# 使用虚拟立方体改进kinfu设计方案

## Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更人 | 变更说明 | 变更时间 |
| V1.0 | 张琛 | 初稿 | 2016/03/06 |

## 引言

论文[1]使用轮廓线约束，改进KinectFusion[2]中ICP点云配准方案，减少相机轨迹漂移的情况，提升相机姿态 M={R, t}求解的准确性、鲁棒性。其核心在于，对于当前深度图（第i帧）上提取的轮廓线Ci，在(i-1)帧模型上，利用“视线与物体边缘法向近似正交”特性[3]，提取对应点候选点集，然后对构建kdtree，进行对应点匹配，将轮廓点对应关系集合作为加权项，增加到原来的优化目标函数中：

(1)

这一方法解决了传统的ICP及其变种[4]，在平滑表面场景中常见的配准失败问题。

论文[1]的难点在于对于，**如何找到相对准确的**，以对配准漂移形成有效约束。其关键贡献是，提出了一种**基于水平集方法求解曲面法向的算法**，使得场景物体边缘法向与视线的正交特性更鲁棒，从而得到更稳定的候选点集。

本设计报告中，同样尝试在第i帧和第i-1帧之间，利用场景中真实存在的高精度立方体，进行“虚拟立方体”的生成与定位，以获得更好的轮廓线，即，从而约束ICP配准，具体步骤如下所述。

## 步骤

### 虚拟立方体初始化

#### 自动版本（识别）

持续检测真实深度图中是否存在三个相邻的正交面，当存在时，确定三个面种子点，使用法向进行区域增长，在各个面上获得较多的点，然后各自进行平面拟合，得到三个平面参数模型，进而求交，即得到虚拟立方体一个顶点，以及三条可视棱边。以顶点和棱边为参考，生成整个虚拟立方体。

#### 手动版本（预处理生成）

【已实现】 对一组采集好的离线数据，使用kinfu算法，生成场景点云模型，根据场景中真实立方体的姿态M={R, t}，手动生成高精度虚拟立方体，并调整姿态，使其与真实立方体大致重合。

假设第一次生成（自动或手动）虚拟立方体的帧为第i帧，记虚拟立方体为。

### 真实场景立方体识别、参数化

从真实场景中识别立方体参数模型，我们采用迭代的方式。

给定输入数据：第i帧虚拟立方体（参数模型or点云？），第i+1帧深度图，期望求解第i+1帧虚拟立方体参数模型。步骤如下：

1. 虚拟立方体可视面获取，

在任意视点下，立方体可能有1~3个可视面。我们采集数据时，保持相机姿态为从场景斜上方俯视，因此立方体可视面有3个（情形a）或者2个（情形b）。

通过光线投射的方式，我们判定哪些面为可视面，并分别将各个面的点标记label为<1, 2[, 3]>，得到label图（分辨率与深度图相同），表示为。

1. 深度图邻面识别

输入：及其对应label图；；

输出：深度图上label图

对于像素，求其对应3D点坐标以及法向 ，求对应位置的点坐标以及法向，若满足：

其中，【暂定】，则标记。

1. 根据 上label图，将各个面进行平面拟合参数化。

情形a：若中有三种标记，则有三个可视面，得到三个平面参数模型。因其两两正交，必然可以解得三条交线，一个顶点。根据三边应有长度，修正三边测得物理长度，进而生成整个虚拟立方体。

情形b：若中有两种标记，则只有两个可视面，这种视角也是原来kinfu容易产生漂移的情况。两个可视面的点集各自拟合得到两个参数平面，得到一条交线。但是因为深度图边缘锯齿问题，此交线的两端点无法准确定位，即与这两个可视面正交的邻面无法定位。试图将这个问题变为优化问题，还没有想好。已知信息有：(a) 此交线应有长度; (b) 带锯齿的边缘点3D坐标（由深度图得到的相机参考系下的坐标）

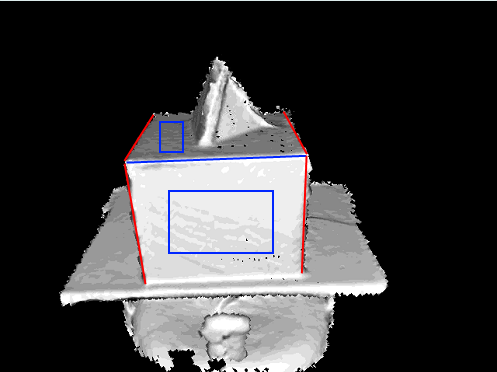


图 1 立方体只有两个可视面（蓝色矩形框标记）的情形。图中蓝色线段，为这两个可视面交线，其直线参数方程可以较准确拟合；红色线段，为带锯齿的物体边缘深度图，这里不精确导致蓝色线段的端点定位不准确。

### 立方体线框用作ICP配准加权项

假设我们得到了第i帧、第i+1帧立方体线框（全部或仅用可视部分），如论文[1]方法，使用kdtree（目前在CPU上），进行二者的对应点匹配，然后放入优化目标公式(1)中，运行算法，并调整权重w，观察此加权项对配准结果的影响。

(1)

## 参考文献

[1] Zhou, Qian-Yi, and Vladlen Koltun. "Depth camera tracking with contour cues." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[2] Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]//Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011: 127-136.

[3] Cipolla R, Blake A. Surface shape from the deformation of apparent contours[J]. International journal of computer vision, 1992, 9(2): 83-112.

[4] Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm[C]//3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on. IEEE, 2001: 145-152.