Draft version October 13, 2024
Typeset using LaTeX **twocolumn** style in AASTeX63

Automatic Detection of Ocean Forcing Events in Antarctic Ice Shelves

Linnea M. Wolniewicz,[1,2] Dhiman Mondal,[2] Pedro Elosegui,[2] John Barrett,[2] and Chester Ruszczyk[2]

[1] *Department of Astrophysical and Planetary Sciences, University of Colorado, 2000 Colorado Ave, Boulder, CO 80305, USA*
[2] *Haystack Observatory, Massachusett's Institute for Technology, 99 Millstone Rd, Westford, MA 01886, USA*

ABSTRACT

Antarctic ice shelves buttress glaciers on land and restrain ice from sliding into the ocean. Due to climate change, the ocean and atmosphere surrounding these ice shelves have warmed and as a consequence they are less stable and more susceptible to ocean forcing. Seismic stations positioned on the Ross Ice Shelf (RIS) have recorded the motion of the ice shelf caused by the ocean forcing events, such as flexural-gravity and lamb waves. These waves can trigger calving events and ice shelf collapse. Detecting these ocean wave events is important to understanding ice shelf stability, however only a few events have been documented in the literature. In this REU project, I have developed a software package which can automatically detect ocean wave events in seismic data collected on the RIS in Antarctica. I have applied a Gaussian Mixture Model (GMM) which clusters seismic spectrograms to detect high-power ocean wave events. I also use OpenCV's `fitEllipse` function to identify dispersive wave signals in the clustered spectrograms. This software package can be applied to future seismic data to create a labelled dataset for training a Convolutional Neural Network (CNN) to classify ocean wave events impacting the RIS.

1. INTRODUCTION

Antarctic ice shelves play an important role in buttressing glaciers and preserving the stability of the entire Antarctic ice sheet. Volume loss of ice shelves reduces buttressing, which raises concerns of glacier discharge into the ocean and a subsequent rise in sea level (Bromirski et al. 2015; Chen et al. 2018). Ice shelves are vulnerable to the warming sea and atmosphere as they extend from the land mass over the ocean, as shown in Figure 1. Ocean gravity waves and storms influence the ice shelf by providing stress perturbations and inducing flexural gravity and lamb waves within the shelves (Bromirski et al. 2010; Bromirski & Stephen 2012; Bromirski et al. 2017). The effect of ocean forcing events on the ice shelves can lead to calving events in icebergs as well as trigger ice shelf collapse events which lead to a subsequent rise in the global sea level (Macayeal et al. 2009). Due to the vulnerability of ice shelves with respect to ocean forcing, the detection of ocean wave events is pertinent to our understanding of ice shelf stability.

The three ocean wave events which induce the highest response in the Ross Ice Shelf (RIS) can be differentiated by the frequencies at which they propagate. Swell waves are generated by storm systems through the coupling of wind energy and ocean swell and propagate between $0.03 - 0.1$ Hz. Infragravity (IG) waves are generated by the superposition of ocean gravity waves impacting nonlinear coastlines and propagate at frequencies between $0.003 - 0.02$ Hz. Finally, VLP waves are induced by storm-systems such as tsunamis and propagate at frequencies less than 0.003 Hz (Bromirski et al. 2017). These ocean wave events are visually detected by researchers analyzing seismic data. However, to gain a deeper understanding of the number of ocean wave events impacting the RIS, it is necessary to detect a larger number of events than can be done by hand. These ocean wave events are often difficult to detect as some of those such as the infragravity and VLP waves propagate at noisy frequencies. Additionally, ice-quakes and other non-events induce high amplitude responses

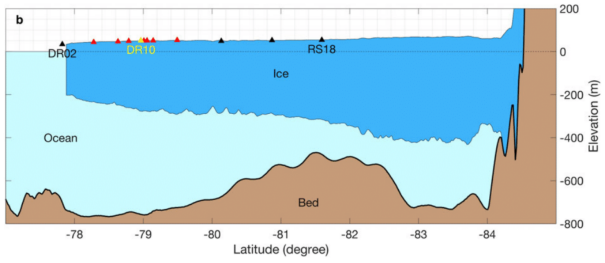

Figure 1. Cross-section view along the 180° meridian of the Ross Ice Shelf. Triangles represent seismic stations already positioned upon the RIS, with red representing the center sub-array of stations centered around station DR10. DR02 is off the ice front because the shelf has migrated Northward but the ice thickness model has not been updated (Modified after Chen et al. (2018))

in the ice shelves so simply thresholding seismic data is not enough to detect signals amidst the noise.

Machine Learning and Artificial Intelligence is a burgeoning field and it's applications to geophysics are vast, including applications for automatic detection and classification of seismic events (Yu & Ma 2020). Seismic stations positioned on the RIS have been collecting data since before 2000. Therefore, it is possible that the stations already recorded a number of ocean forcing events that can be utlilized to train a machine learning algorithm. However, this data is completely unlabelled as only a handful of ocean wave events impacting the RIS have been identified previous studies by Bromirski et al. (2010); Bromirski & Stephen (2012); Bromirski et al. (2015); Macayeal et al. (2009) and others. Thus to train and test a machine learning algorithm to automatically detect and classify ocean wave events, it is first necessary to prepare a processed and labelled dataset of events.

The Seismo-Geodetic Ice Penetrator (SGIP) instrument will monitor the response of the Ross Ice Shelf to ocean forcing. The SGIP, which is expected to be air-dropped into the RIS in 2022-2023, will send large amounts of seismic data to Haystack Observatory to be processed and analyzed. My research project this summer has been to develop a software package that can take in the seismic data collected by the SGIP and prepare it for a future machine learning algorithm that will automatically detect and classify ocean wave events in the data. Automatically detecting ocean wave events as they impact the RIS is important to understanding their stability and preventing a rise in the global sea level.

2. METHODOLOGY

Seismic stations positioned on the RIS (see Figure 1) have recorded seismic data which is available to the public through IRIS webservices[1].

In this research project, I analyzed existing seismic data from station DR10 positioned on the RIS. The data was downloaded using Obspy seismic data analysis package. The same package was used to remove the instrument response from the data and generate displacement timeseries. I have only used vertical component of the displacement from DR10 sampled at 1.0 Hz (channel LHZ). I analyze 100 days worth of data, beginning on November 21st of 2014 and ending on March 1st, 2015. This 100-day dataset is broken into 87 scrolling 10 day intervals (i.e. $set_1 = 11.21.2014 - 12.01.2014$, $set_2 = 11.22.2014 - 12.02.2014$, etc.). Each interval is converted into the frequency domain using the `Matplotlib.pyplot.specgram`[2] function which applies 1024 fast Fourier-transforms (FFTs) on time-series data to convert it into a power spectral density spectrogram in the dB scale. An example spectrogram is shown in Figure 2.

The raw spectrogram shown in Figure 2 is then normalized by subtracting the median power spectrum over the entire 10-day period. This removes much of the bright yellow noise located in the bottom of Figure 2 and makes the swell event propagating at ≈ 0.04 Hz between December 13th to December 16th. This normalized power spectrum is shown in Figure 3.

The normalized power spectrogram shown in figure 3 displays very clearly the wave event we are trying to detect. It is propagating at $0.03 - 0.06$ Hz between $2014 - 12 - 13$ to $2014 - 12 - 16$. This yellow signal is likely the result of a swell wave as it propagates above the IG wave threshold of 0.02 Hz. The goal of this software package is to separate this dispersive swell event (dispersive meaning the phase velocities of the wave differ from the group velocity) from the bright yellow vertical signals which are also present in the image. These vertical signals are non-dispersive and thus not due to ocean forcing.

Next, the normalized power spectrum of Figure 3 is used to train and test a Gaussian Mixture Model (GMM) which clusters it into 3 classes. A GMM performs much like the well-known K-means algorithm does; it takes in unlabelled data and classifies it into k-clusters in an unsupervised fashion. Unlike K-means, a GMM uses Gaussian distributions centered over cluster centroids to determine the probability that each datapoint belongs to a each cluster. Data is clustered based on it's highest probability. This allows a GMM to perform well with data that isn't circularly distributed about a centroid value. In the case of our spectrograms the GMM clusters pixels based on their power values. The GMM is trained on the first 43 images of the 87 image dataset and tested on the remaining 44. The resulting clustered Figure 3 is shown in Figure 4.

Next, the clustered image shown in Figure 4 is thresholded so only the signal cluster (bright yellow cluster) stands out from the noise. At this point in the process the methodology switches from signal processing to image processing so the time and frequency axes have been removed from Figure 5.

Finally, the thresholded image of Figure 5 is fed into the computer vision package OpenCV's `fitEllipse`[3]

[1] https://service.iris.edu/

[2] https://matplotlib.org/matplotlib.pyplot.specgram.html

[3] https://opencv.org/

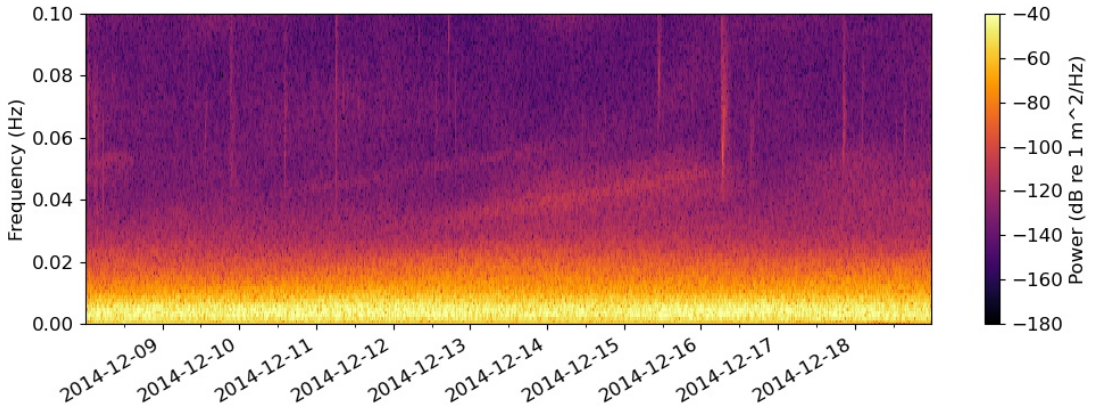

Figure 2. Spectrogram of displacement data collected at Station DR10 on the RIS from December 8th, 2014 to December 19th, 2014. Color corresponds the power in dB scale.

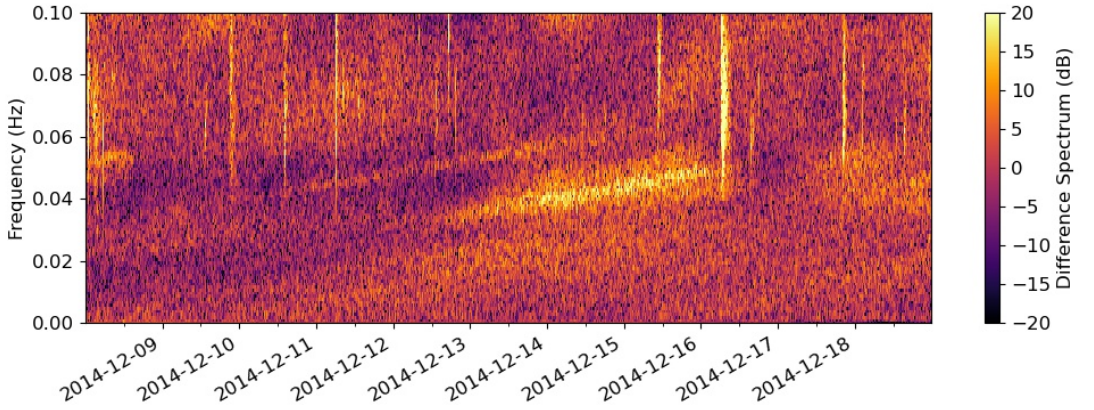

Figure 3. Spectrogram of normalized displacement data collected at Station DR10 on the RIS from December 8th, 2014 to December 19th, 2014. The median spectrum over the 10 day period has been subtracted, meaning the power is a difference spectrum without physical meaning.

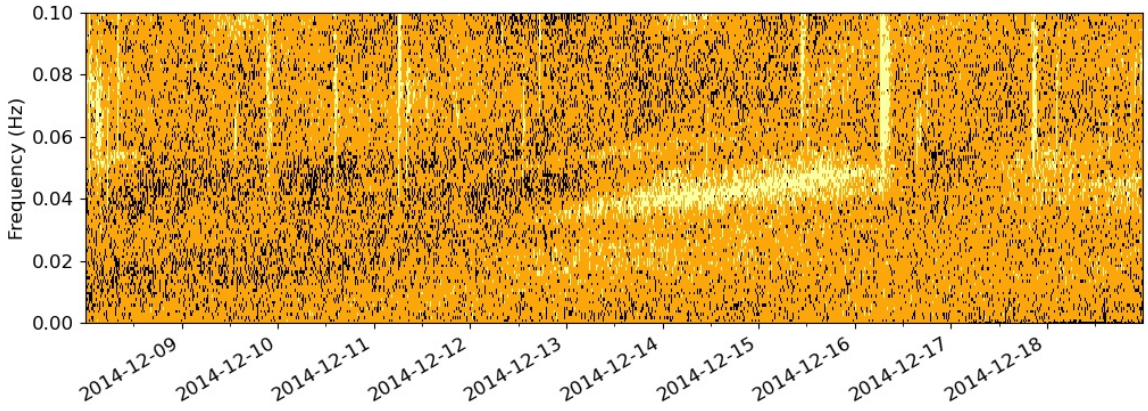

Figure 4. Clustered spectrogram of normalized displacement data collected at Station DR10 on the RIS from December 8th, 2014 to December 19th, 2014. The spectrogram is clustered using a GMM which identifies 3 clusters based on normalized power values.

function, which takes in images and fits ellipses to the contours within them. I have filtered out ellipses which

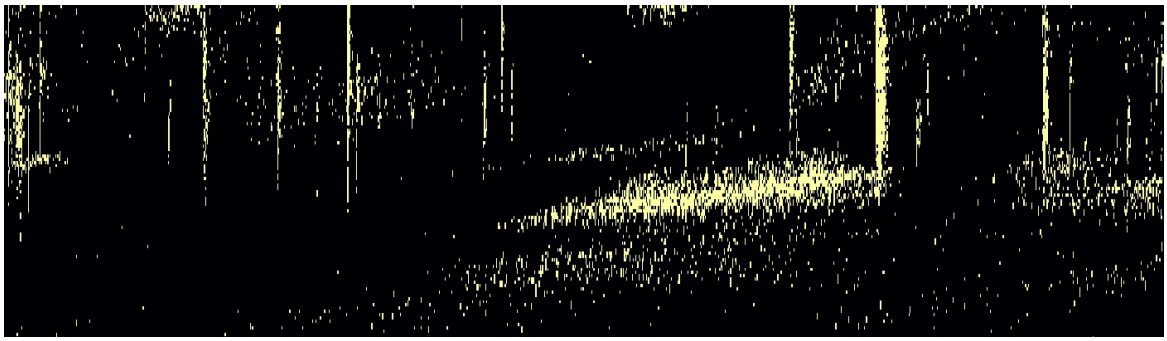

Figure 5. Thresholded and clustered spectrogram of normalized displacement data collected at Station DR10 on the RIS from December 8th, 2014 to December 19th, 2014. The image has been thresholded so events stand out clearly.

are too small to be the signal of a physical event or are vertically-oriented and thus non-dispersive.

The ellipses detected in figure A.4 define the swell event noted in previous figures well. OpenCV's `fitEllipse` function also removes non-dispersive vertical signals from the final image.

3. RESULTS AND DISCUSSION

My software package takes in raw seismic data and detects ocean wave events impacting the RIS using the methodology explained in Section 2. The package does not perform perfectly (see the Appendix for more examples) as it misses a handful events out of the 87 samples and classifies some ocean wave events using many ellipses as opposed to one. Despite this by my hand-counted number of times the software identifies true events and removes non-dispersive signals, I find it to be $\approx 90\%$ accurate based on the results listed in Table 1.

Table 1. Results (Hand-counted)

True Event Detections:	115
Accurate Noise Removals:	45
Missed Detections:	15
Missed Noise Removals:	4

Data is 87 10-day interval spectrograms spanning 100 days (11/21/2014 - 3/1/2015). Data is recorded at station DR10 on the RIS.

This software package can be fine-tuned to create a catalog of the start and end times of ocean wave events impacting Antarctic ice shelves. This catalog would not only be useful for understanding the number of events which impact the ice shelves annually, but will also be applied to create a dataset of labelled spectrograms upon which a CNN can be trained to automatically classify ocean wave events. CNN's are a sub-group of neural networks that perform well in image processing and with the aid of a labelled train and test set would perform well when classifying the ocean wave events that impact Antarctic ice shelves.

Automatically detecting and classifying ocean wave events is an important step in predicting the stability of Antarctic ice shelves, which buttress glaciers on land and are susceptible to ocean forcing. Automatic event detection saves researchers time, provides a count of ocean wave events impacting the ice shelves, and may allow for the discovery of new kinds of ocean wave events. We are excited to see how this summer project influences the push for machine learning in the study of Antarctic ice shelves as well as how a deeper understanding of ice shelf stability may affect humanity's reaction to climate change.

ACKNOWLEDGEMENTS

We would like to thank MIT's Haystack Observatory for hosting this research project and allowing it to move forward, as well as all the people involved at Haystack for offering their advice during this process.

L.M.W. would like to thank the National Science Foundation for funding the project under the REU grant given to MIT's Haystack Observatory. She would also like to thank MIT's Haystack Observatory for their kind hospitality during the REU project.

Thank you to the stations collecting data on the RIS which have allowed this work to be possible. All seismic data were downloaded through the IRIS Wilber 3 system (https://ds.iris.edu/wilber3/) or IRIS Web Services (https://service.iris.edu/).

Software: Matplotlib (Caswell et al. 2021), ObsyPy (Team 2020), SciKit-learn (Grisel et al. 2021), OpenCV (Sekachev et al. 2020), NumPy (Rougier et al. 2016)

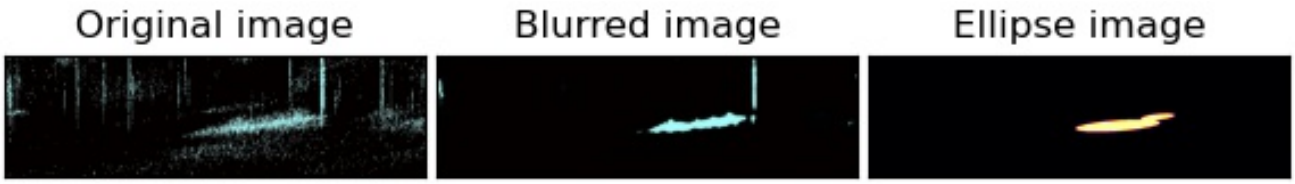

Figure 6. Result of detecting ellipses in the spectrogram of normalized displacement data collected at Station DR10 on the RIS from December 8th, 2014 to December 19th, 2014. The input, blurred, and resulting images are shown. The result image is created by running OpenCV's `fitEllipse` function on the blurred image.

REFERENCES

Bromirski, P. D., Diez, A., Gerstoft, P., et al. 2015, Geophys. Res. Lett., 42, 7589, doi: 10.1002/2015GL065284

Bromirski, P. D., Sergienko, O. V., & MacAyeal, D. R. 2010, Geophys. Res. Lett., 37, L02502, doi: 10.1029/2009GL041488

Bromirski, P. D., & Stephen, R. A. 2012, Annals of Glaciology, 53, 163, doi: 10.3189/2012AoG60A058

Bromirski, P. D., Chen, Z., Stephen, R. A., et al. 2017, Journal of Geophysical Research (Oceans), 122, 5786, doi: 10.1002/2017JC012913

Caswell, T. A., Droettboom, M., Lee, A., et al. 2021, matplotlib/matplotlib: REL: v3.4.3, v3.4.3, Zenodo, doi: 10.5281/zenodo.5194481

Chen, Z., Bromirski, P. D., Gerstoft, P., et al. 2018, Journal of Glaciology, 64, 730, doi: 10.1017/jog.2018.66

Grisel, O., Mueller, A., Lars, et al. 2021, scikit-learn/scikit-learn: scikit-learn 0.24.2, 0.24.2, Zenodo, doi: 10.5281/zenodo.4725836

Macayeal, D. R., Okal, E. A., Aster, R. C., & Bassis, J. N. 2009, Journal of Glaciology, 55, 193, doi: 10.3189/002214309788608679

Rougier, N. P., Hamrick, J. B., ibah, et al. 2016, numpy-100: Version 1.1, 1.1, Zenodo, doi: 10.5281/zenodo.61020

Sekachev, B., Manovich, N., Zhiltsov, M., et al. 2020, opencv/cvat: v1.1.0, v1.1.0, Zenodo, doi: 10.5281/zenodo.4009388

Team, T. O. D. 2020, ObsPy 1.2.1, 1.2.1, Zenodo, doi: 10.5281/zenodo.3706479

Yu, S., & Ma, J. 2020, arXiv e-prints, arXiv:2007.06183. https://arxiv.org/abs/2007.06183

APPENDIX

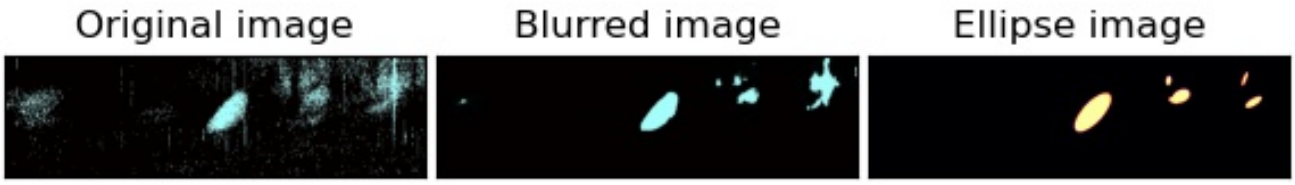

Figure A.1. Result of detecting ellipses in the spectrogram of normalized displacement data collected at Station DR10 on the RIS. The input image, the blurred image, and the result of OpenCV's `fitEllipse` function are showed.

Figure A.2. Result of detecting ellipses in the spectrogram of normalized displacement data collected at Station DR10 on the RIS. The input image, the blurred image, and the result of OpenCV's `fitEllipse` function are showed.

Figure A.3. Result of detecting ellipses in the spectrogram of normalized displacement data collected at Station DR10 on the RIS. The input image, the blurred image, and the result of OpenCV's `fitEllipse` function are showed.

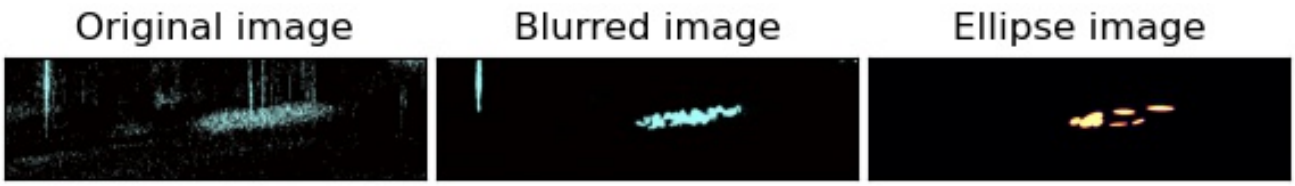

Figure A.4. Result of detecting ellipses in the spectrogram of normalized displacement data collected at Station DR10 on the RIS. The input image, the blurred image, and the result of OpenCV's `fitEllipse` function are showed.