

MIREX 2016: SiMPle-BASED COVER SONG IDENTIFICATION USING SMOOTHED CRP FEATURES

Diego Furtado Silva^{1,2}, Chin-Chia Michael Yeh²,
Gustavo E. A. P. A. Batista¹, Eamonn Keogh²

¹ Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Brazil

² Department of Computer Science and Engineering, University of California, Riverside, USA
diegofsilva@icmc.usp.br, myeh003@ucr.edu, gbatista@icmc.usp.br, eamonn@ucr.edu

ABSTRACT

This document presents a solution to the Audio Cover Identification task submitted to the Music Information Retrieval Evaluation eXchange (MIREX) 2016. Our method uses a recent proposed primitive to assess similarity between music recordings: Similarity Matrix Profile (SiMPle). The first step is to describe the songs by extracting Chroma DCT-Reduced log Pitch (CRP) features, which is a timbre-invariant chroma representation. Given that SiMPle is based on local similarities, we applied smoothing techniques to the CRP to provide tempo invariance. Then, we transpose one of the feature vectors in order to provide key invariance, by applying the Optimal Transposition Index (OTI). Finally, the similarity between two tracks is given the median value of the SiMPle obtained by comparing the query to the reference recording.

1. INTRODUCTION

“Cover song” is the generic term used to denote a new performance of a previously recorded track. For example, a cover song may refer to a live performance, a remix or an interpretation in a different music style. The automatic identification of covers has several applications, such as copyright management, collection organization, and search by content.

In order to identify different versions of the same song, most algorithms search for globally or locally conserved structure(s). In this document, we describe a method which explore both strategies. Specifically, we used a recent proposed primitive to assess similarity in music recordings: Similarity Matrix Profile (SiMPle) [4]. With this representation, we are able to find the (global) similarity between songs, based on the similarities of their subsequences (local).

Before we introduce the general overview of our system to recognize cover songs, we describe the primitive SiMPle.

2. SiMPle: SIMILARITY MATRIX PROFILE

We begin by describing the operation for producing the matrix profile, a *similarity join*. For clarity, we use the term *time series* to refer to the ordered set of features that describe a whole recording and *subsequence* to define any continuous subset of features from the time series.

Definition 1: *Similarity join*: given two time series A and B with the desired subsequence length m , the similarity join identifies the nearest neighbor of each subsequence (with length m) in A from all the possible subsequence set of B .

Through such a similarity join, we can gather two pieces of information about each subsequence in A , which are: 1) the Euclidean distance to its nearest neighbor in B and 2) the position of its nearest neighbor in B . Such information can be compactly stored in vectors, referred as *similarity matrix profile* (SiMPle) and *similarity matrix profile index* (SiMPle index) respectively.

We describe our method to calculate SiMPle in Algorithm 1. In line 1, we record the length of B . In line 2, we allocate memory and initialize SiMPle P_{AB} and SiMPle index I_{AB} . From line 3 to line 6, we calculate the *distance profile* vector D which contains the distances between a given subsequence in time series B and each subsequence in time series A . The particular function we used to compute D is *MASS* (Mueen’s Algorithm for Similarity Search), which is the most efficient algorithm known for distance vector computation [1]. We then perform the pairwise minimum for each element in D with the paired element in P_{AB} (i.e., $\min(D[i], P_{AB}[i])$ for $i = 0$ to $\text{length}(D) - 1$). We also update $I_{AB}[i]$ with idx when $D[i] \leq P_{AB}[i]$ as we perform the pairwise minimum operation. Finally, we return the result P_{AB} and I_{AB} in line 7.

Algorithm 1. Procedure to calculate SiMPle and SiMPle index

Input: Two user provided time series, A and B , and the desired subsequence length m

Output: The SiMPle P_{AB} and the associated SiMPle index I_{AB}

```
1  $n_B \leftarrow \text{Length}(B)$ 
2  $P_{AB} \leftarrow \text{infs}, I_{AB} \leftarrow \text{zeros}, \text{idxes} \leftarrow 1:n_B-m+1$ 
3 for each  $idx$  in  $\text{idxes}$ 
4      $D \leftarrow \text{MASS}(B[idx:idx+m-1], T_A)$  // c.f. [1]
5      $P_{AB}, I_{AB} \leftarrow \text{ElementWiseMin}(P_{AB}, I_{AB}, D, idx)$ 
6 end for
7 return  $P_{AB}, I_{AB}$ 
```

The method *MASS* (used in line 4) is important to speed-up the similarity calculations. This algorithm has a time complexity of $O(n \log n)$. For brevity, we refer the reader interested in details of this method to [1].

3. SYSTEM OVERVIEW

Figure 1 shows a block diagram of our system. In addition, this section explains each step of the method.

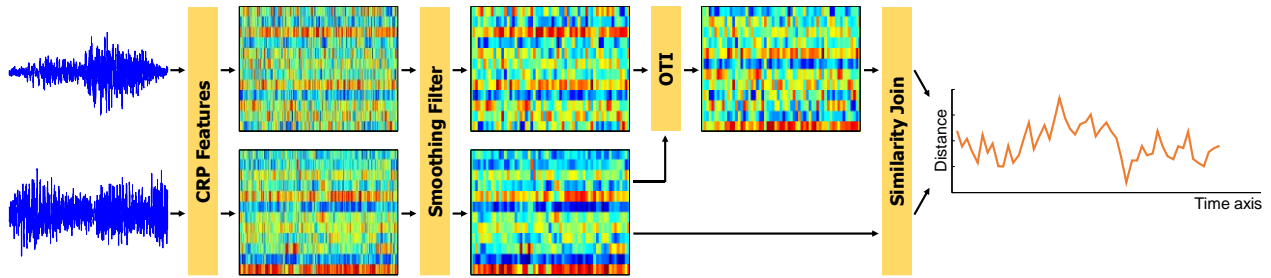


Figure 1. Block diagram of our identification system. For each raw audio, we extract CRP features and smooth them to provide local tempo variation. For a given pair of recordings, we apply OTI to provide key invariance and, then, calculate the SiMPle. Finally, the distance between the two recordings is given by the median of the SiMPle.

3.1 CRP Features

The first step of our system is to load the raw files and describe them by tonal patterns. Specifically, we extract Chroma DCT-Reduced log Pitch (CRP) features from the signals. In summary, the CRP is a chroma representation which provides timbre invariance by discarding timbre-related information. For the purpose of extracting these features from the raw audio, we used the Matlab Chroma Toolbox [2].

3.2 Smoothing Filter

Given that SiMPle is based on the Euclidean distance-based join, it is sensitive to tempo variations. However, given that this operation looks for local similarities, providing local tempo invariance significantly improves its use. So, in the next step of our method, we applied smoothing techniques to the CRP features. Once again, we used the smoothing filter implemented in the Matlab Chroma Toolbox [2] in this phase.

3.3 Key Invariance by OTI

In addition to the previous steps, we preprocessed the feature sets in each comparison to provide key invariance. Before calculating the similarity between songs, we transpose one of them in order to have the same estimated key using the optimal transposition index (OTI) [3].

3.4 SiMPle-based Distance

For each pair of chroma features (already transposed to the same key), we estimate the global distance between them by using the values which compose the SiMPle between them. Specifically, given a query Q and a reference song R , the SiMPle-based distance between them is defined by Equation 1.

$$\text{dist}(Q,R)=\text{median}(\text{SiMPle}(Q,R)) \quad (1)$$

Note that several other central tendency functions could be used instead of the median. However, the median is robust to outliers in the matrix profile. Such distortions may appear when a performer decides, for instance, to add a new segment (e.g., an *improvisation* or *drum solo*) to the song.

4. INITIAL RESULTS

We compared the results obtained by the proposed system to previous results using SiMPle in a database of popular music. In [4], the authors reported 0.591, 0.140 and 7.91 of MAP, precision at 10 and mean position of the first correctly identified cover, respectively. Using our system, we achieved 0.635, 0.141 and 7.27 for the same evaluation measures in the same dataset.

5. CONCLUSION

We proposed a new cover song identification system based on the similarity matrix profile (SiMPle). In addition to be simple, our method seems to be quite effective. The results obtained in our experiment are better than the previous results obtained by similar techniques, which had been shown to be more effective than state-of-the-art algorithms.

6. REFERENCES

- [1] A. Mueen, K. Viswanathan, C. K. Gupta and E. Keogh. “The fastest similarity search algorithm for time series subsequences under Euclidean distance”, url: www.cs.unm.edu/~mueen/FastestSimilaritySearch.html (accessed 24 May, 2016).
- [2] M. Müller. “Chroma Toolbox: MATLAB Implementations for Extracting Variants of Chroma-Based Audio Features”. *International Society for Music Information Retrieval Conference*, 2011.
- [3] J. Serrà, E. Gómez, and P. Herrera. “Transposing chroma representations to a common key”. *CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, 2008.
- [4] D. F. Silva, C.-C. M. Yeh, G. E. A. P. A. Batista, and E. Keogh. “SiMPle: Assessing Music Similarity Using Subsequences Joins”. *International Society for Music Information Retrieval Conference*, 2016.