知乎

KinectFusion论文阅读

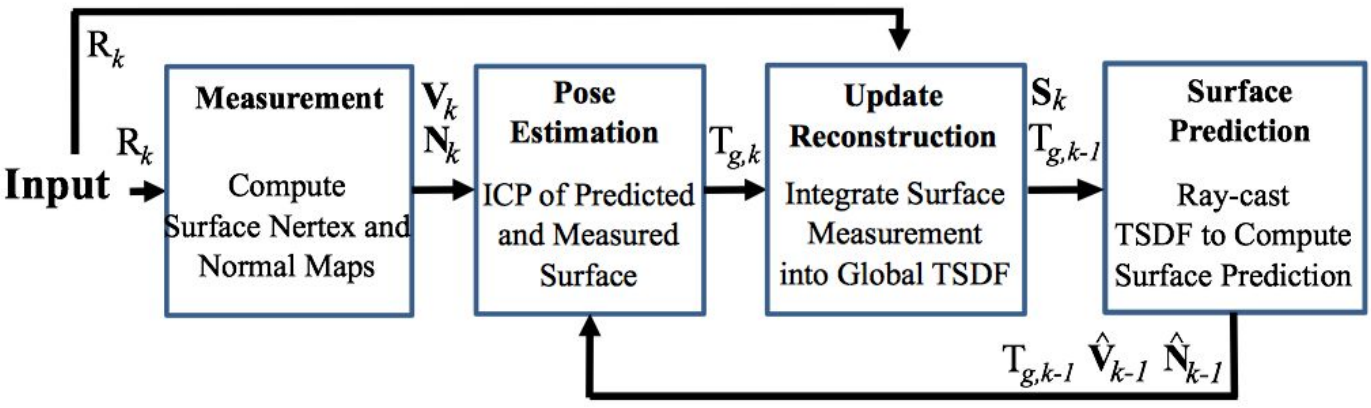
灯灯

95 人赞同了该文章

「必读论文之——

KinectFusion: Real-time dense surface mapping and tracking
ieeexplore.ieee.org

方法概述



Overall system workflow

Surface measurement

知乎

Surface reconstruction update

假设pose estimation已经计算出来，就可以把本次测量结果融合到全局地图（global model）中了。这里的model使用的是TSDF地图。TSDF后面再详细介绍。

Surface prediction

将TSDF raycast到估计的frame上

Sensor pose estimation

使用ICP算法对准predicted surface与当前的sensor measurement，得到估计的pose。

下面我们来看看每一步分别是怎样进行的，以及和其他步骤的联系。

0. 符号约定

$\mathbf{T}_{g,k} = \begin{bmatrix} \mathbf{R}_{g,k} & \mathbf{t}_{g,k} \\ \mathbf{0} & 1 \end{bmatrix}$ ：**相机到世界坐标（global frame）**的transformation matrix

$\mathbf{p}_k$ ： 某点在相机坐标系中的坐标

$\mathbf{p}_g$ ： 某点在世界坐标系中的坐标，二者满足： $\mathbf{p}_g = \mathbf{T}_{g,k}\mathbf{p}_k$

$\mathbf{u} = (u, v)^\top$ ： 像素点坐标

$\dot{\mathbf{u}} = (\mathbf{u}^\top | 1)^\top \in \mathbb{R}^3$ ： 坐标齐次化

$\mathbf{q} = \boldsymbol{\pi}(\mathbf{p}) \in \mathbb{R}^2$ ： 投影变换，对于 $\mathbf{p} = (x, y, z)^\top$ 有 $\boldsymbol{\pi}(\mathbf{p}) = (x/z, y/z)^\top$

$\mathbf{K}$ ： 相机内参，从相机坐标系中一点到像素平面的转换方程为 $\mathbf{u} = \boldsymbol{\pi}(\mathbf{K}\mathbf{p}_k)$

1. Surface Measurement

在时刻 k 从Kinect传来一帧**原始**深度图 $\mathbf{R}_k$ ，像素 $\mathbf{u} = (u, v)^\top$ 对应的深度为 $\mathbf{R}_k(\mathbf{u})$ ，为了降

逐像素计算就得到了所谓的vertex map，也就是相机坐标系下的一帧点云图。接下来，我们通过

$$\mathbf{N}_k(\mathbf{u}) = \boldsymbol{\nu}[(\mathbf{V}_k(u+1,v) - \mathbf{V}_k(u,v)) \times (\mathbf{V}_k(u,v+1) - \mathbf{V}_k(u,v))] \quad (2)$$

计算得到像素 $\mathbf{u} = (u,v)^\top$ 对应的空间点 $\mathbf{V}_k(\mathbf{u})$ 的法向量 $\mathbf{N}_k(\mathbf{u})$ 。

注意到目前计算的vertex和normal都是在相机坐标下的表示，若要将它们转换到世界坐标系，只须通过

$$\begin{aligned} \mathbf{V}_k^g(\mathbf{u}) &= \mathbf{T}_{g,k}\dot{\mathbf{V}}_k(\mathbf{u}) \\ \mathbf{N}_k^g(\mathbf{u}) &= \mathbf{R}_{g,k}\mathbf{N}_k(\mathbf{u}) \end{aligned} \quad (3)$$

计算得到。这样我们就从深度图得到vertex map和normal map了，至于它们的作用，要到第3节才能看到。

2. Surface Construction

这里暂时与上一步没有联系，我们先关注这样一个问题：给定相机的pose estimation $\mathbf{T}_{g,k}$ ，要求将一帧深度图融合到当前的3D地图（也可以称作3D model）中，其中，地图用TSDF表示。现在我们就需要来稍微了解下TSDF是什么东西了。

TSDF全称truncated signed distance function，是一种三维地图的表示方法，在SLAM中用得还是比较多的，它的key idea很简单，就是用一个大的*volume*作为要建立的三维地图/模型，volume由许多个小的*voxel*组成，每个voxel对应空间中一个点，用 $\mathbf{S}_k(\mathbf{p})$ 记录两个量：

1. $F_k(\mathbf{p})$ ：该voxel到最近的surface（一般称作zero crossing）的距离
2. $W_k(\mathbf{p})$ ：weight（融合一帧新的图像时需要用到）

$$\mathbf{S}_k(\mathbf{p}) \mapsto [F_k(\mathbf{p}), W_k(\mathbf{p})]. \quad (4)$$

我们参考下图（原文链接）来更清楚地理解以上提到的新概念：

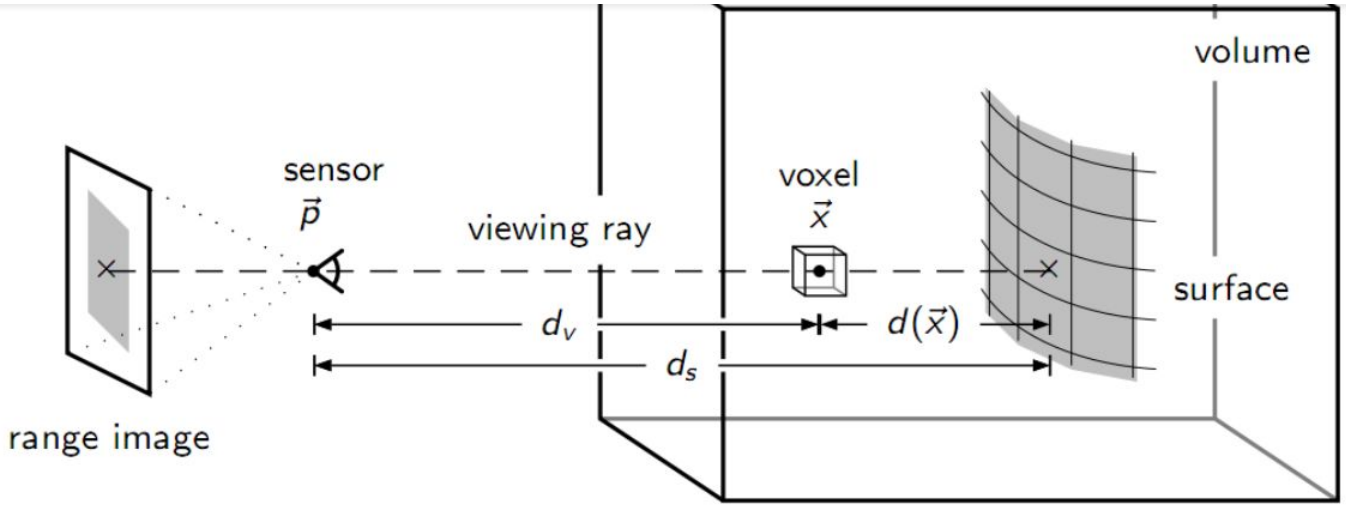


Figure 1: Signed distance $d(\vec{x}) = d_s - d_v$

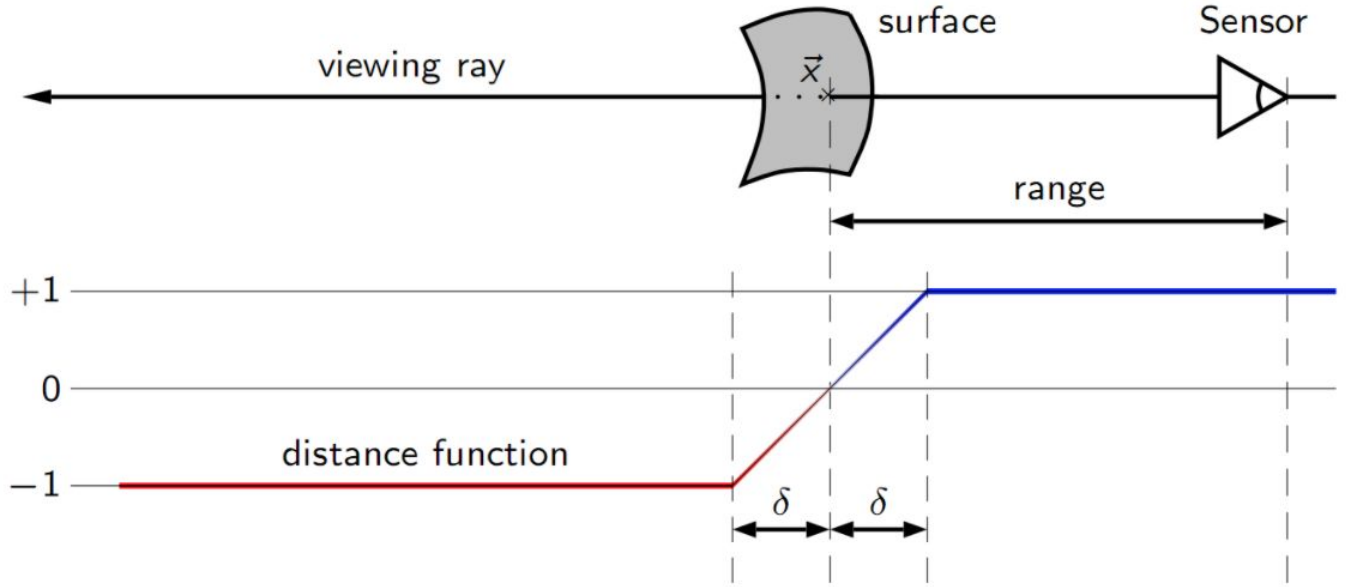


Figure 2: Scaled and truncated signed distance function

TSDF示意图

图中，线性变化区域被限制在 $[-\delta, \delta]$ ，即所谓的**truncated**；幅值被除以 δ ，即所谓的**scale**。

假设Kinect提供的深度信息存在 $\pm\mu$ （相当于上图的 δ ）的不确定度，即：某像素点的深度测量值为 d ，那么我们认为 $d \pm \mu$ 的范围内都是可能是surface上的点。设 r 是距离相机光心的范

space；**从surface顺着ray方向**（对应sensor视角下不可见区域）超过 μ 距离的voxel，它们记录的 $F_k(\mathbf{p})$ 一律计为null（因为此时观察不到这些区域，本次测量针对它们的 $F_k(\mathbf{p})$ 应视作无效，不参与接下来的global TSDF fusion操作）。

这个将原始深度图 $\mathbf{R}_k$ 转换到世界坐标系global frame下的TSDF*的完整过程，用公式来描述如下（看起来稍微有点复杂）：

$$
\begin{aligned}
F_{R_k}(\mathbf{p}) &= \Psi(\lambda^{-1}||\mathbf{t}_{g,k} - \mathbf{p}||_2 - \mathbf{R}_k(\mathbf{x})), \qquad (5)\\
\lambda &= ||\mathbf{K}^{-1}\dot{\mathbf{x}}||_2,\\
\mathbf{x} &= \lfloor \boldsymbol{\pi}(\mathbf{K}\mathbf{t}_{g,k}\mathbf{p}_k) \rfloor,\\
\Psi(\eta) &= \begin{cases} \min(1, \frac{\eta}{\mu}\mathrm{sgn}(\eta)) & \text{iff } \eta > -\mu \\ \text{null} & \text{otherwise} \end{cases}
\end{aligned}
$$

其中， $\lfloor \cdot \rfloor$ 表示最邻近查找函数，因为通过透射变换计算得到的 $\boldsymbol{\pi}(\mathbf{K}\mathbf{t}_{g,k}\mathbf{p}_k)$ 未必是整数，而像素坐标都是整数，所以我们查找 $\boldsymbol{\pi}(\mathbf{K}\mathbf{t}_{g,k}\mathbf{p}_k)$ 最邻近的整数坐标，再根据 $\mathbf{R}_k(\mathbf{x})$ 得到对应的深度。

注意，这里 $||\mathbf{t}_{g,k} - \mathbf{p}||$ 前面除了一个系数 λ ，是因为 $\mathbf{R}_k(\mathbf{x})$ 是深度值，所以需要把相机到点的**距离**转为**深度**。至于为什么不是给 $\mathbf{R}_k(\mathbf{x})$ 乘一个系数呢？作者的解释为：

> we found no considerable difference in using SDF values computed using distances along the ray or along the optical axis

权重 $W_{R_k}(\mathbf{p})$ 也有对应的公式，这里省略。现在，我们计算得到了第k帧深度图对应的TSDF，我们需要将最新的这个TSDF与之前的global TSDF进行融合。

记第1到 $k-1$ 帧深度图融合得到的TSDF的记录值为 $[F_{k-1}(\mathbf{p}), W_{k-1}(\mathbf{p})]$ ，那么我们的更新公式为：

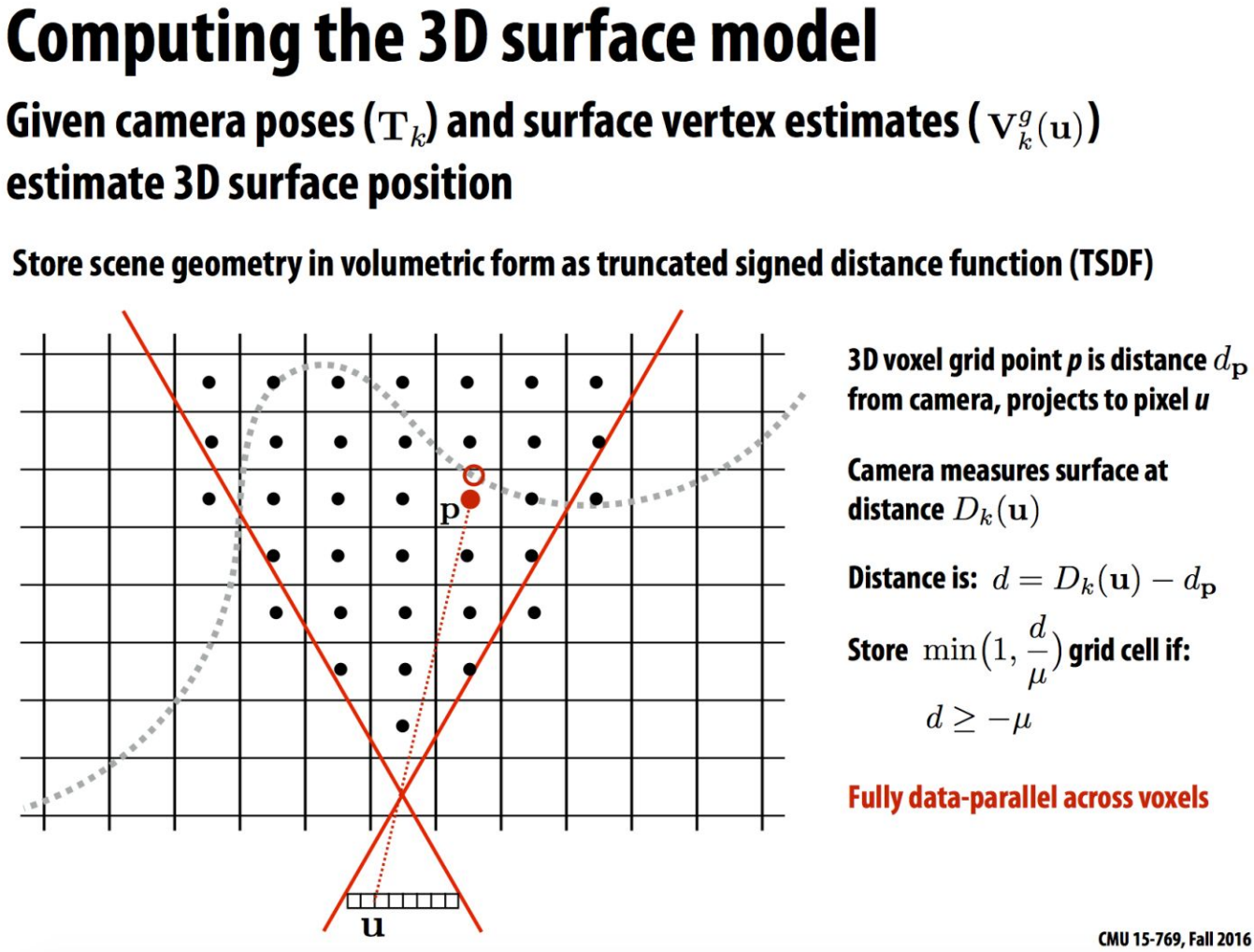
$$
F_k(\mathbf{p}) = \frac{W_{k-1}(\mathbf{p})F_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p})F_{R_k}(\mathbf{p})}{W_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p})} \qquad (6)
$$

$$
W_k(\mathbf{p}) = W_{k-1}(\mathbf{p}) + W_{R_k}(\mathbf{p}) \qquad (7)
$$

注意区分 $F_k(\mathbf{p})$ 和 $F_{R_k}(\mathbf{p})$ 以及 $W_k(\mathbf{p})$ 和 $W_{R_k}(\mathbf{p})$ ，它们的含义是不同的，一个是累计更新量，maintain全局数据；一个是当前帧计算值（相当于incremental amount）。

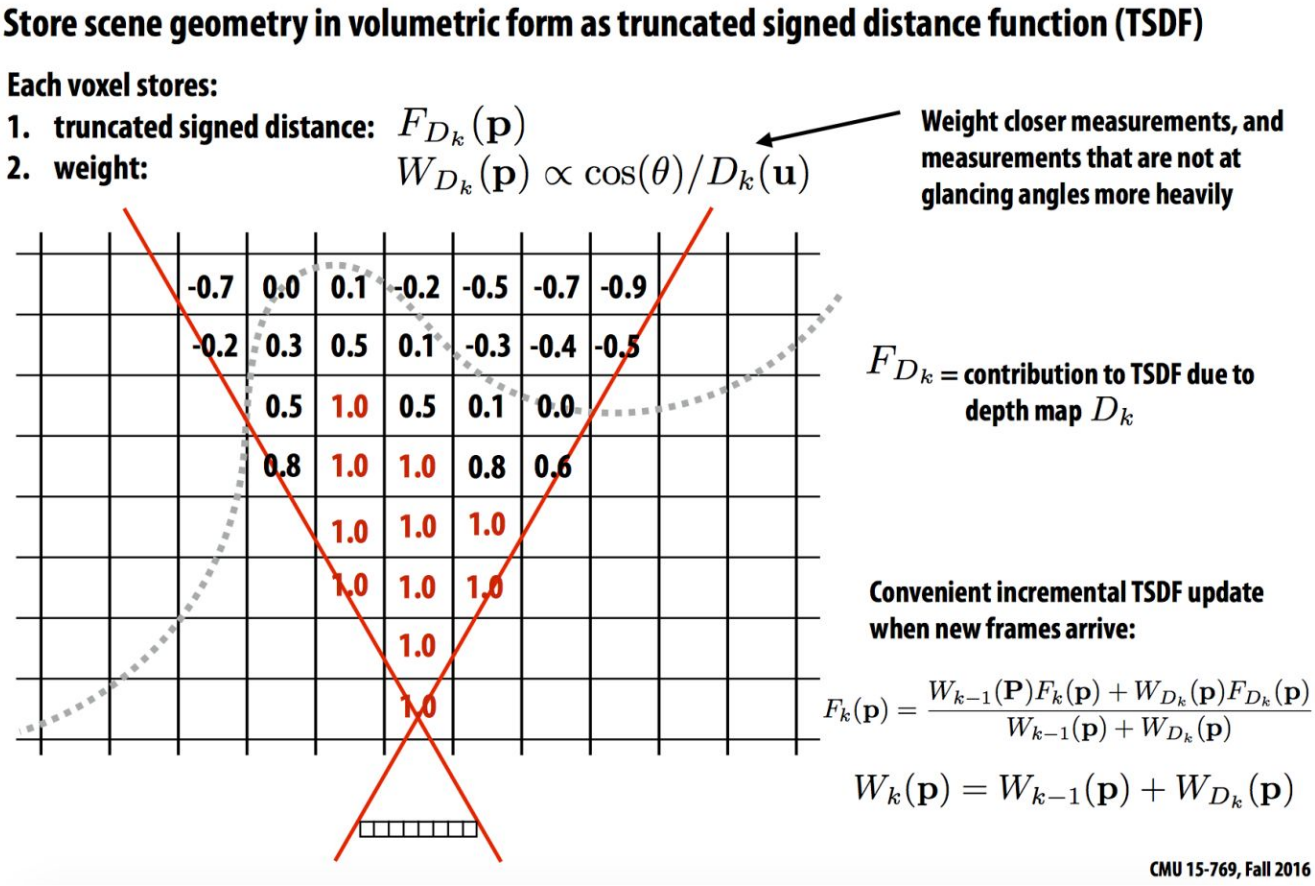
知乎

下图可以更好地帮助理解：



Surface Construction 1 对应公式(5)

知乎



Surface Construction 2 对应公式(6)(7)

3. Surface Prediction from Ray Casting the TSDF

上一节我们能进行surface construction的前提是给定相机的pose estimation $\mathbf{T}_{g,k}$ ，所以在本节和下一节中我们主要解决pose estimation的问题。

问题描述：已有 $k-1$ 时刻的global TSDF（即上一节中最后得到的update 后的TSDF）、位姿估计 $\mathbf{T}_{g,k-1}$ 以及 k 时刻的深度信息 $\mathbf{R}_k$ ，估计 k 时刻的位姿 $\mathbf{T}_{g,k}$ 。

解决的方法分为两步：

1. 根据 $k-1$ 时刻的global TSDF得到surface prediction
2. 利用 k 时刻的surface measurement（见第1节）与第1步中surface prediction，使用ICP算法预测 k 时刻的位姿 $\mathbf{T}_{g,k}$

知乎

注意：这里的相机是虚拟的（virtual camera），因为camera的实际测量结果已经在上一节中融合到global TSDF中去了，所以我们让一个**相同pose** $\mathbf{T}_{g,k}$ 的camera去观察updated global TSDF，计算此时得到的vertex map $\hat{\mathbf{V}}_k$ 和normal map $\hat{\mathbf{N}}_k$ ，所以才称作surface prediction。之所以不直接用camera的实际测量结果，而使用reconstructed TSDF，是因为：

> “frame-to-model tracking” intuition: TSDF model is more accurate/complete than the single measured depth map from frame k-1

本文使用的是**ray casting（光线透射）**方法，注意这里的讨论都是在世界坐标下进行的。光线从像素 $\mathbf{u}$ 可以测量的最小深度（受限于传感器）出发，沿着方向 $\mathbf{T}_{g,k}\mathbf{K}^{-1}\dot{\mathbf{u}}$ ，直到遇见zero crossing（TSDF从正变到负， $+ve \rightarrow -ve$ ）认为找到visible surface，加入到vertex map中。有两种情况认为没有找到surface：

1. 遇到back face（TSDF从负变到正，$-ve \rightarrow +ve$ ）
2. 直到沿着ray搜索完working volume还没有遇到zero crossing或back face

这是原理上寻找intersection的方法，实践中还用到了interpolation以及增加march along the ray的步进距离（因为一次前进一个voxel的效率比较低，可以使用一个小于 μ 的step加速寻找过程）。

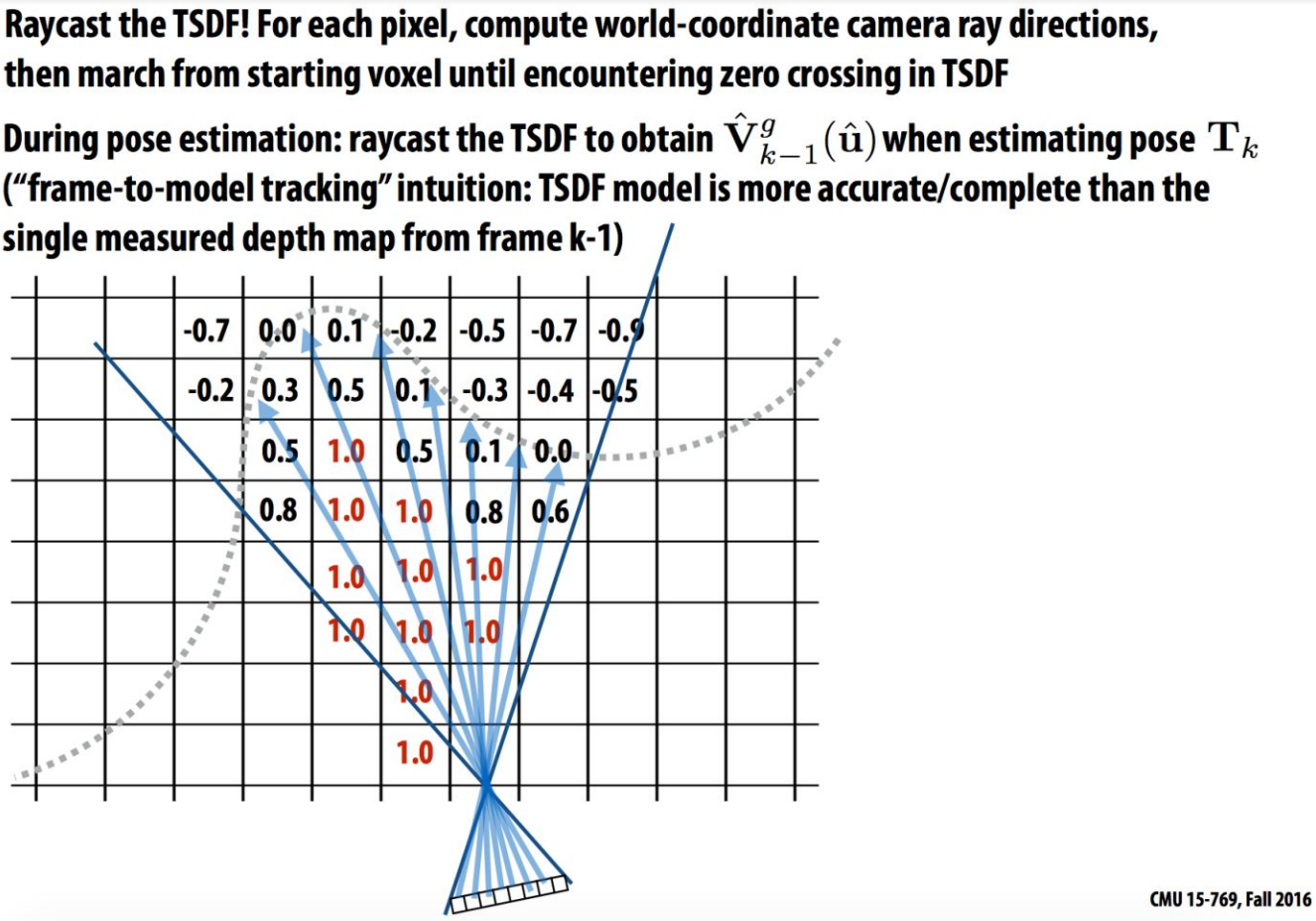
在near surface区域，法向量通过梯度计算得到：

$$\mathbf{R}_{g,k}\hat{\mathbf{N}}_k(\mathbf{u}) = \hat{\mathbf{N}}_k^g(\mathbf{u}) = \nu[\nabla F(\mathbf{p})] \qquad (8)$$

对每个pixel进行一次ray casting，我们完成了surface prediction，结果存储在**global frame**下的vertex map和normal map中。

由于Kinect的深度测量是有一定范围的（本文取0.4m～0.8m），所以在ray casting的时候还要加上这个限制得到的预测结果才是合理的。

知乎



Raycast the TSDF

4. Pose Estimation

和第3节结合起来，我们接下来就要实现pose estimation了。从数据流角度分析的话：

输入：

- 上一帧计算出的的surface prediction $[\hat{\mathbf{V}}_{k-1}, \hat{\mathbf{N}}_{k-1}]$
- 当前帧深度图计算出的surface measurement $[\mathbf{V}_k, \mathbf{N}_k]$ （参见**第1节Surface Measurement**）

输出：

- 当前相机的位姿估计（pose estimation） $\mathbf{T}_{g,k}$ 。

知乎

想一想：这里的 $\hat{\mathbf{V}}_k^g(\hat{\mathbf{u}})$ 为什么由上标 g ？括号内的变量是 $\hat{\mathbf{u}}$ 头上带尖括号？和 $\mathbf{u}$ 的关系是？

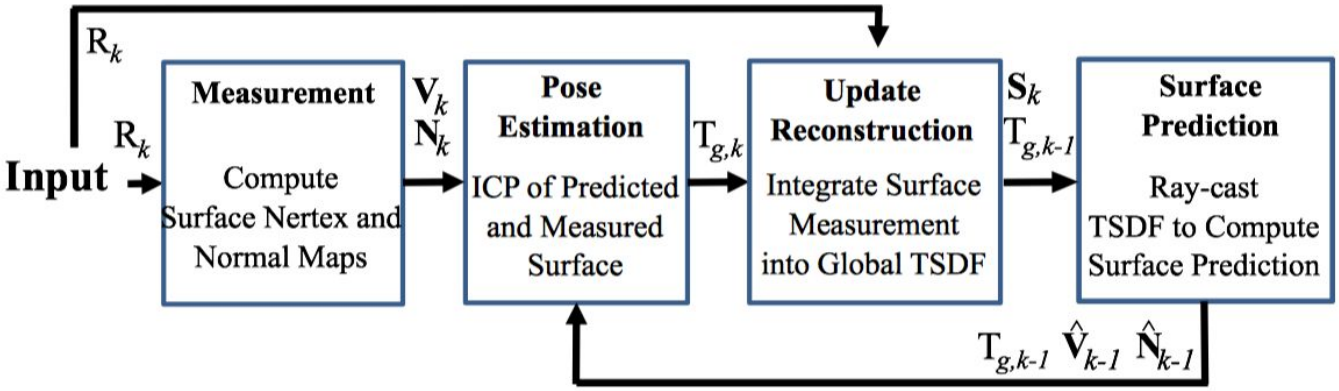
这里关于 $\hat{\mathbf{u}}$ 和 $\mathbf{u}$ ，要注意ICP算法要求每两个点都必须一一对应，这是SLAM中常常用到的**data association**概念。这里默认已经使用projective data association algorithm将两个vertex map集配准了，所以在(9)式中的相减才是有意义的。

$\Omega_k(\cdot)$ 可以看作一个筛选函数，排除掉误差过于大的点，其评判依据可以参见论文中的公式(17)。

接下来我们只需要对(9)式进行非线性优化（论文中使用的Gauss-Newton法），就可以得到最后的最佳pose estimation $\mathbf{T}_{g,k}$ 了，这里就不赘述非线性优化的方法。这里需要给 $\mathbf{T}_{g,k}$ 设定一个初始值，一般直接设为上一帧的pose estimation $\mathbf{T}_{g,k-1}$ 。

5. Summary

再回到开头来回顾整个流程

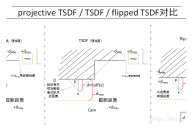


1. 首先，从Kinect读入一帧深度图，计算出measured surface：vertex map和normal map备用。
2. 然后读入前一帧计算出的surface prediction，结合当前帧的measured surface，执行ICP算法，得到当前帧对应的相机位姿 $\mathbf{T}_{g,k}$ 。
3. 有了相机位姿 $\mathbf{T}_{g,k}$ ，就可以将本帧的深度数据融合到此前maintain的global TSDF $[F_{k-1}(\mathbf{p}), W_{k-1}(\mathbf{p})]$ 中，更新TSDF为 $[F_k(\mathbf{p}), W_k(\mathbf{p})]$ 。
4. 最后，根据reconstructed的TSDF $[F_k(\mathbf{p}), W_k(\mathbf{p})]$ 以及 $\mathbf{T}_{g,k}$ ，进行该帧的surface prediction（反馈到下一个循环的第2步），close the loop。

知乎

* 本文采用的projective TSDF，并非true TSDF，关于二者的区别参见下文

projective TSDF/TSDF/flipped TSDF 三种截断符号距离函数比较的个人…
blog.csdn.net



** 在本节中使用的深度图**不是**第1节中描述的双边滤波后的深度图，是原始深度图：

> Finally, we note that the raw depth measurements are used for TSDF fusion rather than the bilateral filtered version used in the tracking component, described later in section 3.5. The early filtering removes desired high frequency structure and noise alike which would reduce the ability to reconstruct finer scale structures.

参考资料

1. Volumetric Range Image Integration
2. Visual Computing Systems : Fall 2016
3. 高翔，视觉SLAM十四讲5.1 相机模型

编辑于 04-19

同时定位和地图构建（SLAM） Kinect 计算机视觉

推荐阅读

SLAM论文阅读

要开始阅读SLAM