一种基于倒排索引的音频检索方法

张雪源* 贺前华 李艳雄 叶婉玲
(华南理工大学电子与信息学院 广州 510640)

摘 要：传统的基于实例的音频检索算法采用顺序索引，检索时需遍历数据库并导致难以忍受的等待时间。针对传统的顺序的索引方法，该文提出基于倒排索引的音频检索算法。该方法首先利用多种音频特征构成的超向量，通过多层音频分割方法将连续音频流分割为特征数值波动幅度小的短时音频段；然后利用事先训练好的音频字典，将短时音频段序列转换为可以表征音频内容的音频字序列，并建立倒排索引；检索时，将用户提交的查询转换为音频字后利用倒排索引无须遍历数据库即可直接定位候选段落，并根据候选段落与查询的内容相似度大小对候选段落进行排序，将排好序的列表作为检索结果。仿真实验以匹配项排名、同类检索结果比例、定位准确性和检索用时 4 个方面作为评价指标，实验结果显示，该算法能够在平均 1.101 s 时间内实现 92.58%的检索准确率。

关键词：音频信号处理；音频检索；内容相似度；倒排索引
中图分类号：TN912.3 **文献标识码**： A **文章编号**：1009-5896(2012)11-2561-07
DOI: 10.3724/SP.J.1146.2012.00510

An Inverted Index Based Audio Retrieval Method

Zhang Xue-yuan He Qian-hua Li Yan-xiong Ye Wan-ling
(*School of Electronic and Information Engineering*, *South China University of Technology*, *Guangzhou* 510640, *China*)

Abstract: Traditional example based audio retrieval algorithms use forward index, with which, retrieval processing need to traverse the whole database, resulting in intolerable response time. This paper proposes an inverted-index based audio retrieval method. Through constructing super-vector comprising several audio features, audio stream is first segmented into short segments with small feature fluctuation; Based on a pre-trained audio word dictionary, short audio segment sequence is then transformed into audio word sequence, from which inverted index is constructed; During the retrieval phase, the query audio sample is transformed into audio words and retrieval is carried out, candidate segments are ranked according to the similarity with the query. Match term ranking, same type ratio, overlap ratio and retrieval time are used to evaluate the performance of the proposed algorithm. The experiment gives 92.58% retrieval precision within average response time of 1.101 s.

Key words: Audio signal processing; Audio retrieval; Content similarity; Inverted index

1 引言

随着多媒体信息爆炸式的增长，如何从海量的、未经手工标注的多媒体数据库中快速、准确地搜索到最理想的内容逐渐成为当前学术研究的热点。音频信息作为多媒体数据的重要组成部分，有时甚至是唯一的形式(如广播、音乐和电话录音等)，其索引和检索算法受到了越来越多的关注[1]。

音频的检索方法可分为两种：一种基于语义，另一种基于相似度。基于语义的检索方法利用先验模型对音频数据进行语义索引，其索引内容多为音效类型、音频场景类型[2,3]。检索时依据用户提交的音效语义描述、场景语义描述进行基于文本的检索。这种方法实现了多媒体内容的自动标注和检索，并且依靠语音识别系统实现了语音文档检索。但由于音频内容十分复杂、多变，文字在描述音频内容上不够精细，如“枪声”这一音效类型忽略了枪声的强弱、密集程度以及枪的类型等重要信息，因此难以跨越的“语义鸿沟”严重影响了这种方法的效果。此外，这一类方法需要预先定义音效、场景类型，检索时只能从事先定义好的类型中选择检索条目。近年来越来越多的研究开始关注基于相似度的检索算法[4,5]。该方法直接利用音频检索音频，不需要进行音效、场景识别，也不需要提前定义和训练模型。用户提交一段现有的或者自己制作的音频作为查询，检索系统从数据库中寻找与之最为相似的片段。该方法将数据库的音频流切割成片段作为候选，检

2012-05-02 收到，2012-07-09 改回
国家自然科学基金(60972132, 61101160)资助课题
*通信作者：张雪源 zhang.xueyuan@mail.scut.edu.cn

索时采用遍历的方法从头至尾依次计算查询音频和候选音频之间的距离。由于采用遍历策略，并且音频段之间距离计算的计算量较大，此类方法在面对海量数据库的检索时的等待时间难以忍受。此外，在对音频切割后，每一个候选片段便已经确定，无法检索到候选片段的一部分或者由连续多个候选片段组成的段落。文献[6]提出了一种快速浏览检索算法，该算法以直方图描述经矢量量化后的特征分布，并以动态步长跳过相似度低的部分从而加速检索。虽然该方法在速度上有较大提升，但其本质仍为顺序索引，并且没有很好地利用音频时序信息。

遍历式检索方法的计算量与数据库大小成线性关系，对于动辄上千小时的电影数据库，遍历式的检索在响应时间上难以满足要求。倒排索引由于其出色的表现被广泛应用在文本检索中。目前，倒排索引在音频检索领域的应用主要集中在语音文档检索[7]，依赖语音识别器进行语音识别后建立文字索引。此外，文献[8]将其应用在音乐检索上，文献[9]将其应用在了鸟叫声音的检索上，目前尚未见文献应用于一般音频数据的索引和检索。本文利用文本搜索引擎中的倒排索引方法，为音频建立音频字典和倒排索引，提出基于倒排索引的音频检索方法。本文将连续音频流转换为音频字序列并建立倒排索引文档。在检索时利用倒排索引实现候选段落的直接定位，利用基于音频字相似度的距离公式对候选段落进行排序。

2 倒排索引

文本检索中，顺序的检索意味着遍历每一篇文章的每一个字并判断是否与查询关键字相匹配。当文章数量极大时，这种做法的效率极低。由于用户等待时间、处理器、内存上的限制，该方法不适用于现代大规模数据库的搜索。

倒排索引是现代搜索引擎最常用的索引方式[10]。所谓“倒排”与顺序的索引不同，它将文档-词信息流转换为词项-文档信息[11]。该方法首先建立一个由索引项构成的索引项词表，索引项是描述数据库文档的最小单元，可以是单词、字甚至音素(常见于基于网格的语音文档检索中)。之后，需要将每个索引项在文档中的位置添加到倒排索引中。如以下 3 个文档：文档 1：书本知识；文档 2：书中的知识；文档 3：这本书很好。

倒排索引表由两部分构成，即索引项列表和每个索引项自身的事件表，事件表中的每一项均是一个指针，指向了含有该索引项的内容在文档中的具体位置，每一个位置 (a,b) 中 a 代表文档编号，b 代表该索引项在文档中的具体位置。文档 1～文档 3 的倒排索引如表 1 所示。

表 1 倒排索引

索引项	事件表
书	(1,1),(2,1),(3,3)
本	(1,2),(3,2)
知	(1,3),(2,4)
识	(1,4),(2,5)
中	(2,2)
的	(2,3)
这	(3,1)
很	(3,4)
好	(3,5)

当用户提交一个查询时，如查询为“书”，通过查询索引项列表及该索引项对应的事件表，可以快速定位所有该索引项出现的位置，即{(1,1),(2,1),(3,3)}。当用户提交的查询为“书本”这样的短语时，首先依次确定各个索引项的位置：

“书”：{(1,1),(2,1), (3,3)}

“本”：{(1,2), (3,2)}

其次确定所有索引项都命中的文档，即对各索引项所命中的文档编号取交集，$\{1,2,3\}\cap\{1,3\}=\{1,3\}$，即第 1 个文档和第 3 个文档命中所有索引项，但只有第 1 个文档与查询的索引项顺序一致，因此第一返回结果应为文档 1 的位置 1～位置 2。

3 基于倒排索引的音频检索方法

搜索引擎必须具备两种功能：索引处理和查询处理。索引处理的目的是建立可快速查找的数据结构，查询处理根据这些数据结构和用户提交的查询生成排好序的检索结果。

索引阶段，对音频流提取特征并进行分割，将持续若干小时的连续音频分成短时音频片段。之后根据事先训练好的音频字典，将短时音频段序列转换为音频字序列。每一个音频字都是这一段音频的集中表示，可以看作文本索引中的“关键字”。之后为音频字建立倒排索引。检索阶段，对用户提交的查询同样进行特征提取、分割并转换为音频字，查找倒排索引后，将查找结果按照和查询的相似度由高到低排序并返回给用户。其过程如图 1 所示。

3.1 特征提取

为了能充分反映各种类型音频的特性，本文的研究中使用多种特征构成的超向量作为音频特征，包括短时能量、短时过零率等时域特征，谱能量、子带能量比、谱质心、谱带宽等频域特征，能反映

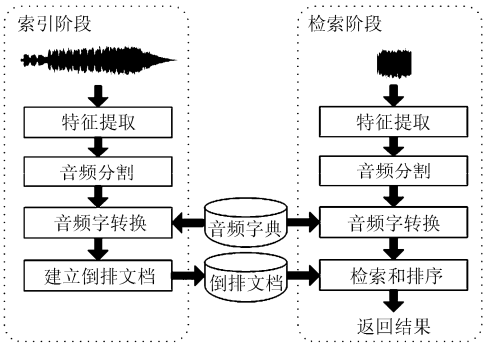

图 1 音频索引和检索系统方法

人耳听觉效果的梅尔(Mel)频率倒谱特征[12]。每种音频特征均描述了音频信号某一方面特性，为了充分表征音频内容，本文将该 15 维超向量作为特征以充分表征语音、音乐和一般音频等各类音频信号的特性。由于各维特征数值范围上有很大不同，如子带能量比数值位于区间[0,1]，而谱质心的数值可轻易达到 1000 以上，在后续处理中，例如计算欧氏距离时，数值范围大的特征会轻易覆盖掉其他特征的影响。因此需要对各维特征进行归一化。事先收集各种音频数据，包括电影、电视剧、电视访谈节目、体育比赛、广播、颁奖典礼、音乐会等类型数据共 210 h，提取特征后分别计算各维特征的均值和标准差当作规整向量。之后对其他数据提取特征时，用该规整向量按式(1)对特征进行规整。

$$f_d' = \frac{f_d - \mu_d}{\sigma_d}, \quad d = 1,\cdots,D \tag{1}$$

其中 D 为特征总维数，f_d 和 f_d' 分别为原始特征和规整后的特征，μ_d 和 σ_d 分别为规整用的均值和标准差。该方法会将各维特征规整到均值为 0，方差为 1 的分布中。

3.2 音频分割

分词是文本索引中的重要模块，与之类似，连续音频流的自动分割是音频索引的首要步骤。音频分割的目的是检测出连续音频流中的声学特征变化点，从而把音频流切分成具有某种相同性质(如同一说话人、同一信道或者同一音效类型)的音频片段。为了对连续音频流用音频字进行表示，本文将音频数据切割成声学特征变化较小的短时音频段。由于多媒体数据动辄持续 1 h 以上(如电影或者音乐会)，有的甚至始终不停止(如 24 h 播放的广播或者新闻)。采用计算十分准确的分割方法会造成巨大的计算量需求，采用计算量低的方法分割效果又不令人满意。本文提出一种 3 阶段的自顶向下多层分割方法，使用由粗到细的多层分割策略，将长达数小时的连续音频流分割成短时音频段：第 1 阶段利用能量对音频进行粗略的分割；第 2 阶段利用 T^2 距离将音频分成音频类型一致的片段；第 3 阶段利用声学特征一阶、二阶统计量保证每个短时音频段内的声学特征数值处于较小的变化范围内。

在多媒体数据中，静音是最明显的分割点，当超过连续 2 s 的音频短时帧能量均小于能量门限则认为该音频片段为静音段。由于不同音频文档其能量浮动范围很大，各文档利用能量值自动确定能量门限。$E_{\max}$，$E_{\min}$ 和 E_{mean} 分别代表当前音频文档短时帧能量的最大值、最小值和均值，E_{range} 表示能量的浮动范围。能量门限 E_{th} 应当在 $E_{\min}$ 和 $E_{\min}+E_{\mathrm{range}}$ 之间，以静音因子 λ_s ($\lambda_s \in [0,1]$)决定能量门限具体数值，计算方法为

$$\begin{aligned} E_{\mathrm{th}} &= \lambda_s \cdot E_{\mathrm{range}} + E_{\min} \\ &= \lambda_s \cdot \min\left\{\frac{1}{2}(E_{\max} - E_{\min}), E_{\mathrm{mean}} - E_{\min}\right\} + E_{\min} \end{aligned} \tag{2}$$

λ_s 的取值由实验确定，为涵盖尽可能广的音频类型，实验数据选用 3.1 节中用于计算规整向量的 210 h 音频，实验结果显示 λ_s 取 0.1 时有最好的分割效果。

在静音分割的基础上，继续将音频分割成音频类型相同的段落。本文利用逐渐增长的分析窗，依次计算分析窗内部各测试点左右两边数据窗之间的 Hotelling's T^2 距离[13]，将超过预设门限的峰值位置当作改变点，距离公式计算如下：

$$T^2 = \frac{b(N-b)}{N}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{3}$$

其中 N 和 $\boldsymbol{\Sigma}$ 分别为分析窗总长度和数据的协方差矩阵，b 和 $\boldsymbol{\mu}_1$ 分别为测试点左侧数据窗长度和特征均值，$\boldsymbol{\mu}_2$ 为右侧数据窗特征均值。设定分析窗初始长度为 3 s，如果窗内未发现音频类型改变点，则窗长增加 1 s，如果窗内找到改变点则将分析窗长重置为初始长度，并以该改变点位置作为起点继续搜索下一改变点直至搜索至数据尾端。

利用 T^2 公式可以将音频数据分割为单一音频类型段，下一步根据音频特征的均值和方差进行分割。首先初始化长度为 30 帧的分析窗，计算其协方差矩阵为 $\boldsymbol{\Sigma}$，以中点分割分析窗得到分别为 15 帧的左右两个数据窗，其特征均值分别记为 $\boldsymbol{\mu}_1$ 和 $\boldsymbol{\mu}_2$，D 为特征维数，计算两数据窗均值的欧氏距离为

$$\mathrm{dis}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sqrt{\sum_{d=1}^{D} \left|\boldsymbol{\mu}_1(d) - \boldsymbol{\mu}_2(d)\right|^2} \tag{4}$$

假设特征各维独立，协方差矩阵 $\boldsymbol{\Sigma}$ 简化为对角阵，则

$$|\boldsymbol{\Sigma}| = \prod_{d=1}^{D} \sigma_{d,d}^2 \tag{5}$$

当左右两数据窗均值距离或者方差超过预设门

限时，均认为分析窗内部存在较大的数据变化，则当前分割点即为改变点。如果均值距离和方差均没有超过门限，则将左侧数据窗向后增加 5 帧，右侧数据窗向后平移 5 帧，继续计算直至找到改变点或者搜索至数据尾端。

经过上述 3 层分割，可以将任意长度的连续音频流准确而有效地分割为音频特征数值波动幅度较小的短时段落。由于前两分割阶段分别采用能量极小值点和音效类型改变点进行分割，只有第 3 阶段依赖均值和方差的统计，因此，当音频数据起点略有偏移时，第 3 分割阶段所造成的分割偏差累积主要存在于音频的起始和结尾部分，而对音频中间主体部分的分割影响较小，而音频的主要内容一般存在于中间主体部分，因此该分割方法具有起点鲁棒性。

3.3 建立音频字典

在文本检索中，关键字或者关键词是文档的索引项，索引项的集合称为索引项词表。与之类似，我们定义音频中划分单元内容的表示为“音频字”，所有音频字的集合构成了“音频字典”。本节介绍如何构建音频字和音频字典。

音频字的个数将影响检索的效果：如果音频字个数过多，极限情况是将每一帧特征都视为一个音频字，一方面会产生尺寸十分巨大的音频字典，另一方面也会将十分相似的音频内容判别为不同的音频字，造成十分相似的音频却有完全不同的音频字序列；反之，音频字个数过少会导致音频字对音频内容的区分能力下降，产生过多音频字序列完全一致的段落。

本文依据音效类型(语音、枪声、爆炸声、欢呼声等)的个数确定音频字典大小。文献[14]根据音源的不同将音效分成了 400 种类型。为了避免将不同音效判别为同一音频字，每一种音效至少需要一个音频字表征。同时，每一种音效由于存在多种变化形式，因此本文对每一种音效使用 3 个音频字进行表征。训练音频字典时，首先为每一种音效收集 100 个样本，进行特征提取后，将该音效的所有特征使用 k-均值聚类算法[15]聚成 3 个类，所有音效总共产生 1200 个类中心。每一个聚类中心作为一个音频字 w_i，由此产生的音频字典为 $W=\{w_1,w_2,\cdots,w_K\}$，$K=1200$。

3.4 音频索引

本节讨论如何利用音频字典将连续音频数据转换为音频字序列并建立倒排索引。

对包含 N 个音频文档的集合 $\{D_1,D_2,\cdots,D_N\}$ 中某个音频文档 D_i，利用 3.2 节的方法将其划分为若干音频片段，记为 $D_i=\{\boldsymbol{d}_1^i,\boldsymbol{d}_2^i,\cdots,\boldsymbol{d}_{N_i}^i\}$。每一个片段都是由若干特征构成的矩阵 $\boldsymbol{d}_j^i=[f_1^{i,j},f_2^{i,j},\cdots,f_{N_{i,j}}^{i,j}]$，$f_k^{i,j}$ 表示第 i 个音频文档第 j 个音频片段的第 k 帧特征。3.2 节的音频分割保证了音频片段的特征在一阶、二阶统计量上均没有显著变化，因此用均值代替音频片段以减少数据冗余，

$$m_j^i=\frac{1}{N_{i,j}}\sum_{k=1}^{N_{i,j}}f_k^{i,j} \tag{6}$$

对所有文档的所有片段进行取均值处理，得到音频文档库的均值表示 $\{M_1,M_2,\cdots,M_N\}$，其中 $M_i=\{m_1^i,m_2^i,\cdots,m_{N_i}^i\}$，$m_j^i$ 表示第 i 个音频文档中的第 j 个均值特征。利用 3.3 节生成的音频字典 $W=\{w_1,w_2,\cdots,w_K\}$，将音频特征均值 m_j^i 转换为音频字 id_j^i：

$$id_j^i=\arg\min_{1\le k\le K}\operatorname{dis}\left(m_j^i,w_k\right) \tag{7}$$

$$\operatorname{dis}\left(m_j^i,w_k\right)=\sqrt{\sum_{d=1}^{D}\left|m_j^i(d)-w_k(d)\right|^2} \tag{8}$$

对所有文档进行音频字转换，得到 $\{ID_1,ID_2,\cdots,ID_N\}$，每个文档的音频字序列为 $ID_i=\{id_1^i,id_2^i,\cdots,id_{N_i}^i\}$。一段音频的音频字序列示意图如图 2 所示，$(i,j)$ 表示文档 i 的第 j 个位置，w_k 表示某音频字，阴影表示该音频字出现在该位置。

之后为文档建立倒排索引，倒排索引由两部分构成，一部分是由所有音频字构成的列表，该列表即为音频字典 W，另一部分称为事件表。每一个音频字都对应了事件表中的一行，事件表中的每一项都记录了该音频字在音频文档中出现的位置。当要

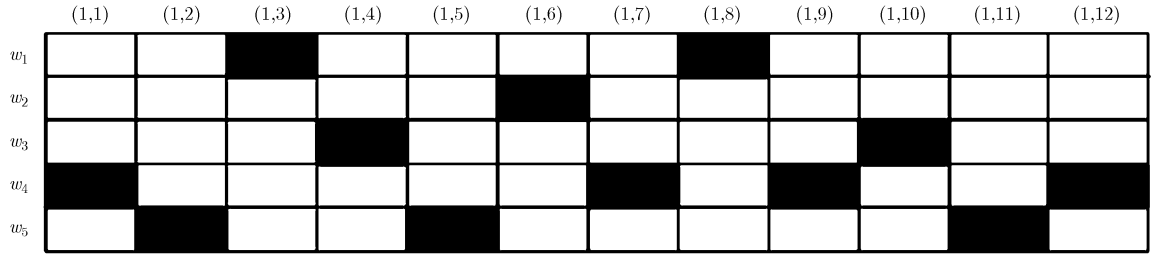

图 2 音频字序列示意图

添加音频字序列 $ID_i=\{id_1^i,id_2^i,\cdots,id_{N_i}^i\}$ 到倒排索引中时，在和 id_j^i 一致的音频字 w_k (即 $id_j^i=k$)所对应的事件表末端，添加 (i,j)。图 2 所示的序列生成的倒排索引为如表 2 所示。

表 2 音频倒排索引表

音频字	事件表
w_1	(1,3),(1,8)
w_2	(1,6)
w_3	(1,4),(1,10)
w_4	(1,1),(1,7),(1,9),(1,12)
w_5	(1,2),(1,5),(1,11)

3.5 音频检索

在文本检索中，检索的任务是接受用户的查询并利用检索、排序算法返回给用户一个排好序的文档列表，排名越靠前的结果与用户提交查询的相似度应当越高。

在基于内容相似度的音频检索中，用户提交的查询是一段音频。检索过程如下：首先利用和索引阶段一样的特征提取、音频分割方法将音频数据切分成音频段，并且根据音频字典将其转换为音频字序列，表示为 $Q=\{q_1,q_2,\cdots,q_{N_Q}\}$；接下来检索系统利用查询 Q 中所有音频字进行“命中”检测，并确定所有候选段落的起止点；最后将所有候选段落依据与查询的相似度进行排序并将结果返回给用户。

所谓“命中”是指存在某音频字既出现在查询中也出现在音频文档中。假设查询 Q 中 q_n 对应的音频字为 w_k，则 q_n 命中的位置为：倒排索引表中 w_k 对应事件表里的所有位置。对 q_n 命中的任意一个位置 (i,j)，可确定一个候选段落：文档 i 中以第 $j-(n-1)$ 个音频字为起点，第 $j+(N_Q-n)$ 个音频字为终点的段落，其长度和查询一致，为 N_Q。

在检测出所有候选段落后需要对其依据与查询的相似度进行排序。查询和候选段落越相似则两者应当有越多时序一致的音频字，因此以命中音频字在每一个候选段落中所占的比例作为相似度。某候选段落 $C=\{c_1,c_2,\cdots,c_{N_Q}\}$ 和查询 Q 的相似度通过计算命中音频字的个数比例求得，即

$$R(Q,C)=\frac{\sum_{n=1}^{N_Q}\mathrm{hit}(Q_n,C_n)}{N_Q} \tag{9}$$

$$\mathrm{hit}(Q_n,C_n)=\begin{cases}1, & Q_n=C_n\\ 0, & Q_n\neq C_n\end{cases} \tag{10}$$

计算查询和所有候选文档的相似度，并按照由大到小的顺序排列，将排好序的结果返回给用户，完成检索过程。

4 实验仿真

为验证本文方法，使用服务器(Intel(R) Xeon(R) CPU, E5606@2.13 GHz 2.13 GHz (2 处理器，共 8 核)，32.0 GB 内存，64 位 Windows7 操作系统)进行实验仿真，以 Matlab 2010 为仿真平台。实验数据为来自互联网的 187 部电影，总时长 200 h，依据 IMDb 分类标准[16]，各类电影数量见表 3。

表 3 各类电影数量分布

电影类型	战争片	动作片	灾难片	纪录片	音乐剧	情景喜剧
数量(部)	15	19	12	38	23	80
总时长(h)	40.6	33.9	27.2	35.6	32.9	29.8

4.1 评价指标

从数据库音频中随机截取不同长度的音频段落作为查询，记录下该查询音频在数据库中的音频文档编号及其在文档中的段落位置(起点和终点)作为标签，该标签用来评价检索算法的性能。检索系统的返回结果列表中每一个结果项均为一个和查询时长一样的音频片段，当该结果项所属的音频文档编号和查询一致，并且该结果项在该音频文档中的段落位置和查询标签所标示的段落位置重叠超过 50% 时，认为该结果项为一个“匹配项”。返回结果列表中匹配项排名越靠前意味着检索算法性能越好。

本文以以下 4 项作为检索性能评价指标，与文献[6]提出的经典快速顺序搜索方法进行比较：

(1)在检索结果列表中，匹配项的排名分布。本文分别统计第 1 返回结果和前 10 名返回结果中含有匹配项的查询数占总查询数的比例，分别记为 PT1(Precision in Top 1)和 PT10(Precision in Top 10)；

(2)前 10 名检索结果中和查询属于同一类型电影的平均个数 STR(Same Type Ratio)。该项指标的值越高，则前 10 名返回结果中和查询属于同类电影的结果越多，和查询无关的结果越少；

(3)匹配项和查询在时间上的平均重叠比例 OR (Overlap Ratio)；

(4)平均检索用时 RT(Retrieval Time)。

4.2 实验结果及分析

本文分别以 5 s, 10 s, 15 s, 20 s 作为查询长度 QL(Query Length)进行 4 次实验，以验证不同长度的查询对检索性能的影响，每次实验均提交 21615 次查询。表 4 显示了本文提出的方法和文献[6]中的方法的实验结果。

表 4 检索性能比较

QL(s)	所用方法	PT1(%)	PT10(%)	STR	OR(%)	RT (s)
5	本文方法	76.02	83.95	0.814	90.7	0.532
	文献[6]的方法	75.04	84.01	0.782	93.1	51.196
10	本文方法	83.70	86.85	0.915	92.6	0.731
	文献[6]的方法	81.35	86.77	0.853	93.5	53.814
15	本文方法	88.41	90.12	0.936	93.9	0.947
	文献[6]的方法	85.20	89.79	0.884	93.4	58.905
20	本文方法	92.58	95.59	0.945	93.8	1.101
	文献[6]的方法	88.47	95.22	0.897	93.7	60.270

由实验结果可以看出，PT1 方面，本文的方法和传统方法都随着查询长度的增加而有更好的查询效果；任一查询长度，本文提出的方法均优于传统方法，相对提升比例分别为 1.31%, 2.89%, 3.77%和 4.65%。该统计说明，随着查询长度的增长，本文方法在性能上的提高越来越明显。该指标性能提升的原因在于本文使用的相似度测度考虑了时序上的匹配，只有当查询和候选段落在内容和时序上均相似时才能得到很高的相似度值。在 PT10 方面，本文的方法和传统方法性能相当，说明两种方法在前 10 名的返回结果中搜索到匹配项的能力相当。在 STR 方面，本文的方法较传统方法均有一定的提升，即本文所产生的返回结果列表中，和查询“相关”的结果更多，其原因在于本文所使用的音频字典，能够将相似的音频内容划分到一个音频字上，即能够很好地检测到相似的音频内容。在 OR 方面，本文方法和传统方法性能相当，均能保证平均在 90%以上的重叠比例，即偏移小于 10%。在时间上，由于本文使用了倒排文档作为索引结构，检索用时大大减少，分别为原来用时的 1.04%, 1.36%, 1.61%和 1.83%，速度提升均在 50 倍以上。

5 结束语

本文针对音频实例检索提出了基于倒排索引的检索算法。通过音频字转换可以将连续音频流转换为音频字序列，通过建立倒排文档可以实现候选段落直接定位，通过融合时序信息的相似度计算公式可实现候选段落排序。实验结果显示了该算法的准确性和有效性。

参 考 文 献

[1] Heryanto H, Akbar S, and Sitohang B. Direct access in content-based audio information retrieval: a state of the art and challenges[C]. 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, July 17–19, 2011: 1–6.

[2] Ghoraani B and Krishnan S. Time-frequency matrix feature extraction and classification of environmental audio signals[J]. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011, 19(7): 2197–2209.

[3] Fu Zhou-yu, Lu Guo-jun, Ting Kai-ming, *et al*.. Music classification via the bag-of-features approach[J]. *Pattern Recognition Letters*, 2011, 32(14): 1768–1777.

[4] Su Ja-hwung, Wu Cheng-we, Fu Shao-yu, *et al*.. Empirical analysis of content-based music retrieval for music identification[C]. 2011 International Conference on Multimedia Technology, Hangzhou, China, July 26–28, 2011: 3516–3519.

[5] Jurkas P, Stefina M, Novak D, *et al*.. Audio similarity retrieval engine[C]. Proceedings of the Third International Conference on Similarity Search and Applications, Istanbul, Turkey, Sep. 18–19, 2010: 121–122.

[6] Kashino K, Kurozumi T, and Murase H. A quick search method for audio and video signals based on histogram pruning[J]. *IEEE Transactions on Multimedia*, 2003, 5(3): 348–357.

[7] Matthews B, Chaudhari U, and Ramabhadran B. Fast audio search using vector space modeling[C]. IEEE Workshop on Automatic Speech Recognition & Understanding, Kyoto, Japan, Dec. 9–13, 2007: 641–646.

[8] Cha Guang-ho. An effective and efficient indexing scheme for audio fingerprinting[C]. 5th FTRA International Conference on Multimedia and Ubiquitous Engineering, Loutraki, Greece, June 28–30, 2011: 48–52.

[9] Bardeli R. Similarity search in animal sound databases[J]. *IEEE Transactions on Multimedia*, 2009, 11(1): 68–76.

[10] 黄少林, 王华, 张玉红, 等. 基于Lucene的索引系统的设计与实现[J]. 现代情报, 2009, 29(7): 169–171.
Huang Shao-lin, Wang Hua, Zhang Yu-hong, *et al*.. The design and implementation of indexing system based on lucene[J]. *Journal of Modern Information*, 2009, 29(7): 169–171.

[11] Bruce C, Donald M, and Trevor S. Search Engines: Information Retrieval in Practice[M]. Upper Saddle River: Addison-Wesley, 2010: 22–23.

[12] 韩纪庆, 冯涛, 郑贵滨, 等. 音频信息处理技术[M]. 北京: 清华大学出版社, 2007: 32–46.
Han Ji-qing, Feng Tao, Zheng Gui-bin, *et al*.. Audio Information Processing Technology[M]. Beijing: Tsinghua University Press, 2007: 32–46.

[13] Zhou Bo-wen and Hansen J H L. Efficient audio stream segmentation via the combined T2 statistic and Bayesian information criterion[J]. *IEEE Transactions on Speech and Audio Processing*, 2005, 13(4): 467–474.

[14] 刘成太. 音效分类数据库[OL]. http://www.yinxiao.net/, 2012-04-20.

[15] Lloyd S. Least squares quantization in PCM[J]. *IEEE Transactions on Information Theory*, 1982, 28(2): 129–137.

[16] 亚马逊公司. 在线影视数据库[OL]. http://www.imdb.com, 2012-04-20.

张雪源：　男，1987 年生，博士生，研究方向为音频信息检索.

贺前华：　男，1965 年生，教授，博士生导师，研究方向为语音及音频信号处理、嵌入式系统开发.

李艳雄：　男，1980 年生，讲师，研究方向为语音及音频信号处理.