Searching and Classifying Affinities in a Web Music Collection

Nicola Orio

Department of Cultural Heritage, University of Padua, Italy
nicola.orio@unipd.it

Abstract. Online music libraries available on the Web contain a large amount of audio content that is usually the result of digitization of analogue recordings or the direct acquisition of digital sources. The acquisition process is carried out by several persons and may last a number of years, thus it is likely that the same or similar audio content is present in different versions. This paper describes a number of possible similarities, which are called *affinities*, and presents a methodology to detect the kind of affinity from the automatic analysis and matching of the audio content.

1 Introduction

The automatic detection of duplicates and near duplicates of textual documents has become an important research trend after the development of the Web [3]. In fact the same textual information may be contained, with minor modifications, in several web pages maintained by different organizations or individuals. One of the reasons why there exists a large number of near duplicate pages can be tracked back to a general tendency on the Web to underestimate the importante of copyright. And in fact, near duplicate identification has also important applications in patent analysis and in plagiarism detection [6].

With the increasing availability of multimedia content on the Web and in cloud services duplicate detection is gaining relevance also for media other than text, in particular to help managing large video collections [9] and to improve image retrieval tasks [2]. Most of these approaches are based on the concept of *fingerprinting* as a way to reduce the very high dimensionality of the problem. The basic idea of fingerprinting is that multimedia objects can be represented by a compact array of features, with a size orders of magnitude smaller than the original object, allowing feature indexing and in general faster processing. Moreover, a robust fingerprinting algorithm is able to extract features that are mostly related to human perception, in order to identify duplicates of a given multimedia object even when some post-processing has been applied.

The approach can be applied also to the music domain, and in fact acoustic fingerprinting is a well-known technique commercially exploited for music identification, which is at the basis of *Shazam!*, one of the most popular music services on the Internet [14], and of many others systems, such as the *MusicID* software patented by Gracenote [4] and the *AudioID* software used by MusicBrainz [8] based on an application of computer vision [7].

However, detection of duplicates and near duplicates has been relatively less investigated in the case of music perhaps considering it a marginal problem in comparison with audio identification. A focus on remixing, which is one of the reasons why music near duplicates exist, has been given in [1] where Locality Sensitive Hashing has been applied as an alternative to audio fingerprinting. An interesting approach [12] proposes to model the processing operators that possibly create music duplicates and near duplicates.

Although it does not apply to the test collection used in this work, in many cases near duplicates are created ad-hoc to dodge digital rights management software and publish copyrighted material on the web [10]. Yet, in the music domain most of duplicates and near duplicates exist as a natural process of artistic creation, which is intrinsically based on resemblance and differentiation with existing music, often using already published tracks as the basis to create new music. This paper focuses on this latter problem, the detection and classification of *affinities* between music tracks in a music digital library that is the basis for an online web service of music delivery.

2 Affinities in a Music Collection

The goal of the project described in this paper is to improve the access of large music collections, as the one available from music web services, by detecting variants of the stored songs and by classifying the kind of variant. The results can be applied both to large web collections, where digital objects can be provided by the end users and thus there is basically no control on the inclusion of new files in the existing collection, and to audio digital libraries, where management can be improved by the detection of content similarities. The objectives, for both domains, can be summarized as follows:

1. Duplicate removal helps saving storage space; although the increasing number of cloud services reduced its cost, storage is still a relevant cost for institutions.
2. Near duplicate detection can highlight inconsistencies in metadata information, which is the typical case when content is uploaded by the end users; moreover, detection can be carried out while new content is uploaded thus, in case near duplicates are already present, the use can be suggest with suitable metadata.
3. It has been shown that metadata insertion is an error prone process even in the case of digitization campaigns for music digital libraries [11], because usually digitization is carried out as a separate process in respect to metadata creation; the identification of near duplicates can be used to discover the presence of errors in the cataloguing process.
4. The content-based music search engine of the digital library should be aware of the presence of duplicate material; similarity matches tend to cluster around duplicates of a given track, possibly hiding additional relevant tracks.

5. The presence of subtle differences between tracks may be of interest for musicologists, musicians and eventually for the simple music fans; alternate takes of a given composition or different live versions of a studio recording are likely to be presents in the collection and be both relevant for the final user.
6. Music composition is increasingly a collaborative process, where the final product is often the result of manipulation of existing material that is remixed, looped, sampled, and so on; the possibility to track this process, which goes beyond the mere identification of the new track, can improve music enjoyment and partially guarantee correct attribution to different authors.

This paper presents a research carried out in collaboration with the staff of a music digital library which is the basis of a web service for online music broadcast and delivery. The methods have been developed to address the real needs of the music experts who created and manage the music collection. Although it addresses the specific needs of a single web service, it is expected that the methodology can be extended also to other similar collections and, possibly, to social networks where content is directly provided by end users..

3 The Test Collection

The music collection used to train the model and run the tests contains more than 350,000 audio tracks in MP3 format for an estimate global duration of about 20,000 hours. The collection has been created in more than ten years by a group of music experts, starting from commercially available CDs that have been individually bought and converted in MP3 format. Descriptive metadata are managed by a DBMS while audio tracks are maintained by an external storage. For this experiment, the owner of the collection granted access to a limited amount of cataloguing information – basically title, authors and main performer – and full access to the MP3 content. The collection focuses on pop and rock genres, with less than 10% of the tracks belonging to classical, jazz and other repertoires. Clearly the used collection is orders of magnitude smaller that the one of popular web services, such as Spotify of Last.fm, but we considered it large enough to obtain significant results.

Since popular songs are likely to be included in different CD editions – first release, remastering, best of, compilations – a certain redundancy was expected with a number of duplicates inside the collection. These can be, as it has been shown by the initial results, *exact duplicates* when the same audio source was present in different CDs, and *near duplicates* when different takes of the same song have been published or when remastering heavily affected the audio content. Because of the long time span required to create the collection, a number of different tools has been used for MP3 ripping, resulting in a different quality of the lossy compression and thus in audible differences between songs, that thus become *near duplicates* a well. It has also to be considered that a number of different persons was involved in the cataloguing process, with potential inconsistencies in the metadata creation that make metadata not completely reliable.

Being used as the source material for the creation of the soundtrack of TV programs of a major Italian broadcaster, the collection contains also the result of post-processing of the original tracks. Hence the collection includes also what can be called *far duplicates*. In this context, far duplicates are considered two tracks that share a consistent part of audio content like in the case of remixing of song with additional instruments, loops used as the basis of new songs, mashups using more than one audio source and different montages of the same audio material. We define all the kind of duplicates – exact, near and far – with the general term *affinities*. The typology of affinity thus depends on a number of factors: the amount of audio material that is shared between two songs, the acoustic differences of the same source due to post-processing and re-mixing, the presence of different editing.

All the tracks in the collection were already fingerprinted because an audio identification engine was already in place as the result of a previous project. The audio fingerprinting engine aims at identifying the usage of the audio tracks inside TV broadcasts in order to manage legal rights of authors, editors, performers and labels. The existing fingerprints, which are described in the next section, were computed in order to identify also very short music excerpts also in the presence of additional signals, mainly speech and environmental noise (e.g. clapping, car engines, crowd cheering, and so on). The computation of the $350,000$ audio fingerprints required approximately two months on a octa-core machine with processors at 1.6 GHz. This relatively long computation time is comparable to the one required to compute grab music from an audio CD or to download/upload the files.

4 Detection of Affinities

Given the size of the audio collection, a pairwise comparison of all the tracks was impracticable. Even on the fast 8-core machine available for the experiments, the existing audio fingerprinting engine would have completed the identification of affinities within all the songs in an estimated time of about three months. For this reason we decided to divide the procedure in two steps.

4.1 First Step: Pruning Candidate Affinities

A common approach to audio fingerprinting consists of summarizing with a sequence of integer numbers the audio content of short overlapping parts of the audio signal. A complete song is thus transformed in a sequence of integers, with the characteristic that similar audio excerpts are represented by the same integer. Thus, we can view this approach as audio hashing where collisions between buckets happen when the original audio excerpts are perceptually similar. A general approach exploits Locality Sensitive Hashing to create a set of hash function that guarantees at least a collision in case of similar audio content [13].

The fingerprints used in this work were computed following a simpler approach, proposed in [5], which uses a single hashing function computed from the frequency representation of the signal.

Given an audio track t^k sampled at the common CD rate of 44.1 kHz, we divide it in frames of about 0.1 seconds and compute their Fast Fourier Transform. Hash values are computed according to the distribution of the signal energy in a number of spectral bands. Thus the original track t^k can be represented by a sequence of time ordered hash values

$$l^k = (h_1^k, h_2^k \ldots, h_L^k) \quad \text{with} \quad h \in \mathbb{N} \tag{1}$$

where L depends on the length of the audio track and in an even more compact way as a set of unordered hash values

$$s^k = \{h_1^k, \ldots, h_D^k\} \quad \text{with} \quad h \in \mathbb{N} \quad \text{and} \quad h_i^k \neq h_j^k \quad \forall i \neq j \tag{2}$$

where D is the number of distinct hash values.

A first approximation of the affinity between two tracks t^h and t^k can thus be computed as the percentage of hash values they have in common, that is

$$af(t^h, t^k) = \frac{\|s^h \cap s^k\|}{min(\|s^h\|, \|s^k\|)} \tag{3}$$

where the normalizing factor guarantees that the maximum affinity value is 1 when a track is completely contained into the other (or the two tracks are identical).

The results of the first step are summarized in Table 1. The analysis showed that the collection contained 1057 exact duplicates (0.3% of the whole collection), at least from the point of view of the audio content because the actual size and content of the files may slightly differ. Although this has not been tested extensively, it is likely that almost all of these pairs can be identified with simple hashing techniques such as MD5. Another 104 pairs overlapped by more than 90% of their audio fingerprints. Since this high overlap is likely to be related to the use of different lossy compression software applied to the same CD track (according to the collection managers three different software were used along the years), this result seems to show that the fingerprinting technique is quite robust to lossy compression. These 1161 song pairs have been directly reported to the collection administrators in order to have one of the two files removed without additional manual checking. These files have not been used in subsequent analyses. It is interesting to note that in many cases the two songs of a pair were catalogued with different titles, which explains the double acquisition of the same material. Thus the analysis had a major impact on metadata correction, while the effect on MP3 cleanup was not particularly relevant in terms of storage reduction. The selection of the correct title in case of inconsistencies was carried out by a pool of experts.

Yet, the first step aimed at pointing out near and far candidates, to be checked in the second step of the analysis. There were 712 song pairs that overlapped

Table 1: Amount of common fingerprints between song pairs in the collection.

Overlap	# song pairs	% song pairs
Complete ($af = 1$)	1057	0.3%
High ($af > 0.9$)	104	0.03%
Partial ($af > 0.5$)	712	0.2%
Low ($af > 0.25$)	2098	0.6%
Minimal ($af > 0.1$)	1041	0.3%
Total	5012	1.43%

for more than a half of their audio content while the largest group of song pairs (2098) had an overlap between one quarter and a half of their content. Finally, a group of 1041 song pairs had an overlap between one tenth and one quarter. We decided not to consider in further analyses song pairs with an overlap smaller than one tenth. The choice of the thresholds was made according to the collection managers, in order to prioritize the process of manually investigating the identified affinities. The choice of ignoring overlaps smaller than 10% was another requirement in order to let the human intervention affordable and reduced the number of false positives basically to zero. A second experiment on affinity detection will be organized in the future in order to deal also with the remaining song pairs and to investigate in more details how false positives can impact the overall process of tracking affinities. False negatives were not measured with this collection. Yet, in a previous experiment carried out with a selection of 1000 songs the number of erroneous detection was about 6.3%.

After the first step we obtained a total of 3851 song pairs to inspect in more detail during the second step of the analysis. Having reduced consistently the size of the problem, the second step can focus on effectiveness without having to deal with scalability issues.

4.2 Second Step: Pairwise Match between Affinities

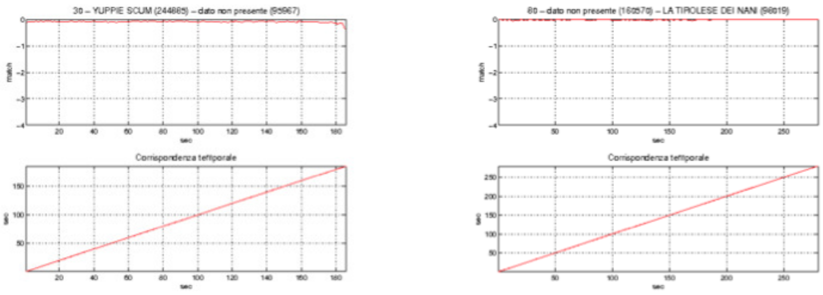
The output of the first step is a list of song pairs annotated with their affinity value as an overall measure of the shared audio content. The second step aims at refining the computation of affinities with a more descriptive representation of the similarities between songs. For this reason we represent each track as a sequence of time ordered hash values and, for the sake of clarity, we assume that a generic song pair is always in the form $p = \{l^1, l^2\}$ where the length of $l^1 = (h^1_1, \ldots, h^1_N)$ is always shorter or equal to the length of $l^2 = (h^2_1, \ldots, h^2_M)$.

According to [5] it is possible to compute hash values in order to define a similarity function between them. For instance, if hash values are in binary form the similarity $d(h^1_i, h^2_j)$ can be set inversely proportional to the hamming distance between h^1_i and h^2_j (which can be easily normalized in the interval $[0, 1]$). It is then possible to compute, for any short time interval in l^2, the best matching time interval in l^1, according to equations

$$\begin{aligned} m_j(l^1, l^2) &= \max_i \sum_{k=1}^{N} d(h^1_{i+k}, h^2_{j+k}) \\ p_j(l^1, l^2) &= \arg\max_i \sum_{k=1}^{N} d(h^1_{i+k}, h^2_{j+k}) \end{aligned} \tag{4}$$

where m_j is the similarity value of the best match between the two tracks around time position j of l^2 and p_j is the corresponding time position in l^1. The plot of these two functions can give interesting insight if paired with a manual inspection of the corresponding audio tracks.

For instance, figure 1 shows an example on how exact duplicates are represented, in order to have a better understanding of the results presented in the subsequent figures. The top graphs depict the trend of m_j and the bottom graphs depict p_j. For all the graphs, the x-axis represents time of the longer track l^2, in seconds. The y-axis in the m_j (top) graphs represents the value, in log scale, of the best match between the two tracks, while the y-axis in the p_j (bottom) graphs represents the time position, in seconds, of the best match on the shorter track. Thus two identical tracks have the top graph consistently equal to zero and the bottom graph coincident to the bisect of the first quadrant (in order to save space on the page, the aspect ratio of the bottom graphs has been compressed along the y-axis).



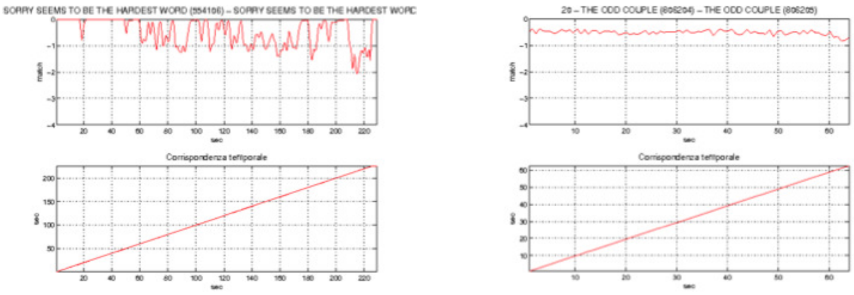
(a)

(b)

Fig. 1: Exact duplicates: result of (a) and (b) ripping with different lossy compression software of the same PCM audio source.

Two typologies of near duplicates are shown in Figure 2 and in Figure 3. In particular, Figure 2(a) represents the effect of heavy audio remastering of the same original material. While the two tracks are perfectly aligned (bottom) the value of the best match m_j (top) reveals the result of post-production which, in this particular case, is almost negligible at the beginning of the tracks becoming more relevant towards the end. Figure 2(b) represents the effect of a lighter remastering, which affects consistently the value of the best match m_j (top). Another case of near duplicates is encountered when the two tracks slightly differ in the orchestration. For instance, in Figure 3(a) the two songs are almost identical apart from three short excerpts towards the end of the song, where one of the takes has a choir doubling the main voice. Similarly, Figure 3(b) shows differences in two longer parts, which correspond to two choruses where a clearly audible synthesizer has been added in the orchestration. In all these cases the linear monotonic trend of p_j is a good evidence of a near-duplicate, while the trend of m_j may help discriminate between remastering and alternative takes.

Clearly, the choice of whether maintaing or not both tracks depends on the usages of the music collection. For the purpose of musicological analyses the two tracks, either remastered or different takes, are equally interesting and should be maintained, possibly with the indication of their differences. For the purpose of a TV broadcaster, the tracks are basically interchangeable since the average audience will never notice their differences.

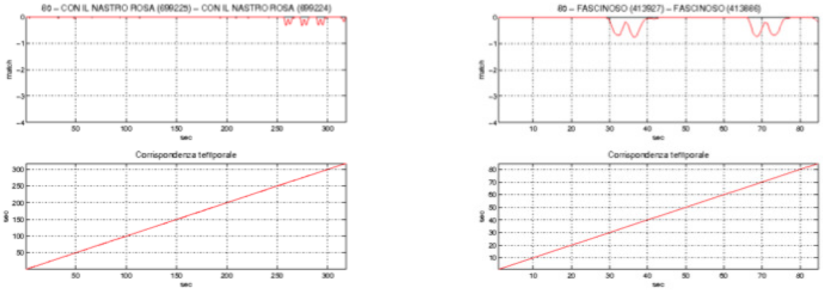


(a)

(b)

Fig. 2: Near duplicates: effect of heavy (a) and light (b) remastering of the same audio source.

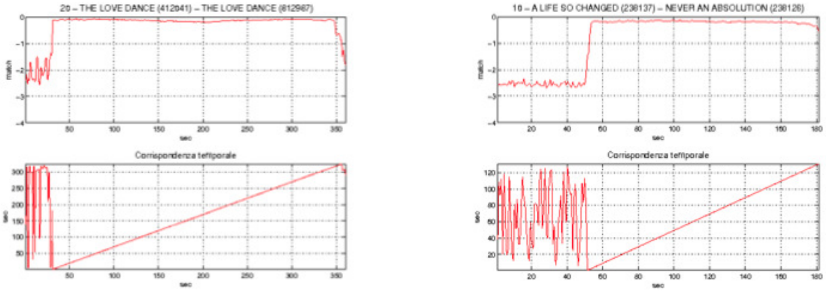
Among the results of the first step we identified a number of far duplicates, that is song pairs that share a substantial part of the audio content but cannot be considered simple variants of the same audio source. We considered two main typologies: mashups and montages. Examples of mashups are shown in Figure 4, where in both cases the audio material contained in track l^1 is used to create l^2, possibly in combination with additional content taken from other tracks. This can easily be seen comparing the initial part of the top and bottom graphs. The best match m_j is quite low and corresponds to random time correspondences of p_j. When the mashup track starts using the audio material of the other track, the best match increases its value and the corresponding positions proceed aligned as in the case of near duplicates. Examples of different montages are shown in Figure 5. Here the two tracks share part or even all of the audio content, which



(a) (b)

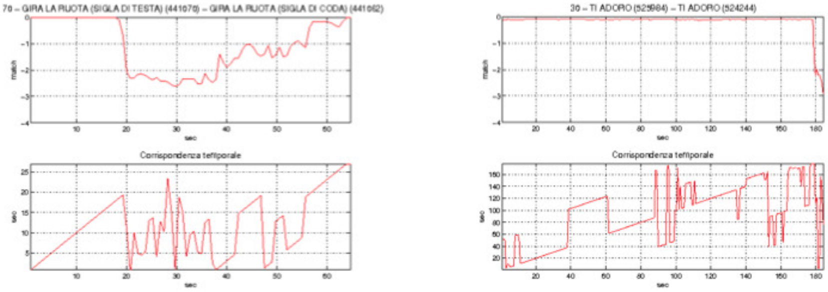
Fig. 3: Near duplicates: different takes of the same song, with (a) additional choir in three short parts at the end and (b) additional synthesizer in two long parts.

is organized in different ways along time. This operation can be surprising if we consider normal pop songs, but it is not uncommon in instrumental music – especially when sound samples are used instead of real music instruments – where the author composes and performs independent parts and then combines them in different ways to create variants final track. For instance, Figure 5(a) compares the opening and closing tracks of a TV program. The two tracks have basically the same beginning and almost the same ending while from 38 to 55 seconds of l^2 the diagonal lines show that different parts of l^1 have been used. Figure 5(b) shows that there is a very high value of the best match m_j along all l_2 but the corresponding elements of l^1 have been combined differently as shown by the discontinuities of p_j.



(a) (b)

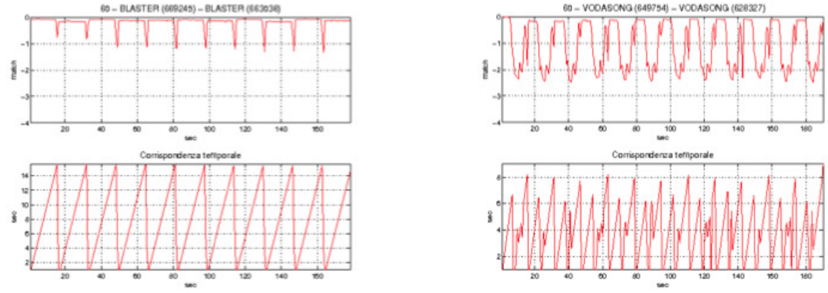
Fig. 4: Far duplicates: mashups with (a) short and (b) long songs not present in the collection that both precede the identified ones.



(a) (b)

Fig. 5: Far duplicates: montages with the same audio material (a) with additional sources after an identical intro and (b) with pure permutations of the audio content.

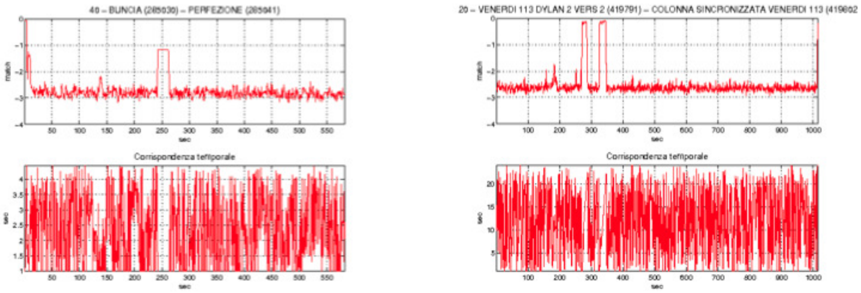
A particular case of far duplicates are loops. It may be argued that loops are more likely to be near duplicates, because one track uses and repeats many times the audio content of the other. This is shown in Figure 6(a), where the longer track contains almost exact repetitions of the shorter one with the only instants with lower m_j values at the joints between repetition. Yet, the use of loops is common practice for hip-hop artists, who compose new songs directly from already published recordings. Although this is probably not the case of the comparison shown in Figure 6(b) – the two songs have the same name – the trends of the two graphs show that the longer track is based on the repetition of the shorter plus additional instrumentation, which may be the typical situation where part of a song is looped and combined with additional material to create a new song.



(a) (b)

Fig. 6: Far duplicates: loops, with (a) simple juxtaposition and (b) additional edit of the original audio source.

Clearly, the simple approach of counting the common fingerprints of the first step can results also in false positives. Figure 7 shows two cases of false positives, where the percentage of common fingerprints may be explained by the use of the same audio samples probably taken from the same sound library. The trend of both bottom and top graphs are quite different from the ones shown in the previous figures, so the task of identifying false affinities does not present high complexity.



(a)

(b)

Fig. 7: False positives: the same libraries of audio samples have been used (a) once in the middle of both songs and (b) in two short parts of one song.

5 Discussion

Large multimedia digital collections are increasingly available on the Web, posing new challenges in organizing, storing and accessing the material. This paper focuses on a typical problem of music collections that is the presence of similar material with different levels of variations, which results in exact, near and far duplicates. We proposed the term *affinities* to refer to all these variations.

From the results of our initial experiments, it seems to be possible to efficiently search for affinities even in a large music collection and, furthermore, to describe the typology of affinity between two audio files with the aid of a pair of graphs representing the level of match between two parallel audio excepts and the alignment curve. The trend of the two graphs that can be interpreted in order to identify the kind of affinity. It is expected that visual identification would be faster and, in case of long audio excerpts, even more reliable than identification based on pure listening. Yet, the next step in the approach will be the automatic classification of affinities, which can be based on the statistical properties of the graphs and on the parallel analysis of the matching and the alignment curves.

Acknowledgments

The author wishfully thanks the company LaCosa s.r.l. for granting access to a large digital audio collection, which has been the basis for the tests, and for providing useful insights on how interpreting the results.

References

1. Casey, M., Slaney, M.: Fast recognition of remixed music audio. In: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing. pp. IV:1425–IV:1428 (2007)
2. Datta, R., Joshi, D., Li, J., Wang, J.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 5:1–5:60 (2008)
3. Fetterly, D., Manasse, M., Najork, M.: On the evolution of clusters of near-duplicate web pages. Journal of Web Engineering 2(4), 228–246 (2003)
4. Gracenote: Gracenote music solutions. http://www.gracenote.com/music/recognition/ (2015), [Online; accessed 20-July-2015]
5. Haitsma, J., Kalker, T.: A highly robust audio fingerprinting system with an efficient search strategy. Journal of New Music Research 32(2), 211–221 (2003)
6. Imran, N.: Electronic media, creativity and plagiarism. ACM SIGCAS Computers and Societies 40(4), 25–44 (2010)
7. Ke, Y., Hoiem, D., Sukthankar, R.: Computer vision for music identification. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition. pp. 597–604 (2005)
8. Lalinský, L.: How does Chromaprint work? https://oxygene.sk/2011/01/how-does-chromaprint-work/ (2011), [Online; accessed 20-July-2015]
9. Liu, J., Huang, Z., Cai, H., Shen, H., Ngo, C., Wang, W.: Near-duplicate video retrieval: Current research and future trends. ACM Computing Surveys 45(4), 44:1–44:23 (2013)
10. Liu, J., Huang, Z., Shen, H., Cui, B.: Correlation-based retrieval for heavily changed near-duplicate videos. ACM Transactions on Information Systems 29(4), 21:1–21:25 (2011)
11. Montecchio, N., Di Buccio, E., N., O.: An efficient identification methodology for improved access to music heritage collections. Journal of Multimedia 7(2), 145–158 (2012)
12. Nucci, M., Tagliasacchi, M., Tubaro, S.: A phylogenetic analysis of near-duplicate audio tracks. In: Proc. of IEEE International Workshop on Multimedia Signal Processing. pp. 99–104 (2013)
13. Slaney, M., Casey, M.: Locality-sensitive hashing for finding nearest neighbors. IEEE Signal Processing Magazine 25(2), 128–131 (2008)
14. Wang, A.: The shazam music recognition service. Communications of the ACM 49(8), 44–48 (2006)