基于 Matlab 的听音识曲系统的设计与实现

徐清钏 赵彦钧 张丹露 弓 创 杨 光
(黄河科技学院,河南 郑州 450000)

【摘 要】文章设计了一个基于音乐片段的“听音识曲”系统,系统分为两部分:第一部分是对音频指纹进行提取。具体来说就是对已经下载好的音乐进行指纹提取,以音频指纹与乐曲名称匹配的方式建立音乐库。第二部分是对音频特征的识别,用“音乐文件特征匹配”或“录音歌曲匹配”的方式将待识别的音频导入系统,分别使用 MFCC、动态时间规整(DTW)的算法对音频信号进行特征提取以及匹配,系统在音乐文件识别上的正确率可以达到 100%,在录音文件的识别上正确率达 50%~60%。

【关键词】音频指纹;听音识曲;动态时间规整;Matlab

中图分类号:G642.423;TP391 文献标识码:A DOI:10.19694/j.cnki.issn2095-2457.2021.18.37

0 引言

随着信息时代的发展,语音识别技术在生活中的应用越来越广泛,国内有很多机构都在研究这方面的工作。“听音识曲”研究是当前的热门领域,本文通过研究原有的听音识曲系统, 掌握识别的基本原理,并使用 Matlab 程序搭建一个简易的听音识曲系统。

1 系统整体架构

整体流程分是:运行 GUI 界面,然后择曲库的路径导入曲库,在导入曲库后可以选择识别方式,最后点击开始识别等待识别输出结果即可。图 1 即为整体流程。

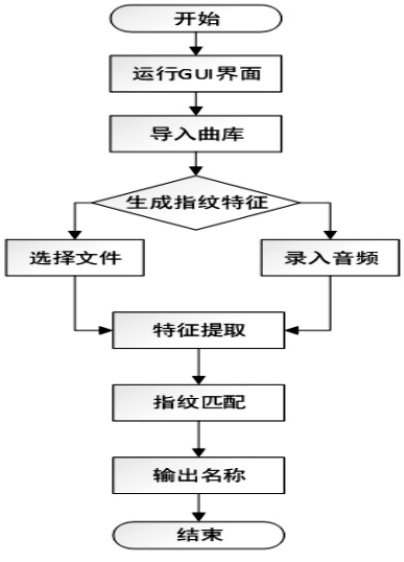

图 1 整体流程

2 音频指纹提取

整个提取的过程是:添加曲库的音乐文件、对转换过格式的文件进行预加重、分帧、加窗、快速傅里叶变换、梅尔滤波器设计、输出梅尔系数。音频指纹提取整体流程如图 2 所示。

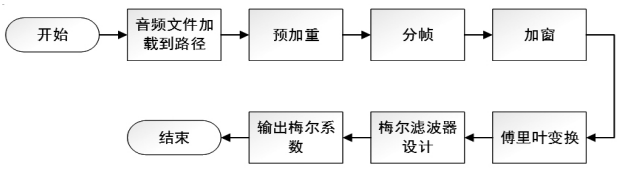

图 2 音频指纹提取整体流程

2.1 分帧加窗

音频信号处理之前要进行截短处理[7],通过函数来完成分帧效果。

加窗分帧操作通过窗函数实现,每移动一次窗函数就得到一帧的音乐片段。把音频段分割成大小为 20 ms 的帧,音乐片段之间需要重叠 1/3,并使用相同长度的窗函数对每一帧平滑处理。时域波形和频谱图如 3 所示。

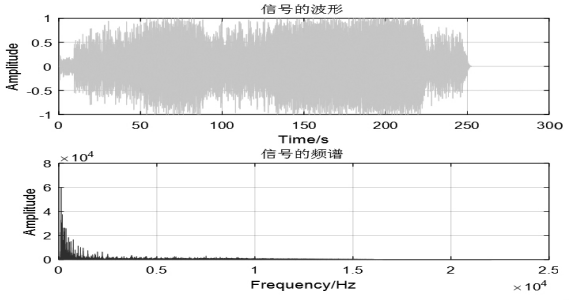

图 3 时域波形和频谱图

※基金项目:河南省教育厅 2020 年省级大学生创新创业训练计划项目(S202011834079)。

作者简介:徐清钏(2001—),男,河南洛阳人,本科,就读于黄河科技学院,研究方向为通信与信息处理。

本系统使用汉宁窗，用 $\omega_N(n)$ 表示窗函数，用 N 表示窗函数的长度，其表达式如式(1)所示：

$$\omega_N(n)=\begin{cases}0.5-0.5\cos[2\pi n/(N-1)], & 0\leqslant n\leqslant N-1\\ 0, & \text{其它 } n\end{cases} \quad (1)$$

2.2 音频指纹划分

将经过 FFT 变换的音频信号，按照帧长、帧移和峰值幅度进行分帧，每一帧都是音频指纹的特征，将其划分为 30 个不同的子代可以更好更准确地提取指纹信息。若音频匹配比较粗糙的话，可以分割更多帧长，以达到后期更好的匹配效果。音频指纹划分的公式如式(2)所示：

$$f(m)=\exp(\log F_{\min}+(m-1)\frac{\log F_{\max}-\log F_{\min}}{M})$$
$$m=1,2,...M+1 \quad (2)$$

式中，$F_{\min}$ 为最小频率值，即 310Hz，$F_{\max}$ 为最大频率值，即 2100 Hz，M 表示子带个数，$f(m)$ 表示第 m 子带的起始频率，同时也是第 $m-1$ 子带的终止频率。

2.3 MFCC 算法原理

梅尔倒谱系数[6]，这个参数的计算首先就要对语音信号通过一个高通滤波器预加重处理，然后对音频信号进行分帧，以及梅尔滤波器、窗函数的设计和傅里叶变换的应用。

3 音频识别算法

3.1 DTW 算法应用

待识别片段与曲库中的音乐片段绝大多数不一致，一首完整的音乐大概 300 s，而识别过程中，识别过程在 15 s 左右，两段音频长度不同，因此在一定程度上需要特殊的算法，对特征参数序列(见图 4)重新进行时间的校准。所以可以选用 DTW 算法[3]。

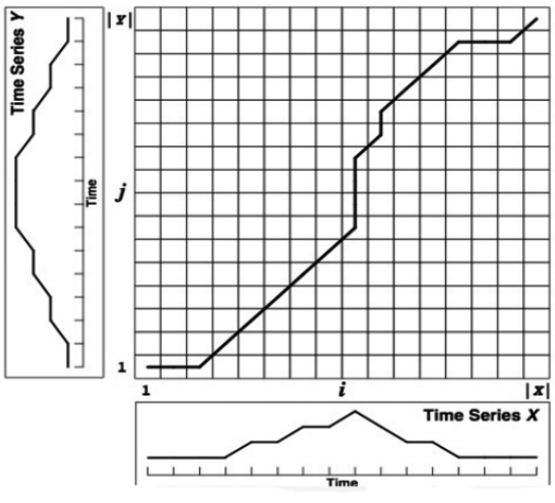

图 4 特征参数序列

3.2 最小似然距离拟合匹配

此拟合算法首先生成一个 i×j 的矩阵，i 是矩阵的行数，j 是矩阵的列数，人们称它为似然距离，用 p 表示，似然系数用 q 表示，因为系数矩阵的特征值具有独特性，所以每一帧音频的似然系数是不同的，分别对每一首待识别的音频进行计算，求出似然系数，并将与所建立音乐库里的歌曲的似然系数值作差再取绝对值，能得到一个最小值，这个最小值就是最小似然距离(见图 5)，并通过此算法筛选出这个最小值，它对应的歌曲就是识别的结果。原理公式如式(3)：

$$[p,q]=|\min[i*(i-1),j*(j-1)]|+\text{Dist}(i,j) \quad (3)$$

式中，p 为 ；q 为 ；i 为 ；j 为 。

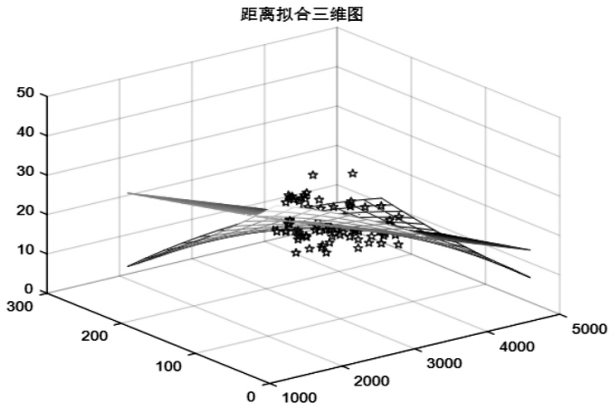

图 5 最小似然距离

4 GUI 界面设计

GUI 界面是该系统的操作界面(见图 6)，可以根据界面上的文字提示来逐步完成对音乐的识别：打开 MATLAB 平台，点击运行即可。

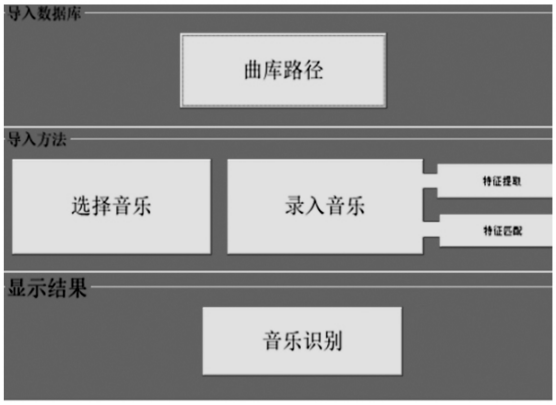

图6 GUI 界面

点击曲库路径按钮，可以自行跳转到所建立的音乐库路径，任意点击音乐库里的一首歌，可以将全部歌导入到检索曲库里。

点击选择音乐,选择一首待识别的音乐对该首音乐提取特征,生成索引值,与最初的模板库里多个索引信息建立的索引库进行匹配。

5 歌曲识别率

由于“听歌识曲”系统主要针对歌曲进行设计,因此,随机挑选出当下较热门的 5 首流行歌曲进行测试。为了避免误差以及确保验证系统的稳定性,每首歌曲分别进行十次测试,记录下正确次数并且取平均值,计算正确率,系统在音乐文件识别上的正确率可以达到 100%,在录音文件的识别上正确率达 50%~60%。具体识别结果见表 1、表 2。

表 1 音乐文件识别率

乐曲名	识别次数	平均正确次数	平均正确率/%
Shape of You	10	10	100
错位时空	10	10	100
十年	10	10	100
Here We Are	10	10	100
起风了	10	10	100

表 2 录音文件识别率

乐曲名	识别次数	平均正确次数	平均正确率/%
Shape of You	10	6	60
错位时空	10	7	70
十年	10	5	50
Here We Are	10	6	60
起风了	10	7	70

6 结语

本文所设计的“听音识曲”系统,可以帮助人们识别一首不知名的音乐片段,使人们不再为错失一首好听的音乐而烦恼。系统主要完成了将音频信号转化成音频特征、乐库的建立、特征匹配系统的搭建和 GUI 页面的生成等工作;基本达到了预期的要求,能够满足人们的需要。当然,系统在设计方面还存在一些缺陷,譬如说,通过录音的方式进行识别的准确率还不够高,如果需要提高准确率,还要用一些更专业的录音设备和对滤波除噪声算法进行更深层次的研究,对 MFCC、DTW 以及最小似然距离拟合匹配算法的优化,减少其他噪声的干扰;本文中的曲库做得还不够丰富,只有一少部分的音乐,能够识别的乐曲有限。今后,可以从增加乐库歌曲信息与反应时间两方面进行优化,尽可能多地增加乐曲,尽可能少地缩短识别时间,尽量使用更加专业的录音设备,将录音识别方式识别歌曲的准确率上再做一些提升。

【参考文献】

[1]朱布裔.基于指纹匹配的音乐检索系统的设计与实现[D].武汉:华中科技大学,2017.
[2]胡俊,李胥,陈毅.一种音频指纹检索算法的改进方法[J].工业控制计算机,2018,31(2):92-93.
[3]张学帅,邹学强,胡琪,等.基于指纹权重的音频模板检索方法[J].中国科技论文,2018,13(20):2295-2300.
[4]王伟,陈志高,孟宪凯,等.基于熵的音频指纹检索技术研究与实现[J].计算机科学,2017,44(S1):551-556.
[5]何旭.基于旋律特征的实时音乐检索系统[D].南京:东南大学,2017.
[6]张新彩.基于内容的音乐检索技术研究与实现[D].西安:西北大学,2009.
[7]李艳凤,黄琳琳,陈后金,等.数字信号处理“听音识曲系统”综合实验设计[J].信息通信,2019(11):81-83.
[8]姚姗姗.音频大数据检索关键技术研究[D].太原:太原理工大学,2018.