INSECT IDENTIFICATION IN PULSED LIDAR IMAGES USING CHANGEPOINT DETECTION ALGORITHMS

Nathaniel Sweeney[1], *Caroline Xu*[2], *Joseph A. Shaw*[1,3], *Toby D. Hocking*[4], *Bradley M. Whitaker*[1,3]

[1]Electrical and Computer Engineering Department, Montana State University, USA
[2]Electrical Engineering and Computer Science Department, University of Michigan, USA
[3]Optical Technology Center, Montana State University, USA
[4]School of Informatics, Computing, and Cyber Systems, Northern Arizona University, USA

2023 Intermountain Engineering, Technology and Computing (IETC) | 979-8-3503-3590-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/IETC57902.2023.10152205

ABSTRACT

Noninvasive entomological insect monitoring often utilizes a variety of tools such as LiDAR to gather information without interfering with the insects in their habitat. These collection methods often result in large amounts of data that can be tedious and lengthy to interpret and analyze. Machine learning has been previously used in the past in order to analyze LiDAR images to detect insects, but often suffers from pitfalls such as long training times and large computational power requirements. In an attempt to offer an alternative that takes little to no training on the data and much less computational power, this paper looks at the use of changepoint detection algorithms to analyze LiDAR images containing insects. By analyzing the rows or columns of a LiDAR image, the algorithms should be able to detect abrupt changes in the image that would represent the insects. While not as accurate, the changepoint detection algorithms give comparable results to a machine learning algorithm tested on the same dataset without the need for supervised training.

***Index Terms*—** LiDAR, Changepoint Analysis, Anomaly Detection

1. INTRODUCTION

1.1. Background

Entomological methods for the monitoring of insects traditionally uses manual inspection techniques such as traps [1] or catching insects [2] in order to gather insects. These methods are often labor intensive, time-consuming, and disruptive to the insects, thus leading to the use of noninvasive insect monitoring with tools such as LiDAR [3]. These noninvasive methods can capture entomological information without disrupting the insects in their habitat, while also reducing the amount of manual labor required for collecting insect data. Previous work has been done using LiDAR to identify disease-carrying mosquitoes [4, 5] and characterizing agricultural pests [6, 7]. Other pulsed LiDAR methods have been developed specifically to detect the insect wingbeat modulations [8, 9, 10]. However, much of the analysis of the information from these techniques often requires large amounts of time and effort to manually analyze the information. Previous work has been done with the use of machine learning algorithms to help automate the analysis of LiDAR images and cut down on the amount of time it takes to analyze the data [11, 12]. While these have been shown to be successful, some of the many shortcomings of machine learning are often long training times of algorithms and large amounts of computational power required for the algorithms.

The contributions of this paper are to explore an alternative way to analyze LiDAR images that requires far less time and computational power than machine learning. This paper shows work done on two different datasets of insect-containing LiDAR images using two different changepoint detection algorithms, a graph-constrained changepoint detection algorithm, gfpop [13, 14], and MATLAB's `findchangepts` function.

1.2. Changepoint Detection

Changepoint detection algorithms are developed to detect abrupt changes in a variety of values such as mean, variance, or standard deviation. They are often utilized in fields such as medicine, neuroscience or genomics where it is necessary to find abrupt changes in large data sequences.While there are a large number of changepoint detection algorithms, the two that were used in this paper are gfpop (Graph Constrained Functional Pruning Optimal Partitioning) [13, 14] and MATLAB's `findchangepts` function.

The first algorithm that was used was gfpop. This changepoint detection algorithm was proposed by Hocking et al. [13] and later implemented into an R interface by Runge et al. [14]. In gfpop, input data is analyzed based on of a user-implemented graph with nodes to represent the various states that the data could be in at any given point and penalized edges representing the transitions between the various states.

This work was funded by the Air Force Research Laboratory (AFRL) on a subcontract from S2 Corporation under prime award No. FA8650-16-C-1954, with a fundamental research exemption.

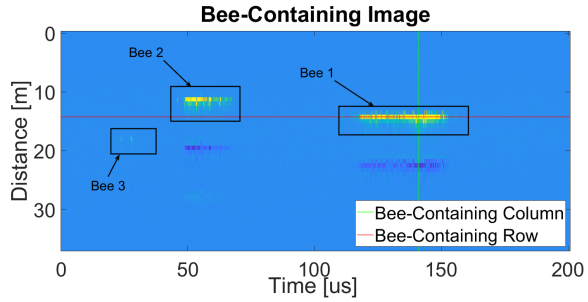

Fig. 1. Example LiDAR image containing three bees.

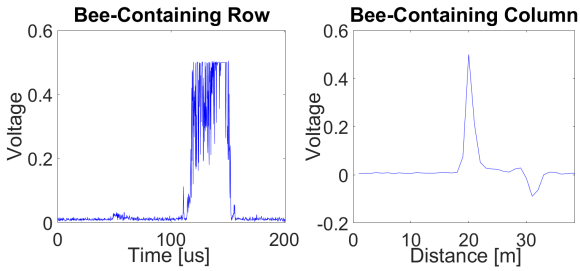

Fig. 2. Example of an insect row and column

The algorithm has a variety of loss functions that can be utilized as well as several different types of transition edges and other edge parameters such as the penalty, decay, threshold for biweight and Huber losses, and the slope for the Huber robust loss. While the work in this paper only utilized the penalty parameter, other applications are more likely to utilize these parameters. The reason for using this algorithm is due to the fact that the user could have a good idea of what the dataset should look like prior to collection, allowing them to quicky generate a graph and iterate through the dataset with little to no training and testing outside of tweaking the penalties on the edges.

MATLAB's `findchangepts` function is a part of the Signal Processing Toolbox and analyzes the input data to the function while returning the most significant changes in the data for several different types of statistics such as mean, RMS value, or standard deviation.

2. METHODS

2.1. Datasets

2.1.1. Beehives Dataset

The first dataset used for both gfpop and `findchangepts` is a set of LiDAR images taken from beehives located at the Horticulture Farms of Montana State University in Bozeman during June 2022. The dataset contains 4131 images (178x1024), 1309 of which contain a bee based on labels manually generated by five different people over the course of a combined 35 hours. Each bee also has a confidence level, ranging from 1 (less confident) to 5 (most confident). In these images, the bee is often represented by a spike in energy if the laser hit their body, or a series of smaller spikes in energy if the laser hit the wings of the bee (see Fig. 1). In the case of the latter, the frequency of the flutters of energy can show the harmonics of the oscillating wings. Both types of spikes can be seen in both the rows and columns (see Fig. 2).

2.1.2. Hyalite Creek Dataset

Both algorithms were further tested on previously taken LiDAR images of insects at Hyalite Creek outside of Bozeman, MT during September 2020. This dataset contains 10,079 images (178x1024) that represent similar info to the beehives. However, this dataset is far more sparse than the beehives dataset, with only 172 of the 10,079 images labeled as containing an insect, and with a large sum of those also labeled only as maybe an insect. Similarly to the beehives, these images were manually labeled over a long period of time that isn't exactly known. Testing was done on this dataset in order to allow a direct comparison to machine learning algorithms that were tested on the dataset by Vannoy et al. [11].

2.2. Preprocessing

For the gfpop algorithm, the only preprocessing that was done involved shifting any of the negative values present in the image, mostly artifacts from the LiDAR return, to zero. Each row or column, depending on which the algorithm was going to be applied to, was then smoothed with a three point moving mean window to reduce the open air noise and the noise at the top of the rows containing an insect.

For the `findchangepts` algorithm, additional preprocessing was performed. All of the rows that were guaranteed to not contain an insect were removed. This was done by splitting the row into eight windows and finding the average of each. If all of these were less than 90% of the average of the whole image, it was considered an empty row. Hard targets were then removed in a similar fashion, this time checking if each of the averages of the windows was greater than 0.8. This value was chosen because the images were normalized from 0 to 1. An additional threshold of 2.25*[Image Average] was then applied in a similar fashion. Noisy rows were then removed by dividing rows into 16 windows and finding their average. The percent difference between each section was found and if the difference was within 5% then the row was considered insignificant and removed. A wavelet transform is then performed on the remaining rows using the `cwt` function in order to distinguish between the remaining rows that contain wings of insects and other targets. If the maximum value that is returned is less than 0.4, the row is considered unimportant. The remaining scalograms are then split into 8 sections and the average of each rectangle is found. If the

percent different between them is less than 10%, then the row is removed. If there were no rows remaining in the image, it would be considered to have no insects.

2.3. gfpop

The main methodology of applying gfpop to the LiDAR images involved sending data through the algorithm with the changepoint graph shown in Figure 3. Both the rows and columns were sent through the algorithm to compare the results between the two, although the same graph and penalties were used. The selected graph was designed to allow gradual increases to the insect states, resulting in higher accuracy as opposed to a single jump to the insect state and then back down. To prevent the algorithms from continuously increasing the state value, the graph is also specified to start and end in the "air" state. Each state has a null edge on it which allows the dataset to remain in the state that it is in if the changes in the data aren't significant enough to trigger a state change. The increasing and decreasing states also both have an up edge transition and a down edge transition respectively, which is what lets the graph gradually increase to the peak, although this is mostly only relevant in the analysis of the rows since the insect columns are usually only three data points in the series. Lastly, the top of the graph which represents the actual insect has a penalty for decreasing back down from the peak.

After sending each row and column of the image through the algorithm, it saves the location, state, and parameter of the changepoints. After getting results back from the algorithm, the parameter of any insect changepoints were checked on whether they were double the mean of the row where the changepoint was detected and 1.5 times the mean of the overall image. This helped to remove any false positives that can be attributed to noise in the air or hard targets that were present in any of the images. Because the insect data was normalized, while the beehive dataset uses voltages, the penalty was set to 0.005 for the beehives dataset and 0.1 in the Hyalite Creek dataset. These values were determined after being trained on a handful of images.

2.4. MATLAB's `findchangepts`

For MATLAB's `findchangepts`, an algorithm using the function was developed for both the bees and insects datasets separately, mainly due to the fact that the insect dataset was far more sparse than the bee dataset. For the insect algorithm, the remaining rows from preprocessing were all put into the `cwt` function, and the absolute value was calculated for the 71x1024 matrix that resulted from it, along with the maximum value of the matrix. Since peaks in the dataset are desired, any value in the matrix below a threshold (.225 times the maximum value) was set to 0, and the matrix was cropped to contain only the 40 top rows (other rows correspond to lower frequencies that do not have wingbeat modulation data) and normalized between 0 and 1. The average of

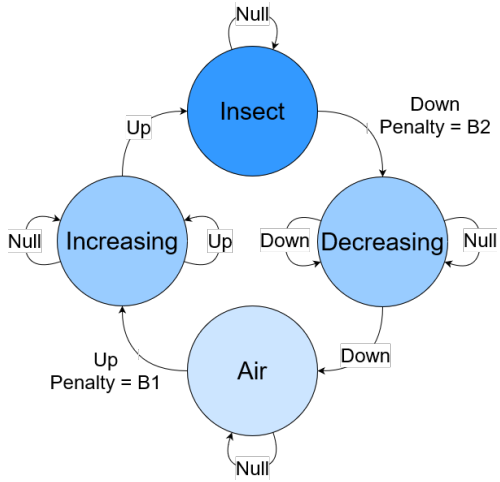

Fig. 3. gfpop graph for insect detection on both datasets.

the remaining columns was stored into a vector and the matrix was then smoothed and normalized again. This final vector was then sent through the `findchangepts` function. This analyzes the mean of the input with a minimum threshold of 3.9. The original signal is then smoothed and normalized and sent through the same function with a minimum threshold of 3. If no changes are found in this second result, there are no insects present in that row. If changepoints are found, then the changepoints of the scalogram and the original image are compared to each other. If they are within 21 pixels of each other, roughly a 2% difference, then the scalogram changepoint is kept, otherwise it is discarded. After this, the location of the insect is found by finding the brightest points in the scalogram, however this process only accommodates for two insects per row. The locations of these insects are then saved, although exact locations weren't verified, only the overall images. After these locations are found, they are discarded if they are over 605 pixels long or less than 15 pixels long. The remaining changes are considered insects.

The bee algorithm is similar, with the same preprocessing but with reduced thresholds since there were significantly more changes in the bee dataset. The first changepoint threshold was reduced from 3.9 to 2 and the second was reduced from 3 to .8. The maximum length that a bee signal could be was also increased from 605 pixels to 700. These rows were then grouped by insect and the middle column of the signal was found. It was then normalized and put into the `islocalmax` function with a minimum prominence of .6. If a peak was located at a row containing an insect signal it was considered an insect.

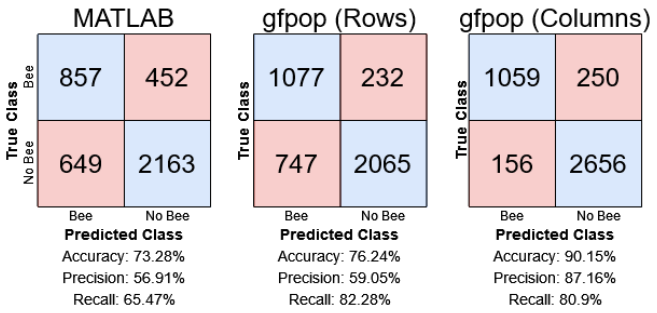

Fig. 4. Results from the beehives dataset.

3. RESULTS

The results of both algorithms were fairly comparable to one another, and can also be compared to some supervised machine learning results in the case of the Hyalite Creek insect dataset. The beehive results cannot be compared to any machine learning results due to the fact that no results from this dataset have been published. For the MATLAB algorithm, the results were found with a runtime of of 8066 seconds, resulting in a 73.26% accuracy. Of the 1309 insect-containing images, the algorithm detected 857 of them (65.47%). The algorithm also resulted in 649 false positives. For the gfpop algorithm on the rows, the results were compiled with a runtime of 1733 seconds, resulting in an accuracy of 76.24%. It was able to detect 1077 of the insect images, or 82.28%. There were 747 false positives. When gfpop was applied to the columns, it took a runtime of 2721 seconds, resulting in an accuracy of 90.15% and correctly identifying 1059 of the insect images, or 80.9%. It also resulted in 156 false positives.

For the Hyalite Creek dataset, the MATLAB algorithm was able to identify 122 of the 172 insect-containing images, or 71.1%, with a runtime of 14,474 seconds. It also incorrectly identified 196 images as insect-containing. For the gfpop algorithm on the rows, it properly identified 140 of the insect-containing images, or 81.4%, in a runtime of 4009 seconds with 1137 false positives. When it was applied to the columns, it found 106 of the insect-containing images with a runtime of 5906 seconds and 1840 false positives. In the case of the machine learning algorithms, specifically a neural network, the algorithm identified 32 of the 44 insect images that were in the testing set, or 72.7% respectively. When going over these results, accuracy was not considered due to the fact that the dataset was so sparse that simply guessing that there are no insects would result in an accuracy of 98%.

4. CONCLUSIONS AND FUTURE WORK

The results from the two algorithms were quite promising and were able to accomplish the main goals of reducing computational power and reduce the time it takes to analyze the

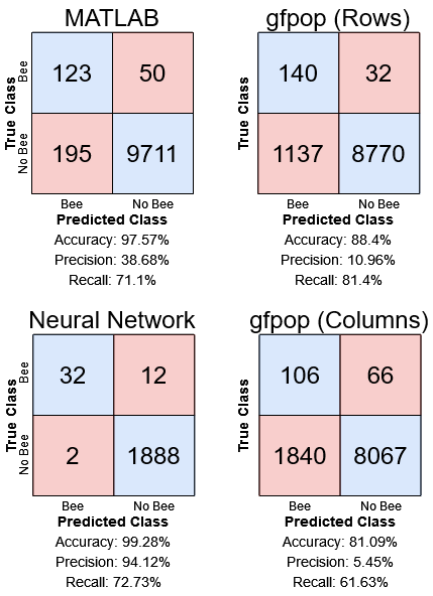

Fig. 5. Results from the Hyalite Creek dataset.

data, while also generating comparable results to the machine learning algorithms. The results between the two datasets were comparable to each other, although the Hyalite Creek dataset is far more sparse than the beehive dataset, which is why the accuracy is significantly differet. Compared to the 35 hours that it took to manually label the beehive dataset, and an even larger amount of time to label the Hyalite Creek dataset, the changepoint algorithms were able to cut down on the analysis time greatly. The analysis of the data could also identify individual insects as opposed to images.

Future work for this will focus on improving the results and attempting to use the algorithms as real-time detection methods when taking LiDAR images. For gfpop, further testing will involve making changes to the graphs and penalties that are used, while also applying more in depth post-processing to verify any of the rows or columns that were marked as containing a bee. For `findchangepts`, future work would involve improving preprocessing on the images and adjusting thresholds for the function. The algorithm can also benefit from using parallel processing in order to decrease the runtime of the algorithm as opposed to running it linearly. The algorithm could also be tested on the voltage matrices of the LiDAR images as opposed to the normalized data. This would result in more consistent values between each image for the air, hard targets, and bees. Since both algorithms are also implemented in MATLAB, testing could be done to use them as a real-time detection algorithm when taking LiDAR images. Future work will also consider developing penalty learning algorithms, rather than using fixed penalties, with gfpop [13].

5. REFERENCES

[1] Hitoshi Kawada, Sumihisa Honda, and Masahiro Takagi, "Comparative Laboratory Study on the Reaction of Aedes aegypti and Aedes albopictus to Different Attractive Cues in a Mosquito Trap ," *Journal of Medical Entomology*, vol. 44, no. 3, pp. 427–432, 05 2007.

[2] Nicole L. Achee, Laura Youngblood, Michael J. Bangs, James V. Lavery, and Stephanie James, "Considerations for the use of human participants in vector biology research: A tool for investigators and regulators," *Vector-Borne and Zoonotic Diseases*, vol. 15, no. 2, pp. 89–102, 2015, PMID: 25700039.

[3] Mahaveer Dwivedi, Malik Hashmat Shadab, and V. R. Santosh, *Insect Pest Detection, Migration and Monitoring Using Radar and LiDAR Systems*, pp. 61–76, Springer Singapore, Singapore, 2020.

[4] Gustavo E.A.P.A. Batista, Yuan Hao, Eamonn Keogh, and Agenor Mafra-Neto, "Towards automatic classification on flying insects using inexpensive sensors," in *2011 10th International Conference on Machine Learning and Applications and Workshops*, 2011, vol. 1, pp. 364–369.

[5] Samuel Jansson, Peggy Atkinson, Rickard Ignell, and Mikkel Brydegaard, "First polarimetric investigation of malaria mosquitoes as lidar targets," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 1, pp. 1–8, 2019.

[6] Y. Y. Li, H. Zhang, Z. Duan, M. Lian, G. Y. Zhao, X. H. Sun, J. D. Hu, L. N. Gao, H. Q. Feng, and S. Svanberg, "Optical characterization of agricultural pest insects: a methodological study in the spectral and time domains," *Applied Physics B*, vol. 122, no. 8, pp. 213, 2016.

[7] L. Mei, Z. G. Guan, H. J. Zhou, J. Lv, Z. R. Zhu, J. A. Cheng, F. J. Chen, C. Lofstedt, S. Svanberg, and G. Somesfalean, "Agricultural pest monitoring using fluorescence lidar techniques," *Applied Physics B*, vol. 106, no. 3, pp. 733–740, 2011.

[8] Joseph A Shaw, Nathan L Seldomridge, Dustin L Dunkle, Paul W Nugent, Lee H Spangler, Jerry J Bromenshenk, Colin B Henderson, James H Churnside, and James J Wilson, "Polarization lidar measurements of honey bees in flight for locating land mines," *Optics express*, vol. 13, no. 15, pp. 5853–5863, 2005.

[9] Kevin S Repasky, Joseph A Shaw, Ryan Scheppele, Christopher Melton, John L Carsten, and Lee H Spangler, "Optical detection of honeybees by use of wingbeat modulation of scattered laser light for locating explosives and land mines," *Applied optics*, vol. 45, no. 8, pp. 1839–1843, 2006.

[10] Joseph A Shaw, Kevin S Repasky, John L Carlsten, Lee H Spangler, and David S Hoffman, "Optical detection of oscillating targets using modulation of scattered laser light," Mar. 31 2009, US Patent 7,511,624.

[11] Trevor C. Vannoy, Trey P. Scofield, Joseph A. Shaw, Riley D. Logan, Bradley M. Whitaker, and Elizabeth M. Rehbein, "Detection of insects in class-imbalanced lidar field measurements," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1–6.

[12] Hannah B. Madsen, Trevor C. Vannoy, Elizabeth M. Rehbein, Riley D. Logan, Joseph A. Shaw, and Bradley M. Whitaker, "Automated insect recognition in unlabeled lidar field measurements," in *Laser Radar Technology and Applications XXVII*, Gary W. Kamerman, Lori A. Magruder, and Monte D. Turner, Eds. International Society for Optics and Photonics, 2022, vol. 12110, p. 1211007, SPIE.

[13] Toby Dylan Hocking, Guillem Rigaill, Paul Fearnhead, and Guillaume Bourque, "Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data," *J. Mach. Learn. Res.*, vol. 21, no. 1, jan 2020.

[14] Vincent Runge, Toby Dylan Hocking, Gaetano Romano, Fatemeh Afghah, Paul Fearnhead, and Guillem Rigaill, "gfpop: an R package for univariate graph-constrained change-point detection," 2020.