华东理工大学研究生院

《音频信号处理》课程报告

开课学院： 信息科学与工程

专 业： 信号与信息处理

姓 名： 何林飞

学 号： Y30160609

任课教师： 陈宁

2017年 4月13

音频指纹技术研究

摘要 听歌识曲，版权保护，多屏互动等多个领域都有音频指纹技术的应用，基于landmark的音频指纹技术对于普通场景的识别已经非常成熟，而对于复杂场景的音频识别没有完善的解决方案。算法的鲁棒性和检索的快速性是指纹技术发展的方向，本文将概述性的介绍音频指纹最新的发展，并详细介绍传统经典landmark算法以及近几年的新型算法，并做简单的程序仿真。

关键词：音频指纹，Landmark，图像处理，指纹检索

**目录**

[1 绪论 1](#_Toc480294698)

[1.1音频指纹技术概述 1](#_Toc480294699)

[1.2 国内外发展现状 2](#_Toc480294700)

[1.3 内容简介 3](#_Toc480294701)

[2 音频指纹算法 4](#_Toc480294702)

[2.1 shazam音频指纹算法 4](#_Toc480294703)

[2.2 Shazam音频指纹算法的改进算法 6](#_Toc480294704)

[2.3 一种基于图像处理的音频指纹算法 7](#_Toc480294705)

[2.3.1 基于图像的音频指纹 7](#_Toc480294706)

[2.3.2检测匹配 10](#_Toc480294707)

[3 基于原始Shazam算法的程序仿真 13](#_Toc480294708)

[4 总结与展望 14](#_Toc480294709)

[参考文献 15](#_Toc480294710)

1 绪论

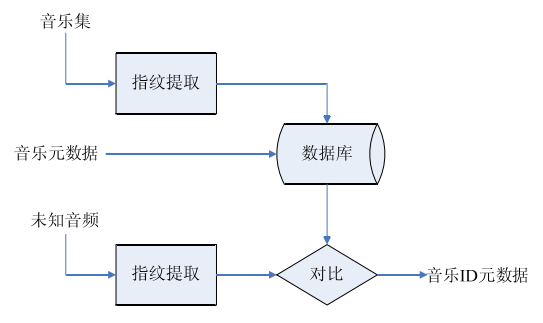
# 1.1音频指纹技术概述

音频压缩技术的进步以及大容量存储器的出现使得互联网上出现了以音乐为主的海量音信息，手工选取某首歌曲很多时候已经变得不可能，这直接促使产生了可以进行音乐自动识别的数字音频指纹技术。

音频指纹是指可以代表一段音乐重要声学特征的基于内容的数字签名，其主要目的是建立一种有效机制来比较两个音频数据的感知听觉质量。注意这里不是直接比较通常很大的音频数据本身，而是比较其相应通常较小的数字指纹。大量音频数据的指纹和其相应的元数据比如歌曲名称、词曲作者、歌词等内容一起存储在一个数据库中，并采用指纹作为相应元数据的索引[1]。

一个音频指纹系统通常包括两个部分:即一个计算听觉重要特征的指纹提取算法和一个在指纹数据库中进行有效搜索的比对算法。当要识别一段未知音频时，首先按照指纹提取算法计算其音频特征，然后和数据库中存储的大量音频指纹相比对从而进行识别。一个有效的指纹提取算法和指纹比对算法能够在数据库中正确识别出可能经受各种信号处理失真的未知音频的原始版本。若识别到对应的原始指纹，则可提取出相应的元数据信息返回给用户[2]。

整个过程模仿人耳识别音乐的过程。如图1-1所示，许多音频作品的指纹在离线情况下计算出来，连同一些重要的元数据如歌曲名称、演唱者、词曲作者、歌词等一并存储到数据库中，在识别过程中未知音频按照同样的算法计算出指纹，然后与数据库中存储的指纹进行比对，如果查找到就返回查询者感兴趣的元数据信息。



**图1-1 基于指纹的音乐检索算法流程**

大多数指纹提取算法基于以下方法[2]：首先将音频信号分成互相重叠的帧，对每一帧计算一系列特征，这些特征需要对各种音频信号处理至少在一定程度上保持不变。已经提出的特征有傅立叶系数FFT、迈尔倒谱系数Mel Frequency Cepstral Coefficients (MFCC)、频谱平滑度Spectral Flatness、尖锐度Sharpness、线性预测编码系数Linear Pre-dictive Coding (LPC)等，还有这些基本量的导出量如均值和方差。通常这些特征使用分类器技术映射到一个更简洁的表示，如隐含马尔可夫模型Hidden Markov Models(HMM)或量化技术。每帧算出的指纹叫做子指纹(sub-fingerprint)，一个子指纹通常并不能包含足够的信息来进行音频识别，足以识别完整音频的未知音频单元叫做指纹块(fingerprint block)，即指纹粒度。

# 1.2 国内外发展现状

目前，对于音频指纹的分类还没有形成统一的共识。如文献[2]将音频指纹分为语义特征和非语义特征两类，而根据音频指纹所提取的特征属将音频指纹分为时间域音频指纹算法、频域音频指纹算法、时频域音频指纹算法三类。

音频指纹检索系统可以分为两部分，一是特征提取，二是特征的存储与检索。系统首先提取数据集中音频的特征，并存储在一个数据库中，当接收到一段未知音频片段的查询请求时，使用同样的算法提取特征，并在数据库中的进行特征的检索和匹配。

在音频特征提取方面，文献[3]提出一种基于短时平均能量，短时平均幅度，短时过零率的一种基于互相关的音频检测方法，这种方法较为简单，计算速度快，但是会造成较多误识。

美国的 Muscle Fish 公司在文献[4]中先将带标识的数据加窗处理，对每帧数据提取音调、响度、亮度、带宽属性，对属性序列计算其均值、方差和自相关值，加上能量共 13 个特征，则此 13 征即为音频数据的特征矢量，检索时采用马氏距离，比较样本特征矢量与库据的特征矢量，从而输出检索结果。

Hastima 等人介绍了 Philips 公司的基于音频指纹的音频检索系统[5]。音频指纹是音频内容关键特征的一种压缩、紧凑的表示，音频指纹特征提取函数将数据冗余较大的音频波形空间映射到指纹空间。音频指纹应具有鲁棒性，即使存在由于压缩算法、传输信道的差异等影响造成的信号畸变，也可以用其在海量未知音频数据中检索音频片段。通过对用户提供的音频片段进行指纹提取，利用提取的指纹采用鲁棒匹配算法实现对音频的自动识别，最后将搜索结果以某种方式发送给用户。在音频特征存储方面，Hastima 等人介绍了 Philips 公司的倒排表存储结构[5]，并被广泛应用。

文献[11]提出节拍跟踪和色度谱特征可用于标记含节奏和乐器的变化的音频，并可以用相关性进行检测。文献[12]提出基于色度向量的和弦序列提取，并用DP算法校正。文献[13]提出将色度向量拼接成高维向量并用最小邻接搜索。文献[14]提出一种改进的色度特征。

Shazam公司在2003发表的论文中提出了landmark音频指纹算法[6]，该算法利用提取音频语谱图的局部峰值点，利用有关联的两个峰值点计算hash值作为音频指纹，实验结果显示该算法计算和检索都非常快速，对噪声有很好的鲁棒性。但对于一些劣化的音频如time-stretching和pitch-shifting，以及翻唱歌曲等都无法识别。T. Bertin-Mahieux和D. P. W. Ellis在2011年的论文[22]中提出了基于色度谱的landmark算法，能有效识别翻唱歌曲。J. George 和 A. Jhunjhunwala在2015年的论文中提出另一种改进的landmark算法，该算法提出了一种新的峰值点提取方法，实验结果显示该算法能有效识别含time-stretching问题的音频。

图像处理技术在音频指纹中也有应用。文献[24]使用SIFT提取音频语谱图的特征点作为音频指纹，具有抗音频劣化的效果。文献[23]提出一种基于梯度直方图的改进landmark算法，使得移动设备在噪声环境下也能很好的识别音频。文献[8]使用Hough变换检测二值化语谱图的海明相似性矩阵，该算法在现场录制等复杂场景下有较好的识别效果。由于图像处理计算量大，文献[25]提出了一种使用K-modes 和LSH进行检索，并用多GPU并行加速计算，实验结果显示该方法的准确率和检索速度超过现有商业音频识别系统。

# 1.3 内容简介

本文详细介绍shazam的音频指纹及其整个检索系统[6]，然后介绍针对其指纹算法中的time-stretching问题的改进算法[7]，之后还会介绍一种基于图像处理的音频指纹算法[8]，该方法改进了pitch-shifting问题，对现场录制音频有较好的识别率。最后，对于原始shazam音频识别系统进行简单的程序仿真。

2 音频指纹算法

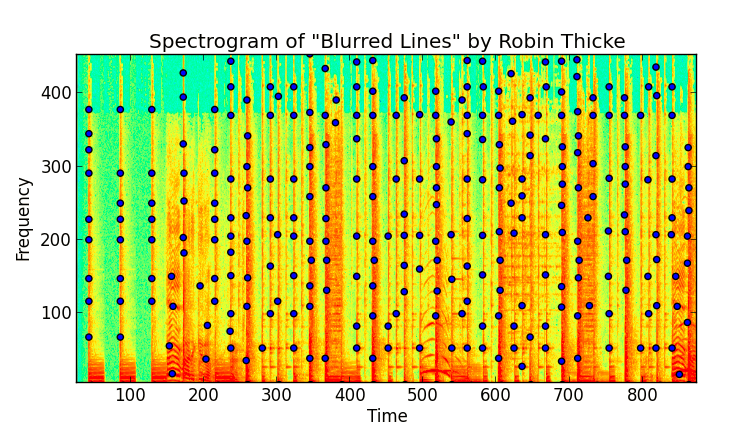
# 2.1 shazam音频指纹算法

Shazam是一家专注于音乐识别的公司，旗下音乐识别软件也叫做Shazam，Avery在2003年公开了Shazam的音频指纹提取算法[6]。

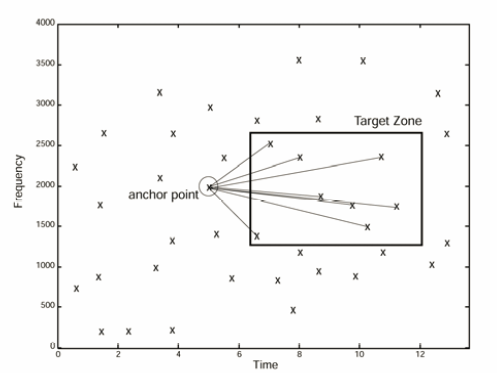
在经过傅里叶变换之后生成的频谱图中，在一个区域内，如果一个点的能量大于其周围的所有点，则将其选为候选点。候选点的选择要遵循一定的密度规则，以保证声音文件的时间频率带的合理分布。在每个时间频率带中的极值点也会根据振幅大小进行过滤，因为振幅较大的点最容易在各种信号畸变中存活下来。

图2-1是歌曲Blurred Lines的音频经过短时傅里叶变换（STFT）加海明窗得到的语谱图[9]，并检测标记了峰值点。峰值点的选取满足一定的条件：幅度大于最小阈值A，与邻近的峰值点有一定的距离D，这两个都是人为可调参数。

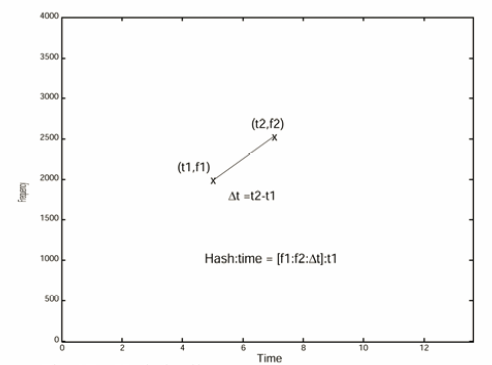
图2-2和图2-3展示了音频指纹的生成过程[6]。在图2-1生成的峰值点中选择一个作为锚点，之后并在该点之后选择一定区域作为目标区域。锚点一般依次迭代所有峰值点，目标区域可以选择锚点之后的10~20个点，这些都是人为可调参数。目标区域中的每个点都和锚点形成一对关联对，如图2-3所示。一对关联对中的两个峰值点的各自的频率以及两点的时间差这三个量生成32位的hash值。例如可以设定字符串“”，用于生成hash。指纹存储时建议把锚点的时间一同记录，在检索音频片段时可以知道该片段在原音频中的时间。



**图2-1 包含峰值点的语谱图**



**图2-2 锚点和目标区域**



**图2-3 hash值生成过程**

在创建一个音频指纹数据库时，对每一首歌曲执行上述步骤，生成对应hash值列表。歌曲的ID可以被附加到后面，与hash值一起存储。

针对上述音频指纹特征的匹配算法如下所示[10]：

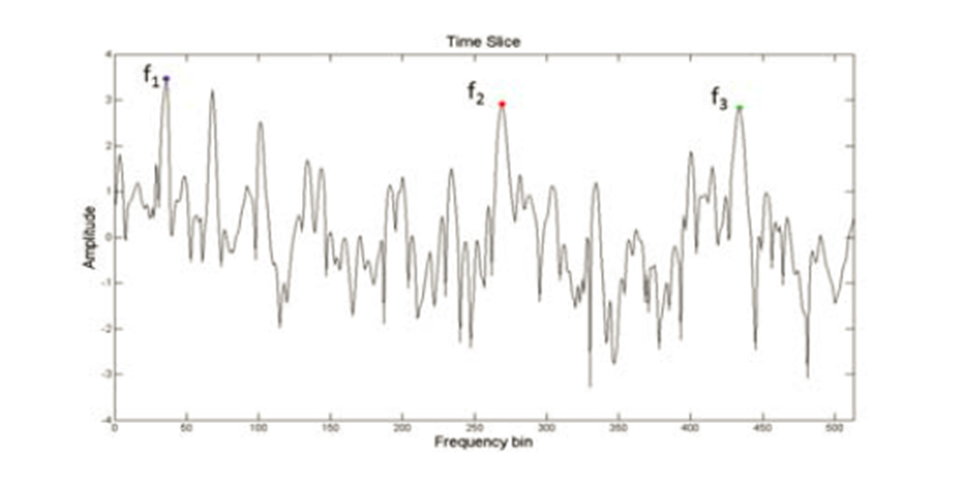
1. 首先对查询片段执行特征提取过程，提取一系列 hash 值。
2. 对每一个hash值，在数据库中进行查找，并取出对应的歌曲ID和时间信息。
3. 将所有取出的结果按照歌曲ID划分成多个组。
4. 对每一组里的值，按照时间排序，并计算时间相对连续的匹配点的个数。
5. 取匹配点最多的歌曲ID作为结果。

# 2.2 Shazam音频指纹算法的改进算法

Shazam算法计算成本很低，对噪声有很好的鲁棒性，适合普通环境下的音频检测。由于该算法指纹中包含时间差信息，所以无法检测含time-stretching畸变的音频。而演唱会现场录制的音频一般会有pitch-shifting，time-stretching，噪声等问题，所以识别效果不好。J. George and A. Jhunjhunwala 2015年发表的论文中提出了基于Shazam指纹算法的改进算法，对time-stretching问题有明显改善[7]。

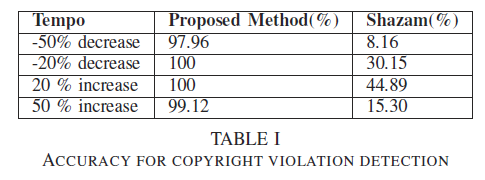
该论文的音频指纹识别过程如下：

1. 对音频进行短时傅里叶STFT变换，该过程和Shazam算法相同
2. 对语谱图每一帧进行一次高通滤波，该操作过滤了该帧中的低频扰动成分，使得峰值点更加显著
3. 每一帧取三个峰值点，每个点最小距离d个频率帧，如图2-4所示[7]。d的选取根据频率成分而确定。根据经验，对于1024点的FFT，8KHZ采样的音频数据中，d可以选40
4. 利用这三个点的频率，用和Shazam同样的方法生成30位的hash值
5. 检索方法和Shazam相同



**图2-4 每隔最小40频率帧选择三个峰值点**

**Table1**显示这种方法对time-stretching有很好的鲁棒性[7]。



# 2.3 一种基于图像处理的音频指纹算法

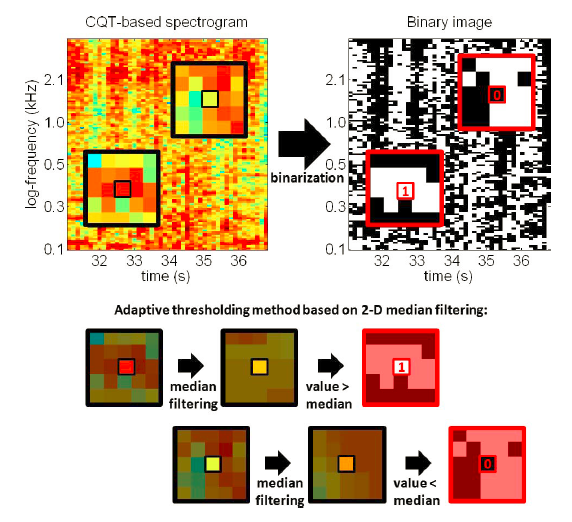
基于Shazam的改进算法虽然改进了time-stretching问题，但是对于现场录制中的pitch-shifting问题并没有改善，因为它时基于峰值点的频率信息生成指纹的，频率改变的话算法就失效了。

Z. Rafii, B. Coover and J. Han在2014年提出了一种对音频语谱图进行图像处理生成指纹和检索的方法，该方法对于现场录制的音频识别有较好的鲁棒性[8]。但该方法的计算量较大。

## 2.3.1 基于图像的音频指纹

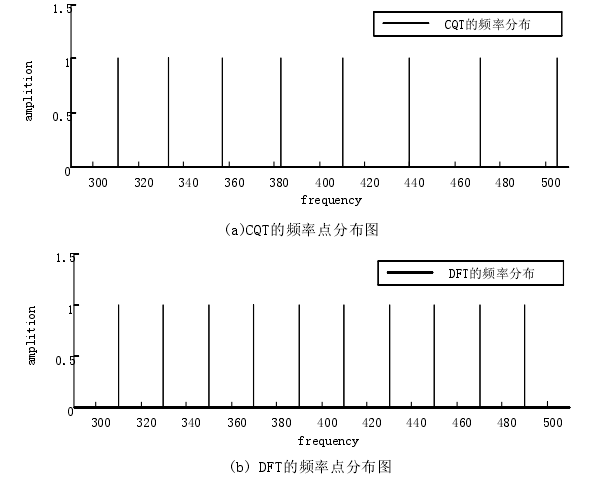
指纹提取过程如图2-5，基本步骤如下：

1. 将音频信号进行恒定Q值变换（CQT）[15]得到带log的语谱图。CQT使用快速算法得到[17]，其中时间分辨率为0.13s每帧，频率范围从C3 (130.81Hz) 到C8 (4186.01 Hz)，分辨率为1/4音调每通道
2. 通过一个自适应滤波器，对语谱图进行图像处理，使其二值化。滤波器模板大小为35x15，以模板范围内的像素中值为阈值，高于该值变为1，其他变为0



**图2-5 指纹生成过程**

1990年，Brown 提出了Constant Q变换(CQT)，将它作为一种计算对数频率分辨率频谱图的一种方法[15]。Constant Q变换是将信号的时域转换到频率的一种方法，基于这个特征Constant Q变换与离散傅里叶变换(DFT)相同。但是离散傅里叶变换（DFT）的频率点是线性分布的，是等间隔的，而Constant Q变换的频率点是按指数规律分布的，如图2-6所示。



**图2-6 CQT，DFT的频率分布图**

从滤波器的角度分析，DFT是中心频率等间隔分布，带宽相同的滤波器组，CQT是中心频率与带宽的比为定值Q的滤波器组对音乐信号来说，它的音阶频率是按指数规律分布的，这一点与CQT相同。所以在音乐的信号处理方面，CQT有很大的优点。CQT的时频域窗口是可调整的，一般对快变信号，它有较高的时域分辨率，较低的频域分辨率；对于慢变信号，有较低的时间分辨率，较高的频率分辨率。变换后信号的频率点成指数分布，更符合音乐特征的本质。西方音乐的基音频率是对数分布的，离散傅里叶变换虽然能够用FFT快速算法，但是计算出来的频率是成线性分布的，与音乐频率不对应，但是CQT的频率分辨率与西方音乐中常用到的十二平均率相匹配，CQT是从音乐和听觉感性两方面出发的[16]。

在CQT中，若给出初始最低频率，可以得到各频带中心频率，CQT中心频率与带宽比值是一个恒定的值，在不同时间有不同的频率分辨率，DFT中有固定的频率间距和固定的频率分辨率。

如果用半音C3的频率（130.813Hz）为基础频率，对于DFT，在谱线k处的数字频率为：

其中N为采样点个数，k为频率指数。

所对应的模拟频率（中心频率）为：

相邻谱线的频率间隔（带宽）为：

对于CQT，第k个半音的频率（中心频率）为：

其中b为一个八度内划分的音符的个数，一般情况下，b为12或者24。

相邻半音的频率间隔（带宽）为

并非为常数，所以一个八度内的频率分布是非均匀的，是指数分布。中心频率和带宽的比是一个常数，即

我们定义这个常数为Q，其第k个半音的频率分辨率为。

其第k个半音的频率幅度为：

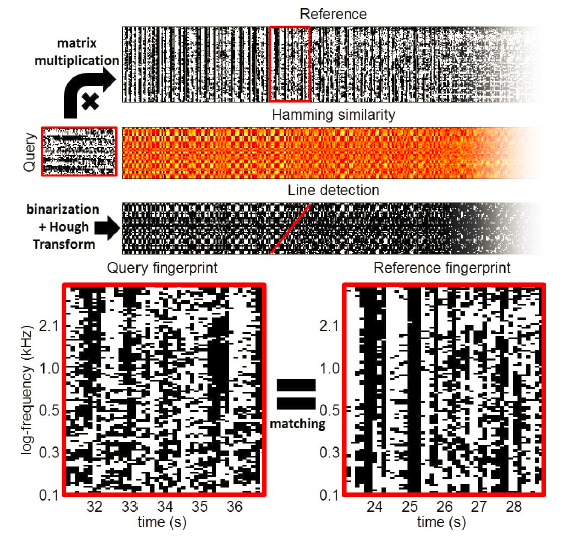
信号作时频分析时，一般对快变信号，希望它有较高的时间分辨率以观察快变信号。而其快变信号对应的是高频信号，所以我们就需要较高的时域分辨率，较低的频域分辨率；对于慢变信号，我们就需要较低的时间分辨率，较高的频率分辨率；这就体现了傅里叶变换中在时间和频率分辨率方面固有的矛盾。CQT

解决了这一矛盾，体现了其优越性。

## 2.3.2检测匹配

检测的过程如图2-7所示，基本过程如下：

1. 用海明相似性（Hamming Similarity）[18]计算测试样例和已知样例的相似性，得到相似矩阵
2. 设定一个阈值（文献中为0.6）将相似矩阵二值化，并以图像的形式表达。利用Hough变换[19]检测二值化后图像中呈45°的直线，如果存在则匹配成功。



**图2-7 音频检测过程**

海明相似性检测过程如下：

1. 将待检测样例所生成的指纹转置矩阵Q和参考样例R中的每一个元素都经过线性变换，使其值域变为为-1和1
2. 计算Q和R的矩阵乘积，之后对每一个元素都计算
3. 最后再除以频率帧数（即Q的列数）进行归一化得到海明相似矩阵

Hough变换是图像处理中从图像中识别几何形状的基本方法之一。Hough变换的基本原理在于利用点与线的对偶性，将原始图像空间的给定的曲线通过曲线表达形式变为参数空间的一个点。这样就把原始图像中给定曲线的检测问题转化为寻找参数空间中的峰值问题。也即把检测整体特性转化为检测局部特性。比如直线、椭圆、圆、弧线等。

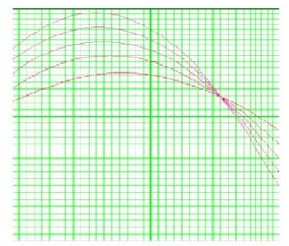
对于直线检测，Hough变换的基本原理如下[20]：

设已知一黑白图像上画了一条直线，要求出这条直线所在的位置。我们知道，直线的方程可以用来表示，其中*k*和*b*是参数，分别是斜率和截距。过某一点(,)的所有直线的参数都会满足方程。即点(,)确定了一族直线。方程在参数*k--b*平面上是一条直线，(你也可以是方程对应的直线)。这样，图像*x--y*平面上的一个前景像素点就对应到参数平面上的一条直线。我们举个例子说明解决前面那个问题的原理。设图像上的直线是*y=x*, 我们先取上面的三个点：A(0,0)，B(1,1)，C(2,2)。可以求出，过A点的直线的参数要满足方程*b=*0, 过B点的直线的参数要满足方程1=*k+b*, 过C点的直线的参数要满足方程2=2*k+b*, 这三个方程就对应着参数平面上的三条直线，而这三条直线会相交于一点(*k*=1,*b*=0)。　同理，原图像上直线*y=x*上的其它点(如(3,3),(4,4)等)　对应参数平面上的直线也会通过点(*k*=1,*b*=0)。这个性质就为我们解决问题提供了方法，就是把图像平面上的点对应到参数平面上的线，最后通过统计特性来解决问题。假如图像平面上有两条直线，那么最终在参数平面上就会看到两个峰值点，依此类推。

简而言之，Hough变换思想为：在原始图像坐标系下的一个点对应了参数坐标系中的一条直线，同样参数坐标系的一条直线对应了原始坐标系下的一个点，然后，原始坐标系下呈现直线的所有点，它们的斜率和截距是相同的，所以它们在参数坐标系下对应于同一个点。这样在将原始坐标系下的各个点投影到参数坐标系下之后，看参数坐标系下有没有聚集点，这样的聚集点就对应了原始坐标系下的直线。

在实际应用中，形式的直线方程没有办法表示*x=c*形式的直线(这时候，直线的斜率为无穷大)。所以实际应用中，是采用参数方程。这样，图像平面上的一个点就对应到参数--平面上的一条曲线上，其它的还是一样。同理，极坐标中的一个点对应x--y坐标中的一条直线。

为了检测出直角坐标*x--y*中由点所构成的直线，可以将极坐标--量化成许多小格。根据直角坐标中每个点的坐标(*x, y*)，在 = 0~180°内以小格的步长计算各个值，所得值落在某个小格内，便使该小格的累加记数器加1。当直角坐标中全部的点都变换后，对小格进行检验，计数值最大的小格，其(, )值对应于直角坐标中所求直线。如图2-8所示是直线上五个点在极坐标下所对应的曲线，根据相交点(, )可求出直线。图中曲线所经过的小格的累加计数器加1，明显可得在点(, )处有最大的计数值。



**图2-8 直线y=x+20上五个点在极坐标中对应的正弦曲线**

文献[8]提出通过以上方法找到的(, )必须满足小于参考样例的时间帧总数（即R的列数）。的范围在，这个数值是针对最大的时间偏移和的节奏变化所确定的经验阈值。该方法对现场录制的音频有较好的识别效果。

3 基于原始Shazam算法的程序仿真

该程序仿真是根据原始Shazam音频识别系统框架编写，参考源码[21]。实验平台：win10，64位，I5-3210M [CPU@2.5GHZ，4G](mailto:CPU@2.5GHZ，4G)内存。开发工具Pycharm，Python-2.7（Pydub，Numpy，Scipy，Pyqt5，Matplotlib），ffmpeg-3.2.4，Mysql-5.5.52。

程序功能：UI交互，能输入音频文件并使用ffplay播放，没有格式限制。能添加文件进数据库，或者对选择的音频文件进行识别，如图3-1所示。



**图3-1 简单音频识别系统**

程序模块划分为五个模块：

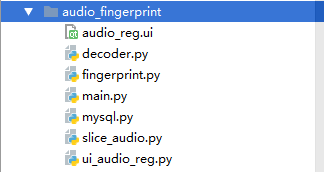
decoder.py：使用pydub和ffmpeg对音频进行解码，得到音频数据和频率

fingerprint.py：输入音频数据，用原始Shazam音频指纹算法得到hash值

mysql.py：将生成的hash值存储到mysql数据库，含添加，检索功能

ui\_audio\_reg.py：UI界面

main.py：程序主入口，启动UI交互，多线程防止阻塞



实验条件有限，只测试原始音频的局部片段，对于20首歌有100%的检出率。

4 总结与展望

普通环境或者本地的音频检测已经有非常完善的商业应用，而复杂场景的音频检测是近几年的研究热点，比如演唱会现场录制的音频含大量噪声以及频谱劣化的问题。算法的过于复杂会对降低音频检索效率，所以设计一个高鲁棒性和快速检索的音频识别系统是今后音频指纹研究的主要方向。目前绝大多数的音频指纹技术都会在变换域中进行操作，基于landmark的音频指纹对复杂场景有一定的局限性，但计算量小，符合商业应用场景，属于传统经典算法；而基于图像处理的方法能更好的应对复杂场景的识别，计算量大的问题可以通过GPU加速改善，是目前效果最好的算法。随着深度学习的快速发展，相信更多精致的算法将会运用到音频指纹技术当中。

参考文献

1. P. Cano, E. Batle, T. Kalker and J. Haitsma, "A review of algorithms for audio fingerprinting," 2002 IEEE Workshop on Multimedia Signal Processing., 2002, pp. 169-173.
2. 李伟,李晓强,陈芳,王淞昕. 数字音频指纹技术综述[J]. 小型微型计算机系统,2008,(11):2124-2130.
3. 李超,熊璋,朱成军. 基于距离相关图的音频相似性度量方法[J]. 北京航空航天大学学报,2006,(02):224-227.
4. E. Wold, T. Blum, D. Keislar and J. Wheaten, "Content-based classification, search, and retrieval of audio," in IEEE MultiMedia, vol. 3, no. 3, pp. 27-36, Fall 1996.
5. Haitsma J, Kalker T. A Highly Robust Audio Fingerprinting System[C]// Ismir 2002, International Conference on Music Information Retrieval, Paris, France, October 13-17, 2002, Proceedings. DBLP, 2002:107--115.
6. Wang A. An Industrial Strength Audio Search Algorithm[C]// Ismir 2003, International Conference on Music Information Retrieval, Baltimore, Maryland, Usa, October 27-30, 2003, Proceedings. DBLP, 2003.
7. J. George and A. Jhunjhunwala, "Scalable and robust audio fingerprinting method tolerable to time-stretching," 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, 2015, pp. 436-440.
8. Z. Rafii, B. Coover and J. Han, "An audio fingerprinting system for live version identification using image processing techniques," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014, pp. 644-648.
9. WILLDREVO, Audio Fingerprinting with Python and Numpy,  
   <http://willdrevo.com/fingerprinting-and-audio-recognition-with-python/>,
10. 王润涛. 海量音频指纹数据的存储与检索研究[D].天津大学,2014.
11. Daniel P.W. Ellis and Graham E. Poliner, “Identifying ’cover songs’ with chroma features and dynamic programming beattracking,” in 32nd International Conference on Acoustics,Speech and Signal Processing, Honolulu, HI, USA, April 15-20 2007, vol. 4, pp. 1429–1432.
12. Juan Pablo Bello, “Audio-based cover song retrieval using approximate chord sequences testing shifts, gaps, swaps and beat,” in 8th International Conference on Music Information Retrieval, Vienna, Austria, September 23-27 2007, pp. 239–244.
13. Michael A. Casey, Christophe Rhodes, and Malcolm Slaney,“Analysis of minimum distances in high-dimensional musical spaces,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 5, pp. 1015–1028, July 2008.
14. Joan Serr`a, Emilia G´omez, Perfecto Herrera, and Xavier Serra,“Chroma binary similarity and local alignment applied to cover song identification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 6, pp. 1138–1152, August 2008.
15. Judith C. Brown, “Calculation of a constant Q spectral transform,” Journal of the Acoustical Society of America, vol. 89, no. 1, pp. 425–434, January 1991.
16. 韩雅欢. 基于Constant Q变换的音符起始点检测算法研究[D]. 南京邮电大学, 2013.
17. Judith C. Brown and Miller S. Puckette, “An efficient algorithm for the calculation of a constant Q transform,” Journal of the Acoustical Society of America, vol. 92, no. 5, pp. 2698–2701, November 1992.
18. Moses S. Charikar, “Similarity estimation techniques from rounding algorithms,” in 34th ACM Symposium on Theory of Computing, Montr´eal, Qu´ebec, Canada, May 19-21 2002, pp.380–388.
19. Richard O. Duda and Peter E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” Communications of the ACM, vol. 15, no. 1, pp. 11–15, January 1972.
20. AndyJee, Hough transform(霍夫变换), <http://www.cnblogs.com/AndyJee/p/3805594.html>
21. sa-matiny, Audio-Fingerprinting, <https://github.com/sa-matiny/Audio-Fingerprinting>
22. T. Bertin-Mahieux and D. P. W. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, 2011, pp. 117-120.
23. T. Park, S. Beack and T. Lee, "A noise robust audio fingerprint extraction technique for mobile devices using gradient histograms," 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, 2015, pp. 287-290.
24. Xiu Z, Zhu B, Li L, et al. SIFT-based local spectrogram image descriptor: a novel feature for robust music identification[J]. EURASIP Journal on Audio, Speech, and Music Processing, 2015, 2015(1):6.
25. C. Ouali, P. Dumouchel and V. Gupta, "Fast Audio Fingerprinting System Using GPU and a Clustering-Based Technique," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 6, pp. 1106-1118, June 2016.