HIGH-THROUGHPUT ANALYSIS AND ADVANCED SEARCH FOR VISUALLY-OBSERVED PHENOTYPES

A Dissertation

presented to

the Faculty of the Graduate School

University of Missouri – Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

JASON M . GREEN

Dr. Chi-Ren Shyu, Dissertation Supervisor

MAY 2012

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

HIGH-THROUGHPUT ANALYSIS AND ADVANCED SEARCH FOR VISUALLY-OBSERVED PHENOTYPES

presented by Jason M. Green,

a candidate for the degree of

Doctor of Philosophy

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Chi-Ren Shyu

Dr. Heidi Appel

Dr. Jianlin Cheng

Dr. Dmitry Korkin

Dr. Mary Schaeffer

DEDICATION

To my Mom, Dad, and family, for their constant love and support.

To my friends for being there for me during this long journey.

To God for giving me the strength, the persistence, and the ability to accomplish this goal.

ACKNOWLEDGEMENTS

First and foremost, I would like to offer my most sincere thanks to my advisor, Dr Chi-Ren Shyu. He has consistently challenged me to be a better student and researcher, provided invaluable suggestions and ideas when obstacles were encountered, and been a constant source of encouragement and motivation. My growth as an educator and a researcher is in large part due to his guidance and wisdom. Additionally, I would like to thank the members of my committee for their feedback and suggestions during this process, which have made this dissertation stronger.

This work would also not be possible without the expertise and support of our plant experts. A special thanks goes out to Dr Ed Coe and Dr Gerry Neuffer for their seemingly limitless maize wisdom; to Dr Peter Balint-Kurti and Dr Guri Johal for their expertise with plant pathology; to Dr Mary Schaeffer and Dr Toni Kazic for their suggestions and general maize expertise; and to Dr Heidi Appel and Dr Erin MacNeal Rehrig for fruitful discussions and suggestions regarding interactions between plants and insects.

These acknowledgements would not be complete without mentioning my colleagues in the MedBio Research Lab. I would specifically like to thank Dr Jaturon Harnsomburana, Jing Han, and Hongfei Cao for their ideas, suggestions, assistance, and support throughout the development of this work.

A final thanks goes to my funding sources, including grants from the National Science Foundation (#DBI-0447794, #DBI-1053024) and the National Library of Medicine (#2T15LM007089-17) as well as the Shumaker Endowment for Bioinformatics.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

HIGH-THROUGHPUT ANALYSIS AND ADVANCED SEARCH FOR VISUALLY-OBSERVED PHENOTYPES

JASON M. GREEN

Dr. Chi-Ren Shyu, Dissertation Supervisor

ABSTRACT

The trend in many scientific disciplines today, especially in biology and genetics, is towards larger scale experiments in which a tremendous amount of data is generated. As imaging of data becomes increasingly more popular especially in experiments related to traits and phenotypes, the ability to perform high-throughput big data analyses and to efficiently locate specific types of information within these data based on increasingly complicated and varying search criteria is of great importance to researchers and scientists.

This research develops several methods for high-throughput phenotype analysis, which notably includes a two-tier floating approach to variable object pattern matching, thereby facilitating in-depth quantitative temporal analysis of lesion burden progression. This algorithm utilizes spatial configurations to address three tasks: 1) handling difficulties in registering multiple indistinct and dynamic objects across images, 2) handling variations in translation, rotation, and local distortion, and 3) detecting the presence of missing or extra objects. These methods aim to produce accurate and unique trait data that can help researchers gain further understanding and insight into the underlying biological and genetic mechanisms that mediate phenotypic expression.

Research accomplishments also resulted in a number of unique advanced search mechanisms including a retrieval engine that integrates multiple phenotype text sources

and domain ontologies to improve search results. Another new and significant search method developed herein retrieves objects based on temporal semantics and behavior. These search mechanisms represent the *first* of their kind in the phenotype community. While this computational framework is developed primarily for the plant community, it has potential applications in other domains including the medical field.

CHAPTER ONE

INTRODUCTION

The trend in many scientific disciplines today, especially biology and genetics, is towards larger and larger scale experiments in which a tremendous amount of data is generated. Couple this with the fact that imaging of data is becoming more and more popular especially in experiments and studies related to traits and phenotypes, and it is evident that researchers and scientists must have the ability to 1) perform large-scale high-throughput analyses of these image data and 2) efficiently find specific types of information within the data based on increasingly complicated and varying search criteria.

Many disciplines, including genetics and ecology, require accurate characterization of traits and complex phenotypes in individual samples in order to facilitate scientific discovery. In the genetics realm, the ability to perform this task can lead to the assignment of biological functions to genes. This can be accomplished in combination with either forward (traditional) genetics [1], in which the detection of mutants with observable changes in growth, morphology, or color is followed by the identification, cloning, and characterization of the gene causing a mutation, or reverse genetics, which operates from the opposite direction by "knocking out" or over-expressing a known gene and then measuring the phenotypic expression that results. Both techniques, which pervade biological research today, are dependent on the ability to detect and quantify phenotypic distortions or changes in order to elucidate and understand gene function. Ecological experiments also utilize accurate and precise quantification of traits to help

researchers gain insight into the defense mechanisms and environmental variables that affect organism interactions.

One difficulty in accurately characterizing some trait appearances is that some of the most interesting and complex traits are often not directly measurable, e.g., disease resistance or root patterns, or are very tedious to measure. One way in which these cases have been traditionally handled is either through subjective evaluation [2-4], either by reduction of the phenotypic appearance to binary (e.g., the organism is or is not resistant to this disease, the plant did or did not have a color change) or through the use of scoring rubrics [5-6] . Both types of approaches introduce some loss of information, and this is increased oftentimes by the reliance on human perception, subjectivity, and inconsistency in the scoring process.

Trait imaging in conjunction with computational approaches represent one route that can address this issue. By collecting phenotype imagery, computer vision and image processing algorithms [7-8] can be used to perform more sensitive and accurate measurement and/or analysis of phenotypic expression. This may even allow detection and quantification of situations in which there are no human-observable changes or, at best, only very subtle changes in phenotype or trait appearances. Furthermore, computational methods have the added advantage of enabling high-throughput phenotyping, which is of great value as phenotyping is often the bottleneck in genetics research.

Additionally, despite the fact that the appearance of many traits progress or change over time (e.g., height, disease resistance), in-depth quantitative studies of these traits' temporal behavior are often lacking due to the difficulty in collecting and analyzing

these data. This is especially true of plant defense and disease resistance data. Computational methods to facilitate analysis of time-series image data are needed to allow the temporal component of these phenotypes to be studied.

Not only do researchers need to be able to analyze their data quickly and accurately, but with the overwhelming amount of data being collected, advanced retrieval mechanisms to aid researchers in locating specific data in their collections are also necessary. This is especially true in the context of image data, as there is an insufficient amount of time, money, and personnel to manually annotate all collected data; therefore, search mechanisms that make use of image content itself or measurements derived from image content will be increasingly important. To illustrate, consider a researcher who happens across an abnormal phenotypic appearance. That person may want to know if this appearance is unique to this sample or if something very similar occurs elsewhere in the collected data. Without a retrieval engine to support such a search, the researcher would be forced to manually sift through all the image data in order to answer that question.

1.1 Problem Statement and Summary of Contributions

Due to the common nature of large-scale experiments in biology and ecology and the increasing reliance on imaging to capture and analyze phenotypes, methods are needed to support high-throughput computational processing of these data. In this dissertation, methods are developed specifically for a few select plant phenotypes, although the techniques can be generalized to other phenotypes and domains.

Imaging of phenotypes is often performed in environments that can be controlled as much as possible, wherein attempts are made to minimize or eliminate variations in

lighting, obstructions from other objects, camera artifacts, etc. While methods to analyze phenotype images in these controlled environments are very useful and have produced a large amount of knowledge, they are not applicable in all situations. This is particularly true when the environment cannot be controlled (i.e. in the field), which incidentally is the location where many large-scale plant experiments take place, and oftentimes when *in situ* imaging is needed. The question remains: How can one achieve high-throughput processing of phenotypes imaged in the field where a host of imaging complications is present? This dissertation (see Chapter 2), will answer that question and show the step-by-step development an imaging protocol for field imaging which facilitates standardized capture of some phenotypic traits. This protocol allows for nondestructive imaging of plants in a variety of growing conditions and can be used in less than optimal imaging conditions. A robust normalization module capable of transforming these images to a common baseline in terms of color and scale is developed herein, which can be used to facilitate high-throughput processing.

There are many traits or phenotypes, i.e., herbivory and disease resistance, that are difficult or tedious to accurately measure. As such, it is common with these types of traits, especially in larger experiments in both the lab and field, for scoring rubrics to be utilized; however, as these methods rely heavily on human perception, estimation, and memory, measurement can be subjective and inconsistent. Can methods be developed, especially high-throughput computational approaches that can be used to analyze larger experiments, to more objectively, accurately, and consistently measure these traits? In this dissertation, we develop high-throughput methods that utilize imagery to extract meaningful trait information for both herbivory and disease resistance. In particular,

both unsupervised and supervised techniques for automatically scoring disease resistance to Southern Leaf Blight in maize from imagery of important leaves are developed (see Chapter 3). A publically available web application called PhenoPhyte is also developed (in Chapter 5) which facilitates automatic high-throughput measurement of area, growth, and herbivory from imagery of rosettes and leaves.

In-depth quantitative studies of the temporal progression of phenotypes are less common than more static evaluation of phenotypic expression, possibly because of the resources required to collect such data and the increased difficulty in collecting and processing the data. However, the temporal characteristics of phenotypes may be important, and computational methods for analyzing these types of datasets are needed. In this dissertation (see Chapter 4), a high-throughput method is developed for analyzing time course data of maize lesion mutant leaves to facilitate the study of lesion formation and progression.

With the massive amounts of phenotype data being collected and the increasingly complex search needs of researchers, novel advanced search mechanisms are needed to help support research and to aid scientists in efficiently and creatively explore these data collections. Here, a series of advanced retrieval methods are developed to support phenotype search (see Chapter 6). This includes a text-based search engine that utilizes domain ontologies and free text fields in model organism databases, a content-based retrieval system for lesion mimic mutants in maize, and a technique for retrieving objects based on their temporal behaviors.

In summary, this research's main contributions are the development of (1) a robust module for normalization of phenotype imagery containing a mini color checker with

respect to color and scale, (2) a supervised method for automatically scoring disease resistance to Southern Leaf Blight in maize from imagery of leaves, (3) a method to register necrotic lesions on maize leaves from time course experiments to facilitate the study of the progression and formation of these lesions, (4) a publically available web application called PhenoPhyte that enables plant scientists to automatically measure herbivory and growth, and (5) a series of advanced search mechanisms for various phenotype data.

1.2 Dissertation Organization

This dissertation is organized as follows. Chapter 2 contains a presentation of capturing visual phenotypes using imagery, which notably includes our custom image normalization algorithm for standardizing phenotype imagery. Chapter 3 discusses the quantification of phenotypic traits from imagery using computational approaches including a number of processing pipelines developed for images of a variety of species and traits. Two methods for translating these low-level features derived from phenotype images into more high-level phenotype scores or values are also described. In Chapter 4, a novel method for registering lesions in time course studies to facilitate temporal analysis is described. Chapter 5 discusses PhenoPhyte, a publically available web application available to the plant community for measuring herbivory and growth in leaves and rosettes. Attention is then turned to various visual phenotype search methods in Chapter 6, where a variety of advanced techniques for helping biologists efficiently search through data is presented. The dissertation then ends with conclusions and future directions in Chapter 7.

CHAPTER TWO

VISUAL PHENOTYPE CAPTURE

Since the onset of genetic research in plants, scientists and researchers have been devising techniques for measuring phenotypic traits in plants of interest. While some plant traits lend themselves to being measured easily (plant height, seed weight, stem diameter, etc), other traits, especially more complex traits like disease resistance or 2D or 3D growth, are more difficult to quantify. The advent of imaging equipment as well as the now ubiquitous nature of these devices, ranging from microscopy to flatbed scanners to digital cameras, however, permits the acquisition of phenotypes through imagery, and as expected, there have continual increases in the imaging of plants and phenotypic data in the past several years. These images, however, are of limited value from a computational processing standpoint as imaging standards are not defined. A concerted effort by the plant community is needed to both standardize the means of image capture and to persuade researchers to digitize, store, and share phenotype image data adhering to these standards. In this chapter, we discuss such an imaging protocol that has been used to image multiple phenotypic traits in multiple plant species.

2.1 Literature Review

Imaging of plant structures for phenotyping has been ongoing for years. Different techniques have been developed depending heavily on the structure a researcher is interested in capturing. Various techniques for imaging root systems, for example, include X-ray computer tomography [9-10] and nuclear magnetic resonance imaging (MRI) [11] in native soil. With the use of transparent growing mediums, digital cameras and laser scanners have been applied for both 2D [12] and 3D [12-13] imaging of root

systems. For leaves, scanning [14-15] or photography [16] of detached or harvested leaves is common. More high resolution imaging has also been conducted. Examples include [17], in which leaf imaging was performed using microscopy in combination with chlorophyll fluorescence, and [18], where a technique for imaging leaves using MRI was explored. More advanced imaging chambers, like the Lemnatec's Scanalyzer HTS [19] and Scanalyzer 3D [20-21], that provide imaging over a wide range of wavelengths including visible, infrared, near infrared, and fluorescence have also been developed. These chambers can also be used for 3D reconstructions using images from multiple angles. In addition to imaging individual plant structures and even whole plants, studies exploring the use of thermal infrared imaging on field plots have also been explored [22]. Though various imaging options are available, they are mostly non-portable systems in very controlled environments. Standards for imaging phenotypes in the field or other locations where environmental conditions cannot be so easily controlled are still needed.

2.2 Rationale for a Standardized Imaging Protocol

The impetus for standardizing image capture stems from two main sources: (1) the well recognized benefits of capturing phenotypic expression through imagery, and (2) the lack of development, much less consensus, of an imaging protocol to standardize two-dimensional capture of many types of visual phenotypes.

There are a few notable benefits that can be realized from digitizing phenotypic expression. One of the most compelling advantages is that saving phenotype data in a machine-readable format facilitates computational quantification of phenotypic expression. Many computer vision and image processing (CVIP) algorithms exist that

can be used to directly or indirectly measure various phenotypic characteristics accurately, precisely, and quickly. In many cases, a fully automated phenotyping pipeline can be constructed using these algorithms to permit high-throughput phenotype extraction from a collection of images. If the phenotype data were captured digitally and stored, the same data could be reused in the future in new and different analyses, which may become possible as more sophisticated and accurate processing algorithms are developed or when new research questions are asked of existing data. This has the potential to greatly reduce experimental costs and improve the speed of research.

Standardization of the imaging protocol also has the potential for high impact in the plant sciences. Currently, many experiments are performed in isolation with little means to re-utilize phenotypic data for other purposes. In addition to allowing a research group to reuse their own data, standardizing imaging across groups facilitates the possibility of combining and/or comparing multiple small, previously disparate, datasets for more large-scale analyses. This could be vastly important in the future as it provides a means for the community to pool data in order to conduct more comprehensive studies of the effects of environmental conditions, treatments, and genotype. This benefit, however, is highly dependent on more widespread adoption of such an imaging protocol.

In order for plant scientists to be willing to adopt an imaging protocol, it must meet several criteria. First, the protocol must be easy to use and flexible for the plant researcher collecting data. It should work with common digitizing equipment, e.g., scanners and digital cameras, and ideally allow capture of multiple plant species, body parts, and traits. The protocol should allow capture in various plant growth conditions (in the lab, greenhouse, and field). Finally, it should seek to facilitate accurate

segmentation of plant structures and robust measurement of phenotypic traits via CVIP algorithms.

2.3 The Imaging Protocol

We have carefully designed and implemented an imaging protocol capable of capturing an array of visual phenotypes that supports robust and flexible phenotype expression quantification via computer vision and image processing algorithms. We believe this protocol meets the above described criteria. There are four main components to the protocol, and though each is relatively simple, they are all critically important to the success and accuracy of this automated phenotyping approach.

(a) (b)

Figure 1: Example images of maize leaves taken in natural field settings illustrating the differences between not using (a) and using (b) our imaging protocol.

The first component is the use of a homogeneous background in the image that provides high contrast to the object of interest. This is important to facilitate accurate computational segmentation of the plant area being imaged. To understand why this is important, consider the images of maize leaves in Figure 1. Without a background to highlight the leaf of interest, it can be very difficult for computer programs to automatically separate a leaf or plant from the other objects in the image, especially

considering that images may be taken in field conditions where plants are grown closely together, as is the case with Figure 1(a). In particular, it is difficult for even the human eye to trace the bottom edge of the leaf in Figure 1(a), and so training a computer to identify this boundary or distinguish between foreground and background leaves is of considerable difficulty. That difficulty is eliminated in Figure 1(b) with the use of a high contrast background. Backgrounds should be solid, uniform colors. Good color choices for backgrounds are bright colors that will not appear in expression of the phenotype. Empirically, we have typically found it undesirable to use a grayscale background as this can make the edge of the object of interest more difficult to visualize and segment, especially in images that are underexposed and in images that contain shadows. In our datasets of maize leaves, we chose a bright blue background to contrast the green leaves and the necrotic and chlorotic lesions on the leaf (Figure 1(b)).

Figure 2: Image of a GretagMacbeth Mini Color Checker, which is the one used in all our experiments.

The second aspect to the protocol is the purposeful placement of a color checker (Figure 2) completely within the field of view. These color references are well known in the imaging industry and consist of 24 predefined color wells that can be used to evaluate and adjust image color profiles. Because lighting conditions can vary between

images, and more dramatically when taking images outdoors, the color checker provides a reference that can be used for normalizing or standardizing images with respect to color. In addition, the known size dimensions of the color checker provide a size reference that can be utilized for scaling purposes. By placing the color checker in the field of view, images taken from different distances with different cameras and in different lighting conditions can be transformed to a common baseline to allow standardized colorimetric and size measurements and to facilitate accurate comparisons between phenotype images.

The third component of the protocol is only applicable in the context of processing choice assays and includes the placement of a thin, straight, red marker between plants to facilitate separation of the plants. This is necessary as plants are grown in close proximity for these experiments, and overlaps of leaves complicates plant separation and may compromise algorithm performance.

The fourth and final part of the protocol involves some specific characteristics of the acquired images. The lens should be pointed approximately orthogonal to the plane of the plant or leaf being imaged in order to maximize the amount of leaf visualized. In addition, while different lighting conditions are tolerable, care should be taken to provide lighting that is as uniform as possible over the image and to minimize shadows in the image, especially shadows on or near the plant of interest. We also recommend that plants not be photographed in direct sunlight, as this tends to create large amounts of glare on many plant surfaces. If imaging in the field is being done, we advise the use of a white (or grayscale) umbrella to block the direct sunlight, which permits a sufficient amount of light for image capture but does not cause glare. Finally, if multiple

specimens are positioned in a single image, care should be taken to ensure separation of specimens, as well as separation from the color checker.

One very important thing to note about this protocol is that it facilitates *nondestructive* imaging of phenotypes, as the protocol was designed specifically to allow imaging of phenotypes in the field, lab, and greenhouse. Using this approach, many phenotypes can be imaged and quantified without having to destroy or damage the plant. This has the very beneficial side effect of allowing repeated measurements to be taken on the same plant, which can be useful for measuring changes in a specific plant's appearance over time.

2.4 Image Normalization Algorithm

A high-quality normalization algorithm that utilizes the color checker aspect of the protocol to transform images to a common baseline is a critical piece of the imaging protocol. Such an algorithm is essential for allowing phenotype imaging in the field, where lighting conditions change by the hour or minute (see Figure 3 for examples of the variation in lighting when imaging in the field.) It is also an important component when one considers the possibility of taking smaller datasets obtained by various research groups and attempting to merge those together for more large-scale analyses.

This research's approach for image normalization has three main steps: (1) color checker identification, (2) mapping of color wells, and (3) calculation and application of color channel transformations.

Identification of the color checker in each image is accomplished primarily using edge detection and a set of heuristics. The input image is first smoothed using a 5x5 pixel median filter to reduce noise. Edge detection using the Sobel operators [8] is

performed for each color channel independently, and an edge image is formed by combining the individual channel edge images (where values correspond to the magnitude of the Sobel operators) using a max operation. The Otsu threshold [23] is computed for the resultant edge image, and thresholding is applied to retain only the heaviest edges in the image. In a sufficiently focused image, the borders of each color well will form very distinct and strong edges against the black boundary of the color checker and will be retained with this step. After an 8-neighbor connected components algorithm [7] is applied, components are screened by a set of heuristics to determine candidates for the color checker wells. These heuristics include constraints on (1) "squareness" and relative size of the components, (2) compactness of groups of similarly sized, square-like components, and (3) color distribution of these component groups.

Once a set of components are determined as the set of viable candidates for the color checker wells, individual components must be correctly mapped to the corresponding color well. This step is accomplished by first calculating the representative color for each component, defined as the average RGB color in each well. The median RGB color could also be utilized as the representative, as averages are known to be affected by outliers such as pollen and other debris on specific wells of the color checker which may have some effect on the channel transformation coefficients. Then, the angular orientation of the color checker is computed utilizing the angles between a color well and its nearest neighbor. Components are next spatially organized into a 4x6 inch grid according to their locations. The highest valued component is selected as a reference point for finding the optimal mapping function ψ. A brute-force search is then conducted which identifies both ψ and the relative orientation of the color checker itself.

This is accomplished using the function (see Equation (1)) below, which minimizes mapping error.

$$\psi = \arg \min_{m \in f(\mathbb{N}) \to \mathbb{N}} \sum_{i=1}^{24} hueDiff(i, m(i)) \tag{1}$$

where

$$hueDiff(x, y) = \min\left(|h(x) - h(y)|, |360 - h(x) - h(y)|\right) \tag{2}$$

and $h(x)$ is the median hue value of the x^{th} component. The *hueDiff* function computes the smallest positive difference between the corresponding hue values. This function also takes into account the circular nature of the hue space in computing this difference.

Once these pieces of information are known, complete mapping of all the candidate components can be performed by utilizing ψ and the already computed spatial arrangement of the components. The algorithm is designed in such a way as to permit successful identification and mapping of the color checker even in cases where significant portions of the color checker are occluded. Our algorithmic requirement is a minimum of at least 5 of the 24 color wells to be represented by candidate components.

Finally, once color wells are mapped, appropriate transformations can be computed for normalizing the image. Preliminary data suggested a simple linear regression (see Equation (3)) using a least squares approach to be sufficient for normalization, and this was confirmed by experimentation.

$$y_i = \grave{x_i}\beta + \epsilon_i \tag{3}$$

where, in this case, $x_i^{`}$ is the matrix of RGB values for a particular color well, y_i is the matrix of true color values, and β is the optimal transformation matrix obtained by

$$\hat{\beta} = \arg\min_{b} \left(\sum_{i=1}^{n} (y_i - x_i^{`} b)^2 \right) \quad (4)$$

Representative color values for each well along with the corresponding true color values, provided by the color checker manufacturer are used to determine the best fit linear transformation for each channel. Color normalization is then accomplished by applying this transformation on a channel by channel basis to each pixel in the image.

The normalization algorithm not only includes a means to standardize the image with respect to color, but also to normalize the image with respect to size. The color checker provides the necessary information for this step, as each color well has known dimensions of 1 x 1 cm. Because the color checker can appear in any angular orientation in the image, the size normalization aspect of the algorithm must take this into account. Using the already computed angular orientation of the color checker, the image is rotated to align the color check along the standard x- and y-axes. The median lengths and widths of the components corresponding to the color wells in the rotated image are computed and used to calculate the number of pixels corresponding to a 1 cm^2 box in the area. This information can then be utilized to convert pixel counts of length or area into physical distances.

2.5 Results and Discussion

An experiment was first conducted to evaluate the color normalization aspect of the algorithm. Using a set of 60 randomly selected images from a maize leaf dataset, the

effect of normalization is shown visually and quantified. This dataset was selected because its comprising images were taken in the field in a variety of lighting conditions. The algorithm's ability or inability to standardize this set is a good indication of its performance.

The effect of visualization is first demonstrated visually. Figure 3 shows five images that represent the range of image penetration in the dataset, from slightly overexposed at the top left to increasing amounts of underexposure as one travels down the left column, as well as the net effect of the normalization algorithm on these images. The result images all appear to be at the same color levels, particularly when looking at the color checker in each image. To quantify the improvement, one can plot the values at each color well against the actual values for both the original and normalized image (see Figure 4 for a few examples). After compiling the differences between the color channel value in the normalized image and that of the expected value across the entire dataset, the average error is just under seven units. This takes into account several fairly severe underexposed images, which have higher errors, most typically in the red channel. Figure 5 illustrates this with one of the most underexposed images in the dataset. In this figure, both the green and blue channels are corrected well, but the red channel shows some larger deviations, as the normalized line (blue triangles) does not approach the green line (perfect normalization) as well. This is because values for the red channel cannot go below 0, as would be needed for the linear relationship. This truncation is seen in the bottom six values for the red channel and causes the error in the normalization of the red channel. This phenomenon generally only occurs in the red channel.

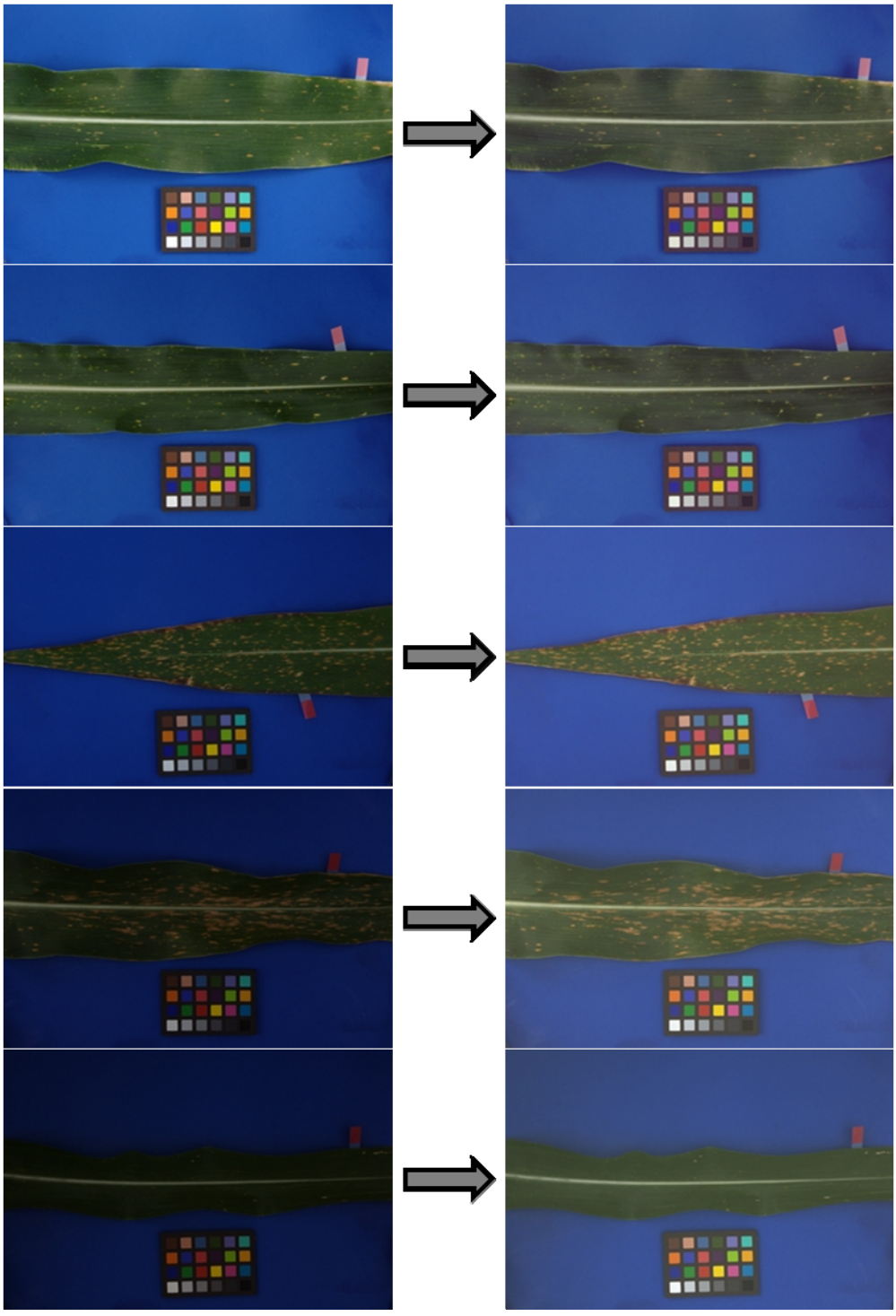


Figure 3: The left column shows a set of maize leaf images that were taken in field conditions that demonstrate the variability of image exposure due to lighting conditions. The right column shows the same images after the normalization algorithm is applied.

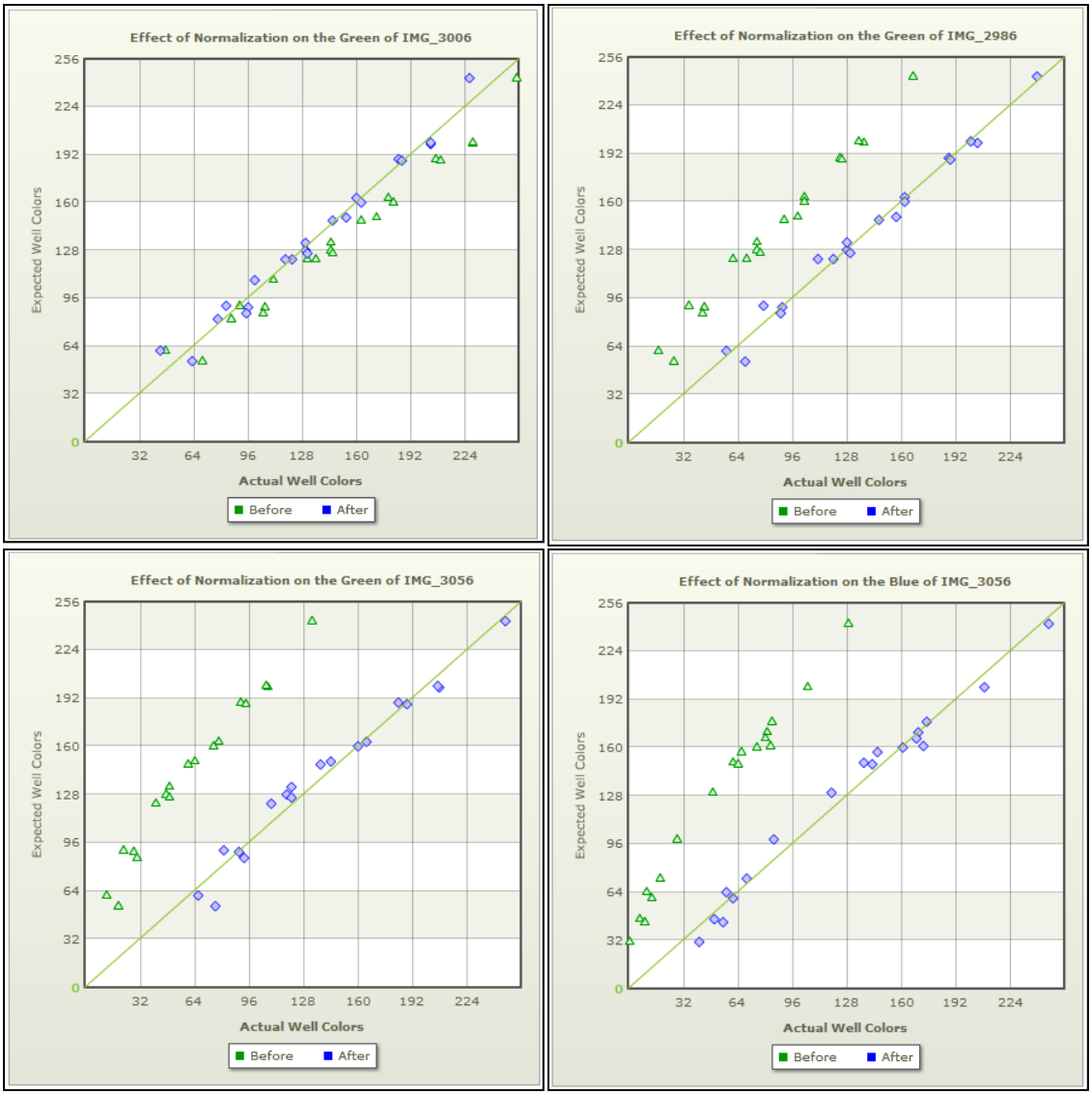


Figure 4: Effect of the image normalization for selected channels of the images from Figure 3, by plotting actual color well values against expected values for both original (green) and normalized (blue) images. The plots show corrections of the (top left) green channel for the overexposed image, (top right) green channel for a moderately underexposed image, and (bottom) green and blue channels for a more severely underexposed image.

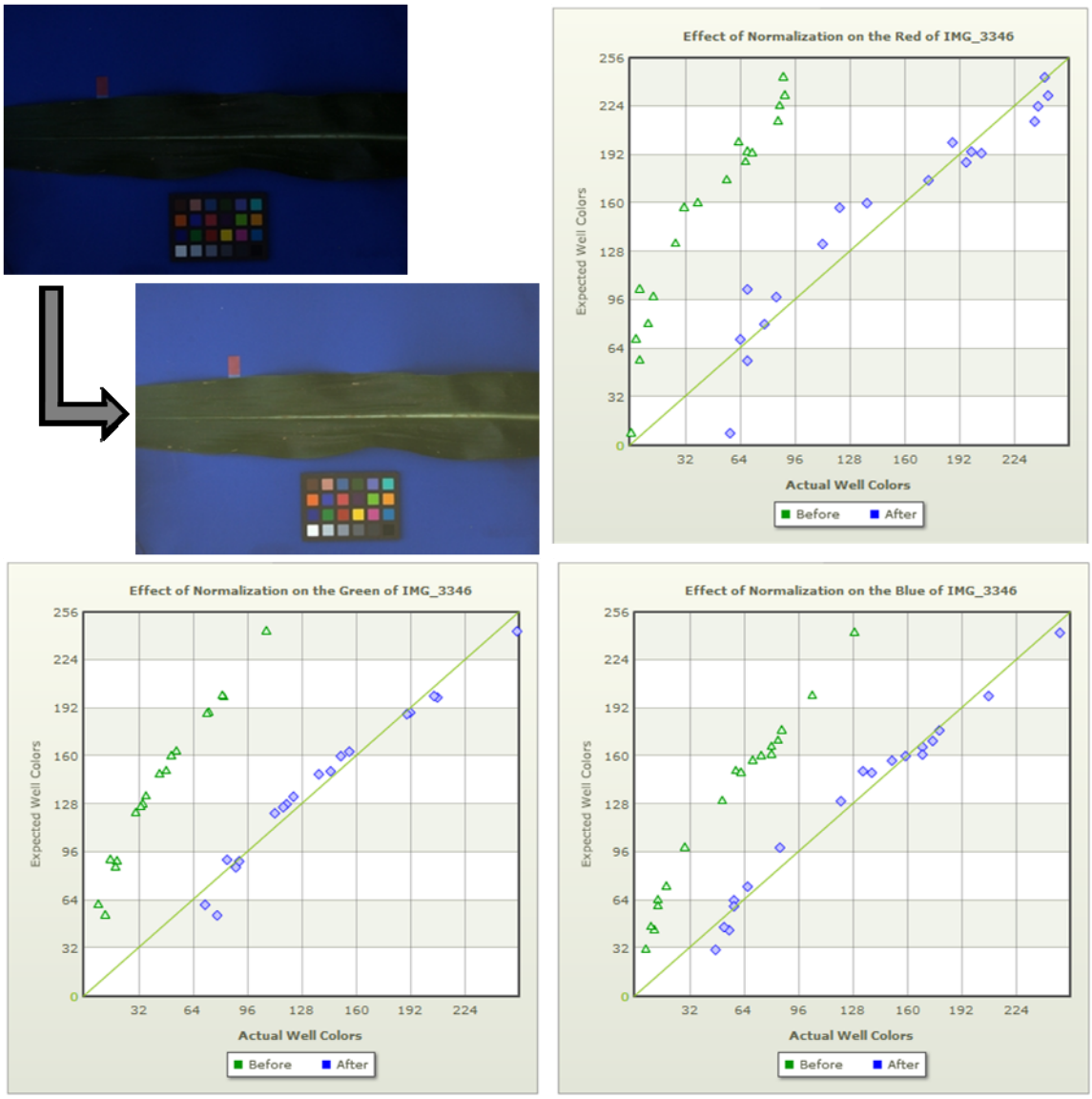


Figure 5: Effect on normalization on one of the most underexposed images in the test set. The top left shows the effect of normalization on the image itself. Plots of the actual versus expected color well values for the original and normalization images for the red channel (top right), green channel (bottom left), and blue channel (bottom right) are also shown.

Table 1: This table shows the correlation coefficients between the average error per channel in the original image and the average error per channel after normalization. These data suggest the average error in the red channel to be the most correlative to normalization quality.

		Final Error Average			
		Red	**Green**	**Blue**	**Combined**
Initial Average Error	**Red**	0.85	0.67	0.66	0.86
	Green	0.72	0.73	0.74	0.84
	Blue	0.48	0.78	0.81	0.76
	Combined	0.72	0.74	0.76	0.84

Since the quality of the normalization appears to depend on the magnitude of the difference between the expected color well values and those in the original image, we quantified the ability of certain initial errors to predict the quality of the normalization. The four factors utilized were the average deviation in color values between the original image and the expected image for the red channel, the green channel, the blue channel, and the average over all three channels. The correlation coefficients between these values and the average error in the normalized image are shown in Table 1. Because the red channel seems to be the most affected channel, these data suggest that the average error for the red channel is the best choice for predicting overall normalization quality.

2.6 Summary

In this chapter, a method was developed to allow for imaging of leaf phenotypes in the field and facilitate high-throughput processing of such data. To account for image variations due to field conditions, a robust normalization algorithm was designed to standardize imagery in terms of color and scale. The algorithm was shown to produce

quality results across a range of exposure levels, and a proxy for algorithm performance was found based on initial average error in the red channel of the image.

Though the color checker utilized by the protocol and algorithm contains 24 color wells, only a small subset of these were found to be necessary for obtaining good normalization results. As such, a smaller color reference with fewer color wells would suffice, so long as at least some of the color wells could be reliably detected despite the image exposure level. This would be particularly useful for the imaging of phenotypes on smaller leaves and other objects.

The ability to standardize phenotypic imagery in this way has a number of benefits. First, it permits flexible imaging of leaf phenotypes in a variety of environments (in the lab, greenhouse, or field; *in situ* or destructive). Second, it allows data (images of the same class of phenotype) from different experiments, locations, and research groups to be combined or compared, for more large-scale analysis of genetic, epigenetic, and environmental factors.

CHAPTER THREE

VISUAL PHENOTYPE QUANTIFICATION

Representation of visual phenotypes is an important consideration. While raw image data does indeed contain a large deal of useful information, the relevant information is often hidden, and image processing and pattern recognition techniques need to be employed to extract higher-level data.

As such, our approach to visual phenotype quantification involves building a digital signature (feature vector) for each phenotype image. From a computational standpoint, this involves two major steps: (1) segmentation of plant structures and (2) extraction of features that measure relevant characteristics of the phenotype. As our datasets involve phenotypes from maize leaves and leafs and rosettes from several other plant species (*Arabidopsis thaliana*, *Brassica*, cabbage, soybean), the particular segmentation methods and feature extraction techniques vary by dataset.

Following the construction of digital signatures describing phenotypic appearance, these features are oftentimes then utilized or combined in various ways to produce phenotypic measurements that are more akin to the information plant researchers typically use in their experiments. In this way, trait data based on digital signatures can be compared to data obtained from other known techniques, oftentimes noncomputational methods, for generating phenotypic scores.

In this chapter, a review of phenotyping techniques and a survey of general techniques for segmentation and feature extraction are provided. We then demonstrate two pipelines developed to convert phenotype images into digital signatures as well as

the methods for conversion of extracted features into phenotype scores. Evaluation of these methods follows.

3.1 Literature Review

3.1.1 Plant Phenotyping and Scoring

Plant phenotyping has been ongoing for decades with researchers developing a myriad of methods of varying accuracy, precision, and efficiency for measuring phenotypic characteristics both manually and computationally. Oftentimes, especially in this age of genetics where experiments based on forward or reverse genetics are commonplace, the ability to measure or characterize changes in phenotype is critical in order to elucidate and understand gene function. The ease with which phenotypic measurement can be performed is directly related to the phenotype being measured and the desired precision of measurement. For many phenotypes (e.g. height, flowering time, survivorship), trait measurement is simple either because identification is easy or measurement is straightforward. However, there are other phenotypic traits for which measurement of phenotypic change is more difficult, especially if precise quantification is desired. This may be because even though the observable change is obvious, there is no direct or easy means of accurately measuring the phenotype, e.g., significant plant growth or severe infection of a leaf. Alternatively, quantifying phenotypic change may be complicated by the fact that many mutants have no observable change or, at best, only very minute changes in phenotype from a wild type plant, which requires more sensitive analyses of growth, color change, or defense responses.

While there has been development of some professional equipment that facilitates some automated phenotypic quantification (such as LemnaTec's Arabidopsis

Morphological Phenotyping Assessment system which is currently available for purchase and performs some of these analyses), these are not cost-effective solutions for most research budgets, and so many researchers must look elsewhere for methods that are less expensive but still provide accurate measures of these more difficult to quantify phenotypes. For example, Boyes *et al.* developed a robust, standardized method for analyzing phenotypes based on growth stages [24]. Their method takes into account germination time, days until full cotyledon development, and flowering time, among many other parameters. In combination with genomic information, the authors suggest this method can provide a rapid protocol for identifying gene function in Arabidopsis. However, this method is limited to "standard environmental conditions" and was designed only for measuring growth phenotypes in Arabidopsis plants growing on Murashige & Skoog (MS) media in plates, thus making phenotypic evaluation of growth and many biotic stresses in soil and in field conditions impractical.

In recent years, several computational systems have been developed in an attempt to improve the accuracy, precision, consistency, and automation of plant phenotypic measurement. Primarily, these systems are restricted to particular plant structures. For example, several methods [12-13, 25] for imaging root systems and quantifying associated traits have been recently developed. Iyer-Pascuzzi *et al.* developed an imaging platform and analysis application for measuring plant root systems [12]. Their system nondestructively images plant root systems grown in cylinders and, with the use of a turntable mechanism, makes a 3D reconstruction of the root system. Using a similar setup in conjunction with the custom software RootReader3D, Clark et al. produced a high-throughput phenotyping system for root traits during seedling

development [13]. Fang *et al.* utilized a 3D laser scanner to make high quality reconstructions of maize and soybean root systems grown in transparent gel [25].

In addition to software on root systems, leaf traits, specifically leaf shape, has received a considerable amount of attention from the computational community with several systems developed, including the LeafAnalyser program, which primarily uses principal component analysis (PCA) to automatically determine leaf shape variation [15]; LAMINA, a tool that provides automatic measurement of a variety of characteristics related to leaf size and shape [14]; and LeafProcessor, an application that semi-automatically supplies a number of single-metric parameters and PCA analysis for leaf size and shape including contour bending energy [16]. While all these methods provide accurate and useful methods and are applicable to leaves of various plant species, they all have limitations. One of the most significant limitations for these approaches is the restrictions on the type of images that can be processed. Though LeafAnalyser and LAMINA utilize only scanned images and LeafProcessor relies on digital photography, all methods require that leaves be harvested. Would these methods provide equivalent results when leaves are imaged *in situ* using cameras in the greenhouse or in field conditions?

Another option for computational measurement of many phenotypes, including leaf area, is ImageJ, a popular utility available from the National Institutes of Health [26]. Though this utility is quite powerful and extensible, a nontrivial amount of image processing and programming background is required to (1) determine the functions and parameters needed to obtain the desired phenotype measurements and (2) write a macro or script to automate the processing pipeline for batch processing.

When these approaches are able to directly quantify a phenotypic trait, the measurements are immediately useful; however, for many traits (e.g. leaf shape and root systems) most systems will generate a number of features that correspond to indirect measurements of the trait. In this situation, the difficulty lies in deciding which features to use and also the best way to utilize those features.

Oftentimes, for mutant phenotypes that cannot be directly measured, identification and characterization of these phenotypes is based on subjective observation [2-4], especially in cases when no software exists to measure the phenotype indirectly. In many of these types of studies, the phenotypes are obvious and can be reduced to binary (yes/no) phenotypes that correspond to easy-to-detect changes in pigmentation or survivorship. For example, to assess the function of the gene MYC2/JIN1 in Arabidopsis, in [4] Dombrecht *et al.* sprayed methyl viologen onto jin1 and 35S:JIN1 plants. Because this compound induces oxidative stress on the plants, the authors could visually measure browning and plant death among mutants versus wild type plants. Repression of a transcription factor, PAP1, controlling the phenylpropanoid pathway resulted in an evident attenuation of anthocyanin pigmentation in Arabidopsis seedlings (see Figure 3B in [3]). These results show an observable difference in purple coloring and did not require additional colorimetric methods to confirm the findings. This is an example where the lack of coloring generates a “yes” response indicating a phenotypic difference.

While identifying binary (yes/no) phenotypes is relatively simple, it can be considerably more difficult to compare mutants where the resulting growth, color change, or defense response is subtle or ranges in scale. In these cases, phenotype

quantification is often performed using rubrics of some kind. For example, pathogenesis can be evaluated using infection scores [27], which can introduce error into the measurements, especially if multiple people are scoring or if a sample does not easily fit into one of the predefined scores. Of course, it is likely that some amount of error will be introduced in any situation in which phenotype quantification is dependent upon human perception and consistency. While rubrics are the cheapest and easiest method of quantifying these phenotypes, alternatives do exist. One could also measure pathogenesis by grinding plant tissues and doing time-intensive serial dilutions to determine the number of Colony Forming Units (CFUs) of bacteria in each sample [28-29]. One may also consider harvesting tissue for pathogen analysis; however, this can introduce an additional wounding treatment which would render the plant inadequate for further measurements such as flowering time, inflorescence production, or seed set.

3.1.2 Relevant Image Processing Techniques

A number of methods have been developed by the image processing and computer vision community that can be utilized to help process visual phenotype images. This specifically includes techniques for separating objects of interest from the background (called segmentation) and for extracting relevant information from the images. In this section, we first review the basic popular techniques for performing each of these tasks.

3.1.2.1 Object Segmentation

Image segmentation seeks to divide an image into disjoint and homogeneous regions [30], and is considered one of the most challenging tasks in image processing [7]. It is typically the first step for any computational approach that aims to process an image and produce features or measurements from objects within the image. Image segmentation

is largely considered one of the most critical steps because it determines the quality of the final image analysis results [31]. Classic segmentation approaches operate on monochrome or grayscale images and can be divided into two main classes: 1) those that create divisions in the image based on quick changes in intensity (discontinuity approaches) and 2) those that build regions by adding pixels with similar characteristics (similarity approaches). The vast majority of approaches for segmenting color images represent variations and extensions of these monochrome approaches with detailed explanations and examples of each type of segmentation found in [32]. As such, the popular monochrome segmentation approaches are reviewed.

One of the oldest and most popular techniques for segmentation is known as thresholding [32], which operates on the assumption that objects within an image will contain pixels of similar intensities. Approaches under the thresholding umbrella may be classified into a number of categories including global, where a single threshold is used for the entire image, and local or adaptive, where the images are divided into sections with a threshold determined for each section. In any case, a histogram is constructed based on intensity values for an image or region, and an appropriate threshold is determined for highlighting objects of interest. These thresholds can be calculated from training examples (supervised approach), may be empirically determined, or may be computed using only the information in the image (unsupervised approach). The unsupervised methods can be thought of as classification problems, attempting to optimally separate foreground pixels from background pixels. Several such approaches have been proposed that assume underlying distribution models [33-35]

or rely on entropy measures [36-38]. Otsu minimized the in-class variance while maximizing the between-class variance [23].

Whereas thresholding methods group pixels by similarlity, segmentation is also often performed by identifying areas of discontinuity. One of the most popular ways for finding these areas is through edge detection. This segmentation technique relies on an assumption that the boundaries of objects of interest in an image will correspond to sharp changes in pixel values, i.e., that there is a detectable difference in adjacent background and object pixels. Many edge detection filters are kernel-based methods that rely on calculating discrete derivatives, which include the Roberts, Sobel, Prewitt, and Laplace operators [7]. While these methods do find pixels near areas of discontinuity, they do at times generate spurious edges that can mistakenly identify isolated points as edges, and they have no means to assess whether areas with high signal actually correspond to real edges in the image. For this reason, Canny devised an edge detector that built upon these primitive edge detection filters to remove likely false edges and reduce thick edges to thinner ones [39]. The Canny edge detector also attempts to grow existing edges using a region growing process called hysteresis.

3.1.2.2 Feature Extraction

In many pattern recognition, image processing, and information retrieval systems, feature extraction plays an important role. With regard to images, this process involves extracting meaningful measurements (or features) from an image in order to produce a more compact representation or image signature. The difficulty lies in choosing features that actually extract the relevant information found in the image. Often, features are chosen to mirror human reasoning about the image data. While raw image

input (image pixel values in this case) can be used for many machine learning tasks, the use of higher-level feature descriptors derived from the low-level data often results in better performance [40]. This is because the more useful higher-level information often lies hidden in massive amounts of image data, and computational techniques to identify and aggregate that information into feature descriptors are required. While there are a large number of measurements or features that can be utilized, the most appropriate ones are dependent on the particular domain and task. Common types of measurements commonly extracted from imagery include those related to color and texture.

Spectral (color) characteristics are likely the most common type of feature extracted from imagery. This is partly because we as humans routinely utilize colors or variations in colors to describe and differentiate objects. Furthermore, color features are useful because they are invariant to translation and rotation and are relatively insensitive to changes in scale and occlusion [41]. As such, a more compact representation of the color in an image or a region of interest is often of great use.

In the image realm, colors are represented according to a particular color model. The Red Green Blue (RGB) color space, a three-channel model, is the most widely used with each pixel represented by the triple {R,G,B} where each position corresponds to the intensity value for that color channel. While formulating specific colors as combinations of these three color channels can be difficult for humans, the rationale for this color space stems from the way color vision actually functions in humans, where a given color is formed by a mixture of responses from three types of photoreceptors that pick up red, green, and blue light. Alternative color space models also exist, notably the Hue Saturation Value (HSV) color space. In this space, the hue represents the base

colors (blue, yellow, etc.), saturation represents the strength of that color (how pale or vibrant a color is), and value represents lightness or darkness of the color. Where the RGB color space resembles the physiology of how color vision works in humans, the HSV color space more closely resembles the way humans perceive and reason about color, and for this reason, it is quite useful in representing spectral characteristics in images.

The most common representation for spectral characteristics is that of the histogram which measures the global distribution of colors in the image. Construction of histograms is typically performed by discretizing the color space into bins and determining the frequency of each bin in the image [41]. For multi-channel images (which includes color images), this can be done as a single histogram, where each position corresponds to a discrete combination of intensities from each channel or by constructing histograms for each channel. In this way, an RGB image would result in three histograms, one for each of the red, green, and blue channels. Though several methods for binning are available, two popular methods include equi-width binning in which the width of each bin is constant, and equi-depth binning in which the same proportion of data falls into each bin. The most appropriate binning relies heavily on the distribution of the data in the histogram.

Texture features are also common in image processing and information retrieval systems. These measures attempt to represent and quantify the characteristics of the subpatterns or regularities that together form a more complex visual pattern [42]. The most commonly used classes of texture analysis techniques are: (1) statistical, (2) transform, and (3) model-based.

Statistical methods, representing the earliest work on texture and containing some of the most commonly used techniques, rely on correlations and relationships between intensity values. The most notable and popular method for texture analysis is a method by Haralick that utilizes second-order statistics computed from a co-occurrence matrix [43]. With this method, a co-occurrence matrix is constructed by iterating over all pairs of pixels with a specified relative position and computing the frequency of co-occurring intensity values. Relative positions are often specified as linear distances in specified directions. For rotational invariance, the signals from each direction may be combined together.

Instead of utilizing co-occurrence values of various gray levels, model-based approaches were developed to mathematically model the distribution of gray levels, most commonly using a stochastic or fractal model. These methods notably include those based on Gaussian Markov random fields [44] and Gibbs distribution [45].

Transform-based techniques approach the texture analysis problem by converting the image data to a different coordinate system in which periodic signal patterns can be more easily detected. Examples of texture analysis methods in this category include wavelet [46], Gabor [47], and Fourier transforms [48].

3.2 Processing Pipelines and Phenotype Quantification

To aid in the development of discriminate features for each plant structure in our datasets, domain experts were consulted in an attempt to define the visual properties that were most important or characteristic of the particular phenotypes studied. Features were then defined to directly measure the important characteristics of the phenotype where possible. In cases where direct measurement was not possible or unclear, indirect

measures were utilized. After appropriate and interesting features were determined for each set of plant structures, algorithms were developed to extract these features, and there were packaged into processing pipelines. Initial development was based on datasets from leaves in *Zea mays* (maize), with application of similar techniques applied later to other species and plant structures. A discussion of developed processing pipelines follows.

3.2.1 Rosettes/Leaves of Other Species

One processing pipeline this research developed was designed for phenotyping traits related to leaf and rosette area, and this has been used in several plant species including *Arabidopsis thaliana, Brassica rapa,* and *Glycine max*. These datasets all involved capture using a digital camera; some of the data represents nondestructive capture of the phenotype, while others involved leaf harvesting.

Images submitted to this pipeline are required to have a color checker in the field of view for normalization purposes, a light background (white), and physical separation of leaves or plants, if multiple specimens are captured simultaneously. Processing proceeds as illustrated in Figure 6, which also includes example images to demonstrate image transformation at each point in the process.

The pipeline includes four steps:

1) **Image Normalization:** This step is performed to both determine physical pixel distance and to adjust color profiles based on lighting and image exposure;

2) **Specimen Separation:** Various specimens present in the image are segmented utilizing Otsu threshold method [23] and connected components [7]. The image is partitioned into individual pots, leaves, etc.

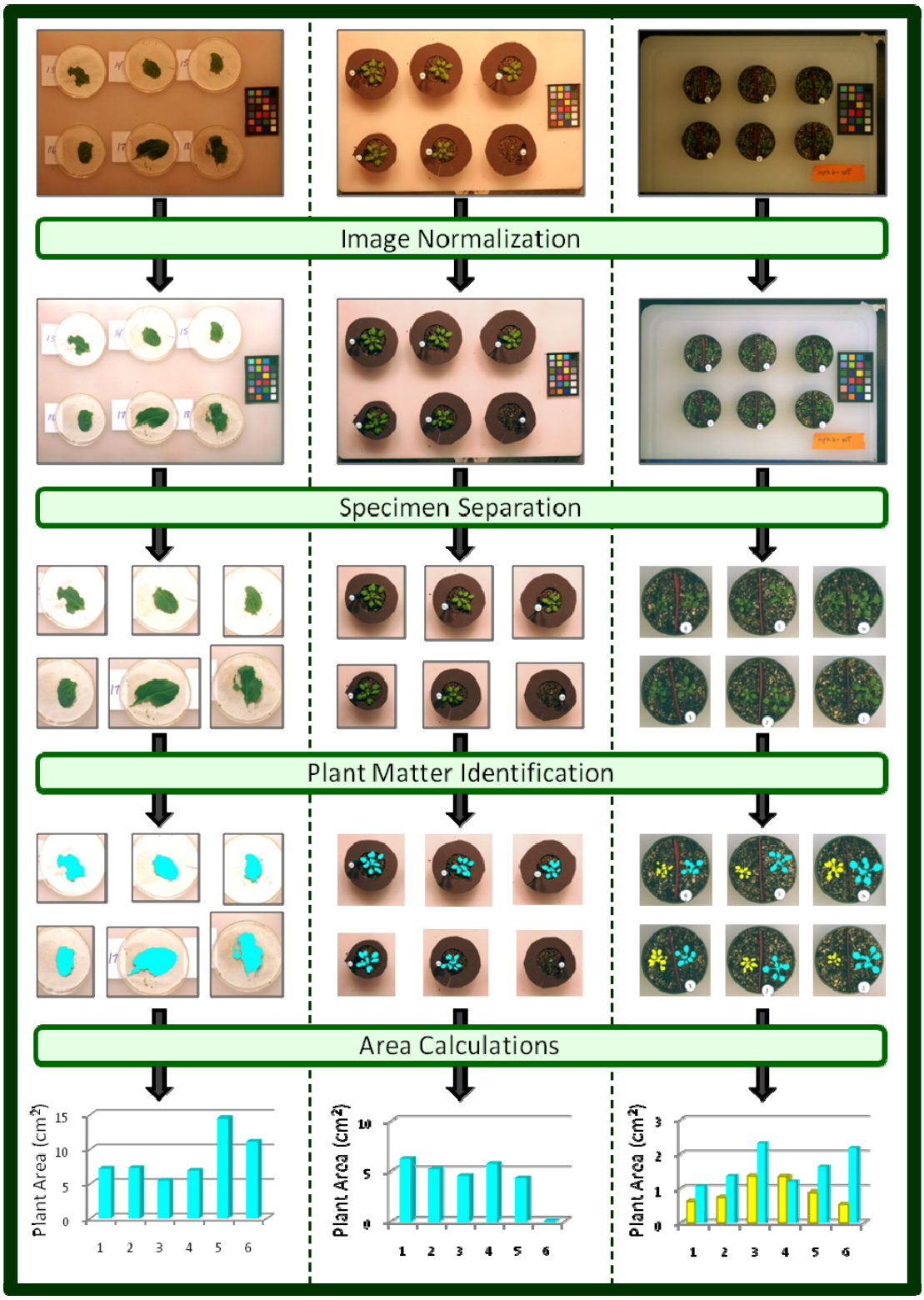


Figure 6: Processing pipeline for the calculation of area, herbivory, and growth of leaves and rosettes.

3) **Plant Matter Identification:** For each specimen in the image, plant matter identification is then accomplished by means of HSV color thresholds, which may be adjusted by users based on experimental needs.

4) Area Calculations: This step corresponds to the feature extraction step with the only feature extracted being the area of the plant.

This pipeline also supports the processing of "choice assays" (see the examples on the right in Figure 6) which involves two plants being grown in the same pot. The protocol for this situation has an additional stipulation that involves placement of a straight red marker in the pot to separate the plants.

After the images have been normalized and appropriately segmented, features are extracted for each plant. In this case, because we are interested in plant area traits, the only feature that is extracted is the area of the plant, which is simply a count of the "plant" pixels, which again are determined by user-defined HSV thresholds.

3.2.2 Maize Leaves

The other major processing pipeline was developed for maize leaves, particularly those with lesion phenotypes, which represent considerably more difficult phenotypes to quantify, and the majority of our datasets fall into this category. These datasets can be divided into two categories. First, there are sets of images of leaves from maize plants inoculated with Southern Leaf Blight (SLB), a fungal disease that affects crop yield [49-50]. The phenotypes of these images are various degrees of necrotic (brown) lesions. Second, there are set of images from various lesion mimic mutants (*les* mutants) or other mutants that invoke responses from the programmed cell death pathway in maize. These phenotypes consist of varying combinations of chlorotic and necrotic lesions. The vast

majority of images from both sets were photographed *in situ* in the field, though there are some sets that involve harvesting leaves from plants to either image with a camera or flatbed scanner.

Processing of these maize leaves is significantly more complicated primarily because imaging is conducted in the field instead of a controlled environment. For this reason, there is a much higher degree of variability in the lighting and exposure levels in the images. Also, since imaging is done *in situ*, it is more difficult to take the pictures due to the presence of other leaves and plants. The waviness of the leaf also makes imaging, specifically obtaining focus throughout the field of view, more difficult. The processing flow for the maize leaves is shown in Figure 7 along with example segmentation results. The first processing step is to normalize the leaf images with respect to color and lighting utilizing the algorithm discussed in Section 2.4. Again, placement of a color checker in the field of view facilitates this step. Following normalization, segmentation of the leaf is performed. This is accomplished utilizing a thresholding approach. Knowledge of the background color is required to properly set the threshold values. Because the expression of both SLB and lesion mimic datasets involves the appearance of various lesions on the leaf surface, a second round of segmentation is performed in an attempt to extract these lesions. Necrotic lesions can be extracted with good accuracy using appropriate thresholds; however, the extraction of chlorotic lesions is a much more challenging segmentation task and is still an open research problem, as oftentimes these lesions appear only subtly different from normal tissue and lesion boundaries are often gradual and not distinct. Empirically determined spectral thresholds are used for this purpose.

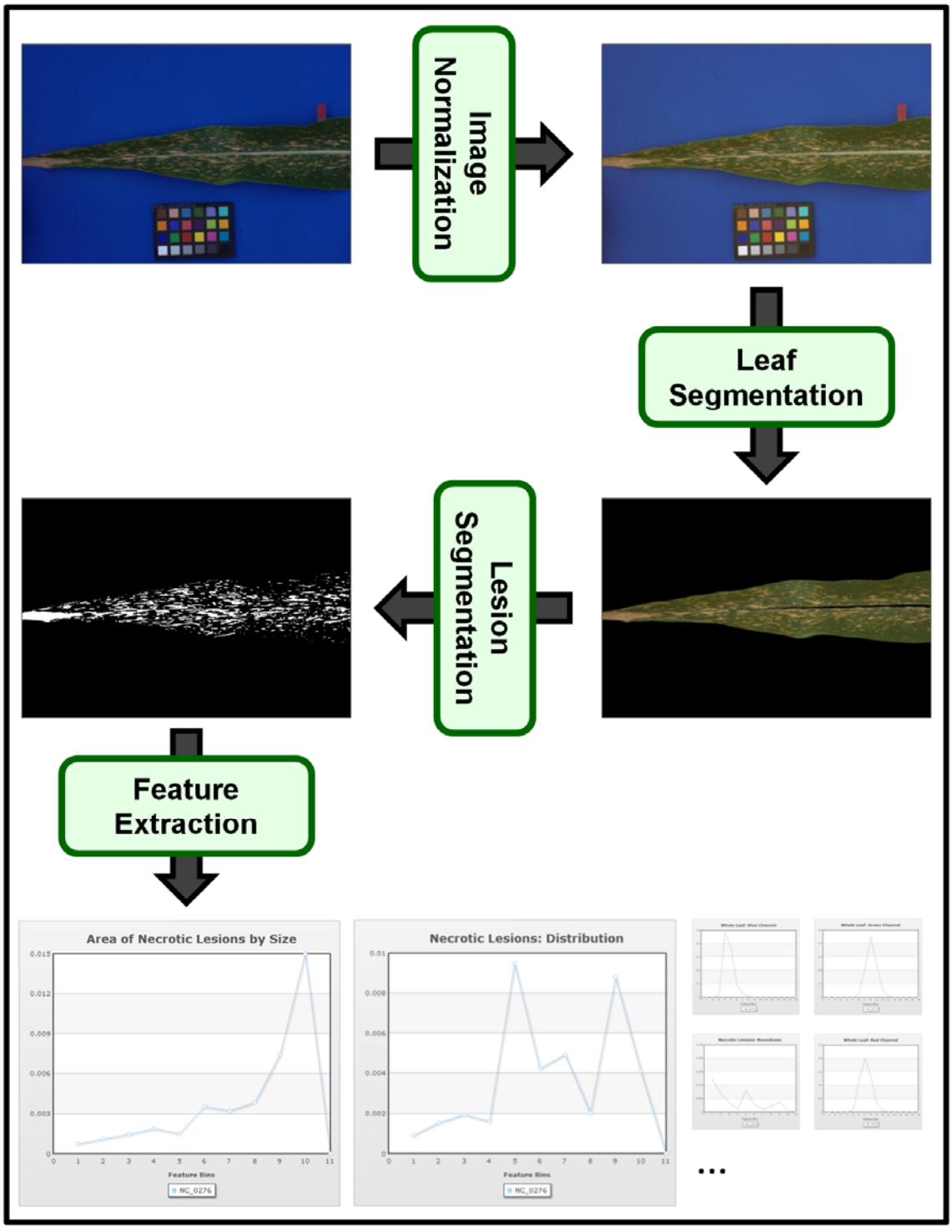


Figure 7: Processing pipeline for maize leaves. Images are first normalized, then the leaf and lesions are segmented, and finally a number of features are extracted.

Once the leaf and lesions are extracted, a leaf image progresses to feature extraction. In the case of SLB in which the trait being measured is disease resistance, there is no means to directly measure this characteristic, and so several indirect measures are utilized instead. Similarly for lesion burden in lesion mutants, there is no means to directly measure the phenotype and so various indirect measures are used. After careful study and categorization of descriptions of maize *les* mutants and corn afflicted with SLB and after discussion with maize experts, the prevailing characteristics in these images were determined to be color, size, shape, and distribution. With these types of mutants and diseases, color can be used to separate normal tissue from abnormal tissue (lesions) and, furthermore, can be used to classify lesions as necrotic or chlorotic, which can be helpful in differentiating between mutants since different mutants produce different color lesions. Lesion size can also be an important indicator, as larger lesions tend to translate to a more severe infection in SLB and since different mutants produce different sized lesions. With SLB and most *les* mutants, lesions usually begin as small circles and grow outward. More advanced expression or infection is oftentimes associated with the merging of expanding lesions. For this reason, a measure of lesion size can also help to determine severity of infection or categorize expression. Finally, various mutants are known to have different distributions of lesions; some may produce very sparse lesions, while others have dense lesion coverage or clustered lesions. Thus, some measure of lesion distribution should also be measured.

Table 2: List of feature ranges collected for maize leaves.

Features	Feature Class	Utilized Pixels	Number of Features
001 - 048	RGB histogram	Entire leaf	48 (16, 16, 16)
049 - 096	RGB histogram	Necrotic-like	48 (16, 16, 16)
097 - 144	RGB histogram	Chlorotic-like	48 (16, 16, 16)
145 - 192	RGB histogram	Non-lesions	48 (16, 16, 16)
193 - 239	HSV histogram	Entire leaf	47 (15, 16, 16)
240 - 286	HSV histogram	Necrotic-like	47 (15, 16, 16)
287 - 333	HSV histogram	Non-lesions	47 (15, 16, 16)
334 - 380	HSV histogram	Chlorotic-like	47 (15, 16, 16)
381 - 392	Size histogram	Necrotic-like	12
393 - 404	Size histogram	Chlorotic-like	12
405 - 416	Roundness histogram	Necrotic-like	12
417 - 428	Roundness histogram	Chlorotic-like	12
429 - 440	NN distance histogram	Necrotic-like	12
441 - 452	NN distance histogram	Chlorotic-like	12
453 - 537	Texture measures	Entire leaf	84

Each of these characteristics is represented in the leaf image's digital signature (see Table 2). First, spectral features in the form of histograms are extracted both for the entire leaf as well as for those pixels identified as part of lesions. These histograms are computed on a channel-by-channel basis using 16 bins per histogram. Following color are several sets of features aggregated from measurements on individual lesions. Regarding lesion shape, the roundness of each extracted lesion is computed using Equation (5),

$$roundness = \frac{lesion\ area}{\pi\left(\hat{d}/2\right)^2} \tag{5}$$

where *lesion area* is the number of pixels in the segmented lesion and $\hat{d}$ is the maximum diameter of the lesion. All the roundness measures are combined into a single histogram for the leaf, and then converted to a 10-bin equi-width histogram of lesion roundness for a describing lesion shape on the leaf. A 10-bin equi-width histogram of lesion size is also computed. To quantify the density of lesions on the leaf, a measure called the nearest neighbor distance is computed. In this measure, the distance from the center of one lesion to the center of its nearest neighbor is computed, as in Equation (6) below,

$$NNdistance(i) = \min_{1 \le i \ne j \le N} \left[\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right] \quad (6)$$

where i is the index of the lesion for which the nearest neighbor is being computed, N is the total number of lesions on the leaf, and x_k and y_k are the *x* and *y* coordinates of the center of lesion k. Compiling these distances for every individual lesion in the image generates a histogram, which is then partitioned into 12 equi-depth bins.

In addition, because some mutants contain tiny lesions that appear densely and diffusely across the entire surface of the leaf, individual segmentation of these lesions is intractable. In order to provide features that can help to measure this appearance, various texture features from second-order statistics [43] (shown in Equations (7)-(12) below) are computed.

Entropy $$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j}) \quad (7)$$

Contrast $$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \quad (8)$$

Dissimilarity $$\sum_{i,j=0}^{N-1} P_{i,j}|i-j| \quad (9)$$

Homogeneity $$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (10)$$

Uniformity $$\sum_{i,j=0}^{N-1} {P_{i,j}}^2 \quad (11)$$

Energy $$\sqrt{Uniformity} = \sqrt{\sum_{i,j=0}^{N-1} {P_{i,j}}^2} \quad (12)$$

3.3 Computational Scoring

Once features are extracted, oftentimes additional work must be done to convert them into meaningful phenotype scores. In the case of the pipelines described in the previous section, the pixel areas may need to be converted to herbivory or growth values and the large number of features extracted for measuring disease resistance to SLB need

to translated into a phenotype score that can be used for analyses like QTL mapping. The method for accomplishing this is dependent on the phenotype but in general can be divided into two classes. First, methods can attempt to learn rubric scores or other measurements through associations between the extracted low-level features and training examples provided by researchers (supervised approach), or phenotype scores can utilize the features exclusively to generate measurements (unsupervised approach). Here, the discussion of our computational scoring is divided by trait: A simple method for area-based traits (e.g. area, herbivory, and growth) is first described, followed by two methods for utilizing the features extracted for lesion burden in SLB to determine a disease resistance score.

3.3.1 Scoring of Traits Related to Leaf Area

The measurement of whole or partial leaf area is a relatively simple trait to obtain; however, this feature can also be utilized to construct more complicated and scientifically more useful quantities. Two such examples are herbivory (the amount of plant matter consumed by an herbivore during an allotted feeding period) and growth (the increase in plant size over time). Because the more classic and traditional means of estimating growth and herbivory (rubrics, manual estimation, etc) are slow, laborious, and error-prone, and the more sophisticated methods are typically quite expensive, this method is of interest to plant scientists as it provides a less expensive option for high-throughput and accurate measurement of these quantities.

The conversion of the extracted feature (area of the specimen in terms of a raw number of pixels) into these more useful quantities is relatively straightforward but can be dependent on the experimental parameters.

Once leaf or rosette areas are extracted from imagery, the next step to translating these values to growth and herbivory measures is conversion of the pixel count features to area measurements in physical units (such as cm^2). This is done not only because researchers prefer to work with data in standard units, but more importantly because the pixel counts may come from images at different scales (perhaps if there is a difference in the height at which the images were taken) and thus are not directly comparable unless they are standardized. This conversion to physical units can be accomplished by utilizing the normalization factor, i.e., the *normFactor*, that is recorded for each image according to Equation (13).

$$areaCM = \frac{pixelCount}{(normFactor * 100)^2} \tag{13}$$

where $pixelCount$ is the raw pixel count from an image and $area$CM is the translated area in cm^2.

Let C_i where $i = 1,2, \ldots N$ be a set of N images of the same specimen arranged chronologically from the earliest image (i=1) to the latest image (i=N). The case of N=1 corresponds to no temporal tracking of the specimen, and the pixel count area can be transformed to a physical area as discussed above. When N >1, however, temporal progression of the leaf area can be computed and tracked. The simplest situation is a growth situation, in which no external force is removing plant matter between time points. In this case, a simple subtraction of normalized areas can be computed to determine the change in plant area.

While a simple subtraction of areas can also be done with herbivory experiments (N=2), the difference in areas may not correspond precisely to the amount of plant matter eaten or removed by an organism. The reported difference will actually be the

combination of (1) plant growth over the interval between the before and after images and (2) the amount of plant eaten or removed by the herbivore. If the feeding interval is short, plant growth may be negligible, in which case it can be safely ignored. However, for experiments with longer feeding periods, herbivory calculations must take into account plant growth.

The typical method utilized to account for such growth is the use of a second set of plants to serve as growth controls. In this case, the amount of growth (however it is measured or estimated) in the control plant can be utilized in the experiment to represent the growth that should have taken place in the experimental plant. While this is an acceptable solution, it is not ideal, as there is no guarantee that the plant in the experiment would have grown at the same rate as the control plant(s) since there are natural variations in growth curves within a particular genotype.

The ability to accurately track individual growth over time, however, has the added potential to offer a more elegant and possibly more accurate solution. If plant growth is consistently tracked from germination, the data could be used to model a growth curve for the plant. If the model were sufficiently accurate, control plants would not be necessary, as corrections for growth could be made by projecting the area that plant would have had if the insect had not been permitted to feed. Let a_1 be the area of the plant at the before image, a_2 be the area of the plant at the after image, and $\hat{a}_2$ be the projected area of the plant due to the modeled growth curve had herbivory not occurred. In this situation, the following equation would be used to calculate herbivory.

$$
\begin{aligned}
herbivory &= projectedGrowth - actualGrowth \\
&= (\hat{a}_2 - a_1) - (a_2 - a_1) \\
&= (\hat{a}_2 - a_2)
\end{aligned}
\tag{14}
$$

3.3.2 Scoring of Disease Resistance

The scoring of a trait like disease resistance to Southern Leaf Blight in maize is much more complicated than scoring herbivory or growth. Because disease resistance or susceptibility cannot be directly measured, common phenotype scoring for a trait such as this is conducted with the use of a scoring rubric. With these methods, one or more individuals navigate through the field examining the severity of disease one row of plants at a time by assigning the numeric score whose corresponding textual description appears to have the best match to the observed infection.

The conversion of the extracted features for the maize leaves to a score of disease resistance is more difficult and involved than that discussed in the previous section. This is because one must use indirect evidence to infer the severity of the infection. Instead of just a simple subtraction of normalized areas, more sophisticated approaches to combine and fuse the information in the features must be used. Two different approaches have been developed for scoring disease resistance.

The first approach for scoring disease resistance is a supervised approach in which the plant researcher would utilize a scoring rubric to assign numeric scores to a small subset of images. Using these scored examples, the system can learn the scoring patterns of the researcher and compute a score for an unscored image, which should result in more objective and consistent scoring. This is accomplished by utilizing

content-based image retrieval (CBIR) [51] techniques to find the *n* most visually similar images from the training set, i.e., those training images whose feature values best match those of the unscored image (see Figure 8, where *n=2*).

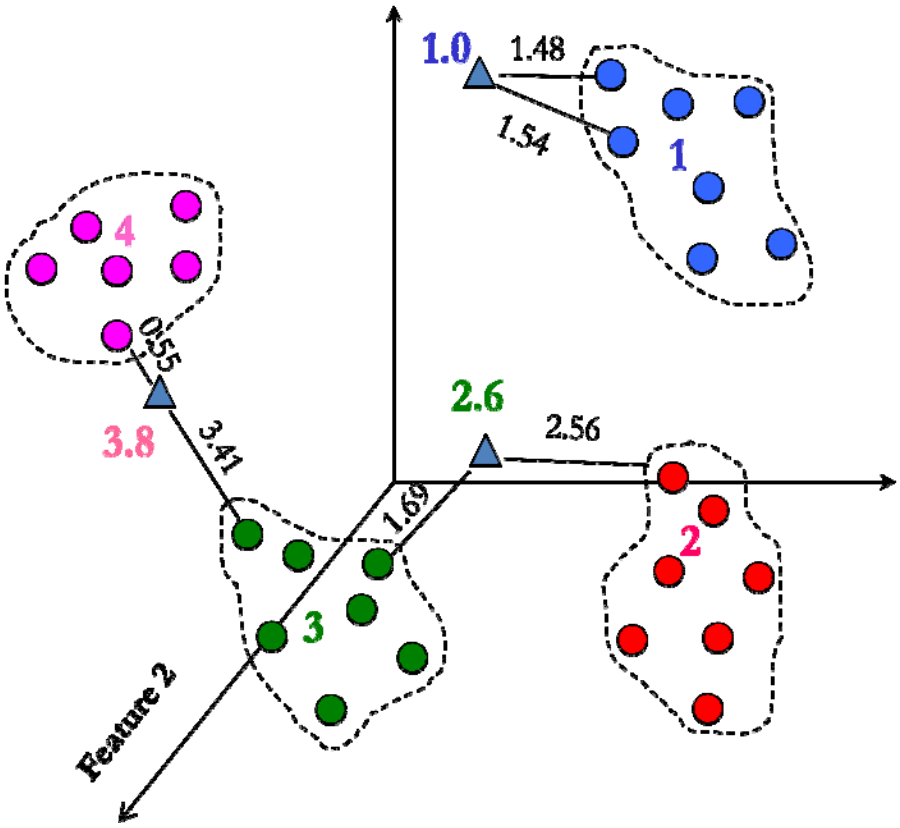


Figure 8: Example scenario for our supervised method of calculating lesion burden scores (with *n=2*). The colored circles represent images already scored by a domain expert. The blue triangles represent the query images. The scoring is computed using Equation (15) on the *n* nearest neighbors.

A score for the unlabeled image can then be computed by utilizing the scores of the retrieved images as well as their distances (in feature space) from the unscored image using the formula below:

$$score(q) = \frac{\sum_{i=1}^{n} score(t_i) * \left[\frac{1}{dist(q,t_i)}\right]}{\sum_{i=1}^{n} \left[\frac{1}{dist(q,t_i)}\right]} \quad (15)$$

where q is the unscored image, t_i is the i-th most visually similar image in the training set, and $dist(x, y)$ is the Euclidean distance between images x and y in feature space. The inverse distance is utilized to bias the score being computed towards the scores of the closer scored images in feature space. For best results, there should exist at least *n* images indexed for each score; otherwise, a computed score may be skewed due to sparseness in the feature space in the vicinity of the unscored image.

In addition to this supervised approach, an unsupervised approach can also be taken. The simplest unsupervised approach would be the direct utilization of an individual feature as the phenotype score. The most natural feature in our set to utilize for this unsupervised scoring is the percentage of the leaf covered by necrotic lesions, as it makes sense that the more covered the leaf is in necrotic lesions, the more severe the infection.

3.4 Results and Discussion

In this subsection, the quality of object segmentation and extracted features are evaluated. We also validate our approaches for computational scoring of both herbivory and disease resistance through comparisons with other methods. Because scoring of these traits is often performed using scoring rubrics, comparisons and detailed discussions of the differences between the manual and automatic methods are provided. Comparisons to computational methods are also made where possible.

3.4.1 Plant Area Traits

In this subsection, all evaluation and discussion related to plant area traits (area, herbivory, and growth) are provided. This includes a detailed comparison of our automated method against the quite common estimation method for measuring herbivory.

3.4.1.1 Verifying the Quality of Plant Area Measures

To illustrate the accuracy of our ability to measure plant area, which is truly a joint evaluation of both specimen separation and plant identification, an experiment was first conducted to compare leaf area calculations using PhenoPhyte, our method, with measurements from a leaf area meter, the accepted standard in quantifying plant area which requires plants or leaves to be harvested prior to measurement, and ImageJ [26]. In total, 24 leaves from Brassica *rapa* (IMB218) plants were imaged individually and scanned using a leaf area meter (CID 202, CID Instruments, Inc., Camas, Washington). Images were then processed using ImageJ and our image processing method.

The areas (in cm^2) produced by PhenoPhyte and the leaf area meter are compared in Figure 9. The two methods were found to produce highly similar results ($R^2 = 0.979$), and this benchmarking against the leaf area meter demonstrates the accuracy of our approach in detached leaf assays. Despite the almost identical results, there are three major differences between the methods. First, the two methods deal with leaf curvature in different manners; the leaf area meter flattens leaves while the inherent curvature of the leaves is retained and reflected in photographic imaging.

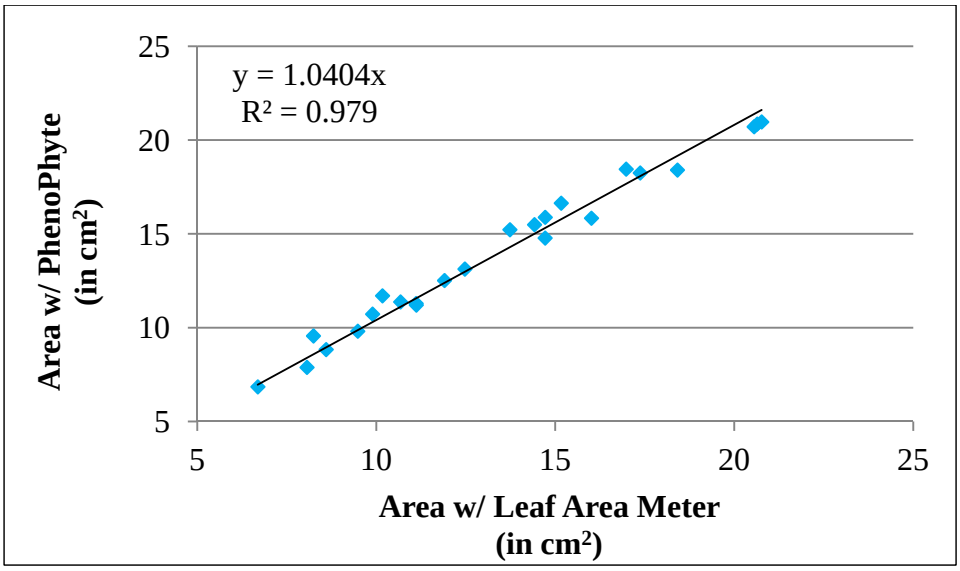


Figure 9: Benchmarking of the accuracy PhenoPhyte against a leaf area meter. The figure shows results consistent with the leaf area meter with slight variation due to differences in handling leaf curvature.

For inherently non-wavy leaves, the flattening of a leaf *can* result in a more realistic leaf area, as leaf curling is minimized by flattening but may translate to less visible leaf area in a top-down 2D projection of a leaf. Wavy leaves (e.g. many mature maize leaves), however, are difficult for both methods. Because a 2D image does not capture surface curvature, both methods can introduce error into the measured leaf area as area measurements from an image can underestimate the true area, and attempts to flatten wavy leaves with a leaf area meter can result in distortion of the leaf and areas where the leaf folds in on itself. Second, most leaf area meter models require harvesting of the leaf for measurement, whereas imaging using our approach can be nondestructive and done in any setting (field, lab, greenhouse, etc). Finally, our method is more versatile in that it can be used for measuring whole plant area in rosettes, where these situations are not conducive for use of a leaf area meter.

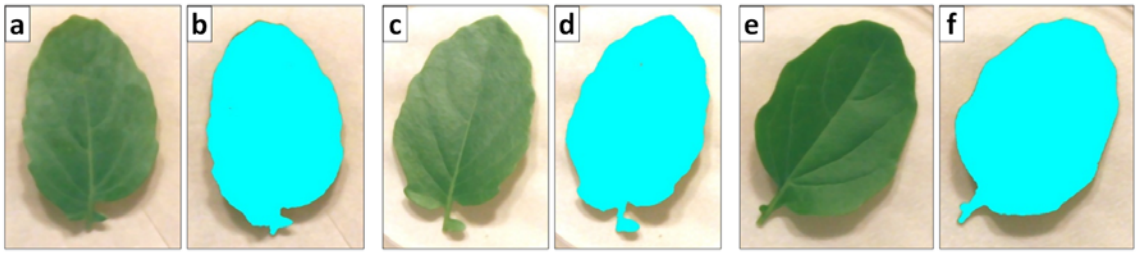


Figure 10: Example images and segmentation results from the leaf area meter comparison experiment.

To illustrate the effect of leaf curvature, Figure 10 shows three example images, along with the areas measured by these two approaches. The first two images (Figure 10 (a)-(d)) show nearly identical areas (difference less than 1 mm^2), while the third example (Figure 10 (e)-(f)) shows a more significant difference. The increased curvature in this third leaf example is evident by the variations in leaf color (Figure 10 (e)) (the right side has glare, while the left side is darker and at times much darker), which are not present in the first two examples.

The comparison of PhenoPhyte and ImageJ is shown in Figure 11. In this comparison, as expected, we see high similarity in leaf areas with a slightly higher correlation ($R^2 = 0.996$) than with the leaf area meter. This comparison also benchmarks the accuracy of our algorithms against the established ImageJ program.

Again, despite the similarities in output, there are three major advantages to the use of our approach over ImageJ. First, we have a built-in image normalization routine that standardizes imagery through color correction. To attain the ImageJ results in Figure 11, manual color correction of each image had to be performed in Photoshop before processing could begin with ImageJ.

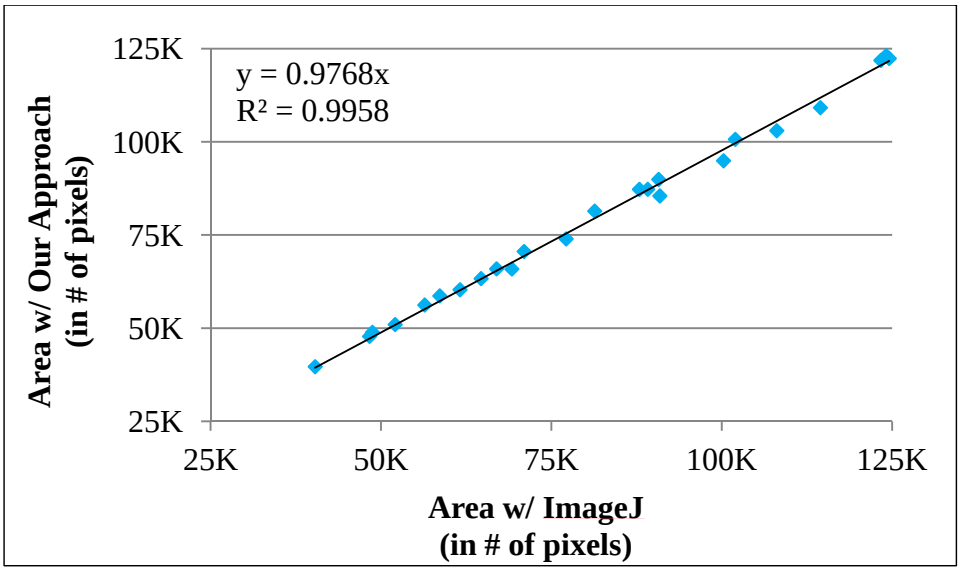


Figure 11: Benchmarking of the accuracy of our imaging method against the imaging program ImageJ. The results are nearly identical those from ImageJ.

Without such a step, the lighting and color variations across images would make a general-purpose leaf detection process very difficult to write in ImageJ. Second, the normalization routine that is packaged into our approach contains a size reference that allows conversion of pixel areas to physical distances (e.g. cm^2), whereas ImageJ provides results in number of pixels and, without extra user-supplied information, cannot provide area results in physical units. Finally, PhenoPhyte is more user friendly. Whereas automating these processes in ImageJ requires the creation of a macro, which is a nontrivial task for researchers with little to no programming background, our processing pipeline already exists and requires little technical background from users to operate.

3.4.1.2 Effect of Orientation on Plant Area

We also performed an experiment to determine the effect that leaf rotation or orientation plays on the calculation of leaf area. This was accomplished by taking six of the same leaves, rotating each one in 20 degree increments, and examining the resultant leaf areas.

Figure 12 shows the results for a single leaf. Though for a single leaf, variations in areas obtained by pixel counting were noticed. The maximum difference in all tested images was only 57 pixels across all leaf orientations, which for the single leaf data in Figure 12 corresponded to roughly 0.88 mm^2. When the pixel counts for all six of the tested leaves are shown together (see Figure 13), these slight variations are insignificant. These results are in concert with the results reported in [52], which showed leaf area calculations via pixel counting were independent of object orientation for regular geometric objects.

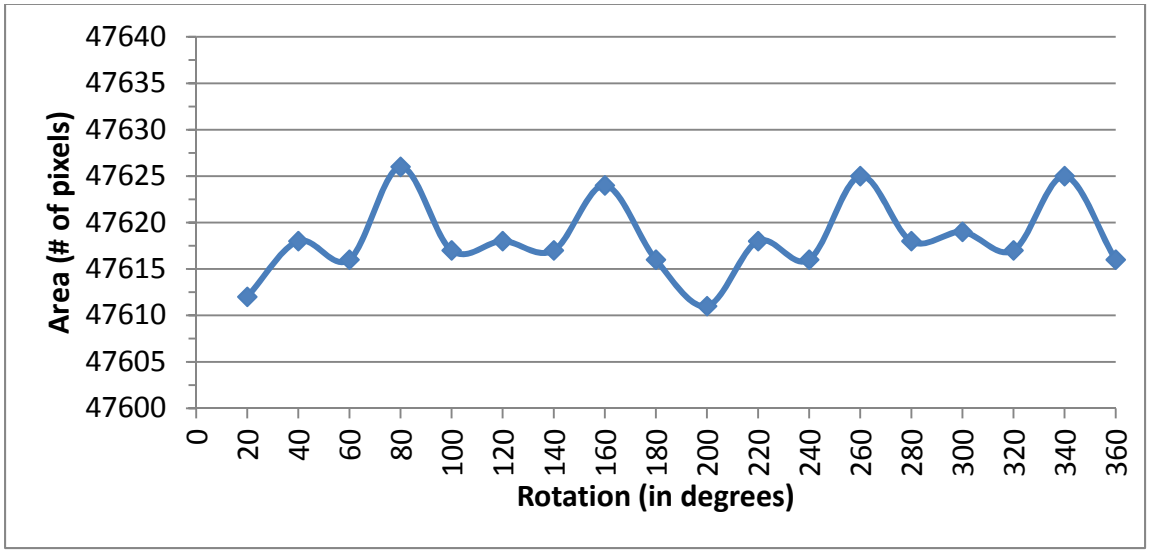


Figure 12: Close-up view of the effects of leaf rotation on leaf area. The maximum difference in this image is only 57 pixels, corresponding to a mere 0.88 mm^2.

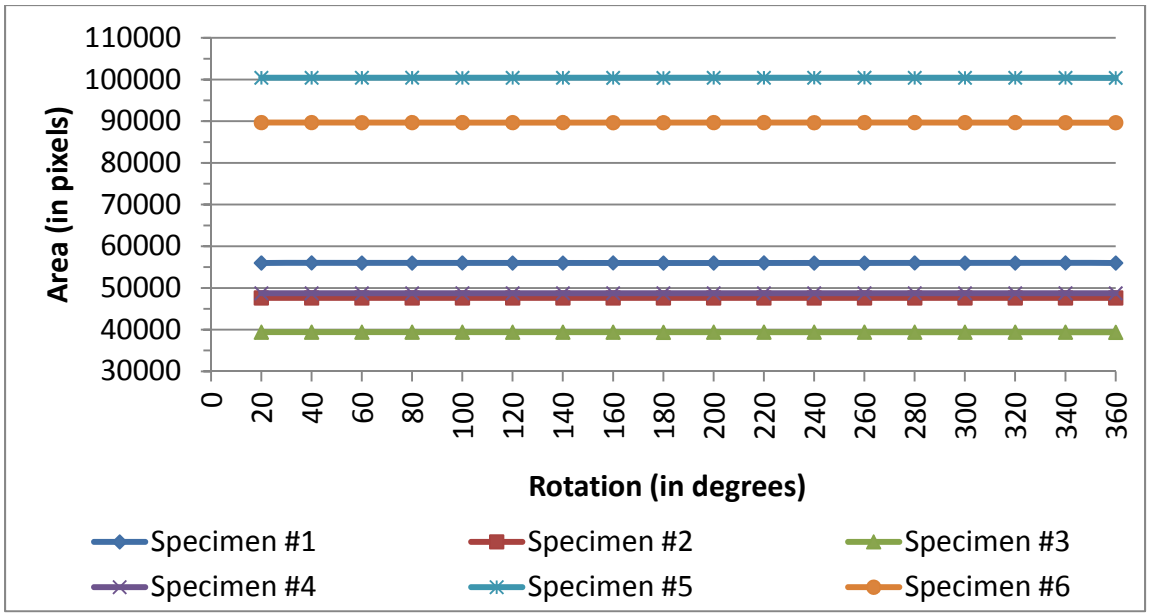


Figure 13: Comparison of leaf areas between six leaves across 20 degree increment orientations in leaf rotation. The effect of orientation is negligible from this viewpoint.

3.4.2 Comparing approaches for measuring herbivory

There are several methods for determining leaf area changes caused by insect herbivory on intact plants or detached leaves, including measurement using grid paper, digital imaging combined with ImageJ, and leaf harvesting combined with a leaf area meter or paper tracing [53]. One of the most widely used and simplest methods, however, for calculating the extent of herbivory is a leaf damage range estimation method [6] that utilizes a scale based on ranges of percentages of leaf area removed (see Table 3). Due to its common nature, it was chosen as the main comparison to our approach. Additionally, we also compare our herbivory results with that from ImageJ.

Table 3: Scoring rubric for the commonly used leaf damage estimation method.

Score	% Leaf Area Removed
0	[0% - 5%)
1	[5% - 13%)
2	[13% - 23%)
3	[23% - 37%)
4	[37% - 55%)
5	[55% - 77%)
6	[77% - 100%]

Using a test set of 30 *Arabidopsis thaliana* rosettes, we compared herbivory calculations using PhenoPhyte with the leaf damage estimation method, and then also with ImageJ. Images were taken of each pot before and after caterpillar feeding. Images were processed with PhenoPhyte and ImageJ to automatically measure the amount of leaf material in both before and after images. Five biologists were recruited to manually assess herbivory using the leaf damage estimation method based on three kinds of information: the live plants right after herbivory (LIVE after), images of the plants after herbivory (IMAGE after) and images of the plants before and after herbivory (IMAGE before & after). This experimental design was used not only to facilitate a direct comparison between our approach and the estimation method, but also to try and answer a number of interesting questions related to the use of manual and automated scoring methods, including:

1. How consistent is human estimation of herbivory across scorers?
2. Does human inspection of “after feeding” images produce similar scores as inspection of the plant itself?
3. If a “before” image is provided, does human estimation of herbivory improve? If so, by how much?

We first compared the leaf damage estimation method of the live plants against our automated approach. Figure 14 shows comparisons of these two approaches for each of the five biologists and the average human score. On average (bottom right), the two methods agreed (i.e., when the blue triangle falls within the colored rectangle) in only 20% of the cases (6 out of 30) and differed by at most one scoring category in 73% of the cases (22 out of 30). The remaining 27% showed more significant differences (in at least two scoring categories) in herbivory measurement. In general, manual scoring tends to underestimate small to moderate amounts of leaf damage and tends to overestimate large amounts of leaf damage (see Figure 14). The experiment as a whole demonstrates the need for objective, consistent, automated scoring, as the correlation coefficient between the two scoring methods ranged from 0.58 to 0.77 depending on the scorer. Note that these results are similar to an earlier preliminary experiment consisting of 18 pots with two plants per pot scored by a single expert, in which the two methods agreed 25% of the time and differed by at most one scoring category in 75% of the cases (see Figure 15). The correlation coefficient between the two scoring mechanisms in that experiment was also similar, at 0.64.

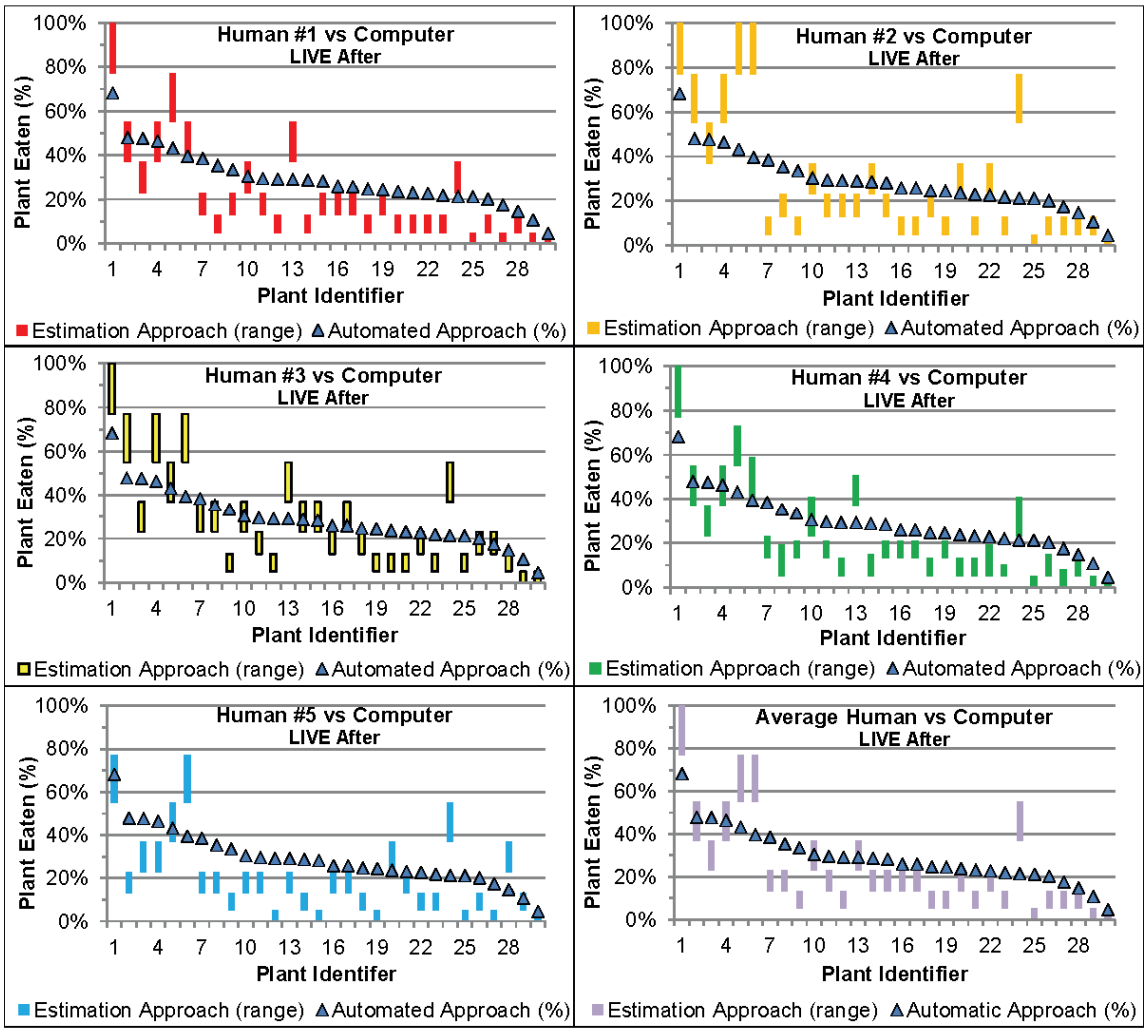


Figure 14: A comparison of the accuracy and precision between the leaf damage estimation method using live plants and our computational method for measuring herbivory for each of the five scorers as well as for the average leaf damage estimation score. The black bars indicate the range of leaf damage assigned using the estimation method, and the gray triangles show the exact percentage calculated using our image processing approach.

The differences in leaf damage assignments may be attributed to several factors. First, because the human-scoring method does not document the initial state of the plants (while the automated approach does), a mental reconstruction of the original state of the entire plant is required to estimate leaf damage. This can be particularly difficult in cases where large portions of a leaf or entire leaves are consumed or lost. Second, the human-scoring method relies heavily on human perception and consistency. In addition to identify all places where portions of the plant are missing, a scorer must be able to accurately estimate the percentage of plant removed and replicate this accuracy over time. All inconsistencies in reconstruction and assignment of damage classes result in error incorporated into the damage estimation. The automated approach is not without limitation, however. First, the images used in the computer-scoring method can only measure the amount of plant matter visible from a top-down two- dimensional view of the plant. Inherent leaf curvature may cause some underestimation of leaf area. Also, because rosette leaves can overlap, the automatic method cannot measure damage done to "hidden" leaves, although in this experiment the visible leaves were primarily the ones eaten by the herbivores.

The computer-scoring method has additional advantages over the human-based scoring method. First, the information that interests most plant scientists is the absolute amount of plant eaten, not the percentage relative to the original plant area that the human-based scoring provides. The computer-scoring method can calculate an absolute leaf area eaten, which when combined with a sampling of leaf thickness can be used to determine the volume of plant matter consumed.

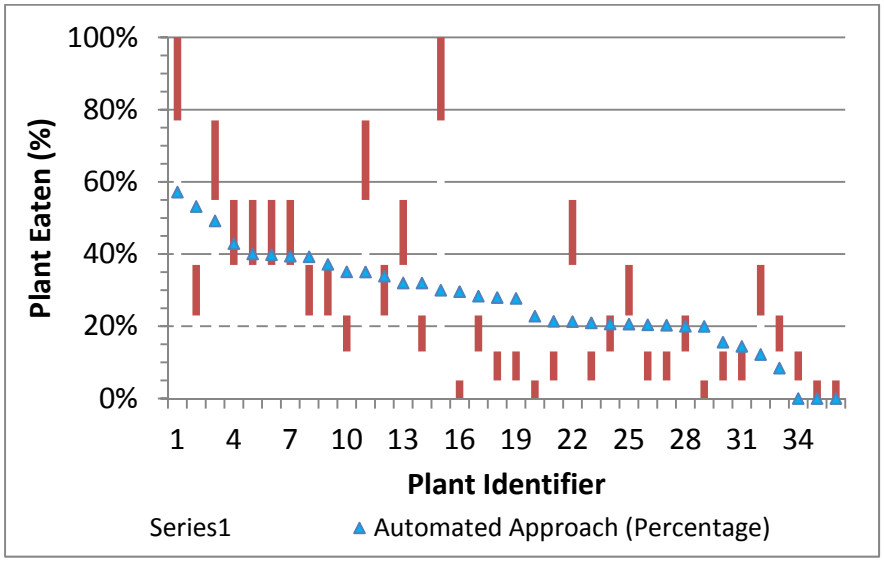


Figure 15: A comparison of the accuracy and precision between the leaf damage estimation method and our computational method for measuring herbivory. The red bars indicate the range of leaf damage assigned using the estimation method, and the blue triangles show the exact percentage calculated using our image processing approach. The computational method is more accurate, consistent, and precise than human estimation.

Second, the human-scoring method relies on arbitrary intervals to score plants, which means that plants with only subtly different leaf damage could end up with different scores. To demonstrate, consider Figure 16, in which the actual percentage from the computer-scoring method is plotted against the score this percentage maps to on the leaf damage estimation scale. In particular, consider the percentages near the cutoff between scores 2 (13% - 23%) and 3 (23% - 37%). The plants that flank this boundary have a measured plant area removed of 22.8% and 23.1%, a difference of only 0.3%, and yet their damage scores are different. This loss of precision introduces noise that may affect phenotype analyses. Because the computer-scoring method can generate

more precise percentages, there is no need to partition the percentages into these ranges, and thus this issue is avoided.

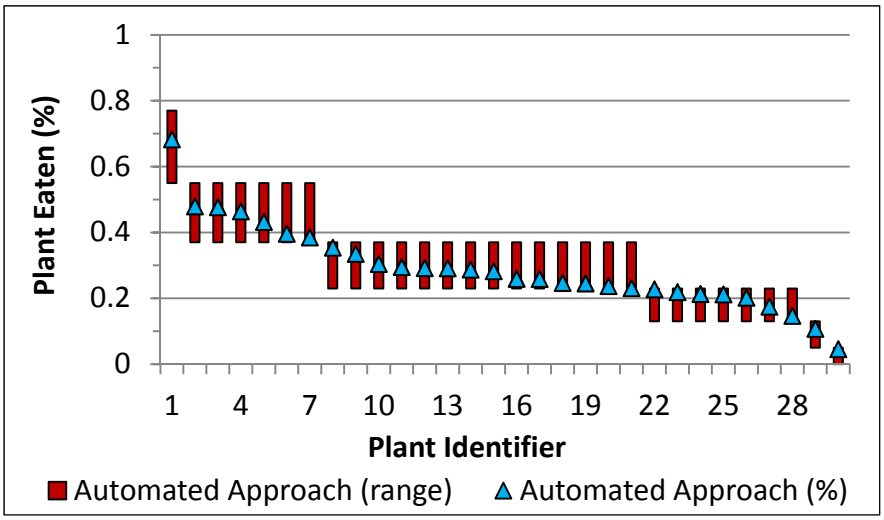


Figure 16: Comparison of raw percentages of plant eaten from the automated approach (gray triangles) with those raw percentages converted to the leaf damage estimation scale (black bars).

In addition to scoring of the live plants (LIVE after), we also had each human scorer assign a leaf damage estimation value in two additional ways, by visual inspection of only the "after" images of each plant on a computer screen (IMAGE after), and utilizing both the "before" and "after" images on a computer screen (IMAGE before & after). Each of these rounds of scoring was done independently at least one day apart. The experiment was designed in this manner to not only allow a comparison of the two methods, but also to examine the consistency and accuracy of human estimation on this task.

We then examined the within-scorer differences in leaf damage assignment across the three estimation tasks. The first two tasks require biologists to do a mental

reconstruction of the “before” state of the plant for herbivory assessment, whereas the latter two tasks involve scoring based on a 2D projection of the plant. The results from this experiment show little difference in scoring between the first two tasks, indicating little information difference between viewing the plant itself and viewing a top-down image of the plant; however, there is evidence of improvement in scoring with the third task, the task that requires less mental reconstruction. Across the five scorers, the average number of perfect matches with the automated approach was 22.0% and 25.3% for tasks 1 and 2, respectively, compared to 32.0% for task 3, with this improvement being just outside the 5% significance level (p-value = 0.0695 when comparing task 3 against both tasks 1 and 2 using the Fischer’s Exact Test). Similarly, the average number of times the two methods were separated by at most one scoring category was 66.7% and 69.3% for tasks 1 and 2, respectively, compared to 82.0% for task 3; this is a statistically significant difference (p-value = 0.0016 using the same test as above). Figure 17 shows these same trends visually using plots from an individual scorer (the scorer who over the three tasks best matched the automated scoring) as well as from the average scores for all humans across the three tasks. These results provide further support for the improved accuracy and consistency in using imagery and automated scoring for assessing change in plant area.

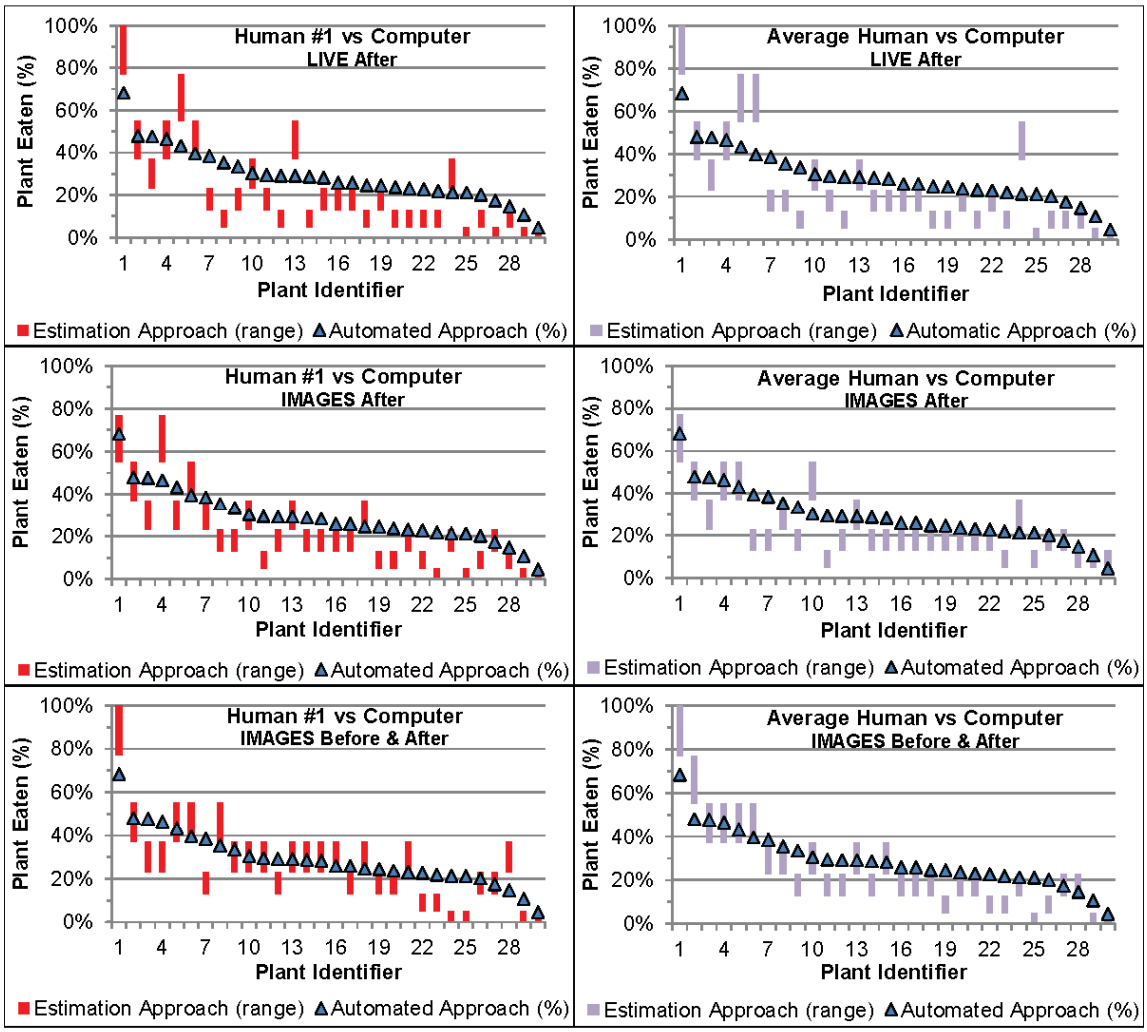


Figure 17: A comparison of the accuracy and precision between the leaf damage estimation method and our computational method for measuring herbivory for one of the five scorers as well as for the average leaf damage estimation score over the three labeling tasks. The colored bars indicate the range of leaf damage assigned using the estimation method, and the blue triangles show the exact percentage calculated using our image processing approach.

We then compared the results produced by PhenoPhyte and ImageJ with these herbivory images. Figure 18 shows the relationship between the methods in terms of raw areas for each plant and each timestamp. Again, we see extremely similar, highly correlated areas ($R^2 = 0.995$) for individual plants both before and after herbivory. This again shows the accuracy of our approach in measuring leaf area, as compared to ImageJ. Secondly, we plot the herbivory measurements from the two methods against one another (Figure 19). As expected, the correlation values are slightly lower, though still high ($R^2 = 0.948$), as herbivory includes the combination of two area measures, each of which may be slightly different from one another. Note that the ImageJ herbivory results required manual linking of areas for corresponding before and after images, whereas PhenoPhyte automatically linked the images together and computed the difference in areas. In large experiments, this could save researchers a significant amount of data analysis time.

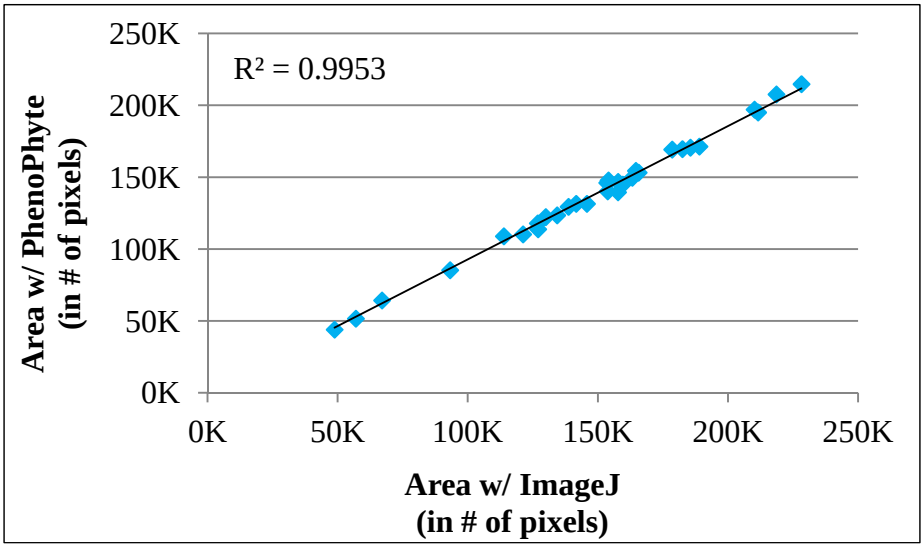


Figure 18: Comparison of PhenoPhyte and ImageJ in terms of rosette areas for *Arabidopsis thaliana* rosettes.

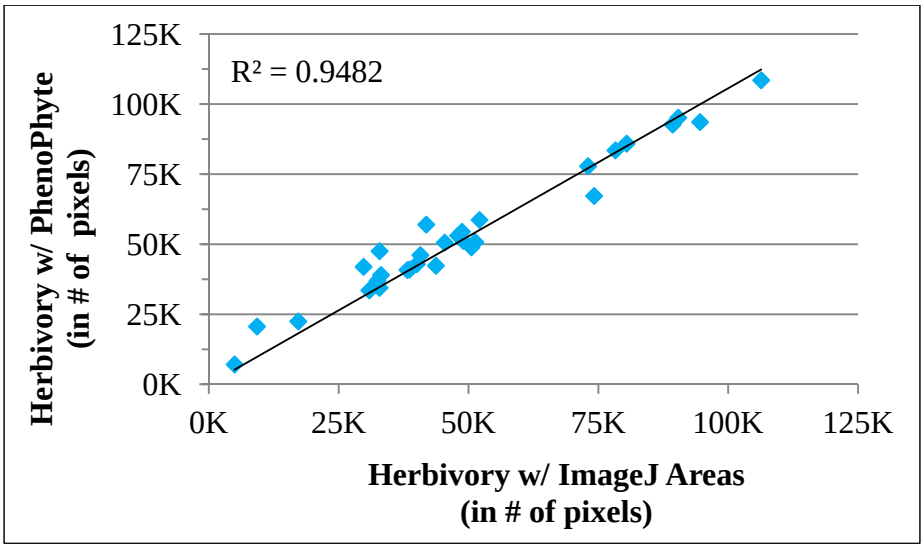


Figure 19: Comparison of PhenoPhyte and ImageJ in terms of herbivory for *Arabidopsis thaliana* rosettes.

3.4.3 Maize Leaf Traits

3.4.3.1 Verifying Maize Leaf Features

In order to evaluate the quality of the features extracted for the maize leaves, we examined the ability of the features to discriminate classes of maize mutants. In this experiment, 135 images from 8 classes of maize *les* mutants [54] were selected for testing: *les1, les6, les7, les8, les13, les17, les18,* and *les19.* These images were collected from the South Farms genetic fields at the University of Missouri during the summer of 2007. Each of the selected les mutant phenotypes has patterns of chlorotic lesions (areas of reduced chlorophyll shown as light green or yellowish spots on the leaves) and necrotic lesions (dead tissue seen as brown or white spots on the leaves) in various sizes, spatial distributions, and severities. Figure 20 shows an example of each *les* mutants.

The experiment was conducted using tenfold validation. For each iteration of the experiment, the 10% reserved for training was used to build an advanced multidimensional indexing tree, specifically an EBS k-d tree [55]. For each test image, classification was performed by finding the 5-nearest-neighbors in feature space and assigning the *les* mutant that occurred most often in those neighbors. The experiment yielded an overall classification accuracy of 90.4%. Detailed results are shown using a confusion matrix in Table 4. The small number of misclassifications arose from images in different lesion classes with similar visual patterns, such as *les8* and *les19* in our image collection. This high classification accuracy is a strong indicator that our extracted features are capturing the discriminating characteristics in these phenotypes.

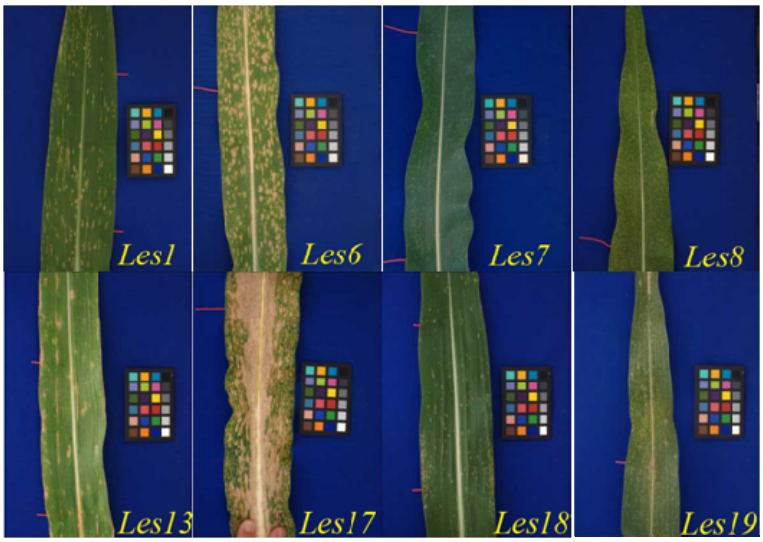


Figure 20: Example of each of the eight classes of les mutant used in the classification experiment.

Table 4: Confusion matrix showing classification results of *les* mutants.

		Predicted							
		les1	*les6*	*les7*	*les8*	*les13*	*les17*	*les18*	*les19*
Actual	*les1*	**30**	0	0	0	0	0	0	0
	les6	1	**17**	0	0	0	0	0	1
	les7	0	0	**17**	0	1	0	0	0
	les8	0	0	0	**12**	0	0	1	4
	les13	0	0	0	0	**14**	0	0	0
	les17	0	1	0	0	0	**14**	0	0
	les18	0	0	0	0	0	0	**11**	1
	les19	0	0	0	1	0	0	2	**7**

3.4.3.2 Comparing approaches to scoring disease resistance to Southern Leaf Blight in Zea mays

In terms of measuring disease resistance phenotypes, which are much more complicated traits to quantify than herbivory, scoring rubrics are also widely used and thus again will be our primary means of comparison and evaluation. As with the herbivory evaluation, a number of interesting questions related to disease resistance scoring can be asked:

1. How do our computer-generated scores compare to manual scoring?
2. What is the across-curator consistency of scoring disease resistance in the field?
3. How does scoring based on images compare to field scores? Do field conditions affect a person's ability to score accurately and consistently?
4. Our imaging of two specific leaves was based on text descriptions of the scoring rubric. Do these images capture enough information to predict field scores, given that in the field scorers can base their decisions on full-plant visual patterns for an entire row?

In this experiment, the resistance to Southern Leaf Blight (a disease caused by the fungus *Cochliobolus heterostrophus* that causes necrotic lesions to appear on the leaves and can affect yield) in maize using a B73 x Gaspe introgression library was quantified. Infection severity was first scored manually using a 1 to 9 scale where 9 indicates no phenotypic expression of the disease and 1 indicates death of the plant, using a rubric like that in [5] except inverted. While only integer values are defined in the rubric, scorers were permitted to give intermediate scores, in 0.25 increments, as necessary. Multiple experts scored the 447 rows of the introgression library separately. Field

scoring was performed by examining the expression of whole plants (all 10-12 plants per row), giving the most importance, however, to the appearance of top two ear leaves. For the automated approach, imaging was performed by selecting two representative plants from each of ~400 rows and nondestructively imaging the top two ear leaves for a total of ~1600 images. These leaves were chosen for imaging as they corresponded to the leaves of the plant that were deemed most important in differentiating resistance levels, according to the infection scoring rubric. Automatic processing of these images (as seen in Figure 7) was used to compute infection severity.

As we cannot directly measure pathogen growth by examining a leaf image, an indirect measurement was utilized as a proxy for the degree of disease resistance. In particular, the automated approach used the percentage of the leaf covered by necrotic lesions as its phenotype score. For example, plants whose imaged leaves were completely dead were given a score of 100, and those whose leaves were half covered by necrosis were given a score of 50.

Four plant scientists were asked to assess disease resistance both in the field (LIVE) as well as using images of the representative leaves (IMAGES) using the rubric. For each expert, phenotyping in the field was completed in a single day; scoring between experts varied over a four-day period with two experts scoring on the same day—imaging was conducted over a three-day period within this timeframe. In general, resistance phenotypes do not appreciably change during this time period.

As before, the human-supplied scores from the field were first compared to the computer scoring. Figure 21 shows scatter plots of the three human scorers against the automated scoring approach for both live plants (left column) and images (right

column). When examining the plots from live plants (left column), one first notices the marked overall variation in the plots between scorers as well as the wide range of amounts of necrosis for individual scores. For example, for scorer #1 (top left) and rubric value 5, the amount of necrosis on the top two leaves ranged from 13.5% to 70.9%. These differences may be partially attributed to a number of factors. First, when phenotyping in the field, scientists utilize more information than is captured with imaging: all plants are considered instead of just the two representative plants imaged, and the entire plant is examined versus the top two ear leaves with imaging (though those leaves are the most important, and descriptions of scores 3-7 in the rubric refer only to these leaves). Second, there may be some changes in the phenotype between the times when the plants were scored, though again resistance phenotypes do not appreciably change over this period, and this cannot fully explain the variation as the two scorers who phenotyped on the same day had starkly different patterns. Third, the percent necrosis measure may not be detailed enough to distinguish between different disease resistance scores as, for example, the sizes of individual lesions on the leaf are lost in the aggregate percent necrosis measure. Finally, there may be inconsistencies and inaccuracies in human labeling, which is supported in particular by the variation in plots between the two scientists who scored on the same day (see Figure 22). See Bock et. al. [56] for additional discussion of sources of phenotyping errors from visual assessment and imagery.

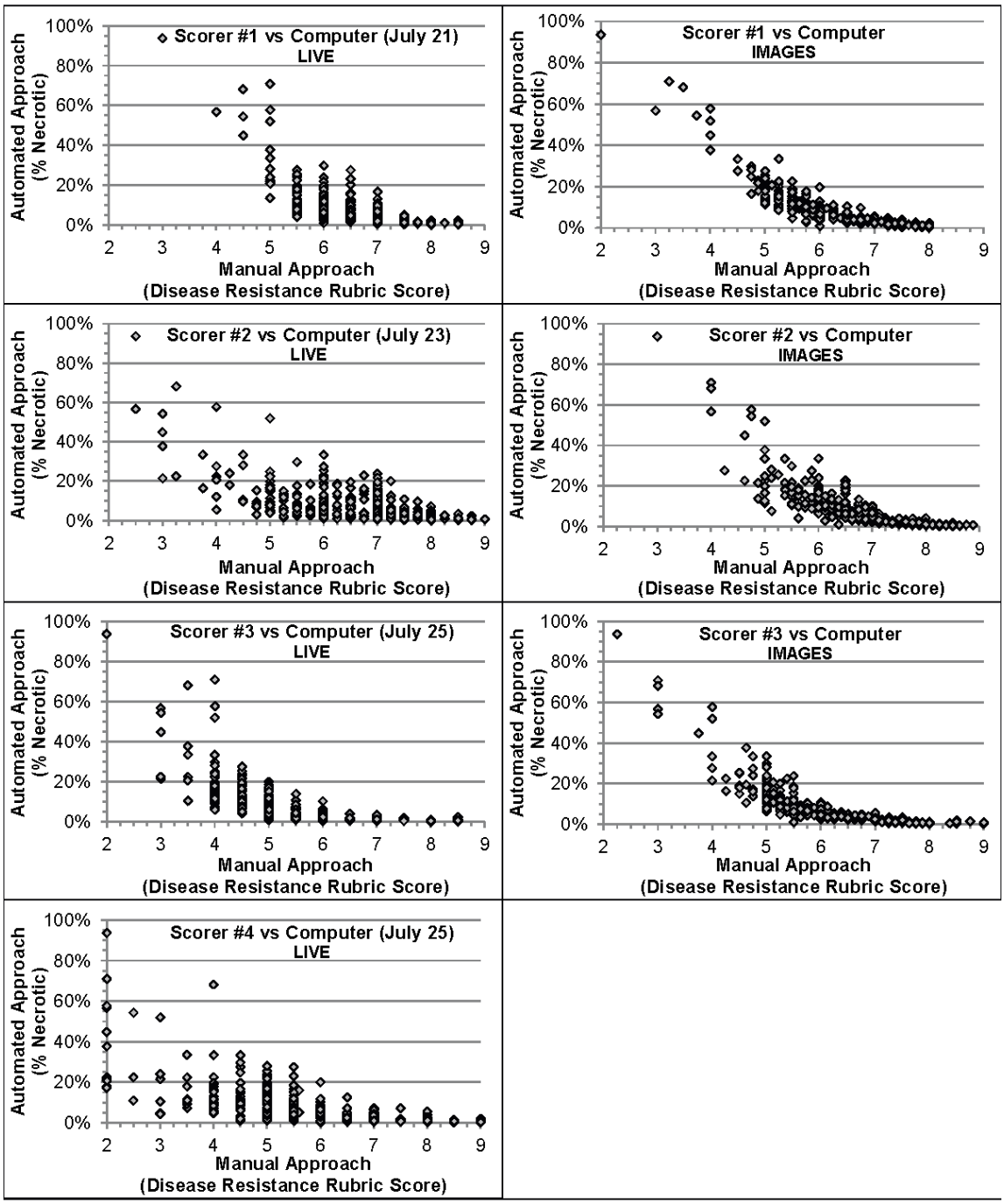


Figure 21: Scatter plots showing phenotyping based on LIVE plants (left column) and IMAGES (right column) for each of the human scorers against the automated approach. The manual labeling was conducted using a disease resistance rubric, and automated measure calculated the percentage of the leaf covered by necrotic lesions. Scorer #4 did not phenotype based on images.

Plant scientists also phenotyped the plants based on images (right column). This was done to further investigate between-scorer variation while eliminating variation due to information not included in the captured images and changes in plant appearance due to phenotyping timing. It should be noted that only three of the four experts who had previously scored the LIVE plants participated in this portion of the study. As expected, the scatter plots are much more similar when phenotyping is based on the images, though obvious human-related variations remain. To explore this in more depth, Figure 23 shows the pair-wise comparison of manual labeling. As these graphs compare scores on a plant-by-plant basis, they allow better visualization of the variation between scorers. To quantify this variation, a linear regression was performed on each plot, and the amount of variance explained by the regression ranged from 0.755 to 0.879.

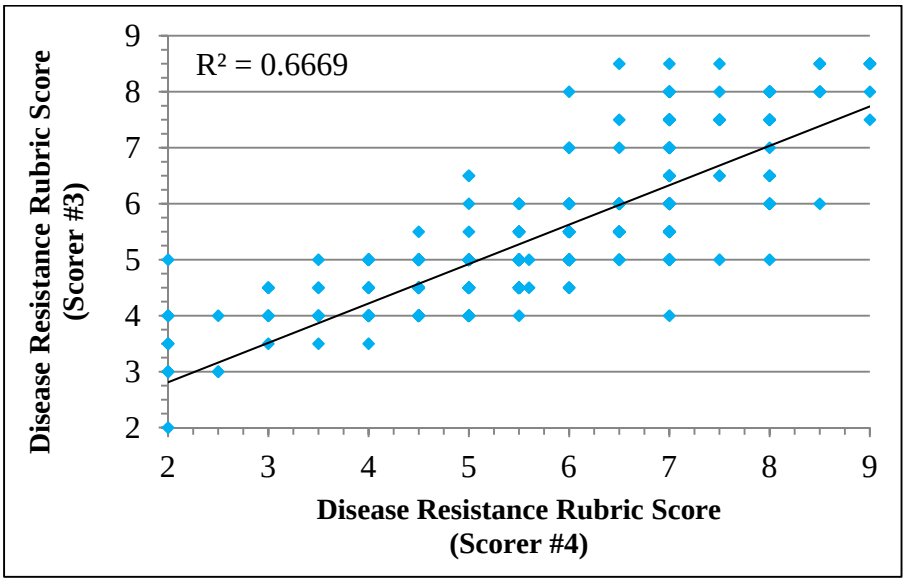


Figure 22: Pair-wise comparisons of manual phenotyping based on LIVE plants between the two scientists who scored on the same day. The amount of variance explained by a linear regression of the data is 0.667.

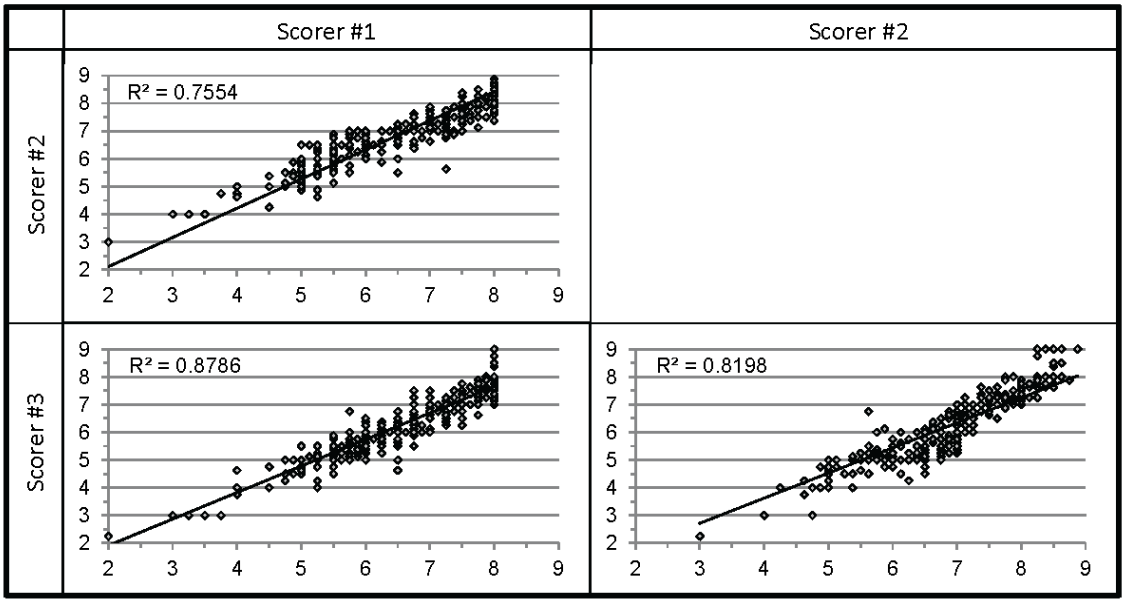


Figure 23: Pair-wise comparisons of manual phenotyping based on IMAGES using the disease resistance rubric, with the amount of variance explained by linear regression of the data ranging from 0.755 to 0.879. Even when viewing the same phenotypic information, variations in scoring remain.

3.5 Summary

In this chapter, high-throughput methods were developed for capturing trait and phenotypic information utilizing image content, notably including the extraction of relevant features for describing lesion burden in leaves from lesion mimic mutants and maize plants inoculated with Southern Leaf Blight.

For both herbivory calculation and the scoring of disease resistance to SLB, in-depth comparative studies were conducted with commonly utilized human estimation methods. With the disease resistance studies, some variation was demonstrated between different scorers, with the correlation coefficients between scorers ranging from 0.71 to 0.87. Interestingly, the highest correlation value was noted between the two scorers who

scored the most days apart (four days). However, when scorers utilized imagery, their correlations increased to between 0.92 and 0.93. These highly similar results suggest some differences in the scorers' perception and focus in the fields.

With the herbivory studies, an increased amount of variation was demonstrated between different scorers. Correlation coefficients between measured herbivory through imagery and human scoring of live plants ranged from 0.58 to 0.77. Comparison of scoring live plants versus scoring with after images showed little difference; however, there was an improvement (+12-13%) when the scorers were allowed to see both before and after images.

These kinds of variations demonstrate the noise that is introduced in these kinds of manual phenotyping and the need for more objective and consistent scoring using automated approaches.

CHAPTER FOUR

VISUAL PHENOTYPE TEMPORAL ANALYSIS

The HR (hypersensitive cell death reaction) response is one of the most important immune responses in the plant kingdom, and yet it is not fully understood in plants. Researchers interested in studying this process oftentimes utilize lesion mimic mutants, in which the response is activated in the absence of disease, or plants with specific mutations that are known to affect this process. For example, Johal et al proposed using Rp1-D21, an autoactive allele of the Rp1 disease resistance gene that spontaneously activates HR, to study this process in terms of the natural variation in maize germplasm. While researchers have examined these phenotypes with regard to a number of conditions and have identified a number of known variables that affect lesion formation, including host factors like developmental stage and genetic background as well as environmental conditions like light, temperature, and other growing conditions [54, 57], there have not been, to the author's knowledge, any methods developed to facilitate in-depth quantitative study of the temporal behavior of these phenotypes. As the temporal characteristics of these phenotypes are likely influenced by factors relevant to HR response, the development of such an automated method that can generate temporal trait data has the potential to help researchers gain further insight into the signaling pathways and underlying genetic and epigenetic mechanisms that underlie the HR response.

In this chapter, we propose a method for processing time course studies of maize leaves that allows tracking of necrotic lesions across frames and models the temporal behavior of those tracked lesions.

4.1 Literature Review

Image registration is the task of aligning two or more images of the same scene or part of the same scene that are taken from different viewpoints, different modalities, and different time points. Two broad classes of approaches for image registration exist: area-based methods and feature-based methods.

Area-based methods refer to the class of techniques that utilize template matching to perform registration. These methods attempt registration without finding meaningful features in the image; instead, similarity is computed for window pairs between the two image frames with the maximum similarity used to define the corresponding pairs. The normalized cross-correlation method [7] and its variants measure similarity between window pairs and seek to find pairs with maximum similarity. Area-based methods based on Fourier transforms have also been proposed, which look for the peak in the phase correlation in the frequency domain and then map the peak back to the original images, and include modifications that can handle frequency-dependent noise [58], rotation [59], and changes in image scale [60]. There is also a set of area-based methods that utilize the theory of mutual information [61-63] which are most widely used in registering images from different modalities, including aligning MRI images, combining medical images with functional images, and mapping 2D and 3D object models to real scenes. Area-based methods, in general, are best suited for situations in which the relevant information can be obtained by examining intensity changes instead of by using shapes and structure [64], which notably includes most medical imaging modalities. The disadvantages of these approaches is that they can generally only handle

translation and small amounts of image rotation before becoming quite computationally expensive.

Feature-based methods, on the other hand, operate by first extracting relevant features or objects from the images (called control points), which are used to represent the image, and then using some technique to establish correspondences or mappings between pairs of control points. The set of techniques that are most relevant are those of the point pattern matching problem, in which each image is reduced to a set of control points and pair-wise point correspondences are identified. A variety of approaches have been proposed for solving this problem. In [65], a clustering technique is utilized in which affine transformation parameters are calculated for point pairs, and the largest cluster of parameters is used to perform the match; however, the point pairs chosen are those with similar structure (size and shape). A number of relaxation methods have also been developed, which utilize iteration to progressively arrive at the final mapping of points. A merit score is assigned to objects based on the relative mapping of all other objects and refined during the mapping of other objects in [66], while [67] uses fuzzy relaxation and geometrical transformations to find correspondences between labeled point patterns. Additionally, Irani and Raghavan, [68] utilize a Monte Carlo random search to match points using both transformation parameter estimation and geometric hashing. There are also nearest neighbor search techniques proposed in [69-70]. Most of the point pattern techniques can handle differences in translation, rotation, and scaling. There is an assumption in many of the algorithms for the same number of points in the two images; however, this is not universal as a few (the nearest neighbor techniques and one clustering approach) are able to handle some local distortion and

extra/missing points. These feature-based methods are not without disadvantages, however. One of these is that the features may be difficult to detect or unstable in time [64], which can affect algorithm performance. Ideally, these methods need invariant descriptors, which are unique, stable, and invariant to expected image deformations.

4.2 Problem Description and Challenges

This research's approach to studying of the temporal characteristics of necrotic lesion formation and progression on leaves of maize lesion mimic mutants can be broken down into four main steps. First, a method for high-throughput processing of individual leaf images to extract necrotic lesions (objects) and compute a set of features relevant to each object, including size and shape measurements, is required. For this study, the processing pipeline developed in Section 3.2.2 is utilized.

Secondly, an algorithm to register objects across images of maize leaves is needed. The algorithm must be able to track individual necrotic lesions across multiple images of maize leaves, with multiple objects (ranging from 10s to 100s) that need to be monitored. While several registration algorithms have been previously developed (see the review above), there are a number of items that make this task uniquely challenging and an extended approach is required for this task.

The data collection process for generating time course series of maize leaves in the field is the causal factor for several challenges with this lesion registration task. First, there are translation and rotation differences between images, though the majority of point pattern algorithms described above are able to handle such changes. In addition to just translation, there is variation in the portion of the leaf imaged between frames as well as in the rotation of the leaf in each frame. Thus, lesions in one frame are not

guaranteed to be in other frames. This also means that the tip of the leaf, which is perhaps the landmark with the most potential for establishing relative orientation on each leaf, is not guaranteed to be in each image and oftentimes is not in the image, as that portion is typically the first part of the leaf to die or be damaged in the field due to weather or insects. In fact, except for the lesions themselves, the leaf surfaces are fairly uniform and contain no anatomical landmarks with which to try and define relative positions for lesions (except in relation to one another).

Additionally, data collection is difficult because of the inherent waviness of maize leaves, which can cause variation in the two-dimensional projection of the leaf margins and the leaf itself. The result of this waviness is the possibility of local and regional distortions in the leaf, which can also distort the spatial configuration and even the visibility of some lesions. Leaf waviness has also been found to produce glare-type appearances on portions of the leaves, which can affect lesion segmentation results. Finally, of course, images that are not completely in focus may also have an effect, as blurriness can affect the visibility and detection of lesions by segmentation algorithms.

Other difficulties with this task relate to the characteristics and behavior of the lesions themselves. First, all the objects are generally indistinct, having the same spectral characteristics. Lesion sizes and shapes may vary, but they are typically similar and rarely distinct. Furthermore, these lesions expand and grow between image frames at different rates, meaning lesion size and shape are not guaranteed to be similar between the template and target frames, and there is no known method for determining how a particular lesion is going to react between any two given image frames. Moreover, as lesions grow, they can and do merge together. Finally, new lesions may appear

spontaneously in any location on the leaf, and it is not rare for there to be materialization of many new lesions in an image frame. This is likely to be troublesome for the standard point pattern matching algorithms, even the ones that can handle missing/extra points, as there is an assumption that the number of points in a source and destination frame will be similar.

This problem does not appear amenable to area-based registration algorithms for a number of reasons including potential rotation and leaf distortion, uniformity of the leaf surface, indistinct nature of the lesions, and the potential for large changes between images including growth and merging of lesions and the materialization of many new lesions.

Because the item that is most constant between consecutive image frames in these time courses is the spatial configuration of lesions, the point pattern methods may be the most relevant towards solving this problem. However, while some point pattern methods handle some of the challenges of the task, including translation, rotation, local distortions, and the ability to handle missing/extra points, there are a number of remaining issues that appear to not be handled by these algorithms. First, most point pattern methods assume that control points are just that, points. They do not utilize any associated relevant information to help with the mapping process. In the case of lesion registration, there are intuitive constraints on the size of the corresponding objects – they should in general be roughly the same size or larger. In more general lesion registration, including that of chlorotic lesions, other object parameters including shape and color may also represent important constraints in the mapping process. For example, chlorotic lesions could map to necrotic lesions but the opposite is not true. Second, it is suggested

that because point pattern methods think of control points as mass-less points, they may also not be able to account for growth, especially significant growth of objects, as this may affect the point centers, which corresponds to the noted disadvantage of feature-based methods when features are unstable over time. Finally, and most importantly, none of the point pattern methods allow multiple points to map to the same target point, and this undoubtedly will be needed whenever lesions merge. Additionally, though lesions do not in practice split, the algorithm may need to handle the case of objects splitting. This can occur when distortion in one frame causes the erroneous appearance of lesion merging but is resolved in future frames, or alternatively in the case of suboptimal segmentation results.

For these reasons, we first designed an algorithm for Variable Object Pattern Matching whose aim is to find object-to-object correspondences between a source and destination frame. This algorithm takes advantage of object spatial configurations and allows object characteristics to change significantly between the source and destination frame. Afterwards, a method for registering objects across entire time courses using this algorithm is proposed.

After objects have been registered across images, temporal features need to be calculated based on changes in individual object features over time, followed by modeling of the temporal changes, which is discussed in Section 4.5.

4.3 Variable Object Pattern Matching Algorithm

Let $P = \{p_i \,|i = 1,2,3 \ldots n_p\}$ be the set of objects in a source frame and $Q = \{q_i \,|i = 1,2,3 \ldots n_q\}$ be the set of objects in a destination frame. The goal of the Variable Object Pattern Matching algorithm is to find the maximal set of object-to-

object correspondences between *P* and *Q*, where objects do not move on the leaf but their characteristics (in terms of size, shape, etc.) may change substantially between frames. More formally, this translates into finding the set of object-to-object correspondences $\Psi = \{ p_i \rightarrow q_j \mid p_i \in P \wedge q_j \in Q \}$ for which the objective function $cost(\Psi)$, which provides a measure of the quality of object mappings, is minimized.

The algorithm is comprised of two major steps. In the first step, called Establishing Anchor Correspondences, the two frames are examined from a global perspective to try and identify some highly confident (anchor) mapping between the frames. In other words, this step aims to orient the destination frame with respect to the source frame. Two approaches for finding these anchor correspondences are described. In the second step, called Growing Anchor Correspondences, the algorithm utilizes a relaxation technique, in which the algorithm iteratively expands and refines the computed correspondences through more fine-tuned mapping of local neighborhoods. Both steps rely on spatial configurations of objects for establishing correspondences.

Because of the inherent uncertainty in assigning object-to-object correspondences, probabilistic mechanisms are employed to provide a measure of confidence for each individual mapping. This allows the algorithm to maintain a set of candidate target objects for each object to be mapped. Because it is possible for the algorithm to make a mistake, it is important that the algorithm has the ability to modify and change object correspondences, and this is accomplished via the iterative nature of the algorithm which allows it through the refining of mapping probabilities to make corrections as it progresses. The goal during each iterative loop is to increase the probability of correct mappings and decrease the probability of likely erroneous mappings.

Let $prob(i,j) = (s_p, s_t)$ represent the current probability that $p_i \rightarrow q_j$, where the probability is given by s_p/s_t. Additionally, let $\omega(\cdot) = (x, y)$ be the coordinates of the center point or centroid of an object from the source or destination frame.

4.3.1 Establishing Anchor Correspondences

In this first step of the Variable Object Pattern Matching algorithm, the goal is to identify anchor correspondences between a source and destination frame. These objects are likely to correspond to objects that are stable between the two frames, or objects that grow symmetrically so that their centers or centroids remain stable with respect to other objects on the leaf. The anchor correspondences, denoted as $\varsigma_{P,Q}$, are defined as those with sufficient probabilistic evidence to establish confident relative linkages between the two frames. Two methods for finding these correspondences are provided below.

4.3.1.1 Spatial Configurations

The first method for identifying anchor correspondences relies on matching spatial configurations between small sets of objects from P and Q (called template and target object configurations, respectively). The pseudo-code for this method is shown below.

Let $C_P = \{c_i^P \subseteq P \mid i = 1,2,\ldots n_c \wedge |c_i^P| = n_s\}$ be the set of template object configurations from P, where n_c is the number of template configurations and c_i^P is a set of n_s sampled objects from P. Analogously, C_Q is defined as the set of target configurations from Q where each c_i^Q consists of n_s sampled objects from Q.

Though each configuration consists of the same number of objects, the differences between the template and target configurations are (1) the number of configurations in

each set and (2) the manner in which the pool of candidate objects is chosen for each type of configuration.

1. FOR i = *1 to* n_c
2. c_i^P = Random selection of *k* (qualified) objects from *P*
3. Compute spatial signature for c_i^P
4. END FOR

5. *sThresh* = area of smallest object from C_P
6. Let *Q'* $\subseteq$ *Q* such that objects are equal to or bigger than *sThresh*

7. FOREACH combination of n_s objects in *Q'*
8. c_i^Q = current set of objects
9. Compute spatial signature
10. END FOR

11. Γ = findSimilarConfigurations(C_P, C_Q);
12. Compute mapping probability based on aggregated mappings

Algorithm 1: Pseudo-code for the Spatial Configurations algorithm for identifying anchor correspondences.

The algorithm begins by constructing a number of template object configurations by random selection of objects from the source frame *P* (lines 1-4). Because there is no guarantee that any of the objects selected from *P* in a template configuration will appear in Q, choosing multiple template configurations, each with a different set of random objects, increases the likelihood that all the objects in at least some of the template configurations will appear in Q. Moreover, ensuring some overlap of objects between different template configurations can also provide additional evidence of the correctness of individual object mappings.

Note that both the number of configurations n_c and the number of objects in each configuration n_s are tunable parameters in the algorithm. We are currently using $n_c = 20$, unless there are not enough objects in the image to generate this many configurations, in which case n_c is set to the number of all possible combinations of n_s objects.

The value of n_s, the number of objects in each configuration, must be at least three, as it takes at least three points to define a plane. However, when $n_s = 3$, there is a chance of multiple sets of three objects with the same spatial configuration. The more n_s is increased, the less likely this situation is to occur due to the increased geometrical constraints; however, increasing n_s also has the negative side effect of increasing the size of the set of objects that must appear in the target frame and also increasing the number of target configurations to iterate through during the matching process, as shown for a sample set of images in Table 5.

Table 5: The number of target configurations needed by the algorithm for the first several frames of a time course using various number of objects in each configuration.

		Value of n_s				
		3	**4**	**5**	**6**	**7**
Frame #	**0**	816	8855	53,130	296,010	3,365,856
	1	1540	12,650	98,280	736,281	5,379,616
	2	680	3876	26,334	230,230	1,184,040
	3	1771	8855	42,504	177,100	1,184,040
	4	4960	14,950	201,376	1,344,904	6,724,520
	5	120	2380	8568	27,132	116,280
	6	12,341	23,751	575,757	6,096,454	

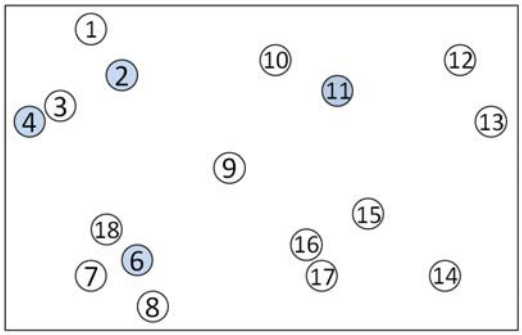
The n_s objects for each template configuration c_i^P are initially chosen as follows: From among the largest $m_c > n_s$ objects from P, n_s objects are chosen randomly such that $c_i^P \neq c_j^P$ for $i \neq j$. The rationale for limiting the selection to the largest m_c objects is twofold. First, these larger objects are more likely (though not guaranteed) to reappear in frame *Q*. Cases when objects have been found to not be present in Q include when portions of the leaf are removed, when the segmentation algorithm finds no corresponding lesion or a markedly smaller corresponding lesion, when there are differences in the field of view between *P* and *Q*, and when variations in leaf waviness cause a lesion to appear significantly smaller in Q. The second reason is that this method for selecting objects reduces the number of objects to consider in Q when generating target configurations. Since objects in *P* should appear as roughly the same size or larger in Q, we will only need to consider objects in *Q'*, those that are roughly equal in size or larger than the smallest of the largest m_c objects from P. The size threshold and the construction of Q' correspond to lines 4-5 in the pseudo-code.

One could consider constructing all possible combinations of n_s, which would be $\binom{|Q|}{n_s}$ combinations; however, as typical images have a few hundred objects, this could generate a tremendous number of object combinations, which may render this approach computationally intractable. Using the described approach to reduce the search space, the number of target configurations can be reduced significantly, to $\binom{|Q'|}{n_s}$, though this still has been shown to produce hundreds of thousands to millions of target configurations with the potential for generating even large numbers.

After the Q' set is constructed, all possible combinations of n_s objects are formed, and collectively these sets of objects comprise the set of target configurations (see lines 7-10). The algorithm then seeks to identify similar pairs of target/template configurations by means of their spatial signatures.

Compact representations of the spatial configurations, called spatial signatures or descriptors, are also computed for each template and target object configuration (lines 3 and 9). Let $\delta(\cdot)$ be a function that takes a template or target configuration as input and outputs a vector representing the spatial signature. We define this function as the sorted pair-wise distances between the n_s objects in a configuration. As an example, the situation in Figure 24 where n_s=4.

Set of Objects *P*



(a)

Pair-wise Distances

	p_2	p_4	p_6	p_{11}
p_2	-	0.56	1.00	1.17
p_4	0.56	-	0.95	1.68
p_6	1.00	0.95	-	1.42
p_{11}	1.17	1.68	1.42	-

(b)

Associated Spatial Descriptor

$$\delta(c_1^P) = \{\, 1.68, 1.42, 1.17, 1.00, 0.95, 0.56 \,\}$$

(c)

Figure 24: (a) A sample set of objects *P* with the blue shaded objects indicating those selected for a template configuration. (b) The pair-wise distance matrix for the template configuration. (c) Spatial descriptor from the pair-wise distance matrix.

This representation of the spatial configuration encodes the spatial distances between objects. Additionally, because all pair-wise distances are utilized, it also maintains some notion of the angles between objects. Most importantly, this simple vector can be computed quickly, which is crucial when the number of target configurations reaches the hundreds of thousands or millions. Additionally, because the elements are sorted, similar configurations can be identified without object-to-object correspondences being explicitly determined. This allows the algorithm to quickly prune a large portion of the target configurations before more closely examining the more promising candidate configurations.

Once descriptors have been computed for each target and object configuration, the algorithm then searches for the most similar template/target configuration pairs (line 11) with the following

$$\Gamma = \left\{ \left(c_i^P, c_j^Q \right) \middle| \, configMatch\left(c_i^P, c_j^Q \right) \leq t_{match} \right\} \tag{16}$$

where

$$configMatch\left(c_i^P, c_j^Q \right) = \left\| \delta(c_i^P) - \delta\left(c_j^Q \right) \right\| \tag{17}$$

After similar template/target configuration pairs are discovered, the individual object-to-object correspondences between each pair must be determined. Because the object sets are small, a brute-force search is conducted, with $\psi(\Gamma, \mathrm{i})$ representing the i-th combination of 1-to-1 mappings of objects. For a given pair of configurations, two objects from the template configuration are assigned to two objects from the target configuration. Then, using these assigned mappings, the expected locations of the

remaining objects in the target configuration are computed by taking advantage of the concepts of similar triangles and geometric triangulation.

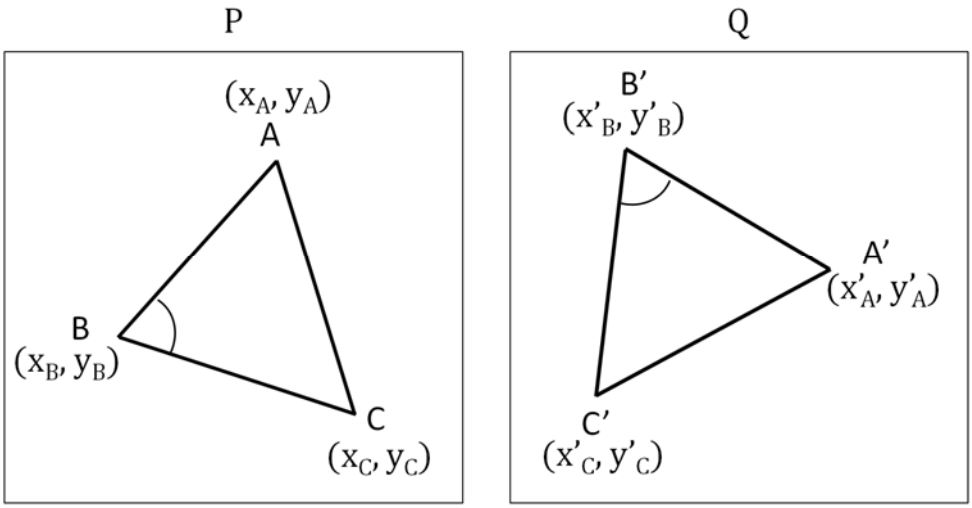


Figure 25: Three sets of points from P and Q, labeled appropriately to demonstrate the computations (Equations 5-8) for determining the coordinates of C' given all the distances and angles of the triangle in P and the object correspondences of A→A' and B→B'.

Let $p_A \rightarrow q_{A'}$ and $p_B \rightarrow q_{B'}$ be two assigned mappings, and $p_C \in c_l^P$ be one of the currently unassigned objects from the template configuration, as in Figure 25. The expected position of $q_{C'}$ in Q is computed using the function below:

$$tri(p_A \rightarrow q_{A'}, p_B \rightarrow q_{B'}, p_C) = (x_p, y_p) = B' + \|BC\| * \vec{v} \qquad (18)$$

where $B' = (x'_B, y'_B)$ corresponds to the center coordinates of $q_{B'}$ and $\vec{v}$ is the unit vector pointing from $q_{B'}$ to $q_{C'}$ and is represented by

$$\vec{v} = (v_x, v_y) \tag{19}$$
$$= (\cos ABC * u_x - dir * (\sin ABC * u_y), cosABC * u_x + dir * (\sin ABC * u_y))$$

where ABC represents the angle at p_B between p_A, p_B, and p_C, $\vec{u}$ is the unit vector pointing from q_B, to q_A, computed by

$$\vec{u} = (u_x, u_y) \tag{20}$$
$$= \left(\frac{(x'_B - x'_A)}{\sqrt{(x'_B - x'_A)^2 + (y'_B - y'_A)^2}}, \frac{(y'_B - y'_A)}{\sqrt{(x'_B - x'_A)^2 + (y'_B - y'_A)^2}} \right)$$

and finally dir determines the direction of the angle at p_B (clockwise or counterclockwise) from p_A to p_C which can be found using Equation 8.

$$dir = (x_B - x_A)(y_C - y_A) - (y_B - y_A)(x_C - x_A) \tag{21}$$

The alignment error e_p between the expected position of an object and the actual position of an object is then

$$e_p(p_A \to q_{A'}, p_B \to q_{B'}, p_C \to q_{C'}) \tag{22}$$
$$= \|\varpi(p_C) - tri(p_A \to q_{A'}, p_B \to q_{B'}, p_C)\|$$

This process is repeated for all combinations of objects between each similar template and target configurations. The quality of each set of object-to-object correspondences is determined by the accumulated alignment error (e_{acc}), and a particular set of object-to-object correspondences is retained if $e_{acc}\big(\psi(\Gamma, \mathrm{i})\big) \leq t_{acc}$.

Consider the situation in Figure 26, in which there is a template configuration from P on the left and a target configuration from Q on the right. Suppose that the colored items were paired by the algorithm (e.g. 1→a and 2→b). Because the spatial configuration should be maintained in Q, using Equations (18)-(21), we can compute the expected position of object 3 in Q, which is indicated in the figure by an "X". The alignment error ($\mathrm{e_p}$) is then the distance of the red line labeled $\mathrm{e_{p,3}}$. Analogously, object 4's position can be predicted and its alignment error also computed. The cost of this object-to-object mapping would then be calculated by: $e_{acc}\big(\psi(\Gamma, \mathrm{i})\big) = e_{p,3} + e_{p,4}$.

After high quality object-to-object correspondences are found between similar template/target configuration pairs, the Aggregated Match Matrix (AMM) is constructed. This is an $n_P \; x \; n_Q$ matrix used to combine all the mapping correspondences from each similar template/target configuration. Each position in the matrix $AMM(j,k)$ corresponds to the number of times $p_j \rightarrow q_k$ appears across all object-to-object mappings from all template/target configuration pairs.

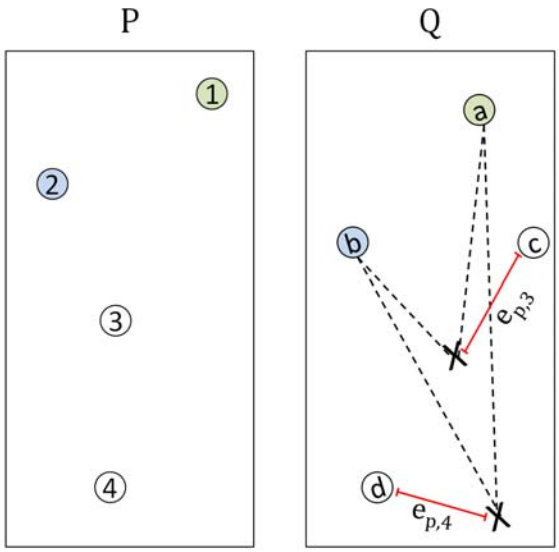


Figure 26: A template configuration of four objects from P (left) and a target configuration of objects from Q (right). Pre-assigned correspondences are denoted by matching colors (1→a, 2→b). The computed expected locations of objects 3 and 4 in Q based on the pre-assigned correspondences are marked by X's, with the alignment errors shown in red.

In the ideal case, all object-to-object correspondences would agree, and the AMM matrix would appear similar to Figure 27(a), in which each row or column only contains nonzero values in a single position. While these situations do occur in practice, oftentimes the AMM matrix contains a degree of uncertainty for some object correspondences (see Figure 27(b)). In these situations, an object from P may have correspondences to multiple objects in Q, or an object from Q may have been mapped to multiple objects in P.

(a)

	q_o	q_1	q_2	q_3	q_4	q_5	q_6	q_7
p_o	16	0	0	0	0	0	0	0
p_1	0	0	10	0	0	0	0	0
p_2	0	0	0	0	0	0	18	0
p_3	0	19	0	0	0	0	0	0
p_4	0	0	0	11	0	0	0	0
p_5	0	0	0	0	0	6	0	0
p_6	0	0	0	0	0	0	0	0

(b)

	q_o	q_1	q_2	q_3	q_4	q_5	q_6	q_7
p_o	8	0	0	0	8	2	0	0
p_1	0	0	10	0	0	0	0	0
p_2	0	0	0	0	0	1	18	0
p_3	0	17	0	0	2	0	0	0
p_4	0	0	0	11	0	0	0	0
p_5	0	0	0	0	0	6	0	0
p_6	0	0	0	0	0	0	0	1

Figure 27: (a) An ideal AMM (Aggregated Match Matrix) with full agreement on all object-to-object correspondences since there is only one nonzero field in each row or column. (b) A second AMM showing some uncertainty among object assignments.

The algorithm utilizes the certainty and/or uncertainty in the AMM matrix by converting these values into probabilities for object mappings, according to Equation (23), where s_t corresponds to the number of times this p_j has been mapped, $s_p$corresponds to the accumulated quality of all the s_t mappings, and s_p/s_t is the current probability of this mapping. These operands are maintained separately to facilitate the later refinement of object correspondence probabilities.

$$\text{prob}(p_j, q_k) = (s_p, s_t) \tag{23}$$
$$= \left(AMM(j,k), \max\left(\sum_{l=0}^{n_P} AMM(l,k), \sum_{l=0}^{n_Q} AMM(j,l) \right) \right)$$

Only those object correspondences with sufficient probabilistic evidence, those whose probability meet or surpass the t_{anchor} threshold, are labeled as "anchor correspondences," as noted below.

$$\varsigma_{P,Q} = \{ p_j \rightarrow q_k \mid prob(p_j, q_k) = (s_p, s_t) \wedge s_p / s_t \geq t_{anchor} \}$$

This Spatial Configurations algorithm requires at least two anchor correspondences to continue to the next step. If $|\varsigma_{P,Q}| < 2$, then the algorithm is repeated using a new set of template configurations.

4.3.1.2 Modified Point Pattern Matching Approach

The second method for identifying anchor correspondences is a modified version of the nearest neighbors point pattern matching algorithm described in by van Wamelen in [70]. The modifications to the original algorithm, which appear in Algorithm 3, are mentioned clearly where they occur. The rationale for this approach is to try and utilize the underlying idea behind point pattern matching to attempt to align as many stable points as possible between *P* and *Q*. This approach is not expected to match objects that have merged or undergone significant growth or match areas where a regional distortion has occurred; however, it can provide a baseline set of correspondences from which a more complete set of correspondences can be computed.

```
1.  FOR EACH p ∈ P
2.      FOR EACH q ∈ Q
3.          IF (nearest neighbors of p and q match)
4.              IF (local match extends to global match)
5.                  Output transform and object correspondences
6.                  RETURN Success
7.              END IF
8.          END IF
9.      END FOR
10. END FOR
11. RETURN Failure
```

Algorithm 2: Pseudo-code for the main loop of the modified point pattern matching algorithm for establishing anchor correspondences.

The overall structure of this algorithm is provided in Algorithm 2. The algorithm begins by exhaustively matching every object in *P* with every object in *Q*. For each pair of selected objects *p* and *q*, the algorithm first determines if there is a local match by attempting to match nearest neighbors. This is performed by iteratively matching each pair of nearest neighbors. Let *a* and *b* be a nearest neighbor of *p* and *q*, respectively. First the unique affine transform $T = \{s, \theta, t_x, t_y\}$ that maps p to q and a to b is computed according to Equations X-Y, where s is the scale, θ is the rotation angle, and t_x and t_y are the x- and y- translations.

$$s = \frac{|\overrightarrow{qb}|}{|\overrightarrow{pa}|} \tag{24}$$

$$\theta = cos^{-1}\left(\frac{\overrightarrow{pa} \cdot \overrightarrow{qb}}{|\overrightarrow{pa}| \cdot |\overrightarrow{qb}|}\right) \tag{25}$$

$$t_x = q_x - p_x \cdot s \cdot cos\theta + p_y \cdot s \cdot sin\theta \tag{26}$$

$$t_y = q_y - p_x \cdot s \cdot sin\theta - p_y \cdot s \cdot cos\theta \tag{27}$$

The percentage of p's nearest neighbors that map to q's nearest neighbors using this transform is then determined. If this percentage is sufficient, the local match criteria is satisfied and the algorithm then tries to extend this local match globally according to the pseudo-code in Algorithm 3. Otherwise, the algorithm continues with other nearest neighbors or other objects.

```
1.  DO
2.      Compute T_avg, the best least-squares transform for matched objects
3.      MatchedSet = { }
4.      FOR EACH p ∈ P
5.          Compute projected position T_avg(p) of object in destination frame
6.          IF(∃ q ∈ Q such that |T_avg(p) − q| < t)
7.  *           IF(q ∉ MatchedSet AND q meets object criteria)
8.                  MatchedSet = MatchedSet ∪ {p → q}
9.              END IF
10.         END IF
11.     END FOR
12. *WHILE (MatchedSet grows in size)
13. *IF ( MatchedSet large enough) RETURN Success
14. RETURN Failure
```

Algorithm 3: Pseudo-code for attempting to extend a local nearest neighbors match to a global match. The lines with * indicate places where the original algorithm has been modified.

Let $MatchedSet$ initially be the set of nearest neighbors that were matched locally. Using the objects in this set, the best least-squares transform $T_{avg} = \{t_x, t_y, s \cdot sin\theta, s \cdot cos\theta\}$ that minimizes $\sum_{i=1}^{l} |T(a_i) - b_i|^2$, which can be computed (line 1) based on the following equations.

$$\mu_{x_A} = \sum_{i=1}^{l} x_{a_i} \qquad \mu_{x_B} = \sum_{i=1}^{l} x_{b_i} \tag{28}$$

$$\mu_{y_A} = \sum_{i=1}^{l} y_{a_i} \qquad \mu_{y_B} = \sum_{i=1}^{l} y_{b_i} \tag{29}$$

$$l_{A+B} = \sum_{i=1}^{l} \left(x_{a_i} x_{b_i} + y_{a_i} y_{b_i}\right) \qquad l_{A-B} = \sum_{i=1}^{l} \left(x_{a_i} y_{b_i} - y_{a_i} x_{b_i}\right) \tag{30}$$

$$l_A = \sum_{i=1}^{l} \left({x_{a_i}}^2 + {y_{a_i}}^2\right) \qquad D = l * l_A - {\mu_{x_A}}^2 - {\mu_{y_A}}^2 \tag{31}$$

$$T_{avg} = \frac{1}{D} \begin{pmatrix} l_A & 0 & -\mu_{x_A} & \mu_{y_A} \\ 0 & l_A & -\mu_{y_A} & -\mu_{x_A} \\ -\mu_{x_A} & -\mu_{y_A} & l & 0 \\ \mu_{y_A} & -\mu_{x_A} & 0 & l \end{pmatrix} \begin{pmatrix} \mu_{x_B} \\ \mu_{y_B} \\ l_{A+B} \\ l_{A-B} \end{pmatrix} \tag{32}$$

Once this transform is computed, it is utilized to compute the projected position of every object *p* in *P*. If an object in Q exists that is close to this projected position that has not been mapped to another object and that does not violate any constraints on the object mapping, then that object in *Q* is marked as matching the current *p* in the *MatchedSet*. These object constraints represent the first modification to the original algorithm. For the current task of mapping necrotic lesions, the relevant object constraints are that objects should not shrink; areas of necrosis either remain of stable size or grow. Other constraints may be appropriate in other situations. For example, if chlorotic lesions (areas of reduced chlorophyll) were also being mapped, the algorithm would not want to map necrotic lesions to chlorotic lesions, but all other combinations of mapping these lesions together would be valid.

After the mapping of all objects in *P* have been attempted with the average transform, the size of *MatchedSet* is compared to its size when the loop began. If the number of correspondences has increased, the entire process is repeated again. Once the maximal number of correspondences has been reached, the success of the global mapping is decided by the size of *MatchedSet*, specifically whether it surpasses the percentage of objects provided as an input parameter to the algorithm. If it does not, then, as discussed earlier, the algorithm returns to trying to find a local match. Otherwise, the set of correspondences in *MatchedSet* are retained as the set of anchor correspondences. These loop conditions also represent a modification to the original algorithm. In the original version, the looping continues until *MatchedSet* stops growing or until that set exceeds the success percentage.

4.3.2 Growing Anchor Correspondences

Following the identification of anchor correspondences, the algorithm then proceeds to the second step in which these anchor correspondences are utilized to iteratively determine additional object mappings and to refine existing object mappings. This step is performed through finer mapping by examining local neighborhoods around each object in *P* to improve or reduce the confidence in an object's mapping. The pseudo-code for this portion is provided below in Algorithm 4.

During this portion of the algorithm, a local neighborhood around each object *p* in *P* is utilized for mapping (line 3). Let N_i be the neighborhood of the *k* nearest objects around p_i. The size of the neighborhood is a parameter of the algorithm.

1. DO
2. FOR EACH $p \in P$
3. N = getKNearestNeighbors(p)
4. *C1* = { $q \mid q \in Q \wedge \left|T_{avg}(p) - q\right| < d_1$ }
5. *C2* = { $q \mid \exists\, pixel(q) \in Q \wedge \left|T_{avg}(p) - pixel(q)\right| < d_2\} \cup \{ noObject\}$
6. *Voters* = getReferenceSet(p)
7. *curMap* = { }; *accAlignmentError* = 0;
8. FOR EACH $c \in \{Cands1 \cup Cands2\}$
9. *curMap* = *curMap* $\cup$ $\{ p \rightarrow c\}$
10. RecursiveNeighborHoodMapping(*curMap, N, Voters, accAlignmentError*)
11. *curMap* = *curMap* – $\{p \rightarrow c\}$
12. Update probabilities of best mappings
13. END FOR
15. END FOR
16. WHILE (stopping criteria)

Algorithm 4: Pseudo-code for Growing Anchor Correspondences step.

Using the best least-square transform for the anchor correspondences, the projected position of an object *p* is computed. The algorithm then determines candidate objects *q* in *Q* that may correspond to *p*. Two sets of candidates are found. The first set of candidates *C1* (line 4) corresponds to the objects in *Q* whose centers are sufficiently close, determined by the distance threshold d_1, to the project position $T_{avg}(p)$. The second set of candidates *C2* (line 5) are those objects in *Q* whose centers are not in *C1* but who contain a pixel that is also sufficiently close to the project position of *p*. The rationale for this second set of candidates is that it allows the algorithm to match to objects whose centers have shifted substantially because of growth or merging. Additionally, *C2* also contains an option that allows this object to have no corresponding match in *Q*, which allows some objects to disappear.

The algorithm proceeds by attempting a mapping of N_i based on the pairing of p and each candidate from the sets above (lines 8-12). In order to find the projected positions of the remaining neighbors in Q, a set of reference objects is utilized. These reference mappings are a subset of anchor correspondences. While a single reference point in conjunction with the pairing $p \rightarrow c$ is sufficient to triangulate the positions of the remaining neighbors, multiple reference points are utilized to provide consensus on the projected positions. In this way, even if some of the anchor correspondences are incorrect, the predicted positions of the neighbors will not be affected so long as half the reference correspondences are correct.

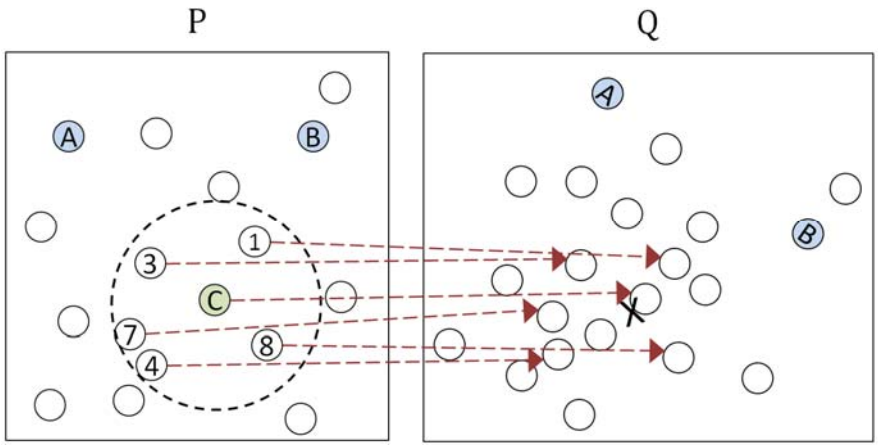


Figure 28: (Left) Illustration of a local neighborhood (enclosed by the dashed circle) around the object p_C in P, with the reference points A and B shown in blue. The radius that defines the neighborhood is adapted to contain the k nearest objects. (Right) Illustration of the projected objects in Q. The X indicates the median expected position of object p_C based on all the reference points, and the dotted arrows show the object correspondences for the entire neighborhood.

Mapping of all the neighbors of p with respect to the reference points and the pairing $p \rightarrow c$ is performed recursively (see Figure 28). At each step, the alignment error between the predicted position using the reference points and the actual position of an object in *Q* is accumulated. The total alignment error is maintained for each mapping as an indicator of mapping quality. The highest quality mappings, determined by a threshold on average alignment error, are maintained. The object-to-object correspondences for the mapping of $p \rightarrow c$ is then updated according to Algorithm 5.

1. FOR EACH good mapping of $p \rightarrow c$
2. $eAvg$ = average alignment error
3. $prob(p \rightarrow c) = (s_p + e^{-eAvg/d}, s_t + 1)$
4. END FOR

Algorithm 5: Pseudo-code for updating mapping probabilities.

The average alignment error for each high quality mapping is computed by dividing the total alignment error by the number of neighbors mapped. A negative exponential is then used to compute the probability of that mapping based on the average alignment error, with larger errors corresponding to smaller probabilities.

Once the probabilities for that candidate mapping are updated, the loop repeats with a different candidate for *p*. The local mapping continues until all candidates are attempted, and then this is repeated for all the other objects in *P*. This completes one iteration of the Growing Anchor Correspondences set. During this iteration, a measure of the overall quality of all mapping, the total mapping alignment error, is computed by summing the alignment errors for all objects based on their highest quality mappings. The entire process of iteratively remapping local neighborhoods continues while the

total mapping alignment error decreases. The algorithm returns all object-to-object correspondences as well as their mapping probabilities.

4.4 Analyzing Entire Time Courses

Let $F = \{f_i | \, i = 1,2, \ldots n_f \}$ be a time course study consisting of a series of image frames f_i. The strategy for mapping an entire time course *F* is to utilize the Variable Object Pattern Matching algorithm described above in a two-tier iterative design. In the first tier, the Variable Object Pattern Matching algorithm is performed where $P = f_i$ and $Q = f_{i+1}$ for each $f_i \in F$, as shown in Figure 29.

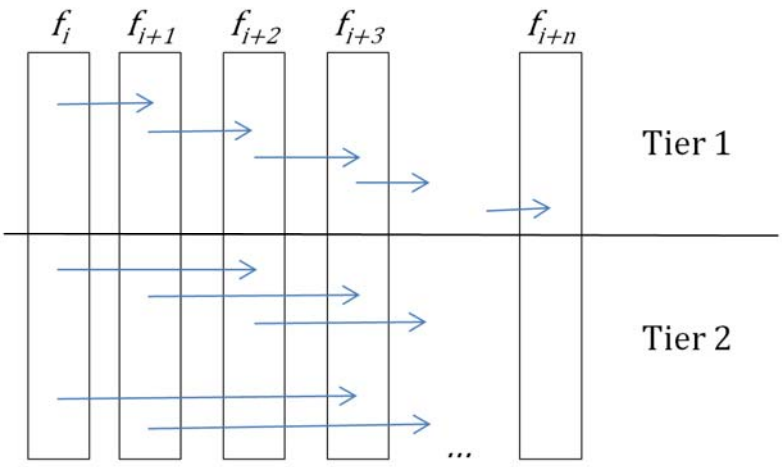


Figure 29: Illustration of the progression of frame registration in the proposed algorithm. The first tier is depicted as the top set of arrows, in which objects are registered across consecutive frames. Underneath that is the second tier, in which objects are attempted to be registered across nonconsecutive images

Ideally, highly confident mappings will be identified for all objects after the first tier; however, in practice, uncertainty undoubtedly remains. For example, situations where additional examination may be required include when objects do not have a

corresponding lesion in f_{i+1} or when the probability of the highest mapping is not satisfying. In an attempt to decrease this uncertainty, the second iterative tier can be initiated. In this stage, the Variable Object Pattern Matching algorithm is utilized for nonconsecutive images, i.e. $P = f_i$ and $Q = f_{i+t}$ where $t > 1$. For a given frame f_i, this process starts at $t = 2$ with t continuing to increase until anchor correspondences can no longer be found between the two frames, until all objects are accounted for completely, or until prediction confidence in mapping falls below a threshold.

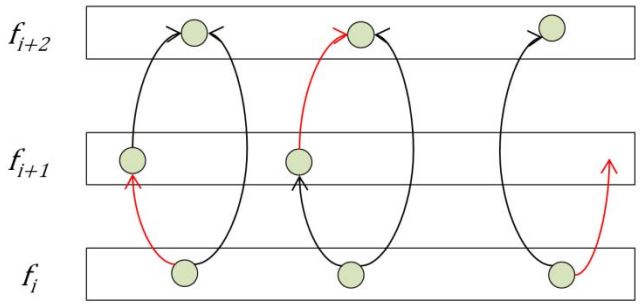


Figure 30: Depictions of three situations that can be resolved by mapping to more than the consecutive frame. In each situation, the black arrows indicate confident mappings made by the algorithm, and the red lines represent links missed by the algorithm. Both the left and middle situations utilize confident mappings to the f_{i+2} frame to resolve a mapping missed by an earlier frame. The right situation demonstrates a situation where a lesion disappeared in a frame, but the mapping to the next frame allows the object to continued to be tracked in future frames.

Some situations that can be resolved through the mapping of nonconsecutive images as shown in Figure 30. In the left situation, frames f_i and f_{i+1}both contain reliable mappings to the same object in f_{i+2}, which is denoted by the black links. However, this

same object from f_i does not have a reliable mapping to f_{i+i}, the red line. The ability to attempt mapping across nonconsecutive images allows the algorithm to fill in the missing red link in this situation. An analogous situation (see the middle of Figure 30) occurs when f_i contains a reliable mapping to objects in both f_{i+1} and f_{i+2}, but f_{i+1} shows no corresponding lesion or contains an unreliable mapping to an object in f_{i+2}. Again, the mapping across nonconsecutive images allows this situation to be detected and potentially resolved. Additionally, the example on the right in Figure 30 demonstrates the algorithm's potential ability to reestablish links to an object that disappeared in a frame. This may occur due to variations in the field of view between images. While the object's behavior in f_{i+i} is unknown, continued tracking of the object is possible. Examples of some of these cases from real data are shown below in the results section.

4.5 Modeling Temporal Changes

Once objects have been mapped across a series of frames, the temporal behavior of individual objects can be assessed. Let $\mathrm{M}(o_j^i) = \{o_l^k | o_j^i \gg o_l^k\}$ be the sequence of objects to which object o_j^i corresponds, where $o_j^i \gg o_l^k$ means that object o_j^i eventually corresponds to o_l^k. If $k = i + 1$, this object corresponds to the mapping in the next consecutive frame; otherwise, the mapping represents a jump across multiple frames.

Additionally, let $\mathrm{H}(o_j^i) = \{h_{i,j,k}\}$ be a set of k features calculated on object o_j^i that describes various static properties or characteristics of the object. Calculated features for the maize lesions include area, roundness, perimeter, and maximum diameter. These features describe in various ways both the size and shape of the objects. A feature

vector representing the temporal signature for an object o_j^i could then be constructed of the direct feature values, as in $v_j^i = \{h_{x,y,k} | o_y^x \in \mathrm{M}(o_j^i)\}$, or using the difference between feature values of mapped objects, as in $v_j^i = \{h_{x,y,k} - h_{x+1,z,k} | o_y^x \in \mathrm{M}(o_j^i) \wedge o_y^x \rightarrow o_y^x\}$.

The temporal behavior for individual objects can then be modeled using Markov chains. Let $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ be a graph corresponding to a Markov chain in which the set of vertices $\mathrm{V} = \{\mathrm{X}_1, \mathrm{X}_2, \ldots\}$ represents a set of states an object may attain during its temporal progression and the set of directed edges $\mathrm{E} = \{\mathrm{e}_{\mathrm{i,j}}\}$ where $\mathrm{e}_{\mathrm{i,j}} = p(X_i | X_j)$ represents the probability of an object transitioning from state X_i to state X_j. This probability is computed according to:

$$p(x_i | x_j) = w_j * \left(\frac{Count\ of\ transition\ from\ x_j\ to\ x_i}{All\ transition\ start\ from\ x_j} \right) = w_j * \left(\frac{\mathrm{c}_{x_j x_i}}{\sum_{x_i} \mathrm{c}_{x_j x_i}} \right)$$

Each state in the Markov chain can be thought of as representing a semantic term describing a particular temporal behavior. For example, in the case of maize lesions, one might be interested in an object's size transitions, in which case the states might be defined as: "Shrink", "Stable", "Mild Growth", "Moderate Growth", and "Substantial Growth."

In order to construct the Markov chain for an object's behavior, the object's temporal signature must be translated into a series of states. This can be accomplished by manually defining ranges of feature values that correspond to different states. An alternative method in which the feature ranges that correspond to individual states are automatically learned from user input is described as future work in Section 7.2.4. The probabilities for transitioning between each state are then computed using the equation

above, and the Markov chain model is constructed. An example of an object's feature vector as well as its translation into a sequence of states and the corresponding Markov chain model are shown in the results section.

4.6 Results and Discussion

The developed Variable Object Pattern Matching algorithm facilitates the mapping of objects between image frames based on the spatial configurations of otherwise indistinct objects. Via a two-tier iterative process, this algorithm can be utilized to map objects across time course studies.

To evaluate mapping accuracy of the Variable Object Pattern Matching algorithm, experiments were conducted. Comparisons were made between using Spatial Configurations and the Modified Nearest Neighbors Point Pattern Matching algorithms for step 1. Additionally, the original point pattern matching algorithm by [70] was included as a comparison.

Furthermore, case studies are provided to demonstrate the benefits of mapping to nonconsecutive frames and also to illustrate the process of modeling temporal behavior for both individual objects and full leaves. All experiments were conducted on a standalone development server with Intel X5570 dual 2.93 GHz quad core CPU and 72 GB of RAM.

4.6.1 Mapping Accuracy

In order to evaluate the accuracy of the algorithm's predicted object-to-object correspondences, several studies were conducted. The first experiment involved evaluating the algorithm accuracy with respect to all objects in an image. To

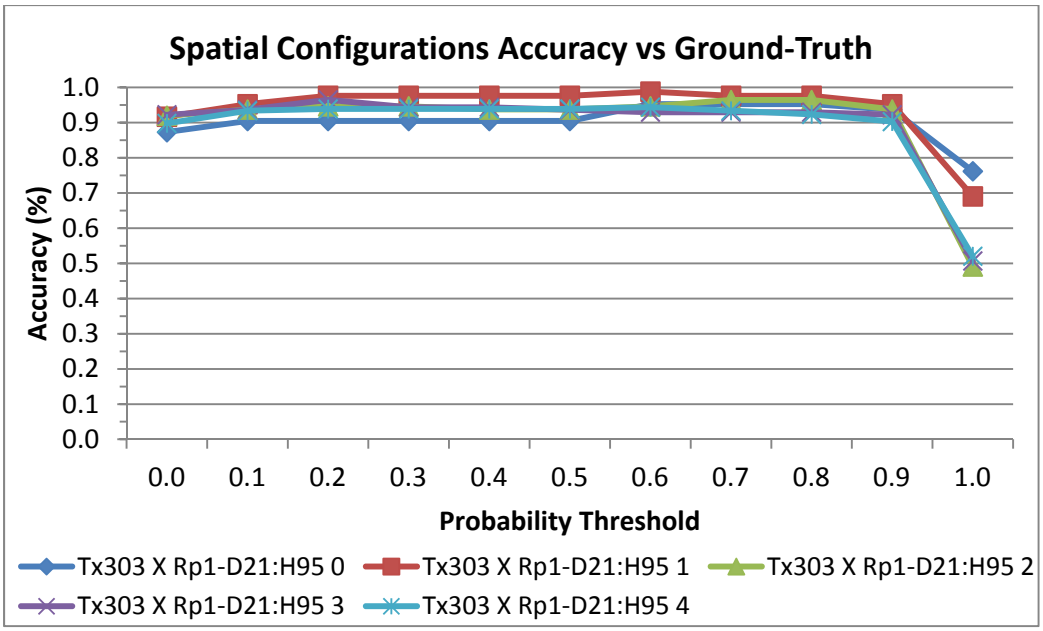
accomplish this, a series of five consecutive frames (see **Error! Reference source not found.**) with ground-truth object correspondences were utilized to compare algorithm results. In total, this represented an evaluation of ~600 object mappings.

Table 6: Information about the image sequence with ground-truth object correspondences.

Line	Frame #	# of Necrotic Objects
Tx303 X Rp1-D21:H95	0	63
Tx303 X Rp1-D21:H95	1	84
Tx303 X Rp1-D21:H95	2	112
Tx303 X Rp1-D21:H95	3	142
Tx303 X Rp1-D21:H95	4	196

Despite the number of objects increasing in each frame, which in and of itself is evidence of phenotype progression, not all objects in f_i have a corresponding object in f_{i+1}. This may be due to variation in a lesion's appearance because of leaf waviness and/or imperfection of the lesion segmentation algorithm. Regardless, a probability threshold was used in this experiment, in which all mappings with probabilities above the threshold are maintained (i.e., the mapped object is compared to the manually verified object) and all those below the threshold are discarded (i.e., the object is treated as having no corresponding lesion).

Figure 31 (top) shows the overall mapping results when using the Spatial Configuration algorithm for step 1 for each of these images based on varying the probability threshold, with the aggregated average mapping results shown in Figure 31 (bottom).



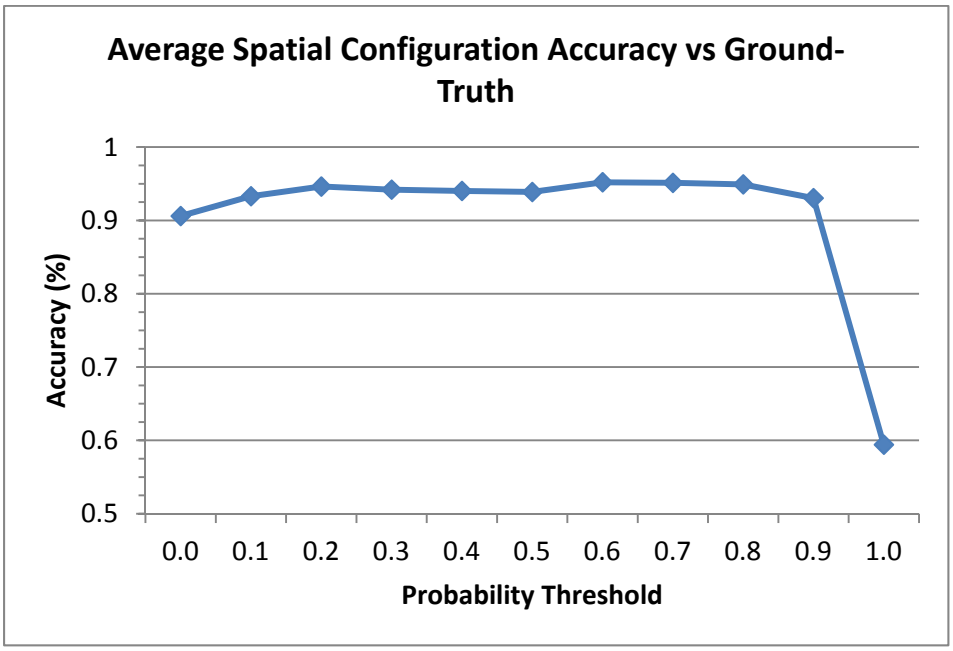


Figure 31: Mapping accuracy for the image sequence with ground-truth correspondences based on a probability threshold, showing (top) the accuracy for each individual image and (bottom) the average accuracy over all images. For a given threshold, if the probability of a mapping is below the threshold, the initial object is said to have no corresponding object in the subsequent frame. The large drop in accuracy between thresholds of 0.9 and 1.0 indicates that most of the correct mappings for these images had probabilities above 0.9.

The accuracy results are fairly stable across all values for the threshold (with the exception of requiring 100% confidence) meaning that most of the probabilities in this set are above the 0.9 level. However, for those which the algorithm has more uncertainty, the best results are obtained when the probability threshold is set around 0.6, and so this threshold will be utilized for the remainder of this section.

Study of the bad predictions reveals a few classes of errors. The most common error occurs when the correct object in f_{i+1} is significantly smaller than the original object in f_i. This is actually an expected error, as the algorithm assumes that objects will roughly maintain their size or grow. This assumption makes sense intuitively with regard to the real behavior of necrotic lesions and it also reduces the search space by imposing a size constraint when choosing candidate target objects during mapping. In this experiment, if the correct mapped object has an area that is less than 80% of the area of the original object, it will be skipped in the selection process. These accounted for 35% of the total errors. Other causes of error include confusion with adjacent objects and significant relative changes in the center point of objects due to large growth.

The two variations of the Variable Object Pattern Matching algorithm (using spatial configurations and the modified NN point pattern matching (modified NN PPM) for the initial step) were compared against the original nearest neighbors point pattern matching (NN PPM) algorithm proposed by van Wamelen [70].

First, these three algorithms were compared in terms of the image sequence with ground-truth labels (see Table 7). Both approaches to Variable Object Pattern Matching consistently had higher accuracy than the original NN PPM algorithm.

Table 7: Comparison of mapping accuracy between the Variable Object Pattern Matching algorithm [using Spatial Configurations (left) and the Modified Nearest Neighbor Point Pattern Matching (Modified NN PPM*) algorithm (middle) for identifying anchor correspondences] and the original nearest neighbor point pattern matching (NN PPM*) algorithm.

Source Frame	Spatial Configurations	Modified NN PPM*	NN PPM*
Tx303 x Rp1-D21:H97 - Frame 1	95.2%	93.7%	90.5%
Tx303 x Rp1-D21:H97 - Frame 2	98.8%	95.2%	85.7%
Tx303 x Rp1-D21:H97 - Frame 3	94.6%	98.2%	87.5%
Tx303 x Rp1-D21:H97 - Frame 4	93.0%	97.9%	84.5%
Tx303 x Rp1-D21:H97 - Frame 5	94.4%	74.5%	70.9%

On all frames but the last frame of this sequence, the modified NN PPM approach had a similar to slightly better accuracy than the spatial configurations approach. The last frame, however, showed a drop in performance for both the modified and original NN PPM approaches, whereas the spatial configuration method maintained its accuracy. In addition to a frame-by-frame comparison, this same series was also utilized to evaluate the accuracy of object tracking across all images. Here, an object is said to be tracked successfully if there is a correct frame-to-frame mapping from the first to last frame of the series. Using this criterion, the accuracy of objects that first appeared in frame 0 and continued throughout is 88% using the spatial configurations approach. To assess the algorithm less focally but more broadly, two subjects were recruited to randomly select a subset of objects from a larger pool of images. In particular, they were asked to evaluate mappings of at least 10-20 objects from at least 10 images. Three of these images were to overlap with the images with ground-truth labels, and the remaining were to represent mappings that had not been manually verified. The results

from the two subjects are shown below in Table 8. In this case, only the variations of the Variable Object Pattern Matching algorithm were compared, since the original NN PPM algorithm had poorer performance in the first experiment.

Across the images evaluated, the average accuracy of the object-to-object correspondences was 93% for the Spatial Configuration algorithm and 74.7% for the modified NN PPM. The average running times over these images for the two algorithms also favored the Spatial Configurations algorithm, with running times of 33.1 s and 1203 s, respectively. These data suggest that the Spatial Configurations algorithm outperforms the modified NN PPM algorithm.

In many cases when the modified NN PPM algorithm had a poor performance, an interesting phenomenon was observed. Portions of the leaf would have excellent correspondences (i.e. the center of the leaf, or the right side), but in the remainder of the leaf, the object correspondences would be erroneous. Since all these correspondences were utilized as anchor correspondences, it is not surprising that the overall results suffered in these cases. If the good and bad matches in these sets can be differentiated, the final results using this approach could be improved significantly.

Additionally, though the time for the modified NN PPM algorithm in this case was slower on average, the worst case performance of this algorithm is far better than that of the Spatial Configurations algorithm. Moreover, there is hope for improving the average case performance of this algorithm by ordering the objects from P and Q in such a way as to make it more likely that a match is found quicker. Possible orderings that may improve the run time include ordering the objects by size or location on the leaf.

Table 8: Accuracy by evaluating subsets of objects mappings. Two subjects were recruited to each score the mapping results starting from each image above for the two variations of the Variable Object Pattern Matching algorithm. For each algorithm, the average percentage of objects evaluated is provided (% selected), the average accuracy of those mappings (% correct), and the running time of the algorithm (time).

Source Frame	# Objects in Image	Spatial Configurations			Modified Nearest Neighbors		
		% Selected	% Correct	Time (s)	% Selected	% Correct	Time (s)
M162W x Rp1-D21:H96 Frame 1	29	0.62	97.8%	16.5	0.76	20.6%	0.6
M162W x Rp1-D21:H96 Frame 2	126	0.21	92.3%	175.8	0.28	83.3%	1687.5
M162W x Rp1-D21:H96 Frame 3	220	0.13	96.3%	26.7	0.16	99.1%	1119.8
M162W x Rp1-D21:H96 Frame 4	210	0.26	83.7%	14.2	0.20	82.2%	185.6
Tx303 x Rp1-D21:H97 Frame 1	63	0.45	88.4%	1.5	0.52	98.8%	7.1
Tx303 x Rp1-D21:H97 Frame 2	84	0.44	99.2%	2.5	0.38	100%	36.4
Tx303 x Rp1-D21:H97 Frame 4	142	0.37	92.2%	9.5	0.26	96.8%	269.9
Tx303 x Rp1-D21:H97 Frame 5	196	0.25	97.1%	26.5	0.20	62.5%	463.5
Tx303 x Rp1-D21:H97 Frame 6	276	0.18	93.1%	31.1	0.18	79.2%	7978.1
Tx303 x Rp1-D21:H97 Frame 7	285	0.32	92.8%	69.5	0.16	32.8%	734.3
Tx303 x Rp1-D21:H97 Frame 9	269	0.12	94.8%	31.8	0.14	48.8%	2587.4
Tx303 x Rp1-D21:H97 Frame 11	284	0.09	79.1%	57.2	0.13	44.5%	1771.4
CML228 x Rp1-D21:H99 Frame 9	19	0.77	100.0%	0.9	0.70	100%	0.3
CML228 x Rp1-D21:H99 Frame 10	41	0.71	94.7%	0.4	0.55	97.6%	1.2

The run times for both of these algorithms could be further improved by utilizing massive parallelization with multiple threads or processors. For the Spatial Configurations approach, the set of target configurations to consider could be divided among a set of threads and processed in parallel. Similarly, local matches for pairs of objects from *P* and *Q* could be performed in parallel.

Another area of possible improvement for the Variable Object Pattern Matching algorithm in terms of accuracy would be to use a combination of the options for step 1 in determining anchor correspondences. For example, both variations could be performed, and consensus used to retain the most confident object correspondences.

4.6.2 Case Study for Nonconsecutive Frame Mapping

Consider the mapping of $f_0 \rightarrow f_1$ in the set of frames described below, which have been manually verified. As described below in the results section, the Spatial Configuration algorithm typically makes around six erroneous predications out of 63, and there are 11 objects that have no corresponding lesion in f_1 based on the ground-truth labels.

By including the mapping of $f_0 \rightarrow f_2$, we are able to demonstrate some of the advantages described above in a real-world example. First, we demonstrate the ability to reestablish connections with some objects vanished in f_1. This in fact happened with four objects located near the tip of the leaf in this set (all are shown in Figure 32).

The inclusion of the mapping of $f_0 \rightarrow f_2$ also demonstrates how a previous past mistake might be corrected, as demonstrated in the left example of Figure 30. Object 15 (the object enclosed by the top right circle) was initially labeled as having no corresponding lesion in the mapping of $f_0 \rightarrow f_1$; however, the mapping to f_2 generated a correct mapping for object 15 with a confidence of 0.676 (which passes the confidence threshold), and in the mapping of $f_1 \rightarrow f_2$, there is a very high probability mapping (0.983) of object 59 to object 15. Together, these data support the relabeling of object 15 in f_0 to the correct object (59).

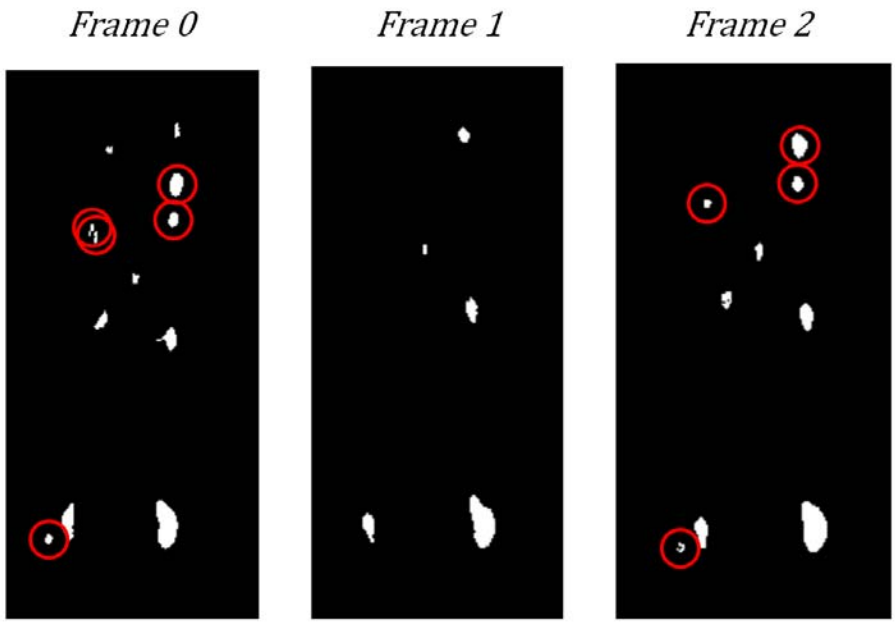


Figure 32: Examples of lesion segmentation from frames 0, 1, and 2 from the test set above. Within the shown portion of these leaves are four lesions (shown in red circles) that appeared in frame 0, disappeared in frame 1, but then reappeared in frame 2. The ability to map $f_0 \rightarrow f_2$ allows continued tracking of these lesions.

4.6.3 Temporal Object Modeling

Using the procedure described in Section 4.5, Markov models can be constructed based on an individual object's temporal behavior. Here, we model temporal behavior in terms of lesion size using specifically the percentage change in object area as the tracked measurement. Table 9 contains a listing of the states and the criteria for belonging to each state.

Table 9: Defined states for modeling lesion growth along with the membership criteria.

State	Criteria
Shrink	$\Delta area \leq -30\%$
Stable	$-30\% < \Delta area \leq 10\%$
Mild Growth	$10\% < \Delta area \leq 50\%$
Moderate Growth	$50\% < \Delta area \leq 100\%$
Substantial Growth	$\Delta area > 100\%$

As examples, consider the vector of area $v_{orig} = \{27, 107, 112, 125, 132\}$ measurements for an object from the manually verified series, where each position corresponds to its area in that frame (starting with frame 0). This can be converted to the following vector of growth percentages $v_{growth} = \{296.3\%, 4.7\%,\ 11.6\%,\ 5.6\%\}$, where each position in this vector represents the percentage increase in area for the object between subsequent frames. Using the criteria above, the growth vector is converted to a sequence of states $v_{states} = \{Substantial\ Growth,\ Stable,\ Mild\ Growth,\ Stable\}$, which can be converted to the Markov chain model in Figure 33.

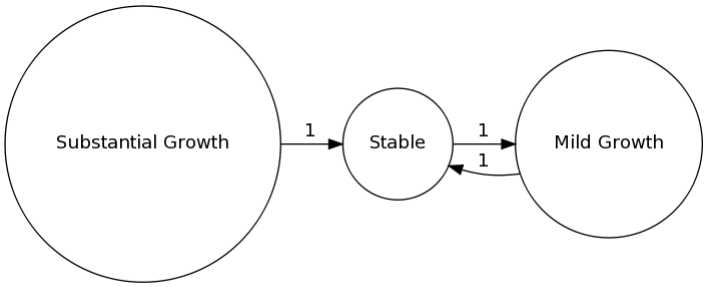


Figure 33: Corresponding Markov chain for the example object's temporal behavior based on growth.

Additionally and more interestingly, individual object Markov chains can be aggregated to model average lesion behavior on an entire leaf over time. This is done by using the same set of states and computing transition probabilities over all objects on the leaf. Figure 34 (top) shows the generated Markov model for the objects originating in frame 0 of the manually verified image sequence. As a comparison, the corresponding model obtained using the algorithm results for the same frames as shown in Figure 34(bottom). Additionally, Figure 35 shows a third Markov chain model for the same time course that includes tracking objects for an additional six frames.

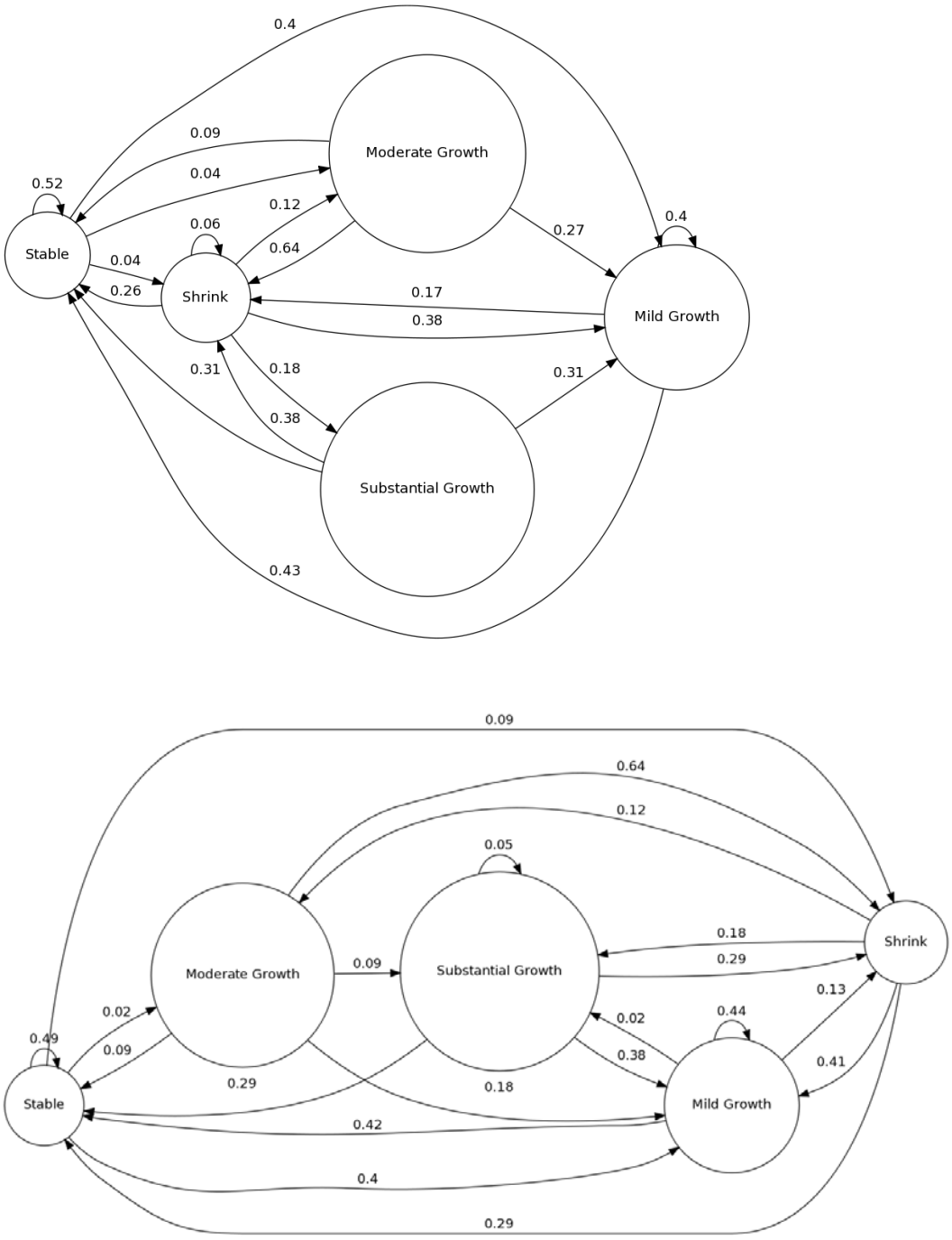


Figure 34: Aggregated Markov chain model of average lesion growth behavior. (Top) Model with correct transition probabilities, which were computed using manual mapping results. (Bottom) Model with transition probabilities computed using automated mapping frames for the same image frames.

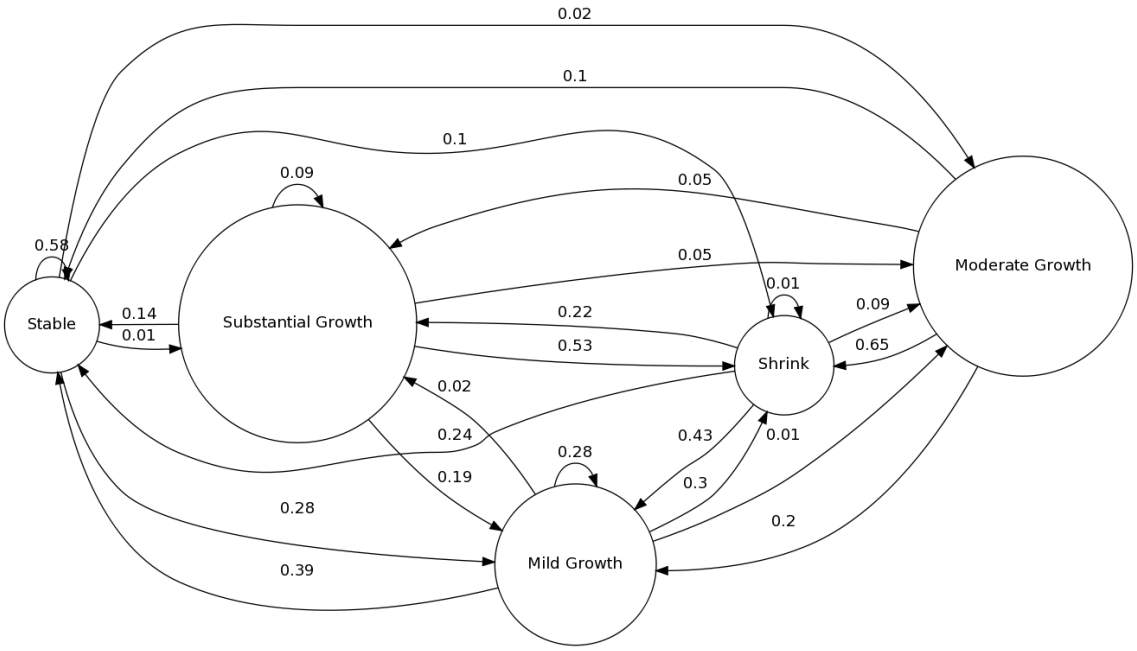


Figure 35: Aggregated Markov chain model of average lesion growth behavior for the same leaf as Figure 34 except with the inclusion of an additional six frames.

CHAPTER FIVE

WEB APPLICATION PHENOPHYTE

To provide the plant community with the ability to easily and readily utilize the processing pipeline related to area-related phenotypic traits (including herbivory and growth), a publically available web-based application called PhenoPhyte was developed. This software, available at http://PhenomicsWorld.org/PhenoPhyte, allows users to automatically and efficiently measure these traits from imagery using a variety of experimental setups. The web application was designed to handle images of individual leaves (both detached and *in situ*) and rosettes and it also permits imaging of multiple objects in a single frame. To save plant scientists from tedious spreadsheet work after extraction of phenotype trait data from single images, PhenoPhyte also provides a means to track objects across images and provide measurements of temporally varying traits such as growth and herbivory.

5.1 Description of Software

The computational mechanisms underlying PhenoPhyte represent a unique combination of existing image processing techniques and novel algorithms. Existing CVIP algorithms chosen for their efficiency and accuracy include Sobel and Canny edge detection, connected components labeling, and segmentation via thresholding. Novel algorithms include (1) the image normalization algorithm discussed previously, (2) a component to track relative plant/pot positions across image sequences, and (3) linking objects and their measurements across image sequences to calculate phenotypic changes.

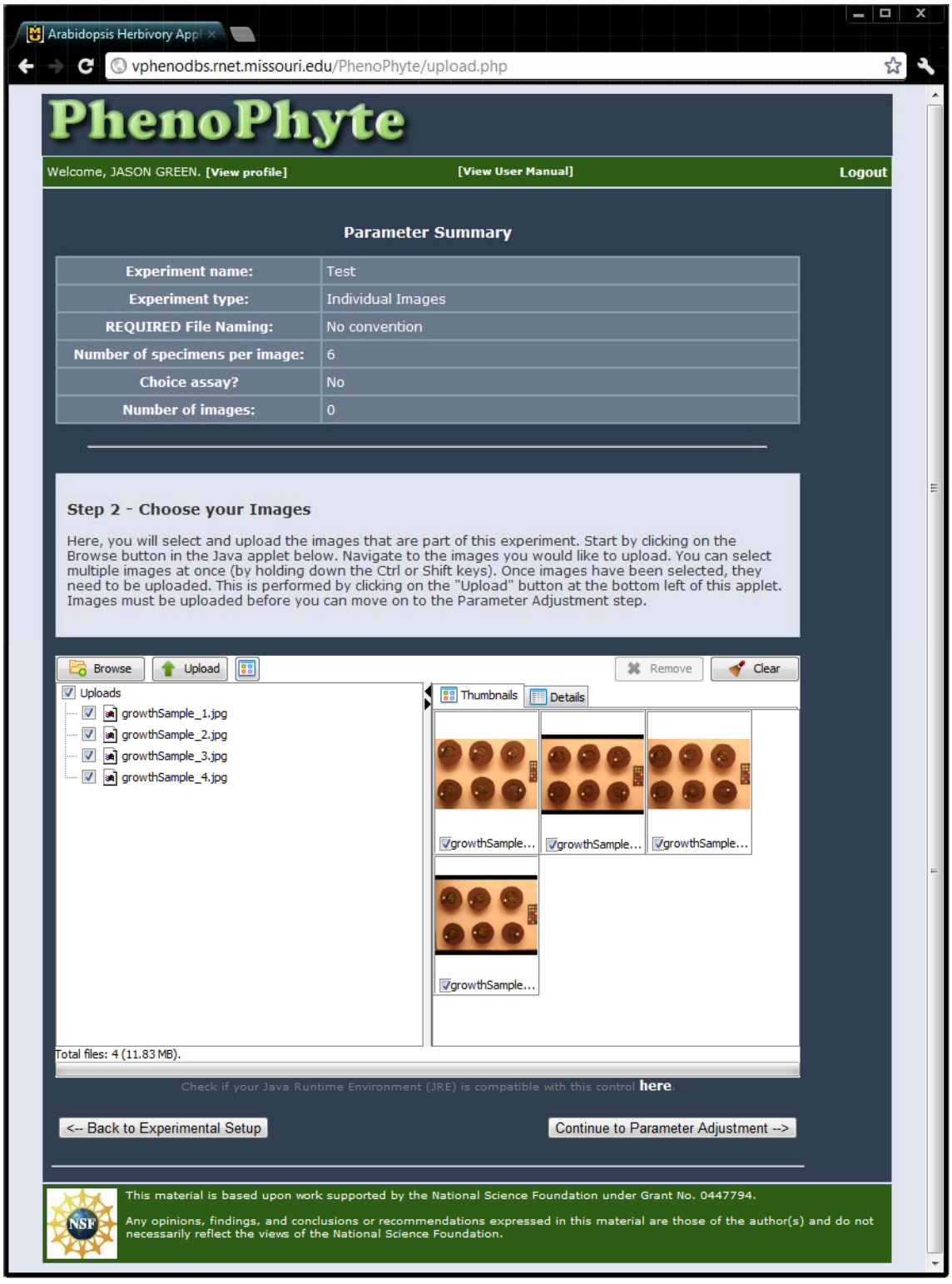


Figure 36: Screenshot of image upload page, in which multiple images can be uploaded simultaneously.

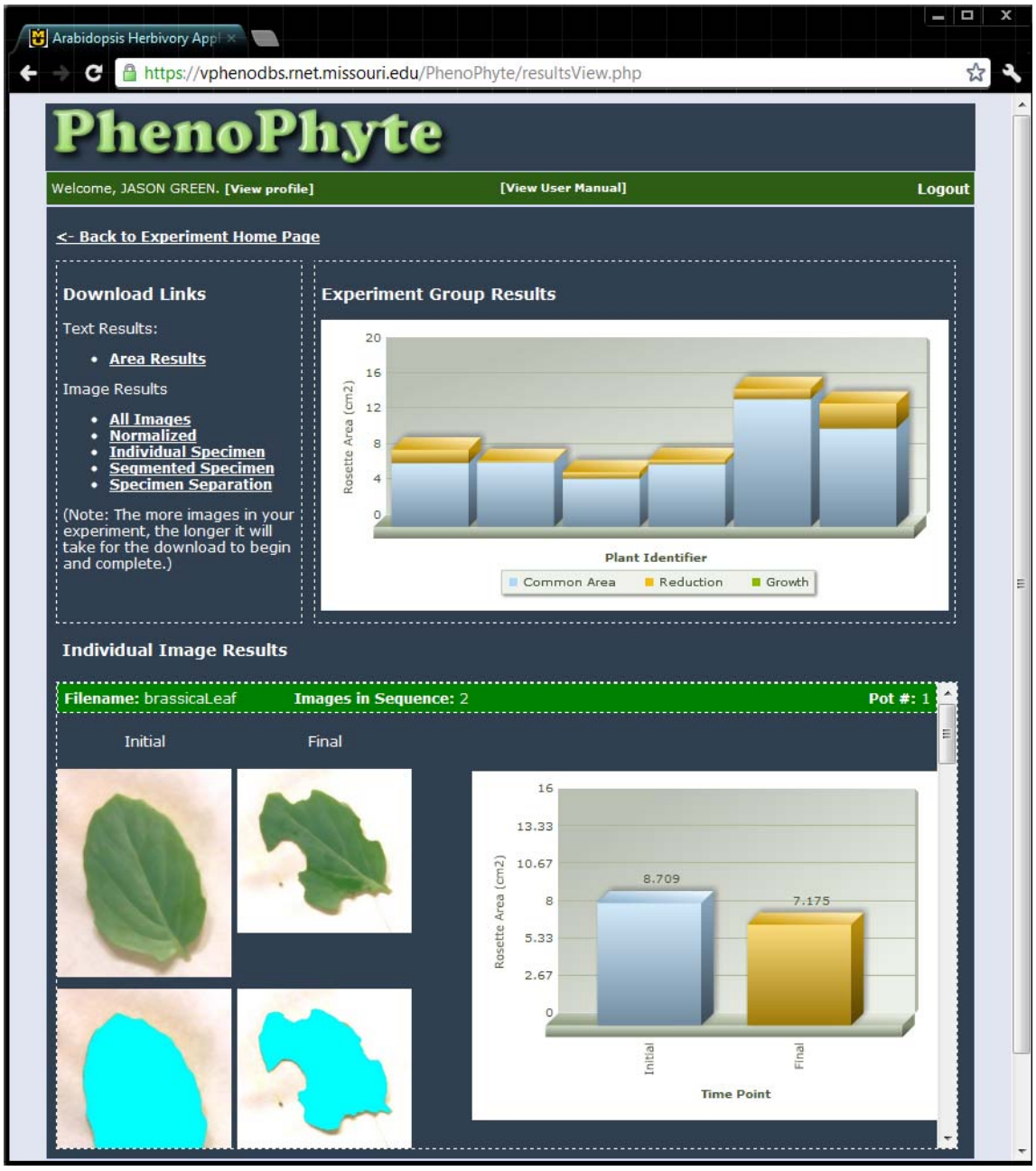


Figure 37: Screenshot of result summary page. Stacked bar charts are shown for all the plants in the experiment in the top right. Links to download both text and image results are found at the top left. The bottom portion shows image and graphical results of each individual plant in the experiment.

Using PhenoPhyte, users are able to upload large amounts of imagery to the server for batch processing (see Figure 36). Each upload can handle up to 2GB or 500 images, whichever comes first; however, if users have more images, they may repeat this process until the entire set is uploaded. Following processing, users are able to review the image results and make adjustments where necessary. Additionally, PhenoPhyte automatically generates graphs showing both experiment-wide and individual plant results (see Figure 37). This may include line graphs showing growth curves, stacked bar graphs illustrating herbivory, or simple bar charts for area traits. Finally, the web application allows users to download image results (normalized images and segmentation results) and measured trait values in CSV format, which can be viewed directly in Microsoft Excel. A detailed manual is available at the website to aid current and interested users in navigating the software.

5.2 Applications utilizing PhenoPhyte

PhenoPhyte was designed to facilitate measurement of plant area, growth, and herbivory in a number of settings and species, and here we illustrate several of them.

5.2.1 Measuring growth in Arabidopsis thaliana

By repeatedly and nondestructively imaging Arabidopsis rosettes over time, PhenoPhyte can provide accurate measurements of growth. To illustrate, 30 *Arabidopsis* plants were imaged (6 plants per image) once daily for 30 days. The amount of plant growth per day and the change in plant area over the one-month study are illustrated in Figure 38. In this experiment, the algorithm was able to detect plants as small as 0.004 cm^2. It was also able to detect natural variations in the growth curves of

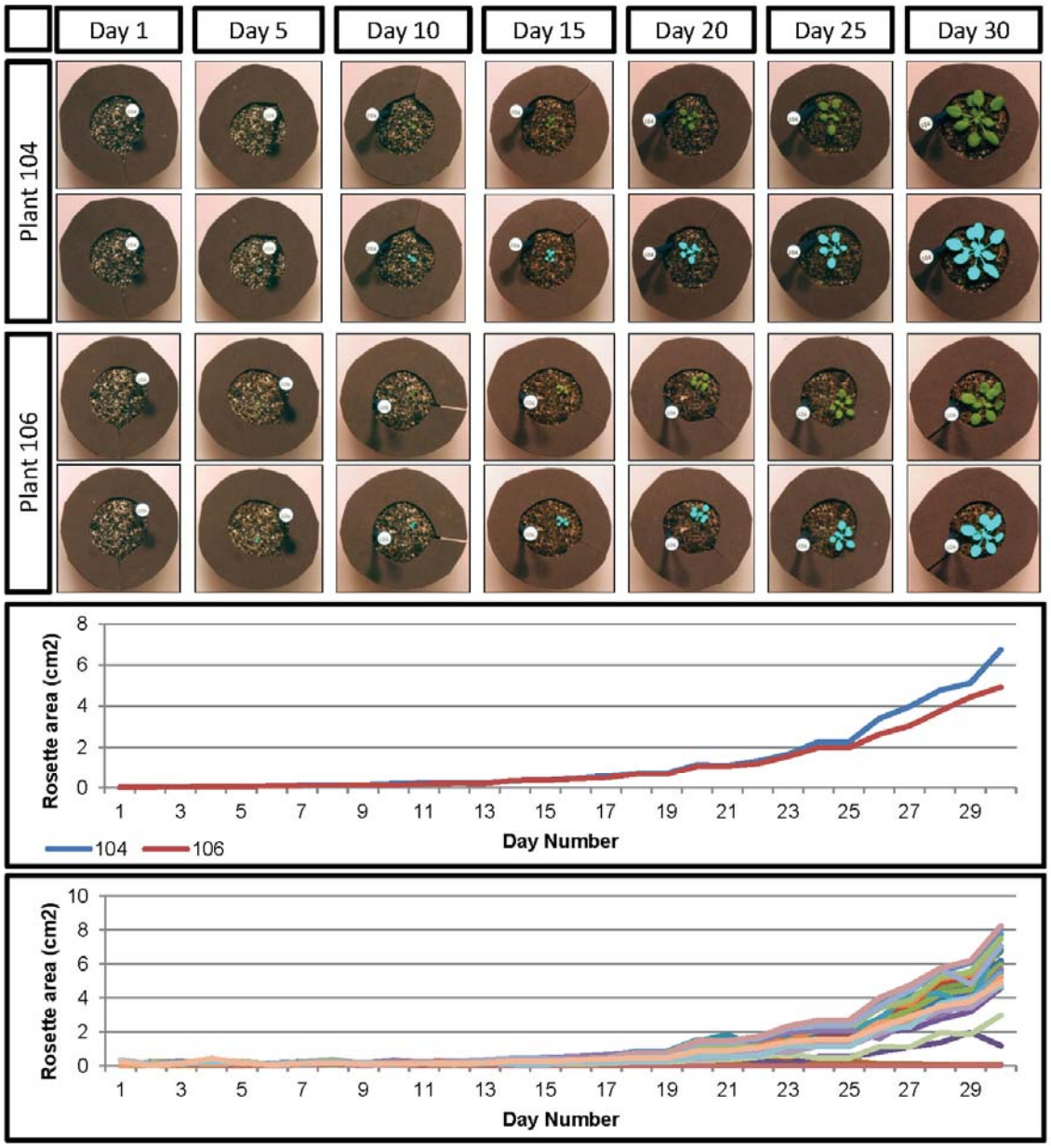


Figure 38: Example results from the growth experiment. The top panel includes normalized and processed images for two plants at five-day intervals. The middle panel shows the growth curves for those two example plants, showing the ability to distinguish between individual curves within genotypes. The bottom panel shows the growth curves for all 60 plants in the experiment.

individual plants of the same genotype. This sensitivity of growth analysis can be useful in identifying and studying more subtle factors (both genetic and environmental) that affect plant growth.

5.2.2 Measuring herbivory in Arabidopsis thaliana

By imaging plants before and after insect feeding, the software can provide accurate measurements of leaf herbivory. Plant traits known or hypothesized to influence insect feeding are often assessed by experiments in which insects are allowed to feed on different plant genotypes and the amount of plant matter consumed is used as an indicator of the presence or lack of chemical defense mechanisms.

To demonstrate, 30 Arabidopsis thaliana rosettes were exposed to cabbage butterfly caterpillars (Pieris rapae L. Pieridae) in cages and 30 plants served as insect-free controls. Images were taken before the introduction of the caterpillar and after feeding concluded 2 hours later.

The full results are shown in Figure 39. Original and processed images for three plants, before and after feeding, are shown in the top panel. In the middle panel, the total height of the bars in the graph represents the initial plant matter for each rosette as calculated by the software. The black fill indicates the final plant area, and the yellow fill shows the difference between the initial and final plant area, i.e., the measurement of herbivory. In this case, a simple difference of the plant area before and after insect feeding was used to calculate the leaf area removed by the caterpillar. All but three plants experienced damage from caterpillar feeding as measured by leaf area loss. In three plants (#3, #6, and #20), there were small increases in plant area (denoted by the red “growth” bar fill in the bottom panel), indicating either no caterpillar feeding

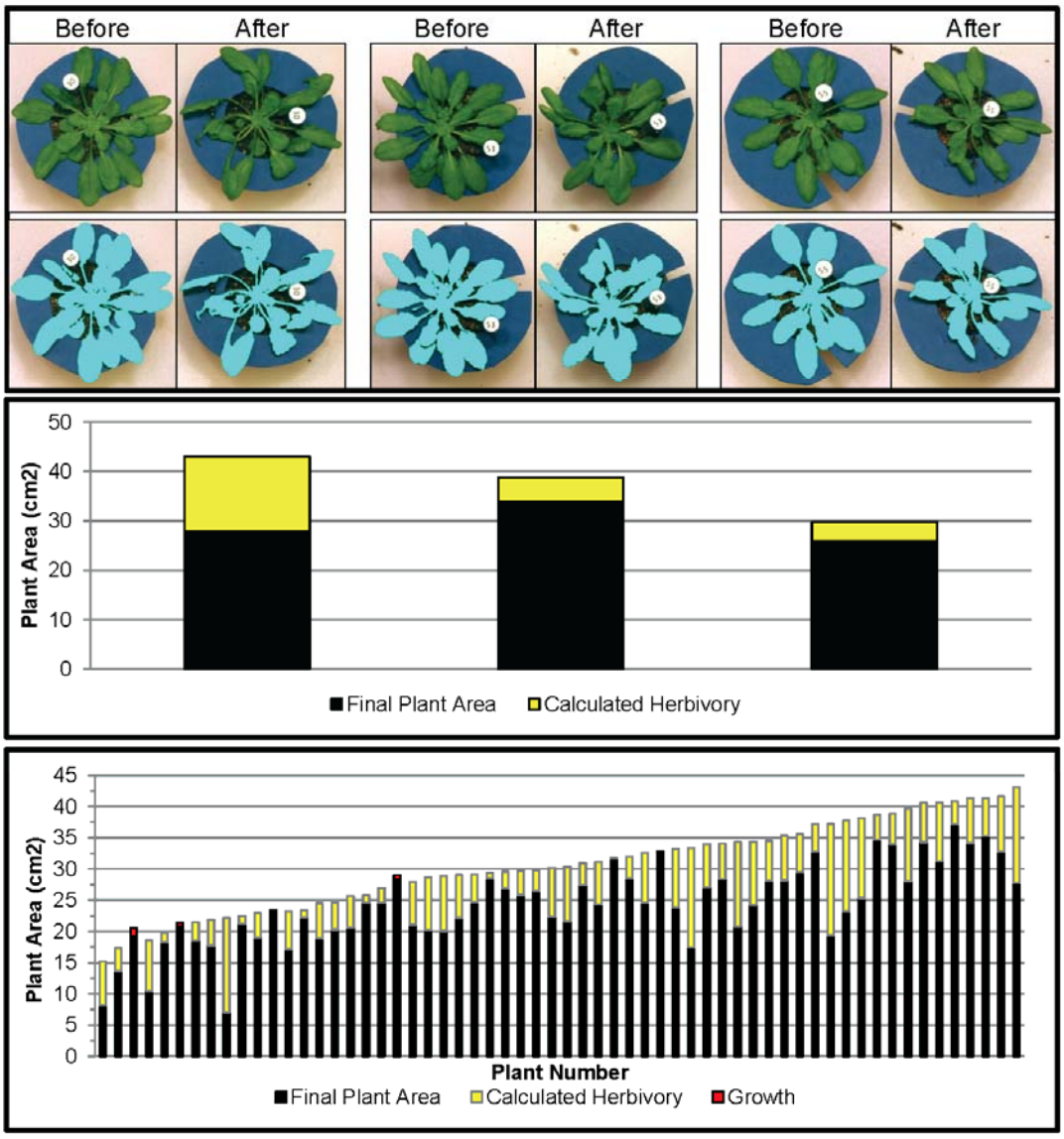


Figure 39: *Arabidopsis thaliana* herbivory results. The top panel shows before and after images for three plants from the experiment, as well as the processing results underneath. The middle panel illustrates the change in plant area between before and after images with the total height representing the initial area, the black fill representing final plant area, and the yellow fill indicating the amount of herbivory. The bottom panel shows these same results for the full experiment. Red fill indicates increases in plant area between, before and after images.

occurred or little enough feeding that it was compensated for by the small amount of plant growth which occurs in 2 hours.

5.2.3 Measuring herbivory in Brassica rapa

PhenoPhyte can also measure these traits and phenotypes in other species with similarly flat rosettes or by using detached leaves. In this example, cabbage leaves (Brassica rapa) were detached before the experiment began, and images were taken both before and after caterpillar feeding (2 hours).

The results are shown in Figure 40. The identification of plant matter is shown in the top panel for three leaves before and after caterpillar feeding (a-f). The middle panel again visually quantifies the difference in visible plant area between the before and after images (see Section 0 for a full description of the meaning of each color). All but two plants experienced damage from caterpillar feeding. Two plants (#7 and #16) had very slight increases in plant area, which resulted from the combination of no observable caterpillar feeding and differences in leaf curvature between images. In this experiment, since the plants were all of the same genotype and grown under identical conditions, variation in the amount of herbivory largely reflects the biological variation in feeding among individual caterpillars.

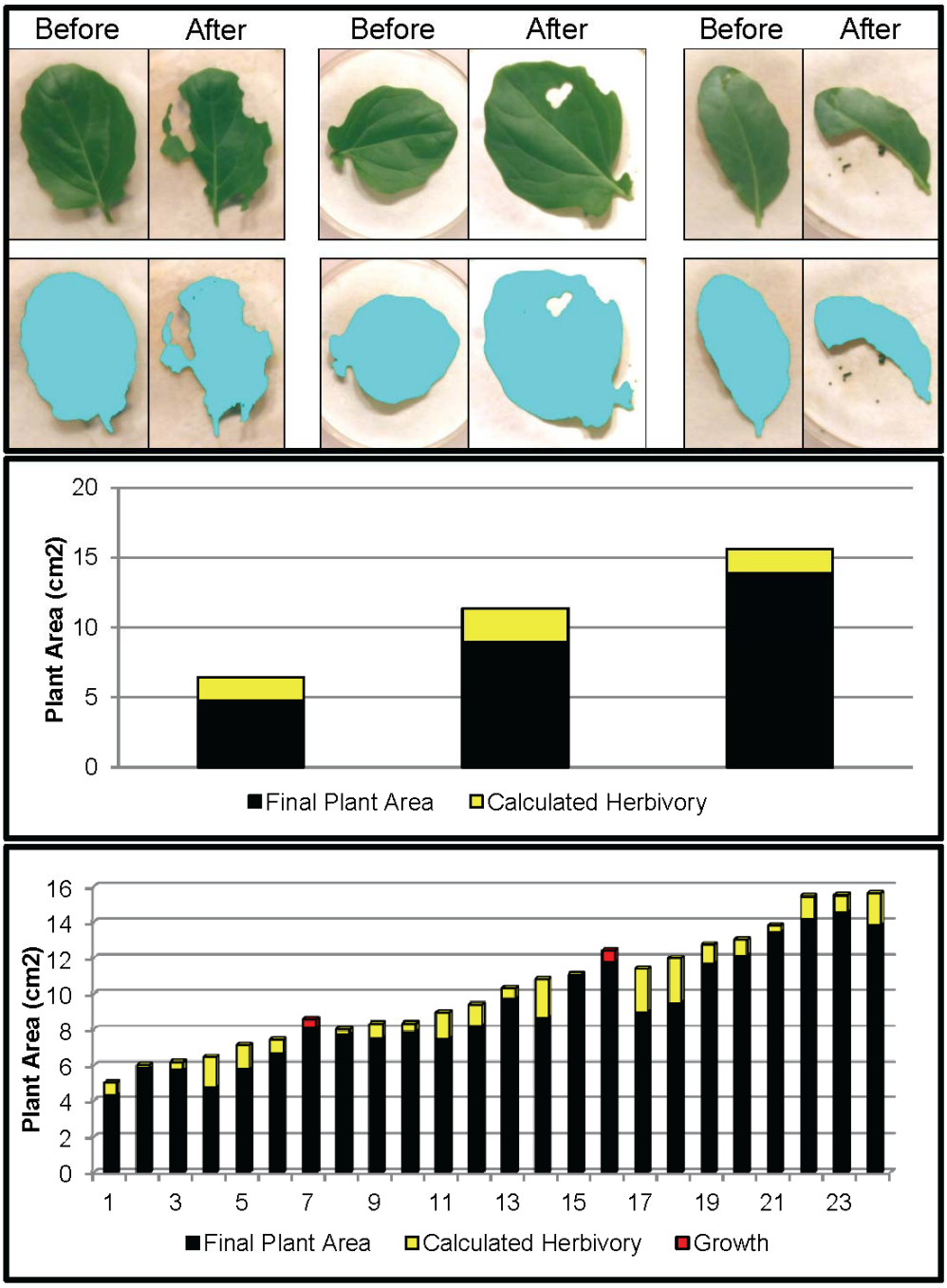


Figure 40: *Brassica rapa* herbivory results. The top panel shows before and after images for three leaves from the experiment, as well as the processing results underneath. The middle panel illustrates the change in leaf area between before and after images with the total height representing the initial area, the black fill representing final plant area, and the yellow fill indicating the amount of herbivory. The bottom panel shows these same results for the full experiment. Red fill indicates increases in left area between, before and after images.

5.2.4 Measuring herbivory in Glycine max

In this example, we demonstrate another way to utilize PhenoPhyte to measure herbivory. Here, soybean (Glycine max) leaves were removed after feeding by beet armyworm caterpillars (Spodoptera exigua Hübner (Noctuidae)) that were caged on individual leaflets of intact plants for 5 days. The leaves were imaged only at the end of the experiment, and the initial area of each leaf was estimated by recreating the original leaf margins in Photoshop. PhenoPhyte was then used to determine the amount of defoliation. Since this herbivory caused some discoloration around the margins of damage, the herbivory calculation included both tissue that was missing and tissue that was still present but was damaged and discolored by the caterpillar feeding (yellow or brown). Results are presented in the same format in Figure 41, with three example leaves from this experiment shown in the top panel, the area results for these sample leaves in the middle panel, and the full experimental results in the bottom panel. In this case, the mutants did not differ in the amount of leaf area removed by the caterpillar.

Although more time consuming than the detached leaf method described above with cabbage, simulating original leaf areas of attached leaves has the advantage that the leaves stay on the plant during caterpillar feeding and maintain their normal physiological condition and response to damage. This is especially important for feeding studies that last longer than a few hours. Simulation of original leaf areas is most accurate for leaves with entire margins like soybean, and its accuracy is likely to decline when the leaf margin is more complex. Images of soybean leaves could also be taken in situ on the plant at the beginning and end of the treatment by setting up an imaging station in the greenhouse or field.

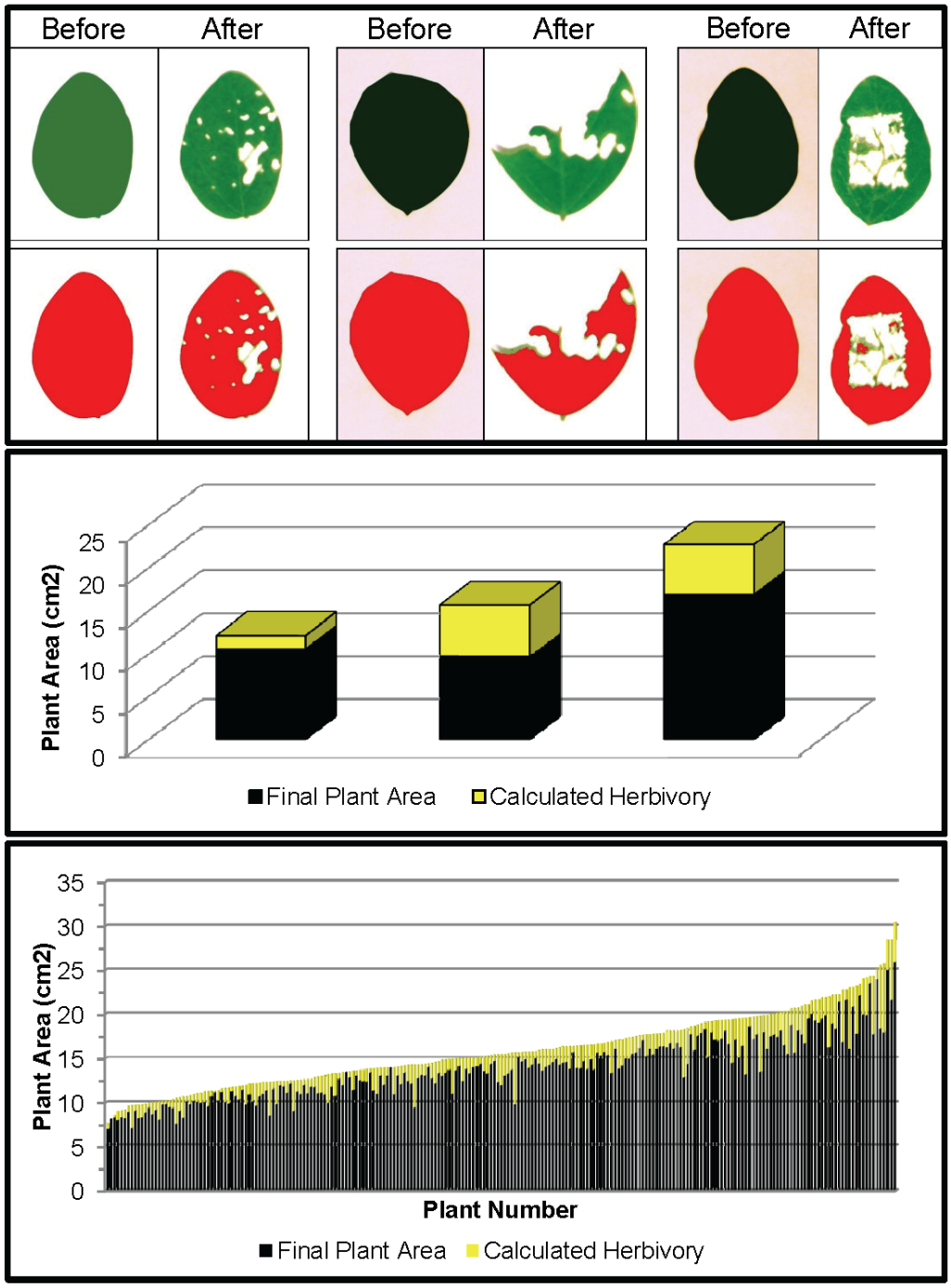


Figure 41: *Glycine max* herbivory results. The top panel shows before and after images for three leaves from the experiment, as well as the processing results underneath. The middle panel illustrates the change in leaf area between, before and after images with the total height representing the initial area, the black fill representing final plant area, and the yellow fill indicating the amount of herbivory. The bottom panel shows these same results for the full experiment.

5.2.5 Measuring herbivory with choice assays in Arabidopsis thaliana

Uniquely, PhenoPhyte also supports the analysis of herbivore choice assays, in which two different genotypes or treatments are present in one pot, separated by a marker. Choice assays can provide a better estimate of plant resistance to insects because they reflect insect feeding preferences not measured by no-choice assays.

To illustrate, we used 120 pots, each containing two small Arabidopsis rosettes: one wild-type and one mutant, and let first instar beet armyworm caterpillars feed on them for 15 hours. Figure 42 shows three examples from this experiment. A plot of the areas of the (left) wildtype and (right) mutant plant of each example is shown in the middle panel. The yellow fill indicates the amount of herbivory, whereas the red fill indicates growth between initial and final images. This same display is repeated for all the plants in the experiment in the bottom panel. In this case, the wild type and mutant genotypes did not differ in size and the caterpillars had a 1.8-fold higher preference for wild type plants ($p=0.0058$, Wilcoxon Two-Sample Test).

5.3 Summary

In this chapter, we introduced PhenoPhyte, a web-based application that encapsulates a high-throughput computational pipeline for processing imagery of leaves and rosettes to provide measurement of area, herbivory, and growth. Through the use of several case studies, the software was shown to be capable of processing a variety of experimental setups and a large number of images. Notably, the software is able to process choice assays and includes a means to link images together to automate the measurement of herbivory and growth. For interested users, the user's manual is available online at (http://PhenomicsWorld.org/PhenoPhyte).

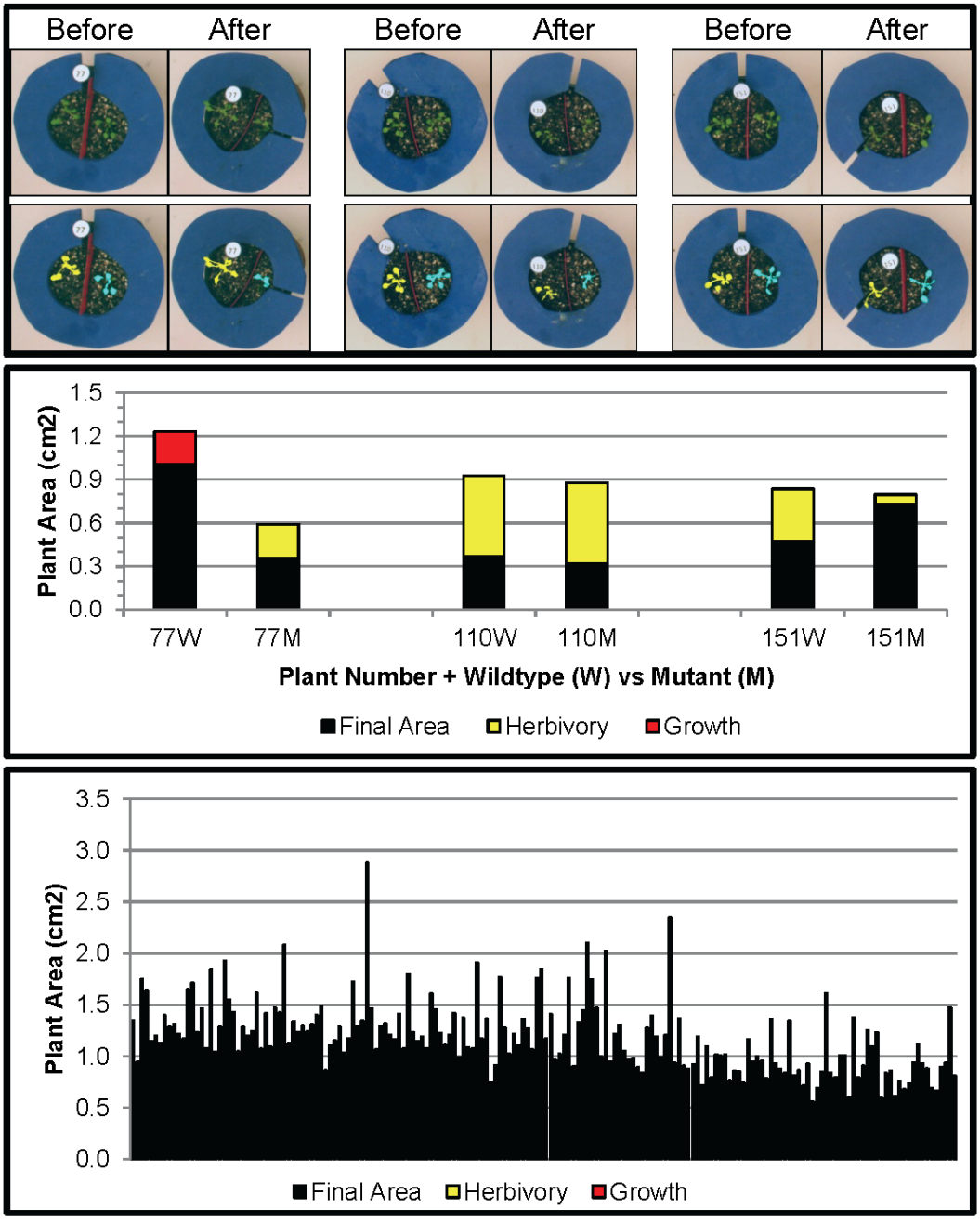


Figure 42: Choice assay herbivory results for *Arabidopsis thaliana*. The top panel shows before and after images for three pots from the experiment, as well as the processing results underneath. The middle panel demonstrates plant areas both before and after feeding for the (left) wildtype and (right) mutant plants in each pot. The bottom panel shows the same plant area comparisons for all pairs of plants in the experiment.

While the software currently only provides measurements related to area, it could be extended quite easily to provide additional types of relevant measurements, for example features related to shape and color. Adjustment of the backend would simply require the inclusion of algorithms to extract these additional features to the processing pipeline. Reflecting this change in the user interfaces would also be required and would be more time consuming.

One software feature that may be important for PhenoPhyte version 2.0 is the ability to define multiple criteria in order to simultaneously detect several regions of the leaf. This would be especially useful for phenotypic appearances with multiple colors, as the software would then be able to, for example, extract and quantify the amount of anthocyanin (purple pigments), necrosis, and chlorosis simultaneously.

CHAPTER SIX

VISUAL PHENOTYPE SEARCH

With the massive amounts of collected phenotype data and the varied and often complex information needs of researchers, search mechanisms will have to be developed to answer current questions posed by users as well as in anticipation of future retrieval needs. In this chapter, several advanced and unique search mechanisms that this research developed for phenotypes are discussed. The first is a text-based retrieval engine that ties various information sources together and utilizes domain ontologies to help users locate information. The second is a content-based retrieval mechanism that allows users to search with a phenotype image for visually similar images. The third is a search mechanism for finding objects with similar temporal characteristics. Search engines are constructed for both *Drosophila* mitochondria as well as necrotic lesions of maize lesion mimic mutants.

6.1 Literature Review

In this subsection, we will examine the current types of search mechanisms that are available to plant researchers as well as the disadvantages and limitations of these search styles. In addition, a review of content-based image retrieval, a more advanced retrieval approach, will be conducted.

6.1.1 Available Search Techniques in MODs

Major plant genome databases like MaizeGDB [71], Gramene [72], TAIR [73], SGN [74], Soybase [75], and Oryzabase [76] have compiled and organized large quantities of data for their respective research communities. Though enormous amounts of new data

are being generated and submitted, completion of the respective genomes is expected to increase the rate of data collection even more. Currently, each group provides text browsing capabilities to allow individuals to sift through the available data as well as basic text search mechanisms, mainly in the form of Structure Query Language (SQL) queries, to aid users in locating specific information. While these search mechanisms meet basic needs of plant science researchers, they will become decreasingly useful because of several shortcomings.

One limitation that will cause problems in the future is a failure to rank search results by similarity to the query, which is a by-product of the Boolean style retrieval [51] used by most current plant search mechanisms. Instead, search results are sorted idiosyncratically by some field, which is sufficient when databases and result sets are relatively small. However, as database sizes increase causing result sets to become larger, manual perusal through a list of arbitrarily ordered or alphabetized results to find meaningful information will become tedious, inefficient, and slow. To locate desired information more quickly, ranking of retrieved results according to their similarity to the query will be necessary.

A second issue is the limited utilization of free-text fields. A limited number of search mechanisms exist in the plant community that make use of free-text descriptions; most appear to rely on making selections from predefined lists of characteristics. Those mechanisms that do search free-text fields treat the description as a single string for character matching (description prefix, description suffix, description contains, etc.), which is not sufficient due to variations in phenotype descriptions. More advanced

information retrieval methods can be implemented to take better advantage of the wealth of information available in these fields.

A third shortcoming is the limited utilization of domain ontologies in current retrieval methods. Large amounts of time, money, and energy have been put forth towards the development of several plant-related ontologies [77-78]. Though the rich knowledge embedded in these ontological structures is particularly well suited to retrieval, especially for improving the efficiency and coverage of retrieved results, the domain ontologies remain underutilized In the information retrieval community, ontologies have been used extensively for three purposes: word sense disambiguation, which is the complex task of determining the correct meaning of a polysemous word from its context [79-81]; thematic summarization and concept mapping, which involves identifying broad concepts in free text [82-84]; and query expansion, which is a technique of enriching a query by adding additional relevant terms to it [85-87]. State-of-the-art query expansion via ontologies generally utilizes hierarchical relationships, typically parents and children [87], and this is one technique that can be employed to improve the accuracy and coverage of plant phenotype searches.

In addition, current search mechanisms exploit primarily the text-based information stored in the databases. While most of the information collected and stored in these databases is textual, there are other kinds of data for which searching could prove extremely beneficial. One of these types of data which has already received attention from a retrieval perspective is genomic data like sequences and proteins. Online database resources like PlantGDB [88] house various resources including genome annotation utilities, genome browsers, and sequence alignment and assembly tools. In

addition, with the increasing number of fully sequenced genomes, a number of genome browsers [72, 89-90] and sequence and protein retrieval engines [91-94] are publically available for researchers to utilize. What is still lacking in the plant community, and in many other domains, are search mechanisms that allow searching based on image data, which are being increasingly collected. (This is a different concept than search of image captions or metadata, which does occur in the plant community.) This type of search is called content-based retrieval and is reviewed in the next subsection.

6.1.2 Content-Based Image Retrieval

The concept of content-based image retrieval (CBIR) is the task of performing searches utilizing image content itself or features derived directly from image content. The typical steps to performing CBIR include identifying the important characteristics in a set of images, designing computer algorithms to directly or indirectly measure these features, and finally utilizing existing or developed indexing structures for fast and efficient retrieval of visually similar images. This technique first appeared in the mid 1990s with the development of systems like QBIC [95], PhotoBook [96], and VisualSEEK [97]. Over the past decade, the field has steadily grown and matured, and these techniques have since been applied to a wide variety of applications, including geospatial intelligence [98], astronomy [99], and protein structure comparison [92]. See [100-101] for extensive reviews of current CBIR systems. Most of the CBIR systems regarding phenotypes have been from the medical domain. The interested reader is referred to [102] for a review of CBIR applications in medicine. Regarding plant phenotypes, there appear to be very few studies. Yahiaoui and Boujemaa developed

IKONA, which contains one dataset in which CBIR techniques are used to compare genetically modified *Arabidopsis thaliana* plants [103].

6.2 Multi-Source Ontology-Based Phenotype Annotation Retrieval Engine

This research developed a flexible, advanced search mechanism for textual search of phenotype information that seeks to overcome the described shortcomings in most available text retrieval mechanisms in the plant community. This retrieval engine is designed to be multi-source, meaning it integrates multiple related free-text sources to aid the user in retrieving desired information. The current version of the search engine combines three text sources (phenotype image captions, locus descriptions, and gene product descriptions) that are associated with phenotypic variations in MaizeGDB. The retrieval engine is also designed to utilize the knowledge and structure of existing domain ontologies for query expansion, with the expectation of improving the contextualization of the query. Given the above text sources, the Plant Ontology (PO) [78] and Gene Ontology (GO) [77] were relevant and pertinent choices for inclusion in the search engine.

The MaizeGDB resource itself is in the early stages of a full redesign. One component of the redesign will be the inclusion of the system described here as one of MaizeGDB's phenotype search tools. This search mechanism is scheduled to be released as part of MaizeGDB in March 2013.

When a user submits a query to the system, a list of top-ranked phenotype variations is returned. Each entry in this list corresponds to a *folder* of documents, which contains all the phenotype image captions, locus descriptions, and gene product descriptions associated with this variation. From among this group of documents, a single

representative document is selected (using Equation (33)) from each text source for use in determining the rank of the folder, or the similarity of the folder to the query.

$$g\big(f(j,k)\big) = \Big\{ d_i^k | i = max_i \big\{ sim(d_i^k, q) \big\} \Big\} \tag{33}$$

where $f(j,k)$ is the set of documents from text source k related to phenotype variation j and d_i^k is the document in text source k with the highest similarity to the query q, where document similarity is calculated using the standard cosine similarity measures, shown in Equation (34).

$$sim_{document}\big(d_j^k, q\big) = \frac{\sum_{i=1}^{t} w_{j,i} * w_{q,i}}{\sqrt{\big(\sum_{i=1}^{t} w_{j,i}^2\big)} * \sqrt{\big(\sum_{i=1}^{t} w_{q,i}^2\big)}} \tag{34}$$

where $w_{a,b}$ is the standard term frequency (TF) and inverse document frequency (IDF) weightings for term b in document a (or query q).

A variation of the standard vector model [104] is applied to determine the ranking. In short, the folder vector is formed by a concatenation of vectors of the representative document from each text source. The folder similarity measure (shown in Equation (35)) used is the traditional cosine similarity measure adapted for multiple documents. This measure is utilized to determine folder rankings.

$$sim_{folder}(F(j,K), q) = \frac{\sum_{k \in K} \big(\sum_{i=1}^{t} \big(w_{g(f(j,k)),i}^k * w_{q,i}^k \big) \big)}{\sqrt{\sum_{k \in K} \Big(\sum_{i=1}^{t} {w_{g(f(j,k)),i}^k}^2 \Big)} * \sqrt{\sum_{k \in K} \Big(\sum_{i=1}^{t} {w_{q,i}^k}^2 \Big)}} \tag{35}$$

where $F(j, K)$ is the set of all documents from all text sources K related to phenotypic variation j.

In performing the search, query expansion is performed by matching query terms exactly to ontology terms. Then, any term synonyms, parents, or children may be included in the query and weighted appropriately, through user controls. The search and results pages are shown in Figure 43 and

Figure 44, respectively. The search interface is repeated to allow adjustment to the query or weights. The main results page shows phenotypic variations ordered by relevance. Within each result is the most similar document from each text source, along with an accompanying phenotype image and links to the MaizeGDB pages of the most similar genetic loci and gene products.

6.2.1 Evaluation

The developed multi-source ontology-based maize mutant phenotype search engine utilizes interconnected free-text fields from MaizeGDB, specifically descriptions of phenotypes, loci, and gene products, as well as two domain ontologies, the Gene Ontology and Plant Ontology. Users are given the ability to search with any combination of text sources and ontologies and can adjust the weights of these entities as desired to customize retrieval of specific queries. This is the first retrieval tool in the plant community that utilizes sophisticated information retrieval techniques to search free-text fields and provide ranking of results according to similarity to the query and that utilizes domain ontologies in this manner for query expansion.

The search engine was evaluated in terms of retrieval performance, specifically speed and accuracy, and scalability. The setup and results of the corresponding experiments are described below. All these experiments were conducted on a standalone development server with Intel X5570 dual 2.93 GHz quad core CPU and 72GB of RAM.

Figure 43: Query interface for the multi-source ontology-based retrieval engine for maize mutant phenotypes.

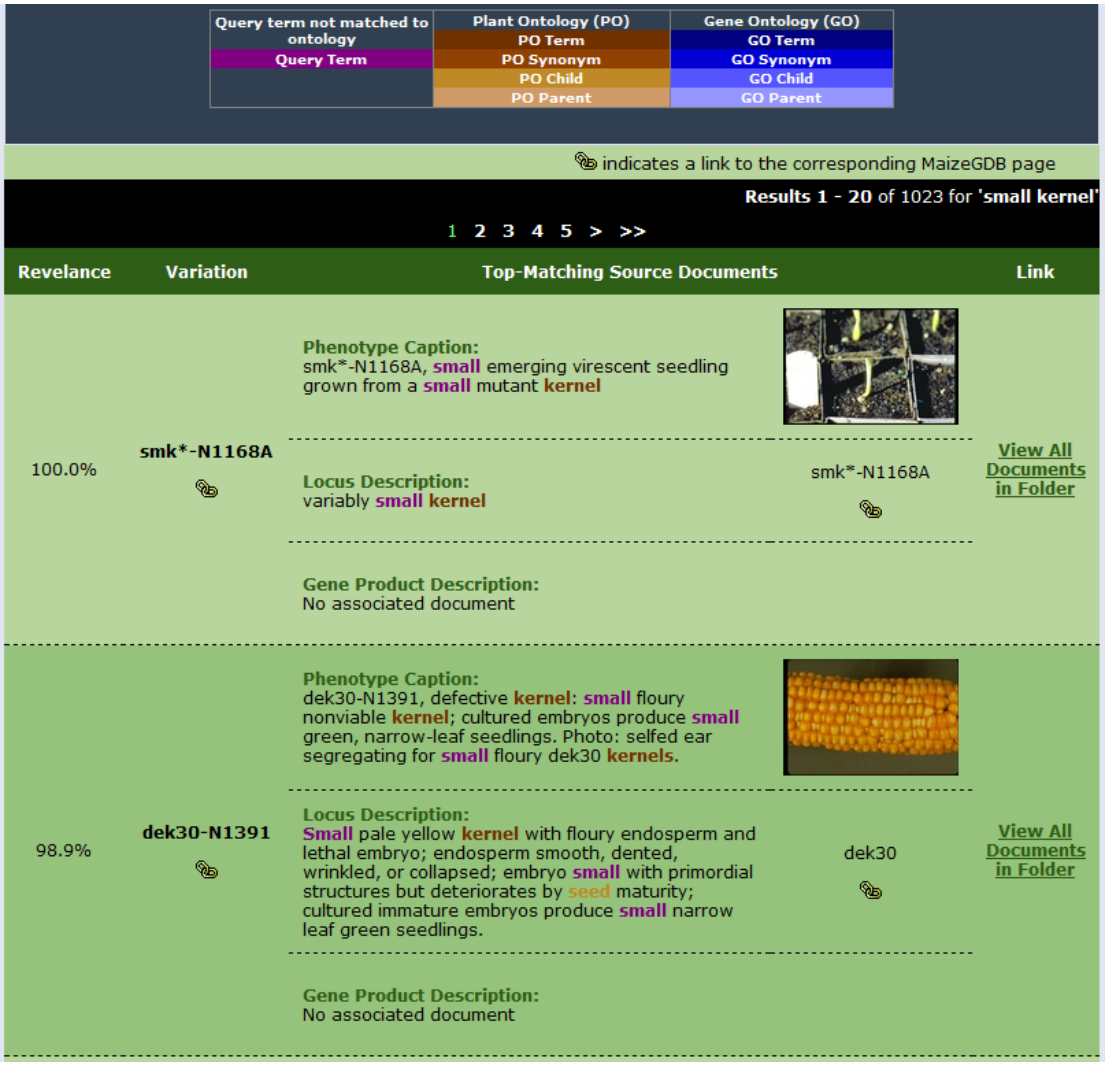


Figure 44: Sample search results for our retrieval engine using the query “small kernels.” Highlighted terms indicate matches to the query (including original terms and terms included through query expansion), with the color providing information about the term.

Retrieval Performance

Experiments were conducted to measure both the speed and accuracy of the developed search engine. For all experiments, the list of 29 test queries in Table 10 was used. These test queries included a set of anatomical terms, most of which were linked to phenotypic variations in MaizeGDB, anatomical terms plus one or more modifiers, and other miscellaneous queries.

To determine the overall speed of the retrieval system, each query was executed 20 times against one, two, and all three text sources. The overall average retrieval time for all the test queries in all situations was measured at 2.16 seconds/query. The fastest query was “lysine”, which completed in 0.28 seconds but only had 7 matching documents in the database. The slowest query was “leaf blade with white stripes”, which without query expansion partially matched 452 phenotypic variations and finished in 5.67 seconds.

The quality and accuracy of the search mechanism were also measured. We designed an experiment that utilized eight body part terms (“kernel”, “ear”, “leaf”, “tassel”, “aleurone”, “pericarp”, “endosperm”, “seedling”) to query the system, with the expectation of retrieving variations whose affected body part matches the query. The queries were executed in four environments: with one source (phenotype captions) or all three sources and with or without query expansion. In each situation, precision was measured at 10% recall intervals and was also measured for the top 10, 20, 30, 40, and 50 results. The affected body parts, which have been manually curated by humans, in MaizeGDB for each variation were used as the standard to assess the relevance of each ranked result.

Table 10: List of experimental queries

ID	Query
1	Lysine
2	Gibberellins
3	Aleurone
4	Pericarp
5	mottled appearance
6	Tassel
7	purple aleurone
8	Endosperm
9	tassel branch
10	andromonoecious plant
11	Leaf
12	leaf blade
13	necrotic tissue
14	Seedling
15	narrow leaf
16	Kernel
17	yellow leaf
18	Ear
19	broad leaf
20	segregating ears
21	collapsed kernels
22	floury kernel
23	immature ear
24	small kernel
25	broad green leaf
26	opaque dented kernel
27	small floury kernel
28	small yellow kernel
29	leaf blade with white stripes

The standard equations for precision and recall [51] were used and are shown below.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{36}$$

and

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{37}$$

The precision values at the top 10 through 50 results for the four scenarios are shown in Figure 45. It is noteworthy to mention that the scenario with the best performance is the one that uses all three text sources but does not perform query expansion. This result helps to establish the usefulness of one of the major components of this search engine – the integration of multiple text sources to improve search – is demonstrated in this situation by the increased precision when all the text sources are utilized. The scenario with the second best performance occurs when the phenotype caption source is used alone in conjunction with query expansion, which gives credence to the utilization of ontologies, particularly with respect to query expansion, to help improve search results.

These ideas are reinforced further by looking at the some of the precision-recall plots for the individual queries. Consider Figure 46, which shows the results for the "endosperm" and "ear" queries. While all the scenarios for "endosperm" perform comparably at the early recall levels, it is the scenario with all sources and with query expansion that maintains the highest precision for the higher recall levels. In addition, by utilizing these features, the result set is able to pick up more than 90% of all the variations whose affected body part is "endosperm" versus roughly 45% for the single

source search without expansion. The “ear” query is a quite clear example of how extra text sources and query expansion can improve the search results, as the top 10 results all retrieve correct variations, and the precision stays above all the other scenarios for all but one recall level.

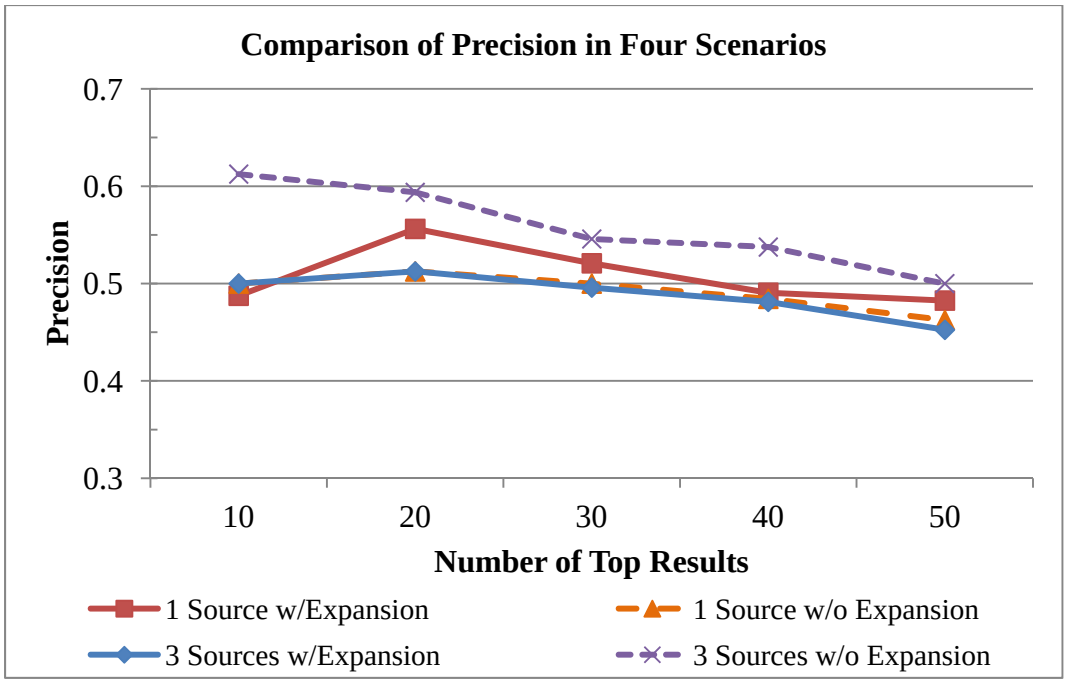


Figure 45: Precision was measured at the top 10, 20, 30, 40, and 50 results for four scenarios: one text source with query expansion, one text source without query expansion, all three text sources with query expansion, and all three text sources without query expansion. The precision values at each of those levels for each scenario are shown.

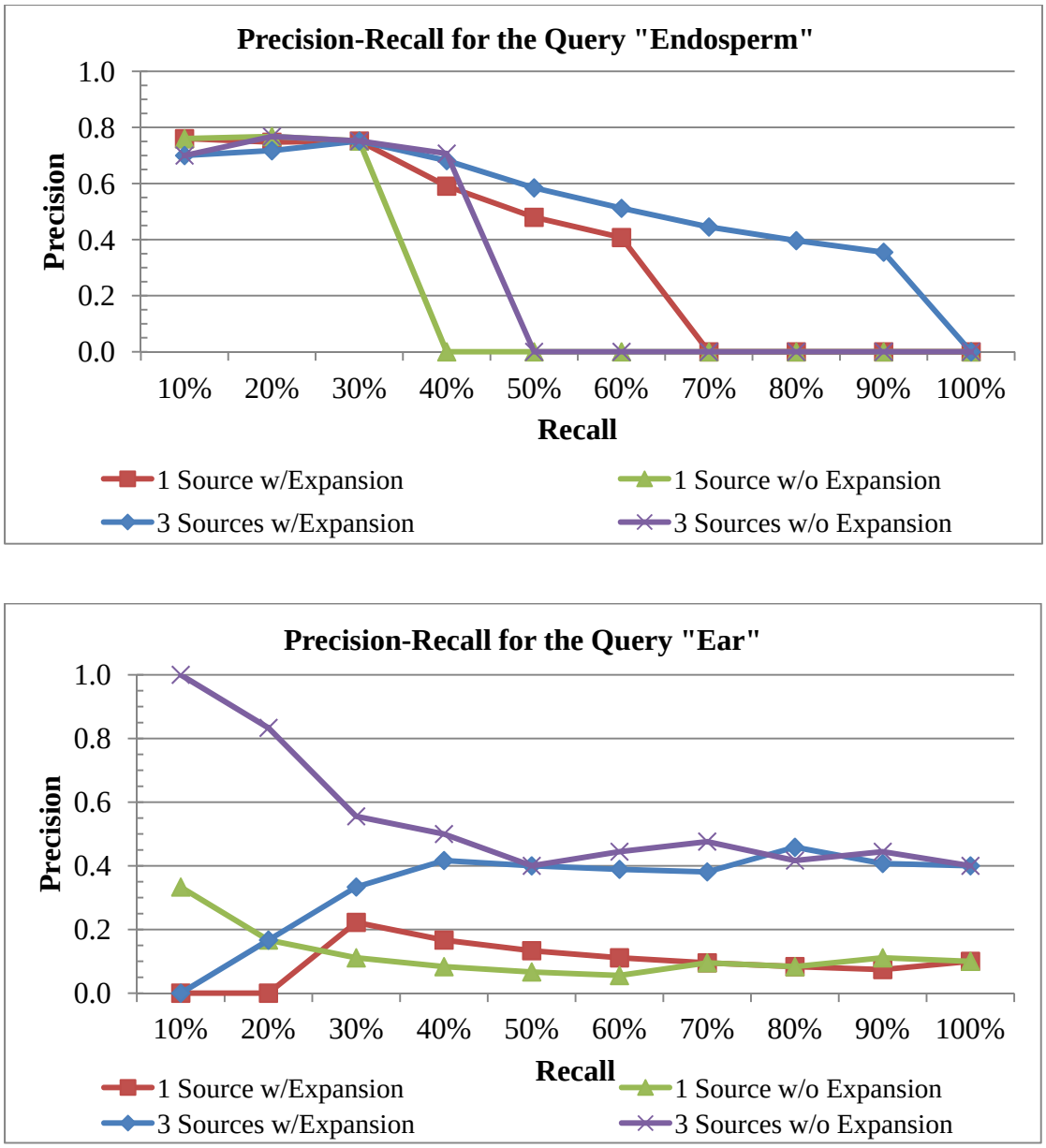


Figure 46: Precision-recall plots for two individual queries: “endosperm” and “ear.” For the “endosperm” query (top), all four scenarios have similar precision values at low recall levels; however, the scenario with all the text sources and query expansion includes many more of the relevant documents in the result sets than the others, as indicated by the higher precision at the higher recall levels. For the “ear” query (bottom), the best performance is achieved by all the text sources with no query expansion.

System Scalability

The scalability of the system was assessed using three separate experiments. First, we designed an experiment to examine the effect of query length on retrieval time. Retrieval speed was measured for each of the test queries, which were composed of one, two, or three terms. The average query speeds were then used to illustrate the trend in retrieval time as the length of the query increases. Figure 47 shows the average query time for each query using only the caption text source. The figure is organized by query length with the single-word queries on the left, the two-word queries in the middle, and the three-term queries on the right. While the average times did increase as query length increased (0.43 seconds between query lengths of one and two terms, and 1.55 seconds between query lengths of two and three words), there were overlaps in the retrieval times of individual queries across the sets. It was noted that the more dominant factor in retrieval time was not the number of query terms, but rather the number of descriptions matched by the query terms.

Second, in order to determine the effect of the number of text sources on query performance, each of the test queries was executed using just one text source (phenotype captions), two sources (captions and locus descriptions), and all three text sources. For each scenario, each query was executed 20 times with the average retrieval speeds measured and shown in Figure 48. This figure shows slight increases in retrieval time with the inclusion of additional text sources, but not a significant increase. The average increase in retrieval time for all the test queries in adding the second and third text sources was found to be 0.49 and 0.12 seconds, respectively.

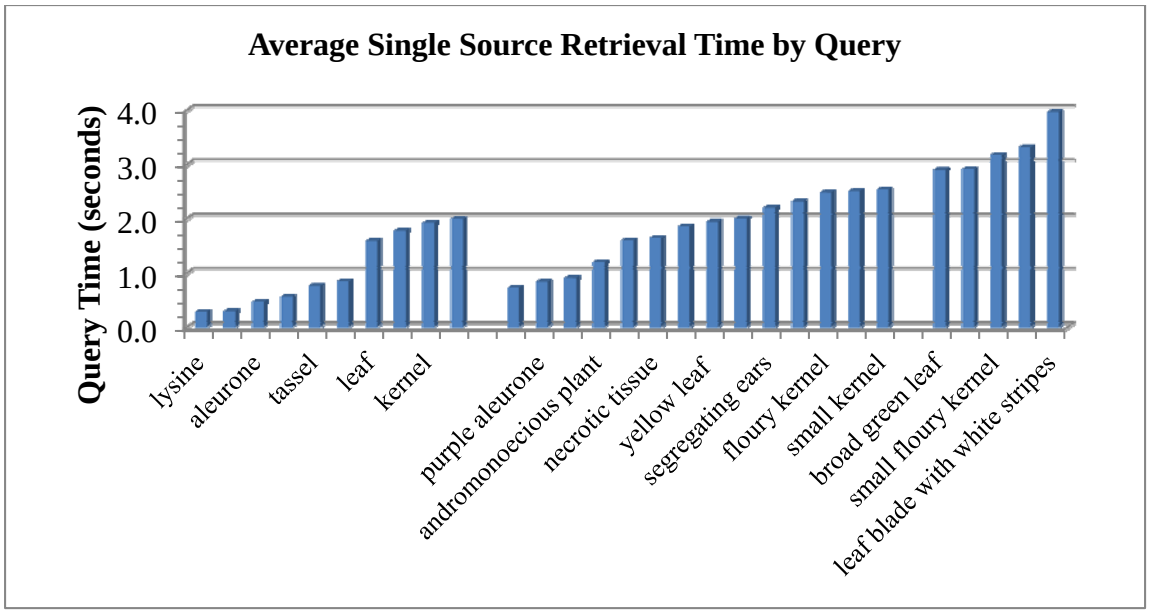


Figure 47: Average retrieval speeds for queries, ordered by query length. The leftmost group corresponds to the single term queries, and these have the quickest execution times. The middle group contains queries consisting of two terms, and the query speeds for these are on average slightly slower than the single term queries. The rightmost group of queries all contain three terms, and these are the most time consuming of the queries tested.

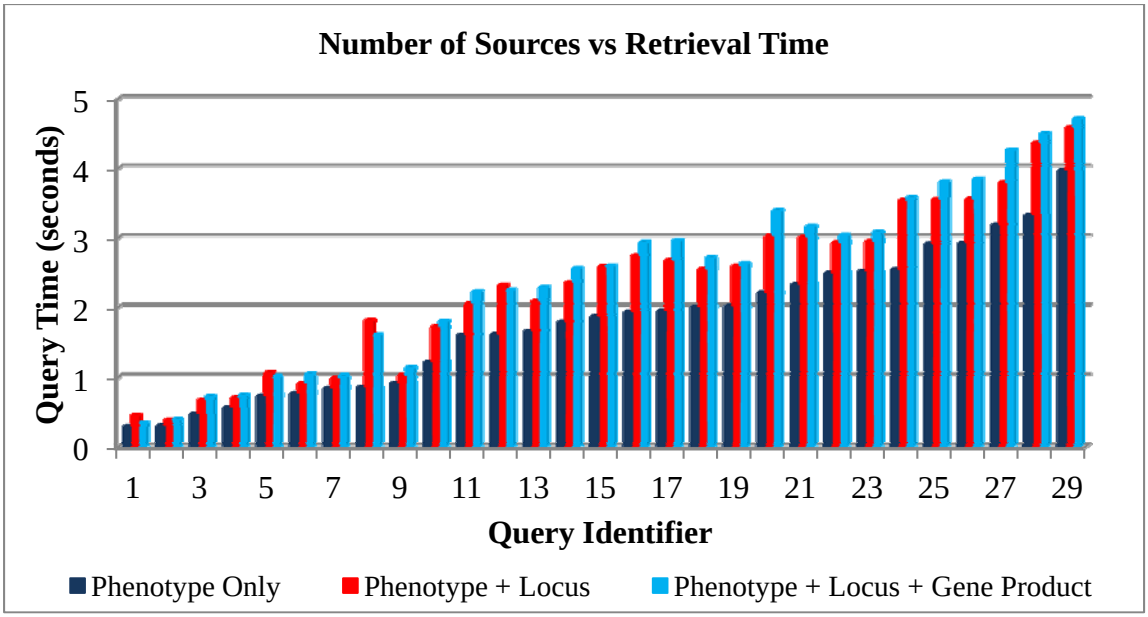


Figure 48: Comparison of retrieval speeds for each of the test queries executed on one text source (phenotype captions only) [left bars], two sources (phenotype caption + locus descriptions) [middle bars], and three sources [right bars]. There is a slight increase in execution as the number of sources increase.

Finally, we designed an experiment to examine how the size of the database, in terms of the number of documents, affects retrieval time. To perform this experiment, smaller databases were generated from the current dataset by decreasing the size of the phenotype caption table. Six versions of each phenotype caption table size, which included 20%, 25%, 33%, 50%, and 75% of the original table size, were constructed. Each query was executed five times on each of the constructed databases. Figure 49 shows the trend in query performance as the number of documents in the database increased. The five-fold increase in database size depicted shows an increase in retrieval time from 1.19 seconds to only 2.42 seconds. Should the size of the database eventually cause retrieval times to become unacceptable, various strategies could be employed to

speed up the search. Performing the search on each text source in parallel and then merging the results of each of those threads could make significant inroads in decreasing retrieval speed. In addition, because the search engine is currently implemented via a PHP application that communicates with a MySQL database, implementing portions of the search procedure in a higher performance language, like C++, could also allow for substantial gains in query speeds.

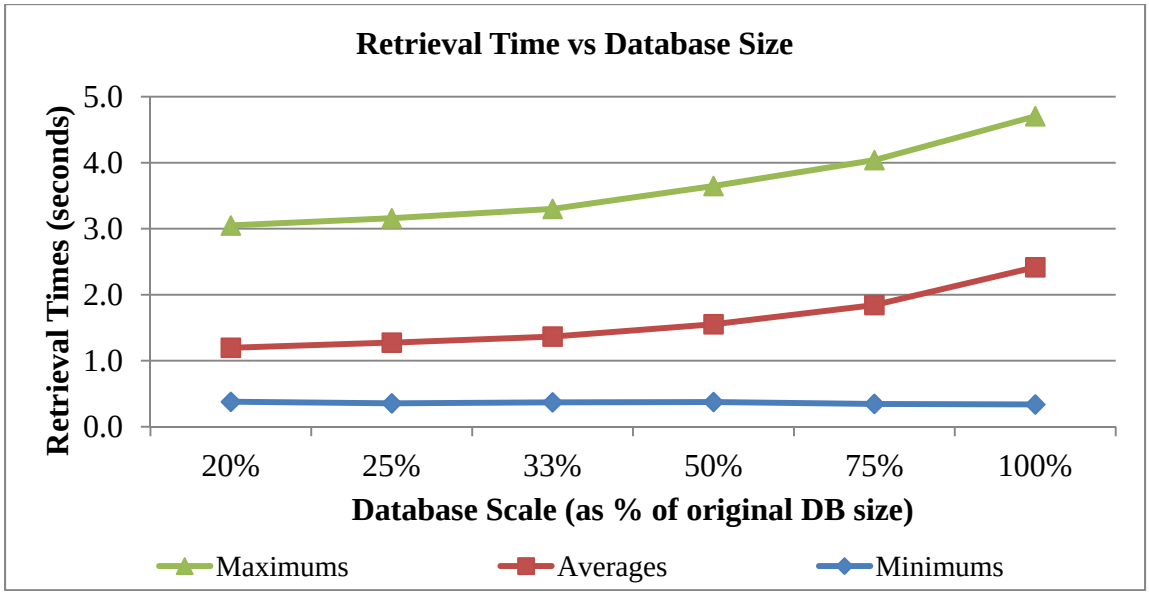


Figure 49: Effect of database size on retrieval speeds. Six different-sized test databases were constructed from subsets of the original dataset. All the test queries were executed against each of these test databases, with the minimum, average, and maximum query times measured. The trend suggests a nonlinear complexity for the search task

Limitations

As with any system, there are some limitations to this search mechanism. The first limitation is the requirement for exact matches when pairing terms to ontology concepts. This, at times, prevents the matching of concepts in documents to ontologies and thus prevents the ability to perform query expansion on those terms. For example, the term 'aleurone' on its own is not in the PO; however, 'aleurone layer' is a concept. Due to the large number of concepts that can be returned by performing a partial match to an ontology concept, it was decided that missing an occasional ontology match was a better alternative than falsely identifying many ontology pairings. Though stemming is performed on words in the text documents, ontologies, and queries in an effort to reduce each term to its base word, variations in terms do exist that stemming does not account for. The result of this is the inability to match some variations of words (e.g. because "necrosis" and "necrotic" map to "necrosi" and "necrot", respectively, these terms would not match each other in the system). A third limitation of this approach is the potential steep learning curves for users. Because of the built in flexibility, it may take some time for users to get accustomed to the available weighting options and it may take some exploration of the capabilities of the system to learn how to best utilize it.

6.3 Lesion Mimic Mutant Image Search

In addition to utilizing features to determine phenotype scores or measurements for direct use in experimental analysis, extracted features from imagery can also be exploited to provide image searches using content-based image retrieval (CBIR). To this end, as part of the Visual Phenotype Database System (VPhenoDBS) project, a

CBIR search engine was developed for maize leaf images and can be publically accessed at http://phenomicsworld.org/QBE.php.

This retrieval mechanism utilizes images from maize lesion mimic mutants, a class of corn mutants that exhibit lesions (varying by characteristics such as color, size, shape, and distribution) in the absence of infection, stress, or injury [57]. In particular, 726 leaf images from 16 different *les* mutant classes (see Table 11 for a listing of the specific *les* mutants) are represented by the search engine. For each of the mutant images, a subset of the extracted features discussed in Section 3.2.2 is utilized for performing searches; specifically, features related to leaf color, leaf texture, lesion color, lesion size, lesion shape, and lesion distribution are indexed. With each of those six classes of features, an EBS k-d tree [55] is constructed. This tree is a high-dimensional indexing tree that utilizes image labels and feature values to partition feature space in such a way as to provide a highly efficient and accurate search.

Table 11: *Les* mutants contained in the database underlying the VPhenoDBS Query by Image Example retrieval engine.

Searchable Les Mutants			
les1	les6	les10	les18
les2	les7	les12	les19
les3	les8	les13	les21
les5	les9	les17	les23

Figure 50: Query page for the Query by Phenotype Image search mechanism of the VPhenoDBS project.

Figure 51: Results page for the Query by Phenotype Image feature of the VPhenoDBS project.

When an image is submitted to the system for searching (see Figure 50), each of the six indexes is queried, and the top results from each are retrieved. This represents a deviation from the traditional CBIR search mechanisms, as the use of multiple indexes requires additional and nontrivial processing of the individual indexes' result sets. Using the method developed by Klaric *et al*., the results from each index are merged together to form a final ranked listing of results that are returned to the user.

The results page is illustrated in Figure 51. This page divides the screen in three panes: top, middle, and bottom. The bottom pane contains the ranked results, ordered from left to right. The middle pane contains information about the currently selected image from the top results. This data includes the image, any textual information stored about that image, as well as predicted semantic terms related to the image. (These semantic terms are computed based on the method by Barb *et al*. [105] which utilizes association rule mining in conjunction with parameterized functions to model semantic relevance in relationship to specific features and values. It is noteworthy to stress that the semantic terms are retrieved based on image content, not textual labeling of the image.) The top pane of the results page contains both the query image (left) as well as a figure showing the distribution of different *les* mutants in the top ranked results.

The current retrieval engine functions in demonstration mode only, meaning that users may only select available images from our collection to use for search. Extension of the search mechanism to allow users to upload new images and perform searches on-the-fly is planned.

6.3.1 Evaluation

The quality of this search engine is directly related to the quality of the features used to represent the images. Evaluation of the extracted features was previously discussed in Section 3.4.3.1 with an experiment that determined the ability of the extracted features to differentiate between several *les* mutants. The accuracy of this classification task was 90.4%. The reader is referred to Section 3.4.3.1 for full details of the experiment.

6.4 Temporal Behavior Search

In Chapter 4, a method was described for modeling temporal behavior using Markov chains. This involved extracting features over a time series relevant to the temporal behavior being studied, defining a set of states, translating the extracted features into a sequence of states, and finally using that information to build a Markov chain, in which the transition between two states represents the probability that the object would move between these states.

In this subsection, we describe how these Markov models can be utilized to build a retrieval engine that finds objects whose temporal characteristics are similar to that of a query object. As in Chapter 4, let $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ be a graph corresponding to a Markov chain where V represents the set of states and E the transition probabilities between states. We first represent each Markov chain as a vector of features. The most natural manner to construct this vector is for each position in the vector to correspond to the transition probability between a defined set of points. For example, where $\vec{M} = \{v_{i*|V|+j} = e_{i,j}\}$. Once these vectors are constructed from Markov chains, a similarity measure needs to be defined. Though several similarity metrics have been proposed, including the family of K-norm distances and the Kullback-Leibler Divergence, the

cosine similarity (see Equation (38)) was chosen due to its extensive usage in information retrieval systems.

$$cosSimilarity\left(\overrightarrow{M_1}, \overrightarrow{M_2}\right) = \cos^{-1}\left(\frac{\overrightarrow{M_1} \cdot \overrightarrow{M_2}}{\sqrt{\left\|\overrightarrow{M_1}\right\| * \left\|\overrightarrow{M_2}\right\|}}\right) \tag{38}$$

With these pieces in place, a retrieval system to search for objects with similar Markov chains or temporal behavior can be constructed.

Using the example provided in Section 4.6.3 to define states related to growth and the criteria for an object belonging to each state, a small-scale search mechanism to retrieve necrotic lesions based on growth was constructed. In total, 63 necrotic lesions were tracked over a 5-frame image sequence. As described in Section 4.6.3, their area measurements were converted first to growth percentages, then to state sequences, and finally to Markov chains.

When selecting a query lesion, the temporal behavior is displayed to the user as an animated GIF file (see Figure 52). After a lesion has been selected, the system retrieves the lesions whose Markov chains best match the Markov chain of the query lesion using the procedure described. The ranked results are then displayed (see Figure 53) to the user. Both animated files showing lesion progression and the corresponding Markov chains are provided for each result.

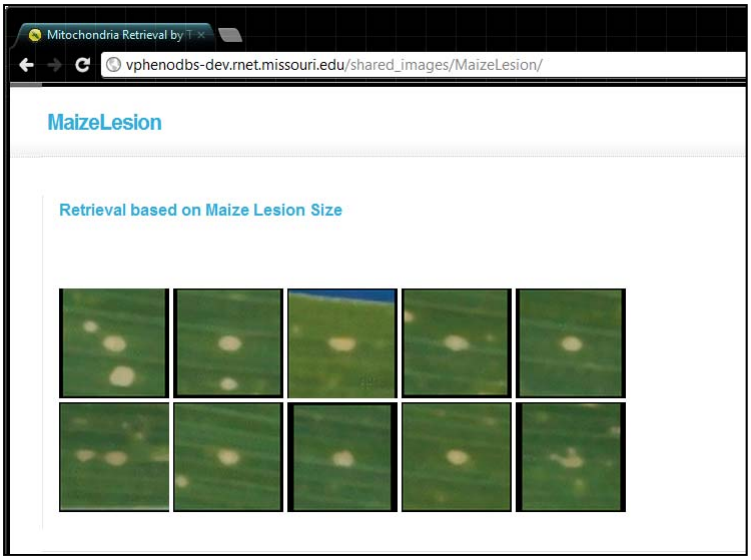


Figure 52: Query interface for the search mechanism designed to retrieve necrotic lesions based on temporal behavior, growth in this case.

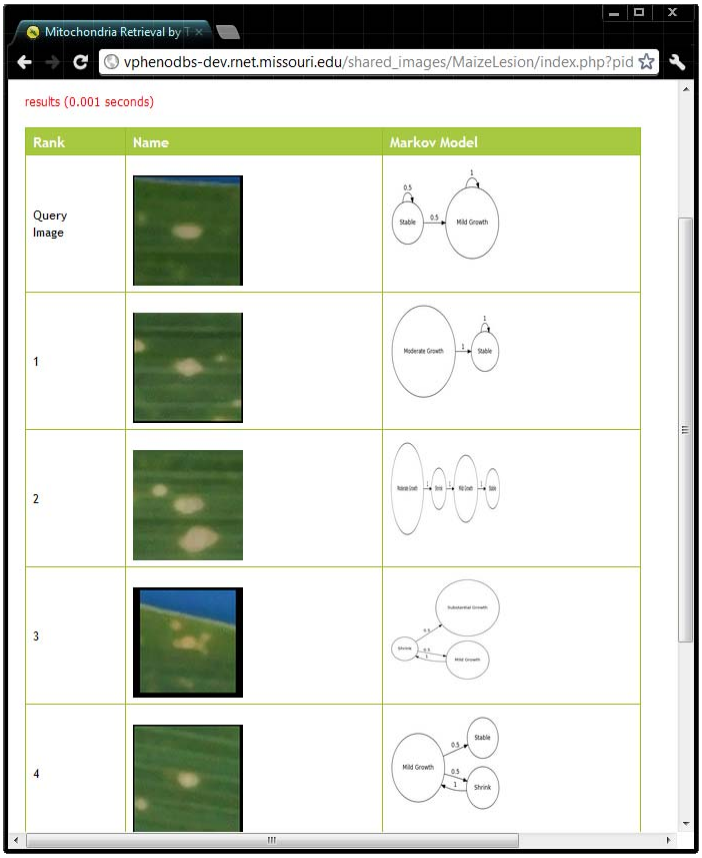


Figure 53: Results screen for the temporal search engine for necrotic lesions.

The concept of this temporal behavior search is illustrated further by another more extensive system that is part of the BioShapes project. This dataset contains time course studies of mitochondrial movement along axons in *Drosophila*. The mitochondria were segmented and their positions were tracked across image frames. Individual features related to object position, size, and shape were extracted.

Three types of temporal information were then modeled from these features: trajectory, size, and shape. In the trajectory models, an objects velocity and acceleration were calculated at each time point. Velocity and acceleration were binned into positive, negative, and zero, with states defined as combinations of velocity and acceleration values (e.g. positive velocity/positive acceleration). For both the size and shape modeling, users labeled a small subset of mitochondria with semantic labels. Using the algorithm described in [105], the feature subspaces corresponding to these labels were learned. This knowledge was applied to the entire dataset for state sequence generation. In the size models, the mitochondria were labeled as either small or large, and mitochondria shapes were labeled with the terms in Figure 54. Markov chains were built for each behavior modeled for each object and then indexed for retrieval, which allows the construction of a search mechanism for retrieving mitochondria with similar temporal behaviors.

Bead 	Elbow 	Pear 
Branched 	Horseshoe 	Rod 
Dumbbell 	Irregular 	Spade 

Figure 54: Shape labels for describing mitochondria in Drosophila.

The query page for the search mechanism is shown in Figure 55 below. The query thumbnails are animated images, which allow the user to see the movement of the mitochondria. Clicking on an image brings up a larger view of the object's temporal behavior and includes an option to search by trajectory, size, or shape. Upon selecting an option, the search is executed and the results appear as in Figure 56. Both animated images and Markov models are shown for the user to peruse.

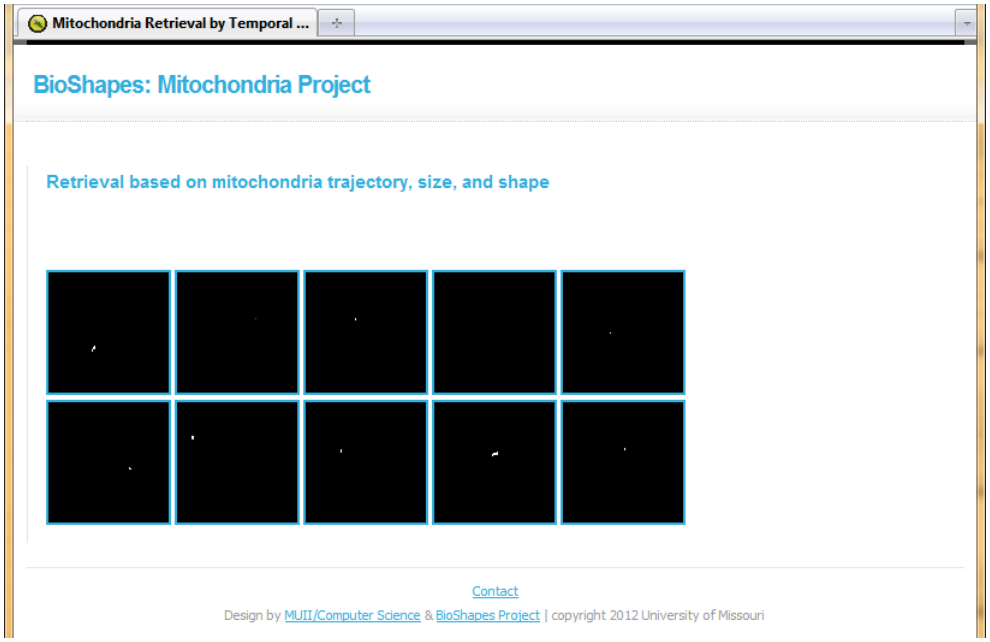


Figure 55: Query page for a search mechanism that retrieves mitochondria based on temporal behavior, either trajectory, size, or shape.

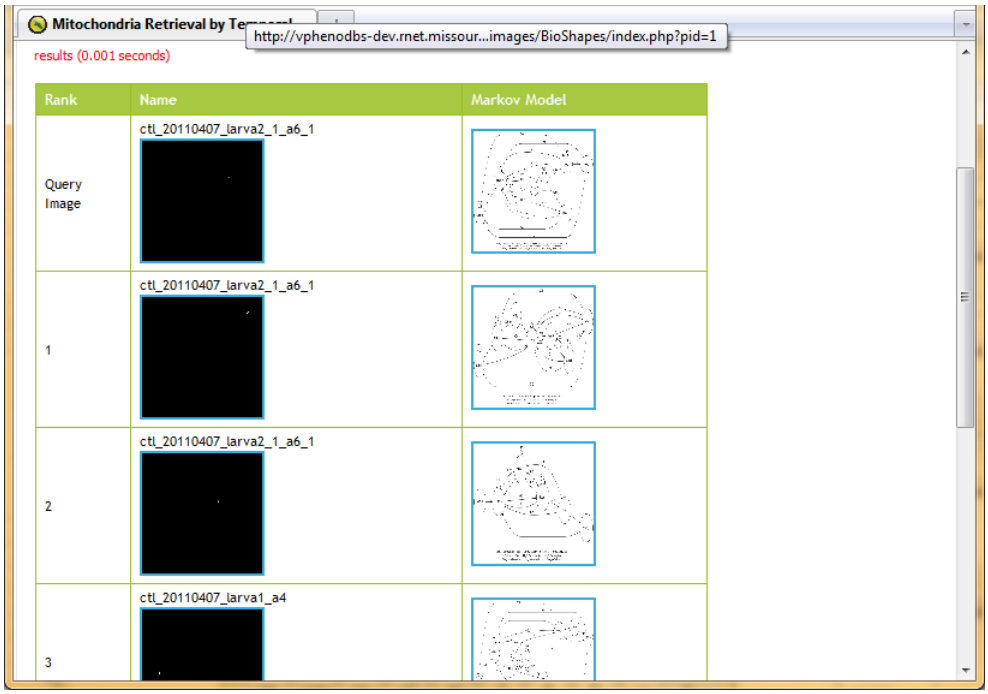


Figure 56: Results page for the mitochondria search mechanism, showing temporal behavior via animated images and Markov models.

CHAPTER SEVEN

CONCLUSIONS AND FUTURE WORK

In this chapter, we will review the major contributions presented in this dissertation as well as discuss some directions for future research based on the developed methods.

7.1 Conclusions

This dissertation centers around the development of a series of computational methods designed to support high-throughput analysis and search of visual phenotypes, with an emphasis placed on visual plant phenotypes. A summary of these contributions follows:

The notion of phenotype capture was first addressed in the dissertation, specifically in terms of an imaging protocol designed to standardize phenotype imaging using commonly found digital equipment. The rationale for such an undertaking stemmed from the current lack of standards for imaging phenotypes as well as a desire to facilitate the merging and/or comparison of disparate image datasets, which is most easily obtained via image standardization. Accompanying this protocol, we developed an image normalization method that utilizes a mini color checker, a required component of the imaging protocol, to standardize images in terms of color and scale. This method is invariant to color checker orientation and insensitive to obstruction of the color checker and image exposure. A proxy of expected algorithm performance as it relates to the degree of image exposure was also provided.

Additionally, methods for phenotype measurement and quantification were developed. Custom processing pipelines for measuring several traits (e.g. disease resistance, lesion burden, herbivory, growth) from a variety of species (e.g., *Zea mays*,

Arabidopsis thaliana, Glycine max, Brassica rapa) were designed that consisted of unique combinations of existing computer vision and image processing techniques. Low-level features relevant to the trait of interest were extracted into feature vectors. The quality of these features was evaluated. The features extracted for lesion mimic mutants yielded an overall classification accuracy of 90.4% in a difficult dataset with multiple classes of similarly-appearing mutants; those extracted for herbivory and leaf area were evaluated against the "gold standard" leaf area meter and ImageJ, and showed near identical results. The herbivory results were also compared against the popular leaf damage estimation method, which showed the advantage of using a more objective computational approach as well as the disadvantages of human scoring.

Supervised and unsupervised methods for translating low-level features into scores reflecting the degree of resistance to the fungal disease, Southern Leaf Blight, in maize were also developed, and these were compared to a traditional rubric-based method for measuring this trait. The results first show the inherent subjectivity and inconsistency in human perception. They also demonstrate the ability of both computational methods to more objectively score this trait with the supervised method obviously more closely matching the human labeling.

Next, an approach was developed for analyzing time course studies of maize leaves from lesion mimic mutants. This notably included an algorithm to register necrotic lesions across images as well as a method to model both the temporal behavior of individual necrotic lesions as well as the aggregated temporal characteristics of all lesions on a leaf using Markov chains. The registration algorithm was evaluated over a number of images in terms of mapping accuracy.

Finally, a series of applications were developed for the research community's use. First was a publically available web application called PhenoPhyte for helping plant scientists to automatically process leaf or rosette imagery for high-throughput analysis of area, herbivory, or growth. Additionally, a group of advanced search mechanisms were developed to aid researchers efficiently locate interesting information in large data collections. This includes a content-based image retrieval engine for lesion mimic mutants that utilizes multiple EBS kd-tree indexes for search. Additionally, a multi-source ontology-based text retrieval engine was developed for MaizeGDB phenotype data to more fully utilize free text fields. Searches utilize descriptions of phenotype image captions, genetic loci, and associated gene products and are linked to both the Gene and Plant Ontologies to increase the precision and accuracy of phenotype searches. Finally, a search mechanism was introduced for retrieving objects based on temporal behavior represented by Markov chains. This was demonstrated using both maize necrotic lesions and mitochondria. All these search mechanisms represent the *first* of their kinds in the plant community.

7.2 Future Work

7.2.1 Development of Algorithms for Other Species and Traits

In both biology and ecology, there are likely a large number of traits and phenotypes in multiple species and on various anatomical body parts that would benefit from the development of computational methods to facilitate higher-throughput analyses. The challenge here would be finding good collaborations, high quality data, and high impact phenotypes to work on. In addition to the work presented in this dissertation, we already have ongoing projects regarding phenotypes related to maize kernels and ears.

7.2.2 Expanded Time Course Study Analyses

Though the method for analyzing time course studies was developed in this work, expanded analyses, which likely require the careful collection of additional data, could be performed.

First, this could be utilized to answer some very interesting questions related to the temporal behavior of the same lesion mimic mutants. The aggregated temporal behavior of necrotic lesions could be modeled (1) on all or a specific subset of leaves of plants over the course of the growing season to quantify the differences in expression and temporal progression on different leaves and in different developmental stages of the plant as well as (2) on the same leaves of multiple plants of a specific genetic line within a particular row, across replications, and across sites to assess the variation in temporal progression as it relates to genetic and environmental factors.

Second, the same technique could be applied to other phenotypes in maize and other species. For example, *Arabidopsis* is known to have lesion mimic mutants as well, and thus the temporal behavior of the same HR response in that species could also be quantified and potentially compared or correlated with the corresponding behavior in maize. In addition to lesion mimic mutants, the progression of disease could also be monitored and quantified this way, including fungal diseases like Southern Leaf Blight in maize and various mildew phenotypes in grape.

7.2.3 Including Chlorotic Lesions in Time Course Study Analyses

In the work described in this dissertation, analysis of the formation and progression of necrotic lesions over time was performed via a lesion tracking/registration algorithm. Necrotic lesions, however, are not the only type of lesion that occurs in lesion mimic

mutants in maize. Chlorotic lesions, which correspond to areas of reduced chlorophyll and present as yellow or green lesions, are also common and oftentimes occur prior to the development of necrosis. The development of a more accurate and robust segmentation algorithm that captures and differentiates necrotic and chlorotic lesions could be explored. Once this is performed, additional insight into the temporal behavior of these mutants in maize could be explored by tracking the development of both chlorotic and necrotic lesions. The inclusion of chlorotic lesions adds an additional difficulty in that some mutants consist of necrotic lesions that appear amid a diffuse background of tiny chlorotic spots. In these cases, even though the broad area of chlorosis may be detectable, the separation of individual chlorotic lesions may not be possible. This adjustment would allow for the quantification of lesion progression in terms of spectral characteristics. In addition, the positions of lesions that initially appear later in time sequences could be traced backward to before the segmentation algorithm registered them as lesions to investigate for any subtle retrospective changes in the leaf that may serve as early indicators of lesion formation.

7.2.4 Modeling Semantics Related to Temporal Behavior

Plant scientists and researchers often use semantic descriptions to explain biological processes, including the dynamics of temporal behavior. As such, semantic descriptions could be elicited of the temporal behavior of maize lesions from experts. In doing so, the tacit knowledge used by these experts in assigning these semantic labels could be mathematically modeled from the extracted low-level features by utilizing computational methods like that in [105]. The results of applying such a method would allow the automatic annotation of temporal behavior using these semantic terms based

solely on the extracted low-level features. This also facilitates construction of a semantic search engine that allows users to specify using more natural language to describe the type of temporal phenomena they are interested in with subsequent retrieval of maize lesions that match such behavior.

7.2.5 Extending the Multi-Source Ontology-Based Phenotype Text Retrieval Engine

The text retrieval engine could be expanded in multiple ways. First, while query expansion is performed automatically in our search engine, we could also provide user-assisted query expansion, which allows the user to see the list of candidate expansions to the query and remove any unwanted terms from that list before the search is performed. In addition, we would like to investigate using relevance feedback techniques to automatically determine user-driven weights for the various parameters. Currently only free-text sources are searchable through this mechanism; however, the integration of other kinds of sources, e.g., attribute fields, into this search engine could be explored. Most interesting would be the possibility of including non-text sources like images and sequences into the search mechanism. Phenotype searches using these modalities alone have already been developed: MaizeGDB has a sequence search mechanism called POPcorn [106-107] available, and we have already explored in a previous work [108] how to represent phenotype images as feature vectors and perform image searches. In both cases, however, an approach to intelligently combine these various types of sources remains unstudied. Nevertheless, such a hybrid search mechanism that merges varied information sources would have great potential.

7.3 Final Thoughts

In this age of genetics, phenotyping is often the bottleneck in experimental studies. As such, the ability to efficiently and accurately characterize traits and phenotypes is critical to being able to produce the quantity and quality of data needed for scientific discovery. This dissertation focused specifically on developing high-throughput methods for phenotype analysis of traits in more challenging environmental conditions. These methods aim to produce accurate and unique trait data that can help researchers gain further understanding and insight into the underlying biological and genetic mechanisms that mediate phenotypic expression.

BIBLIOGRAPHY

[1] B. A. Pierce, *Genetics: A Conceptual Approach*, 2nd ed. New York: W.H. Freeman and Company, 2005.

[2] N. N. Artus, "Two mutants of Arabidopsis thaliana that become chlorotic in atmospheres enriched with CO2," *Plant, Cell & Environment*, vol. 13, pp. 575-580, 1990.

[3] K. Hiratsu, *et al.*, "Dominant repression of target genes by chimeric repressors that include the EAR motif, a repression domain, in Arabidopsis," *The Plant Journal*, vol. 34, pp. 733-739, 2003.

[4] B. Dombrecht, *et al.*, "MYC2 differentially modulates diverse jasmonate-dependent functions in Arabidopsis," *The Plant Cell Online*, vol. 19, pp. 2225-2245, 2007.

[5] K. L. Kump, *et al.*, "Genome-wide association study of quantitative resistance to southern leaf blight in the maize nested association mapping population," *Nat Genet*, vol. 43, pp. 163-168, 2011.

[6] H. U. Stotz, *et al.*, "Induced Plant Defense Responses against Chewing Insects. Ethylene Signaling Reduces Resistance of Arabidopsis against Egyptian Cotton Worm But Not Diamondback Moth," *Plant Physiology*, vol. 124, pp. 1007-1018, November 2000.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed.: Academic Press, Inc., 1982.

[9] S. R. Tracy, *et al.*, "The X-factor: visualizing undisturbed root architecture in soils using X-ray computed tomography," *Journal of Experimental Botany*, vol. 61, pp. 311-313, January 2010.

[10] P. J. Gregory, *et al.*, "Non-invasive imaging of roots with high resolution X-ray micro-tomography," *Plant and Soil*, vol. 255, pp. 351-359, 2003.

[11] P. A. Bottomley, *et al.*, "NMR imaging shows water distribution and transport in plant root systems in situ," *Proceedings of the National Academy of Sciences*, vol. 83, p. 87, 1986.

[12] A. S. Iyer-Pascuzzi, *et al.*, "Imaging and analysis platform for automatic phenotyping and trait ranking of plant root systems," *Plant Physiology,* vol. 152, pp. 1148-1157, 2010.

[13] R. T. Clark, *et al.*, "Three-dimensional root phenotyping with a novel imaging and software platform," *Plant Physiology,* vol. 156, pp. 455-465, 2011.

[14] M. Bylesjö, *et al.*, "LAMINA: a tool for rapid quantification of leaf size and shape parameters," *BMC Plant Biology,* vol. 8, p. 82, 2008.

[15] C. Weight, *et al.*, "LeafAnalyser: a computational method for rapid and large-scale analyses of leaf shape variation," *The Plant Journal,* vol. 53, pp. 578-586, 2008.

[16] A. Backhaus, *et al.*, "LeafProcessor: a new leaf phenotyping tool using contour bending energy and shape cluster analysis," *New Phytologist,* vol. 187, pp. 251-261, 2010.

[17] M. J. Fryer, *et al.*, "Imaging of photo-oxidative stress responses in leaves," *Journal of Experimental Botany*, vol. 53, pp. 1249-1254, May 2002.

[18] J. S. Veres, *et al.*, "Magnetic Resonance Imaging of Leaves," *New Phytologist,* vol. 123, pp. 769-774, 1993.

[19] S. Arvidsson, *et al.*, "A growth phenotyping pipeline for Arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects," *New Phytologist,* vol. 191, pp. 895-907, 2011.

[20] K. Rajendran, *et al.*, "Quantifying the three main components of salinity tolerance in cereals," *Plant, Cell & Environment,* vol. 32, pp. 237-249, 2009.

[21] B. Harris, *et al.*, "A water-centred framework to assess the effects of salinity on the growth and yield of wheat and barley," *Plant and Soil,* vol. 336, pp. 377-389, 2010.

[22] H. G. Jones, *et al.*, "Thermal infrared imaging of crop canopies for the remote diagnosis and quantification of plant responses to water stress in the field," *Functional Plant Biology,* vol. 36, pp. 978-989, 2009.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 9, pp. 62-66, 1979.

[24] D. C. Boyes, *et al.*, "Growth stage–based phenotypic analysis of Arabidopsis: A model for high throughput functional genomics in plants," *Plant Cell,* vol. 13, pp. 1499-1510, 2001.

[25] S. Fang, *et al.*, "Crop Root Behavior Coordinates Phosphorus Status and Neighbors: From Field Studies to Three-Dimensional in Situ Reconstruction of Root System Architecture," *Plant Physiology,* vol. 155, pp. 1277-1285, 2011.

[26] M. D. Abràmoff, *et al.*, "Image processing with ImageJ," *Biophotonics international,* vol. 11, pp. 36-42, 2004.

[27] P. L. Nurmberg, *et al.*, "The developmental selector AS1 is an evolutionarily conserved regulator of the plant immune response," *Proceedings of the National Academy of Sciences,* vol. 104, p. 18795, 2007.

[28] A. Al-Daoude, *et al.*, "RIN13 is a positive regulator of the plant disease resistance protein RPM1," *Plant Cell,* vol. 17, pp. 1016-1028, 2005.

[29] J. R. Vidal, *et al.*, "Evaluation of transgenic 'Chardonnay'(Vitis vinifera) containing magainin genes for resistance to crown gall and powdery mildew," *Transgenic Research,* vol. 15, pp. 69-82, 2006.

[30] L. Lucchese and S. Mitra, "Color image segmentation: A state-of-the-art survey," *PINSA,* vol. 67, pp. 207-221, 2001.

[31] H. D. Cheng, *et al.*, "Color image segmentation: advances and prospects," *Pattern Recognition,* vol. 34, pp. 2259-2281, 2001.

[32] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition,* vol. 26, pp. 1277-1294, 1993.

[33] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," *Pattern Recognition,* vol. 11, pp. 191-204, 1979.

[34] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition,* vol. 19, pp. 41-47, 1986.

[35] N. Pal and D. Bhandari, "Object background classification: Some new techniques," *Signal Processing,* vol. 33, pp. 139-158, 1993.

[36] T. Pun, "Entropic thresholding, a new approach," *Computer Graphics and Image Processing,* vol. 16, pp. 210-239, 1981.

[37] J. Kapur, *et al.*, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing,* vol. 29, pp. 273-285, 1985.

[38] A. K. C. Wong and P. Sahoo, "A gray-level threshold selection method based on maximum entropy principle," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 19, pp. 866-871, 1989.

[39] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 679-698, 1986.

[40] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[41] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision,* vol. 7, pp. 11-32, 1991.

[42] A. Materka and M. Strzelecki, "Texture analysis methods-a review," *Technical University of Lodz, Institute of Electronics, Cost B11 Report,* vol. 11, pp. 9-11, 1998.

[43] R. M. Haralick, *et al.*, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 3, pp. 610-621, 1973.

[44] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 33, pp. 959-963, 1985.

[45] H. Elliott, *et al.*, "Application of the Gibbs distribution to image segmentation," presented at the Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP, 1984.

[46] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *Image Processing, IEEE Transactions on,* vol. 2, pp. 429-441, 1993.

[47] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America,* vol. 2, pp. 1160-1169, 1985.

[48] Z. Feng, *et al.*, "Texture feature based on local Fourier transform," presented at the International Conference on Image Processing 2001.

[49] D. Fisher, *et al.*, "Leaf infection and yield loss caused by four Helminthosporium leaf diseases of corn," *Phytopathology,* vol. 66, pp. 942-944, 1976.

[50] P. Balint-Kurti, *et al.*, "Precise mapping of quantitative trait loci for resistance to southern leaf blight, caused by Cochliobolus heterostrophus race O, and flowering time using advanced intercross maize lines," *Genetics,* vol. 176, pp. 645-657, 2007.

[51] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM Press, 1999.

[52] C. Igathinathane, *et al.*, "Interactive computer software development for leaf area measurement," *Computers and Electronics in Agriculture,* vol. 51, pp. 1-16, 2006.

[53] P. D. Coley, "Herbivory and defensive characteristics of tree species in a lowland tropical forest," *Ecological monographs,* vol. 53, pp. 209-234, 1983.

[54] M. G. Neuffer, *et al.*, *Mutants of Maize*: Cold Spring Harbor Laboratory Press, 1997.

[55] G. Scott and S. Chi-Ren, "Knowledge-Driven Multidimensional Indexing Structure for Biomedical Media Database Retrieval," *Information Technology in Biomedicine, IEEE Transactions on,* vol. 11, pp. 320-331, 2007.

[56] C. H. Bock, *et al.*, "Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging," *Critical Reviews in Plant Sciences,* vol. 29, pp. 59-107, March 2010.

[57] G. S. Johal, *et al.*, "Disease lesion mimics of maize: a model for cell death in plants," *Bioessays,* vol. 17, pp. 685-692, 1995.

[58] C. D. Kuglin and D. C. Hines, "The phase correlation image alignment method," *IEEE Conference on Cybernetics and Society,* pp. 163-165, 1975.

[59] E. De Castro and C. Morandi, "Registration of Translated and Rotated Images Using Finite Fourier Transforms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. PAMI-9, pp. 700-703, 1987.

[60] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *Image Processing, IEEE Transactions on,* vol. 5, pp. 1266-1271, 1996.

[61] P. Viola and W. M. Wells Iii, "Alignment by Maximization of Mutual Information," *International Journal of Computer Vision,* vol. 24, pp. 137-154, 1997.

[62] P. Thevenaz and M. Unser, "An efficient mutual information optimizer for multiresolution image registration," presented at the International Conference on Image Processing, 1998.

[63] C. Studholme, *et al.*, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition,* vol. 32, pp. 71-86, 1999.

[64] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing,* vol. 21, pp. 977-1000, 2003.

[65] G. Stockman, *et al.*, "Matching Images to Models for Registration and Object Detection via Clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. PAMI-4, pp. 229-241, 1982.

[66] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognition,* vol. 12, pp. 269-275, 1980.

[67] H. Ogawa, "Labeled point pattern matching by fuzzy relaxation," *Pattern Recognition,* vol. 17, pp. 569-573, 1984.

[68] S. Irani and P. Raghavan, "Combinatorial and experimental results for randomized point matching algorithms," presented at the Proceedings of the 12th Annual Symposium on Computational Geometry, Philadelphia, PA, 1996.

[69] S. H. Chang, *et al.*, "Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes," *Pattern Recognition,* vol. 30, pp. 311-320, 1997.

[70] P. B. van Wamelen, *et al.*, "A fast expected time algorithm for the 2-D point pattern matching problem," *Pattern Recognition,* vol. 37, pp. 1699-1711, 2004.

[71] C. J. Lawrence, *et al.*, "MaizeGDB: the maize model organism database for basic, translational, and applied research," *International Journal of Plant Genomics,* 2008.

[72] P. Jaiswal, *et al.*, "Gramene: a bird's eye view of cereal genomes," *Nucleic Acids Research,* vol. 34, pp. D717-D723, 2006.

[73] D. Swarbreck, *et al.*, "The Arabidopsis Information Resource (TAIR): gene structure and function annotation," *Nucleic Acids Research,* vol. 36, pp. D1009-D1014, 2008.

[74] L. A. Mueller, *et al.*, "The SOL Genomics Network. A comparative resource for Solanaceae biology and beyond," *Plant Physiology,* vol. 138, pp. 1310-1317, 2005.

[75] D. Grant, *et al.*, "SoyBase, the USDA-ARS soybean genetics and genomics database," *Nucleic Acids Research,* vol. 38, pp. D843-D846, 2010.

[76] N. Kurata and Y. Yamazaki, "Oryzabase. An integrated biological and genome information database for rice," *Plant Physiology,* vol. 140, pp. 12-17, 2006.

[77] M. Ashburner, *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics,* vol. 25, p. 25, 2000.

[78] P. Jaiswal, *et al.*, "Plant Ontology (PO): a controlled vocabulary of plant structures and growth stages," *Comparative and Functional Genomics,* vol. 6, pp. 388-397, 2005.

[79] S. Banerjee and T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet," *Computational Linguistics and Intelligent Text Processing,* pp. 117-171, 2002.

[80] X. Li, *et al.*, "A WordNet-based algorithm for word sense disambiguation," presented at the Proceedings of the 14th International Joint Conference on Artifical Intelligence, 1995.

[81] D. Widdows, *et al.*, "Unsupervised monolingual and bilingual word-sense disambiguation of medical documents using UMLS," 2003.

[82] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," presented at the Proc AMIA Symp, 2001.

[83] S. Harabagiu and F. Lacatusu, "Topic themes for multi-document summarization," presented at the Proc of the 28th ACM SIGIR Conference on Research & Development in Information Retrieval, 2005.

[84] C. S. Lee, *et al.*, "A fuzzy ontology and its application to news summarization," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 35, pp. 859-880, 2005.

[85] G. Fu, *et al.*, "Ontology-based spatial query expansion in information retrieval," *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE,* vol. 3761, pp. 1466-1482, 2005.

[86] R. Navigli and P. Velardi, "An analysis of ontology-based query expansion strategies," presented at the Workshop on Adaptive Text Extraction and Mining, 2003.

[87] D. Wollersheim and J. W. Rahayu, "Ontology based query expansion framework for use in medical information systems," *International Journal of Web Information Systems,* vol. 1, pp. 101-115, 2005.

[88] J. Duvick, *et al.*, "PlantGDB: a resource for comparative plant genomics," *Nucleic Acids Research,* vol. 36, pp. D959-D965, 2008.

[89] W. J. Kent, *et al.*, "The human genome browser at UCSC," *Genome Research,* vol. 12, pp. 996-1006, 2002.

[90] Broad Institute, "Argo Genome Browser," ed, 2010.

[91] S. F. Altschul, *et al.*, "Basic local alignment search tool," *Journal of Molecular Biology,* vol. 215, pp. 403-410, 1990.

[92] C. R. Shyu, *et al.*, "ProteinDBS: a real-time retrieval system for protein structure comparison," *Nucleic Acids Research,* vol. 32, pp. W572-W575, 2004.

[93] J. Reneker, *et al.*, "ACMES: fast multiple-genome searches for short repeat sequences with concurrent cross-species information retrieval," *Nucleic Acids Research,* vol. 32, pp. W649-W653, 2004.

[94] S. F. Altschul, *et al.*, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research,* vol. 25, pp. 3389-3402, 1997.

[95] W. Niblack, *et al.*, "The QBIC project: Querying images by content using color, texture, and shape," presented at the Proc Storage and Retrieval for Image and Video Databases, Bellingham, WA, 1993.

[96] A. P. Pentland, *et al.*, "Photobook: Tools for content-based manipulation of image databases," presented at the Proc Storage and Retrieval for Image and Video Databases, Bellingham, WA, 1995.

[97] J. R. Smith and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," presented at the Proc of the 4th ACM International Conference on Multimedia, 1996.

[98] C. R. Shyu, *et al.*, "GeoIRIS: Geospatial information retrieval and indexing system—Content mining, semantics modeling, and complex queries," *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 45, pp. 839-852, 2007.

[99] A. Csillaghy, *et al.*, "Content-based image retrieval in astronomy," *Information Retrieval,* vol. 3, pp. 229-241, 2000.

[100] R. Datta, *et al.*, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys,* vol. 40, pp. 34-94, 2008.

[101] M. Oussalah, "Content based image retrieval: review of state of art and future directions," *Image Processing Theory, Tools, & Applications,* pp. 1-10, 2008.

[102] H. Müller, *et al.*, "A review of content-based image retrieval systems in medical applications—clinical benefits and future directions," *International Journal of Medical Informatics,* vol. 73, pp. 1-23, 2004.

[103] I. Yahiaoui and N. Boujemaa, "Content-based image retrieval in botanical collections for gene expression studies," *IEEE Intl Conf on Image Processing*, vol. 3, pp. 1240-1243, 2005.

[104] G. Salton, *et al.*, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, pp. 613-620, 1975.

[105] A. S. Barb and C. R. Shyu, "Visual-Semantic Modeling in Content-Based Geospatial Information Retrieval Using Associative Mining Techniques," *Geoscience and Remote Sensing Letters, IEEE*, vol. 7, pp. 38-42, 2010.

[106] T. Z. Sen, *et al.*, "MaizeGDB becomes 'sequence-centric'," *Database: The Journal of Biological Databases and Curation*, vol. 2009, 2009.

[107] E. K. S. Cannon, *et al.*, "POPcorn: an online resource providing access to distributed and diverse maize project data," *International journal of plant genomics*, vol. 2011, 2011.

[108] C. R. Shyu, *et al.*, "Searching and Mining Visually Observed Phenotypes of Maize Mutants," *Journal of Bioinformatics and Computational Biology*, vol. 5, pp. 1193-1214, 2007.

VITA

Jason Green received his Ph.D. degree in computer science from the University of Missouri in 2012. He previously received his M.S. degree in computer science from the University of Missouri in 2009, and graduated *summa cum laude* in 2005 from the University of Missouri with three B.S. degrees in computer science, computer engineering, and mathematics.

Since 2005, he has worked as a Graduate Research Assistant in the Medical and Biological Digital Library Research Laboratory under the direction of Dr. Chi-Ren Shyu at the University of Missouri. From 2006-2011, he also served as a Pre-Doctoral Fellow for the National Library of Medicine Biomedical and Health Informatics Research Training Program. During this time, he was twice (in 2009 and in 2012) named the Outstanding Graduate Student in the Computer Science Department as part of the annual Missouri Honor Awards. In addition to research duties, he served as a Graduate Instructor for both junior and senior level database courses for the Computer Science Department.

During his graduate studies at the University of Missouri, Jason conducted research on several collaborative projects that included designing and maintaining databases for both biological and healthcare data, developing and implementing several advanced search mechanisms for a variety of biomedical data, and developing algorithms for high-throughput phenotype analysis as well as for analyzing healthcare data. These works resulted in several publications, conference posters, and presentations.