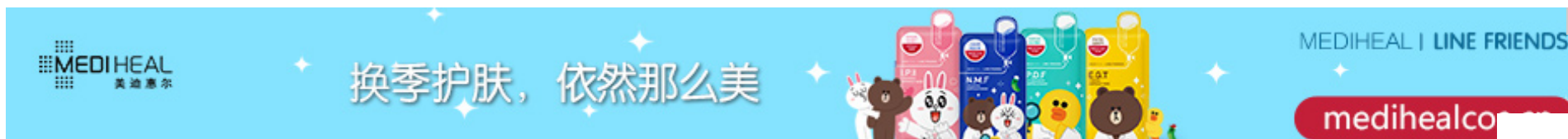




请输入您要搜索的内容

[首页](#)[语录](#)[生活](#)[搞笑](#)[游戏](#)[萌宠](#)[明星](#)[趣玩](#)[红人](#)[美女](#)[时尚](#)[健康](#)[旅游](#)当前位置：[首页](#) [微信公众平台推荐](#) [IT技术微信公众号](#) [Python程序员](#) [正文](#)

Python和Numpy 处理音频指纹 —

2016-11-04 [Python程序员](#)

肥城恒源环保设备生...

高分子微曝气管服务完
善,质量可靠,值得信赖..

搬家上58同城先行赔...

提供各类车型价格统一
专业单位/住宅搬家,2...

中电昆辰核

空间定位精度可达1厘
米,一维测距可达1毫...

分享到

微信版



微信扫一扫关注！

喜欢就关注我 有惊喜哦！

Python部落

(python.freelycode.com)组织翻译，禁止转载，欢迎转发。

第一天试用Shazam, 我大吃一惊。除了手机GPS和从一段楼梯掉下后还能使用，我见过手机做过的最难以置信的事情就是能够从一个巨大的音频语料库中识别出一首歌。这个识别工作原理叫做音频指纹。例子有：

Shazam

微信公众号：[Python程序员](#)微信账号ID：[pythonbuluo](#)

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)[Chromaprint](#)[Echoprint](#)

几周后，通过学术论文和编写代码的研究，我做了一个Python的音频指纹的开源项目Dejavu。你可以在GitHub上查看：

<https://github.com/worldveil/dejavu>

在我的测试数据集中，Dejavu能够100%识别从硬盘读取的一个未知波形文件或监听的一个至少5秒的录音。

以下是所有你需要了解的关于音频指纹从基础到识别的知识。那些有信号经验的人可以直接跳到“寻峰”。

音乐就是信号

作为一名计算机科学家，我熟悉的快速傅里叶变换(FFT)只是一个很酷的解决多元多项式乘法的方法，时间复杂度为 $O(n\log(n))$ 。幸运的是它的典型应用——信号处理——比这酷多了。

音乐，事实证明，只是经过数字编码后的一长串数字。在一个未压缩的 .wav 文件中有很多这样的数字——44100每秒每声道。这意味着一首3分钟长的歌几乎有1600万个样本。

```
3 min * 60 sec * 44100 samples per sec * 2 channels = 15,876,000 samples
```

一个声道是一个扬声器

可以播放的单独样本序列。比如有两个耳塞——这就是一个“立体声”或者说双声道设置。单一的声道被称为“单声道”。今天，现代的环境声系统可以支持更多的声道。但是，除非声音是在相同数量的声道中录音或混音，否则额外的扬声器就是多余的，这些扬声器将只会播放相同的样本。

采样

为什么每秒取样44100次？每秒取样44100次的神秘选择似乎很随意，但它涉及到奈奎斯特-香农采样原理。这是一个很长的数学方法，说是当我们录音时，为了我们可以准确地取样有一个理论上的最大频率的限。这个最大的频率是基于我们采样的速度。

如果这很难理解，想想观察一个风扇叶片，旋转一周刚好每秒1次（1赫兹）。现在想象一下，闭上眼睛，但是每秒睁开一次。如果风扇仍然恰好是每秒旋转一周，就会发现风扇叶片像是没有移动！每次你睁开眼睛，叶片都会在同一个地方。但是有一个问题。事实上，如你所知，风扇叶片可以每秒旋转0, 1, 2, 3, 10, 100, 甚至100万次，而且你永远不会知道-它仍然会看起来像是静止的！因此，为了保证你能正确的采样



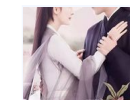
热门文章 最新收录



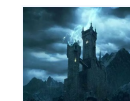
这部同志片，拍出了99%的人的青春



《三生三世十里桃花》大概是史上CP最多的剧了吧！



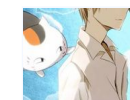
《三生三世十里桃花》经典台词: 用我无端宠溺, 免你半



全球十大魔幻电影排行榜，简直炫酷！



如果生活是跟童话故事一样的话，我就会有特美好的



《夏目友人帐》：你的温柔令人心痛，与你一起的光阴

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)

率的速度去采样，以确保我们能够检测到。

录音时，通用原则是，我们只需要记录22050Hz以下的声音，因为人类正常听力甚至不能听到超过20000Hz的声音。因此根据奈奎斯特-香农采样原理，采样频率的两倍：

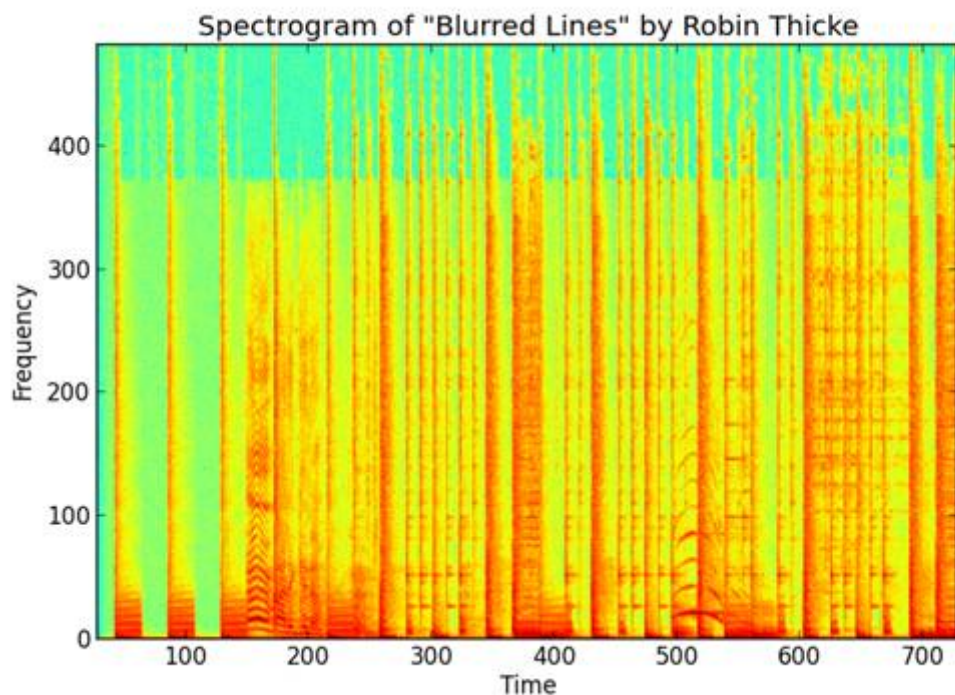
```
Samples per sec needed = Highest-Frequency * 2 = 22050 * 2 = 44100
```

MP3压缩格式的出现

是为了1) 节省硬盘空间，和2) 刺激音乐发烧友，但你电脑上的纯 .wav格式文件只是一个16bits的整形列表（和一个小标题）。

频谱图

由于这些样本是多种信号，我们可以重复对小窗时间的歌曲样本使用FFT来创建一个歌曲的频谱图。下面是Robin Thicke 的 "Blurred Lines" 前几秒的频谱图。



如你所见，它只是

一个以时间和频率为维度的表现振幅的二维数组。FFT把特定频率信号的强度（振幅）显示为一列。如果我们



刘若英深情演唱《原来



女大学生竟被食堂员工



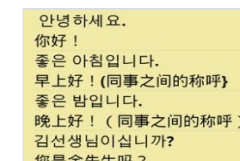
丈夫发现妻子在宾馆， 拜金女第一次相亲就上



怎么除眼袋



眼部去皱的方法



韩语口语



python课程

微信推荐文章



三儿一女这样给老母亲过生日，看哭了！必须火！



在西峡做过以下10件叫少女，20件叫熟女，30件就

张大拙新春大吉微信特卖展

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)

重要的是要注意到频率和时间值是离散的，每个代表一个“bin”，而振幅是实值。颜色显示在离散坐标（时间，频率）振幅的实值（红色->更高，绿色->更低）。

作为一个思想实验，如果我们记录和创建一个单音节的频谱，我们会得到一个水平直线。这是因为频率不随窗口到窗口变化。

棒极了.那么，这如何帮助我们识别音频？好的，我们想用这个频谱图来标识这首歌唯一性。问题是，如果你在你的车里使用手机，你想从电台里识别出一首歌，你会得到很多噪声——背景声里有人的聊天声，另一辆车的喇叭声等等。我们得找到一个可靠的方法来抓住音频信号里的特殊“指纹”。

寻峰

现在，我们已经有了一个音频信号的频谱图，我们可以从寻找振幅的峰开始。我们定义峰是一对（时间，频率）所对应的振幅值是局部邻近值中的最高值。它周围其他（时间，频率）对应的振幅都比它低，从而不太可能产生噪声。

寻峰本身就是一个完整的问题。最后我把频谱处理成图像，利用SciPy中图像处理工具包和技术来寻峰。结合高通滤波器（强调高振幅）和SciPy局部最大值方法来实际地处理。

当我们提取了这些抗噪峰值，我们就发现了标识一首歌的兴趣点。我们只要有效地“挤压”下频谱就能发现峰。振幅已经达到目的，就不需要了。

让我们画出来看看是什么样的：



中国书法家协会入会条件细则（2016年12月最新发布）



全世界254个民族的共同传说！史前灭绝人类的大洪



小程序开发公司

python是干嘛的

公众号推广

python 爬虫

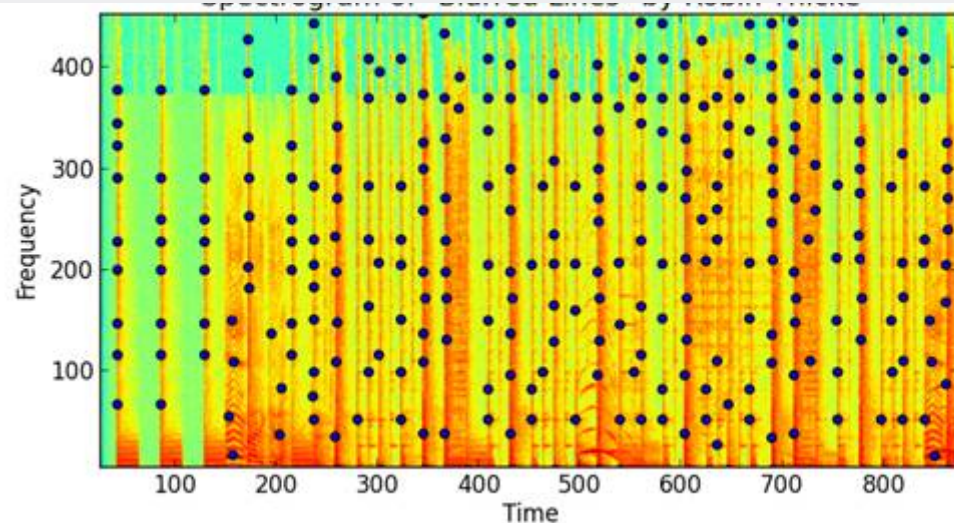


免费推广平台

python课程

python 是什么

python 学习



你会注意到有很多

兴趣点。每首歌实际上有成千上万个点。美妙的是，因为我们已经完成了振幅，我们只需要两样东西，时间和频率，就可以很方便的生成离散的整形值。基本上我们已经转变好了。

我们有一个有点精神分裂的情况：一方面，我们有一个系统，能把一个信号的峰转变成一个个离散数值对（时间，频率），让我们可以免除一些噪音。另一方面，由于我们已经完成离散化，我们已经把峰的信息从无限减少到了有限，这意味着在一首歌中发现的峰能够（提示：也确实会！）和其他歌曲中提取出的峰重复。不同的歌曲很有可能会出现相同峰！那么现在怎么办？

音频指纹哈希化

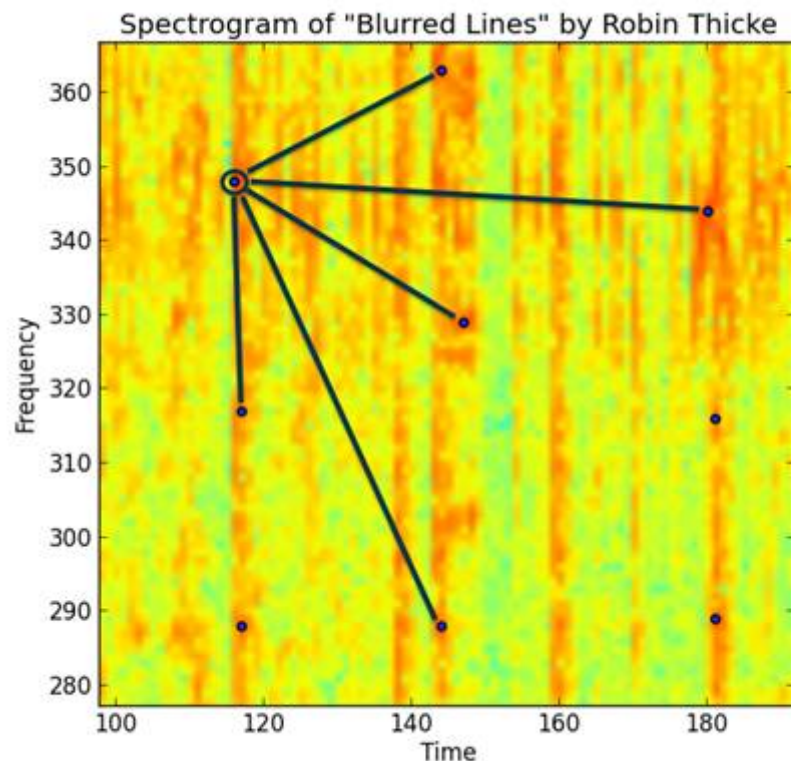
所以也许会有相似的峰。没问题，让我们把峰转化为指纹！我们将使用哈希函数来做这个。

一个哈希函数需要输入一个整数，并且返回另一个整数作为输出。美妙的是一个好的哈希函数不仅每次相同的输入就会返回相同输出，而且有稍微有点不同的输入也会出现相同的输出。

看一看我们的频谱峰，结合峰的频率和峰之间的时间差，我们可以创建一个哈希值，表示这首歌的一个独有的音频指纹。

处理这个问题，Shazam有他们自己的，SoundHound也有，还有其他。你可以阅读我的源码，看看我是怎么做的，但问题是，考虑到你创建的音频指纹包含一个以上的峰会有更多的信息熵，因此包含更多信息。因此，这有利于标识歌曲，因为它们与其他歌曲出现重合的可能性也更少。

你可以从下面放大的频谱片断中标记看看是什么情况：



Shazam的白皮书把

这些有序的峰比作“星座”用来识别歌曲。实际中，他们使用的是峰值和峰值之间的时间差。你可以想象有很多不同的方法来分组这些点和音频指纹。一方面，在一个音频指纹中有多峰，这种罕见音频指纹比较好识别歌曲。但多峰也意味着面对噪声不太好处理。

学习一首歌

现在我们可以开始了解这样一个系统是如何工作了。一种音频指纹识别系统做到两点：

2.通过在数据库中搜索已经学习过的歌曲来识别一首未知歌曲

为此，我们将用到以上的所有知识和MySQL数据库功能。我们的数据库将包含两个表：

1.指纹记录

2.歌曲记录

音频指纹记录表

音频指纹记录表有如下字段：

```
CREATE TABLE fingerprints (  
  hash binary(10) not null,  
  song_id mediumint unsigned not null,  
  offset int unsigned not null,  
  INDEX(hash),  
  UNIQUE(song_id, offset, hash)  
);
```

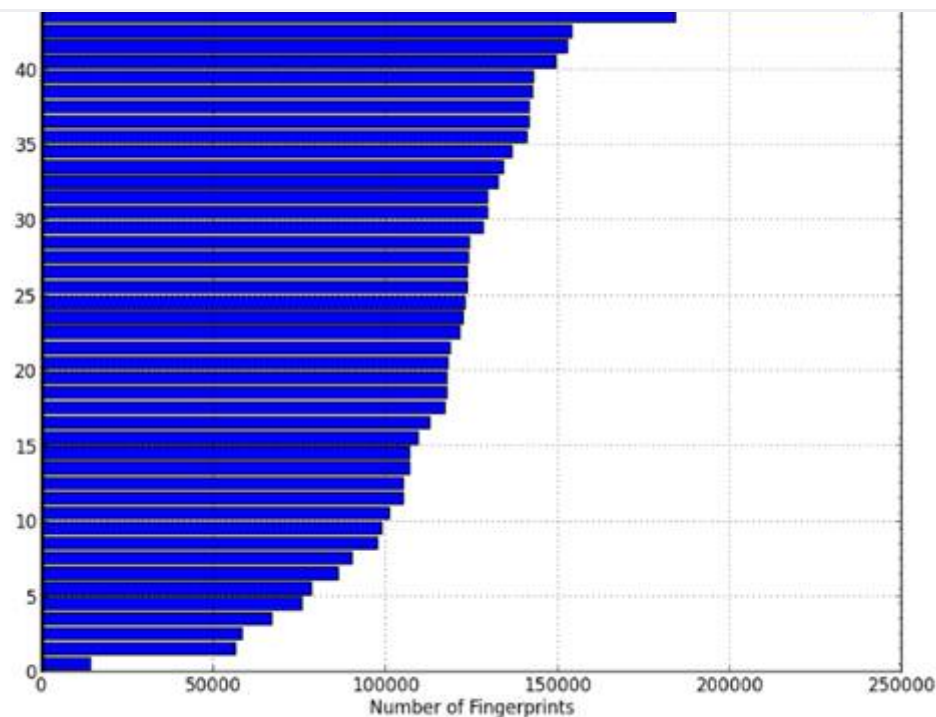
首先，注意到我们不

仅有一个hash和song_id 字段，还有一个offset 字段。这对应于哈希来源的谱图上的时间偏移量。这会在后面我们在通过匹配哈希值来过滤时用到。只有哈希值与真实信息一致才是我们真正要识别出的（更多看下面音频指纹比对）。

其次，我们有很好的理由把hash 设置成INDEX。因为所有的查询都将需要做匹配操作，所以这里我们需要一个真正的快速检索。

然后，UNIQUE只是确保我们没有重复。无需浪费空间或者因为重复的音频影响匹配查询速度。

如果你在绞尽脑汁地想我为什么把 hash 设置成 binary(10)，原因是，哈希值通常太长，设置少点有得于减少存储。下面是每首歌的音频指纹数图：



最前面的是Justin

Timberlake 的"Mirrors"，音频指纹数超过240K，其次Robin Thicke 的"Blurred Lines" 也有180k。底部是acapella 演艺的"Cups"，是一首乐器很少，仅有人声和和声的歌曲。做个对比，听听 "Mirrors"。你会发现明显的乐器声组成的“噪音墙”并且填充的频谱数从高到低分类，即频谱丰富与否与峰的频率高低是一致的。这个数据集里每首歌平均超过100k个音频指纹数。

有这么多的指纹，我们需要从哈希值水平上减少不必要的硬盘存储。对于指纹哈希，我们将开始使用SHA-1哈希，然后减少一半大小（只有前20个字符）。这使我们每个哈希值减少了一半的字节数：

```
char(40) => char(20) 从 40 bytes 到 20 bytes
```

下一步，我们将采取十六进制编码，并将其转换为二进制，再次大幅削减空间：

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)

hash字段从320 bits降到了80 bits，减少了75%。

我在系统中第一次尝试时，我把hash字段设置成了char(40)-这导致了单单音频指纹表就占据超过了1GB的空间。

设置成binary(10)后，我们把表的大小降低到只需377M就能存储520万个音频指纹。

我们确实丢失了一些信息——从统计学的角度来说我们的哈希值现在碰撞的更频繁。我们降低了哈希相当多的“信息熵”。然而，重要的是要记住，我们的熵（或信息）也包括offset字段，这有4个字节。这使得我们每个音频指纹的总信息熵为：

```
10 bytes (hash) + 4 bytes (offset) = 14 bytes = 112 bits = 2^112  
≈ 5.2+e33 possible fingerprints
```

就说，我们已经节省

了自己75%的空间，但仍然有一个巨大无比的指纹空间要处理。要保证每个字段的理想分布是很困难的，但是我们已经足够的信息熵进行接下来的工作了。

英文原文：<http://willdrevo.com/fingerprinting-and-audio-recognition-with-python/> 译者：Steven



赞

[原版阅读](#)[加入收藏](#)[分享给好友](#)

分享到微信朋友圈：

第一步 打开微信底部扫一扫

第二步 扫左边的文章二维码

第三步 右上角点击转发

相关标签：[Python](#) [Numpy](#) [音频](#) [指纹](#) [处理](#)

[上一篇](#)：“多米诺林”在部落论坛问了一个问题，快去帮帮

[下一篇](#)：Python和Numpy 处理音频指纹 二

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)

昵称：

验证码：

[发表评论](#)

相关文章 Relevant

**建网站要多少钱**

h5制作

公众号推广

**短信验证码平台**

北大核刊 游戏备案

**奥迪a6l二手?**

·让 Python 更加充分的使用 Sqlite3

·狗肉场死里逃生、四肢被截，它从金毛变柯基，如

·世界顶级妖孽建筑设计，见过2个以上算你厉害！

·10分钟前，微信公布了2017年最新数据报告！

·刚刚，人类史上第一例换头手术在中国成功了！

·双11后最奇葩的淘宝差评...

·用 Linux、Python 和树莓派酿制啤酒

·她画出当妈后的苦逼生活，带娃画画，屎尿不及！

微信公众账号推荐

[显示更多](#)[PHP开发者](#)[马哥Python笔记](#)[阿铭linux](#)[php网站建设](#)[Android安卓控](#)[LinuxStory](#)

[首页](#) | [微信最新文章](#) | [微信精选文章](#) | [微信公众账号推荐](#)[提交微信公众账号](#)[微信下载](#)[手机版](#)[加入收藏](#)[Python入门](#) | [Python进阶](#) | [Python应用](#) | [Python案例](#) | [Python教程](#) | [Python资源](#)

那角落

[关于我们](#)[广告合作](#)[联系我们](#)[意见箱](#)

微信公众平台

[微信网页登录](#)[微信FAQ](#)[微信头像](#)[微信二维码](#)

微信公众账号大全

[语录](#)[生活](#)[搞笑](#)[游戏](#)[萌宠](#)[明星](#)[趣玩](#)[红人](#)[美女](#)[时尚](#)[健康](#)[旅游](#)[运动](#)[影音](#)[IT技术](#)[科技](#)[教育](#)[品牌](#)[汽车](#)[职场](#)

关注我们

[网站地图](#)[微信头条](#)[计](#)

版权声明

站长
统

本站内容全部来自于腾讯微信朋友圈,属于第三方自助推荐平台.

不承担任何法律责任,如需删除可联系我,给我发邮件

© 2015 那角落(www.najiaoluo.com) Inc. All Rights Reserved.

浙ICP备13002564号