Locating Crop Plant Centers From UAV-Based RGB Imagery

Yuhao Chen[1] Javier Ribera[1] Christopher Boomsma[2] Edward Delp[1]

[1]Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

[2]Department of Agronomy
School of Agriculture
Purdue University
West Lafayette, Indiana, USA

Abstract

In this paper we propose a method to find the location of crop plants in Unmanned Aerial Vehicle (UAV) imagery. Finding the location of plants is a crucial step to derive and track phenotypic traits for each plant. We describe some initial work in estimating field crop plant locations. We approach the problem by classifying pixels as a plant center or a non plant center. We use Multiple Instance Learning (MIL) to handle the ambiguity of plant center labeling in training data. The classification results are then post-processed to estimate the exact location of the crop plant. Experimental evaluation is conducted to evaluate the method and the result achieved an overall precision and recall of 66% *and* 64%*, respectively.*

1. Introduction

Phenotyping is the process of measuring the physical properties of a plant, and it is critical in determining how genetic and environmental factors influence the physical and chemical properties of a plant [1, 2, 3]. Traditional field phenotyping requires a great deal of human labor to routinely obtain traits measurement on a plant by plant basis, making data acquisition very time-consuming. For example, workers first need to identify an individual plant from its neighbors, because plants in the field are typically planted densely to maximize yield. Then, leaves are untangled and a number of traits are recorded with multiple devices. Since measurements are done plant by plant, field size directly relates to the time needed for collecting data. Thus, labor requirements become the bottleneck of plant phenotyping in the field. In addition, damage done to plants during measurements may affect future experiments.

While traditional phenotyping techniques are labor intensive, modern imaging technologies have provided faster and more cost-effective plant phenotyping [3, 4, 5, 6, 7]. Phenotyping experiments can be conducted in greenhouses or outdoor fields. In greenhouse environments, phenotyping systems have been developed to provide fast and accurate data collection [8, 9, 10]. In [10], sorghum plants are 3D reconstructed from depth images. Greenhouse phenotyping systems can provide high precision measurements, but they are limited in terms of scalability and reflecting real growing conditions in the field. For example, in [9], plants must be individually moved inside a phenotyping chamber, thus limiting scalability. Also, plants are grown in individual pots, as opposed to field conditions where they share the same ground. In outdoor environments, phenotyping using Unmanned Aerial Vehicle (UAV) imagery has become popular due to its efficiency and non-invasiveness [11, 12, 13, 14]. One issue is that plants in UAV images have lower resolution and less consistent lighting conditions compared to plants in a greenhouse, making UAV images more difficult to analyze.

Identifying individual plants is a crucial step in analyzing UAV image data in field based plant-level phenotyping. In [15], the centers of the plants are selected as a starting point to extract phenotypic data. Since leaves spread out from a plant's center, the location of the center is important to study the plant structure and distinguish leaves from leaves of neighboring plants. Distinguishing individual plants from their neighbors is essential to track the growth of each plant, and obtain precise estimates of plant traits. However, locating field based crop plant centers is not a trivial task due to heavy occlusion.

In this paper, we describe some initial work in estimating plant centers from UAV images. In order to handle the ambiguity of the plant centers in our training data, we use Multiple Instance Learning (MIL) [16, 17] because it can

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000593. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Address all correspondence to Edward J. Delp, ace@ecn.purdue.edu

Figure 1: Image of a subrow of a sorghum field acquired from a UAV at an altitude of 40m.

tolerate some error in the labels. We approach the problem by classifying/labeling each pixel as a plant center or not a plant center. After the candidate locations are classified, we generate a mask indicating the estimated plant centers. From this mask, we find the plant centers by finding the centroids of large clusters. We focus our analysis on the crop plant sorghum [*Sorghum bicolor* (L.) Moench] [18, 1].

2. Related Work

Previous work in estimating plant centers generally work only on non-overlapping plants or require additional inputs. In [19], a mixture of Gaussian color model of plants is estimated by using expectation maximization. The model is used to segment the image into plants and background. A connected components analysis is used and each plant is identified by their component mask. In [15], the skeleton of the plant segmentation mask is estimated. The shortest paths between endpoints of the skeleton are detected. Then, the center of the plant is estimated by the center of the most frequently used path segment. In [20], leaf edges are detected and used to segment leaves. The longest line across each segmented leaf shape is considered as the orientation line of the leaf. The center of the plant is estimated by finding a point that is close to all orientation lines. The number of plants is assumed to be known in [21]. The centers are randomly initialized and moved iteratively to minimize a cost function. The cost function is modeled to account for the plants' vertical and horizontal alignment and the distance between its center and surrounding segmentation mask pixels.

Deep learning has been used to effectively segment objects in clustered scenes. In [22, 23], neural networks are used to segment multiple objects. Large labeled data sets are required to achieve accurate results. In order to employ deep learning to locate each plant, we would need a large quantity of labeled samples. There is no such publicly available dataset.

3. Multiple Instance Learning

In this section, we describe how we train a classifier using Multiple Instance Learning (MIL) [17, 16] to determine if a location is a plant center. Many studies have successfully used MIL with a boosting framework [24] to solve object detection problems [17, 25, 26, 27, 28]. In this paper, we implement the MILBoosting presented in [17].

In MIL, each sample is considered an instance. All the instances are put into bags. A bag must contain at least one instance and an instance may exists in multiple bags. We define the label of each bag based on the following criteria. A bag is considered positive if at least one of the instance in the bag is positive. A bag is considered negative if all of the instances in the bag are negative. Unlike in a common classification problem, the label for each instance can tolerate some error. As long as the label of the bag is correct, the classifier can be properly trained. Therefore, we can train the classifier with an approximate label of the plant location.

For a candidate plant location $P = (P_x, P_y)$ in image, we train classifier C to determine if location P is the center of the plant. The pixel value at location P may not be sufficient to discriminate between plant center or not plant center. We design the classifier C to consider the surrounding leaf structures of the plant for classification. Let $W(P)$ be the gray scale image that corresponds to a square window of size w centered at P (Figure 2a). The classifier is defined as:

$$C(W(P)) = \begin{cases} 1 & P \text{ is plant's center} \\ 0 & P \text{ not plant's center.} \end{cases} \quad (1)$$

Let $W(P)$ be an instance. We denote the i-th bag as x_i, and the j-th instance of the i-th bag as x_{ij}. Each bag x_i has N_i instances. We assign y_i to be the label of i-th bag, and assign y_{ij} to be the label of j-th instance in i-th bag.

We train a strong classifier from K weak classifiers. Let $c_k(x_{ij})$ be the classification result of the k-th weak classifier on instance x_{ij}. Then, we define the strong classifier as a linear combination of K weak classifiers:

$$C(x_{ij}) = \sum_{k=1}^{K} \lambda_k c_k(x_{ij}). \quad (2)$$

The label y_{ij} for instance x_{ij} is obtained as

$$y_{ij} = \begin{cases} 1 & \text{if } C(x_{ij}) > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Based on the definition of a bag, the label y_i for bag x_i is:

$$y_i = \max_j y_{ij} \quad (4)$$

To train the classifier, we model the probabilities that the instances are positive and use them to compute the probability for each bag being positive. Then we compute the likelihood function for all bags. Finally, we construct the strong classifier finding the best K weak classifier that maximize the likelihood function iteratively. The probability p_{ij}

of the j-th instance in the i-th bag is modeled as a sigmoid function of its classification result $C(x_{ij})$.

$$p_{ij} = \sigma(C(x_{ij})) = \frac{1}{1 + e^{-C(x_{ij})}} \tag{5}$$

Once we estimate the instance probabilities, we use the noisy-or (NOR) boosting frameworks presented in [17, 26] to compute the bag probability. The probability of a bag being positive is

$$p_i = 1 - \prod_j (1 - p_{ij}). \tag{6}$$

After the probabilities of all I bags are computed, we compute the likelihood L of all bags being correctly classified by classifier C as

$$L(C) = \prod_{i=1}^{I} p_i^{y_i} (1 - p_i)^{1 - y_i}, \tag{7}$$

and

$$\log(L(C)) = \sum_i (y_i \log p_i + (1 - y_i) \log(1 - p_i)). \tag{8}$$

As in the Milboost presented in [17], the weight w_{ij} of each sample is computed by the derivative of the log likelihood function:

$$w_{ij} = \frac{\partial log(L)}{\partial C(x_{ij})} = \frac{y_i - p_i}{p_i} p_{ij} \tag{9}$$

k-th weak classifier is selected to maximize the sum score of all instances $\sum_{i,j} c_k(x_{ij}) w_{ij}$. Then, we use exhaustive search to determine the weight λ_k of each classifier over the range [0,1] that maximize the updated log likelihood with classifier $C + \lambda_k c_k$. Algorithm 1 shows the pseudocode to train the classifier.

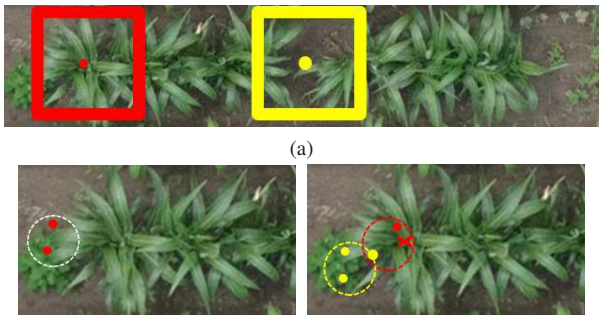
4. Features

In this paper, we use Gabor features [29]. Gabor features has been widely used in many recognition applications because of its spatial locality and orientation selectivity [30, 31, 32, 33]. Leaves exhibit thin triangle-like shapes with different orientations that Gabor features will be able to capture while accounting for the illumination changes [30].

We denote the value of the n-th feature computed on instance x_{ij} as $v_{n,ij}$. Gabor wavelets are defined as

$$f(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{-j\omega(x \cos\theta + y \sin\theta)}, \tag{10}$$

where (x, y) are the 2D image coordinates, σ is the variance of the Gaussian, ω is the oscillation frequency, and θ is



(a) (b) (c)

Figure 2: a) A row of plants with positive sample (red dot) and negative sample (yellow dot). The window for classification is marked as a square for each sample. b) A bag (white circle) contains 2 instance (red dot). c) A positive bag (red circle) contains a positive instance (red dot) and a negative instance (yellow dot), and a negative bag contains 3 negative instances. Ground truth location is marked in red X.

Algorithm 1: Training Classifier

Input : Data set: $\{X_1, X_2..X_N\}$, where $X_i = \{x_{i1}, x_{i2}, ...x_{iN_i}\}$,
Label set: $\{y_1, y_2..y_N\}$

Initialize $w_{ij} \leftarrow 1$ for all positive instances
Initialize $w_{ij} \leftarrow -1$ for all negative instances
for $k = 1$ *to* K **do**
  $c_k \leftarrow \underset{c}{\text{argmax}} \sum_{i,j} c(x_{ij}) w_{ij}$
  $\lambda_k \leftarrow \underset{\lambda}{\text{argmax}} \log L \left(\sum_{l=1}^{k-1} \lambda_l c_l + \lambda c_k \right)$
  for *all i* **do**
    for *all j* **do**
      $p_{ij} \leftarrow \sigma\left(\sum_{l=1}^{k} \lambda_l c_l(x_{ij}) \right)$
    end
    $p_i \leftarrow 1 - \prod_j (1 - p_{ij})$
  end
  Update $w_{ij} \leftarrow \frac{y_i - p_i}{p_i} p_{ij}$ for all i, j
end
$C \leftarrow \sum_{k=1}^{K} \lambda_k c_k$

the orientation angle. Gabor wavelets with orientations $\theta = \{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3}{8}\pi, \frac{\pi}{2}, \frac{5}{8}\pi, \frac{3}{4}\pi, \frac{7}{8}\pi\}$ are generated. The wavelets are also generated with frequencies $\omega = \{\frac{\pi}{\sqrt{2}}, \frac{\pi}{2}, \frac{\pi}{2\sqrt{2}}, \frac{\pi}{4}, \frac{\pi}{4\sqrt{2}}\}$.

In order to account for how leaves relate to the plant center, we divide the classification window into 9 equally spaced sections as shown in Figure 3a and Figure 3b. In a top-view sorghum image, leaves spread out from the center. We assume leaves in each section follow a specific orien-

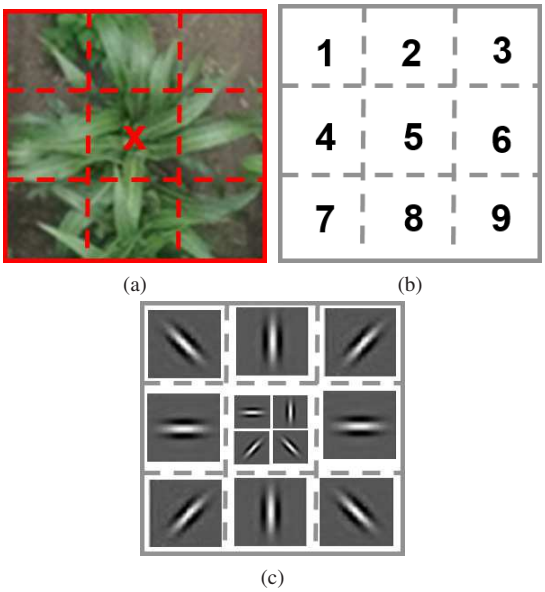


Figure 3: a) The lines that divide the classification window. b) Section numbers of the classification window. c) Orientations of the Gabor feature in each section.

Section	Orientations	Features / pixel
1, 9	$\frac{5}{8}\pi, \frac{3}{4}\pi, \frac{7}{8}\pi$	15
2, 8	$\frac{7}{8}\pi, 0, \frac{\pi}{8}$	15
3, 7	$\frac{\pi}{8}, \frac{\pi}{4}, \frac{3}{8}\pi$	15
4, 6	$\frac{3}{8}\pi, \frac{\pi}{2}, \frac{5}{8}\pi$	15
5	$0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3}{8}\pi, \frac{\pi}{2}, \frac{5}{8}\pi, \frac{3}{4}\pi, \frac{7}{8}\pi$	40

Table 1: Orientations of the Gabor wavelets

tation as shown in Figure 3c. Section 1, 3, 7, and 9 have diagonal leaves. Section 2 and 8 have vertical leaves. Section 4 and 6 have horizontal leaves. Section 5 may have horizontal, vertical and diagonal leaves. Any leaf that does not follow this assumption is from a neighbor plant. With this design, cases where the classifier incorrectly classifies a neighbor plant can be avoided. Table 1 shows the orientations of Gabor wavelets we used and were assigned to each window section. Gabor features are computed by convolving the window section image with a Gabor wavelet. The result of the convolution at each pixel represents a Gabor feature. We choose these wavelets based on the work in [32, 31]. The variance σ of the gabor wavelets is selected to be $\frac{pi}{2}$ to account for small window section size.

5. Weak Classifier

We use naive Bayes classifier to replace the Sign classifier used in [17]. The Sign classifier maps the feature values to two discrete values, 1 for a positive result and value 0 for a negative result. We are unable to determine if a classifier with one feature is better than the classifier with another feature when the classification results are both positive. A Sign classifier is selected based on the number of instances it correctly classified. In our application, the data labels can be wrong. Thus, the quantity of correct classification is not a reliable measurement for the performance of a classifier. We need a classifier that can estimate how strong an instance is classified to be positive or negative. Naive Bayes classifier is a better choice. A Naive Bayes classifier that uses n-th feature is given by:

$$c^{(n)}(x_{ij}) = \log \frac{p(y_{ij} = 1|v_{n,ij})}{p(y_{ij} = 0|v_{n,ij})}. \tag{11}$$

By using Bayes rule, the above equation becomes

$$c^{(n)}(x_{ij}) = \log \frac{p(v_{n,ij}|y_{ij} = 1)p(y_{ij} = 1)}{p(v_{n,ij}|y_{ij} = 0)p(y_{ij} = 0)}. \tag{12}$$

We let $p(y_{ij} = 1) = p(y_{ij} = 0)$, which means we have equal amount of positive instances and negative instances. We model $p(v_{n,ij}|y_{ij} = 1)$ to be Gaussian with mean μ_1 and variance σ_1, and $p(v_{n,ij}|y_{ij} = 0)$ to be Gaussian with mean μ_0 and variance σ_0. The means μ_0, μ_1, and the variances σ_0, σ_1 can be computed as

$$\mu_1 = \frac{1}{N_t} \sum_{i,j} v_{n,ij} y_{ij}, \tag{13}$$

$$\mu_0 = \frac{1}{N_t} \sum_{i,j} v_{n,ij} (1 - y_{ij}), \tag{14}$$

$$\sigma_1 = \sqrt{\frac{1}{N_t} \sum_{i,j} y_{ij} (v_{n,ij} - \mu_1)^2}, \tag{15}$$

$$\sigma_0 = \sqrt{\frac{1}{N_t} \sum_{i,j} (1 - y_{ij}) (v_{n,ij} - \mu_0)^2}, \tag{16}$$

where N_t is the total number of instances.

6. Post-Processing

After the locations in an image are classified, we obtain a map M with classification labels as shown in Figure 4c and Figure 4d, where $M(P_x, P_y) \in \{0, 1\}$. Because our classifier is trained to classify areas and specific location is not given, the classification label map M shows the clusters of possible plant center location.

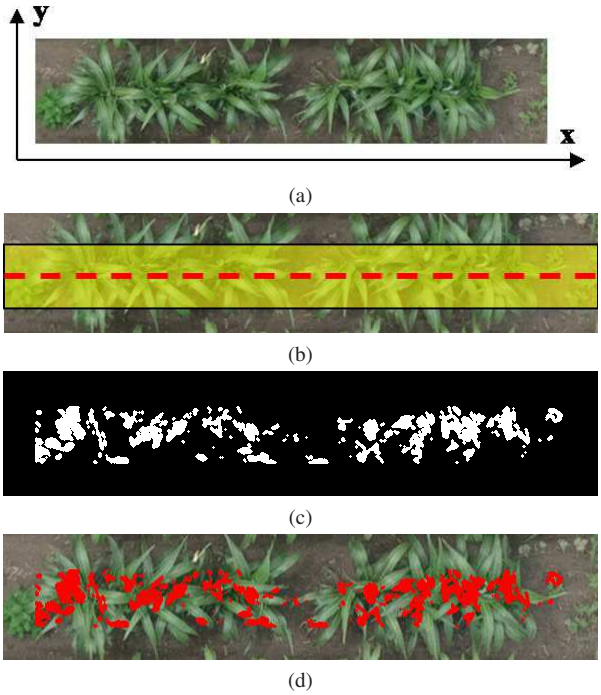


Figure 4: a) Original subrow with coordinates system. b) Plant location region (yellow region) and the $\mathcal{Y}$ direction mid line (red dash line). c) Classification label map M (positive: white, negative: black). d) Classification label marked in red in original image.

We further process the label map M to estimate the plant center. We assume small clusters in M are noise. The connected component analysis [34] is used to the label map M to obtain individual clusters. Small components are removed by thresholding the number of pixels in the components with threshold τ_a. Let the horizontal direction be denoted as $\mathcal{X}$, and let the vertical direction be denoted as $\mathcal{Y}$, as shown in 4a. We assume all the plants in a subrow are somewhat aligned in the $\mathcal{X}$ direction, so we only consider results that are within a distance τ_b to the $\mathcal{Y}$ mid line (see Figure 4b). Some candidate centers may come from the same plant but may be in different components. We merge components whose centroids are within a distance of τ_c of each other in $\mathcal{X}$ direction. After the merging, the centroid of each component is considered as the center of plants for our results.

7. Experiment and Results

Our dataset was collected from a UAV in a field of sorghum on July 19th, 2016. The images were taken at an altitude of 40 meters at UAV velocity of 8m/s. Figure 5 shows an example image in our dataset. We cropped 26 subrows from the original image, and converted them into grayscale images. We used 6 subrows for training, and 20 subrows for testing. The cropped subrows were selected to have low lens and perspective distortion.

Figure 5: Image of a section of a sorghum field acquired from a UAV at an altitude of 40m. The pixel resolution is 0.59 cm/pixel.

We ground truthed the center locations for each plant. Each ground truthed location corresponds to one positive bag. We labeled the locations around each ground truth as positive if the location is within 5 pixels of the ground truth. Those positive instances are put into the same bag as the ground truthed location. The locations that are at least 30 pixels away from any of the ground truth are considered as negative instances, and each negative instance is put into one bag. Positive instances and negative instances are randomly selected to reduce the computation cost. We used 800 positive instances and 800 negative instances in the 6 subrows for training. We tested our method with 10 high leaf density subrows and 10 low leaf density subrows.

We set the classification window width w to be the average plant diameter, 99 pixels. The number of weak classifiers K is set to be 10. The threshold τ_a to remove small components is set to be 100 pixels empirically. The threshold τ_b is set to be $\frac{1}{3}w = 33$. The merge distance τ_c is set to be 30 pixels. To reduce the computational cost, we used locations in the plants segmentation mask as candidate locations for classification, and the mask is obtained by applying the color threshold method described in [21].

Precision and recall was used to evaluate our method [35, 36]. Precision is defined as

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{17}$$

and recall is defined as

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

If a center is detected within 30 pixels of the ground truth, the center will be considered as a true positive (TP). If more

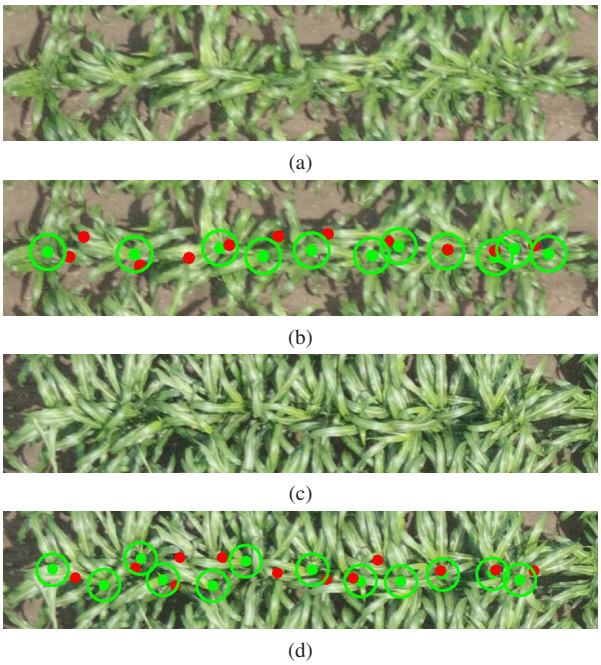


Figure 6: a) A subrow with low leaf density. b) The center of the red dots are the detected plant centers, the center of the green dots are the ground truth. If a detected plant falls inside the green circumference, we consider it a true positive. c) A subrow with high leaf density. d) Detected centers (red dots) of high leaf density plants.

Type	TP	FP	FN	Precision	Recall
High Density	779	506	431	61%	64%
Low Density	701	242	399	74%	64%
Overall	1480	748	830	66%	64%

Table 2: Results for plants with different leaf density after training and evaluating 10 times. Our dataset contains 121 high density plants and 110 low density plants.

than one center is detected within 30 pixels of any ground truth, the extra detection will be counted as false positive (FP). Note that this distance is the same as the distance we use to generate negative samples during training. False negatives (FN) occur when no center is detected for a ground truth location. As the instances are randomly selected during training, the precision and recall may vary when evaluated. We report the results after training and evaluating 10 times. Table 2 shows the results for high leaf density data and low leaf density data. Our result obtained an average overall precision of 66%, and a recall of 64%. Figure 6 shows an example output for a low leaf density subrow, and an example output for a high leaf density subrow.

The results show that the the method produces more false positive with high leaf density data. This is due to the fact that the classifier is trained to look for the structures of surrounding leaves. When plants have small amount of leaves, the structures around the plant centers are clear. Leaves from the same plant point toward the plant center. If a pixel location is not a plant center, then the leaves around it will converge to different location. When plants have high leaf density, there is more leaf overlapping. The overlapping makes the leaf structures more complex, and confuses the classifier. As a result, there are more false positives.

The performance of our method can be interpreted as follows. In our application, a plant consists of an indefinite number of leaves. Each leaf has a unique shape including length, width, and leaf curvature. Plants have different number of leaves in various orientations. A combination of simple features may not be enough for describing heterogeneous sorghum plants. In addition, our negative training data may have false labels due to human error during labeling. Plant centers can easily be missed during labeling, and may accidentally be used as negative training data.

8. Conclusions

This paper presented a method to find the centers of sorghum plants from UAV imagery. Pixel locations are classified as plant center or not plant center. Areas of possible plant centers are estimated by MIL. Then, the areas are merged based on their centroid distance. Finally, we estimate the center of the plants to be the centroids of each merged area. Experiments were conducted on plants with high and low leaf densities. Results show that the method works better when the plants have low leaf density. Results also show the limitation of using simple features to describe diverse sorghum plants. In the future, we investigate the use of multiple simple features that can be combined into secondary features. Boosting frameworks will be used with the features to find a feature set that defines sorghum plants centered in image. Future work will also include using deep learning for classification and refining the post-processing method with a statistical approach.

References

[1] S. K. Panguluri and A. A. Kumar, "Phenotyping in sorghum," in *Phenotyping for Plant Breeding: Applications of Phenotyping Methods for Crop Improvement*, vol. 1, pp. 73–110. Springer New York, New York, NY, 2013. 1, 2

[2] F. Fiorani and U. Schurr, "Future scenarios for plant phenotyping," *Annual Review of Plant Biology*, vol. 64, pp. 267–291, February 2013. 1

We thank Professor Ayman Habib and the Digital Photogrammetry Research Group (DPRG) from the School of Civil Engineering at Purdue University for providing the images used in this paper.

[3] R. T. Furbank and M. Tester, "Phenomics-technologies to relieve the phenotyping bottleneck," *Trends in Plant Science*, vol. 16, no. 12, pp. 635–644, November 2011. 1

[4] J. W. White, P. Andrade-Sanchez, M. A. Gore, K. F. Bronson, T. A. Coffelt, M. M. Conley, K. A. Feldmann, A. N. French, J. T. Heun, D. J. Hunsaker, M. A. Jenks, B. A. Kimball, R. L. Roth, R. J. Strand, K. R. Thorp, G. W. Wall, and G. Wang, "Field-based phenomics for plant genetics research," *Field Crops Research*, vol. 133, pp. 101–112, July 2012. 1

[5] G. A. Johnson, G. P. Cofer, S. L. Gewalt, and L. W. Hedlund, "An engineering approach to image-based phenotyping," *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 381–383, July 2002, Washington, DC. 1

[6] A. Mutka and R. Bart, "Image-based phenotyping of plant disease symptoms," *Frontiers in Plant Science*, vol. 5, no. 734, January 2015. 1

[7] A. Bucksch, J. Burridge, L. M. York, A. Das, E. Nord, J. S. Weitz, and J. P. Lynch, "Image-based high-throughput field phenotyping of crop roots," *Plant Phisiology*, vol. 166, no. 2, pp. 470–486, October 2014. 1

[8] C. Granier, L. Aguirrezabal, K. Chenu, S. J. Cookson, M. Dauzat, P. Hamard, J. Thioux, G. Rolland, S. Bouchier-Combaud, A. Lebaudy, B. Muller, T. Simonneau, and F. Tardieu, "Phenopsis, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit," *New Phytologist*, vol. 169, no. 3, pp. 623–635, Novermber 2005. 1

[9] N. Fahlgren, M. Feldman, M. A. Gehan, C. Shyu M. S. Wilson, D. W. Bryant, S. T. Hill, C. J. McEntee, S. N. Warnasooriya, I. Kumar, T. Ficor, S. Turnipseed, K. B. Gilbert, T. P. Brutnell, J. C. Carrington, T. C. Mockler, and I. Baxter, "A versatile phenotyping system and analytics platform reveals diverse temporal responses to water availability in Setaria," *Molecular Plant*, vol. 8, no. 10, June 2015. 1

[10] R. F. McCormick, S. K. Truong, and J. E. Mullet, "3D sorghum reconstructions from depth images identify QTL regulating shoot architecture," *Plant Physiology*, vol. 172, no. 2, pp. 823–834, October 2016. 1

[11] F. A. Vega, F. C. Ramírez, M. P. Saiz, and F. O. Rosúa, "Multi-temporal imaging using an unmanned aerial vehicle for monitoring a sunflower crop," *Biosystems Engineering*, vol. 132, pp. 19–27, January 2015. 1

[12] S. Sankaran, L. R. Khot, C. Z. Espinoza, S. Jarolmasjed, V. R. Sathuvalli, G. J. Vandemark, P. N. Miklas, A. H. Carter, M. O. Pumphrey, N. R. Knowles, and M. J. Pavek, "Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review," *European Journal of Agronomy*, vol. 70, pp. 112 – 123, October 2015. 1

[13] S. C. Chapman, T. Merz, A. Chan, P. Jackway, S. Hrabar, M. F. Dreccer, E. Holland, B. Zheng, T. J. Ling, and J. Jimenez-Berni, "Pheno-copter: A low-altitude, autonomous remote-sensing robotic helicopter for high-throughput field-based phenotyping," *Agronomy*, vol. 4, no. 2, pp. 279–301, June 2014. 1

[14] R. Sugiura, N. Noguchi, and K. Ishii, "Remote-sensing technology for vegetation monitoring using an unmanned helicopter," *Biosystems Engineering*, vol. 90, no. 4, pp. 369–379, April 2005. 1

[15] M. V. Giuffrida, M. Minervini, and S. Tsaftaris, "Learning to count leaves in rosette plants," *Proceedings of the Computer Vision Problems in Plant Phenotyping*, pp. 1.1–1.13, September 2015, Swansea, UK. 1, 2

[16] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, January 1997. 1, 2

[17] P. Viola, J. C. Platt, C. Zhang, et al., "Multiple instance boosting for object detection," *Proceedings of the Neural Information Processing Systems Conference*, January 2005, Vancouver, Canada. 1, 2, 3, 4

[18] National Research Council, "Sorghum," in *Lost Crops of Africa. Volume I: Grains*, vol. 1, pp. 127–144. The National Academies Press, Washington, DC, 1996. 2

[19] J. D. Vylder, F. Vandenbussche, Y. Hu, W. Philips, and V. D. S. Dominique, "Rosette tracker: An open source image analysis tool for automatic quantification of genotype effects," *Plant physiology*, vol. 160, no. 3, pp. 1149–1159, November 2012. 2

[20] O. L. Tessmer, Y. Jiao, J. A. Cruz, D. M. Kramer, and J. Chen, "Functional approach to high-throughput plant growth analysis," *BMC systems biology*, vol. 7, no. 6, pp. S17, December 2013. 2

[21] J. Ribera, F. He, Y. Chen, A. F. Habib, and E. J. Delp, "Estimating phenotypic traits from UAV based RGB imagery," *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Workshop on Data Science for Food, Energy, and Water*, August 2016, San Francisco, CA. 2, 5

[22] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," *Proceedings of European Conference on Computer Vision*, pp. 695–711, September 2016, Amsterdam, The Netherlands. 2

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, June 2015, Boston, MA. 2

[24] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–I, April 2001, Kauai, HI. 2

[25] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, August 2011. 2

[26] Y. Xu, J. Zhu, E. Chang, and Z. Tu, "Multiple clustered instance learning for histopathology cancer image classification, segmentation and clustering," *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pp. 964–971, June 2012, Providence, RI. 2, 3

[27] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu, "Multiple component learning for object detection," *Proceedings of The European Conference on Computer Vision*, pp. 211–224, October 2008, Marseille, France. 2

[28] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, "On-line semi-supervised multiple-instance boosting," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1879–1879, June 2010, San Francisco, CA. 2

[29] T. Lee, "Image representation using 2D gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, October 1996. 3

[30] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern Analysis and Applications*, vol. 9, no. 2-3, pp. 273–292, August 2006. 3

[31] S. Shan, P. Yang, X. Chen, and W. Gao, "Adaboost Gabor Fisher classifier for face recognition," *Proceedings of International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 279–292, October 2005, Beijing, China. 3, 4

[32] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, April 2002. 3, 4

[33] 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 A*, vol. 2, no. 7, pp. 1160–1169, 1985. 3

[34] E. R. Davies, "Binary shape analysis," in *Machine Vision: Theory, Algorithms, Practicalities*, pp. 161–167. Morgan Kaufmann, San Francisco, CA, 2004. 5

[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006. 5

[36] D. M. W. Powers, "Evaluation: from precision, recall and F-factor to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, December 2011. 5