Audio Engineering Society

Convention Paper

Presented at the 138th Convention
2015 May 7–10 Warsaw, Poland

A real-time system for measuring sound goodness in instrumental sounds

Oriol Romani Picas[1], Hector Parra Rodriguez[1], Dara Dabiri[1], Hiroshi Tokuda[2], Wataru Hariya[2], Koji Oishi[2] and Xavier Serra[1]

[1] *Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain*

[2] *Technology Development Dept., KORG Inc., Tokyo, Japan*

Correspondence should be addressed to Oriol Romani Picas (oriol.romani@upf.edu)

ABSTRACT

This paper presents a system that complements the functionality of a typical tuner by evaluating the sound quality of a music performer in real-time. It consists of a software tool that computes a score of how well single notes are played with respect to a collection of reference sounds. To develop such a tool we first record a collection of single notes played by professional performers. Then, the collection is annotated by music teachers in terms of the performance quality of each individual sample. From the recorded samples, several audio features are extracted and a machine learning method is used to find the features that best described performance quality according to musician's annotations. An evaluation is carried out to assess the correlation between systems predictions and musicians criteria. Results show that the system can reasonably predict musicians annotations of performance quality.

1. INTRODUCTION

Although a formal definition of musical performance quality is complicated due to its subjectivity [1], in music teaching there is a certain consensus on what is a good and a bad sound for a particular instrument. Bearing this in mind, Knight et al. [2] worked on automatic classification of sound characteristics in terms of quality using a machine learning approach. This study uses a similar approach but aims to grade isolated musical notes according to their performance quality or sound goodness. Together with a group of music teachers, five sound attributes that affect the goodness of a note are defined. These attributes are *dynamic stability*, *pitch stability*, *timbre stability*, *timbre richness* and *attack clarity*.

This work is carried out by first creating a training dataset of single note recordings including six classes of sounds per studied instrument (i.e., clarinet, flute and trumpet). Five of these classes include examples of note recordings which are intentionally badly played according to the five aforementioned sound attributes (e.g., sounds with bad dynamic stability, pitch stability, etc.). The sixth class includes examples of note recordings which are considered to be well played. A subset of the dataset is manually labeled in terms of the relevant dynamic segments of each note: attack, decay, sustain and release. From this subset, an automatic learning process is performed to get the parameters to segment the notes automatically. Then, several audio features are extracted from the different segments of the notes to train a classifier that predicts a goodness score for each of the sound attributes. This features are extracted using Essentia [3], an open-source library for audio analysis and audio-based music information retrieval. An overall sound goodness score is also computed as the average of the former predicted scores. Finally, the automatic segmentation and the classifier are implemented in a real-time application. As each musical instrument has its sound particularities, a different model is built for each one. In this work, we describe and evaluate the models we build for clarinet, flute and trumpet instruments.

2. METHODS

2.1. Data Collection

The recordings of the training dataset were done using two different microphones: Neumann U87 Ai and the iPhone 5 microphone plugged into a Yamaha DM 2000 VCM console. The microphone positions were the same for the whole recording session of an instrument. Recordings were done at a 48 kHz sample rate and 24 bit depth. Instrument players are professional musicians with a music degree and music teaching background. For each instrument, two musicians recorded isolated notes for all the semitones contained in the primary range of the instrument and the six sound classes previously defined.
These ranges are:

- Clarinet: D3 - G6
- Flute: C4 - C7
- Trumpet: D3 - E6

For the good sound class, musicians were asked to play in a mezzoforte dynamic level, while for the other classes no dynamic restriction was imposed. Recordings were done in two different time periods. One was done at the beginning of the research and the other after a first preliminary evaluation of the prototype. For the second set of recordings, the classes with worst classification results in our preliminary evaluation were prioritized. A summary of the number of recorded notes for each sound class is presented in Table1. The dataset was finally stored in Freesound [4].

	Clarinet	Flute	Trumpet
Good sound	258	248	206
Bad dynamic stability	503	369	284
Bad pitch stability	268	94	203
Bad timbre stability	206	112	133
Bad timbre richness	346	74	51
Bad attack clarity	722	199	307

Table 1: Number of recordings in the training dataset per instrument and per sound class.

2.2. Note segmentation

The first processing step of our system includes the segmentation of notes and identification of their sustained part in order to evaluate its stability independently of its attack or release. For that purpose we automatically detect the following note boundaries:

- *onset*: the perceptual beginning of the note.
- *beggining of sustain*: the beginning of the sustained part of the note.
- *release*: the end of the sustained part of the note.
- *offset*: the perceptual end of the note.

The following features were developed to detect each boundary:

- *onset amplitude threshold*: calculated as the loudness envelope value at which the note starts.
- *offset amplitude threshold*: the amplitude envelope value at the end of the note.

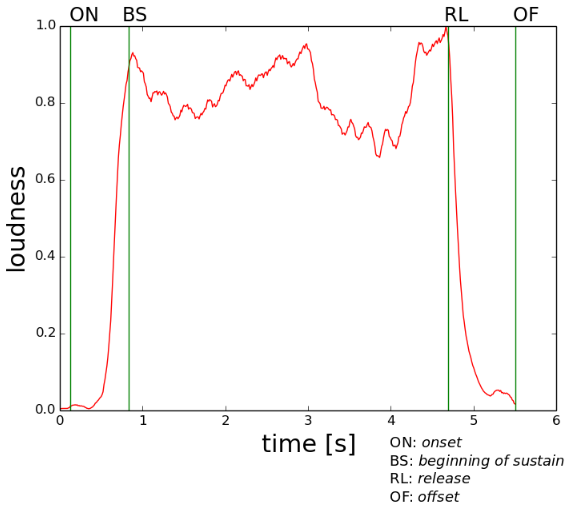


Fig. 1: Automatic segmentation of a single note.

- *attack slope ratio*: computed using the Adaptative threshold method (ATM)[5]
- *release slope ratio*: computed also with ATM but starting from the offset and going backwards.

The values of the previous segmentation features are obtained using a grid search that minimizes the average error of all parameters. In order to do this, a subset of sounds is manually segmented to create a ground truth. Then, the error is measured as the distance (in time units) from detected boundaries to the ground truth. This process is repeated for every instrument in order to adapt the segmentation to different envelope shapes. An example of the final automatic segmentation is shown in Fig.1.

2.3. Feature selection

Using Essentia, different audio descriptors are selected and used for each one of the sound attributes, e.g. YinFFT [6] algorithm for pitch stability or Tristimulus [5] for timbre stability. They are computed at different segments of the sounds depending on the analyzed sound attribute. For example, descriptors for stability attributes are computed in the sustained part of the sound while the descriptors for the attack clarity are computed only in the attack part. After this analysis, several statistical measures are computed for each descriptor to represent its behavior. These measures over the descriptors are the final analysis features. Almost 200 features are used for each sound attribute. Then, one single feature is used to determine the goodness of a single sound attribute. This feature is selected using a machine learning process approach: OneR [7] classifier using the Weka [8] data mining software. This feature is the one that best discriminates between the good sound class and a bad attribute class. A summary of the selected features and their classification accuracies is shown in Table 2 and a description of them is given in Section 6.

	Clarinet	Flute	Trumpet
Dynamic stability	env_flatness (94.61%)	env_std_norm (80,72%)	env_std_norm (82,89%)
Pitch stability	pitch_regenv _std (98,43%)	pitch_regenv _max (86,95%)	pitch_regenv _std (100%)
Timbre stability	tristimulus_log1 _std (87,90%)	tristimulus1_std (89,27%)	kurtosis_std (81,88%)
Timbre richness	flatness_mean (87,80%)	richness_std (89,47%)	skewness_mean (88,39%)
Attack clarity	pitch_stderr_ median (77,64%)	attack_time (80,09%)	pitch_tukey _std (76,63%)

Table 2: Descriptors from Essentia chosen for each sound attribute and for each instrument with their classification accuracy.

2.4. Score computation

The overall goodness score is computed as the average of the five sound attribute scores. Each sound attribute score is achieved using a function that links the value of a feature to the score. We consider the value of the feature for which the classificaton is performed as the medium score. This is due to the fact that this value is the one that discriminates between a good and a bad sound class. We then take into account the distribution of the feature value for the whole dataset to build a function that maps feature values to score. An example of one of this functions is shown in Fig.2.

3. SYSTEM OVERVIEW

The system components are a learning process and a real-time part. (Fig.3). In the first component, the data collection and the segmentation parameters are used to find the features that best describe each sound attribute. Using this, a real-time component is implemented, which is able segment an input

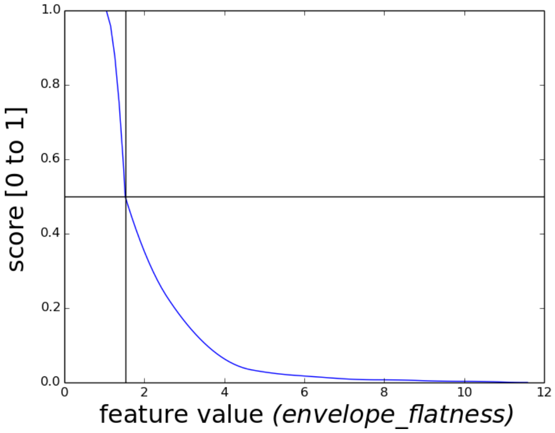


Fig. 2: Mapping function from feature value to score

note into the segmentation boundaries according to the parameters learnt in previous stages. Then the selected features found in the previous component are computed for each sound attribute. Finally, the mapping function is used to compute the score of each attribute and the final goodness score. A prototype interface of this real-time component is shown in Fig.4.

4. EVALUATION

The goal of the evaluation is to assess the correlation between the output of the system and the musicians' criterion. Since a perceptual concept such as timbre stability is subjective, grading such a concept could lead to different results for different users. We assume that there is a consensus on what is a good sound so the grades of several users should show clear trends. Two different processes are performed: a *user grading test* and a *real-time experiment.*

4.1. User grading test

The six musicians that recorded the sounds of the training dataset are also asked to grade 100 sounds according to their criteria. The set is randomly selected from the database in order to have examples of the different sound classes and no extra information is given to the users. In this case, only the goodness score (or overall score) of the note is considered. The results shown in Table 3 are the error of the machine score respect to users grades.

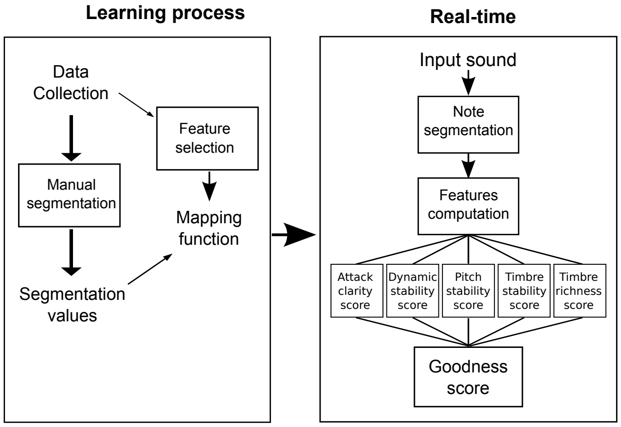
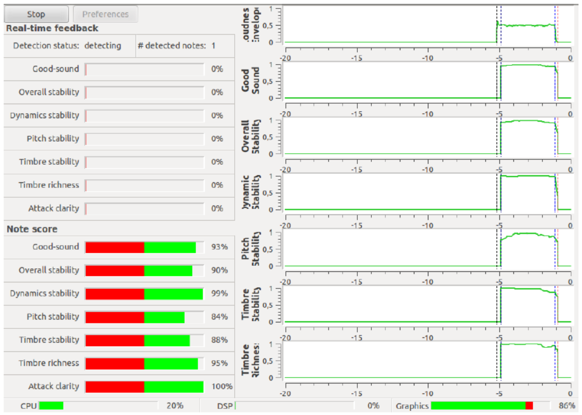


Fig. 3: System diagram.



Real-time feedback

Detection status: detecting # detected notes: 1

Good-sound 0%
Overall stability 0%
Dynamics stability 0%
Pitch stability 0%
Timbre stability 0%
Timbre richness 0%
Attack clarity 0%

Note score

Good-sound 93%
Overall stability 90%
Dynamics stability 99%
Pitch stability 84%
Timbre stability 88%
Timbre richness 95%
Attack clarity 100%

CPU 20% DSP 0% Graphics 86%

Fig. 4: Interface of the prototype.

4.2. Real-time experiment

We designed an experiment to test the correlation between the scores given by the system and the musicians criteria in a real-time scenario. Two musicians for each one of the models (clarinet, flute and trumpet) are asked to play single notes of each one of the sound classes while using the prototype. For each example they are asked to grade each sound attribute of the sound class. The results shown in Table 4 are again computed as the difference between the machine score and the musicians grade.

The real-time experiment shows that the system goodness score fits the musicians criterion. If we consider the sound attributes separately, the error

	Flute	Clarinet	Trumpet
Mean absolute error	13,30%	13,83%	18,28%
Root mean squared error	16,94%	17,50%	21,80%

Table 3: Estimation error of the system

	Flute	Clarinet	Trumpet	TOTAL
Goodness	0,00%	3,25 %	0,00%	1,08 %
Dynamic stability	8,00%	17,50 %	4,75 %	10,08 %
Pitch stability	20,00%	7,00%	12,5 %	13,25 %
Timbre stability	14,75 %	2,5 %	7,5 %	8,25 %
Timbre richness	28,25 %	18,75 %	0,00%	15,67 %
Attack clarity	26,25 %	17,00%	20,00%	21,08 %
TOTAL	16,25 %	11,00%	7,46 %	

Table 4: Estimation error of the real-time prototype

tends to be below 2 points, a magnitude of error that could also occur between different musicians grading the same sound. The users were also asked to rate the usefulness of the application. The average score is 8.7, which suggests that the system is potentially helpful for music teaching.

5. CONCLUSION

The work carried out in this research has been used as the basis of Artistry© (patent pending), a technology developed by Korg and the Music Technolgy Group that is used in the mobile app Cortosia© by Korg Inc.

For future work we want to improve the system to give scores even more correlated to the musicians criteria. We think the effectiveness of the system is linked to the amount of sounds contained in the dataset and the metadata of the users opinion on the sounds. Thus we want to implement a methodology to automatically add sounds and metadata to the database, for instance through a social community of musicians sharing and commenting sounds. The system will be extended to work with new instruments and we are already working in the models for cello and violin. We want also to extend the system to deal not only with isolated notes but with any possible musical exercise.

6. APPENDIX

In this section we decribe the features that were obtained in the automatic learning process using the data collection described previously:

env_flatness: flatness coefficient of the signal envelope. [3]
pitch_regenv_std: is computed as the standard deviation of the difference between the pitch in the sustained part of the note and a linear regression of this pitch.
tristimulus_log1_std: standard deviation of the first tristimulus [5] of the logarithmic spectrum magnitude.
flatness_mean: mean of the spectrum flatness.
pitch_stderr_median: standard deviation of the pitch median.
env_std_norm: normalized standard deviation of the envelope.
pitch_regenv_max: the maximum of the difference between the pitch and its linear regression.
tristimulus: standard deviation of the first tristimulus.
richness_std: the relevance of harmonics respect to the total, an opposite concept to the tristimulus.
attack_time: the duration of the attack.
kurtosis_std: the standard deviation of the kurtosis[5].
skewness_mean: the mean value of the skewness[5].
pitch_tukey_std: the standard deviation of the pitch windowed by a tapered cosine window [9].

7. REFERENCES

[1] J. Geringer and C. Madsen. (1998) "Musicians ratings of good versus bad vocal and string performances", Journal of Research in Music Education, vol. 46, 1998, p. 522-34.

[2] T. Knight, F. Upham, and I. Fujinaga. (2011). "The Potential for Automatic Assessment of Trumpet Tone Quality" in Anssi Klapuri & Colby Leider, ed. ISMIR, University of Miami, pp. 573-578.

[3] D. Bogdanov, N. Wack, E. Gmez, S. Gulati, P. Herrera, O. Mayor, et al. (2013). "ESSENTIA: an Audio Analysis Library for Music Information Retrieval" International Society for Music Information Retrieval Conference (ISMIR'13). 493-498.

[4] F. Font, G. Roma, and X. Serra. (2013) "Freesound technical demo." Proceedings of the 21st ACM international conference on Multimedia. ACM, 2013.

[5] G. Peeters (2004) "A large set of audio features for sound description (similarity and classification) in the CUIDADO project", Proceedings of the 12th International Society for Music Information Retrieval Conference. Miami (Florida), USA. October 24-28. pp. 573–578.

[6] P. M. Brossier (2007) "Automatic Annotation of Musical Audio for Interactive Applications, Ph.D. Thesis, Queen Mary University of London: UK, 2007.

[7] R. C. Holte (1993). "Very simple classification rules perform well on most commonly used datasets" Machine Learning. 11:63-91.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P.Reutemann, I. H. Witten (2009) The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[9] J. W. Tukey(1967). "An introduction to the calculations of numerical spectrum analysis". Spectral Analysis of Time Series.(B. Harris, ed.) 25-46. Wiley, New York.