**毕业设计（论文）题目：1. 基于手机的室内导航与浏览系统设计与实现**

**学 院：** 计算机学院

**专 业：** 物联网工程

**班 级：** 07121301

**姓 名：** 刘云飞

**指导教师：** 张 磊

**校外指导教师：**

**摘要**

随着信息技术与计算机科学的发展，不同地方的地理信息展示形式呈现出多样化，以Google地图、百度地图、高德地图、Bing地图等为代表的地理信息平台已经融入到人们的日常生活之中。然而，与发展迅速成熟的室外定位技术相比，可靠的室内定位还未能被完全实现。比如说在结构稍稍复杂一些的大型商场或是地下停车库中，我们只能纯凭记忆力和直觉判断路线，而很难利用智能手机来查找相关信息。常见的G定位技术只能定位到建筑物，而个人在建筑物中的位置却由于信号减弱等原因，往往不能被准确找到。随着现代智能可移动设备的普及以及人们对其依赖性的不断提升，使用手机来实现室内可靠、快速以及普适的定位方法有着十分重要的意义，并且逐步受到相关领域研究人员的广泛关注。

本文在对传统的非G定位的技术基础上，通过使用Android智能机自带的传感器数据，针对现有的导航与定位算法的不足提出一些改进，设计并实现了一套基于惯性导航的室内定位算法，并且为Android智能机开发出一个可以使用的软件来测试。论文的工作主要有以下三个方面：

1. 对已有的室内定位技术以及算法进行研究分析，并且结合目前大多数手机所具备的传感器种类，提出了一套稳定、精确的定位算法；

2. 分析了目前常见的惯性导航以及传感器融合算法，提出了新的适合手机处理与展示的算法模型，并且实现原型系统；

3. 通过交互计算机图形学相关技术将移动数据在手机上绘制为方便用户交互的三维数据序列，并且能够实现室内漫游路径的记录与场景图片的关联。

通过最终软件测试与体验，本文设计的室内定位算法以及为此开发出来的便于交互的数据可视化系统，能够在室内提供精确的定位以及与场景进行关联，最终结果表明这项对于处理未来的室内导航与定位技术可以用于大型超市、博物馆、大型办公建筑以及医院等等。

关键词：室内定位 传感器融合 漫游路径 场景图片 图形图像 惯性导航

**Abstract**

With the development of information technology and computer science, the geographical information share a diverse future, and the geographic information platform represented by Google Maps, Baidu map, Amap and Bing map has been integrated into people's daily life. However, reliable indoor positioning has not been fully realized compared to the rapid development of outdoor positioning technology. For example, in a complex large shopping malls or underground garage, we can only rely on memory and intuitive judgments to route, and it is difficult to use smart phones to find relevant information. Common G positioning technology can only give the building location, but cannot get the location of the individual in the building because of the signal weakened and other reasons. With the popularity of modern intelligent mobile devices, the use of mobile phones to achieve reliable, fast and universal positioning method has a very important significance, and get the relevant areas and researchers attention gradually.

Based on the traditional non-G positioning technology, this paper puts forward some improvements to the existing navigation and positioning algorithm based on the built-in sensors in an Android smartphone, designs and implements a set of indoor positioning algorithm based on inertial navigation, also, design a software that can be used for test. The work of the paper mainly has the following three aspects:

1.According to the existing indoor positioning technology and algorithm research and analysis, with the combination of most of the current mobile phone with the kind of sensor, I put forward a stable and accurate positioning algorithm;

2.This paper analyzes the common inertial navigation and sensor fusion algorithms, proposes a new algorithm model suitable for mobile phone to process and display, and realizes the prototype system.

3. Through the interactive computer graphics related technology to make mobile data on the phone drawn to facilitate the user interaction of the three-dimensional data sequence, and can achieve the indoor roaming path records and scenes associated with the picture.

Through the final software testing and experience, the indoor positioning algorithm designed in this paper and the data visualization system developed for this purpose can provide accurate positioning and correlation with the scene in the room. The final result shows that this is the future of indoor navigation and positioning technology can be used for large supermarkets, museums, large office buildings, hospitals and so on.

Key words: indoor positioning sensor fusion roaming path scene picture graphic image inertial navigation

**目录**

[第1章 绪论 1](#_Toc482780546)

[1.1 课题的背景和意义 1](#_Toc482780547)

[1.2 国内外研究现状 2](#_Toc482780548)

[1.3 论文的主要工作和章节安排 4](#_Toc482780549)

[1.4 本文专有词汇对照 4](#_Toc482780550)

[1.5 测试项目开发的平台参数 5](#_Toc482780551)

[第2章 手机传感器 5](#_Toc482780552)

[2.1手机中可用于定位的传感器 5](#_Toc482780553)

[2.2 Android智能机的主要传感器分类及其架构 9](#_Toc482780554)

[2.3 Android传感器使用与分析 10](#_Toc482780555)

[第3章 导航算法设计 12](#_Toc482780556)

[3.1数据同步算法设计 12](#_Toc482780557)

[3.2传感器数据误差来源 15](#_Toc482780558)

[3.3坐标系定义及其变换 16](#_Toc482780559)

[3.4 EKF引入设计与姿态求解 18](#_Toc482780560)

[3.5 坐标系转换与解算位置信息 26](#_Toc482780561)

[3.6 小结 32](#_Toc482780562)

[第4章 用户交互设计及软件实现 32](#_Toc482780563)

[4.1 基于Android的导航算法实现 33](#_Toc482780564)

[4.2 三维场景设计与实现 34](#_Toc482780565)

[4.2.1 数据可视化 35](#_Toc482780566)

[4.2.2 人机交互 37](#_Toc482780567)

[4.3 软件架构设计与实现 39](#_Toc482780568)

[4.4 路径优化与结果展示 41](#_Toc482780569)

[4.5 小结 42](#_Toc482780570)

[第5章 总结与展望 43](#_Toc482780571)

[5.1 总结 43](#_Toc482780572)

[5.2 展望 44](#_Toc482780573)

[致谢 45](#_Toc482780574)

[参考文献 46](#_Toc482780575)

# 第1章 绪论

## 1.1 课题的背景和意义

随着人们生活条件以及居住水平的不断提升，出现了大量的大型购物商场以及一些内部结构较为复杂的大楼。越来越多的大型建筑提供了内部三维结构图来方便人们快速了解周围情况，但同时也存在一个问题就是人们需要先根据周围墙壁、物品等布局与地图进行匹配来得到自己当前的位置，这种方法并不快速以及准确。虽然通过GPS以及网络的方式定位已经逐渐满足人们在户外的交通和旅行，但是在室内由于GPS信号衰减以及定位精度有限，不能很好的得到人们室内的位置。同时，大型商场为了给顾客很好的导航功能或者在机场、火车站、地铁站等公共大型室内不再到处寻找内部地图以及自己的位置，更加快速的抵达出入口抑或人们在复杂建筑里行走能够更快的找到目的地而不至于在里面迷失方向，也在路口显眼的地方布置地图来提醒人们当前位置用来弥补建筑的复杂可能导致的人们迷路。

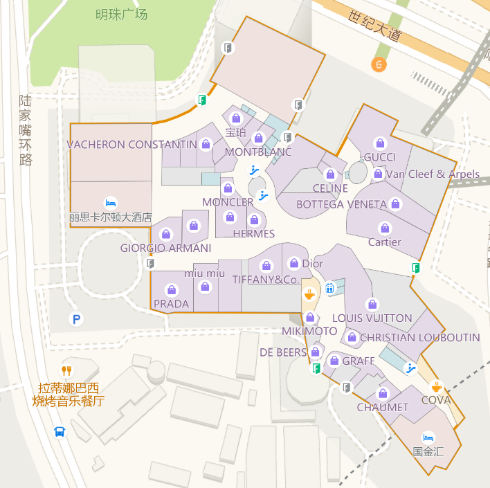




图1-1 复杂室内结构图展示

图1-1中左上图是上海国际会议中心一层的室内地图，右上图为广州美食博览中心室内截图以及左下为地铁站内部三维结构图和右下上海国金中心商场一层的室内地图。

考虑到手机在生活中的便捷性以及不可或缺性，人们已经希望通过手机来快速查看自己在任何地方的具体位置来方便下一步的判断。也希望即使第一次到达，也可以不使用线球来走出米诺斯的迷宫。日前，以Google Earth、 高德地图、百度地图等为代表的传统地图行业开始绘制以城镇建筑物模型为核心的三维场景，其中包括了大型建筑中的每个楼层的内部结构图。为了方便没有游览过城市的旅客能够迅速的对城市有所认识，并切身的感受到城市的魅力，欣赏到建筑内部美丽的景色。方便游客的出行，大量的室内定位技术出现在各种场合之中。

目前比较常见的室内定位技术主要是通过多个以Wi-Fi信号源为代表的信号发射源为基础，利用自由空间传播的无线通信信号强度只和工作频率与传播距离有关，并且在无线信号发射源工作频率已知的情况下，通过采集室内不同位置每一个信号源的不同方向上的信号强度作为识别一个地点的唯一依据，作为这个地点的指纹，并尽可能多的为室内每一个位置通过算法测量出其指纹并且记录到一个总得指纹库里面。之后再次通过采集当前位置的指纹信号，并通过从指纹库里搜索匹配与该信号最为接近的指纹作为当前位置信息。但是这种定位比较依赖信号源的布置，并且需要花费相当多的时间来制作位置指纹库。定位的精度十分依赖这个前期处理，换而言之，指纹库的丰富程度直接影响了定位精度，而搜索的算法则影响定位速度，可移植性比较差。

惯性导航系统（INS）是广泛用于航空航天和机器人及其自动化领域的用来导航以及定位的系统。其原理是通过采集惯性测量单元（IMU）中传感器信息数据，通过传感器融合、坐标系变换以及求解微分方程的方式达到实时的得到系统的位置信息用以导航。

随着Android智能机的硬件技术的迭代，手机内部传感器也包括了IMU主要的传感器类型：加速度计、陀螺仪以及磁罗盘。但是由于仅仅使用手机IMU传感器数据质量精度不够，另一方面没有Ground Truth信息来用以纠正，导致用以计算位置比较困难。如何很好的利用这些传感器信息来解算出一个准确的位置信息，是课题研究的重要方面之一。该课题涉及到多个传感器数据同步、传感器数据融合、姿态估计、微分方程的解算、计算图形学以及软件设计等学科的内容，而这些基本的内容需要使用计算机科学与导航制导学科以及物联网学科的交叉来实现。那么我们对于室内导航与浏览这一系统算法的实现不仅仅对于室内定位，而且还对其他的课题具有支持和推进作用，具备非常深远的显示意义。

## 1.2 国内外研究现状

关于定位这一技术实现的方式比较多，涉及到的领域也比较繁杂。可以将室内定位的方式分为下面四种：

1. 通过一些无线信号发射源作为基站，手机传感器接收一个或多个信号源，通过对信号强度进行处理或者解算，来定位手机当前的位置；

2. 通过卫星以及地面信号基站或者无线路由器通过GPS准确定位或者网络大致定位；

3. 通过web camera或者一系列摄像头来采集比较全面的图像信息[1],然后通过物体检测的方式来实现物体定位；

4. 通过IMU等一些测量惯性的传感器来通过导航算法进行定位。

在文献[2]里面实现了一个通过估计无线热点的位置信息来实现定位的算法，但是为了消除使用过程中的误差，需要在最开始使用者载入当前的楼层地图信息并且询问使用者当前在地图中的准确位置，以能够进行接下来的使用者的位置定位。在文献[3]中，设计了一个额外的硬件来获取使用者的当前精度比较高的运动状态信息，通过一个装载有激光测距仪以及光纤陀螺仪的机器人来实现一个能够自动将标定好的Wi-Fi数据信息处理成机器人地图的自动化程序。MapCraft[4]使用地图匹配算法以及无向图模型，通过线性链条件随机场算法来处理解算位置定位，但是这一方法大大的增加了定位的实现难度。在MapCraft中，一个楼层的地图信息是需要作为这个系统的输入才能进行接下来的操作，在此期间需要使用大量的变量信息来定义一个楼层的全部地图信息，并且在不同楼层的地图物体结构变换比较明显的这一描述地图信息的结构体也需要进行相应调整。因此，在文献[5]中，提出了仅仅依赖手机传感器来实现当前定位，用户只需最开始指出自己在当前地图中的位置即刻进行接下来的定位。更加深入地，Navshoe[6]以及FootSLAM，通过使用特制的硬件来完成室内导航定位，但是通过使用装在脚上的惯性传感器对于手机而言并不实际来实现，并且因此限制了它的推广使用。另外文献[7]提供的CompAcc方法，通过记录人行走时候的手机内部带有的磁罗盘以及加速度计数据特征信息，然后通过将其与地图上可能存在的路径信息进行匹配来实现定位，尽管此方法提供了一个定位精度达到行人的一步距离这样亚米级的定位方式，但是这种方式是为室外活动设计的，并且需要预先载入地图来进行处理。在SLAM[8]领域，主要是通过跟踪当前的当前的位置信息来构造未知地方的地图，以此方式实现物体的定位，然而构造的是一系列的特征点而不是便于直观理解的室内地图信息来便于理解位置信息。另外在文献[9]中提供了一个SmartSLAM的方法，通过收集人们行走时候的Wi-Fi信号来生成地图，这种方式把注意力放到人们经常行走的路径而不是整个楼层的地图，所以局限性也十分明显，必须先一遍生成才能下一步实现定位，对未知地方的定位支持不友好。CrowInside[10]通过使用智能手机的传感器数据，通过众包的方法来自动估计当前楼层是这一栋楼的哪一层，进而实现室内楼层级别的定位，但是这种方法需要将室内所有边界走一遍来估计这一楼层的形状，进而用来匹配它所在的楼层位置，但是此方法缺点也十分明显，并不是所有室内边界人是可以走一遍的，例如有家具等其他设备的存在导致某些地方人们无法接近室内边界。在文献[11]中展现了一种通过Wi-Fi指纹的方式来确定并且枚举所有的房间，通过内部传感器来估计过道的形状。然而这种方法缺点依然明显：只能估计出室内结构比较简单的建筑，对于定位来说没有突破性进展。LiFS[12]预先假设了一个楼层模型，通过自动将采集过来的Wi-Fi信号与存在的楼层Wi-Fi指纹库进行匹配来获取手机当前的位置，只要具有某一楼层的楼层地图信息与指纹库，这种方法就能通过匹配的方法很好的实现室内定位。换而言之，这个楼层地图信息以及与之对应的Wi-Fi指纹库信息需要提前人为录取生成。

## 1.3 论文的主要工作和章节安排

本论文的重点是通过使用手机INS传感器数据估算手机姿态与路径信息并且将数据通过手机软件的方式可视化，能够通过将路径信息与场景相关联的方式进行室内导航。

论文的主要工作包括：

在充满噪声的传感器数据中获取出有效的手机姿态、状态等相关信息，通过传感器融合算法得到手机精确度比较高的相关状态信息，并对过程、结果进行优化。

使用手机状态信息解算手机位置、路径信息，完成导航算法的设计与实现。

通过Android开发完成相关室内导航系统的软件设计，完成手机三维路径数据以及状态数据可视化处理，能够将路径状态信息与场景进行关联，并且可以回放历史数据信息。

论文各章节的具体安排如下：

第一章首先介绍课题的背景和意义，并介绍该课题在国内外的发展现状，在本章的最后给出论文完成的工作，词汇对照表以及开发平台的参数介绍；

第二章主要介绍了手机中搭载的可以中来定位的传感器种类及其工作原理和数据特征以及使用与分析；

第三章主要介绍了导航算法的设计，包括数据同步、误差分析与抑制，坐标系变换以及通过EKF进行传感器融合的算法推导与设计；

第四章主要介绍了用户交互设计与软件的实现，主要是第三章提出的算法通过Android开发实现并且完成三维场景的设计与实现和整个软件的设计与实现，详细介绍了数据可视化以及人机交互的设计与实现算法过程，最后优化了路径求解算法以及路径结果显示；

第五章给出论文的总结和展望。总结论文完成的工作，并对该课题未来的研究方向做出进一步讨论

## 1.4 本文专有词汇对照

表1-1 词汇对照表

|  |  |
| --- | --- |
| 词汇名称 | 词汇详细解释 |
| 设备 | 小米5s Android智能手机 |
| 惯导 | 惯性导航 |
| 惯导传感器 | 加速度计、磁罗盘与陀螺仪的统称 |
| 机体坐标系 | 手机的坐标系，下文会给出定义 |
| 大地坐标系 | 世界坐标系、参考坐标系 |

## 1.5 测试项目开发的平台参数

表1-2 开发平台参数

|  |  |
| --- | --- |
| 设备属性 | 详细信息 |
| 处理器与内存 | 3GB内存+64GB闪存  骁龙821 最高主频 2.15GHz  Adreno 530 图形处理器 624MHz  3GB LPDDR4 1866MHz 双通道  64GB 机身存储 UFS2.0 |
| 重量与尺寸 | 高度：145.6 mm 宽度：70.3 mm  厚度：8.25 mm 重量：145 g |
| 导航定位 | GPS AGPS GLONASS 北斗定位 |
| 感应器 | 陀螺仪 加速度 传感器 距离感应器 环境光传感器 霍尔感应器 电子罗盘 气压计 |

# 第2章 手机传感器

此章节主要介绍当下手机中主要搭载的传感器种类以及详细介绍可以用来对手机进行定位的传感器的工作原理以及Android架构中传感器数据特征与调用。其中改论文中主要通过使用惯性传感器，采取惯性导航的算法进行手机的相关定位。

## 2.1手机中可用于定位的传感器

**1. Wi-Fi**

Wi-Fi是IEEE 802.11通信标准的另一个更为常见的名称，任何一个具备Wi-Fi模块的设备都可以通过无线网络接口来访问网络，例如笔记本电脑，游戏机、智能手机以及现在的具备物联网功能的家电和播放设备。

使用无线网络通信是一个双工无线电通信的过程：

1. 一个设备的无线适配器将数据转化成无线电信号并且通过外部天线或者内部天线将数据发送出去；
2. 无线路由器接收信号并且将无线电信号解码为数字信号，然后将信息通过网线与互联网通信。

反之亦然，无线路由器通过网线从互联网中获得信息，然后转化为无线电信号发给具有Wi-Fi模块的无线适配器。

每一个无线路由器通过广播信息来使得一定区域内的设备能够接收到Wi-Fi信号，这些设备能够获得无线路由器的信号强度，这个信号强度可以转化为数字信息，定义为接收的信号强度指示（RSSI）。具备Wi-Fi功能的智能手机，通过手机上的软件能够完成这项任务，然后通过Wi-Fi指纹或者Wi-Fi SLAM来实现定位。

1. **GPS**

全球定位系统是由美国国防局设计制造的基于卫星的导航系统，之前主要用于军队使用。这套系统于1994年秘密设计，但是现在GPS已经开放部分功能给民用，已经成为我们生活中密不可或缺的部分。

GPS分为3个部分：宇宙空间部分（卫星）、控制部分以及用户地面端（GPS接收器或者GPS模块）。

1. 宇宙空间部分主要包括24颗卫星，其中每6颗在赤道之上间隔55°倾角绕地球旋转，如此这样布置是为了保证地球表面的每一点都至少能收到6颗卫星的信号
2. GPS发送导航信息是每秒钟50bit，其中主要包括的信息有：消息发出的具体时间信息、具体轨道信息、整个系统健康度信息、大致的所有卫星的轨道信息
3. 控制部分主要包括一些基站和天线。通过他们来监视卫星的健康程度以及向卫星发送控制指令，如有必要，可以立刻调整卫星的时钟。
4. 地面用户部分分为军用GPS精密导航服务部分和民用GPS标准导航部分，地面GPS接收器主要包括一个天线，一个十分稳定的内部时钟，以及通过固件实现的解算用户位置、速度信息，往往也会演示相关信息给用户。

GPS接收器使用三边测量算法，通过结合不同卫星的数据信息来解算接收器正确的位置信息。综上所述，GPS信息包括GPS数据发送时刻时间、精确的卫星信息、GPS系统健康度信息、以及较为粗糙的其他卫星的状态信息。接收器主要使用时间信息来计算与卫星的距离，使用三边测量的算法，通过和多个卫星的距离解算出当前接收器的位置信息。

使用卫星系统定位的好处是接收器可以解算出经纬度、海拔等位置信息，并且具有比较高的准确度。然而，视距传输（LOS）的特性导致GPS不能很好的用于屋顶和墙壁遮挡的室内环境。

1. **Bluetooth**

Bluetooth也是两个以及两个以上的设备通过无线通信的标准，在比较短的距离内，Bluetooth具备比较明显的优势，因为它可以无线通信、很便宜、比较小并且可以随便移动。

Bluetooth相比红外线技术（另一种无线通信技术，主要用于远程遥控系统）有诸多明显的优势，虽然红外线通信也是非常可靠并且比较廉价，但是红外通信有着许多缺点：它必须直线控制与使用，使用时必须对准信号接收器才可以通信，另外红外通信一般是一对一式的通信。Bluetooth在红外通信的优势基础上很好的解决了他的缺点：通过适配调频技术，可以支持同时多个设备相互通信，并且Bluetooth不受直线传输的限制，因此Bluetooth可以同时控制多个房间的不同设备。

一代的Bluetooth的最大传送速率为1Mbps，二代的Bluetooth，也是现在主要的手持设备的使用标准，可以支持3Mbps的传送速率，并且最大支持传送距离为10米。

1. **惯性导航系统**

惯性导航系统（INS）主要基于独立的导航技术，主要通过加速度计和陀螺仪的测量数据（有的包含磁罗盘数据）。通过跟踪物体相对于起始点的位移、姿态与速度来实现定位。

INS主要分为两个坐标系，依照不同的传感器数据信息。导航器系统的参考坐标系为载体坐标系，所运动的环境的坐标系称为全局坐标系。

INS分为两种：第一种在载体坐标系中包含平台增稳系统，在此内部传感器与外部的姿态变化隔离，这个平台的坐标系与全局坐标系保持一致。第二种被称为捷联系统，内部传感器与载体刚性连接，也因此采集的数据信息与载体姿态直接相关，数据信息为载体坐标系数据而非全局坐标系。

* 1. **加速度计**

加速度计是测量较为准确的加速度的相关数据重要设备。数据不需要与加速度坐标系保持一致，会随着载体的姿态和速度改变而改变，但是加速度的参考坐标系与地球上的重力系统有着比较强的关联。

总体来讲，当受力时，加速度计测量的是当前所受的多大的作用力。当它静止在地球表面时候，它收到的作用力为1G向上的力，因为在地球表面上的任意一个点相比于当前坐标（以自由落体建立的坐标系）系有一个相对的向上的加速度，是由地心引力造成的。为了获取准确的加速度需要考虑地球的运动规律，也因此必须减去引力加速度。

* 1. **陀螺仪**

大致来讲，陀螺仪是基于角动量原理用来测量或维持方位的传感器。一个传统的陀螺仪包括一个安装在两个旋转轴上的转子，这样能保证它有三个自由度。通过这个物理原理，当整个系统的方位姿态发生变化的时候，转子能够保持自己本身的姿态不变化。只是通过那两个轴在变化。陀螺仪主要用于测量方位。之后的包含陀螺仪的微控制器（MEMS），主要用来测量角速度。

具备微控制器的陀螺仪包括震动部分和通过科氏效应的测量部分。一个物体通过一个驱动轴来振动，当陀螺仪旋转时候，一个二级震动就会由于科氏力通过传递给一个垂直的感应轴。也因此角速度可以通过测量这个二级旋转来解算出来。

科氏效应表达的是在参考坐标系中，物体以的角速度在转动，一个质量为m的物体以v的线速度旋转时候受到的力为：

一个比较关键的地方在于加速度计和磁力计能测量加速度和角度是相对于地球，而陀螺仪测量出来的角速度则是相对于载体的。

* 1. **磁力计**

磁罗盘是用来测量设备周围磁场方向以及大小的工具，磁力计分为两种：一种为尺度磁力计用来测量周围环境磁场的强弱，另一种为矢量磁力计，主要用来捕获和测量设备方位与周围环境的方向信息。

## 2.2 Android智能机的主要传感器分类及其架构

手机中很多模块都可以称为传感器，例如触摸是一种传感器，摄像头也是一种传感器。在Android的软件架构中不适用触摸与摄像头，其中使用的传感器主要包含以下三类传感器：

表2-1 手机传感器种类

|  |  |
| --- | --- |
| 类别 | 包含传感器种类 |
| 运动传感器 | 加速度计/重力传感器/陀螺仪传感器/旋转向量传感器 |
| 环境传感器 | 气压计传感器/压力传感器/温度传感器/湿度传感器 |
| 方位传感器 | 方向传感器/磁罗盘传感器 |

这几类传感器的共同特点为：

1. 产生数据量少
2. 实时性较强
3. 物理构造相对简单

当今绝大多数Android手机所携带有的传感器种类大致分为11种，其中可以通过Android的SDK中的Sensor类来获取传感器技术参数的方法，其中种类与获取方法如下表所示：

表2-2 Android SDK自带传感器获取方式

|  |  |  |
| --- | --- | --- |
| 序号 | 传感器 | Sensor类中定义的TYPE常量 |
| 1 | 加速度传感器 | TYPE\_ACCELEROMETER |
| 2 | 温度传感器 | TYPE\_AMBIENT\_TEMPERATURE |
| 3 | 陀螺仪传感器 | TYPE\_GYROSCOPE |
| 4 | 光线传感器 | TYPE\_LIGHT |
| 5 | 磁场传感器 | TYPE\_MAGNETIC-FILED |
| 6 | 压力传感器 | TYPE\_PRESSURE |
| 7 | 临近传感器 | TYPE\_PROXIMITY |
| 8 | 湿度传感器 | TYPE\_RELATIVE\_HUMIDITY |
| 9 | 方向传感器 | TYPE\_ORIENTATION |
| 10 | 重力传感器 | TYPE\_GRAVITY |
| 11 | 线性加速度传感器 | TYPE\_LINER\_ACCELERATION |
| 12 | 旋转向量传感器 | TYPE\_ROTATION\_VECTOR |

其中，1~8是硬件传感器，9是软件传感器，其中方向传感器的数据来自重力和磁场传感器，10~12的传感器视具体情况而定，可以为硬件或软件传感器。

对于传感器的整体架构上具备如下图所示：



图 2-1 Android传感器架构

传感器整体逻辑结构中，其中包括一个控制流（蓝色向下箭头），一个数据流（红色向上箭头）。控制主要包括开关传感器，设置传感器采样频率；数据流则是从驱动到应用的整个过程。

## 2.3 Android传感器使用与分析

该系统主要使用运动传感器以及方位传感器两大类别的传感器，其中具体传感器详细数据信息如下：

表2-3 Android 运动传感器详细信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 传感器名称 | 作用 | 单位 | 坐标系 | 备注 |
| 1 | 加速度传感器（G-Sensor） | 返回x, y, z轴的加速度数值 | m/s^2 |  | 将手机平放在桌面上，x轴默认为0，y轴默认0，z轴默认9.81.  将手机向左倾斜，x轴为正值。  将手机向上倾斜，y轴为负值。  一般提供±2G至±16G的加速度测量范围 |
| 2 | 磁力传感器（M-Sensor） | 返回x, y, z三轴的环境磁场数据。 | uT  微特斯拉 |  | uT， micro-Tesla  单位也可以是高斯（Gauss），1Tesla=10000Gauss。  硬件上一般没有独立的磁力传感器，磁力数据由电子罗盘传感器提供（E-compass）。  电子罗盘传感器同时提供下文的方向传感器数据。 |
| 3 | 方向传感器（O-Sensor） | 返回三轴的角度数据，方向数据的单位是角度。  为了得到精确的角度数据 | 角度 | azimuth：方位，返回水平时磁北极和Y轴的夹角，范围为0°至360°。  0°=北，90°=东，180°=南，270°=西。  pitch：x轴和水平面的夹角，范围为-180°至180°。  当z轴向y轴转动时，角度为正值。  roll：y轴和水平面的夹角，由于历史原因，范围为-90°至90°。  当x轴向z轴移动时，角度为正值 | 为了得到精确的角度数据，E-compass需要获取G-sensor的数据，  经过计算生产O-sensor数据，否则只能获取水平方向的角度。  方向传感器提供三个数据，分别为azimuth、pitch和roll。 |
| 4 | 重力传感器（GV-Sensor） | 输出重力数据 | m/s^2 | 坐标系统与加速度传感器相同 | 当设备复位时，重力传感器的输出与加速度传感器相同。 |
| 5 | 线性加速度传感器  （LA-Sensor） | 线性加速度传感器是加速度传感器减去重力影响获取的数据 | m/s^2 | 坐标系统与加速度传感器相同。 | 加速度 = 重力 + 线性加速度 |
| 6 | 陀螺仪传感器（Gyro-Sensor） | 返回x、y、z三轴的角加速度数据。 | Rad/s |  | ST的L3G系列的陀螺仪传感器 |
| 7 | 旋转矢量传感器  （RV-Sensor） | 将坐标轴和角度混合计算得到的数据。 | Rad/s | RV的方向与轴旋转的方向相同。  RV的数据没有单位，使用的坐标系与加速度相同。 | RV-sensor输出三个数据：  x\*sin(theta/2)  y\*sin(theta/2)  z\*sin(theta/2)  sin(theta/2)是RV的数量级。  RV的三个数值，与cos(theta/2)组成一个四元组。 |

# 第3章 导航算法设计

惯导传感器主要应用于惯性导航系统中对系统进行实时导航以及导航算法的数据来源，通过采集惯导传感器数据来实时获取系统的相关状态信息。其中加速度计、磁罗盘传感器以及陀螺仪是主要的传感器数据来源，这三种传感器的物理原理在上一章节已经详细介绍，下面我们将详细分析以及使用这些传感器数据来解算手机姿态以及手机的位移信息，接下来本论文将从设计传感器数据同步算法以提高数据准确度、分析传感器数据误差以在不同情况选择并使用合适的传感器数据、坐标系定义以及坐标系之间的相互转换、传感器数据融合以及EKF的设计来求解手机姿态以及物理运动学公式来计算手机位移信息。

## 3.1数据同步算法设计

由于三种传感器数据来自三个具备独立系统的传感器，通过订阅手机传感器服务的数据响应事件中每一个传感器数据均带有数据发生的时间戳，数据产生的数据需要严格保证时间最为接近的传感器数据进行组合来解算每一个时刻的手机姿态。

由于加速度计和陀螺仪来自传感器MPU6050，而地面磁罗盘数据来自三轴磁罗盘传感器，其产生的数据不同步。并且加速度计与陀螺仪的数据更新频率为200Hz，磁罗盘更新数据的频率为100Hz。将数据同步来处理对于提高导航精度计算手机姿态准确度，减少误差来说很重要。

将传感器产生从数据通过加上时间戳的方式，将这三种传感器数据：分别放进三个消息队列中，如下图所示：



图3-1 不同传感器数据时间戳示意图

实际传感器数据是通过注册传感器系统服务时候获取的传感器实际产生的时间戳在不同传感器中接收的数据之间有偏置的，如果直接使用会导致系统误差增加，这是在环境观测噪声本身就比较多的前提下无疑降低系统的鲁棒性。所以通过一个消息队列来缓冲一定数据量的传感器数据，然后每种传感器数据对应的队列中数据的时间戳简化得到上图效果。

我们需要将数据通过时间戳来对齐，实现算法如下：



图3-2 多传感器数据同步算法

通过此算法，将三个传感器的数据同步为频率100Hz的数据，进行下一步的姿态解算和位置计算。

## 3.2传感器数据误差来源

**1. 加速度计**

最重要的加速度的误差来源是它的偏差，这个偏差主要是加速度计的输出结果与实际值存在一个偏差，单位是，一般是一个常量（不考虑测量噪声）。通过二次积分，这个误差导致的定位结果将是以二次的速度增加，在定位种的误差为：

上式中t表示积分的时间。

此误差可以通过让加速度计长时间静止，通过计算此段时间的输出数值的平均值来校准加速度计的输出误差，这个误差也会严重影响定位精度，所以必须被消除掉

1. **陀螺仪**

陀螺仪测量的偏差与加速度计有些类似，就是陀螺仪的输出结果与实际角速度之间存在一个偏差，单位是，也是一个常量，这里是一次积分，导致角度误差是随着时间线性增长。

这个常量偏差也可以通过长时间静止求出陀螺仪的输出值的平均值作为这个偏差值，通过减去这个偏差值来进行校验陀螺仪。

另外一个误差被称为陀螺仪的标定误差。主要与陀螺仪本身的尺寸因素、排列以及陀螺的线性因素导致的，此误差只能在设备旋转时候发现。这个误差会导致角度积分时候出现漂移误差，这个误差的数量级由测量的频率和积分时间确定。

1. **磁罗盘**

磁罗盘的两个主要测量误差为：传感器自身的受损以及周围含铁钴镍化学元素的影响。如果传感器本身随着测量而发生旋转，也会导致误差产生。

## 3.3坐标系定义及其变换

此时引入姿态角（Euler角）。将飞行器的姿态角引入进行描述手机的姿态信息为下一步的位移信息做铺垫。飞行器姿态角不是特指那个角度，而是这三个角度的统称，也叫欧拉角。分别为俯仰，偏航，滚转。对应物理意义为手机围绕XYZ三个轴分别转动形成的夹角。

地面坐标系（earth-surface inertial reference frame）——O ，表示图如下图所示：

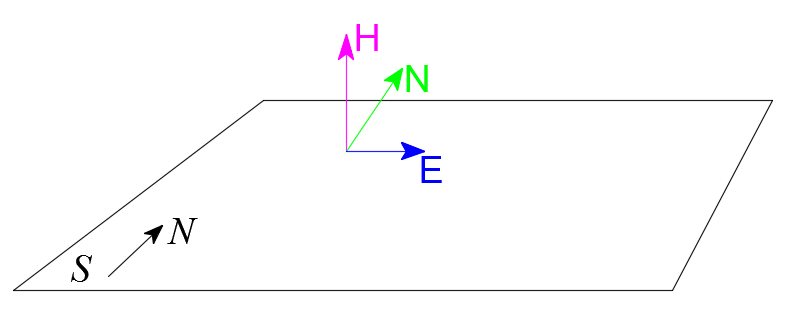


图3-3 地面坐标系示意图

其坐标选取方式如下：

1. 在地面上选一点
2. 使轴在水平面内并指向某一方向
3. 轴垂直于地面并指向地心(重力方向)
4. 轴在水平面内垂直于轴，其指向按右手定则确定

机体坐标系（Aircraft-body coordinate frame）——O　X　Y　Z，其对应于手机的手机坐标系，将它定义为如下：

图3-4 手机坐标系下示意图

其坐标选取原则如下：

1. 原点O取在手机质心处，坐标系与飞机固连
2. X轴在飞机对称平面内并平行于手机的设计轴线指向机头
3. Y轴垂直于飞机对称平面指向手机机身右方
4. Z轴在飞机对称平面内，与x轴垂直并指向手机机身下方

其中的姿态角roll，pitch，yaw,分别代表手机绕x，y，z进行右手螺旋方向的旋转，其中负值表示反方向旋转角度，更确切说由下图来表示：

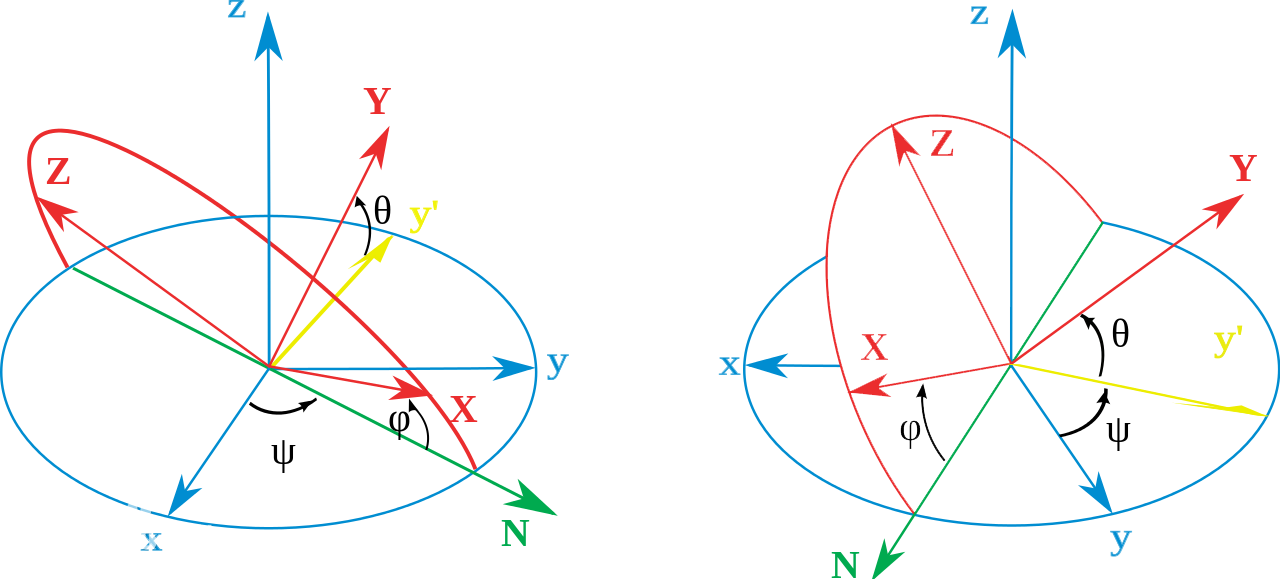


图 3-5 欧拉角

通过手机的运动传感器中可以获取出三轴的加速度和角速度，分别对应传感器类型为TYPE\_LINEAR\_ACCELERATION以及TYPE\_ROTATION\_VECTOR。通过订阅传感器服务可以获取到手机姿态的六个数据（三个加速度AD值，三轴加速度AD值）这些描述的均为手机的姿态。通过姿态融合就可以得到Pitch**、**Roll、Yaw角。此外也可以通过使用磁罗盘（TYPE\_MAGNETIC\_FIELD）获取手机的yaw，可以将其用于姿态融合的互补滤波。

通过加速度和角速度都可以得到Pitch和Roll角（加速度不能得到Yaw角）。就是说有两组Pitch、Roll角。

加速度计的优点如下：

1. 加速度值没有累计误差，并且可以通过tan函数得到倾斜角
2. 包含噪声太多，待测物体运动时会产生加速度，放置人身上的抖动也会产生加速度，不能直接使用

角速度的数据来源为陀螺仪，其优点如下：

1. 陀螺仪受外界振动影响小，精度高，
2. 倾角需要通过对角速度积分得到，会产生累计误差

综上不能单独使用加速度计或者单独使用陀螺仪，所以需要将这两个传感器进行数据融合，把他们结合到一起得以修正。最简单的方式是采取互补滤波，由下公式表示：

其中和分别表示不同传感器采集数据的来源，单位保持一致的情况下，采用和表示对其两个数据源的置信度，其中。

如此通过互补算法只能得到一个倾角，为了得到三个姿态需要三个互补算法即可。

通过互补需要人为参与调整置信度，需要不断调整，在测试之前不能通过数学方式直接的评判其系数优劣，所以引入EKF进行自适应数据融合。

## 3.4 EKF引入设计与姿态求解

卡尔曼滤波是一种高效率的递归滤波器（自回归滤波器），他能够从一系列的不完全以及包含噪声的测量中，估计动态系统的状态。应用到此处，手机的陀螺仪，加速度计以及磁罗盘的数据是时域上具有高斯白噪声特性的数据，通过建立数学模型，得到手机姿态的最优估计。

由于手机这三个Motion传感器采集的数据对手机的运动描述是有限的，并且包含系统噪声以及环境噪声，通过对手机的位置（通过积分得到的）物体位置会有偏差，为了是这个误差达到最小，就引入了卡尔曼滤波。由于手机的位置、速度以及加速度信息在任何时候都有噪声，卡尔曼滤波利用目标的动态信息，通过最优化原理去掉噪声的影响，得到一个关于手机位置的比较优的估计。这个估计对于应用的不同主要分为三种：对当前目标位置的估计（滤波）；对手机将来位置的估计（也叫预测），也可以是对过去位置的估计（通过插值或者平滑）。

卡尔曼滤波的基本动态模型：KF建立在线性代数以及隐马尔科夫模型的基础上。他的基本动态系统可以用一个马尔科夫链来表示，这个马尔科夫链建立在一个被高斯白噪声干扰的线性算子上面。系统的状态可以用一个元素为实数的向量来表示。随着离散时间的每一个增加，这个线性算子就会作用在当前的状态上，通过运算产生一个新的状态，并且引入一些新的噪声，如果有控制信号，控制信息也会被加入。同时，英一个受噪声干扰的线性算子产生出这些隐含状态的可见输出。

为了从一系列有噪声的观察数据中用卡尔曼滤波器估计出被观察过程的内部状态，通过在卡尔曼滤波的框架下建立物理模型。换而言之，对于每一步k，定义矩阵，，以及。于是卡尔曼滤波器的模型假设k时刻的真实状态是同（k-1）时刻的状态演化而来，符合下面的公式：

其中

是作用在手机上一个时刻状态上的变量变换模型（雅可比矩阵）

是过程噪声，并且假定其为均值为零的高斯白噪声，具有协方差矩阵为的多元正态分布

在时刻k，与真实状态相对应的观测满足下面的式子：

其中是观测模型，它把真实状态空间映射成观测空间，而是观测噪声，也是均值为零的高斯白噪声，其协方差矩阵同样服从正态分布。

在运行这个滤波器需要设置初始状态，初始状态以及每一时刻的噪声{}都认为是相互独立的。

在此期间，KF是一种递归的估计，只需要知道上一时刻状态的估计值以及当前状态的观测值就可以计算出当前时刻状态的估计值，也因此不需要记录观测或者估计的历史信息。这个与其他滤波器最大的不同就是不需要像低通滤波器等频域滤波器那样，需要在频域上设计再转换到时域上实现。

手机的状态主要有下面两个变量来表示：

，在时刻k状态的估计；

，后验估计误差协方差矩阵，是用来度量估计值的精确程度。

于是，对于KF的操作主要包括两个阶段：预测与更新。在预测阶段，滤波器采用上一时刻的状态估计，结合当前的观测做出对当前状态的估计。在更新阶段，滤波器利用对当前状态的观测值优化在预测阶段获得的预测值，来获得一个更为精确的估计值。

其中预测通过以下公式来实现：

预测状态

预测估计协方差矩阵

更新通过计算下面三个量来实现

测量余量

测量余量的协方差

得到的最优卡尔曼增益

通过上面求解的结果来更新滤波器的状态x和预测p

更新之后的状态估计

更新之后的协方差估计

其中上述式子的不变量有：

由于从手机的加速度计是手机坐标系下的瞬时加速度，磁罗盘是获得的手机在周围环境的具有噪声的绝对的手机的轴角数据，陀螺仪获得的是较为准确的手机角速度信息，所以我们的EKF滤波器的状态可以设计为：

其中Position，Velocity，Accelerate均为世界坐标系下的运动信息数据，Pose表示当前手机的姿态。并且每一个数据为这样的一个在欧式空间里的一个向量，对于前三个分别表示通过坐标投影下的在世界坐标系下x轴, y轴,以及z轴上的位置、速度以及加速度数据，对于Pose表示当前的姿态的分别围绕x轴、y轴，z轴转过的轴角

观测设计为：

其中Accelerate是手机坐标系下加速度计数据回传的数据，Magnitude是三轴磁罗盘传感器回传的数据以及Gyroscope是陀螺仪回传的数据，并且他们也都是一个数据为这样的一个在欧式空间里的一个向量，分别表示对应的正交分解在不同轴上的加速度、角偏移以及瞬时角速度。

为了方便计算，将状态向量中的元素简化记为：

则当前时刻t的状态为：

上一时刻t-1的状态为：

初始时刻状态为：

从状态到的更新过程通过物理学的运动模型可以得到：

其中表示状态更新的时间间隔。

这里面的Pose是手机当前的姿态，数据为，其中x表示roll，y表示pitch，z表示yaw。这里的Pose表示一个旋转，即从世界坐标系到手机坐标系的旋转。也可以采用四元数来表示，那么数据将为，虽然形式不同，但是表达的意义均为从世界坐标系到手机坐标系的旋转。

将手机获得的观测（测量）矩阵简化为：

那么，状态则可以通过下面的公式计算得出：

其中R表示从世界坐标系到手机坐标系的旋转矩阵。下面给出从欧拉角到旋转矩阵以及从四元数到旋转矩阵的推导公式：

1. 欧拉角到旋转矩阵

其中：

1. 四元数到旋转矩阵

通过以上分析发现，对于状态向量的更新，其中的数据流结构如下：



图3-6 EKF数据融合数据流

通过数据流分析，在状态S1，S2以及S3与滤波器不存在耦合关系，即可以将上述模型进分解、化简得到如下模型：



图3-7 EKF简化之后的数据融合数据流

左边的位置计算过程可与右侧EKF相互独立分别进行计算。由此，EKF的数据更新部分由此更新为通过Z1，Z2以及Z3三种传感器数据求解手机姿态的问题，手机定位则化简为由已知手机机身坐标系下的三轴加速度以及手机姿态，通过坐标系变换求取手机在世界坐标系下的各个坐标轴投影的加速度进而通过求解微分方程获得手机的速度、位移信息。

1. 基于EKF数据融合算法的手机姿态求解

此处涉及此种方法定位的基础：使用传感器数据。在Android智能手机里面提供了三种基础的运动跟踪传感器：

1. 加速度传感器测量的是手机任意时刻的加速度信息，返回一个手机坐标系下的三个轴的加速度信息，但是加速度由于传感器特性导致数据信息十分粗糙，下面是通过加速度信息计算的手机关于X轴旋转的角度信息：



图3-8 加速度解算的关于x轴旋转角度

1. 陀螺仪测量的是旋转，返回的数据信息是在手机坐标系下的旋转角度矢量。陀螺仪的输出数据结果十分平滑，对于比较微小的旋转信息十分具有代表性。陀螺仪可以被用来估计手机当前姿态，通过在每一个时间戳修正其历史数据，结合一个新的角度旋转向量。通过积分的方式获取手机当前姿态可以比较好的工作，但是存在累计误差。即当积分一段时间之后，即使将手机静止放置，得到的数据依然存在偏移。并且通过将手机恢复到初始姿态，通过积分得到的数据不能很好的得到最初的姿态数据信息。下图是通过陀螺仪数据积分得到的手机围绕X轴的旋转信息：



图3-9 陀螺仪积分的关于x轴旋转角度

1. 磁罗盘传感器测量的是磁场，返回的是一个手机周围环境的近磁场（包含地球磁场）向量。这个传感器类似于一个指南针，可以获得手机的朝向信息，与手机加速度计结合可以得到手机的姿态，因为可以通过磁罗盘数据向量弥补加速度计无法求解得出的手机yaw角信息。但是磁罗盘的数据极其容易手到手机周围的磁场干扰，包括手机内部的电子电路产生的电磁场。

通过对比通过加速度计与磁罗盘解算的手机姿态以及通过积分角速度得到的手机姿态这两者的姿态数据，其中当手机仅仅旋转x轴得到的结果如下：

图3-10 两者对比的关于x轴旋转角度

其中src1为陀螺仪积分得到的姿态角数据，src2为通过加速度计与陀螺仪解算的姿态数据。每一个传感器都有自身的优点和缺点，陀螺仪不知道全局下的信息，然而加速度计的噪声又十分严重并且没有yaw角信息。传感器融合的目的是采用每一个传感器数据信息并且将传感器之间互相取长补短，得到一个更为可靠的结果。

由于加速度传感器的数据矢量始终受到地球重力影响，方向指向失重方向，即与地心方向相反，另外通过使用磁罗盘，可以得到手机朝向与正南方向上的旋转矢量。为方便计算，记为A，为E由此，可以通过下面的式子求解第三个轴的方向旋转信息：

则旋转矩阵R（从东北天坐标系到手机坐标系的转换）为：

可以通过下图来证明：

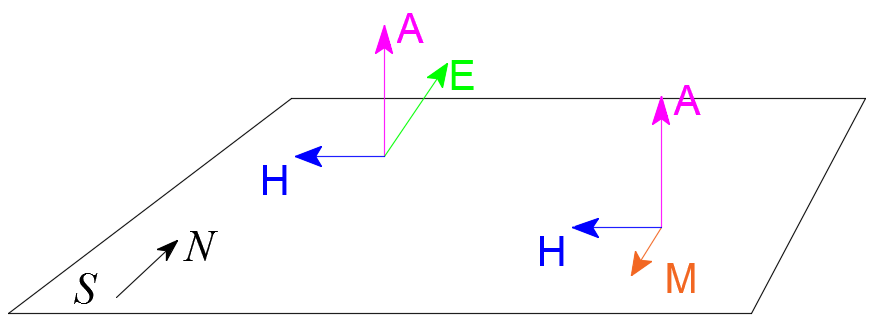


图3-11 旋转矩阵求解证明

即通过将加速度获得的与朝向地心的旋转角度矢量信息与磁罗盘获得的与正南方向的旋转角度矢量进行单位化并且进行叉乘，即可得到手机姿态与正东方向的旋转角度矢量。分别将手机坐标系三个轴在大地坐标系下的三个轴的向量归一化，从而得到手机姿态与正东、正北以及指向天空的坐标系（东北天坐标系）的旋转矩阵。

通过传感器数据融合之后得到的姿态数据如下：

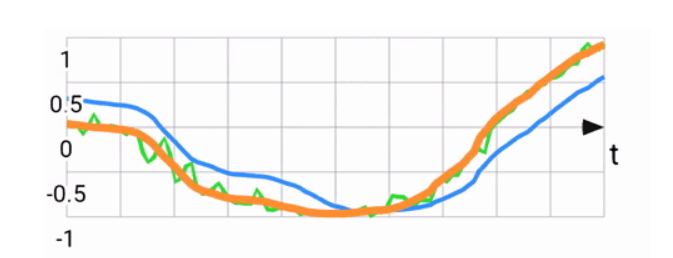


图3-12 数据融合之后的姿态信息(橙色)

其中绿色为加速度计与磁罗盘计算出来的噪声比较严重的数据，而蓝色的为陀螺仪积分得到的具有累积误差的数据，橙色数据为通过传感器数据融合得到的数据，相比之下比较理想。

## 3.5 坐标系转换与解算位置信息

由于旋转矩阵为的矩阵，过程运算比较繁琐，所以考虑将其转换成更为简洁的旋转描述方法上面去，因此考虑采取欧拉角或者四元数来进行计算。

首先，欧拉角的描述仅为三个值：roll，pitch和yaw，相比于四元数的四个值而言，更为简洁明确容易理解：分别为当前姿态绕x轴，y轴以及z轴旋转的角度。但是欧拉角有一非常严重的缺点，也就是他的奇异值问题，又叫万向锁。由于物体旋转过程中坐标轴随着物体做相同的转动，所以目前手机的运动属于动态的欧拉角。当手机的yaw角先旋转45°，然后pitch旋转90°，最后的结果与先pitch旋转90°与再roll旋转45°的结果是等价的。可以通过证明得到，一旦pitch角旋转±90°，就会导致第一次旋转yaw与等价，整个旋转表示系统被限制在只能绕竖直旋转轴旋转，从而丢失了一个维度，也因此这种±90°的第二次旋转使得第一次和第三次旋转轴相同的现象，称为万向锁。除此之外，当表达3D旋转，旋转顺序对于欧拉角来说十分重要，所以也有了不同的情况：

1. 不同旋转角的组合次序
2. 相对于旋转物体的坐标系和绝对坐标系
3. 左右旋转以及右手旋转
4. 不同旋转角的标记名字
5. 使用欧拉角表达的两个旋转之间的过程是非线性的

由于种类之多导致没有一个被广泛接收的标准来表达欧拉角，虽然欧拉角可以直观表示手机的姿态，但是在运算过程中遇到万向锁直接导致结果出错以及上面一些特征，所以在过程运算引入四元数，通过四元数来进行过程姿态计算。

四元数主要是用来表示一个三维的旋转，与欧拉角不同，四元数是一个四维空间的向量，由四个变量组成，其中一个是一个实数来表达旋转，记为w，另外三个变量为虚数，来表示三个正交的维度i，j以及k，这三个虚部的平方根为-1，所以四元数可以由下面式子表达：

在上一步，通过使用加速度计与磁罗盘得到手机噪声比较严重的姿态对应的旋转矩阵，下面给出将旋转矩阵转化为四元数的公式推导：

由于旋转矩阵R满足以下条件：

1. 该矩阵是通过三个向量组合而成的
2. 组成矩阵的特征值以及行列式均为+1
3. 该矩阵表示一个纯粹的旋转特征

所以该矩阵为正交矩阵，所以将的正交矩阵转化成四元数过程为：

同时，通过陀螺仪得到的手机旋转向量表示手机在每一个轴当前旋转的角速度，通过乘上时间间隔即可得到手机在采样间隔的每一个轴所旋转过的角度，称之为轴角。即手机围绕x，y以及z轴旋转之后的角度，此阶段的可以通过下面方式来计算:

其中

1. a表示旋转的角度
2. x，y，z表示旋转角度在每一个轴上的投影。

所以更直观的理解，四元数中i表示一个关于x轴的旋转，j表示一个关于y轴的一个旋转，k表示一个关于z轴旋转，均为180°，其中若表示一个关于x轴旋转了360°。

手机姿态的通过四元数更新，可以通过四元数乘法进行状态更新，适用于通过陀螺仪来更新手机姿态：

将通过加速度计与磁罗盘数据解算手机姿态获得的手机关于绝对坐标系（也称东北天坐标系）的旋转，通过陀螺仪累计得到的手机当前姿态旋转为，下面开始设计EKF数据融合器。

首先通过前期计算，我们的简化之后的观测有两个：以及，这作为滤波器的输入，同时我们需要的手机姿态q，作为滤波器需要维护的状态，这个这状态也是我们用以求解手机绝对坐标系下的重要依据。

算法流程如下：



图3-13 EKF数据融合与姿态估计算法

其中，初始化部分尤为重要，由于需要初始化4个部分，即初始化状态矩阵、初始化模型、计算雅可比矩阵以及初始化Q、R矩阵。在解算手机姿态这个模型中，主要采取的模型为互补滤波模型，由下面公式表示：

其中和分别表示不同传感器采集数据的来源，单位保持一致的情况下，采用和表示对其两个数据源的置信度，其中。

由于这里面采取的用以描述手机姿态的四元数，即得到的是两个描述手机姿态的，以及。于是观测矩阵如下表示：

其中q的表达式为，但是我们这里将其当作一个整体来运算。

状态矩阵变为：

关于状态更新的模型中，，

由于陀螺仪存在累计误差，所以将Q，R矩阵设置为

在其中四元数的运算，通过陀螺仪更新的姿态如下计算：

融合加速度计与磁罗盘的解算姿态如下计算：

其中slerp表示线性插值，在陀螺仪解算的姿态累计计算出来的q与加速度计与磁罗盘求解的暂时的姿态q的一个中间插值。

通过上一步骤求解得出手机机体坐标系与参考坐标系，大地坐标系的姿态旋转q，通过线性加速度获得的手机加速度，此时的加速度是手机坐标系的加速度信息，为方便计算，加速度可以通过下面的方式表示：

其中表示加速度的模，表示加速度方向与手机坐标系的x，y，z轴的夹角信息，单位为rad，记为，计算方式如下：

其中，根据物理动力学公式有：

其中每一个位移都是一个向量，通过积分的方式获得路径信息。通过Dead reckoning算法来计算路径信息。Dead reckoning 算法又叫航位推测法，它主要是通过前一时刻的已知的或者估算出来的位置信息，以及上一个时刻估计出来的方向信息和经过时间间隔的平均速度来估算现在时刻的位置信息。由于上一个时刻的位置信息是由之前一系列的积分获得，会存在累计误差，需要在过程中减少这个累计误差。方向信息同通过手机姿态获得，手机姿态通过3.4章节的姿态融合解算得到，利用传感器数据，通过误差抑制的方式得到的姿态是得到的误差控制在最小，所以姿态求解得到的方向的误差是严格小的，时间差由传感器数据更新以及传感器同步算法所控制，更新频率为100Hz，即0.01s，为一个固定的值。所以通过以上分析，需要根据用户的行为来对速度进行优化以及约束以减少过程误差。

在实际情况中，手机静止于桌面，突然受到震动，其在世界坐标系下的加速度、速度以及路径有如下图所示：

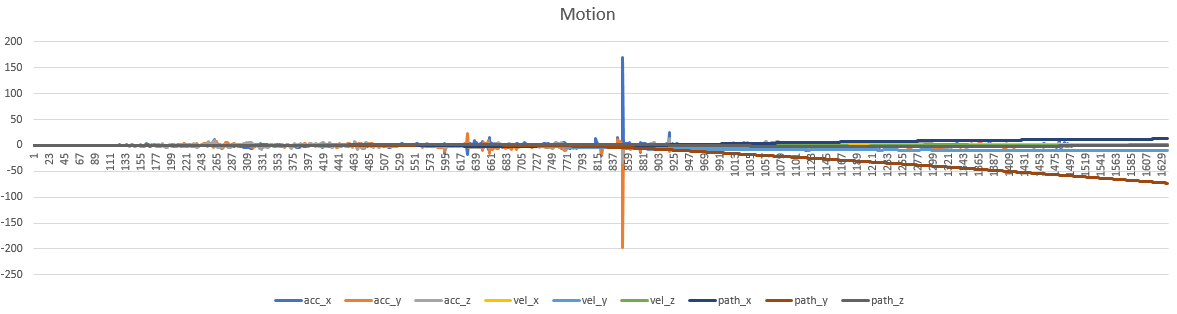


图3-14 受到噪声干扰的加速度、速度以及位移信息

在中间时刻投影在x轴与y轴的加速度发生一个尖锐噪声，之后导致速度x轴的速度与y轴的速度偏离坐标x轴，随着时间累积，得到手机在现实中x轴与y轴的偏移越来越大导致数据出现严重的累积误差，影响数据的准确性。为了减少或者抑制这种误差，首先需要从数据源头来进行误差的检测与抑制。

由于线性加速度数据噪声比较多，为了降低尖锐噪声对求解位移精度的影响，所以通过下面的策略来对原始数据进行滤波：



图3-15 加速度滤波算法

在得到参考系下的加速度时候，由于采样频率为100Hz，所以将其经过一个滑动窗口为13的中指滤波器过滤尖锐噪声，为了是数据尽可能的平滑，将中指滤波输出的数据在经过一个滑动窗口为13均值滤波器得到相邻13个数据的平均值作为当前时刻的参考坐标系下的加速度。由于两次滤波的滑动窗口大小为26，即数据从传感器采样到路径更新的时间间隔为0.26s。如果在用户不涉及高速移动的情况下，这个延时视为正常情况，为了解决这个延时问题，可以在姿态融合的时候通过预测的方式对手机姿态进行预测，以及使用历史加速度数据对0.26s之后的加速度进行预测来解决时间延迟问题，具体做法本论文不予深究。

## 小结

通过手机IMU传感器（加速度计，磁罗盘、陀螺仪）获取手机的一阶状态数据信息（磁罗盘数据）以及二阶状态数据信息（加速度计、陀螺仪数据）。通过导航算法获得手机的姿态以及位置信息。由于传感器数据本身受到系统噪声以及环境噪声的影响，源数据出现的误差会随着时间的积分，极大的影响定位精度，所以需要对软件的整体数据流从数据采集到中间数据转化与求解，以及通过多个传感器数据来通过数据融合来提高最终结果的精度，这一过程使用传感器数据同步与基于Kalman Filter的数据融合，在计算位移信息的过程中，需要通过滤波算法将速度、加速度异常的状态给过滤掉，然后使用Dead reckoning算法解算手机位置信息，从而提高手机定位精度。

# 第4章 用户交互设计及软件实现

在上一章节中详细介绍并且通过证明与推算，通过采集传感器数据然后解算得到手机的方位信息以及速度、位移信息，由于需要通过场景关联来实现室内不同场景之间的路径导航与手机定位，并且将数据以直观的方式展现以及便于浏览历史路径信息与场景回放，在这一章节将详细展开用户交互的设计以及软件的架构设计与实现。

## 基于Android的导航算法实现

在Android操作系统中，传感器数据主要通过系统服务的方式来实现对传感器数据的采集。对于传感器数据的获取主要遵循下面的步骤：



图4-1 Android传感器数据采集步骤

其中传感器属性包括传感器数据的来源种类、传感器采集此数据对应的时间戳（单位：ns）。通过得到这些信息，可以参考第3.1章节的传感器同步算法将传感器数据同步并且打包，包结构如下表所示：

表4-1 数据包格式

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 数据名称 | 详细 | 数据类型 |
| 0 | 磁罗盘数据 | X轴投影数据 | double |
| 1 | Y轴投影数据 | double |
| 2 | Z轴投影数据 | double |
| 3 | 陀螺仪数据 | X轴投影数据 | double |
| 4 | Y轴投影数据 | double |
| 5 | Z轴投影数据 | double |
| 6 | 加速度数据 | X轴投影数据 | double |
| 7 | Y轴投影数据 | double |
| 8 | Z轴投影数据 | double |
| 9 | 线性加速度数据 | X轴投影数据 | double |
| 10 | Y轴投影数据 | double |
| 11 | Z轴投影数据 | double |
| 12 | 时间间隔 | | double |

通过将同步好的传感器数据按照上述方式进行打包，打包成一个13×1的double数组来发送给解算线程来计算手机的姿态信息以及位移信息，具体的解算方式按照第3.2~3.5章节的误差抑制、坐标系定义以及转换、姿态求解以及位移求解的算法来实现。主线程（UI线程）与求解姿态的线程之间关系如下：



图4-2 位置姿态解算线程与主线程关系

通过上述方式可以将数据显示与数据解算部分分割开来，通过多线程通讯信机制将软件功能模块化划分。

## 三维场景设计与实现

此部分是软件实现的关键部分，需要将解算出来的路径数据以及姿态数据以一种更加直观以及方便用户查看的方式将数据呈现在界面并且支持用户多种方式查看历史数据。

### 4.2.1 数据可视化

数据可视化中需要将行走位移的历史数据以三维路径的方式显示出来，并且通过带有方向向量的手机坐标系表达在不同时刻下的手机姿态信息，通过将不同状态的数据特征通过改变描述该状态的点的大小以及颜色来区分不同状态的信息，例如当前位置状态、不同行走模式以及有场景关联点的状态表示。

为了形象的表示坐标点的参考关系，需要在场景中首先布置地平面（X-O-Y平面）以及垂直参考线（Z轴上半轴），最终设计效果图入如下所示：

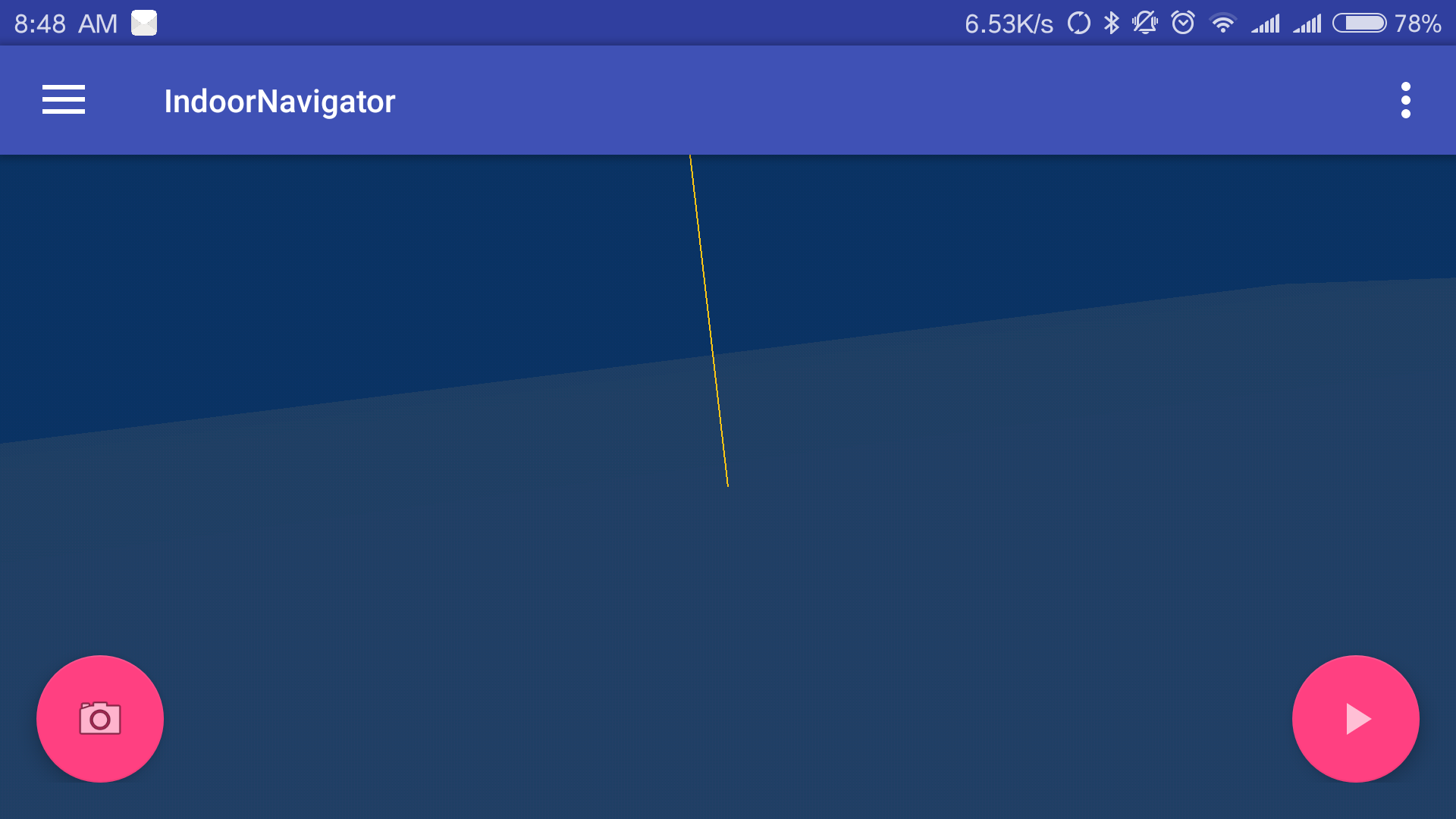


图4-3 场景地图效果图

其中下面的浅蓝色部分表示X-O-Y平面，也就是大地，中间的那条蛋黄色的线表示Z轴的上半轴，也就是垂直参考线。

三维数据的可视化主要通过OpenGL来进行绘制、渲染实现，所以数据可视化的设计模式符合OpenGL的设计模式。其中OpenGL坐标系与三维点的坐标系对比如下：

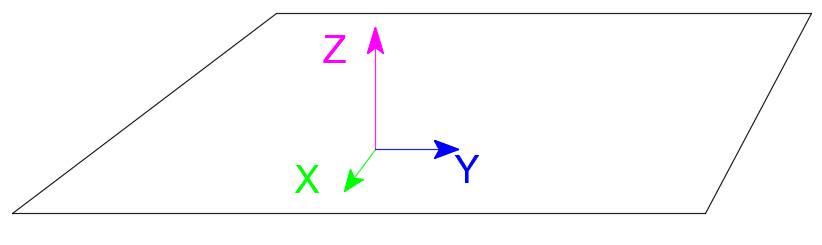
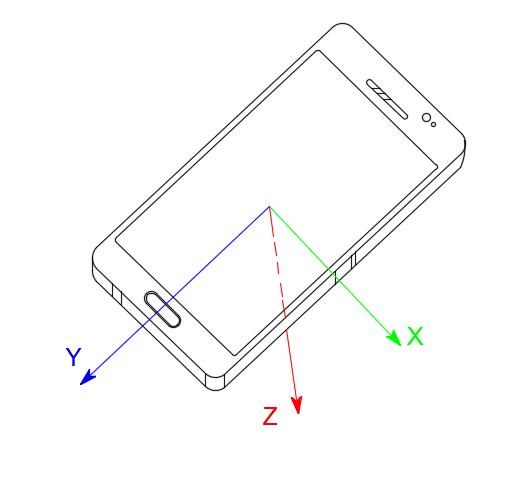


图4-4 OpenGL坐标系(左)与参考系(右)对比

所以将一个带有姿态的欧式空间的一个点转化为OpenGL绘制的带有具备表达手机姿态的一个空间三维几何图形的流程如下：



图4-4 通过OpenGL数据可视化流程图

通过此种方式就可以直观描述路径中每一个状态的详细信息，为了减少CPU的占用以及内存的消耗，此软件中采用异步请求刷新的方式来更新三维场景的绘制以及渲染，具体做法是只有当有用户与软件交互以及解算线程完成位姿解算时候通知主线程重新绘制渲染场景。

场景绘制中采取下面的树形图进行相关设计：



图4-5 数据场景结构图

World是整个3D场景的环境，其中包括光源的位置以及个数设置、相机位置与朝向设置以及需要展示的场景的结构渲染。其中光源与相机的位置与World相关联，通过设置好光源位置，不同材质的场景有着相应的反光与阴影，一个3D物体，颜色相同的不同面，有着不同的显示效果，使场景有更好的显示效果。通过操作场景的方式来对可视化的数据进行交互。

对于每一个状态点，均采用坐标轴+带有颜色的立方体来表达，如下图所示：

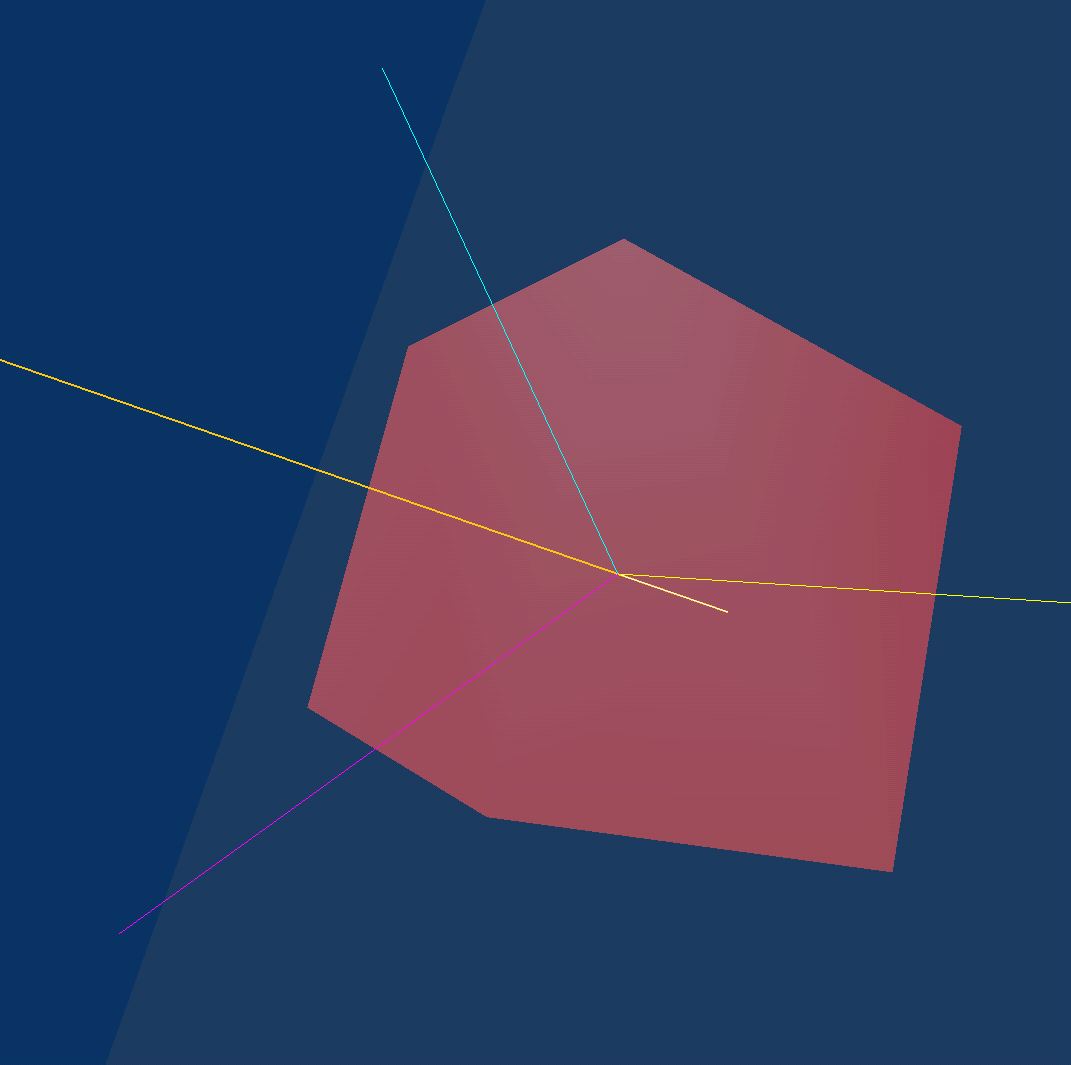


图4-6 状态点效果图

其中比较粗的黄线是世界坐标系的z轴，另外三个比较细的三根线，品红、靛青以及黄色分别为手机坐标系的z，y，x轴，用来直观表达手机坐标系与参考系的旋转关系。与机体系对应的半透明红色立方体表示当前时刻手机的状态。颜色以及大小对应关系为：

表4-2 可视化状态点与场景颜色对照表

|  |  |  |
| --- | --- | --- |
| 手机状态 | 颜色 | 大小 |
| 正常历史路径 | 绿色 | 小 |
| 选中或当前状态 | 红色 | 小 |
| 与场景关联的状态 | 紫色 | 大 |

### 人机交互

此部分主要目的是对三维场景的操控实现，主要包括触摸的手势识别以及场景中相应相关手势的设计实现。具体响应的操作有如下：

表4-3 人机交互手势对照表

|  |  |
| --- | --- |
| **操作** | **手势** |
| 物体拾取 | 点击 |
| 围绕中心旋转 | 单指拖动 |
| 缩放 | 双指不同方向拖动 |
| 平移 | 双指并拢拖动 |

为了便于人机交互，在建立场景时候采取下面的树形结构进行添加以及操作场景。

其中围绕手机旋转操作主要通过使用OpenGL中的rotate系列函数来实现，具体做法如下：

1. 监听手机手指在屏幕上的事件，如果为单指触摸并且手势为拖动，记录最开始点击位置(x, y)并且进行下面的操作；
2. 获取手机当前触摸位置，并且减去上一时刻的位置得到触摸移动位移(Δx, Δy)，并更新上一时刻的触摸位置(x, y)；
3. 通过调用OpenGL中的rotateX(Δx·w)与rotateY(Δy·w)来实现对图4.2.1.4中的场景进行旋转操作，其中w表示触摸旋转的灵敏度；
4. 拖动触摸结束旋转结束。

缩放和平移操作主要通过OpenGL中的translate系列函数来实现，具体做法如下：

1. 监听手机触摸事件，如果为两个触摸点，记录两个触摸点(x1, y1)与(x2, y2)，进行下面操作
2. 如果两个触摸点的距离不超过阈值，认为平移操作，记录第一个触摸点的位移(Δx, Δy)并更新上一时刻的触摸位置(x1, y1)与(x2, y2)，否则认为缩放操作，通过计算新的两点之间的距离与上一时刻的距离求差得到Δz
3. 通过调用OpenGL中的translate(Δx·w, Δy·w, Δz·w)来对图4.2.1.4中的场景进行平移操作，其中w表示旋转的灵敏度；
4. 双指拖动结束时场景平移结束。

物体拾取算法主要通过下面的算法来实现：

1. 初始化，将每一个通过路径求解算法得到的新的状态点标记为选中点并且高亮显示，将每一个场景中的状态点的3DObject对象存入一个数组中；
2. 监听手机触摸事件，如果为点击事件则记录手机点击位置(x, y)；
3. 通过当前渲染帧中所有场景物体的帧缓冲以及相机位置和手机触摸位置，通过函数reproject2D3DWS()可以得到触摸点在三位场景中三维的方向向量;
4. 通过世界中相机的位置信息，以及归一化的触摸方向向量，调用calcMinDistanceAndObject3D()可以计算出来所有与该射线相交的三维物体的集合；
5. 将所有计算出来的三维物体的中心与相机位置求解距离得到最小距离的物体即为选中的物体；
6. 为了避免选中X-O-Y平面或者z轴，所以需要第1步的所有状态点的数组来查找与选中物体ID相同的状态点，如果查找成功，则将上一个选中的状态设置为它原本的颜色，将查找得到的状态点设置为选中点并且高亮显示。

通过以上操作方式，可以实现较为全面的人机交互方式来方便用户对数据的浏览以及操作观察。

## 软件架构设计与实现

将数据与显示交互实现分离，整个软件架构采取下图的设计模式：



图 4-7 软件架构

其中数据分析性是基于良好的手机位移数据来进行更高层次的数据分析，主要将用户的路径状分为如下几类以及对应状态颜色如下：

表4-4 高级状态点与可视化数据颜色对照表

|  |  |  |
| --- | --- | --- |
| 手机状态 | 颜色 | 大小 |
| 正常历史路径 | 绿色 | 小 |
| 上楼梯 | 黄色 | 小 |
| 上电梯 | 靛青 | 小 |

这些判别方式主要通过手机历史路径数据特征来对之前走过的历史路径重新设置状态信息来体现不同的状态数据。

数据调试与阈值设置在下一节的路径优化与结果展示来阐述。

历史场景图片与其关联的状态浏览通过将所有与场景关联的状态点存入一个列表中，当与场景关联点被选中时候会从本地读取之前存储的对应的场景图片并且显示到手机界面上面便于查看。当需要浏览历史场景图片时候则加载所有与场景图片关联的状态点对应的图片并且进行展示。

分享功能则是将当前移动的路径信息通过截图的方式发送给QQ朋友或者空间。实现的界面菜单如下：

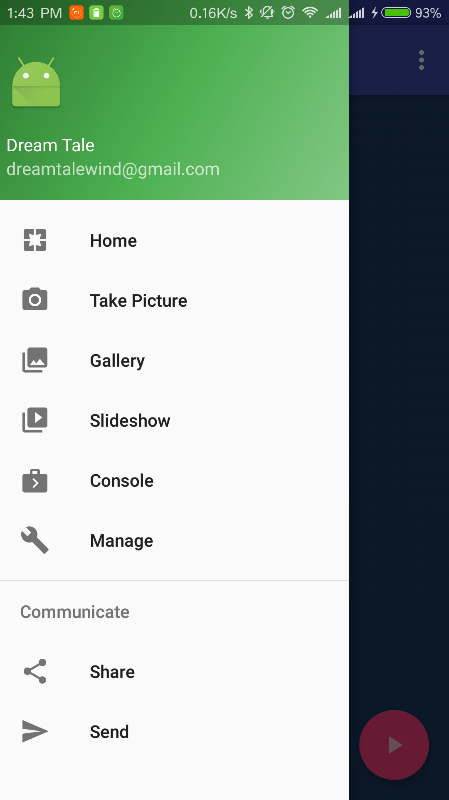


图 4-8 软件功能菜单

其中home为三维数据显示的主界面，Take Picture用于将当前状态点与拍摄的场景进行关联，Console显示过程解算的手机姿态数据以及速度、加速度和位移信息。

## 路径优化与结果展示

考虑到用户主要是在室内通过手机进行导航为行走，所以通过对速度以及手机的高度变化进行约束是可以有效减少过程误差的，在行走过程中，选择恰当的时刻初始化速度对于抑制路径求解过程中的累积误差也是必要的。

考虑到室内行走，所以默认将速度大于12m/s的值视为异常值舍弃，并且这一阈值可以通过Manage界面中手动设置。默认情况下是没有开启楼梯或者电梯检测，通过约束z轴数据到默认值1m来减少z上的累积误差。

由于速度信息是通过累积加速度数据得来，所以速度信息本身会存在累积误差，当加速度的值趋近于0的时候以及手机姿态角变化趋于稳定的时候，将速度置为0来减少累积误差，加速度的阈值默认为0.0001g，手机姿态的三个轴角变化默认最小值为0.005rad，所以当加速度值低于此值并且轴角变化低于此阈值的时候将速度初始化为0。

为了在合理的时机绘制更新数据可视化界面，默认设置当位移大小阈值在0.3m时进行数据更新与绘制。

其中得到的一些结果如下所示：

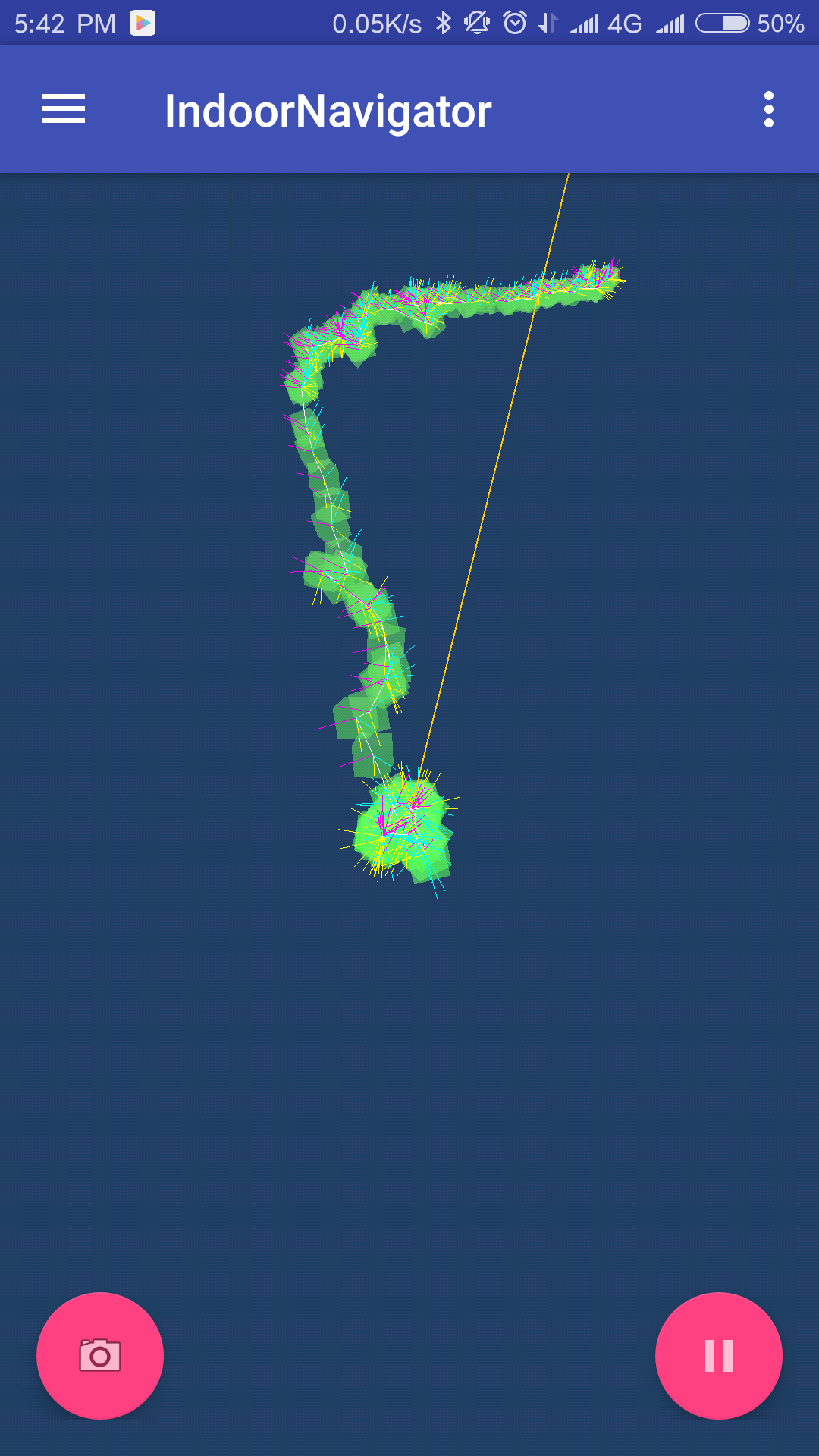
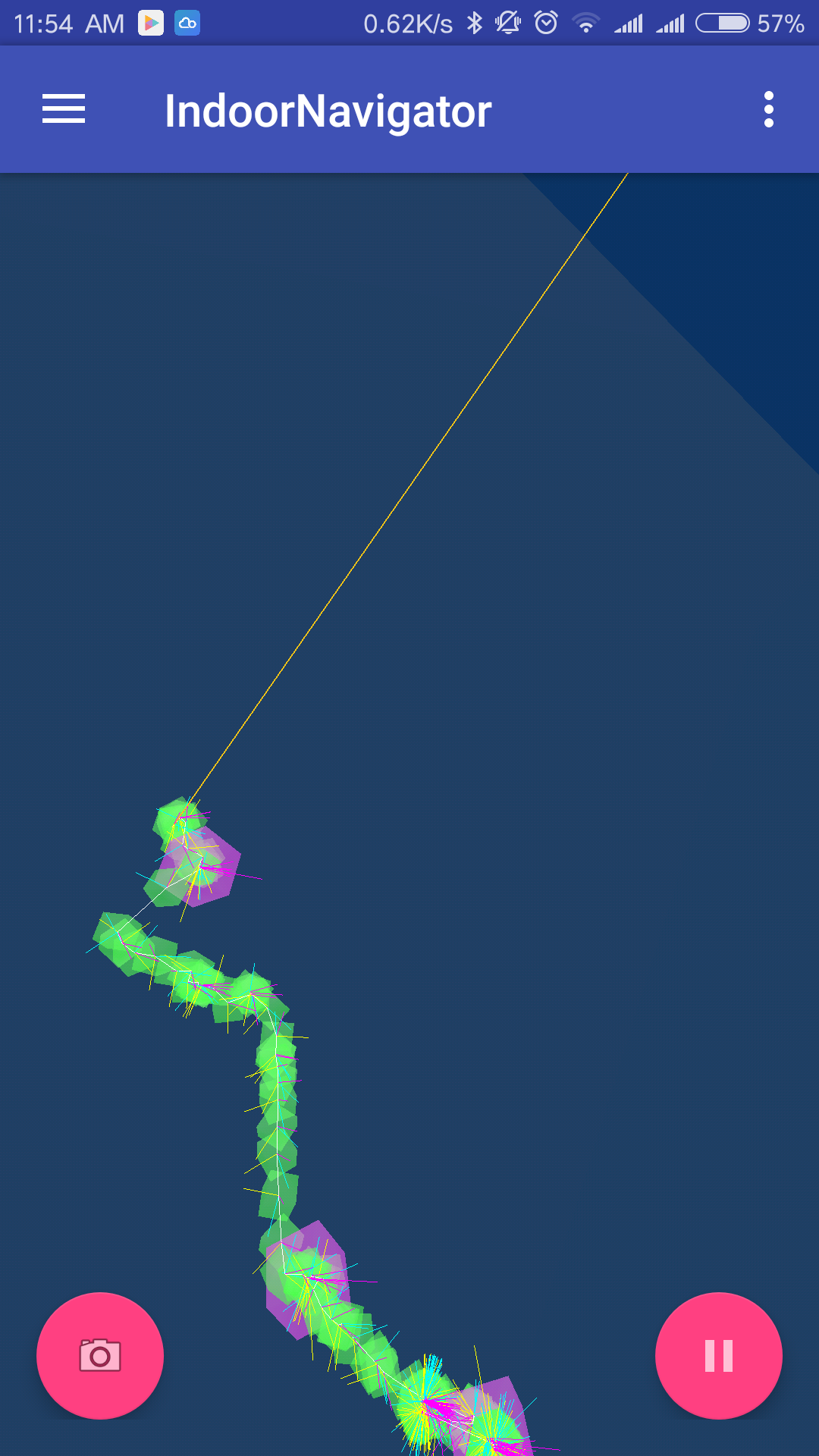
 

图 4-9 软件路径数据可视化展示

左图最开始为下楼数据，由于高度默认为固定值，所以得到一团数据，然后时向前走约50m之后向右拐弯；右图是将路径与场景进行关联得到的路径信息，其中紫色的比较大的立方体是与场景关联的点。

## 小结

通过Android手机上软件开发来通过订阅监听手机INS传感器事件获取手机的最基本的惯性数据，通过实现第三章的导航算法的相关设计来解算得到手机的位移、路径信息以及手机姿态信息；通过使用OpenGL开发来将手机的姿态数据以及手机在三维空间中的路径信息以一种直观、友好的方式展示出来，并且加入多种手势识别的方式来操作数据可视化的场景；将不同状态的数据以不同的颜色来展示，增加不同数据的直观表达性；对速度数据信息进行相关约束滤波处理以及初始化，可以有效地降低路径求解的过程误差；使用导航菜单的方式来分配软件的不同功能，使软件在使用以及调试都比较方便。

# 第5章 总结与展望

## 总结

在GPS信号数据无法使用的室内，，使用手机内部硬件自带的传感器来实现室内定位，从而尽可能少的对外部设备的依赖，如何合理有效的利用利用手机传感器数据来得到准确的手机位置姿态方向等信息是手机室内定位的核心问题。通过监听手机传感器数据，从充满大量系统以及环境噪声里获取廉价但是大量的手机状态信息，利用手机较为强大的计算能力来通过特定算法合理运用手机各个传感器信息数据来解算手机位置与姿态信息，通过将人在室内行走的路径信息与室内地图或者室内场景进行关联，可以很好的实现室内定位与导航。可以在大型复杂建筑内部实现对人的定位来起到导航作用，与GPS定位系统结合可以实现任何地方的定位从而实现导航地区的全覆盖。

本文通过对比多种室内定位方法，对室内定位的相关技术进行了仔细的研究，完成了以下的工作：

1. 通过手机INS传感器数据，通过使用EKF算法来综合加速度计、磁罗盘以及陀螺仪数据信息从而获得更加精确的手机姿态信息，利用准确的手机姿态来解算手机在参考系下的位置路径信息从而得到手机参考系下的位置实现定位。
2. 利用Android操作系统来开发相应软件，通过监听传感器数据信息来获取手机INS数据，并通过实现导航算法来得到手机的三维位置数据以及手机姿态信息。通过OpenGL将这三维信息可视化，来直观展示定位位置信息
3. 设计并实现了场景关联的室内定位系统，利用手机的摄像头对场景拍摄并且与路径进行关联，从而完成手机路径的相对形状与场景关联从而进行导航，设计并实现了这一导航系统。

## 展望

本文主要针对基于Android手机的室内导航进行研究，目前该系统仍然存在一些不足之处等待改进。结合上文提到的问题，主要提出以下几点作为今后进一步深入研究的出发点：

1. 本文仅仅使用INS传感器数据进行位置解算，虽然在各个环节进行误差抑制的设计，但是累积误差依旧无法消除，只能描述路径的大致形状，定位精度有限。在以后的工作中可以加入Wi-Fi指纹数据作为Ground Truth信息来进行传感器数据融合从而用来纠正累积误差。
2. 论文中手机的高度信息误差噪声比较严重，系统中通过约束高度来使用户在参考系中X-O-Y平面中运动路径信息更为精确，在实际测试中主要在同一楼层进行相关测试，效果满足要求，接下来需要进一步对高度的误差进行抑制。
3. 通过加载室内3D地图，结合地图来进行室内定位可以更加符合实际情况，将实际拍摄场景与3D室内地图与路径信息结合得到更好的室内定位导航效果。
4. 通过使用手机摄像头可以通过实现ORB-SLAM来得到手机在室内的位置信息作为Ground Truth信息进行融合来提高手机定位精度。

# 致谢

时光荏苒，大学四年的本科生涯转瞬间即将画上句号，我的本科生毕业设计也即将结束。在大学四年里，我感触良多，从刚刚开始军训刚刚开始接触大学什么都不懂的小孩终于成为现在稍微有自己见解的大人了大学四年里，接触了很多人，也结识了很多靠谱的伙伴，学到了很多知识，渐渐的领悟到如何做人如何处事，也深刻的意识到学习的重要性，同时也埋下了终身学习的种子，感谢大学里的老师们，同学们，伙伴们，你们教会了我很多。

在这里，首先感谢我的指导老师张磊老师。张磊是一个优秀的老师，他言传身教，善于聆听学生的想法并提出合理的建议，并且处处为同学着想。张磊老师也十分关心我的毕业设计工作，经常通过邮件等联系方式了解我的学习进度与毕业设计进展，在我的毕业设计中有着不可替代的作用。张老师还每周组织我汇报毕业设计进度，并时刻关心我的学习和生活。

接着，我还要感谢我的师兄们，在毕业设计过程中，师兄为我提供了大量参考文献和论文编写技巧，并且与我一起讨论算法的实现等具体问题。我们一起讨论系统的设计方案并且对方案进行不断的改进，在他们的帮助下我对毕业设计以及以后研究生的方向和生活有着更加深入的了解。

然后，我要感谢学院的老师同学们，学院老师同学都很nice，为我答疑解惑授业，在大学四年中承蒙你们照顾，感谢你们的陪伴与支持。

最后，真挚地感谢我的母校——北京理工大学，今日我以母校为荣，今后我依奋发向上，竭我所能，不辜负母校对我们寄予的厚望，亦不负与之交换的年华青春！

# 参考文献

[1] Wang H, Mou W, Suratno H, et al. Visual odometry using RGB-D camera on ceiling vision[C]// IEEE International Conference on Robotics and Biomimetics. IEEE, 2012:710-714.

[2] Han D, Jung S, Lee M, et al. Building a Practical Wi-Fi-Based Indoor Navigation System[J]. IEEE Pervasive Computing, 2014, 13(2):72-79.

[3] Shen S, Michael N, Kumar V. Autonomous multi-floor indoor navigation with a computationally constrained MAV[C]// IEEE International Conference on Robotics and Automation. IEEE, 2011:20-25.

[4] Zhu L, Ling J G, Zhang P, et al. Research on quadric orthogonal regression experiment on MAP craft of waxberry[J]. Science & Technology of Food Industry, 2012, 33(19):326-329.

[5] Hellmers H, Norrdine A, Blankenbach J, et al. An IMU/magnetometer-based Indoor positioning system using Kalman filtering[C]// International Conference on Indoor Positioning and Indoor Navigation. IEEE, 2014:1-9.

[6] Niemeyer I, Richter B, Dürr M, et al. Possible applications of “NavShoe/FootSLAM” for Indoor Navigation in Safeguards[J].

[7] Constandache I, Choudhury R R, Rhee I. CompAcc: Using Mobile Phone Compasses and Accelerometers for Localization[J]. Infocom, 2010.

[8] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-Time Single Camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007, 29(6):1052-67.

[9] Faragher R M, Harle R K. SmartSLAM - An efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing[J]. Proceedings of International Technical Meeting of the Satellite Division of the Institute of Navigation, 2013:1006-1019.

[10] Alzantot, Moustafa, Youssef, et al. CrowdInside: automatic construction of indoor floorplans[J]. 2012, 2012:99-108.

[11] Wu C, Yang Z, Zhou Z, et al. DorFin: WiFi Fingerprint-based Localization Revisited[J]. Eprint Arxiv, 2013.

[12] Wu C, Yang Z, Liu Y. Smartphones Based Crowdsourcing for Indoor Localization[J]. IEEE Transactions on Mobile Computing, 2014, 14(2):444-457.