Chapter 10
Sound Sharing and Retrieval

Frederic Font, Gerard Roma, and Xavier Serra

Abstract Multimedia sharing has experienced an enormous growth in recent years, and sound sharing has not been an exception. Nowadays one can find online sound sharing sites in which users can search, browse, and contribute large amounts of audio content such as sound effects, field and urban recordings, music tracks, and music samples. This poses many challenges to enable search, discovery, and ultimately reuse of this content. In this chapter we give an overview of different ways to approach such challenges. We describe how to build an audio database by outlining different aspects to be taken into account. We discuss metadata-based descriptions of audio content and different searching and browsing techniques that can be used to navigate the database. In addition to metadata, we show sound retrieval techniques based on the extraction of audio features from (possibly) unannotated audio. We end the chapter by discussing advanced approaches to sound retrieval and by drawing some conclusions about present and future of sound sharing and retrieval. In addition to our explanations, we provide code examples that illustrate some of the concepts discussed.

Keywords Sound sharing • Sound retrieval • Multimedia • Audio metadata • Sound description • Audio database • Audio indexing • Audio features • Similarity search • Query by example • Sound taxonomy • Machine learning • Sound exploration • Sound search

10.1 Introduction

Multimedia sharing is one of the areas in which the social web has experienced the largest and quickest growth in recent years [73]. Just to name a few examples, every minute 100 h of video are uploaded to YouTube [75], 2400 photos are uploaded

F. Font (✉) • X. Serra
Music Technology Group (MTG), Universitat Pompeu Fabra, Barcelona, Spain
e-mail: frederic.font@upf.edu; xavier.serra@upf.edu

G. Roma
Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK
e-mail: g.roma@surrey.ac.uk

T. Virtanen et al. (eds.), *Computational Analysis of Sound Scenes and Events*,
DOI 10.1007/978-3-319-63450-0_10

to Flickr [32], and 12 h of music are uploaded to SoundCloud [79]. The case of *sound* sharing—understanding sound as any kind of reusable audio material like sound effects, environmental recordings, or music building blocks, but typically not finished music tracks—even if at a smaller scale, is not an exception. Websites such as Freesound, Looperman, CC-mixter, and the Radio Aporee project[1] are examples of sound sharing platforms in which users can not only search and browse content but also contribute new audio recordings including sound scenes such as field and urban recordings, sound events such as *foley* or animal sounds, and music samples such as loops, melodies, and single notes. Furthermore, consumer-oriented websites such as Sound Dogs, Sound Snap, and A Sound Effect[2] sell sounds from extensive collections of audio recordings that users can also navigate. In this chapter we focus on the particular context of online sound sharing and give an overview of different ways to approach sound sharing and retrieval challenges.

On a prototypical scenario of online sound sharing, a user might record a sound and upload it to a web application so that other users can listen to it and possibly download it. The intent with which users upload and share multimedia content can vary widely, but we can identify some general patterns according to the usage that the uploaders may expect of the contributed content. On the one hand, we can identify content that is meant to be accessed and viewed or played through the online sharing platform itself. Hence, the *end use* of the resource is its online consumption. For example, someone may upload photos of an event to a photo sharing site so that other participants of that event can have access to the photos, or a musical artist can upload a music album to a music sharing site so that other users can listen to it. On the other hand, there is an additional type of uploaded content which is meant to be reused outside the sharing platform where it is hosted. Here, the display in the sharing platform does not represent an end use per se. Some examples of this situation include sharing recordings of sound events that can be later used in video games, drum loops in music compositions, video backgrounds or transitions to be used in audiovisual installations, or images to be used in collages or as a desktop wallpaper. These latter cases of multimedia sharing particularly support Lawrence Lessing's definition of *read/write culture* [39]. In read/write culture, users are both consumers and producers of content that is easily shared and reused through the internet [80].

Such content potentially represents an incredibly valuable resource that can serve several purposes, ranging from business and research applications to artistic creation and the preservation of cultural heritage [34]. Nevertheless, the value of this content is significantly dimmed by the ways in which it can be accessed and reused, i.e., the ways in which it can be retrieved. As the amount of content grows, so does the difficulty of browsing and locating what one needs, and so do the challenges that search engines have to face. For the content to be accessible, it needs to be properly indexed. However, the quantity and variety of available content turns

[1] https://freesound.org, https://looperman.com, http://ccmixter.org, http://aporee.org/maps.

[2] https://sounddogs.com, https://soundsnap.com, https://asoundeffect.com.

proper indexing into a very difficult task. This is particularly true for multimedia resources like video, pictures, and audio which, as opposed to other kinds of media, do not have a direct textual representation [5]. At the same time, the amount of content generated is simply too much to be curated in scalable ways by groups of experts.

The description, indexing, and retrieval of audio content is therefore a challenge that needs to be faced in order to make audio shareable and increase its value. Especially in the context of read/write culture, users need sophisticated and specialized ways of accessing online resources that fit their particular requirements. Users searching for content in sound sharing sites might be looking for audio clips with very specific and detailed characteristics that can be represented by a wide range of audio properties. For example, one user might be searching for the sound of an opening door with a particular duration, size, and material of the door, while another user might be searching for the sound of a melody being played by a particular instrument with a specific tonality, tempo, and mood. Being able to successfully retrieve such specific content poses a number of issues to both the users and the sharing platform. Another relevant aspect of sound sharing is that the assessment of the results returned by a search engine of a sound sharing site requires the time to listen to them, and cannot be done as instantly as it could be done with the search results of, for example, a photo sharing site. From this point of view, the cost of iterating over several queries in order to find the desired resource is higher for sounds than for images. This is one of the reasons why good quality descriptions are crucial for indexing audio.

But how should sounds be described so that users can effectively search them? As it might be expected, this question does not have a single definitive answer. Nevertheless, we can intuitively differentiate at least two ways in which sounds can be described. On the one hand, sounds can be generally described by referring to the source that produces them. In other words, we can describe a sound by denoting an object and (possibly) an action that produces it (e.g., "the sound of a closing door"). On the other hand, sounds can be described by referring to their perceptual qualities regardless of their source, that is to say, by describing the timbre and acoustic qualities of a perceived sound (e.g., "a loud high-pitched sound"). Both approaches are complementary and both bear relevant information for indexing and retrieval purposes [43]. Source-based descriptions can be effectively indexed by using metadata annotations such as labels and textual descriptions. Conversely, some perceptual qualities can be better represented using automatically extracted audio features. In addition, other sound properties such as audio format and editorial information can be used when indexing content. Once content is described and indexed, different browsing and searching strategies can be implemented such as text-based search, sound browsing based on category filtering, or search based on audio similarity. All these strategies and many other possible ones ultimately enable sound search and discovery.

In this chapter we go through the different components and the typical issues and solutions of sound sharing systems, and show step by step how to build a basic system with references to code examples. The components we describe are similar

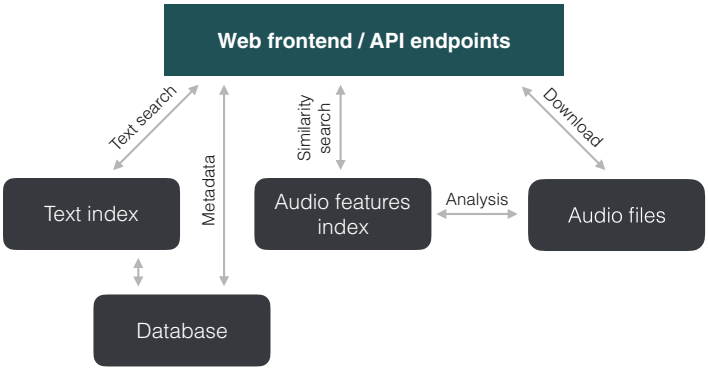

Fig. 10.1 Block diagram of Freesound's main components. The *arrows* correspond to main functionalities of Freesound and show the components that they need to access. Note that metadata information and audio features are indexed separately and provide different functionalities. Also note that audio files are not included in the database itself (see Sect. 10.2). This diagram serves as a general example of the different components of a sound sharing and retrieval system

to those that can be found in a site such as Freesound, which we take as an example (Fig. 10.1). We start by describing how to build an audio database (Sect. 10.2), and continue by explaining metadata-based retrieval strategies (Sect. 10.3) as well as other retrieval strategies based on audio analysis (Sect. 10.4). We end this chapter with a discussion about advanced audio retrieval topics (Sect. 10.5) and a conclusions section (Sec. 10.6). Code examples, written in the Python[3] programming language and based on open source technologies, are available in the book's accompanying website[4] and demonstrate some of the concepts discussed in Sects. 10.2–10.4.

10.2 Database Creation

The concept of a database is traditionally associated with text and numerical information. While many database programs can store binary objects, standard practice for applications involving image, audio, or video files is to leave them in the file system and store the paths in the database. The expression *audio database* has been used during the last decades in research on audio analysis to refer to collections of audio files (e.g., [20, 23, 25]). Much of this research was precisely trying to find a way to index audio files and facilitate search and discovery. In this chapter, we will refer to an audio database as the set of information used for indexing collections of

[3] https://python.org.

[4] www.TODO:bookwebsite.

audio files. In the following subsections, we summarize the most important design issues for designing an audio database. First we discuss practical aspects such as file formats and licensing, then we highlight the key design issues related with the two main types of information used for indexing audio: metadata and audio features.

The code examples provided in the accompanying website of this book show how to build an audio database taking into account the design issues discussed below. We show how to download a small number of sounds from Freesound in a standardized preview format using the Freesound API.[5] We retrieve sounds that match the keywords *dog bark*, *cat meow*, *lion roar*, and *nightingale*, and also retrieve its associated metadata and some pre-computed audio features. Then we show how to store this information in a data structure and create a text index using popular Python libraries. Both data structures are used in later sections of this chapter to demonstrate metadata-based and audio-based retrieval strategies (Sects. 10.3 and 10.4, respectively).

10.2.1 Licensing, File Formats, and Size

Licensing terms for audio are more of a legal than a technical issue. However, they will often be a key factor for anyone building an audio database, especially for shared usage. Copyright licensing will be generally needed for audio recordings but additional licensing may be required. For example, if the performance of a musical composition appears in the recording, a license for the composition also applies. There are two interactions to consider: first, the database developer or administrator needs a license from the content author. Second, database users willing to play or download audio files for reuse need a license too. In many social media applications (such as Freesound or Flickr) a license is chosen by the content author at upload time and is propagated by the database to the end users. The development of Creative Commons[6] (CC) licenses was instrumental in the emergence of this use case because these allow re-distribution under clear terms. Re-distribution is generally not permitted under traditional copyright. While their applicability depends on the audio material (e.g., whether existing protected content is considered) and/or possible commercial agreements, CC licenses provide mature legal terms curated by copyright law experts that may be useful to developers with little or no expertise on this subject.

File format can be naively seen as a trivial issue. Even though it may be trivial for a database designed for individual usage, for collective usage it is convenient to support multiple formats. Choice is complicated due to the plethora of available options, each with its own implications, often due to commercial factors. For this reason, early planning is advised. As a general rule, variety of formats is convenient

[5] https://freesound.org/docs/api/resources_apiv2.html.

[6] https://creativecommons.org.

for collecting sounds from different authors (who may use different equipment), and uniformity is convenient to facilitate reuse. An obvious general way to classify formats is to distinguish between *uncompressed*, *lossy* (i.e., when the compression implies loss of information), and *lossless* compressed formats. In some cases, a distinction between *container* and *codec* may be relevant, especially when dealing with video files. In pure audio formats there is often no distinction between container and codec, so we will not make that distinction in this chapter either. In addition to the format, there are numerous possible combinations of sampling rate and sample bit depth. Audio production textbooks are a good source for conventional choices of sampling rate and bit depth [29]. The most universal is the one defined by the compact disc standard, 44.1 kHz and 16 bits. With respect to uncompressed formats, WAV and AIFF are the most common. As an example of non-trivial issue, 24-bit integers are a very common choice, yet some high-level languages such as Python lack a specific 24-bit primitive data type, and as a consequence reading 24-bit wav files in some Python packages will not work. Similar situations may be encountered with 32-bit audio and for any obscure combinations: libraries for dealing with different audio formats may surprisingly fail to recognize many files. Lossy compression has been instrumental in the popularization of digital audio in the Internet age. Perhaps the main issue is that many formats are covered by patents. At the time of this writing, MP3 patents are progressively expiring. Open formats such as OGG are free to use but often not supported in commercial players and devices. A similar situation is found with respect to lossless compressed formats: while open formats such as FLAC are available, companies often develop their own formats and protect them, so they are mostly used in specific platforms. Compressed formats will introduce more possibilities for variation, typically a bit rate or quality parameter. A common sense strategy is to allow authors to contribute content with the format they want, and then convert to a standardized format. For large scales, if an uncompressed format is chosen as standard and/or the originals are preserved, this approach may require large amounts of storage space.

File size and duration is also something to consider when creating an audio database, not only in order to plan the storage needs but also for deciding the way in which files are accessed. Short audio recordings will usually contain sound events, while longer recordings may contain music, speeches, or environmental sound scenes. Analysis and segmentation of longer recordings may be of interest for some applications in order to isolate specific events or to allow streaming. Large file sizes also complicate transmission for authors, for example, when using HTTP to upload files to a database server.

10.2.2 Metadata

Metadata can be defined as “data about data.” In the case of audio it usually refers to textual information that is used to describe and index an audio file or segment. Virtually all existing sound sharing platforms implement some kind of metadata-

based retrieval strategy. Text is the most established way to deal with any sort of information stored in computers, so for most applications, some kind of metadata is necessary. Audio files already contain some sort of metadata in the headers, such as sampling rate, bit depth, bit rate, and potentially editorial information, that can be added to the database for indexing. Nevertheless, sound sharing platforms often delegate the responsibility of providing metadata to the content authors or editors. This will typically include a name (which may or may not coincide with the file name), a textual description of the content of the sound, a number of labels or *tags*, or other more structured bits of information such as audio file format properties, time of recording, or geo-location information. Even though audio features could also be considered to be metadata, these are typically excluded from the definition.

A general concern is the consistency of the provided metadata, which is affected by the original design of the data model. Like in the case of file formats, some degree of freedom in terms of required metadata will allow to make an audio database more attractive to different users. For example, tags have become a very popular way to attach text labels to pieces of information without any predefined structure. Conversely, more structured and strict metadata layouts may benefit indexing and retrieval in some cases. In Sect. 10.3.1 a more detailed discussion is given regarding metadata fields and consistency.

In order to index and retrieve content based on text metadata, a full-text search engine is especially useful. Full-text search engines are specialized software programs that are useful for searching in text documents or textual representations of documents as metadata fields. The choice of indexing algorithms depends on the type of information stored in a database and the way it is to be retrieved. The availability of implementations will typically determine the choice of a given database program or library. Traditional relational databases typically rely on B+ trees for indexing. Other tree structures can be used for spatial queries as described in Sect. 10.4. While the plethora of available database software is beyond the scope of this chapter, it is useful to distinguish three main groups that are commonly helpful for large-scale applications. Traditional relational databases, also known as relational database management systems (RDBMS), are used via the structured query language (SQL) in many corporate and web applications. The more modern trend of *NoSQL* databases comprises a heterogeneous group including document databases, key-value stores, and graph databases. Some of these use text formats commonly used for audio metadata, such as XML or JSON. For information indexed using complex ontologies, specialized graph databases or triple-stores may be needed (see Sect. 10.5) [59].

10.2.3 Audio Features

Audio features or *descriptors* are numerical representations obtained through automatic analysis of audio, often attempting to capture some aspect of human perception. A large number of such descriptors have been developed over the

years through research in automatic speech recognition (ASR), music information retrieval (MIR), and environmental sound recognition (ESR), including sound scene and sound event analysis (see Chap. 4). The use of audio features has been historically different in each of these domains. For speech, features were used most of the time as a spectral representation of sound used to train discrete models (e.g., hidden Markov models) informed by human speech and language. For music, features have been traditionally organized in different levels, according to their proximity to music theory concepts. For environmental sound most of the time very generic features are used in order to recognize specific semantic categories. Creating a database of audio features involves feature extraction software. One example can be found in the Freesound Extractor,[7] which extracts a number of audio features using the Essentia [8] audio analysis library. The most important question when designing a database using audio features is to know which features are relevant to the expected type of content and use case. So far there are mainly two interaction paradigms that have been extensively researched for audio retrieval based on features: range queries can be used with descriptors that are understandable for humans. A simple example is finding pitched sounds within a given range of pitches. Query by example (often also called *similarity search*) refers to using an example sound to find similar sounds in the database (see Sect. 10.4). In the field of data-driven music creation a special case is to find sequences of shorter sounds in the database that are optimally close to an audio query. This technique has been named *musical mosaicing* [81] or *concatenative sound synthesis* [67].

10.3 Metadata-Based Sound Retrieval

Metadata is the most common way through which audio databases can be navigated and their content retrieved. In this section we describe in more detail the use of user-provided metadata for indexing audio content (Sect. 10.3.1) and explain some of the most common sound retrieval strategies based on metadata (Sect. 10.3.2). The code examples referenced along with the explanations build up from the examples referenced in the previous section.

10.3.1 Metadata for Audio Content

Content authors or editors typically provide metadata in the form of a number of *annotations* or *descriptions*. As described in Sect. 10.2, it is common for sound sharing platforms to rely on such user-provided metadata for audio indexing and retrieval. Nevertheless, the nature of content annotations may vary on each particular

[7] https://github.com/MTG/essentia/tree/master/src/examples/freesound.

sharing platform, and is highly dependent on the description strategy used in every particular site. Description strategies that look for the most uniform annotations can use forms with a number of predefined metadata fields with fixed responses. For each field, users will chose one of the available responses when describing a resource. For example, users might be asked to classify a sound effect by selecting a category from a fixed list of categories. However, these strategies lack flexibility when new resources are uploaded because their characteristics can be unexpected and not contemplated in the description form [27, 42, 71]. Other description strategies provide more flexibility by not limiting metadata fields to a specific set of responses. In that case, annotations typically consist of a textual description and a number of tags which are not restricted to a particular vocabulary.

Using tags as keywords for annotating resources has become standard practice in many online sharing sites of very different nature. Just to name a few examples, multimedia sharing sites like YouTube, Vimeo, Flickr, SoundCloud, Bandcamp, Last.fm, or Freesound[8] have content labeled using tags. However, despite the popularity of tagging systems and their successful implementation in many online sharing sites, there are a number of well-known problems which limit the possibilities of these functionalities [26]. These problems range from the use of different tags to refer to a single concept (synonymy) and the ambiguity in the meaning of certain tags (polysemy), to tag scarcity and typographical errors [24, 27]. Furthermore, the quality of the indexing, searching, and browsing functionalities enabled by tagging systems strongly relies on the coherence and comprehensiveness of the tags assigned to the resources. It is not only important that individual resources are properly tagged, but also that descriptions are consistent across the database. For that reason, it has been often discussed whether a tagging system, after a certain time of being in use, reaches a point of implicit consensus where the vocabulary converges to a certain set of tags and tagging conventions that are widely adopted by all users of the system [27, 58, 70, 74, 78]. According to these authors, the point of consensus may be reached because of imitation patterns and users' shared cultural knowledge. Reaching that point of consensus is desirable to improve indexing and overall sharing experience [26]. It is a common strategy in sharing platforms to use tag recommendation methods to help users during the description process [19, 24, 27, 44]. By using such methods, user annotations are expected to be more uniform and comprehensive, thus helping in reaching the aforementioned point of consensus.

Overall, the choice of using a flexible description strategy or a more strict one strongly depends on the nature of the data that needs to be collected. Flexible systems are a better fit for heterogeneous data, while strict systems can work well for cases in which the information to annotate is very well defined. In the case of audio material a mixed approach can be a good option, using well-defined metadata fields relating to aspects such as audio format (e.g., sample-rate, bit depth, number

[8] https://youtube.com, https://vimeo.com, https://flickr.com, https://soundcloud.com, https://bandcamp.com, https://last.fm, https://freesound.org.

of channels) or other recording properties (e.g., duration, date of recording, duration of recording, used microphone, etc.), and also using more flexible fields such as a textual description of the activities being recorded. In the specific case of sound events and sound scenes, it is important to put an emphasis on describing the sound sources that are captured in a recording (i.e., *what* produces the sound). Because of the potential variety of sound sources, tagging systems are typically appropriate for annotating that kind of information. For sound scenes, it is also desirable to attach annotations to particular regions of a recording, being able in this way to provide specific descriptions for different fragments of the scene.

10.3.2 Search and Discovery of Indexed Content

As previously mentioned, a common choice for indexing metadata is the use of a full-text search engine. In that case users typically introduce some search terms (i.e., words) as a query. The search engine then matches these terms with indexed metadata fields and returns a sorted list of results (sounds in our examples). For each sound in the index, the search engine will compute a relevance score based on how well the input terms match the information in the metadata fields and how relevant the matched terms are. The classic relevance score in information retrieval is based on calculating the relevance of a term with respect to a given document by the TF*IDF measure [72]. TF stands for "term frequency," and number of times the specific term appears in a document. IDF stands for "inverse document frequency," and represents the inverse of the number of documents in the index that contain the given term. The idea is that a given term will be relevant with respect to a document if it appears many times, but the relevance will be penalized if it also appears in many other documents. Using such a relevance function and given a number of input query terms, a global score can be computed by aggregating the relevance of each query term for the different metadata fields of each document in the index. Other common score functions include the BM25F, which is a variation of TF*IDF based on probabilistic information retrieval [57], and the PageRank algorithm [50], which calculates the relevance of a document based on the relevance of documents that point to it.

Besides the scoring functions for sorting results, search engines can include *query expansion* mechanisms which perform pre-processing of user queries before matching it with the index contents [14]. The idea behind query expansion is that the input terms provided by a user can be expanded with other relevant terms before matching with the index, potentially increasing the number of results. The way in which new terms are added to the query can be based on simple strategies such as using synonym lists or on more complex strategies such as the analysis of previous queries or the use of domain-specific knowledge (see Sect. 10.5). Search results may be further refined by allowing users to specify filtering criteria. In this way metadata fields that are not taken into account in the scoring function can be used in the search process to restrict the searchable space.

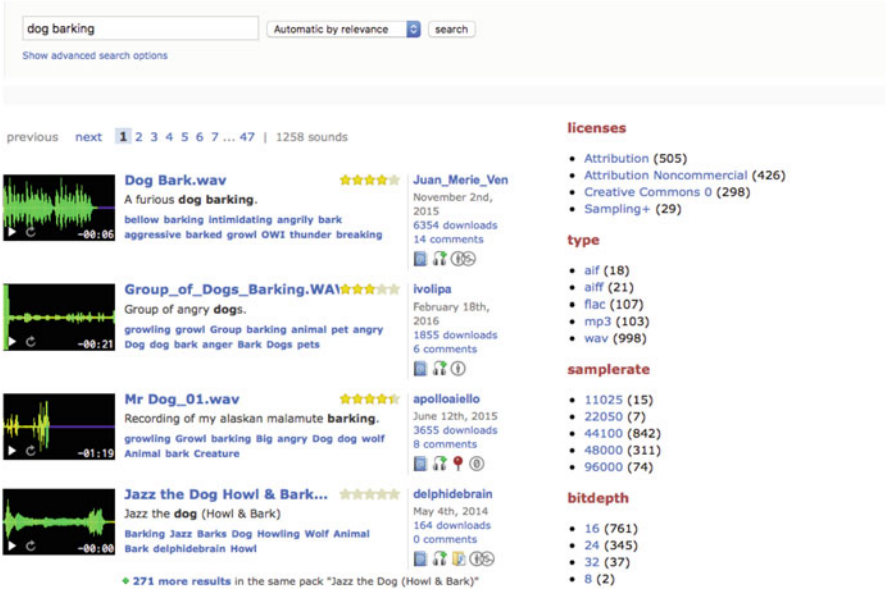

Fig. 10.2 Example of faceted search as implemented in Freesound. Facets for *licenses*, *type*, *samplerate*, and *bitdepth* metadata fields are displayed for the results of the query "dog barking." Clicking on these facets further filters the query results

Another typical retrieval strategy based on metadata is the use of *facets* when displaying search results. This is typically referred to as *faceted search* [76]. Given the results of a query, faceted search extends conventional search by dynamically summarizing the distribution of values in a number of information facets (i.e., metadata fields) and showing this information to users. In this way, users can use the information displayed in facets to further filter and update their queries (see Fig. 10.2). Faceted search has become increasingly popular in sharing platforms and provides a foundation for interactive information retrieval by allowing iterative results-informed query refinement.

Faceted search allows the discovery of the database beyond conventional search by providing users with a way to *navigate* the content (even without specifying initial query terms). A particularly successful faceted search application is the use of a *tag cloud* as a browsing interface. A tag cloud shows the most commonly used tags in a database with the size of each tag set proportional to its frequency of occurrence (Fig. 10.3) [33]. Users can typically navigate a collection by applying query filters based on the tags in the tag cloud, and for each new filter a new tag cloud can be computed and displayed only considering the filtered set of documents. Note that when new content is indexed in a database, the tag cloud can be automatically updated. Therefore tag clouds show an up to date overview of the contents of a database.

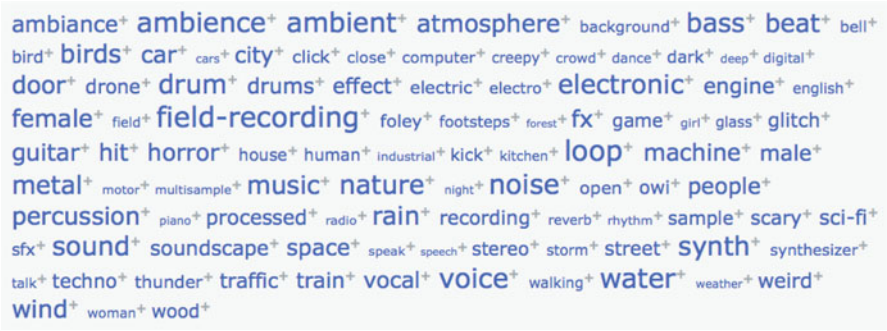

Fig. 10.3 Example tagcloud taken from Freesound (retrieved September 27th 2016)

Our provided code examples show a simple implementation of a number of metadata-based sound retrieval strategies. We provide a basic text search system which is configured to match input query terms with the information in *name*, *description*, and *tags* metadata fields. We also provide examples for filtering search results based on *duration* and sound *license* fields. Furthermore, our example code shows how to define facets and group search results based on these (again using *license* and *duration* metadata fields). Finally, we provide code to generate a tagcloud which summarizes the contents of an audio database by displaying its most important tags.

10.4 Audio-Based Sound Retrieval

Audio-based retrieval (also known as content-based retrieval) refers to the use of descriptors computed automatically from the waveform in order to find audio files in a database. The obvious advantage of using automatic descriptors is that it allows indexing content when no labels or metadata are available. Manual labelling may require significant amounts of work, which could be avoided by using automatic analysis. On the other hand, currently available descriptors do not always bear an intuitive meaning for non-expert (or even expert) users. Often they are used in conjunction with machine learning algorithms in order to obtain meaningful labels (i.e., typically through segmentation and/or classification). In this section we will review basic retrieval techniques using audio descriptors. More details on specific descriptors can be found in Chap. 4.

10.4.1 Audio Features

One of the most critical aspects for audio-based retrieval is devising a set of features that are relevant for the application. The standard representation of audio as a waveform in the time domain allows for visual inspection and graphical editing, but conveys only limited information (i.e., amplitude) about the sound. Since time-frequency transforms such as the Short-time Fourier Transform (STFT) became affordable and ubiquitous in audio processing, the spectrogram has been used as a more intuitive representation. Most descriptors used for content-based indexing of audio data are related to some time-frequency transform, and attempt to describe some aspect of it that is relevant to some application or is intuitively useful. An obvious example would be detecting the pitch for harmonic signals. A large number of software libraries are available for feature extraction, mainly in the context of MIR research [8, 10, 11, 36, 45, 46, 54, 77]. It is common in MIR to distinguish low-level features (i.e., closer to the spectral representation but with little intuitive meaning) and higher-level features related with musical concepts. Environmental audio can be seen as a very general case where it is still possible to find music (e.g., street music, the radio in a car) and very likely pitched sounds such as human or other animal vocalizations, or human-originated sounds like alarms. Finding a generic representation is not straightforward and in most cases it may depend on the application. Low-level features can still be used for most kinds of sounds. A popular set of low-level features was compiled in the definition of the MPEG-7 standard [55]. As generic descriptors, Mel-frequency cepstral coefficients (MFCCs) are still very widely used, like in the case of music and speech. More recently, deep learning architectures make it possible to automatically learn the required representations from spectral frames [38], or even from raw audio waveforms [30]. A strategy for supporting many potential applications is computing a comprehensive set of features. This is the strategy adopted in the aforementioned Freesound Extractor (Sect. 10.2.3), and does not bear a large computational cost in respect to the whole operation.

10.4.2 Feature Space

Audio features are typically aggregated along time in order to represent each audio file as a single vector. This process may depend on the content of the file: for sound events, the temporal evolution of spectral features may be taken into account, while for sound scenes global statistics may suffice. Statistics can also be computed from delta features, producing higher-dimensional vectors that capture some of the short-term temporal evolution. The idea of ignoring the actual order of spectral features and computing statistics has been dubbed the “bag of frames” approach [3], by analogy to the “bag of words” model in text retrieval. Feature vectors can also be extracted from events detected in long recordings.

The set of selected features can then be seen to form a vector space, defined by some distance metric between feature vectors. Typical choices are distance metrics associated with p-norms, or cosine distance for large dimensions. Feature vectors represent audio documents, and can be added as records to any conventional database software. On the other hand, it is very convenient if the collection can be kept in computer memory. As an example, for an arbitrary distance metric, a naive query-by-example approach would require traversing all the vectors and compute the distance to the query vector (see Sect. 10.4.4). Since there may be some redundancy in the feature space (for example, when a very large set of features is computed), dimensionality reduction algorithms (e.g., principal component analysis) can be helpful. Another common practice is to quantize features into typically sparse spaces, e.g., using data clustering [37, 64]. However, keeping the original features allows defining distance metrics over meaningful subsets. This is especially useful if many different kinds of sounds are mixed, so different feature sets can be used by different applications.

10.4.3 Descriptor-Based Queries

Using descriptors directly in user interfaces is not very common, especially when dealing with general audio, since low-level features often lack an intuitive meaning. Moreover, the distribution of features across the database must be taken into account in order to find sensible values.[9] Queries based on hand-picked descriptor targets or ranges are still possible for some low-level features. An intuitive example is a rough division of the power spectrum in a few frequency bands (as is used frequently in audio production and mixing, e.g., *low, mid-low, mid-high, high*). A non-specialist user could use these bands to select sounds where most of the energy is in the higher frequencies (e.g., selecting bird sounds), or in the lower frequencies (e.g., to find sounds of passing cars). Descriptor-based queries can also be devised by an expert and presented as discrete choices to the user. An even simpler example is using the spectral centroid (see the code examples provided in the book website). Since audio descriptors are often floating point numbers, a common strategy is to specify a query range for a given descriptor or for a set of them. However, more complex queries can be made using specialized languages like SQL or other database query languages. In concatenative sound synthesis research, it was also common to use audio features as axes of interactive scatterplots of audio collections [68]. The exploration can also be driven by gestures [17].

[9]For this reason, histograms are provided as part of the documentation of the Freesound API: https://www.freesound.org/docs/api/analysis_docs.html.

10.4.4 Query by Example

Query by example (Qbe) refers to a kind of content-based retrieval technique where the target descriptors are extracted from examples, so users do not need to know about them. The idea of QbE has become widespread for music (e.g., singing a melody, or recording a fragment of audio to retrieve a song title and author). The same idea has been tried for general audio, where users try to imitate the sound they are looking for [6, 15] (see [62] for an example implementation using Freesound). For Qbe to work, the query example must be analyzed by the same extractor program that has been used to create the database, since small differences in parameters can lead to complete different ranges for descriptors. Again it is also possible to devise ideal targets by experts, for example, specifying a given value for pitch. Similarity queries will normally return a list of records ordered by similarity to the target. This can be seen as a nearest-neighbors search in the vector space. Common algorithms for nearest-neighbor search are KD-trees, R-trees, and ball trees. Relaxation of the problem to approximate nearest neighbors (where the returned vector is not guaranteed to be the nearest one to the target) may perform several orders of magnitude faster, and thus is indicated for large data sets and high-dimensional spaces.

The accompanying code examples include a simple implementation of both descriptor and nearest-neighbors queries. We first query the database to get general statistics about extracted audio features which allows to observe their distribution. We focus on the spectral centroid feature, which is a rough indicator of how energy is distributed across the frequency spectrum. The database is then queried for sounds with a centroid below 50 Hz, which returns a roar sound with low frequencies, and a nearest-neighbors ball tree algorithm is used to find the ten nearest neighbors using MFCC statistics. This returns a list with mostly roar sounds and also some dog barks.

10.4.5 Audio Fingerprints and Thumbnails

An audio fingerprint summarizes an audio recording into a small description (typically an alphanumeric string) that is ideally unique. This is used to identify copies of the same recording, since applying the same algorithm should result in the same fingerprint. Systems are often designed to be robust to some distortions, such as ambient noise or reverberation, but in general fingerprinting only works for copies of the same recording (i.e., the same waveform), as opposed to multiple recordings of similar sounds, such as a given utterance or a musical piece. The techniques used for fingerprinting are generally based on feature extraction as described above, typically with a more complex step of summarization of the time series of audio features. For example, vector quantization or hidden Markov models can be used in order to obtain a short and hopefully unique representation (see [12] for a review).

Fingerprinting can be seen as a special case of hashing. Generic hashing algorithms can be used to find and prevent exact duplicates of the same file in the database (e.g., MD5 is used in Freesound). Audio fingerprinting was developed mostly for music but it can be used also for general audio content (e.g., commercial monitoring [31]). Fingerprinting has also been used for identifying room ambiance [4], so it could be used to group recordings from the same location in an audio database. Finally, fingerprinting-like hashing has been proposed also as experimental indexing for creative applications [16].

While fingerprints can be seen as summaries that uniquely identify audio recordings for machines, audio thumbnails can be seen are sound fragments that humans can use as previews to identify and remember recordings. Such previews are required for browsing audio databases or analyzing search results. For musical audio, the most common approach is identifying frequently repeated passages[2]. Contrastingly, for environmental sound, particularly for long recordings of sound scenes, it is more useful to apply some detection strategy in order to find salient events [82].

10.5 Further Approaches to Sound Retrieval

In the previous sections we have introduced standard sound retrieval strategies based either on metadata or on audio information. In this section we introduce some advanced strategies which are not as common as those introduced in the previous sections but are also very relevant for sound retrieval.

If we have a closer look at the metadata-based strategies described in Sect. 10.3, we will see that none of them are in fact particular or restricted to the sound sharing domain. In other words, no knowledge specific to the audio domain is used for any of the scoring functions, faceting or tagcloud examples shown above. The inclusion of domain-specific knowledge is therefore something that can be considered for enhancing sound retrieval strategies [48]. A simple form of domain-specific knowledge that is relevant in sound retrieval are, for example, taxonomic classifications of sound events as seen in Chap. 7. Such taxonomies can be used at different stages of the information retrieval process. For example, a taxonomy can be used to perform domain-specific query expansion [7] and increase in this way the recall of search results. Taxonomies can be used to group search results in specific concepts and present them accordingly [35].

Another more complex form of domain-specific knowledge is that represented by ontologies [18]. Ontologies provide, for a given domain, an unambiguous formalization of its concepts, entities, and their relations. Besides the work by Nakatami and Okuno [49] in which an ontology for sound is provided, the use of ontologies has not been much explored in the field of sound scene analysis. Typically, simpler forms for representing structured domain-specific knowledge are used as exemplified by Gaver's *map* of everyday sounds [22] and the recent Urban Sound Taxonomy [65]. Nevertheless, one advantage of using ontologies is that

content can be annotated with labels which feature a very specific semantic meaning. Hence, where tagging systems feature free-form textual labels with no predefined semantic meaning, ontologies feature detailed concept hierarchies interlinked with semantically meaningful relations. The accurateness and rigidity of ontologies is often opposed to the flexibility and ambiguity of tagging systems, but these can also be complementary [40, 47]. One common approach in this direction is the mapping of user-provided tags with specific concepts of an ontology. This allows tackling typical synonymy and ambiguity problems of tagging systems, but requires methods for automatically matching tags with concepts of the ontology [1, 53]. Ontologies can also be used in a sound retrieval context for optimizing sound annotations provided by users. For example, an ontology that embeds information about types of sounds and their relevant characteristics can be used during an annotation process to suggest users to provide annotations about particularly relevant information facets [19].

In the context of online sharing platforms in which users contribute and consume audio content we can also think of retrieval strategies that take advantage of user behavior information. The most prominent example of this type of retrieval strategies are recommendation systems [56]. Recommendation systems can be defined in different ways, but in the context of sound sharing a common application is the recommendation of potentially relevant sounds for a user given previous sounds that the user has retrieved. This problem is typically approached using collaborative filtering techniques [66]. Such techniques are able to recommend items to a user based on items that other similar users interacted with in the past. For example, if user A has downloaded sounds 1, 2, and 3 and user B has downloaded sounds 1 and 3; the recommendation system could recommend sound 2 to user B. Collaborative filtering techniques can be used for discovery through sound recommendation in a way that evolves along with users' activity. The more users interact with sounds, the more information the system has to perform better informed recommendations.

With respect to content-based approaches, sound retrieval often benefits from machine learning approaches that map low-level features to more intuitive representations. Machine learning algorithms for retrieval can be generally classified between supervised and unsupervised. As shown elsewhere in this book, supervised learning approaches have applications in acoustic event classification (Chap. 5), annotation (Chaps. 6 and 7), and detection (Chap. 8) among others. Its application to sound retrieval often implies dealing with scalability both in terms of computational cost and concept generalization. For example, the statistics of a large set of features have been used along with K-NN classifiers for large-scale applications [13]. Another example approach for tackling concept generality is combining classification based on existing taxonomies with free text queries [63].

Unsupervised machine learning methods are well suited for browsing and discovery, typically by using clustering to discover underlying groupings in the database. The most common approach is to map a collection of sounds to two-dimensional space. Self-organizing maps (SOM) were used in a number of efforts for this purpose [9, 28, 51, 52]. Another approach is using graph layout algorithms

for visualizing nearest-neighbor graphs [69]. Nearest-neighbors graphs can also be clustered using graph clustering to provide unsupervised hierarchical organizations [60]. These browsing and discovery mechanisms do not require a textual query to initially filter content, but can be effectively used in combination with textual queries or supervised approaches to provide focused unsupervised interfaces. These are especially useful when many kinds of sounds are mixed in the database. Two recent examples of such interfaces are shown in Fig. 10.4. The first example, *Floop*[10] (top), is an experimental system for graphically browsing rhythmic sounds. Rhythimc sounds are detected and classified in an unsupervised fashion using the Beat spectrum [21], a classic descriptor that estimates the main periodicities in any kind of sound [61]. A force-directed graph layout is used to organize a nearest-neighbors graph (computed from content-based timbral similarity) for a subset of sounds that share the same repetitive period and therefore can be played rhythmically together. The second example (Fig. 10.4, bottom) shows an interface for exploring an audio database in which the search results of a given textual query are organized according to timbral similarity. Similarly to previous work by Heise et al. [28], results are displayed in a map that can be explored and in which sounds can be listened to.[11] The map is computed using the t-SNE [41] dimensionality reduction technique on MFCC audio descriptors. Closer sounds in the map have closer timbral similarity. In this way search results are placed in different parts of the map and users can browse content by combining the semantic properties specified via the text search and the timbre characteristics represented in the map of results.

10.6 Conclusions

The increasing popularity of sound sharing and the growing capabilities of portable recording devices, including mobile phones, pose new challenges for sound retrieval techniques. Sound retrieval is therefore a timely topic which will probably attract more and more attention in the coming years.

In this chapter we have introduced the most important concepts related to sound sharing and retrieval and have described the different ways in which content from an audio database can be indexed, searched, and navigated. We have illustrated the different parts of a sound retrieval system with code examples showing the creation of an audio database and the addition of both metadata-based and audio-based retrieval functionalities. This code can easily be extended to incorporate more features and further experiment with sound retrieval techniques.

The introduction given in this chapter should be understood as a starting point for future developments. In particular, promising research directions such as the use of deep learning for the annotation of audio content and the use of domain-specific ontologies for structuring metadata are likely to play an important role in future sound sharing and retrieval systems.

[10] https://labs.freesound.org/floop/.

[11] https://ffont.github.io/freesound-explorer/.

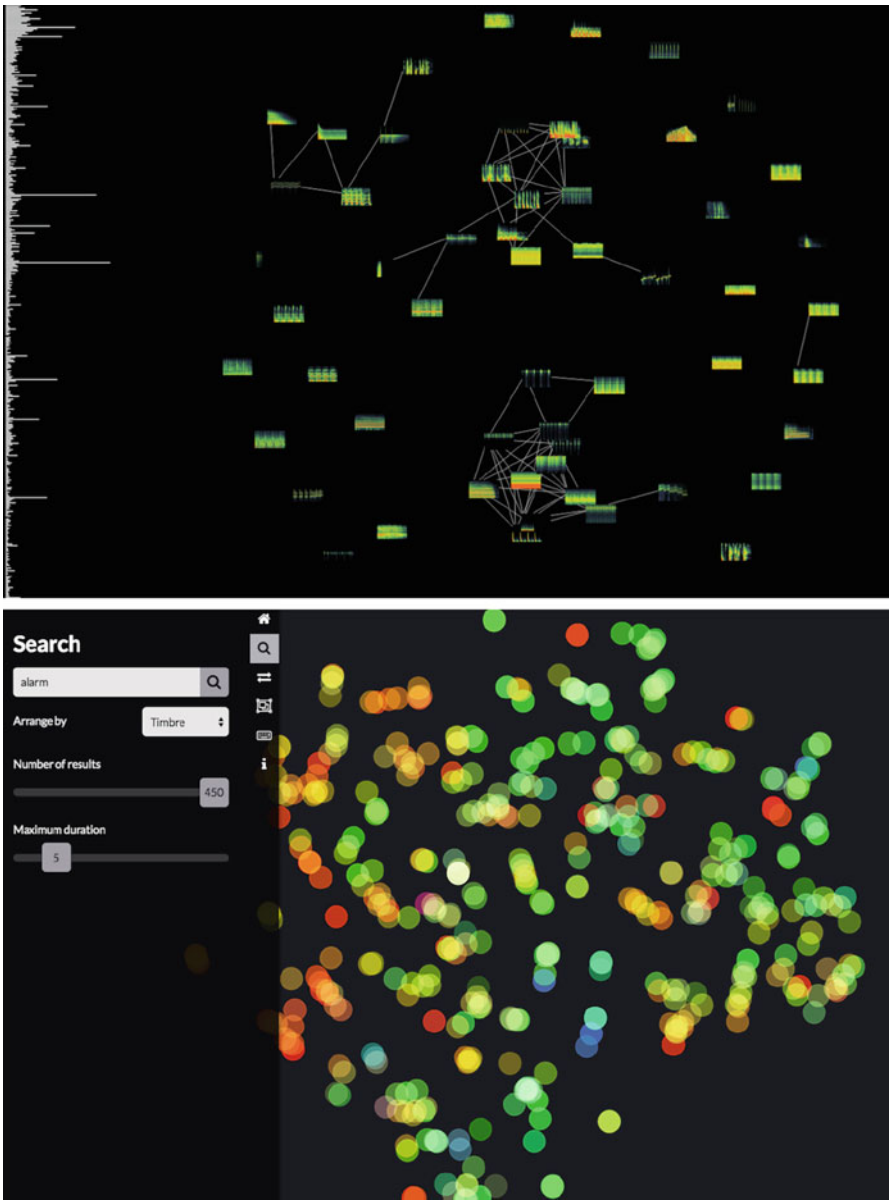

Fig. 10.4 Examples of two interfaces that exploit audio-features information to display sounds in a two-dimensional space. The *top figure* shows Floop, an interface that organizes rhythmic sounds according to an estimated periodicity. At the *left*, an interactive histogram indicates the number of sounds available for each rhythmic cycle duration. When clicking on one of the bars, the corresponding sounds are displayed and organized by timbral similarity. The *bottom figure* shows a map in which sounds are organized by timbral similarity. Users can introduce some textual query terms which are used to query Freesound and the results are displayed in a map where each circle represents a sound. Closer circles in the map tend to sound more similar than circles which are farther away

References

1. Angeletou, S., Sabou, M., Motta, E.: Semantically enriching folksonomies with FLOR. In: Proceedings of the European Semantic Web Conference (ESWC) (2008)
2. Aucouturier, J.J., Sandler, M.: Finding repeating patterns in acoustic musical signals: applications for audio thumbnailing. In: Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio. Audio Engineering Society, New York (2002)
3. Aucouturier, J.J., Defreville, B., Pachet, F.: The bag-of-frames approach to audio pattern recognition: a sufficient model for urban soundscapes but not for polyphonic music. J. Acoust. Soc. Am. **122**(2), 881–891 (2007)
4. Azizyan, M., Constandache, I., Roy Choudhury, R.: Surroundsense: mobile phone localization via ambience fingerprinting. In: Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 261–272. ACM, New York (2009)
5. Bischoff, K., Firan, C.S., Nejdl, W., Paiu, R.: Can all tags be used for search? In: Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), pp. 193–202 (2008)
6. Blancas, D.S., Janer, J.: Sound retrieval from voice imitation queries in collaborative databases. In: Proceedings of the AES Conference on Semantic Audio. Audio Engineering Society, New York (2014)
7. Bodner, R.C., Song, F.: Knowledge-based approaches to query expansion in information retrieval. In: Proceedings of the Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI), pp. 146–158. Springer, New York (1996)
8. Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J.R., Serra, X.: Essentia: an audio analysis library for music information retrieval. In: Proceedings of the International Music Information Retrieval Conference (ISMIR), pp. 493–498 (2013)
9. Brazil, E., Fernstroem, M., Tzanetakis, G., Cook, P.: Enhancing sonic browsing using audio information retrieval. In: Proceedings of the International Conference on Auditory Display (ICAD), Kyoto, pp. 132–135 (2002)
10. Brossier, P.M.: The aubio library at MIREX 2006. In: Proceedings of the Music Information Retrieval Evaluation Exchange (MIREX), p. 1 (2006)
11. Bullock, J., Conservatoire, U.: Libxtract: a lightweight library for audio feature extraction. In: Proceedings of the International Computer Music Conference (ICMC), pp. 22–28 (2007)
12. Cano, P., Batlle, E., Kalker, T., Haitsma, J.: A review of audio fingerprinting. J. VLSI Signal Process. Syst. **41**(3), 271–284 (2005)
13. Cano, P., Koppenberger, M., Wack, N.: An industrial-strength content-based music recommendation system. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, p. 673 (2005)
14. Carpineto, C., Romano, G.: A survey of automatic query expansion in information retrieval. ACM Comput. Surv. **44**(1), 1:1–1:50 (2012)
15. Cartwright, M., Pardo, B.: Vocalsketch: Vocally imitating audio concepts. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI), pp. 43–46. ACM, New York (2015)
16. Casey, M.A.: Acoustic lexemes for organizing internet audio. Contemp. Music Rev. **24**(6), 489–508 (2005)
17. Comajuncosas, J.M., Barrachina, A., O'Connell, J., Guaus, E.: Nuvolet: 3d gesture-driven collaborative audio mosaicing. In: Proceedings of the New Interfaces for Musical Expression Conference (NIME), pp. 252–255 (2011)
18. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2001)
19. Font, F.: Tag Recommendation using Folksonomy information for online sound sharing platforms. Ph.D. thesis, Universitat Pompeu Fabra (2015)

20. Foote, J.: An overview of audio information retrieval. Multimed. Syst. **7**(1), 2–10 (1999)
21. Foote, J., Uchihashi, S.: The beat spectrum: a new approach to rhythm analysis. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) (2001)
22. Gaver, W.W.: What in the world do we hear?: An ecological approach to auditory event perception. Ecol. Psychol. **5**(1), 1–29 (1993)
23. Ghias, A., Logan, J., Chamberlin, D., Smith, B.C.: Query by humming: musical information retrieval in an audio database. In: Proceedings of the ACM International Conference on Multimedia (MM), pp. 231–236. ACM, New York (1995)
24. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
25. Guo, G., Li, S.Z.: Content-based audio classification and retrieval by support vector machines. IEEE Trans. Neural Netw. **14**(1), 209–215 (2003)
26. Guy, M., Tonkin, E.: Folksonomies: tidying up tags? D-Lib Mag. **12**(1) (2006)
27. Halpin, H., Robu, V., Shepard, H.: The dynamics and semantics of collaborative tagging. In: Proceedings of the Semantic Authoring and Annotation Workshop (SAAW), pp. 1–21 (2006)
28. Heise, S., Hlatky, M., Loviscach, J.: Soundtorch: quick browsing in large audio collections. In: Proceedings of the 125th AES Convention. Audio Engineering Society (2008)
29. Huber, D.M., Runstein, R.E.: Modern Recording Techniques. Taylor & Francis, London (2013)
30. Jaitly, N., Hinton, G.: Learning a better representation of speech soundwaves using restricted boltzmann machines. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5884–5887. IEEE, New York (2011)
31. Jang, D., Jin, M., Lee, J.S., Lee, S., Lee, S., Seo, J.S., Yoo, C.D.: Automatic commercial monitoring for TV broadcasting using audio fingerprinting. In: Proceedings of the AES Conference on Audio for Mobile and Handheld Devices. Audio Engineering Society, New York (2006)
32. Jeffries, A.: The man behind Flickr on making the service 'awesome again' (2013). http://www.theverge.com/2013/3/20/4121574/flickr-chief-markus-spiering-talks-photos-and-marissa-mayer. Last accessed 15 Nov 2016
33. Kaser, O., Lemire, D.: Tag-cloud drawing: algorithms for cloud visualization. In: Proceedings of the International World Wide Web Conference (WWW) (2007)
34. Krumm, J., Davies, N., Narayanaswami, C.: User-generated content. IEEE Pervasive Comput. 10–11 (2008)
35. Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R.: A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 658–665. ACM, New York (2004)
36. Lartillot, O., Toiviainen, P., Eerola, T.: A MATLAB toolbox for music information retrieval. In: Proceedings of the Data analysis, Machine Learning and Applications Conference, pp. 261–268. Springer, Berlin, Heidelberg (2008)
37. Lee, K., Ellis, D.P.W.: Audio-based semantic concept classification for consumer video. IEEE Audio Speech Language Process. **18**(6), 1406–1416 (2010)
38. Lee, H., Pham, P., Largman, Y., Ng, A.Y.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Proceedings of the Neural Information Processing Systems (NIPS), pp. 1096–1104 (2009)
39. Lessing, L.: Remix: Making Art and Commerce Thrive in the Hybrid Economy. Penguin Press, Harmondsworth (2008)
40. Limpens, F., Gandon, F.L., Buffa, M.: Linking folksonomies and ontologies for supporting knowledge sharing: a state of the art. Tech. rep., Institut National de Recherche en Informatique et Automatique (INRIA) (2009)
41. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
42. Macgregor, G., Mcculloch, E.: Collaborative tagging as a knowledge organisation and resource discovery tool. Libr. Rev. **55**(5), 291–300 (2006)

43. Marcell, M.M., Borella, D., Greene, M., Kerr, E., Rogers, S.: Confrontation naming of environmental sounds. J. Clin. Exp. Neuropsychol. **22**(6), 830–864 (2000)
44. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, Tagging paper, taxonomy, Flickr, academic article, to read. In: Proceedings of the ACM Conference on Hypertext and Hypermedia (Hypertext), pp. 31–41 (2006)
45. Mathieu, B., Essid, S., Fillon, T., Prado, J., Richard, G.: YAAFE, an easy to use and efficient audio feature extraction software. In: Proceedings of the International Music Information Retrieval Conference (ISMIR) (2010)
46. McFee, B., Raffel, C., Liang, D.: librosa: Audio and music signal analysis in python. In: Proceedings of the Python in Science Conference (SciPy) (2015)
47. Mika, P.: Ontologies are us: a unified model of social networks and semantics. Web Semant.: Sci. Serv. Agents World Wide Web **5**(1), 5–15 (2007)
48. Nagypál, G.: Improving information retrieval effectiveness by using domain knowledge stored in ontologies. In: Proceedings of the OTM Confederated International Conferences - On the Move to Meaningful Internet Systems, pp. 780–789. Springer, New York (2005)
49. Nakatani, T., Okuno, H.G.: Sound ontology for computational auditory scene analysis. In: Proceedings of the Innovative Applications of Artificial Intelligence Conference (IAAI), pp. 1004–1010 (1998)
50. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Tech. rep., Stanford InfoLab (1999)
51. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. In: Proceedings of the ACM International Conference on Multimedia (MM), pp. 570–579. ACM, New York (2002)
52. Pampalk, E., Hlavac, P., Herrera, P.: Hierarchical organization and visualization of drum sample libraries. In: Proceedings of the International Conference on Digital Audio Effects (DAFx), Naples, pp. 378–383 (2004)
53. Passant, A., Laublet, P., Breslin, J.G., Decker, S.: A URI is worth a thousand tags: from tagging to linked data with MOAT. In: Semantic Services, Interoperability and Web Applications: Emerging Concepts, p. 279 (2011)
54. Pedregosa, F., Varoquaux, G.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
55. Peeters, G.: A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM (2004)
56. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)
57. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)
58. Robu, V., Halpin, H., Shepherd, H.: Emergence of consensus and shared vocabularies in collaborative tagging systems. ACM Trans. Web **3**(4) (2009)
59. Rohloff, K., Dean, M., Emmons, I., Ryder, D., Sumner, J.: An evaluation of triple-store technologies for large data stores. In: Proceedings of the OTM Confederated International Conferences - On the Move to Meaningful Internet Systems, pp. 1105–1114. Springer, New York (2007)
60. Roma, G.: Algorithms and representations for supporting online music creation with large-scale audio databases. Ph.D. thesis, Universitat Pompeu Fabra (2015)
61. Roma, G., Serra, X.: Music performance by discovering community loops. In: Proceedings of the Web Audio Conference (WAC), Paris (2015)
62. Roma, G., Serra, X.: Querying Freesound with a microphone. In: Proceedings of the Web Audio Conference (WAC) (2015)
63. Roma, G., Janer, J., Kersten, S., Schirosa, M., Herrera, P., Serra, X.: Ecological acoustics perspective for content-based retrieval of environmental sounds. EURASIP J. Audio Speech Music Process. **2010**, 1–11 (2010)
64. Salamon, J., Bello, J.P.: Feature learning with deep scattering for urban sound analysis. In: Signal Processing Conference (EUSIPCO), 2015 23rd European, pp. 724–728. IEEE, New York (2015)

65. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: Proceedings of the ACM International Conference on Multimedia (MM), pp. 1041–1044 (2014)
66. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 285–295. ACM, New York (2001)
67. Schwarz, D.: Corpus-based concatenative synthesis. IEEE Signal Process. Mag. **24**(2), 92–104 (2007)
68. Schwarz, D., Cahen, R., Britton, S.: Principles and applications of interactive corpus-based concatenative synthesis. J. d'Informatique Musicale 1 (2008)
69. Schwarz, D., Schnell, N.: Sound search by content-based navigation in large databases. In: Proceedings of the Sound and Music Computing Conference (SMC), p. 1 (2009)
70. Sen, S., Lam, S., Rashid, A., Cosley, D.: Tagging, communities, vocabulary, evolution. In: Proceedings of the Conference on Community Supported Cooperative Work (CSCW), pp. 181–190 (2006)
71. Shirky, C.: Ontology is overrated: Categories, links, and tags (2005). http://www.shirky.com/writings/ontology_overrated.html. Last accessed 15 Nov 2016
72. Singhal, A.: Modern information retrieval: a brief overview. Bull. IEEE Comput. Soc. Tech. Commun. Data Eng. **24**(4), 35–43 (2001)
73. Smith, T.: The social media revolution. Int. J. Mark. Res. **51**(4), 559–561 (2009)
74. Sood, S.C., Owsley, S.H., Hammond, K.J., Birnbaum, L.: TagAssist: automatic tag suggestion for blog posts. In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM), pp. 1–8 (2007)
75. The YouTube Team: Here's to eight great years (2013). http://youtube-global.blogspot.com/2013/05/heres-to-eight-great-years.html. Last accessed 15 Nov 2016
76. Tunkelang, D.: Faceted search. Synth. Lect. Inf. Concepts Retr. Serv. **1**(1), 1–80 (2009)
77. Tzanetakis, G., Cook, P.: Marsyas: a framework for audio analysis. Organised Sound **4**, 169–175 (2000)
78. Wagner, C., Strohmaier, M., Huberman, B.: Semantic stability and implicit consensus in social tagging streams. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 735–746 (2014)
79. Wahlforss, A.L.: SoundCloud is 5! (2013). http://blog.soundcloud.com/2013/11/13/soundcloud-is-5/. Last accessed 15 Nov 2016
80. Wikipedia: Remix culture (2014). https://en.wikipedia.org/wiki/Remix_culture. Last accessed 15 Nov 2016
81. Zils, A., Pachet, F.: Musical mosaicing. In: Proceedings of the International Conference on Digital Audio Effects (DAFx), p. 135 (2001)
82. Zlatintsi, A., Maragos, P., Potamianos, A., Evangelopoulos, G.: A saliency-based approach to audio event detection and summarization. In: Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, pp. 1294–1298. IEEE, New York (2012)