VERSION DETECTION FOR HISTORICAL MUSICAL AUTOMATA

Bernhard Niedermayer[1]
[1]Dept. of Computational Perception
Johannes Kepler University Linz
music@jku.at

Gerhard Widmer[1,2]
[2]Austrian Research Institute for
Artificial Intelligence, Vienna
music@jku.at

Christoph Reuter[3]
[3] Dept. of Musicology
University of Vienna
christoph.reuter@univie.ac.at

ABSTRACT

Musical automata were very popular in European homes in the pre-phonograph era, but have attracted little attention in academic research. Motivated by a specific application need, this paper proposes a first approach to the automatic detection of versions of the same piece of music played by different automata. Due to the characteristics of the instruments as well as the themes played, this task deviates considerably from cover version detection in modern pop and rock music. We therefore introduce an enhanced audio matching and comparison algorithm with two main features: (1) a new alignment cost measure – *Off-Diagonal Cost* – based on the Hough transform; and (2) a *split-and-merge strategy* that compensates for major structural differences between different versions. The system was evaluated on a test set comprising 89 recordings of historical musical automata. Results show that the new algorithm performs significantly better than the reference system based on Dynamic Time Warping and chroma features without the above-mentioned new features, and that it may work well enough to be practically useful for the intended application.

1. INTRODUCTION

Over the past 30 years, the *Phonogram Archive* of the Austrian Academy of Sciences (www.phonogrammarchiv.at) has compiled a large collection of recordings of a variety of historical musical automata (e.g., musical boxes, 'flute clocks', violin playing automata, barrel organs,...). [1] Thousands of recordings have been collected, representing a large repertoire of music that was popular during certain periods of the 18th and 19th centuries – from opera arias to folk songs. Musical automata were widespread in private homes long before the invention of the phonograph. Such a collection is thus a unique source of information to study musical tastes, popular repertoire, and other musical trends during parts of the pre-phonograph era.

A major problem with the audio collection is that while the instruments themselves are relatively well documented, the musical pieces being played are often unknown. At the same time, there is reason to believe that there will be multiple versions and interpretations of many of the songs in the collection. One way to automatically identify some of the unknown pieces would thus be to systematically search for subsets of recordings that might represent the same piece and then unify their meta-information. Given the size of the collection and the specific characteristics of the recordings (see below), this is a very difficult and tedious task.

The research described here represents a first attempt at developing audio analysis and matching technology that could help with this problem. The goal is a kind of 'cover version' detection: for each recording in the collection, search for other recordings that might represent (parts of) the same piece – perhaps played on an entirely different instrument, in a different key, possibly only sharing some sub-sections of the piece. Recordings of such historical music automata present new challenges to sound and music computing: apart from the sometimes extremely inharmonic sounds of the instruments (see section 3), a central problem is the often extreme ornamentation and/or arpeggiations and other asynchronies that obscure the main melody (to the extent that it is sometimes hard even for experienced listeners to recognize a song, at least at first hearing).

We present here a first pilot study that starts with a small collection of recordings of musical boxes and flute clocks, and with 3 pairs of recordings which are known to pertain to the same composition. We first experiment with fairly 'standard' audio matching technology (chroma features, dynamic time warping), and then, based on insights into specific problems posed by our data, develop and test an enhanced audio matching strategy (including the use of a Hough transform). Experiments show that the latter method improves results considerably and gives us reason to believe that the general problem is not unsolvable.

The remainder of this paper is organized as follows. In section 2 we give a brief overview of related work in the field of (cover) version detection. Section 3 then describes relevant characteristics of the mechanical musical instruments under consideration. The proposed version detection system is explained in Section 4, and the exact similarity measures used are defined in Section 5. An evaluation is presented in Section 6.

2. RELATED WORK

A detailed overview over current version detection systems is given in [4]. Also, an annual comparison of different al-

[1] For a report on the background of this project see http://www.phonogrammarchiv.at/Mechanical_Music/mechreal.html.

gorithms is carried out as part of the MIREX [2] contest. The evaluation is performed on two test sets, one comprising pieces of popular music, the other one consisting of performances of Chopin's mazurkas by different pianists. Best results were obtained in 2009 by [1] and [2]. Both approaches are based on chroma descriptors. [1] used cross recurrence plots based on a state space representation of two songs to rank songs according to their similarity. [2] calculated three different similarity features – two based on cross-correlation and one based on dynamic programming – in combination with three different tempo assumptions and trained a support vector machine on this data.

A completely different approach was presented in [3], where similarity estimation is not done based on audio features themselves, but on a discrete text representation. Strings are obtained from the feature sequences by clustering all chroma vectors and subsequently replacing each individual vector by the hash value assigned to the cluster it belongs to. The actual query is then performed by exploiting the open-source search engine *Lucene* [3].

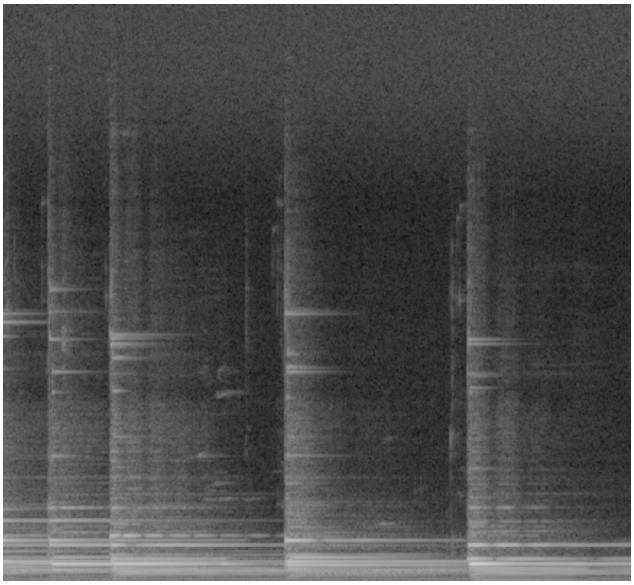
Though not directly focusing on cover version detection, also [5] is relevant to our work. Here, instead of identifying cover versions, the plausibility of audio-to-midi alignments is assessed. Several metrics were investigated, including a *relative path cost* measure that will also be employed in our matching algorithm (see Section 5).

3. ACOUSTIC CHARACTERISTICS OF MUSICAL AUTOMATA

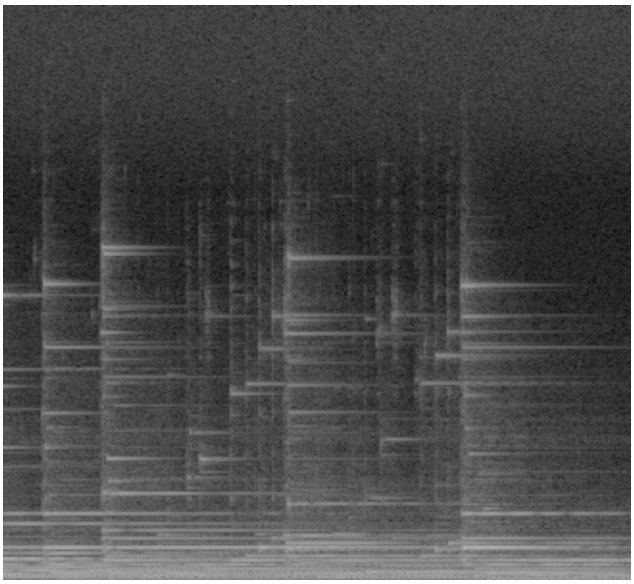
Musical automata come in wide varieties. Computer controlled pianos and similar modern instruments aside, there are musical boxes, flute clocks, violin playing automata of various kinds, barrel organs, etc. – each of them revealing individual characteristics. While flute clocks, for example, are largely consistent with the spectral envelopes one would expect from wind instruments – i.e., harmonics at the odd integer multiples of the fundamental frequency – musical boxes, where metal plates are struck, are highly inharmonic.

Another issue are different arrangements of a same theme for different automata. Besides transpositions into different keys, additional ornamentations or arpeggiations can alter the sound impression of a piece significantly. Figure 1 shows spectrograms of the ending of a theme from Mendelssohn's oratorio *Elias* as played by two different musical boxes. While in (a) only the main melody notes are played, (b) features luscious ornamentations that make it hard even for human listeners to hear the relation to the main theme.

A third challenge for a version detection system are major changes in the structure of the piece. As described in more detail in section 6 (see Table 2), when two recordings relate to the same piece, this does not necessarily mean that they both comprise the same musical sections. In our data there are samples where one musical box plays only the second half of what another one is playing. Moreover, some recordings are *potpourris* (*medleys*) of several themes (e.g., popular melodies from an opera); pairs of such medleys pertaining to the same set of themes will match only in part.



(a)



(b)

Figure 1. The ending of the same theme from Mendelssohn's oratorio *Elias* played by two different musical boxes.

4. VERSION DETECTION

The proposed system works in two steps. First, features are extracted from the audio signals of the individual recordings. Then, a compact similarity measure is obtained, such that pairs of audio files can be ranked accordingly. In doing so, to account for transpositions and major structural changes, each piece is split into several chunks of a fixed

[2] MIREX: Music Information Retrieval Evaluation eXchange (organized by the IMIRSEL at the University of Illinois at Urbana-Champaign) http://www.music-ir.org/mirex

[3] http://lucene.apache.org

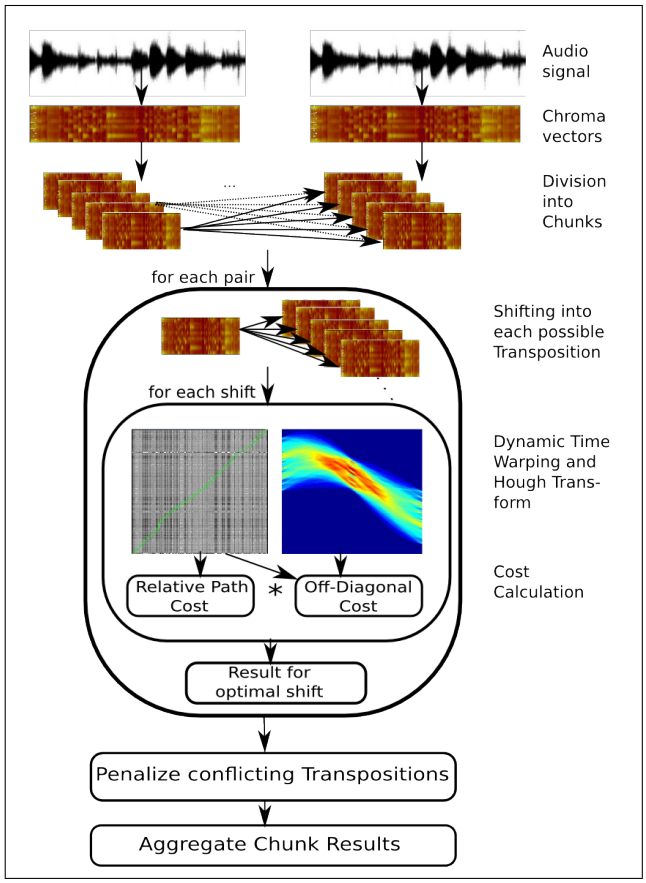


Figure 2. Calculation of the similarity measure

length which are then aligned to each other considering all possible transpositions. The final similarity measure is obtained by accumulating the fitness ratings of these alignments. An overview of the whole process is given in Figure 2.

4.1 Feature Extraction

Chroma vectors are a common feature in cover version detection or audio-to-audio alignment. They have been shown to be robust to several potentially problematic aspects, such as different instrumentation, varying degrees of polyphony, or different recording conditions. A comparative study has been presented in [6], showing that chroma vectors outperform several other features, such as MFCCs or pitch histograms, in an audio matching task.

Chroma vectors consist of a 12-dimensional vector per time frame, representing the relative energies within the individual pitch classes (i.e. C, C#, D,...). The calculation starts by transforming the audio signal into the frequency domain. Then the energies of frequency bins corresponding to the same pitch classes are accumulated. To this end the center frequencies of all bins are folded into the same octave. The mapping onto pitch classes is then done such that a frequency can also contribute to two classes if it lies in the middle of the fundamental frequencies of two adjacent pitches. Therefore, the distance of a bin's folded center frequency $\overline{f}$ to the fundamental f_0 of a certain pitch prototype is determined on the cent-scale as

$$d_{f_0}(\overline{f}) = 1200 * \log_2 \frac{f}{f_0} \tag{1}$$

The energy of $\overline{f}$ then contributes to each prototypical fundamental f_0 with a weighting of

$$w(\overline{f}, f_0) = \begin{cases} 0 & \text{, if } d_{f_0}(\overline{f}) < l \\ \cos\left(\frac{\pi}{2l} d_{f_0}(\overline{f})\right) & \text{, else} \end{cases} \tag{2}$$

In the system proposed here, l was chosen to be 75. Concerning the transform into the spectral domain, an STFT with window length 4096 and a hop size of 1024 was applied. This results in a frequency resolution fine enough to also resolve relatively low pitches and a time resolution enabling an accurate alignment between two pieces.

4.2 Transposition

A common deviation of a version of a piece from its original is a change of key. Often, the reason for these transpositions is to adapt a piece to a different lead instrument or another singer. Transpositions can also be motivated by artistic considerations, such as to give a piece a different mood.

Most similarity measures will inevitably fail if the main key has been changed between two versions of a piece. The one used here, for example, depends on the Cosine distance $d_c(A_i, B_j)$ (see equation 3 in section 4.4) that measures the error made when matching two chroma vectors A_i and B_j. The Cosine distance, however, is not robust to transpositions, i.e., shifts of one of the chroma vectors.

To compensate for possible changes of key, when computing the similarity measures between two feature sequences, each of the 12 shifts is considered. The one yielding the best result is kept and its deviation from the original key is remembered.

4.3 Segmentation

Listening to many recordings of different musical automata has revealed that the lengths of the themes played by these instruments vary significantly from less than 30 seconds up to several minutes. This can also be the case when the original piece or even the particular theme was the same. For instance, there are pairs of musical boxes where one performs only the second half of what the other one is playing.

To address this problem, the recordings were split into fragments of equal length. Instead of comparing two whole pieces, each combination of two such chunks was considered. This might seem inefficient from a computational costs point of view. However, the effects are moderate since the effort required to run the alignment algorithm (which is of complexity $O(n^2)$) and compute the similarity measure (including operations of complexity $O(n^3)$) is reduced accordingly. In addition, the processing of pairs of fragments is fully parallelizable and also, due to the fixed chunk size, the amount of memory needed is limited and independent of the actual length of the full audio recordings.

A fragment length of 25 seconds and an overlap ratio of about 50% have been found to yield good results. The overlap ratio was adapted slightly, such that the last fragments are positioned at the very end of each piece and there is no remainder left. When trying to align pairs of fragments from two pieces, all possible transpositions are considered, and different fragments from the same piece are allowed to be transposed by different numbers of semitones. Conflicts arising from this are handled later when the results from the individual fragment pairs are merged (see Section 5.3).

4.4 Alignment

To calculate a compact similarity measure, two sequences of features have to be aligned first. A well known method to perform this task is Dynamic Time Warping (DTW). Here, one starts by defining a cost function, measuring the error made when aligning a frame A_i within one feature sequence A to a frame B_j within another feature sequence B. Preliminary experiments have shown that the cosine distance d_c between two chroma vectors, defined as

$$d_c(A_i, B_j) = \frac{\langle A_i, B_j \rangle}{|A_i| * |B_j|} \tag{3}$$

yields good results.

Given the cost function, the next step is to calculate the dissimilarity matrix D, where each element $D_{i,j}$ equals $d_c(A_i, B_j)$. From this data an accumulated cost matrix C can be computed efficiently. Here, each cell $C_{i,j}$ gives the minimum cost of an alignment between the two subsequences $A_0 - A_i$ and $B_0 - B_j$. Starting from $C_{0,0} = D_{0,0}$, it is calculated iteratively by

$$C_{i,j} = \min \begin{cases} C_{i-1,j-1} + D_{i,j} \\ C_{i-1,j} + D_{i,j} \\ C_{i,j-1} + D_{i,j} \end{cases} \tag{4}$$

The alignment is finally determined by the path through D that leads to the optimal global alignment cost given by $C_{N-1,M-1}$. Backtracking the path can easily be done by remembering which of the three options in equation 4 was used in each step of the calculation of C. For a more detailed description of the Dynamic Time Warping algorithm and its properties or a possible refinement strategy, we refer the interested reader to [7] and [5] respectively.

5. SIMILARITY MEASUREMENT

Given the alignment, an intuitive similarity measure for two pieces of music would be the average cost along the alignment path. However, preliminary experiments have shown that this measure is too simple. One the one hand, it allows for insertions or deletions of notes and thus a change of melody up to certain amount, while on the other hand, it penalizes a change in the structure of a piece. In addition, although chroma vectors are relatively robust to these effects, higher average alignment costs can still be caused be changes in instrumentation or accompaniment.

5.1 Relative Path Cost

One approach to account for differences in instrumentation or accompaniment is to calculate the average cost along the alignment path *in relation to the overall average cost* $\overline{D}$ over all $D_{i,j}$. Assuming that the two pieces of music under consideration share some similar sections, the pitch classes in the chroma feature will have similar underlying distributions. In such cases the overall average cost $\overline{D}$ is a good baseline, estimating the average path cost of a random alignment. It can account for specifics of the two audio signals that the chroma feature is not invariant to, such as changed recording conditions, varying levels of noise, or different arrangements. In [5] a similar metric is proposed and shown to outperform others that fully rely on absolute costs along the alignment path.

5.2 Off-Diagonal Cost

An inherent property of the DTW algorithm is that it is robust to small deviations of one piece compared to another one. This is necessary in order to compensate for different performance styles or playing errors. However, the DTW algorithm's flexibility can also result in relatively low alignment costs when two meloies are compared that are really to be considered different – especially in cases where one performance contains a lot of additional ornamentation. The algorithm may decide to delete the main melody notes while matching the auxiliary notes, producing a good-looking alignment of what are really different melodies.

This undesired behavior can be detected by investigating the shape of the alignment path. Matching melodies, although with varying ornamentations, are likely to result in approximately linear paths along the main diagonal. On the other hand, different melodies which still yield low global alignment costs are characterized by major stretches and compressions of notes. This corresponds to significant horizontal or vertical segments within the alignment path. To measure this effect, we define the *Off-Diagonal Cost* as the deviation of the optimal alignment path from the best strictly linear alignment path.

A method to retrieve linear segments, known from the field of image processing, is the Hough transform [8, 9]. It transforms points $(i, j)^T$ from the image domain – in our case, the element-wise inverse of the dissimilarity matrix D – into the Hough space H, where each point $(\rho, \theta)^T$ represents a line given by θ – the angle with respect to the image domain's positive x-axis – and ρ – the distance from the origin, such that

$$\rho - i \cos\theta - j \sin\theta = 0 \tag{5}$$

A single point $(i, j)^T$ of the image domain lies on infinitely many lines. Thus, its Hough transform is a function $r_{i,j}$, following from equation 5 as

$$r_{i,j}(\theta) = i \cos\theta + j \sin\theta \tag{6}$$

$r_{i,j}$ is a sinusoid of period 2π having a magnitude of $|(i, j)^T|$ and a phase of $\arctan j/i$.

In the discrete case the Hough space is sampled and represented by an accumulator array $\hat{H}$ of size $N \times M$. The function $r_{i,j}$ then becomes a set of corresponding cells, defined as

$$R_{i,j} = \{\hat{H}_{t,P^{-1}(h_{i,j}(\Theta(t)))} : t \in [0, N-1]\} \quad (7)$$

where $\Theta(t)$ is the angle θ corresponding to index t and $P^{-1}(\rho)$ is the sampling index ρ resolves to.

When applying the Hough transform to the dissimilarity matrix D, for each element $(i,j)^T$, all accumulator cells in $R_{i,j}$ are increased by $D_{i,j}^{-1}$. High values within the resulting $\hat{H}$ indicate prominent lines in the image domain. Figure 3 shows the accumulator arrays $\hat{H}$ that result from comparing two versions of the same piece of music and and two independent recordings, respectively.

In the proposed system, the size of the accumulator array was set to the size of the dissimilarity matrix D. Doing so yields fine resolutions if D is large and coarser resolutions if input data is scarce. In addition, the angle θ was restricted to have a maximum deviation from the main diagonal of $\pm 30°$. Since the dominant line is assumed to be the best linear alignment path, this restricts the slopes – i.e., the tempo deviations – to reasonable values.

In summary, the Hough transform is used as a line detector. Applied on the inverse alignment costs, it finds linear segments within the dissimilarity matrix D along which the two pieces under consideration match relatively well. Such alignments allow only for an offset between the beginnings and a constant tempo change. The highest value of the accumulator array $\hat{H}$ represents the best alignment under these constraints.

To finally calculate the Off-Diagonal cost, for each point along the alignment path as computed by the DTW algorithm, the shortest distance to the linear one is measured. These distances are then squared and averaged to obtain the final cost measure. In doing so, the first and last 5% of the paths are disregarded, so as not to penalize different offsets.

5.3 Data Merging

So far, we have proposed splitting the audio recordings into chunks to compensate for structural changes between versions of the same piece of music, described a basic method to align two such chunks, and introduced two similarity measures that indicate whether the alignment has indeed matched notes of a same melody. To finally obtain a compact similarity measure these pieces of information need to be integrated.

First, the two similarity measures need to be combined. The Relative Path cost describes the difference between feature vectors along the alignment path in comparison to a specific baseline influenced by changes in instrumentation or recording conditions between the two audio recordings. The Off-Diagonal cost, on the other hand, measures the severity of changes in rhythm or local tempo. Preliminary experiments have shown that simply taking the product of these two measures results in a meaningful aggregated matching cost.

Next, conflicts between evidence for transpositions of segments by different intervals need to be resolved. To this end, a majority vote is taken from the n most similar pairs of segments. To determine this number n, the lengths of valid alignment paths given different scenarios are considered. Let one piece be split into a chunks and the one it is compared to into b fragments, then the minimum number n_{min} of pairs needed to fully reflect an alignment path over the whole recordings along the diagonal is $\max(a,b)$. On the other hand, the maximum number n_{max} of pairs needed to cover an alignment path in the worst case – if it consists of many horizontal and vertical segments – is $a+b-1$. Therefore a reasonable number of pairs of chunks to take into consideration is chosen as $n = \alpha \max(a,b)$ with $1 \leq \alpha < 2$, depending on how much deviation from the main diagonal an overall alignment path should be allowed to exhibit.

The main idea of splitting pieces into chunks was to compensate for major structural changes, e.g. the insertion or deletion of a prominent section of a piece. A large difference in performance time would be a cue for such a modification. Therefore, instead of forcing two whole pieces to be aligned, parts of the longer recordings are allowed to be left out. To this end, the length of the shorter recording was chosen to be the determining factor, resulting in $\tilde{n} = \alpha \min(a,b)$. We set α to 1.5, to still give consideration to deviations in tempo. Experiments have shown that the number $\tilde{n}$ of pairs taken into account outperforms the original n.

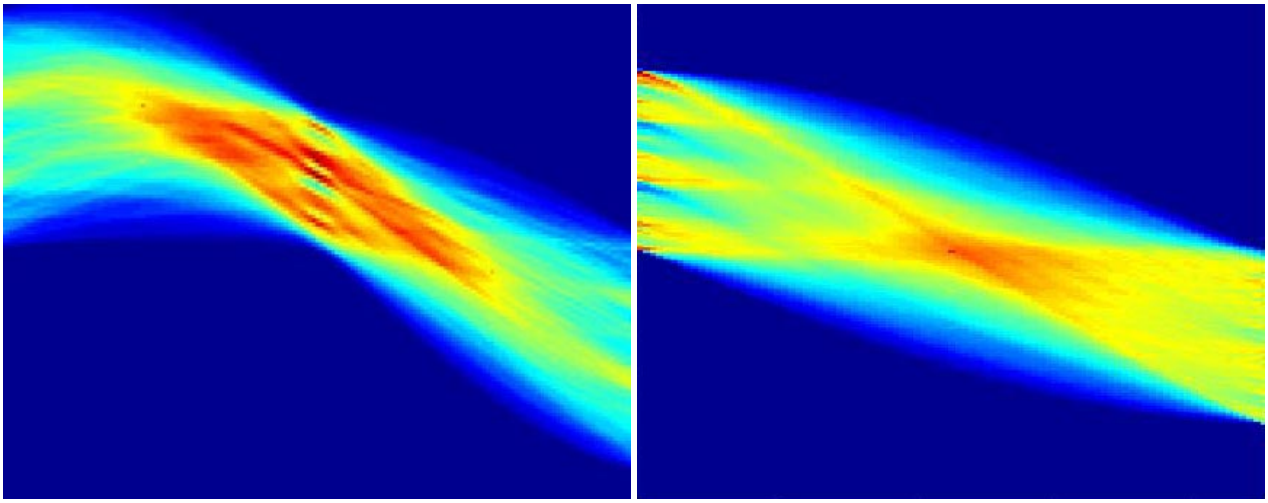
Once the main transposition interval has been obtained via voting among these $\tilde{n}$ selected pairs of segments, deviating transposition intervals of individual pairs are penalized by multiplying the respective matching costs by a factor β. In the context of our data, $\beta = 2$ is sufficient to prevent low matching costs as a result of arbitrarily many different transpositions. The final matching cost is then obtained by averaging over the costs of the $\tilde{n}$ most similar pairs of fragments.

6. EVALUATION

6.1 Data Corpus

The data corpus used for the evaluation comprises recordings of 89 mechanical music instruments, collected by the *Phonogram Archive* of the Austrian Academy of Sciences. About half of the pieces are played by flute clocks while the other half is performed by musical boxes. As described in Section 3, there are also significant differences in performance style and accompaniment. While some instruments only play the main melody notes, others make use of rich ornamentations.

Amongst the test data are three pairs of recordings pertaining to the same underlying piece (all of them performed by music boxes). They comprise (several) themes from Auber's opera *Fra Diavolo*, Mendelssohn's oratorio *Elias*, and Haydn's oratorio *The Creation*, respectively. These are the 'cover versions' we wish to discover in the experiment.



(a) (b)

Figure 3. Hough transform of two versions of the same piece of music (a) and two independent recordings (b).

Query Piece	Proposed System		Standard Alg	
	Ver. 1	Ver. 2	Ver. 1	Ver. 2
Fra Diavolo	1	3	5	8
Elias	4	3	12	6
The Creation	5	9	8	11

Table 1. Rank of the corresponding version of the same piece within a list of 88 candidates, i.e, rank of version 2 when the query was version 1 and the other way around. 'Standard Alg' refers to a DTW-based matching algorithm without our two new extensions *split-and-merge* and *off-diagonal cost* (but with *relative path cost*, which was already proposed by [5]).

Query Piece	Ver. 1	Ver. 2
Fra Diavolo	2:13	3:25
Elias	0:58	1:53
The Creation	0:58	1:55

Table 2. Performance times of different versions of same piece of music.

6.2 Results

In analogy to the MIREX audio cover song detection task, each of the 3×2 test recordings is in turn used as the query file. The respective ranking of the 88 remaining candidates is then examined and given in Table 1.

Although there is only one perfect match, we consider the results promising, given the nature of the data. In comparison to popular music, a different version of a piece is not only played on a different instrument, but, as can be presumed from the durations in Table 2, there are significant differences in which subset of the underlying piece is performed at all.

Table 1 also shows that the proposed system significantly outperforms a reference system – a 'standard' audio matching algorithm without the *split-and-merge* approach and the *off-diagonal cost* (but with *relative path cost*, which was already proposed by [5]). Looking at the mean rank of the corresponding version of the 6 query recordings, the two systems achieve values of 4.2 and 8.3 respectively.

Clearly, our algorithm is not precise enough for the fully automatic identification of matching recordings in a music collection as difficult as the one we are targeting. (In fact, neither are other state-of-the-art cover version detection algorithms in their domains of pop and rock music.) However, it may be useful as a component in an interactive search process. Also, we do have some ideas for possible improvements via quasi-transcription and higher-level representations (see below).

7. CONCLUSIONS

The paper has presented a first approach towards the automatic detection of versions of the same piece of music played by different musical automata. We have described the difficulties arising from the characteristics of these kinds of musical instruments and the different ways of arranging pieces for them. The proposed system is designed to be robust to the degrees of freedom instrument makers have, such as implementation of different subsections of the same theme, transpositions, or slight variations in tempo. To this end, each piece is split into several chunks which are compared separately, allowing each possible transposition. The comparison is based on an alignment obtained by the DTW algorithm and is evaluated via a similarity measure that combines match quality along the alignment path and plausibility of this path itself. Results from individual pairs of chunks are then combined to a final judgment about the similarity between two recordings.

The data set used for testing comprised 89 pieces including 3 pairs of recordings which share the same original. (Finding these three pairs of matching recordings in the

collection involved quite some effort.) That leaves us with only a small number of possible test setups. Future work will focus on extending the data set, including 'ground truth' concerning subsets of recordings that relate to the same piece.

Generally, historical mechanical instruments are limited in various ways – for instance, they generally have a rather restricted tonal range, little freedom or variation in terms of how tones are produced or modulated, etc. That might make it possible to perform some kind of automatic transcription, or at least a mapping onto a high-level representation (e.g., a list of played pitches), which again would facilitate a comparison at a higher level. On the other hand, each instrument has different tonal characteristics. Therefore, for each piece, individual tone models would need to be learned in an unsupervised manner. Given that the length of many recordings is less than 30 seconds, this is error prone as well. Still, the idea of introducing a higher-level representation of the audio signals is intriguing and will be investigated in future work.

Acknowledgments

This research is supported by the Austrian Research Fund (FWF) under grants TRP109-N23 and Z159. We would like to thank Helmut Kowar (Phonogrammarchiv of the Austrian Academy of Sciences) for providing the audio recordings for our experiments.

8. REFERENCES

[1] J. Serrà, X. Serra, and R. G. Andrezejak: "Cross Recurrence Quantification for Cover Song Identification", *New Journal of Physics*, Vol. 11, Issue 9, 093017, 2009.

[2] S. Ravuri and D. P. W. Ellis: "Cover Song Detection: From High Scores to General Classification", *Proceedings of the IEEE International Conference on Audio, Speech, and Signal Processing (ICASSP)*, pp. 65–68, Dallas, 2010.

[3] E. Di Buccio, N. Montecchio, and N. Orio: "A scalable cover identification engine", *Proceedings of the International Conference on Multimedia (MM'10)*, Firenze, Italy, 2010.

[4] J. Serrà, E. Gómez, and P. Herrera: "Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation and Beyond", *Advances in Music Information Retrieval (Z. W. Ras and A. A. Wieczorkowska Eds.), Studies in Computational Intelligence*, Vol. 247, pp. 307–332, Springer, Berlin, 2010.

[5] R. J. Turetsky and D. P. W. Ellis: "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses", *Proceedings of the 4th International Symposium of Music Information Retrieval (ISMIR)* Baltimore, MD, 2003.

[6] N. Hu, R. B. Dannenberg, and G. Tzanetakis: "Polyphonic Audio Matching and Alignment for Music Retrieval", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, 2003.

[7] L. R. Rabiner and B. H. Juang: "Fundamentals of speech recognition". Prentice Hall, Englewood Cliffs, NJ, 1993.

[8] J. Princen, J. Illingworth, and J. Kittler: "A formal definition of the Hough transform: Properties and relationships", *Journal of Mathematical Imaging and Vision*, Vol. 1, Issue 2, pp. 153–168, Springer Netherlands, 1992.

[9] M. Atiquzzaman and M. W. Akhtar: "Complete line segment description using the Hough transform", *Image and Vision Computing*, Vol. 12, Issue 5, pp. 267–273, 199.