# KinectFusion一些实现在立方体数据上的测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更人 | 变更说明 | 变更时间 |
| V1.0 | 张琛 | 初稿 | 2016/03/20 |

## 引言

本报告目的是对现有的一些KinectFusion算法代码进行测试（如，PCL Kinfu[1], VoxelHashing[2], 我们自己基于论文[3]利用PCL Kinfu改写的版本, 以及kfusion[4]），并展示他们目前的主要误差细节。

## 测试数据

我们以高精度立方体作为基座，在上面放置不同的石膏模型（三棱锥、球体、十二面体、贝多芬头像）。立方体被放置在一个可转动的平台上，置于XBOX 360 Kinect (V1)相机视野中，保持**Kinect设备静止**，稳定地旋转平台约三周，得到一组RGB-D视频序列。数据采集环境如下：

**采集设备**：XBOX 360 Kinect (V1)

**采集程序**：NiViewer (**Registration**=ON, **Mirror**=OFF)

**采集数据**：RGB彩色图、Depth数据，**无IR**数据 (与RGB无法同时采集)

**已知高精度立方体模型尺寸**：30.5\*28.0\*25.0 ，暂时无彩色标记物（如棋盘格定标板等）

**目标几何体**：编号与描述见表1。

表 1 待扫描物体编号与描述

|  |  |  |
| --- | --- | --- |
| 编号 | 描述 | 剪裁参数 |
| 00 | 正四面体 | -cropz 1333 -xywh 110 50 400 400 |
| 01 | 球体 | -cropz 1333 -xywh 60 30 480 400 |
| 02 | 六棱锥 |  |
| 03 | 二十面体 |  |
| 04 | 圆锥 |  |
| 05 | 十二面体 | -cropz 1333 -xywh 60 30 480 400 |
| 06 | 正方体 |  |
| 07 | 圆锥+三棱锥 |  |
| 08 | Beethoven头像 | -cropz 1333 -xywh 60 10 480 480 |

注1：本次测试使用00, 01, 05, 08号物体作为重建目标 //2016-3-17 15:35:39

本次测试的数据均为保持Kinect设备静止，旋转目标物体得到，要使三维重建算法正常工作，需要对这些数据进行剪裁预处理。表1第三列标记了剪裁参数。

## 三维重建结果

可以从两个方面评估三维重建的结果：一是相机轨迹；二是重建模型几何结构。

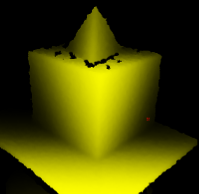
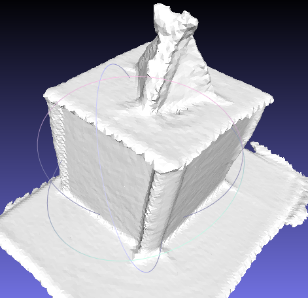
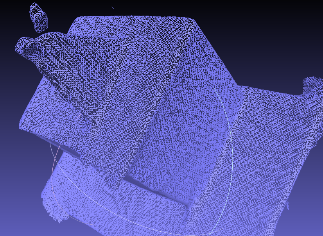
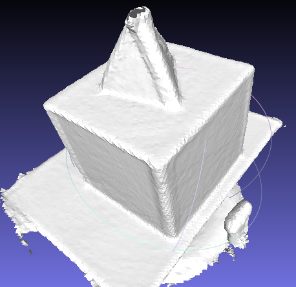
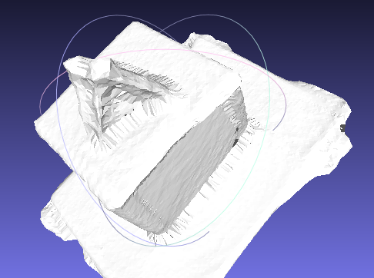
对于相机轨迹，论文[5]使用基于视觉的高精度运动捕捉系统，来精确确定相机的轨迹；对于我们自己采集的数据，有两种方式：一种是固定Kinect相机，稳定转动平台（本次测试目标）；第二种是保持目标物静止，手持Kinect设备环绕扫描。手持设备环绕扫描得到的数据，无法得到确切的相机轨迹，暂不在本次测试范围；对于第一种采集方式，相机与目标物的相对位置关系理论上为一个**正圆**。我们采集的数据本身没有相机轨迹，但是可以通过配准过程得到的相机轨迹估算圆的位置与半径（根据无漂移数据段的圆弧，求解半径）。相机轨迹误差评价指标由[5]定义为: **绝对轨迹**误差(Absolute Trajectory Error (**ATE**))， 以及**相对姿态**误差(Relative Pose Error (**RPE**))。

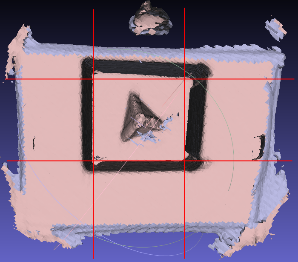
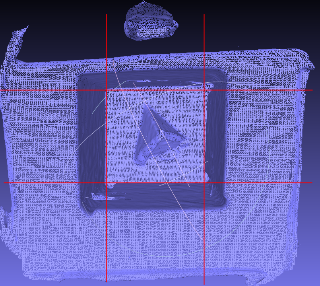
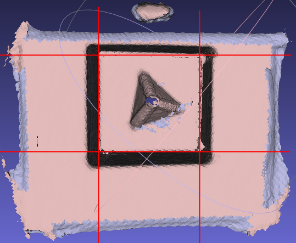
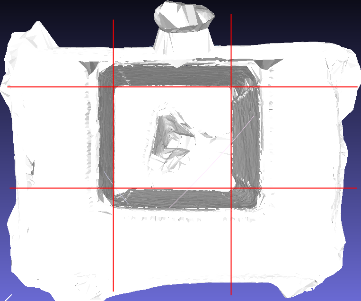
对于重建模型的几何结构评估，定性方式为，肉眼观察不同算法在特定时刻、视角下的配准精度；定量方式为，从重建得到的mesh模型中，尽量精确地抠出目标物（四面体、球体、etc），与使用高精度激光扫描仪得到的mesh模型进行对比，对比方法为，调整两个mesh模型，使其尽量配准（利用ICP算法），计算他们的**RMSE**误差（也可用**热度图**演示？未测试验证 2016-3-21 03:20:54）。

本文**目前仅对重建模型的几何结构进行定性评估**。所用的算法如表2所示。

表 2 本文测试算法列表

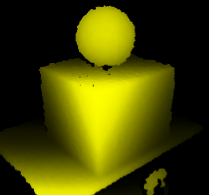
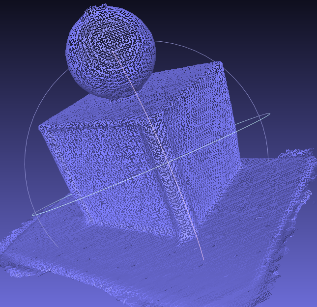
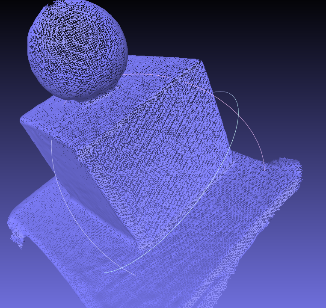
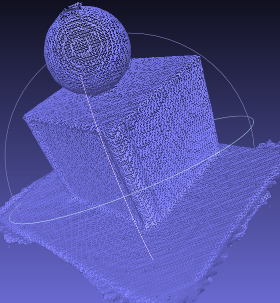
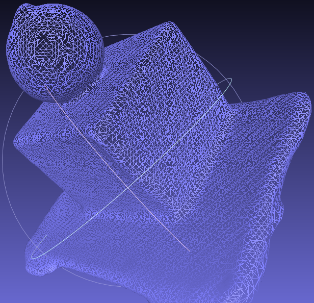
|  |  |
| --- | --- |
| 算法序号 | 备注 |
| 1 | PCL Kinfu[1] |
| 2 | VoxelHashing[2] |
| 3 | Contour Cues约束法[3], 我们基于Kinfu[1]的实现 |
| 4 | Kfusion[4] |

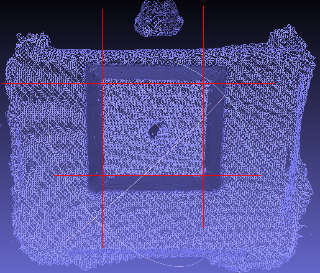
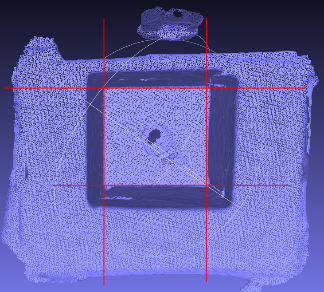
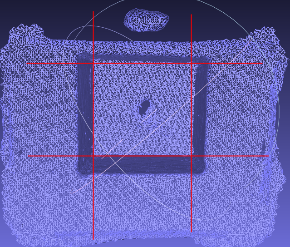
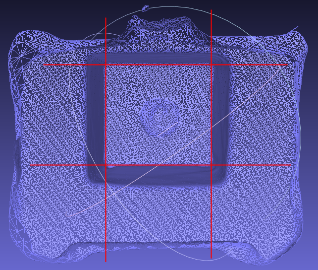
    

(a)RGBD原始数据 (b)PCL Kinfu结果 (c)VoxelHashing结果 (d)ContourCue结果 (e)Kfusion结果

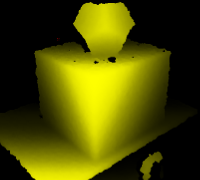
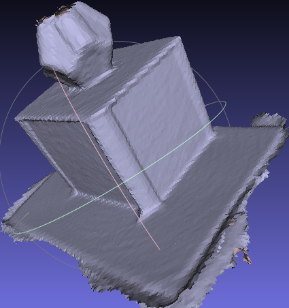
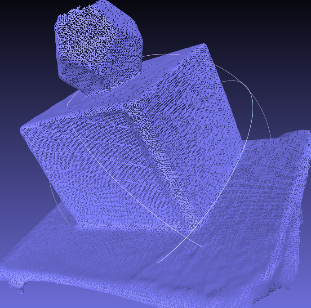
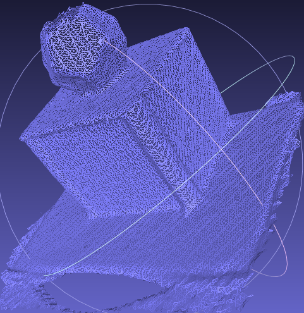
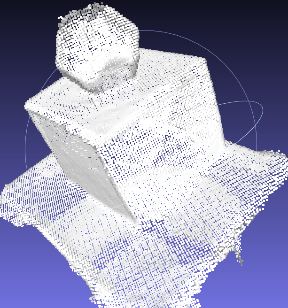
图 1 正四面体为目标的重建结果。(a)中上下分别为深度图/彩色图数据; (b)~(e)为表2中对应算法1~4的结果图, 其中第一行为重建模型的侧上方俯视图, 第二行为模型从底面观察图，红色平行线用于辅助对比（立方体基座理想状态下应该邻面正交，对面平行）

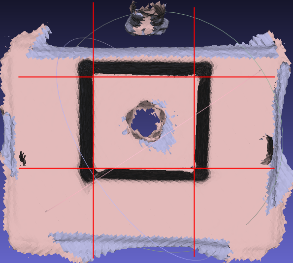
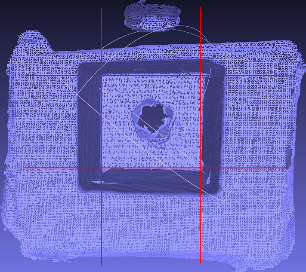
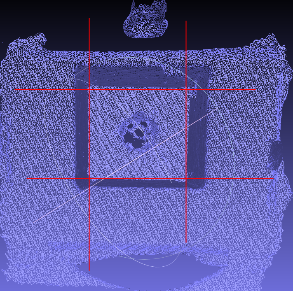
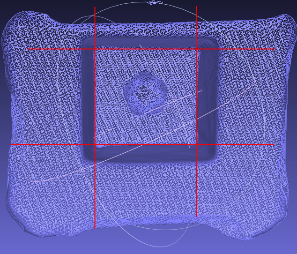
    

(a)RGBD原始数据 (b)PCL Kinfu结果 (c)VoxelHashing结果 (d)ContourCue结果 (e)Kfusion结果

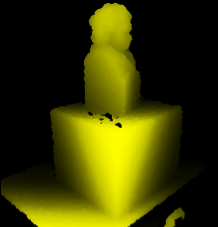
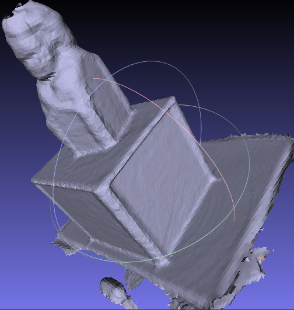
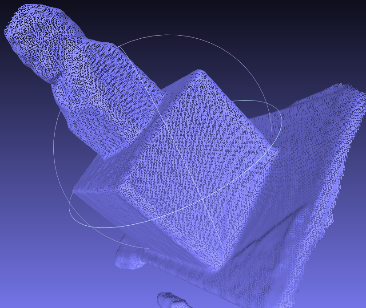
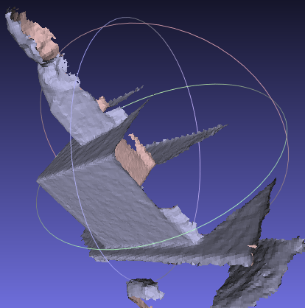
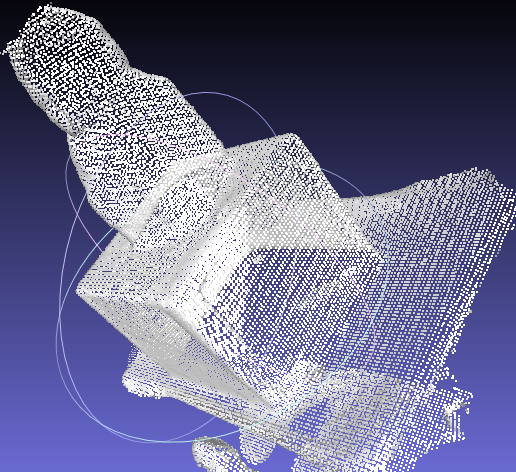
图 2 球体为目标物的重建结果

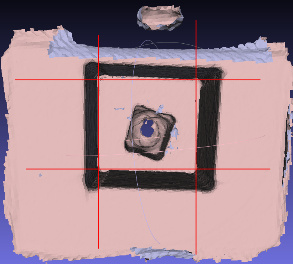
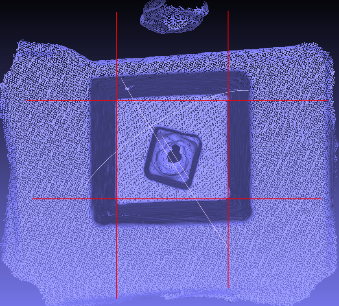
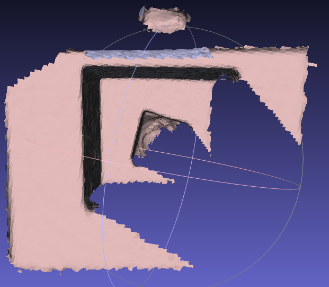
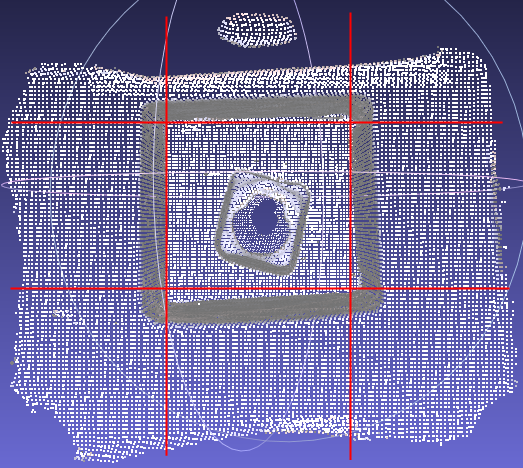
    

(a)RGBD原始数据 (b)PCL Kinfu结果 (c)VoxelHashing结果 (d)ContourCue结果 (e)Kfusion结果

图 3 十二面体为目标物的重建结果

(a)RGBD原始数据 (b)PCL Kinfu结果 (c)VoxelHashing结果 (d)ContourCue结果 (e)Kfusion结果

图 4 贝多芬头像为目标物的重建结果。注意(d)ContourCue算法重建失败，可能原因是环绕过程中重建算法多次重置

## 参考文献

[1] <https://github.com/PointCloudLibrary/pcl/tree/master/gpu/kinfu>

[2] Nießner, Matthias, et al. "Real-time 3d reconstruction at scale using voxel hashing." ACM Transactions on Graphics (TOG) 32.6 (2013): 169.

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

[4] <https://github.com/Nerei/kinfu_remake>

[5] Sturm, Jürgen, et al. "A benchmark for the evaluation of RGB-D SLAM systems." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.