



南京大學

本科畢業論文

院 系 匡亞明學院

專 業 計算機科學與技術

題 目 基於超聲波和 IMU 的室內定位系統

年 級 2016 級 學 號 161240005

學生姓名 陳勇虎

指導老師 謝磊 職 稱 副教授

提交日期 2020 年 5 月 20 日

南 京 大 学

本科生毕业论文（设计）指导情况记录

| | |
|------------------|---|
| 开 题 简 况 | 论文题目： |
| | <div>1、选题质量（简述选题与专业培养目标、专业要求关系、题目难度、工作量、创新性、理论性、实用性）</div> <div>2、开题意见：</div> <div>指导教师签名：</div> <div>年 月 日</div> |
| 中 期 检 查 | <div>指导教师检查论文的进展情况：（指导和培养学生查阅文献资料、综合运用知识、研究方案设计、研究方法和手段运用和外文应用等能力简况）</div> <div>指导教师签名：</div> <div>年 月 日</div> |

南 京 大 学

本科生毕业论文（设计）指导教师评阅意见

指导教师评语：

指导教师签名：

年 月 日

南 京 大 学

本科生毕业论文（设计）评阅教师评阅意见

评阅教师评语：

评阅教师签名：

年 月 日

南 京 大 学

本科生毕业论文（设计）答辩记录、成绩评定

答辩记录：

答辩记录人签名：

答辩小组评语：

答辩小组成员：_____

成绩_____

组长签名：

答辩时间： 年 月 日

南京大学本科生毕业论文(设计、作品)中文摘要

题目：基于超声波和 **IMU** 的室内定位系统

院系：匡亚明学院

专业：计算机科学与技术

本科生姓名：陈勇虎

指导老师（姓名、职称）：谢磊副教授

摘要：

基于 IMU 和智能手机的室内定位系统

关键词：IMU; 智能手机;

南京大学本科生毕业论文(设计、作品)英文摘要

THESIS: IMU

DEPARTMENT: School of Computer Science

SPECIALIZATION: Computer Science and Technology

UNDERGRADUATE: Yonghu Chen

MENTOR: Professor Lei Xie

ABSTRACT:

IMU

KEY WORDS: IMU, smart phone,

目 录

| | |
|--------------------------------|-----------|
| 目 录 | V |
| 1 绪论 | 1 |
| 1.1 研究背景及意义 | 1 |
| 1.2 国内外研究现状 | 1 |
| 1.3 研究内容与本文主要工作 | 1 |
| 1.4 本文的整体结构 | 1 |
| 2 声源定位相关研究工作 | 3 |
| 2.1 声源定位技术介绍 | 3 |
| 2.2 声源定位方案 | 3 |
| 3 智能手机感知与定位原理 | 5 |
| 3.1 双麦克风的节点定位 | 5 |
| 3.2 智能手机的传感器 | 6 |
| 3.2.1 加速度传感器 | 7 |
| 3.2.2 陀螺仪 | 7 |
| 3.2.3 方向传感器 | 7 |
| 3.2.4 磁力计 | 7 |
| 3.3 信号时延计算原理 | 7 |
| 3.4 基于到达时间差的定位算法设计 | 9 |
| 3.4.1 二维空间下的定位分析 | 9 |
| 3.5 | 10 |
| 4 基于双麦克风手机的声源定位方案 | 11 |
| 4.1 离线数据处理流程 | 11 |
| 4.2 数据采集 | 12 |
| 4.3 声音信号处理 | 16 |
| 4.3.1 信号除噪 | 16 |

| | |
|--------------------------------|-----------|
| 4.3.2 异常点移除 | 16 |
| 4.3.3 有效声音提取 | 18 |
| 4.3.4 TDOA 计算 | 18 |
| 4.3.5 TDOA 处理 | 19 |
| 4.4 offline 定位方法 | 21 |
| 4.5 offline 到 online 的迁移 | 22 |
| 4.5.1 概念阐述 | 22 |
| 4.5.2 online 算法模型 | 24 |
| 5 模型系统实现与实验分析 | 25 |
| 5.1 开发环境与平台 | 25 |
| 5.2 系统设计 | 25 |
| 5.2.1 多模块设计 | 25 |
| 5.2.2 界面与运行方式 | 25 |
| 5.3 实验与分析 | 25 |
| 5.4 本章小结 | 25 |
| 6 总结与展望 | 27 |
| 6.1 工作总结 | 27 |
| 6.2 前景展望 | 27 |
| 参考文献 | 29 |
| 致 谢 | 31 |

第一章 绪论

1.1 研究背景及意义

在室内环境(办公室,住所)中想要寻找一个小物品,诸如钥匙,硬币,可以说是一件令人费神的事情了^[1]。在游戏世界和虚幻的电影场景中,我们都或多或少接触过类似于“寻宝罗盘”的神奇宝物,它可以类似一个掌上的指南针,时刻指示“宝物”的方向。随着各式各样移动智能设备的广泛涌现,例如智能手机,智能手表,这些设备不仅自带了诸多传感器,同时可以为用户提供很好的UI界面。因此,如果可以利用这些设备实现类似于寻宝罗盘的功能,用于寻找我们室内环境中的小物品,无疑是有趣且富有创造性的。

随着对位置精度要求的提高,对于定位技术的研究也越加深入。传统的全球定位系统(Global Positioning System,GPS)^[2]是现代定位系统的首选,然而,在室内环境中,由于卫星信号受到建筑物的削弱,遮蔽,以及在复杂的室内环境下引起的信号反射,折射,透射等造成的多径和非视距传输现象,导致定位误差较大,所以卫星定位的精度已经无法满足室内定位的需求,因此以无线传感网为媒介的定位系统逐渐发展起来。

1.2 国内外研究现状

1.3 研究内容与本文主要工作

1.

1.4 本文的整体结构

本文共分为个章节,每个章节具体安排如下:

第一章绪论。

第二章相关工作

第三章实验

第四章总结与讨论

。 。 。

第二章 声源定位相关研究工作

2.1 声源定位技术介绍

噪声和异响是日常生活和工业生产中所常见的现象，在很多情况下，这些声音是令人困扰的。至于如何去解决这些噪声，我们首先需要去识别噪声并且能够定位出噪声的方向和位置，这就是声源定位问题的一个实例。当然，声源定位技术并不局限于噪声的定位，在特定的场合中，也有不同的应用。例如在会议室中，定位正在说话人的位置同样是声源定位技术的应用。

日常生活中我们似乎很容易根据听到的声音来辨别声源的大致方向，比如从身边急驶而过的汽车的来向，甚至车辆距离自己大概的距离。经过专业的训练，我们甚至可以让盲人参与到踢足球的运动中。这种常见的“听声辨位”尽管有一定的局限性，但是依然可以满足我们生活中很多的定位需求。借助于双耳效应的启发，为了提高声源定位的分辨力，声源定位技术提出了很多高效的定位方案。本章将对声源定位方案，双麦系统等进行调研。

2.2 声源定位方案

在现代城市中，利用机器视觉实现的机器人导航系统已是非常普遍。然而由于机器视觉本身存在的问题，使得利用机器视觉进行机器人导航在一些环境下表现的差强人意，比如，在光线不足的情况下，或者是在机器人的“视野”所不能到达的地方，机器人就找不到目标在空间中的位置，如果给机器人安装上“耳朵”，给予机器人声源采集装置，在进行人机交互的时候，人就可以通过语音告诉机器人自己现在所处的位置，机器人对声源信号进行处理，求解出声源的具体位置，然后制定关节运动决策，使交互对象重新进入其视野之内。

第三章 智能手机感知与定位原理

近些年来，基于声学传感器的定位系统越来越受到人们的关注，依赖于多个同步的麦克风，采集到的多路同步信号，用以后续的信号处理。随着无线传感网的不断发展进步，无线传感网定位问题也是一个比较热门的课题。而节点定位作为目标定位的前提条件，节点的定位精度也影响着目标定位的结果。然而，尽管目前有很多节点定位算法，但是因为功耗，成本等客观条件的限制，这些定位算法在很多场景中并不能很好的应用。因此本文提出了一种可以在室内环境下利用智能手机进行定位的方案。

随着智能设备的普及以及诸多传感器的集成，智能手机与硬件技术的协同发展有目共睹。无论是 ios 系统还是 Android 系统下的智能手机，内置传感器也越加丰富，也为智能手机应用的发展带来了更多的可能。

本文将会对利用智能手机进行双麦结点定位进行简要描述，并对本文中后续所涉及智能手机内置传感器进行讲解，最后系统描述本文中所使用的一些定位原理和算法设计。

3.1 双麦克风的节点定位

随着可移动智能设备的普及，尤以智能手机更为常见。自 2018 年起，大部分智能手机都开始集成两个甚至多个麦克风，除了底部的通话话筒以外，智能手机的顶部都还有一个“小孔”(不同的手机可能有所区别，例如 iPhone 手机的次麦克风可能位于主摄像头的附近)(如图 3-1)，这就是手机的降噪麦克风，它的主要作用是实现有噪声环境下的高质量通话。

手机上设有的两个电容式麦克风尽管作用不同，但是其性能相同，对声音信号的收集能力相同。利用双麦克风系统的时间同步(两个麦克风是集成于同一个设备的，因此完全时钟同步)特性，即使两个麦克风(如图 3-2)之间的距离较短(16cm 左右，不同手机略有差异)，也可以根据声音到达两个麦克风时刻的差异计算 TDOA，因此，利用双麦克风系统手机结点作为实验设备有效解决了研究过程中时钟同步的问题。

目前已经有很多利用双麦克风结点实现声源定位或者其他有趣的研究成果。例如利用双麦克风智能手机进行手势轨迹感知^[3];Aarab 使用双麦克风节点



图 3-1: iPhone8 的次麦克风手机位于摄像头附近

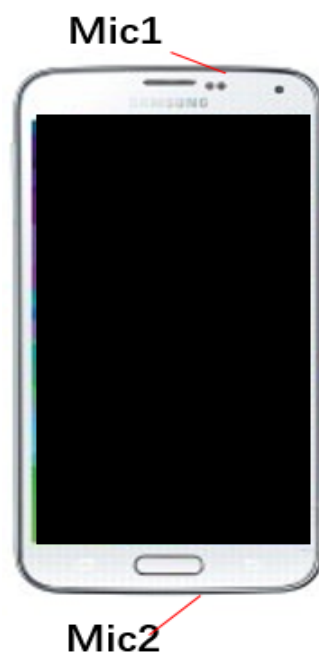


图 3-2: Dual-microphone phone

定位的思路测算说话人的位置^[4]。

3.2 智能手机的传感器

随着技术的进步，手机已经不再是一个简单的通信工具，而是具有综合功能的便携式电子设备。大多数的 ios 和 Android 设备都有许多内置传感器，用来测量运动，屏幕方向和各种环境条件，这些传感器能够提供高度精度的原始数据，非常适合用来测量设备的三维移动和定位，或者检测设备周围环境的变化，例如可以跟踪设备的重力传感器推断用户的手势和动作，同样天气应用中将会使用设备的温度传感器和湿度传感器。智能手机的内置传感器有硬件实现

和软件实现之分^[5]。基于硬件的传感器是内置在手机或者平板设备中的物理组件，这类传感器通过直接测量待定的环境属性 (如加速度，地磁场强度或者角度变化等) 来采集数据。基于软件的传感器不是物理设备，从一个或者多个硬件传感器获取数据，例如线性加速度传感器。具体可参考表 3-1。本章将会简述实验中涉及使用的几种传感器。

表 3-1: Android 平台支持的传感器类型

| 传感器 | 类型 | 说明 | 常见用途 |
|---------------------|-------|-----------|--------------|
| ACCELEROMETER | 硬件 | 加速力 | 动态检测 |
| AMBIENT_TEMPERATURE | 硬件 | 环境室温 | 监测气温 |
| GRAVITY | 软件或硬件 | 重力 | 动态检测 |
| GYROSCOPE | 硬件 | 旋转速率 | 旋转检测 |
| LIGHT | 硬件 | 环境光级 (照度) | 控制屏幕亮度 |
| LINEAR_ACCELERATION | 软件 | 加速力 | 监测单个轴向上的加速度 |
| MAGNETIC_FIELD | 硬件 | 环境地磁场 | 创建罗盘 |
| ORIENTATION | 软件 | 旋转角度 | 确定设备位置 |
| PRESSURE | 硬件 | 环境气压 | 监测气压变化 |
| PROXIMITY | 硬件 | 物体到屏幕的距离 | 通话过程中手机的位置 |
| RELATIVE_HUMIDITY | 硬件 | 环境的相对湿度 | 露点，绝对湿度和相对湿度 |
| ROTATION_VECTOR | 软件或硬件 | 设备 | 动态检测和旋转检测 |
| TEMPERATURE | 硬件 | 设备 | 监测温度 |

3.2.1 加速度传感器

3.2.2 陀螺仪

3.2.3 方向传感器

3.2.4 磁力计

3.3 信号时延计算原理

智能手机的两个麦克风处于相同的环境下，因此采集到的两路音频信号具有较强的相关性。当求解某一声源的位置时，声源信号到达两个麦克风的时延往往具有重要的作用。对于到达时延差 (TDOA) 的获取主要有两种方法。

第一种方法依赖于信号到达时间 (TOA) 的测量, 通过 TOA 的差值来求得。TOA 对系统的时间同步要求就会很高, 在信道传输特性相似的情况下, 这种办法可以减少多径效应带来的误差。如图 (3-3), 根据 TOA 可以得到关于 TDOA 的方程如下:

$$\sqrt{(x-x_1)^2+(y-y_1)^2}-\sqrt{(x-x_3)^2+(y-y_3)^2}=c(t_1-t_3) \quad (3-1)$$

$$\sqrt{(x-x_2)^2+(y-y_2)^2}-\sqrt{(x-x_3)^2+(y-y_3)^2}=c(t_2-t_3) \quad (3-2)$$

由于依赖于 TOA 的值, 微小的测量误差就会引起 TOA 较大的误差, 进而可能导致 TDOA 的精度降低, 因此在后续的实验中并没有采用这种方式求解时延。

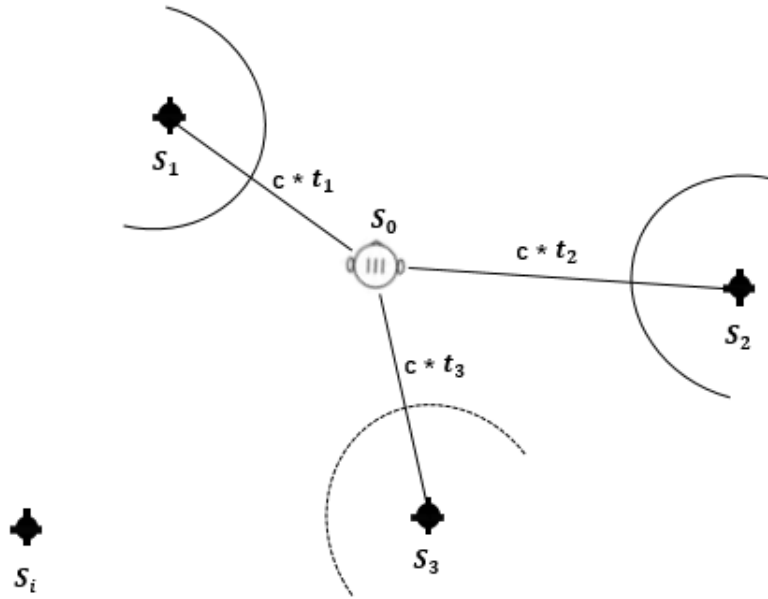


图 3-3: TDOA 算法示意图

第二种方法是将接受到的信号进行相关运算, 从而得到 TDOA 的值。在实际应用中, 系统往往很难做到严格的同步, 因此常常用相关估计的方法求解 TDOA 的值, 再进行后续的定位计算。在无线传感网中, 这种方法对网络的要求比较低, 因此更具有实际意义。

广义互相关的基本思想是对两路信号 $x_1(t)$ 和 $x_2(t)$ 进行预滤波, 然后再求互相关函数, 其原理框架如图 (3-4) 所示, 公式表达为:

$$\tau_{\text{delay}} = \arg \max_{t \in \mathbb{R}} ((f \star g)(t)) \quad (3-3)$$

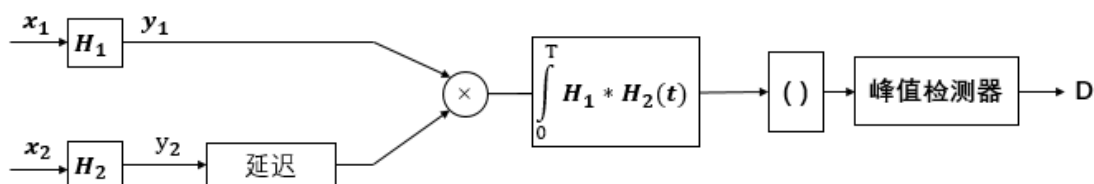


图 3-4: 互相关函数求解时延流程

在利用线性卷积计算时，常常可以借助线性卷积的关系来实现利用频域 FFT 的方式计算互相关，这种方法将比传统的时域计算法极大的减少计算量。根据维纳辛钦定理: 平稳信号的自相关函数与其功率谱密度互为傅里叶变换对，因此相对应的，利用频域下的功率的互功率谱可以计算时域的互相关函数。最后通过峰值检测器求解出时延。需要说明的是，麦克风采集的信号是离散的，因此时延可以看做是几个采样点数的差距，通过麦克风的采样率，我们将可以把计算的采样点数的差距转换成时间的差值。

3.4 基于到达时间差的定位算法设计

上一节简要描述了利用互相关方法求解时延的方法，这一小节将会描述利用求解出的 TDOA 进行后续的声源定位。

3.4.1 二维空间下的定位分析

我们首先从二维空间下进行定位分析。首先可以明确的是，根据到达时间差，结合声速 (v_{sound}) 我们可以求解出声源到两个麦克风的距离差 (δd)，因此在二维空间下，相对于手机中心和手机坐标系，如图 (3-5) 所示，我们可以得到一个以智能手机麦克风节点为焦点的双曲线，而声源则在其中的某一支曲线上。同理，当我们移动手机至别的位置，那么就又可以得到一支曲线，如图 (3-5) 而两个曲线的交点就是声源的所在位置。

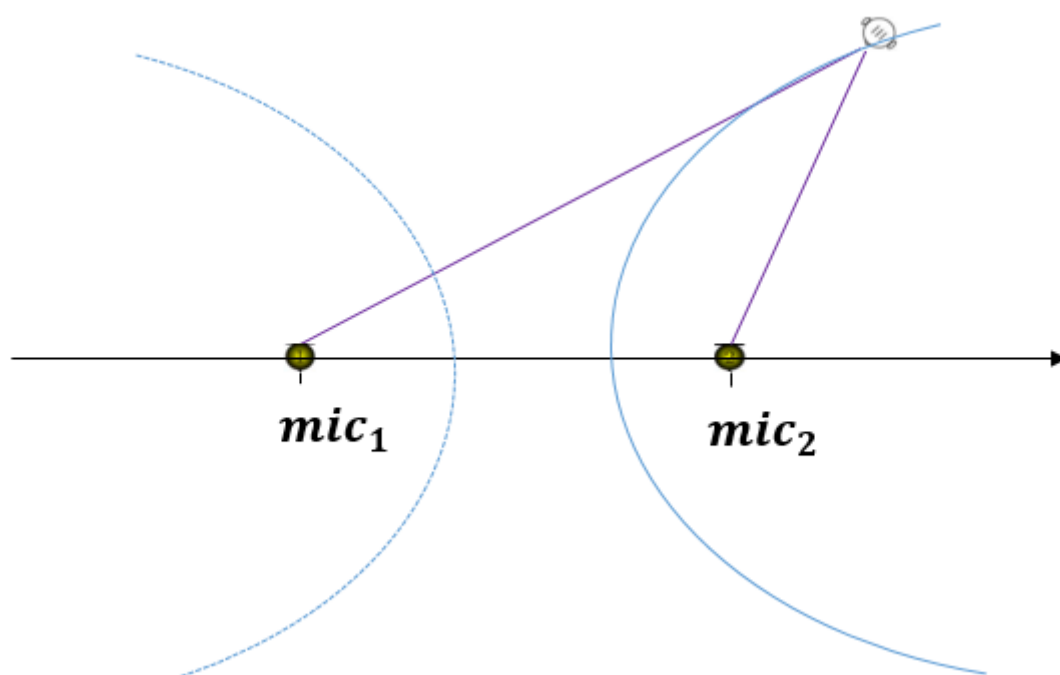


图 3-5: 2 维双曲线定位

3.5

第四章 基于双麦克风手机的声源定位方案

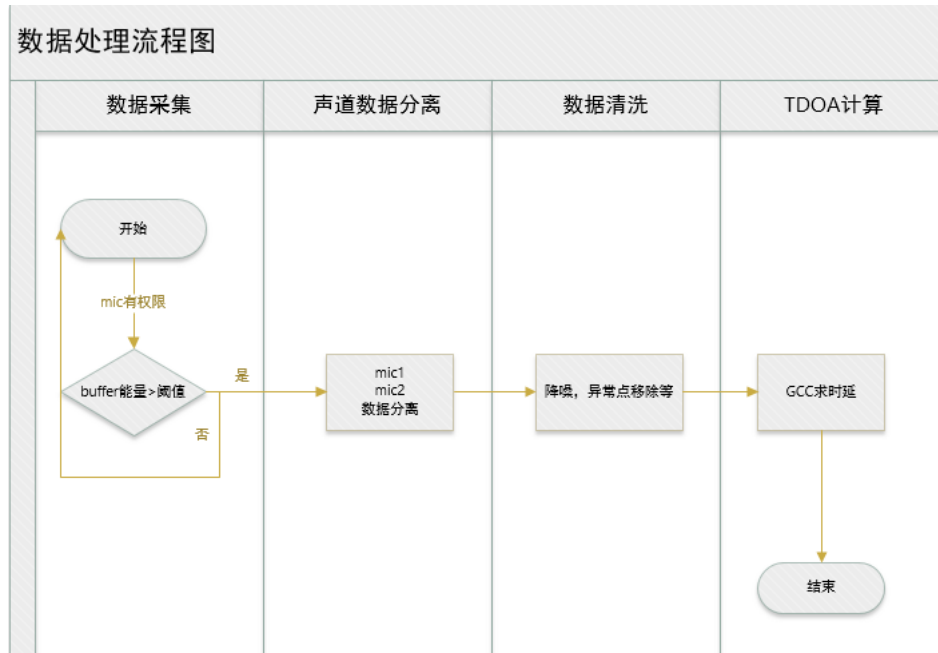
近些年来，智能手机已经成为人们生活中不可缺少的一部分，手机上集成的传感器的发展，使得智能手机的功蟹越来越强大。在前面我们已经简述到大部分智能手机已经配备两个麦克风，用于消除我们在日常通话中的环境噪声，并且可以很好的解决的时钟同步问题。因此，利用智能手机的优势，通过获取同一个手机两个麦克风的声数据，我们从中过滤出我们想要的声音信号，得到两路同步的声音信号，利用智能手机计算出两个麦克风的到达时间差 (TODA), 对 TDOA 信号进行后续的处理，就可以实现一些有意义的功能。

在前一章的基础上，本章详细描述利用双麦克风手机实现定位功能的诸多细节性工作。

4.1 离线数据处理流程

本小节中，首先介绍 offline 算法的处理流程，并逐渐过渡到最终系统使用的 online 算法的处理流程。offline 算法中，智能手机会收集声音文件并产生一个 wav 文件，通过 MATLAB 分析该 wav 文件，计算出 TDOA。该 offline 算法处理流程是为了通过处理一段时间窗口内，设备没有发生大幅度状态变化 (包括角度，位置等参数) 的情况下，计算出静态声源相对于该位置的方向 (Speaker Direction Finding (SDF))。在一段时间窗口内的处理流程如下图所示 (4-1)。

图 4-1: offline flowchart



4.2 数据采集

本文利用具有双麦克风系统的智能手机 (Samsung 5) 作为传感器结点的进行声音信号的手机, 在分析阶段, 使用 MATLAB 科学工具对采集到的信号进行解析显示。我们利用另一台手机发出了一个频率在 2000Hz-2700Hz 变换的线性调频信号, 如图 4-2 所示, 图中蓝色的标记为 LY 的数据以及红色的标记为 RY 的数据, 分别代指两个麦克风采集的数据。图中的上一部分图中, 横坐标代指数据采样点, 纵坐标是对数据进行标准化后的结果, 因此无量纲。但是在下部分图中, 横坐标是标准化的频率, 单位为 $\pi * \text{radians/sample}$ (频率标准化的相关概念不做赘述, 可以参考^[6]), 纵坐标为能量谱, 这里我们很容易看出两个麦克风的采集的数据的能量谱是非常接近的, 换句话说, 两个麦克风的采集能力非常接近。这里我们对原始数据取样其中发生部分连续的 400 个点。如图 4-3 所示, 同样我们依然可以看出两个麦克风的采集能力是非常接近的, 另一方面, 即便我们暂时无法确定 TDOA 的值, 但是我们可以清晰看出两路声音数据是差了几个样本点的, 因此只要计算出相差的样本点数, 根据采样率 (44100Hz) 我们就可以求解出 TDOA 的近似的结果。

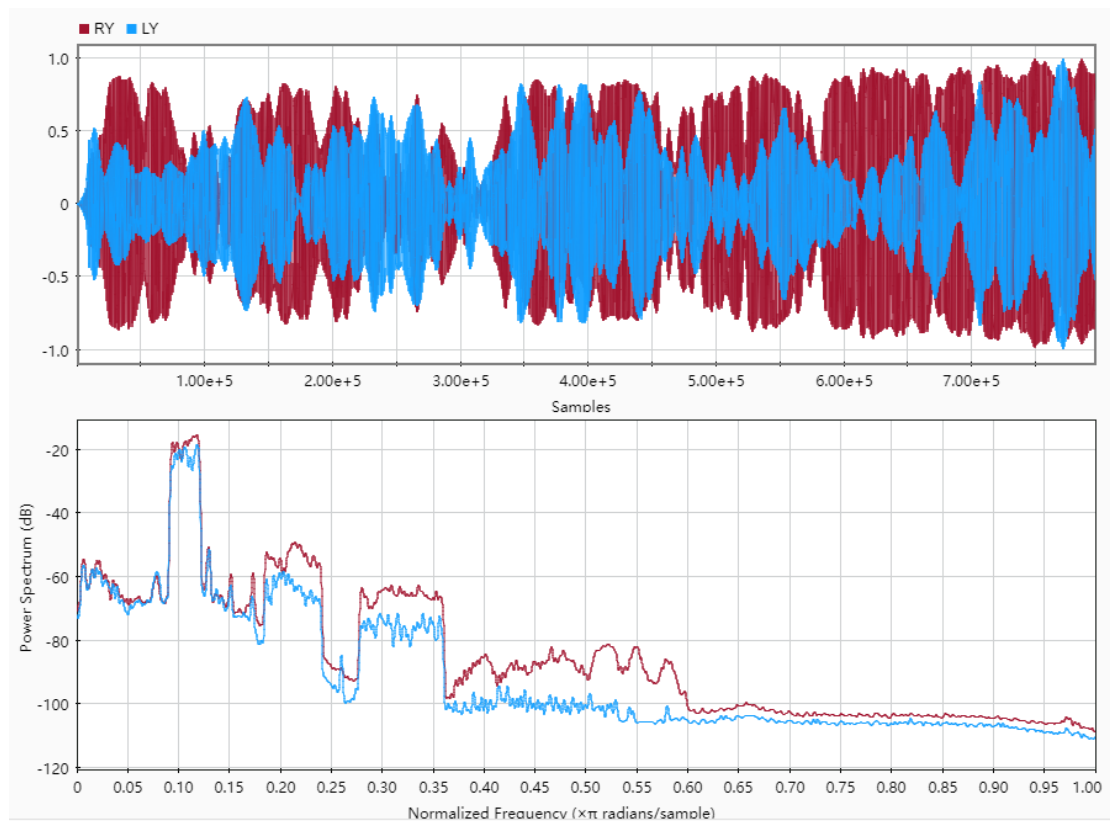


图 4-2: 双麦克风信号

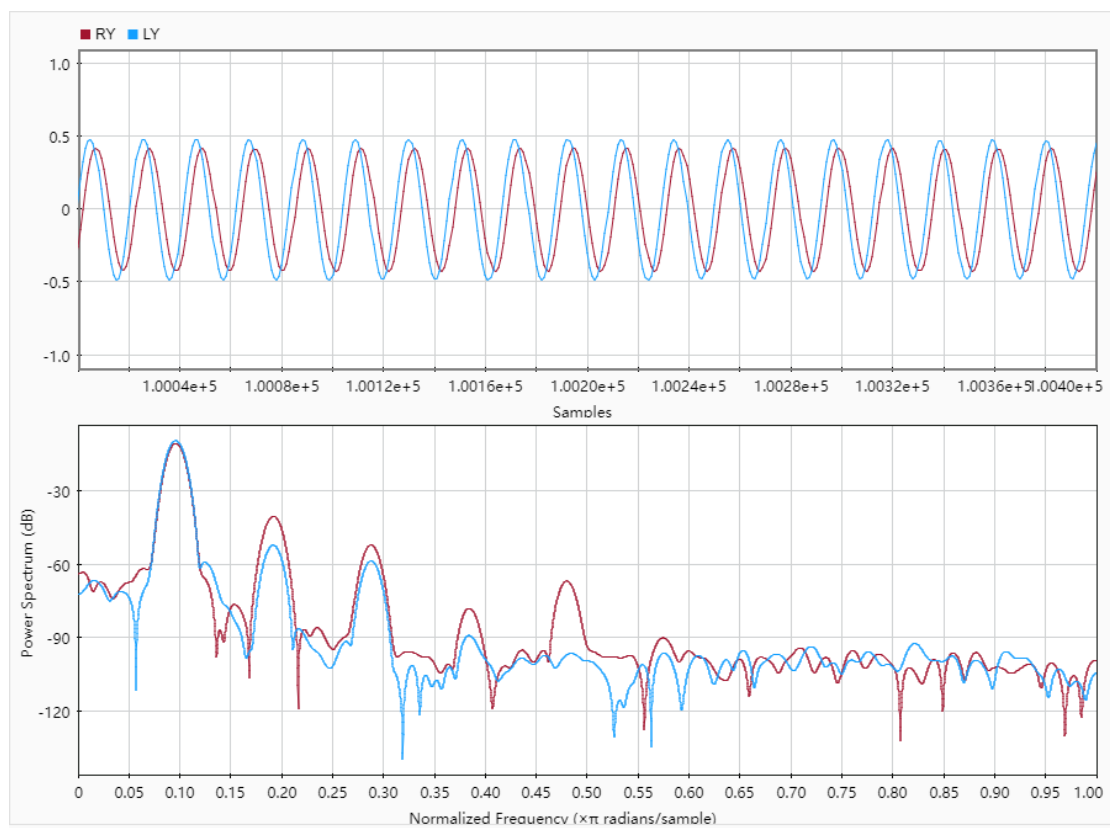


图 4-3: 双麦克风信号取样

下表中 4-1 展示了采集声音信号的手机参数，实验中使用了 SamSung S5 手机作为实验设备，S5 集成了两个同步的麦克风，分别位于手机的底部和顶部，而这个两个麦克风的声​​音采集能力完全相同。目前的智能手机最高采样率都可以达到 44100Hz，每秒钟就可以获得 44100 个声音信号的样本值。

表 4-1: SamSung S5 设备参数

| 参数 | 描述 |
|-------|---------|
| 灵敏度 | -26dB |
| 麦克风距离 | 16cm |
| 采样频率 | 44100Hz |

本文利用内置双麦克风的 S5 手机进行实时采集声音信号，首先介绍 offline 处理的，对信号进行汇总处理和分析。在底层获取的声音信号实际是个一维数组，当然利用 MATLAB 进行读取，会转换成两个一维数组，分别代表两个麦克风的数据。在原始的底层存储中，一维数组的值即是采集的到数据。如图 4-4 所示，实验中，我们采用 16bit 的 stereo 的格式进行采集，因此根据图中的存储方式很容易分别获得两个麦克风的数据，并且两组数据是完全满足时钟同步的。为了环境噪声或者其他声音的干扰，本文中设置了一个基于能量的硬阈值，当每个 buffer 信号的能量低于该阈值时，则认为没有声源发生，当平均能量高于该阈值时，认为有声源发生，从而进行后面的处理。



图 4-4: 数据存储方式

数据采集阶段核心代码如 4.1 所示. 手机调用麦克风不断检测周围环境的声​​音信号，当信号端的平均能量高于阈值时，系统判断声源发声，并且将两路的声​​音信号进行分离，用于后续的处理。

算法 4.1 dual-microphone signals acquisition(双麦克风信号采集)

```

1: mAudioRecord.startRecording()
2: //系统检测周围环境中的声音信号
3: while isRecording do
4:   //读取两路声音信号
5:   read = mAudioRecord.read(mBuffer, 0, BUFFER_SIZE)
6:   if read <= 0 then
7:     return false
8:   else
9:     //根据设定的阈值判断是否有声源发声
10:    sum = 0
11:    for i = 0 to read do
12:      sum += Math.abs(mBbuffer[i])
13:    end for
14:    //达到设定的阈值，认为有声源发声
15:    if sum / read > thresholdValue then
16:      //两路信号分离
17:      index = 0
18:      for i = 0 to mBuffer.length/2 do
19:        if i % 2 == 0 then
20:          System.arraycopy(mBuffer, i * 2, mBuffer1, 0, 2)
21:          LY[index] = ((mBuffer1[0]&0x000000FF) | (((int)mBuffer1[1])<<8))/
22:            32768.0
23:          LYList.add(LY[index])
24:          index = i / 2
25:        else
26:          System.arraycopy(mBuffer, i * 2, mBuffer2, 0, 2)
27:          RY[index] = ((mBuffer2[0]&0x000000FF) | (((int)mBuffer2[1])<<8))/
28:            32768.0
29:          RYList.add(RY[index])
30:        end if
31:      end for
32:    end if

```

```

31:      //other code for further data-processing
32:  end if
33: end while

```

4.3 声音信号处理

从底层获得的两路原始信号包括了真实声源信号，环境噪声以及其他各种异常点，本小节将会利用滤波器降低环境噪声的影响，同时使用加窗移动的方式去除采集中的异常值，对原始的信号进行清洗处理。

4.3.1 信号除噪

常见的去噪方式就是使用滤波器，这里我们使用带通滤波器，降低环境噪声的干扰。在 MATLAB 通过 `filter`, `designfilter` 函数我们很容易实现我们需要的功能，在后续的 online 算法处理流程中，因为程序运行在 Java 平台上，所以 Android 方面使用 `dsp-collection` 这一个 jar 包即可满足我们需要的功能。

例如当我们的声源频率范围在 2000Hz-2700Hz 时，那么在除噪的过程中，我们就需要通过一个带通滤波器，减益非该频率范围的成分。如图 (4-5) 所示，横纵坐标信息同图 (??) 通过滤波，不在制定频率范围的数据被大大减益，从而降低了对实验结果的影响。

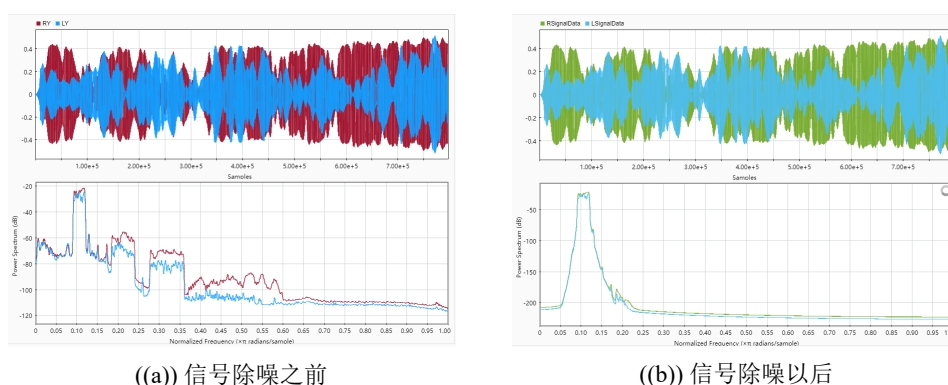


图 4-5: (a) 数据除噪之前。(b) 数据除噪以后。

4.3.2 异常点移除

对实验结果影响较大的还有异常数据，例如在稳定的信号中仍会有一些突变的信号。为防止信号之间的干扰，这里采用分别对两路信号进行处理。值得一提的是，由于采样率高达 44100Hz，所以处理过程中去掉一些数据

并不会造成较大的影响。这里，我们采用滑动窗口的方式，使用加权平均移动 (WMA^[7]) 的方式进行处理。加权平均移动认为：同一个窗口内的不同的数据对目标值都有着不相同的权重因子，因此目标值就可以看成窗口内数据和一个固定的权重函数的卷积。由于数据本身异常点并不是很多，并且使用传统的WMA就可以获得较好的效果，因此记 signal_t 为 t 时刻的数据，采用下面的公式更新 t 时刻的信号值：

$$\begin{aligned} \text{signal}_{t_{\text{new}}} = \text{WMA}_t &= \frac{np_t + (n-1)p_{t-1} + \cdots + 2p_{(t-n+2)} + p_{(t-n+1)}}{n + (n-1) + \cdots + 2 + 1} \\ &= \frac{n \times \text{signal}_{t-1} + (n-1) \times \text{signal}_{t-1} + \cdots + 1 \times \text{signal}_{t-n+1}}{n + (n-1) + \cdots + 2 + 1} \end{aligned} \quad (4-1)$$

经过上述处理后，数据变得更加光滑，并且从图4-6中已经可以隐约可见TDOA的大致范围。这里，LSD为第一个麦克风降噪后是数据，LSignaData代表降噪后数据经过异常点去除操作后的数据。

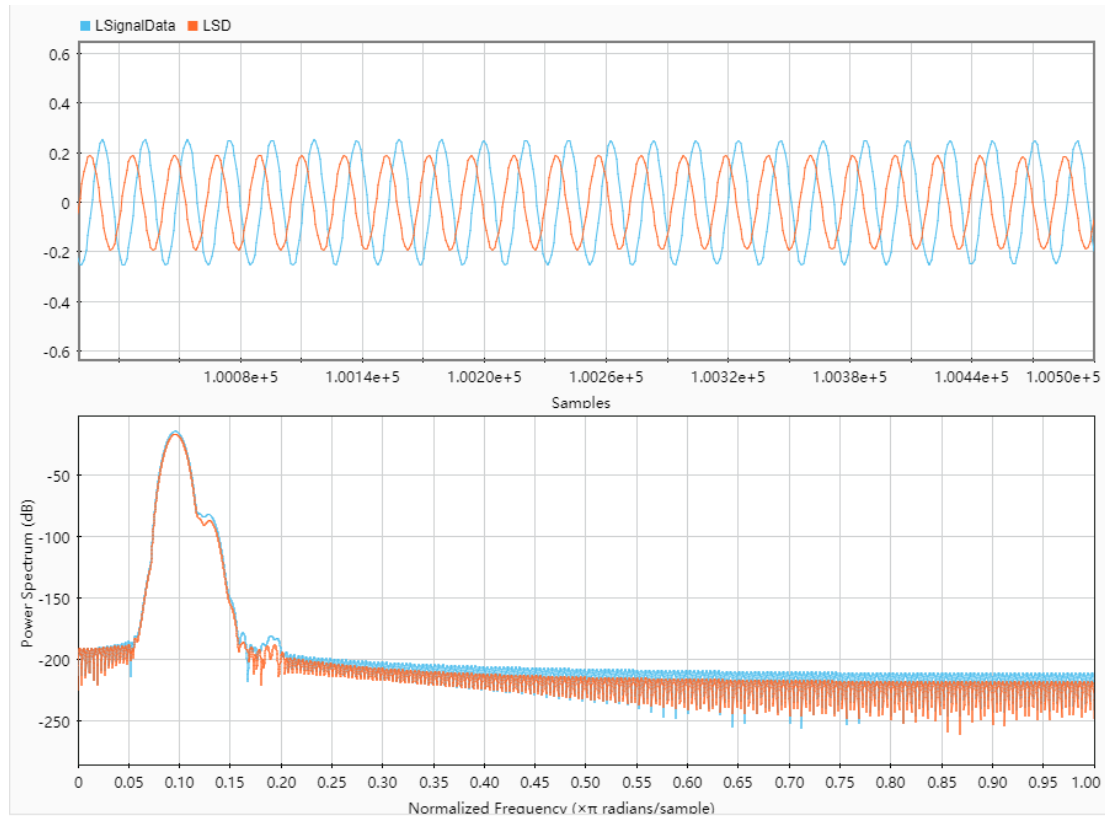


图 4-6: 异常点移除后的数据

4.3.3 有效声音提取

阈值中值判断，online 算法中去除

TODO

4.3.4 TDOA 计算

本小节基于前期的信号处理获取到两路同步的声音信号，调用传统的 GCC 算法对两路同步信号进行 TDOA 计算 (估计)。

由于接收到的声音信号来自于同一个声源，因此两路信号之间具有较强的相关性而与噪声无关。广义互相关方法恰好可以得到两路信号之间的时延差。其基本思想是对两路信号 $x_1(t)$ 和 $x_2(t)$ 进行预滤波，然后再求互相关函数，其原理框架如图 (3-4) 所示，公式表达为：

$$\tau_{\text{delay}} = \arg \max_{t \in \mathbb{R}} ((f \star g)(t)) \quad (4-2)$$

需要注意的是，这里计算出的“时延”实际是两个信号相差的采样点数，但是，由于采样率为设定好的 44100Hz，因此计算出相差的采样点数，经过转换就可以得到对应的时延，也就是所求的 TDOA，后面我们用相差的采样点数代指 TDOA，转换公式无需赘述。可以明确的是，当 $\text{TDOA} < 0$ 时，声音先到达第二个麦克风， $\text{TDOA} > 0$ 时，声音先到达第一个麦克风， $\text{TDOA} = 0$ 时，声音同时到达。降至二维空间下，详情如图 (4-7) 所示。

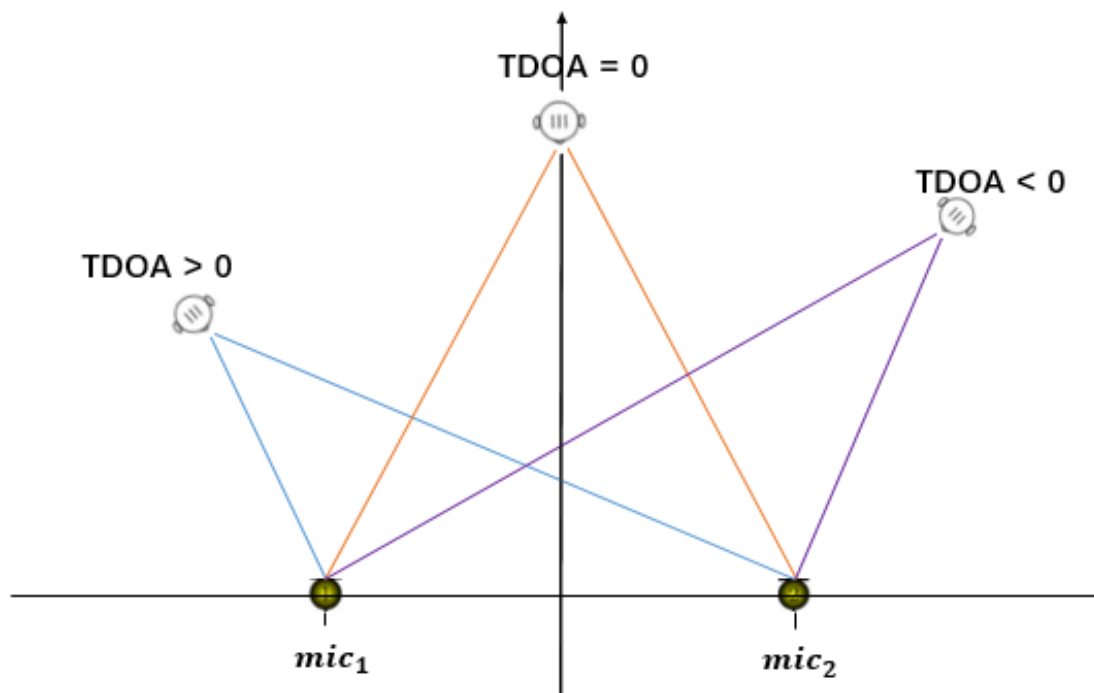


图 4-7: $TDOA > 0$, $TDOA < 0$, $TDOA = 0$

4.3.5 TDOA 处理

利用智能手机作为传感器，计算指定声源到达同一个手机麦克风的 TDOA，可以得到大量的数据并且计算出很多个数据段的 TDOA。因此鉴于数据量多的优势，那么我们就可以利用此优势进行数据的筛选。即便我们的某些 TDOA 测量并不是足够“精确”，但是依然可以辅助进行方向探测。

回到最初的系统设置，我们设置的采样率 44100Hz，编码为 16bit，我们暂时不考虑高编码编码的精确度，但显然的是，由于使用的 bit 数较大，因此数据量化的范围也更大，因此数据也是更精确的。对于 44100Hz 的采样率，麦克风每手机一秒钟，就可以采样到 44100Hz 的采样点。实验中发现，如果将太多的数据一次性进行计算，由于我们没有加窗等操作，加上声源功率较小，信号衰减等，大部分测量结果都是不准确的。然而，另一方面，即便数据过多造成了结果不够准确，但是对数据分段，对该一秒钟的部分数据进行 tdoa 计算，在原理上也是整段数据的 tdoa 的估计，因为延迟是“全局”的延迟，通过实验我们也发现这一个问题。因此，鉴于这种思想，我们可以将信号进行分段，当然，由于在后续实验中设备会一直处理数据，因此我们并不是对 1s 数据分段，而是每 k 个连续数据进行分段。从而，对两个麦克风相同次序的数据段进行 TDOA，依然可以得到全局的 TDOA，如图 (4-8)。

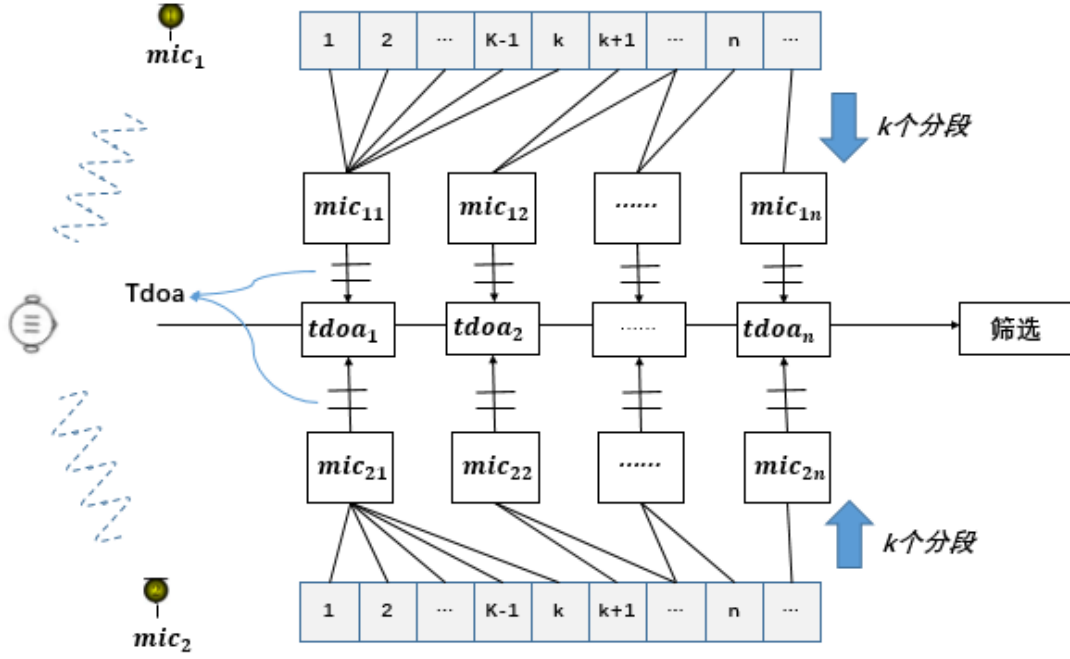


图 4-8: 数据分段示意

数据分段之后，并进行对应的数据段的 tdoa 计算，从而可以产生一系列的 tdoa 值。在实际计算中，我们设定 $k = 2048$ ，很明显的是，对 k 的设定是一个 trade-off 的问题，若 k 值太小，数据点太少，基本无法得到合理的 tdoa 值，如果 k 值较大，精度会有影响，同时，也就没有分段的意义了，实验中 k 暂取用 2048。另一方面需要明确的时，实验中手机麦克风距离约为 16cm, 因此 tdoa 绝对值的最大值 $|Tdoa_{max}|$ 应该满足下式：

$$|Tdoa_{max}| = 16/v * 44100 \leq 16/33700 * 44100 \approx 20.9 < 21 \quad (4-3)$$

因此 Tdoa 的范围应该在 -21 至 21 之间，这就是 TDOA 合理值的区间。另一方面，虽然由于离散采样的问题，Tdoa 的值最后应该取整数，但是我们可以肯定的是，即便有 0.5 个采样点的误差， $\delta distance = 0.5/44100 * v \leq 0.5/44100 * 34000 \approx 0.38cm$, 对于 16cm 的麦克风距离来说，这个值微不足道。

前面中我们提及到，部分信号计算的 tdoa 应该符合或者接近于全局的 tdoa 值，基于这种思想，我们使用以下的筛选处理方案：

(1) 每 m s 中，准确来说，应该是每 $[44100 * m/2048] * 2048$ 个数据，进行一个 TDOA 计算，在这一次 TDOA 的计算中，按照前文叙述的方法，每 $k(k=2048)$ 个数据计算一次 TDOA 的计算，因此可以产生个 $[44100 * m/2048]$ 个 TDOA 的值。

(2) 对于这些 TDOA 的值，如果有超过一半的值相同并且介于合理值区间，那么我们就可以认定该 TDOA 有效，并且其符号包括数值都可以用于后续的处理算法中。

(3) 如果无法满足 (2) 的条件，但是大部分值甚至所有值都是大于 0 或者小于 0，或者等于 0，只有极少的符号与整体不一致，这里我们定义极少为占比不超过 1/7，那么仍然可以认为该组 TDOA 有效，不过此时我们只使用 TDOA 的符号，也就是大部分值的符号，用于后续的方位判断。对于特殊情况，我们考察这些数值的众数，如果只有一个众数，如果众数在“大部分值”占比超过一半，那么我们使用该众数作为 TDOA 的数值；如果存在多个众数，并且数值差值不超过 2，并且所有众数在“大部分值”中的占比超过一半，我们可以使用这些众数的均值作为 TDOA 的数值使用，否则，在 (3) 中，我们只使用 TDOA 的符号。

(4) 如果 (2)(3) 的处理条件都无法满足的话，我们认为这一组的 TDOA 无效。

但是，当 TDOA 不为 0 的时候，信号能量会有较为明显的对比，因此对于符号的判断基本都可以保证，因此大部分测量结果至少可以返回一个合理的 TDOA 符号，因此大致方向可以确定。并且在 offline 实验中，取 $m=1$ ，就可以满足我们的实验需求。

4.4 offline 定位方法

图 (4-7) 中，如果我们以 mic_1 为左，以 mic_2 为右，根据 TDOA 的符号 (0,-1,1) 实际可以确定对于声源与中心关系，这里我们只简单定义为左，前，右三个方向。后续说明方位的时候，我们所指方向就是这里说明的左，前，后。

由 TDOA 我们很容易确定一个双曲线方程，对于双曲线的渐近线方向，就是对声源的很好一个指示，至少在 2D 的范围中是很好的一個标准。这里我们根据 TDOA 值和麦克风距离，就可以计算出渐近线的角度。后续说明角度的时候，我们所指角度就是这里说明的角度。

在室内环境 (图 4-9) 下，我们需要在整个空间坐标系下确定每次设备的所处的位置，也就是说，已经确定了图 4-9 中 Position_i 的位置，随后我们在每个位置下录取一段音频，并进行 TDOA 的计算，由于设备在三维空间中可以有很多的位置，因此，我们可以得到一系列的 Position_i ，以及对应的 $X_i, Y_i, Z_i, \text{TDOA}_i$ 。我们利用已经比较成熟的 Chan 算法即可计算出后续的结果。

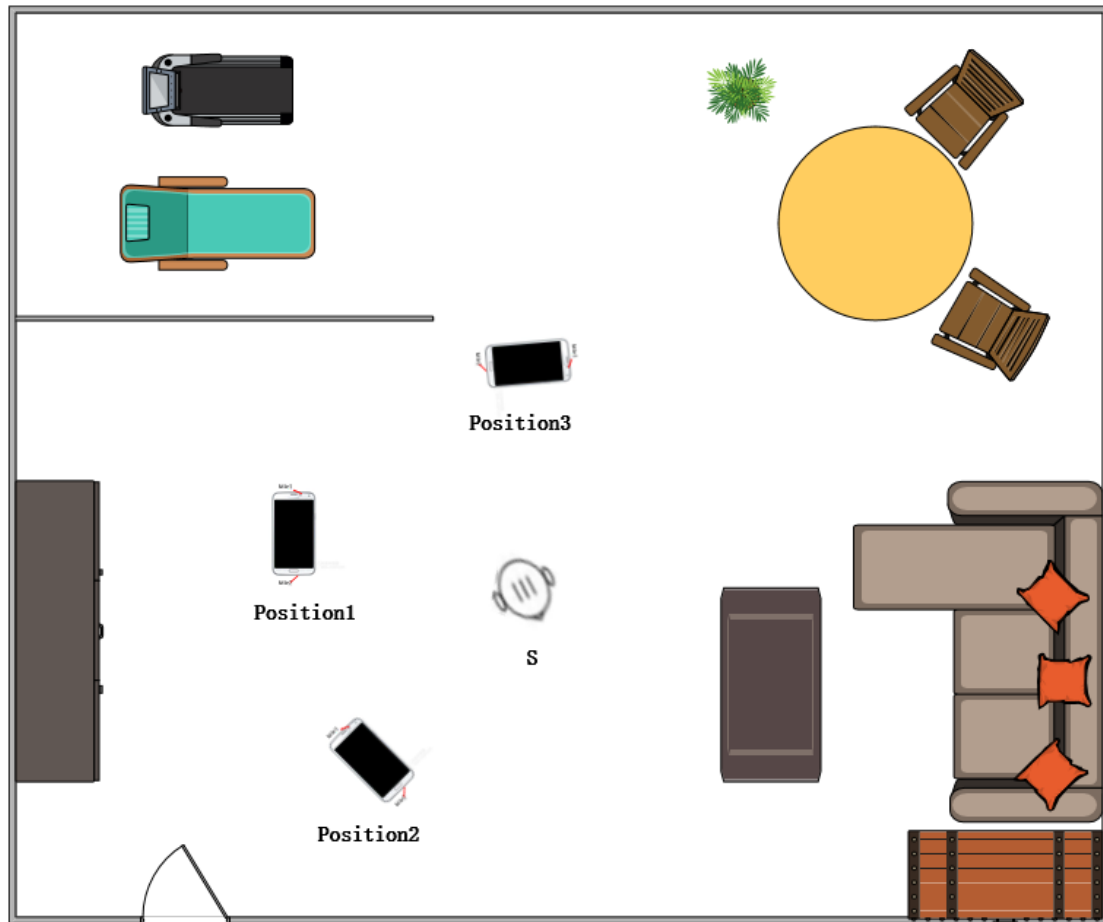


图 4-9: 室内环境

TODO Chan 算法简述精度数据下一章说明

4.5 offline 到 online 的迁移

4.5.1 概念阐述

offline 处理流程尽管可以获得更为精确的结果，但是在实际操作中过于复杂，并且在实际寻找的过程中，我们也并不是“视而不见”，简单的提示信息或许就可以帮助我们的进行探测，因此我们降低了实际需求，从精确到 3D 指引降至 2.5D 的探测范围。因此在 offline 算法的处理的基础上，我们提出了 online 的处理流程。

这里我们需要更新之前的处理流程，因为在 online 的处理流程中，麦克风处于一直的录音状态，因此有效声音提取部分也就不是那么重要，在前面的讲解中，我们指出无效声音段并不是声音信号的主体，因此可以直接抛弃一些数据。尽管 online 的处理流程也可以看做多个 offline 的组成，但是在实际操作

中，我们还是不希望在寻找过程中，在一个位置上长期停留，后续，我们引入一些机制进行一些状态的判断。

首先，我们需要提出一种降维的思想。如图4-10, 记智能手机屏幕所在平面为 P_1 , 声源实际位置 S 与两个麦克风 mic_1, mic_2 所在平面为 P_2 , 记 P_1 和 P_2 构成二面角为 α , 在 P_2 中, 声源位置 S 与智能手机中心的连线 l_1 与麦克风所在直线 l_2 夹角为 β , 声源在平面 P_1 的投影 S' 与智能手机中心的连线 l_3 与麦克风所在直线 l_2 夹角为 γ . 显然很容易得到 $\tan \gamma = \cos \alpha \cdot \tan \beta$. 图中 l_4 为平面 P_1 上与麦克风所在直线 l_2 夹角为 β 的直线。尽管 α 的值我们无法获知，但是计算出一个粗略的 β , 依然有一定的指导意义，因为在平面 P_1 上，声源投影的位置 S' 的方向角 $\gamma = \arctan(\cos \alpha \cdot \tan \beta)$ 。

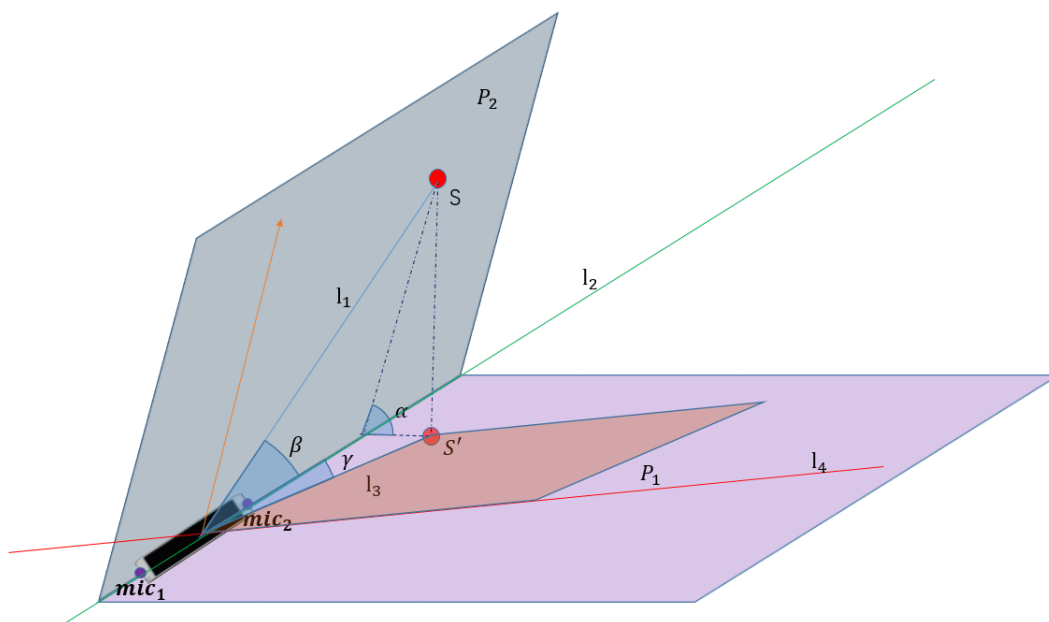


图 4-10: 3D 降维指引

如果在平面 P_1 上仍使用 β 作为指引的方向，那么，这里就会产生一个角度上偏离误差 $\delta\theta = \beta - \arctan(\cos \alpha \cdot \tan \beta)$ ，显然当 α 越大时，误差 $\delta\theta$ 越大 (这里 α 不超过 90 度)。显然在实际运行过程中，我们并不需要考虑这种情况。这是因为实际过程中，智能手机并非是固定不动的，因此，稍微的转动就可以使得二面角发生较大的变化。如图4-11所示，当手机位于 Loc_1 时，二面角 α 达到 90 度，但是只要稍微转动至 Loc_2 处，就可以获得一个新的 α ，甚至我们可以把手机移动到 Loc_3 处。当手机的转动已经不能造成二面角 α 较大的变化时，我们也就来到了声源投影位置的附近区域，这样，声源的位置我们也就大致获得了。

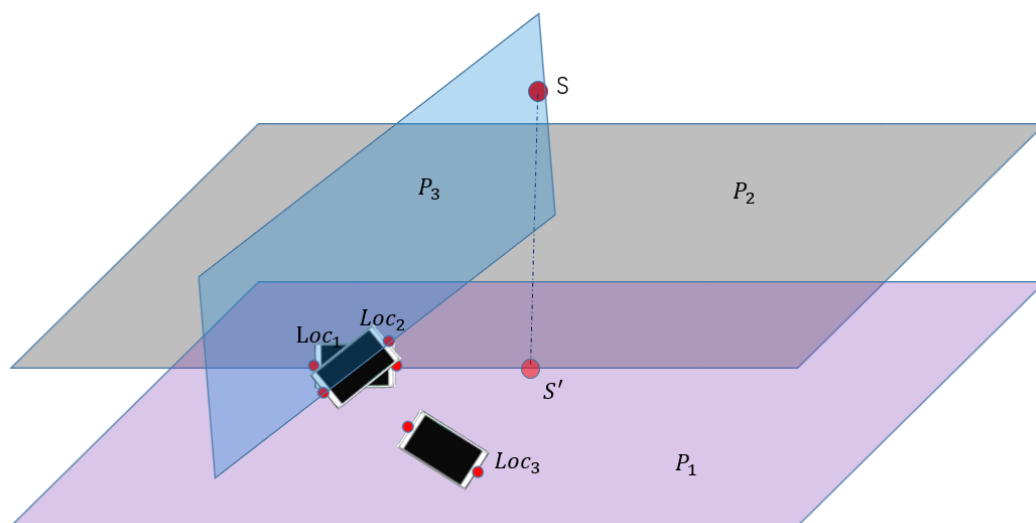


图 4-11: 手机状态改变

4.5.2 online 算法模型

第五章 模型系统实现与实验分析

5.1 开发环境与平台

5.2 系统设计

5.2.1 多模块设计

5.2.2 界面与运行方式

5.3 实验与分析

5.4 本章小结

第六章 总结与展望

6.1 工作总结

6.2 前景展望

参考文献

- [1] HONGZI Z, YUXIAO Z, ZIFAN L, et al. HyperEar: Indoor Remote Object Finding with a Single Phone[J]. IEEE 39th International Conference on Distributed Computing Systems(ICDCS), 2019.
- [2] Wikipedia contributors. Global Positioning System[EB/OL]. Wikipedia, The Free Encyclopedia, 2020 (2020/4/2) [2020/5/20].
https://en.wikipedia.org/wiki/Global_Positioning_System.
- [3] KE S, TING Z, WEI W, et al. VSkin: Sensing Touch Gestures on Surfaces of Mobile Devices Using Acoustic Signals[J]. ACM MobiCom, 2018.
- [4] AARABI P. The Fusion of Distributed Microphone Arrays for Sound Localization[J]. EURASIP Journal on Applied Signal Processing, 2003, 2003(4): 338–447.
- [5] Google Inc. Android Developers Sensors Overview[EB/OL]. 2020 [2020/5/20].
https://developer.android.com/guide/topics/sensors/sensors_overview.html.
- [6] Wikipedia contributors. Normalized frequency[EB/OL]. Wikipedia, The Free Encyclopedia, 2020 (2017/7/16) [2020/5/20].
[https://en.wikipedia.org/wiki/Normalized_frequency_\(unit\)](https://en.wikipedia.org/wiki/Normalized_frequency_(unit)).
- [7] Wikipedia contributors. Moving averagey[EB/OL]. Wikipedia, The Free Encyclopedia, 2020 (2020/2/19) [2020/5/20].
https://en.wikipedia.org/wiki/Moving_averag.

致 谢

非常感谢在 Dislab 实验室度过的两年时光，让我在实验室里拓宽眼界的同时，也不断纠正了自己的学习方法和学习态度。我的毕业论文撰写和系统实现中，也或多或少的遇到许多困难，也走过很多弯路。我很感谢我的导师谢磊副教授，从我毕设选题阶段，到我研究工作的开展，再到最后的论文撰写中，谢老师都给予了我细心的指导和点拨，给我纠正了很多错误的方向，避免我走了很多弯路。在实验室的两年，谢磊老师也给我的研究工作不断指引方向，在此，我向谢磊老师致以最真挚的谢意。

同样，我也要感谢在 Dislab 实验室的同一课题组的各位师兄师姐，鲁欣然师兄在我的成长道路上指引了很多，从细节方面给予我帮助，王楚豫师兄同样，给了我研究工作上的很多帮助，让我的研究工作变的富有创新，也是他们给我提出的宝贵建议，让我可以不断改进自己。

感谢大学四年来四种相伴的同学和朋友，因为你们的存在，我的大学才会在忙碌中透露出精彩。感谢每一位悉心教导我的老师，感谢钱柱中教授的指引，让我能够去香港游学增长知识和阅历，让我在问题求解的课程中感受到计算机和算法的魅力。