

华中科技大学

本科生毕业设计[论文]

基于多个 TOF 相机的 3D 场景重建

院 系 电子信息与通信学院

专业班级 信息类数理提高班 1301

姓 名 曾 晨

学 号 U201313736

指导教师 喻 莉 教授

2017 年 5 月 17 日

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名: 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密口，在 年解密后适用本授权书

2、不保密口。

(请在以上相应方框内打“√”)

作者签名: 年 月 日

导师签名: 年 月 日

摘 要

过去一些年,深度相机已吸引了足够的目光。通过提供三维信息,深度相机可以使得很多原本的计算机视觉等任务相对容易。包括物体跟踪,物体检测,人类行为分析,手势分析,室内 3D 建模等。

对于 3D 场景重建,既可以用单个深度相机实现,也可以用多个深度相机实现。相比于运用单个深度相机重建 3D 场景,多个深度相机的重建有一些优点,它可以使得表面轮廓更加紧密平滑,也可以使得重建覆盖范围更大,此外还可以补偿单个相机采集频率不够的问题。

本文通过采用多个 Kinect v2(内含 TOF 相机)采集的图像和点云数据作为素材,着重讨论了多个 TOF 相机的点云配准问题、多相机干扰问题和多路径干扰问题。用张正友标定和 ICP 算法实现了点云配准,用多帧求期望解决了多相机干扰问题,用 SRA 算法解决了多路径干扰问题,实验对象是对于 Kinect 采集的实际场景和微软官方提供的数据进行测试,取得了很好的效果。

本课题对于三维场景重建有着实际意义。

关键词: TOF 相机; 3D 场景重建; 多路径干扰; 多相机干扰

Abstract

In the past few years, depth cameras have attracted enough attention. They make some tasks of computer vision become much easier by providing three dimension information. And these tasks include object tracking, proposal detection, behavior analysis, gesture analysis and reconstructing 3D scene.

As for problem of 3D scene reconstruction, taking data by using single camera or multiple cameras are both acceptable. Comparing with using single camera, using multiple cameras has its advantages. It can make the surface more smooth and make the range of taking data to be bigger. Also, it can compensate for the problem that the frequency of single camera is not enough.

In this paper, we using the data and point cloud caught by ToF camera set in Kinect-V2. And this paper concentrate on the analysis of registration of 3D point cloud, cancellation of noise made by multiple cameras and made by multiple paths. Here we use Zhengyou Zhang camera calibration method and ICP algorithm to implement calibration. And we calculate expect for multiple frames to resolve noise led by multiple cameras and use SRA algorithm to solve multiple paths problem. Based on the data we captured and the data provided by Microsoft, the results show our methods work well.

This topic is important for 3D scene reconstruction.

Key Words: TOF Camera; Multiple Cameras Interference; Multiple Paths Interference; Reconstructing 3D Scene

目 录

1.绪论	1
1.1. 课题意义	1
1.1.1. 3D 场景重建的意义	1
1.1.2. 用深度相机进行 3D 场景重建的意义	1
1.1.3. 用多个 ToF 相机进行 3D 场景重建的意义	2
1.2. 国内外研究现状及发展趋势	2
2. 3D 场景信息的获取	5
2.1. 3D 场景获取设备	5
2.2. 单帧数据的获取和实验结果	6
2.3. 多帧数据的获取和实验结果	8
2.4. 多相机的数据获取和实验结果	9
3.多个 TOF 相机的 3D 场景重建	12
3.1. 多个 ToF 相机的点云获取和预处理	12
3.2. 多个 ToF 相机的点云配准	13
3.2.1. ICP 算法	13
3.2.2. ICP 算法实验结果	16
3.3. 结合相机标定的多相机 3D 场景重建	17
3.3.1. 张正友标定法	17
3.3.2. 结合相机标定的多相机 3D 场景重建结果	17
3.4. 多个 ToF 相机的点云融合	19
4.多个 TOF 相机干扰消除	20
4.1. 多个 TOF 相机干扰问题	20
4.2. 多 TOF 相机干扰消除	21
4.3. 实验和结果	23
5. TOF 相机多路径干扰消除	25
5.1. 多路径干扰问题	25
5.1.1. ToF 测量	25
5.1.2. 多路径	25
5.1.3. 反射光的特性	26
5.2. 多路径干扰消除	27
5.2.1. SRA(Sparse Reflections Analysis)算法	27
5.3. 快速计算	29
5.4. 实验和结果	29
6.结论	31

1. 绪论

1.1. 课题意义

1.1.1. 3D 场景重建的意义

当今, 3D 重建技术已经是人工智能的重要研究课题, 该问题的研究成果, 可直接应用于机器导航、物体识别、医学仪器、精密工业测量、虚拟现实以及军事等方面的应用。物体的 3D 重建工作是计算机视觉、计算机图形学(CG)、计算机辅助几何设计(CAGD)、科学计算和虚拟现实、医学图像处理、计算机动画、数字媒体创作等领域的核心技术和共性科学问题, 解决 3D 场景重建的问题对于这些领域有重要意义。

为满足各类应用对三维信息于真实世界中的需要, 各领域的科研人员都在致力于新型获取手段的探索, 和现有的重建技术改进。如计算机视觉中需要获取目标的三维特征从而实现自动目标识别和分类等任务, 机器人学中需通过借助环境信息的获取实现自动导航, 而测绘学领域需要获取高精度的地形信息以及建筑物、植被、道路等非地形信息, 而考古学需要获取文物及遗址的三维模型从而实现对它们的保护和修理等。来自多个领域的需求促使当前各种三维重建技术层出不穷, 精度不断提高, 速率趋于实时, 成本也不断降低, 也不断促进 3D 重建技术的广泛应用。尤其是促进其在军事医疗等领域的深度应用。

除了上述领域外, 近年来 3D 重建技术成为我国医学等别的研究领域的新热点, 3D 重建技术所用于医学体数据研究, 2D 重建及其变换的应用科研上很多已达到国际先进水平。经 3D 重建技术, 在 CT、MRI 或 DSA 等影像指导下, 再经计算机精确计算和仿真校验之后, 最后利用新型高精度脑立体定向手术仪进行各种脑部手术脑血管疾病、脑血管造影 3D 重建成像等技术在脑动脉瘤、颅脑损伤诊断和治疗中所用。另外, 在医疗领域, 还有一些虚拟现实和 3D 重建运用的结合, 比如基于虚拟现实的人体颌面部和女性盆腔的 3D 重建已经有一些成果, 它们能够帮助医疗人员更好的观察病人病理。

1.1.2. 用深度相机进行 3D 场景重建的意义

过去一些年,深度相机已经吸引了足够的目光。通过提供三维信息,深度相机使得很多的计算机视觉任务相对容易。包括物体跟踪,物体检测,人类行为分析,手势分析,室内 3D 建模。

对于 3D 场景重建,既可以用单个深度相机实现,也可以用多个深度相机实现。相比于运用单个深度相机重建 3D 场景,多个深度相机的重建有一些优点,它可以使得表面轮廓更加紧密平滑,也可以使得重建覆盖范围更大,此外还可以补偿单个相机采集频率不够的问题。总结而言,运用多个深度相机进行 3D 场景重建,提高了重建的精度、增加了重建的灵活性、拓宽了重建的适用范围。

1.1.3. 用多个 ToF 相机进行 3D 场景重建的意义

在深度相机技术中,ToF 相机最近变得很火。TOF 是 Time of flight 的简写,直译为飞行时间的意思,即是通过 TOF 相机上的发射装置给目标发送连续的光脉冲,然后再用传感器接收从物体返回的光,通过探测光脉冲在飞行(往返)过程中所需时间来得到距离目标物的距离[1]。

在现阶段,我们一般使用 ToF 相机,对于 ToF 的优势:与传统的立体相机或三角测量系统比,TOF 相机体积非常小巧,跟普通相机大小相近,非常适合一些需要轻便和小巧的相机的场合。并且 TOF 相机能够实时快速的计算深度信息,达到几十甚至到 100fps。与之对比,双目立体相机则需要用到复杂的相关性算法,处理速度比较慢。TOF 相机的深度计算不会受物体表面灰度和特征影响,可以非常准确的进行三维探测。与之对比,双目立体相机则需要目标具有良好的特征变化,否则将无法进行深度计算。TOF 的深度计算精度完全不随距离改变而改变,基本可以稳定在厘米级别,这对于一些大范围运动的应用场合是非常有意义的。

1.2. 国内外研究现状及发展趋势

现阶段,多种三维测量设备的迅速发展,为三维场景重构提供了更多的实现手段,三维场景重构的研究经历了从最初对中小型物体的模型构建,再到后来对大型建筑室外场景的三维重构,重构技术一直影响着人类的生活。虽然重构技术在科研领域已发展了许久,但是受困于受传感器方面的限制,目前还完全没有达到普及的程度,在国内的研究,甚至是国外的研究,都不多。

要探究 3D 重建,首先需要了解 3D 场景获取的手段,双目视觉法是用两个彩色相机来获取场景的 3D 信息,激光逐点扫描法是从逐行逐列进行扫描从而获得整个场景的 3D 信息,结构光深度相机则是通过投射模板到场景然后对接收到的场景图像和模版之间进行匹配从而获得深度,以及 TOF 深度相机,直接投射近红外光到场景中,在接收端从每个像素点接收到的信号来同时获取整个场景的 3D 信息。

Kinect 的出现,特别是 2 代的出现,激发了更多的研究者投身到三维技术的科研中。Kinect1 代运用的结构光,而 Kinect2 代中正是使用了 ToF 相机,由此 ToF 相机相关的实验,较之前相比成本大大降低。随着 Kinect 的逐渐推广,三维重构技术也开始走进大众视野。真实场景现在可以依托于多种设备,如激光测距、立体相机、单目相机及 Kinect 等手段来实现多种重构表现形式。

对于 Kinect,它是一种简洁低价的可以同时获得彩色信息和深度信息的传感器设备,可将日常生活中的场景或物体数字化,连接了虚拟和真实世界。基于 Kinect 的三维场景重构及地图构建等技术也日趋成熟起来。Henry 等人利用移动的 Kinect 采集室内场景数据,联合形状和表面信息进行数据集之间的配准,产生室内场景的三维模型。建立了室内的三维场景地图并完成了 Kinect 的定位,解决了 SLAM 问题。最后全局模型利用面元表现出来。Endres 等人基于手持 Kinect,从彩色图像提取视觉关键点,在三维空间中构建可视化的 SLAM。随着 Kinect 的推广使用,相信会出现更多的三维应用[5][6]。

ToF 刚出现时研究主要集中在去噪,去多周期混叠,而现在,ToF 相机的研究重点在提高 ToF 相机分辨率,ToF 相机成像质量,重建质量和速度上面。

对于 3D 重建,由于在场景过于庞大的情况下,单个相机重建 3D 场景有困难,并且有时候单个相机采集频率不够,所以有时候,3D 场景重建需要多个相机同时工作,这样便克服了单个相机采集频率及准确率不足的问题。但是,ToF 相机的主要问题是其会遇到多相机干扰和多路径干扰。光源发射出的光线会在每个场景点上发生漫反射,反射后会有多条路径,每个像素点可能会接收多个场景点反射射出的光线,这就是多路径干扰,甚至是也会接收别的相机发出的光线,这就是多相机干扰。由此可见,多相机和多路径干扰问题会导致测量错误,这种测量错误不像普通的噪声,其错误很大,并且导致深度估计不准确。所以对于 ToF 相机来说,移除多相机干扰和多路径干扰是一个非常重要的问题。

解决多相机干扰问题已经成为多相机 3D 场景重建中一个热门话题。Young Min Kim 等人于 Stanford 已经构造了整个的多相机测量系统，而更多关于解决多相机干扰问题的算法也在不断提出。[7]

为了移除多相机干扰，可以使用时域上的分时使用。这种情况下，每个 ToF 相机在不同时间段上工作，进而避免相互之间干扰。这种方法需要相机之间时钟同步，十分不便。另一种方法是在频域上分开，使得不同相机的工作时调制频率不同，这样相机之间也就不会有干扰。这种方法需要相机有自己的调制频率，并且这种调制频率将会决定最大检测范围和测量精度。但是商用相机的调制频率很有限，所以不同相机之间还是很难同时工作，使得这个方法难以实施。伪随机编码法也是一种重要方法，其可以使不同相机之间的工作频率不相关，进而避免相机之间干扰，但是这种方法需要拥有调幅连续波技术的商用相机。所以，现在的方法在不同程度上，这些方法都还存在一些问题与不足。

为了移除多路径干扰，之前的工作通过增加传感器移除了多路径干扰，而最近更多的工作尝试着直接通过测量的传感器本身移除多路径干扰。但在不同程度上，这些方法都还存在一些问题与不足。这些方法有的不接近实时，对于每个框架都需要 1-10 分钟处理时间。而有的方法虽然实时，但是所需的调制频率数量上需要较多。例如，Bhandari 等人 2014 年刊登在 ECCV 上的作品，效果好速度快，但是需要很多调制频率，多达 77 个，实际上，这么多的调制频率在很多场景下是不可能实现的。

相较于之前的方法，Daniel Dreedman 的工作 SRA 算法是基于 L1 优化，在实际运用中效果更好。L1 的缺点在于其很慢，但是本算法基于 DUT 快查表进行了创新优化，其允许这个方法既具备 L1 的准确性，又减小了 L1 的速度劣势。这也是本次毕业设计主要参考的算法。

2. 3D 场景信息的获取

2.1. 3D 场景获取设备

我们这里采用的获取 3D 场景数据的设备是 Kinect v2。

Kinect for Xbox one, 简称 Kinect v2, 是由微软开发, 应用于 Xbox one 主机的周边设备。Kinect 2 代有三个镜头, 最边上的镜头是 RGB 彩色摄影机, 用来采集彩色图像。中间镜头是 Depth 传感器, 采用的是 TOF 的方式, 通过从投射的红外线反射后返回的时间来取得 Depth 信息。Depth 传感器看不到外观, 不过 Color Camera 旁边是红外线 Camera 和投射脉冲变调红外线的 Porjector, 用来采集深度数据(场景中物体到摄像头的距离)。彩色摄像头最大支持 1280*960 分辨率成像, 红外摄像头最大支持 512*424 成像。Kinect 也内建阵列式麦克风, 由四个麦克风同时收音, 比对后消除杂音, 并通过其采集声音进行语音识别和声源定位。

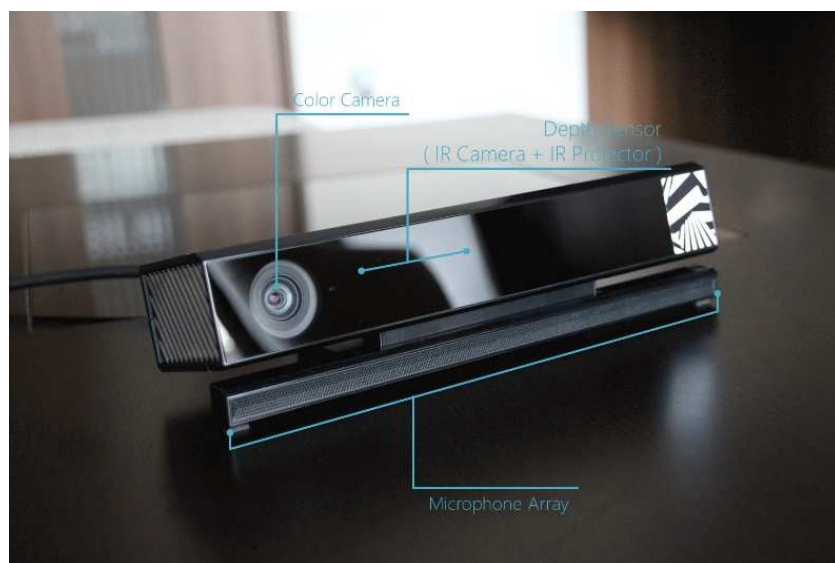


图 2-1 Kinect2 代实物图

Kinect 的官方 SDK 提供了音频支持、调整倾角的转动电机、在全身跟踪骨骼跟踪方面: 非标准姿势检测, 头部、手、脚、锁骨检测以及关节遮挡等细节上的处理更为细致(但精度是否更高还不能确定)。此外, 支持多传感器(多台 Kinect)。Kinect for Windows SDK 可让使用 C++、C# 或 VisualBasiC 语言搭配 MicrosoftVisualStudio 工具的程序设计师轻易开发使用。

但是由于微软对非商业使用的限制, 很多功能没有提供。因此在实际使用中,

可以根据需要采用非官方 OpenNI/NITE 等开源驱动。

这里我们采用 Matlab 和 Kinect Studio SDK 获取数据。在 Matlab 的 SDK 安装中,可以找到对 Kinect v2 的支持,而该工具非常容易帮助我们获取 Kinect 的点云数据,并且输出点云数据在之后的实验中可以容易被 matlab 调用。而 Kinect Studio SDK 是微软官方的开发 Kinect 工具,可以通过不同模式调用采集彩色图、深度图、红外图,以及.xef 格式的视频。总之,结合 Matlab 和 Kinect Studio SDK 获取 Kinect v2 的数据使得我们之后的实验有足够和易处理的数据。

2.2. 单帧数据的获取和实验结果

在讲解单帧数据获取之前,我们需要知道,虽然彩色相机采集的数据跟深度相机采集图像尺寸不一样,但是经过 matlab 内置函数剪裁,点云数据中二者的尺寸是一样的,点云会在 matlab 中很好的呈现。

(1) 点云数据的获取

如前文所说,matlab 中的 SDK 有对 Kinect 2v 的支持,安装后可以轻易调用 Kinect 摄像头采集数据,而输出点集格式为.mat,具体代码在如下 Github 链接:<https://github.com/ZengChen94/3DReconstructionByKinect2V/blob/master/src/catchScene.m>,这里面的核心代码是 `ptCloud = pcfromkinect(depthDevice, depthImage, colorImage)`,即是采集实时深度数据和彩色数据,并且以点云形式存储,点云在 matlab 下显示如下:

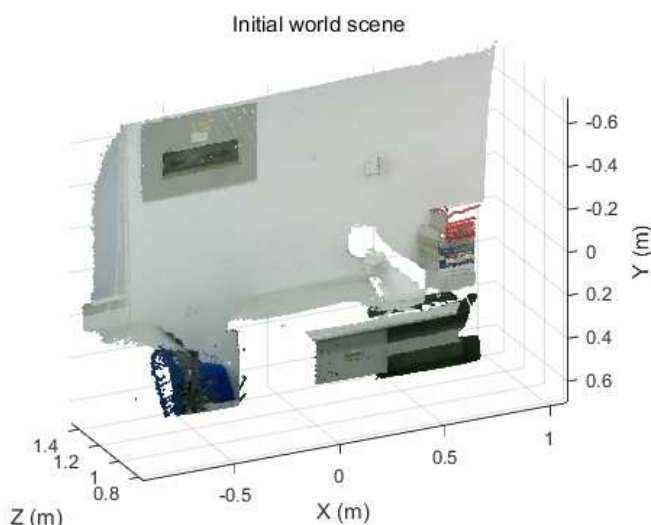


图 2-2 点云数据

(2) 彩色图像的获取

打开 SDK Browser(Kinect for Windows) v2.0, 进入模式 Color Basics-WPF, 在链接了 Kinect 的情况下, 进入后可以观测到彩色相机拍摄的图像, 点击右下角截取图片, 图片便保存在文档下的图片文件夹, 效果如下所示:



图 2-3 彩色图

(3) 深度图像的获取

打开 SDK Browser(Kinect for Windows) v2.0, 进入模式 Depth Basics-WPF, 在链接了 Kinect 的情况下, 进入后可以观测到事实的深度相机拍摄的图像, 点击右下角截取图片, 图片便保存在文档下的图片文件夹, 效果如下所示:



图 2-4 深度图

(4) 强度图像（红外图像）的获取

打开 SDK Browser(Kinect for Windows) v2.0, 进入模式 Infrared Basics-WPF, 在链接了 Kinect 的情况下, 进入后可以观测到深度相机拍摄的图像, 点击右下角截取图片, 图片便保存在文档下的图片文件夹, 效果如下所示:

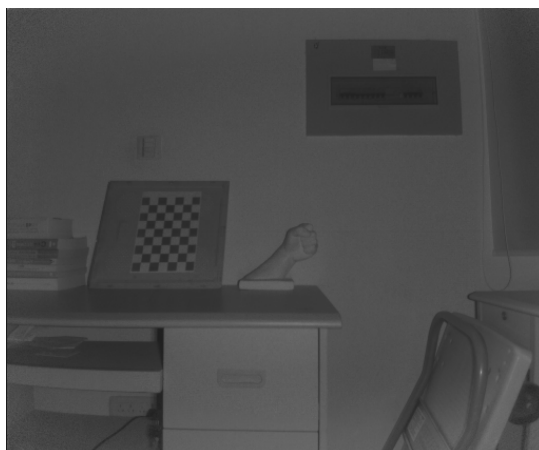


图 2-5 红外图

2.3. 多帧数据的获取和实验结果

对于研究多相机干扰问题时，我们需要连续的多帧深度图数据，由于 Kinect Studio SDK 并没有提供对应的采集连续帧图像的功能，我们于是编写 C#程序自行截取视频中的连续帧。

编写 C#程序，读取包含多相机干扰环境下的.xef 视频，增添线程模拟 Kinect 采集数据，实时将数据流以帧为单位存储为.png 格式深度数据。之所以要这么做是由于 Microsoft 并没有对 Kinect 中提取视频提供直接接口，而需要第三方自行编写 C#利用 Kinect 的某些接口模拟采集数据并不断截取每一帧屏幕。

该 C#项目下有 3 个文件，Program.cs / Player.cs / Extractor.cs。其中 Program.cs 调用了 Player.cs / Extractor.cs，调用时需要设置参数运行程序，传入输入文件和输出文件夹，具体的调用方法为： `KinectFrameExtractor.exe C:/Users/itsy/Desktop/test C:/Users/itsy/Desktop/test/20170223_074431_00.xef`。Player.cs 实现模拟.xef 文件的播放，Extractor.cs 实现对于模拟播放的视频进行不断截屏。总体而言采用多线程，一个线程模拟播放，一个线程不断截图。

最后便能把.xef 格式的视频数据分解成连续的单帧的深度图数据，同理对彩色图等数据也能这么处理，分解后的文件夹，示意图如下：

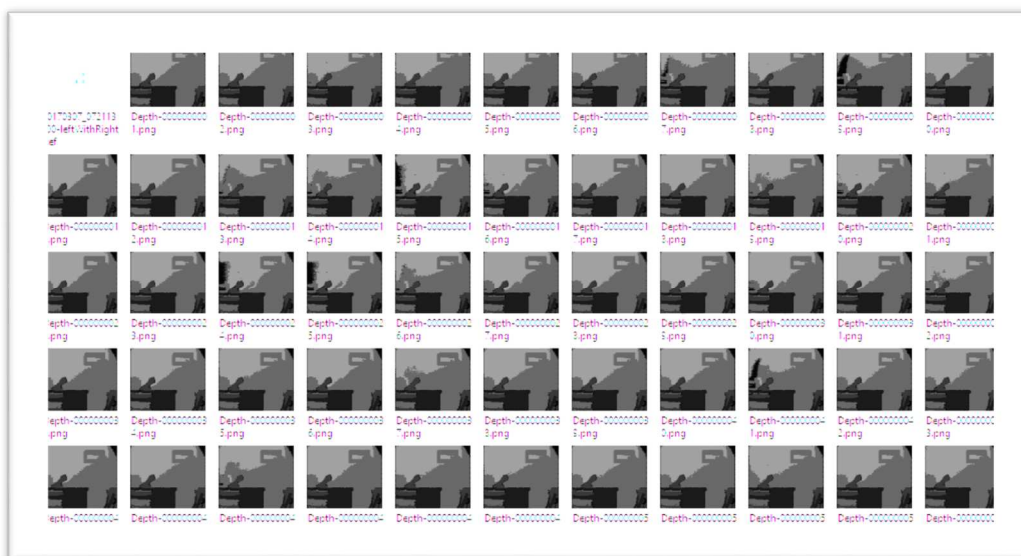


图 2-6 分解 .xef 格式视频为连续多帧深度图

这里之所以大部分图像显示黑色是因为缩略图分辨率和图像实际每个像素点数值所致，点开看其实每张图都是单帧的深度图数据，每个像素点的数据大小都代表实际深度大小，以米为单位。

2.4. 多相机的数据获取和实验结果

为了实现多相机之间的串扰，我们在实验中实际采用多个相机同时采集数据，这样相机之间便会有串扰，采集数据的实验示意图如下，如图我们可以看到三台 Kinect2 代相机在同一高度和水平面上对同一场景进行不同角度拍摄。

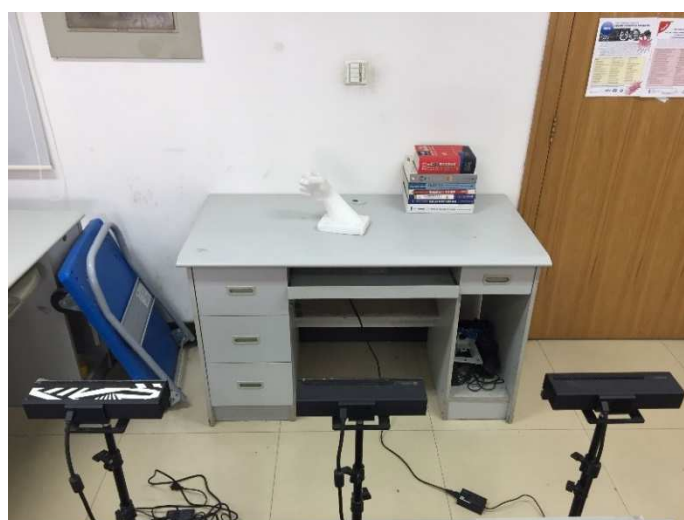


图 2-7 采集数据场景

下面展示采集的左/中/右相机三份点云集数据：

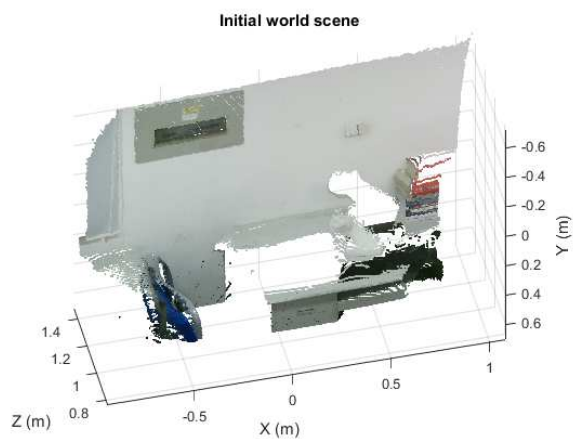


图 2-8 左相机点云数据

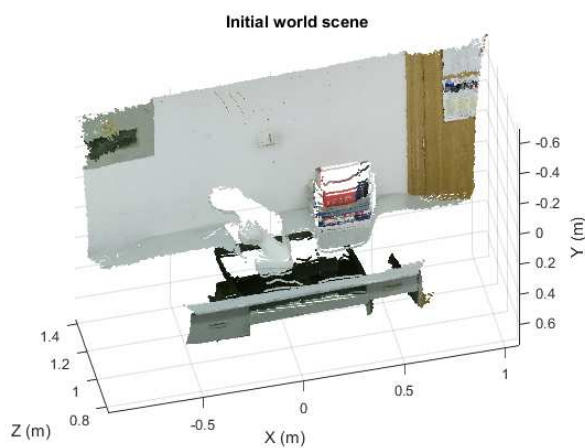


图 2-9 中相机点云数据

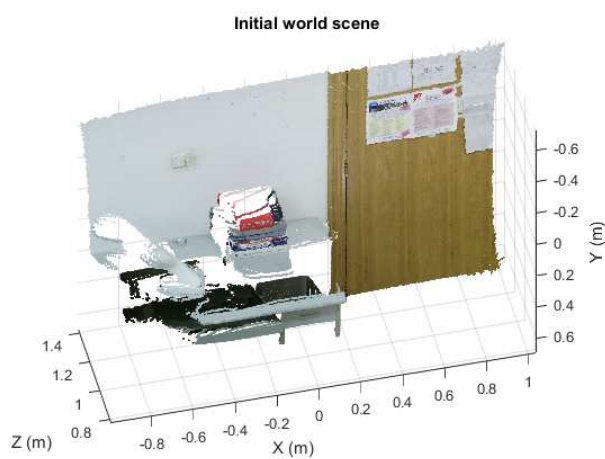


图 2-10 右相机点云数据

由上面三张图可以看到三个点云数据在同一水平线上的不同位置上, 而每两个相机之间的距离相同。除了采集三份点云数据, 还采集了三份彩色图, 三份深度图,

三份强度图。由于篇幅所限，这里不一一展示。

3. 多个 TOF 相机的 3D 场景重建

3D 重建是将多个 ToF 相机采集数据的深度图，通过滤波器移除噪声并且下采样后，将处理后的点集，配准到一个坐标系下，并且融合，最后形成 3D 模型实现重建的过程，整个过程如下图所示。

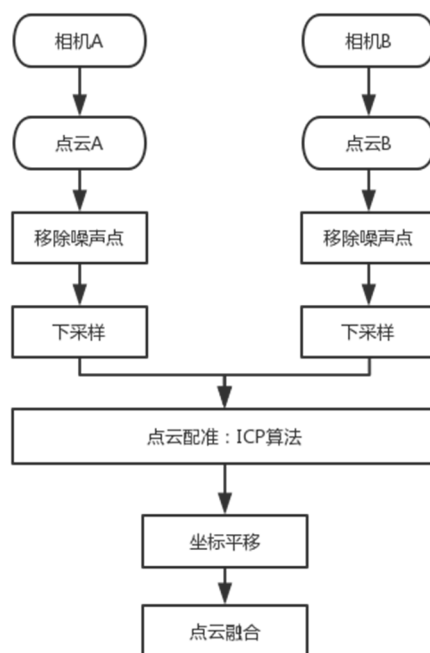


图 3-1 多个 ToF 相机的 3D 场景重建示意图

3.1. 多个 ToF 相机的点云获取和预处理

(1) 点云获取

参考前文 2.3 节可以获取多个相机的点云数据。

(2) 去噪

Kinect 输出的纯点云数据含有非常大的噪声，对于后面的工作其实有较大影响。这里我们使用双边滤波器对深度图进行双边滤波，而滤波后一定要很好的保持边缘特征，在我们的项目下添加了一个 `bfilter2` 工程可以使得我们轻松调用双边滤波器，调用方法为 `bfilter2(img, w, sigma)`，其中 `img` 是二维图像，`w` 是双边滤波的半波长，`sigma` 是双边滤波器的标准差，对于双边滤波器具体的算法细节可以参考

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html[12]。

(3) 下采样

过多的数据点不仅会使得 ICP 算法速度大大变慢,并且会因为参杂噪声使得其准确率变低,所以要通过下采样减少样本点。Matlab 中可以直接调用下采样函数,如下: `fixed = pcdownsampling(ptCloudMiddle_copy, 'gridAverage', gridSize)`, 其中 `gridSize` 是采样大小,我们这里设为 0.05 比较合适。

3.2. 多个 ToF 相机的点云配准

对于点集配准,即有两个点集,原始数据和目标数据,目标数据始终不变,原始数据经过旋转和平移甚至加上尺度变换,使得变换后的原始数据点集尽量和目标数据点集重合。而点云配置过程就是要求这里的旋转和平移矩阵。

在我们的实验中,我们使用 ICP 算法进行点云配准[2][3]。

在我们的实验中,用了两个相机配置,我们以中间的相机为参考中心,左边的相机坐标往中间相机坐标系配准。

3.2.1. ICP 算法

迭代最近点算法 ICP (Iterative Closest Point)算法最初由 Besl 和 Mckey 提出,是一种基于轮廓特征的点配准方法。基准点在图像坐标系及世界坐标系下的坐标点集 $P = \{p_i, i = 0, 1, 2, \dots, k\}$ 和 $Q = \{q_i, i = 0, 1, 2, \dots, n\}$ 其中, P 与 Q 元素间不必存在一一对应关系,元素数目亦不必相同,设 $k \geq n$ 。配准过程就是求取 2 个坐标系间的旋转和平移变换矩阵,使得来自 P 与 Q 的同源点间距离最小。对于 3D 空间中任意两个点 p_i, q_i , 定义距离为

$$d(p_i, q_i) = \|p_i - q_i\| = \sqrt{(x_{p_i} - x_{q_i})^2 + (y_{p_i} - y_{q_i})^2 + (z_{p_i} - z_{q_i})^2}$$

三维点云匹配问题的目的是找到 P 和 Q 变化的矩阵 R 和 T , 利用最小二乘法对于下述待求式

$$E = \sum_{i=1}^N |(Rp_i + T) - q_i|^2$$

求最优解, 使得其最小, 求得此时的 R 和 T

那么其过程如下:

(1) 数据预处理

利用 Geomagic Studio 中删除点的工具, 去除原始数据中的一些隔离的噪点。

(2) 平行移动和旋转的分离

分别求得 P 和 Q 的重心

$$p = \frac{1}{n} \sum_{i=1}^N p_i$$

$$q = \frac{1}{n} \sum_{i=1}^N q_i$$

那么我们分别将点集 P 和 Q 平移至中心点处:

$$p' = p_i - p$$

$$q' = q_i - q$$

则上述最优化目标函数可以转化为:

$$E = \sum_{i=1}^N |(R(p' + p) + T) - (q' + q)|^2 = \sum_{i=1}^N |(Rp' - q') + (T - Rp - q)|^2$$

则最优化问题化简为求使得 E 最小的 R, 和 $T = Rp - q$

(3) 取控制点, 求初始旋转矩阵

在确定对应关系时, 所使用的几何特征是空间中位置最近的点。这里, 我们甚至不需要两个点集中的所有点。可以指用从某一点集中选取一部分点, 一般称这些点为控制点(Control Points)。这时, 配准问题转化为:

$$E = \sum_{i=1}^N |Rp_i - q_i|^2$$

这里 p_i, q_i 是最近匹配点。

对于每对点 i, 计算点对的矩阵 A_i :

$$A_i = \begin{bmatrix} 0, (p' - q')^T \\ p' - q', (p' + q')^T \end{bmatrix}$$

对于所有 A_i 计算 B

$$B = \sum_{i=1}^N A_i A_i^T$$

再就是求解旋转矩阵, 找到上述四元数矩阵 B 对应的最大特征值和对应的特

征向量 $[f_0, f_1, f_2, f_3]$ ，那么旋转矩阵 R 如下：

$$R = \begin{bmatrix} f_0^2 + f_1^2 - f_2^2 - f_3^2, 2(f_1f_2 - f_0f_3), 2(f_1f_3 + f_0f_2) \\ 2(f_1f_2 + f_0f_3), f_0^2 - f_1^2 + f_2^2 - f_3^2, 2(f_2f_3 - f_0f_1) \\ 2(f_1f_3 - f_0f_2), 2(f_2f_3 + f_0f_1), f_0^2 - f_1^2 - f_2^2 + f_3^2 \end{bmatrix}$$

(4) 求平移矩阵

由上述 $T = Rp - q$ 可以求得。

这样我们就得到了 P 和 Q 的配准，得到配准变换矩阵 R ， T ，其中 R 是 3×3 的旋转矩阵， T 是 3×1 的平移矩阵。

(5) 计算误差，迭代

计算坐标变换，即对于集合 P ，用配准变换矩阵 R ， T 进行坐标变换，得到新的点集 P_1 ，即 $P_1 = PR + T$ ，计算 P_1 与 Q 之间的均方根误差，如小于预设的极限值 ε (或者到达迭代次数限定)，则结束算法，否则，以点集 P_1 替换 P ，重复上述步骤，直到达到终止条件。

除了上述方法，用奇异值分解法 (SVD 方法) 来求解 ICP 算法过程中的几何参数最初是由 Arun 等提出来的，其并没有建立目标函数等式，而是通过矩阵变换的相关性质，直接求解出最优的参数解。

在 ICP 算法中，很重要的一点就是特征点的选择，不同特征点集对应不同的平移和旋转矩阵，这样对应不同的结果。最近点搜索方法主要分为一下两种：

(6) 点到点(Point to Point)最近点搜索法

在标准 ICP 算法中，Besl 和 McKay 直接采用的是点到点的欧氏距离。首先利用点到点的最小欧氏距离得到点到集合的距离，从而寻找到对应点，再对这些对应点到集合的距离进行求和得到求解刚体变换的目标函数，如上面所示。简而言之，标准 ICP 算法使用的是点到点(point-to-point)的欧式距离。

(7) 点到平面(Point to Plane)最近点搜索算法

该方法利用的是场景点集中的点的法线与模型点集合的交点来确定对应点，得到对应点后，目标函数则采用的是点到面(point-to-plane)的距离，点到面的距离是指，场景数据集中的点到模型数据集合中的经过对应点的切平面的距离。

具体的，搜索算法是根据源曲面上 P 点的切平面的法线，确定发现于目标曲面的交点 Q' 。根据目标曲面上 Q' 点求出的过 Q' 点切平面，然后求源曲面上 P 点到过

Q' 点切平面的垂线的交点 Q ，那么 P 和 Q 就视作对应点。

点到平面的距离减少了迭代次数，能够以更快的速度收敛到给定的阈值。Pulli 对点到点的方法和点到面的方法进行了对比讨论，他们指出与点到点的 ICP 算法相比，运用点到平面的距离的方法大大减少了计算量以及迭代次数，但是该方法的鲁棒性并不是太好。

关于 ICP，最后要讨论一下 ICP 中各个环节和因素对算法效果的影响，论文 Efficient Variants of the ICP algorithm 详细解答了影响 ICP 算法的各种因素改变下，其结果与性能如何。最终这些因素根据 ICP 算法分为 6 个步骤：a. 筛选：点集或曲面的筛选（滤波）；b. 匹配：两个点集之间的点进行配对；c. 权重：给每个匹配的点分配权重；d. 去除：去除不符合条件的点对；e. 误差度量：基于以上点对，给出每个点对的误差计算方法；f. 最小化：最小化误差度量[10]。

由上可见，在需要改进 ICP 算法效果时，可以考虑调整以上参数。

3.2.2. ICP 算法实验结果

下图表示 ICP 算法进行点云配准的结果，我们可以看到效果非常糟糕，原因是两个相机之间的初始位置差距太大，背景又为较平坦的墙面，使得 ICP 算法中在寻找特征点时很容易匹配错误，导致后面的点云配准出现较大误差，所以我们需要在 ICP 算法之前预处理，采用的方法是用棋盘格先进行相机标定，来获得相机间的位置关系，得到一个初步的点云配准，使得两个相机的点云数据统一在一个坐标系下。

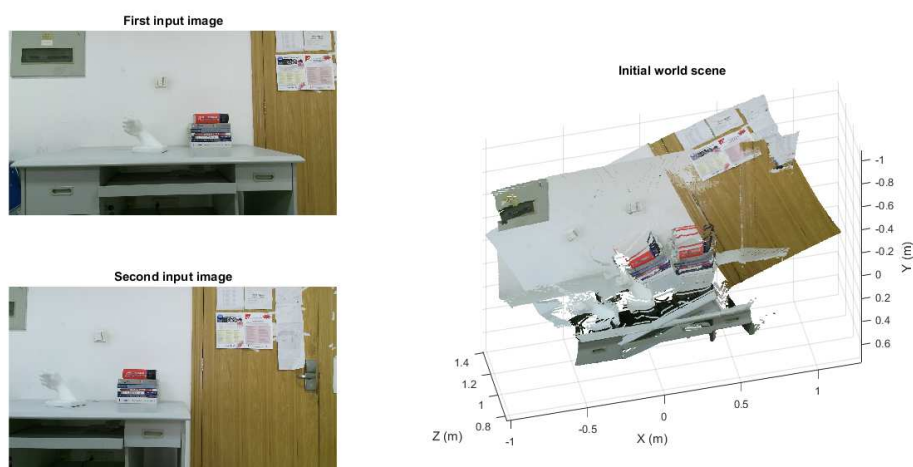


图 3-1 ICP 算法点云配准结果

3.3. 结合相机标定的多相机 3D 场景重建

如前文所说,对于我们的采集数据,单纯用 ICP 算法效果不好,所以需要采用相机标定先把点云数据大致统一到同一坐标系下,再用 ICP 精确配准。这里我们采用微软的张正友标定法。

3.3.1. 张正友标定法

张正友标定法是一种灵活的标定多个相机位置的方法,该方法只需要相机从很少的位置(大于等于两个)观测平面,相机和平面都可以以未知速度自由移动,透镜畸变被考虑在该模型中。该方法,先给出封闭解,然后再基于最大似然给出理想的非线性解,解出相机的内参和外参[11]。

该方法需要棋盘格的辅助,而具体方法在 matlab 中已经有完善的封装,我们可以直接调用。这里之所以要用张正友标定法,是因为我们采集的数据两个点云初始位置和数据差距比较大时,可以应用该方法先大体标定到同一坐标下,再通过 ICP 具体配准,最后点云融合。

3.3.2. 结合相机标定的多相机 3D 场景重建结果

对于多个相机的 3D 场景重建,我们分别实验并比较只用棋盘格和既用棋盘格又用 ICP 的效果。具体实验步骤如下:

步骤 1: 运用棋盘格进行双目标定,运用棋盘格进行双目标定,Matlab 自带应用程序 Stereo Camera Calibrator 可以根据成对带棋盘格的深度图或者彩色图计算两个相机之间旋转矩阵和平移矩阵。Stereo Camera Calibrator 界面示意图如下:

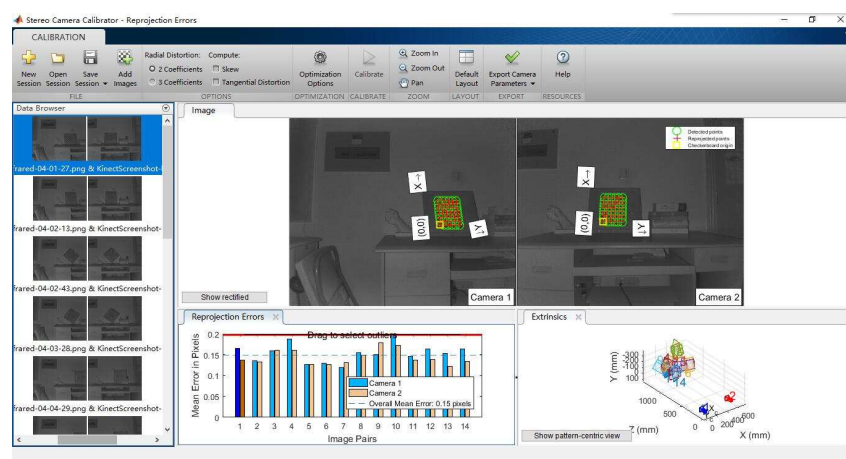


图 3-2 Stereo Camera Calibrator 界面

步骤 2: 通过双目标定结果将点云配准到一个坐标系下并融合, 这里我们发现前景的细节匹配的很准确, 但是后面的墙面有明显的分层现象。

效果图如下:

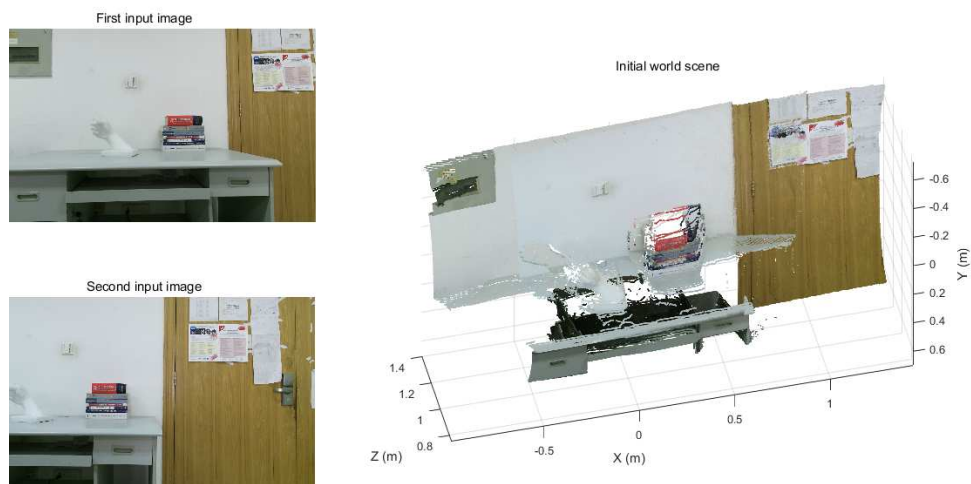


图 3-3 棋盘格双目标定效果

步骤 3: 再上述双目标定后, 再运用 ICP 进行点云配准, 并且进行点云融合。之所以这么做, 是为了补偿前面墙面匹配不准导致融合后分层的现象, 但是使用 ICP 后我们发现后面的墙面虽然匹配更准确了, 但是前景反而效果变差了。

效果图如下:

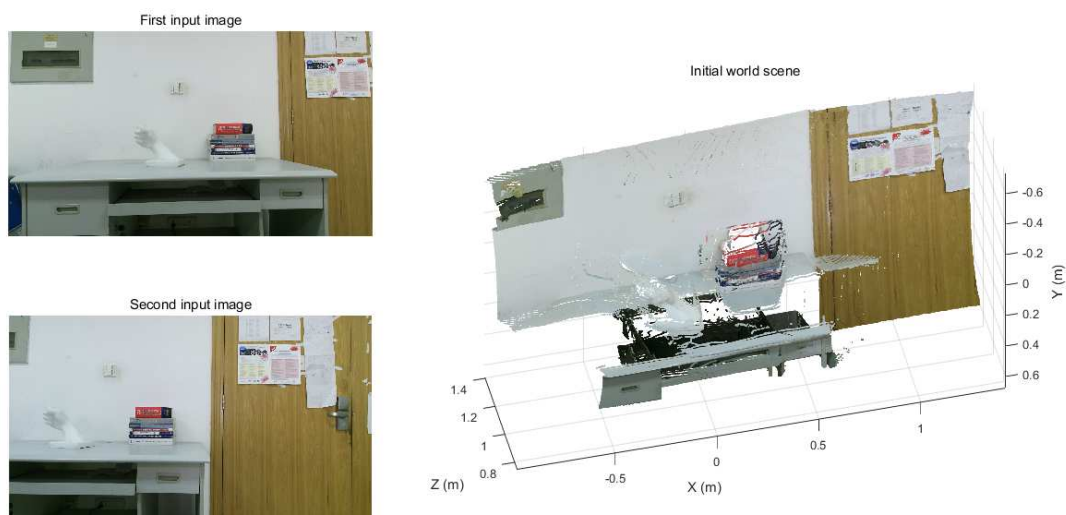


图 3-4 棋盘格双目标定并且 ICP 效果

所以比较而言, 只用棋盘格进行双目校准可以大概将多个点云配准, 但是对于

一些平面并不是匹配的很准确，而使用 ICP 算法并且对其参数设置得当的话，对于墙面等平面会得到更好的配准，但是对于前景中比较复杂的物体，结果相反会变得略差，在 Fast ICP 中对此有所讨论，我们认为是噪声过大和前景石膏像过于复杂造成的。这里可以考虑加双边滤波器进行滤除噪声，然后再进行 ICP 配准，但是由于三维的双边滤波需要消耗大量运算资源，而我们的计算设备能力不足，所以这里没有使用。而如果采用计算量较小的对深度图进行二维双边滤波，再还原回点云，发现重建后结果依然不太好。总体而言，棋盘格双目配准和 ICP 配置的方案已经能够比较好的解决多个点云的配准问题。

3.4. 多个 ToF 相机的点云融合

对于之前的点云配准后的点云融合过程，我们使用了 matlab 中自带的融合函数 `pcmerge` 进行工作，使用方法具体如下 `pcmerge(ptCloudRight2, ptCloudMiddle, mergeSize)`，其中前面两个参数是要被融合的两个点云，`mergeSize` 参数代表融合的单元格大小，具体含义可以参考 TSDF 算法，而 TSDF 具体算法可以参考以下网址：<http://blog.csdn.net/fuxingyin/article/details/51417822>。最后输出的就是融合后的场景点云。

4. 多个 TOF 相机干扰消除

4.1. 多个 TOF 相机干扰问题

多相机问题即是相机在接收的时候, 信号不仅包含自己发射的信号, 也包括其他相机的信号。例如下图所示。

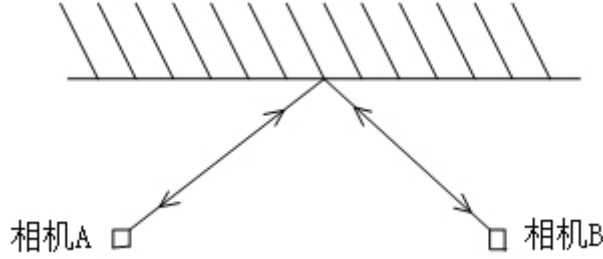


图 4-1 多相机干扰示意图。相机 B 不仅接收自己发出后反射回的信号, 也接收到相机 A 发射的信号。

对于多相机干扰问题, 我们做出如下定义, 发送信号:

$$s(t) = 1 + \cos(2\pi f_0 t)$$

这里调制频率 $f_0 = 1/T_0$

接受信号:

$$r(t) = B + A\cos(2\pi f_0 t + \varphi)$$

这里相位 $\varphi = 4\pi f_0 d/c$, 幅度 A 和传输损失和折射率有关, 偏置 B 是环境光产生的, c 是光速。所以 TOF 相机可以根据幅度 A 和相位 φ 分别计算场景的颜色和深度。要计算它们, 我们需要进行让接收信号和原发射信号按照采样间隔 t_p 进行如下解调:

$$C(t)|_{kt_p} = (s \otimes r)(t), k \in Z$$

这里 \otimes 代表了求相关函数。

一般的, 求相关正是一个根据时间间隔 Δt 采样的过程。大多数 TOF 相机是 4 节拍设备, 也就是说 $t_p = \frac{kT_0}{4}, k = 0, 1, 2, 3$, 那么我们可以以此估计 $\{A, \varphi\}$, 如下:

$$A = P_0 \sqrt{(C(3t_p) - C(t_p))^2 + (C(0) - C(2t_p))^2} / 2$$

$$\varphi = \arctan\left(\frac{C(3t_p) - C(t_p)}{C(0) - C(2t_p)}\right)$$

这里 P_0 是个和 Δt 和 f_0 相关的常数。这里距离 d 可以如下计算:

$$d = c\varphi/4\pi f_0$$

以上是单个相机的深度 d 估计问题, 那么对于多个相机, 假如 2 个相机 C_1, C_2 , 我们进行如下讨论计算深度 d 。二者信号分别是 S_1, S_2 , 他们的调制频率相同。二者对应的接收信号分别是 r_1, r_2 , 用 r_{11} 表示 C_1 发送 C_1 接收的信号, r_{12} 表示 C_2 发送 C_1 接收的信号, r_{22} 表示 C_2 发送 C_2 接收的信号, r_{21} 表示 C_1 发送 C_2 接收的信号, 那么有如下关系式

$$\begin{cases} r_1 = r_{11} + r_{12} \\ r_2 = r_{22} + r_{21} \end{cases}$$

其中每一项都可以表示如下:

$$r_{ij} = B_{ij} + A_{ij} \cos(2\pi f_0 t + \varphi_{ij}), \quad i, j = 1, 2$$

那么我们可以继续化简

$$\begin{aligned} r_1 &= (B_{11} + B_{12}) + A_{11} \cos(2\pi f_0 t + \varphi_{11}) + A_{12} \cos(2\pi f_0 t + \varphi_{12}) \\ &= \tilde{B} + \tilde{A} \cos(2\pi f_0 t + \tilde{\varphi}) \end{aligned}$$

我们可以看到, 其中

$$\begin{aligned} \tilde{B} &= B_{11} + B_{12} \\ \tilde{A} &= P_0 \sqrt{(A_{11}^2 + A_{12}^2 + 2A_{11}A_{12}\cos(\varphi_{11} - \varphi_{12}))/2} \\ \tilde{\varphi} &= \arctan\left(\frac{A_{11}\sin\varphi_{11} + A_{12}\sin\varphi_{12}}{A_{11}\cos\varphi_{11} + A_{12}\cos\varphi_{12}}\right) \end{aligned}$$

由此可见, 如果我们据此估计深度 d , 会是错误的。

4.2. 多 TOF 相机干扰消除

多相机干扰消除就是要准确的估计幅度 A 和相位 φ , 从而得到深度 d , 传统方法无法从频域解决这个问题, 但是从时域, 能够从统计方法上消除相机之间的串扰 [8]。

尽管 TOF 相机的工作光源是连续波, 但是他的调制作用只发生在取样时间段内, 而在每一帧其他的时间, 是关闭的。也就是说 TOF 相机在每一帧内完成全部采样工作, 而和其他的帧是没有关系的。此外, 在每一个像素传感器内, 测量值独

立同分布的。所以说对于 N 帧工作，我们将相机 C_1 的接收信号视作一个序列为 $\{r_1(i), i = 1, 2 \dots N\}$ 的独立同分布变量，其也可以被写为

$$r_1(i) = \tilde{B}(i) + A_{11}(i) \cos(2\pi f_0 t + \varphi_{11}(i)) + A_{12}(i) \cos(2\pi f_0 t + \varphi_{12}(i))$$

这里 N 表示帧的数量，因为每一帧之间都是独立的，所以说每一帧之间信号的初始相位不同。由于多相机之间的同步关系，我们可以视作它们在每帧上都串扰。所以相机每次测量自己的相位是相同的，而接受的别的相机的光每次相位不同。除此之外，对于同一个参考点，光的传播可以视为恒定，假设物体静止不动。所以如果忽略损失， $\tilde{B}(i), A_{11}(i), A_{12}(i), \varphi_{11}(i)$ 是不变量，而 $\varphi_{12}(i)$ 是变化的量，其在不同帧中体现了 C_1, C_2 相位的不同。所以上式可以继续化简

$$r_1(i) = X_1 + I(t)$$

这里有不变项 $X_1 = \tilde{B}(i) + A_{11}(i) \cos(2\pi f_0 t + \varphi_{11}(i))$ ，串扰项 $I(t) = A_{12}(i) \cos(2\pi f_0 t + \varphi)$ ，和随机相位 $\varphi = \varphi_{12}(i)$ 。

那么我们消除串扰的原因是想从 $r_1(i)$ 估计 X_1 。作为最优估计， X_1 的矩估计 \widehat{X}_1 就是 $E[X_1]$ ，如下

$$\widehat{X}_1 = E[X_1] = E[r_1(i)] - E[I(t)]$$

其中串扰项 $I(t)$ 可以被视作随机相位过程，这里 φ 是 $(0, 2\pi)$ 之间的均匀分布，我们可以根据如下计算期望

$$E[I(t)] = m_I = \int_0^{2\pi} A_{12}(i) \cos(2\pi f_0 t + \varphi) \frac{1}{2\pi} d\varphi = 0$$

这里串扰项 $I(t)$ 是广义稳态随机过程，可以由各态历经证明。我们可以使用任意取样函数取时间平均去取代稳态过程的平均。除此以外，对于多帧工作的时域，深度测量是一个离散的过程，当 N 足够大时，上述式子离散条件下依然接近 0。所以要估计串扰，我们可以计算 $r_1(i)$ 的期望，并且因此得到

$$\widehat{X}_1 \approx E[r_1(i)]$$

误差为 $A_{12} \sum_{i=1}^N \cos[2\pi f_0 t + \varphi_{12}(i)] / N$ ，上述式子中， N 不断增大则误差不断减少，最后会接近于真实值。所以说，基于多帧联合处理，用矩估计，可以得到多相机干扰情况下真实的景深数据。

4.3. 实验和结果

对于研究多相机干扰问题, 由于 Kinect Studio SDK 并没有提供对应的采集连续帧图像的功能, 我们于是编写 C#程序自行截取视频中的连续帧, 然后通过求期望的方式实现去除多路径干扰, 具体步骤如下。

步骤 1: 根据前文 2.3 节讲到的方法提取连续多帧深度图。

步骤 2: 根据之前讨论的多相机去噪方法, 编写 Matlab 程序, 对以上提取.png 格式深度图做期望, 如 2.3 节中图所示, 于是便能消除多相机之间的干扰。注意没有意义的点不要计入计算, 没有意义的点包括深度为 0 和深度明显超出场景范围的点。如果把没有意义的点也计入计算, 最后结果会依然错误。

步骤 3: 编写 Matlab 程序, 利用每个相机采集的彩色图和深度图恢复点云, 其中.png 格式的深度图每一个像素点的值代表深度数据 (单位 mm)。注意在恢复点云的时候要连接 Kinect 相机, 否则无法使用 Kinect SDK 中对应的接口函数。

以下是多相机去干扰前后的比较:

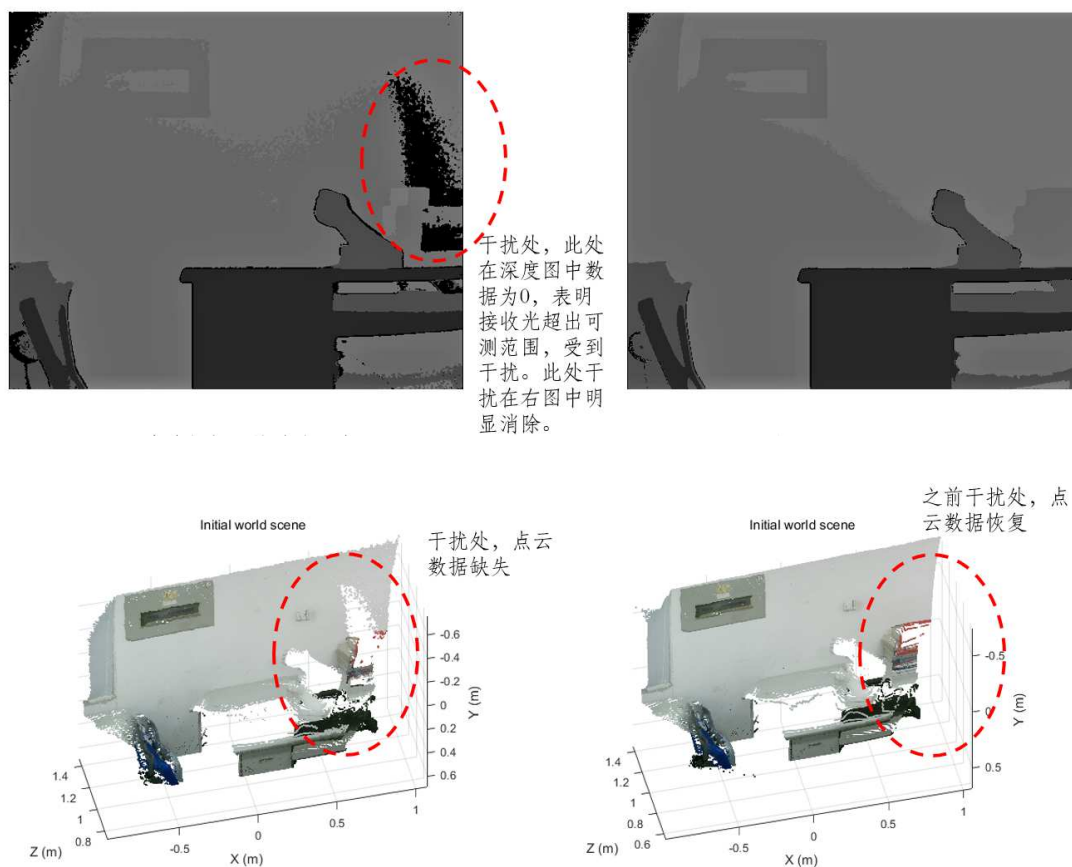


图 4-2 去多相机干扰效果

这里可以清楚地看到，在书的上方这一块，即图中用红圈圈出来处，在去干扰之前，由于多相机之间的干扰，会造成数据点的缺失，而在我们的多相机去干扰之后，补偿了之前缺失的点，很好的解决了多相机干扰问题。

5. TOF 相机多路径干扰消除

5.1. 多路径干扰问题

5.1.1. ToF 测量

从 ToF 相机测出的向量作为开始进行描述。对于一个给定的像素点，传感器发出多个频率调制的红外线。光打在了场景的平面上，有一些光线反射到检测器。对于 m 个调制频率中的每一个，是相同频率的正弦形式光聚合，使得 v 的相位测量取决于跟平面的距离：

$$v \in \mathbb{C}^m, \text{with } v_k = x e^{\frac{2\pi i d}{\lambda_k}}, k = 1, 2, \dots, m \quad (1)$$

这里 d 表示到平面的距离， $\lambda_k = c/2f_k$ 是半波长，相对应于第 k 个调制频率 f_k ，而 x 是一个和接收光强度相关联的量。对于 m 一个典型的选择是取 3，这足够避免混叠效应。

5.1.2. 多路径

等式(1)假设场景中没有多路径。如果只有一条多余的路径，即称作双路径，公式将会改写为如下所示：

$$v_k = x_1 e^{\frac{2\pi i d_1}{\lambda_k}} + x_2 e^{\frac{2\pi i d_2}{\lambda_k}} \quad (2)$$

这里 d_1 和 d_2 是两条路径的距离， x_1 和 x_2 给了这两条路的长强度，如果 $d_1 < d_2$ ，那么 d_1 就是真实距离而 d_2 是多出来的那条路径， x_2/x_1 给了多路径的强度。

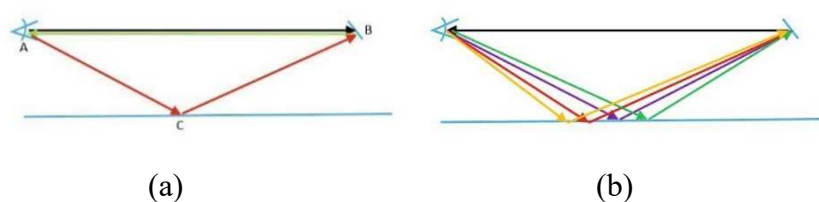


图 5-1 多径效应示意图。(a)双径效应，相机 A 和平面 B 用蓝色显示，用其他颜色表现光线。正确的路线是 A-B-A；不正确的光路是 A-C-B-A。(b)散射多径，这是由于有很多相邻点反射的光线方向近似，这里显示了 4 种颜色的光路。

当然了，还可以用公式表示更一般的情况，包括 3 条或者更多条，我们将公式化为如下：

$$v_k = \sum_{j=1}^n x_1 e^{\frac{2\pi i d_j}{\lambda_k}} \quad (3)$$

这里 d_j 是相关间隔。在典型例子中, 我们的物体距离为 20cm 到 450cm, 每次增加 1cm。因此, 这里 $n=431$, 向量 x 被称作反射。

等式(3)包括散射多路径。一个理想的朗伯表面接收已知方向的光线, 并且反射极小量的到别的所有方向。实际上, 一个无限数量的相近点在平面上反射极小量, 结果是有限的。多路径散射对于我们来说, 一般可以表示为 $v_k = x_l e^{\frac{2\pi i d_l}{\lambda_k}} + \sum_{j=l+\Delta}^n x_L(d_{j-\Delta}) e^{\frac{2\pi i d_j}{\lambda_k}}$, 对于 $\Delta \geq 0$ 。 $x_L(\cdot)$ 的形状可以在查找散射的仿真得到, 它大约为 $x_L(d) \approx A d^\alpha e^{-\beta d}$, 这里 α 和 β 取决于场景的几何形状。

可以用向量矩阵形式重新写等式(3), $v = \Phi x$, 这里 $\Phi \in \mathbb{C}^{m \times n}$ 并且 $x \in \mathbb{R}^n$, 再将复数化为实数, 通过堆其实部于虚部之上, 用以下表示:

$$\Phi \in \mathbb{R}^{2n \times m}, \text{ with } \Phi_{kj} = \begin{cases} \cos\left(\frac{2\pi d_j}{\lambda_k}\right) & \text{if } k = 1, \dots, m \\ \sin\left(\frac{2\pi d_j}{\lambda_{k-m}}\right) & \text{if } k = m+1, \dots, 2m \end{cases}$$

然后我们可以将它写作

$$v = \Phi x \quad (4)$$

而现在所有变量都是实数了。

5.1.3. 反射光的特性

现在让我们探究多路径情况下反射光 x 的特点。

特点 1: 非负性。即对于所有距离必须大于等于 0。

特点 2: 可压缩性。这个特性很有趣, 我们发现两条路径场景下, 除了 2 以外所有下标对应的 x 都是 0 (对于等式(2)中的 d_1 和 d_2), 所以 x 是稀疏的。

另一方面说, 对于散射多路径, 有基本形式 $x_L(d) \approx A d^\alpha e^{-\beta d}$, 不是稀疏的。取而代之, 离散化的版本有以下几点属性: 当 x 的相关性被由大到小排列时, 结果向量很快就会收缩到 0, 这个属性被称为可压缩性。

规范的, 给定 $x = (x_1, \dots, x_n)$, 让 $(x_{l(1)}, \dots, x_{l(n)})$ 表示这些向量由大到小的排列,

那么如果在 $r \leq 1$ 的前提下 $x_{I(i)} \leq Ri^{-\frac{1}{r}}$, 那么称 x 为可压缩。也就是说 x 的排列很迅速的递减。

5.2. 多路径干扰消除

5.2.1. SRA(Sparse Reflections Analysis)算法

上文已经通过反射光 x 表示了多路径, 并且进而探究了反射光 x 的属性, 即非负性和可压缩性。我们接下来急需讨论如何使用这些信息去除多路径效应, 并且找到一个鲁棒和准确的方法来通过 ToF 测量的向量来估算场景的深度[4]。

(1) 多解

给定 v , 那么我们知道通过反射光 x 可以得到 v , 例如, 根据等式(4), $v = \Phi x$, 给定 $x \in R^n$, 并且 $v \in R^{2m}$, 这里 $n \gg 2m$, 这样很多可能的 x 都能产生 v 。但根据我们对于反射光的定于, x 是一个非负的并且可压缩的, 那么就约束了可能的 x 的数量。

实际上, 由于传感器的噪声, 我们得不到 $v = \Phi x$, 取而代之, 我们得到的可能是 $v = \Phi x + \eta$, 这里 η 被称作高斯噪声, 期望为 0, 协方差矩阵为 C^2 。

(2) L0 最小化

假设, 在某个时刻, x 是稀疏的而不是可压缩的, 也就是说 x 有很少量的非零输入, 我们将非零元素称作它的 0 范数。比如 $\|x\|_0$ 。在这个情况下, 我们要解决的是如下问题:

$$\min_{x \geq 0} \|x\|_0 \quad \text{subject to} \quad (\Phi x - v)^T C^{-1} (\Phi x - v) \leq \epsilon^2 \|v\|^2 \quad (5)$$

对于参数 ϵ , 在我们的实验中, 被设为 0.05。也就是说, 我们想要稀疏化反射光 x 进而避免在一定噪声容忍限度下去测量 v 向量。不幸的是, 上述问题, 是一个 NP 问题, 很难解。

(3) L1 最小化

那么, 也就是说给定一个 Φ , 来解决以下问题:

$$\min_{x \geq 0} \|x\|_1 \quad \text{subject to} \quad (\Phi x - v)^T C^{-1} (\Phi x - v) \leq \epsilon^2 \|v\|^2 \quad (6)$$

这样就会回避[3,2,4]中相似的对于(5)优化的解决。注意到这两个优化唯一的区别是, 我们替换了 0 范数为 1 范数。这样在(6)中就变成了一个凸问题, 就能在有限的时间

间内得到解决。

实际上,对于前文提到的情况,比如等距约束属性,对于我们的 Φ 矩阵是不普遍适用的。这是由于 Φ 中列之间会有冗余,比如相邻列之间由多余的进而之间影响。然而,我们在(6)中讨论的优化问题,在等距约束属性等情况中遇到的问题就不会出现了。注意到这些工作都是计算机视觉中同根源的,比如 Wright 等人对于人脸识别的工作,就是使用了 L1 优化,并且它的情况和这里关注的点不一样。

所以说,讨论到这里,我们的 x 都是稀疏的,而不是可压缩的,即使 x 是可压缩的而不是稀疏的,那么解决(6)中的 L1 优化问题还是承认了解答的正确性。

(4) L1 约束

尽管优化问题(6)是个凸的,但是它是个二阶锥规划问题,实际运用中很慢。我们因此做出如下改进。注意到 L2 约束,可以写为 $\|C^{-1/2}(\Phi x - v)\|_2 \leq \epsilon \|v\|_2$ 。我们可以用其等价 L1 约束来近似,比如 $\|C^{-1/2}(\Phi x - v)\|_1 \leq \epsilon \|v\|_1$ 。所以优化问题就变成了:

$$\min_{x \geq 0} \|x\|_1 \quad \text{subject to} \quad \|C^{-1/2}(\Phi x - v)\|_1 \leq \epsilon \|v\|_1 \quad (7)$$

它相较于(6)的优势在于它可以化作一个线性函数。所以,它可以相对较快的被解出。首先,注意到因为 $x \geq 0, \|x\|_1 = \sum_{i=1}^n x_i = 1^T x$ 。其次,注意到 $\|z\|_1 \leq \gamma$ 对于 $z \in R^l$ 可以被转化为线性约束 $Q_l z \leq \gamma 1$, 这里 Q_l 是一个 $2^l * l$ 的矩阵。它的行由 $\{-1, +1\}^l$ 组成。它在 l 太大时被禁止,我们这里 $l=2m$, 而通常取 $m=3$, 这导致有另外 64 个约束,比非负约束要少很多,也就可以化作如下方程:

$$\min_{x \geq 0} 1^T x \quad \text{subject to} \quad Ax \leq b$$

这里有 $A = Q_{2m} C^{-1/2} \Phi$ 并且 $b = Q_{2m} C^{-1/2} v + \epsilon \|v\|_1 1$

(5) 通过反射光计算深度

刚才提到的多种优化问题不用考虑反射光 x , 当然了,最后的目标,是估计深度,那么需要通过 x 估计景深。

主要路径一定是最短路径,这是有图像处理得到的几何形状得到的。所以,我们可以简化问题为,求第一个非零 x 的索引,使得 $i_1(x) = \operatorname{argmin}_i \{i: x_i > 0\}$, 然后深度就是 $\sigma = d_{i_1(x)}$ 。实际运用中,由于一些问题,将会有很多非零但是很小的 x ,

所以我们求 $i_1(x) = \operatorname{argmin}_i \{i: x_i > c \max_i x_i\}$, 我们这里用 $c=0.01$ 。

如果我们有合适的准确的噪声模型, 我们可以做得更复杂。对于每个反射光的峰值, 我们可以计算避免这些峰值是由噪声而不是信号产生。那么第一个峰值是噪声和第一个峰值是信号的可能性的和是 1, 第二个峰值是真实信号的概率是第一个峰值是噪声的概率乘以 1 减第二个峰值是噪声的概率的差, 以此类推。如果没有返回值的可能不超过阈值 0.9, 我们可以认为这个像素点不合法。也就是说, 我们不知道真实信号是多少。

5.3. 快速计算

SRA 算法允许我们通过传感器测量 v 得到反射光 x , 通过反射光, 我们可以得到深度。现在的问题就是如何加速计算。解决上述 L1 优化问题典型的是对于普通 CPU 需要大概 50ms 每个实例。给定的 ToF 图像一般有成百上千个像素点, 也就是说大概对于每个框架需要将近一个小时的计算。为了使得码率得到 30Hz, 需要一种更为激进的方法去实现。我们现在描述的算法对于 424×512 的图像, 可以用 SRA 以一定码率在普通硬件上运行。

这里 SRA 的降维方法主要是预计算得到快查表作为辅助降一维, 然后通过正则计算再降一维, 一共降两维, 加速了计算。

5.4. 实验和结果

我们利用 SRA 方法对微软提供的场景数据 (包括正常部分和有干扰部分) 进行了全部处理, 其结果中有的地方效果变好了, 有的地方结果相反变差了, 特别是原本没有干扰的地方经过估计求解后, 分层效果过于明显, 这是由于在 L1 的过程中本身线性规划就有一个分层的过程。为了解决这个问题, 我们最后实际处理时, 人工大致截取明显出现了干扰的地方, 然后对其进行 SRA 估计求解后, 再重新生成场景点云数据, 如图 5-2 所示。可见, 对由于多路径产生的模糊和扭曲物体成功进行了纠正。

另外, 由于微软没有给我们提供具体的噪声模型, 所以我们的噪声都是自己采集数据测量计算的, 跟微软在论文中采用的噪声不一样, 这可能是导致我们的结果没有论文中那么好的原因。

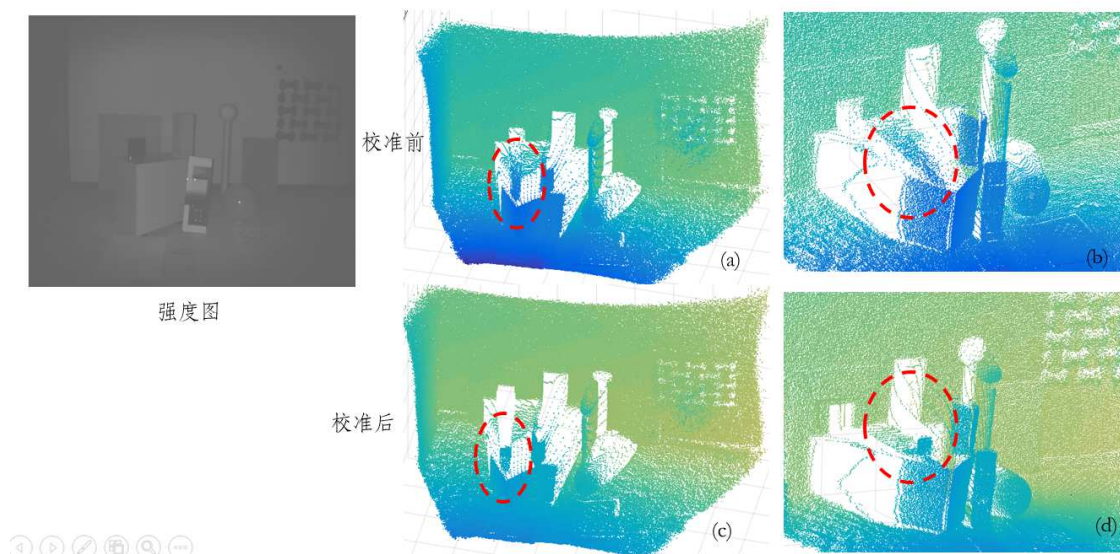


图 5-2 去多路径干扰效果

本实验中还有个问题就是速度比较慢，约两三个小时才能算出结果，由于我们没有原论文中提供的快查表作为参考，所以 L1 优化的速度很慢，在今后的实验中如果有条件，还需要在这方面进行改进。

6. 结论

本次毕业设计的目的是研究多个 ToF 相机的三维场景重建问题, 需要考虑多路径干扰和多相机干扰对点云数据的干扰。通过实验, 我们成功使用棋盘格法相机标定和 ICP 算法实现了对多个点云的配准问题, 并且重建了较好的 3D 场景, 再通过求多帧期望消除了多相机干扰问题, 最后通过 SRA 算法基于 L1 优化成功消除了多路径干扰问题。在实验中我们发现, ICP 算法对于点云过于凌乱复杂的情况性能还有待加强, SRA 算法对于噪声比较敏感, 这些都是在今后研究中需要改进的地方。本课题对于三维场景重建有着实际意义。

致 谢

完成本文首先需要感谢喻莉老师的细心指导，在实验做不出来的时候给出建议，在文章看不懂的时候指点方法，在工作完成后提供改进意见，正是在喻老师的指点下，本文才能完成的如此顺利。而在实际实验过程中，感谢李连华学长和实验室其他师兄给出的宝贵指导。其次，我想感谢大学四年遇到的每一个人，感谢老师对我的传授知识和教导方法，感谢同学朋友之间的互帮互助和共同进步。最后，我想感谢含辛茹苦把我精心呵护到现在的父母，无论在外面有怎么样的大风大浪，父母提供的家总是为我遮风避雨，并且不求回报的提供一切支持。

感谢所有人，没有你们的付出与支持，本文难以完成，谢谢！

参考文献

- [1] Remondino F, Stoppa D. TOF range-imaging cameras[M]. Heidelberg, Germany:: Springer, 2013.
- [2] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Robotics-DL tentative. International Society for Optics and Photonics, 1992: 586-606.
- [3] Bergevin R, Soucy M, Gagnon H, et al. Towards a general multi-view registration technique[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(5): 540-547.
- [4] Freedman D, Smolin Y, Krupka E, et al. SRA: Fast removal of general multipath for ToF sensors[C]//European Conference on Computer Vision. Springer International Publishing, 2014: 234-249.
- [5] 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.
- [6] Choi S, Zhou Q Y, Koltun V. Robust reconstruction of indoor scenes[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5556-5565.
- [7] Izadi S, Kim D, Hilliges O, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera[C]//Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011: 559-568.
- [8] Li L, Xiang S, Yang Y, et al. Multi-camera interference cancellation of time-of-flight (tof) cameras[C]//Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015: 556-560.
- [9] Yan Z, Yu L, Xiong Z. Texture-free large-area depth recovery for planar surfaces[C]//Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on. IEEE, 2015: 1-6.
- [10] 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.

[11] Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(11): 1330-1334.

[12] Jiakai Zhang. 3D Real-time Reconstruction.

附录

代码: <https://github.com/ZengChen94/3DReconstructionByKinect2V>

数据集及文档: <http://pan.baidu.com/s/1bJX4xo>