

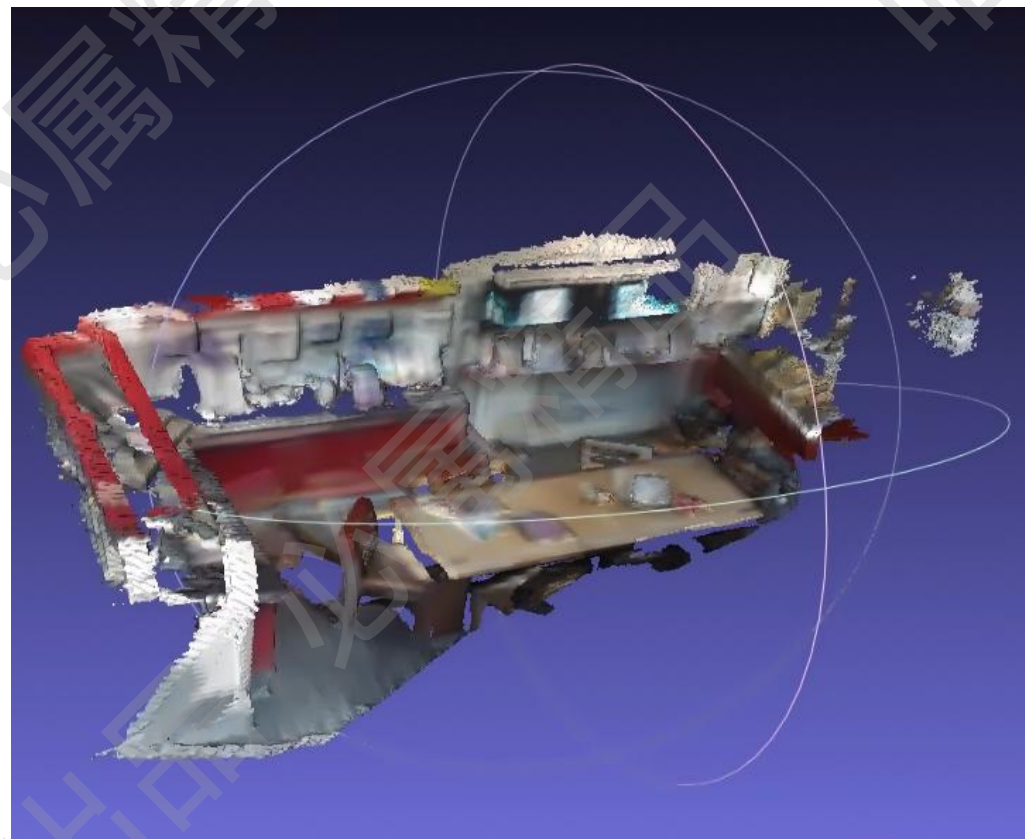
✓ 整体概述: (Truncated Signed Distance Function)

✎ 如何得到右图的重建结果呢?

✎ 先看看咱们手里有啥东西:

✎ 1.原始图片; 2.对应的深度信息

✎ 3.每张图的相机位姿; 4.相机内参



TSDF

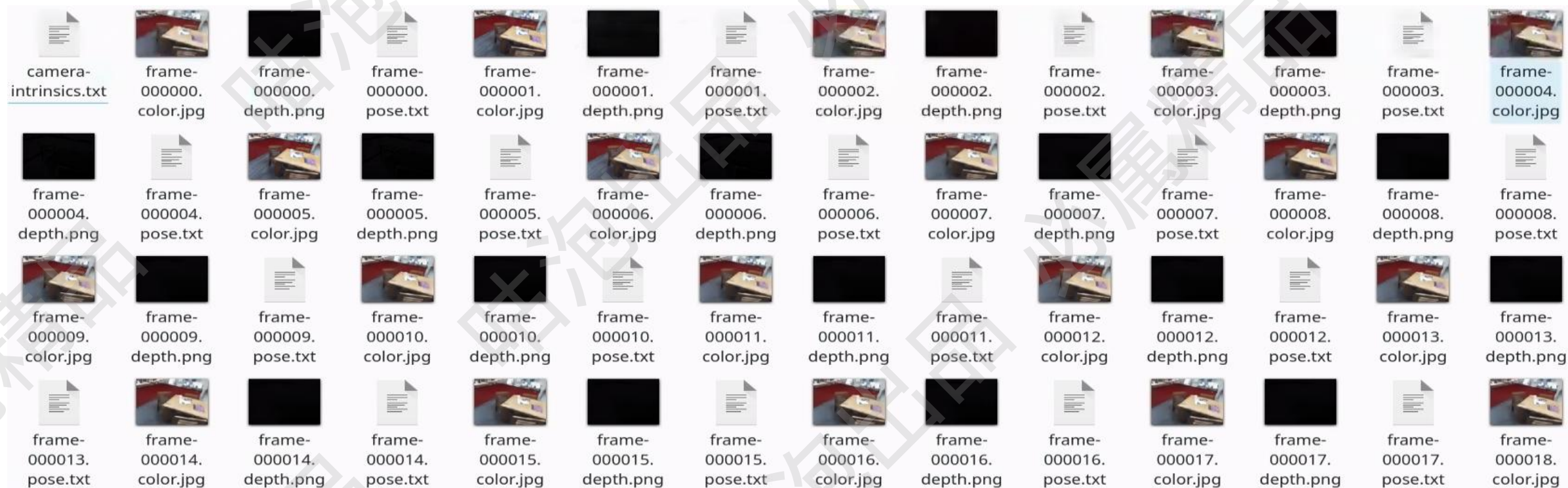
✓ 手里有的东西:

📎 原始输入图像数据 (就是一个场景多个视角拍摄的结果)



✓ 手里有的东西:

✎ 输入图像的深度信息, 位姿信息等 (一般由相机得到, 也可以通过算法得到)



TSDF

✓ 整体概述：

✎ 这个也就是TSDF的更新和组合的过程

✎ 一步步计算各个体素（位置的）TSDF值再把他们拼接起来



TSDF

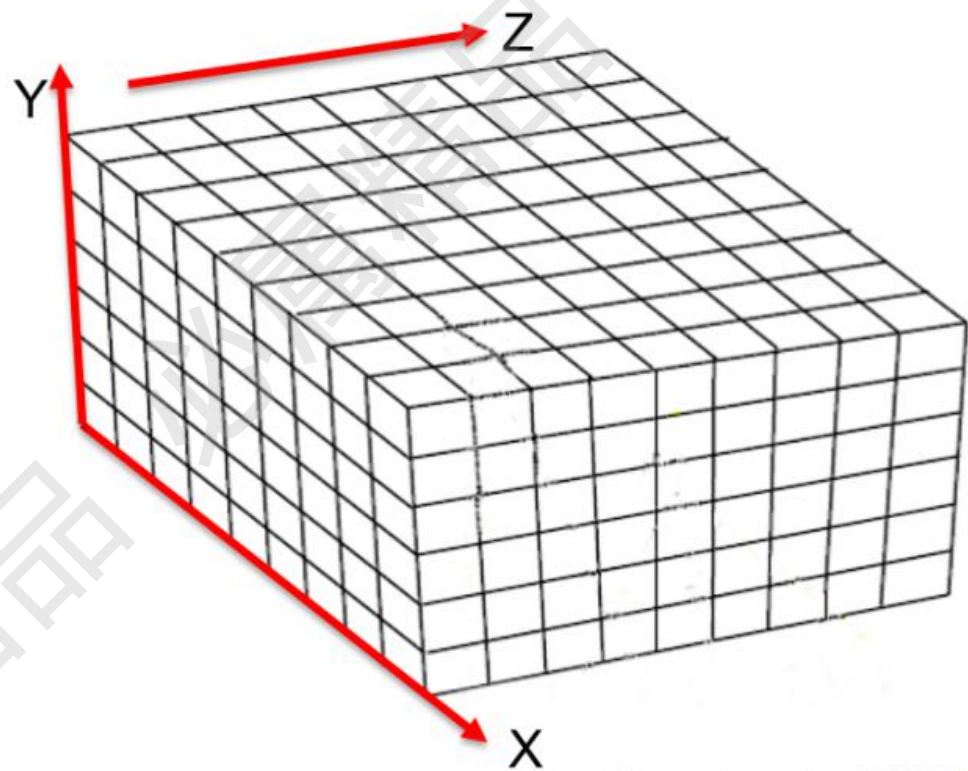
✓ 流程分析：

✎ 首先需要构建一大块空区域（不能放大，GPU放不下。。。）

✎ 一般咱们把这个大区域叫做volume

✎ 它由很多个小区域组成，每个叫voxel(体素)

✎ 需要计算每个voxel的TSDF值及其权重



TSDF

✓ 初始化操作:

✎ 一个三维的大区域(TSDF图)由 $L \times W \times H$ 个三维小方块组成

✎ 其中每一个小方块的大小也是自己定义的, 例如 $0.05\text{m} \times 0.05\text{m} \times 0.05\text{m}$

✎ $5\text{m} \times 5\text{m} \times 5\text{m}$ 的真实三维场景中有 10^6 个体素, 计算量还是很大的

✎ 一般都用GPU来算, CPU也可以, 但是会很慢

TSDF

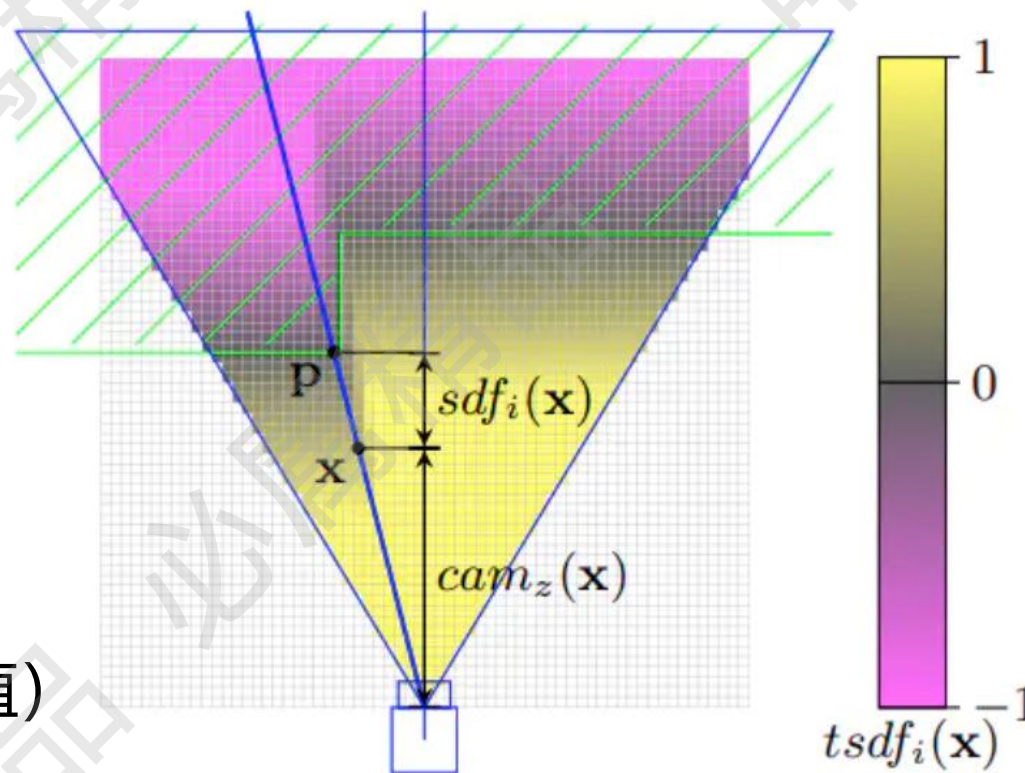
✓ 要对每个体素计算什么呢？(一个体素的 SDF 值，是它到最近的表面的距离)

✎ X是其中一个体素，绿色是离它最近平面

✎ P点是平面上离X最近的一点

✎ 它俩之间的距离就是SDF(要计算的结果)

✎ 其中T表示截断的意思 (太近太远的设为定值)

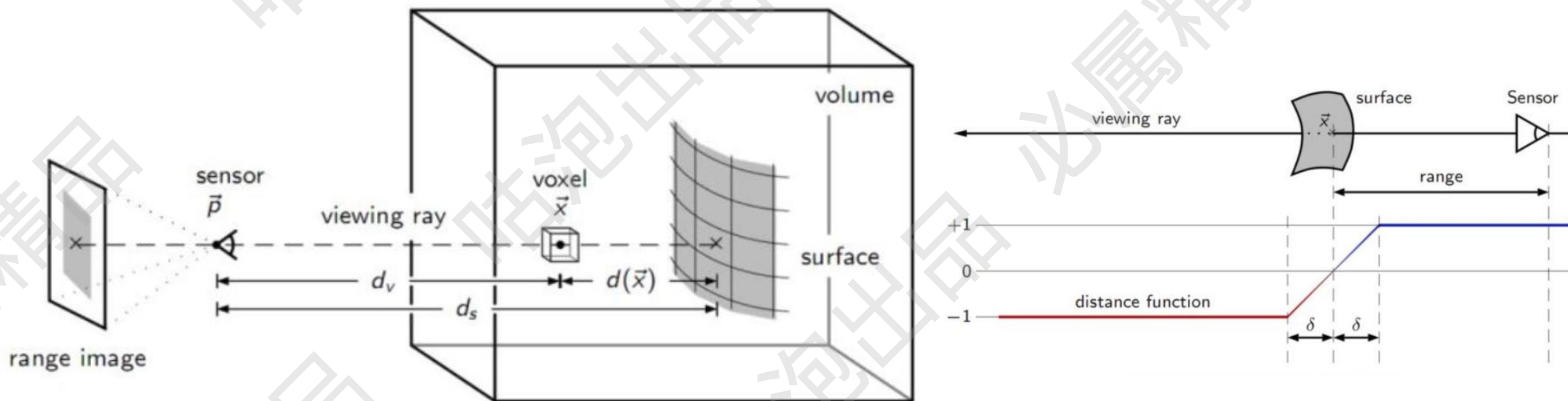


TSDF

✓ 如何计算TSDF呢?

✎ 面到相机的深度是 d_s ，相机采集到的深度 d_v ，则 $d(x) = d_s - d_v$

✎ $d(x) > 0$ 说明该体素在真实的面的前面；反之则说明该体素在真实的面的后面



- ✓ 转换操作 (ds , 其实就是我们的深度图中的数值)
- ✎ 遍历每一个体素, 计算其在世界坐标系中的位置
- ✎ 根据初始化“大块”时设置的极点位置和体素大小决定
- ✎ 再将其在世界坐标中的值转换成其在相机坐标系中的位置(位姿 RT 已知)
(其到相机坐标原点的距离就是 dv)
- ✎ 再根据相机内参, 转换到像素坐标 (因为我们由深度图, 得到像素坐标就能得到其深度值)

✓ TSDf结果:

✎ 正负1表示离的很远)

 最终得到了多有体素的结果

 当前只表示其中一帧，还要融合



✓ TSDF组合

✎ 我们的输入由很多视角的图像组成，但是大块就那么一个

✎ 所以咱们的大块是一直在更新的，但是并不是算完所有的一起更新

✎ 相当于每一个体素的值是一种平均的感觉，各个视角的图都会对其产生影响

✎ 但是第二帧是由其结果与第一帧组合得到，第三帧就是组合了第二帧
(就像咱们之前的动图，是一个增量式的过程，并不是最后一次组合)

$$\text{tsdf}(t) = g(\text{tsdf}(t-1), \text{tsdf}(t)), t \geq 1$$