<https://www.youtube.com/watch?v=6oMZb3yP_H8>

想要理解 marching cube，先理解 marching square，这两者的区别只是维度的问题，

Marching square是二维平面上的算法，而 marching cube 则是三维的算法。

我简单描述下 Marching square，二维平面上有个图形，是用隐函数的形式表现得，比如一个圆： x^2 + y^2 - r^2 = 0 这样的隐函数，假设我们只有他的隐函数，想绘制出它得轮廓来，该如何处理？

Marching square提供了一套框架去解决这类问题，它得原理就是轮廓上得点一定是另外两个点之间：在轮廓内，在轮廓外（中值定理）。

A black background with white text and numbers

Description automatically generated

基于前面得信息，想方设法去搜索这两类点，但是瞎找肯定是不行的，为了更好的逼近这个轮廓，这些点我们需要足够得靠近，因此算法就寻找一个像素得四个点，如果四个点都是在轮廓得一边，则不用处理。如果四个点分别位于轮廓的两边，则可以通过插值得方式粗略估算出轮廓得点在哪（这个插值是通过计算四个点代入到轮廓函数的值，里面和外面的正负是相反的），因为是四个点，要在四个点之间生成一条直线，用来模拟这个像素中得轮廓，分了好几种情况（绿色点是轮廓内得点，黄线是生成得估计轮廓）。

当然，有时候为了计算快速，连插值都不查了，直接用中点，毕竟是在一个像素内。

A screenshot of a computer

Description automatically generated

建议详细看这个视频，是非常好用的效果

<https://www.bilibili.com/video/BV18z421d7gm/?vd_source=2fd56c06057ed87c77543a25d43c6a6e>

那么 Marching cube呢？

同理，marching cube也是要绘制出三维得轮廓表面函数，不过是用面片的形式连接起来。上面我们不是用插值点找出来后连城直线么，而三维的情况，就是插值点弄出来，连成面片。另一个不同点就是采样的情况，marching square是用像素，而三维空间是用体素，所以你做这个算法的前需要规定体素的大小，当然把空间划分的越精细越好

Marching cube的常规用法，是生成地形中的山谷，峡谷，洞穴等，因为这些东西用高度图是无法生成的（比如洞穴，同一个高度有多个值，但是高度图无法实现这些），怎么做呢？

<https://www.youtube.com/watch?v=M3iI2l0ltbE>

<https://www.bilibili.com/video/BV1iL4y1K74G?spm_id_from=333.788.videopod.sections&vd_source=2fd56c06057ed87c77543a25d43c6a6e>

其实marching cube不是看到的那么简单，他的优化以及实现都挺复杂，这里不再展开，具体看视频

Todo：

<https://0x00000000.dev/terrain-generation/#Marching-Cubes-Algorithm>

等学到unreal的shader编程后来看看这个以及，compute shader