PROGRESS REPORT

Techniques for Rapid Data Analytic of Remotely Sensed Data for Phenotypic and Precision Agriculture Applications

Ashvi Dua, Ajay Sharda, and William Schapaugh

Introduction

Plant phenotyping is the characterization of and quantification of physical traits of plants. Researchers used simple devices to measure traits for a few plots in traditional phenotyping. While these methods are simple and easy to perform for a small number of plots, they are not efficient for large-scale phenotype plots in terms of labor efforts, cost, and time.

On the other side, extraordinary advances over the 10-15 years in precision agriculture in crop monitoring and management have geared towards automated crop monitoring. In the realm of crop breeding plots, this approach is commonly referred to as "High Throughput Phenotyping"; it utilizes sensor systems & computational tools to extract phenotypic data for large populations. One of the most renowned, widely used, and effective tools and platforms is remote sensing, which has streamlined the rapid acquisition of imagery data. Unmanned Aircraft Systems (UAS) equipped with frame cameras make it possible to acquire images of large fields of thousands of small phenotype plots.

However, the major limitation of the widespread adoption of this technology in agriculture is the complex data processing systems and data analysis to estimate numerical targets. Here, the major disadvantage is locating plots precisely in high-resolution imagery of the field and delineating plot boundaries within the Ortho mosaic images of field experiments. When dealing with fewer than 50 plots, the manual creation of plot boundaries and integration with phenotype data for analysis requires minimal effort. However, as the number of plots increases into the thousands, this task becomes significantly more laborious.

Therefore, the aim of the study was to create multiple polygon shape files with unique identifiers that can be overlaid on drone imagery data with cm-level accuracy. The core idea was developing a pipeline without assumptions of field uniformity, plot spacing, size, or number of plots, and eliminating the need for manual adjustments and orientation of individual plots. To utilize precision agriculture techniques with high-accuracy plot position data from a precision planter and georeferenced UAV (Unmanned Aerial Vehicles) image data to generate plot boundaries. And to create an automated program capable of producing maps, multi-polygon shapefiles, and CSV files of plot boundaries for use in external software and downstream analysis. The proposed objective also included the idea of drawing plot boundaries that are derived without relying on image features and can be drawn regardless of vegetation presence. The goal was to design an open-source, efficient, adaptable, and replicable automated pipeline that minimizes time, labor, and user involvement while facilitating the extraction of zonal statistics for individual plots.

Material and methods

The methodology section outlines the systematic approach undertaken to conduct this research, elucidating the techniques and procedures in different steps. Each step contributed to a different outcome.

Datasets

The sample dataset consisted of non-nodulating soybean breeding plots planted for some experiments. This dataset served as the foundation for initializing a pipeline designed to create boundaries around various phenotype plots. The plot was planted in 2022 using 4 rows (12’ planted and 4’ foot alley) with 30 inches (0.762 meters) spacing between each row. The SRES 4-row planter was used to sow a specified and recommended number of seeds in each

phenotype plot. The spatial position of each plot was recorded using GPS (Global Positioning System) receiver and a data logger that was mounted on the planter exactly at the center of the 4-row planter (width 90”).

For each recorded position, the Trimble system logs information such as spatial coordinates (latitude and longitude), HDOP, VDOP, GPS Status, date, time, speed, and a unique event ID. The Trimble system records the spatial positions of the planter at three specific points within each plot: one at the beginning, one near the middle, and one at the end of the plot. These positions are determined based on remote output event signals that are triggered when the planter reaches those key locations. Since the three points within each plot have different event IDs, they are used to group the GPS data together, effectively associating them with the same plot. The collected data is exported to a CSV file, which is subsequently utilized as input for the planter's logged GPS data.

The second input was raster file/UAV imagery acquired by the five-band multispectral sensing system flown on canopy closure at 30 meters above the ground. Individual camera captures were imported into Agi soft's Meta Shape photogrammetry software to generate orthoimages of the field. Our photogrammetry workflow was based on vendor-recommended settings. The input image consisted of all the background metadata and pixel values.

The third part comprises four coordinates defining the region of interest (ROI). These coordinates can be directly input into the code, or alternatively, a shapefile for the ROI can be generated initially. This shapefile serves the purpose of delineating the experimental area and removing undesired edges from the multispectral imagery obtained during the image stitching process. All spatial data files were in WGS 1984, it is the coordinate system used by the Global Positioning System (GPS) for worldwide positioning.

The fourth file serves as the experimental area map, containing the grid layout of plots within the region of interest.

The fifth file, on the other hand, is the phenotype data file, encompassing a comprehensive set of phenotype data, including location, entry name, treatment, stand, harvest, yield, plot weight, moisture, lodging, height, seed quality, seed weight, protein content, oil content, and more.

Image post-processing/spectral data manipulation

In the first step, raster imagery was imported in Python using the raster IO library, and its metadata was explored to retrieve information about the raster, understand the data's properties, and conduct geospatial analysis, or visualize the raster. Then the imagery coordinate system was converted from WGS 1984 (decimal degrees) to UTM zone 14 (meters) for cm level and improved accuracy. UTM projections are designed for specific zones, which results in minimal distortion within each zone. UTM coordinates are given in meters, making them well-suited for distance and area calculations, this can reduce spatial computational load for spatial operations. The metadata of the imagery was examined once more to verify the consistency of all properties, excluding the coordinate system. A pixel histogram for the imagery was made to explore the distribution of pixel values. This pixel histogram is a graphical representation of pixel statistics, providing a visual impression of the data's distribution. It was observed that the histogram showed positive skewness, indicating that most pixel values across all bands were concentrated toward the right side. Also, a few

multiple peaks were observed, indicating the presence of soil, plants, land, etc. However, outliers were detected, which disrupted the overall appearance of the raster image. The outliers were single and very sharp peaks of some bands

Therefore, the region of interest using 4 coordinates was created to eradicate unwanted pixels and the unsmoothed edges of the image near the boundary. The four coordinates initially captured in the field were in the WGS 1984 coordinate system. Subsequently, these coordinates were transformed into the UTM zone system to overlay them onto the imagery and clipping all five bands of the Ortho mosaic. Many outliers in the image were removed, but still, a few outliers were left. Therefore, the next step involved was pixel thresholding. In previous approaches, vegetation segmentation using VI thresholds or pixel thresholds was employed to eliminate undesired pixels, including bare soil and shadows, to exclude potentially biasing factors. However, in this study, the decision was made to solely remove unwanted pixels by setting a predefined threshold value from the histogram. This choice was driven by the intention to retain the maximum amount of data within the image.

After pixel thresholding, the image was subjected to a filtering process for unwanted noise removal. In the noise removal process, a Gaussian filter was used. A Gaussian filter is a type of linear filter used for image processing and spatial filtering. It is widely used for tasks like noise reduction, image smoothing, and feature detection in raster data. The Gaussian filter is a weighted moving-average filter that gives more importance to the central pixel and decreases the weights as you move away from the center. The Gaussian filter kernel is a 2D matrix that determines the weights to be applied to the neighboring pixels during filtering. The kernel is characterized by its size and standard deviation (sigma). A larger sigma value results in a broader smoothing effect, while a smaller sigma value provides sharper feature preservation. The values define the behavior to suit specific image processing goals, finding the right balance between noise reduction and feature preservation.

Now, after noise removal VI index segmentation was done to remove soil, stones, water, land area, shadows, and undergrowth. The Vegetation Index (VI) values tend to exhibit higher values when compared to those of soil, stones, water, and shadows. Based on a literature review, it has been established that Normalized Difference Vegetation Index (NDVI) values can effectively be employed to differentiate vegetative foreground from the background. Typically, NDVI values below 0.33 are either omitted from the vegetation category or regarded as undesired components within the NDVI values linked to crops.

The segmented image was superimposed onto various bands to create masks that are applied across all bands. This process is undertaken to derive additional vegetation indices and values from the imagery. The methodology workflow with spectral manipulation is depicted in Figure 1.

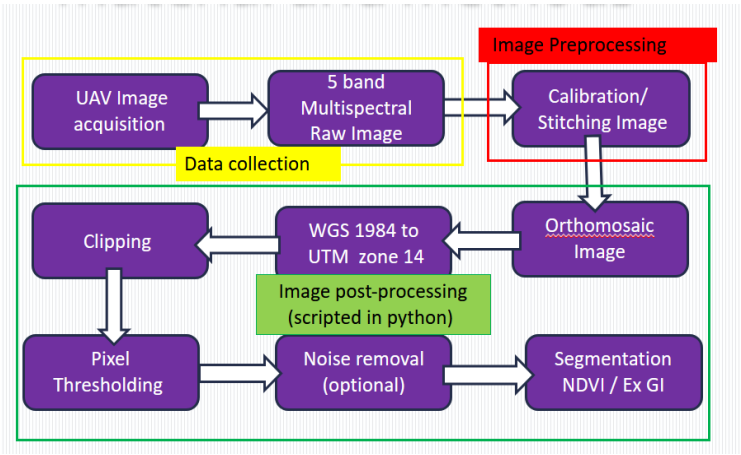

Figure 1. Workflow for spectral manipulation

Creation of plot and row boundaries

First, planter logged RTK GPS data was converted from WGS 1984 to UTM zone 14, so that they can spatially align with cm level accuracy. Next, the planter-logged GPS coordinates was sliced from a CSV file based on the 4 coordinates of the region of interest of the field. The dataset of Planter GPS points represented the movement of planters in a field; the points may correspond to the paths followed by the planters as they move across the field. In some cases, these paths can be organized into distinct rows, where each row represents a separate trajectory followed by a planter as it moves through the field.

Using HDBSCAN to cluster these GPS points can help identify and separate these various rows. The density-based clustering algorithm can automatically discover clusters of varying shapes and sizes without requiring you to specify the number of clusters in advance. This is particularly advantageous when dealing with datasets where the number of rows or trajectories is unknown beforehand or when the rows have varying densities and shapes.

By applying HDBSCAN to the GPS points, you can effectively group together points that belong to the same row or trajectory. Points belonging to different rows will form separate clusters, which can help you distinguish and separate the various rows in the field.

However, it is important to note that the effectiveness of HDBSCAN, or any clustering algorithm, depends on the characteristics of the data and the underlying patterns of movement in the field. While HDBSCAN can be a reasonable approach, evaluating the results, tuning the algorithm's parameters, and exploring other methods to ensure the accuracy of row separation in your GPS points dataset is essential. After separating GPS points of different rows, a moving window approached with a size of 3 in the dataset was used. Within each window of three elements, all three elements were assigned a common identifier (ID). This technique was employed to group individual plots with 3 GPS points in rows based on their relative positions in the dataset. If any individual plot had less than or greater than three GPS points in the plot, then a different approach was followed to give a unique ID to each plot. In this code, the code is iterated through different event Id through each row. "If the initial event for a plot was 'TRIP_REQ,' it assigned the same unique identifier to all GPS points within that plot until the next 'TRIP_REQ' event occurred. The same rule applied to two other event IDs as well." Afterwards, the GPS points with common group or unique id at the center of the planter width spacing, were connected to form line strings. Each GPS point corresponded to

a specific location where the planter operated, it was collected precisely at the center of the planter's width spacing. The width of the planter was known and the distance between each row of the planter was calculated. This distance represented the spacing between adjacent rows of each plot as the planter moved through the field. After creating an initial centerline representing the path the planter took, additional lines were drawn by offsetting them based on the determined row spacing. These offset lines represented the path of the planter for each row it planted in the region of interest. With the offset lines representing the path of each row, buffers were created around these lines. Each buffer's width depended on the desired coverage area around each row, which was determined by the planting equipment or the characteristics of the crop being planted.

Finally, polygon boundaries for each row of the crop were formed. Each polygon represented the area covered by a single row of the planter, considering the buffer width. So, in the ultimate step, a buffer was created for each line string. However, the line strings did not have consistent lengths due to delays in collecting GPS points. The resulting line string lengths differed since the distance between GPS points varied between different plots. All buffers were resized to the same length to standardize the buffer lengths for each plot. In the last step zonal statistics, the table consisted of plot ID, row ID, unique ID, a polygon object with coordinates of the four plot corners, centroid, area, length, width, pixel count, and VI (Vegetation Index) values. The methodology for spatial manipulation done by building a Python script is depicted in Figure 2. There are some variations on the methodology based on different datasets that will produce consistent results.

Result and Discussion

The maps given below in Figure 3. illustrate the transformation process from raw imagery to Vegetation Index (VI) maps and then the alignment of planter GPS data on the imagery. The preprocessing of the Ortho mosaic was carried out in Python, primarily utilizing the Raster IO library and straightforward existing algorithms and logic. To clip the imagery, the coordinates were input in the WGS 1984 coordinate system. In the pixel thresholding step, a threshold value of 10,000 was applied to eliminate unwanted pixels from the imagery. This was because the pixel values across all bands ranged from 0 to 10,000. A Gaussian filter with a sigma value of 2 was employed for noise reduction. It is important to note that these threshold and filter values can be adjusted as needed for several types of imagery.

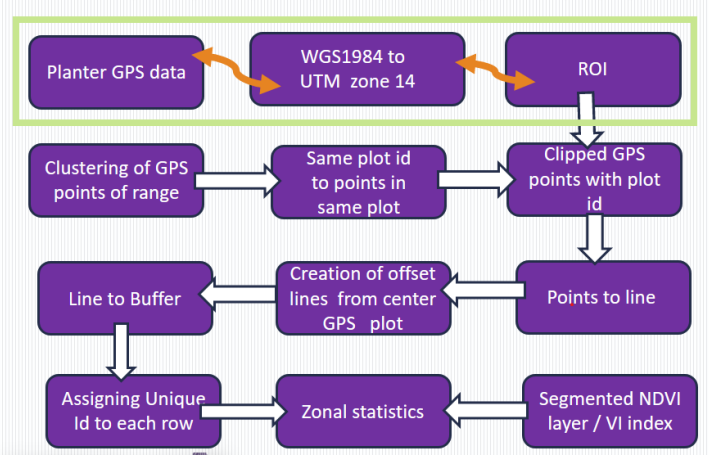

Figure 2. General work flow for creation of plots

Visually, it was observed that plot boundaries formed successfully aligned on vegetation. Afterward, the boundary files were exported to a shapefile file as well as a CSV file, which includes information such as plot ID, row ID, unique ID, a polygon object with coordinates of the four plot corners, centroid, area, length, width, pixel count, and VI (Vegetation Index) values. It was observed that the workflow reduced the need for many inputs to adjust plot boundaries. Overlapping canopies or crop lodging from adjacent plots or crop effects do not limit the effectiveness of the workflow. Automated workflow was obtained for both single-row and multiple-row plot boundaries. Unlike other studies, the pipeline avoided complex algorithms for processing images and managing spatial data. This also includes no assumptions about spacing between rows and columns of whole plots, length, number of rows or columns etc. Plot positioning was done with high accuracy RTK GPS precision planter. Plot boundaries can be extracted in any growth stage without issues of overlapping plots and insufficient ground cover. The pipeline obtained was an open-source, streamlined, flexible, and reproducible pipeline to reduce time, effort, and user intervention while obtaining zonal statistics for individual plots. The only limitation was high precision planter with accurate GPS devices is needed to ensure accurate and precise coordinates of ROI and research plots. The breeder should pay attention to the number of GPS points collected on each plot. It was also observed that the agronomic data was not recorded with spatial coordinates of that plot. Therefore, either each plot in the field is assigned a unique plot ID, and the corresponding agronomic data (e.g., yield, height etc.) is recorded along with the spatial coordinates of that plot or the plot IDs should be assigned based on the movement of the planter equipped with GPS.

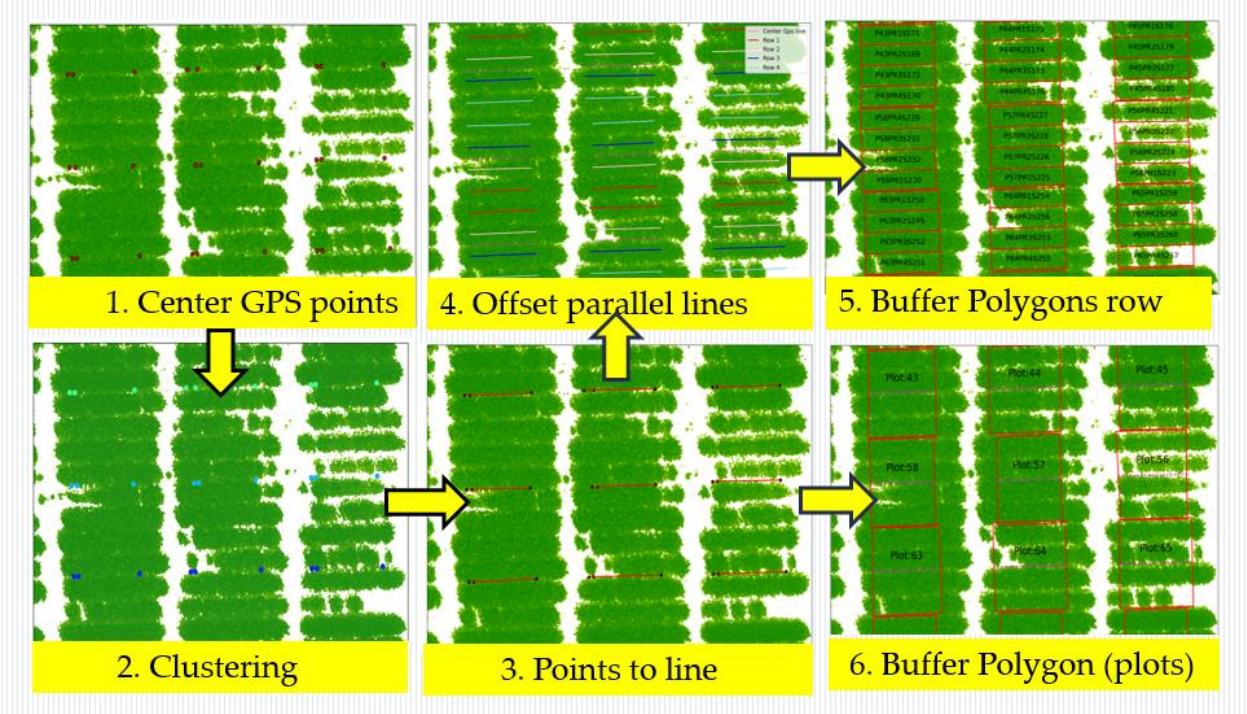

Figure 3. Close-up view for creating plot boundaries from Planter GPS data on segmented raster

Conclusion

The plot boundary extraction methodology presented in the study provided an accurate and efficient method. This research methodology used simpler existing algorithms to extract spatial signatures from imagery and plot boundary extraction from high accuracy precision planter.

Future work

It is essential to merge the map file containing information about phenotype plots with spectral, spatial data, and phenotype data. This integration is necessary to facilitate statistical inferences and informed decision-making. Researchers sometimes need to deal with multiple Ortho mosaics to conduct time series analyses. Due to errors in stitching and the collection of imagery data, as well as precision issues in planter data, there can be instances where multiple images do not align correctly with the vegetation boundaries. In such cases, including a boundary adjustment function within an interactive mapping system becomes essential. Additionally, there is a requirement to assess this pipeline by comparing it and identifying correlations with an existing methodology. In the second study, create a foundational statistical analysis framework for generating automated reports and visualizations and to seamlessly integrate it with the boundary creation pipeline. Subsequently, the goal is to develop a user-friendly GUI that allows users to utilize the data extraction and analytics tool, enabling them to generate interactive maps and automated reports with minimal human intervention. This tool will be effective for extracting field plot features and analysis reports, which can be used in high-put phenotype and further analyses in agricultural research.