Visualization and sonification of earthquake events

Andras Dörömbözi, 11921675

Main supervisor: Univ.Ass. Dipl.-Ing. Matthias Steiner, BSc
Co-supervisor: Univ.Ass. Victor Schetinger, PhD

May 1, 2022

1 Abstract

Earthquakes are sudden movements across faults that release elastic energy stored in rocks and radiate seismic waves that travel throughout Earth. Every day there are about fifty earthquakes worldwide that are strong enough (magnitude > 2.5) to be felt locally, and every few days an earthquake occurs that is capable of damaging structures. In addition, a multitude of smaller earthquakes (magnitude < 2.5) are happening that are too weak to be felt, but that are readily recorded by modern instruments. These small earthquakes provide valuable information about earthquake processes. Seismology is a data-rich and data-driven science, and the rate of data acquisition is accelerating as seismic sensors get steadily less costly. The massive and rapidly growing amount of data highlights the need for more effective tools for the efficient processing and extraction of as much useful information as possible to enable scientist to realize the full potential to gain new insights into earthquake processes from them. Seismologists use only a portion of the recorded data to understand the physics of earthquakes and learn about Earth's deep interior, where direct observations are impossible. Most seismic data sets have not been fully analyzed and important discoveries can result from reanalysis of data sets using new data analysis tools. [1]

2 Introduction

The goal of the project is the development of a dashboard, which represents past earthquake events on map, as well as corresponding receiver stations. The plotted events can be filtered by attributes, such as depth or the magnitude of the recorded earthquake event, as well as filtering on the time of recording, or the provider of the event recordings. The source dataset will be extracted from the STanford EArthquake Dataset [1]. That dataset will also include seismic signals of the recorded events, which will be represented for the events after selecting them on the map: the waveforms are represented in image form, as well as high-pass filtered spectrogram form, which scoops out unwanted low frequencies from any audio source, which low-frequencies might be irrelevant from the perspective of sonification since these frequencies are not audible for human ears. Additionally, the dashboard will include an audio player component, which provides the sonified amplitude-normalized form of earthquake events in order to make the earthquake waveform audible for human ear.

2.1 State-Of-The-Art

2.1.1 SeisSound

The SeisSound visual component includes the seismogram and corresponding spectrogram, presented in movie format indicating how the data evolve with time. An auditory sound file (WAVE format) of the data that are time compressed accompanies the visual information so the frequency content of the seismogram can be easily heard. Combining audio and visual information allows the user to both hear and see complexities in the frequency-time distribution of the seismogram that are often otherwise hidden in large-amplitude signals. These SeisSound video products provide a unique way to watch and listen to the vibration of the Earth, and help introduce more advanced topics in seismology. [2]

2.1.2 Alberta Earthquake Dashboard

The Earthquake Dashboard provides information relating to earthquakes in Alberta, and ground motion data from Alberta Geological Survey (AGS) seismic stations within the RAVEN monitoring network. The dashboard includes interactive maps that show the locations and magnitudes of earthquakes from 2006 to present, along with the locations of AGS RAVEN seismic monitoring stations. These earthquakes include both natural and induced seismic events, but exclude events relating to mining activities. The dashboard also provides access to real-time and archived waveform data, as well as an ability to view a summary of the seismic events (P phase, which are compression waves, material moves back and forth in the direction in which the wave propagates, and S phase or shear waves, material moves at right angles to the propagation direction [1]) that were identified on the AGS RAVEN seismic stations. [3] The waveforms are available from all three sensors, the orthogonal sensors are namely the North-South (N), East-West (E), and the vertical denoted as Z.

3 Visualization techniques

3.1 Focus + Context

Information Visualization often deals with data that users have no mental image of. A visualization imposes a graphical structure – a mapping from data to screen space – on the data that the user has to learn. It is therefore necessary to change this mapping as little as possible; but there is often not enough space on the screen to display all information with enough detail. Focus+Context (F+C) methods make it possible to show more, more detailed, or more targeted information – and at the same time, giving the user a sense of where in the data the zoomed-in, more detailed, or pointed out information is. [4]

3.2 Overview + Detail

Overview + Detail interfaces support navigation of the information space they display, through traditional panning and zooming mechanisms such as dragging the detail view to move it in the desired direction and changing magnification level.[5]

3.3 Dynamic Query

In dynamic queries the query is represented by a number of widgets such as sliders. A slider consists of a label, a field indicating its current value, a slider bar with a drag box, and a value at each end of the slider bar indicating minimum and maximum values. Sliding the drag box with the mouse changes the slider value.[6] Sliders are useful for filtering numeric variables. For categorical variables, dropdown boxes are useful in order to select which categories should be involved in the current query. In order to filter for dates intervals, calendar-styled elements are effective solutions, where the start date and end date can be selected. The combination of a graphical query and graphical output matches well the ideas of direct manipulation.

3.4 Clustering

Cluster visualization renders data clustered on an interactive map allowing to see a quick overview of similar items, where similarity means spatially close events are grouped to the same cluster in the current context. Additionally, clustering helps to render large amount of data efficiently on the map without exceeding the computational resources. By zooming in and out, the clusters were changed by drilling down to further cluster from super-cluster or rolling up where several clusters are grouped to super cluster on the visualization.

4 Seismic data extraction

4.1 Amplitude normalization

To normalize audio is to change its overall volume by a fixed amount to reach a target level. It is different from compression that changes volume over time in varying amounts. It does not affect dynamics like compression, and ideally does not change the sound in any way other than purely changing its volume. [7]

In order to sonify the earthquake events, loudness normalization methodology which adjusts the recording based its perceived loudness will be applied, namely the RMS (Root Mean Square) normalization. RMS normalization is the methodology, where the perceived loudness level is determined using the root mean square of the signal. The result is then used to compute the gain value used in the normalization. Since the gain value is constant and applied across the entire recording, the normalization does not affect the signal-to-noise ratio and the relative dynamics.[8] [9] In order to apply the normalization, the original signal has been multiplied with scaling factor a. The calculation of such scaling factor has been calculated:

$$y[n] = \sqrt{\frac{N \cdot 10\left(\frac{r}{20}\right)}{\sum_{i=0}^{N-1} x^2[i]}} \cdot x[n] \tag{1}$$

where r denotes the desired RMS level which has to be reached in decibel (Decibels relative to full scale, dBFS), N is the length of the input signal $x[n]$. The maximum possible RMS value is **0 dBFS**, which is equivalent to the amplitude of 1 on the normalized scale, and the possible maximum volume of the signal.

4.2 Spectrogram representation

A spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, but one can also see how energy levels vary over time. In other sciences spectrograms are commonly used to display frequencies of sound waves produced by humans, machinery, animals, whales, jets, etc., as recorded by microphones. In the seismic world, spectrograms are increasingly being used to look at frequency content of continuous signals recorded by individual or groups of seismometers to help distinguish and characterize different types of earthquakes or other vibrations in the earth.

Spectrograms are basically two-dimensional graphs, with a third dimension represented by colors. The horizontal axis represents the time dimension, the vertical represents the frequencies occur in each time frame, which can also be thought of as pitch or tone, with the lowest frequencies at the bottom and the highest frequencies at the top. The amplitude (or energy or "loudness") of a particular frequency at a particular time is represented by the colormap. [10]

5 Proposed solution

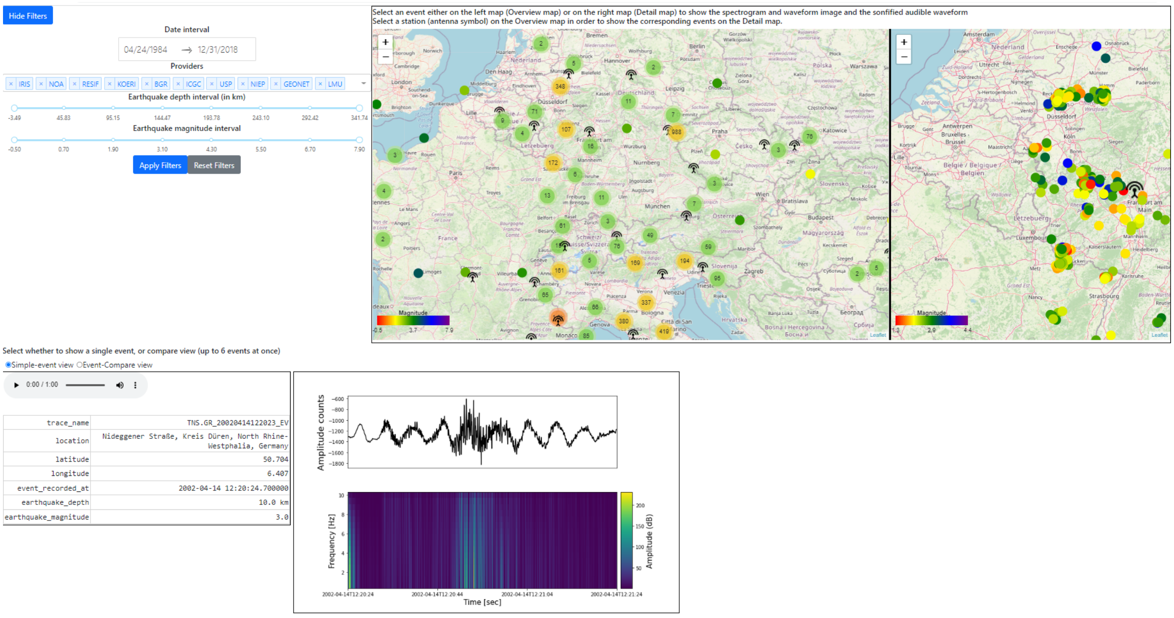

Figure 1: Proposed dashboard

5.1 Dataset description

The metadata provided by the STanford EArthquake Dataset [1] consist information regarding earthquake events, as well as the corresponding receiver station. In case of earthquake events, the project-relevant variables are the epicenters of earthquakes (**source_latitude** and **source_longitude**) are given in units of latitude and longitude, The depth of earthquake event (**source_depth_km**) relative to the mean sea-level, or the average elevation of the seismic stations that provided arrival-time data for the earthquake location. Earthquake origin times (**source_ origin_time**) when an earthquake began to rupture, have been estimated by seismic networks using earthquake location methods based on observed phase arrival times at multiple stations. Magnitude (**source_magnitude**) is approximately related to the released seismic energy and provides an estimate of the relative size or strength of an earthquake. There are different methods (scales) for measuring the magnitude. The data set contains seismograms associated with a wide range of earthquake sizes from magnitude -0.5 to magnitude 7.9. **source_id** is a unique identification number provided by monitoring network that can be used to retrieve the waveforms and metadata.[1]

Regarding the receiver station, variables **network_code**,**receiver_code**, **receiver_latitude**, **receiver_longitude** are relevant in order to determine the stations.[1]

In order to represent the receiver stations as well on maps, based on the STEAD metadata, and by the extraction of further information through ObsPy [11] framework, a dataset has been constructed which include the individual receiver stations, their location determined by longitude and latitude, and the corresponding provider service, like Incorporated Research Institutions for Seismology (**IRIS**).

5.2 Visualization of events

The earthquake events are shown in maps. The maps are provided via the Dash Leaflet [12] framework, the points are plotted as geojson instances as colored circle-markers, which color-coding belongs to the magnitude value of the recorded events. Additionally, the recording stations are also plotted with antenna symbol.

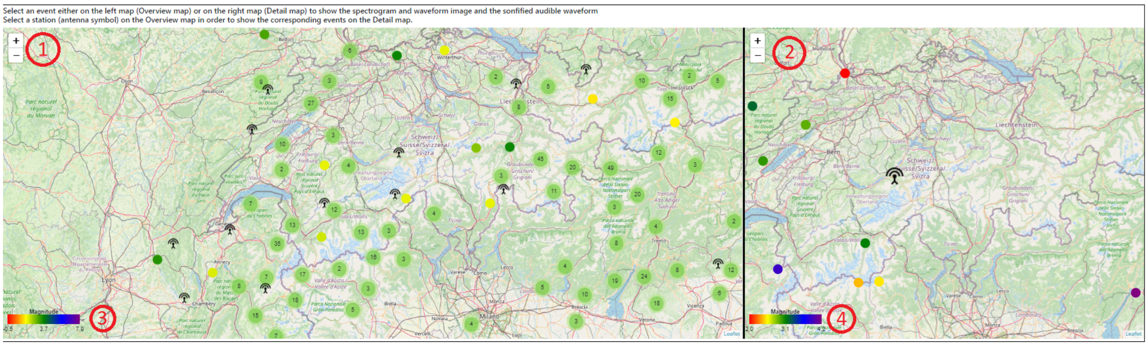

Figure 2: Overview and detail maps

The map visualization has been split up to two components, an **Overview** map (marked with **1**) and a **Detail** map (marked with **2**): if a station has been selected on the **Overview** map, the earthquake events which belongs to the selected station will be focused on the **Detail** map. Furthermore, the colors of the events on the **Detail** map will be re-scaled in order to distinguish the difference within them. The color-bars are marked with **3** and **4** upon.

5.3 Filtering

The visualized events can be filtered based on several attributes in order to focus on specific events.

As the upon screenshot shows, the filtering interface can be shown and hide by triggering the 'Hide Filters/Show filters button' (**1**) in order to make appropriate space for the map and the corresponding event detail representations. The events can be filtered by the date range of event recording (**2**), as well as the provider of the event (belongs to the station) with dropdown box by selecting them such as categorical variables, marked with **3**, and the sliders which filter of earthquake depth (**4**) and magnitude (**5**) to be represented. The application of set filtering and the default settings reset can be done with the corresponding buttons, denoted with **6**.

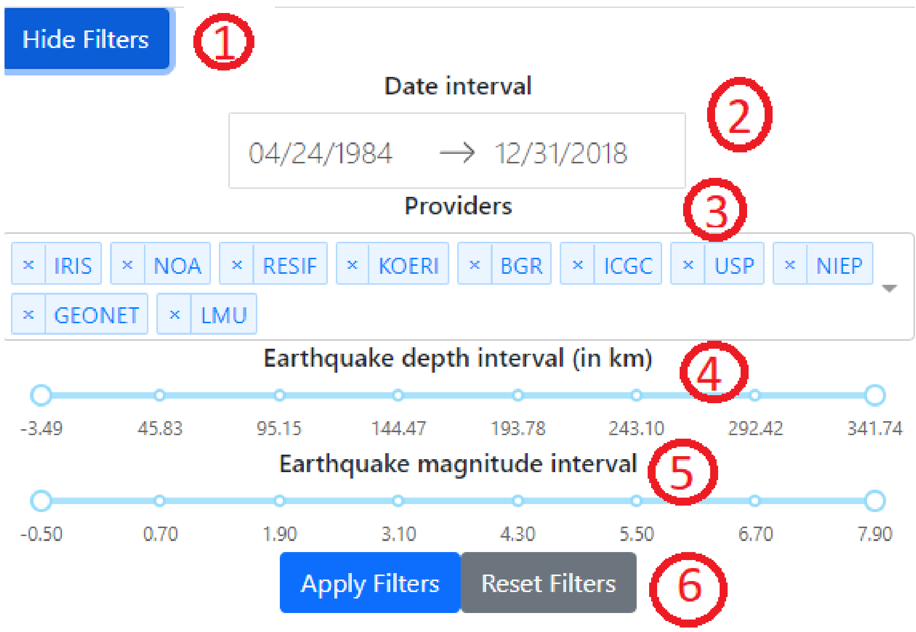

Figure 3: Filtering options

5.4 Waveform visualization and sonification

The events selected on the map were shown in waveform and spectrogram representation, and sonified and playable in audible form

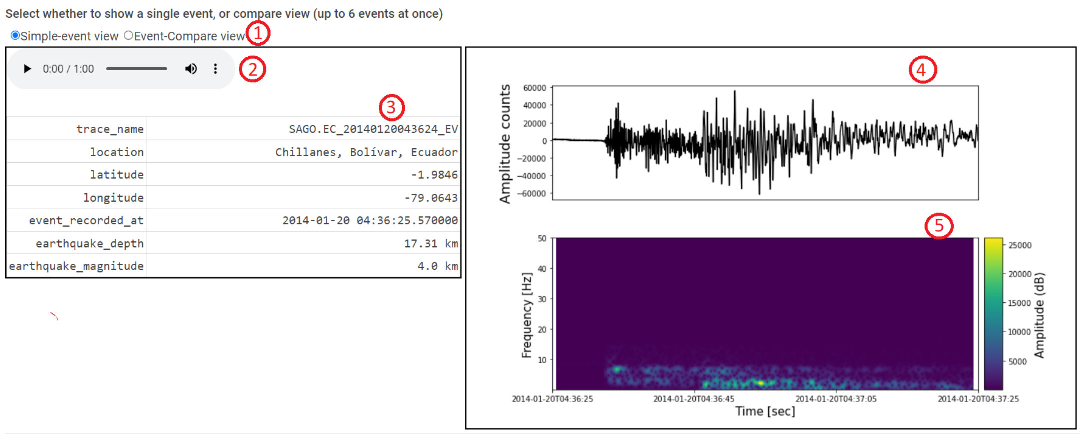

Figure 4: Waveform visualization and sonification component

The radio buttons provides a selection within single-event and comparison view (marked with **1**). The comparison view is able to show six events at once The high-pass filtered and sonified/amplitude-normalized waveform is playable via the audio player component marked with **2**, alongside the details of the event shown in table representation (**3**), where the approximated country and state extracted based on longitude and latitude (in case of earthquake below water, or other non-determinable location, N/A will be shown). The earthquake were represented in waveform (**4**) and spectrogram view, with colormap legend in order to interpret the amplitude measures (**5**) as well. The comparison view consist up to six events in paralell. Additionally, a button has been implemented to clear the view.

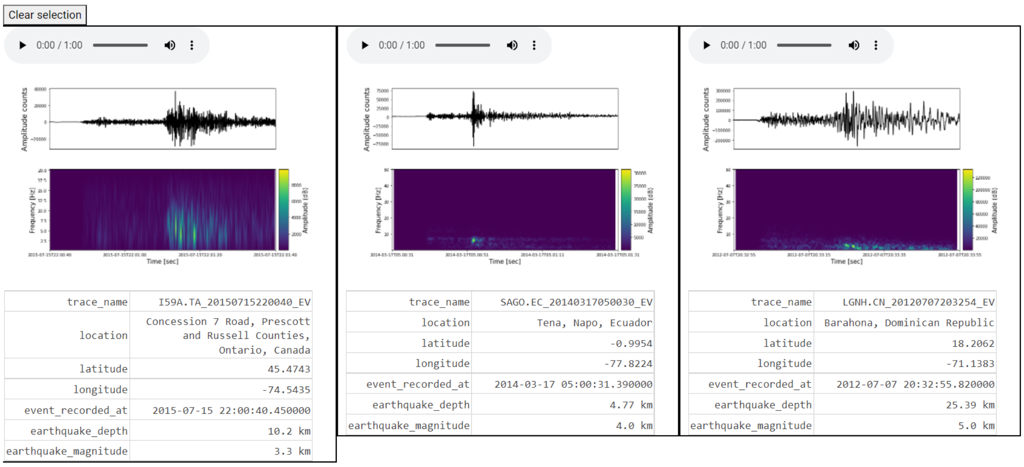

Figure 5: Waveform comparison

6 Implementation

The visualization dashboard is based on plotly Dash [13]. The metadata files has been constructed with pandas framework [14] and also the dynamic query interaction elements are connected with the dataframe-style managed objects in order to filter and manage efficiently. The map visualization is based on dash leaflet [12], where the events are handled as geojson components, which provide wide scale of manipulation opportunities, such as clustering and filtering these events. Additionally, the leaflet-defined methodologies assure to convert that the records of the dataframe can be handled in geojson components by converting these records to geopandas records. The extraction of waveforms from the providers are developed by the solutions of ObsPy [11] framework. The representation of waveforms are processed by matplotlib [15] framework, which provides efficient solutions to plot the waveform and spectrogram representations, The implementation of waveform sonification were based on numpy framework [16], which provides opportunity to manage the waveforms as numeric arrays and apply the proper mathematical methods to normalize the signals to be audible for human ears. Additionally, librosa framework [17] used to resample the normalized waveform, which was necessary to make the waveform audible by the dashboard's audio component. The resampled waveforms are stored in byte format and read directly by the Dashboard's appropriate component, which helps to avoid the bottlenecks of Dash framework, namely in case of changes in files (which would be the case if new event would be sonificated, then would be read from an updated audio files), the dashboard has to be re-loaded.

7 Evaluation

The implemented Dashboard is able to represent more than 1 million events at once in clustered view. The booting time of the dashboard might be a bit slow ($\tilde{6}5$ seconds, including the loading of the dashboard in browser), however, with the large number of scaling and clustering, and data loading ($\tilde{4}00$ MB of metadata), the booting time still can be considered as acceptable. Regarding the extraction of waveform events, if such event is available by the corresponding provider, in align with stable network connection, the construction of waveform and spectrogram plots, as well as the sonification of the waveform and the extraction of the event related information to table format takes approximately 10 seconds. Regarding the Dashboards bottlenecks and disadvantages could be named that due to the waveforms and informations are requested from online provider, stable and continous internet connection needed, and if the provider has been requested too many times in a small timeframe, request denial problems might occur. However that requesting methodology is a must since the local version of waveforms of the corresponding events might take approximately 80 GB of storage space, which could be infeasible by the storage or the computation perspective as well, even if only one record has to be loaded into the application at once.

Comparing to the Seissound application, the advantage might be that all the applied frameworks and programming environments are open-source and free-of-charge (Python), while the Seissound has dependencies with MATLAB, which is only free with specific licenses. The disadvantage comparing to Seissound, that the lack of video product, which is following the current audio playback moment on the spectrogram and waveform plots with a vertical line, which helps to interpret better the current sonified moment visually as well.

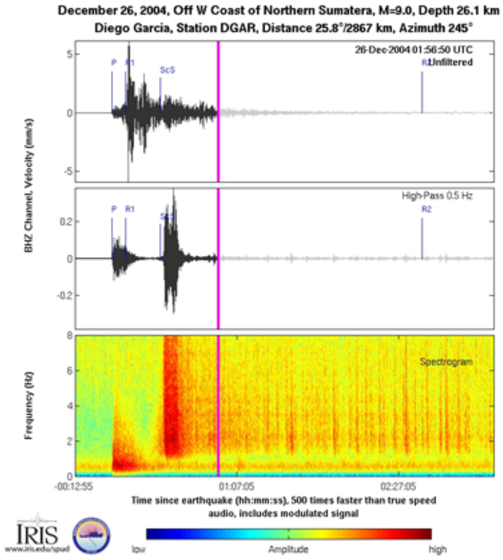

Figure 6: Seissound sample image source

Comparing to **Alberta Earthquake Dashboard**, the advantage is events availability also in the context of location availability, since the proposed dashboard represent events world-wide, as well as in in the context of amount. Additionally, the proposed dashboard includes the sonified audio content for (most) the events. The disadvantage comparing to the upon State-Of-The-Art approach that the statistics and representation at the Alberta Earthquake Dashboard is much more detailed, also in the context of individual events, namely the waveform representation can be examined in details by highlighting specific time frames, as well as in the context of summary statistics, since the State-Of-The-Art approach consist descriptive statistics not just for waveform events, such as grouped barplots to summarize the amount of waveform for specific magnitude ranges, but for individual stations as well. Additionally, the Alberta Earthquake Dashboard is processing real time data, not just with events from the past.

8 Conclusion

The developed Dashboard is able to provide an appropriate solution for earthquake event representation and lower level analysis, also in visual or audible context, with the combination of information visualization techniques in alignment with audio and waveform processing methodologies. The final product might be useful for educational purposes in order to gain knowledge of seismic data, since the graphical interface can be relatively easily learn-able, and able to provide most of the useful information in one screen without additional user interactions such as horizontal scrolling or keyboard commands. Comparing to the State-of-the-art solutions, the main advantage is that each components of the solution is developed with open-available frameworks, and the extracted and composed dataset's source are publicly available as well

9 Future Scope

The current solution only able to deal with events within 1984 and 2018, therefore recent earthquake events cannot be analyzed on the dashboards. The ObsPy [11] framework provides event request methodologies, which might be able to query recent events as well. However, based on the current experiments in order to extract recent events, comparing to the local extraction were time-consuming and would make the dashboard's loading/filtering procedure multiple times slower. However, that would be an interesting approach for further development of the current solution.

10 Corresponding lectures

From the domain-specific lectures in the curriculum, the relevant lectures are **Data Retrieval in Earth Observation" (120.034 and 120.035)**.

From the curriculum of the studies, the mostly affected area is module **Visual Analytics and Semantic Technologies**, and lectures namely **Information Visualization** and **Visualization 2**. Additionally, due to the audio-processing task of the project, lecture **Intelligent Audio and Music Analysis** is partially affected as well.

References

[1] S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza, "Stanford earthquake dataset (stead): A global data set of seismic signals for ai," *IEEE Access*, 2019.

[2] D. Kilb, Z. Peng, D. Simpson, A. Michael, M. Fisher, and D. Rohrlick, "Listen, watch, learn: Seissound video products," *Seismological Research Letters*, vol. 83, no. 2, pp. 281–286, 2012.

[3] "Alberta earthquake dashboard." https://ags.aer.ca/publication/iam-005. Accessed: 2022-04-24.

[4] R. Kosara, S. Miksch, and H. Hauser, "Focus+ context taken literally," *IEEE Computer Graphics and Applications*, vol. 22, no. 1, pp. 22–29, 2002.

[5] S. Burigat and L. Chittaro, "On the effectiveness of overview+ detail visualization on mobile devices," *Personal and ubiquitous computing*, vol. 17, no. 2, pp. 371–385, 2013.

[6] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 619–626, 1992.

[7] "Audio normalization." https://www.hackaudio.com/digital-signal-processing/amplitude/rms-normalization/. Accessed: 2022-04-25.

[8] "Root mean square normalization in python." https://superkogito.github.io/blog/2020/04/30/rms_normalization.html. Accessed: 2022-04-25.

[9] "Rms normalization." https://www.hackaudio.com/digital-signal-processing/amplitude/rms-normalization/. Accessed: 2022-04-25.

[10] "What is a spectrogram." https://pnsn.org/spectrograms/what-is-a-spectrogram. Accessed: 2022-04-25.

[11] T. O. D. Team, "Obspy 1.3.0," Mar. 2022.

[12] "Plotly dash." https://github.com/thedirtyfew/dash-leaflet. [Online; accessed 2022-02-23].

[13] "Plotly dash." https://plotly.com/dash/. [Online; accessed 2022-02-23].

[14] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[15] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[17] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, A. Weiss, D. Hereñú, F.-R. Stöter, P. Friesch, M. Vollrath, T. Kim, and Thassilo, "librosa/librosa: 0.9.1," Feb. 2022.