Naval Research Laboratory
Washington, DC 20375-5320

NRL/7340/MR—2023/3

Human Attuned Machine Learning Tool for Dune Annotation in ArcGIS Pro

NICHOLAS M. STUDER
BRADLEY LANDRENEAU
BENJI J. LEE
STEVEN M. DENNIS
CHRISTOPHER J. MICHAEL

Cognitive Geospatial Systems Section
Ocean Sciences Division

July 13, 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*: 13-07-2023

2. REPORT TYPE: NRL Memorandum Report

3. DATES COVERED *(From - To)*: 06/18/2021 – 03/01/2023

4. TITLE AND SUBTITLE

Human Attuned Machine Learning Tool for Dune Annotation in ArcGIS Pro

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nicholas M. Studer, Bradley Landreneau, Benji J. Lee, Steven M. Dennis, and Christopher J. Michael

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

1Y03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/7340/MR--2023/3

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Geological Survey
St. Petersburg Coastal Marine Science Center
600 4th St S, St. Petersburg, FL 33701

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Human-Attuned Machine Learning (HAML) Dune Annotation Tool (DAT) is a custom ArcGIS Pro plugin that facilitates quick, simple, and accurate annotation of coastal dune formations based on digital elevation models collected from LiDAR or other remote sensing methodologies. It implements an intuitive user interface consisting of basic lines and vertices which are first generated by a machine-learning backend and then adjusted or corrected by the user, if necessary, in an iterative process following the methods of previous HAML work. The intended purpose of the tool is to enable geospatial analysts to efficiently and accurately annotate coastal dune formations, so the tool is designed to follow the standard procedures for detecting dune crests and dune toes that have previously been used to conduct studies on the coastal response to storm surges and other adverse coastal phenomena. It utilizes several fail-safes and quality control checks on the output to minimize the amount of manual user corrections needed. This tool is part of ongoing research into the further optimization of geographic region annotation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	U	12	Nicholas Studer

19b. TELEPHONE NUMBER *(include area code)*: (985) 590-9413

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

Human Attuned Machine Learning Tool for Dune Annotation in ArcGIS Pro

Developed in 2022 by U.S. Naval Research Laboratory at Stennis Space Center, MS for sponsor:

U.S. Geological Survey
St. Petersburg Coastal Marine Science Center
600 4th St S
St. Petersburg, FL 33701

Contact:

- Brad Landreneau, NRL Code 7343
- Benji Lee, NRL Code 7343
- Nick Studer, NRL Code 7343
- Steven Dennis, NRL Code 7343
- Christopher Michael, NRL Code 7343

Preface

The Human Attuned Machine Learning (HAML) Dune Annotation Tool (DAT) is a custom ArcGIS Pro plugin which facilitates simple and efficient annotation of coastal dune formations. It implements an intuitive user interface consisting of simple lines and vertices which are first generated by a machine-learning backend and then adjusted or corrected by the user, if necessary, in an iterative process following the methods laid out in previous work[1].

The tool creates elevation profiles along the coastline of a user-provided GeoTIFF consisting of elevation data (DEM), typically collected via LiDAR or other remote sensing methodologies. These profiles are first analyzed internally by the tool to suggest the location of annotation points to the user. If the user approves the locations suggested to them, this data is then stored in a GeoDatabase along with other useful metadata for further scientific use. If the annotations need correction, a simple click-and-drag interface is available for the user to adjust them as necessary.

Because the intended purpose of the tool is to enable geospatial analysts to efficiently and accurately annotate coastal dune formations, the tool is designed to follow standard procedures for detecting dune crests and toes that have previously been used to conduct studies on the coastal response to storm surges and other adverse coastal phenomena[2][3]. It utilizes several failsafes and quality control checks on the suggested annotations to minimize the amount of manual user corrections needed.

This tool is part of ongoing research into the further optimization of geographic region annotation.

Table of Contents

Overview

The Dune Annotation Tool (DAT) is an ArcPro plugin that facilitates the labeling of dune profiles. Each profile has a dune crest (high point), dune toe (low point), and shoreline point (mean high water value). There is also support for berm formations, which is where the coastline is either flat or slopes very gently and thus does not have a toe.

DAT can be used with or without a machine learning backend that attempts to assist the user in placing the annotation points. It is enabled by default, but is easily disabled. When machine learning is disabled, a naive algorithmic method is used that will place the high point at the highest point on the profile and the low point at the location of maximum change in downward curvature (i.e., where the terrain levels out) w.r.t. the high point's location.

Adjusting each profile's high, low, or shoreline point is as simple as clicking and dragging the point in the mapview or profile view. Previous profiles may also be edited, even those created in previous sessions.

When profiles are saved, DAT will write the relevant annotation data to a geodatabase file for later use. See the geodatabase section for more details. These saved profiles can be edited in future runs of the tool via the History section in the HAML toolbar. See Editing Historical Profiles for more details.

The HAML Toolbar

The HAML toolbar (Fig 1) is where the user will interact with DAT. The functionality of each of these buttons is covered in more detail below.

The toolbar buttons will be enabled or disabled based on the current state of the application and/or tool. For example, the "Refine" button will not be clickable until the tool has been initialized. See the initialization section for further instructions on initializing DAT.

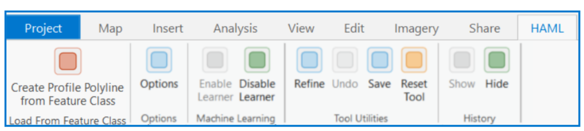

Fig 1: The DAT Toolbar in ArcPro

Button Descriptions

- **Create Profile Polyline from Feature Class**: launches the tool
- **Options**: opens the options menu
- Machine Learning
  - **Enable Learner**: turns on machine learning
  - **Disable Learner**: turns off machine learning
- Sketch Utilities
  - **Refine** (bound to K button): extends the shoreline and generates profiles
  - **Undo**: removes the last extension
  - **Save**: saves the current state of the plugin
  - **Reset Tool**: resets the MHW point, landward side, and starts the app from a "cold-start" state
- Historical Utilities
  - **Show**: shows historical profiles
  - **Hide**: hides historical profiles

Options Menu

Before using the tool, users should open the options menu (Fig 2) and configure the settings to match their needs. These include:

- **Output Geodatabase**: select the geodatabase file you wish to read from and write to

Manuscript approved July 11, 2023.

- **Transect Size**: toggles between using *Fixed* profile size (recommended) or a *MapView*-based profile size
- **High Annotation Placement**: toggle the method of finding the "high point" or crest of each dune, between a simple algorithm placement or ML placement. For more details, see the appendix.
- **Landward** and **Seaward Transect Lengths**: sets the size of profiles when **Transect Size** is set to *Fixed*.
- **Mean High Water (MHW) Point**: sets the height value where shoreline points will be created
- **Profile Spacing**: determines the distance between profiles when the user uses Refine to extend the shoreline
- **Color Options**: customizes the colors used for various tool elements

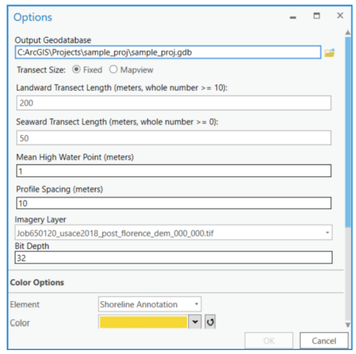

Fig 2: Options Menu

These values can be updated at any point while the annotation tool is active.

Initialization

The tool can only be initialized when two conditions are met:

1. Selection of the baseline feature class (via the MapView)
2. Selection of the raster containing the elevation data (via the Contents pane)

Once these have been selected, the user may then click the **Create Profile Polyline from Feature Class** button.

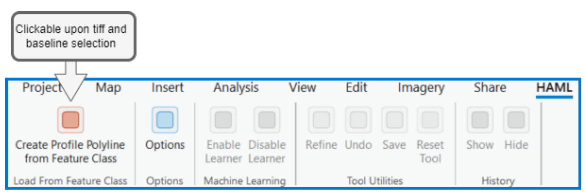

Fig 3: DAT Initialization

Once this button is clicked, the user will be prompted to provide two core pieces of information.

1. The user must **select the most seaward mean high water point** given the current MapView. This information is necessary to help find the appropriate shoreline point when two or more raster cells contain the MHW value for a given profile.
2. Then the user must **select the landward side** of the current MapView. This information helps with suggested placement of annotation points, particularly when there is significant land mass on both sides of the shoreline.

Warning: The initialization of the tool requires the MapView to comfortably contain all initialization points, including the mean high water point and the initial shoreline (dotted red line). Most failures to initialize are due to the MapView not enclosing the working area.

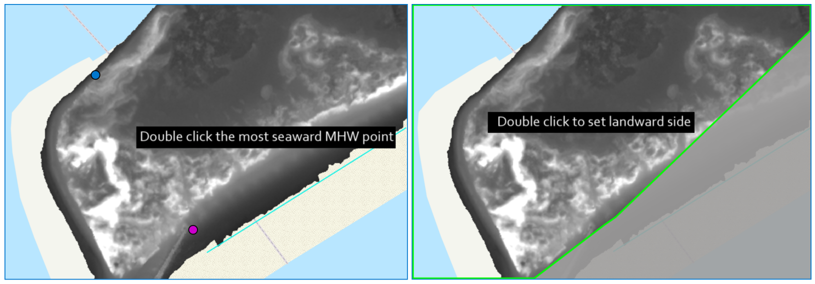

Fig 4: MHW Selection (left), Landward Select (right)

Once the MHW point and landward side are set, the shoreline is initialized (represented by the red dashed line). Now the tool will create the first profile in the tool using the algorithmic placement. The profile will then be displayed in the MapView and profile view (detailed in Fig.s 5 and 6).

In the MapView, the profile will be magenta, indicating that it is the selected profile while the high, low, and shoreline points will be green, red, and yellow, respectively. The user can then click and drag the high, low, or shoreline points to their appropriate location via the MapView or profile view.

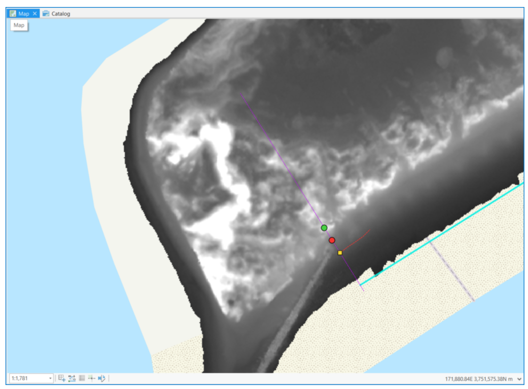

Fig 5: Initial Transect

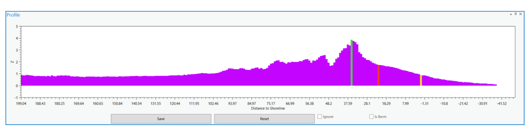

Fig 6: Profile View

Profile View

The profile view (Fig 6) displays the elevation data of the profile. The user can left click and drag each point of interest (high, low, shoreline) and adjust their positions as needed. Additional mouse controls are also supported:

- Right clicking and dragging with the mouse will pan the graph
- Scrolling up and down with the mouse wheel will zoom the x-axis of the graph
- Clicking the scroll wheel over a bar will display the height value at that point

Tip: The user may click and drag *near* a point to move it, they do not have to click exactly on the point.

The Profile View also contains several buttons:

- **Save**: saves the current profile and update its values in the geodatabase
- **Reset**: places the points of interest in the positions they were first displayed in
- **Ignore**: flag the profile in the geodatabase as *ignored* and will cause this profile to be slighty transparent
- **Berm** : removes the low point from a profile and flag the profile as a berm in the geodatabase.

Note: Ignored profiles and berms will not be used for training the machine learning model

Refining

Once the tool is initialized and the first profile is annotated, the user may click the **Refine** button in the HAML toolbar or press the 'Space' key to further extend the current shoreline and populate it with profiles. The selected profile will then change to the first profile of the newly extended shoreline and the profile view will update accordingly. The user can switch between profiles by either clicking on them in the MapView or pressing the 'D' or 'F' keys to move to the previous and next profile, respectively.

Tip: Annotation can be done entirely with the mouse by clicking to select profiles and then clicking to drag the annotation points, or the user can utilize the 'Space', 'F', and 'D', keys to Refine, select next profile, and select previous profile, respectively. In the profile view, the user can use 'J' and 'K' to move the currently active vertex (crest or toe), and the 'I' key will swap which vertex is active.

After the user adjusts the profiles accordingly, they continue refinement. Each refinement will extend the tool 50 meters, with a profile being placed in even intervals given by the *profile spacing* setting (found in options). With each refine, all new or modified profiles are saved to the geodatabase.

In the event of the refinement extending beyond the current mapview, the mapview will automatically pan to where the tool is extending.

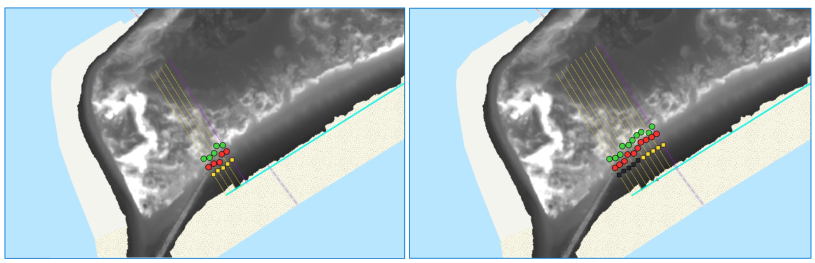

Fig 7: First refine (left), second refine (right)

Note that all profiles in Fig 7 that are unsaved have yellow shoreline points. Once a profile is saved to the geodatabase, its shoreline point will change color to black by default. Any time the user edits a profile after it has been saved, the profile's shoreline point will change back to its edited color.

Geodatabase Functionality

When the tool is activated, a Feature Dataset called *Profile Points* and a Feature Class called *Baseline Points* are created. The profile points feature dataset contains high, low, and shoreline point data. The baseline points feature class contains the MapPoints associated with each segment of the baseline.

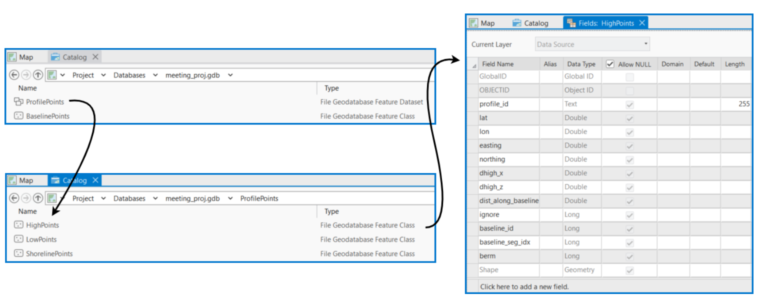

Fig 8: An example GDB entry for a high point annotation

As a user annotates and/or edits profiles, the geodatabase is automatically updated (and can be triggered to update by clicking a "Save" button). When a user stops using the plugin or closes ArcPro, when the tool is reactivated, annotations will continue where the user last left off.

Tip: The active GDB can be changed in the Options menu

Editing Historical Profiles

Editing profiles that were previously annotated and saved in a prior tool run can be done by activating the tool as usual, then clicking the "Show" button under the History section of the HAML toolbar. Any profiles that have been saved since the user last used ArcPro or the plugin will then be editable. Editing historical profiles works the same as editing current profiles. It is important to note that when the "Show" button is clicked or active, that *all historical profiles* in the mapview will be loaded. These historical profiles are dynamically loaded as the user pans or zooms in the current mapview.

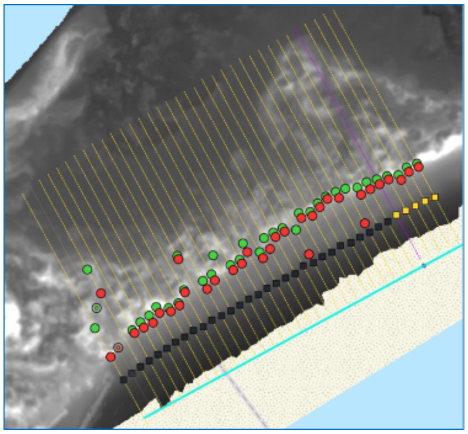

Fig 9: Displaying Historical Profiles. Note the black shoreline vertices that are not connected by a dotted red line; these are the historical profiles loaded from the GDB.

Reporting Issues

If you encounter any unexpected behavior, performance concerns, crashes, or any other problems, please send a message to the developers by email.

Please include the log files which can be found in the logs directory of your /ESRI/ArcGISPro/ installation.

Additional information which may be helpful:

- A screenshot of the issue occuring
- Geocoordinates of the location where the issue occured
- If possible, include the geotiff used

Acknowledgements

We acknowledge Alexander Seymour, Kara Doran, and the rest of the St. Petersberg, FL USGS team for their assistance with feedback, testing, and advice. Their input was significantly helpful in the development of this plugin.

Appendix: Algorithmic and Machine Learning Placements

Overview

When the tool is initialized, the machine-learning backend will be in "cold-start" conditions -- it will have no previous training data to work with. The tool will thus rely on algorithmic annotation placement (details below) until it has accumulated a reasonable amount of training data (a couple thousand raster points, which is roughly a few full profiles' worth). Once it has enough data, the Machine-Learning (ML) backend (details below) will be used instead. To avoid obvious errors, there is a system of quality-control checks (details below) that we run the ML output through; if any of these checks should fail, we revert to an algorithmic placement for the profile which failed a check.

To Summarize:

1. Begin in cold-start conditions, so rely on algorithmic placement
2. When enough training data is accumulated, use ML placement*
3. Perform quality-control checks on the ML output
4. If any quality-control check fails, revert to algorithmic placement for the profile which failed a check

The above steps are repeated when the user resets the tool.

*: By default, only the "low point" or toe will use ML placement. ML placement for the "high point" or crest can be enabled in the user Options menu.

Algorithmic Placement

Let a profile P be the set of points $(x, z(x))$ along a line perpendicular to the baseline with size $d = d_\mathrm{s} + d_\mathrm{l}$ where d_s and d_l are the seaward and landward length, respectively, set by the user. Also, let x be the distance from the shoreline point given by the user-specified MHW value x_MHW and $z(x)$ is the elevation curve of profile P.

Then, our method to find the crest, or "high point" x_high is given by Eq. (1):

$$x_\mathrm{high} = ^{\arg\max}_{x \in P} \; z(x) \tag{1}$$

In other words, x_high is simply the point of highest elevation in P.

The algorithm to detect x_low is quite a bit more complex. First, we restrict the possible domain of $x_\mathrm{low} \in S$, to be the points between x_MHW and x_high where the slope is positive, which is written more explicitly in Eq (2):

$$S = \{x | x \in (x_\mathrm{MHW}, x_\mathrm{high}) \;\; \mathrm{and} \;\; \frac{\partial}{\partial x} z(x) > 0\} \tag{2}$$

Then, x_low is given by Eq. (3):

$$x_\mathrm{low} = ^{\arg\max}_{x \in S} \frac{\partial^2}{\partial x^2} z(x) \tag{3}$$

Put more simply, x_low is the point between x_MHW and x_high where the second derivative is largest and the first derivative is positive. Even more simply, x_low is the point where the elevation profile slopes upward the fastest when moving from the MHW point to the crest. Fig 10 shows all of the terms used above on an example profile.

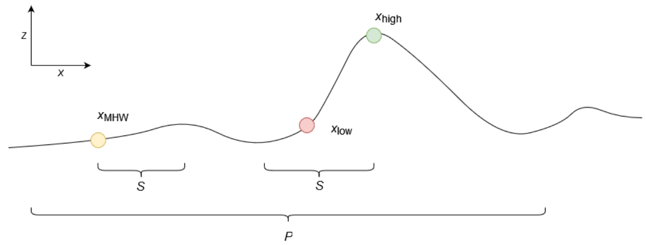

Fig 10: An example profile view with annotations

Machine Learning Placement

Training

We utilize a multiclass k-Nearest Neighbors (KNN) model for our machine learning classifier which is trained on feature vectors constructed from points on the elevation profile of the GeoTIFF when rasterized. Each feature vector contains the following data:

1. The elevation: $z(x_\mathrm{i})$
2. The first-derivative of the elevation profile:

$$\frac{\partial}{\partial x}(z(x_\mathrm{i}))$$

3. The second-derivative of the elevation profile:

$$\frac{\partial^2}{\partial x^2}(z(x_\mathrm{i}))$$

4. The second-derivative of the moving local average of the elevation profile:

$$\frac{\partial^2}{\partial x^2}(\mathrm{Mean3}(z(x_\mathrm{i})))$$

5. The second-derivative of the moving local standard deviation of the elevation profile:

$$\frac{\partial^2}{\partial x^2}(\mathrm{Stdev3}(z(x_\mathrm{i})))$$

where i is the index of the raster point along profile P, $\mathrm{Mean3}(z(x_\mathrm{i}))$ is the ± 3 window-size mean of z around x_i, and $\mathrm{Stdev3}(z(x_\mathrm{i}))$ is the ± 3 window-size standard deviation of z around x_i.

To train the learner, these feature vectors are inserted into a k-d tree which is regularly optimized every 2000 new data points for performance. Additionally, the size of the k-d tree is capped at 10,000 data points to reduce slowdown in performance over longer sessions. For this reason, it is encouraged to occasionally reset the learner as you move great distances along the shore, as this will increase performance and accuracy in most cases.

Note: Data in the feature vectors are min-max normalized to [0,1] on a per-profile basis. Additionally, NaN values in the underlying raster are never used for training or classification

Classification

Our KNN learner utilizes $k = 8$ nearest neighbors for the classification of each feature vector. The neighbors contribute weight based on Dual Inverse Distance Weighting[4], and those weights are passed to a Softmax function[5] to convert the weights to a list of measures analogous to class-membership probabilities. Then, the point is classified as the class with the highest class-membership probability.

Placement

In the dune annotation problem there are three possible classes for any point along the profile P:

1. Points "above" or more inland of x_{high}
2. Points between x_{low} and x_{high}
3. Points "below" or more seaward than x_{low}

Then, the annotations shall be placed by first classifying every point in P, then algorithmically determining the "boundaries" listed above.

Once we have classified each point in P, we utilize a *near-continuous* strategy for placing each annotation. For example, if we start from the MHW point and walk along P, the points near the MHW should be classified as class 3 from above. Then, x_{low} will be placed at the first point where there the next 3 consecutive classification are all of class 2.

Quality Control Checks

Depending on the quality of the training data as well as the geography of the land being annotated, ML can occasionally find it difficult to make accurate placements. In the event of ML making obvious or glaring mistakes, we have several quality-control checks on the output of ML. If any of these checks fail, we will fall back on algorithmic placement for that profile and that profile only.

Tip: If ML placement is making frequent errors, consider resetting the learner or disabling it altogether. Information about how the ML is failing can be found in the log files, and the GDB keeps track of which profiles were successfully placed via ML

The quality control checks currently in place are:

1. The **crest should not be below a certain percentage of the maximum elevation** in the profile. Sometimes a crest is not the highest point in the profile (in cases of more land-ward dunes which are taller, for example), but this check makes sure that the crest placement is still at relatively high elevation.
2. The **crest and toe should not be similar to each other in elevation**. If the ML attempts to place the crest and toe at similar elevations (relative to a percentage of the maximum elevation in the profile), it has either made a mistake or the profile is that of a berm.
3. **The MHW point, toe, and crest should always be in order** in both x and z. In other words, the toe should never be more landward or taller than the crest. The same is true for the MHW point and the toe.

Each of these events trigger a unique log event when its criteria is not satisfied and algorithmic placement is utilized instead.

References

1: Michael, C. J., Dennis, S. M., Maryan, C., Irving, S., & Palmsten, M. L., 2019, A General Framework for Human-Machine Digitization of Geographic Regions from Remotely Sensed Imagery, Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), 259-268, https://doi.org/10.1145/3347146.3359370

2: Stockdon, H.F., Doran, K. S., & Sallenger, A.H., 2009, Extraction of Lidar-Based Dune-Crest Elevations for Use in Examining the Vulnerability of Beaches to Inundation During Hurricanes, Journal of Coastal Research, 2009(10053), 59-65, https://doi.org/10.2112/SI53-007.1

3: Stockdon, H.F., Doran, K.J., Thompson, D.M., Sopkin, K.L., Plant, N.G., & Sallenger, A.H., 2012, National assessment of hurricane-induced coastal erosion hazards: Gulf of Mexico: U.S. Geological Survey Open-File Report 2012-1084, 51, https://pubs.usgs.gov/of/2012/1084/

4: Li, Z., Zhang, X., Zhu, R. et al. Integrating data-to-data correlation into inverse distance weighting. Comput Geosci 24, 203–216 (2020). https://doi.org/10.1007/s10596-019-09913-9

5: Wood, T., Softmax Function, deepai.org, https://deepai.org/machine-learning-glossary-and-terms/softmax-layer