一种基于“乐纹”的海量音乐检索系统*

徐英进[1+]，蔡　锐[2]，蔡莲红[3]

（清华大学计算机科学与技术系,北京　100084）

摘　要: 本文提出了一种优化的乐纹提取方法，并通过实验分析了该提取方法的有效性、抗噪性和对于查询速度的影响等。本文还提出了一种快速的查询方法，并通过实验定量的分析了该查询方法的速度和准确率。并建立了一种基于“乐纹”的海量音乐检索系统。

关键词: 乐纹;音乐检索

1. 引言

根据音乐片段从海量音乐数据库中快速找到对应的音乐，具有很大的实用价值，也是一个极具挑战的研究课题。随着基于内容音乐检索系统逐步获得应用，将给广大喜爱音乐的听众带来很多方便。

音乐的物理特性表现相当复杂，但一首音乐的某些特征是相对确定的，可以看作表征该音乐“身份”的“指纹”。能否用音乐的“指纹”(Music-Fingerprinting，MFP)来表征一段音乐，并根据该“指纹”来查询我们所需要的音乐？答案是肯定的。本文中，把代表一段音乐的有效特征序列（音乐指纹）命名为“乐纹”，并建立了一种基于“乐纹”的海量音乐检索系统。

关于乐纹的提取，学者们已经进行了很多的研究工作。广泛被使用的方法是从经过短时-傅里叶变换（Short-time Fourier Transform）以后的频谱图里面选择一些特征序列为该片段的乐纹。典型的方法有两种：一种是荷兰的Philps研究所提出的基于全局信息的方法。这个方法首先将整个频谱分成很多小块，每个小块由0或者1来表示，这样整个频谱可以用二进制数的序列来表示。这种方法的特点是可以表示整个频谱的全局信息，缺点是信息量代表性较差，且抗噪性能欠佳。另一种方法是英国的Shazam公司提出的基于特征点的方法。这个方法是从频谱里寻找一些特征点，组成特征点对（Peak-Pairs），把特征点对的序列作为该片段的乐纹。此方法的特点是不需要保留整个频谱的全局信息，优点是有效信息量集中，抗噪性比较好。

本文基于Shazam公司的思路，提出了一种优化的乐纹提取方法，并通过实验分析了该提取方法的有效性、抗噪性及其对查询速度的影响等。同时，我们还注意到，海量音乐检索系统的另一个难题是查询的精确率和速度问题。一方面，由于用户所输入的音

* 国家自然科学基金资助 （60433030，60418012）

* 联系作者, Email: clh-dcs@tsinghua.edu.cn

乐片段可能包含很强的噪音，而且提取乐纹的过程中还可能产生一些误差，因此查询的精确率不但重要而且难于提高；另一方面，考虑到实际应用对适时性的要求，查询速度也是很重要的一个问题。本文提出了一种快速的查询方法，并通过实验定量分析了该查询方法的速度和精确率。

本文安排如下。首先，第二节介绍了整个基于乐纹的海量音乐检索系统的结构设计。第三、四节则分别介绍了乐纹的提取方法和相应的快速查询算法。最后在第五节中给出相应的实验结果。

2. 音乐检索系统

海量音乐检索系统的总框架如图 1 所示。主要包括两大部分：建库过程和查询过程。建库过程将容量庞大的音乐库转换成乐纹库；查询过程基于输入样本片段匹配检索获得用户所需要的音乐信息。具体的各个模块的功能如下。

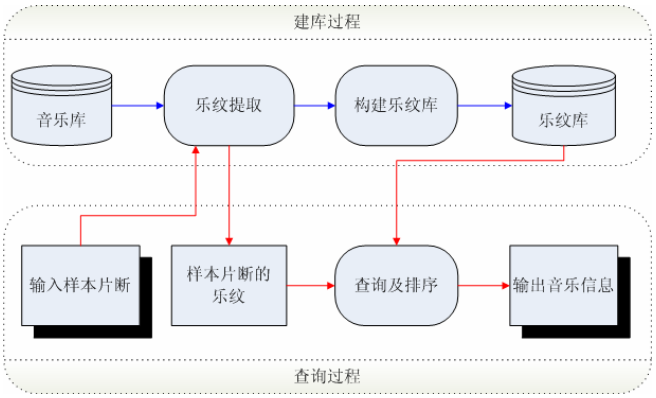


图 1 基于“乐纹”的音乐检索系统

建库过程包括特征计算、乐纹提取、乐纹库构建。为了快速查询，乐纹的存储采用 hash 表结构。把从音乐库的每首歌曲提取的乐纹按照 hash 表的数据结构保存到乐纹库里。最终完成把容量庞大的音乐库转换成乐纹库的工作。

考虑音乐库里的每首歌曲的乐纹与用户所输入的样本片段的乐纹的交集，这个交集不为空集的所有歌曲都可以成为“匹配候选者”（matching candidate）。首先求出样本片段的匹配候选者集，然后利用一定的度量算法计算出每个候选者与输入片段之间的“距离”（distance）。经过排序以后选择距离最小的前 N 首歌曲作为检索结果的输出。

3 一种优化的乐纹提取方法

参照 Shazam 公司的方法，基于特征点的乐纹提取方法流程如图 2。

图 2 “乐纹”提取流程

预处理是归一化过程，这个过程将输入的音乐片段转换成单声道、8K 采样率的 wav 格式。分帧后进行短时-FT 变换，然后从频谱中选择峰值点（Peak）作为特征点。对于 2 维频谱上的每一个峰值点，在固定大小矩形（$\Delta f \times \Delta t$）范围内寻找其它峰值点，配对组成峰值点对（Peak-Pairs）。这些特征点对的序列成为该片段的乐纹。

实际上，短时-FT 变换得到的频谱包含了太多的信息，又加上噪声的影响，峰值点对（Peak-Pairs）的正确提取是十分困难的。本文重点研究了：①选择最有物理意义的特征点；②提高查询速度和抗噪性能。

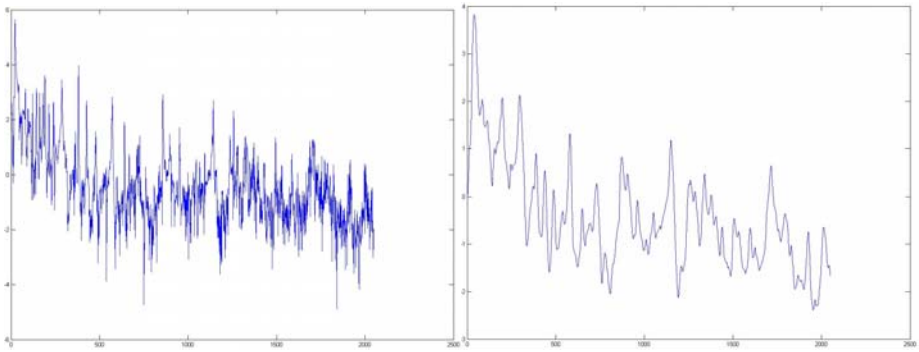
3.1 选择最有物理意义的特征点

选择频谱上的峰值点作为特征点，是因为这些点一般能代表某个较强频率分量的突现，而这些频率分量往往代表特定时刻特定乐器/人声的出现。实际处理中，经过傅里叶变换以后的频谱存在很多毛刺，有非常多的伪峰值点。为了能更好的选择最有物理意义的特征点、即在短时间内最突出的频率分量，本文提出一种频谱优化方法来去掉频谱包络上的毛刺。同时，在特征点选择时还面临的会出现重复的拥有相同物理意义的特征点的问题，即能量很强的某一个频率分量在持续一定时间的情况下相关音频帧都会出现相近取值的频率分量的特征点，导致物理意义的重复。本文还提出了的一种峰值过滤的方法来解决这一问题。

3.1.1 频谱优化

本方法利用低通滤波器来去掉帧级频谱中毛刺，只保留能量最突出的少量峰值点。

如果将帧级频谱包络也看作一个时域上的波形，则包络上的毛刺可以看成是由很多高频分量产生的。因此，可以通过低通滤波器来去掉包络上的毛刺。图 3 是频谱优化之前的和频谱优化之后的两个帧级频谱图的例子。



频谱优化之前　　频谱优化之后

图 3 频谱优化之前的和频谱优化之后的某帧频谱

3.1.2 峰值过滤

这个过程使用最大值法过滤掉一些在很短时间内重复的频率分量峰值点，即能量很强的某一个频率分量持续一定时间的情况下，选择该时间段内能量最大的一个点，之外时间点的频率分量都认为是重复的，不为特征点。

对于每个帧级的峰值点（x, y）都进行如下的操作

If　$E_{xy} == \max(E_{ij})$　E_{xy} 为特征点，Else　E_{xy} 不为特征点

其中 x 是该峰值点的频率值，y 是该峰值点的时间值。有 $x-df \le i \le x+df$ ，$y-dt \le j \le y+dt$ 。df 为允许频率误差，在这个误差范围内的两个频率被认为是同一个频率分量。dt 为最大时间限制，在这段时间内的同一个频率分量被认为是重复的频率分量。

3.2 提高查询速度和抗噪性能

一方面，“选择最有物理意义的特征点”已经保证了对于能量很小的噪音的过滤。同时，乐纹提取还必须考虑到查询时的需求，为了保证查询的快速和更可靠的抗噪性，本文提出了“特征点置信度（confidence）”的概念。

对于每一个特征点，其置信度被定义为：$C=1-\frac{1}{2}(e^{-x_1}+e^{-x_2})$

其中 x_1 表示帧内的能量突出性，x_2 表示同一个频率分量在一段时间内的能量突出性。

置信度是衡量每个特征点的唯一的标志，特征点对的置信度是两个特征点的置信度的平均值，本文还设置了一个置信度的阈值（0-1 之间的很小的一个值），置信度小于阈值的特征点对被认为是噪音特征点，被去掉。

本文最终采用的基于峰值对的乐纹提取方法的流程图为如下：

图 4 基于峰值对的乐纹提取方法的流程

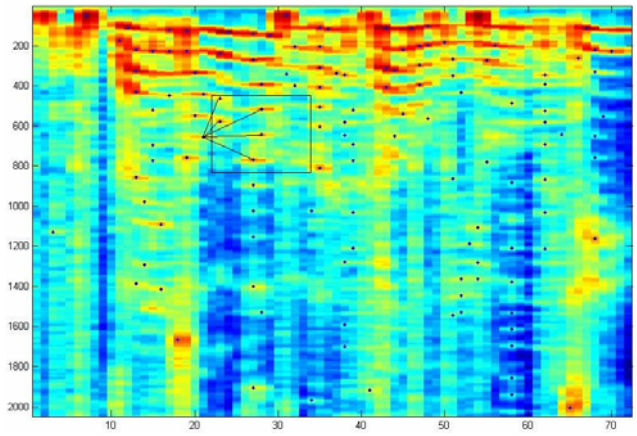


图 5 为从经过优化的 2 维频谱当中提取的特征点和特征点对的例子。

图 5 从优化的频谱中提取的特征点和特征点对

4 快速查询

考虑到现实应用的需要，查询必须满足正确性和快速性。本文提出了两种保证快速性的机制，分别为乐纹库的 Hash 结构和三重链表查询结构。本文还提出了一种保证正确性的机制，及优化的距离度量方法。

乐纹库的结构

乐纹库采用的是 hash 表结构，哈希表的每个桶的数据结构为一个特征即峰值点对，具体的 $(f_1, \Delta t, \Delta f)$，

其中，f_1 为第一个峰值（Peak）的频率，Δt 为第一个峰值和第二个峰值之间的时间差值，Δf 为第一个峰值和第二个峰值之间的频率差值。每个桶对应着一条链表，链表的每个元素记录着包含该特征的音乐的基本信息，如该特征在该音乐里出现的位置和该特征在该音乐里的置信度等。

这样，当某一个输入片段需要查询时，首先提取该片段的乐纹，进而根据提取得到的乐纹访问乐纹库，乐纹库当中相应乐纹桶对应链表的中的每首歌曲均被放到“匹配候选者集”（matching candidate set）。这样避免了线性搜索所带来的巨大时间开销。

三重链表查询结构

匹配候选者集的构造是查询过程的第一步，而匹配候选者集时采用的数据结构也是决定查询效率和速度的另一个重要的一个因素。其中需要解决的一个关键问题是：一个特征可能在同一首歌的不同位置多次出现。

本文提出的匹配候选集的三重链表结构为如下：

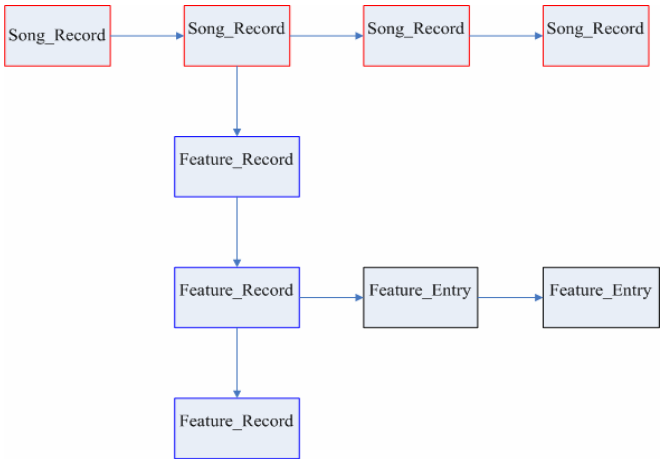


图 6 匹配候选集的三重链表结构

其中 Song_Record 为匹配候选集的每首歌曲的信息，Feature_Record 为该歌曲包含的每个特征，Feature_Entry 为该特征在该歌曲里出现的每个位置。

距离度量方法

度量每个匹配候选者与输入片段之间的距离的算法直接决定了查询的正确性。

本文提出了一种基于匹配率和时间偏移集中度的距离度量方法。具体的，一首歌曲和一个输入片段之间的距离为：

$$d = \frac{1}{R_M^{W_1} \times R_O^{W_2}}$$

其中 R_M 为匹配率，R_O 为时间偏移集中度， W_1 和 W_2 分别为匹配率和时间偏移集中度的权重。

匹配率表示该片段和该歌曲之间的特征的匹配程度，计算公式为

$$R_M = \frac{M}{T}$$

其中，T 为该片段的特征总数，M 为这些特征当中与该歌曲匹配的特征数。

时间偏移集中度的计算公式为

$$R_O = \frac{P}{M}$$

其中 M 为匹配的特征数，P 为分布图的最高峰的高度。

时间偏移集中度表示匹配的特征序列的有效性。如果某一个片段属于某一首歌曲，则从片段提取的特征和跟它相应的歌曲的特征之间的时间偏移是一定的，因此查询成功的情况下匹配的特征序列的时间偏移分布高度集中，而在失败的情况下时间偏移分布较为分散。

5 实验结果及讨论

根据上述的设计实现了音乐检索系统，并进行了相关的性能和查询速度的测试。

音乐库的大小为 531 首歌曲，包括中文、英文和日文的流行歌曲，还有一些伴奏音乐。测试样本为从 9 首歌曲当中提取的 27 个无噪音的片段，，以及同样 27 个片段在四种噪音环境下的 80 个带噪音的片段。所有片段的长度均为 15 秒。具体四种噪音及其相应的样本数如表 1：

表 1 ：检索样本说明

	具体环境说明	片段数
噪音 1	办公环境安静，偶尔有点击鼠标键盘的声音	12
噪音 2	汽车环境较吵，无人说话，偶有喇叭的声	12
噪音 3	办公环境安静,有空调声	27
噪音 4	汽车环境较吵，有空调声	29

相应的测试结果如表 2：

表 2：测试结果

	平均特征个数	平均检索速度（秒）	识别率（%）
无噪音片段	1193.4	6.70	100
带噪音 1 片段	790	5.33	100
带噪音 2 片段	1189.8	7.08	75
带噪音 3 片段	437	4.41	100
带噪音 4 片段	395.2	3.90	82.76

从结果可以看出，带噪音片段的特征点对的个数比无噪音的原始音乐片段少了很多，这是因为在实际噪音环境中使用录音设备录音时丢失了很多细节特征。但是，需要注意，在强噪音环境下（如噪音 2），由噪音而产生的特征也会导致片段的特征数的增加。从结果还可以看出，检索速度随着平均特征数的增加而变慢。从结果来看，识别率已经令人满意，除了无噪音原始片段以外，在办公室录的片段的识别率也已经达到了 100%。像汽车环境那种噪音很强的条件下，识别率还稍微低一些，而且噪音特征比较多还影响到检索速度，这些是下一步研究中还需解决的问题。

参考文献

[1] Jaap Haitsma, Ton Kalker. “A Highly Robust Audio Fingerprinting System”.

[2] Jaap Haitsma, Ton Kalker and Job Oostveen. “Robust Audio Hashing for Content Identification”.

[3] Michael Mandel, Audio Fingerprinting for Recognition

[4] Shazam website. http://www.shazamentertainment.com

[5] Philips (audio fingerprinting) website. http://www.research.philips.com/InformationCenter/Global

A Music Fingerprinting-based Large-Scale Music Retrieval System

SO Yong-jin[1], CAI Rui[2], CAI Lian-hong[3+]

[1](Department of Computer Science and Technology, Tsinghua University, 100084, China)

[2](Department of Computer Science and Technology, Tsinghua University, 100084, China)

[3](Department of Computer Science and Technology, Tsinghua University, 100084, China)

+ Corresponding author: Phn: +86-10-6277-1587, E-mail: clh-dcs@tsinghua.edu.cn

Key words: music fingerprinting; music retrieval

Abstract: In this paper, we proposed an improved approach for music fingerprint extraction. This approach was evaluated in our experiments to show its effectiveness, noise robust, and efficiency. We also proposed an efficient query algorithm to build a large-scale music retrieval system based on music fingerprinting. Experiments showed the system can provide high precise query results in a short response time.