

武汉大学教学实验报告

电子信息学院 通信工程 专业 2021 年 10 月 5 日

实验名称 哼唱识谱 指导教师 卜方玲

姓名 周轩洋 年级 2019 学号 2019302120083 成绩

一、 预习部分

1. 实验目的

2. 实验基本原理

1、实验目的

1. 巩固傅里叶变换的知识
2. 掌握使用 python 对信号进行短时傅里叶变换
3. 学会使用 python 对语音信号进行处理

2、实验基本原理

1. 音频信号

音频信号(Audio)是通过麦克风、A/D 等数据采集设备将声音转换而成的电信号,是声波频率、幅度变化的信息载体。音频信号的特征定义如下:

1) 基频与音调

基频指一个复杂声波中最低的一个频率(其他频率叫谐波)。音调主要由声音的频率决定,同时也与声音强度有关。对一定强度的纯音,音调随频率的升降而升降;对一定频率的纯音、低频纯音的音调随声强增加而下降,高频纯音的音调却随强度增加而上升。

2) 谐波与音色

谐波是指周期函数或周期性的波形中不能用常数与原函数基频的正弦函数和余弦函数的线性组合表达的部分。 $n\omega_0$ 称为 ω_0 的 n 次谐波分量,也称为 $(n-1)$ 次泛音。音色是由混入基音的泛音所决定的,高次谐波越丰富,音色就越有明亮感和穿透力。不同的谐波具有不同的幅值 A_n 和相位偏移 y_n ,由此产生各种音色效果。

3) 音宽与频带

音频信号的频带宽度简称为音宽,它是描述组成复合信号的频率范围。借助傅里叶变换,信号可以时间函数或频率函数两种形式描述,特别是周期信号和准周期信号(前者由一个基频成分和若干谐波成分,后者虽可分解为几个正弦分量,但它们的周期没有公倍数),从频率域可以很清楚地了解它们由哪些正弦分量组成。而对于非平稳信号,最典型的例子就是语音信号,它是非周期的,频谱随时间连续变化,因此由傅里叶变换得到的频谱无法获知其在各个时刻的频谱特性。最直观的想法就是用中心在某一时刻的时间窗截取一段信号,对其做傅里叶变

换,得到这一时刻的频谱;然后将窗在时间轴上移动,从而得到不同时刻的频谱,这就是短时频谱的原理。最简单的窗就是矩形窗,即直接从原信号中截取一段。

4) 音频格式

本次实验使用的音频格式为 wav 格式

2. python 语音信号处理之 librosa 库

Librosa 是一个用于音频、音乐分析、处理的 python 工具包,一些常见的时频处理、特征提取、绘制声音图形等功能应有尽有,功能十分强大。

二、 实验操作部分

1. 实验方法

2. 实验结果

1. 实验方法

1) 实验大致流程



图 1 实验大致流程

2) 对信号的处理方法

使用 librosa.load 对 wav 文件进行读取;进而用 librosa 自带的短时傅里叶变换工具对哼唱片段进行 stft 得到 stft 矩阵;之后对信号进行滤波处理;建立频率-音调对应库;再将滤波后的信号与库中的频率进行比对;最后绘制输出结果图

2. 实验结果

1) 读入 Hum.wav 文件

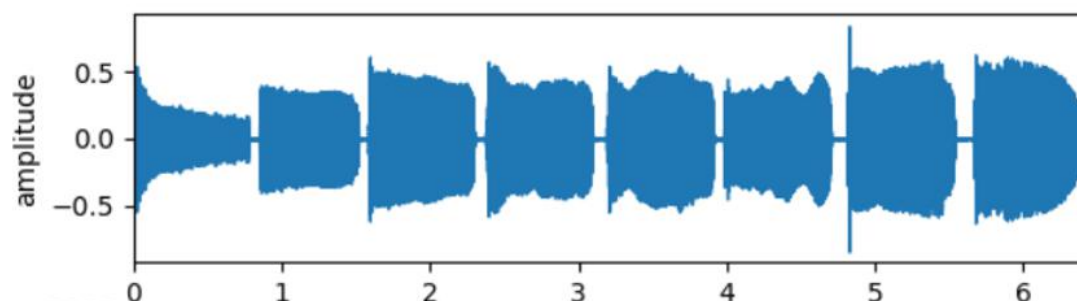


图 2 Hum.wav 文件的时域图

2) 将 stft 矩阵绘制出来

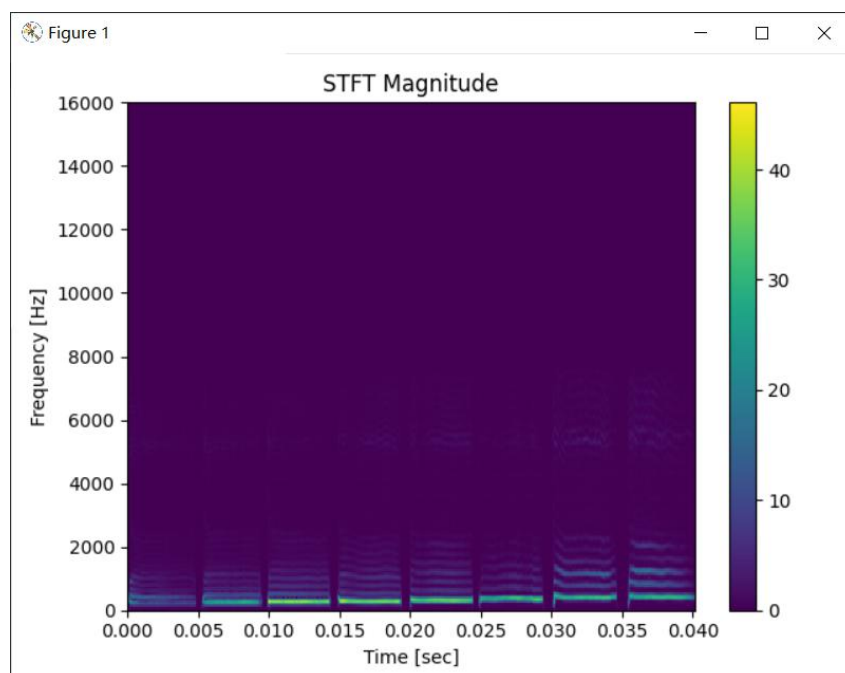


图 3 stft 矩阵

3) 进行滤波操作

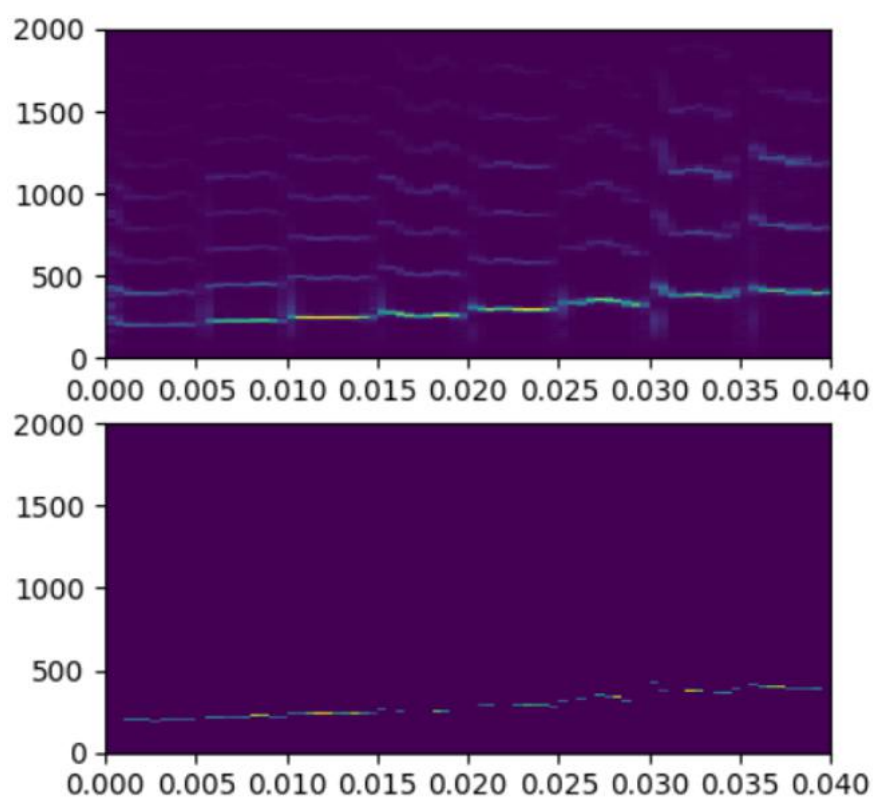


图 4 滤波前后对比图

4) 建立频率-音调对应表

Frequency in hertz (semitones above or below middle C)										
Octave → Note ↓	0	1	2	3	4	5	6	7	8	9
C	16.352 (-48)	32.703 (-36)	65.406 (-24)	130.81 (-12)	261.63 (0)	523.25 (+12)	1046.5 (+24)	2093.0 (+36)	4186.0 (+48)	8372.0 (+60)
C#/D♭	17.324 (-47)	34.648 (-35)	69.296 (-23)	138.59 (-11)	277.18 (+1)	554.37 (+13)	1108.7 (+25)	2217.5 (+37)	4434.9 (+49)	8869.8 (+61)
D	18.354 (-46)	36.708 (-34)	73.416 (-22)	146.83 (-10)	293.66 (+2)	587.33 (+14)	1174.7 (+26)	2349.3 (+38)	4698.6 (+50)	9397.3 (+62)
D#/E♭	19.445 (-45)	38.891 (-33)	77.782 (-21)	155.56 (-9)	311.13 (+3)	622.25 (+15)	1244.5 (+27)	2489.0 (+39)	4978.0 (+51)	9956.1 (+63)
E	20.602 (-44)	41.203 (-32)	82.407 (-20)	164.81 (-8)	329.63 (+4)	659.26 (+16)	1318.5 (+28)	2637.0 (+40)	5274.0 (+52)	10548 (+64)
F	21.827 (-43)	43.654 (-31)	87.307 (-19)	174.61 (-7)	349.23 (+5)	698.46 (+17)	1396.9 (+29)	2793.8 (+41)	5587.7 (+53)	11175 (+65)
F#/G♭	23.125 (-42)	46.249 (-30)	92.499 (-18)	185.00 (-6)	369.99 (+6)	739.99 (+18)	1480.0 (+30)	2960.0 (+42)	5919.9 (+54)	11840 (+66)
G	24.500 (-41)	48.999 (-29)	97.999 (-17)	196.00 (-5)	392.00 (+7)	783.99 (+19)	1568.0 (+31)	3136.0 (+43)	6271.9 (+55)	12544 (+67)
G#/A♭	25.957 (-40)	51.913 (-28)	103.83 (-16)	207.65 (-4)	415.30 (+8)	830.61 (+20)	1661.2 (+32)	3322.4 (+44)	6644.9 (+56)	13290 (+68)
A	27.500 (-39)	55.000 (-27)	110.00 (-15)	220.00 (-3)	440.00 (+9)	880.00 (+21)	1760.0 (+33)	3520.0 (+45)	7040.0 (+57)	14080 (+69)
A#/B♭	29.135 (-38)	58.270 (-26)	116.54 (-14)	233.08 (-2)	466.16 (+10)	932.33 (+22)	1864.7 (+34)	3729.3 (+46)	7458.6 (+58)	14917 (+70)
B	30.868 (-37)	61.735 (-25)	123.47 (-13)	246.94 (-1)	493.88 (+11)	987.77 (+23)	1975.5 (+35)	3951.1 (+47)	7902.1 (+59)	15804 (+71)

图 5 不同音高频率-音调对应表

5) 进行比对并输出识别结果

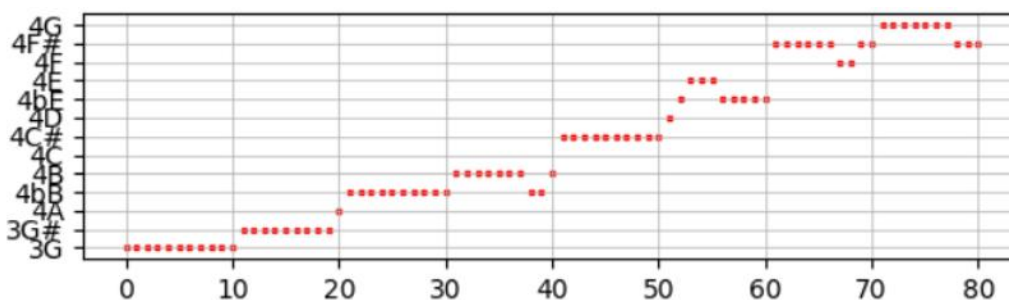


图 6 识别结果图

三、 实验效果分析（包括仪器设备等使用效果）

- 1、成功识别出了哼唱曲谱，并成功区别了同一音调的不同音高；但最终结果图仍有部分点有所偏离，原因我认为一方面是哼唱者音准有所偏离，另一方面可能是取窗不合适；可以增加用 librosa 的特征提取函数提取出哼唱的节奏，再根据哼唱节奏自适应取窗的函数来改进；
- 2、由于手动建立完整的频率-音调对应表过于繁琐，所以本人只选择了一部分频率建立的对应的表，这就造成了对视唱-8k.wav 文件不能很好的识别；日后可以将完整的对应关系封装成查找表或函数，以增加其普适性
- 3、滤波操作较为简单，采用了普通的低通+结合功率的方法，即选取功率较大且频率较低的音频成分通过；可以增加去噪、取众数、自相关等方法以改进滤波算法。

四、 教师评语
<div data-bbox="603 452 772 506" data-label="Text"><p>指导教师</p></div> <div data-bbox="1051 452 1332 506" data-label="Text"><p>年 月 日</p></div>