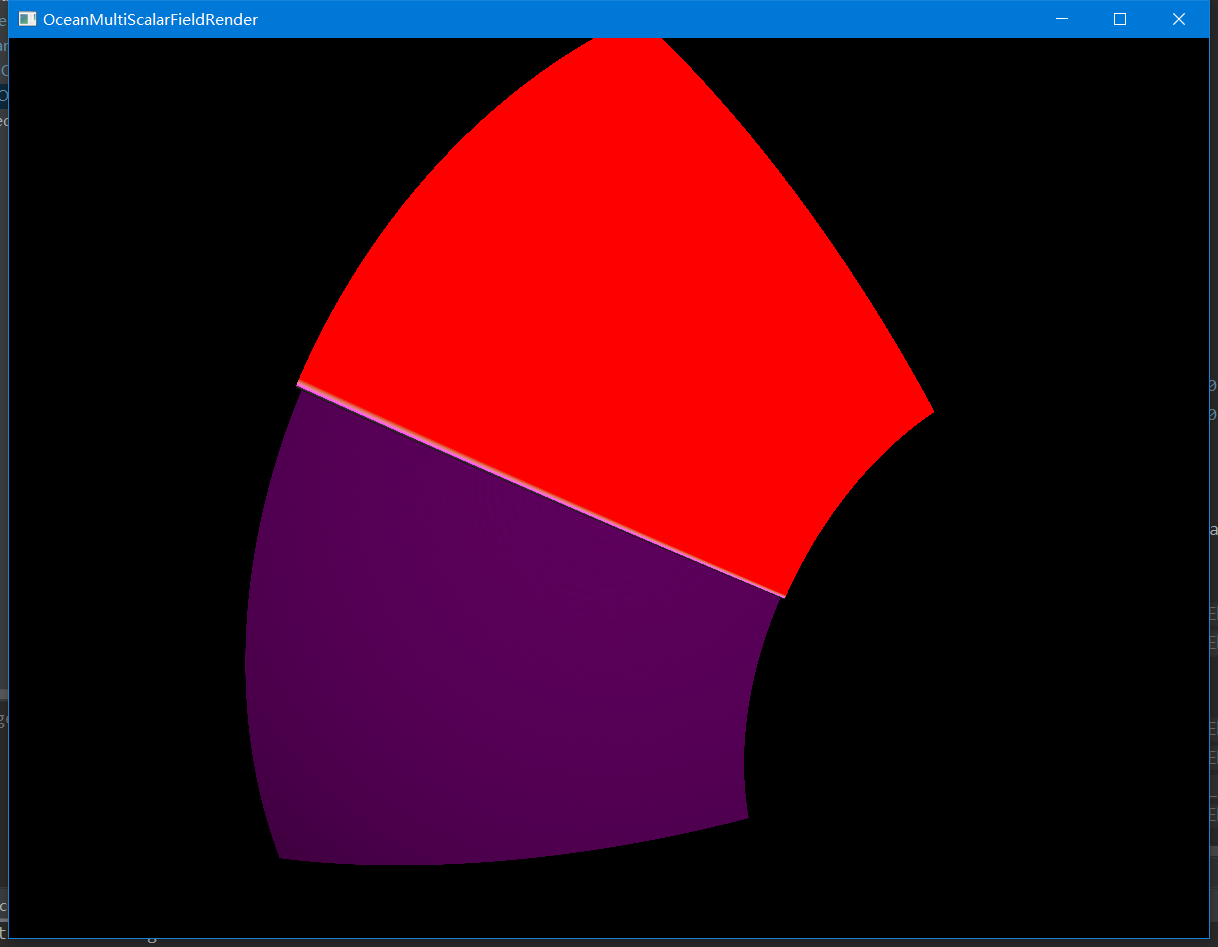


生成相似性的数据函数，输入是两个数据各自的数据路径配置，得到一个差异性结果的OceanData。真正用于计算差异性的部分为



calculateGradientSimilarity函数在GradientSimilarityMeasure.hpp中

**两个体的混合标量场绘制**



适用于一些不同属性但是同一范围的数据一起绘制，最终呈现的颜色是两个数据各自的颜色乘以透明度后相加混合的结果

**两个等值面的混合绘制**

等值面绘制就是绘制出指定值的那一个面，所以颜色是单一的，需要指定，不会普通体绘制那样的颜色渐变。混合等值面绘制就是各自的等值面