TOWARD CHARACTERISTIC AUDIO SHINGLES FOR EFFICIENT CROSS-VERSION MUSIC RETRIEVAL

Peter Grosche and Meinard Müller

Saarland University and MPI Informatik
{pgrosche,meinard}@mpi-inf.mpg.de

ABSTRACT

The general goal of cross-version music retrieval is to identify all versions of a given piece of music by means of a short query audio fragment. To speed up the retrieval process, hashing techniques have been proposed, where the audio material is split up into small overlapping shingles (used as hashes) that consist of short feature subsequences. In this paper, we extend this work with the goal to minimize the number of hash lookups. To this end, one requires larger shingles that characterize the underlying piece of music to a high degree, while being robust to variations that occur across different versions. As our main contribution, we report on extensive experiments to highlight the delicate trade-off between the query length, feature parameters, shingle dimension, and index settings. These insights are of fundamental importance for building efficient cross-version retrieval systems that scale to millions of songs.

***Index Terms*—** Audio shingle, music retrieval, audio matching, cover song identification, locality sensitive hashing

1. INTRODUCTION

In the last decade, content-based music retrieval based on *audio fingerprinting* has become of commercial interest with various applications ranging from broadcast monitoring to automatic organization of music collections [1]. Recent systems allow for identifying an audio recording by means of a small query audio fragment even in the presence of signal distortions and, employing efficient index structures, scale to millions of songs. Being based on a rather strict notion of similarity close to identity, fingerprinting systems are designed to basically detect exact duplicates of the queried audio fragment.

The requirements on retrieval systems change significantly when considering cross-version retrieval tasks such as audio matching, opus retrieval, or cover song identification [2, 3, 4, 5]. Here, given an audio fragment as query, the general goal is to automatically retrieve from a given collection all documents that contain fragments musically similar to the query. Such documents may include various interpretations, arrangements and cover songs of the piece underlying the query fragment. For such cross-version scenarios, one needs retrieval systems that can handle variations with regard to musical properties such as tempo, articulation, timbre or instrumentation. Dealing with a much lower specificity level as in the fingerprinting scenario, the development of efficient cross-version retrieval systems that scale to huge data collections still faces challenging problems.

In this paper, we address the fundamental issue on how cross-version retrieval can be accelerated by employing index structures that are based on suitably designed elementary building blocks. Building upon ideas of two recently proposed retrieval systems [2, 4], we first summarize these approaches and then describe our contributions.

In [4], a matching procedure is described that allows for a fragment-based retrieval of all audio excerpts musically related to a given query audio fragment. To this end, the query and all database documents are converted to sequences of chroma-based audio features that correlate to harmonic properties. To cope with temporal variations, global scaling techniques are employed to derive multiple queries that simulate different tempi. Finally, feature quantization techniques in combination with inverted file indexing is applied to speed up the retrieval process. The authors report on speed-up factors of 10-20 for medium sized data collections. However, using a codebook of fixed size, this approach does not scale well to collections of millions of songs.

In [2], a different approach is described. Instead of considering long feature sequences, the audio material is split up into small overlapping *shingles* that consist of short chroma feature subsequences. These shingles are indexed using locality sensitive hashing. While being very efficient (the authors report on a speed-up factor of 100) and scalable to even large data collections, the proposed shingling approach has one major drawback. To cope with temporal variations, each shingle covers only a small portion of the audio material (three seconds in the proposed system). As a result, an individual shingle is too short to characterize well a given piece of music. Therefore, to obtain a meaningful retrieval result, one needs to combine the information retrieved for a large number of query shingles. As a consequence, many hash-table lookups are required in the retrieval process. This becomes particularly problematic, when the index structure is stored on a secondary storage device.

Based on ideas of these two approaches, we systematically investigate in this paper how one can significantly reduce the number of hash-table lookups. The main idea is to use a shingling approach, where an individual shingle covers a relatively large portion of the audio material (between 10 and 30 seconds). Compared to short shingles, such large shingles have a higher musical relevance so that a much lower number of shingles suffices to characterize a given piece of music. However, increasing the size of a shingle comes at the cost of increasing the dimensionality and possibly loosing robustness to variations. Building on well-known existing techniques, the main contribution of this paper is to systematically investigate the delicate trade-off between the query length, feature parameters, shingle dimension, and index settings. In particular, we experimentally determine a setting that allows for retrieving most versions of a piece of music when using only a single 120-dimensional shingle covering roughly 20 seconds of the audio material. Furthermore, we show that such large shingles can still be indexed using locality sensitive hashing with only a small degradation in retrieval quality.

The remainder of this paper is organized as follows. In Sec-

The authors are supported by the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University.

tion 2, we introduce the overall retrieval approach. Then, in Section 3, we report on our systematic experiments. Conclusions and prospects on future work are given in Section 4. Further related work is discussed in the respective sections

2. BASIC RETRIEVAL STRATEGY

In our cross-version retrieval scenario, given a short fragment of a music recording as query, the goal is to retrieve all music recordings (documents) that contain a passage similar to the query from a large dataset. The retrieval result for a query is given as a ranked list of document identifiers.

To this end, we proceed in three steps. Given a query Q and a document D to be compared, the first step consists in converting Q and D into sequences of feature vectors $X = (X(1), \ldots, X(M))$ and $Y = (Y(1), \ldots, Y(N))$, respectively. In our system, as in [4, 2], we use 12-dimensional chroma-based audio features, which are a powerful mid-level representation for capturing harmonic content in music recordings, while being robust to other musical aspects. More precisely, we use a chroma variant referred to as CENS[1] features [6], which involve a temporal smoothing by averaging chroma vectors over a window of length w and downsampling by a factor of d. In our experiments, we use a feature rate of 10 Hz for the basic chroma vectors. Then, for example, setting $d = 10$ and $w = 41$ results in one feature vector per second (a feature resolution of 1 Hz), where each vector is obtained by averaging over 41 consecutive frames, corresponding to roughly 4 sec of the audio. The resulting features $\texttt{CENS}(w, d)$ show an increased robustness to local tempo changes and allow for flexibly adjusting the temporal resolution.

In the second step, the sequence X is compared with subsequences $Y_t^M := (Y(t), \ldots, Y(t + M - 1))$ of length M for $t \in [1 : N - M + 1]$. Here, we adopt the idea of audio shingles [2] and reorganize the sequences of feature vectors into shingle vectors. In our system, we represent each query Q as a single shingle of dimension $M \times 12$. Then, we use the cosine measure to obtain a similarity value between X and all subsequences of Y of length M defined as $s(X, Y_t^M) = \langle X | Y_t^M \rangle / (\|X\| \cdot \|Y_t^M\|)$, where $\|\cdot\|$ denotes the Euclidean norm. In the third step, we then express the document-wise similarity of Q and D as

$$S(Q, D) = \max_{t \in [1:N-M+1]} \left(s(X, Y_t^M) \right) \ . \tag{1}$$

Given Q and a dataset $\mathcal{D}$ containing $|\mathcal{D}|$ documents, we compute S between Q and all $D \in \mathcal{D}$ and rank the result by descending $S(Q, D)$. In practice, however, such an exhaustive search strategy is not needed to find the relevant documents. Instead, one tries to efficiently cut down the set of candidate subsequences using index-based strategies such as locality sensitive hashing (LSH) and computes S in Eq. (1) using only the retrieved shingles (setting $s(X, Y_t^M) = 0$ for non-retrieved shingles Y_t^M).

Given the set $\mathcal{D}_Q \subset \mathcal{D}$ of documents that are relevant to the query Q, we follow [7] and express the retrieval accuracy in terms of the *mean of average precision measure* (MAP) denoted as $\langle \overline{\psi} \rangle$.[2] To this end, we obtain the precision ψ_Q at rank $r \in [1 : |\mathcal{D}|]$ as

$$\psi_Q(r) = \tfrac{1}{r} \textstyle\sum_{i=1}^{r} \Gamma_Q(i) \ , \tag{2}$$

where $\Gamma_Q(r) \in \{0, 1\}$ indicates whether a document is contained in $\mathcal{D}_Q$. Then, the average precision $\overline{\psi}_Q$ is defined as

$$\overline{\psi}_Q = \tfrac{1}{|\mathcal{D}_Q|} \textstyle\sum_{r=1}^{|\mathcal{D}|} \psi_Q(r) \Gamma_Q(r) \ . \tag{3}$$

Furthermore, using several queries, we compute $\overline{\psi}_Q$ for each Q and average over all values to obtain the MAP value $\langle \overline{\psi} \rangle$. Furthermore, we determine $\langle \overline{\psi} \rangle_{\text{null}}$ expected under the null hypothesis of a randomly created sorted list as in [7].

Typically there are tempo differences in the versions considered in our retrieval scenario. As a result, a musical passage represented by a query can be realized in another version with significant temporal differences. In that case, our choice of representing each query as a single shingle would require a comparison of shingles representing feature sequences of differing length. One approach to this problem is to use similarity measures based on dynamic time warping (DTW) or Smith-Waterman [5]. However, regarding computationally efficiency and an application in the indexing context, such procedures are problematic. Instead, we employ the *query scaling strategy* as proposed in [4]. Here, tempo differences are handled by creating R scaled variants of the query $Q^{(1)}, \ldots, Q^{(R)}$, each simulating a global change in the tempo of the query. The similarity value between D and Q is then defined as

$$S(Q, D) = \max_{r \in [1:R]} \left(S(Q^{(r)}, D) \right) \ . \tag{4}$$

Furthermore, as a baseline strategy, we handle tempo difference between Q and D using an offline *DTW-based procedure* [8] that ensures that corresponding feature sequences coincide in all versions. This idealized procedure serves as reference in our experiments as it provides an optimal estimate of $S(Q, D)$ even in the case of strong non-linear temporal distortions.

3. EXPERIMENTS

In this section, we describe our systematic experiments to investigate the trade-off between efficiency and shingle characteristic. First, in Section 3.1, we introduce our dataset. Then, in Section 3.2, we investigate how long a query Q needs to be to accurately characterize all versions and what a suitable feature resolution is. In Section 3.3, we analyze how well tempo differences can be handled by the query scaling approach (avoiding warping procedures). In Section 3.4, we further reduce the shingle dimension using principal component analysis (PCA). Finally, in Section 3.5, we use locality sensitive hashing (LSH) to accelerate cross-version retrieval.

3.1. Datasets

In our experiments, we use a dataset $\mathcal{D}$ of 2484 audio recordings with an overall runtime of 162 hours, see Table 1. A subset (denoted $\mathcal{D}_{\text{Queries}}$) of 359 recordings is used for obtaining queries. These recordings correspond to classical music pieces by three different composers. For each piece, there are 7 to 88 different recorded versions available. More precisely, the first part `Chop` consists of 298 piano recordings of five Mazurkas by Frédéric Chopin.[3] The second part `Beet` consists of ten recorded performances of Beethoven's *Symphony No. 5* in orchestral as well as piano interpretations. The third part `Viva` contains seven orchestral performances of the *Summer* from Vivaldi's *Four Seasons*. Additionally, we add 2125 recordings of various genre to enlarge the dataset. In our experiments, we randomly select 100 queries from each of the three parts of $\mathcal{D}_{\text{Queries}}$ and average the results over the resulting 300 queries.

[1] *Chroma Energy Normalized Statistics* features, provided by the Chroma Toolbox www.mpi-inf.mpg.de/resources/MIR/chromatoolbox

[2] MAP is also used in MIREX Cover Song Identification, see www.music-ir.org/mirex/wiki/2011:Audio_Cover_Song_Identification

[3] This data is provided by the Mazurka Project http://mazurka.org.uk/

	Composer	Piece	Description	#	Dur. (min)
Chop	Chopin	Op. 17, No. 4	Mazurka	62	269
	Chopin	Op. 24, No. 2	Mazurka	64	147
	Chopin	Op. 30, No. 2	Mazurka	34	48
	Chopin	Op. 63, No. 3	Mazurka	88	189
	Chopin	Op. 68, No. 3	Mazurka	50	84
Beet	Beethoven	Op. 67, 1. Mov.	Fifth	10	75
	Beethoven	Op. 67, 2. Mov.	Fifth	10	98
	Beethoven	Op. 67, 3. Mov.	Fifth	10	52
	Beethoven	Op. 67, 4. Mov.	Fifth	10	105
Viva	Vivaldi	RV 315, 1. Mov.	Summer	7	38
	Vivaldi	RV 315, 2. Mov.	Summer	7	17
	Vivaldi	RV 315, 3. Mov.	Summer	7	20
$\mathcal{D}_{\text{Queries}}$				359	1145
$\mathcal{D}$				2484	9725

Table 1: The music collection used in our experiments. The last two columns denote the number of different performances and the duration in minutes.

3.2. Query Length and Feature Resolution

In a first experiment, we investigate how much of a recording needs to be captured by the query Q to robustly characterize all versions of the underlying piece. Furthermore, we analyze to what extent the temporal resolution of the features can be reduced without negatively affecting the retrieval quality. Here, we exploit the downsampling and smoothing parameters d and w of the $\texttt{CENS}(w, d)$ features. The goal is to reduce the overall dimensionality of the query while retaining as much of the retrieval accuracy as possible. For the moment, we use the DTW-based procedure to account for tempo differences between the versions.

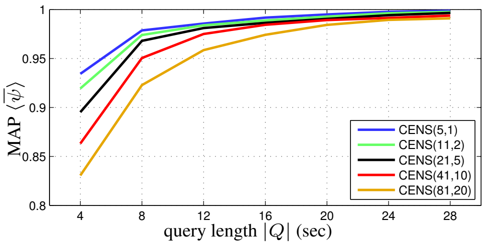

Fig. 1: MAP values as a function of query length $|Q|$ using $\texttt{CENS}(w, d)$ in different feature resolutions. Null hypothesis $\langle\overline{\psi}\rangle_{\text{null}} = 0.015$.

Fig. 1 shows MAP values obtained using $\texttt{CENS}(w, d)$ features with seven different query lengths $|Q|$ and five different feature resolutions. Obviously, the longer $|Q|$ the higher the retrieval quality. For example, for $|Q| = 28$ sec, one obtains MAP values of $\langle\overline{\psi}\rangle \approx 0.99$, regardless of the feature resolution. Short queries, however, can not accurately capture the characteristics of a piece, leading to significantly lower MAP values. Reducing the feature resolution, one observes lower MAP values, too, in particular in combination with short queries. For example, using $|Q| = 4$ sec, one obtains $\langle\overline{\psi}\rangle \approx 0.94$ for $\texttt{CENS}(5, 1)$ (10 Hz resolution) and $\langle\overline{\psi}\rangle \approx 0.83$ for $\texttt{CENS}(81, 20)$ (0.5 Hz resolution). Increasing the query length, however, this effect vanishes. In particular for $|Q| \geq 20$ sec one obtains similar MAP values, independent of the feature resolution. Using $d = 10$ (1 Hz) as in $\texttt{CENS}(41, 10)$ with $|Q| = 20$ sec constitutes a good trade-off between query dimensionality and query characteristic. This setting results in shingles with a dimensionality of 240.

3.3. Matching Strategy

In this experiment, we investigate how much of retrieval accuracy is lost when using the query scaling approach for handling tempo differences instead of the idealized DTW-based technique. Fig. 2(a) shows the retrieval quality using $\texttt{CENS}(41, 10)$ for different query scaling settings. Here, we use R variants of the query with scaling factors specified by the set T. $R = 1$ means that only the original query is used. Furthermore, we use $R = 3$ with $T = \{0.8, 1, 1.25\}$, meaning that the query is also stretched by a factor of 0.8 and 1.25 (thus simulating tempo changes of roughly $\pm 25\%$). Similarly, we use $R = 5$ with $T = \{0.66, 0.8, 1, 1.25, 1.5\}$ and $R = 9$ with $T = \{0.66, 0.73, 0.8, 0.9, 1, 1.1, 1.25, 1.35, 1.5\}$. The red line indicates the DTW-based result as shown in Fig. 1. From these results, we draw two conclusions. Firstly, the scaling strategy ($R > 1$) significantly increases the retrieval quality in comparison to only using the original query ($R = 1$). The actual choice of parameters does not seem to be crucial. In the case of our dataset, already a small number of additional queries ($R = 3$) seems to be sufficient. Secondly, the scaling strategy leads to very similar results as the computationally expensive DTW-based strategy, in particular when using a large smoothing window (e. g., $w = 41$ in $\texttt{CENS}(41, 10)$). In the case of the smaller smoothing window $w = 5$ in $\texttt{CENS}(5, 1)$ (see Fig. 2(b)), the difference is more significant. In summary, a local feature smoothing in combination with a global scaling strategy yields a robust yet computational simple alternative to warping procedures.

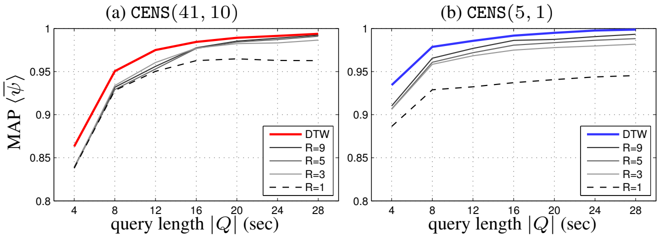
(a) $\texttt{CENS}(41, 10)$
(b) $\texttt{CENS}(5, 1)$

Fig. 2: MAP values obtained for four query scaling strategies and the DTW-based strategy using **(a)** $\texttt{CENS}(41, 10)$ and **(b)** $\texttt{CENS}(5, 1)$.

3.4. Dimensionality Reduction

In a third experiment, we investigate in how far statistical data reduction based on Principal Component Analysis (PCA) can be applied to CENS features to further reduce the dimensionality of the query.

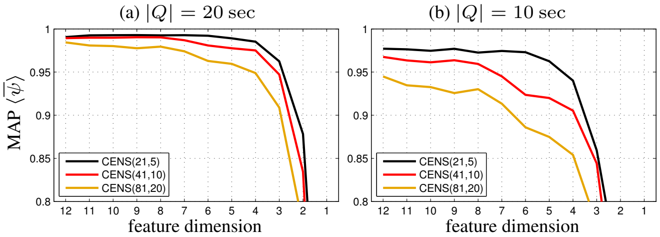
(a) $|Q| = 20$ sec
(b) $|Q| = 10$ sec

Fig. 3: MAP values as a function of feature dimension obtained by PCA-based dimension reduction of $\texttt{CENS}(w, d)$.

We estimate the principal components using all non-query documents of our dataset and project all feature sequences onto the most dominating components. Fig. 3 shows MAP values obtained for PCA-reduced variants of CENS features with 1-12 remaining dimensions. For a query length $|Q| = 20$ sec (Fig. 3a), MAP values are nearly unaffected when reducing the number of dimensions from 12 to 4, in particular for higher feature resolutions. However, in combination with shorter queries of $|Q| = 10$ (Fig. 3b), the retrieval quality is more affected by a dimensionality reduction.

In the following, we use the first 6 components of $\texttt{CENS}(41, 10)$ features, denoted as $\texttt{CENS}(41, 10)\texttt{-6}$.[4] Using $|Q| = 20$, this results

[4] Further experiments revealed that $\texttt{CENS}(41, 10)\texttt{-6}$ is very similar to the

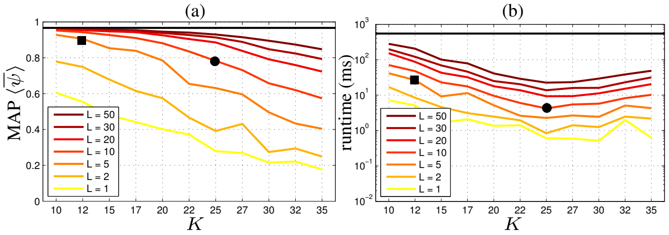

Fig. 4: Illustration of the MAP values and runtimes obtained using $\texttt{CENS}(41, 10)\texttt{-6}$ with different parameter settings in the LSH-based retrieval experiment. **(a)** Retrieval quality MAP. **(b)** Overall runtime per query including index lookup time and document ranking time. Horizontal black lines indicate values obtained by the exhaustive search.

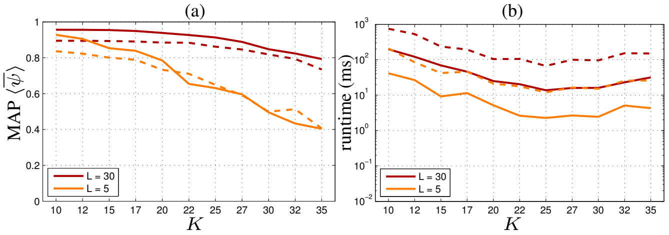

Fig. 5: Comparison of two LSH-based retrieval strategies. Warping strategy (solid line) and query scaling strategy ($R = 5$, $T = \{0.66, 0.8, 1, 1.25, 1.5\}$) (dashed line) using $\texttt{CENS}(41, 10)\texttt{-6}$. **(a)** MAP values. **(b)** Overall runtime per query.

in 120-dimensional shingles, which constitutes a reasonable trade-off between shingles dimensionality and shingle characteristic.

3.5. Locality Sensitive Hashing

We now investigate whether it is possible to index shingles of this size using locality sensitive hashing (LSH) for accelerating the retrieval. LSH is a hash-based approach for finding approximate nearest neighbors based on the principal that similar shingles are indexed with the same hash value. In our experiment, we use an implementation of the Exact Euclidean LSH (E^{2}LSH) algorithm [10]. We index all shingles of the entire dataset $\mathcal{D}$ using L parallel indices and K hash functions. For a query shingle Q we retrieve all shingles from the index with the same hash value as the query. Given this (typically small) set of candidate shingles, we derive the ranked list of documents and compute MAP values as described in Section 2.

Fig. 4 shows MAP values (Fig. 4a) and runtime per query in milliseconds[5] (Fig. 4b) as a function of K for different L.[6] These are crucial parameters having a tremendous influence on the trade-off between retrieval quality and runtime. For example, setting $K = 12$ and $L = 5$ results in a MAP $\langle\bar{\psi}\rangle \approx 0.90$, see black square in Fig. 4a. This is only slightly lower than the MAP value one obtains for the exhaustive search (horizontal black line). However, the runtime for this setting is significantly (by a factor of 25) faster than for the exhaustive search, see black square in Fig. 4b. K and L allow for controlling the trade-off between speed and quality of the results. Setting $K = 25$ and $L = 10$, the MAP drops to $\langle\bar{\psi}\rangle \approx 0.80$ (black circle). However, this goes along with a decrease of query runtime to 5 ms, a speed-up of 100 in comparison to the exhaustive search.

The results shown in Fig. 4 are again obtained using the ideal DTW-based procedure for handling tempo differences. Fig. 5 now shows the comparison of the warping (solid line) with the query scaling approach (dashed line) for $L = 5$ and $L = 30$. Similar as for the exhaustive search discussed in Section 3.3, using $R = 5$, one observes only a small drop in retrieval quality (see Fig. 4a). Using this strategy, the runtime per query linearly increases with the number of scaled queries R (see Fig. 4b).

4. CONCLUSIONS

Concluding the experiments, one can say that even when using large shingles (covering roughly 20 seconds of audio material), LSH-based indexing techniques can be applied for obtaining a significant speed-up of the retrieval process (up to factor of 100). At the same time, most of the accuracy of an exhaustive search can be retained. To facilitate this, we determined suitable parameter settings with regard to query length, feature resolution and smoothing, as well as shingle dimension. The advantage of using shingles that represent a large audio fragment is that most versions of a given piece can be characterized and retrieved by using a single shingle. In future work, we exploit this to significantly reduce the number of hash-table lookups needed for performing cross-version retrieval. The number of lookups becomes a crucial bottleneck when the index structure is stored on secondary storage devices, which is unavoidable when dealing with collections of millions of songs.

5. REFERENCES

[1] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma, “A review of algorithms for audio fingerprinting,” in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, St. Thomas, Virgin Islands, USA, 2002, pp. 169–173.

[2] Michael Casey, Christophe Rhodes, and Malcolm Slaney, “Analysis of minimum distances in high-dimensional musical spaces,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 5, 2008.

[3] Daniel P. W. Ellis and Graham E. Poliner, “Identifying ‘cover songs’ with chroma features and dynamic programming beat tracking,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, Apr. 2007, vol. 4.

[4] Frank Kurth and Meinard Müller, “Efficient index-based audio matching,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 382–395, Feb. 2008.

[5] Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra, “Chroma binary similarity and local alignment applied to cover song identification,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 1138–1151, Oct. 2008.

[6] Meinard Müller, *Information Retrieval for Music and Motion*, Springer Verlag, 2007.

[7] Joan Serrà, Xavier Serra, and Ralph G. Andrzejak, “Cross recurrence quantification for cover song identification,” *New Journal of Physics*, vol. 11, no. 9, pp. 093017, 2009.

[8] Sebastian Ewert, Meinard Müller, and Peter Grosche, “High resolution audio synchronization using chroma onset features,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[9] Christopher Harte, Mark Sandler, and Martin Gasser, “Detecting harmonic change in musical audio,” in *Proceedings of the ACM Workshop on Audio and Music Computing Multimedia*, Santa Barbara, California, USA, 2006, pp. 21–26.

[10] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *Symposium on Computational Geometry*, Jack Snoeyink and Jean-Daniel Boissonnat, Eds. 2004, pp. 253–262, ACM.

musically motivated 6-dimensional *tonal centroid* proposed in [9].

[5] obtained on a Xeon X5560 CPU with 72GB of RAM

[6] The quantization parameter denoted r in [10] is found as proposed in [2].