

基于 PCL 的局域网内 Kinect 点云数据采集传输系统设计^{*}

陆 凯 肖 力 陈雪权

(华中科技大学自动化学院 武汉 430074)

摘 要 为了实现局域网内 Kinect 点云数据的共享,提出了基于 PCL 的 Kinect 点云数据局域网内采集传输系统设计方案,该方案利用了模块化的 C++ 模板库 PCL(点云库)中提供的通用采集接口,可以直接获取到实际坐标空间的三维信息,三维坐标信息保存为点云数据,并在流式套接字通信模式下,利用 CArchive 对象传输点云数据,使得局域网内两台计算机可以串行化的发送和接收点云数据,提高了传输点云数据的效率,实现了局域网内点云数据的远程访问。

关键词 PCL; 点云数据; 数据采集; 数据传输

中图分类号 TP393 **DOI**:10.3969/j.issn1672-9722.2013.11.017

Kinect Point Cloud Data Acquisition and Transmission System Based on the PCL within the LAN

LU Kai XIAO Li CHEN Xuequan

(School of Automation, Huazhong University of Science and Technology, Wuhan 4300074)

Abstract In order to share Kinect point cloud data within the LAN, a acquisition and transmission system of Kinect point cloud data within the LAN based on the PCL is proposed, which utilizes a modular C++ template library PCL(point cloud library) providing the generic collection interface. It can directly access to the actual three-dimensional coordinate information, three-dimensional coordinate information is saved as point cloud data, under the communication mode of socket flow, by using the CArchive object transfer point cloud data, making two computers within the LAN serializing send and receive point cloud data, thus improve the transmission efficiency of point cloud data and make the point cloud data remote accessed within the LAN.

Key Words PCL, point cloud data, data acquisition, data transmission

Class Number TP393

1 引言

PCL^[1](Point Cloud Library)是一种大型跨平台的开源 C++ 编程库,它拥有大量点云相关的高效数据结构和先进处理算法,涉及到点云获取、滤波、分割、配准、特征提取、追踪、曲面重建、可视化等。PCL 和二维信息获取和处理的函数库 OpenCV^[2]一样,在三维信息获取与处理上具有重要地位,并可支持多种操作系统平台,PCL 经由开源协议 BSD 授权,在商业和学术应用中可免费使用。

如图 1^[1]所示,对于三维点云信息获取和处理来说,PCL 是一个基于 Boost、Eigen、FLANN、VTK、CUDA、OpenNI、Qhull 第三方库的模块化的现代 C++ 模板库,可实现点云的获取、滤波、分割、配准、特征提取、识别、曲面重建、可视化等。

本文研究了如何利用 PCL 点云库,使得局域网内某一计算机可直接获取到 Kinect 点云数据,并将点云数据传输给局域网内另外一台计算机,并以 PCL 提供的 PCD 文件格式来保存传输过来的点云数据,整个采集传输系统,实现了点云数据的高效采集、高效传输,使得局域网内的计

算机可以共享点云数据。

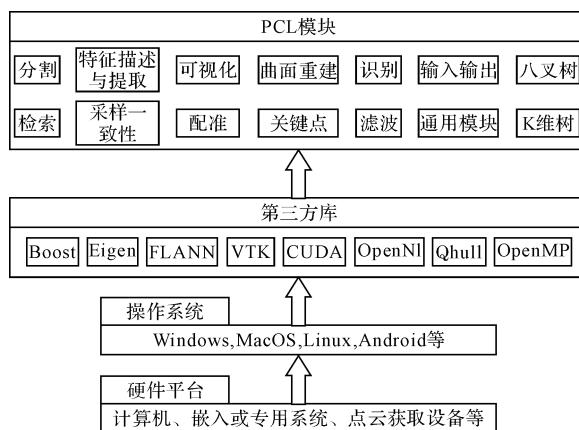


图 1 PCL 架构图

2 总体设计

系统整体框架如图 2 所示,由采集和传输两部分组成。其中采集部分是由局域网内一台计算机负责 Kinect 点云数据的采集,它将采集到的图像深度信息转换为实际空间

^{*} 收稿日期:2013 年 5 月 15 日,修回日期:2013 年 6 月 31 日

作者简介:陆凯,男,硕士研究生,研究方向:三维点云数据采集与传输。肖力,男,博士,讲师,研究方向:传感器网络与捷联惯性导航。陈雪权,男,硕士研究生,研究方向:网络控制系统与实时信号处理。

上的点的三维坐标信息;传输部分则是采集点云数据的计算机通过 TCP/IP 协议,用套接字与局域网上另外一台计算机实现点云的传输,并将传输来的点云数据保存为 PCD 文件,从而实现了局域网内的点云数据共享。

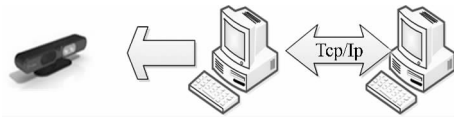


图 2 整体框架图

2.1 基于 PCL 的 Kinect 点云数据获取

2.1.1 Kinect 技术

Kinect^[3]是微软在 2010 年 6 月 14 日对 XBOX360 体感周边外设正式发布的名字。如图 3 所示,Kinect 共有 3 个摄像头,左边镜头为红外线发射器,中间的镜头是 RGB 彩色摄像机,右边的镜头为红外线 CMOS 摄像机。

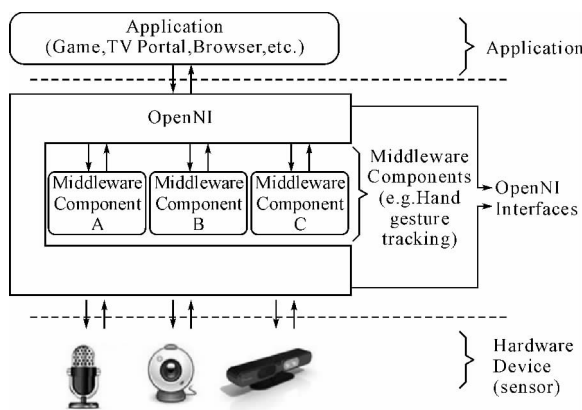


图 3 Kinect 结构图

Kinect 深度成像原理不是 TOF 技术,而是光编码 (Light Coding) 技术^[4],光编码技术关键是雷射光散斑,利用连续光(近红外线)对测量空间进行编码,经感应器读取编码的光线,交由晶片运算进行解码后,产生一张具有深度信息的图像。

2.1.2 点云获取分析

OpenNI^[5]是一个多语言、跨平台的开放自然交互框架,用来抓取 Kinect 设备中的点云数据,目前已集成到 PCL 中成为设备相关的第三方库,它已经形成了一套标准的 API,如图 4 所示,分为三层:

图 4 OpenNI 应用框架^[1]

- 1) 顶层:基于 OpenNI 接口的自然交互应用层;
- 2) 中间层:表示 OpenNI 本身,提供了传感器以及中间件模块交互的通信接口;
- 3) 底层:获取视频和音频数据的硬件设备。

从 PCL 1.0 版本开始,PCL 提供了一个通用采集接口 Grabber,可以方便地连接到不同设备及其驱动、文件格式和其他数据源。其中 OpenNI Grabber 是 PCL 集成的第一个数据获取驱动的接口^[7],十分方便的从 OpenNI 兼容设备中获取数据流。PCL 中 IO 模块提供了 OpenNI Grabber

类来捕捉数据,整个点云数据获取流程如下:

- 1) 首先创建了一个 OpenNI Grabber 接口对象;
- 2) 定义回调函数,用回调函数地址创建 boost::bind 对象,该 bind 被固定到 boost::function 对象中,boost::function 对象的函数类型是实例化的模版函数,这里使用的是 PointXYZ 的点数据类型,包含了三维 XYZ 坐标的信息;
- 3) 调用 registerCallback 函数返回一个 boost::signals2::connection 对象,则实现了对回调函数的注册;
- 4) 调用 start()开始接收点云数据;
- 5) 接收完成后,则调用 stop()停止采集。

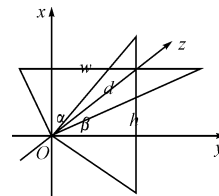


图 5 Kinect 三维坐标系 Generator 的 GetDepthMap() 函

数获得。如图 5 所示,该坐标系的坐标原点是深度图的中心投影到 Kinect 所在平面上的点,Kinect 所在平面即为 X-Y 平面,Kinect 镜头所对方向的则为 Z 轴方向。此外,由于点云数据是三维信息,需要将深度数据转换为实际坐标空间的三维数据,首先建立投影坐标空间下三维数据: x 、 y 的值为深度图中点的索引位置的值, z 值是获得的深度值。再利用深度发生器 DepthGenerator 中的坐标转化函数实现投影坐标到实际空间坐标的转换,坐标转换函数 ConvertProjectiveToRealWorld^[8] (XnUInt32 nCount, const XnPoint3D aProjective[], XnPoint3D aRealWorld[]) const,其中 nCount 表示点数,aProjective 表示投影坐标下坐标值,aRealWorld 表示实际空间下坐标值。即 aProjective[i].x = column, aProjective[i].y = row, aProjective[i].z = d (column 为点所在行数,row 为点所在列数,d 为点的距离),因此,须将投影坐标下所有 (x, y, z) 的值转为实际坐标空间 (\hat{x} , \hat{y} , \hat{z}) 值,其中坐标转换后 z 值不改变。

坐标转换函数 ConvertProjectToRealWorld() 实现方式如下:

- 1) 获取视角值:获取 Kinect 水平和垂直视角, α 是水平视角角度值, β 是垂直视角角度值;
- 2) 换算比例:换算关系 $\text{RealWorldXtoZ} = 2 \tan(\alpha/2)$, $\text{RealWorldYtoZ} = 2 \tan(\beta/2)$,即得到了深度图的宽度值 w 、高度值 h 分别与距离 d 的比值;
- 3) 求出实际的 \hat{x} , \hat{y} 值: $\text{NormalizedX} = x/640 - 0.5$,即深度图像左侧在 x 轴负方向上,深度图像右侧在 x 轴正方向上; $\text{NormalizedY} = 0.5 - y/480$,即深度图像上侧在 y 轴正方向上,深度图像下侧在 y 轴负方向上,最后算出实际坐标空间 (\hat{x} , \hat{y} , \hat{z}) 值, $\hat{x} = \text{NormalizedX} * z * \text{RealWorldXtoZ}$, $\hat{y} = \text{NormalizedY} * z * \text{RealWorldYtoZ}$, $\hat{z} = z$;

4) 实际坐标的保存: aRealWorld[i].x = \hat{x} , aRealWorld[i].y = \hat{y} , aRealWorld[i].z = \hat{z} ,即将计算出的三维数据保存于实际坐标空间 aRealWorld[] 中。

2.2 点云数据局域网内传输

2.2.1 套接字

套接字^[9]是一个通信端点,用来在网络上不同主机之

间发送或接收数据包的对象。套接字可分为流式套接字和数据报套接字,流式套接字使用 TCP 协议使得数据流可以准确无误按顺序无重复地被送达;数据报套使用 UDP 协议,数据不能可靠地被送达,易出现丢包或重复。

CSocket 是 MFC 在 CAsyncSocket^[10] 基础上派生的一个同步阻塞 Socket 的封装类,它支持流方式和数据报方式,但更多的是在流方式下进行数据通信。在流方式的通信模式下,CSocket 类与 CSocketFile 类、CArchive 类一起使用,CArchive 类对象缓冲区内可存取数据,并通过 CSocketFile 类对象与 CSocket 类对象建立联系,从而使得数据可以在网络上串行化的发送和接收。数据传输通信模式^[11]如图 6 所示。

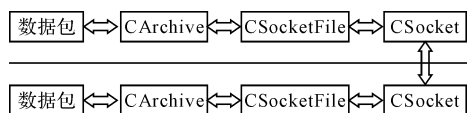


图 6 CSocket 通信模式

2.2.2 三维点云数据传输分析

Kinect 的可视区域的分辨率为 640×480 , 因此采集一次即可得到 640×480 个点, PCL 提供多种可用的 PointT 类型, 这里使用包含三维 XYZ 坐标信息的 PointXYZ 数据类型, 可用 `points[i].data[0]` 或 `point[i].x` 保存点的 x 的坐标值, `points[i].data[1]` 或 `point[i].y` 保存点的 y 的坐标值, `points[i].data[2]` 或 `point[i].z` 保存点的 z 的坐标值, PCL 提供一种新的文件格式即 PCD^[12] 文件格式, 用于存储点云数据, 每一个 PCD 文件都包含一个文件头, 它确定和声明文件中存储的点云数据的点云特性, 且该文件头必须用 ASCII 码来编码。为了使局域网内两台电脑共享 Kinect 的点云数据, 可通过 Socket 网络编程实现两台电脑之间的通信。这里使用流式套接字, 基于 C/S 模型采用 MFC 下的 CSocket 类编程实现了点云数据的串行化传输, 具体做法如图 7、8 所示。

整个传输过程中使用了三个辅助类, 即 CPoint、CListenSocket 和 CClientSocket 类。CPoint 类是一个串行化的点云数据类, 通过重载 CPoint 类的构造函数 `CPoint(float point[i].x, float point[i].y, float point[i].z)` 更新点云数据, 三维数据的传输也只需通过重载函数 `Serialize(CArchive &ar)` 中流操作符对缓冲区进行存取就可以实现; CListenSocket 类是接收端负责监听管理的套接字类, 调用 `listen()` 函数监听发送端的连接请求, 重载基类 CSocket 的 `OnAccept` 函数对发送方的连接请求作出响应, 当该套接字接收到连接请求时则调用 `accept()` 函数来建立发送端与接收端之间的连接, 并返回一个新的 CClientSocket 类对象, 用于接收点云数据; CClientSocket 类是主要负责传输串行化点云数据的类, 创建该类发送端的对象, 调用 `connect()` 函数用于向接收端请求建立连接, 同时定义一个连接到 CSocket 对象的 CSocketFile 类型的指针 `m_File` 和分别用于发送端发送、接收端接受的 CArchive 类型的指针 `m_ArchiveOut`, `m_ArchiveIn`, 该类提供了成员函数 `SendPcd()`、`ReceivePcd()` 用于点云数据的发送和接收, 接收端数据的接收由重载的 `OnReceive` 函数实现, 每当有点云数据在 CArchive 对象提供的缓冲区中, CClientSocket 类对象自动调用 `OnReceive` 函数, 逐个写入 PCD 文件中。

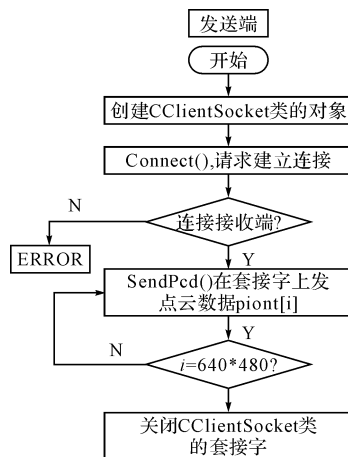


图 7 发送端点云数据传输流程

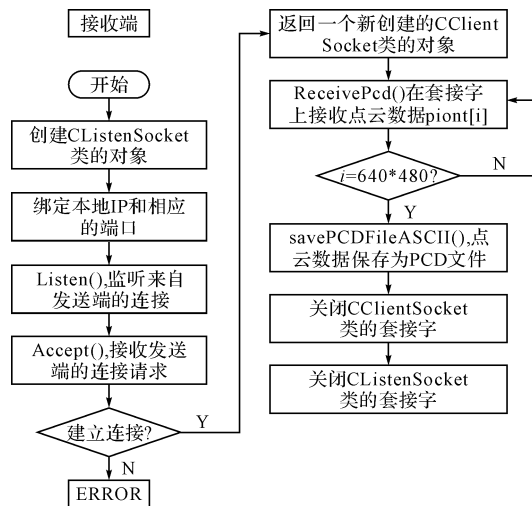


图 8 接收端点云数据传输流程

3 结语

本文研究了局域网内点云数据的采集和传输, 利用 PCL 提供的通用采集接口, 可以直接将 Kinect 采集到的深度图像的像素值直接转换为实际坐标空间的三维信息, 同时利用在流方式的通信模式下, CSocketFile 类配合 CArchive 类使用, 使得局域网内两台计算机通过套接字可以串行化的发送和接收点云数据, 提高了点云传输的效率, 实现了局域网内的计算机可以远程访问点云数据。

参考文献

- [1] 朱德海. 点云库 PCL 学习教程[M]. 北京: 北京航空航天大学出版社, 2012.
ZHU Dehai. The learning tutorial of Point cloud library PCL [M]. Beijing: Beijing University of Aeronautics and Astronautics press, 2012.
- [2] 张晓宇, 彭四伟. 基于 OpenCV 的运动目标识别算法与实现[J]. 现代电子技术, 2009, 32(22): 99-101.
ZHANG Xiaoyu, PENG Siwei. Recognition algorithm and Realization of moving target Based on OpenCV[J]. Modern Electronic Technology, 2009, 32(22): 99-101.
- [3] 黄康泉, 陈壁金, 郑博, 等. Kinect 在视频会议系统中的应用[J]. 广西大学学报(自然科学版), 2011, 36(z1): 308-313.
HUANG Kangquan, CHEN Bijin, ZHENG Bo, et al. Kinect

- in video conference systems[J]. Journal of Guangxi University (NATURAL SCIENCE EDITION), 2011, 36(z1): 308-313.
- [4] 李红波, 丁林建, 冉光勇. 基于 Kinect 深度图像的人体识别分析[J]. 数字通信, 2012(4): 21-25.
LI Hongbo, DING Linjian, RAN Guangyong. Human analysis recognition based on Kinect depth image[J]. Digital Communication, 2012(4): 21-25.
- [5] 黄佳. 基于 OPENCV 的计算机视觉技术研究[D]. 上海: 华东理工大学, 2012: 46-48.
HIANG Jia. The research of Computer visual technology based on OPENCV[D]. Shanghai: East China University of Science, 2012: 46-48.
- [6] 楚发. 基于体感的三维配准系统的设计与实现[D]. 上海: 上海交通大学, 2011: 21-22.
CHU Fa. Design and Realization of 3D registration system based on Body Feeling[D]. Shanghai: Shanghai Jiaotong University, 2011: 21-22.
- [7] PCL 中的 OpenNI 点云获取框架(1)[EB/OL]. <http://www.pclcn.org/study/shownews.php?lang=cn&id=87>.
OpenNI point cloud acquisition framework in PCL(1) [EB/OL]. <http://www.pclcn.org/study/shownews.php?lang=cn&id=87>.
- [8] heresy. 透过 OpenNI 建立 Kinect 3D 点云[EB/OL]. http://viml.nhc.org.tw/blog/paper_info.php?CLASS_ID=1&SUB_ID=1&PAPER_ID=217.
- heresy. The establishment of Kinect 3D point cloud through OpenNI[EB/OL]. http://viml.nhc.org.tw/blog/paper_info.php?CLASS_ID=1&SUB_ID=1&PAPER_ID=217.
- [9] 刘丹华, 黄道君. 利用套接字开发网络通信程序[J]. 微机发展, 2003, 13(1): 33-35.
LIU Danhua, HUANG Daojun. Developing network communication program with Socket[J]. Microcomputer Development, 2003, 13(1): 33-35.
- [10] 李天文, 董祥军, 张洁. 用 MFC CAsyncSocket 类实现计算机间的通信[J]. 山东轻工业学院学报, 2001, 15(1): 21-24.
LI Tianwen, DONG Xiangjun, ZHANG Jie. Using MFC CAsyncSocket class to implement the communication between computers[J]. Journal of Shandong light industry academy, 2001, 15(1): 21-24.
- [11] 罗艳丽, 秦开宇, 金燕华. VC++6.0 环境下三种方法实现计算机通信的讨论[J]. 电子设计工程, 2012, 20(18): 145-147.
LUO Yanli, QIN Kaiyu, JIN Yanhua. Three kinds of methods to achieve the computer communication under VC++6.0 environment [J]. Electronic Design Engineering, 2012, 20(18): 145-147.
- [12] PCD 文件格式[EB/OL]. http://pointclouds.org/documentation/tutorials/pcd_file_format.php#pcd-file-format.
The PCD(Point Cloud Data) file format[EB/OL]. http://pointclouds.org/documentation/tutorials/pcd_file_format.php#pcd-file-format.

~~~~~

(上接第 1767 页)

- [4] Champandard A. J. Behavior trees for next generation game AI [C]//Proceedings of the Game Developers Conference, 2007.
- [5] Orkin J. 2004. Applying goal-oriented action planning to games [M]. AI Game Programming Wisdom. Hingham: Charles River Media, 2004: 217-228.
- [6] Champandard A. Verweij, T. Straatman, R. The AI for Killzone 2's multiplayer bots[C]//Proceedings of Game Developers Conference(2009), Paris.
- [7] Tan C T, Cheng H. An automated model-based adaptive architecture in modern games[C]//Proceedings of The Artificial Intelligence and Interactive Digital Entertainment Conference (2010). AAAI Press, 2010.
- [8] 尹全军, 冯磊, 胡记文, 等. 基于游戏引擎的人类行为建模与仿真[J]. 系统仿真学报, 2009, 21(3).  
YIN Quanjun, FENG Lei, HU Jiwen, et al. Game Engine Based Simulation for Modeling Human Behavior[J]. Journal of System Simulation, 2009, 21(3).
- [9] Ballinger C A, Turner D A, Conception A I. Artificial Intelligence Design in a Multiplayer Online Role Playing Game[C]//Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. IEEE, 2011: 816-821.
- [10] Auslander B, Lee-Urban S, Hogg C, et al. Recognizing the enemy: Combining reinforcement learning with strategy selection using case-based reasoning[M]. Advances in Case-Based Reasoning. Springer Berlin Heidelberg, 2008: 59-73.
- [11] Gillespie K, Karneeb J, Lee-Urban S, et al. Imitating inscrutable enemies: Learning from stochastic policy observation, retrieval and reuse[M]. Case-Based Reasoning. Research and Development. Springer Berlin Heidelberg, 2010: 126-140.
- [12] 王振宇. 计算机游戏中智能角色行为的研究与实现[D]. 长沙: 湖南师范大学, 2010.  
WANG Zhenyu. Research and implementation of intelligent role's behavior in computer games [D]. Changsha: Hunan Normal University, 2010.
- [13] M. Benda, V. Jagannathan, R. Dodhiawala. On optimal co-operation of knowledge sources-an empirical investigation[R]. Washington: Boeing Advanced Technology Center, 1986.
- [14] Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective[J]. Autonomous Robots, 2000, 8(3): 345-383.
- [15] Barrett S, Stone P, Kraus S. Empirical evaluation of ad hoc teamwork in the pursuit domain[C]//The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2. International Foundation for Autonomous Agents and Multiagent Systems, 2011: 567-574.