NEON NIST - Tree Crown Delineation Final Report

Arvind Kumar Sugumar
M.S. Computer Science
University of Florida
Gainesville, FL, USA
ak681443@ufl.edu

Nishant Agarwal
M.S Computer Science
University of Florida
Gainesville, FL, USA
nishantagarwal@ufl.edu

***Abstract*— Automated Tree Crown detection and delineation is of great importance for forest management and with the availability of high resolution camera images along with HyperSpectral and Lidar point cloud data, the delineation of the tree crown structure and application of species classification methods can secure results of high accuracy. Recognizing ecological patterns across the geographical scale is crucial in understanding the effect of ecological change on human life, hence this study is aimed at figuring out the best crown delineation algorithms suitable for the NEON data set using an Ensemble approach and assessing their performance across multiple image segmentation process including the modified version of marker controller watershed using geodesic distance, and the region growing algorithm that considers the homogeneity between the crown shapes and size. The methods that we propose holds great potential in delineating trees across different vegetation and obtaining the topological attributes that can facilitate in the classification process.**

Keywords—NEON NIST identification; Crown delineation; image segmentation; watershed; region growing

Introduction

ITC crown delineation is the most essential component in discovering features pertaining to the plant vegetation which includes canopy structure and closure, stem volume and DBH, and vegetation type. These data features computed traditionally by humans requiring time consuming field surveys, could be automated to avoid human intervention which facilitates generation of more resource data needed for efficient forest management. Researchers have proposed that a strategic combination of the hyperspectral, Lidar and camera image can help achieve much better results in tree crown delineation, since we put all the information from the tree reflectance pattern, canopy heights from the ground surface and the raster camera image pixels to use.

Related Work (A)

Individual Tree Crown (ITC) delineation is the first step in building a pipeline which can perform automatic forest inventory tracking and hence there has been many different approaches proposed to handle this task either automatically or semi-automatically with minimal human intervention. For this body of work we are primarily concerned about the automatic ITC delineation problem and hence we can split most of the available algorithms as a special case of one of the following classes :

A. *Valley Following*: This algorithm, proposed by Gougeon in 1995 for automatic tree crown delineation works by identifying points of local minima i.e. valley's in the image by finding a pixel of lower value sandwiched between two high value pixels. From here this algorithm applies a five level rule to identify tree crowns and iteratively applies additional special cases to check for cases like individual branches sticking out and other anomalous entities. This approach has a reported error of about 7.7% but in reality this translates to about 81% of tree crowns getting delineated on an average when errors of omission and commission are taken into account.

B. *Watershed Segmentation :* This algorithm works by considering the forest region as a topographical surface where the reflectance value of each pixel is in a way some measure of the height of that region so it stands to reason that if we image pouring water on this surface we will get pockets of water stagnating in all the local minimas and hence we can effectively filter out individual segments. The vanilla version of this algorithm is really sensitive to noise and hence will always tend to over segment trees by combining a lot of them into one so to avoid this Meyer and Beucher in 1990 proposed a marker controlled variant which takes additional cues as to how many distinct colors of water to fill the basin with in order to better segment the surface.

C. *Region Growing :* Region growing is a traditional image segmentation algorithm which helps recognize individual objects in a composite image by taking as input few seed pixels from which the algorithm tries to expand to the neighbouring area using a typically 4 or 8 connectivity neighbourhood matrix, until a particular threshold is reached or if the growth cannot happen anymore. This approach was used by Culvenor in 2002 to identify individual trees after using a local maxima technique to predict the seed pixels

Dataset (A)

The dataset is obtained as part of the NEON Airborne Observation Platform (AOP) program, funded by the National Science Foundation (NSF), which collects data about its plots of forested region at a high volume and velocity, so to handle this influx of continental scale information we expect the need for an interdisciplinary collaborative effort between data science and ecology to build the state of the art system for automatic forest management. Towards this end University of Florida is joining NEON to host a NIST (National Institute for Standards & Technology) Data Science Evaluation (DSE) track to help foster new study and research in this field.

As a first step we propose the study of one NEON plot and add new plots as and when the contest proceeds to further years. In particular we want to consider the Ordway Swisher Biological Station (OSBS) which is jointly maintained by University of Florida as a year round observation plot.

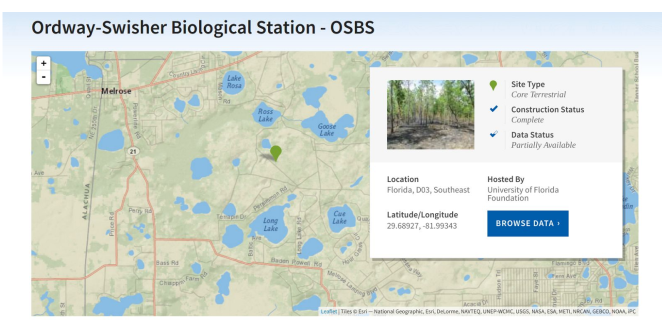

Figure 1: Ordway Swisher Biological Station

The participants in this DSE will be given the following data products as input to their algorithms

1.Raster Data Products:

<u>A.</u> <u>Camera</u> <u>Image</u> <u>:</u> RGB Image of the resolution 320 X 320 pixels will be given on a plot level which would represent a region of about 80 m^2 which would mean that each pixel would cover about 0.25m^2. These are given out as geoTIFF files so that the participants can easily project these on the world coordinates if required

<u>B.</u> <u>Canopy</u> <u>Height</u> <u>Model</u> <u>:</u> Processed LiDAR data product which gives the height of a particular spot on the RGB image. This is computed by sampling the LiDAR point cloud to a 80 x 80 pixel array which would again mean that each pixel point holds the average height of a 1 m^2 region of the plot

<u>C.</u> <u>Hyperspectral</u> <u>Data</u> <u>:</u> The hyperspectral data is the reflectance pattern observed by the spectrometer in about 428 spectral bands ranging from 380 nm to 2510 nm with a spectral sampling of about 5 nm.

2. Vector Data Products :

<u>A.</u> <u>Tagged</u> <u>Tree</u> <u>Shapefile</u> <u>:</u>Tagged trees are the tree crowns given by NEON which serve as the ground truth. This is usually collected by doing a field survey and is in the form of a shapefile which contains polygons which are represented on a projected square region on the earth. This shapefile also contains other useful attributes like the tree id, the breast height of the tree stump, its species identifier and so on.

<u>B.</u> <u>Untagged</u> <u>Tree</u> <u>Shapefile</u> <u>:</u> These are trees which are not part of the NEON dataset and were collected by some of the students in the University of Florida

<u>C.</u> <u>LiDAR</u> <u>Point</u> <u>Cloud:</u> LiDAR (Light Detection And Ranging) is a technique of getting the topography by bouncing of Laser beams and measuring the reflected rays to get a detailed map of points in three dimensional space. This data product is available as a x,y,z coordinate list where we can find an obstacle.

Evaluation Pipeline

As a part of evaluating the participant's submissions we have proposed an automated evaluation pipeline which works in tandem with the evaluation metrics defined in this paper. Apart from providing the actual results for the three tasks of crown delineation, alignment and classification, we intend to furnish other useful metrics which might help them fine tune their models to attain higher accuracy rate.To obtain these automated reports we have used template driven report generation system. In order to evaluate or visualize the ground truth data, reprojection is required. The shapefiles which contains the geographical location for the ground truth data have their coordinates mentioned in the WGS84 UTM Zone 17 North projection. We use an affine transformation to project the predicted polygons on the world coordinate system using the extend and dimension of the raster images. Once the test and ground data are mapped on the same coordinate system the evaluation metrics are calculated.

Metrics for Crown Delineation task:

1. Overall score based on the average Jaccard coefficients.
2. Histogram of precision: for each testing polygon x, precision is defined as

$$p(x) = max_n \frac{OverlappingArea(g_n, x)}{Area(x)}$$

where *gn* is a groundtruth polygon. A bar chart of the distribution is plotted with precision and count in x and y axis respectively.

3. Histogram of recall: For each groundtruth polygon g, recall is defined as

$$p(g) = max_n \frac{OverlappingArea(t_n, g)}{Area(g)}$$

tn is a testing polygon. A bar chart is plotted with precision and count in x and y axis respectively.

4. Confusion matrix showcasing the amount of false positives, false negatives and true positives.
5. Top-5 predicted samples with highest Jaccard score.
6. Top-5 predicted samples with lowest Jaccard score.

Metrics for Alignment task:

1. Overall score using the metric defined below
2. Histogram plot having X axis as the plot number and count of correct alignments in Y axis.

Metrics for Classification Task:

1. Average cross-entropy loss
2. Precision for each specie type.
3. Recall for each specie type.
4. Confusion matrix

Model Architecture

A. Laplacian of Gaussian Model:

This model works on the principle of finding the Laplacian i.e. second order derivative of the image matrix after applying the gaussian kernel to smoothen and remove most of the noise present in the image. This technique is highly sensitive to big changes in reflectance values and hence we can expect this to highlight regions of tree and ground. This technique is usually plagued by spurious edges which can be removed by using the zero crossings instead of the actual edges to find the initial approximate polygons. The found polygons can now be filtered in a systematic manner to prune ones that clearly are wrong, some of the filters we use are based on average height of the polygon, maximal clustering, geodesic distance transforms, polygon hierarchy (polygon contains another). This approach clearly gets most of the contours correctly and if we carefully set the window size it could identify almost all of the tree crowns but if we consider some cases where most of the trees are densely packed together this model over segments the image which can be broken down by either running morphological erosion to separate them into islands or run watershed algorithm again to get finer clusters.

$$\begin{pmatrix} 1000. & 1000. & 1000. & 1000. & 1000. \\ 1000. & 1000. & 1000. & 1000. & 1000. \\ 1000. & 1000. & 1000. & 1000. & 1000. \\ 1000. & 1000. & 1000. & 1000. & 1000. \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 2: Geodesic Distance Example

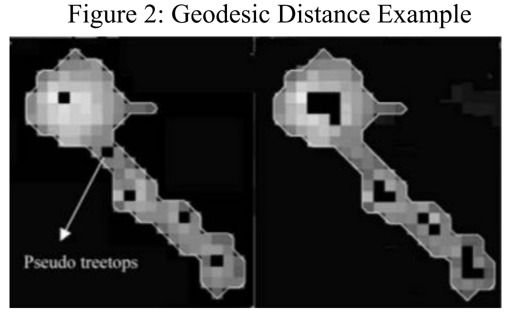

Figure 3: Geodesic Distance On Images

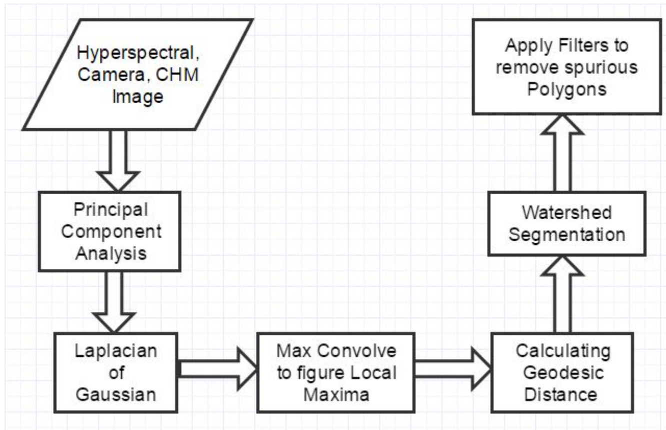

Figure 3: Laplacian of Gaussian Model

B. Region Growing Model

This model is based on the fact that the variation within a tree crown in terms of its height from the ground surface is less as compared to variation among different surrounding trees. As a preprocessing step we apply Principal Component Analysis (PCA) on the Hyperspectral images to reduce the dimensions of the image to 3 X 80 X 80, where 3 represents the color channels. After converting the image to grayscale, Otsu's thresholding scheme is employed on the resultant image. Watershed segmentation is applied on the image after performing distance transformation to obtain regions of sure background and foreground. We extract the initial polygons from the markers generated using approxPoly detection algorithm.These detected polygons suffers from the problem of over segmentation which is eradicated using the Region Growing Algorithm.

Only polygons having area greater than a threshold value are processed further. For each of these probable contours, local maxima points within that region is determined using a the CHM data and a convolutional filter of 5X5 which are termed as the initial seeds or the focal points.

Assuming these focal points as the starting point having the height greater than its surrounding pixels, an eight connected region is calculated using the expansion algorithm, which takes the difference between the maximum tree height and the current pixel height in making a decision to consider the new pixel as the starting point. This algorithm is based on BFS graph traversal where each pixel values are the nodes and the height difference between them as the edge value.The algorithm terminates when no new pixel can be added to the candidate region of the tree.

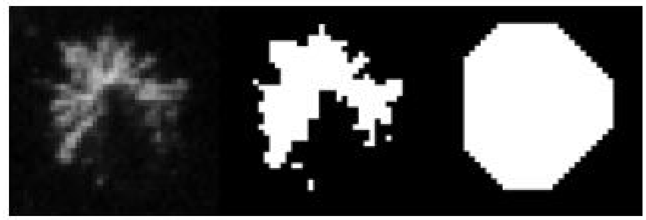

Fig 4: The middle image shows approximate tree contours obtained through Region Growing. The last image shows the convex hull drawn to estimate polygon surrounding this region.

The above step is repeated for all the points of local maxima.This approach suffers from a problem of overlapping polygons which need to removed using various filter mechanisms

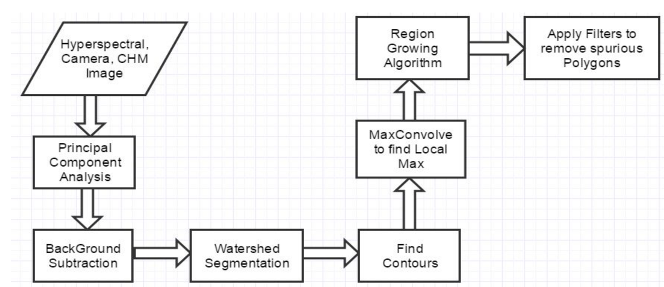

Fig 5: WorkFlow of Region Growing Algorithm

C. Ensemble:

By experimenting with many different architectures we have observed that each model has a different performance trait which makes it more suitable to one plot type rather than the others. This nicely allows us to train a bunch of models and use the model which works best for each case using the test data to make the choice at runtime thus enabling better dynamic performance without losing on the static optimizations which seem to help boost the performance on a plot level. Our model is actually an ensemble of three different models each optimized to perform under a certain combination of traits of the input volume.

Evaluation Methods

D. Evaluation Metric

1. Crown Delineation: Performance for the task of crown delineation is calculated based on a modified version of the F1 score. Assuming that the predicted crowns are non-overlapping, for each ground truth data, we find the best overlapping polygon having the maximum area of intersection with the ITC crowns. The above metric is a variation of the Jaccard Index also known as the Jaccard similarity coefficient and is calculated as :

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Here A and B represents the area of the predicted and ground truth polygon respectively. A perfect delineation of a tree crown would yield a score of 1. Similarly non overlapping polygons will fetch a score of 0. Finally an average similarity score is computed from scores for all ground truth polygons across all the plots. The above metric is continuous in nature and considers cases of true positive, false positive and false negative quite efficiently.

2. Crown Alignment : Performance of the pairing of field stems to ITC crowns will be conducted using the F1 score. In the testing stage, suppose we have a set of probe data (ITC) denoted as $\{p_n|n=1,..,N\}$, and ground truth data denoted as $\{g_n|n=1,..,N\}$,. We know in advance that there is a unique one-to-one mapping between P and G sets. Without loss of generality, let's assume pn should be mapped to gn for n=1,..,N. For each probe data point pi, a program predicts a non-negative confidence score that should be aligned with ground truth data point gj, which forms a prediction matrix M = (mi,j) where i,j = 1,...,N. Then, the quality of prediction can be measured by the following scoring function:

$$F_1 = \frac{trace(M)}{\sum_{i,j} m_{i,j}}$$

where trace(M) represents trace of a matrix M.

3. Species Classification: We will evaluate performance using two metrics, one that focuses only on the accuracy of the most likely classification and one that takes into account uncertainty in classification. Suppose we have N testing samples denoted as $T = \{t_1, \ldots, t_N\}$ which we want to classify into K different classes denoted as $c_1, \ldots, c_K$. For each testing sample t_n, a classification system assigns a probability $p_{nk} \in [0, 1]$ which represents the probability that t_n should be classified as class c_k, where $\Sigma_k p_{nk} = 1$ for all n. The two methods we will use are:

a) Rank-1 Accuracy: popular in face recognition, calculates the percentage of testing samples that can be correctly classified based on the top-1 matching class. It is calculated as:

$$\text{accuracy} = \frac{\sum_{n=1}^{N} \delta(\text{argmax}_k p_{nk}, g_n)}{N},$$

where g_n is the groundtruth class label of testing sample t_n such that $g_n = k$ if t_n is belonged to class c_k. The $\delta(x, y)$ is an indicator function takes value 1 when $x = y$.

b) Average Cross-Entropy: popular in neural networks classifier training, and is defined as:

$$\text{cost} = \frac{-\sum_{n,k} ln(p_{nk}) * \delta(g_n, k)}{N}$$

given that $p_{nk} \neq 0$, to avoid the singularity. The $\delta(x, y)$ is an indicator function takes value 1 when $x = y$.

E. Report Generation

In order to provide near real time feedback to the contestants we have also designed a pipeline which is capable of generating reports after automatically running the evaluation metrics on the received submission. These reports will contain additional metrics which will allow the contestants to get a better sense of where they stand when compared to the baseline version of the model and also learn where they are lagging behind so that they can utilize the testing slots allotted to them. Each report will have the following metrics in addition to each overall metric discussed earlier.

Results

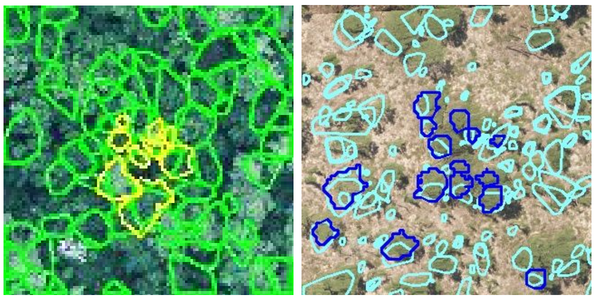

Fig 6 : Results of LoG delineation

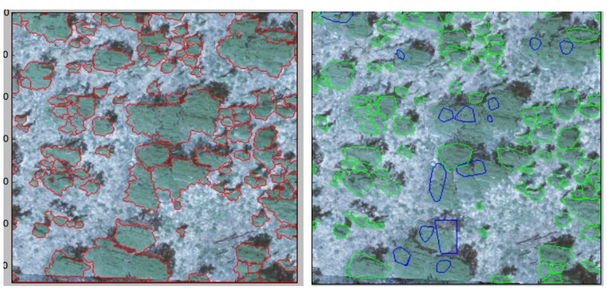

Fig 7: Results from Region Growing algorithm

Conclusion And Summary

1. Any real world data is bound to be noisy, so we must first account for that before embarking on application of any model.

2. Improvements to the model might lead us towards a local maxima so we must be careful about premature optimization

3. An ensemble of models, each tuned to work in a particular setting, would prove to be really beneficial to this task.

4. Watershed being a standard image segmentation algorithm needs further optimization process to achieve desired results

References

[1] Delineating individual trees from lidar data: a comparison of vector- and raster-based segmentation approaches. Jakubowski et.al. 2013

[2] Detection of individual tree crown in airborne lidar data. B.Koch et.al. 2002

[3] Individual tree crown delineation using local maxima approach and seeded region growing technique Jan NOVOTNÝ et. al 2011

[4] Individual Tree-Crown Delineation and Treetop Detection in High-Spatial-Resolution Aerial Imagery Le Wang, Peng Gong, and Gregory S. Biging

[5] Individual Tree Crown Delineation Techniques For Vegetation in Power Line Corridor Zhenzhong Li, Ross Hayward