DMRN+13: DIGITAL MUSIC RESEARCH NETWORK
ONE-DAY WORKSHOP 2018

QUEEN MARY UNIVERSITY OF LONDON
TUE 18 DECEMBER 2018

Evaluating musical score difference: a two-level comparison

Francesco Foscarin[1], Raphael Fournier-S'Niehotta[1] , Philippe Rigaux[1] and Florent Jaquemard[3]

[1]Vertigo, CNAM Paris, France, francesco.foscarin@cnam.fr
[3]INRIA, Paris, France

***Abstract*— The comparison of music scores is a crucial step for many tasks, such as collaborative score creation, version control systems and evaluation of music transcription. We propose two measures of score difference aiming at comparing the scores at semantic and syntactic level.**

I. INTRODUCTION

The Unix *diff* utility is a line-based comparison of two text files, useful in particular for source code files with few differences.

As recently observed in [1] it's not meaningful to apply the Unix *diff* utility to XML scores because the line structure of a XML file does not reflect the its musical structure. They propose a hierarchic *diff* algorithm for XML MEI files based on a tree edit distance applied to xml trees. We follow here a different approach, proposing a comparison on two different objectives. The first level is semantic: we compare the musical content of the score (seen as a sequence of events characterized by their pitches and durations), with techniques similar to melodic similarity [2][3]. The second level is syntactic, *i.e.* the graphical content of the score (beamings, bar positions, dots, ties, etc.). For both cases, we compute a measure of the difference and a list of modifications necessary to transform one score into the other.

II. SEMANTIC DIFFERENCE

For this case we use an intermediate lossy representation of the scores called timeline (similar to MIDI files) that contains the explicit note pitches and temporal positions.
Considering first the monophonic case, we build a sequence of couples (pitch, duration) from each timeline. We then use an edit distance with the usual operations (insert, delete and update) [4] to compute a similarity index and to retrieve the list of transformations of one score to the other (Fig.1).

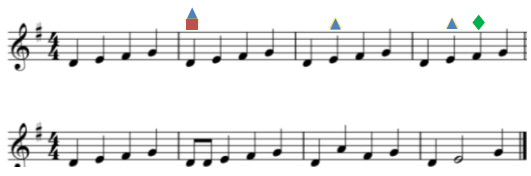

Figure 1. Example of the differences computation and visualization on two monophonic scores. The edit operations are notated by different figures: square (insert), triangle (update), rhombus (delete).

This difference is feature-based: we can compute it for pitch only, for durations only or for both features, and we can have different cost/weight values for the edit operations, depending on the needings.
We may generalize to a polyphonic score, computing a similarity index between the voices of the two scores and consider the pairs that give the maximum similarity.

III. SYNTACTIC DIFFERENCE

For the syntactic difference we model the graphical representation of the score with notation trees [5], building the trees from each voice in each measure. Considering first the monophonic case, the computation of the differences reduces to the problem of finding the longest common subsequences (LCSs)[4] for the sequences of tree hash of the two scores. This is similar to compute an edit distance with only insertion and deletion operations.

The generalization to a polyphonic score is done similarly to the semantic difference.

IV. CONCLUSIONS

We propose a tool that allow the computation of two kinds of differences between two scores. It is partially implemented in the digital music library NEUMA (http://neuma.huma-num.fr/) to be easy to access and use. Our future goals are to upgrade the visualization tool and to perform a finer tree distance computation for the notation trees, that will allow to compute and visualize a fine grain comparison of the measures (*e.g.*, beaming breaks or continuations and different tuplet groupings).

REFERENCES

[1] Antila, C., & Weaver, G. (2017). A hierarchic diff algorithm for collaborative music document editing. In *Third International Conference on Technologies for Music Notation and Representation (TENOR)*.
[2] Lemstrom, K. (2000). String matching techniques for music retrieval. PhD thesis, University of Helsinky.
[3] Mongeau, M., & Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, *24*(3), 161-175.
[4] Hunt, J. W., & MacIlroy, M. D. (1976). *An algorithm for differential file comparison*. Murray Hill: Bell Laboratories.
[5] Foscarin, F., Fournier-S'Niehotta, R., Jacquemard, F., & Rigaux, P. (2018). Évaluation de la correction rythmique des partitions numérisées. In *Journées d'Informatique Musicale (JIM 2018)*.

*Research supported by ANR France-Quebec MUNIR Project.