# Towards Cover Group Thumbnailing

Peter Grosche
Saarland University
and MPI Informatik
Saarbrücken, Germany
pgrosche@mpi-inf.mpg.de

Meinard Müller
International Audio
Laboratories Erlangen
Erlangen, Germany
meinard.mueller@audiolabs-erlangen.de

Joan Serrà
Artificial Intelligence Research
Institute (IIIA-CSIC)
Bellaterra, Spain
jserra@iiia.csic.es

## ABSTRACT

In this paper we investigate whether we can extract the commonalities shared by a group of cover songs or versions of the same musical piece. As a main contribution, we introduce the concept of cover group thumbnail, which is the most representative, essential subsequence for an entire group of versions. Opposed to previous approaches, we jointly consider all versions of a given song to compute a single cover group template, which then shows a high degree of robustness against version-specific aspects. To compute such a template, we introduce a modification of a recent audio thumbnailing technique. To evaluate the reliability of our conceptual contribution, we consider the task of template-based version identification, where we show comparable accuracies to existing systems.

## Categories and Subject Descriptors

H.5.5 [**Sound and Music Computing**]: Methodologies and Techniques, Systems.

## Keywords

Cover song essence, music retrieval, audio thumbnailing.

## 1. INTRODUCTION

Cover songs are alternative versions or performances of a previously recorded musical piece. They often differ from the original in several musical aspects such as timbre, tempo, song structure, tonality, arrangement, lyrics, or language of the vocals. The goal of cover song or version identification is to automatically detect all versions of the same piece of music within a pool of documents [4, 11]. Version identification is usually interpreted as a document-level retrieval task, where a single similarity measure is used to globally compare entire documents. However, successful methods perform this global comparison on a local basis, obtaining the final similarity measure by comparing parts (or subsequences) of the documents. The global similarity measure can then be derived from the best matching subsequence (e.g. [3, 6, 11]).


*MM* '13 Barcelona, Spain


State-of-the-art approaches based on subsequence matching show a significant robustness to musical variations (see [11] for a review). However, requiring pairwise comparisons of the query and all subsequences of all available documents, they typically do not scale to large audio collections. One step towards better scalability is to employ indexing schemes to exhaustively compute similarities between subsequences [2, 3]. Another, possibly complementary strategy is to reduce the number of subsequences that need to be compared. This approach, which motivates the present paper, has been rather unexplored, with the notable exceptions of [6, 8].

In an ideal case, for a given song, a single subsequence that acts as a compact descriptor or *template* for the entire song would be sufficient to identify all other versions. In [6, 8], *music thumbnailing* approaches were employed to determine the most representative and repetitive segment of every single song. Typically, such a segment has many (approximate) repetitions covering large parts of the song [1, 7, 10]. The underlying assumption is that a segment which represents a song well might also capture aspects of the different versions. However, as thumbnails are extracted for each version individually, they often do not correspond to the same musical section in all versions. As a result, performing comparisons of entire songs only on the basis of the individually extracted templates might not reveal the desired similarities.

Here, our goal is to extract a single robust template that captures the essence of a group of cover songs. The idea is to compute a *cover group thumbnail*: a single segment that is the most representative and repetitive segment for an entire group of versions of a piece. Conceptually, our approach differs from previous approaches as we jointly consider all versions of a given song to compute a single thumbnail, whereas previous approaches consider individual songs to compute a thumbnail for each version separately [6, 8] or consider groups in a post-processing step [11]. As shown in [11], considering entire groups of versions provides meaningful information which can enhance current systems.

As our main technical contribution, we introduce a modification of a recent audio thumbnailing technique [10], so that the following problem can be solved: Given a group of sequences (let say $K$ sequences $Y_1, \ldots, Y_K$) and a reference sequence $X$, find a subsequence of $X$ that simultaneously "explains" in each of the sequences $Y_1, \ldots, Y_K$ a suitable subsequence. The resulting subsequence of $X$ is called *cover group thumbnail* and the resulting $K$ subsequences are called *induced subsequences*. The hypothesis is that these induced subsequences are similar to the cover group thumbnail. To obtain a *cover group template* that captures aspects of all

versions, we compute an average representation of the cover group thumbnail and all induced subsequences.

Cover group templates have two main advantages over individual song templates. First, based on a joint analysis of an entire group, they show a higher degree of robustness against some version-specific aspects. Second, a single template is sufficient to represent the entire group, which could potentially allow reducing the computational load in a retrieval scenario. In our experiments, we analyze to which extent an entire cover group can be characterized by a single segment and show that cover group thumbnails can be effectively used for template-based version identification.

## 2. COVER GROUP TEMPLATES

**Audio features:** For our approach, we employ chroma features capturing information that closely correlates to harmonic and melodic properties of the audio recording. Such features have become a widely used tool for processing and analyzing music data in general [1, 3, 9] and cover song identification in particular [3, 6, 11]. The 12-dimensional chroma vectors express how the short-time energy of the audio signal is distributed over the twelve chroma bands. Following [9], we decompose the audio signal into subbands that correspond to the semitones of the equal-tempered scale. Then, adding up the bands that belong to the same pitch class, we obtain a chroma representation. Finally, applying suitable quantization, smoothing, downsampling, and normalization operations results in enhanced chroma features referred to as CENS[1] [9]. In the following, we use a feature resolution of 1 Hz (one feature vector per second).

**Similarity matrices:** Let $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$ be two chroma sequences. Furthermore, let $\mathbf{s}$ be a similarity measure that allows for comparing two chroma features (we here use the cosine measure). Then, an $M \times N$ *similarity matrix* (SM) is obtained by comparing the elements of $X$ and $Y$ in a pairwise fashion: $\mathcal{S}(m, n) := \mathbf{s}(y_m, x_n)$ for $m \in [1 : M]$ and $n \in [1 : N]$. Furthermore, we apply a smoothing filter [9], which results in an emphasis of diagonal information in $\mathcal{S}$. For handling tonality differences across the versions, we adopt the concept of transposition-invariant similarity matrices [7]. We first compute the similarity between the sequence $X$ and each of the twelve cyclically shifted versions of $Y$ resulting in twelve similarity matrices. Then, the transposition-invariant SM $\mathcal{S}$ is obtained by taking the point-wise maximum over these matrices. Subsequently, we apply a thresholding operation with the goal to achieve that relevant paths lie in the positive part of $\mathcal{S}$, whereas all other cells receive a negative penalty $\delta \leq 0$. As proposed in [11], we use a relative threshold that identifies cells that belong both to the 30% of the cells having the highest value in each column and to the 30% having the highest value in each row (the remaining cells are set to $\delta = -1$). These experimentally found parameter values did not have a significant impact on the results.

In our scenario, we jointly consider similarity matrices for an entire group of $K$ cover songs. Let $Y_k$ denote the feature sequence of the version $k \in [1 : K]$ and $Y$ the concatenation of the $Y_k$. Furthermore, we fix a version $k_0 \in [1 : K]$ to serve as reference and define $X = Y_{k_0}$. Note that any version $k_0 \in [1 : K]$ could act as a reference. Let $N$ denote the length of $X$, $M_k$ the length of $Y_k$, and $M := \sum_{k \in [1:K]} M_k$ the length of $Y$. Then, we compute an $M \times N$ similarity matrix $\mathcal{S}$ for the sequence $Y$ and the reference $X$.

[1] *Chroma Energy Normalized Statistics*, an implementation of these features is available online: `http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/`.

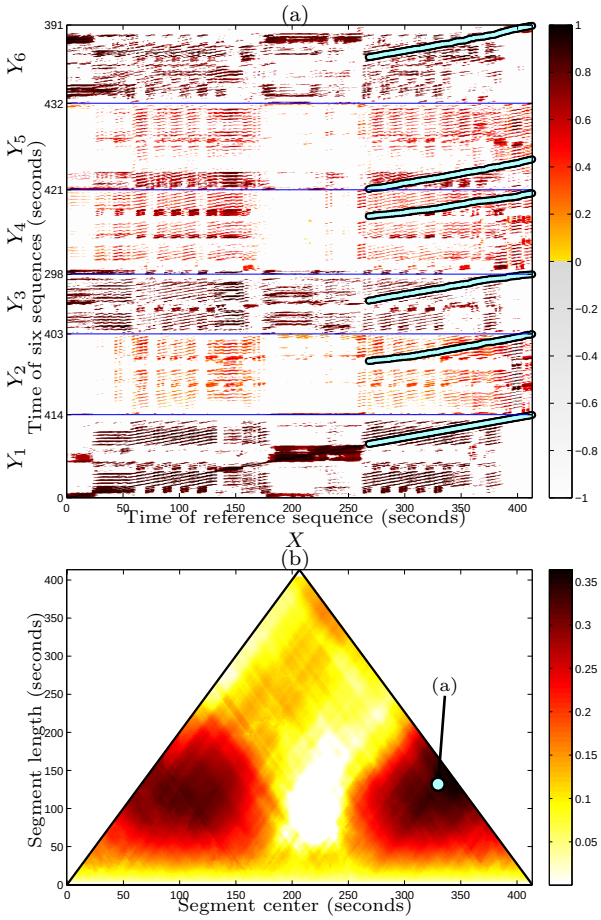


**Figure 1: (a) Similarity matrix for a group of six versions of Kraftwerk's "Radioactivity" and optimal path family. Horizontal lines indicate boundaries between versions. (b) Fitness scape plot.**

Fig. 1a shows $\mathcal{S}$ for a group of $K = 6$ versions of Kraftwerk's *Radioactivity*. In this example, $X = Y_1$ corresponds to the original version and the sequences $Y_k$, $k \in [2 : 6]$ are covers. Each path of cells of high similarity within $\mathcal{S}$ indicates the similarity between subsequences of $X$ and $Y$ given by the projections of the path onto the horizontal and vertical axis, respectively. In the case that two versions are very similar, one observes a long path. In the case of musical variations, however, the paths are often fragmented.

**Cover group thumbnail:** Let $\alpha = [s : t] \subseteq [1 : N]$ denote a subsequence of $X$ specified by its starting point $s$ and end point $t$. In [10], a fitness measure is introduced that assigns to each $\alpha$ a fitness value $\varphi(\alpha) \in \mathbb{R}$ that simultaneously captures two aspects. It indicates (i) how well $\alpha$ explains other subsequences of $X$ and (ii) how much of $X$ is covered by these subsequences. The thumbnail of $X$ is then defined to be the subsequence $\alpha^*$ with maximal fitness $\varphi$.

In the computation of the fitness measure, the main technical idea is to assign to $\alpha$ a so-called *optimal path family* that reveals the relations between $\alpha$ and all other similar

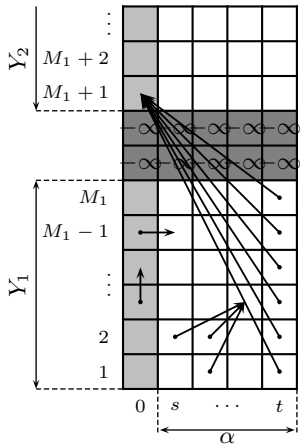


**Figure 2: Illustration of the modifications of the thumbnailing algorithm for an excerpt of the similarity matrix $\mathcal{S}(m,n)$ for $n=[s:t]$ and $m=[1:M]$ corresponding to the subsequence $\alpha=[s:t]$.**

subsequences. A path family over $\alpha$ is defined to be a set $\mathcal{P} := \{p_1, p_2, \ldots, p_I\}$ of size $I$, consisting of paths $p_i$ over $\alpha$, $i \in [1:I]$. Fig. 1a shows a path family consisting of $I=6$ paths for the segment $\alpha=[269:414]$. Each path over $\alpha$ of length $L$ is a sequence $p_i = ((m_1,n_1),\ldots,(m_L,n_L))$ of cells $(m_\ell,n_\ell) \in [1:M]\times[1:N]$, $\ell \in [1:L]$, satisfying $n_1=s$ and $n_L=t$ as well as $(m_{\ell+1},n_{\ell+1})-(m_\ell,n_\ell) \in \Omega$, where $\Omega$ denotes a set of admissible step sizes. We use $\Omega=\{(1,2),(2,1),(1,1)\}$, as it has been shown to perform well for the task of version identification [9, 11]. Importantly, a path family induces an entire family of related subsequences $\alpha_i$ that are similar to $\alpha$, given by the projection to the vertical axis $\pi(p_i) := [m_1 : m_L]$ of each path $p_i$.

Instead of analyzing songs individually, we here analyze a group of versions simultaneously. In particular, our goal is to extract for each segment $\alpha$ of the reference sequence $X$ exactly one subsequence in each of the versions. To do so, we modify the original algorithm presented in [10] and use additional constraints to enforce that the number of paths equals the number of versions ($I=K$) and that $\alpha_i \subset Y_k$ for $i=k\in[1:K]$, i.e., there is exactly one $\alpha_i$ in each $Y_k$. Among all possible subsequences of $Y_k$, $\alpha_i$ corresponds to the one that is most similar to $\alpha$.

In [10], a dynamic programming algorithm is introduced for computing optimal path families. Our technical modification is twofold (Fig. 2). First, to ensure that no path $p_i$ induces a subsequence in more than one $Y_k$, we insert cells with a score of $-\infty$ between $Y_k$ and $Y_{k+1}$ for $k\in[1:K-1]$. As a result, a path which crosses the boundary between two versions gets a score of $-\infty$ (Note that, because of the step size condition $\Omega=\{(1,2),(2,1),(1,1)\}$, two entries with $-\infty$ are needed). Second, to ensure that there is a path $p_i$ inducing a subsequence in each sequence $Y_k$, we add the condition that the extraction of a path is stopped at the end $t$ of $\alpha$ and a new path is started in the next version $Y_{k+1}$ (Fig. 2, arrows starting in the last column). As this is the only way to cross the $-\infty$ rows, the combination of both modifications ensure that the resulting path family $\mathcal{P}$ consist of exactly one path for each version $Y_k$.

Fig. 1b shows an example of fitness values $\varphi(\alpha)\in\mathbb{R}$ for all reference segments $\alpha$ in the form of a scape plot representation, where each point corresponds to one subsequence $\alpha$ represented by its center $c(\alpha) := (s+t)/2$ and its length $|\alpha| := t-s+1$. The fitness $\varphi(\alpha)$ indicates how well $\alpha$ explains subsequences of $Y$ and how much of $Y$ is covered by all these subsequences. The subsequence $\alpha^*=[269:414]$ having maximal fitness $\varphi$ is considered to be the *cover group thumbnail*, the subsequence that best explains and is most similar to the entire group. The optimal path family over $\alpha^*$ encodes the relation between $\alpha^*$ and the induced subsequences $\alpha_k^*$, one in each of the versions, see Fig. 1a.

**Template extraction:** To obtain a cover group template for a given group and reference $X=Y_{k_0}$, we compute the thumbnail $\alpha^*$ and extract the chroma features of the induced subsequences $\alpha_k^*$. The cover group template is then obtained by averaging all chroma sequences. Here, we exploit the multi-alignment between $\alpha^*$ and $\alpha_k^*$ given by the paths $p_k$ to determine for each chroma vector $x_n$ the corresponding vectors $y_{k,m}$ in all versions $k\in[1:K]$. Actually, this operation results in a temporal warping which compensates for temporal differences in the versions. Furthermore, we compensate for possible transpositions by employing a circular shift strategy [11]. The necessary cyclic shift index is determined by estimating the global similarity between $X$ and the warped chroma sequences using the cosine measure. The shift index with maximum similarity is used to compensate for a transposition. The final cover group template $\mathcal{T}_{k_0}$ with respect to the reference $k_0$ is then obtained by point-wise averaging all $K$ chroma sequences.

## 3. TEMPLATE-BASED RETRIEVAL

To quantify our assumption that cover group templates capture characteristic aspects of a group of versions, and to evaluate the reliability of cover group thumbnails, we consider the task of cover song identification. Given a cover group template, we investigate if it is possible to retrieve all versions of the group from a dataset. We use a dataset $\mathcal{D}$ obtained from [11] that consists of $G=17$ groups, each containing $K=6$ versions ($|\mathcal{D}| = G\times K = 102$).

Since our goal is to gauge the potential of cover group templates, in this proof-of-concept experiment we perform training and evaluation on the same dataset (notice however that we do not perform any exhaustive parameter tuning). For each group $\mathcal{G}_g\subset\mathcal{D}$ with $g\in[1:G]$, we compute $K$ templates $\mathcal{T}_{g,k_0}$ by selecting, in turn, each song $k_0\in[1:K]$ as reference. Then, we perform retrieval on the whole dataset employing a subsequence matching strategy. Specifically, we compare $\mathcal{T}_{g,k_0}$ locally with all chroma subsequences of the database using a DTW-based distance measure [9]. The final distance values for a song are obtained by minimizing the distances of all subsequences of that song.

Following standard practice [11], we express the retrieval accuracy in terms of *mean of average precision* (MAP). Given the group $\mathcal{G}_g$ of $K$ versions that are relevant to the template $\mathcal{T}_{g,k_0}$, we obtain the precision $\psi_{g,k_0}$ at rank $r\in[1:|\mathcal{D}|]$ as $\psi_{g,k_0} = \frac{1}{r}\sum_{i=1}^{r}\Gamma_{g,k_0}(i)$, where $\Gamma_{g,k_0}(r)\in\{0,1\}$ indicates whether the version at rank $r$ is contained in $\mathcal{G}_g$. The average precision $\overline{\psi}_{g,k_0}\in[0,1]$ is then defined as $\overline{\psi}_{g,k_0} = \frac{1}{K}\sum_{r=1}^{|\mathcal{D}|}\psi_{g,k_0}\Gamma_{g,k_0}(r)$. Table 1 shows $\overline{\psi}_{g,k_0}$ values when using the version $k_0\in[1:K]$ as reference (the higher, the better). MAP values $\overline{\psi}_g = 1/K\sum_{k_0=[1:K]}\overline{\psi}_{g,k_0}$, $\overline{\psi}_g^+ = \max_{k_0}(\overline{\psi}_{g,k_0})$ and $\overline{\psi}_g^- = \min_{k_0}(\overline{\psi}_{g,k_0})$ are also shown for

| $g$ | $\overline{\psi}_{g,1}$ | $\overline{\psi}_{g,2}$ | $\overline{\psi}_{g,3}$ | $\overline{\psi}_{g,4}$ | $\overline{\psi}_{g,5}$ | $\overline{\psi}_{g,6}$ | $\overline{\psi}_{g}$ | $\overline{\psi}_{g,k_0{}^*}$ | $\overline{\psi}_{g}^{+}$ | $\overline{\psi}_{g}^{-}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.41 | 0.45 | 0.12 | 0.57 | 0.15 | 0.19 | 0.31 | 0.45 | 0.57 | 0.12 |
| 2 | 0.90 | 0.88 | 0.93 | 1.00 | 0.97 | 0.86 | 0.92 | 0.90 | 1.00 | 0.86 |
| 3 | 0.47 | 0.78 | 0.39 | 0.42 | 0.24 | 0.55 | 0.47 | 0.78 | 0.78 | 0.24 |
| 4 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 | 0.96 | 1.00 | 1.00 | 0.74 |
| 5 | 0.93 | 0.62 | 0.75 | 0.80 | 0.78 | 0.34 | 0.70 | 0.80 | 0.93 | 0.34 |
| 6 | 0.64 | 0.07 | 0.81 | 0.81 | 0.67 | 0.67 | 0.61 | 0.67 | 0.81 | 0.07 |
| 7 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 8 | 0.59 | 0.59 | 0.68 | 0.51 | 0.54 | 0.35 | 0.54 | 0.68 | 0.68 | 0.35 |
| 9 | 0.94 | 0.69 | 0.59 | 0.67 | 0.71 | 0.64 | 0.71 | 0.71 | 0.94 | 0.59 |
| 10 | 0.88 | 0.97 | 0.59 | 0.78 | 0.77 | 0.77 | 0.79 | 0.88 | 0.97 | 0.59 |
| 11 | 0.97 | 0.90 | 1.00 | 0.88 | 1.00 | 0.94 | 0.95 | 1.00 | 1.00 | 0.88 |
| 12 | 0.74 | 0.61 | 0.65 | 0.69 | 0.73 | 0.79 | 0.70 | 0.73 | 0.79 | 0.61 |
| 13 | 0.61 | 0.67 | 0.58 | 0.37 | 0.74 | 0.20 | 0.53 | 0.74 | 0.74 | 0.20 |
| 14 | 0.42 | 0.70 | 0.67 | 0.65 | 0.79 | 0.86 | 0.68 | 0.79 | 0.86 | 0.42 |
| 15 | 0.86 | 1.00 | 1.00 | 0.86 | 0.87 | 1.00 | 0.93 | 1.00 | 1.00 | 0.86 |
| 16 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 17 | 1.00 | 1.00 | 1.00 | 0.89 | 1.00 | 1.00 | 0.98 | 1.00 | 1.00 | 0.89 |
| ø | 0.78 | 0.76 | 0.75 | 0.76 | 0.76 | 0.70 | 0.75 | 0.83 | 0.89 | 0.57 |

**Table 1: MAPs for each of the 17 cover groups, consisting of 6 songs each (see text).**

each group $g \in [1:G]$. The values $\overline{\psi}_{g,k_0{}^*}$ are obtained by choosing for each group the reference $k_0{}^* \in [1:K]$ that has maximum fitness $k_0{}^* = \text{argmax}_{k_0}(\varphi^*_{k_0})$.

The values $\overline{\psi}_g^+$ indicate the best possible results that can be achieved by our template-based approach. It turns out that the average over all groups $\overline{\psi}^+ = 0.89$ is similar to the results for the state-of-the-art subsequence matching algorithm reported in [11], which indicates the potential of a template-based approach. Note that instead of comparing all subsequences of all $|\mathcal{D}|$ documents with all subsequences of all $|\mathcal{D}|$ documents, one main advantage of our template-based approach is that we need only to compare $G$ templates with the subsequences of $|\mathcal{D}|$ documents. Thus, as the number of comparisons and the complexity of the similarity measure is reduced, the template-based approach also facilitates efficient retrieval.

Importantly, we observed that the selection of a single template representing a group turned out to be a crucial step. For some groups (e. g., $g = 7, 11, 16, 17$) one obtains rather consistent results for all choices of a reference $k_0$. For other groups, however, the choice of a reference has a large influence on the retrieval results. For example, in the case of the group $g = 6$, the maximum average precision is $\overline{\psi}_{6,k_0} = 0.81$ when using the reference $k_0 = 4$. However, with $k_0 = 2$, one only obtains $\overline{\psi}_{6,k_0} = 0.07$. One strategy to select a proper reference is to use the reference $k_0^* \in [1:K]$ that has maximum fitness (this could also be a potential application of our approach to querying databases containing cover groups). For the group $g = 6$, $k_0^*$ corresponds to $k_0 = 5$ which results in $\overline{\psi}_{6,k_0{}^*} = 0.67$. In average over all 17 groups, one obtains $\overline{\psi}_{k_0{}^*} = 0.83$, only a minor reduction in accuracy in comparison to the ideal case $\overline{\psi}^+ = 0.89$.

## 4. CONCLUSION

The modification of a recent thumbnailing approach allowed us to analyze all versions of the same piece simultaneously and to extract a cover group template that is invariant to version-specific aspects: the cover group thumbnail. We evaluated the reliability of our conceptual contribution under a cover song retrieval scenario, assuming that all versions of a song have a common, essential subsequence. First experiments showed that a template-based cover song identification system may have the potential of yielding similar results as state-of-the-art approaches and, as a by-product, reducing the number of necessary comparisons.

The selection of a reference version, however, turned out to have a large influence on the resulting templates, which weakens our assumption of a common subsequencefor all versions . Being based on chroma features, our approach can only capture harmonic similarities. In many groups, however, cover versions are characterized by similarities of the melody line, bass line, or rhythm pattern [11]. A more robust cover group template extraction requires additional features which capture the many facets of similarities between cover versions (cf. [5]).

## 5. ACKNOWLEDGMENTS

This work was supported by the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University, JAEDOC069/2010, FP7-ICT-2011-8-318770, and 2009-SGR-1434.

## 6. REFERENCES

[1] M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Trans. on Multimedia*, 7(1):96–104, 2005.

[2] T. Bertin-Mahieux and D. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proc. of IEEE WASPAA*, pages 117–120, 2011.

[3] M. A. Casey, C. Rhodes, and M. Slaney. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Trans. on Audio, Speech and Language Processing*, 16(5):1015–1028, 2008.

[4] D. Ellis and G. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. of IEEE ICASSP*, IV:1429–1432, 2007.

[5] R. Foucard, J.-L. Durrieu, M. Lagrange, and G. Richard. Multimodal similarity between musical streams for cover version detection. In *Proc. of IEEE ICASSP*, pages 5514–5517, 2010.

[6] E. Gómez, B. S. Ong, and P. Herrera. Automatic tonal analysis from music summaries for version identification. In *Proc. of the AES Convention*, 2006.

[7] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Trans. on Audio, Speech and Language Processing*, 14(5):1783–1794, 2006.

[8] M. Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proc. of ISMIR*, pages 280–285, 2006.

[9] M. Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[10] M. Müller, P. Grosche, and N. Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. In *Proc. of ISMIR*, pages 615–620, 2011.

[11] J. Serrà. *Identification of versions of the same musical composition by processing audio descriptions*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2011.