单位代码 10006
学　　号 39022602
分 类 号 TP37

北京航空航天大學
BEIHANG UNIVERSITY

毕业设计(论文)

面向室内场景实时监控的视频无线传输与复现

学 院 名 称　电子信息工程学院
专 业 名 称　信息对抗
学 生 姓 名　孙沁璇
指 导 教 师　李洪革，苑晶（南开）

2013 年 6 月

面向室内场景实时监控的视频无线传输与复现

学　　生：孙沁璇
指导教师：苑晶（南开）
指导教师：李洪革

摘要

由于人们对于安全的需求，以及特殊人群在移动存在障碍情况下能够实现对整个室内场景监控的需要，应用于室内场景实时监控的视频无线传输与复现系统有其实现以及面向应用的价值。但传统意义上的实时监控系统只能在接收端实时再现当前视频帧的场景，不能对已采集区域有一个全面的把握，针对此问题，本文提出并初步实现了一种在实时监控的同时利用三维点云实现场景复现的方法。

Kinect 作为图像采集装置，同时采集到彩色图像及其对应像素点的深度信息，并在移动终端进行处理得到三维场景的点云信息。移动终端在获得三维点云后对彩色图像以及点云进行预处理后，通过无线局域网传输给上位机，再由上位机对接收到的每帧彩色图像进行实时的显示，并对每帧点云信息进行位姿的变换完成点云的配准工作，从而完成三维场景的复现。

本文所实现的内容，是在实时监控的同时，利用 Kinect 采集到信息得到室内环境点云数据，并在上位机完成三维场景的重建。与传统实时监控系统不同的是，本文中需要运用计算机视觉相关理论以及移动机器人在定位和构图中所涉及的一些方法，在实时监控的基础上运用 ICP 算法进行点云的配准从而完成全局场景的生成，与此同时为了提高系统的实时性，在彩色图像上利用 SURF 特征检测算子来进行特征的抽取与匹配，并以此为基础来进行点云数据的筛选工作。

关键词：实时监控，场景复现，Kinect，无线传输，点云配准

Wireless Video Transmission and Reproduction of Indoor Scenes Based on the Real-time Monitoring

Author: Sun Qinxuan

Tutor: Yuan Jing

Tutor: Li Hongge

Abstract

It is proved that the video wireless transmission and reproduction system for indoor scene real-time monitoring is not nonsense towards implementation and application, for people's requirements for security, and for the special populations to see the indoor scenes where they can't reach. But the traditional real-time monitoring system could only show us the scene at the current video frame. To solve this problem, this thesis proposes and implements a method for real-time monitoring and reproducing the map of the area using the 3d point cloud simultaneously.

Kinect has been used as an image acquisition device, which receives color images and the corresponding depth information for every pixel at the same time. This information will be processed in the mobile terminal to acquire the point cloud. The images captured by the Kinect will be transmitted to the principle computer and displayed by the principle computer. And the images as well as the point cloud will be processed in the terminal before transmission. The principle computer will finish the registration of the point cloud, to complete the map generating.

The method mentioned in this paper, using the Kinect to get point cloud, collects information to reconstruct the scene for real-time monitoring. Unlike traditional real-time monitoring systems, some related theories of the computer vision and some methods in the SLAM (Simultaneous Localization and Mapping) of the mobile robot are needed. On the basis of the real-time monitoring of point cloud registration, ICP (Iterative Closest Point) algorithm is used in order to complete the global scenario generation, at the same time, in order to improve the real-time performance of system, the use of Surf on color image features detector to extract and match, based on which point cloud data filtering can be achieved.

Key words: Real-time Monitoring, Map Reconstruction, Kinect, Wireless Transmission, Registration of Point Cloud

目录

1 绪论

1.1 研究背景及意义

实时监控技术从最初实现以来，随着科学技术水平的不断进步，大体经历了三个发展阶段[1]。首先是模拟监控阶段，在这个阶段中，摄像头采集到的视频基带信息不经过任何的调制处理直接通过传输电缆传送到中央控制台，并在中央控制台的界面上进行显示。这个阶段的实时监控实现原理较为简单，投资相对较少，但是易受到不同程度的干扰，可靠性以及可扩展性都不是很强。第二个阶段是数字监控阶段，随着数字化的步伐，各类技术也逐步进入了数字化处理的时代，实时监控在数字监控阶段是通过数字信号的处理方式来对视频图像进行传输和处理，在模拟信号经过采样以及量化等过程转换为数字信号时，由于分辨率以及清晰度等方面的要求，数字信号的信息量通常都较大，所以必要时也会进行图像的压缩工作。数字化的处理方式使得视频图像的传输效率和质量都有了一定程度的提高，与此同时还引入了模块化的管理模式，使整个系统更加易于管理。第三个阶段是网络化多媒体监控阶段，近些年来随着网络及多媒体技术的飞速发展，使得实时监控技术在时间及空间上的限制进一步的减小，不仅仅局限于一个固定的空间和设备，真正可以实现一个网络化的多媒体监控系统[2]。

随着互联网的高速发展，近些年国内外有很多基于互联网的监控系统已先后面向应用并投入市场，例如 iSecure[3], DigiEye, TeleEye Pro, Digital Surveillance and Vision-Based Traffic Surveillance System[4]等。它们都是网络化多媒体监控阶段的产物，能够通过网络传输视频数据，且能够保证实时性的要求，其中有些还具有智能识别与处理，录像及回放的功能，为此类产品提供了更广阔的应用前景。还有一些系统具备网络情况监控及传输控制的功能，即根据网络情况来控制传输的一些选项，以达到更加优越的性能。例如在网络情况较为理想时，可以选择传输分辨率较高的视频图像，以确保用户的观看质量，而当检测到网络较为拥堵的情况下，可以自动选择传输分辨率较低的图像，在不影响人的视觉感受的前提下保证传输的速度能够在实时性的可接受范围内，这样虽然是牺牲了一些清晰度，却保证了传输的速率，从而确保系统可以满足实时性要求。基于互联网的实时监控系统相比于传统的监控系统有很多优势，比如可以利用现成的网络资源，省去了额外布线和安装设备的麻烦，另外随着网络的发展，带宽和误码率等性能较以前也有了明显的改善，这样可以为实时传输视频的质量提供更好的保障。

但是一般意义上的实时监控系统，不论是否具有记忆及回放功能，都只是以视频即

图像流的形式将采集到的图像信息显示在用户界面上。以室内场景为例，用户在看到连续的每一帧图像的时候，只能通过想像来构建三维场景信息，即场景中各个物体立体空间信息与物体之间的位置关系。鉴于此，并借鉴了移动机器人在同时定位与建图中的相关思路和算法，本文提出并初步实现了基于室内实时监控的场景复现，利用由 Kinect 获取的深度信息生成的三维点云数据来构建摄像头已采集区域的点云地图。

通过摄像头采集到的图像信息来构建空间地图这项技术在现阶段主要应用于移动机器人的同时定位与建图（SLAM）中。移动机器人在导航与建图中主要涉及到两方面的问题，同时也是 SLAM 的两方面问题[5]：一是机器人在未知环境下如何描述周围的环境特征并构建环境的地图，这一点在机器人导航定位以及与周围环境的交互中是非常重要的前提，二是机器人在未知环境下如何确定自身的定位，而对自身定位的确定也是构建环境地图的一个重要的步骤。当需要对三维场景进行重建并基于此来确定自身定位的时候，移动机器人需要具有对三维空间环境的感知能力，这样才可能获取到一些信息来完成三维场景的重建工作。而目前对三维环境的感知主要可以通过以下三种方法来实现：一是激光扫描，二是双目视觉，最后就是摄像头具有获取深度信息的能力（如 TOF 相机，Kinect 等）。

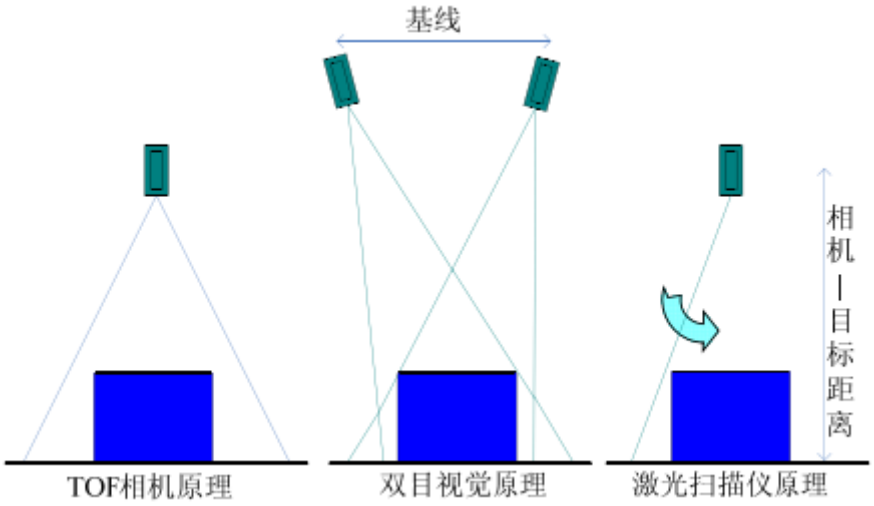


图 1.1 几种感知三维环境的实现方法

如果是通过双目视觉原理来获得三维信息[6]，则最重要的工作便是根据计算机视觉的有关原理，对两个摄像头所得到的同一场景不同视角的图像等信息进行匹配。其实双目立体视觉的根本原理，便是利用多图像的成像来获取场景中物体的深度信息，与直接获取深度信息的传感器不同的是，在双目视觉中，深度信息是在对两幅图像进行处理之后所得到的，而不是传感器直接获得的。所以在双目视觉成像的过程中，需要从两幅图

像中寻找对应于真实场景中同一点或同一区域的信息，再根据计算机视觉相关理论来计算相应点的深度信息，从而实现三维场景的还原。由于寻找两幅图像对应点的这个过程经常并不是能达到很精确的程度，所以双目甚至多目视觉在场景还原的时候一般都无法做到很高的精度。

基于激光扫描来获取三维信息的方法[7]，相对于双目视觉方法具有精度高，信息直观等优点，近年来已成为研究热点。激光扫描仪一般可以快速高精度地获得被测物体表面的三维点云坐标数据，可以用于快速建立物体的三维影像模型。由于激光扫描具有实时性，不接触性，高精度，高密度，自动化以及数字化等等特点，使其应用得到了快速的推广。其应用领域极其广泛，例如测绘工程和结构测量方面，对建筑、地形或者工具设备等进行测量与绘制，另外还用于虚拟场景的构建以及 3D 游戏的开发等项目。

基于深度信息进行三维场景构建，最重要的是完成深度信息的获取。与双目视觉不同的是，这里的深度信息是通过摄像头直接来获取的，我们可以直接获得深度信息来进行下一步的处理，而双目视觉中深度信息是利用已获取到的信息（如两幅图像及其对应关系）经过二次计算处理得到的。常见的直接获取深度信息的摄像头有 TOF（Time of Flight）相机，是通过发射经过调制的红外光，经物体表面反射后返回到相机，相机上的光学传感器则可以根据反射光的亮度以及相位差等信息来获得空间中物体表面各点的深度信息。而 Kinect 的出现，以其较低廉的价格和优异的性能，在很大程度上促进了这个领域的发展，使得三维场景的构建有了一个更加方便直观的解决办法。

1.2 课题研究内容

由于传统意义上的实时监控系统都只将采集到的信息以图像流的方式展现给用户，不能使用户对于采集到的区域的三维信息有一个整体的把握，故本文提出并初步实现了一种在室内实时监控的基础上，利用 Kinect 采集到的信息来构建三维点云并结合 Kinect 获取到的彩色图像一起，进行每帧点云之间的配准工作，从而完成三维场景的重建，并在以图像流形式为用户展示出实时监控结果的同时，以点云地图的方式展示三维场景的构建与复现的结果，从而得到 Kinect 摄像头扫描过的区域整体的空间信息。

从移动终端获取到图像信息并传输完成，到上位机处理并显示的这一过程中，由于实时性的要求，在传输之前移动终端的处理器会完成一定的传输前预处理的工作，例如对三维点云进行滤波处理，一是为了减少点云的数据量从而减少处理的时间，二是通过一些简单的滤波处理过程过滤掉无用的干扰点，提高后期处理过程的准确性。另外在移

动终端还需要进行在彩色图像上完成特征点的提取并利用深度信息将其投影到三维空间中得到两帧点云的对应特征信息，将这些信息再通过无线局域网传送给上位机来进行彩色图像的实时显示以及点云匹配与场景复现的工作。这些在移动终端的处理器中进行的传输前处理工作，由于在一定程度上减少了传输的数据量，所以对于整个系统的实时性以及快速性都是有必要的，另外由于在前期处理的过程中一定程度上去除了一些信息获取过程中的干扰，所以对传输后在上位机中进行的点云匹配工作准确性的提高也有一定作用。

在三维场景重建的过程中，借鉴了移动机器人在同时定位与建图（SLAM）中的一些原理及方法。在 SLAM 中，机器人主要需要完成两个方面的内容，即环境地图的创建以及自身的定位。对于基于视觉的 SLAM，要从获取到的相邻两帧图像或点云信息中，得到带有空间位置信息的视觉特征，从而得到两帧之间的配准关系来完成地图的创建。本文正是运用在两帧点云之间找到的对应特征进行匹配，得到两帧之间的摄像头位姿变换关系，并将其作用于整个点云来进行两帧之间摄像头坐标系的转换，进而完成了增量式的点云地图创建。

1.3 论文组织结构

第一章介绍了课题研究的相关背景，包括实时监控的发展与现状，以及移动机器人视觉 SLAM 的相关情况，概括了课题研究的主要内容以及一些相关的研究方法。

第二章对 Kinect 以及计算机视觉中摄像头模型理论进行了简要的介绍，阐述了摄像头标定的必要性和标定的一些相关方法，并给出了本文中对 Kinect 进行标定及校正的一些基本结果。

第三章对现阶段室内实时监控的方法进行了简单的介绍，并阐述了通过无线局域网进行传输的优势，同时说明了本文中为达到传输的实时性所做的工作及采取的措施。

第四章主要说明场景重建的过程中所采用的一些理论及方法，并给出了在研究过程中对不同方法进行试验比较所得出的结果，着重阐述了在特征点提取中所采用的 SURF 方法以及点云配准中所采用的 ICP 算法以及二者的结合来进一步提高系统的实时性。

第五章主要说明了系统的设计与实现，包括开发环境，软硬件平台搭建，以及实时监控和场景重建的效果。

第六章为全文总结与今后工作的展望。

2 Kinect 简介与点云生成相关原理

2.1 Kinect 简介及工作原理

Kinect 是美国微软公司为 XBOX 360 游戏机和 Windows PC 机开发的一款体感外设，最初的适用领域只是游戏领域，它可以使游戏者脱离鼠标、键盘、控制板等设备，而是直接利用动作和语音等来控制游戏的整个进度以及进行一系列的人机交互。但是基于其优越的性能以及低廉的成本，之后很多其他领域的研发者们都在不同领域中进行开发，将其应用于游戏领域之外，如人工智能，人机交互，体感互动等领域[8]，其探究和研发取得了不错的进展。如[8]中所提到的，欧洲名为 Topshop 的时装店莫斯科旗舰店中，安装了运用 Kinect 体感外设以及增强现实的技术的虚拟试衣间，顾客可以不用真正进行穿与脱的动作，而只需要站在镜子前选择想要的衣服，就可以看到虚拟的 3D 着装效果。还有国外一个名为 BlablabLAB 的小组在街头进行了实验，用三个 Kinect 为游客扫描建模之后用 3D 打印机快速制作出一个对应游客的迷你雕像。除此之外，Kinect 还在很多领域有了成功应用的案例，例如虚拟物理实验，虚拟手术或尸检等等。

Kinect 的外观如图 2.1 所示，它由一个基座和一个感应器组成，基座和感应器之间有一个电动的马达，开发者可以通过程序调整感应器的俯仰角度，在一定的应用场合中可以通过俯仰的调整获得很方便的应用效果。在上面的感应器中有一个红外投影仪，两个摄像头，四个麦克风和一个风扇。打开外面的盖子可以看到里面的构造（图 2.2）。这些感应器可以用来捕捉彩色和深度数据，而通过程序可以直接获得这些数据，再加上已知的摄像头内部参数等就可以完成三维坐标点的计算。在面对 Kinect 方向看，最左边是红外光源，下一个是 LED 指示灯，再下一个是彩色摄像头，最右边是红外摄像头用来采集深度数据。彩色摄像头用来收集 RGB 数据，且成像时支持的最大分辨率是 1280*960，红外摄像头是用来采集深度数据，成像支持的最大分辨率是 640*480[9]。

图 2.1 Kinect 外观[9]

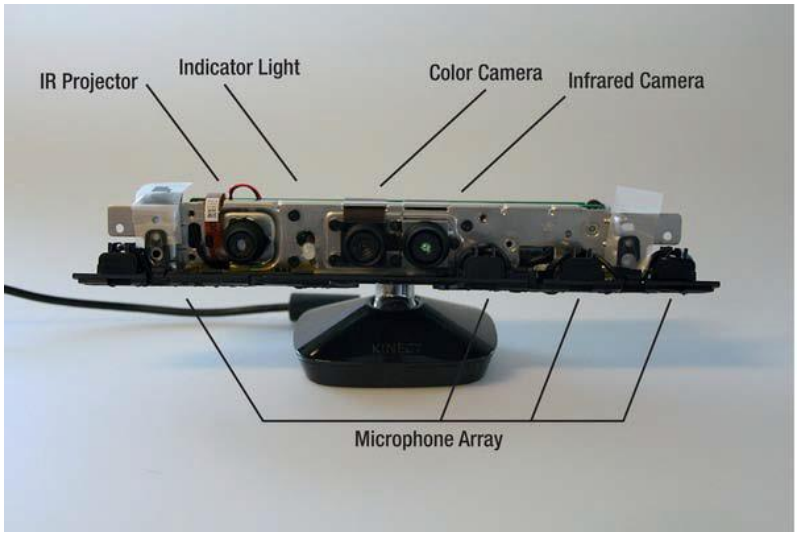


图 2.2 Kinect 组成[9]

Kinect 将一个标准 RGB 摄像机与深度传感器相结合，对于图像中的每一个像素点，都能得到其对应的颜色信息及深度信息，利用颜色与深度信息以及摄像头的内部参数便可以直接得到被采集区域环境的三维特征信息[10]。但由于 Kinect 的红外摄像机对深度敏感距离有限制，一般为距摄像头 4 米以内，所以一般不能得到完整的深度信息，甚至产生一些无效的点，如果不经过处理会影响后期的处理过程。在理想条件下，深度信息的分辨率可以达到 3 毫米。

传统的摄像头无法直接获取深度信息。专业的深度摄像头大多采用 TOF （time of fight），即摄像头主动射出红外光线，红外光线经过物体表面反射后再被摄像头接收，并根据往返的相位差来得到深度信息。Kinect 使用的并不是 TOF 技术，而是 PrimeSense 公司提供的光编码（Light Coding）技术[11]。所谓光编码技术，是用红外光线为测量空

间进行编码，从而获得深度信息，Kinect 通过红外摄像头向采集区域中投射满足一定规律的点阵信息，当场景深度发生变化时，摄像头捕捉到的点阵信息也会随之发生变化，基于此，通过分析点阵模式的变化情况便可以推断出场景的深度距离信息。与 TOF 相比，Light Coding 不需要特制的感光芯片，仅依靠普通的 COMS 感光芯片和连续的照明，就能得到周围环境的图像信息。

2.2 摄像机模型理论

摄像机通过将三维物体投影到二维平面上来完成成像过程，用来投影三维物体的二维平面是在焦点处，与焦点和摄像头所成直线平行的平面。在将三维空间的坐标点投影到成像平面的过程中，实际上丢弃了三维空间点的深度信息，所以这是个不可逆的过程，如果没有额外的条件，不能由成像平面的二维坐标点直接得到三维空间的坐标点。理想情况下，假设摄像头没有发生畸变，为了分析与计算方便，摄像机成像模型可以用针孔模型来近似。

为了描述这一成像过程[12]，首先定义三个坐标系：图像坐标系，摄像机坐标系和世界坐标系。三个坐标系之间的关系可以从图 2.3 中看到。

世界坐标系为客观世界的绝对坐标系，因为通常摄像机的位置不是固定的，所以需要一个世界坐标系为基准坐标系，用来定义每个摄像机坐标系的空间位置。理论上来讲，三维空间中任意一点都可以作为世界坐标系的原点（如图 2.3 中$O_W - X_W Y_W Z_W$）。

摄像机坐标系是以针孔模型的中心点为原点，以指向摄像机焦点的方向为 z 轴方向，以竖直向下为 y 轴建立的一个右手坐标系（如图 2.3 中$O - X_C Y_C Z_C$）。摄像机坐标系是随着摄像机的运动而运动的，每获取一帧的深度图像就会对应一个不同的摄像机坐标系，一般情况下三维地图的生成等都需要将每一帧所对应的摄像机坐标系转换到世界坐标系下进行处理。

图像坐标系是在成像平面上的二维坐标系。通常情况下，若以像素作为单位则称为图像像素坐标系（如图 2.3 中$O_2 - uv$），若以物体单位（毫米）为单位则为图像物理坐标系（如图 2.3 中$O_1 - xy$）。图像像素坐标系的原点一般都设在成像平面左上方的一点，而图像物理坐标系的原点一般都设在成像平面的中心位置。

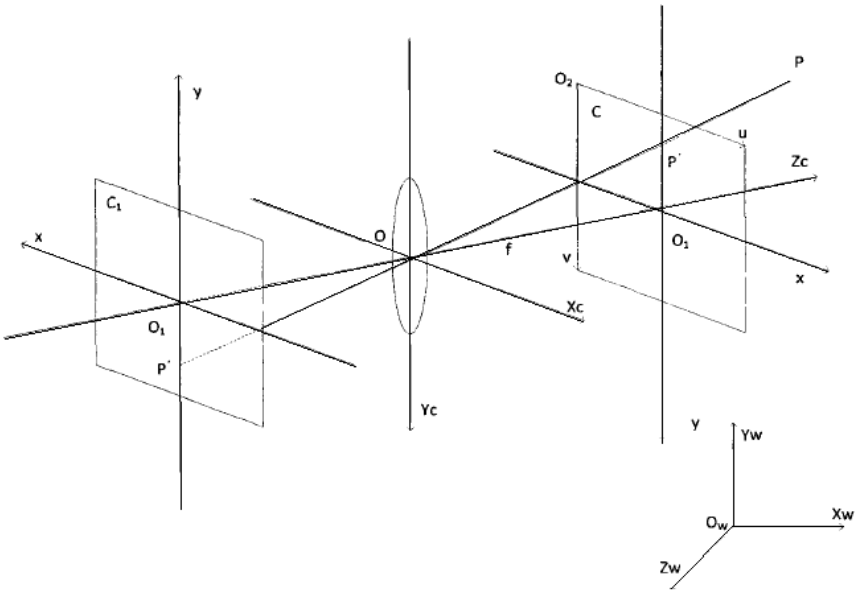


图 2.3 摄像机成像模型

对于图像像素坐标系，一个像素点坐标(u,v)可以认为是该像素在位于整个图像所构成数组中的行数与列数。假设图像物理坐标系的原点在像素坐标系下的坐标为(u_0,v_0)，且每一个像素点在 x 轴方向与 y 轴方向的物理尺寸分别为 dx 和 dy，则图像中每一个像素点在像素坐标系与物理坐标系之间的转换关系可以表示为：

$$x=(u-u_0)dx$$

$$y=(v-v_0)dy \tag{2.1}$$

若用矩阵形式则可以表示为：

$$\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}dx & 0 & -u_0dx\\0 & dy & -v_0dy\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix} \tag{2.2}$$

假设空间一点 P，在世界坐标系下的坐标为(X_W,Y_W,Z_W)，在摄像机坐标系下的坐标为(X_C,Y_C,Z_C)，而 P 在图像物理坐标系下的投影点坐标为(x,y)，由图 2.3 可得以下关系：

$$x = \frac{fX_C}{Z_C}$$

$$y = \frac{fY_C}{Z_C} \tag{2.3}$$

其中，f 为摄像头的焦距。

而世界坐标系与摄像机坐标系的转换关系可以通过三维空间中刚体的旋转和平移来表示：

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.4}$$

其中 $T = (t_x, t_y, t_z)^T$ 是三维平移向量， R 是旋转矩阵。

由式（2.2）（2.3）代入（2.4）进一步可得：

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.5}$$

式（2.5）中， dx, dy, u_0, v_0, f 只与摄像机内部结构有关，故称之为内部参数，而 R，T 完全是由摄像机相对于世界坐标系的位姿关系来决定，所以称之为外部参数。

由式（2.1）~（2.5）可知，若已知摄像机的内外参数以及像素点对应的三维空间坐标点距离摄像头的深度信息，由（2.5）就可以直接计算并唯一确定图像平面上的一个像素点对应于三维空间点的坐标。对于 Kinect 来说，由于它可以直接获取场景中物体表面各点的深度信息，所以只要已知 Kinect 摄像头的内外参数就可以获得三维空间的深度信息。这便是 Kinect 进行标定以及利用内外参数获得三维坐标点的理论依据。

2.3 三维点云的生成

由式（2.3）可知，在已知摄像机内部参数的情况下，若能够得到每一个像素点的深度信息，就可以求出成像平面上每一个像素点在三维空间对应的三维空间点坐标，而每一帧图像中的所有像素点全部投影到三维空间就会得到一个三维空间点集，这个点集就

构成了这帧图像所对应的当前摄像机坐标系下的点云数据（图 2.5），对应的彩色图像如图 2.4。

从图 2.5 可以看出，对于每帧图像（640*480 分辨率）来说，生成的点云数据量是非常大的，而且有一定程度上的冗余和干扰，有时这些冗余信息非但影响了运算处理的速度，对处理结果的准确性也并没有起到积极的作用。所以出于传输和处理的实时性要求，需要在传输前对每帧点云进行一个预处理，去除一些冗余信息来减小传输量与运算量，从而减少传输和处理的时间，提高整个系统的实时性。经过下采样预处理后的点云如图 2.6 所示，可以看到处理后的点云数据基本保留了采集场景的表面形状信息，与此同时数据量却大大减少，整体上有利于提高整个系统的处理性能和运算速度。

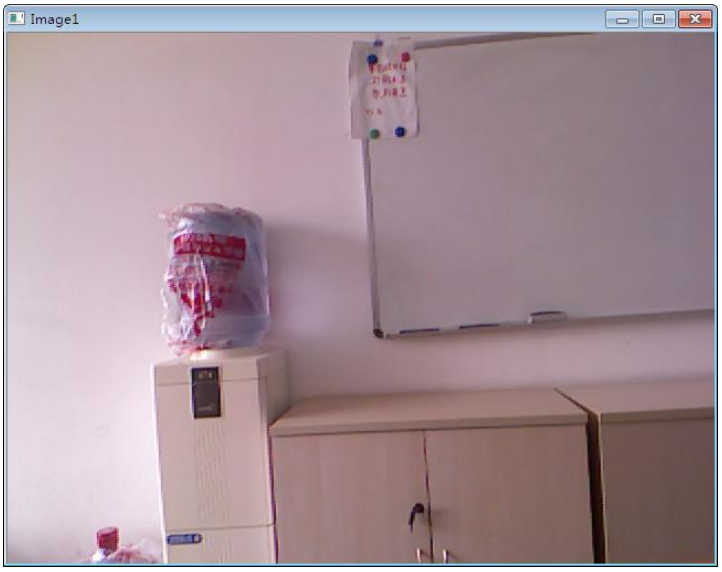


图 2.4 采集到的彩色图像

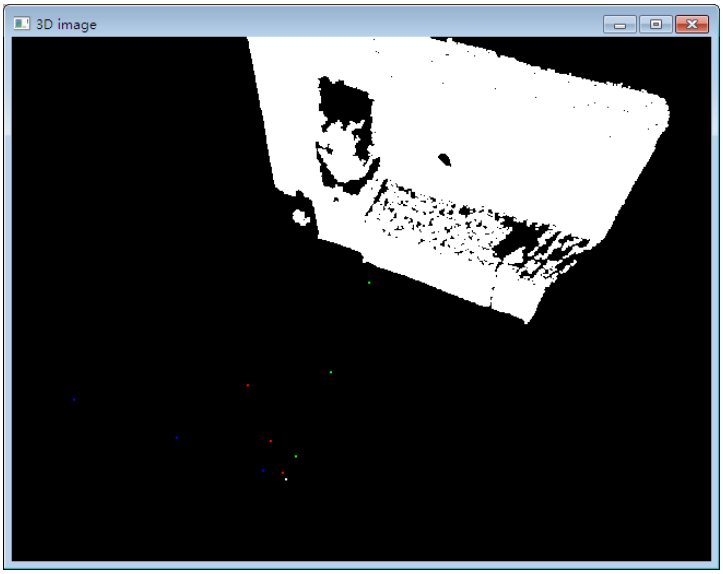


图 2.5 生成的三维点云

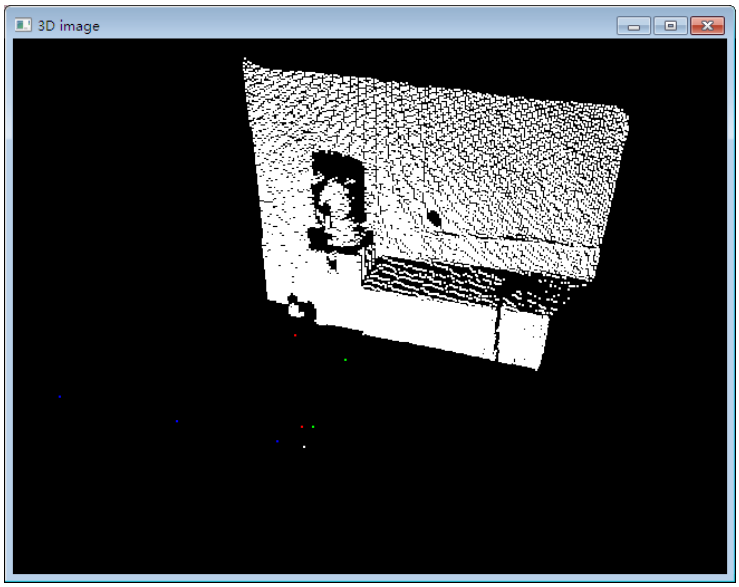


图 2.6 滤波后的三维点云

3 基于无线局域网传输的室内实时监控

3.1 无线传输技术

近年来随着技术发展，无线传输技术在各个领域都得到了广泛的关注。相比于有线传输而言，无线传输技术有其得天独厚的优势，如便捷性、适应性，以及不必布线，节约成本等等。在现在的社会中，人们对于“随时”与“随地”的需求进一步增加，各种移动网络和设备的遍布与普及充分地验证了这一点。但一般来讲，无线传输的传输效率往往不能与有线传输相比，这也使其在很多方面的应用受到了限制。不过由于技术的不断发展，无线传输的效率很大程度上也已经能满足实时性的要求，这也为其开拓了更广阔的应用空间。

下面对现在已经得到广泛应用的多种无线传输技术进行简单的介绍与比较[13]。

（1）CDMA，GSM 等 2.5G 无线传输技术。GSM（Global System for Mobile Communications，全球移动通信系统）技术的特点是频谱效率比较高，且 GSM 不仅提供空中接口，还提供了网络与网络之间或者实体设备之间的接口。而对于 CDMA（Code Division Multiple Access，码分多址）来说，相比而言在频率资源相同时，其移动网络容量较大，另外其软切换技术，一定程度上克服了硬切换技术中容易掉线的缺点，另外 CDMA 所提供的语音编码技术相较 GSM 而言，可以保证更好的通话质量。

（2）3G 无线传输技术。3G（3rd-Generation，第三代移动通信技术）数据业务所面向的重点是多媒体业务，所以对其传输速率有着一定的要求，具体要求是传输速率在高速移动时能达到 144kbps，静止状态时能达到 2Mbps。3G 网络的带宽在一定程度上已经能够保证多媒体业务中对实时性的要求，所以对于无线视频传输的应用较为适合[14]。

（3）Wifi 无线传输技术。如今在无线网络监控的领域，Wifi 数字视频监控的应用最为广泛。因为 Wifi 的带宽能达到 11Mbps 到 54Mbps，基本可以保证视频流的稳定持续传输，另外 Wifi 使用的是无线网桥，不存在布线等问题，且成本较低，扩充性比较强，基于上述优势，使其得到了广泛的应用。但与移动数据网络相比，Wifi 的覆盖范围十分有限，并不能达到与移动数据网络相当的覆盖率，也因此使其应用有了一定的限制。

（4）卫星无线传输技术。卫星通信是利用卫星中的转发器作为中继站，转发无线电波，来实现两个或多个地面站之间的通信。卫星通信具有覆盖范围广，抗干扰能力强，终端性能卓越等优势，但卫星信道租用费用高，维护技术偏高等问题也限制了其广泛应用。

由于实时性的要求和成本的限制，另外由于本文的应用场景是室内环境，而现如今多数单位和家庭都已实现了 Wifi 网络的覆盖，所以选择使用 Wifi 无线传输技术来实现实时监控中的无线传输部分。

3.2 TCP/IP 网络体系结构及其协议

TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/因特网互联协议）是一个协议族，它定义了各类设备该如何接入因特网以及网络中数据传输的标准，利用这组协议可以使具有计算机、调制解调器、Internet 服务提供者的用户访问和共享因特网信息。与 OSI（Open System Interconnect，开放系统互联）七层参考模型不同的是，TCP/IP 的参考模型由四层组成：应用层，传输层，网络层和链路层[15]（图 3.1）。

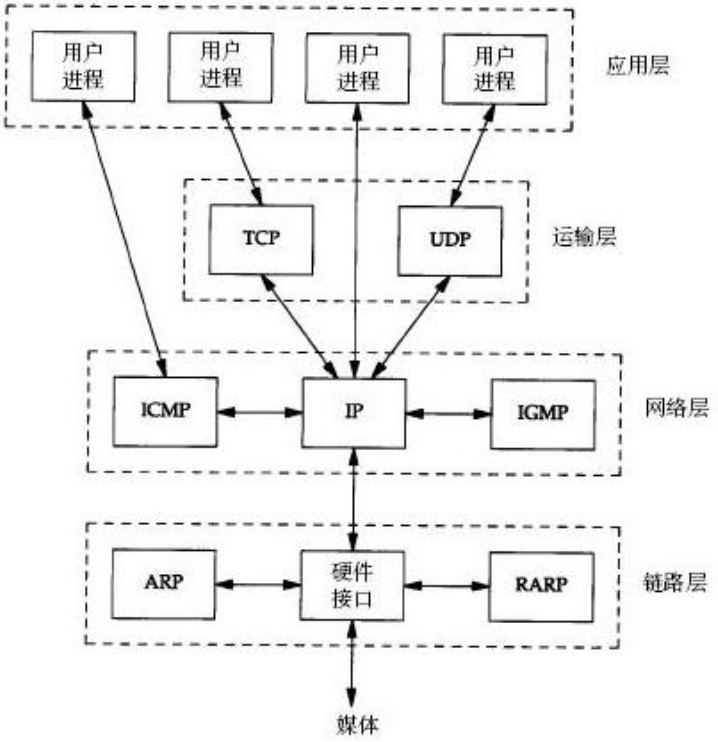


图 3.1 TCP/IP 参考模型

应用层向用户提供一些常用的应用程序，包括 SMTP（Simple Mail Transfer Protocol，简单邮件传输协议）、FTP（File Transfer Protocol，文件传输协议）、HTTP（Hypertext Transfer Protocol，超文本传输协议）、Telnet（用户远程登录服务）等协议，以及一些相关的服务。传输层主要是用来提供应用程序之间的通信，传输层的协议包括 TCP（Transport Control Protocol，传输控制协议）和 UDP（User Datagram Protocol，用户数据报协议）。其中 TCP 为两台主机提供可靠性高的数据通信，需要进行“三次握手”的

过程才可以建立连接，而 UDP 则是为应用层提供一种简单的数据报传输服务，没有提供任何可靠性的保证，也因而节省了开销，提高了通信速率。网络层则主要是负责相邻计算机之间的通信，并处理报文的路由管理，根据接收报文的报头信息等来确定报文的去向。而链路层则是管理网络的连接并提供网络报文的输入输出。

TCP/IP 实际上是一个相关协议族，并不仅仅代表了 TCP 与 IP 两个协议。其中 IP 协议主要是网际层的协议，主要是用来保证连接的，同时也被 TCP 和 UDP 所使用。TCP 是可靠的面向连接的传输层协议，能够建立端到端的可靠的字节流。UDP 是一个简单的，尽力转发的数据报协议，面向无连接，提供高效但不可靠的传输层服务。UDP 协议的突出优点便是高效率，不用建立及释放连接，节省了很多开销，常用于视频音频等多媒体数据的传输，因为此类传输中需要保证的是传输的实时性，而对于可靠性的要求并不如实时性的要求那么高。

基于此，本文中所涉及到的彩色图像流以及点云数据的无线传输所选用的是 UDP 协议，移动终端将经过处理的数据分割成数据包缓存在发送缓冲区，并另外开出单独的发送线程进行数据包的发送，这样一来数据的获取与预处理以及数据包的发送就可以在各自独立的线程中并行进行，有利于实时性的提高。同样的在上位机接收数据包的时候也是在独立的接收线程中，并将接收缓冲区的数据取出存到内存中进行后续的处理。

由于在 UDP 的实现过程中并没有任何握手连接或者检验重发的机制，所以在传输的过程中不可避免会存在一些丢包的情况。对于此类情况，本文采用了简单的回包机制进行解决，具体方法是接收端在接收每一个数据包后都给发送端发送回包，发送端接收到回包之后再进行下一个数据包的发送。这样一来，相当于在 UDP 中加入了一些 TCP 的思想，进行了最简单的接收确认的工作，基本可以解决前期出现过的丢包问题。

3.3 Winsock 网络通信实时传输技术

Winsock（Windows Sockets）网络接口是根据 U.C. Berkeley 大学 BSD UNIX 中流行的 Socket（套接字）接口制定的在 Windows 下运行的网络编程接口规范。Windows 上的应用程序可以通过调用 Winsock 的 API 来实现应用程序之间的通信，而 Winsock 则利用下层的网络通信协议来完成实际的通信工作。

套接字 Socket 是通信的基础和通信中某一个可以被命名和寻址的通信端点的抽象，也是支持 TCP/IP 协议网络通信的基本操作单元，一个正在运行中的套接字通常都有着对应的类型以及与其相关的进程。Socket 提供了一种发送和接收的机制。在 Windows

下的 Winsock 提供了一种应用程序的接口，可以方便地利用 TCP、UDP 等下层网络协议进行数据通信。一般情况下，套接字都是在某一种通信域（如 Internet 域）中进行通信，套接字之间交换数据时也是在同一个通信域中。

Socket 有两种类型[16]：流式 Socket 和数据报 Socket。通常而言，TCP 是面向连接的协议，使用的是流式 Socket，传输通常是可靠有序且无重复的，而 UDP 是面向无连接的协议，使用的是数据报 Socket，传输不能保证可靠有序和无重复。

Winsock 并不是一种网络协议，而是一种网络编程接口，在 Windows 下可以支持多种协议，较常使用的是 TCP/IP 协议。Winsock 编程主要分为两种方式：一种是面向连接的方式，对应的 TCP 协议（图 3.2），两个通信进程之间应先建立一种连接关系，其特点是通信较为可靠，对数据有校验和重发的机制，通常用作文件的传输；另一种是面向无连接方式，对应的 UDP 协议（图 3.3），通信双方之间不必要建立确定的连接关系，传输的最大数据量取决于具体的网络，因为缺少握手连接和校验重发的环节，所以数据在传输过程中有可能会丢失、重复或者无序等问题，但是通信速率较高，通常用于一些对数据可靠性要求不高的场合，如实时语音或视频传输等。

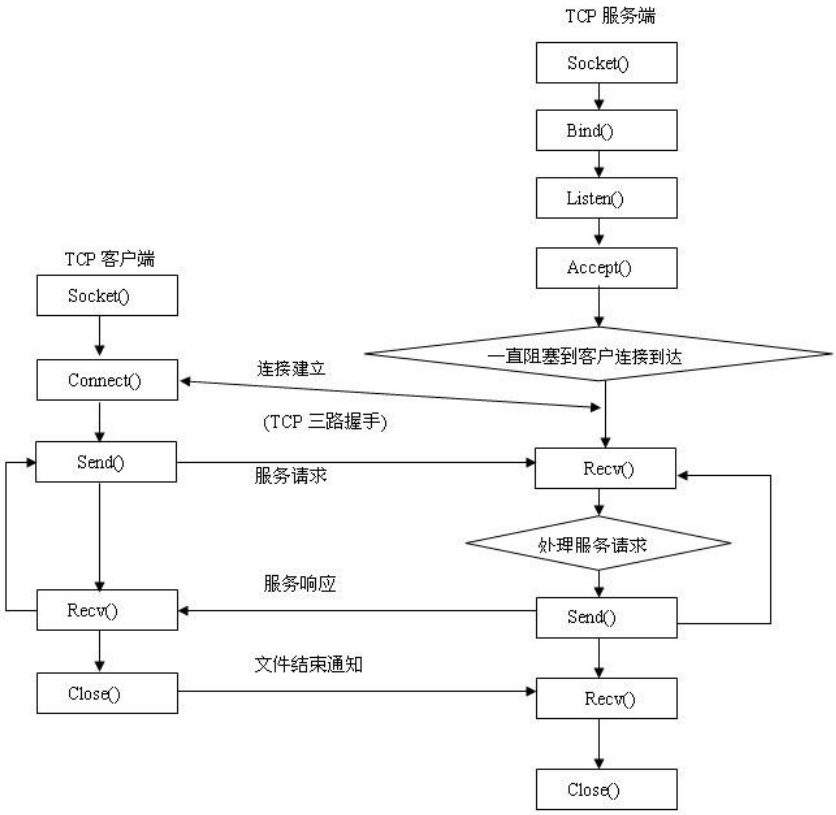


图 3.2 TCP 连接过程

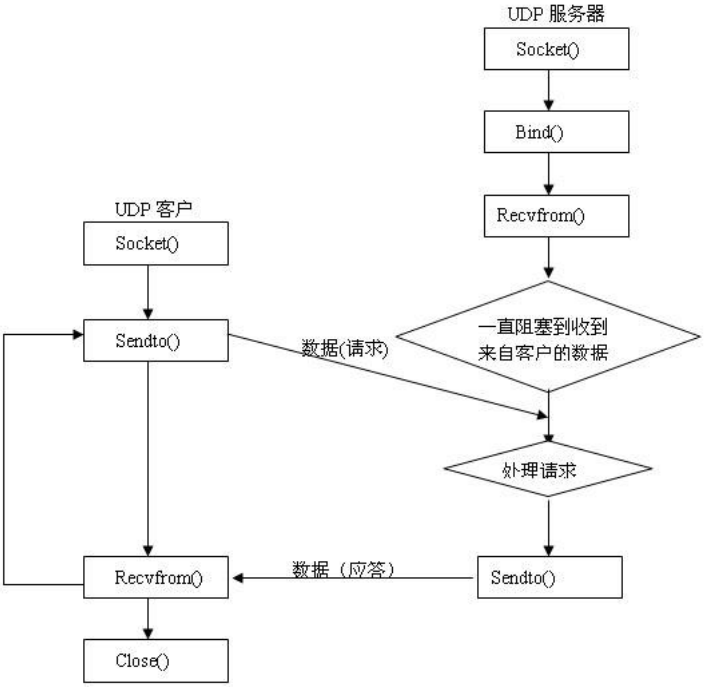

UDP 服务器
Socket()
Bind()
Recvfrom()
UDP 客户
Socket()
Sendto()
数据(请求)
一直阻塞到收到
来自客户的数据
处理请求
Recvfrom()
数据（应答）
Sendto()
Close()

图 3.3 UDP 连接过程

4 三维点云地图的生成

4.1 关于同时定位与地图创建

SLAM（Simultaneous Localization and Mapping，同时定位与建图）问题[17]的提出最初是针对移动机器人导航问题。未知环境下移动机器人导航需要解决的两个关键问题即机器人自身的定位（Localization）以及周围环境地图的创建（Mapping），这两个问题通常是相互关联的，因为如果机器人不知道自己的确切位置便无法创建准确的环境地图，而知道自己位置的前提便建立在已知环境地图的基础上，这就需要机器人在自身位置不确定的情况下，根据传感器所获取的信息进行位置估计，同时增量式地创建地图，并利用自己创建的地图进行自主定位与导航[18]。为了解决这个问题便有了同时定位与地图创建即 SLAM 理论。当前解决 SLAM 问题的方法大致分为两种：基于概率估计方法和非概率方法，其中很多关于卡尔曼滤波的方法如压缩滤波，完全 SLAM， FastSLAM 等就属于概率估计方法，SM-SLAM，扫描匹配，数据融合等是属于非概率估计方法，而目前的相关研究大部分都是基于概率估计的方法。

机器人对地图的创建实际是为了获得机器人所在环境的空间模型，所创建的地图通常用于机器人导航。在本文中，我们借鉴 SLAM 中地图创建中的有关思想和方法，利用移动终端传感器采集到的视觉信息来进行空间场景的复现。而在 SLAM 近些年来的发展过程中，也存在一些因素在限制其发展和应用。首先为了得到外部环境的信息，机器人必须要装有能够感知外部信息的传感器，因为机器人需要根据传感器获得的信息来进行位置的估计和增量式地图创建，而这些传感器通常都会受制于其自身在获取信息的过程中产生的测量误差。由于机器人的地图创建是一种增量式的过程，所以在这个过程中也导致了误差的不断累积，这也是 SLAM 一直面临的一个重大挑战和需要重点着手解决的问题[19]。

还有一个很难解决的问题就是场景中的匹配问题，即传感器在某个场景不同位置不同角度所获取的测量数据是否对应于真实三维空间中的同一物理对象。由于传感器在获取信息的时候执行的一种相对独立的过程，如实的对外部场景进行感知，所以如果不经过任何的处理过程，传感器并不具备对场景中某些特点进行识别的能力，也就无法得到两个在不同时刻所获得信息的对应关系。在本文中，Kinect 获取的每一帧图像都可以得到其对应的三维点云数据，而且每一帧的点云都在其独立的摄像头坐标系中，而不同两帧之间的对应关系是要经过后续的处理才可以得到。若要进行点云地图创建必须将每一

帧点云都配准到同一个坐标系即世界坐标系中，所以需要解决的问题就是找到空间中同一物理对象对应在每一帧点云中所描述的某些点，在此基础上才能完成点云的匹配和点云地图的创建。

4.2 基于 ICP 算法的点云匹配

因为 Kinect 在采集图像并生成点云数据的时候，是在每帧独立的摄像头坐标系下，而 Kinect 本身是处在不断运动的过程中，不同两帧点云之间是相对独立的关系，所以首要的问题就是将每一帧独立的点云数据统一在同一个坐标系即世界坐标系下。对于相邻的两帧数据生成的三维点云而言，会有一定范围的重叠区域，而对这部分重叠区域来说，理论上来讲它们在经过一定的旋转和平移变换之后是能够达到完全重合的，同时如果计算出使两帧点云数据完全重合的旋转矩阵和平移向量，也就相当于得到了两帧的摄像头坐标系之间的旋转和平移变换，即完成了两帧点云之间的匹配工作。基于此，本文中考虑应用 ICP（Iterative Closest Point，迭代最近点）算法来实现点云的匹配工作。

迭代最近点算法即 ICP 算法是近几年在点云配准中用的最为广泛的算法，由 Paul J. Besl 和 Neil D. Mckay 在 1992 年首次提出[20]，是一种基于自由形态曲面的配准方法，定义一个阈值，通过迭代来搜索最近点，最终完成多视图的拼合，通常可以用于三维重建中的三维点云或形状的匹配，而且不需要事先设定对应点。

ICP 算法的基本原理如下。假设三维空间 R3 中存在两个点集 P_d 和 P_m，这两个点集各自包含 n 个坐标点，分别为 $P_d=\{P_{d0},P_{d1},P_{d2}...P_{dn}\}$ 和 $P_m=\{P_{m0},P_{m1},P_{m2}...P_{mn}\}$，点集 P_d 中的点经过三维变换后可以与 P_m 中的点一一对应，即

$$P_{mi}=R\cdot P_{di}+T \tag{4.1}$$

其中 R 是三维旋转矩阵，T 是平移向量。

在 ICP 配准方法中，空间变换参数向量可表示为 $RT=\begin{bmatrix}q_0 & q_1 & q_2 & q_3 & t_0 & t_1 & t_2\end{bmatrix}^T$，其中满足约束条件：$q_0^2+q_1^2+q_2^2+q_3^2=0$。而旋转矩阵 R 与 RT 的对应关系为：

$$R=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{4.2}$$

ICP 算法流程如下：

（1）计算两个点集 P_d 和 P_m 的重心，并分别对点集进行中心化的处理生成新的点集。

（2）由新的点集计算正定矩阵 N，并计算 N 的最大特征值及其对应的最大特征向量，对应于空间变换参数向量 RT，进而得到旋转矩阵 R。

（3）由两个点集的重心点与旋转矩阵 R 确定平移向量 T。

（4）计算点集 P_d 经过旋转矩阵 R 与平移向量 T 变换后得到的点集 $P_d^{'}$，并计算两个点集之间的距离平方和值作为迭代判断数值。

（5）当满足一定条件后停止迭代，若不满足则重复迭代过程。

通过以上步骤就可以计算出旋转矩阵 R 以及平移向量 T，进而得到变换矩阵，将两个数据点集以最小的误差配准到同一个坐标系下。

但是利用 ICP 算法完成点云匹配时有两个重要的问题需要解决。

首先是当前处理器下算法运行速度的问题。根据 Kinect 采集的信息所生成的三维点云每帧的数据量是很大的，滤波前每帧数据量为 480*640 即 307200，经过滤波后每帧也有 19000 左右的数据量，而 ICP 算法的运算量是 O（N*N），所以尽管点云数据在传输会经过一定的滤波处理大幅地降低了数据量也排除了一些额外的干扰，但对于一般 PC 机的处理器来说每帧如此巨大的运算量也无法满足实时性的要求。通常来讲，针对 ICP 算法迭代过程运算速度无法达到要求这一问题，可以采用 kd-tree（k-dimensional tree）的方法来加速最近点的搜索过程进而提高迭代过程整体的运算速度。kd-tree 是一种数据结构，用来分割 k 维空间，主要可以用在多维空间的数据搜索等领域。

其次，ICP 算法能够实现精确配准的前提条件是，两个点集经过变换后能够完全重合，即点集中的每一个点与另一点集中的每一个点是一一对应的关系，也就是说，存在一种旋转和平移变换，使得某一点集在变换过后可以与另一点集实现完全重合。而在本文的应用环境中， Kinect 的位置是在不断的变化中，所以相邻两帧点云所对应的真实环境中的场景是一种存在重叠部分但并不完全一致的情况，这样情况下 ICP 的配准效果往往会大打折扣。因为 ICP 算法过程中，两个点集中的每个点都会参与运算，而这种情况下，其中有很多的点其实并不能在另外一个点集中找到对应点，对于 ICP 算法来说，两个点集中的点并没有严格的对应关系，只是在不断迭代的过程中使整体的配准能达到

一个最优的状态，于是这一部分无法在另一点集中寻找到对应点的坐标点一定程度上就变成了一种干扰，使得迭代过程中将它们也纳入一种最优状态的考虑中，影响了配准的效果。

为了解决这两个问题，除了要在配准前对点云进行滤波处理，降低点云数据量之外，还需要运用一定的方法对两个点集中的点进行筛选，过滤掉那些在另一点集中找不到对应点的点，使得筛选出来的两个点集尽量满足变换后能够重合的条件，从而提高 ICP 算法的准确性。这就涉及到两个点云中对应点或对应区域的选取，这也是上文所提到的 SLAM 过程中经常会遇到的难题之一，也就是两个不同视图中的某些点或区域是否对应于真实三维空间的同一物理对象。因为若要做到筛选出的两个点云子集数据能满足 ICP 算法的变换后重合的条件，就必须要保证这两个点云子集在三维空间中的对应关系，只有这种对应关系得到满足，才有可能会实现两个点云子集中的点达到一一对应或近似一一对应的条件。考虑到在利用 Kinect 获取到的信息生成点云数据的时候，实际上通过计算得到的点云数据是有序的，即点云中的每一个点都对应于彩色图像中的一个像素点，所以本文采用的解决办法是在彩色图像上进行特征点的提取和匹配，再用这些提取到的特征点来定位点云中的对应区域。

4.3 图像特征提取与 SURF 特征检测算子

由上文可知，若想要找到两帧点云的对应区域，可以对相邻的两帧图像进行特征点的检测与匹配，再将这些特征点投影到三维空间的坐标点云中来确定两帧图像对应的点云区域。要完成这一任务，要求特征提取的过程有较强的鲁棒性，对于尺度和旋转都具有一定的不变性，而且能够基本满足实时性的要求。基于以上考虑，本文拟采用 SURF 特征检测算子来完成对特征点的提取。

在 SURF 算法之前 Lowe 等人提出的 SIFT（Scale-invariant Feature Transform，尺度不变特征转换）算法也是一种鲁棒性较好的尺度不变的特征描述方法[22]，广泛应用于人脸识别、图像拼接、图像配准等领域。但是 SIFT 算法的计算数据量大，复杂度高，计算耗时长，适合于可以进行离线数据处理的系统，并不适于实时性的方法和系统。2006 年 Bay 等人在 SIFT 的基础上首次提出了 SURF（Speeded Up Robust Feature）算法。SURF 算子由 SIFT 算子改进而来，最大的特征在于采用了 haar 特征以及积分图像（Integral Image）的概念，在基本保证了 SIFT 算法原有的鲁棒性强以及尺度不变性等特点的基础上，大大加快了程序的运行速度，为实时性运行的系统提供了一个更好的选择。

与 SIFT 类似，SURF 特征点检测依然基于尺度空间的理论，假设图像中一点 $p(x,y)$，在尺度 σ 上的 Hessian 矩阵定义为：

$$H(p,\sigma)=\begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{xy}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix} \tag{4.3}$$

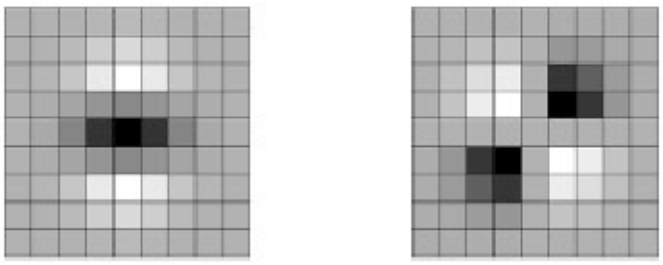
其中 L_{xx} 的含义是高斯滤波二阶导数 $\frac{\partial^2}{\partial x^2}g(\sigma)$ 在 $p(x,y)$ 点处与整个图像卷积的结果，L_{xy} 与 L_{yy} 的含义与 L_{xx} 类似。

Bay 等人在论文中提出用方框滤波近似代替二阶高斯滤波，用积分图像来加快卷积的计算速度，用盒子型滤波器对实际的滤波器进行近似（图 4.1，图 4.2）。积分图像计算的是图像中某一个矩形区域内的像素和，例如图像 $I(x,y)$ 在像素点 $p(x,y)$ 处的积分图像定义为

$$I_{\Sigma}(p)=\sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(i,j) \tag{4.4}$$

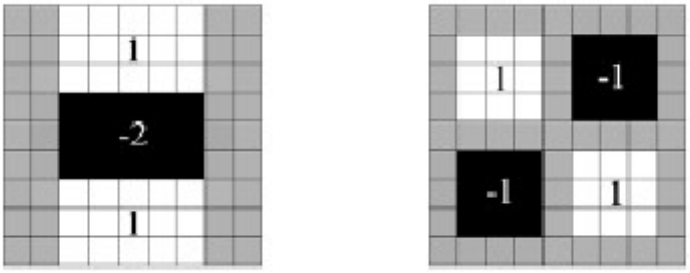
如果一个图像的每一个像素点的积分图像都已经计算出来，则对于这个图像来说，计算任意一个矩形区域内像素和的时候，只需要利用这个矩形区域的四个顶点对应的积分图像，进行三次加（减）法的运算便可以完成，因此在总体上便提高了效率。

而对于盒子滤波器来说，可以假定其每一个连续区域内的权值是相同的，这样一来，计算盒子滤波器与图像在某点处的卷积等价于计算原图像在该点的积分图，而积分图的计算如上文所说只需要三次加（减）法就可以完成。大大减少了计算量。



（a）计算L_{yy} （b）计算L_{xy}

图 4.1 计算 $L(p,\sigma)$ 的实际模板[21]



（a）计算L_{yy} （b）计算L_{xy}

图 4.2 计算$L(p,\sigma)$的近似模板（Box Filter）[21]

若用D_{xx}，D_{yy}，D_{xy}表示采用近似模板后得到L_{xx}，L_{yy}，L_{xy}的近似值，则 Hessian 矩阵的行列式可以近似表示为：

$$\det(H) = D_{xx}D_{yy} - (0.9D_{xy})^2 \qquad (4.5)$$

其中 0.9 是通过实验得到的一个调节参数，理论上此调节参数的值与尺度 σ 有关，但一般都被设置为常数，实验证明对结果不会有太大的影响。

在构建尺度空间时，SIFT 是通过不断对图像进行高斯平滑与降采样的过程来得到金字塔，在这个过程中高斯滤波器的大小是不会变的，而与 SIFT 不同的是，SURF 只是改变滤波器的大小，图像本身不进行变化（图 4.3）。

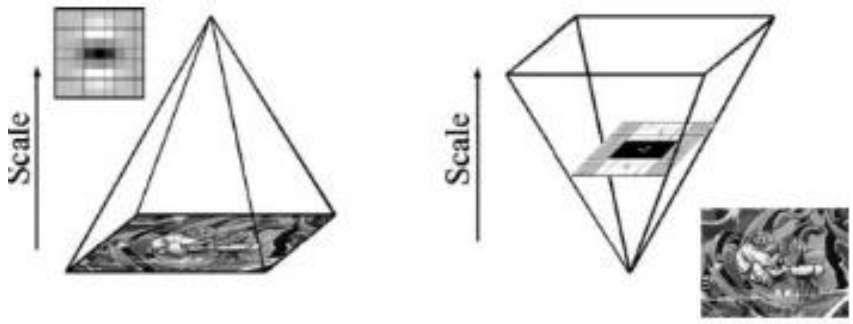


图 4.3 构建尺度空间（左 SIFT，右 SURF）

与 SIFT 相类似，在尺度金字塔的每一阶中，选择四层的尺度图像，如果图像尺寸远大于模板大小，则继续增加阶数。用 Hessian 矩阵求出极值后，在3×3×3邻域内进行非极大值抑制，然后在尺度空间和图像空间中进行插值运算，得到稳定的特征点位置及所在的尺度值。

为了保证旋转不变性，需要对每个特征点进行主方向的确定。具体方法是计算一定半径（6σ）内邻域内的点在 x、y 方向的 Haar 小波响应，再将高斯权重系数加在这些响

应上，使得离特征点越近的响应贡献越大，离特征点越远的响应贡献越小，然后遍历整个圆形区域的60°范围内响应相加从而形成了新的矢量，这样一来最长的矢量方向即可确定为该特征点的主方向。之后便可以沿主方向构建一个正方形，并针对每一个特征点，形成一个 64 维的描述向量。

描述向量也包含了特征点邻域的信息，用向量的最近邻匹配法便可以完成两帧图像特征点的匹配。

4.4 基于 SURF 特征提取与 ICP 算法的点云匹配

如前所述，在运用 ICP 算法进行点云匹配的过程中，一是由于数据量比较大，以致于运算量也随之变大，导致 ICP 算法中的迭代过程耗时过长，所以存在实时性不能满足的情况；二是两帧之间那些无法重叠部分的点云由于找不到其对应点，所以在 ICP 算法运行的过程中，这些点其实并没有对结果产生任何帮助，反而会成为匹配过程中的一种干扰，影响匹配的效果。

因此在本文中所采用的方法是，首先运用 SURF 特征提取算子对 Kinect 采集到的彩色图像进行特征的提取以及匹配（图 4.4），然后将这些特征点映射到三维空间中，找到相应点云中的对应点便得到了两帧点云之间的对应关系。得到这个对应关系是为了对相邻两帧点云中的点进行筛选，在减少数据量与运算量的同时，剔除那些非重叠区域的点，提高匹配过程的准确性。但并不是直接对这些特征点运行 ICP 算法来进行点云的匹配，原因如下：一是相对于每一帧整个的点云数据来说，特征点数量非常少，最多的情况下也只有 500 到 1000 的数量，并且利用 SURF 特征检测算子提取到特征点的数量直接取决于当前帧中所采集到的场景，如果场景发生变化则特征点的数据可能随之发生很大变化，具有相当程度的不确定性，不利于整个系统实现过程的稳定性；二是利用 SURF 特征检测算子检测到的特征点基本都处于图像中变化较为剧烈的区域，如角点或物体的边缘区域，而在这些区域内一般深度信息也会存在剧烈的变化，甚至有可能不能够在这些像素点获得有效的深度信息，也就意味着在将图像上检测到的特征点投影到三维空间的时候，有可能不能对应到有效的三维坐标点。综上所述可知，直接用特征点在三维点云中的对应来进行点云匹配的方法是不可取的。

对此本文所采取的解决办法是，首先对匹配好的特征点进行筛选，得到匹配结果较为理想的特征点，再以这些特征点为中心，在两帧图像上分别选取适当大小的区域，作为每帧点云的一个子集，即完成每帧点云数据的筛选过程。而选取出来的这两个子集，

由于有匹配好的特征点信息的限制，所以可以作为两个对应点云来运行 ICP 算法，从而得到这两个点云子集之间旋转和平移的变换关系。将得到的这种变换关系应用于整个点云，即可完成点云数据的匹配（图 4.5，图 4.6）。

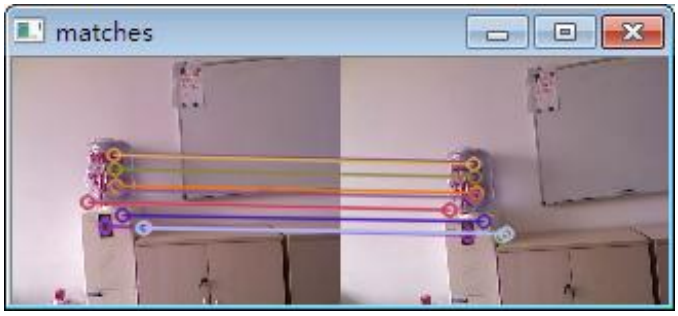


图 4.4 特征点提取与匹配结果

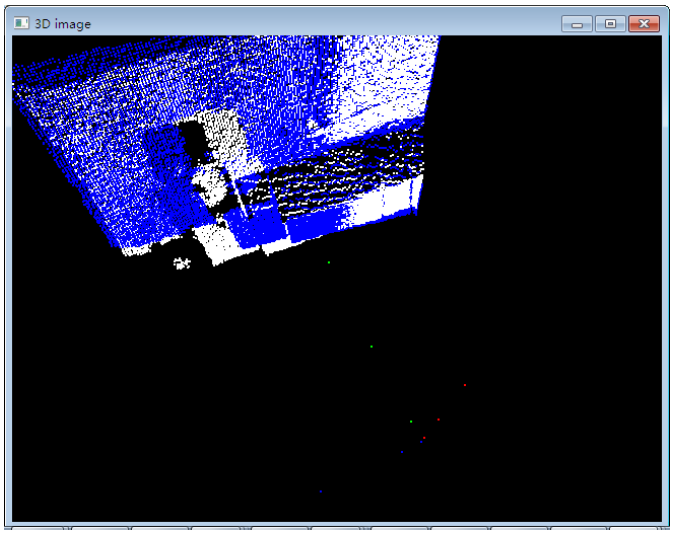


图 4.5 点云匹配前

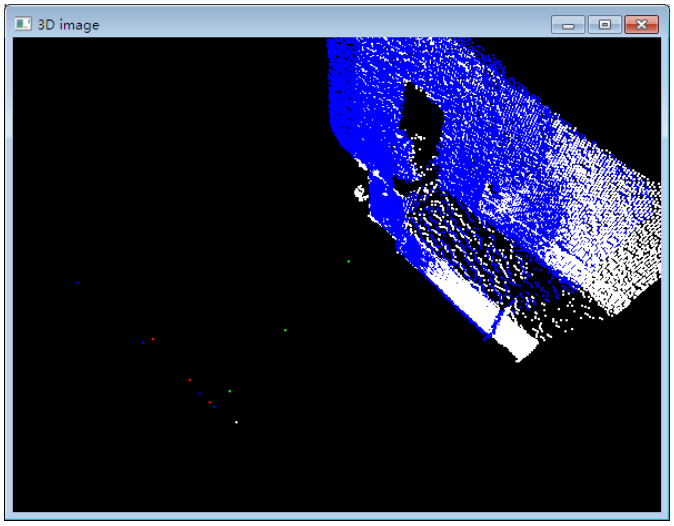


图 4.6 点云匹配后

以 SURF 所检测到的特征点为中心选取特定区域来进行匹配，充分利用了 Kinect 在图像获取同时得到深度信息的优势，完成了特征点向三维空间的投影以及对应点云区域的选取，相比于直接运行匹配算法或者直接用特征点来进行匹配有以下好处：

（1）减少了运行时间，提高了实时性。对于 ICP 算法来说，寻找最近点的迭代过程的耗时通常会严重影响系统整体的实时性。由于选取出的区域所包含点云的数据量与整个点云的数据量相比有很大程度的减少，甚至已不是一个量级，运算量也因此成倍地降低。

（2）消除了在寻找图像特征点所对应点云中特征点的过程中一些不确定因素，相比直接利用特征点进行匹配具有更强的可靠性。如上文所述，利用 SURF 特征检测算子在彩色图像上提取的特征点，在数量上不能满足要求，而且提取到的特征点若直接投影到三维点云中，通常都会处于深度信息剧烈变化的区域，具有很大程度的不稳定性。所以本文中所采用的根据已匹配特征点位置来选取两帧点云对应区域的方法，与直接利用特征点进行匹配的方案相比可靠性更好。

（3）由于选取对应区域的时候是以特征点为中心点，所以可以近似认为选取到的两个对应区域能够变换后完全重合。这样一来，就近似满足了 ICP 算法的另外一个条件，即两个待匹配点集中的点存在一一对应的关系，也就可以排除在用整个点云做匹配时总会存在找不到对应点的一部分非重叠区域的干扰，运行 ICP 算法的时候就可以认为将可能产生干扰的点排除在外，相比直接运行 ICP 算法具有更高的准确性。

5 系统设计与实现

5.1 系统构成

系统由移动终端和上位机构成，由一台 PC 机连接 Kinect 作为视觉信息采集传感器来进行模拟移动终端，上位机是由另一台 PC 机构成。终端在室内移动的时候，上位机实时显示终端所采集到的当前帧的彩色图像，同时利用接收到的三维点云信息来进行场景的复现，终端和上位机之间通过无线局域网来完成数据的传输。

移动终端完成的工作首先是图像数据的采集，包括当前帧的彩色图像以及对应每个像素点的深度信息，其次是完成数据的预处理工作，例如点云的生成与滤波处理，图像的预处理以及无线传输前的准备工作。

而上位机完成的则是利用接收到的移动终端处理的信息进行当前帧彩色图像的显示，即实现实时监控的功能，以及利用生成的点云信息与彩色图像信息一起，进行每帧点云的匹配，从而实现三维点云地图的创建工作，即实现场景复现的功能。在上位机上呈现给用户的主要有两个，一是实时监控的窗口，用户通过这个窗口可以实时地看到当前帧的彩色图像，实现实时监控的功能，二是场景复现的窗口，显示的是当前帧与之前若干帧所形成的摄像头采集过的区域的三维场景信息，实现场景复现的目的。

在系统功能实现的时候，移动终端可以利用 Kinect 摄像头针对室内的某一区域进行扫描，扫描的同时将获得的信息进行处理并传输给上位机，最终由上位机完成最终的处理以及场景复现，并呈现给用户进行查看。

5.2 系统软件搭建平台与流程

本文选用 C++语言，并利用 Visual Studio 平台进行开发，在对 Kinect 进行开发时选用了 OpenNI2（Open Natural Interaction，开放式的自然交互）。与 OpenNI1 不同的是，OpenNI2 的环境下只能选择运用微软官方的 Kinect for Windows SDK（Software Development Kit，软件工具开发包）来进行驱动了。

Kinect for Windows SDK 是由微软开发的用于 Kinect 驱动的开发包，是一个非商业授权许可，且只能在 Windows 7 操作系统下使用，开发环境是 Visual Studio 2010 及以上版本。Kinect SDK 支持三种开发语言：C++、C#以及 VB。Kinect for Windows SDK 可以提供非常复杂的软件库与工具，帮助开发者进行基于 Kinect 自然输入，信息捕获及事件反应等开发工作。

在 Kinect 信息捕获与处理中使用了 OpenNI。OpenNI[23]定义了一些开发开放式自然

交互程序的 API，是一个多语言跨平台的框架。OpenNI 是非盈利，开源的且具有良好的兼容性，主要是为了形成标准的 API，搭建音频和视觉传感器之间的通信标准，使得开发人员进行开发的时候可以不用考虑传感器相关的一些细节，而是直接利用传感器输出的标准化数据来进行处理。OpenNI 的框架主要分为三层，底层是捕获场景音频和视频信息的硬件设备，中间层代表的是传感器和中间组件进行交互的通信接口，而顶层则是执行自然交互的应用软件。OpenNI 提供多种的生成器，可以提供多种的功能，针对 Kinect 设备主要涉及的生成器有：映射生成器，用来提供产生任何映射数据生成器的基本接口功能；深度数据生成器，用来产生深度数据对象；彩色图像生成器，用来产生彩色图像映射对象；红外生成器，用来生成红外映射对象。

在进行彩色图像特征点提出的过程中使用了 OpenCV[24-25]。OpenCV 是由 Intel 在 1999 年建立，是基于 BSD 许可证授权的开源发行的跨平台计算机视觉库，可以运行在多种操作系统上，也提供了多种语言的接口，实现图像处理和计算机视觉方面的一套成熟的通用算法接口，可以通过调用简单的计算机视觉的框架来帮助用户建立比较复杂的视觉应用。

5.3 系统实现

系统的移动终端与上位机分别打开套接口，利用 Windows 中的 Winsock 进行通信，通信过程运用的是面向无连接的 UDP 协议。

在信息的发送端，Kinect 作为视觉信息采集设备，在进行过初始化工作之后，先是以字节流的形式采集到每一帧的彩色图像数据流以及深度数据流，再以数组的方式进行暂时的存储，并利用深度信息，根据前面章节所说明的计算机视觉相关理论完成空间三维坐标点的生成，这样每一帧的所有坐标点即构成了一帧初始的点云数据。在终端处理器中还需要完成的工作有点云数据的滤波处理（如去除噪声点等），这样采集到的视觉信息便经过了移动终端处理器的处理形成待发送的数据包暂存在发送缓冲区内进行发送。发送的过程是在单独打开的线程中完成的，发送线程是在 Kinect 初始化之后打开并完成一系列的套接字初始化的工作。将发送过程放在单独的线程中是为了提高整个系统的实时性，因为对于一个进程而言，各个线程是并行工作的，这样可以使数据发送与数据传输前的预处理过程同时进行，不会因为数据包的发送以及回包的接收而导致整个进程的阻塞从而影响传输与处理的速度。

上位机负责对数据包进行接收以及后续的处理及显示工作。与发送端类似，接收端

的接收过程与数据的处理过程也是处于单独的两个线程中，也就不会因数据的接收以及回包的发送而导致进程的阻塞。

在数据接收端完成的两个重要过程就是用 SURF 特征检测算子进行特征提取的过程，以及运行 ICP 算法进行点云数据匹配的过程。其中，特征提取的过程是在数据包接收的线程中进行的，因为在发送端会对每帧的数据进行分割，所以在数据包接收后有一个重新组合成一帧完成数据集合的过程，在这个过程完成之后就对接收到的一帧完整的彩色图像进行特征提取，并与上一帧图像提取到的特征进行匹配，并存储下已匹配好的特征点坐标。而与此同时，ICP 算法的运行是在主线程中进行的，在 ICP 进行之前需要先利用提取到的特征点坐标来筛选出相邻两帧点云数据中运行 ICP 算法所需要的对应区域，再对此区域内的所有点运行 ICP 算法来进行匹配的过程，从而得到两帧点云之间的旋转与平移变换关系，即一个 3×3 的旋转矩阵和一个 3×1 的平移向量构成的变换矩阵，将此变换矩阵作用于整幅点云数据便完成两帧点云的匹配。

当成功完成了两帧点云的匹配，将匹配的过程不断地在连续帧中的相邻帧进行，就可以完成场景地图的生成工作。如图 5.1 是连续六帧点云数据的配准过程，可以看到随着每帧点云数据的加入，整个点云地图中的点更加密集，所覆盖的场景也更大一些。

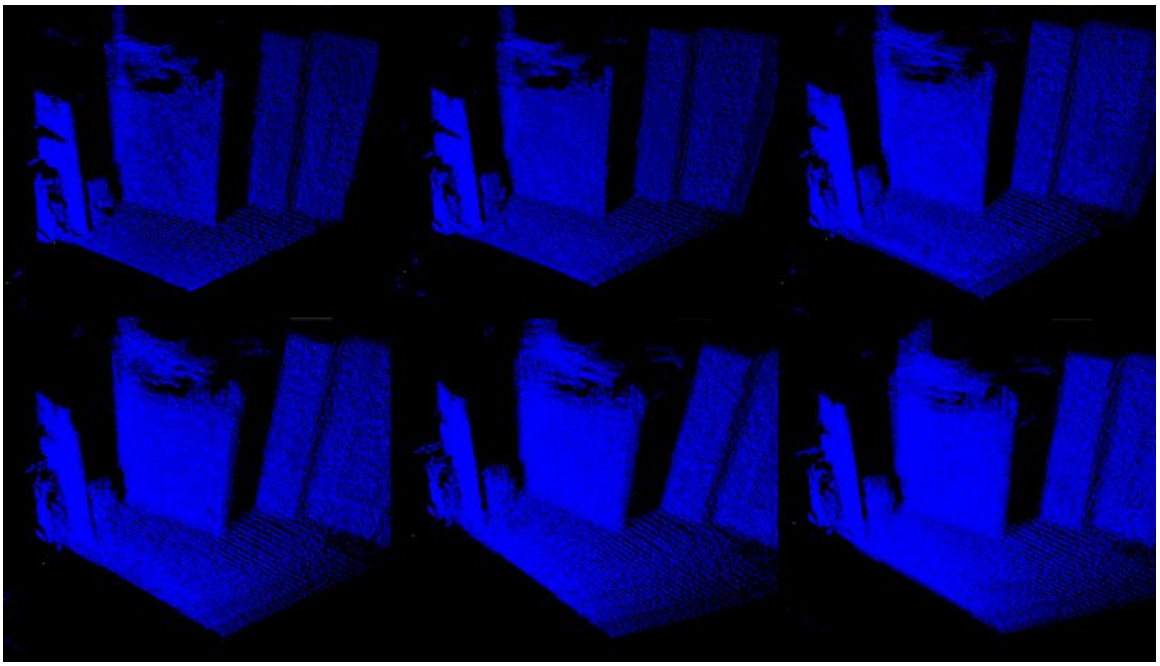


图 5.1 连续六帧点云的配准过程

图 5.2 为连续采集某一区域获取到的四帧彩色图像截图，图 5.3 为扫描后生成的该场景对应全局点云地图，因为全局地图生成的过程中是将配准后的每帧点云加入生成的全局地图，所以点云会随着地图的生成而变得越来越密集。图 5.4 是对该点云地图进行一次降采样后的结果，可以更清楚地看到配准效果。

图 5.2 采集过区域场景

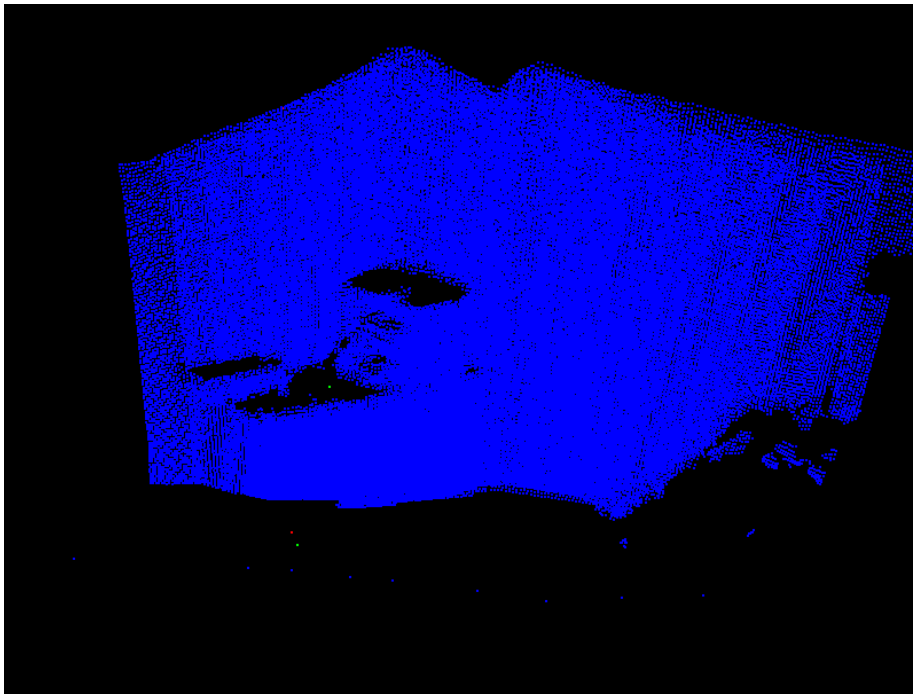


图 5.3 生成的全局点云地图

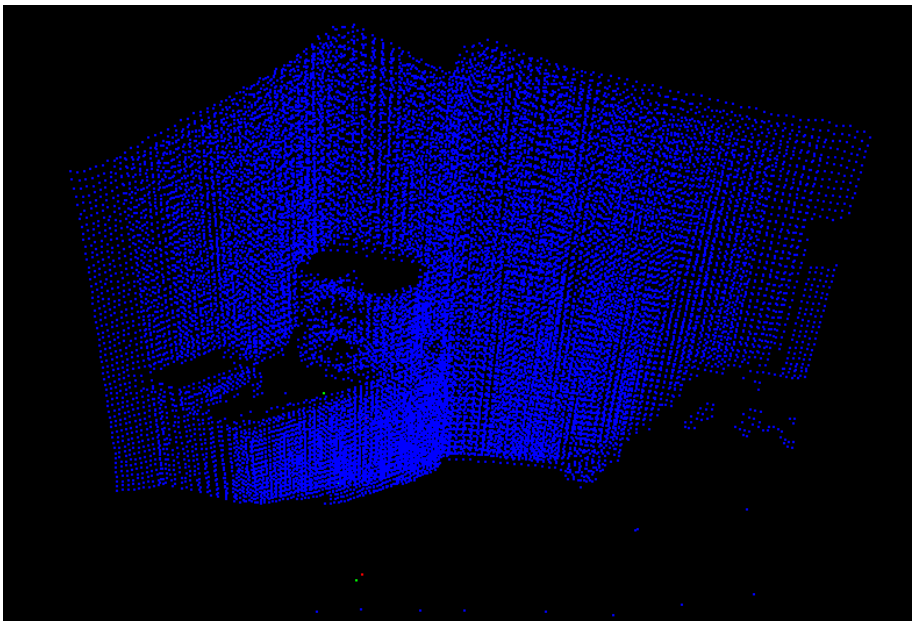


图 5.4 生成的全局点云地图（降采样后）

6 总结与展望

本文在室内场景实时监控的基础上，借鉴移动机器人 SLAM 的有关思想与方法，对监控场景进行三维点云重建，使用户在上位机端能看到当前帧的场景的同时，也可以看到实时增量式生成的点云地图，实现了已采集区域的三维场景记忆和重现的功能。

本文在实现的过程中主要的工作与进展有：

（1）视觉信息的采集与点云数据的生成。视觉信息的采集是通过 Kinect 来完成的，Kinect 的特点是在采集到彩色图像的同时可以得到每个像素点的深度信息，利用这些深度信息加之计算机视觉相关理论便可以直接进行三维点云数据的生成。

（2）彩色图像以及点云数据的无线传输。传输过程是通过 Windows 中的 Winsock 建立套接口，使用 UDP 协议来完成的。

（3）SURF 特征检测算子对彩色图像进行特征提取，利用提取到的特征点来定位点云中的对应区域，之后再运用 ICP 算法来完成点云数据的匹配。这样做的原因是在减少数据量和运算量的同时排除一些变换后无法重合区域内点的干扰，提高匹配的准确性。

另外，由于个人水平以及时间的限制，本文的实现过程还存在一些不足之处，有待完善和提高。例如在进行特征点提取的时候，并不能保证每帧场景都可以提取到合适的特征点，在本文中为了避免一些不必要的错误，在检测不到特征点的特殊区域，则将此帧图像的中心点像素作为对应区域选取的中心点，但这样一来就造成了一些不必要的误差。

除此之外，利用 SURF 进行特征点的提取以及进行特征点匹配的过程，还有 ICP 算法的运行过程都是相对比较耗时的，虽然已将其分开在单独的两个线程中运行，但系统整体的实时性也不能达到很高的要求，这一点仍有待提高。

参考文献

[1] 葛广英. 实时监控技术的发展历程和发展趋势 [J]. 电视技术, 2000, (10): 62-64.

[2] M. P. de Albuquerque, E. Lelièvre-Berna. Remote Monitoring over the Internet [J]. Nuclear Instruments and Methods in Physics Research, 1998, 412(1): 140-145.

[3] S. C. Hui, F. Wang. Remote Video Monitoring over the WWW [J]. Multimedia Tools and Applications, 2003, 21(2): 173-195.

[4] S. Pankanti, R. M. Bolle, A. Jain. Biometrics: The Future of Identification [J]. Computer, 2000, 33(2): 46-49.

[5] 张涛. 大尺度环境移动机器人同时定位与地图构建算法的实现 [D]. 青岛: 中国海洋大学, 2011.

[6] O. Aycard, F. Charpillet, D. Fohr, J. Mari. Place Learning and Recognition Using Hidden Markov Models [J]. Intelligent Robots and Systems, 1997, 3: 1741-1746.

[7] T. Bailey, H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II [J]. IEEE Robotics and Automation Magazine, 2006, 13(3): 108-117.

[8] 张荻. Kinect 应用领域的探讨 [J]. 物流工程与管理, 2012, 34(6): 39-41.

[9] J. Webb, J. Ashley. Beginning Kinect Programming with the Microsoft Kinect SDK [M]. U.S: Apress, 2012: 9-22.

[10] K. Boulos, B. Blanchard, C. Walker. Web GIS in Practice X: A Microsoft Kinect Natural User Interface for Google Earth Navigation [J]. International Journal of Health Geographics, 201l, 10(45): l-14.

[11] J. Salvi, J. Pages, J. Batlle. Pattern Codification Strategies in Structured Light Systems [J]. Pattern Recognition, 2004, 37(4): 827-849.

[12] 马颂德, 张正友. 计算机视觉一计算理论与算法基础 [M]. 北京: 科学出版社, 1998: 20-30.

[13] 范迪才, 荣文广. 远程实时监控技术探讨 [J]. 华北电力技术, 2009, (9): 9-12.

[14] 柴远波, 郭云飞. 3G 高速数据无线传输技术 [M]. 北京: 电子工业出版社, 2009: 24-31.

[15] T. Parker, M. Sportack. TCP/IP 技术大全 [M]. 北京: 机械工业出版社, 2000: 1-28.

[16] 施炜, 李铮, 秦颖. Windows Sockets 规范及应用—Windows 网络编程接口 [M]. 北

京:北京电子工业出版社, 1997: 35-89.

[17] A. Nüchter. 3D Robotic Mapping-The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom [M]. U. S: Springer, 2009: 29-69.

[18] G. Dissanayake, R. Newman, S. Clark, H. F. Durrant-whyte. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem [J]. IEEE Transactions on Robotics and Automation, 2001, 17(3): 229-241.

[19] 罗荣华, 洪炳铭. 移动机器人同时定位与地图创建研究进展 [J]. 机器人, 2004, 26(2): 182-186.

[20] J. Besl, D. McKay. A Method for Registration of 3D Shape [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239-256.

[21] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool. SURF: Speeded Up Robust Features [J]. Computer Vision and Image Understanding, 2008, 3951: 346-359.

[22] D. G. Lowe. Distinctive Image Features from Scale-invariant Keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[23] PrimeSense OpenNI [EB/OL]. http://www.primesense.com/.

[24] 贾小军, 喻擎苍. 基于开源计算机视觉库OpenCV的图像处理 [J]. 计算机应用与软件, 2008, 4: 276-278.

[25] G. Bradski, A. Kaebler. Learning OpenCV [M]. O' Reilly, 2008: 56-76.

致谢

在本人本科毕业设计的完成过程以及毕业论文的写作过程中，得到了很多人的悉心教导和热心帮助，才使我得以顺利完成毕业设计的工作，并在此期间学到很多知识和方法。

首先要感谢在毕设期间一直给予我指导的两位老师，南开大学信息技术科学学院的苑晶老师和北京航空航天大学电子信息工程学院的李洪革老师。感谢苑晶老师在整个毕业设计期间所提供的工作上的指导和帮助，以及为我提供良好的实验环境来完成工作。感谢李洪革老师在我整个毕业设计中对工作流程的把握以及对答辩工作的指导。

也要感谢在北航四年来给过我教育和指导的所有老师和辅导员。感谢各位老师引导我走进了这一片曾经未知的专业领域，并且毫无保留地分享他们的专业知识以及人生感悟，必将终生受益。感谢 3902 大班的所有辅导员，重要通知日常琐事，无私地助我们四年的生活学习。

其次要感谢南开大学智能信息处理实验室的师兄师姐们，感谢他们在毕设的进行过程中以及生活上给予的热心帮助，在我工作遇到困难时也能够不吝赐教，使我尽早熟悉了新的环境，顺利完成毕设任务。也要感谢在南开大学临时宿舍 409 的各位室友在生活上的帮助和鼓励，使我能够在陌生校园快速找到归属感。

另外还要感谢北航 3902 大班尤其是我们 26 班的所有同学以及 417 宿舍各位姐妹四年的陪伴和帮助，是他们让我在北航的生活足够充实快乐，也使我收获到学习之外的更多东西。

最后，感谢父母家人以及各位朋友一直以来的关心和照顾。

谨以此，献给所有给予我热心帮助的老师和同学，献给所有家人朋友，献给我的母校北航和即将就读的南开，献给所有的相逢相识，也献给我自己。愿所有人一切顺利。

此致最真诚的谢意。