Extraction and evaluation of temporal musical features from MIDI recordings of organ music

Dominik Kisić[1 2], Sebastià V. Amengual[2], Aristotelis Hadjakos[2], Malte Kob[2]

[1] Fakultet Elekktrotehnike i Računarstva, 10000 Zagreb, Croatia, Email: dominik.kisic@fer.hr

[2] Hochschule für Musik Detmold, 32756 Detmold, Deutschland, Email: {amengual, kob, hadjakos}@hfm-detmold.de,

Introduction

The influence of room acoustics on musical live performance is a topic under current research and it is based on recording live performances under different acoustic conditions. These recordings are then analyzed in order to extract a set of musical features that explain the adjustments implemented by the musicians in their performance. The majority of instruments require analysis of audio recordings, but in some specific cases, especially regarding keyboard instruments such as piano or organ, it is possible to record a MIDI stream which contains information related to the actions of different keys and pedals on the instrument. The use of MIDI signals for the analysis of music performance decreases the complexity of the analysis in comparison to audio recordings, since note events such as onsets, offsets, pitch and key velocity are coded and can be easily analyzed. This approach has been previously implemented in the study of piano performance [1, 2].

Since the dynamics of the organ are mostly constant (pipe valves are either fully open or closed), the performance analysis of organ playing is limited to temporal features. Existing tools e.g. MIDI toolbox for Matlab [3] - provide a good framework for importing, visualizing and analyzing MIDI data. However, the analysis and comparison of organ performance requires a detailed analysis of different performance features e.g. note duration, breaks duration, tempo fluctuations... - which are not available at the moment.

This paper presents a set of Matlab functions that can be used to separate melodies and analyze temporal features of MIDI recordings of polyphonic key instruments and constitutes a complementary work in the study of performance of organ players presented previously in [4, 5].

Pre-existing functions

The MIDI Toolbox for Matlab [3] is used as a starting point for the import and visualization of MIDI streams. This section presents an overview of these operations.

Notematrix or *nmat* is an alternative representation format of MIDI data for its use in Matlab provided by the MIDI Toolbox (see Fig. 2 for more information). Columns one and two contain the temporal information of the notes - onset time and duration, respectively - in beats. The last two columns (sixth and seventh) contain the same information in seconds. In our particular case, since the recorded performances contain tempo variations which are not coded in a MIDI stream, only the information in seconds is usable.Column four contains the MIDI note corresponding to the pitch. MIDI channel information is given in the third row. In organ recordings every manual (and pedal) is usually coded into a different channels. Column five represents the velocity of a played note, which contains always the maximum value in our organ recordings. That feature could be used for the analysis of dynamics of piano performances or performances of any other instrument, but is excluded from the temporal analysis presented in this paper.

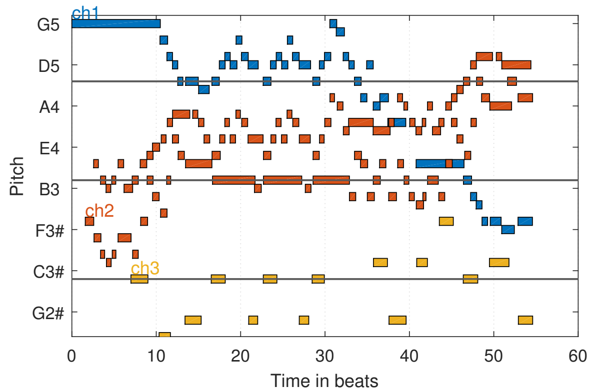

Figure 1: Piano roll

Onset (beats)	Duration (beats)	MIDI channel	MIDI pitch	Velocity	Onset (sec)	Duration (sec)
0	1.2250	7.00	48.00	127.000	0	0.6125
0.0333	0.6354	1.00	63.00	127.00	0.0167	0.3177
0.0490	0.9073	1.00	72.00	127.00	0.0245	0.4536
0.0781	2.0750	1.00	67.00	127.00	0.0391	1.0375
0.1063	1.9458	1.00	55.00	127.00	0.0531	0.9729
1.0240	1.1542	7.00	50.00	127.00	0.5120	0.5771
1.0281	1.0958	1.00	65.00	127.00	0.5141	0.5479
1.0427	1.1396	1.00	71.00	127.00	0.5214	0.5698
3.1719	1.0948	7.00	48.00	127.00	1.5859	0.5474
3.2031	0.4490	1.00	63.00	127.00	1.6016	0.2245
3.2469	1.7719	1.00	55.00	127.00	1.6234	0.8859
3.2490	0.8198	1.00	72.00	127.00	1.6245	0.4099

Figure 2: NMAT format

Pianoroll is a function that gives a graphical interpretation of the *nmat* format (see Fig. 1). Vertical axis can be chosen as a number or a name of the pitch of notes. Horizontal axis can be in beats or in seconds. Different colors represent different channels of the MIDI recording.

Extracted features

In preliminary studies [4, 5] a general view on temporal analysis was implemented, giving only a few features: *overall tempo*, *average duration of notes (Dur_{notes})* and *average duration of breaks (Dur_{breaks})*.

The *overall tempo (Tempo)* feature contains the overall tempo of the whole recording and it is calculated as presented in [6],

$$Tempo = 60 \cdot \frac{B_{beats}}{T_{time}} \tag{1}$$

where B is the number of beats and T is the duration in seconds of the recording, from the first to the last onset.

Equations 2 and 3 describe calculation of the *duration of notes* and *duration of breaks*. They give information about the average duration of all notes and breaks,

$$Dur_{notes} = \sum \frac{T_{notes}}{N_{notes}} \tag{2}$$

$$Dur_{breaks} = \sum \frac{T_{breaks}}{N_{breaks}} \tag{3}$$

where T_{notes} and T_{breaks} are arrays containing the duration of every note and break, respectively. N_{notes} and N_{breaks} are the total number of notes and breaks, respectively.

In order to study more detailed features it is necessary to dispose of some previous information regarding the analyzed piece. This means that it is necessary to use a theoretical performance synthesized from a MIDI score representation in order to obtain the theoretical duration and timing of the notes and compare it then with the recorded performance. Implemented features that benefit from the use of a MIDI score are the *inter-note tempo*, *inter-bar tempo*, *cumulative tempo* and *note overlap*.

The *inter-note tempo ($Tempo_{notes}$)* is an array containing the tempo of every individual note and is defined as

$$Tempo_{notes}(i) = 60 \cdot \frac{b_{on}(i+1) - b_{on}(i)}{t_{on}(i+1) - t_{on}(i)} \tag{4}$$

where b_{on} and t_{on} are the onset time of note i in the score (in beats) and in the recording (in seconds), respectively.

The *inter-bar tempo ($tempo_{bar}$)* is an array that expresses the tempo of every bar.

$$Tempo_{bar}(i) = 60 \cdot \frac{N_{measure}}{tbar_{on}(i+1) - tbar_{on}(i)} \tag{5}$$

where $N_{measure}$ is a positive number indicating the number of beats per bar and $tbar_{on}$ is the onset time of a bar.

Cumulative tempo ($Tempo_{cumulative}$) constitutes an array of average tempo for the first n notes, where the number of notes used for the calculation increases with every index of the array (see equation 6). This feature shows how a certain part of the melody affects the *overall tempo*, and with every new note the average tempo accumulates. The last value of the array would be an average of the entire analyzed melody, being the same as *overall tempo*.

```
Tempo =

          loc_breaks: [0x1 double]
             loc_bar: [6x2 double]
     loc_notes_beats: [34x1 double]
           dur_notes: 0.5740
          dur_breaks: NaN
             overlap: [33x1 double]
         overlap_pos: [23x1 double]
         overlap_neg: [10x1 double]
     overlap_Avg_pos: 0.0237
     overlap_Avg_neg: -0.0815
         tempo_score: 72
        tempo_melody: 76.9239
           tempo_bar: [6x1 double]
         tempo_notes: [33x1 double]
    tempo_cumulative: [33x1 double]
```

Figure 3: Output structure of tempo MIDI analysis function

$$Tempo_{cumulative}(i) = 60 \cdot \frac{b_{on}(i) - b_{on}(1)}{t_{on}(i) - t_{on}(1)} \tag{6}$$

where b_{on} and t_{on} are the onset time of note i in the score (in beats) and in the recording (in seconds), respectively.

Note Overlap (overlap) is calculated as the offset time of a note minus the onset time of the next note. The position of breaks in the score are excluded from the calculation of *overlap* because the goal of this feature is to show articulation changes in the melody. In order to have a more concise descriptor, averages of positive and negative overlaps are calculated. Also, positive and negative values in the overlap array are separated.

$$Overlap(i) = t_{off}(i) - t_{on}(i+1) \tag{7}$$

where t_{off} and t_{on} are the offset and onset times of a note in seconds, respectively.

Implementation

The presented features are implemented in a set of Matlab functions that return the results of the analysis in a structure as shown in Fig. 3. Main features are given in the fields named *tempo_notes, tempo_cumulative, tempo_bar, overlap*. Besides the extracted features, a set of complementary are given in order to simplify the plotting tasks of the results. *Loc_bar* and *loc_breaks* are arrays of indexes of the notes whose onset corresponds with the start of a bar and those notes which have a break after them, respectively. The second column of the *loc_bar* matrix contains a number that represents the duration of each bar. If two bars are coupled with a legato note over

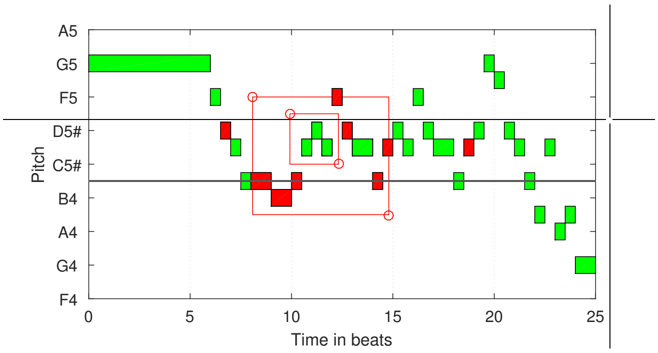

Figure 4: Melody extraction function interface

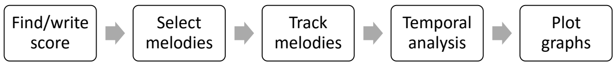

Figure 5: Workflow for the extraction of performance features

the bar the exact beginning of the second bar is unknown. These two bars are then treated as one double bar in further analysis and plotting functions. *Loc_notes_beats* contains the onset positions of each note in beats. *Dur_notes* and *dur_breaks* are the average duration of notes and breaks. The results structure format allows an easy integration of different analyzed performances, making it suitable for the study of large collections of recordings.

As a side effect of this analysis, two additional functions were created so that the entire work flow would work smoother. These two functions are named *nmatTrackMelody* and *nmatExtractMelody*. They are tools for extracting single melodies that can then be used along with the tempo analysis function. The functions were successfully tested on the data-set obtained during the experiments completed in [4, 5]. However, since the purpose of these functions is very specific, their accuracy can not be guaranteed with different data-sets.

NmatTrackMelody is a simple melody tracking function aiming at the automation of the analysis of multiple performances, allowing the user to separate a specific melody from a polyphonic performance. The input of the function is a complete recording of an analyzed performance and a single (monophonic) melody from the score. It is based on a straightforward algorithm which maps the monophonic score with the polyphonic performance using note positions and time information. Yet simple, the algorithm is still capable of disregarding small errors in the performance and robust against tempo variability, allowing overlapping between notes. The primary assumption that was taken into consideration was that the recorded performance does not have many wrong notes.

NmatExtractMelody gives the user an option of graphically extracting a portion of a melody from the entire MIDI recording or the score. It is based on *pianoroll* and *ginput* functions. In a Matlab figure (see figure 4), the options are to click on a specific note that is then selected, changing the color from green to red, or to use a different mode to click in two corners of the box, selecting all the notes inside the box. By using the left mouse

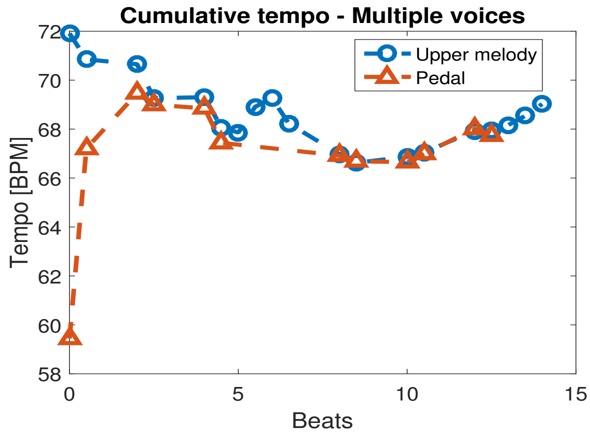

Figure 6: Cumulative tempo of a single performance for two different voices

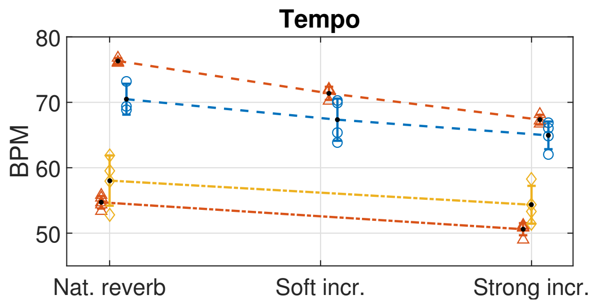

Figure 7: Example of a graph showing tempo changes with different players for different reverberation levels

button, the notes are selected, and by using the right mouse button, the notes can be unselected. The output of the function gives two arrays, the first one being the selected part of the piece and the second one being the rest of the piece that was not selected.

Worflow & Examples

A diagram of the work flow of the toolbox use can be seen in Fig. 5. Firstly, a musical piece for the analysis has to be chosen and a MIDI score of that piece needs to be available. The second step is to use the melody extraction function *NmatExtractMelody* to separate different melodies in the score depending on the melodies that are needed or wanted to be analyzed. The chosen melody is then tracked in a performance using *NmatTrackMelody*. After having an extracted recorded melody and its corresponding score, a MIDI tempo analysis function is used which gives all the analyzed data. The function for the temporal analysis of MIDI data has three inputs: an *nmat* object containing the extracted MIDI melody of the performance with a corresponding MIDI score and information about the measure of the MIDI score (number of beats in a bar).

Plotting the analyzed parameters varies depending on the user's specific demands. As research is still ongoing, only a few examples of plotting options are presented in this paper. Figure 6 presents *cumulative tempo* for two different voices of the same performance, where every marker represents a single note of the melody, showing

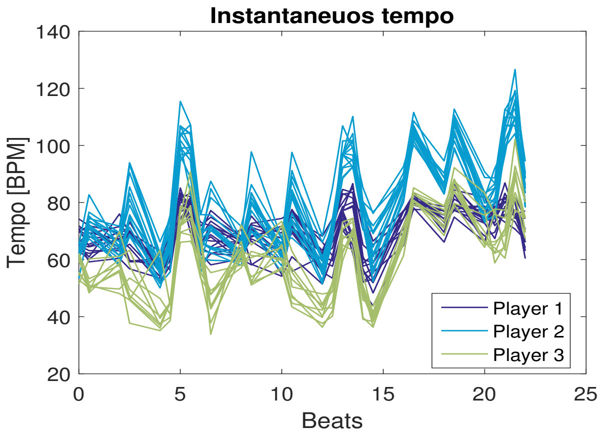

Figure 8: Inter note tempo of the performance for different players

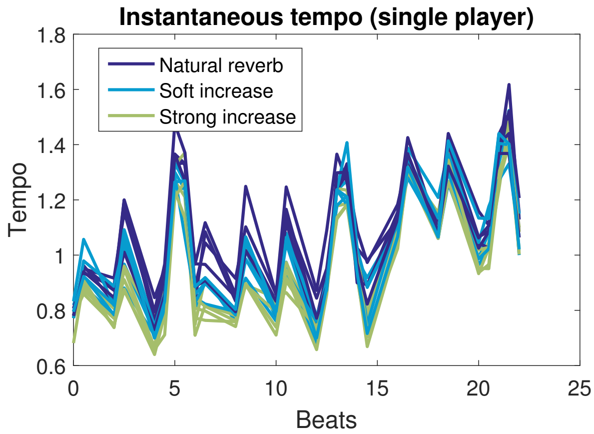

Figure 9: Inter-note tempo of a single player with different acoustic conditions

how a melody with faster rhythm is more affected by tempo changes. Figure 7 shows the variation of the overall tempo of the performance of different players depending on different levels of reverberation. Different colors stand for different players, different line types represent different sessions, and every marker represents a take of the performance. Figures 8 and 9 show different representations of inter-note tempo. On both graphs different plotted lines represent individual performances. Figure 8 is a graph of inter-note tempo variations for different players, where one color represents each player, while in a figure 9 the colors represent different acoustical conditions for the same player. Two different formats of presenting tempo can be seen in the graphs, in *beats per minute* or in normalized tempo, where all tempos are scaled using the overall tempo average.

Conclusion

In this paper a Matlab tool for MIDI analysis of polyphonic performance is presented including examples of their usage. The tool will be used to analyze an extensive data-set generated during previous experiments with organ players.

One of the current issues with the use of the tool is related with the requirement of having a MIDI score for every performance. This could be solved by having a database or a software solution to convert music scores to MIDI.

Possible upgrades of the tool include the implementation of other tempo related features such as *rubato* analysis or statistical tools such as correlation analysis. For improving the melody tracking function, a more robust melody tracking algorithm [7] could be implemented. Finally, a graphical user interface (GUI) including plotting is also being considered for implementation to facilitate the use of the presented tools in other research projects.

Acknowledgments

The authors wish to thank W. Lachenmayr for providing data from recordings, and L. Beslać for proofreading. The research work presented in this article has been funded by the European Commission within the ITN Marie Curie Action project BATWOMAN under the 7th Framework Programme (EC grant agreement no. 605867).

References

[1] Bolzinger, S.; Warusfel, O.; Kahle, E.: “A study on the influence of room acoustics on piano performance”, *Journal de Physique IV Colloque C5, supplement au Journal de Physique III*, Vol.4 C5-617-620 (1994)

[2] Kawai, K.; Kato, K.; Uneo, K.; Sakuma, T.: ”Experiment on adjustment of piano performance to room acoustics: Analysis of performance coded into MIDI data”, *Proceedings of International Symposium on Room Acoustics 2013*, Toronto, Canada, 2013.

[3] Eerola, T.; Toiviainen, P.: ”MIDI Toolbox: MATLAB Tools for Music Research”, *University of Jyväskylä*, Jyväskylä, Finland, 2004.

[4] Amengual Garí, S. V.; Lachenmayr, W.; Kob, M.: ”Study on the influence of acoustics on organ playing using room enhancement”, *In Proceedings of the Third Vienna Talk on Music Acoustics*, Vienna, Austria, 2015.

[5] Amengual Garí, S.; Lachenmayr, W.; Kisić, D.; Kob, M.: ”Preliminary results of the effect of reverberation and sound delay on organ playing”, *Proceedings of FAMA des DEGA*, Hamburg, 2015.

[6] Lerch, A.: ”Softwere-Based extraction of objective parameters from music performances”, *PhD dissertation*, Berlin, Germany, 2008.

[7] Bloch, J.; Dannenberg, R.: ”Real-time computer accompaniment of keyboard performances”, *Proceedings of ICMC*, Cambridge, Massachusetts, 1985.