

# 一种高效过滤-提纯音频大数据检索方法

张兴忠<sup>1</sup> 王运生<sup>1</sup> 曾 智<sup>2</sup> 牛保宁<sup>1</sup>

<sup>1</sup>(太原理工大学计算机科学与技术学院 太原 030024)

<sup>2</sup>(中国科学院自动化研究所 北京 100190)

(1659898176@qq.com)

## An Efficient Filtering-and-Refining Retrieval Method for Big Audio Data

Zhang Xingzhong<sup>1</sup>, Wang Yunsheng<sup>1</sup>, Zeng Zhi<sup>2</sup>, and Niu Baoning<sup>1</sup>

<sup>1</sup>(College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024)

<sup>2</sup>(Institute of Automation, Chinese Academy of Sciences, Beijing 100190)

**Abstract** Fast audio retrieval is demanding due to the high dimension nature and increasingly larger volume of audios in the Internet. Although audio fingerprinting can greatly reduce its dimension while keeping audio identifiable, the dimension of audio fingerprints is still too high to scale up for big audio data. The number of audios to be checked has to be small enough. This paper proposes a robust and fast audio retrieval method for big audio data, which combines audio fingerprinting with filtering-and-refining method. An audio middle fingerprint is devised with considerable small dimension for quickly filtering most likely audios, by applying bag-of-features(BoF) technique on the classical Philips audio fingerprint, which can reduce the search scope with a 130 times speed gain compared with the Fibonacci Hashing retrieval. A matching algorithm is developed to reduce the computational complexity by comparing the samples at fixed interval of two audios with thresholds, which results in a maximal speed gain of 140 times. Experimental results show that the average time of retrieving audio clips of different length in about 100 000 audios is less than 1 s. After applying MP3 conversion, resampling, and random shearing, the recall rates are all above 99.47%, and the theoretical accuracy is close to 100%.

**Key words** big audio data; efficient retrieval; Philips audio fingerprint; filtering-and-refining; audio middle fingerprint

**摘 要** 针对互联网音频大数据的高速检索问题,结合音频指纹技术与过滤-提纯思想,提出一种面向音频大数据的鲁棒高效检索方法.在经典的 Philips 音频指纹基础上,提出了一种基于 bag-of-features (BoF)的音频中间过滤指纹用于快速缩小检索范围,与 Fibonacci Hashing 检索相比提高检索速度约 130 倍;并设计了一种基于阈值的固定间隔抽样匹配方法,大幅减少匹配计算量,进一步提高检索速度可达 140 倍.实验结果显示:使用该方法在约 10 万首音频中对不同时长的音频片段进行批量检索,平均检索时间均小于 1 s;对音频进行 MP3 转换、重采样、随机剪切后再检索,召回率均在 99.47%以上,理论准确率接近 100%.

**关键词** 音频大数据;高速检索;Philips 音频指纹;过滤-提纯;音频中间指纹

**中图法分类号** TP391.42

近年来,随着互联网的普及与大容量存储器的发展,互联网中的数据规模不断膨胀,大数据已成为各行业的研究热点,如何从大数据中挖掘有价值的信息受到了各领域的广泛重视. 互联网数字音频数量的爆炸式增长,使得在音频大数据中进行高速检索成为亟待解决的难题. 音频大数据环境下高速检索技术的突破关键在于解决音频大数据的 Volume 和 Velocity 方面的特点给检索技术带来的困难. 音频指纹检索技术采用简短的音频指纹数据代替音频本身进行检索,可以有效地提高音频检索的效率;但音频大数据对应的指纹数据量也相当庞大,传统的音频指纹检索方法已经难以满足音频大数据环境下的快速精确检索需求.

音频指纹通常都具有高维的特性,而高维指纹的相似度匹配会引起计算和存储代价以指数的形式增长. 对音频指纹检索带来的维数灾难问题,不同的音频指纹需要依据其数据结构特征和应用场景等采取相应合适的指纹检索方法予以解决. 目前的检索方法分为指纹降维和建立索引 2 类,降维技术通过降低指纹数据量大幅简化计算,会导致检索准确度的降低;索引方法则可以快速缩小检索范围,但其性能过分依赖于指纹的鲁棒性.

本文参照过滤-提纯思想,提出并实现了一种面向音频大数据的高效过滤-提纯检索 (efficient filtering-and-refining retrieval, EFRR) 方法,该方法利用 Philips 音频指纹<sup>[1]</sup> 的鲁棒性及数据结构特征,结合 bag-of-features (BoF) 算法<sup>[2]</sup> 思想,生成一种数据量更小的中间指纹进行过滤检索,可以快速地缩小检索范围;同时设计了一种基于阈值的固定间隔抽样匹配方法用于过滤检索的相似度匹配过程中,大幅减少了匹配过程的计算量. 既保留了降维技术大幅减少计算量和索引方法快速缩小检索范围的优点,又能保证检索的召回率和准确度. 其思想也可用于其他数据结构特征类似的指纹检索中.

1 相关研究

音频指纹检索是一种基于内容的音频检索技术<sup>[3]</sup>,它是将检索音频指纹与音频指纹数据库中的指纹进行相似度匹配,通过比较相似度获取检索结果的过程. 目前快速检索方法的研究方向主要有降

维和建立索引 2 类.

降维技术的思想是通过减少指纹数据量来降低指纹相似度匹配过程中的计算量,从而达到提高检索效率的目的. 文献[4]提出的基于 OPCA (oriented principal component analysis) 的降维技术,对流媒体的识别很有效,但存在分类效果不理想、结果不稳定等问题. 基于此,文献[5]提出了一种 w-PCA (weighted principal component analysis) 的音频降维技术,在数据低维化方面有显著优越性,但对维数的选取较为敏感. 文献[6]提出一种求和算法,可以大幅提高检索速度,但只有当最大的特征值远大于其他特征值时适合使用. 另外,文献[7]提出一种加权自相似性的量化方法,在检索前将多维的音频特征向量指纹进行降维处理. 文献[8]对 Delta 梅尔倒频谱系数 (Mel-scale frequency cepstral coefficients, MFCC) 或 Delta 色度特征使用一种基于主成分分析 (principal component analysis, PCA) 的汇总技术进行降维后建立马尔可夫模型,有效地加速了检索过程.

索引方法的目的是通过对指纹建立索引关联,从而快速缩小检索范围,实现高效检索. 文献[9]针对 Philips 指纹提出将所有可能的音频指纹建立一个快速查询表,对音频指纹库中的指纹分别与快速查询表进行关联,可以在查询表中快速找到查询音频指纹所关联的歌曲;但当查询音频失真时,检索性能会大幅度下降. 文献[10]对该方法进行改进,提出一种基于 Fibonacci Hashing 的快速检索算法,可以根据内存的容量调整 Hash 表的大小,有效地节省了内存. 文献[11]提出将音频指纹用量化码本的形式建立索引,可以显著提高检索速度,但在构建索引时由于对特征错误率的限制,会导致虚警率升高. 另外,文献[12]将 CENS (chroma energy normalized statistics) 特征与多种容错和排列机制以及多次查询的策略结合提出一种基于倒排序文件索引的检索方法. 文献[13]对频率特征使用一种 Hash 函数提取指纹,并使用 Hash 编号来划分检索空间,在并行体系结构中具有较高的扩展性.

对于不同的音频指纹,对应的快速检索方法往往需要依据其数据结构特征和应用场景等来设计. 本文提出的方法借鉴降维技术大幅减少计算量和索引方法快速缩小检索范围的思想,针对 Philips 指纹预先生成数据量比原始指纹库更小的中间指纹库,

用来快速缩小检索范围；通过使用基于阈值的固定间隔抽样匹配方法进行指纹相似度匹配，大幅简化匹配计算量。方法经验证比同样针对 Philips 指纹改进的基于 Fibonacci Hashing 的快速检索算法检索效率提高了 800 倍以上，同时又避免了降维技术对检索精度的影响和索引方法对指纹鲁棒性的依赖等问题。

2 预备知识

2.1 Philips 音频指纹

Philips 音频指纹是一种经典的高鲁棒性音频指纹，已经在商业领域取得了成功，目前仍是世界各地科研机构的研究对象。Philips 算法的流程如图 1 所示：

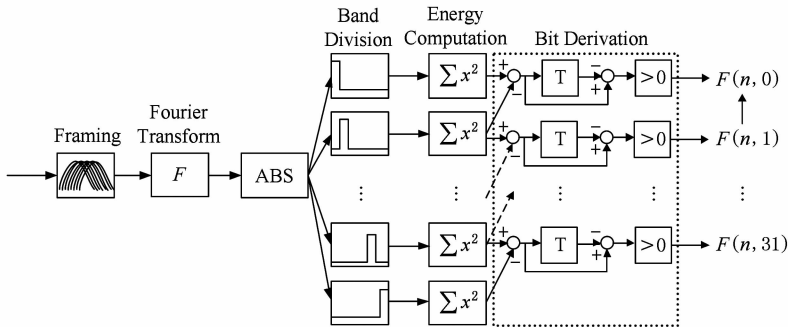


Fig. 1 Process for extracting Philips audio fingerprint.  
图 1 Philips 音频指纹提取过程

首先对每隔 11.6 ms、长度为 0.37 s 的音频信号进行加 Hanning 窗划分重叠帧，相邻帧之间的重叠率为 31/32；对单帧信号进行快速傅里叶变换，获取帧频谱并保留其模值；依据人类听觉系统遵循的 Bark Scale 规律将 300~2 000 Hz 划分成 33 个不重叠的、具有对数间隔的频带；对不同帧的不同频带分别计算其能量值总和及相邻频带的能量差；最后通过相邻 2 帧对应的能量差值确立子指纹的二进制位。假设  $E(n, m)$  表示音频第  $n$  帧在第  $m$  频带的能量， $F(n, m)$  表示第  $n$  帧的子指纹的第  $m$  个二进制位：

$$F(n, m) = \begin{cases} 1, & E(n, m-1) - E(n, m) - (E(n, m) - E(n, m+1)) > 0; \\ 0, & E(n, m-1) - E(n, m) - (E(n, m) - E(n, m+1)) \leq 0. \end{cases}$$

Philips 音频指纹提取算法快速简洁，对多种攻击具有较强的鲁棒性，音频指纹间的相似度采用比特误差率 (bit error rate, BER) 来评价，具有较快的匹配速度。

2.2 BoF 算法

BoF 算法是近年来被广泛地应用在图像检索应用中的一种成熟的快速检索技术。其核心思想是采用图像原始特征的分布统计值代替原始特征，显著地降低了图像特征的数据量，从而简化检索匹配的计算量，达到快速检索的目的。首先通过分类或聚类等手段将图像集的所有局部特征划分为若干类；然

后对每幅图像统计其局部特征在每类中的分布情况；最后使用统计值作为该图像新的特征表示。该算法忽略了原始特征在时间和空间上的顺序性以及特征数据内部的结构和逻辑相关性等，只保留原始特征的分布情况统计值作为新的图像特征。

3 高效过滤检索方法

3.1 高效过滤-提纯检索流程

高效过滤-提纯检索方法流程如图 2 所示，其中步骤如下：

- ① 检索前对参考音频提取原始指纹 (Philips 指纹)，建立音频原始指纹库，并由原始指纹库结合 BoF 算法转换生成数据量相对较小的中间指纹库，2 种指纹库中的音序列号按顺序一一对应。
- ② 检索开始时对检索音频提取原始指纹，并使用 BoF 算法转换生成对应的中间指纹。
- ③ 使用检索音频的中间指纹，在音频中间指纹库中进行过滤检索，快速筛选出可能结果的序号，构成序号集 I，缩小检索范围，作为第①步过滤检索。
- ④ 使用检索音频的原始指纹，在第③步筛选出的可能结果序号对应的原始指纹中进行过滤检索，筛选出可能结果的序号，构成序号集 II，进一步缩小检索范围，作为第②步过滤检索。
- ⑤ 在第④步筛选出的可能结果对应的原始指纹中，用检索音频的原始指纹进行精确匹配检索，获取

最终检索结果.

在上述第③④步的过滤检索中,指纹间的相似

度均使用基于阈值的固定间隔抽样匹配方法进行快速相似判定.

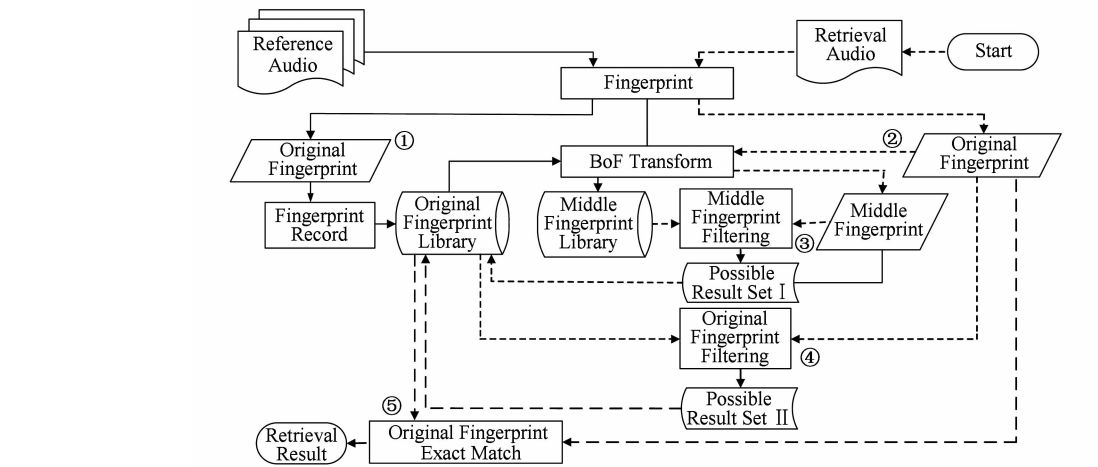


Fig. 2 Process for efficient filtering-and-refining retrieval.

图2 高效过滤-提纯检索流程

3.2 中间指纹生成

本文依据 Philips 指纹的数据结构特征以及对多种音频攻击处理表现出的高鲁棒性等,将 BoF 算法思想引入音频检索中,生成一种新的数据量成倍减少的中间指纹.使用该中间指纹虽然不能进行有效的精确匹配检索,但可以快速过滤掉大量不相关的音频,大幅缩小检索的范围.但由于该方法生成的中间指纹只记录了局部特征的概率分布情况,忽略了局部特征的位置和结构逻辑等信息,在大量的音频片段中可能存在许多局部特征分布相同但内容却毫不相关的情况,因此在本文中用作初步过滤检索的中间特征.

采用 BoF 算法对原始 Philips 指纹转换生成中间指纹,关键在于如何对可能存在的  $2^{32}$  种 Philips 子指纹进行分类.分类方法的优劣将影响中间指纹的区分度及过滤的幅度;同时,分类方法要在尽可能简便快捷的前提下必须满足以下 2 点:

1) 对音频一定程度的攻击所造成的子指纹的轻微变动将对其类别的判定影响很小,使中间指纹可以尽量地延续原始指纹的鲁棒性.

2) 类别的数量要在保证中间指纹具有较强的区分度的同时尽可能少,从而最大化地减少音频指纹的数据量.

**算法 1.** 中间指纹生成算法.

```
/* 指纹分类 */
①  $\forall$  subFingerprint
for  $i=1$  to  $M$  do
    subFingerprint( $i$ ) & 1;
```

```
Sort(bit( $M$ ));
/* 依据子指纹的  $M$  个二进制位分类 */
endfor
/* 统计指纹分布情况 */
②  $\forall$  originalFingerprint
for  $j=1$  to  $FpSize$   $j+=X$  do
/* 每次统计间隔  $X$  个子指纹 */
for  $k=j$  to  $Y+j$  do
    Distribution(subFingerprint[ $Y$ ]);
/* 统计连续  $Y$  个子指纹的类别分布 */
endfor
endfor
```

首先将原始指纹(Philips 指纹)的子指纹按照音频能量通常较高的低频区域对应的  $M$  个固定比特位的不同划分为  $2^M$  类.每向后移动  $X$  帧,对连续的  $Y$  帧( $Y>X$ )原始子指纹统计其在  $2^M$  个类中的分布情况,统计值作为中间指纹的子指纹.

参考原始指纹的提取算法,中间指纹的相邻 2 组指纹也存有  $1-X/Y$  的重叠部分,这样做可以延续原始指纹对随机截取检索的鲁棒性,重叠部分越多检索效果越准确,但检索效率越低.中间指纹对原始指纹的压缩比例在中间指纹具有最优区分度的前提下,比例越小过滤检索效率越高.中间子指纹间的相似度可以由绝对距离或欧氏距离判定.

3.3 基于阈值的固定间隔抽样匹配

本文所涉及的所有检索过程中音频指纹间的相似度匹配均采用滑窗法<sup>[14]</sup>获取,具体过程如图 3 所示:

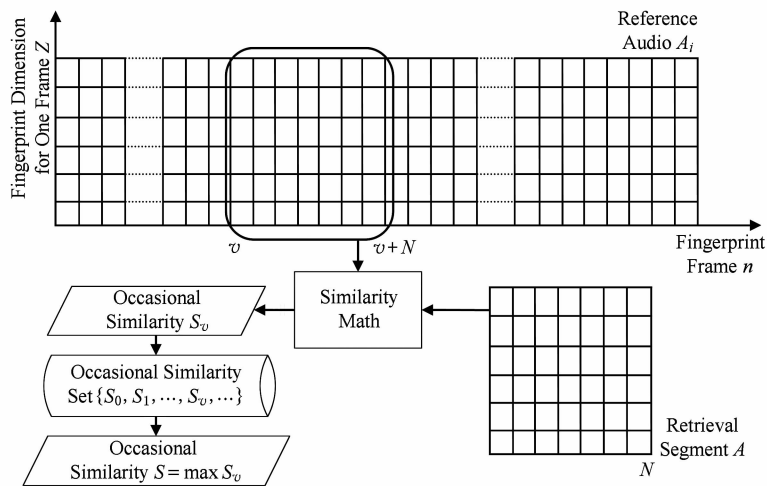


Fig. 3 Process of matching sliding window method in audio fingerprint.

图3 音频指纹滑动窗法匹配过程

对音频指纹库  $\{A_0, A_1, \dots, A_i, \dots\}$  中的参考音频指纹  $A_i$ , 沿帧数向后每滑动  $\delta$  帧, 计算其与帧数为  $N$  的检索音频片段  $A$  指纹之间的相似度, 获得一系列临时相似度值  $(S_0, S_1, \dots, S_v, S_{v+1}, \dots)$ , 最后将最大的临时相似度  $\max S_v$  定义为音频片段  $A$  和音频  $A_i$  之间的相似度  $S$ . 滑动帧数  $\delta$  越小, 相似度  $S$  越精确. 中间指纹单帧数据维数  $Z$  即为划分的类数  $2^M$ , 原始指纹的单帧数据维数  $Z$  即为 Philips 子指纹的比特数 32.

固定间隔抽样匹配是指在 2 个音频指纹的相似度匹配过程中, 每间隔固定的数量  $Q$  帧抽取 1 个子指纹进行相似度计算, 用所有抽取的对应子指纹间相似度代表 2 个指纹间的整体相似度  $S$ . 由于音频指纹是基于音频内容本身而提出的, 可以认为音频中几乎所有的子指纹在整个音频指纹中的地位是基本相当的, 任意子指纹在整体指纹相似度计算中的权重几乎是相同的, 因此可以通过采用固定间隔抽样匹配的方法减少音频指纹相似度匹配过程中的计算量, 提高匹配效率; 但同时也要注意, 间隔过大会导致在短音频片段检索过程中抽取的子指纹数量过少, 而导致子指纹相似度的平均值与指纹整体相似度  $S$  偏差过大, 影响检索精度.

另外, 参考序贯相似性检测算法<sup>[15]</sup>, 本文设计了一种基于阈值的相似度快速匹配方法, 对待检索音频  $A$  在参考音频库  $\{A_1, A_2, A_3, \dots, A_i, \dots\}$  中采用滑动窗法进行相似度过滤匹配, 具体如下:

1) 若音频片段  $A$  的中间指纹总帧数大于参考音频  $A_i$  的中间指纹总帧数, 则直接判定 2 段音频  $A$  与  $A_i$  为不相似.

2) 设定阈值  $\bar{\alpha}$  和  $\alpha$ , 使用中间指纹每次滑动匹配时, 若前  $N_i$  帧指纹的距离平均值  $\frac{1}{N_i} \sum_i \epsilon_i < \bar{\alpha}$  或某一对应帧的距离  $\epsilon_i < \alpha$  时, 直接判定音频  $A_i$  为可能结果.

3) 设定阈值  $\beta$ , 在过程 2) 中, 累计前  $m$  帧中间指纹的距离  $\epsilon_m$ , 若  $\epsilon_m > \beta$  或帧数  $m$  达到一定数量时, 直接判定该次匹配为不相似.

4) 设定阈值  $\gamma$ , 使用原始指纹每次滑动匹配时, 先对比前  $t$  帧指纹的相似度  $S_i$ , 只有当  $S_i > \gamma$  时, 计算整体指纹的相似度  $S_v$ ; 否则直接判定该次匹配为不相似.

5) 设定阈值  $\eta$ , 在过程 4) 中, 累计前  $n (n \ll t)$  帧原始指纹的比特误差  $\epsilon_n$ , 若  $\epsilon_n > \eta$ , 直接判定该次匹配为不相似.

6) 设定阈值  $\theta$ , 由于相邻指纹间具有极高的相似性, 因此当指纹间相似度低于  $\theta$  时, 可以适当向后多滑动几次, 再进行相似度匹配.

实验证明, 加入阈值后, 在略微降低过滤幅度的基础上, 大幅地减少了音频指纹检索过程中相似度匹配的计算量, 可以有效地提高过滤速度和检索速度. 与固定间隔抽样匹配方法相结合, 应用在中间指纹和原始指纹的过滤检索过程中进行快速相似判定, 可以达到更加高效的检索效果.

## 4 实验和结果分析

### 4.1 实验环境

#### 1) 软硬件环境

本文实验在配置为 4 核、16 GB 内存的普通 PC

机上进行,CPU 的频率为 3.40 GHz,操作系统为 64 位 Windows 7 系统,程序运行平台为 Visual Studio 2012.

2) 音频库

实验参考音频库采用 3 341 首网络中随机搜集的音频,大部分音频长度在 3~6 min. 指纹库则对以上音频提取的指纹重复加载 30 次达到规模为 100 230 条的音频指纹库. 测试检索音频均从音频库中随机选取. 为了提高音频指纹提取效率,实验在提取指纹前将音频预先转换为 8 000 采样率、单声道、16 b 的 wav 格式.

4.2 实验设计

实验主要验证本文提出的逐层过滤-提纯检索方法的高效性及鲁棒性,以及中间指纹和基于阈值的固定间隔抽样匹配方法对检索的影响. 为了简化相似度匹配计算量,中间指纹相似度使用绝对距离评价,原始指纹相似度由比特误差率评价. 对实验中所有的检索,当检索音频的帧数大于参考音频帧数时,均按不相似处理. 因此测试音频在检索前首先根据首尾的音频能量去除首尾的静音部分,尽可能确保检索音频的帧数小于等于待命中音频的帧数. 另外,Philips 音频指纹提取算法经验证当指纹比特误差率 BER 阈值为 0.35 时,虚警率只有  $3.6 \times 10^{-20}$ ,因此认为只要有召回结果即是准确的结果,即理论准确率近似为 100%,故而实验在验证检索方法对音频处理的鲁棒性时只统计召回率.

1) 检索方法

① Fibonacci Hashing 检索(Fibonacci Hashing retrieval, FHR):使用基于 Fibonacci Hashing 的快速检索方法.

② 中间指纹过滤检索(middle fingerprint filtering retrieval, MFR):在直接检索前加入中间指纹过滤过程.

③ 高效过滤-提纯检索(EFRR):在中间指纹过滤检索中加入基于阈值的固定间隔抽样匹配方法.

以上 3 种方法都是基于音频的 Philips 指纹特征的检索方法,具有可比性.

2) 检索速度对比实验

从音频库中随机选取 2 100 首,并从中随机裁剪出时长为 10 s,30 s,1 min,2 min,3 min,4 min,5 min 的音频片段各 300 首,分别进行 FHR,MFR 以及 EFRR. 统计对不同时长的音频片段、3 种检索方法需要的平均检索时间;对 MFR,EFRR 分别统计使用中间指纹进行过滤消耗的时间以及过滤的幅度.

3) 召回率对比实验

随机选择 1 500 首音频分别进行 MP3 转换、重采样及随机截取处理后再检索,统计 3 种检索方法经过不同处理后的音频的召回率.

4.3 实验分析

1) 检索时间

从表 1 中可以观察到,对相同的音乐片段,MFR 已经大幅地降低了检索的时间开销,比 FHR 算法速度提高了约 130 倍;将基于阈值的固定间隔抽样匹配方法加入到过滤检索体系中后,可以观察到检索速度在过滤检索的基础上得到了进一步的提高,对不同时长的音频片段,平均检索时间均小于 1 s,比 FHR 算法效率提高了 800 倍以上.

Table 1 Average Retrieval Time of Three Retrieval Methods with Different Audio Length

表 1 3 种检索对不同时长音频检索平均时间 s

| Retrieval<br>Audio Length | FHR       | MFR    | EFRR  |
|---------------------------|-----------|--------|-------|
| 10                        | 528.313   | 3.945  | 0.635 |
| 30                        | 1 875.431 | 14.325 | 0.966 |
| 60                        | 3 132.927 | 24.312 | 0.834 |
| 120                       | 3 772.114 | 29.505 | 0.586 |
| 180                       | 3 479.712 | 27.609 | 0.406 |
| 240                       | 1 513.097 | 12.164 | 0.172 |
| 300                       | 994.146   | 8.796  | 0.063 |

另外,FHR 和 MFR 的平均时间随检索音频时长的增加均明显呈现出向上凸起的抛物型曲线状,如图 4(a)和图 4(b)所示. 检索方法的计算量主要与 3 个因素相关:①对每首参考音频进行匹配时的滑动窗口次数  $H$ ;②每次滑动后需要匹配的长度  $L$ ;③参考音频库中比检索音频长的音频数量  $R$ . 检索计算量与  $H \times L \times R$  成正比,随着检索音频片段时长逐渐增加, $H$  和  $R$  会逐渐降低, $L$  会逐渐增长. 当检索音频片段较短时,因素  $L$  占主导地位,因素  $H$  和  $R$  的影响不够大,因此检索平均时间逐渐增加;当检索音频片段时长达到一定程度时,因素  $H$  和  $R$  的影响逐渐凸显,检索平均时间便开始逐渐降低. 而 EFRR 大体上随着检索音频片段的生长,速度提高效果越明显,如图 4(c)所示,最高能在过滤检索的基础上再提高约 140 倍,这是由于基于阈值的固定间隔抽样匹配方法致使因素  $L$  基本不受检索音频时长的影响,因素  $H$  和  $R$  在检索计算量中占主导地位;中间指纹和基于阈值的固定间隔抽样匹配方法对不同时长音频检索效率的提高均具有较大的影响.

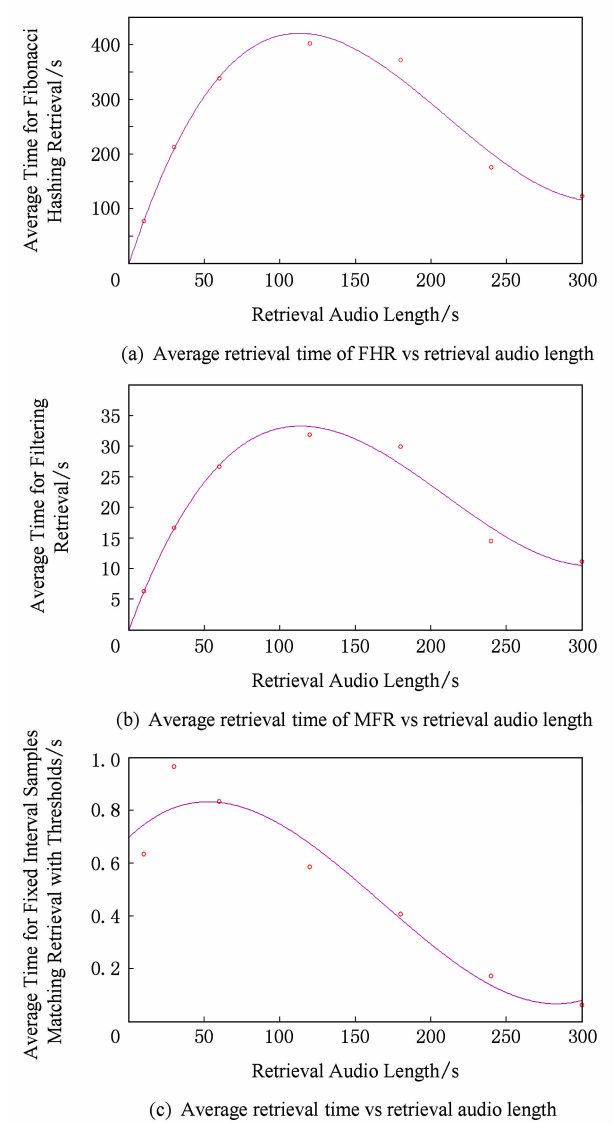


Fig. 4 Average retrieval time with different audio length in retrieval with three retrieval methods.  
图 4 3 种检索平均时间随检索音频时长变化趋势

由表 2 可以看出，中间指纹过滤需要的平均时间约为过滤检索平均时间的一半，过滤幅度均在

Table 2 Several Parameters for Filtering Retrieval with Middle Fingerprint  
表 2 中间指纹过滤检索各项参数

| Retrieval Audio Length/s | Average Filtering Time/s | Average Amplitude/% | Average Retrieval Time/s |
|--------------------------|--------------------------|---------------------|--------------------------|
| 10                       | 0.681                    | 99.498              | 3.945                    |
| 30                       | 5.985                    | 99.581              | 14.325                   |
| 60                       | 11.793                   | 99.648              | 24.312                   |
| 120                      | 16.892                   | 99.775              | 29.505                   |
| 180                      | 14.623                   | 99.837              | 27.609                   |
| 240                      | 7.962                    | 99.941              | 12.164                   |
| 300                      | 4.255                    | 99.947              | 8.796                    |

99.498% 以上。相对于 FHR，中间指纹可以用很少的时间大幅地过滤掉不相关的音频，快速地缩小了检索范围。表 3 将基于阈值的固定间隔抽样匹配方法应用到过滤检索中，虽然稍微降低了过滤的平均幅度，但过滤速度得到了显著提高。

Table 3 Parameters of EFRR  
表 3 高效过滤-提纯检索各项参数

| Retrieval Audio Length/s | Average Filtering Time/s | Average Amplitude/% | Average Retrieval Time/s |
|--------------------------|--------------------------|---------------------|--------------------------|
| 10                       | 0.356                    | 94.586              | 0.635                    |
| 30                       | 0.663                    | 92.386              | 0.966                    |
| 60                       | 0.613                    | 93.812              | 0.834                    |
| 120                      | 0.432                    | 95.857              | 0.586                    |
| 180                      | 0.286                    | 96.480              | 0.406                    |
| 240                      | 0.130                    | 99.451              | 0.172                    |
| 300                      | 0.051                    | 99.720              | 0.063                    |

2) 鲁棒性

由表 4 可以看出，MFR，EFRR 的召回率与 FHR 的召回率均在 99.47% 以上，说明本文提出的方法对 MP3 转换、重采样和随机截取均具有较高的鲁棒性。基于阈值的固定间隔抽样匹配方法由于经过多次阈值调整，对过滤检索的召回率影响不大。没有成功召回的音频经检查发现，存在静音过多或音频在所有频带的能量都很低的问题，这种情况会导致音频在处理指纹发生较大变化。

Table 4 Average Recall Rate for Different Audio Processing  
表 4 不同音频处理后检索的平均召回率 %

| Audio Processing | FHR   | MFR   | EFRR  |
|------------------|-------|-------|-------|
| MP3 Convert      | 99.87 | 99.73 | 99.73 |
| Resampling       | 99.87 | 99.47 | 99.47 |
| Shearing         | 100   | 99.87 | 99.87 |

5 结束语

经实验验证，本文所提出的高效过滤-提纯检索方法在保证召回率和准确率的情况下检索速度有显著提高，并对格式转换、重采样、随机剪切具备较好的鲁棒性；同时，本文设计的中间指纹和基于阈值的固定间隔抽样匹配方法对检索效率的提高均有较大的影响。对于含有过多静音或频谱能量很低的音频片段，检索效果较差。我们考虑寻找更好的原始指纹分类方法，以提高中间指纹的区分度或降低中间指纹的重叠率，进一步加快过滤检索过程。另外，由于

实际中对音频的操作处理极其丰富,因此仍需进行大量对抗不同的失真处理实验,验证检索方法在其他方面的鲁棒性.

参 考 文 献

- [1] Haitsma J, Kalker T. A highly robust audio fingerprinting system [C] //Proc of the 3rd Int Symp on Music Information Retrieval (ISMIR2002). Paris: Ircam-Centre Pompidou, 2002: 107-115

[2] Precioso F, Cord M, Gorisse D, et al. Efficient bag-of-features kernel representation for image similarity search [C] //Proc of the 18th IEEE Int Conf on Image Processing (ICIP2011). Piscataway, NJ: IEEE, 2011: 109-112

[3] Grosche P, Müller M, Serrà J. Audio content-based music retrieval [J]. Multimodal Music Processing, 2012, 3: 157-174

[4] Diamantaras K I, Kung S Y. Principal Component Neural Networks [M]. New York: Wiley, 1996

[5] Hu Yonggang, Wu Yi, Bu Jiang. Dimensionality reduction in audio fingerprint based on weighted PCA [J]. Computer Applications, 2006, 26(9): 2250-2254(in Chinese)  
(胡永刚, 吴翊, 卜江. 基于加权 PCA 的声音指纹降维技术 [J]. 计算机应用, 2006, 26(9): 2250-2254)

[6] Shen L, Guan Y, Wu Y, et al. Fast audio fingerprint search strategy for song identification [C] //Proc of the 2009 Int Conf on Networking and Digital Society (ICNDS 2009). Piscataway, NJ: IEEE, 2009, 2: 259-262

[7] Guibin Z, Miao L, Jiqing H, et al. A fast audio retrieval method based on negativity judgment [C] //Proc of the 5th Int Conf on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP2009). Piscataway, NJ: IEEE, 2009: 1156-1159

[8] Panagiotou V, Mitianoudis N. PCA summarization for audio song identification using Gaussian mixture models [C] //Proc of the 18th Int Conf on Digital Signal Processing (DSP 2013). Piscataway, NJ: IEEE, 2013: 1-6

[9] Haitsma J, Kalker T. An efficient database search strategy for audio fingerprinting [C] //Proc of the 2002 IEEE Workshop on Multimedia Signal Processing. Piscataway, NJ: IEEE, 2002: 178-181

[10] Chen M, Xiao Q, Matsumoto K, et al. A fast retrieval algorithm based on Fibonacci Hashing for audio fingerprinting systems [C] //Proc of the 2013 Int Conf on Advanced Information Engineering and Education Science (ICAIEES 2013). Paris: Atlantis Press, 2013: 219-222
- [11] Kurth F, Ribbrock A, Clausen M. Identification of highly distorted audio material for querying large scale databases [C] //Proc of the 112th Audio Engineering Society Convention(AES112). Los Angeles: AES, 2002: 1-8

[12] Kurth F, Muller M. Efficient index-based audio matching [J]. IEEE Trans on Audio, Speech, and Language Processing, 2008, 16(2): 382-395

[13] Vitola C P J, Sepulveda J, Martinez J I. Fast content-based audio retrieval algorithm [C] //Proc of the 18th Image, Signal Processing, and Artificial Vision (STSIVA2013). Piscataway, NJ: IEEE, 2013: 1-5

[14] Shibuya T, Abe M, Nishiguchi M. Audio fingerprinting robust against reverberation and noise based on quantification of sinusoidality [C] //Proc of 2013 IEEE Int Conf on Multimedia and Expo(ICME2013). Piscataway, NJ: IEEE, 2013: 1-6

[15] Barnea D I, Silverman H F. A class of algorithms for fast digital image registration [J]. IEEE Trans on Computers, 1972, 100(2): 179-186



**Zhang Xingzhong**, born in 1964. MSc and associate professor in Taiyuan University of Technology. His main research interests include pattern recognition and database management.



**Wang Yunsheng**, born in 1987. MSc candidate. His research interests include audio retrieval based on content, rights-managed digital audio(wawa\_yun\_sheng@163.com).



**Zeng Zhi**, born in 1981. PhD, associate professor. His research interests include multimedia content management, pattern recognition(zhi, zeng@ia, ac. cn).



**Niu Baoning**, born in 1964. PhD and professor in Taiyuan University of Technology. His main research interests include autonomic computing and performance management of database system, data management in cloud computing environment, data management and knowledge engineering and pattern recognition(niubaoning@tyut. edu. cn).