

Timbra: An online tool for feature extraction, comparative analysis and visualization of timbre

Filip Szymański, Ewa Łukasik, and Magdalena Chudy

Abstract—DARIAH.lab is a research infrastructure created for digital humanities, consisting of state-of-the-art hardware and dedicated software tools. One of the tools developed for digital musicology is Timbra, a web-based application for conducting research on sound timbre. The aim was to create an easy-to-use online tool for non-programmers. The tool can be used to calculate, visualise, and compare different timbre characteristics of uploaded audio files and to export the extracted parameters in CSV format for further processing, e.g. by classification tools. The application offers extraction and visualisation of scalar features such as zero crossing rate, fundamental frequency, spectral centroid, spectral roll-off, spectral flatness, band energy ratio, as well as feature vectors (e.g. chromagram, spectral contrast, spectrogram, and MFCCs). An interested user can compare selected sound characteristics using various types of plots and run dissimilarity analysis of timbre parameters by means of 2D or 3D multidimensional scaling (MDS). The paper showcases potential applications of the tool based on presented case studies. In terms of implementation, the calculations are performed at the backend Django server using Librosa and standard Python libraries. Dash library is used for the frontend. By offering an easy-to-use tool accessible anytime and anywhere through the Internet, we want to facilitate timbre analysis for a broader group of researchers, e.g. sound engineers, luthiers, phoneticians, or musicologists.

Keywords—timbre; audio descriptors; feature extraction; comparative analysis; visualization

I. INTRODUCTION

DARIAH.LAB [1] is a research infrastructure for digital humanities that has been built by the DARIAH-PL consortium. It is a network of distributed research laboratories equipped with state-of-the-art hardware and dedicated software tools and integrated with digital data resources from various fields of humanities and art sciences.

The tools for digital musicology were developed as part of the Automatic Enrichment Laboratory and based on the existing open libraries including libraries from Music Information Retrieval (MIR) field. In the case of Timbra, or Timbre Analyzer, the aim was to create an easy-to-use online tool for non-programmers that can extract, visualise, and compare different timbre characteristics of uploaded audio files. It was built with an educational purpose in mind to help familiarise with a range of audio descriptors identified by MIR field as

F. Szymański and E. Łukasik are with Poznan University of Technology, Poznan, Poland (e-mail: filip.szymanski@cs.put.poznan.pl, ewa.lukasik@cs.put.poznan.pl).

M. Chudy is with Institute of Art, Polish Academy of Sciences, Warsaw, Poland (e-mail: magdalena.chudy@ispan.pl).

important acoustical characteristics for describing the timbre of sound. By focusing on feature visualisation and comparison, Timbra was designed to support a better understanding of calculated parameters and experimenting with different feature subsets, and as a result, to facilitate timbre analysis for a broader group of researchers, e.g. sound engineers, luthiers, phoneticians, or musicologists.

II. BACKGROUND

Research on musical timbre has been an integral part of the MIR field and its beginnings can be traced back to Helmholtz's pioneering work published in the 19th century [2]. The current definition by ANSI [3] describes timbre as a multidimensional attribute of auditory sensation which enables distinguishing between two sounds having the same loudness, pitch, spatial location, and duration. It also acknowledges that “the timbre of a sound is strongly influenced by its time-varying characteristics, particularly during the initial portion (attack), and is also influenced by its ongoing spectral and temporal characteristics.”

To capture those spectral and temporal characteristics of timbre, a variety of acoustic descriptors were developed, mainly through perceptual studies facilitated by the advancement in computers and audio signal processors. For example, perceived *brightness* of timbre was confirmed by a large body of psychoacoustic research to be one of the main dimensions of timbre spaces [4]–[10]. It has been often correlated with acoustic features such as spectral centroid and spectral roll-off that are indicators of high frequency energy content in the audio spectrum. As an acoustical correlate of proposed timbre spaces, spectral centroid was proved to be a good discriminator between timbres of different musical instruments [11], [12] as well as between timbres of the same instrument class [13]–[15].

The timbre of a musical instrument can be described using “Sines + Noise” model, consisting of a deterministic component, harmonics, represented by sinusoids, and a stochastic component, noise. Spectral flatness, or tonality descriptor, shows how noisy is the signal. The spectral flatness is calculated as the ratio of the geometric mean to the arithmetic mean of a power spectrum. Flatness value close to 1 indicates the spectrum's noise-like character, i.e. the spectrum is flat, and flatness close to 0 depicts the spiky character of the power spectrum, characteristic of tonal sounds.



Cepstral coefficients are indicators of the shape of the spectral envelope. In speech and speaker recognition tasks, as well as in Music Information Retrieval, mel frequency cepstral coefficients (MFCC) are commonly used. They represent the envelope of the power spectrum in the Mel scale.

Based on the findings from the perceptual studies, the standardised definitions of timbre descriptors were incorporated into MPEG-7 as part of the audio data representation framework (ISO/IEC-15938-4, 2002). A comprehensive review of the standard with the descriptors' definitions and Matlab scripts can be found in [16]. Similarly, various sets of temporal, spectral and spectro-temporal parameters have been implemented in a form of practical toolboxes released for the wider research community. They include Matlab-based frameworks such as Timbre Toolbox [17] and Miningsuite [18] (former MIRtoolbox [19]), standalone frameworks and libraries written in C/C++ (Essentia [20], Marsyas [21], Aubio [22], LibXtract [23], YAAFE [24]), Java-based jAudio [25], Python API Librosa [26], and Meyda [27] based on Web Audio API. More detailed overview and the results of tools evaluation with the Cranfield model were described in [28].

From the end user perspective, to be able to deploy any of these toolboxes requires at least to know Matlab scripting or to have basic programming skills in scripting languages such as Python for example. Also, these tools focus on audio feature extraction rather than on feature visualization since plotting data graphs is considered part of a research technical skillset. For these reasons, Timbra was developed to support less technically advanced users in timbre characteristics exploration.

III. TIMBRA

A. Architecture

The Timbra software consists of two main components: backend and frontend applications (Fig. 1). The backend application was implemented using the Python framework Django [29], for a number of reasons. First, Django is a proven and stable tool used in many companies and global organizations. Second, it is a scalable framework, i.e. it can be used to create smaller as well as much larger and more extensive websites. Third, it supports handling a large number of users and enables future development of the application. Fourth, Django has a large and active developer community, so there are many libraries and extensions available.

Apart from Django, other key component libraries include:

- i) Librosa: a library for analyzing and processing audio files. It is optimized in terms of performance (processing large data sets) and offers many functions that calculate parameters such as chromagram, spectral centroid, cepstral coefficients, etc. Librosa supports popular audio file formats including WAV, MP3, OGG, FLAC.

- ii) NumPy: a library for working with lists. It was created in 2005 and its *ndarray* lists are up to 50 times faster than the standard lists.

- iii) SciPy: a library for data analysis and scientific computing based on NumPy.

- iv) Scikit-learn: a library containing tools for building and evaluating machine learning models.

- v) Other libraries: Pandas, Uuid, Pickle, JSON, Math, OS.

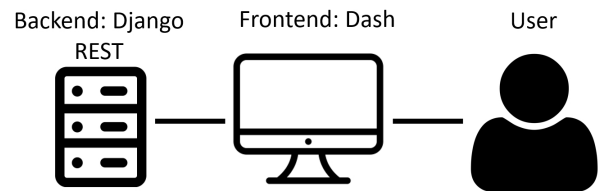


Fig. 1. Timbra. The general architecture diagram.

In the backend, all calculations on sound files are performed using objects belonging to one of two classes:

AudioSample (AS). When the user uploads an audio file, an instance of this class is created, which stores the imported audio file as one of the attributes, along with details such as sampling frequency, format, duration and file name. The class has methods that calculate individual timbre parameters and attributes that store them for further use. It has also setters and getters used for example in the method that packs attributes into CSV files.

AudioExperiment (AE). Objects of this class are created as an aggregation of objects of the *AudioSample* class. These aggregators enable to calculate the parameters of all *AudioSample* objects and compare them with each other. In addition, this class is used to compute the multidimensional scaling of selected parameters, taking either their average values or the values per individual audio frame.

The frontend application, i.e. the user interface and its communication with the Django server, was implemented using the Python framework Dash [30]. It uses the Requests module to send requests to the server, sending data in JSON format. Requests are based on HTTP methods: GET, POST, and DELETE. The reason behind deploying the Dash framework was its integration with the Python library Plotly [31] used to create charts, and the simplicity with which they can be embedded in Dash applications. Another advantage of Dash is that it supports a modular approach to creating applications. A software project can be divided into smaller components, which makes it easier to develop and manage. Dash also has an active programming community, thanks to which it was possible to use extensions such as Dash Bootstrap Components, Dash Mantine Components, and Holoviews.

Plotly is used to create interactive charts and data visualizations that enable the user to zoom in, move and analyze the data on the chart, as well as to add elements such as sliders to modify the data on-the-fly. Available chart types include linear, bar and scatter plots, histograms, and heat maps. They can be customized by changing their colors, style, labels and other elements, and exported, for example, to PNG format. Plotly also enables the generation of 3D charts for analysis of spatial data or data at subsequent points in time.

B. User interface

The user interface consists of three main parts: right-side toolbar, left-side toolbar and the main workspace for user-generated charts (Fig. 2). The left-side toolbar functionalities include creating, selecting and managing objects of the *AudioSample* and *AudioExperiment* classes. At the top, there is

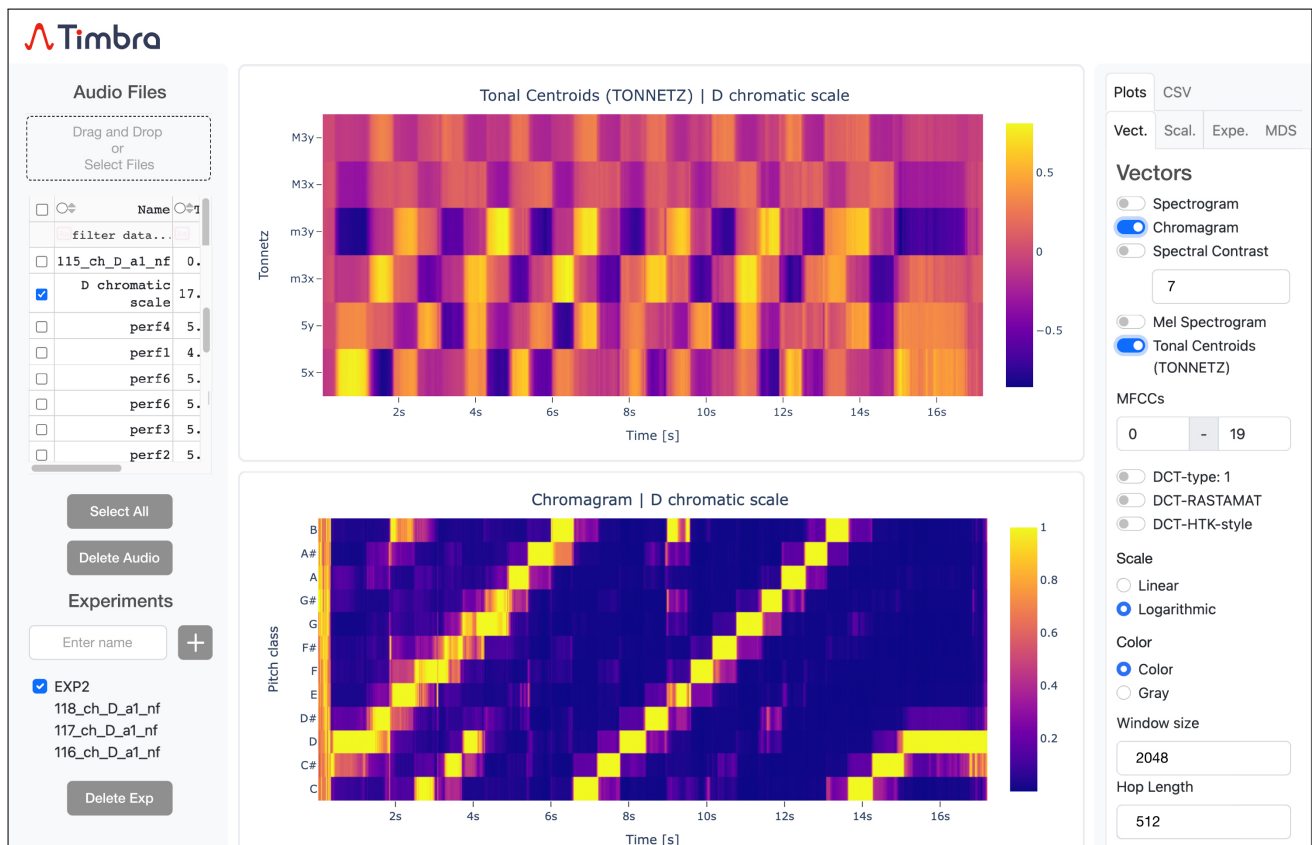


Fig. 2. Timbra. The user interface components: left-side toolbar, the main workspace, right-side toolbar.

a table for managing AS objects that can be sorted by their name, duration, sampling rate, format, or a combination of these elements. After selecting at least one AS object in the table, an AE object can be created by entering its name in a dedicated text field below the table. At the bottom, there is a checklist for managing AE objects. Each AE object has a list of the AS objects that belong to it. Using respective buttons, it is possible to delete objects of both types.

The right-side toolbar is used for creating, selecting and managing charts created from AS and AE objects that have been selected in the left toolbar and for generating CSV files. This toolbar is divided into two main tabs: *Plots* and *CSV*. Both tabs have their own sub-tabs.

In the *Plots* tab, individual sub-tabs (*Vectors*, *Scalars*, *Experiments*, *Multidimensional Scaling*) have the following functions:

Vectors tab is used for generating and displaying graphs with vector characteristics of selected AS objects. Calculated vector features include: spectrogram, chromagram, spectral contrast, Mel spectrogram, tonal centroids (TONNETZ), mel frequency cepstral coefficients MFCCs (DCT-type: 1, DCT-RASTAMAT, DCT-HTK-style).

Scalars tab is used for generating and displaying graphs with scalar characteristics of selected AS objects, presented against the spectrogram background. Calculated scalar features include: fundamental frequency, spectral bandwidth, spectral centroid, spectral flatness, zero crossing rate (as a temporal feature ZCR is plotted with an audio waveform in the

background instead of a spectrogram), spectral roll-off, band energy ratio. All the parameters that require the user to provide additional information, e.g. cepstral coefficients or spectral roll-off, have special fields for selecting the options to be transferred for parameter calculations.

Experiments tab is used for generating and displaying graphs that compare the parameters of the AS objects included in the selected AE objects. Available chart and feature options are: LTAS (RMS) i.e. the root mean square value of a long-term average spectrum, bar plot, scatter plot and frame-based scatter plot, LTAS based MFCCs (DCT-type: 1, DCT-RASTAMAT, DCT-HTK-style), and LTAS based chromagram. Charts such as bar plot, scatter plot, and frame-based scatter plot also have the option of selecting the parameters that the user wants to compare. The choice is made using a drop-down list. In the case of parameters such as spectral roll-off or band energy ratio, which require an additional input from the user, a dedicated field is displayed with the option of entering the values needed for calculations.

Multidimensional Scaling tab (MDS) is used for generating and displaying two- or three- dimensional graphs obtained by performing multidimensional scaling of the parameters selected from the drop-down list for each AS object included in the selected AE object. Available MDS charts are: 2D MDS, 2D MDS (frame by frame), 2D MDS (all frames), 2D MDS (Stress), 3D MDS, 3D MDS (frame by frame), 3D MDS (all frames), 3D MDS (Stress), Stress per number of dimensions in MDS. Additionally, for calculating the MDS, the user can



Fig. 3. Bar plot. Spectral roll-off 85%. LTAS based value per player.

manipulate the following options: Number of Initializations, Number of Max Iterations, Eps - relative tolerance.

At the bottom of the right-side toolbar, there are options that can influence the parameters calculation and displaying charts selected in the first three sub-tabs, i.e. *Vectors*, *Scalars*, and *Experiments*. Parameter calculation options include manipulation of Fourier Transform window size and hop length. Chart display options include: *Color* – to display charts in color or grayscale, *Scale* – to display charts on a logarithmic or linear scale.

In the *CSV* tab, the user can select parameters that are to be packed into CSV files and prepared for download. Each of the sub-tabs (*Vectors*, *Scalars*, *LTAS*, *Experiments*) generates a separate file or a set of CSV files.

IV. CASE STUDIES

A. Exploring timbre dissimilarities of cello players

In this study, we show how Timbra can be helpful in a preliminary investigation on timbre differences between cello players when comparing their recordings made on the same instrument in controlled acoustical conditions. The aim of the study was to reveal whether classical musicians can be discriminated by their tone quality using extracted acoustic parameters. The major experiments and research results were reported in [15].

Sound corpora. Music samples from six advanced cellists were recorded on a cello of a good luthier class, using the same high quality bow. The recording sessions took place in a professional studio where the recording setup was strictly preserved through all the sessions. The audio signal was captured by a high class ambient mono microphone (model Neumann U87Ai P48) located in the near-field to the cello. From the recorded repertoire (more details in [15]), a fragment of the 20th century *Cello Sonata* by Shostakovich: Movement I (bars 3–5: notes 5–15) was selected for this study.

Dissimilarity analysis using Timbra. The Timbra's interface offers calculation and visualization of scalar and vector parameters of each audio file individually or in bulk. Then, the *Experiments* tab can be used to compare extracted features across a selection of audio samples by means of bar and scatter plots.

To identify which acoustic parameters best describe varying timbral characteristics of the players, bar plots of each scalar feature, namely spectral bandwidth (SB), spectral centroid (SC), spectral roll-off (SR95% and SR85%), band energy ratio (BER2000Hz and BER4000Hz), spectral flatness (SF), and zero crossing rate (ZCR) were visually reviewed across six

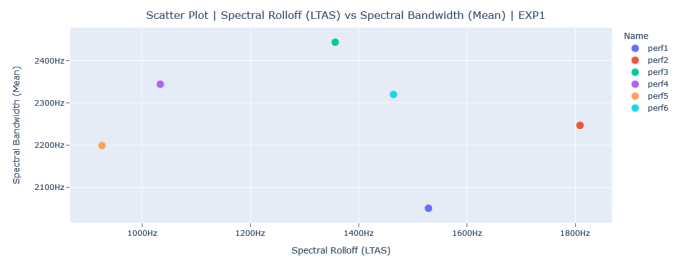


Fig. 4. Scatter plot. Spectral roll-off 85% (LTAS) vs. spectral bandwidth (mean) across six cellists.

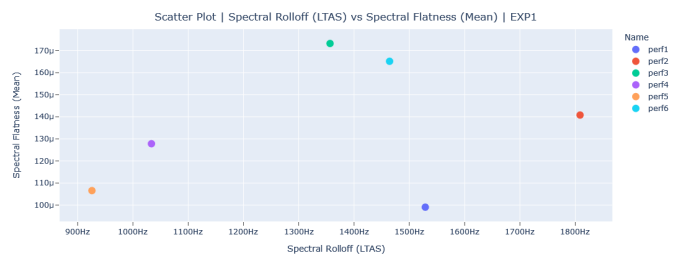


Fig. 5. Scatter plot. Spectral roll-off 85% (LTAS) vs. spectral flatness (mean) across six cellists.

cellists, followed by the inspection of scatter plots displaying the descriptors in pairs. Both LTAS based and mean values across all audio frames were compared.

The strongest separation between the players in 2D space was observed for SR85% (LTAS) vs. SB (mean) and for SR85% (LTAS) vs. SF (mean) (see Figures 4 and 5). Figure 3 shows a bar plot of SR85% LTAS based values per player. In both feature spaces, the samples of six cellists could be separated into four clusters, namely perf4 and perf5, perf3 and perf6, perf1, and perf2, indicating some shared timbre characteristics between the players as well as dissimilarities. In terms of *brightness* (spectral roll-off), the largest difference was observed between perf2 and perf5 and the smallest between perf4 and perf5. Similar results for the three cellists were obtained in [15] for spectral centroid identified as the parameter most strongly correlated with the *brightness* dimension when a larger set of audio samples from various music excerpts was analyzed and compared. While different methods of feature calculations and advanced statistical analyses were employed in the former work, this example shows the usefulness of the tool in gaining preliminary insights into the studied data and helping to formulate initial research hypotheses.

B. Comparing timbre of recorded and resynthesized tones of musical instruments

In the second experiment, we used Timbra to compare tones of recorded original musical instruments with tones of their resynthesized versions. The synthesized sounds were obtained in the project, the aim of which was to build a VST plugin for software synthesizer imitating sounds of recorded musical instruments [32]. Sound analysis/synthesis was performed according to the method introduced by Serra and Smith in [33]. The original tones had to be first analyzed in

the context of the number, amplitude, and variation of building sinusoids (deterministic components) and the noise. The noise is modeled by the LPC filtration method with the changing number of coefficients. Adjusting the number of sinusoids and noise characteristics causes the synthesized sounds to become closer to the original. The perceptual similarity may be evaluated in listening tests, whereas objective similarity has to be examined by analyzing adequate parameters such as spectral flatness that describes the tonality of the sound and indicates how much noise is in the signal, and cepstral coefficients that represent the spectral envelope.

Figure 6 shows the LTAS of five exemplary sounds: original accordion tone (org), synthesized tones composed of 49 sinusoids (f_{0_det_5}, f_{0_det_20}), and tones synthesized from the sum of the same sinusoids with added noise modeled with 5 or 20 LPC coefficients (f_{0_sum_5}, f_{0_sum_20}). Figure 7 presents the bar plot of spectral flatness, indicating how much noise is in the five analyzed signals. The highest flatness value may be seen in the signal f_{0_sum_5}, which concurs with Figure 6. Figure 8 visualizes the similarity between the spectral envelopes in individual frames of the original and synthesized versions of the accordion sound captured in the space of 13 MFCCs using 3D multidimensional scaling. The most distinct from the others is again the synthesized version f_{0_sum_5}, whereas the closest to the original is the version f_{0_sum_20} with better-adjusted noise component.

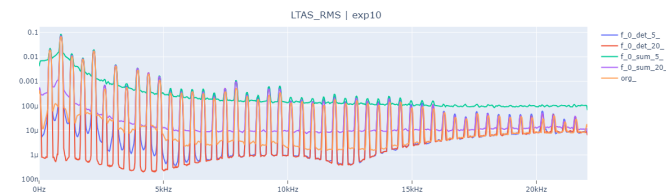


Fig. 6. LTAS (RMS) of the original and synthesized versions of the accordion sound

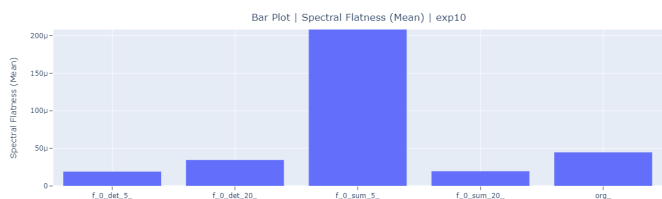


Fig. 7. Bar plot. Spectral flatness (mean) of the original and synthesized accordion tones.

V. CONCLUSIONS

In this paper, we have presented Timbra, the online tool for timbre analysis, that enables calculation, visualization and comparison of features related to timbre. Timbra can be helpful in a preliminary investigation on timbre differences between various sounds either in their averaged feature values, or their values per individual audio frame. Multidimensional scaling enables visualization of distances between sound objects in multiple dimensions reduced to 2D or 3D. We have described

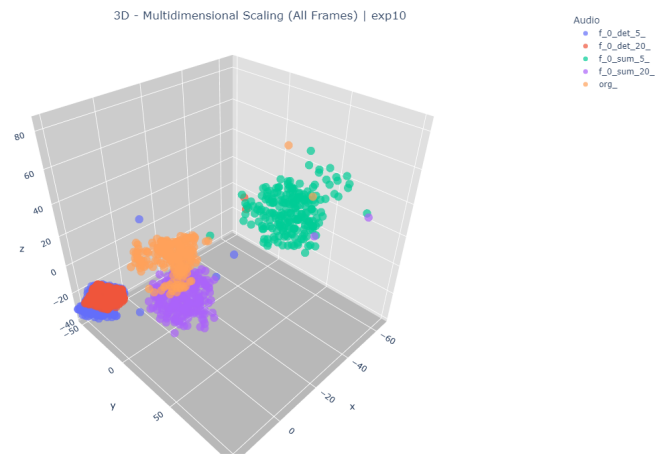


Fig. 8. 3D Multidimensional scaling of all frames' MFCCs (1-13) computed from the original and synthesized accordion tones

the architecture, libraries used, functionality, and user interface of the Timbra tool.

We have showcased two exemplary applications of Timbra: (i) examining timbre differences between cello players when comparing their recordings made on the same instrument in controlled acoustical conditions, and (ii) finding the resynthesized sound of a musical instrument that is the most similar to the original recording. The tool does not require any programming skills from a potential user and may serve musicologists, sound engineers, luthiers, educators, or students who need to get acquainted with the interpretation of timbre descriptors. With the introduction of Timbra, we expect low level computational timbre analysis to become applicable to new research areas. The system will be available from 2024 but can be tested earlier upon request.

ACKNOWLEDGMENT

This work has been supported by the DARIAH-PL project co-funded by the European Union through the Smart Growth Operational Programme 2014-2020, Action 4/4.2/2020 "Development of modern research infrastructure of the science sector".

REFERENCES

- [1] Dariah.lab. (2023) Digital Research Infrastructure for the Arts and Humanities. [Online]. Available: <https://lab.dariah.pl/en/>
- [2] H. Helmholtz, *On the Sensations of Tone: As a Physiological Basis for the Theory of Music*, 4th ed. London: Longmans, Green & Co., 1877, (1912). Translated by Ellis, A. J. [Online]. Available: <https://doi.org/10.1017/CBO9780511701801>
- [3] ANSI, *American Standard Acoustical Terminology, Definition 12.9, Timbre*, New York, 1960. [Online]. Available: <https://asastandards.org/terms/timbre/>
- [4] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *Journal of the Acoustical Society of America*, vol. 61, no. 5, pp. 1270–1277, 1977. [Online]. Available: <https://doi.org/10.1121/1.381428>
- [5] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1493–1500, 1978. [Online]. Available: <https://doi.org/10.1121/1.381843>
- [6] C. L. Krumhansl, "Why is musical timbre so hard to understand?" in *Structure and Perception of Electroacoustic Sound and Music*, S. Nielzen and O. Olsson, Eds., Elsevier, Amsterdam, 1989, pp. 43–53. [Online]. Available: http://music.psych.cornell.edu/articles/timbre/Why_Is_Musical_Timbre_so%20hard_to_understand.pdf

- [7] P. Iverson and C. L. Krumhansl, "Isolating the dynamic attributes of musical timbre," *Journal of the Acoustical Society of America*, vol. 94, no. 5, pp. 2595–2603, 1993. [Online]. Available: <https://doi.org/10.1121/1.407371>
- [8] S. McAdams, S. Winsberg, S. Donnadieu, G. Desoete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychological Research*, vol. 58, no. 3, pp. 177–192, 1995. [Online]. Available: <https://doi.org/10.1007/BF00419633>
- [9] S. McAdams, B. L. Giordano, P. Susini, G. Peeters, and V. Rioux, "A meta-analysis of acoustic correlates of timbre dimensions (A)," *Journal of the Acoustical Society of America*, vol. 120, no. 5, p. 3275, 2006. [Online]. Available: <https://doi.org/10.1121/1.4777215>
- [10] B. L. Giordano and S. McAdams, "Sound source mechanics and musical timbre perception: Evidence from previous studies," *Music Perception*, vol. 28, no. 2, pp. 155–168, 2010. [Online]. Available: <https://doi.org/10.1525/mp.2010.28.2.155>
- [11] R. A. Kendall, E. C. Carterette, and J. M. Hajda, "Perceptual and acoustical features of natural and synthetic orchestral instrument tones," *Music Perception*, vol. 16, no. 3, pp. 327–363, 1999. [Online]. Available: <https://doi.org/10.2307/40285796>
- [12] S. Lakatos, "A common perceptual space for harmonic and percussive timbres," *Perception & Psychophysics*, vol. 62, no. 7, pp. 1426–1439, 2000. [Online]. Available: <https://link.springer.com/content/pdf/10.3758/BF03212144.pdf>
- [13] R. A. Fitzgerald, "Performer-dependent dimensions of timbre: identifying acoustic cues for oboe tone discrimination," Ph.D. dissertation, School of Music, University of Leeds, 2003.
- [14] M. Barthelet, P. Guillemain, R. Kronland-Martinet, and S. Ystad, "From clarinet control to timbre perception," *Acta Acustica united with Acustica*, vol. 96, no. 4, pp. 678–689, 2010. [Online]. Available: <https://doi.org/10.3813/AAA.918322>
- [15] M. Chudy, "Discriminating music performers by timbre: On the relation between instrumental gesture, tone quality and perception in classical cello performance," PhD thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London, 2016. [Online]. Available: https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/18378/CHUDY_Magdalena_PhD_120916.pdf?sequence=1
- [16] H.-G. Kim, N. Moreau, and T. Sikora, *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. Chichester, UK: Wiley, 2005. [Online]. Available: <https://doi.org/10.1002/0470093366>
- [17] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The Timbre Toolbox: Extracting audio descriptors from musical signals," *Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, 2011. [Online]. Available: <https://doi.org/10.1121/1.3642604>
- [18] O. Lartillot, "Miningsuite: A comprehensive Matlab framework for signal, audio and music analysis, articulating audio and symbolic approaches," in *Proceedings of the 16th Sound & Music Computing Conference*, Malaga, Spain, 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.3249435>
- [19] O. Lartillot, P. Toivainen, and T. Eerola, "A Matlab toolbox for Music Information Retrieval," in *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization*, C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, Eds. Springer-Verlag, 2008, pp. 261–268. [Online]. Available: https://doi.org/10.1007/978-3-540-78246-9_31
- [20] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, and et al., "ESSENTIA: An audio analysis library for Music Information Retrieval," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR'13)*, 2013, pp. 493–498, <http://essentia.upf.edu/index.html>. [Online]. Available: <https://doi.org/10.1145/2502081.2502229>
- [21] G. Tzanetakis and P. Cook, "MARSYAS: A framework for audio analysis," *Organised Sound*, vol. 4, no. 3, p. 169–175, 2000, <https://github.com/marsyas/marsyas>. [Online]. Available: <https://doi.org/10.1017/S1355771800003071>
- [22] P. M. Brossier, "The audio library at MIREX 2006," in *Music Information Retrieval Evaluation eXchange (MIREX)*, 2006, <https://aubio.org/>. [Online]. Available: <https://www.music-ir.org/mirex/abstracts/2006/MIREX2006Abstracts.pdf>
- [23] J. Bullock, "Libxtract: A lightweight library for audio feature extraction," in *Proceedings of the 2007 International Computer Music Conference*, Copenhagen, Denmark, 2007, pp. 25–28, <https://www.jamiebullock.com/LibXtract/documentation/>. [Online]. Available: <https://quod.lib.umich.edu/i/icmc/bbp2372.2007.116/1>
- [24] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "YAAFE, an easy to use and efficient audio feature extraction software," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR'10)*, Utrecht, Netherlands, 2010, p. 441–446, <http://yaafe.github.io/Yaafe/>. [Online]. Available: <https://ismir2010.ismir.net/proceedings/ismir2010-75.pdf>
- [25] C. McKay, I. Fujinaga, and P. Depalle, "jAudio: A feature extraction library," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005, p. 600–603, <https://jmir.sourceforge.net/jAudio.html>. [Online]. Available: <https://ismir2005.ismir.net/proceedings/2103.pdf>
- [26] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proceedings of the 14th Python in Science Conference*, 2015, <https://librosa.org/>. [Online]. Available: http://conference.scipy.org/s3-website-us-east-1.amazonaws.com/proceedings/scipy2015/brian_mcfee.html
- [27] H. Rawlinson, N. Segal, and J. Fiala, "Meyda: An audio feature extraction library for the Web Audio API," in *Proceedings of the 1st Web Audio Conference*, Paris, France, 2015, <https://meyda.js.org/>. [Online]. Available: https://webaudioconf.com/posts/2015_17/
- [28] D. Moffat, D. Ronan, and J. D. Reiss, "An evaluation of audio feature extraction toolboxes," in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015. [Online]. Available: https://www.dafx.de/paper-archive/2015/DAFx-15_submission_43_v2.pdf
- [29] Django. (2023) The web framework for perfectionists with deadlines. [Online]. Available: <https://www.djangoproject.com/>
- [30] Dash. (2023) The low-code framework for rapidly building data apps in Python. [Online]. Available: <https://dash.plotly.com/>
- [31] Plotly. (2023) Low-code data apps. [Online]. Available: <https://plotly.com/>
- [32] M. Mrozik, "VST software synthesizer imitating sounds of recorded musical instruments," Master's thesis, Poznan University of Technology, 2023, E. Łukasik, supervisor.
- [33] X. Serra and J. Smith III, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, pp. 12–14, winter 1990. [Online]. Available: <https://doi.org/10.2307/3680788>