Touching Sounds: Praat Data Sonification for Tang-Audible User Experience (In Progress)

Brian Dozier Brown
Department of Music, State University of New York, Stony Brook, New York 95616, USA
(Dated: December 14, 2022)

The scientific study of linguistics is inherently preoccupied with our sense of hearing as it relates to phonetics and draws upon several forms of data visualization to evaluate complex speech phenomena. However, this dependency on translating sonic information to the visual realm presents an accessibility disparity for visually impaired scholars. This paper details the development and design of a web application to sonify formant data from Praat to assist visually impaired users with formant data analysis through non-speech auditory feedback and speech output.

I. INTRODUCTION

Data sonification has always involved the transformation of data into auditory cues and is defined analogously as the "auditory counterpart of data visualization" (Sawe et al. 2020). Over time, a thinning of its description has been adopted by some scholars to specifically pertain to the rendition of non-speech audio to convey data and its relations (Kramer et al. 1999). In any case, the framework on which any system of sonification has been designed is described as its auditory display (Hermann 2008). Data sonification as a field of research itself is inherently interdisciplinary, operating at the nexus of several scientific fields (Kaper et al. 1999). Decisions regarding the appropriate methods and mapping strategies for sonification are highly contingent upon the datasets themselves and require knowledge of the field from which that data is derived (Kramer 1994). Sonification is particularly apt at mining time series data and elucidating latent patterns in complex datasets due to our auditory sensitivity to changes in periodicity (Kramer et al. 1999, Last and Gorelik 2008). It is also frequently used as a sensory substitute for displays that are inaccessible to visually impaired persons, and as a complement to visual displays (Hildebrandt et al. 2016, Zhao et al. 2008).

The type of non-speech audio signal used for data sonification may be either musical or nonmusical and operates somewhat independently from the auditory display's function (Vickers and Hogg 2013). There are roughly 4 categorical areas that may describe an auditory display's function: alerts, status updates, entertainment, and data exploration (Walker and Nees 2011). The nature of each of these categories dictates, to some extent, the design of their auditory display. For instance, alerts that warn users of a potentially life-threatening hazard (such as a fire alarm) are generally designed to be quite loud, spectrally intrusive, and persistent (Terasawa et al. 2015). In the case of data exploration, which has come to be the default function of "sonification" when referenced generally, the underlying responsibility to scientific literacy asks the representative auditory display to have a balance between its fidelity to the data, level of complexity, aesthetic appeal, and accessibility (Sawe et al. 2020, Walker and Nees 2011). This commitment to scientific literacy also dictates the importance of a "query-based" mode of interaction that provides users access to a method to efficiently recycle information for comprehension (Franklin and Roberts 2004). This contrasts with more non-interactive auditory display modes that may trigger without users having any degree of control of the display or the ability to customize the display's characteristics otherwise. Ultimately, these functional dependencies are mostly carried out by the parameter mapping strategies employed by the auditory display.

For data sonification, parameter mapping typically describes the translation of numerical or physical data quantities into a characteristic of an acoustic signal, such as its pitch (frequency), intensity or loudness (amplitude), tone quality (timbre), and location (spatialization) (Zanella et al. 2022). Walker and Nees refer to such systems of sonification as "event-based" approaches and detail how the number of malleable sound attributes augments the potential design space of the auditory display (2011). Thus, not only are direct one-to-one and convergent many-to-one mapping topologies available between data and sonification, low-dimensional datasets may be eligible for divergent one-to-many auditory mappings (Grond and Berger 2011). However, for data exploration and scientific clarity, higher dimensional mappings should only be employed whenever they do not detract from the auditory display's perceptibility (Sawe et al. 2020).

Event-based mappings are also subject to a psychoacoustic implicit bias towards sonic characteristics that humans demonstrate higher sensitivity to, leading sonification practices to correlate the most important data parameters with sound attributes such as pitch and timbre (Hegg et al. 2018). This trend is reflected in Dubus and Bresin's review of mapping strategies for physical quantities, which found over 50% of mappings between physical and auditory dimensions to involve pitch in an analysis of over 495 mapping occurrences across 60 sonification projects (2013).

In addition to mapping parameters according to the most sensitive auditory dimensions, the intuition behind sonification mappings also seems to "follow the logic of ecological perception" (Dubus and Bresin 2013). In other words, there is a tendency towards a mimetic mapping

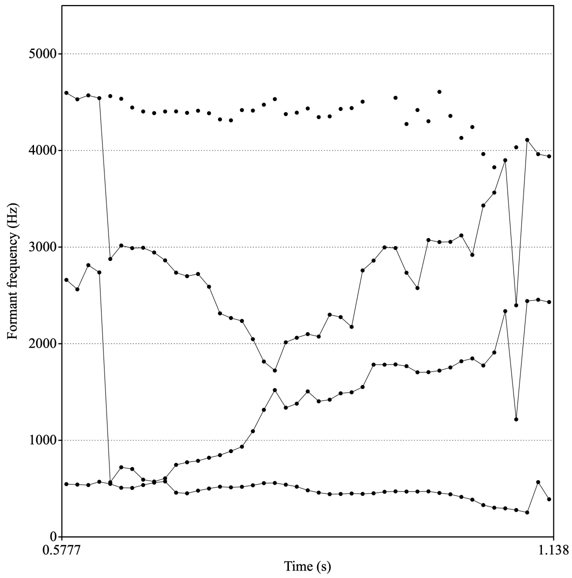

FIG. 1: F_1-F_3 Formant Contours of "Orange" Interpolated through Point Estimations in Praat

from the data's physicality to the auditory characteristic which Hermann and Ritter outlined as an appeal to "universal relationships" (2004). This conscientious attitude wherein researchers employ a more semiotic and thus, signal-referent approach to auditory mappings would result in a stronger distribution of analogic sounds in the literature as opposed to symbolic sounds (Pirhonen et al. 2006, Worrall 2019). It also parallels the more intentional model-based sonification mapping strategy that relies on our facility to associate sounds with the physical characteristics of their perceived source (Hermann 2018).

The present study primarily uses Walker and Nees's taxonomy of sonification as a framework for designing an auditory display for speech formant analysis (2011). Although nonanomalous due to the data's quantitative nature, this study's dataset distinguishes itself amongst some of the most preeminent examples of sonification because the data is extracted from another auditory display — the speech itself. If this study attempted to employ a model-based approach to sonification, any attempt to relate the physical rendering of the data to an auditory display would simply result in the original sound being enacted as the novel display. This evinces a necessity for an event-based sonification approach that performs a task comparable to a type of dimensionality reduction of the complexity of sonic data laden into the raw speech waveform (Zebari et al. 2020). As a result of this reduction, the current model of this application addresses two analytical tasks of speech analysis: formant tracking and estimation (Dissen et al. 2019). Formants are frequencies resonances in the vocal tract (Fant 1960). Formant tracking describes the process of monitoring the trajectory of a formant's contour, and this contour is an interpolation of formant frequencies identified across a time series of speech frames: Fig 1. Formant estimation is the process of identifying the formant frequencies for this latter operation (Yegnanarayana et al. 2020). Accordingly, these tasks conform neatly to Smith and Walker's description of point estimation and point comparison analytic listening tasks. They specify that in order for any auditory graph undertaking point estimation to be successful, "the listener must: 1) listen to the sonification; 2) determine in time when the datum of interest occurs; 3) upon identifying the datum of interest, estimate the magnitude of the quantity represented by the pitch of the tone; 4) compare this magnitude to a baseline or reference tone (i.e., determine the scaling factor); and 5) report the value" 2005. Because this study is designed to benefit academic researchers and scientific literacy, the accuracy and reliability of our auditory display are of chief concern. To ensure this, a combination of both non-speech audio and speech output has been used in this sonification project.

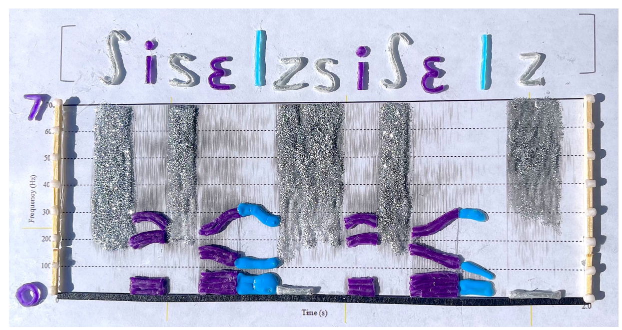

FIG. 2: Crafted Tactile Spectrogram of "She Sells Seashells" produced by Dr. Marie Huffman for the Touching Sounds Exhibit 2022 at Stony Brook University

II. HISTORY OF TOUCHING SOUNDS PROJECT

This project for sonification was initiated as part of a larger initiative to provide visually impaired students and scholars with a means for interpreting formant data outside of explicit values analysis. This initiative sought to stimulate tactility with haptic devices such as hand-crafted spectrograms and 3d-printed spectrograms,(Wolek and McEllroy 2022) and audibility through data sonification: Fig 2. Collectively referred to as the "Touching Sounds" project, early outreach efforts in 2022 took the form of a modest exhibition of hand-crafted spectrograms and the associated website accessible through QR codes.

Prototyping for the website's User Interface (UI) was outlined in Adobe XD before being hard-coded into HTML, CSS, and Javascript: Fig 3. During this early stage of web development is when most of the accessibility considerations were introduced, including the necessity of subtext for screen reader compatibility and high-contrast color schemes to accommodate the spectrum of visual impairment (Chan et al. 2020). Most importantly, a method to verbalize IPA symbols that may be displayed above each syllable of the audio file was devised so that screen readers could communicate these symbols to users. Each of these features, subtext and high color contrast, has been maintained in this subsequent sonification application.

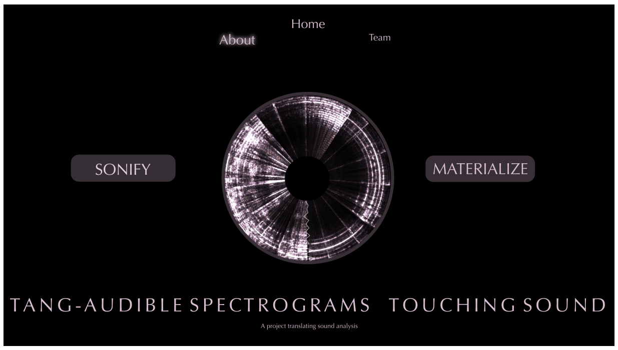

FIG. 3: Early Touching Sounds Website Prototyping in Adobe XD

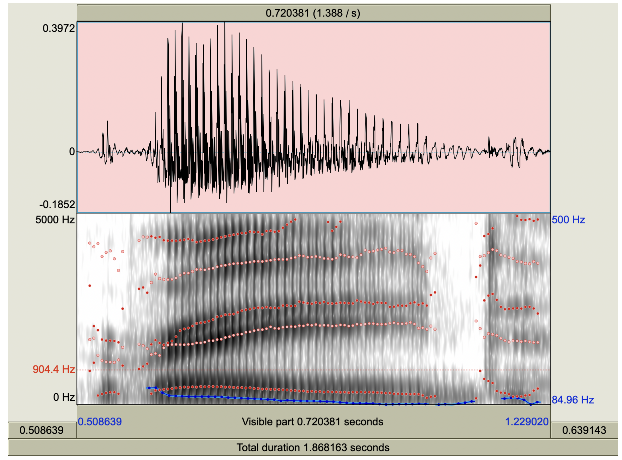

FIG. 4: "Raid" Spectrogram with Superimposed Formant Values in Praat

III. FORMANT TRACKING

One consideration in building the application is its interoperability with Praat,(Boersma and Weenink 2021) which is one of the foremost software tools for speech analysis (Cantoni et al. 2022). In order for the data provided by the application to conform with the results of Praat's spectrographic analysis, it needs to process voice samples using the same algorithm as Praat. Because this application is written in Python, it relies on Parselmouth, a library and API capable of directly accessing Praat's internal C/C++ scripting, to generate spectrogram and formant data (Jadoul et al. 2018). Formant tracking is one of the primary methods of meaningful formant analysis. One of the most integrated methods Praat allows visual users to perform formant tracking with is by superimposing an analysis of the formant estimates upon a spectrogram of the audio sample: Fig 4.

Praat offers a few algorithms for performing spectral analysis; however, each of them processes the target sound sample as a precursor to the formant-tracking algorithm. Such preparations include routing the sound through a resampling algorithm that essentially repurposes the Formant Ceiling as the Nyquist frequency of the resampled output, and a pre-emphasis filter to boost the signal strength. Afterward, Praat computes the LPC (linear predictive coding) coefficients according to either the Burg or Split-Levinson procedures. Due to reliability concerns, this application borrows Praat's implementation of the Burg algorithm, which also removes extraneous formants below 50Hz and above the Formant Ceiling minus 50Hz (Boersma and Weenink 2021).

IV. INTERFACE DESIGN AND FUNCTIONALITY

Details surrounding the design for the application's auditory display were largely dependent on its function as a tool for data exploration and its purpose to assist visually impaired Praat users. Accordingly, the mapping process involved with encoding the formant information in the audio domain primarily considered data function (formant tracking and estimation), keyboard accessibility, and screen reader compatibility. The ensuing design description will be organized according to the application's auditory display feature space.

A. Intensity Detection

Intuitively, formants and harmonics may be visually inferred from a spectrogram by identifying the center frequency within a bandwidth of sound that displays a relative peak in sound intensity measured in SPL (Sound Pressure Level) (Jiang et al. 2021). Praat's spectrograms indicate frequency intensity according to a greyscale colormap wherein darkness maps to increased intensity. Emulating the visual quality that darkness clarity plays in determining formant strength, the audio display correlates signal distortion by means of a noise generator with intensity such that darkness maps inversely with noise. That is, the sine wave of darker formants that demonstrate higher sound pressure levels will be purer and thus, more easily perceptible than light formants that demonstrate lower sound pressure levels and have more noise in their signal. This runs contrary to a parameter mapping based on ecological perception, as a direct mapping

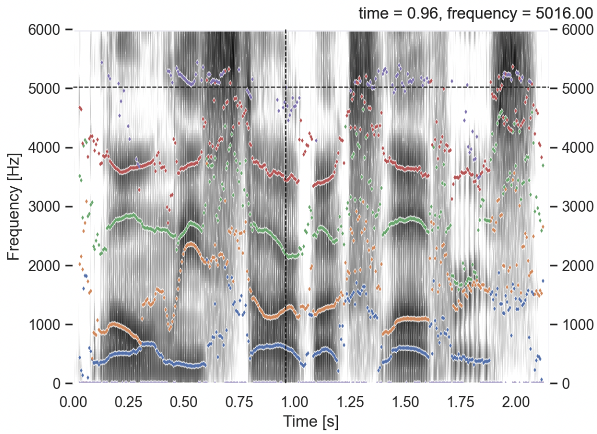

FIG. 5: "Lonely Starbucks Lovers" Spectrogram with Superimposed Formant Estimations in new Application

of sound pressure level to itself(loudness) would instantiate a sound parameter that is significantly less reliable. For one, this mapping strategy would not be optimal for conveying information for scientific literacy because the ability for users to alter volume ranges with their systems internal settings would disrupt the consistency of information delivery. Furthermore, parameter mappings to loudness have been demonstrated to be relatively less effective due to our sensitivity to loudness (Neuhoff et al. 2002). Finally, this sort of mapping would not allow a linear mapping between darkness value and loudness because we perceive loudness logarithmically, meaning a 6 decibel(dB) increase/decrease between lower values of dbSPL would be significantly more noticeable than a 6db shift between higher values (Robin and Plante 2022).

B. Navigation

The controls for data exploration are motivated by the visual tracing method employed during formant contour assessment. Replacing the point of focus with a cursor, this application splits the data into regions according to the formant (F_1, F_2, etc.) and allows users to trace formant data laterally, stepping between adjacent data points within each formant with the '`left arrow`' and '`right arrow`' keys: Fig 5 A bandwidth-skipping feature wherein users can skip to the next spectrally unique bandwidth of sound increases the efficiency and convenience of the application. This feature allows users to skip between words and syllables with ease, while also providing them with a means of skipping through noisy plosives and fricatives without needing to click through each data point. The primary method of interfacing with this cursor is through the keyboard. Speech output functions as an auditory indicator of keyboard focus to update users on their current position in the graph. Chiefly, this speech output will be enacted by shifts in formant (data subset changes from F_1 to F_2) and bandwidth changes (skipped data points). Lastly, users have the ability to initiate an automatic tracing, wherein the cursor moves across the current formant level selected from left to right, allowing users to quickly trace formant values.

Another indication of the user's spatial orientation on the graph is the panning of non-speech audio. Although audio spatialization may potentially provide a means for a 3-dimensional auditory representation of our graph, there were a few motivations for operating within a curtailed sound field. Encoding audio into a binaural audio format may be synthesized through Head-Related Transfer Functions (HRTFs), which would enable spatial data (in this case time (x) and frequency (y)) to be mapped into azimuth (horizontal) and elevation (vertical) angles (Algazi et al. 2001). However, there are differences in the perceptual acuity between the azimuth and elevation planes. For instance, spatial disintegration is notably reduced along the central elevation axis, wherein disorientation renders front, back, up, or down localization indistinguishable. This phenomenon is often referred to at the "cone of confusion" (Risoud et al. 2018).

Another incompatibility arises from the difference between sound localization in the azimuth and elevation angles and our mapping of sound intensity to a signal-to-noise (SNR) gradient. Humans undertake sound localization within the azimuth plane with calculations involving two forms of information — interaural time differences (ITDs) and interaural level differences (ILDs) that are extracted from changes to a sound's condition once it reaches each ear (Algazi et al. 2001). Neither of these binaural processes, detecting discrepancies in phase (time) nor SPL (loudness), are pertinent to the SNR timbral gradient on which we map intensity data. However, elevation angles are monaurally derived from spectral cues, rendering them frequency-dependent functions (Algazi et al. 2001). This perceptive dependency on spectral information suggests a proportional relationship between localization accuracy and spectral richness of the sound source. This would trouble our use of an SNR gradient since the signal end of our gradient is a pure sine tone while the opposite end is the spectrally maximal timbre of white noise. In essence, our gradient becomes more or less localizable, fundamentally transforming the target of our perceptibility spectrum.

For these reasons, stereo-panning of the SNR gradient output relays the cursor's position along the x-axis to users. All speech output remains in a central position due to their role as focus indicators; however, the SNR gradient (non-speech) output indicating formant strength and frequency is subject to stereo positioning (Kavcic 2005). Two-dimensional imaging of the aural space also remedies the issue of front-to-back differentiation, which are two indistinguishable positions without the help of monaural spectral cues (McAnally and Martin 2014). Instead, the front-to-back "cone of confu-

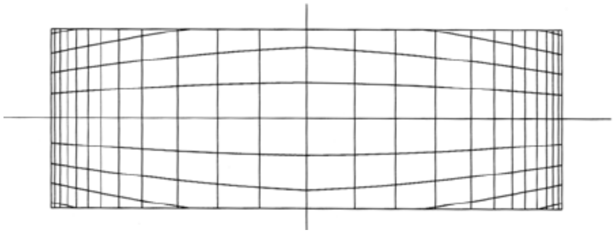

FIG. 6: Panoramic Distortion of X-Axis Values: Adjacent points closer to the center of the diagram are spatially farther apart than adjacent points at the edges, despite their displacement being the same.

sion" is simply inconsequential since the only important point is the central positioning along the x-axis, which both orientations imply. The primary lingering variable is in assessing the efficacy of stereo panning as a parameter of an auditory display, which is troubled by the nature of panning. A panoramic potentiometer (pan pot) stereo control allows users to artificially position audio in a virtual environment by adjusting the amplitude of each speaker's audio channel (Jot 1999). However, the values of this stereo positioning, which is encoded in polar space (r, θ, ϕ; the spherical coordinate system), are subject to panoramic distortion much in the same way that a panoramic photo is optically distorted along the x-axis: Fig 6. When translated to Cartesian space (x, y, z), this results in a distortion whereby adjacent values along the x-axis are spatially further apart at the center of the graph than adjacent values at the margins of the graph (Kopf et al. 2009). Because the spectrogram from which we extract formant data is graphed in Cartesian space, a direct mapping of the cursor's position on the x-axis to panning position may produce an ambiguous stereo imaging of time data.

C. Audio Controls

An appropriate balance between automated and initiated data verbalization is important to prevent users from being inundated with information and induce listening fatigue (Axon et al. 2018). Accordingly, a system of switches has been coded as audio controls through several boolean variables. Focus indicators such as the current formant and its associated frequency and timing data may be reiterated freely through keyboard commands. Users also have the option to export the formant/time data for reference as a JSON file with an array of formant values by pressing '`e`' or '`E`'.

Most importantly, spectrograms are generated from files that the user uploads to the application. This application allows users to play the current bandwidth of sound that the cursor lies within so that users can know precisely what syllable they are currently observing.

V. METHOD

Matplotlib is a comprehensive Python library based on the proprietary computing language MATLAB and is useful when plotting data into both static and interactive visualizations (Hunter 2007). This application accesses Matplotlib through the Pyplot API to provide a display window for the spectrogram and formant data. Fig 5 To draw the spectrogram, frequency, time, and amplitude data is pulled from ParselMouth's Spectrogram object and plotted into a Matplotlib figure. Once displayed, formant values are superimposed over that data with the '`draw_formant`' function. In order to build a cursor that can be used to step through each formant value, the '`time_formant_values`' information is handled as 5 distinct datasets corresponding with each format respectively. The '`time_values`' array is stored directly into a Matplotlib line variable, while the '`formant_values`' arrays are accessed with an additional variable that allows the user to select which index value from the 2-dimensional array to extract. This line data is passed as an argument into a '`blitted_cursor`' object which uses it for '`key_press events`' and '`mouse_motion events`.' Rendering issues with mouse tracking have been mitigated with the implementation of blitting, which is a data procedure that essentially stores a copy of the current bitmap (spectrogram pixel array in this case) and adjusts those pixels once a new image (cursor movement) is drawn to the screen (Pike et al. 1985). Essentially, it reduces the amount of information that needs to be redrawn as updates are detected.

A. Speech Output

As useful as non-speech auditory feedback may be for users, having the option for additional speech output of salient information significantly aids data analysis and table navigation. This application uses Google's Text-to-Speech (gTTS) API to generate a recitation of time and formant values in a human-like voice. Before passing the numerical data through the translate text-to-speech object, the application checks the data for undefined numerical values denoted computationally as Not a Number (NaN). If a NaN value is detected, a temporary variable specified as "Not a Number" is passed to the translate text-to-speech object. If a NaN value is not detected, the application rounds each value to a defined significant figure. This operation is performed to remove extraneous numbers from the data and prevent an excessively long speech output. Time data is rounded to 3 significant figures to accurately capture the time scale of short audio clips while formant data is rounded to 4 significant figures to produce whole number frequency values in Hertz. Typically, the text-to-speech API returns an audio format such as MP3 or WAV that contains the audio file of the text being verbalized. This would require users to generate an audio file that is written to the disk each

Table 1. Parameter mappings between sonified data and auditory dimension

Sonification Data	Auditory Mapping	Key Command
Selected Formant (F_1, F_2, F_3, etc.)	Relative Pitch Range	Up and Down Arrows “F” or “f” to speak current formant
Selected Formant Frequency Estimation	Pitch & Panning	Left and Right Arrows
Intensity	SNR	“I” or “i” to speak current value
Spatial (X, Time) Position	Panning	Left and Right Arrows “P” or “p” for Speech Output
Spatial (Y, Frequency) Position	Pitch	Left, Right, Up, and Down Arrows “P” or “p” for Speech Output
Export Time_Formant Values in JSON		“E” or “e”
Save Current Figure		“S" or “s"

FIG. 7: Key Commands and Mappings between Speech Data and Auditory Display

time they request speech output, which is not optimal for navigation updates while exploring data. Instead, this application uses Python's IO module to temporarily store the text-to-speech output into a file-like object that saves the information to memory. Now stored in a file-like object, it is loaded by the audio segment object from PyDub and output to the user.

B. File Extraction

Formant data is extracted by first converting the source audio file into a Parselmouth ‘`Sound`’ object. Afterward, a ‘`Formant`’ object containing all of the data related to that audio file's formants may be deduced by calling Parselmouth's ‘`formant_to_burg`’ method on that ‘`Sound`’ object. Praat iterates formant calculations for each frame of analysis in the window; these are the same frames used for generating a spectrogram of the source file. This iteration across frames (which signifies time in this case) is what produces a trackable array of values within a formant. Thus, in order for this application to serve as an auditory facsimile to Praat's visual interface, it needs to have the same number of formants with the same exact spacing as Praat by borrowing its unique time-step algorithm. Praat calculates the standard time-steps for formants as 25% of the Window length, which is the “effective” duration of an analysis frame (Boersma and Weenink 2021). To determine the time steps at which we need to save formant frequency information, the application stores the time steps as a list by calling the ‘`frame_number_to_time`’ method to iterate through each analysis frame of the spectrogram from first (1) to last denoted by the ‘`nt`’ attribute of the ‘`Formant`’ object that outputs the total number of frames. Once these time steps have been stored into a list, they are formalized into time values as a 1-dimensional array of size formant length (’`formant.nt`’). Finally, these ‘`time_values`’ are used to find each formant (F_1–F_2) in each frame by calling the ’`Formant`’ object's ‘`get_value_at_time`’ method and iterating through its arguments (formant number, time) in a nested loop. This last process outputs a 2-dimensional Numpy array of size ‘`formant.nt`’×5. Finally, both arrays are concatenated into an array ‘`time_formant_values`’ with the subsequent 2-dimensional structure of size ‘`formant.nt`’× 6: $\big\{$“audio_file_name”: $\big[[t_1, F_1t_1, F_2t_1, F_3t_1, F_4t_1, F_5t_1],$ $[t_2, F_1t_2, F_2t_2, F_3t_2, F_4t_2, F_5t_2], \ldots\big]\big\}$

Users have the option to export the ‘`time_formant_values`’ for reference in JavaScript Object Notation (JSON) by pressing ‘`e`’ or ‘`E`’. JSON files were opted for data extraction over Comma-Separated Values (CSV) to clarify data structure and help file management with name/value pairs for users. Our software executes this by setting up a JSON encoder to appropriately manage NumPy datatypes and then serializing that data into a JSON format before writing and dumping it into a file. The resultant filename is derived from the source audio file, and thus requires a protocol for overwriting that file should it already exist within the system's PATH if it runs natively on macOS.

VI. CONCLUSION

As an ongoing project, the next step would be to engage in a series of assessments with human subjects to optimize the application's efficacy, ergonomics, and accessibility/usability. Chiefly, this investigation will be seeking feedback on the aforementioned auditory interface design to eliminate any especially egregious features. Trials would seek to determine the negligibility of panoramic distortion on sound localization in the azimuth plane and the utility of a SNR gradient with assessing formant clarity. Additionally, another real-time sound synthesis programming environments (Supercollider, ChucK, MusicN, CSound, Max/MSP or PD) may be better suited for the digital sound synthesis that is necessary for this program, as Python's PyAudio environment has not proven the most efficient nor reliable for digital signal processing. Several issues related the API's native callback function has inhibited progress on the application. Lastly, Google's Text-to-Speech API relies an a stable internet connection, which significantly limits the applications mobility and efficiency. Another text-to-speech module may be better suited for this application.

Algazi, V. R., Avendano, C., and Duda, R. O. (2001). Elevation localization and head-related transfer function analysis at low frequencies. *The Journal of the Acoustical Society of America*, 109(3):1110–1122.

Axon, L. M., Alahmadi, B., Nurse, J. R., Goldsmith, M., and Creese, S. (2018). Sonification in security operations centres: what do security practitioners think?

Boersma, P. and Weenink, D. (2021). Praat: doing phonetics by computer [computer program]. version 6.1.38.

Cantoni, M. M., Oliveira, B. G. D., and Nevado, H. M. (2022). Introduction to acoustic speech analysis with praat. *Texto Livre*, 15.

Chan, K. S. C., Yuen, A. H., Yeung, P. P., and Wong, F. (2020). Utilising screen reader to support the transitions of learners with visual impairment. In *Careers for Students with Special Educational Needs*, pages 283–299. Springer.

Dissen, Y., Goldberger, J., and Keshet, J. (2019). Formant estimation and tracking: A deep learning approach. *The Journal of the Acoustical Society of America*, 145(2):642–653.

Dubus, G. and Bresin, R. (2013). A systematic review of mapping strategies for the sonification of physical quantities. *PloS one*, 8(12):e82491.

Fant, G. (1960). *Acoustic Theory of Speech Production.* Mouton Co, The Hague.

Franklin, K. M. and Roberts, J. C. (2004). A path based model for sonification. *Proceedings of the Eighth International Conference on Information Visualization, 2004*, IV 2004:865–870.

Grond, F. and Berger, J. (2011). Parameter mapping strategies. In Hermann, T., Hunt, A., and Neuhoff, J. G., editors, *The sonification handbook*, chapter 15, page 363–397.

Hegg, J. C., Robertson, B. L., and Kennedy, B. P. (2018). The sound of migration: exploring data sonification as a means of interpreting multivariate salmon movement datasets. *Heliyon*, 4:e00532.

Hermann, T. (2008). Taxonomy and definitions for sonification and auditory display. *International Conference for Auditory Display.*

Hermann, T. (2018). Wave space sonification. *Proceedings from the 24th International Conference for Auditory Display.*

Hermann, T. and Ritter, H. J. (2004). Sound and meaning in auditory data display. *Proceedings of the IEEE*, 92:730–741.

Hildebrandt, T., Amerbauer, F., and Rinderle-Ma, S. (2016). Combining sonification and visualization for the analysis of process execution data. In *2016 IEEE 18th Conference on Business Informatics (CBI)*, volume 2, pages 32–37.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Jadoul, Y., Thompson, B., and Boer, B. D. (2018). Introducing parselmouth: A python interface to praat. *Journal of Phonetics*, 71:1–15.

Jiang, Y., Grigorev, V., and Katsnelson, B. (2021). Estimation of parameters of moving nonlinear internal waves using spectrogram of sound intensity fluctuations due to mode coupling. *The Journal of the Acoustical Society of America*, 150(4):A328–A328.

Jot, J. M. (1999). Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia systems*, 7(1):55–69.

Kaper, H. G., Wiebel, E., and Tipei, S. (1999). Data sonification and sound visualization. *Computing in science & engineering*, 1(4):48–58.

Kavcic, A. (2005). Software accessibility: Recommendations and guidelines. In *EUROCON 2005-The International Conference on 'Computer as a Tool'*, volume 2, pages 1024–1027.

Kopf, J., Lischinski, D., Deussen, O., Cohen-Or, D., and Cohen, M. (2009). Locally adapted projections to reduce panorama distortions. In *Computer Graphics Forum*, volume 28, pages 1083–1089, Oxford: UK. Blackwell Publishing Ltd.

Kramer, G. (1994). An introduction to auditory display. In G.Kramer, editor, *Auditory display: Sonification, audification, and auditory interfaces*, pages 1–79.

Kramer, G., Walker, B., Bonebright, T., Perry Cook, J. H. F., Miner, N., and Neuhoff, J. (1999). Sonification report: status of the field and research agenda. report prepared for the national science foundation by members of the international community for auditory display.

Last, M. and Gorelik, A. (2008). Using sonification for mining time series data. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 63–72.

McAnally, K. I. and Martin, R. L. (2014). Sound localizatoin with head movement: implications for 3-d audio displays. *Frontiers in neuroscience*, 8(210).

Neuhoff, J. G., Kramer, G., and Wayand, J. (2002). Pitch and loudness interact in auditory displays: Can the data get lost in the map? *Journal of Experimental Psychology: Applied*, 8(1):17–25.

Pike, R., Locanthi, B., and Reiser, J. (1985). Hardware/software trade-offs for bitmap graphics on the blit. *Software: Practice and Experience*, 15(2):131–151.

Pirhonen, A., Murphy, E., McAllister, G., and Yu, W. (2006). Non-speech sounds as elements of a use scenario: A semiotic perspective. *Proceedings of the 12th International Conference on Auditory Display (ICAD06).*

Risoud, M., Hanson, J. N., Gauvrit, F., Renard, C., Lemesre, P. E., Bonne, N. X., and Vincent, C. (2018). Sound source localization. *European annals of otorhinolaryngology, head and neck diseases*, 135(4):259–264.

Robin, O. and Plante, C. (2022). An illustrated tutorial for logarithmic scales and decibels in acoustics. *The Journal of the Acoustical Society of America*, 152(5):2880–2892.

Sawe, N., Chafe, C., and Treviño, J. (2020). Using data sonification to overcome science literacy, numeracy, and visualization barriers in science communication. *Communication. Front. Commun.*, 5(46).

Smith, D. R. and Walker, B. N. (2005). Effects of auditory context cues and training on performance of a point estimation sonification task. *Journal of Applied Cognitive Psychology*, 19(8):1065–1087.

Terasawa, H., Morimoto, Y., Matsubara, M., Sato, A., Ohara, M., and Kawarasaki, M. (2015). Guiding auditory attention toward the subtle components in electrocardiography sonification. *2015 Proceedings of the International Conference on Auditory Display.*

Vickers, P. and Hogg, B. (2013). Sonification abstraite/sonification concréte: An 'aesthetic perspective space' for classifying auditory displays in the ars musica domain. *arXiv preprint arXiv:1311.5426*.

Walker, B. N. and Nees, M. A. (2011). Theory of sonification. In Hermann, T., Hunt, A., and Neuhoff, J. G., editors, *The sonification handbook*, chapter 2, page 9–39.

Wolek, N. and McEllroy, G. (2022). Designing 3d-prited spectrograms for blind students. *Proceedings of the International Computer Music Conference, Limerick, Ireland.*

Worrall, D. (2019). Sonification: An overview. *Sonification Design*, pages 23–54.

Yegnanarayana, B., Anand, J. M., and Pannala, V. (2020). Enhancing formant information in spectrographic display of speech. *INTERSPEECH*, pages 165–169.

Zanella, A., Lenzi, C. M., Cooke, J., Damsma, P., and Fleming, S. W. (2022). Sonification and sound design for astronomy research, education and public engagement. *Nature Astronomy*, page 1–8.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., and Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2):56–70.

Zhao, H., Plaisant, C., Shneiderman, B., and Lazar, J. (2008). Data sonification for users with visual impairment: a case study with georeferenced data. *ACM Transactions on Computer-Human Interaction*, 15(4):1–28.