

基于 Kinect v2 三维重建的研究与实现

卢灵青¹, 曾玉珠^{1*}, 李俊²

(1. 泉州师范学院, 福建 泉州 362000; 2. 中科院泉州装备制造研究所, 福建 泉州 362000)

摘要: 随着计算机视觉技术的蓬勃发展, 三维模型重建成为研究热点。选用第二代 Kinect 作为三维模型重建的数据采集外设, 获取具有深度信息的点云数据; 然后用 KinectFusion SDK 对物体进行重建, 并利用 Matlab 和 Visual C++ 对点云数据进行降低噪声处理, 并对三维点云数据进行 Delaunay 三角剖分, 建立物体表面的拓扑关系, 实现了三维物体的重建。

关键词: Kinect v2 传感器; 点云数据; 降噪处理; 三维重建

DOI:10.16184/j.cnki.comprg.2018.05.047

1 概述

近年来, 随着计算机视觉技术的蓬勃发展, 三维模型的重建已经吸引越来越多的研究者投入其中, 使之成为热点。三维模型重建技术的发展同时也给社会诸多领域带来了创新与便利, 比如电影特效和游戏制作、教学环境和实验环境的仿真、重建作战环境和仿真大型实验场景、工业上的产品设计和 3D 打印、生活上的医疗诊断和虚拟现实等等。目前, 有多种方法可实现三维模型的重建, 主要有基于体素重建、三角网格、多序列图像的拼接融合和基于激光扫描的方法等。在对现实物体获取信息的方法中, 又可划分为主动式和被动式测量方法, 其中基于三维扫描仪的主动式三维重建虽然能够精确地获取三维模型信息, 但设备太过昂贵, 数据量过大, 不利于后续处理数据, 而且使用的范围也受到限制; 基于二维图像的被动式三维重建^[1]虽然能够低成本地获取三维模型信息, 但它的特征提取、特征匹配都过于依赖光照条件和本身物体表面具有的纹理信息, 并且具有时间的限制性。

2010 年发布的 Kinect 传感器是一种可以实现三维重建的摄像头传感器设备, 虽然它在重建精度方面达不到工业制造程度的标准, 但是具备了快速而又精确、方便而又低廉、外形小巧便于携带等优点, 可以实时地进行扫描及获得物体的彩色和深度信息, 广泛应用到虚拟现实、影视游戏、产品展示等, 是一个有吸引力的替代昂贵激光扫描仪的产品。选用第二代 Kinect 作为研究三维模型重建的数据采集外设, 并利用 Matlab 软件和 Visual C++ 对点云数据进行处理, 通过软硬件的有效结合从而达到三维物体重建的目的。

2 基于 Kinect v2 三维模型的数据采集

2.1 Kinect 摄像头的标定

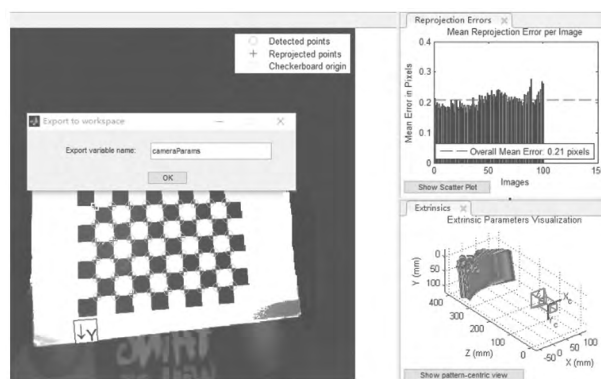


图 1 红外摄像头标定

在使用 Kinect 设备前, 首先要对其摄像头进行空间位置的标定, 以达到捕获两个摄像机得到的数据对应于空间位置之间的关系。这样做的好处有: (1) 找到 RGB 图像与深度图像之间的关系进行坐标转换; (2) 建立三维物体模型时, 可以根据彩色摄像头对模型进行纹理贴图, 达到上色呈现更真实的模型; (3) 可以利用得到的颜色阈值获取目标物体的点云数据, 从而能屏蔽除掉背景点云。使用 Matlab 自带的相机标定程序^[2] (使用张正友的棋盘标定法) 工具箱对 Kinect 进行简单的标定。具体步骤如下: 首先, 在 Matlab R2017a 应用程序中找到 Camera Calibrator; 接着将实现红外摄像头

基金项目: 福建省教育科学“十三五”规划 2016 年度课题 (FJJKCG16-366)。

作者简介: 曾玉珠 (1962-), 女, 通讯作者, 副教授, 研究方向: 大数据、商务智能、软件工程。

收稿日期: 2018-02-20

标定工作, 点击 Add Images, 然后一次导入 10~20 张的由 Kinect 采集到的图片, 过数秒后就可以看到程序会自动找到所有的探测角点, 标定结果如图 1 所示。

2.2 三维点云数据的采集

对指定目标对象进行扫描三维重建, 使用的方式有两种: (1) 手持 Kinect v2 设备, 围绕不动的目标物进行一定角度的旋转一周, 采集时序点云数据; (2) 把 Kinect v2 设备固定在实验框架上, 然后将指定的目标物体放在旋转平台上, 对旋转平台进行旋转获取信息。Kinect 设备能够同时捕获扫描的深度图和彩色 RGB 图, 帧速率约为 30 fps, 深度和颜色数据的融合构成了一个彩色点云, 它每一个帧包含了大约 30 万点。通过对连续深度图像进行注册, 可以获得更高的点密度, 同时也可以实时地创建一个室内环境的完整点云。三维点云是带有三维坐标值的数据, 将 Kinect 传感器提取到的点云三维数据保存为 “.pcd” 格式的文件, 如图 2 所示。

```

1 # .PCD v0.7 - Point Cloud Data file format
2 VERSION 0.7
3 FIELDS x y z
4 SIZE 4 4 4
5 TYPE F F F
6 COUNT 1 1 1
7 WIDTH 51320
8 HEIGHT 1
9 VIEWPOINT 0 0 0 1 0 0 0
10 POINTS 51320
11 DATA ascii
12 -0.3607603 0.37685022 0.63900006
13 -0.42193231 0.44745561 0.76000005
  
```

表示有 51320 个点的无序点云数据

图 2 pcd 点云格式

以下是获取三维点云数据的实现代码:

```

pcl::PointCloud<MyPointDataType>::Ptr cloud(new
pcl::PointCloud<MyPointDataType>);
cloud->width = 512 * 424;
cloud->height = 1;
cloud->is_dense = false;
cloud->points.resize(cloud->width * cloud->height);
  
```

2.3 降噪处理

在三维重建时需要对获取到的原始点云数据进行滤波降噪、简化数据、平滑数据等处理, 重新保存得到一个完整的包含关键特征信息的点云文件。

(1) 中值滤波

中值滤波是运用非线性信号处理的方法, 采用中值滤波方法可以对深度图像进行去除细小部分的滤波。中值滤波的实现步骤是: 1) 先找一个中心点, 确定对应

的邻域, 比如将其设置为 3×3 ; 2) 按邻域为单位, 将单位中的所有像素灰度值进行按顺序排列, 并且用一个新值 (排序后的中间值) 来取代该中心像素的灰度值, 再以邻域为一个模板, 然后将模板在图像中结合滤波算法完成上下左右移动, 使得它可以很好地平滑图像。在 Matlab 平台中调用自身的 medfilt2 函数完成中值滤波。

(2) 高斯滤波

高斯滤波是一种对整个图像实现加权平均的方法, 高斯滤波不仅能够起到平滑图像同时也起到降噪的效果。高斯滤波的实现步骤是: 1) 设置好所需的模板单位, 2) 使用该模板对图像中的所有像素点进行扫描一遍, 进而计算像素的加权平均灰度值, 3) 用这个灰度值来取代该模板的中心点像素值。在 Matlab 平台中调用自身的 fspecial 函数完成高斯滤波。

(3) 双边滤波

双边滤波器^[3]是一种运用非线性处理, 并对图像的空间邻域与像素值进行折中处理的方法。该方法不仅仅考虑了空间信息和灰度相似度, 而且降噪效果也很好, 实现了非迭代, 并且具有简单性、以及局部性等特征。在 Matlab 平台中调用自身的 bfilter2 函数完成双边滤波。

3 三维重建的实现与分析

3.1 KinectFusion 算法简介

KinectFusion 算法用于深度数据处理, 它经过匹配深度数据来构建三维场景, 是一种高效的编程和高水平并行的算法, 实时操作性高。KinectFusion^[4]算法由 4 个部分组成: (1) 深度数据进行转换。Kinect v2 的 SDK 从设备中采集到原始帧数据, 并将其转换为浮点型、单位为米的数据, 然后将上文所提方法得到的摄像机的坐标信息来优化数据, 把这些浮点型的数据与 Kinect 设备摄像头获取的点云数据进行相同方向的转化, 调用 AlignPointClouds 函数实现点云数据的对齐。(2) 计算全局摄像机的位姿, 包含摄像机所在位置与方向。在摄像机进行运动时, 经过采用交互类型的注册配准算法不停地跟踪确定其位置, 所以系统可以时刻了解现在摄像机与第一帧时摄像机的相对位置, 调用 NuiFusionAlign-PointClouds 函数实现点云配准、调用 AlignDepthToReconstruction 函数可以实时高效地对重建对象进行跟踪扫描得以构建模型。(3) 融合当前已知的位姿摄像机采集的深度图像数据, 并表示为视觉场景, 而且深度数据

是对齐每一帧达到完全融合。它不仅完成了降噪平滑算法,同时还对一些场景中的移动变化做处理工作。(4)渲染点云数据。Kinect 传感器可以根据立方体的光投影对目标物体进行重构,并且进行点云渲染呈现出带色彩的三维重建立方体。

3.2 KinectFusion SDK 重建

首先用微软提供的 KinectFusion SDK 算法对场景和目标物体进行三维重建,为减少抖动和确定摄像机坐标等,运用第二种方式将 Kinect 固定,调整最佳视角进行扫描采集数据。图 3 是以一个保温杯作为扫描对象, Kinect v2 和杯子目标对象的实际距离为一米左右,体素分辨率为所重建的结果。可以明显看出,这个距离能够较好地进行重建,融合度相对高,但模型轮廓边缘仍然存在一些不必要的、杂乱的点云数据,使得重建后的杯子表面不够平滑,影响重建的效果。在 SDK 中,还可以通过体素分辨率来改变扫描的范围,当它进行简单场景的物体目标物三维重建时,体素分辨率可以设置小一些,但是在设点的范围内无法自动分离支撑物或者目标对象所在平面;当它进行大型场景重建时,就可以设置为最大的体素分辨率。如果目标对象位置发生变化,则可通过 SDK 不断地对场景进行复位更新。

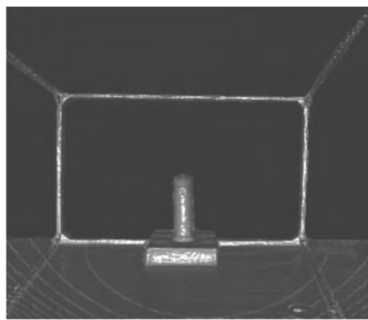


图 3 SDK 重建结果

3.3 点云可视化

三维点云是带有三维坐标值的数据,并且呈现的是离散状态,无法看出有任何明显的特征形态和拓扑关系,所以要对点云数据实现可视化。将 Kinect 设备与 PC 端连接,在 Visual Studio 2015 配置好 PCL^[5]后的编译平台下,完成表面物体三维点云的可视化。具体的步骤如下:(1)获得 Kinect 设备,调用 GetDefaultKinectSensor() 这个函数来得到预设的传感器,设置传感器的开状态;(2)运用 IKinectSensor 类中,调用一些成员函数便能方便采集到相应的资源;(3)通过

OpenReader() 获取 3 个数据帧及引用;(4)使过 AcquireLatestFrame() 能够获得最新的数据,然后把它放到其他的数据帧中,同时判断是否获取成功;(5)数据处理,提取范围内的点,同时去除离群点;(6)显示结果图,并计数点云个数。关键代码如下:

```
(1) hr = GetDefaultKinectSensor(&m_pKinectSensor);
(2) hr = m_pKinectSensor ->OpenMultiSourceFrameReader(
    FrameSourceTypes::FrameSourceTypes_Depth,
    &m_pMultiFrameReader);
(3) cloud->points[i].x = cameraX;
    cloud->points[i].y = cameraY;
    cloud->points[i].z = cameraZ;
(4) hr = m_pMultiFrameReader ->AcquireLatestFrame(&m_pMultiFrame);
(5) pcl::ConditionAnd <MyPointDataType > ::Ptr
    range_cond(new pcl::ConditionAnd<MyPointDataType >());
(6) boost::shared_ptr <pcl::visualization::PCLVisualizer> viewer(new pcl::visualization::PCLVisualizer("3D viewer"));
```

在 Matlab 软件中用如下代码: ptCloud=pcread('cow.pcd');pcshow(ptCloud);可进行读取对应的“.pcd”文件实现杯子模型对应点云显示图,效果如图 4 所示。与 KinectFusion SDK 实验对比之下,该方法可有效地剔除背景和不必要的点云数据,使得物体边缘也少了杂乱的点云,自动进行去噪处理后变得平滑,并显示读取的点云数。

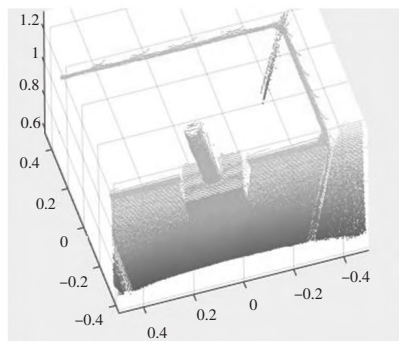


图 4 杯子的点云显示图

3.4 三角网格化

为了获得较好的三维重建的可视化效果,在获取物体的点云数据时,还需要进一步将这些离散点之间映射



相应的拓扑关系, 连接成三角网格。主要使用空间离散点的 Delaunay 三角剖分算法, 因为它是可全自动生成网格, 具有唯一性、最优性等特点。

在 Matlab 中进行三维模型的三角网格化之前, 需要对获取到的原始点云数据运用上文所提到的方法步骤进行滤波降噪、简化数据、平滑数据等处理, 重新保存可以得到一个完整的包含关键特征信息的点云文件。对完整的点云数据建立拓扑关系, 具体步骤为: 首先将“.pcd”格式转化为“.txt”格式, 然后保存到同一文件夹中, 将“.txt”格式转化为“.mat”进行下载读取, 存储为 p 数组, 再通过 Delaunay 分治算法进行三角剖分, tetr 是一个 $n \times 3$ 的矩阵, trisurf 绘制出一个由三角形构成的曲面三维网格图。杯子的三角网格细节如图 5 所示。

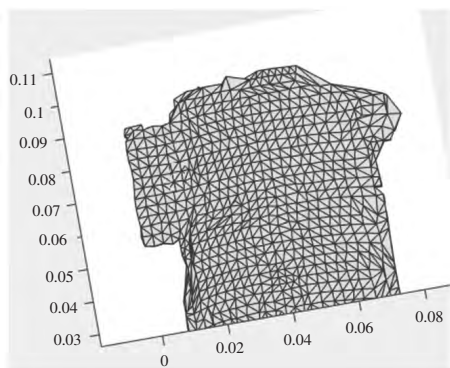


图 5 杯子的三角网格细节图

基于 Delaunay 三角剖分算法简单表面三维重建的代码如下:

(1) tic

```
tetr=delaunayn(a); %创建四面体进行三角剖分
tetr=int32(tetr); %使用整型存储数据
fprintf('三角剖分时间: %4.4f s\n',toc); %输出计时结果
```

```
(2) trisurf (tri, p (:,1) ,p (:,2) ,p (:,3) , 'facecolor',
'c','edgecolor','k') ; %曲面平滑。
```

(上接第 91 页)

实现纳入学校的综合门户等。

参考文献

- [1] 陈洵. 对高校教师工作量统计管理工作存在问题的思考和设想——以韩山师范为例 [J]. 黑龙江教育, 2011, 7: 17-19.
- [2] 牛大伟, 张晓林, 吴志明. 基于 MVC 模式的 Struts

在 Matlab 中利用 tic 和 toc 计时器来计算本实验运行结果所需要的时间, 杯子的三角剖分时间为 3.6804s, 连接时间为 0.4241s, 运行时间为 1.5311s, 总时间为 5.7570s。

4 结语

主要介绍了基于 Kinect v2 的三维模型生成所涉及到的技术与实现。首先使用第二代 Kinect 摄像头采集目标模型中具有深度信息的点云数据, 分别搭建了 Visual C++ 2015 环境和 Matlab 实验平台, 进行滤波和三角网格平滑等处理, 利用 pcd 格式的点云文件结构进行读取, 进而对无数的点云数据可视化, 最后利用 Delaunay 三角剖分对 Kinect 采集到的三维点云模型进行网格化物体的表面。对深度图像、点云数据的处理研究还不够深入, 模型三角网格化后的表面还比较粗糙, 对三维模型的平滑处理等问题, 仍需进一步研究和改进。

参考文献

- [1] 阮孟贵. 基于图像三维模型重建的研究 [D]. 清华大学, 2010.
- [2] 陈兴, 梁洁, 杜希瑞, 等. 采用两台 Kinect 相机的三维人体数字化 [J]. 电脑知识与技术, 2016, 12 (1): 192-195.
- [3] 郑立国, 刘杨, 罗江林. 基于 PCL 的人体实时 3D 重建技术的研究与实现 [J]. 电子技术与软件工程, 2015, (10): 95-97.
- [4] Jianlei Zhang, Xiaorui Wang, Qiping Zhang, Yujiao Chen, Junhui Du, Yiqun Liu. Integral imaging display for natural scene based on KinectFusion [J]. Optik - International Journal for Light and Electron Optics, 2016, 127 (2).
- [5] 郑立国, 刘杨, 罗江林. 基于 PCL 的人体实时 3D 重建技术的研究与实现 [J]. 电子技术与软件工程, 2015, (10): 95-97.

框架的教师工作量管理系统 [J]. 内江师范学院学报, 2010, 4: 46-49.

- [3] 高亮, 等. 基于 ASP.NET 技术的高校教师工作量管理系统的实现 [J]. 中原工学院学报, 2007.
- [4] 刘磊光, 郎莹, 俞玫. 高职院校教师工作量管理系统的设计与应用 [J]. 中国科教创新导刊, 2010, 34: 233-234.