A Case Based Approach to Expressivity-aware Tempo Transformation

Maarten Grachten, Josep-Lluís Arcos and Ramon López de Mántaras
IIIA-CSIC - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain.
Vox: +34-93-5809570, Fax: +34-93-5809661
Email: {maarten,arcos,mantaras}@iiia.csic.es

Abstract

The research presented in this paper is focused on global tempo transformations of music performances. We are investigating the problem of how a performance played at a particular tempo can be rendered automatically at another tempo, while preserving naturally sounding expressivity. Or, differently stated, how does expressiveness change with global tempo. Changing the tempo of a given melody is a problem that cannot be reduced to just applying a uniform transformation to all the notes of a musical piece. The expressive resources for emphasizing the musical structure of the melody and the affective content differ depending on the performance tempo. We present a case-based reasoning system called TempoExpress and will describe the experimental results obtained with our approach.

1 Introduction

It has been long established that when humans perform music from score, the result is never a literal, mechanical rendering of the score (the so called nominal performance). As far as performance deviations are intentional (that is, they originate from cognitive and affective sources as opposed to e.g. motor sources), they are commonly thought of as conveying musical expression. Two main functions of musical expression are generally recognized. Firstly, expression is used to clarify the musical structure (in the broad sense of the word: this includes metrical structure [29], but also the phrasing of a musical piece [9], harmonic structure [25] etc.). Secondly, expression is used as a way of communicating, or accentuating affective content [21, 23, 10].

Furthermore, when a specific musician play the same piece at different tempos, the deviations from the nominal performance tend to differ. Changing the tempo of a given melody is a problem that cannot be reduced to just applying a

uniform transformation to all the notes of a musical piece [5, 20]. When a human performer plays a given melody at different tempos, she does not perform uniform transformations. On the contrary, the relative importance of the notes will determine, for each tempo, the performer's decisions. For instance, if the tempo is very fast, the performer will, among other things, tend to emphasize the most important notes by not playing the less important ones. Alternatively, in the case of slow tempos, the performer tends to delay some notes and anticipate others.

The research presented in this paper is focused on global tempo transformations of music performances. We are investigating the problem of how a performance played at a particular tempo can be rendered automatically at another tempo, without the result sounding unnatural. Or, differently stated, how does expressiveness change with global tempo. Thus, the central question in this context is how the performance of a musical piece relates to the performance of the same piece at a different tempo. We describe *TempoExpress*, a case-based reasoning system for tempo transformation of musical performances, that preserves expressivity in the context of standard jazz themes. A preliminary version of the system was described in [16]. In this paper we present the completed system, and report the experimental results of our system over more than six thousand transformation problems.

Problem solving in case-based reasoning is achieved by identifying a problem (or a set of problems) most similar to the problem that is to be solved from a case base of previously solved problems (also called cases), and adapting the corresponding solution to construct the solution for the current problem. In the context of a music performance generation system, an intuitive manner of applying case-based reasoning would be to view unperformed music (e.g. a score) as a problem description (possibly together with requirements about how the music should be performed) and to regard a performance of the music as a solution to that problem. As we describe in the next section, in order to perform expressivity-aware tempo transformations, only representing the score is not enough for capturing the musical structure of a given melody. Moreover, because we are interested in changing the tempo of a specific performance, the expressive resources used in that performance have to be modeled as part of the problem requirements.

The paper is organized as follows: In section 2 we will present the overall architecture of *TempoExpress*. In section 3 we report the experimentation in which we evaluated the performance of *TempoExpress*. Section 4 points the reader to related work, and conclusions are presented in section 5.

2 System Architecture

In this section we will explain the structure of the *TempoExpress* system. A schematic view of the system as a whole is displayed in figure 1. For the audio analysis and synthesis, *TempoExpress* relies on two separate modules, that have been developed in parallel with *TempoExpress*, by Gomez et al. [14, 13]. The

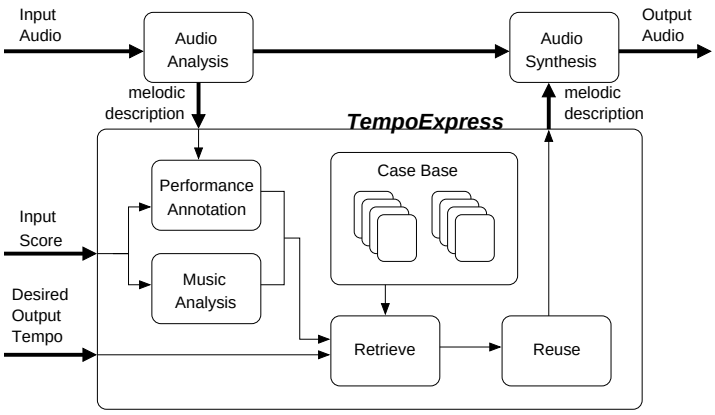

Figure 1: Schematic view of the *TempoExpress* system

analysis module is used to analyze monophonic audio recordings and provides a melodic description of the audio content, using an extension of the MPEG7 standard for multimedia content description [12]. *TempoExpress* takes such a melodic description as input data, together with a MIDI representation of the score that was performed in the audio. Finally, a desired output tempo is specified, the tempo at which the audio performance should be rendered. The melodic description of the performance and the MIDI file are used to automatically annotate the performance, yielding a representation of the expressivity of the performance. The MIDI score file is analyzed by a musical analysis component, that segments the phrase and returns a more abstract description of the melody. The performance annotation, together with the desired output tempo, the score and its analysis, form the problem description for which a solution is to be found.

Based on the problem description, the retrieval component finds a set of similar cases from the case base, and the reuse module composes the melody description for the output performance, based on the solutions from the retrieved cases. This is done in a segment by segment fashion. The motivation for this twofold. Firstly, using complete melodic phrases as the working unit for case based reuse is inconvenient, since a successful reuse will then require that the case base contains phrases that are nearly identical as a whole to the input phrase. Searching for similar phrase *segments* will increase the probability of finding a good match. On the other hand, segment wise retrieval and reuse is to be preferred over note by note retrieval and reuse, because the way a particular note is performed is highly unlikely to depend solely on the attributes of the note in isolation. Rather, the musical context of the note will play an important role.

In the reuse step, the input tempo performances of the retrieved segments are matched to the input performance of the problem and for the matching events, the output performance events are transferred to the context of the current

problem. When output performance events are found for all segments, the performance event sequences are concatenated to form the output performance. Finally the audio synthesis module renders the performance in audio, using the input audio and the revised melody description.

In the remaining subsections, we will give a more detailed explanation of the components of the system.

2.1 Automated Case Acquisition

An important issue for a successful problem solving system is the availability of example data. We have therefore put effort in automatizing the process of constructing cases from raw data. This process involves two main steps: performance annotation, and music analysis of the score. Performance annotation consists in matching of the elements of the performance to the elements in the score. This matching leads to the 'annotation' of the performance: a sequence of 'performance events'. The annotation can be regarded as a description of the musical behavior of the player while he interpreted the score, and as such conveys the musical expressivity of the performance. The second step in the case acquisition is an analysis of the musical score that was interpreted by the player. The principal goal of this analysis is to provide conceptualizations of the score at an intermediate level. That is, below the phrase level (the phrase is the musical unit which the system handles as input and output), but beyond the note level. One aspect is the segmentation of the score into motif level structures, and another one is the categorization of groups of notes that serves as a melodic context description for the notes.

2.1.1 Performance Annotation

It is common to define musical expressivity as the discrepancy between the musical piece as it is performed and as it is notated. This implies that a precise description of the performance alone is not very useful in itself. Rather, the relation between score and performance is crucial. The majority of research concerning musical expressivity is focused on the temporal, or dynamic variations of the notes of the musical score as they are performed [4, 5, 28, 37]. In this context, the spontaneous insertions or deletions of notes by the performer are often discarded as artifacts, or performance errors. This may be due to the fact that most of this research is focused on the performance practice of classical music, where the interpretation of notated music is rather strict. Contrastingly, in jazz music performers often favor a more liberal interpretation of the score, so that expressive variation is not limited to variations in timing of score notes, but also comes in the form of e.g. deliberately inserted and deleted notes. We believe that research concerning expressivity in jazz music should pay heed to these phenomena.

A consequence of this broader interpretation of expressivity is that the expressivity of a performance cannot be represented as a straight-forward list of expressive attributes for each note in the score. A more suitable representation

of expressivity describes the musical behavior of the performer as 'performance events'. The performance events form a sequence that maps the performance to the score. For example, the occurrence of a note that is present in the score, but has no counterpart in the performance, will be represented be a *deletion event* (since this note was effectively deleted in the process of performing the score). Obviously, deletion events are exceptions, and the majority of score notes are actually performed, be it with alterations in timing/dynamics. This gives rise to *correspondence events*, which establishes a correspondence relation between the score note and its performed counterpart. Once a correspondence is established between a score and a performance note, other expressive deviations like onset, duration, and dynamics changes, can be derived by calculating the differences of these attributes on a note-by-note basis.

Analyzing a corpus of monophonic saxophone performances of jazz standards (the recordings that were used to construct the case base), we encountered the following kinds of performance events:

Insertion Represents the occurrence of a performed note that is not in the score

Deletion Represents the non-occurrence of a score note in the performance

Consolidation Represents the agglomeration of multiple score notes into a single performed note

Fragmentation Represents the performance of a single score note as multiple notes

Transformation Represents the change of nominal note features like onset time, and duration

Ornamentation Represents the insertion of one or several short notes to anticipate another performed note

These performance events tend to occur persistently throughout different performances of the same phrase. Moreover, performances including such events sound perfectly natural, so much that it is sometimes hard to recognize them as deviating from the notated score. This supports our claim that even the more radical deviations that the performance events describe, are actually a common aspect of musical performance.

A key aspect of performance events is that they refer to particular elements in either the notated score, the performance, or both. Based on this characteristic, the events can be displayed as an ontology, as shown in figure 2. The primary classes of events are depicted as solid boxed names. The dotted boxed names represent secondary classes (a *Transformation* event belong can belong to any or all of the *PitchTransformation*, *DurationTransformation*, and *OnsetTransformation* classes, depending on the attribute values of the references score/performance elements). The unboxed names represent abstract classes.

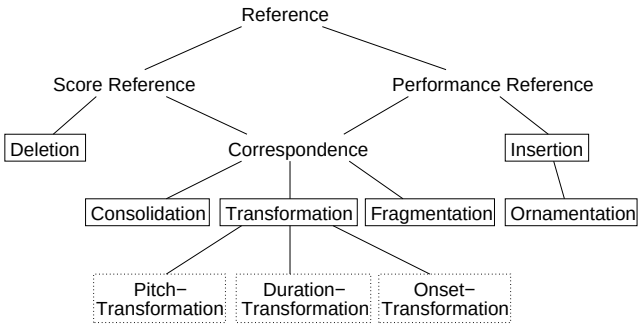

Figure 2: A hierarchical representation of performance events. The unboxed names denote abstract event classes; the solid boxed names denote the primary event classes, and the dotted boxed names denote secondary event classes

In order to obtain a sequence of performance events that represent the expressive behavior of the performer, the elements in the performance are matched to the elements in the score using the edit-distance, as described in a previous paper [1]. For every primary class of performance events, an edit operation is included in the edit distance. Secondary classes are not mapped to edit operations, since a performance element can belong to several secondary classes at the same time (i.e. a note can be changed in both onset and duration), whereas a performance element can only be matched to a single edit-operation. After assigning a cost to every edit-operation, the performance annotation is found by computing the sequence of edit-operations with minimal cost that accounts for all score and performance elements.

This method allows for automatically deriving a description of the expressivity in terms of performance events. With non-optimized edit operation costs, the average amount of annotation errors is about 13%, compared to manually corrected annotations. Typical errors are mistaking consolidations for a duration transformation event followed by note deletions, or recognizing a single ornamentation event , containing two or three short notes, as a sequence of note insertions (which they are, formally, it is supposedly more informative to represent these as an ornamentation). In previously presented work [15], we showed that by evolutionary optimization of edit operation costs using manually corrected annotations as training data, the amount of errors could be reduced to about 3%.

2.1.2 Musical Score Analysis

The second step in the case acquisition is an analysis of the musical score. This step actually consists of several types of analysis, used in different phases of the case based reasoning process.

Firstly, a metrical accents template is applied to the score, to obtain the level of metrical importance for each note. For example, the template for a

4/4 time signature, specifies that every first beat of the measure has highest metrical strength, followed by every third beat, followed by every second and fourth beat. The notes that do not fall on any of these beats, have lowest metrical strength. This information is used in the Implication/Realization analysis, described below, and in the retrieval/adaptation step of the CBR process (see subsections 2.3, and 2.4

Secondly, the musical score is segmented in to groups of notes, using the Melisma Grouper [33], an algorithm for grouping melodies into phrases or smaller units, like motifs. The algorithm uses rules regarding inter-onset intervals, and metrical strength of the notes, resembling Lerdahl and Jackendoff's *preference rules* [22]. The algorithm takes a preferred group size as a parameter, and segments the melody into groups whose size is as close as possible to the preferred size. In *TempoExpress*, the segmentation of melodic phrases into smaller units is done as part of the retrieval and reuse steps, in order to allow for retrieval and reuse of smaller units than complete phrases. We used a preferred group size of 5 notes, yielding on average 4.6 segments per phrase.

Lastly, the surface structure of the melodies is described in terms of the Implication/Realization model [24]. This model characterizes consecutive melodic intervals by the expectation they generate with respect to the continuation of the melody, and whether or not this expectation is fulfilled. The model states a number of data driven principles that govern the expectations. We have used the most important of these principles to implement an Implication/Realization parser for monophonic melodies. The output of this parser is a sequence of labeled melodic patterns, so called I/R structures. An I/R structure usually represents two intervals (three notes), although in some situations shorter or longer fragments may be spanned, depending on contextual factors like rhythm and meter. Eighteen basic I/R structures are defined using labels that signify the implicative/realizing nature of the melodic fragment described by they I/R structure. Apart from its label, the I/R structures are stored with additional attributes, such as the melodic direction of the pattern, the amount of overlap between consecutive I/R structures, and the amount of notes spanned.

The I/R analysis can be regarded as a moderately abstract representation of the score, that bears information about the rough pitch interval contour, and through the boundary locations of the I/R structures, includes metrical and durational information of the melody as well. As such, this representation is appropriate for comparison of melodies. As a preprocessing step to retrieval, we compare the score from input problem of the system to the scores in the case base, to weed out melodies that are very dissimilar.

2.2 Case Base Profile and Case Representation

For populating the case base, several saxophone performances were recorded from 4 jazz standards, each one consisting of 3–4 distinct phrases. The performances were played by a professional performer, at 9–14 different tempos per phrase. This resulted in 14 musical phrases, each with about 12 annotated performances (in total more than 4000 performed notes), as raw data for

constructing the case base.

After applying the case acquisition process described in section 2.1, we obtain performance annotations for each performance, and an I/R analysis for each phrase score. For each phrase, the score and I/R analysis are stored, together will all performance-annotations belonging to the performances of that phrase. Note that this aggregate of information is strictly speaking not a case, containing a problem and a solution, because it does not specify which tempo transformation is the problem, and which performance is the solution for that tempo transformation. Rather it holds the data from which many different tempo transformation cases can be constructed (precisely $n(n-1)$ for n performance-annotations). Hence it is more appropriate to call the aggregate of score, I/R analysis, and performance annotations a *proto case*. At the time the input problem becomes available to the system, the cases can be constructed from the proto cases, by taking the relevant performances annotations from the proto case.

2.3 Retrieval: Case Similarity Computation

The goal of the retrieval step is to form a pool of relevant cases, that can possibly be used in the reuse step. This done in the following three steps: firstly, cases that don't have performances at both the input tempo and output tempo are filtered out; secondly, those cases are retrieved from the case base that have phrases that are I/R-similar to the input phrase; lastly, the retrieved phrases are segmented. The three steps are described below.

2.3.1 Case filtering by tempo

In the first step, the case base is searched for cases that have performances both at the tempo the input performance was played, and the tempo that was specified in the problem description as the desired output tempo. The matching of tempos need not be exact, since we assume that there are no drastic changes in performance due to tempo within small tempo ranges. For example, a performance played at 127 beats per minute (BPM) may serve as an example case if we want to construct a performance at 125 BPM.

2.3.2 I/R based melody retrieval

In the second step, the cases selected in step 1 are assessed for melodic similarity to the score specified in the problem description. In this step, the primary goal is to rule out the cases that belong to different styles of music. For example, if the score in the problem description is a ballad, we want to avoid using a bebop theme as an example case.

We use the I/R analyses stored in the cases to compare melodies. The similarity computation between I/R analyses is based on the edit-distance, using edit-operation costs that were optimized using ground truth data for melodic similarity [36]. This algorithm for melodic similarity won the MIREX 2005

contest for symbolic melodic similarity [6, 17], which shows it performs relatively good compared to other state-of-the-art melody retrieval systems.

With this distance measure we rank the phrases available in the case base, and keep only those phrases with distances to the problem phrase below a threshold value. The cases containing the accepted phrases will be used as the precedent material for constructing the solution.

2.3.3 Segmentation

At the time of case acquisition, a segmentation of the melodic phrase into motifs is performed. In order to be able to work with the cases at a this level, the performance annotations must also be segmented. This is largely a straight-forward step, since the performance annotations contain references to the score. Only in the case non-score-reference events (such as ornamentations of insertions) occur at the boundary of two segments, it is not necessarily clear whether these events should belong to the former or the latter segment. In most cases however, it is a good choice to group these events with the latter segment (since for example ornamentation events always precede the ornamented note).

The set of segment level cases form a pool to be used in the reuse step.

2.4 Reuse: Transfer of Expressive Features

In the reuse step a performance of the input score is constructed at the desired tempo, based on the input performance and the set of retrieved phrase segments. This step is realized using constructive adaptation [26], a technique for reuse that constructs a solution by a best-first search through the space of partial solutions. This search process starts with a state that represents the input problem, without any part of the output performance that forms the solution to the problem. Then it starts to find construct alternative performances for a segment of the input melody, using different precedent segments from the segment pool. For every possible partial solution that was found, the next segment is provided with alternative solutions. By repeating this step, the space of partial solutions is searched until a complete solution is found. The state space is searched using best-first search, using the matching quality between the precedent segment and the problem segment as an heuristic.

Figure 3 shows an example of the reuse of a precedent segment for a particular input segment. We will briefly explain the numbered steps of this process one by one:

The *first step* is to find the segment in the pool of retrieved melodic segments that is most similar to the input score segment. The similarity is assessed by calculating the edit distance between the segments (the edit distance now operates on notes rather than on I/R structures, to have a finer grained similarity assessment).

In the *second step*, a mapping between the input score segment and the best matching retrieved segment is made, using the optimal path trace from the edit-distance calculations from the previous step.

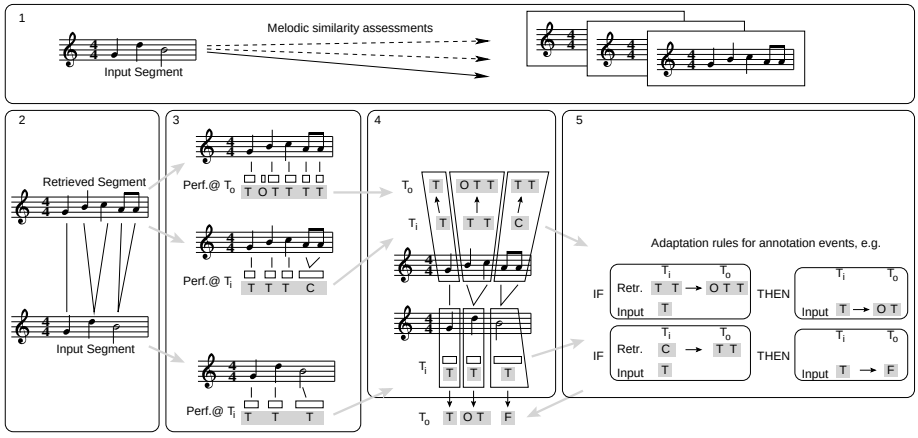

Figure 3: Example of case reuse for a melodic phrase segment. In step 1, a mapping is made between the input score segment and the most similar segment from the pool of retrieved segments. In step 2, the performance annotations for the tempos T_i and T_o are collected. In step 3, the performance annotation events are grouped according to the mapping between the input score and retrieved score. In step 4, the annotation events are processed through a set of rules to obtain the annotation events for a performance at tempo T_o of the input score segment

In the *third step*, the performance annotation events corresponding to the relevant tempos are extracted from the retrieved segment case and the input problem specification (both the input tempo T_i and the output tempo T_o for the retrieved segment case, and only T_i from the input problem specification).

The *fourth step* consists in relating the annotation events of the retrieved segment to the notes of the input segment, according to the mapping between the input segment and the retrieved segment, that was constructed in the first step. For the notes in the input segment that were mapped to one or more notes in the retrieved segment, we now obtain the tempo transformation from T_i to T_o that was realized for the corresponding notes in the retrieved segment. In case the mapping is not perfect and some notes of the input segment could not be matched to any notes of the retrieved segment, the retrieved segment cannot be used to obtain annotation events for the output performance. These gaps are filled up by directly transforming the annotation events of the input performance (at tempo T_i) to fit the output tempo T_o (by scaling the duration of the events to fit the tempo). That is, a uniform time stretching method is used as a default transformation, when the precedent provides no information. Note that such situations are avoided by the search method, because the proportion of un-matched notes negatively affects the heuristic value for that state.

In the *fifth step*, the annotation events for the performance of the input score at tempo T_o are generated. This is done in a note by note fashion, using rules

that specify which annotation events can be inferred for the output performance at T_o of the input score, based on annotation events of the input performance, and the annotation events of the retrieved performances (at T_i and T_o). To illustrate this, let us explain the inference of the Fragmentation event for the last note of the input score segment (B) in figure 3. This note was matched to the last two notes (A, A) of the retrieved segment. These two notes were played at tempo T_i as a single long note (denoted by the Consolidation event), and played separately at tempo T_o. The note of the input segment was also played as a single note at T_i (denoted by a Transformation event rather than a Consolidation event, since it corresponds to only one note in the score). To imitate the effect of the tempo transformation of the retrieved segment (one note at tempo T_i and two notes at tempo T_o), the note in the input segment is played as two shorter notes at tempo T_o, which is denoted by a Fragmentation event (F).

In this way, adaptation rules were defined, that describe how the tempo transformation of retrieved elements can be translated to the current case (in figure 3, two such rules are shown). In the situation an input-problem note and a precedent note have been matched, but there is no adaptation rule that matches their performance events, no output performance event can be found for that note. This happens when the input performance events for the problem segment and the precedent segment were too different. For example, a note might be loud in the input performance, and be matched to a note that was deleted in the precedent performance at the same tempo. In that case, we consider the interpretation of the precedent score too different from the interpretation of the input performance to serve as a good basis for transformation. The solution is again to use the input performance event for the output performance, resulting in a decreased heuristic value for the search state.

3 Experimental Results

In this section we describe experiments we have done in order to evaluate the *TempoExpress* system that was outlined above in comparison to straight forward tempo transformation, that is, uniform time stretching. By uniform time stretching we refer to scaling all events in the performance by a constant factor, namely the ratio between the input tempo and the output tempo. We have chosen to define the quality of the tempo transformation as the distance of the transformed performance to a target performance. The target performance is a performance played at the output tempo by a human player. This approach has the disadvantage that it may be overly restrictive, in the sense that measuring the distance to just one human performance discards different performances that may sound equally natural in terms of expressiveness. In another sense it may be not restrictive enough, depending on the choice of the distance metric that is used to compare performances. It is conceivable that certain small quantitative differences between performances are perceptually very significant, whereas other, larger, quantitative differences are hardly noticeable by the human ear.

To overcome this problem, we have chosen to model the distance measure used for comparing performances after human similarity judgments. A web based survey was set up, to gather information about human judgments of performance similarity. In the rest of this section we will explain how the performance distance measure was derived from the survey results, and give an overview of the comparison between *TempoExpress* and uniform time stretching.

3.1 Obtaining the Evaluation Metric

The distance measure for comparing expressive performances was modeled after human performance similarity judgments, in order to prevent the risk mentioned above, of measuring difference between performances that are not perceptually relevant (or conversely, failing to measure differences that *are* perceptually relevant).

3.1.1 Obtaining Ground Truth: a Web Survey on Perceived Performance Similarity

The human judgments were gathered using a web based survey. Subjects were presented with a target performance (the nominal performance, without expressive deviations) of a short musical fragment, and two different performances of the same fragment. The task was to indicate which of the two alternative performances was perceived as most similar to the target performance. The two alternative performances were varied in the expressive dimensions: fragmentation, consolidation, ornamentation, note onset, note duration, and note loudness. One category of questions tested proportionality of the effect quantity to perceived performance distance. Another category measured the relative influence of the type of effect (e.g. ornamentation vs. consolidation) on the perceived performance distance.

A total of 92 subjects responded to the survey, answering on average 8.12 questions (listeners were asked to answer 12 at least questions, but were allowed to interrupt the survey). From the total set of questions (66), those questions were selected that were answered by at least 10 subjects. This selection was again filtered to maintain only those questions for which there was significant agreement between the answers from different subjects (at least 70% of the answers should coincide). This yielded a set of 20 questions with answers, that is, triples of performances, together with dichotomous judgments, conveying which of the to alternative performances is closest to the target performance. The correct answer to a question was defined as the median of all answers for that question. This data formed ground truth for modeling a performance distance measure.

3.1.2 Modeling a Performance Distance Measure after the Ground Truth

An edit distance metric was chosen as the basis for modeling the ground truth, because the edit distance is flexible enough to accommodate for comparison of sequences of different length (in case of e.g. consolidation/fragmentation) and it allows for easy customization to a particular use by adjusting parameter values. Fitting the edit distance to the ground truth is a typical optimization problem, and as such, evolutionary optimization was used as a local search method to find good costs for the edit operations. The same approach of modeling an edit distance after ground truth by evolutionary optimization of edit operation costs, yielded particularly good results earlier, in a task of measuring melodic similarity [17].

The fitness function for evaluating parameter settings (encoded as chromosomes) was defined to be the proportion of questions for which the correct answer was predicted by the edit-distance, using the parameter settings in question. A correct answer is predicted when the computed distance between the target performance and the most similar of the to alternative performances (according to the ground truth) is lower than the computed distance between the target and the less similar alternative performance.

Using this fitness function a random population of parameter settings was evolved using an elitist method for selection. That is, the fittest portion of the population survives into the next population unaltered and is also used to breed the remaining part of the next population (by crossover and mutation) [11]. A fixed population size of 40 members was used. There were 10 parameters to be estimated. Several runs were performed and the maximal fitness tended to stabilize after 300 to 400 generations. Typically the percentages of correctly predicted questions by the best parameter setting found were between 70% and 85%. The best parameter setting found was used to define the edit distance allowing us to estimate the similarity between different performances of the same melody.

3.2 Comparison of *TempoExpress* and Uniform Time Stretching

In this subsection we report the evaluation results of the *TempoExpress* system on the task of tempo transformation, to the results of uniformly time stretching the performance. As said before, the evaluation criterion for the tempo transformations was the computed distance of the transformed performance to an original performance at the output tempo, using the edit-distance optimized to mimic human similarity judgments on performances.

A leave-one-out setup was used to evaluate the CBR system where, in turn, each phrase is removed from the case base, and all tempo transformations that can be derived from that phrase is performed using the reduced case base. The constraint that restricted the generation of tempo transformation problems from the phrases was that there must be an original human performance available at

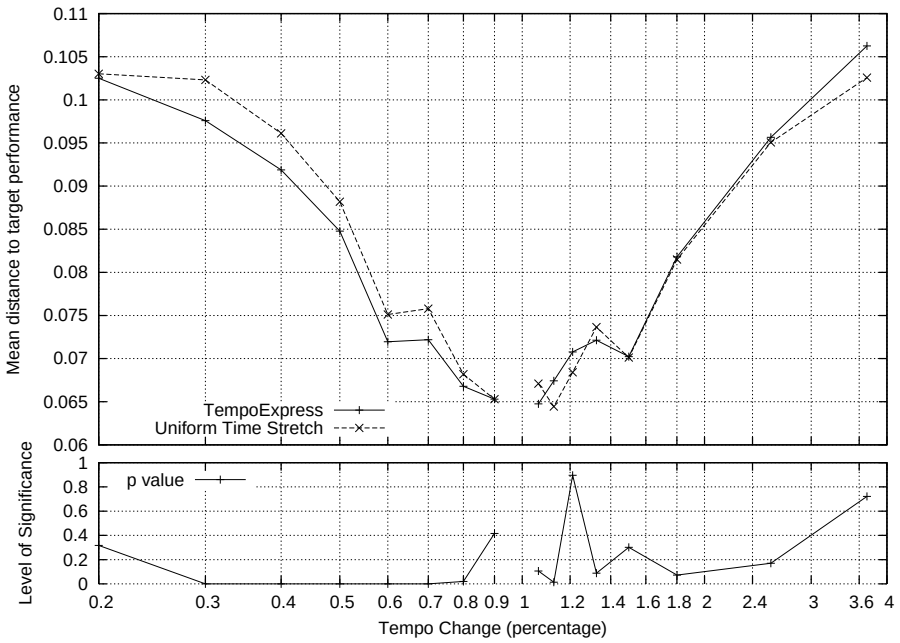

Figure 4: Performance of *TempoExpress* vs uniform time stretching as a function of tempo change (measured as the ratio between output tempo and input tempo). The lower plot shows the probability of incorrectly rejecting H_0 (non-directional) for the Wilcoxon signed-rank tests

the input tempo (the performance to be transformed) and another performance of the same fragment at the output tempo of the tempo transformation (this performance serves as the target performance to evaluate the transformation result). Hence the set of tempo transformation problems for a given phrase is the pairwise combination of all tempos for which a human performance was available. Note that the pairs are ordered, since a transformation from say 100 BPM to 120 BPM is not the same problem as the transformation from 120 BPM to 100 BPM. Furthermore the tempo transformations were performed on a phrase segment basis, rather than on complete phrases, since focusing on phrase level transformations is likely to involve more complex higher level aspects of performance (e.g. interactions between the performances of repeated motifs), that have not been seriously addressed yet. Moreover, measuring the performance of the system on segments will give a finer grained evaluation that measuring on the phrase level.

Defining the set of tempo transformations for segments yields a considerable amount of data. Each of the 14 phrases in the case base consists of 3 to 6 motif-like segments, identified using Temperley's Melisma Grouper [33], and has approximately 11 performances at different tempos (see subsection 2.2). In total there are 64 segments, and 6364 transformation problems were generated

using all pairwise combinations of performances for each segment. For each transformation problem, the performance at the input tempo was transformed to a performance at the output tempo by *TempoExpress*, as well as by uniform time stretching (UTS). Both of the resulting performances were compared to the human performance at the output tempo by computing the edit-distances. This resulted in a pair scores for every problem. Figure 4 shows the average distance to the target performance for both *TempoExpress*and UTS, as a function of the amount of tempo change (measured in as the ratio between output tempo and input tempo). Note that lower distance values imply better results. The lower graph in the figure shows the probability of incorrectly rejecting the null hypothesis (H_0) that the mean of *TempoExpress* distance values is equal to the mean of UTS distance values, for particular amounts of tempo change. The significance was calculated using a non-directional Wilcoxon signed-rank test [18].

Firstly, observe that the plot in Figure 4 shows an increasing distance to the target performance with increasing tempo change (both for slowing down and for speeding up), for both types of transformations. This is evidence against the hypothesis of *relational invariance* [27], since this hypothesis implies that the UTS curve would be horizontal, since under relational variance, tempo transformations are supposed to be achieved through mere uniform time stretching.

Secondly, a remarkable effect can be observed in the behavior of *TempoExpress* with respect to UTS, which is that *TempoExpress* seems to improve the result of tempo transformation only when slowing performances down. When speeding up, the distance to the target performance stays around the same level as with UTS. In the case of slowing down, the improvement with respect to UTS is mostly significant, as can be observed from the lower part of the plot.

Finally, note that the p-values are rather high for tempo change ratios close to 1, meaning that for those tempo changes, the difference between *TempoExpress* and UTS is not significant. This is in accordance with the common sense that slight tempo changes do not require many changes, in other words, relational invariance approximately holds when the amount of tempo change is very small.

Another way of visualizing the system performance is by looking at the results as a function of absolute tempo change (that is, the difference between input and output tempo in beats per minute), as shown in figure 5. The overall forms of the absolute curves and the relative curves (figure 4) are quite similar. Both show that the improvements of *TempoExpress* are mainly manifest on tempo decrease problems.

Table 1 summarizes the results for both tempo increase and decrease. Columns 2 and 3 show the average distance to the target performance for *TempoExpress* and UTS, averaged over all tempo increase problems, and tempo decrease problems respectively. The other columns show data from the Wilcoxon signed-rank test. The p-values are the probability of incorrectly rejecting H_0 (that there is no difference between the *TempoExpress* and UTS results). This table also shows that for downward tempo transformations, the improvement of *TempoExpress* over UTS is small, but extremely significant ($p < .001$), whereas for

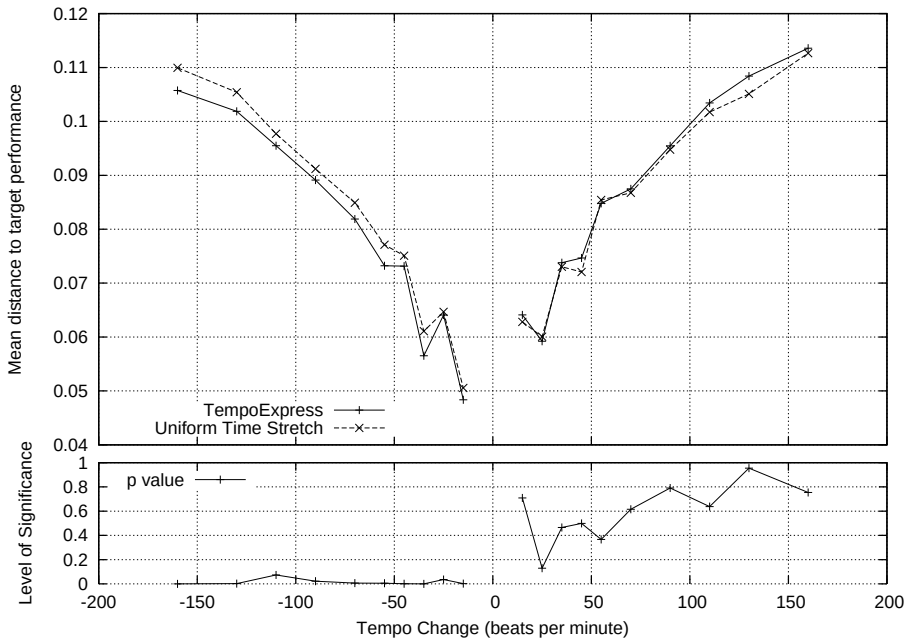

Figure 5: Performance of *TempoExpress* vs UTS as a function of tempo change (measured in beats per minute). The lower plot shows the probability of incorrectly rejecting H_0 (non-directional) for the Wilcoxon signed-rank tests

upward tempo transformations UTS seems to be better, but the results are slightly less decisive ($p < .05$).

How can the different results for tempo increase and tempo decrease be explained? A practical reason can be found in the characteristics of the case base. Since the range of tempos at which the performances were played varies per song, it can occur that only one song is represented in some tempo range. For example in our case base, there is one song with performance in the range from 90 BPM to 270 BPM, whereas the highest tempo at which performances of other songs are available is 220 BPM. That means that in the leave-one-out method, there are no precedents for tempo transformations to tempos in the range from 220 BPM to 270 BPM. This may explain the increasing gap in performance in favor of UTS, towards the end of the spectrum of upward tempo transformations.

4 Related Work

The field of expressive music research comprises a rich and heterogeneous number of studies. Some studies are aimed at verbalizing knowledge of musical experts on expressive music performance. For example, Friberg et al. are work-

	mean distance to target		Wilcoxon signed-rank test		
	TempoExpress	UTS	p <>	z	*df*
tempo increase	0.0791	0.0785	0.046	1.992	3181
tempo decrease	0.0760	0.0786	0.000	9.628	3181

Table 1: Overall comparison between *TempoExpress* and uniform time stretching, for upwards and downwards tempo transformations, respectively

ing on *Director Musices* (DM), a system that allows for automatic expressive rendering of MIDI scores [8]. DM uses a set of expressive performance rules that have been formulated with the help of a musical expert using an analysis-by-synthesis approach [30, 7, 31].

Widmer [37] has used machine learning techniques like Bayesian classifiers, decision trees, and nearest neighbor methods, to induce expressive performance rules from a large set of classical piano recordings. In another study by Widmer [38], the focus was on discovery of simple/robust performance principles rather than obtaining a model for performance generation.

In the work of Desain and Honing and co-workers, the focus is on the validation of cognitive models for music perception and musical expressivity. They have pointed expressivity has an intrinsically perceptual aspect, in the sense that one can only talk about expressivity when the performance itself defines the standard (e.g. a rhythm) from which the listener is able to perceive the expressive deviations [19]. In more recent work, Honing showed that listeners were able to identify the original version from a performances and a uniformly time stretched version of the performance, based on timing aspects of the music [20].

Timmers et al. have proposed a model for the timing of grace notes, that predicts how the duration of certain types of grace notes behaves under tempo change, and how their durations relate to the duration of the surrounding notes [34].

A precedent of the use of a case-based reasoning system for generating expressive music performances is the *SaxEX* system [3, 2]. The goal of the *SaxEX* system is to generate expressive melody performances from an inexpressive performance, allowing user control over the nature of the expressivity, in terms of expressive labels like 'tender', 'aggressive', 'sad', and 'joyful'.

Another case-based reasoning system is *Kagurame* [32]. This system renders expressive performances of MIDI scores, given performance conditions that specified the desired characteristics of the performance.

Recently, Tobudic and Widmer [35] have proposed a case-based approach to expressive phrasing, that predicts local tempo and dynamics and showed it outperformed a straight-forward k-NN approach.

To our knowledge, all of the performance rendering systems mentioned above deal with predicting expressive values like timing and dynamics for the notes in the score. Contrastingly *TempoExpress* not only predicts values for timing and dynamics, but also deals with note insertions, deletions, consolidations, fragmentations, and ornamentations.

5 Conclusion and Future Work

In this paper we presented our research results on global tempo transformations of music performances. We are interested in the problem of how a performance played at a particular tempo can be rendered automatically at another tempo preserving some of the features of the original tempo and at the same time sounding natural in the new tempo. We focused our study in the context of standard jazz themes and, specifically on saxophone jazz recordings.

We proposed a case-based reasoning approach for dealing with tempo transformations and presented the *TempoExpress* system. *TempoExpress* has a rich description of the musical expressivity of the performances, that includes not only timing and dynamics deviations of performed score notes, but also represents more rigorous kinds of expressivity such as note ornamentation, and note consolidation/fragmentation. We apply edit distance techniques in the retrieval step, as a means to assess similarities between the cases and the input problem. In the reuse step we employ constructive adaptation. Constructive adaptation is a technique able to generate a solution to a problem by searching the space of partial solutions for a complete solution that satisfies the solution requirements of the problem.

Moreover, we described the results of our experimentation over a case-base of more than six thousand transformation problems. *TempoExpress* clearly behaves better than a Uniform Time Stretch (UTS) when the target problem is slower than the input tempo. When the target tempo is higher than the input tempo the improvement is not significant. Nevertheless, *TempoExpress* behaves at a similar level than UTS except in transformations to really fast tempos. This result is not surprising because of the lack of cases with tempos higher than 220 BPM. Summarizing the experimental results, for downward tempo transformations, the improvement of TempoExpress over UTS is small, but extremely significant ($p < .001$), whereas for upward tempo transformations UTS seems to be better, but the results are slightly less decisive ($p < .05$).

As a future work, we wish to extend the experiments to analyze the performance of *TempoExpress* with respect to the complete phrases. This experimentation requires of acquiring several recordings of melodies at the same tempo and on defining comparison measures at the phrase level.

Acknowledgments

This research has been partially supported by the Spanish Ministry of Science and Technology under the project TIC 2003-07776-C2-02 "CBR-ProMusic: Content-based Music Processing using CBR" and EU-FEDER funds.

References

[1] J. Ll. Arcos, M. Grachten, and R. López de Mántaras. Extracting performer's behaviors to annotate cases in a CBR system for musical tempo

transformations. In Kevin D. Ashley and Derek G. Bridge, editors, *Proceedings of the Fifth International Conference on Case-Based Reasoning (ICCBR-03)*, number 2689 in Lecture Notes in Artificial Intelligence, pages 20–34. Springer-Verlag, 2003.

[2] Josep Lluís Arcos and Ramon López de Mántaras. An interactive case-based reasoning approach for generating expressive music. *Applied Intelligence*, 14(1):115–129, 2001.

[3] Josep Lluís Arcos, Ramon López de Mántaras, and Xavier Serra. Saxex : a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27 (3):194–210, 1998.

[4] Sergio Canazza, Giovanni De Poli, Stefano Rinaldin, and Alvise Vidolin. Sonological analysis of clarinet expressivity. In Marc Leman, editor, *Music, Gestalt, and Computing: studies in cognitive and systematic musicology*, number 1317 in Lecture Notes in Artificial Intelligence, pages 431–440. Springer, 1997.

[5] P. Desain and H. Honing. Does expressive timing in music performance scale proportionally with tempo? *Psychological Research*, 56:285–292, 1994.

[6] J.S. Downie, K. West, A. Ehmann, and E. Vincent. The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview. In *Prooceedings of the 6th International Conference on Music Information Retrieval*, 2005.

[7] A. Friberg. Generative rules for music performance: A formal description of a rule system. *Computer Music Journal*, 15 (2):56–71, 1991.

[8] A. Friberg, V. Colombo, L. Frydén, and J. Sundberg. Generating musical performances with Director Musices. *Computer Music Journal*, 24(1):23–29, 2000.

[9] A. Gabrielsson. Once again: The theme from Mozart's piano sonata in A major (K. 331). A comparison of five performances. In A. Gabrielsson, editor, *Action and perception in rhythm and music*, pages 81–103. Royal Swedish Academy of Music, Stockholm, 1987.

[10] A. Gabrielsson. Expressive intention and performance. In R. Steinberg, editor, *Music and the Mind Machine*, pages 35–47. Springer-Verlag, Berlin, 1995.

[11] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[12] E. Gómez, F. Gouyon, P. Herrera, and X. Amatriain. Using and enhancing the current mpeg-7 standard for a music content processing tool. In *Proceedings of Audio Engineering Society, 114th Convention*, Amsterdam, The Netherlands, 2003.

[13] E. Gómez, M. Grachten, X. Amatriain, and J. Ll. Arcos. Melodic characterization of monophonic recordings for expressive tempo transformations. In *Proceedings of Stockholm Music Acoustics Conference 2003*, 2003.

[14] E. Gómez, A. Klapuri, and B. Meudic. Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1), 2003.

[15] M. Grachten, J. Ll. Arcos, and R. López de Mántaras. Evolutionary optimization of music performance annotation. In U.K. Wiil, editor, *Computer Music Modeling and Retrieval*, Lecture Notes in Computer Science. Springer, 2004.

[16] M. Grachten, J. Ll. Arcos, and R. López de Mántaras. TempoExpress, a CBR Approach to Musical Tempo Transformations. In *Advances in Case-Based Reasoning. Proceedings of the 7th European Conference, ECCBR 2004*, Lecture Notes in Computer Science. Springer, 2004.

[17] M. Grachten, J. Ll. Arcos, and R. López de Mántaras. Melody retrieval using the Implication/Realization model. In *Online Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*. Queen Mary University of London, 2005. `http://www.ismir.net/all-papers.html`.

[18] R. Hogg and E. Tanis. *Probability and Statistical Inference*. Macmillan, New York, 3rd edition, 1998.

[19] H. Honing. Structure and interpretation of rhythm and timing. *Tijdschrift voor Muziektheorie*, 7(3):227–232, 2002.

[20] H. Honing. Timing is tempo-specific. In *Proceedings of the International Computer Music Conference*, pages 359–362, Barcelona, 2005. Pompeu Fabra University.

[21] P.N. Juslin. Communicating emotion in music performance: a review and a theoretical framework. In P.N. Juslin and J.A. Sloboda, editors, *Music and emotion: theory and research*, pages 309–337. Oxford University Press, New York, 2001.

[22] F. Lerdahl and R. Jackendoff. An overview of hierarchical structure in music. In S. M. Schwanaver and D. A. Levitt, editors, *Machine Models of Music*, pages 289–312. The MIT Press, 1993. Reproduced from Music Perception.

[23] E. Lindström. 5 x "oh, my darling clementine". the influence of expressive intention on music performance, 1992. Department of Psychology, Uppsala University.

[24] E. Narmour. *The Analysis and cognition of basic melodic structures : the implication-realization model*. University of Chicago Press, 1990.

[25] C. Palmer. Anatomy of a performance: Sources of musical expression. *Music Perception*, 13(3):433–453, 1996.

[26] E. Plaza and J. Ll. Arcos. Constructive adaptation. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning*, number 2416 in Lecture Notes in Artificial Intelligence, pages 306–320. Springer-Verlag, 2002.

[27] B. H. Repp. Relational invariance of expressive microstructure across global tempo changes in music performance: An exploratory study. *Psychological Research*, 56:285–292, 1994.

[28] B. H. Repp. Quantitative effects of global tempo on expressive timing in music performance: Some perceptual evidence. *Music Perception*, 13(1):39–58, 1995.

[29] J. A. Sloboda. The communication of musical metre in piano performance. *Quarterly Journal of Experimental Psychology*, 35A:377–396, 1983.

[30] J. Sundberg, Anders Friberg, and Lars Frydén. Common secrets of musicians and listeners: an analysis-by-synthesis study of musical performance. In P. Howell, R. West, and I. Cross, editors, *Representing Musical Structure*, Cognitive Science series, chapter 5. Academic Press Ltd., 1991.

[31] J. Sundberg, Anders Friberg, and Lars Frydén. Threshold and preference quantities of rules for music performance. *Music Perception*, 9:71–92, 1991.

[32] T. Suzuki. The second phase development of case based performance rendering system "Kagurame". In *Working Notes of the IJCAI-03 Rencon Workshop*, pages 23–31, 2003.

[33] D. Temperley. *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, Mass., 2001.

[34] R. Timmers, R.and Ashley, P. Desain, H. Honing, and L. Windsor. Timing of ornaments in the theme of beethoven's paisiello variations: Empirical data and a model. *Music Perception*, 20(1):3–33, 2002.

[35] A. Tobudic and G. Widmer. Case-based relational learning of expressive phrasing in classical music. In *Proceedings of the 7th European Conference on Case-based Reasoning (ECCBR'04)*, Madrid, 2004.

[36] R. Typke, M. den Hoed, J. de Nooijer, F. Wiering, and R.C. Veltkamp. A ground truth for half a million musical incipits. In *Proceedings of the Fifth Dutch-Belgian Information Retrieval Workshop*, pages 63–70, 2005.

[37] G. Widmer. Large-scale induction of expressive performance rules: First quantitative results. In *Proceedings of the International Computer Music Conference (ICMC2000)*, San Francisco, CA, 2000. International Computer Music Association.

[38] G. Widmer. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):37–50, 2002.