Naval Research Laboratory
Washington, DC 20375-5320

NRL/7340/MR—2022/1

An Interactive Machine Learning Approach to 3D Geographical Region Annotation

NICHOLAS M. STUDER

STEVEN M. DENNIS

CHRIS J. MICHAEL

Center for Geospatial Sciences Branch
Ocean Sciences Division

January 19, 2022

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
19-01-2022	NRL Memorandum Report	

4. TITLE AND SUBTITLE

An Interactive Machine Learning Approach to 3D Geographical Region Annotation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
NISE

6. AUTHOR(S)

Nicholas M. Studer, Steven M. Dennis, and Chris J. Michael

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
6C39

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--2022/1

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

Naval Research Laboratory
1005 Balch Boulevard
Stennis Space Center, MS 39529

10. SPONSOR / MONITOR'S ACRONYM(S)
NRL-NISE

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

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

13. SUPPLEMENTARY NOTES
Karles Fellowship

14. ABSTRACT

This document presents the background, implementation detail, and demonstration of operation for the 3D extension of U.S. Naval Research Laboratory's proprietary Geographic Region Annotation Tool (GRAIT). This report serves to explain the need for and demonstrate a fully-featured 3D geographic region segmentation tool with accompanying visualization and analysis suite. Current and near-future features are also discussed and proposed.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Nicholas M. Studer
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	U	18	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(228) 688-4955

This page intentionally left blank.

TABLE OF CONTENTS

FIGURES

An Interactive Machine Learning Approach to 3D Geographical Region Annotation

1. INTRODUCTION

The study of geographical features is an inherently 3D problem. Heights, depths, slopes, and volumes are integral in studying such natural phenomena as coastal dune morphology and glaciology [2]. These quantities can be extracted from satellite imagery, IR- or LiDAR-based measurements, etc., where digital elevation models are also available; however, this requires either significant manual labor or reliance on error-prone fully-automated approaches that also require significant amounts of training data. By utilizing the human-in-the-loop approach of previous work [1], we can enable an analyst to work as a team with the machine from cold start (if necessary) to teach it which geographical features are important. In return, the machine will rapidly locate these features, segment them out, and create a full 3D visualization for the analyst.

In this report we describe a new interactive machine learning methodology to 3D geographic region segmentation which allows analysts to identify, locate, segment, and visualize 3D- geographical regions where imagery is available alongside a digital elevation model. We also demonstrate the use of the tool, its human-machine interface for segmentation, as well as the current state of its visualization features. We also discuss the current and future directions for this tool.

2. BACKGROUND

2.1 Binary Classification and Supervised Learning

Binary classification as applied to image segmentation involves every pixel of an image being labeled as either having full membership to the class, denoted as a positive label, or no membership to the class, denoted as a negative label. Binary classification is a subset of the *supervised* learning methods, which means that some amount of previously-labelled data is required in order to make predictions on future data. This predictive capability is given in the form of function estimation done by the classifier after being exposed to the training data. The *classifier* is the mapping from input data to label class, i.e., a pixel within an image and its associated color channel data will be predicted as belonging to the "positive" label class or the "negative" label class, based on previously-encountered data. In the way, we can make predictions about where in the image our important geographical features might lie. For more information on general supervised learning and classification, see reference [9].

2.2 *k* Nearest Neighbors

k nearest neighbors turns a labelled data point into a corresponding feature vector in *n* dimensional space, where *n* is the number of features in the data set. A feature is any degree of freedom in the data set, such as the "redness" or "height" associated with any given point in the feature space. Our tool operates on pixels in a raster, where each pixel has a value for at least four different features: red, green, blue, and height or depth.

The feature vectors in KNN form the classifier model: when a new, unknown pixel and its associated features are passed to the KNN classifier, it simply compares the new feature vector with all of its previously-labelled feature vectors, and performs Euclidean distance measurements. The *k* nearest feature vectors decide the label class of the new pixel, giving the KNN classifier its name.

In general, it is not a trivial task to locate the nearest members in feature space and use that information to make a mathematical prediction of the label. Our implementation employs an *uncertainty model* where

Manuscript approved January 18, 2022.

we obtain a quantity that reflects how strongly the classifier believes in its predictions. This quantity is normalized between zero and one, with zero corresponding to a prediction with no certainty whatsoever (essentially, a coin-toss) and one corresponding to a prediction with absolute certainty. Specifically, the uncertainty $U_a(\vec{p}_j)$ for some pixel j with feature vector $\vec{p}_j$ is the same as that in [1] and is given by

$$U_a(\vec{p}_j) = 1 - \frac{\left|\sum_{k=0}^{K} \frac{2C_a(\vec{p}_k) - 1}{(1 + \Delta(\vec{p}_j, \vec{p}_k))^n}\right|}{K}$$

where $\Delta(\vec{p}_j, \vec{p}_k)$ is the Euclidean distance between the feature vectors for pixels j and k, $C_a(\vec{p}_k)$ is the previously labelled classification value of 0 (negative class) or 1 (positive class) of the kth pixel and is provided as the user works through an image, n is the dimensionality of the feature set, and K is the number of nearest neighbors to consider and is specified on startup. This chosen form for uncertainty aids us in reducing bias towards strongly-confident and conversely very unconfident predictions, based on experimentation.

2.3 Image Segmentation

The problem of image segmentation is one of separating the important parts of an image from the background. For example, identifying all of the pixels which make up an individual's face in a photograph taken of a crowd. Perhaps known most commonly in the medical imaging field [3] or autonomous vehicle technology [4], image segmentation is also commonly used in geographic information systems [5]. This process can be done completely manually [6], completely automatically [7], or somewhere in between as in our case. The KNN classifier is utilized alongside the user to accomplish this approach. Since the KNN is a pixel-based classifier, individual pixels along with their associated feature vectors are given to the KNN and it will then indicate whether it thinks that pixel is positively or negatively labeled with respect to the class. By performing this pixel-based classification for each pixel in an image, we are able to clearly locate and label larger-scale geographical regions of importance. From there, we may conduct further analysis on the labelled objects within the image such as visualization and measurement.

2.4 Interactive Machine Learning

Interactive machine learning (IML) [8] is a subset of human-in-the-loop methodology that necessitates a user actively confirmed or correcting the machine implementation's actions, while the machine immediately learns and applies those confirmations and corrections. To assist an analyst in performing the task of locating and labelling interesting geographical features from imagery, we employ a KNN pixel-based classifier; this will make up the "machine" portion of the human-machine team. This team-based approach is what gives us the flexibility to operate from cold start: no previous training data is available, but the task is not entirely manual labor either. In an IML approach, we provide labelled training data to the machine as we work with the dataset, in small chunks at a time. The machine then takes what it knows about the system and gives to the user a suggestion for what it thinks is the correct answer. The user can then accept or correct the machine's suggestion, and the corresponding response from the user will generate new ground-truth data for the machine to process and learn from. Over time, the machine's confidence in labelling new data will grow more in line with the data that has previously been given to it by the user's interactions.

3. IMPLEMENTATION

Our approach to semi-automatic segmentation of 3D geographical regions is a multiple-step process which involves the analyst in each step to ensure accuracy and enforce the learning of the machine partner. For this report, we assume that the imagery contains at least one instance of the region or object to segment and analyze. With that assumption, the process can be summarized as follows:

1. Locate a region of interest – be it a dune formation, mountain, or other 3D object within the raster.
2. Determine the 2D spatial extent of the region.
3. Generate additional points within the segmented area to aid in the 3D visualization of the geographical region.
4. Visualize and/or process the now-segmented region for analysis.
5. Either look for more instances within the raster to process or save the results.

Between and during each of these steps, the user is given opportunity to interact with the machine's segmentation process by verifying or correcting it as needed.

3.1 Locating the Region

In the cold-start case, our KNN implementation has no model with which to classify each pixel. In such a circumstance, the analyst will give the machine partner a head start by manually identifying the general region of interest.

If the classifier has already been trained, be it from previous images or previous segmentations within the same image, a search for a candidate insertion point will be performed. For simplicity, this search is currently a column-based approach that utilizes the confidence model in [1], though any instance-based classification model that yields a confidence metric may be used. The insertion point is then determined by the following process: first, the classifier performs a raster-wide classification of each pixel and then searches for which three adjacent columns has the largest total confidence value. Then, the insertion point is simply the point within those three columns with the largest or smallest value in the height map. This behavior can be set by the user at startup, depending on the expected properties of the feature of interest. Finally, once the insertion point is found the user can correct the initial insertion point before proceeding by simply dragging the center or "peak" vertex to the preferred location. This manual correction may or may not be needed depending on the 3D shape of the region in question, as well as the underlying resolution of the elevation model. If there exist many points within the region that have equal or roughly-equal elevations, the physical peak of the object might not be located at the suggested insertion point. For example, the user may want the insertion point to be in the center of a plateau rather than on the outer edge of a region where all of the height values are equal to the maximum height of the segmented region.

3.2 Determining the 2D Spatial Extent of the Region

Once the location of the region has been specified, an initial set of three vertices forming a polygon will then be overlaid onto the raster within the training window centered at the insertion point found in the previous step. It is now up to the human-machine team to add additional vertices to this polygon, which will form the 2D spatial extent of the region.

When the user identifies an edge to place a new vertex upon, a one-dimensional *signal search space* is created perpendicular to that edge both inside and outside of the existing polygon, terminating on the interior of the polygon when the opposing edge is reached. The signal search space consists of the pixels and their associated feature vectors, which will be each be given to the classifier for classification. The

classifier will compute a confidence value for each pixel and a placement algorithm will determine the location for initial insertion. This vertex placement algorithm can be chosen from a set of several, based on the user's needs. Different vertex placement algorithms' performance will vary based on the context. The current choices for vertex placement algorithms are given in Table 1, and are the same as those in [1].

Table 1 — Vertex placement algorithms

Type	Description
Best Edge	The vertex is placed at the largest change in projected classification between neighboring pixels along the search signal space.
Split Point	Places the vertex at the location which maximizes the number of "label" pixels on one side while also maximizing the number of "non-label" pixels on the other side.
Near Continuous	Attempts to find the mid-point of the "phase transition" from "label" to "non-label" by identifying a region with several minimally-certain pixels lie along the search signal space.

After the initial insertion, the user can accept the machine-generated insertion point for the new vertex, or they can drag it to another location as necessary for accuracy. By accepting a newly-placed vertex, the user is creating a new set of labelled feature vectors along the signal search space. The KNN implementation uses these new feature vectors as ground truth data and incorporates them into its classification model each time a vertex is accepted.

This process of vertex insertion and correction (if necessary) is performed until the region is fully segmented from its surroundings.

3.3 Generation of Interior Vertices

The size of an object and the location and shape of its exterior boundaries is insufficient information to visualize it in 3D. Its elevation distribution is also required; or in other words, the digital elevation model must be sampled. There are two basic problems to solve here: how many samples of the elevation data are needed, and how are those samples distributed? In solving the "how frequent" problem, there is a balance to strike based context and user preference: too many samples of the height map will result in a cluttered 3D display and may cause performance issues on slower machines, but too few samples can result in reduced visibility of important 3D features. The distribution of the inner vertices ought to be context-specific. For example, organic shapes such as coastal dunes curve slowly and often have well-defined peaks and troughs. By comparison, synthetic structures such as buildings can curve quite sharply and may have many points at the same elevation which make defining a singular "peak" pixel problematic.

For now, the current implementation of the tool uses a fully-automatic approach to generating vertices that lie within the boundaries of the segmentation polygon once the user accepts the outer vertices. These vertices are placed within the geometry, and the distribution of these interior vertices is determined by an interior vertex generation algorithm (from those listed in Table 2, which is specified by the user at startup). The user should take care to choose which algorithm works best with their data set.

Note that there is currently no adjusting or correction of these vertices once they are placed, as they are for 3D visualization purposes only and do not contribute to the classifier. Future implementations may intelligently place interior vertices which the user can adjust to improve placement.

Table 2 — Interior vertex generation algorithms

Type	Description
Linear	Points are evenly spaced along a line connecting the outer geometry vertices to the initial insertion point. Generally results in a "spider web" appearance. The number of vertices placed along these lines is set by the user.
Grid	Points are evenly spaced on an *x-y* grid within the geometry. The spacing is set by the user.
Contour	Points are placed along contour lines according to the underlying digital elevation model. The number of contour lines to generate is set by the user.

3.4 Visualization

At any time, the user can choose to visualize all existing vertices in 3D. The tool uses JavaFX to generate a point cloud of the vertices along with their associated connecting line segments to form a mesh which aids in visualization. The spacing of the points in 3D is determined by their pixel locations within the annotated raster along with the height map data associated with each pixel location.

Within the visualization window, the user is able to freely move, scale, and rotate the point cloud in order to view the object from any angle or scale. The controls for this operation involve mouse movements similar to those used commonly in other 3D manipulation programs. Specifically, the user can click and drag the "front"-facing features to either rotate the object or translate it across the screen, depending on which mouse button is held. Additionally, the user can zoom in and out of the point cloud via the mouse wheel.

When the visualization is no longer needed, the visualization window can be exited and further annotation or data export can be performed.

3.5 Locating Additional Features and Saving Results

When a geometry is fully accepted, the user can then search the image for another, similar region. For example, if the user wanted to find and segment a second mountain peak or dune formation. But first, the classifier must train on the already-labelled geometries. Care must be taken here, because by previously training on search signals that extended out to the end of the image, the tool may have inadvertently mislabeled pixels which should be part of the next geometry's positive label class. For that reason, a rectangular area is defined around the previous geometries and the tool re-trains from scratch on all of (or a sample of) the feature vectors within that area. Then the tool can proceed to locating the new geometry.

The search for the second feature is performed similarly to the first, using a column-based approach that utilizes pixel-based classification. The primary differences in this case are two-fold: first, the classifier will never be cold-start here, since it has the first geometry's input as training data. Secondly, care must be taken to avoid considering pixels of the first region, or pixels too close to the first region, as constituting an entirely new region of interest. Since the goal is to find a distinct additional region, the user's previously-specified training distance threshold is used to determine how far away a pixel must be from the first region in order to be considered for an additional insertion location.

When the user is fully satisfied with their current imagery and that there are no more remaining features to segment, the user is able to train the classifier on all pixels of the image or just a sampling of them, as

desired. Then the user can move on to another image and use the now-trained KNN classifier to quickly locate a feature and repeat the segmentation process from there.

In the future, the user will be able to save and export their segmented 3D regions for analysis and comparison; this is discussed in Section 6.

4. EXAMPLE DATA SET

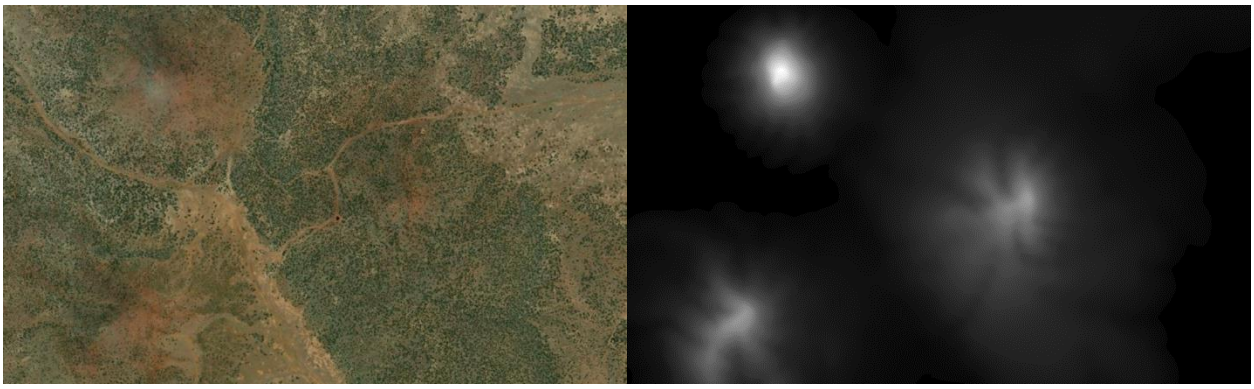

Figure 1: Satellite imagery (left) and digital elevation model (right) of the "Dog Knobs" formation in northeastern Arizona

A data set which would be well suited for analysis with this tool is one that has at least four available channels within its feature set: red, green, blue, and height or depth. This requirement is fulfilled any time satellite imagery is available alongside a digital elevation model of the same region. One can also use RGBD point cloud data generated by commercial LiDAR tools such as the X-Box Kinect. The dataset can be processed by our tool as image files, or given as a set of .csv files where each file represents a data channel and each value corresponds to a pixel value for that channel.

As an example of such a data set, we will use USGS imagery of a desert formation in northeastern Arizona named "Dog Knobs", located at 35.6599° N, -111.9250° E. A satellite image of this area is shown in Figure 1 (left). This area contains several distinct hill formations, as is apparent from the height map data (also provided by USGS) presented in Figure 1 (right).

5. DEMONSTRATION

Upon loading the raster data into the tool, the user will be presented with a "Training Window" which is where all of the steps described in Section 3 will take place. With the "Dog Knobs" data loaded in, the training windows looks as such:

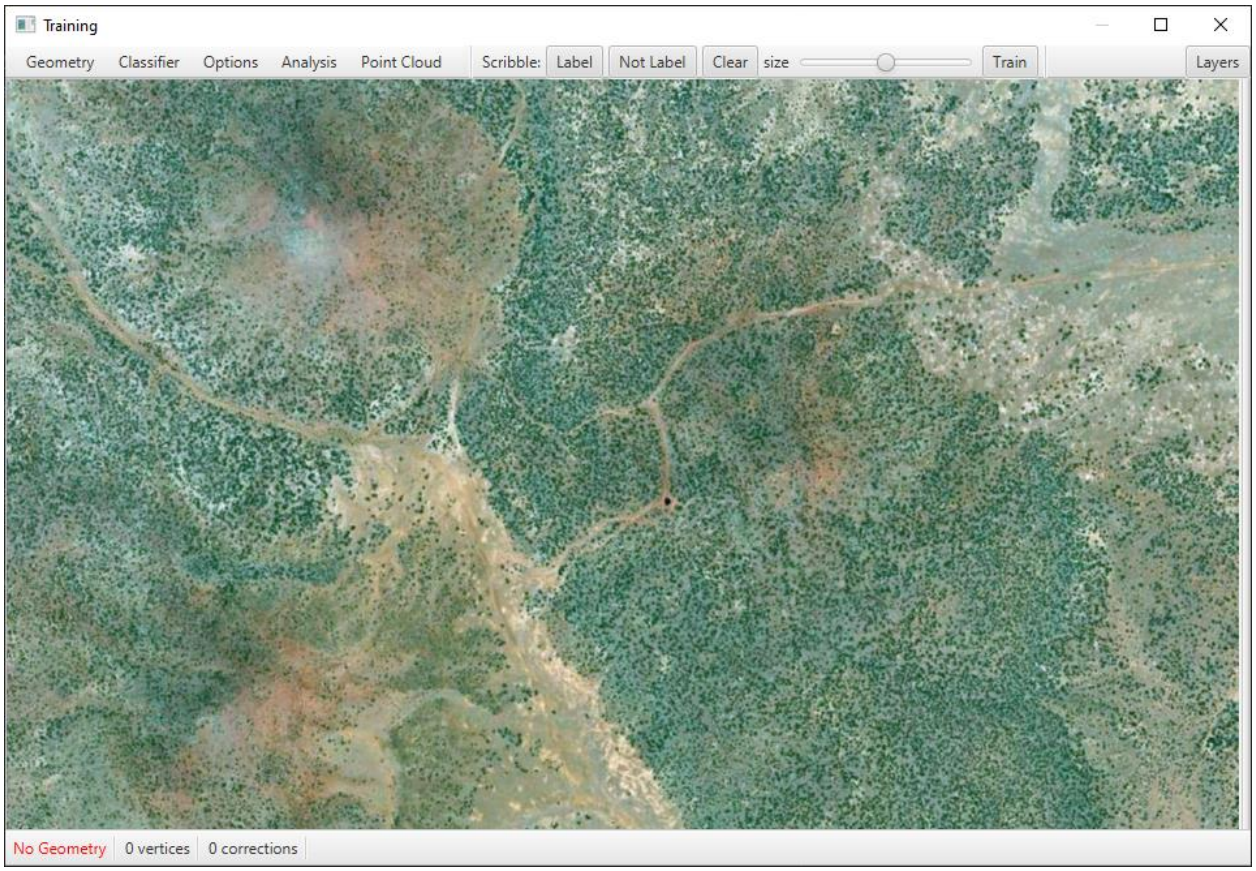

Figure 2: The example data set loaded into the training window

From here, the user has the option of viewing all of the underlying feature layers of the raster by opening the “Layers” button in the upper right hand corner of the window. This is what the user would see if they were to enable the height map data (in our example, the layer named “depth”):

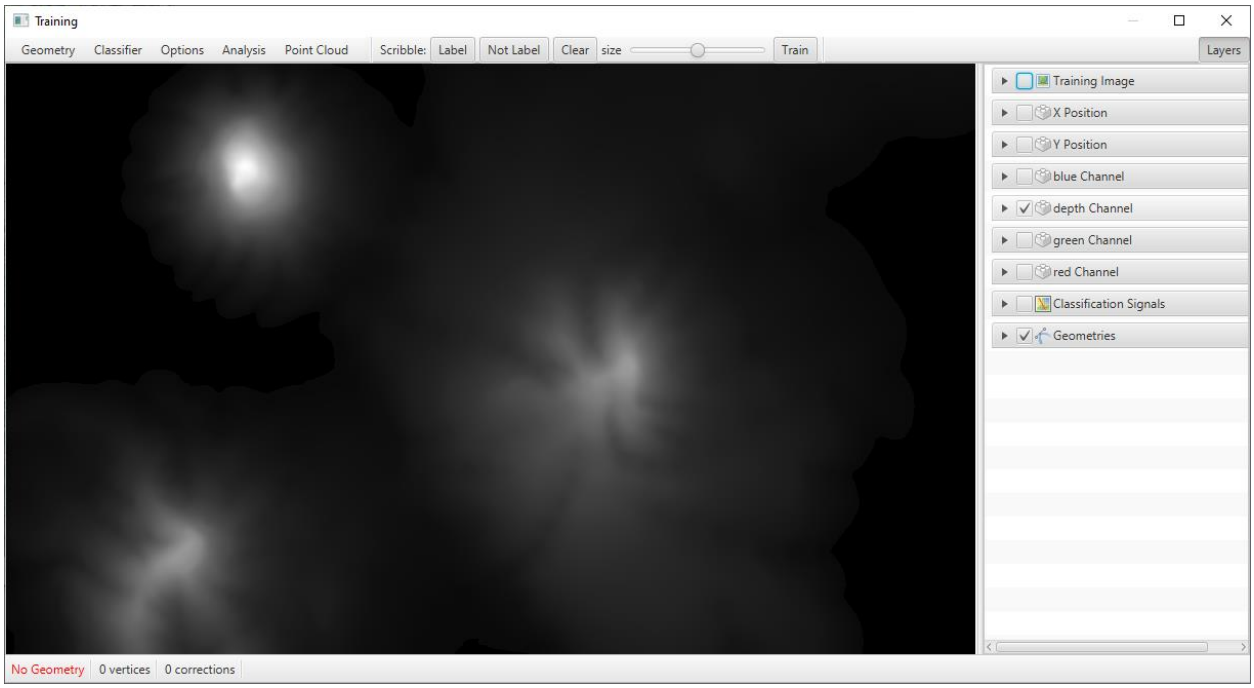

Figure 3: Toggling the elevation map data layer from the training window's "Layers" menu

Returning back to the normal view by unchecking the "depth" channel and re-enabling the "Training Image", the user can then begin segmenting the "Dog Knobs" formation by right-clicking somewhere in the general vicinity of its peak. The user will then be presented by an initially three-sided polygon geometry as such:

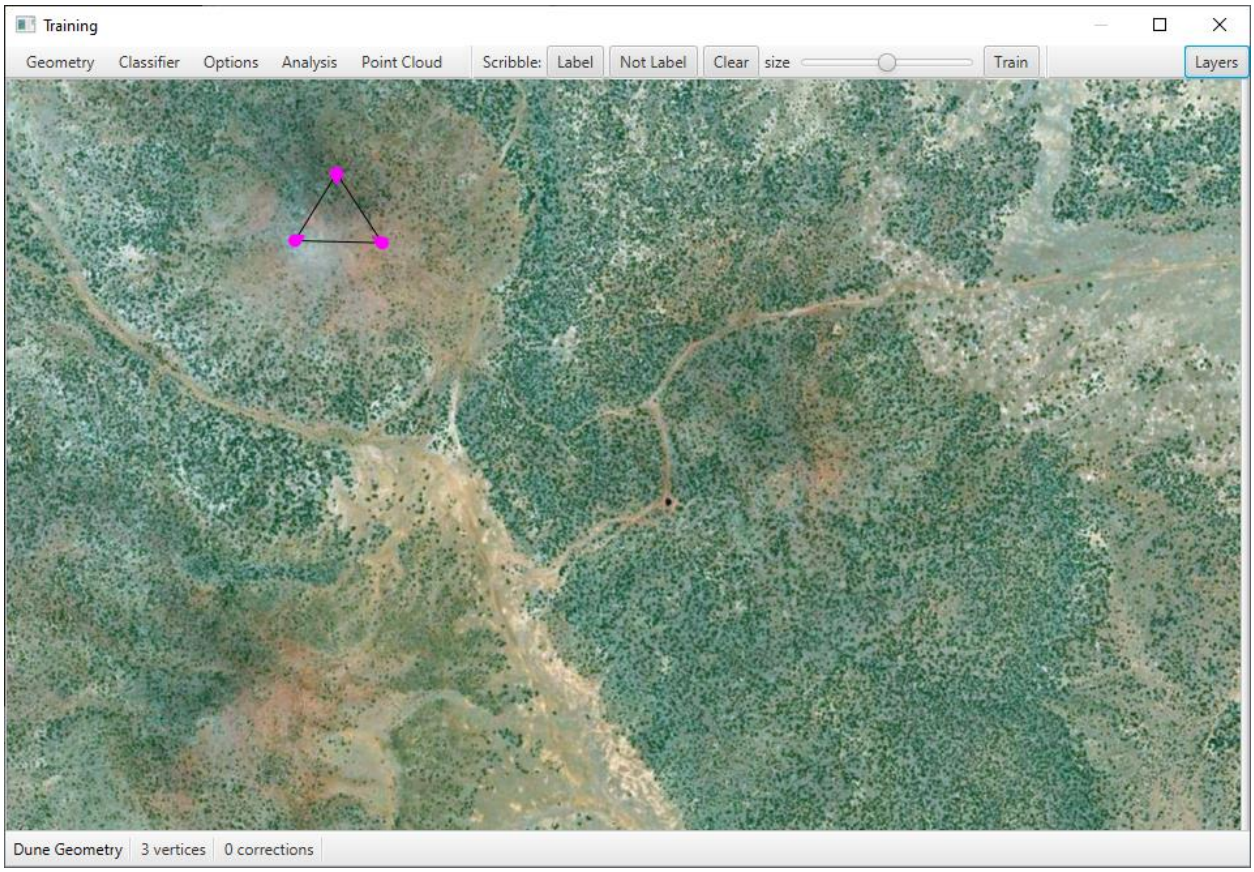

Figure 4: Initial insertion of the annotation geometry

The user will drag these pink (unaccepted) vertices to the desired location to fully contain the physical feature within, and then new vertices can be placed by right clicking again somewhere along the edges of the polygon. If the user were to return to the “Layers” menu and enable the “Classification Signals” layer, the user will see the following feature when adding new vertices:

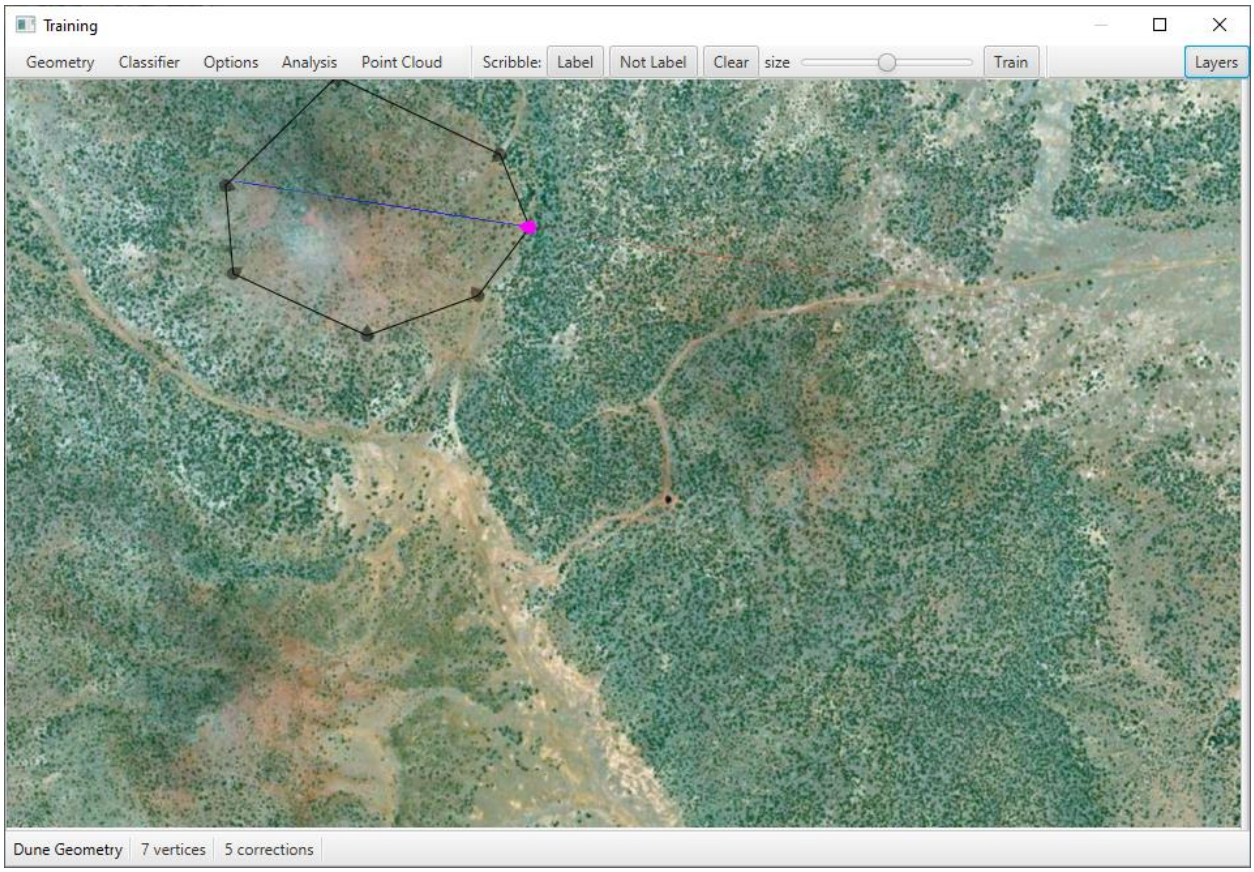

Figure 5: Classification signal for vertex placement is made visible

In Figure 5, a faint blue/red line is displayed along the line that the current unaccepted vertex is restricted to. This reflects the per-pixel confidence based on the current state of the KNN classifier model. Blue pixels denote confidence that the pixel is a positive label pixel, red pixels denote confidence that the pixel is a negative pixel, and the opacity of the signal corresponds to how confident the classifier is in its prediction. From Figure 5 it is clear that, currently, the classifier is quite confident that the pixels inside the polygon are positive pixels. Further, the classifier is identifying pixels within the search space outside of the polygon are negative pixels, but it is less certain about this region of the raster.

Once the user has fully segmented the region in 2D, the user will accept the current geometry and then interstitial vertices will be placed for the purpose of 3D visualization. The completed 3D geometry including interstitial vertices is shown in Figure 6. The distribution of these interstitial vertices will be based upon the user's settings within the Visualization Options menu.

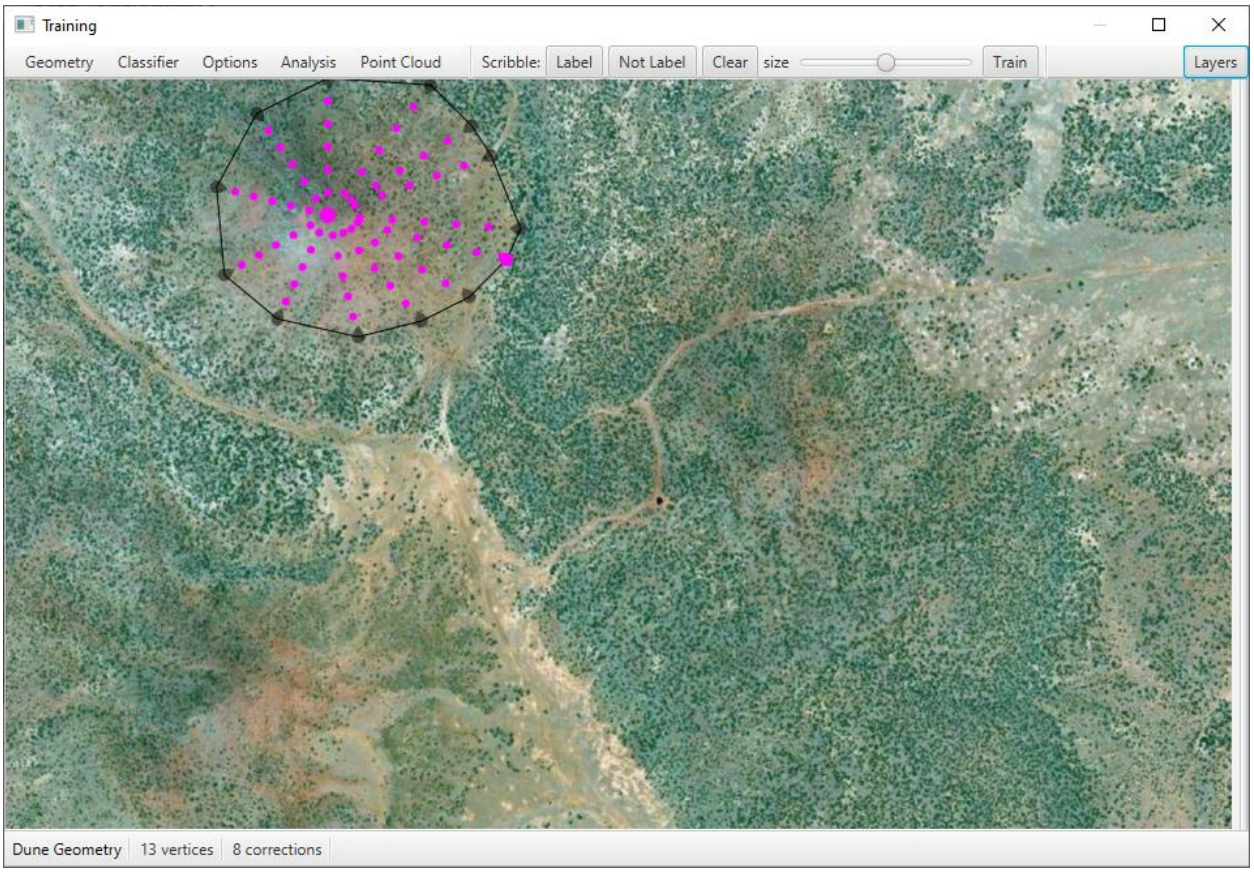

Figure 6: Interstitial vertices are placed prior to 3D visualization

To view the now-segmented 3D geometry, the user can access the visualization tool by opening the "Point Cloud" menu in the top menu bar as shown in Figure 7.

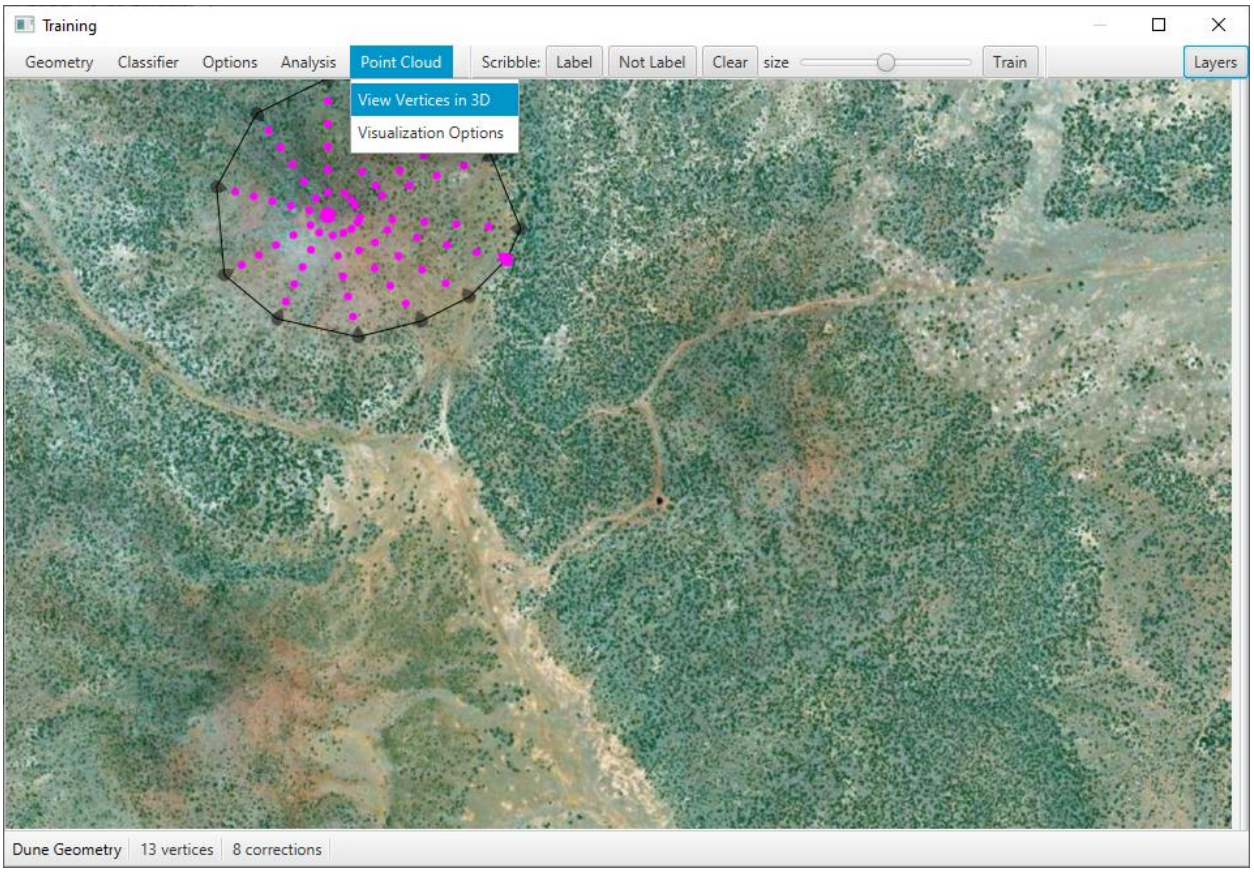

Figure 7: At any point, the user can visualize the currently-placed vertices in 3D by accessing the “Point Cloud” menu

The visualization window will then pull the vertices from the training window and generate an interactive geometric mesh as shown in Figure 8.

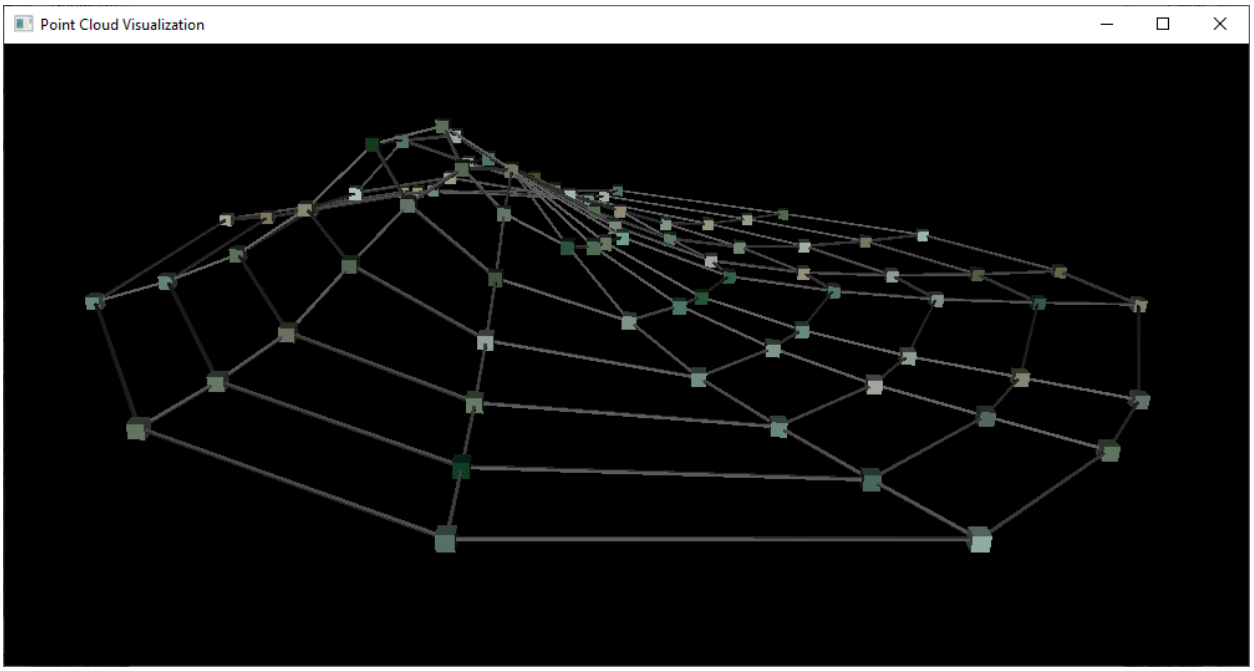

Figure 8: The 3D visualization window. The colored boxes correspond to previously placed vertices from the segmentation process, and the connecting lines form a mesh which serves as a visual aid. Note this view can be rotated, translated, and scaled by the user as they see fit.

6. CURRENT AND FUTURE WORK

This work is currently under rapid, fulltime development and features are added, refined and expanded upon frequently. As such, there are a few choice features on the near horizon that are not quite ready for highlight within this report but present a road map to a more feature rich analysis suite.

6.1 Save and export 3D geometries

To aid in longitudinal studies of geographic features such as the changing morphology of coastal dune formations, analysts should be able to make meaningful, direct comparisons of the shapes and volumes of such features extracted from imagery. To make this possible, the user should have the ability to save and export 3D geometries for future comparison. Currently, we do not have an implementation for this feature but we are investigating which of the various 3D geometry file formats lend themselves most to this task.

6.2 Measurements and analysis

Because the geometry represents real-world geographical features we seek to study, it's critical that we enable the analyst to extract real, physical measurements from the regions segmented by the human-machine team. Under current development is a prototype of an analysis suite which will allow the analyst to, once a region is segmented, view various statistics and measurements about the region. Since pixel spacing is related to real-world distances, and the height map values are also related to real-world heights and depths, one can specify a simple mapping that can for example display areas, volumes, heights, slopes, etc. Another example under development is the ability to select 2D slices of the region for plotting.

7. CONCLUSION

We have described the operation of a human-in-the-loop 3D geographic region segmentation tool which allows the user to import imagery alongside height map data to locate, segment, and visualize important 3D features.

We have also described the types of input data currently accepted, and shown an example data set which highlights the features described by the rest of this manuscript.

A demonstration of the current tool was provided with our example dataset as input. The complete process of locating the feature of interest, segmenting it from the rest of the image, and visualizing it in 3D space was stepped through.

Lastly, we discussed the features currently under development, and given the respective motivations for each proposed addition.

8. REFERENCES

1. C. J. Michael, S. M. Dennis, C. Maryan, S. Irving, and M. L. Palmsten, “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 System, SIGSPATIAL ’19, New York, NY, USA (Association for Computing Machinery), 2019.
2. L. Youcun, S. Bo, H. Tianding and Y. Baisheng, "3D GIS interactive editing method: Research and application in glaciology," The 2nd International Conference on Information Science and Engineering, 2010, pp. 3384-3387, doi: 10.1109/ICISE.2010.5690776.
3. M. A. Balafar, A. R. Ramli, M. I. Saripan, and S. Mashohor, “Review of brain MRI image segmentation methods,” Artificial Intelligence Review 33(3), 261–274 (2010).
4. M. Liu, C. Wu, and Y. Zhang, “A review of traffic visual tracking technology,” Proceedings of the 2008 International Conference on Audio, Language and Image Processing (IEEE), 2008, pp. 1016–1020.
5. M. D. Hossain and D. Chen, “Segmentation for Object-Based Image Analysis (OBIA): A review of algorithms and challenges from remote sensing perspective,” ISPRS Journal of Photogrammetry and Remote Sensing 150, 115–134, 2019.
6. Yi-Ying Chen, Wei Huang, Wei-Hong Wang, Jehn-Yih Juang, Jing-Shan Hong, Tomomichi Kato, and Sebastiaan Luyssaert, “Reconstructing taiwan’s land cover changes between 1904 and 2015 from historical maps and satellite images,” Scientific reports, 9(1):3643 2019.
7. JS Blundell and DW Opitz, “Object recognition and feature extraction from imagery: The feature analyst® approach,” International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(4):C42, 2006.
8. J. A. Fails and D. R. Olsen Jr, “Interactive machine learning,” Proceedings of the Proceedings of the 8th international conference on Intelligent user interfaces, 2003, pp. 39–45.
9. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.