Melody, Bass Line, and Harmony Representations for Music Version Identification

Justin Salamon
Music Technology Group,
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
justin.salamon@upf.edu

Joan Serrà
Artificial Intelligence Research
Institute (IIIA-CSIC)
Campus de la UAB s/n
08193 Bellaterra, Spain
jserra@iiia.csic.es

Emilia Gómez
Music Technology Group,
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
emilia.gomez@upf.edu

ABSTRACT

In this paper we compare the use of different musical representations for the task of version identification (i.e. retrieving alternative performances of the same musical piece). We automatically compute descriptors representing the melody and bass line using a state-of-the-art melody extraction algorithm, and compare them to a harmony-based descriptor. The similarity of descriptor sequences is computed using a dynamic programming algorithm based on nonlinear time series analysis which has been successfully used for version identification with harmony descriptors. After evaluating the accuracy of individual descriptors, we assess whether performance can be improved by descriptor fusion, for which we apply a classification approach, comparing different classification algorithms. We show that both melody and bass line descriptors carry useful information for version identification, and that combining them increases version detection accuracy. Whilst harmony remains the most reliable musical representation for version identification, we demonstrate how in some cases performance can be improved by combining it with melody and bass line descriptions. Finally, we identify some of the limitations of the proposed descriptor fusion approach, and discuss directions for future research.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Content Analysis and Indexing; H.5.5 [**Information Systems**]: Sound and Music Computing

Keywords

Version identification, cover song detection, melody extraction, bass line, harmony, music similarity, music retrieval

1. INTRODUCTION

The challenge of automatically detecting versions of the same musical piece has received much attention from the research community over recent years (see [18] for a survey). Potential applications range from the detection of copyright violations on websites such as YouTube, to the automation of computational analyses of musical influence networks [2]. Version identification on its own also represents an attractive retrieval task for end users.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

Systems for the automatic detection of versions exploit musical facets which remain mostly unchanged across different renditions, primarily the harmonic (or tonal) progression. In most cases, the harmonic progression is represented as a series of chroma descriptors (also called pitch class profiles) and compared using techniques such as dynamic time warping or simple cross-correlation [18]. Another musical representation that has been considered for version identification is the main melody, either by attempting to fully transcribe it [21], or by using it as a mid-level representation for computing similarity [13]. Melodic representations have also been widely used for related tasks such as query-by-humming [3] or music retrieval using symbolic data [22].

Whilst good results have been achieved using single musical representations (in particular harmony [18]), some recent studies suggest that version detection could be improved through the combination of different musical cues [5, 11]. However, surprisingly, not much research has been carried out in this direction. One of the first studies to automatically extract features derived from different musical representations for version identification was conducted by Foucard et al. [5], in which a source separation algorithm was used to separate the melody from the accompaniment. The authors then compared the performance of a version identification system using the melody, the accompaniment, the original mix, and their combination, by employing different fusion schemes. The study showed that considering different information modalities (i.e. main melody and accompaniment) is a promising research direction, but also noted the intrinsic limitation of simple fusion schemes whose capabilities seemed to be limited to merging modalities that carry more or less the same type of information. In the work of Ravuri and Ellis [14], the task of detecting musical versions was posed as a classification problem, and different similarity measures were combined to train a classifier for determining whether two musical pieces where versions or not. However, only chroma features were used to derive these similarity measures. Therefore, they were all in fact accounting for the same musical facet: the harmony.

In this paper we expand the study of version identification using different musical representations. In particular, we explore three related yet different representations: the harmony, the melody, and the bass line. Rather than use source separation [5], we employ a state-of-the-art melody extraction algorithm, which achieved the highest overall accuracy

results in the most recent MIREX[1] evaluation campaign [15, 16]. The bass line is extracted using a modified version of the melody extraction algorithm. Both melody and bass line are evaluated against a state-of-the-art version identification system using chroma features [20], which are closely related to the harmony. This system has achieved the best version identification results to date, according to MIREX[2]. Beyond exploring single musical representations alone, we also study their combination. For this we use the power of a standard classification approach, similar to Ravuri and Ellis [14]. In addition, we compare a number of classification algorithms and assess their ability to fuse the information coming from the three different representations.

The structure of the remainder of the paper is as follows: in Section 2 we describe the musical representations compared in this study, how we compute descriptors to represent them, the computation of version similarity, and our approach for descriptor fusion. In Section 3 we describe the evaluation methodology, including the music collection and evaluation measures used to assess the accuracy obtained using the different descriptors and their combinations. In Section 4 we present the results of the evaluation for both individual descriptors and descriptor fusion. Finally, in Section 5, we discuss the limitations of the proposed approach and suggest future research directions and applications.

2. MUSICAL REPRESENTATIONS, SIMILARITY AND FUSION

In the following subsections we describe the different descriptors evaluated in this study. We start by providing a brief description of the harmonic pitch class profile (HPCP), a harmony based chroma descriptor which has been used successfully for version identification [20]. Next, we describe the melody and bass line descriptors we use, including how they are extracted and subsequently converted into a representation suitable for computing version similarity. Finally, we outline our sequence matching procedure and explain our descriptor fusion strategy. The complete matching process, using either a single musical representation or descriptor fusion is depicted in the block diagram of Figure 1.

2.1 Harmonic Representation

To represent harmony, we compute the sequence of harmonic pitch class profiles (HPCP) [6, 7]. The HPCP is an enhanced chroma feature and, as such, it is derived from the frequency-dependent energy in a given range (typically from 50 to 5000Hz) in short-time spectral representations of the audio signal (e.g. 100ms; frame-by-frame extraction). The energy is mapped into an octave-independent histogram representing the relative intensity of each of the 12 semitones of the equal-tempered chromatic scale (12 pitch classes). To normalise with respect to loudness, the histogram is divided by its maximum value, thus leading to values between 0 and 1. Two important preprocessing steps are applied during the computation of the HPCP: tuning estimation and spectral whitening [6]. This means the HPCP is tuning-frequency independent and robust to changes in timbre, which makes it especially attractive for version identification.

[1] http://www.music-ir.org/mirex/wiki/2011:MIREX2011_Results

[2] http://www.music-ir.org/mirex/wiki/2009:Audio_Cover_Song_Identification_Results

Chroma features are a standard tool in music information research, and the HPCP in particular has been shown to be a robust and informative chroma feature implementation. For more details we refer the interested reader to [6] and references therein. For the purpose of this study, and in order to facilitate the comparison with previous work on version identification, the HPCP is computed using the same settings and parameters as in [20].

2.2 Melody and Bass Line Representations

The melody descriptor is computed in two stages. First, the melody is estimated from the polyphonic mixture using a state-of-the-art melody extraction algorithm [16]. The algorithm produces an estimation of the melody's pitch at each frame (represented as a fundamental frequency, F0). In the second stage, this representation (F0 per frame) must be converted into a representation which is suitable for version identification. The bass line descriptor is computed in the same way, using a modified version of the melody extraction algorithm adapted for extracting bass lines.

2.2.1 *Melody Extraction*

In the first stage of the algorithm, the audio signal is analyzed and spectral peaks (sinusoids) are extracted [16, 17]. This process is comprised of three main steps: first a time-domain equal loudness filter is applied [23], which has been shown to attenuate spectral components belonging primarily to non-melody sources [17]. Next, the short-time Fourier transform is computed with a 46ms Hann window, a hop size of 2.9ms and a $\times 4$ zero padding-factor. At each frame the local maxima (peaks) of the spectrum are detected. In the third step, the estimation of the spectral peaks' frequency and amplitude is refined by calculating each peak's instantaneous frequency (IF) using the phase vocoder method [4] and re-estimating its amplitude based on the IF. The detected spectral peaks are subsequently used to compute a representation of pitch salience over time: a *salience function*. The salience function is based on harmonic summation with magnitude weighting, and spans a range of almost five octaves from 55Hz to 1760Hz. Further details are provided in [17].

In the next stage, the peaks of the salience function are grouped over time using heuristics based on auditory streaming cues [1]. This results in a set of pitch contours, out of which the contours belonging to the melody need to be selected. The contours are automatically analyzed and a set of contour characteristics is computed. In the final stage of the system, the contour characteristics and their distributions are used to filter out non-melody contours. The distribution of contour salience is used to filter out pitch contours at segments of the song where the melody is not present. Given the remaining contours, we compute a rough estimation of the melodic pitch trajectory by averaging at each frame the pitch of all contours present in that frame, and then smoothing the result over time using a sliding mean filter. This mean pitch trajectory is used to minimise octave errors (contours with the correct pitch class but in the wrong octave) and remove pitch outliers (contours representing highly unlikely jumps in the melody). Finally, the melody F0 at each frame is selected out of the remaining pitch contours based on their salience. A full description of the melody extraction algorithm, including a thorough evaluation, is provided in [16].

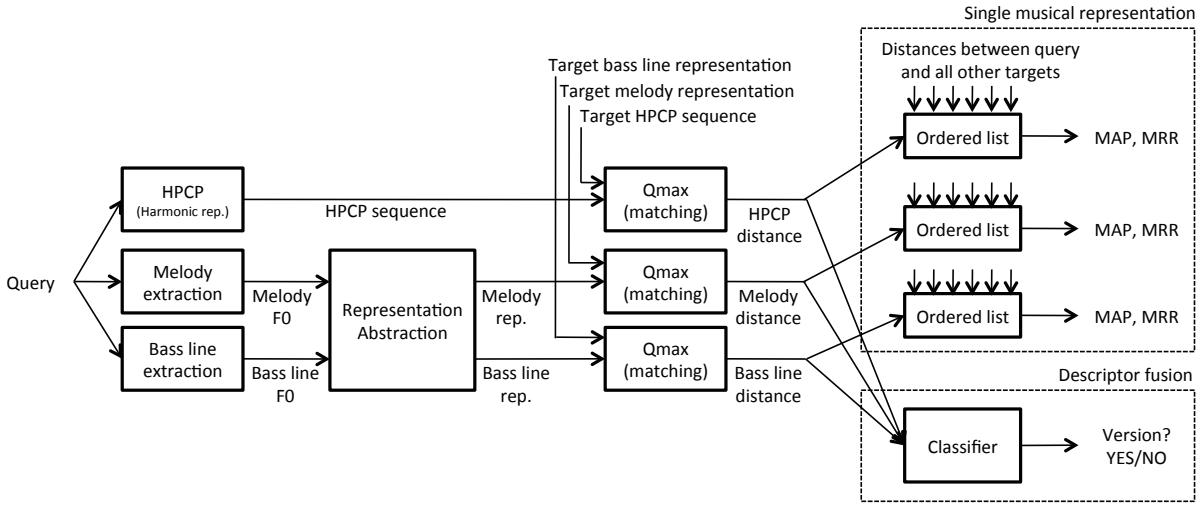

Figure 1: Matching process using either a single musical representation (top right corner) or descriptor fusion (bottom right corner).

2.2.2 *Bass Line Extraction*

The bass line is extracted by adapting the melody extraction algorithm described above. Instead of applying an equal loudness filter (which attenuates low frequency content), we apply a low-pass filter with a cutoff frequency of 261.6Hz, as proposed in [8]. The window size is increased to 185ms, since for the bass we require more frequency resolution. The salience function is adjusted to cover a range of two octaves from 27.5Hz to 110Hz. As before, the salience peaks are grouped into pitch contours. However, since we do not expect other instruments to compete for predominance in the bass frequency range, the detailed contour characterisation and filtering used for melody extraction is less important in the case of bass line extraction. Therefore, the bass line is selected directly from the generated contours based on their salience.

2.2.3 *Representation*

Once the melody and bass line sequences are extracted, we must choose an adequate representation for computing music similarity or, in the case of this study, a representation for detecting versions of the same musical piece. Since the matching algorithm can handle transposition, a first guess might be to use the extracted representation as is, i.e. to compare the F0 sequences directly. However, initial experiments showed that this (somewhat naïve) approach is unsuccessful.

When considering the task of version identification, we must take into consideration what kind of musical information is maintained between versions, and what information is subject to change. In the case of the melody, we can expect the general tonal progression to be maintained. However, more detailed performance information is likely to change between versions. Besides changing the key in which the melody is sung (or played), performers might change the octave in which some notes are sung to adjust the melody to their vocal range. More importantly, the use of expressive effects (such as ornaments, glissando and vibrato) will obviously vary across versions. Overall, this means we should aim for a representation which abstracts away specific performance information and details, whilst maintaining the basic melodic tonal progression. To this effect, we defined the following types of information abstraction:

- Semitone abstraction: quantise pitch information into semitones. This will help in removing some local expressive effects.
- Octave abstraction: map all pitch information into a single octave. This will help in removing potential octave changes of the melody within the piece.
- Interval abstraction: replace absolute pitch information with the difference between consecutive pitch values (a.k.a. delta). This may provide robustness against key changes.

Before applying any abstraction, all frequency values were converted into a cent scale, so that pitch is measured in a perceptually meaningful way. We then ran initial matching experiments comparing the different degrees of abstraction applied to melody sequences: none, semitone, interval, interval+semitone, and semitone+octave (by definition, the interval and octave abstractions are not compatible). For these experiments we used a collection of 76 songs (described in Section 3.1), and evaluated the results as detailed in Section 3.2. We found that results using the semitone+octave abstraction were significantly better than the other types of abstraction, obtaining a mean average precision of 0.73, compared to 0.26-0.28 for all other abstractions considered. Perhaps not surprisingly, we note that this abstraction process is very similar to the one applied when computing chroma features. In particular, the observations above suggest that octave information can be quite detrimental for the task of version identification. For the remainder of the study we use the semitone+octave abstraction for both the melody and bass line descriptors.

The exact abstraction process is as follows: first, all frequency values are converted into cents. Then, pitch values are quantised into semitones, and mapped onto a single octave. Next, we reduce the length of the sequence (whose original hop size is 2.9ms), by summarizing every 150 frames

as a pitch histogram[3]. This produces a shortened sequence where each frame is a 12-bin chroma vector representing a "summary" of the melodic tonal activity over roughly half a second. This window length has been reported to be suitable for version identification by several authors (e.g. [11, 18]). The motivation for the summary step is two-fold: firstly, it reduces the sequence length and therefore reduces the computation time of the matching algorithm. Secondly, it reduces the influence of very short pitch changes which are more likely to be performance specific (e.g. ornamentations). Finally, the chroma vector of each frame is normalised by the value of its highest bin. The steps of the representation abstraction process are depicted in Figure 2 for a melody and in Figure 3 for a bass line.

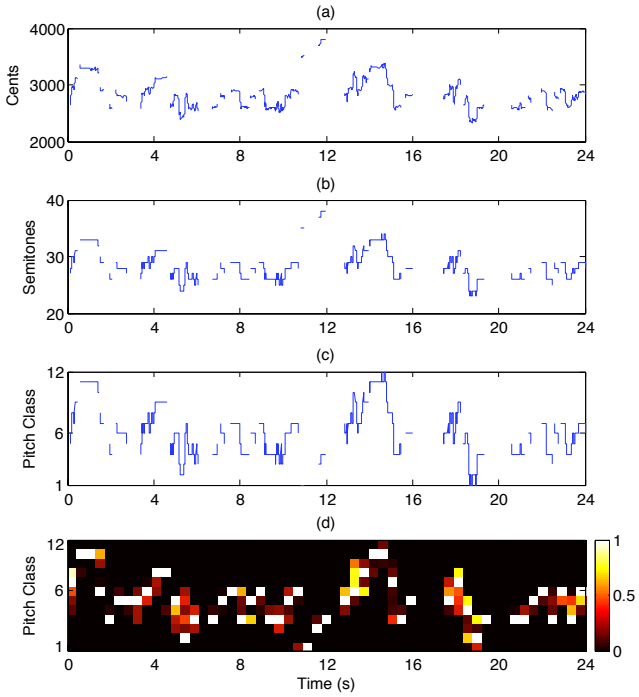

Figure 2: Melody representation abstraction process: (a) melody pitch in cents, (b) quantised into semitones, (c) mapped onto a single octave, (d) summarised as a pitch histogram and normalised.

2.3 Descriptor Sequence Similarity

For deriving a similarity measure of how well two versions match we employ the $Q_{\max}$ method [20]. This is a dynamic programming algorithm which computes a similarity measure based on the best subsequence partial match between two time series. Therefore, it can be framed under the category of local alignment algorithms. Dynamic programming approaches using local alignment are among the best-performing state-of-the-art systems for version identification [18], and have also been extensively used for melody-based retrieval [3].

The $Q_{\max}$ algorithm is based on general tools and concepts of nonlinear time series analysis [10]. Therefore, since the algorithm is not particularly tied to a specific time series, it can be easily used for the comparison of different (potentially multivariate) signals. Furthermore, the $Q_{\max}$ method has provided the highest MIREX accuracies in the version identification task, using only HPCPs [20]. Therefore, it is a very good candidate to test how melody and bass line compare to HPCPs, and to derive competitive version similarity measures to be used in our fusion scheme.

[3]The contribution of each frame to the histogram is weighted by the salience of the melody at that frame, determined by the melody extraction algorithm.

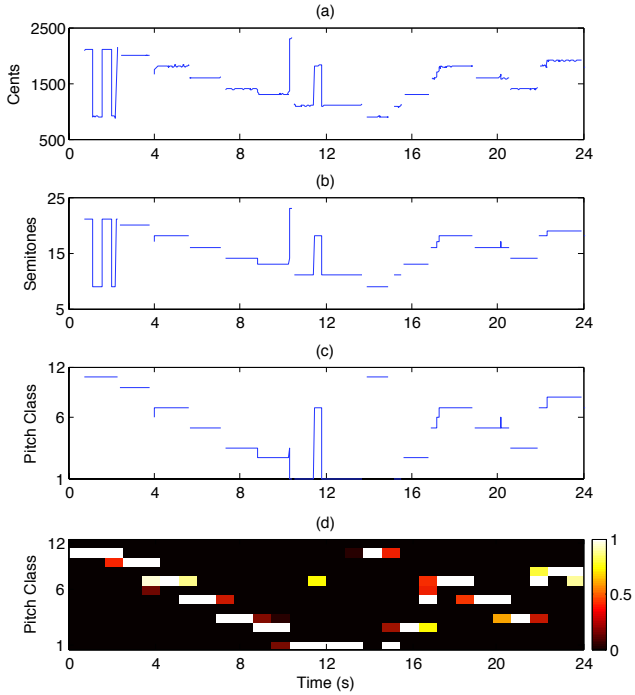

Figure 3: Bass line representation abstraction process: (a) bass line pitch in cents, (b) quantised into semitones, (c) mapped onto a single octave, (d) summarised as a pitch histogram and normalised.

Given a music collection containing various sets of covers, we use the $Q_{\max}$ algorithm to compute the similarity, or in the case of our method, the distance, between every pair of songs in the collection. The resulting pairwise distances are stored in a distance matrix which can then be used either to evaluate the performance of a single descriptor (as explained in Section 3.2), or for descriptor fusion as described in the following section.

2.4 Fusing Descriptors

In addition to evaluating each descriptor separately, the other goal of this study is to see if there is any information overlap between the descriptors, and whether results can be improved by combining them. To this end, we propose a classification approach similar to [14] – each descriptor is used to calculate a distance matrix between all query-target pairs as described in Section 2.3 (4,515,625 pairs in total for the collection used in this study). Every query-target pair is annotated to indicate whether the query and target are versions or not. We then use five different subsets of 10,000 randomly selected query-target pairs to train a classifier for determining whether two songs are versions of the same piece. Note that we ensure each training subset con-

tains an equal amount of pairs that are versions and pairs that are not. In this way we ensure the subsets are not biased and, therefore, the baseline accuracy (corresponding to making a random guess) is 50%. The feature vector for each query-target pair contains the distances produced by the matching algorithm using each of the three representations: HPCP, melody, and bass line (feature columns are linearly normalised between 0 and 1 prior to classification). In this way we can study different combinations of these descriptors, and most importantly, rather than imposing a simple fusion scheme, the combination of different descriptors is determined in an optimal way by the classifier itself. The only potential limitation of the proposed approach is our employment of a late-fusion strategy (as opposed to early-fusion). Nonetheless, in addition to being straightforward, previous evidence has shown that late-fusion provides better results for version identification [5].

The classification is performed using the Weka data mining software [9]. We compare five different classification algorithms: random forest, support vector machines (SMO with polynomial kernel), simple logistic regression, k-star, and Bayesian network [24]. For all classifiers we use the default parameter values provided in Weka. By comparing different classifiers we are able to assess which classification approach is the most suitable for our task. Furthermore, by verifying that any increase (or decrease) in performance is consistent between classifiers, we ensure that the improvement is indeed due to the descriptor fusion and not merely an artefact of a specific classification technique.

3. EVALUATION METHODOLOGY

3.1 Music Collection

To evaluate the performance of our method (using either a single musical representation or the descriptor fusion strategy), we use a music collection of 2125 songs [19]. The collection includes 523 version sets (i.e. groups of versions of the same musical piece) with an average set cardinality of 4.06. The collection spans a variety of genres including pop, rock, electronic, jazz, blues, world, and classical music. We note that the collection is considerably larger than the collection used in the MIREX version identification task, and as such contains a greater variety of artists and styles.

For training the parameters of the $Q_{\max}$ matching algorithm, a small subset of 76 songs from the full collection was used. This 76-song collection was also used for the preliminary experiments on information abstraction outlined in Section 2.2.3. Importantly, we made sure that all songs in this subset have a main melody (and all but 3 have a clear bass line). The full collection, on the other hand, includes versions where there is no main melody (e.g. minus one versions of jazz standards) or no bass line (e.g. singing voice with acoustic guitar accompaniment only), and we can expect this to affect the performance of the melody and bass-line-based representations.

3.2 Evaluation Measures

The distance matrix produced by each descriptor can be used to generate an ordered list of results for each query. The relevance of the results (ideally versions of a query should all appear at the top of the list) can then be evaluated using standard information retrieval metrics, namely the mean average precision (MAP) and the mean reciprocal rank (MRR) [12]. Both measures provide a value between 0 (worst case) and 1 (best case). These metrics, which are standard for evaluating information retrieval systems, are also a common choice for assessing the accuracy of version identification systems based on a single information source [18].

Table 1: Results for single musical representation (76 songs).

Feature	MAP	MRR
HPCP	0.829	0.458
Melody	0.732	0.422
Bass line	0.667	0.387

Table 2: Results for single musical representation (full collection).

Feature	MAP	MRR
HPCP	0.698	0.444
Melody	0.483	0.332
Bass line	0.528	0.355

Since we use classification to fuse different information sources (different descriptors), an alternative evaluation approach is required to evaluate the results obtained using descriptor fusion. Here, the results produced by each classifier are evaluated in terms of classification accuracy (%) using 10-fold cross validation, averaged over 10 runs per classifier. The classification is carried out using a subset of 10,000 randomly selected query-target pairs. We repeat the evaluation process for 5 such subsets (non-overlapping), and average the results over all subsets. As mentioned earlier, the subsets are unbiased and contain the same amount of version pairs as non-version pairs, meaning the random baseline accuracy is 50%. The statistical significance of the results is assessed using the paired t-test with a significance threshold of $p < 0.001$.

4. RESULTS

4.1 Single Musical Representation

We start by comparing the results obtained when using a single descriptor, either the HPCP, the melody, or the bass line. In Table 1 we present the MAP and MRR results for the 76-song subset which was used for training the parameters of the matching algorithm. At first glance we see that the harmonic representation yields better results compared to the melody and bass line descriptions. Nonetheless, the results also suggest that the latter two representations do indeed carry useful information for version identification. Evidence for the suitability of melody as a descriptor for version identification has been reported elsewhere [13, 18, 21]. However, no evidence for the suitability of bass lines has been acknowledged prior to this study. Moreover, no direct comparison between these three musical representations has been performed previously in the literature.

To properly assess the performance of each descriptor, however, a more realistic collection size is required. Thus, we now turn to the results obtained using the full 2125 song collection, presented in Table 2. As expected, there is a drop in performance for all three representations (c.f. [18]). The harmonic representation still outperforms the melody and bass line descriptors, for which the drop in performance

Table 3: Fusion results for the different classifiers considered.

Feature	Random Forest	SMO (PolyKernel)	Simple Logistic	KStar	Bayes Net
H	82.04	87.69	86.42	87.74	87.58
M	69.84	76.73	75.29	77.98	77.90
B	73.34	81.03	78.98	81.31	81.03
M+B	79.80	82.05	80.91	84.62	84.46
H+M	84.29	87.73	86.51	88.01	87.81
H+B	84.72	87.80	86.77	88.32	88.14
H+M+B	86.15	87.80	86.83	88.46	88.24

is more considerable. It is worth noting that the MAP results we obtain using melody or bass line, though lower than those obtained using HPCP, are still considerably higher than those obtained by other version identification systems using similar (and different) types of descriptors [18].

As suggested earlier, one probable reason for the superiority of the HPCP is that some versions simply do not contain a main melody, and (though less often) some songs do not contain a bass line (e.g. a singer accompanied by a guitar only). Still, as seen in the results for the 76-song subset, even when the melody and bass line are present, the HPCP produces better matching results in most cases. This can be attributed to the different degree of modification applied to each musical representation across versions: whilst some versions may apply reharmonisation, in most cases the harmony remains the least changed out of the three musical representations. Differences in the melody and bass line may also be increased due to transcription errors, an additional step which is not necessary for computing the HPCP.

Though the HPCP is considered a harmony based descriptor, it is interesting to ask to what degree is the information it encapsulates different from the melody and bass line descriptors. Since the HPCP is computed using the complete audio mix, it is possible that the melody and bass line are to some degree represented by the HPCP as well. This aspect, albeit very simple, has not been formally assessed before. To answer this question we turn to the second part of the evaluation, in which we examine whether fusing the different descriptors results in improved matching or not.

4.2 Fusion of Musical Representations

The classification results for individual descriptors and fusion approaches are presented in Table 3, where we use "H" for harmony (HPCP), "M" for melody, and "B" for bass line. Several observations can be made from the results. Firstly, we note that for all descriptors and all classifiers the results are significantly above the baseline of 50%. We see that most classifiers perform relatively similarly, though there are some notable differences. In particular, the random forest classifier provides considerably lower results, whilst k-star consistently provides the highest (the difference between the two is for all cases statistically significant). As before, we note that when using only a single representation, the HPCP provides the best performance, followed by the bass line and, finally, the melody.

Perhaps the most interesting results are those obtained by descriptor fusion. For all classifiers, combining the melody and bass line provides increased classification accuracy compared to using either of the two descriptors separately (the increase is statistically significant). Not surprisingly, this confirms that the two musical representations carry complementary information and hence their combination results in increased performance. Still, using melody and bass line together does not outperform using the HPCP on its own. The remaining question is thus whether combining harmony with other descriptors improves classification accuracy.

The results are less straightforward this time. In the case of the random forest classifier, the improvement is clear and statistically significant. However, for the remainder of classifiers the increase is not as considerable. This suggests that the benefits of considering different musical representations are particularly important when the classifier has (relatively) low performance. Nonetheless, if we consider the results of the best performing classifier (k-star), it turns out that the increase in accuracy when combining harmony, melody, and bass line compared to harmony alone is in fact statistically significant. Still, the small increase in accuracy (less than 1%) indicates that the HPCP, to a great extent, carries overlapping information with the melody and bass line.

5. CONCLUSION AND DISCUSSION

To date, the use of different musical representations for computing version similarity has not received the attention it deserves. In this paper we have taken a necessary step in this research direction, which not only holds the promise of improving identification accuracy, but also improving our understanding of the relationship between different musical cues in the context of music similarity. Three types of descriptors were compared in this study, related to the harmony, melody, and bass line. We studied different degrees of abstraction for representing the melody and bass line, and found that abstracting away octave information and quantising pitch information to a semitone level are both necessary steps for obtaining useful descriptors for version identification. The new melody and bass line descriptors were evaluated on a relatively large test collection, and shown to carry useful information for version identification. Combined with the proposed matching algorithm, our melody and bass line descriptors obtain MAP results comparable to (and in some cases higher than) other state-of-the-art version identification systems. Still, it was determined that in most cases the harmony based descriptor gives better matching accuracy. We have also shown that by using a classification approach for descriptor fusion we can improve accuracy, though the increase over using harmony alone is (albeit significant) very small.

To better understand how these different musical representations can complement each other, we manually examined cases where the melody or bass line descriptors produced better matching results than the HPCP. In Figure 4

we present three distance matrices of 10 queries compared to 10 targets, where the same 10 songs are used both as the queries and the targets. The three distance matrices are computed using (a) HPCP, (b) melody, and (c) bass line. The distances in each matrix are normalised by the greatest value in each matrix so that they are visually comparable. Cells for which the query and target are versions of the same musical piece are marked with a black box.

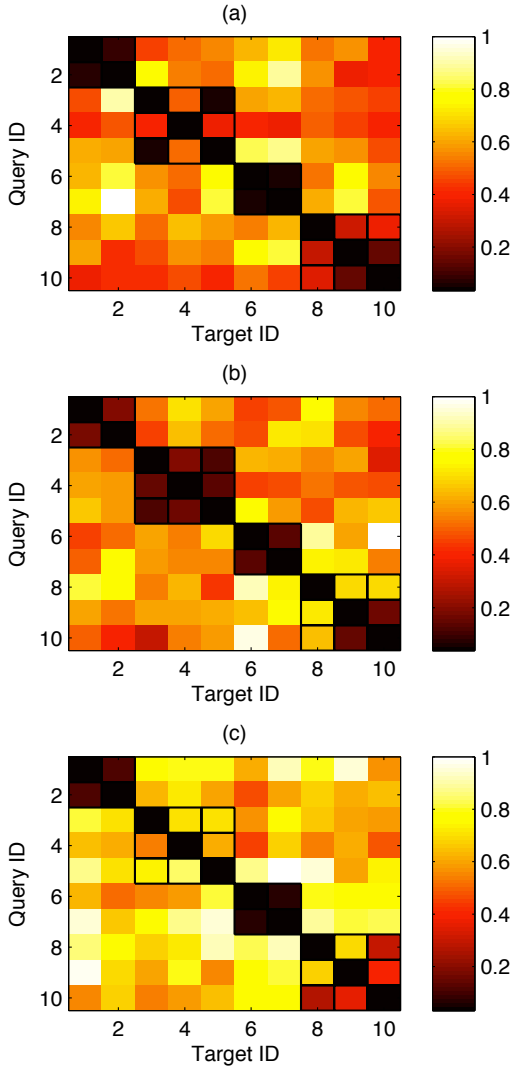

Figure 4: Distance matrices for 10 query and 10 target pieces, produced using: (a) HPCP, (b) melody, and (c) bass line.

An example where melody works better than the HPCP can be seen for the version group with IDs 3, 4, and 5. We see that when using the HPCP, song 4 is considered relatively distant from songs 3 and 5 (light color), whilst the distance is much smaller (darker colour) when using the melody. The three songs are different versions of the song "Strangers in the Night" popularized by Frank Sinatra. Listening to the songs we found that whilst versions 3 and 5 have relatively similar orchestral arrangements, version 4 includes several reharmonisations and entire sections where the melody is played without any accompaniment. It is clear that in such a case using the melody on its own will produce smaller distances between the versions. The bass line descriptor on the other hand does not work well in this example, for the very same reasons.

Another interesting example is provided by the version group with IDs 8, 9 and 10. The three songs are different versions of the song "White Christmas" by Irving Berlin, made famous by Bing Crosby back in 1941. Here we see that whilst song 8 is poorly matched to songs 9 and 10 using either HPCP or melody, it is well matched to song 10 when we use the bass line. When listening to the songs we found that unlike versions 9 and 10, in version 8 there are sections where the melody is solely accompanied by the bass line. In other parts of the song the accompaniment, played by a string section, consists of melodic motifs which interleave with the singing. Furthermore, unlike the more traditional vocal renditions in 9 and 10, the melody in 8 is sung in a more "talk-like" fashion, which combined with the predominant melodic motifs of the string section causes greater confusion in the melody extraction. The various aforementioned differences explain why in this case the bass line succeeds whilst the melody and HPCP do not perform as well. Curiously, whilst song pairs 8-10 and 9-10 are well matched using the bass line, the pair 8-9 is not. Though investigating the exact cause for this inequality is beyond the scope of this study, a possible explanation could be the greater degree of transcription errors in the extracted bass line of song 9. Since the distance computation is not metric, it is possible for transcription errors to have a greater effect on the matching of some songs compared to others.

The results above show that, while in most cases the HPCP (most closely related to the harmony) is the most reliable musical representation for version matching, the melody and bass line can provide useful information in cases where the harmony undergoes considerable changes or is otherwise completely removed (e.g. a-capella singing in unison). Although this observation may seem somewhat obvious, approaches for version matching using descriptor fusion such as [5] and the one proposed in the current study do not take this into account since they always use all descriptors even when one of them may not be appropriate. Thus, a potential approach for improving matching accuracy would be, rather than always using all descriptors, to first attempt to determine which descriptors will provide the most reliable matching results and then use only those. For example, if we detect that one version has accompaniment and the other does not, we might decide to use just the melody rather than melody, bass line and harmony.

Whilst the generality of the matching algorithm employed in this study (Section 2.3) means it can be easily adapted to different types of times series, it is still relevant to ask whether it is the most appropriate matching approach for the melody and bass line sequences. Since the algorithm was originally designed to work with chroma features (HPCPs), it is possible that it introduces a slight bias towards this type of time series. Another conjecture is that the intrinsic lower dimensionality of the melody and bass line features may in part be the cause for the reduced performance of these features. One of our goals for future work will be to address these questions by evaluating and comparing different matching algorithms with the melody and bass line representations proposed in this study.

Finally, the results of the study presented here suggest

that our approach could be successfully applied in the related task of query-by-humming (QBH). Currently, QBH systems (in which a sung or hummed query is matched against a database of melodies) require a large amount of manual labour for the creation of the melody database [3]. In this paper we have shown how combining state-of-the-art melody extraction and version identification systems can be used to automatically generate the melody database and perform the matching. This means that, with some adaptation, our method could be used to create a fully automated QBH system. The melody of the candidate pieces could be extracted with the same algorithm we use here. Furthermore, in a realistic situation, the queries would consist of monophonic melodies sung (or hummed) by the user, which would be easier to transcribe (no interference from other instruments). In the future we intend to test this hypothesis by evaluating the proposed approach in a QBH context.

6. ACKNOWLEDGMENTS

This research was funded by: Programa de Formación del Profesorado Universitario (FPU) of the Ministerio de Educación de España, Consejo Superior de Investigaciones Científicas (JAEDOC069/2010), Generalitat de Catalunya (2009-SGR-1434) and the European Commission, FP7 (Seventh Framework Programme), ICT-2011.1.5 Networked Media and Search Systems, grant agreement No. 287711.

7. REFERENCES

[1] A. Bregman. *Auditory scene analysis*. MIT Press, Cambridge, Massachusetts, 1990.

[2] N. J. Bryan and G. Wang. Musical influence network analysis and rank of sampled-based music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, Florida, 2011.

[3] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis. A comparative evaluation of search techniques for query-by-humming using the MUSART testbed. *Journal of the American Society for Information Science and Technology*, February 2007.

[4] J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell Systems Technical Journal*, 45:1493–1509, 1966.

[5] R. Foucard, J.-L. Durrieu, M. Lagrange, and G. Richard. Multimodal similarity between musical streams for cover version detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5514–5517, 2010.

[6] E. Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[7] E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*, 18(3), 2006.

[8] M. Goto. A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43:311–329, 2004.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1), 2009.

[10] H. Kantz and T. Schreiber. *Nonlinear time series analysis*. Cambridge University Press, Cambridge, UK, 2nd edition, 2004.

[11] C. C. S. Liem and A. Hanjalic. Cover song retrieval: a comparative study of system component choices. In *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pages 573–578, 2009.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[13] M. Marolt. A mid-level representation for melody-based retrieval in audio collections. *Multimedia, IEEE Transactions on*, 10(8):1617–1625, Dec. 2008.

[14] S. Ravuri and D. P. W. Ellis. Cover song detection: From high scores to general classification. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 65–68, 2010.

[15] J. Salamon and E. Gómez. Melody extraction from polyphonic music: Mirex 2011. In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, extended abstract, Miami, USA, October 2011.

[16] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. on Audio, Speech and Language Processing*, In Press (2012).

[17] J. Salamon, E. Gómez, and J. Bonada. Sinusoid extraction and salience function design for predominant melody estimation. In *Proc. 14th Int. Conf. on Digital Audio Effects (DAFX-11)*, Paris, France, September 2011.

[18] J. Serrà, E. Gómez, and P. Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In Z. W. Raś and A. A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, volume 274 of *Studies in Computational Intelligence*, chapter 14, pages 307–332. Springer, Berlin, Germany, 2010.

[19] J. Serrà, H. Kantz, X. Serra, and R. G. Andrzejak. Predictability of music descriptor time series and its application to cover song detection. *IEEE Trans. on Audio, Speech and Language Processing*, 20(2):514–525, 2012.

[20] J. Serrà, X. Serra, and R. G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.

[21] W.-H. Tsai, H.-M. Yu, and H.-M. Wang. Using the similarity of main melodies to identify cover versions of popular songs for music document retrieval. *Journal of Information Science and Engineering*, 24(6):1669–1687, 2008.

[22] R. Typke. *Music Retrieval based on Melodic Similarity*. PhD thesis, Utrecht University, Netherlands, 2007.

[23] E. Vickers. Automatic long-term loudness and dynamics matching. In *Proc. of the Conv. of the Audio Engineering Society (AES)*, 2001.

[24] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, Waltham, USA, 2nd edition, 2005.