根据最终权利要求书

使用图1

1、一种音频子指纹的检索方法，其特征在于，包括：

提取需要检索的音频子指纹；

根据所述音频子指纹，在预先生成的索引表中读取对应所述音频子指纹的存储单元中存储的所述音频子指纹的倒排项地址；

根据所述倒排项地址，在预先生成的倒排表中，获得所述倒排项地址对应的存储单元中存储的所有音频ID和偏移量；

根据所述音频ID和偏移量，通过预先生成的音频ID-音频指纹库地址映射表，获得各个音频ID对应的音频指纹文件在所述音频指纹库中的音频指纹文件存放地址；

根据各个所述音频指纹文件存放地址以及对应的所述偏移量，在音频指纹库中获得所述音频子指纹在音频指纹库中的存放位置。

2、根据权利要求1所述的音频子指纹的检索方法，其特征在于，所述预先生成的索引表中，对应所述音频子指纹的存储单元采用如下方式确定：将所述音频子指纹右移一位，所获得的数值就作为对应所述音频子指纹的子指纹key值；将所述子指纹key值作为存储地址，将其指向的所述存储单元作为对应所述音频子指纹的存储单元。

3、根据权利要求2所述的音频子指纹的检索方法，其特征在于，采用所述音频子指纹的所述子指纹key值，通过完美哈希算法投射到所述倒排表的倒排项地址空间，将所述完美哈希算法获得的地址作为所述音频子指纹对应的倒排项首地址。

4、根据权利要求2所述的音频子指纹的检索方法，其特征在于，所述预先生成的索引表采用三级索引表结构构造。

5、根据权利要求4所述的音频子指纹的检索方法，其特征在于，所述音频子指纹为长度32比特的二进制数；将所述音频子指纹右移一位获得31位的子指纹key值；在此基础上，所述预先生成的索引表采用三级索引表结构构造的方法包括：

以10位二进制比特位为第一key值，建立一级索引表，在每个第一key值为地址的存储位中，对应存储64比特位的第一地址数据；

以25位二进制比特位为第二key值，建立二级索引表，在每个第二key值为地址的存储位中，对应存储32比特位的第二地址数据；

以31位二进制比特位为第三key值，建立三级索引表，在每个第三key值为地址的存储位中，对应存储16比特位的第三地址数据；

将第一地址数据+第二地址数据+第三地址数据加和，形成完整的地址数据，该地址数据用于作为所述音频子指纹的倒排项地址；

所述在预先生成的索引表中读取对应所述音频子指纹的存储单元中存储的所述音频子指纹的倒排项地址，其读取方式如下：

根据所述子指纹key值的前10位，读取所述一级索引表相应的第一key值的存储位中存储的第一地址数据；

根据所述子指纹key值的前25位，读取所述二级索引表相应的第二key值的存储位中存储的第二地址数据；

根据所述子指纹key值的全部31位，读取所述三级索引表相应的第三key值的存储位中存储的第三地址数据；

根据所述第一地址数据+第二地址数据+第三地址数据，获得所述音频子指纹对应的倒排项地址。

6、一种音频子指纹的检索装置，其特征在于，包括：

音频子指纹提取单元，用于提取需要检索的音频子指纹；

倒排项地址获取单元，用于接收所述音频子指纹，并根据所述音频子指纹，在预先生成的索引表中读取对应所述音频子指纹的存储单元中存储的所述音频子指纹的倒排项地址；

倒排表查询单元，用于接收所述倒排项地址，并根据所述倒排项地址，在预先生成的倒排表中，获得所述倒排项地址对应的存储单元中存储的所有音频ID和偏移量；

音频指纹文件地址获取单元，用于接收所述音频ID和偏移量，并根据所述音频ID和偏移量，通过预先生成的音频ID-音频指纹库地址映射表，获得各个音频ID对应的音频指纹文件在所述音频指纹库中的音频指纹文件存放地址；

音频子指纹存放位置获取单元，用于接收所述各个所述音频指纹文件存放地址以及对应的所述偏移量，根据各个所述音频指纹文件存放地址以及对应的所述偏移量，在音频指纹库中获得所述音频子指纹在音频指纹库中的存放位置。

7、一种用于识别音频文件的检索方法，其特征在于，包括：

从所述待识别音频文件中，提取至少一段具有足够播放时间的音频段落，获取其音频指纹，作为待识别音频指纹；

提取所述待识别音频指纹中的音频子指纹作为待识别音频子指纹；

以所述待识别音频子指纹为依据，根据上述权利要求1-权利要求5任意一项提供的音频子指纹的检索方法，检索获得各个音频子指纹在音频指纹库中的可能存放位置；

根据所述每个音频子指纹在所述待识别音频指纹中的位置，在检索获得的所述可能存放位置的音频子指纹所归属的音频指纹中截取对应段落，获得待判别音频子指纹簇；

比较所述待判别音频子指纹簇与所述待识别音频指纹是否为实质相同的音频指纹；

若是，则读取所述待判别音频子指纹簇所在的音频指纹对应的音频ID，并将其确定为所述待识别音频文件的音频ID；

若否，则继续对根据所述待识别音频指纹中的音频子指纹获得的其他待判别音频子指纹簇进行上述比较，直到能够获得的所有待判别音频子指纹簇均已经判断完毕。

8、根据权利要求7所述的用于识别音频文件的检索方法，其特征在于，所述比较所述待判别音频子指纹簇与所述待识别音频指纹是否为实质相同的音频指纹，是根据所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离进行比较，若两者的汉明距离小于预定的阈值，则比较结果为是；否则，比较结果为否。

9、根据权利要求8所述的用于识别音频文件的检索方法，其特征在于，所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离采用如下方式获得：

将所述待判别音频子指纹簇与所述待识别音频指纹之间按位异或；

将异或结果中包含的二进制1的个数作为所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离。

10、根据权利要求9所述的用于识别音频文件的检索方法，其特征在于，所述异或结果中包含的二进制1的个数，采用下述方法获得：

对所述异或结果按照固定位数分段截取；

以截取获得的截取段作为关键值，查询预先设计好的对照表，获得该截取段包含的二进制1的个数；所述对照表为预先设置，对应每个关键值存储该关键值中包含的二进制1的个数；

将上一步骤中获得的各个所述截取段包含的二进制1的个数相加，获得所述异或结果的二级制1的总个数。

11、根据权利要求7所述的用于识别音频文件的检索方法，其特征在于，若能够获得的所有待判别音频子指纹簇均已经判断完毕，仍然未找到符合条件的音频ID作为所述待识别音频文件的音频ID，则进入下述步骤：

对所述待识别音频子指纹的二进制形式的其中1位二进制位进行修改；

将所获得的变化后的音频子指纹作为待识别音频子指纹，进入所述以所述待识别音频子指纹为依据，根据上述权利要求1-权利要求5任意一项提供的音频子指纹的检索方法，检索获得各个音频子指纹在音频指纹库中的可能存放位置的步骤，以及后续步骤；

若上述对1位二进制位进行修改获得的待识别音频子指纹仍然无法获得所述待识别音频文件的音频ID，则按照数量单调递增的方式，修改更多位数的二进制位；直到确定所述待识别音频文件的音频ID；或者，

直到需要修改的二进制位数超过规定的修改位数阈值，或者检索过的音频ID文件数量超出规定的阈值，此时，判断查询失败。

11、一种用于识别音频文件的检索装置，其特征在于，包括：

待识别音频指纹提取单元，用于从所述待识别音频文件中，提取至少一段具有足够播放时间的音频段落，获取其音频指纹，作为待识别音频指纹；

待识别音频子指纹提取单元，用于接收所述待识别音频指纹，并提取所述待识别音频指纹中的音频子指纹作为待识别音频子指纹；

音频子指纹检索单元，用于接收所述待识别音频子指纹，以所述待识别音频子指纹为依据，通过上述权利要求6提供的音频子指纹的检索装置，检索获得各个音频子指纹在音频指纹库中的可能存放位置；

音频子指纹簇获得单元，用于接收所述每个音频子指纹在音频指纹库中的可能存放位置，并据此在检索获得的所述可能存放位置的音频子指纹所归属的音频指纹中截取对应段落，获得待判别音频子指纹簇；

比较判别单元，用于比较所述待判别音频子指纹簇与所述待识别音频指纹是否为实质相同的音频指纹，若结果为是，则启动音频ID读取单元；若否，则对根据所述音频子指纹簇获得单元提供的根据所述待识别音频指纹中的音频子指纹获得的其他待判别音频子指纹簇进行上述比较，直到能够获得的所有待判别音频子指纹簇均已经判断完毕；

音频ID读取单元，用于接收所述比较判别单元的判别结果，若判别结果为是，则读取所述待判别音频子指纹簇所在的音频指纹对应的音频ID，并将其确定为所述待识别音频文件的音频ID。

12、一种音频查询系统，其特征在于，包括调度分发服务器、指纹服务器和指纹库；

所述调度分发服务器，用于接收音频文件查询请求，并解析所述音频文件查询请求，根据解析结果向相应的指纹服务器发出包含音频子指纹的检索请求；以及接收所述指纹服务器返回的检索结果；将所述检索结果返回给提出音频文件查询请求的客户端；

所述指纹服务器，用于维护音频指纹库的倒排表，以及接收所述检索请求，根据所述检索请求中包含的音频子指纹以及音频指纹，对所述音频子指纹进行检索，或者根据所述音频指纹进行识别音频文件的检索，获得检索结果后，将所述检索结果返还给发出检索请求的调度分发服务器；所述对音频子指纹进行检索，采用所述权利要求1-权利要求5任意一项的音频子指纹检索方法；所述根识别音频文件的检索，采用权利要求6-权利要求10提供的用于查询音频文件的检索方法；

所述指纹库，用于存储音频指纹。

13、根据权利要求12的音频查询系统，其特征在于，所述调度分发服务器采用HTTP或者HTTPS接口接收所述音频文件查询请求；所述HTTP或者HTTPS接口根据不同类型的音频查询请求，对客户端提供不同形式的接口。

14、根据权利要求12的音频查询系统，其特征在于，所述调度分发服务器通过GRPC远程调用协议连接所述指纹服务器。

15、根据权利要求12的音频查询系统，其特征在于，所述指纹服务器采用分布式系统提供服务，采用zookeeper分布式应用协调系统协调管理。

16、根据权利要求12的音频查询系统，其特征在于，所述指纹库位于所述指纹服务器的磁盘。

**音频子指纹的检索方法、装置以及音频查询系统**

**技术领域**

本申请涉及音频文件检索领域，具体涉及一种音频子指纹的检索方法以及装置。本申请同时涉及一种用于识别音频文件的检索方法以及装置。本申请同时涉及一种音频查询系统。

**背景技术**

为识别音频文件而对音频内容进行查询，已经被广泛使用。例如，使用手机采集一首歌曲并要求查询歌曲名，或者使用手机中存储的某个音频文件查询该音频文件的相关信息等，都是音频文件查询的应用场合。

为了识别音频文件，现有技术提供了采集获取音频文件的音频指纹的技术方案。通用的音频指纹技术是对音频文件进行如下处理：每隔11.6ms提取出一个反映该时段的音频特点的32bit的帧作为一个子指纹，一段音频的所有子指纹的集合则称为该音频的指纹。对于内容相同的音频文件，提取的音频指纹总是相同的；对于内容不同的音频文件，提取的音频指纹则不可能相同。因此，可以采用对应每个音频文件的音频指纹文件区分各个音频文件。由于音频指纹文件的数据量远远小于对应的音频文件本身，可以采用音频指纹文件对音频文件进行识别。

现有技术下虽然已经广泛使用了音频指纹对音频文件进行标识和识别，但是，如何根据一段已经获得的音频文件从音频文件的指纹库中查询到音频文件的相关信息，仍然是一个非常棘手的方法。

例如，申请号为：201510079596.3，名称为“构建音频指纹库及检索音频指纹的方法和设备”的中国专利申请，提供了一种检索音频指纹的方法，具体是为每一个音频指纹提取key值，并建立具有相同key值的音频指纹与key值的对应关系，并进而根据key值分配音频指纹存储的服务器；然后通过提取待识别音频文件的音频指纹以及key值，通过所述对应关系查找到相关服务器，进一步查询音频文件。

上述现有技术仅仅解决了音频指纹检索中服务器分配的方案，没有提供实现对音频文件快速检索的具体方法，显然不能满足快速查询音频文件的要求；并且该专利文献并未提供如何提取key值的方法，使该专利文献的实用性存在疑问。

**发明内容**

本申请提供一种音频子指纹的检索方法，以及一种用于识别音频文件的检索方法；其中所述用于识别音频文件的检索方法以所述音频子指纹的检索方法为基础。上述方法有助于实现对音频文件的快速检索。

根据最终权利要求书

与现有技术相比，本申请提供的音频子指纹的检索方法借鉴了文本检索中的倒排表方式，将音频指纹中的音频子指纹作为索引依据建立倒排表，该倒排表的每个条目记录了包含某个音频子指纹的音频ID。通过这种倒排表，可以快速获得包含音频子指纹的音频ID。

进一步的，在上述音频子指纹的检索方法的基础上，还可以在音频指纹库中相应的获得所述音频ID的音频指纹；从而可以进一步判断所述音频指纹与音频文件之间是否相符合，从而获取音频文件的相关信息；通过上述方法，可以进行音频文件的识别。

本申请提供的检索方法，创造性的引入了倒排表的方式记录音频子指纹与包含音频指纹的音频文件的对应关系，从而从一个新的维度实现了对音频文件的检索，能够显著提高音频文件的检索效率。

**附图说明**

图1是本申请第一实施例提供的子指纹检索方法的流程图；

图2是本申请第一实施例涉及的索引表、倒排表、映射表以及指纹库的相互关系的示意图；

图3是本申请第一实施例中采用三层索引结构构造索引表的示意图；

图4是本申请第二实施例的流程图；

图5是以本申请第二实施例为基础提供的一种具有容噪机制的用于识别音频文件的检索方法；

图6是子指纹key值变换1位-3位实现容噪机制的情况示意图；

图7是本申请第三实施例提供的音频查询系统的单元框图；

图8是本申请第四实施例提供的子指纹检索装置的单元框图；

图9是本申请第五实施例提供的用于识别音频文件的检索装置的单元框图。

**具体实施方式**

在下面的描述中阐述了很多具体细节以便于充分理解本申请。但是本申请能够以很多不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本申请内涵的情况下做类似推广，因此本申请不受下面公开的具体实施的限制。

本申请第一实施例提供一种音频子指纹的检索方法。请参看图1，该图为本申请第一实施例的流程图。请同时参看图2，该图为本申请第一实施例实现音频子指纹检索所涉及的索引表、倒排表、映射表以及指纹库的相互关系的示意图。以下结合图1、图2对本申请第一实施例提供一种音频子指纹的检索方法进行详细说明。

步骤S101，提取需要检索的音频子指纹。

本步骤用于获得需要检索的对象，即音频子指纹。一般检索目的下，获取音频指纹中的哪一段音频子指纹与本实施例关系不大。本实施例是作为音频文件检索的技术基础而存在。

音频指纹技术（Audio fingerprinting technology）是指通过特定的算法将一段音频中独一无二的数字特征以标识符的形式提取出来，用于识别海量的声音样本或跟踪定位样本在数据库中的位置。

音频指纹作为内容自动识别技术的核心算法，已广泛应用于音乐识别，版权内容监播，内容库去重和电视第二屏互动等领域。

获取音频指纹的方法有多种，其中Philips研究院提出一种经典的音频指纹算法，通过以下方式从音频文件中提取出音频指纹，即：每隔11.6ms提取出一个32bit的帧作为一个子指纹，而一段音频的所有子指纹的集合就是该音频的音频指纹。所述提取的32bit的帧是能够反映其所在的音频时段的音频特征的一个二进制序列，每个这样的32bit的帧就是一个音频子指纹。所有的音频文件的音频指纹都包括若干音频子指纹。理论上，不同的音频文件不可能具有相同的音频指纹，但是，不同的音频指纹可能包含相同的音频子指纹。

所述提取需要检索的音频子指纹，就是指在音频指纹中，选择某个音频子指纹，并将其读取出来的过程。一般的使用需求下，需要检索的都是音频文件，可以首先获取音频文件的音频指纹，然后再根据所获得的音频指纹，读取其中的音频子指纹。本实施例针对所获得的音频子指纹进行检索，其目的是提供一种检索音频文件的技术基础，在此基础上，可以实现多种可能的实际应用方式。

步骤S102，根据所述音频子指纹，在预先生成的索引表中读取对应所述音频子指纹的存储单元中存储的所述音频子指纹对应的倒排项地址。

本步骤用于实现根据音频子指纹获得其在倒排表中的倒排项地址的目的。

所谓倒排表是从文本检索中引入的一个概念，本申请中将其引申到音频文件检索领域使用，为音频指纹建立倒排表是本申请的关键环节，在此做出详细说明。

以下首先介绍传统文本检索领域的倒排表概念，然后再将这个传动概念引入到音频检索领域。

在文本检索领域的所谓倒排表，即倒排索引，由于一般采用表格形式，所以常常用倒排表称呼倒排索引。可以将倒排表理解为记录倒排索引的文件，其是否真正采用表格形式，均无实质影响；以下无论提及倒排表或者倒排文件或者倒排索引均为相同的含义。

所谓索引，是对数据库表中一列或多列的值进行排序的一种结构,使用索引可快速访问数据库表中的特定信息；记录索引信息的文件就是索引文件。对于搜索引擎，由于搜索空间巨大，索引文件对其非常重要，一般搜索引擎对搜索获得的文档均会建立索引。因为索引会记录在文件中，而文件又多用表格形式，所以索引、索引文件和索引表在实际使用中往往具有同样的涵义。

搜索引擎中使用的索引文件包括正排索引和倒排索引。

所述正排索引，即从文档角度看其中的单词，用于表示每个文档（用文档身份标识来标识）都含有哪些单词，以及每个单词出现了多少次（词频，英文缩写为TF）及其出现位置（一般记录单词相对于文档首部的偏移量，英文用offset表示）。正排索引以所述文档身份标识作为排序的依据。

所述倒排索引(inverted index，或inverted files)与上述正排索引相反，是从单词角度看文档；即记录每个单词（关键词）分别在哪些文档中出现(文档ID)，以及在各自的文档中每个单词分别出现了多少次（TF）及其出现位置（offset）。对于搜索引擎而言，一般是接收文本查询请求作为检索条件，所述文本查询请求经过预处理，形成若干分词（Term）以及分词之间的逻辑关系,通过这些信息使用索引表进行检索。

在本申请中使用的倒排表与上述文本检索领域的倒排表类似，只是将其中的关键词改为音频子指纹，所述文档则改为音频文件；即本申请所使用的倒排表，是以音频子指纹为索引关键词，在对应每个音频子指纹的条目中，则记录符合以下条件的音频文件的音频ID：这些音频文件的音频指纹中包含该音频子指纹；同时，对应每个音频ID，还对应记录作为关键词的音频子指纹在音频指纹中的位置，即偏移量。

如上所述，本申请所用的音频文件的倒排表按照以音频子指纹为关键词的条目排列，每个条目称为一个倒排项；每个倒排项中可以包含若干个记录，每个记录的形式为：（音频ID，offset1，offset2……），其中音频ID就是其音频指纹包含作为该倒排项的关键词的音频子指纹的音频文件的文件身份标识，通过所述音频ID可以唯一的确定该音频文件；所述offset1等，就是所述作为该倒排项的关键词的音频子指纹在所述音频ID标识的音频文件的音频指纹中的具体位置，即在音频指纹所包含的所有音频子指纹中，本音频子指纹所在的位置；这里称为偏移量或者偏移地址。

通过以上介绍可以看出，为音频指纹建立倒排表，可以像为文本文件建立倒排表一样，迅速找到其音频指纹包含某个音频子指纹的音频文件的音频ID；从而可以进一步获取所需要的关于该音频ID的相关信息。因此，在检索音频子指纹的方法中，在倒排表中找到对应步骤S101提取的音频子指纹的条目所在的存放位置，就成为了查询该倒排表的关键问题。由于所述倒排表的每个条目包含的记录不同，需要的存储空间不同，为了节省存储空间，又不可能给每个条目预定义足够的空间，因此每个条目的存储位置并不规律。

为解决上述问题，需要预先生成索引表，该索引表用于建立音频子指纹与倒排项存储位置的首地址之间的关系，这样就可以用一个便于查阅的规范的索引表解决倒排项条目的存储地址不规则的问题，建立从所述音频子指纹到倒排项存储地址的对应关系。

如图2所示，其中的索引表的每个存储单元均具有表征该存储单元地址的key0、key1、key2等标识；由于每个存储单元需要存储的是某个倒排项的首地址，记录格式规范，存储空间可以准确计算，因此，在获得某个音频子指纹对应的存储单元地址后，立刻就可以到该存储单元中读取到其中存放的倒排项首地址。

对于任何一个音频子指纹，如何方便的获得其在所述索引表中对应的存储单元地址，是查阅所述索引表的关键，也就是如何从所述音频子指纹获得所述索引表存储单元地址key。

实现上述从音频子指纹到索引表存储单元地址key的方式可以有很多种。只要采用规律并且方便计算的方式即可；在此提供一种简单易行的方案。

根据前述提到的所述音频子指纹的特点可知，所述音频子指纹为32bit的二进制记录，同时，在上述获取音频指纹的方法中，所述32bit的音频子指纹的最后一位实际上永远是“1”，即不具有标识性。

针对上述特点，采用下述方式确定对应音频子指纹的存储单元：将所述音频子指纹右移一位，所获得的数值就作为对应所述音频子指纹的子指纹key值；将所述子指纹key值作为存储地址，将其指向的所述存储单元作为对应所述音频子指纹的存储单元。

当然，所述子指纹key值只是逻辑层面的存储单元地址，真正的存储单元物理地址一般需要加一个偏移量。

通过上述方式，可以方便的进行索引表的存储地址的分配；只要获得音频子指纹，就可以通过两个简单的指令—二进制数右移操作和地址偏移量操作，就可以获得索引表中对应该音频子指纹的存储单元。所述存储单元中存放有所述音频子指纹在倒排表中的对应的倒排项的首地址，这样，通过直接读取所述存储单元中的记录的地址，就可以直接获得所述音频子指纹对应的倒排项在倒排表中的地址。

所述倒排表，可以存储在专门的服务器中。由于需要存储的数据量比较大，并且每个倒排项的存储空间并不能确定，并且还需要为每个倒排项提供足够的存储空间。因此，确定每个倒排项的存储地址并提供足够的空间是关键问题。在上述为每一个音频子指纹提供一个专门的存储空间，一对一的为其分配存储单元的方案中，可以采用下述方式分配所述倒排项地址：采用所述音频子指纹的所述子指纹key值，通过完美哈希算法投射到所述倒排表的倒排项地址空间，将所述完美哈希算法获得的地址作为所述音频子指纹对应的倒排项在倒排表中的倒排项地址。

上述方法需要首先确定一个存储倒排表的各个倒排项的倒排项地址空间，根据所述倒排项地址空间，利用完美哈希算法，就可以获得一个不会出现冲突的倒排项地址分派方案。据此，在索引表中对应子指纹key值的地址存储使用完美哈希算法获得的倒排项地址，即可获得快速查询倒排项地址效果。

使用上述方式进行倒排项地址分配，其查询过程就是简单的查表检索过程，并且作为关键字的所述子指纹key值为有规律的连续量，便于查询；因此，采用这种方式获得的索引表查询速度很快。

上述采用完美哈希算法的问题在于，需要为所述索引表提供足够的存储空间。即31位二进制数的每个可能取值都需要为其在索引表中设置存储空间，这样就需要231个存储单元，加上每个存储单元中存放的倒排项地址的位数，则为了存储这个索引表需要的空间很大。

例如，对于31位的key值，由于key值的个数为2G个，每个key对应的倒排表中的倒排项地址为64bit偏移地址。这样简单的perfect hash直接用子指纹key值作为地址，每个子指纹key值对应的索引值为64bit，这样仅仅索引表就需要16G内存。显然，其占用的空间过大。

由于存在上述占用空间过大的问题，本申请提供另外一种构造索引表的方法，即采用三层索引结构构造索引表，其目的是以时间换取空间，即获得如下效果：存储所述索引表的空间显著减少，代价是在获得所述倒排项首地址时计算量上稍有增加。

请参看图3，该图为采用这种三层索引结构构造索引表的示意图。

如前所述，所述音频子指纹为长度32比特的二进制数；将所述音频子指纹右移一位获得31位的子指纹key值；在此基础上，所述预先生成的索引表采用三级索引表构造的方法如图4所示。需要说明的是，图4所述的流程图只是为了说明的方便，在该构造方法中，步骤S102-1到步骤S102-3都是必须要处理的步骤，但是在次序上并无先后之别。

步骤S102-1，以10位二进制比特位为第一key值，建立一级索引表，在每个第一key值为地址的存储位中，对应存储64比特位的第一地址数据。

所述以10位二进制比特位是第一key值，建立一级索引表；是指将10位二进制比特位可能出现的值全部排序，每一个可能的取值为一个key值，即第一key 值（与后续第二key值、第三key值相区别），以每个第一key值作为地址，提供存储单元，即存储位（使用存储位，是为了与前面提到的存储单元相区别），在每个存储位中，存储一个64比特位的数据作为第一地址数据。

由于是以10位二进制比特位为第一key值，因此，能够获得的所有第一key值的总量为1k；又因为每个存储位中存储一个64比特位的数据，即每个存储位占8个字节；则第一地址数据耗费了8k字节的存储空间。

步骤S102-2，以25位二进制比特位为第二key值，建立二级索引表，在每个第二key值为地址的存储位中，对应存储32比特位的第二地址数据。

所述以25位二进制比特位是第二key值，建立二级索引表；是指将25位二进制比特位可能出现的值全部排序，每一个可能的取值为一个key值，即第二key值，以每个第二key值作为地址，提供存储单元，即存储位，在每个存储位中，存储一个32比特位的数据作为第二地址数据。

由于是以25位二进制比特位为第二key值，因此，能够获得的所有第二key值的总量为32M；又因为每个存储位中存储一个32比特位的数据，即每个存储位占4个字节；则第二地址数据耗费了128M字节的存储空间。

步骤S102-3，以31位二进制比特位为第三key值，建立三级索引表，在每个以第三key值为地址的存储位中，对应存储16比特位的第三地址数据。

所述以31位二进制比特位是第三key值，建立三级索引表；是指将31位二进制比特位可能出现的值全部排序，每一个可能的取值为一个key值，即第三key值，以每个第三key值作为地址，提供存储单元，即存储位，在每个存储位中，存储一个16比特位的数据作为第三地址数据。

由于是以31位二进制比特位为第三key值，因此，能够获得的所有第三key值的总量为2G；又因为每个存储位中存储一个16比特位的数据，即每个存储位占2个字节；则第三地址数据耗费了4G字节的存储空间。

步骤S102-4，将上述第一地址数据+第二地址数据+第三地址数据加和，形成完整的地址数据，该地址数据用于作为所述音频子指纹的倒排项地址。

采用上述三层索引结构构造索引表的方式，实际上需要为索引表耗费的存储空间约为4G，相比采用完美哈希方法耗费的16G空间，该方法可以节约大约四分之三的空间消耗。

采用上述三层索引结构构造索引表后，采用如下方式获取某个音频子指纹的倒排项地址。

首先，将所述音频子指纹的二进制数据右移一位，获得31位的子指纹key值。在此基础上，执行下述步骤：

根据所述子指纹key值的前10位，读取所述一级索引表相应的第一key值的存储位中存储的第一地址数据；

根据所述子指纹key值的前25位，读取所述二级索引表相应的第二key值的存储位中存储的第二地址数据；

根据所述子指纹key值的全部31位，读取所述三级索引表相应的第三key值的存储位中存储的第三地址数据；

根据所述第一地址数据+第二地址数据+第三地址数据，获得所述音频子指纹对应的倒排项地址。

上述操作中，需要对子指纹key值进行两次移位，即获取所述子指纹key值的前10位时，右移21位；获取所述子指纹key值的前25位时，右移6位。另外需要执行三次查询，即分别查询一级索引表、二级索引表以及三级索引表。最后，还需要执行一次加和操作，即将从各级索引表获取的地址数据相加构成最终的倒排项地址。尽管需要增加上述操作，但该构造索引表的方法能够有效的降低索引表消耗的存储空间。由于所述索引表优选的是在指纹服务器的内存中存储的，内存的存储空间一般都比较紧张，减少存储空间带来的效益非常显著；而上述增加的操作都并不复杂，其算法复杂度还是o（1），不会显著延长获得倒排项地址的时间，因此可以接受。

如图2所示，执行本步骤之后，根据所述音频子指纹的子指纹key值key3，在预先生成的索引表中读取地址为key3的存储单元中存储的地址，该地址就是所述音频子指纹的倒排项地址。此处的地址，为倒排项的首地址；所谓倒排项首地址，是指倒排项存储位置的第一个存储单元的地址，这是由于倒排项记录的内容会占据多个连续的存储单元，获得首地址，即可顺次获得倒排项中的所有记录。

步骤S103，根据所述倒排项地址，在预先生成的倒排表中读取相应的倒排项，获得所述倒排项下存储的所有倒排项记录，每个倒排项记录包括音频ID和偏移量。

本步骤用于获取倒排表中存储的与所述待检索的音频子指纹相关的信息。

所述倒排表的结构在上述步骤S102中已经说明，在此不再赘述。使用上述步骤S102获得的倒排项地址，即可在所述倒排表中读取到与所述待检索的音频子指纹相关的信息，即倒排项中的各个记录。

在每一个倒排项中，可以有多个记录；即，某个音频子指纹可以在多个音频文件中出现；此外，对于每一个记录，还可以有多个偏移量，即同一个音频子指纹，可以在一个音频文件中出现多次。

如图2所示，根据上述步骤S102查询获得的对应key3的倒排项地址，查询获得该倒排项地址指向的倒排项条目下的记录，该记录提供了音频指纹中包含该音频子指纹的音频文件的身份标识，即音频ID，具体为fileID；同时提供了一个偏移量，即图中的Offset。当然，在同一个倒排项条目下，往往会有多个倒排项记录，此时会获得多个所述fileID；另外，每个倒排项记录中可能包含多个偏移地址。图2所示的例子为最简单的情形，没有考虑上述情况。

步骤S104，根据所述音频ID和偏移量，通过预先生成的音频ID-音频指纹库地址映射表，获得各个音频ID对应的音频指纹文件在所述音频指纹库中的音频指纹文件存放地址。

本步骤用于获得对应音频ID的音频指纹在音频指纹库中的存放地址。

上一个步骤中获得了音频ID，根据所述音频ID，即可获得音频指纹库中存储的该音频ID的音频指纹。但是，直接根据音频ID本身并不能获得音频指纹文件存放地址，因此还需要建立音频ID-音频指纹库地址映射表（图2中简称为映射表），用于记录所述对应所述音频ID的音频指纹在所述音频指纹库中的存放地址；该映射表与音频指纹库同时建立；其索引顺序即根据所述音频ID的文本顺序，可以非常方便的根据所述音频ID获得其对应的音频指纹库地址。

如图2所示，通过上述步骤S103获得的倒排项记录中的音频ID，具体为fileID，在所述音频ID-音频指纹库地址映射表中，获得了对应fileID的音频指纹库地址address；该地址实际为所述音频ID对应的音频指纹在所述音频指纹库中存放位置的首地址。图2中只考虑了倒排项中只记录了一个倒排项纪录的简单情况，实际情况中，往往一个音频子指纹被多个音频文件的音频指纹包含，会出现需要查询多个音频ID对应的倒排项记录的情况。

步骤S105，根据各个所述音频指纹文件存放地址以及对应的所述偏移量，在音频指纹库中获得所述音频子指纹在音频指纹库中的存放位置。

本步骤用于直接查询音频子指纹。

经过所述步骤S104查询到的音频指纹文件存放地址，就可以在音频指纹库中获得包含待检索的所述音频子指纹的音频指纹文件的存放位置，并前去读取。而根据所述步骤S103查询获得的对应所述音频子指纹的偏移量，还可以获得所述音频子指纹在所述音频指纹中的具体位置。

如图2所示，在查询中，音频指纹库（图2中简称为指纹库）的地址来自步骤S104获得的音频指纹库地址address，该地址实际为所述音频ID对应的音频指纹在所述音频指纹库中存放位置的首地址；同时，在查询中还使用了步骤S103获得的偏移量；该偏移量是对应所述音频ID即fileID的偏移量，在实际情况中，同一个倒排项记录中，可以有多个偏移量，即对应该倒排项记录的fileID这个音频文件的音频指纹中包含多个待检索的音频子指纹。

根据上述第一实施例提供的方法，可以通过待检索的音频子指纹获得音频指纹中包含该音频子指纹的音频ID，还可以获得所述音频子指纹在所述音频ID的音频指纹中的具体位置，实现了上述检索，就可以在其基础上实现一些应用层面上有实际价值的功能。本申请第二实施例即提供一种用于识别音频文件的检索方法，该检索方法在实际应用中用于实现“听歌识曲”这样的功能，并可以作为实现其他音频检索功能的基础。

请参看图4，该图为本申请第二实施例的流程图；以下结合图4介绍第二实施例提供的用于识别音频文件的检索方法。

步骤S201，从所述待识别音频文件中，提取至少一段具有足够播放时间的音频段落，获取其音频指纹，作为待识别音频指纹。

本步骤的作用是获得音频指纹。

所述待识别音频文件，即使用者期望获得其相关信息的文件，可以是某一首乐曲，例如，在商场听到的某个现场背景歌曲，但不知道该歌曲的曲目以及演唱者等情况，希望查询了解，则可以现场录制一段音频内容获得待识别音频文件。当然，所述待识别音频文件也可以是已经在客户端存放的某一个歌曲或其他音频节目。

为了进行查询，首先需要获得该音频的音频指纹，即待识别音频指纹。具体的通过音频获得音频指纹的原来在前面已经介绍，由于属于现有技术，并非本申请的重点，在此不予详述。对一段音频提取音频指纹的计算过程，需要一定的计算能力，此计算能力可以由手机等客户端提供，也可以由手机等客户端将音频存送给服务器，由服务器提取其音频指纹。

要获得由辨识度的音频指纹，则用于提取音频指纹的音频内容不能时间过短，因此需要提取至少一段具有足够播放时间的音频段落。

步骤S202，提取所述待识别音频指纹中的音频子指纹作为待识别音频子指纹。

此步骤用于从音频指纹中获得音频子指纹。

听歌识曲等功能的目的是获得某一音频节目的相关信息，即至少要获得该音频节目的音频ID，然后就可以通过音频ID使用一般的查询方法查询其他相关信息。

在所述步骤S201中获得待识别音频文件的音频指纹后，并不能直接利用该音频指纹来检索对应该音频指纹的音频ID，因为该音频指纹信息量很大，不能作为检索的“关键词”。实际上，在进行文本检索时，一般也不可能将整篇文件作为检索关键词；因此，必须从音频指纹中提取音频子指纹进行检索。

由于所述音频指纹是由大量音频子指纹组成的，因此，可以从所述音频指纹中提取音频子指纹作为检索的“关键词”。

当然，一个音频指纹中可以包含大量的音频子指纹，可以将所有的音频子指纹均提取出来，全部用于后续检索，也可以只提取少量的音频子指纹；提取时，可以顺序提取，也可以从中间某个位置提取音频子指纹。

在本实施例中，只考虑获得最终的检索结论，至于如何从音频指纹中提取音频子指纹，提取多少等问题，只要满足识别音频文件的要求，都是允许的，甚至，提取音频子指纹的过程可以是随机提取的，只要最终获得准确的结果即可。这是由于，对音频文件的检索不像对某个文章的检索，如果期望对某个文章的出处进行检索，阅读者完全可以自己识别关键词，例如题目、作者、关键名词等，将这些作为检索依据。使用音频指纹检索则无法人为识别，只能由设备选择，设备则可以采用各种方法可能的方法在所述音频指纹中获得音频子指纹。在此，对具体从音频指纹中选择代表性的音频子指纹的方案不做限制。

步骤S203，以所述待识别音频子指纹为依据，根据上述第一实施例提供的音频子指纹的检索方法，检索获得音频子指纹在音频指纹库中的可能存放位置。

本步骤在步骤S202已经选择获得了待识别音频子指纹的基础上，使用上述第一实施例提供的方法获得音频子指纹在音频指纹库中的可能存放位置。所述可能存放位置，其含义是音频子指纹实际上很可能具有多个存放位置，但是对应待检索音频文件而言，只有一个存放位置是正确的；并且，即使只找到一个存放位置，也不能认为该位置对应的音频ID就是需要检索的音频文件，还需要在后续步骤中进一步验证；因此，此处获得的只能是“可能存放位置”。

由于前述步骤提取的音频子指纹可能是多个，因此，本步骤中也可以是对多个音频子指纹进行相关检索。类比文本检索，就是选取多个关键词进行检索。

步骤S204，根据所述每个音频子指纹在所述待识别音频指纹中的位置，在检索获得的所述可能存放位置的音频子指纹所归属的音频指纹中截取对应段落，获得待判别音频子指纹簇。

本步骤对检索获得的音频ID是否准确进行验证。

从上述步骤S203可以获得待检索音频子指纹在指纹库中的位置；这个位置具体到所述待检索音频子指纹位于的音频指纹的偏移量。当然，这个位置可能不止一个，因为所述音频子指纹可能出现在不同音频文件的音频指纹中；并且，所述音频子指纹完全可能在一个音频文件的音频指纹中出现多次。本实施例为了简化情况，假设检索获得的音频子指纹在指纹库的位置只有一个；对于多个位置的情况，实际上就是重复相同的验证步骤若干次，不予赘述。

另外一方面，对于被实际检索的音频子指纹而言，其在所述待识别音频指纹中的位置，以及所述待识别音频指纹包含的子指纹个数都可以方便的获得。

例如，所述待识别音频指纹包含192个音频子指纹，被实际检索的音频子指纹位于所述待识别音频指纹的第20位。

在上述信息的基础上，当获得了被实际检索的音频子指纹的位置后，则可以以该可能存放位置为基础，在音频子指纹所归属的音频指纹中截取对应段落。也就是说，在所述检索获得的音频子指纹所在的音频指纹中截取一段音频指纹，该段音频指纹与所述待识别音频指纹长度一致，并且所述实际被检索的音频子指纹在其中的位置也一致。如上述例子，应当截取包含192个音频子指纹的段落，并且，该192个音频子指纹包括所述被实际检索的音频子指纹位置开始前面的第20到后面第172个。

对于被截取的音频指纹段落，由于其既不是完整的音频指纹，也不是单个的音频子指纹，此处称为待判别音频子指纹簇。

步骤S205，比较所述待判别音频子指纹簇与所述待识别音频指纹是否为实质相同的音频指纹；若是，则进入下一个步骤；若否，则继续对根据所述待识别音频指纹中的音频子指纹获得的其他待判别音频子指纹簇进行上述比较，直到能够获得的所有待判别音频子指纹簇均已经判断完毕。

本步骤用于判断所述待判别音频子指纹簇与所述待识别音频指纹是否实质相同，并根据判断结果进入不同的后续步骤。

所述实质相同，其含义是两者可能不是完全相同，但是从比较两段音频信息来说，实际上已经没有差异可以视作相同了。采用实质相同这个概念，是由于待识别音频文件在采集过程中，可能会有环境噪音，采集设备可能会有误差，总之，由于各种原因，两个音频材料尽管来源相同，但是形成的音频指纹却不同之处，但该不同之处并不能说明两者之间有本质差别。如果不采用实质相同的概念，而要求所述待判别音频子指纹簇与所述待识别音频指纹完全相同，则可能由于要求过严，造成大多数音频文件均无法获得识别结果的问题。因此，在进行音频文件识别过程中，必然对两者是否相同采用实质相同的判断标准，并为此设置合适的阈值。

判断所述待判别音频子指纹簇与所述待识别音频指纹是否实质相同，其本质在于计算两者之间的差距有多大。对于数字信息，两个数字信息之间的差距可以用汉明距离来表征。当然，除了汉明距离之外，在现有技术下也提供了一些其他的判断两个数字信息之间差距的方式。

汉明距离表示两个具有相同位数的数字信息，其对应位不同的数量，一般以d（x,y）表示两个相同位数的数字信息之间的汉明距离。

在采用汉明距离作为衡量标准的前提下，需要规定一个阈值，其实质就是对应位不同的数量不能超过多少的一个设定值。考虑到作为比较对象的所述待判别音频子指纹簇与所述待识别音频指纹并没有固定的长度标准，因此，所述阈值可以是一个百分比，即100位的数字信息中，不能超过多少位不同。

对于所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离，可以采用将所述待判别音频子指纹簇与所述待识别音频指纹之间按位异或的方式计算汉明距离，获得异或结果后，所述异或结果中包含的二进制1的个数就是所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离。

对于两个二进制数字进行按位异或，在计算机处理中可以得到底层汇编语言的支持，一般计算机语言中都有简单的指令可以执行，例如汇编语言中的（xor a，b）指令就是将a、b两个二进制数按位异或；由于异或计算的特点是相异为1，也就是结果为1的位就说明相应位置上的二进制数值不同。

在按位异或之后，进一步需要计算结果中包含的1的个数，该个数就是所述汉明距离。

以上是计算汉明距离的理论方式，在实际中存在一些明显的问题，本实施例为此进行了改进。

首先，直接对所述待判别音频子指纹簇与所述待识别音频指纹进行异或存在位数过多，计算量过大的问题。

以前述包含192个音频子指纹的待识别音频指纹的情况为例。

假设需要比较的子指纹序列x和y都是由192个子指纹组成，每个子指纹为32bit，以d(x,y)来表示x和y的汉明距离。有公式如下



其中n=192\*32, 表示异或操作，x和y的汉明距离d(x,y)就是对应二进制位做异或操作后二进制位为1的个数。



显然，上述方式中，采用的是逐个位异或然后计数的操作方式。该操作方式是极慢的，完全没有利用计算机底层采用二进制位设计的特点。

本实施例中，采取每个相对应的子指纹进行异或，然后计算单个子指纹的汉明距离，然后将每个子指纹的汉明距离相加的方式获得汉明距离；这样，由于所述子指纹均为32位二进制数，对于计算机底层硬件而言，这个级别的位运算是最合适的，这样，就可以更快的执行异或运算，x和y的汉明距离，同时，不会出现



其中n=192，和分别为x和y的第i个子指纹，为第i个子指纹的汉明距离。这个方式下，以每32bit为一个单位进行一次异或操作，然后计算异或后的1的个数，异或操作减少为上一方式的1/32，显著减少了计算量。



其次，现有技术下对异或结果计数的过程仍然是一个非常耗时的过程。

现有技术下，需要通过移位判别的方式逐个计算1的个数，这个过程实际上非常耗时，显然不是合适的方式。

本实施例对此进行了改进，改为采用查表法计算。即：以16位二进制数为单位，将所有16位二进制数的整数（0~65535）所包含1的个数以数组的形式存储在一个映射表中。如整数1有1个二进制位为1，则table[1]=1；整数3有2个二进制位为1，则table[3]=2；整数65535有16个二进制位为1，则table[65535]=16。该映射表占据内存64k大小。

采用上述方案后，考虑到是以16位二进制位设计的映射表，因此，可以将原有子指纹序列x和y以16二进制位进行分割。先进行异或操作，然后通过查表方式找到异或后的汉明距离，公式如下：



这样一来每16位二进制数只需要做一次异或操作和一次查表操作，计算x和y整个汉明距离只需要192\*2\*2次基本操作即可，在只消耗了64k内存的情况下，显著减少了计算量。

当然，也可以以32位二进制数设计上述映射表，其缺点是占据的内存量过大。

实际上，也可以对已经获得的异或结果按照其他固定位数（例如8位）分段截取；以截取获得的截取段作为关键值，查询预先设计好的对照表，获得该截取段包含的二进制1的个数，再将各个截取段包含的二级制1的个数相加；所述对照表为预先设置，对应每个关键值存储该关键值中包含的二进制1的个数。

综合本实施例提出的上述两种计算汉明距离的改进方案，本实施例实际使用中，可以以待判别音频子指纹簇与所述待识别音频指纹的子指纹为单位，将对应子指纹进行异或，并将每组的异或结果分为前后两个16位二进制位，分别查表获得各个分段的二进制1的个数，然后相加。或者，在异或计算中，直接将对应的子指纹分为前后两个16位二进制位，分别异或，然后根据每段的异或结果查表，最后将各个段的查表结果相加，得到最终结果。

通过上述改进，可以有效解决汉明距离计算耗时过程的问题。

在本步骤的判断中，判断结果可以为是或者否，如果判断结果为是，即经过判断，所述待判别音频子指纹簇与所述待识别音频指纹之间的汉明距离在规定的阈值范围内，两者实质相同，则进入步骤S206进行后续处理。

如果判断结果为否，则还需要根据其他可能性是否穷尽做出处理，以下予以详细说明。

尽管本步骤对待判别音频子指纹簇与所述待识别音频指纹是否实质相同的判断结果为否，但是，很可能还有其他音频子指纹可以作为待识别音频子指纹。

此种情况又可以分为两种情况。即：所述待识别音频文件还有其他可以使用的音频子指纹可以作为待识别音频子指纹，也就是还可以从所述待识别音频指纹中获得新的“关键词”作为检索依据，即在步骤S202中可以获得其他新的“关键词”；或者，根据当前作为“关键词”使用的待识别音频子指纹还可以查询到其他倒排项记录，也就是对应所述待识别音频子指纹的倒排项还有其他倒排项记录，这些记录对应其他音频指纹，所述待识别音频指纹可以在这些音频指纹中。

总之，一方面，上述步骤S201获得的待识别音频指纹包含很多音频子指纹，可以在步骤S202中提取出的待识别音频子指纹可以有很多选择，选择其他子指纹作为待识别音频子指纹可能获得不同的检索结果，从而在本步骤S205中获得不同的判断结果；另一方面，即使是同一个音频子指纹，在倒排表中也可能找到多个倒排项记录，从而找到不同的音频指纹用于本步骤S205的判断，最终也可能获得不同的结论。

在上述情况没有完全排除完毕的情况下，只要没有找到与所述待识别音频指纹实质相同的待判别音频子指纹簇，都可以继续寻找其他相关的音频指纹，并执行本步骤的比较过程，直到最终找到符合要求的待判别音频子指纹簇。

当然，最终仍然可能出现穷尽了所有可能的线索——包括检索过每个音频子指纹或者对每个音频子指纹在倒排表中可以查到的对应的音频指纹都已经比较过相应的音频子指纹簇，最终还是没有找到合适的音频文件，则本步骤可以结束；可以将未检索到有效音频文件的情况提供给使用者。

但是，在很多情况下会出现待识别音频指纹的采集环境不佳，或者有其他干扰因素等，造成采集到的待识别音频指纹无法找到相关的音频文件的情况。由于听歌识曲等重要应用中，这种情况事实上普遍存在，有必要建立一定程度的容噪机制。为此，本申请提供一种容噪机制，以便在必要时进行更大范围的检索；对于这种进一步的改进方案，在介绍完本实施例的基础方案后，再进行介绍。

步骤S206，读取所述待判别音频子指纹簇所在的音频指纹对应的音频ID，并将其确定为所述待识别音频文件的音频ID。

本步骤是在所述步骤S205的判断结果为是的基础上进行的，有了步骤S205的判断结果，就可以确定所述待判别音频子指纹簇所在的音频指纹就是需要寻找的音频文件的音频指纹，获得该音频指纹对应的音频ID，就获得了识别结果，即所述待识别音频文件的音频ID。使用者如果需要进一步检索该音频文件的相关情况，则可以根据所述音频ID对自己感兴趣的情况进行进一步的检索。

以上第二实施例提供的检索方案，在穷尽可以作为“关键词”的待识别音频子指纹以及所有与待识别音频子指纹可能关联的音频指纹后，仍然有可能无法获得需要的音频文件，对此情况，可以向使用者提示无法获得检索结果。但是，如前所述，也可以考虑可能出现的一些干扰情况，提供容噪机制。以下提供一种作为第二实施例的改进方案的容噪机制。

请参看图5，该图示出了本申请的提供的一种具有容噪机制的用于识别音频文件的检索方法。以下结合图5说明该容噪机制下的检索过程，当然，该检索过程是和第二实施例相互结合的，与第二实施例各个步骤重复的部分在此不再赘述。采用该容噪机制进行检索，可以在已经穷尽了所有可以选择的待识别音频子指纹后进行，也完全可以在以某个待识别音频子指纹作为“关键词”检索过程中进行。请同时参考图6，该图示出子指纹key值变换1位-3位的情况示意图。

步骤S501，设置纠错距离阈值d=0。

本容噪机制的核心，是将待识别音频子指纹的二进制位进行变化；即假设由于某种干扰，造成待识别音频子指纹的二进制位数据发生了变化；由于是二进制，所谓变化，就是1变成0，或者0变为1。因此，本方法采用的就是将待识别音频子指纹中的二进制位变化的方法，实现容噪；显然，这种变化需要从变化位数最少的情况开始。

在容噪机制中，用于对待识别音频子指纹的二进制位数据变化位数进行控制的参数称为纠错距离，其代表符号为d。该数值就是对所述待识别音频子指纹的二进制位进行变化的位数的控制参数。

本步骤将d设置为0，其实质就是要求所述待识别音频子指纹不进行任何变化，即从完全不考虑噪声干扰开始检索。

步骤S502，生成纠错距离小于等于d的子指纹key值。

对于一个待检索音频子指纹，生成纠错距离小于等于d的子指纹key值，就是对所述子指纹key值进行小于等于d的变化。所述子指纹key值在本实施例中和待检索音频子指纹的不同之处在于，其位数为31位，即将位值必然是1的最后一位舍去了，使用子指纹key值进行检索的方式见本申请第一实施例；当然，直接使用待检索音频子指纹进行位值变化也是可以的，考虑到最后一位必然是1，则变化位不能是最后一位，其他任何一位都可以。在此处考虑到与第一实施例提供的检索音频子指纹的方案相一致，采用对子指纹key值进行变化的说法，两者实质没有差别。

由于是从d=0开始循环递增的，而d=0的情况就是第二实施例的情况，在此不做介绍。以下以d=1为例进行说明

所述纠错距离d=1，即所述音频子指纹key值可以有一位的变化。具体是改变哪一个位置的位值并没有规定，例如，可以考虑从最后一位开始向前开始逐位变化，当然也完全可以随机变化，但是要注意由于需要穷尽d=1时的变化情况，则还是最好对发生变化的位置设置规律。

如图6所示，d=1时，可以是音频子指纹key值发生一位的变化，例如，图6所示的源子指纹key值的最后一位从0变换为1，获得变化1位的子指纹key值。

在图示的例子中，本步骤的位值变化发生在子指纹key值的最后一位，实际上，满足d=1的变化有31种，在并行计算的情况下，可以同时对这些变化进行后续步骤S503和步骤S504的查询以及步骤S505的判断。如果不进行并行计算，也可以设置一个循环，逐位变化然后进行后续步骤S503和步骤S504的查询过程，以及步骤S505的判断过程，最终获得所有d=1的可能变化的查询结果。总之，d=1的情况可以有多种可选择的变化后的子指纹key值。在本实施例中，假设这些可能的d=1的情况均一次性并行处理完毕。

步骤S503，在倒排表中查找file\_ID和offset。

本步骤利用倒排表实现对所述音频子指纹的检索，其实现方式在前述第一实施例中已经说明，这里不再赘述。只是本步骤使用的音频子指纹key值时进行过位值变化的，因此检索结果和原来没有经过位值变化或者位值变化没有超过d的情况（例如，如果d=2，则前序步骤已经执行过d=0、d=1两轮判断，都没有获得结果）不同，检索结果也会有所不同。

步骤S504，按照file\_ID和offset查找对应的音频指纹并进而确定待判别音频子指纹簇。

本步骤通过上述步骤S503获得的file\_ID和offset，查找对应的音频指纹；并进一步获得用于与待识别音频指纹比较的待判别音频子指纹簇，这些步骤在上述第二实施例中已经说明，此处不再赘述。

步骤S505，判断所获得的音频指纹簇与所述待识别音频指纹是否实质相同，若是，则返回该识别结果；若否，则进入下一步。

本步骤的实现方式在第二实施例中已经说明；其实质是使用待判别音频子指纹簇与待识别音频子指纹进行比较，如进行汉明距离的计算，判断其是否在预设的阈值范围内，如果是，则说明所述待判别音频子指纹簇就是期望的结果，将其所属的音频指纹的音频ID确定为所述待识别音频文件的音频ID，并返回该识别结果；否则，需要继续下面的步骤。

步骤S506，判断总检索量是否超过了预定的数值范围；若是，则结束整个检索过程；若否，则进入步骤S507。

本步骤用于判断总检索量是否已经过大，避免对一个待识别音频文件投入过大的资源。这是由于，在建立上述容噪机制的基础上，无限制的增加d的数值则会造成与原来的待识别音频文件已经有了较大的差距，已经无法假设仅仅是噪声等干扰因素的影响了；另一方面，容噪机制增加的工作量在不加限制的情况下，可以是非常大的；超过一定范围的检索工作产生的计算量消耗对一般应用而言并不合理，因此，可以在超过一定的范围时停止。如生成并检索过的待识别音频子指纹如果已经超过100个，仍然没有查找到合适的音频文件，或者检索过的音频文件超过100个，均与所述待识别音频指纹不相符合等，就可以停止本次检索。确定停止检索后，可以向使用者发出无法检索到合适结果的通知。

步骤S507，设置d=d+1，返回所述步骤S502。

该步骤是在当前的纠错距离d下无法检索到合适的音频文件的情况下，允许增加纠错距离d，在新的纠错距离下进行下一轮检索，即返回步骤S502。当然，所述纠错距离不能无限制增加，一般在纠错距离d=3还没有检索到合适的音频文件时，总检索量将已经超过设定的阈值，会在步骤S506退出检索。

当然，完全可以设置一个纠错距离d的规定阈值，大于该规定阈值，将不再检索；如果采取此方案，可以在步骤S507后增加一个对d是否超过规定阈值的判断步骤来确定是否需要返回所述步骤S502。

变化1位后的子指纹key值，以及变化2位、3位的情况在图6中分别示出一个例子；不论是d=1或者2、3，都存在多种可能的变化方式，在图中不再一一示出。

通过上述容噪机制，实际上将检索范围作了非常大的扩展，提高了检索到符合要求的音频文件的可能性。

本申请第三实施例提供一种音频查询系统，该音频查询系统用于实现上述第一实施例和第二实施例提供的方法。

请参看图7，该图提供了所述音频查询系统结构示意图；图中同时示出与所述音频查询系统发生交互关系的客户端。

如图7所示，所述音频查询系统包括调度分发服务器701、音频指纹服务器702和音频指纹库703。

所述调度分发服务器701，用于接收音频文件查询请求，并解析所述音频文件查询请求，根据解析结果向相应的指纹服务器发出包含音频子指纹的检索请求；以及接收所述指纹服务器返回的检索结果；将所述检索结果返回给提出音频文件查询请求的客户端704。

以上是从功能角度对调度分发服务器701的描述，实际上，结合图7以及第一实施例和第二实施例提供的方法，可以很清楚的了解调度分发服务器的作用，在此不予赘述。

从硬件角度看，调度分发服务器701具有向客户端704提供的对外接口，以及用于向系统内部的所述指纹服务器702进行交互的接口。

所述调度服务器701对外提供的接口采用http或者https形式提供，用于接收来自客户端的查询请求。根据不同的需求，可以设置多种接口，每个接口均为特定的音频检索应用服务。接口的外部表现形式可以是通过网页提供的入口，或者通过某个APP提供的功能入口。

例如，为听歌识曲功能提供的接口，通过该接口可以获得客户端录制的音频材料，作为音频检索的依据；再如，离线歌曲查询接口，这一接口以客户端本地已经存好的音频材料作为检索依据；还有未知音频映射接口和曲库重排接口，用于读取音频材料并查找曲库中符合的歌曲。所述调度分发服务器601接收的各种检索请求一般来自手机等智能设备，待识别音频文件的音频指纹可以在调度服务器701一侧生成，也可以在远程的客户端生成。如果在客户端生成音频指纹，则客户端发出的检索请求中直接包含音频指纹，就不需要传送音频文件本身了。

调度分发服务器701对内则提供GRPC远程调用协议与指纹服务器602实现交互。

所述音频指纹服务器702，其功能为用于维护音频指纹库的倒排表，以及接收所述检索请求，根据所述检索请求中包含的音频子指纹以及音频指纹，对所述音频子指纹进行检索，或者根据所述音频指纹进行识别音频文件的检索，获得检索结果后，将所述检索结果返还给发出检索请求的调度分发服务器701；所述对音频子指纹进行检索，采用所述第一实施例提供的音频子指纹检索方法；所述识别音频文件的检索，采用第二实施例提供的用于查询音频文件的检索方法。

从功能上而言，音频指纹服务器702具有音频检索本系统的核心位置，对系统主要文件，如倒排表、映射表、索引表的管理就由其完成，实际检索过程也在音频指纹服务器702的控制下进行。

从硬件角度看，所述指纹服务器702采用分布式系统提供服务，可以包括数台或者数十台、数百台服务器，并且采用分布式系统进行管理运行；具体可以采用zookeeper分布式应用协调系统协调管理这些服务器，处理音频指纹检索过程中的资源分配。

所述指纹服务器702在上电之后或者在音频指纹库703更新或者添加数据后，会对音频指纹库的倒排表进行更新；所述倒排表以及使用倒排表时可能需要使用的映射表、索引表等，均在所述服务器的内存或硬盘上保存，根据检索时的需求程度进行调度和分配存储空间。

所述音频指纹库703，用于存储音频指纹。

该音频指纹库703可以位于所述指纹服务器702的硬盘上，采用便于检索的关系式数据库形式或者其他形式。所述音频指纹库703在存储音频文件的数据库发生更新的时候，会相应的更新。由于音频文件本身的存储以及获取的过程并非本申请的范围，在此不再赘述。

通过上述系统，即可较好的实现前述第一实施例提供的音频子指纹检索方法、以及第二实施例提供的用于查询音频文件的检索方法，实现听歌识曲、离线歌曲查询等各种与音频识别相关的功能；当然，由于通过该系统为完善音频检索提供了更多的可能，还可使实现各种进一步的音频资料管理功能成为可能。

本申请第四实施例提供一种子指纹检索装置；该子指纹检索装置与本申请第一实施例听歌的子指纹检索方法对应，在此予以简要介绍。

请同时参考图8；该图为本申请第四实施例的单元框图。

本申请第四实施例提供的音频子指纹的检索装置，包括：音频子指纹提取单元801、倒排项地址获取单元802、倒排表查询单元803、音频指纹文件地址获取单元804、音频子指纹存放位置获取单元805。

所述音频子指纹提取单元801，用于提取需要检索的音频子指纹。

所述倒排项地址获取单元802，用于接收所述音频子指纹，根据所述音频子指纹，在预先生成的索引表中读取对应所述音频子指纹的存储单元中存储的所述音频子指纹的倒排项地址。

所述倒排表查询单元803，用于接收所述倒排项地址，根据所述倒排项地址，在预先生成的倒排表中，获得所述倒排项地址对应的存储单元中存储的所有音频ID和偏移量。

所述音频指纹文件地址获取单元804，用于接收所述音频ID和偏移量，根据所述音频ID和偏移量，通过预先生成的音频ID-音频指纹库地址映射表，获得各个音频ID对应的音频指纹文件在所述音频指纹库中的音频指纹文件存放地址。

所述音频子指纹存放位置获取单元805，用于接收所述各个所述音频指纹文件存放地址以及对应的所述偏移量，根据所述各个所述音频指纹文件存放地址以及对应的所述偏移量，在音频指纹库中获得所述音频子指纹在音频指纹库中的存放位置。

本申请第五实施例提供一种用于识别音频文件的检索装置，该用于识别音频文件的检索装置与本申请第二实施例提供的用于识别音频文件的检索方法相对应，在此予以简要介绍。

请参考图9，该图为本申请第五实施例的单元框图。

本申请第五实施例提供的用于识别音频文件的检索装置，包括：待识别音频指纹提取单元901、待识别音频子指纹提取单元902、音频子指纹检索单元903、音频子指纹簇获得单元904、比较判别单元905、音频ID读取单元906。

所述待识别音频指纹提取单元901，用于从所述待识别音频文件中，提取至少一段具有足够播放时间的音频段落，获取其音频指纹，作为待识别音频指纹。

所述待识别音频子指纹提取单元902，用于接收所述待识别音频指纹，并提取所述待识别音频指纹中的音频子指纹作为待识别音频子指纹。

所述音频子指纹检索单元903，用于接收所述待识别音频子指纹，以所述待识别音频子指纹为依据，通过本申请第四实施例提供的音频子指纹的检索装置，检索获得各个音频子指纹在音频指纹库中的可能存放位置。

所述音频子指纹簇获得单元904，用于接收所述每个音频子指纹在音频指纹库中的可能存放位置，并据此在检索获得的所述可能存放位置的音频子指纹所归属的音频指纹中截取对应段落，获得待判别音频子指纹簇。

所述比较判别单元905，用于比较所述待判别音频子指纹簇与所述待识别音频指纹是否为实质相同的音频指纹，若结果为是，则启动音频ID读取单元；若否，则对根据所述音频子指纹簇获得单元提供的根据所述待识别音频指纹中的音频子指纹获得的其他待判别音频子指纹簇进行上述比较，直到能够获得的所有待判别音频子指纹簇均已经判断完毕。

所述音频ID读取单元906，用于读取所述待判别音频子指纹簇所在的音频指纹对应的音频ID，并将其确定为所述待识别音频文件的音频ID。

本申请虽然以较佳实施例公开如上，但其并不是用来限定本申请，任何本领域技术人员在不脱离本申请的精神和范围内，都可以做出可能的变动和修改，因此本申请的保护范围应当以本申请权利要求所界定的范围为准。

在一个典型的配置中，计算设备包括一个或多个处理器 (CPU)、 输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。

1、计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、 程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、 其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

2、本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。



**图1**



**图2**



**图3**



**图4**



**图5**



**图6**



**图7**



**图8**

**图9**