https://doi.org/10.2352/ISSN.2470-1173.2017.17.COIMG-435

In Situ Height and Width Estimation of Sorghum Plants from 2.5d Infrared Images

Tavor Baharav, Mohini Bariya, and Avideh Zakhor
University of California, Berkeley
{***tavorb, mohini, avz***} ***@berkeley.edu***

Abstract

Plant phenotyping, or the measurement of plant traits such as stem width and plant height, is a critical step in the development and evaluation of higher yield biofuel crops. Phenotyping allows biologists to quantitatively estimate the biomass of plant varieties and therefore their potential for biofuel production. Manual phenotyping is costly, time-consuming, and error-prone, requiring a person to walk through the fields measuring individual plants with a tape measure and notebook. In this work we describe an alternative system consisting of an autonomous robot equipped with two infrared cameras that travels through fields, collecting 2.5D image data of sorghum plants. We develop novel image processing based algorithms to estimate plant height and stem width from the image data. Our proposed method has the advantage of working in situ using images of plants from only one side. This allows phenotypic data to be collected non-destructively throughout the growing cycle, providing biologists with valuable information on crop growth patterns. Our approach first estimates plant heights and stem widths from individual frames. It then uses tracking algorithms to refine these estimates across frames and avoid double counting the same plant in multiple frames. The result is a histogram of stem widths and plant heights for each plot of a particular genetically engineered sorghum variety. In-field testing and comparison with human collected ground truth data demonstrates that our system achieves 13% *average absolute error for stem width estimation and* 15% *average absolute error for plant height estimation.*

1 Background

Increases in energy demand and rising concerns about climate change have led to a major thrust in the search for renewable energy sources, particularly better biofuels. Improved biofuel development involves planting genetically engineered crop varieties. These varieties are evaluated and compared through *phenotyping*-the measurement of plant traits such as height, leaf volume, and stem thickness. Phenotyping is an essential task as phenotypic data is used by geneticists to identify promising genetic strains of biofuel crops such as sorghum. However, when done manually, phenotyping is time-consuming and creates a serious bottleneck in the biofuel research and development workflow.

Manual phenotyping is done with basic equipment such as a tape measure. It requires a person to traverse fields and measure individual plants and, as such, it is laborious and error prone. Thus, in practice, it is unfeasible to measure every plant in a densely packed field, so a few individuals are chosen as representative of each genetic variety and measured. Automated phenotyping techniques promise to relieve this burdensome task. In fact, across phenotyping applications, software and algorithms are already releasing manual bottlenecks in plant biology research [12]. However, most automated phenotyping techniques have limited scope because they are designed to measure plant traits in restrictive laboratory conditions. Recent work by Amean *et. al.* uses a Frangi filter to segment stems and branches, but requires individual plants to be photographed against a white screen [2]. Other phenotyping techniques try to achieve full 3*D* model reconstruction of plants but require multiple images from many angles of a single plant [3, 4], making them unfeasible for most field conditions. Many height estimation techniques require expensive aerial systems for image capture [5, 6]. Other work by Heijden *et.al.* describes in situ phenotyping of pepper plants in a greenhouse setting with wide spacing between plants [1].

Figure 1: Data collection robot in a field. Note the density of the sorghum in the background.

In this work, we leverage autonomous robots mounted with cameras, as shown in Figure 1, along with image processing algorithms to speed up the phenotyping process. This allows larger

quantities of higher quality data to be collected, thereby accelerating and improving biofuel development. Specifically, we develop algorithms to estimate stem width and plant height for sorghum, a widely grown and promising biofuel source, from autonomously captured images. Stem width and plant height metrics are important for determining the biomass of new sorghum strains and therefore their effectiveness as biofuel sources. Our algorithms enable in situ height and width estimation in a typical field setting of rows of closely spaced plants using images captured from only one side of the plant, as shown in Figure 2. This enables non-invasive yet substantial data collection throughout the growing cycle. The images of sorghum used by our algorithms, shown in Figure 4 are collected by cameras mounted on an autonomous robot as shown in Figure 1. The images are captured as the robot travels through the rows in a field. The exact sensor configuration is described in Section 2.

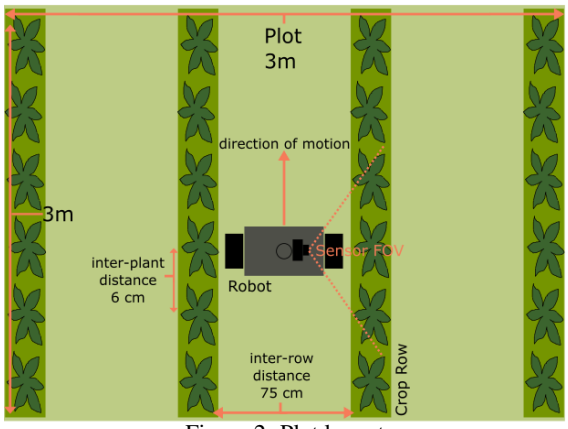


Figure 2: Plot layout

Our algorithms aim to produce useful *comparative* height and stem width statistics for sorghum. Specifically, they are useful for measurement comparisons across plant varieties, where each variety has numerous individual plants in the same plot that are imaged and processed by the algorithms. In practice, geneticists are interested in comparing the height and stem width of an entire plot of one plant variety with a plot of a different variety. Consequently, the fields of sorghum through which the robot travels are divided into plots with each plot containing only one genetic strain of sorghum. Our algorithms produce one histogram of stem widths and one histogram of stem heights per plot. Note that height and width measurements for the same individual plant are not associated by our algorithms, so there is no correspondence between the returned height and width histograms beyond their being from the same plot.

At a high level, our proposed stem-width measurement algorithm uses a Frangi filter and Hough Transform on infrared images to respectively enhance and locate stems [2], followed by a local search to determine stem edges. Our proposed height estimation algorithm uses wide-angle disparity and infrared intensity images from a stereo camera to segment plants. Camera to plant distances are combined with apriori knowledge of the camera height to geometrically compute plant heights. As the images used for stem width estimation and height estimation differ significantly, we present two different tracking algorithms: each suited to one of the image types.

We report the results of our algorithms on image data of planted sorghum collected by a human-driven robot in the field. We compare our estimates with manual measurements over several individual plants of different genetic varieties. The outline of the paper is as follows: In Section 2, we describe the sensor positioning. Section 3 covers the stem width estimation algorithm. Section 4 covers the stem height estimation algorithm. Finally, Section 5 describes results and future directions.

2 Sensor Positioning

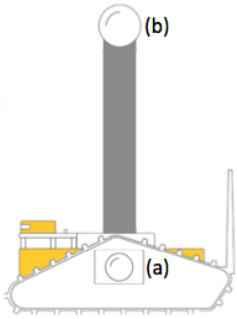


Figure 3: Side view of robot: (a) Infrared TOF width estimation camera, 10cm off the ground, and (b) Stereo height estimation camera, atop the mast

Our algorithms use a time of flight (TOF) camera that captures low resolution 2.5D infrared (IR) images, as seen in Figure 4a, and a stereo camera that captures low-resolution, wide field of view (FOV) images from which it computes a depth map, as shown in Figure 11. Both cameras are mounted on the moving robot, as shown in Figures 1 and 3. We position the TOF camera 10cm off the ground, and 30cm away from the row of plants, to look horizontally towards the base of the plants. Its images are used to isolate stems and measure stem widths. The 170^o FOV stereo camera is mounted on a retractable mast with horizontal line of sight to the top of the plants. Since sorghum can grow as tall as 3 meters, it is essential that the height of the stereo camera be adjustable. Though we use a camera with a large FOV that can theoretically capture the top of the plants during the entire growth cycle, in practice, sorghum develops a thick leaf canopy that occludes the plant tips making it necessary for the camera to always be mounted above the canopy. The camera returns two IR intensity images and a computed depth map, also referred to as a *disparity* image.

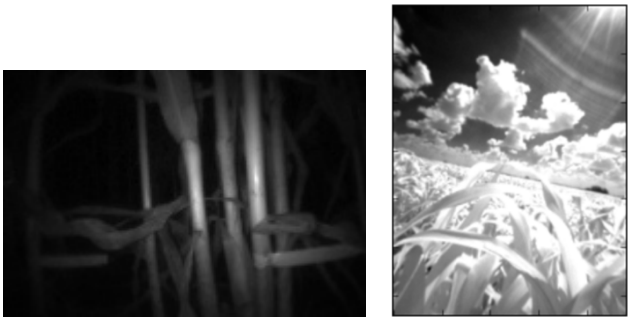

(a) (b)

Figure 4: Sample IR frames from the (a) TOF 2.5D depth camera and (b) stereo camera used

3 Width Estimation

This sections covers the width estimation algorithm that takes as input the 2.5D images captured by the infrared TOF camera mounted at the base of the robot. Our stem width measurement algorithm consists of 5 steps:

1. Pre-processing using Frangi filter [13] - A Frangi filter accentuates tubular and blob-like structures in the infrared image, highlighting the tubular stems, as shown in Figure 5b.
2. Hough transform for stem identification [2]- An iterative Hough transform is applied to the filtered image to identify straight line segments corresponding to stems, as shown in Figure 5c.
3. Geometric analysis of stem features - Once stems are located, the original IR image is processed to identify the width of the stem at every point along its length through edge detection, as shown in Figure 5d. The resulting array of width measurements for each point along the stem height is smoothed to remove outliers.
4. A single width measurement is extracted from the array of width measurements as follows: stem width measurements for the bottom half of the plant are considered, and outliers discarded. The average of the remaining smaller measurements is used as the final width estimate for the plant in that given frame.
5. Tracking stems through frames - To improve stem location, width estimation, and prevent double counting, individual stems are tracked across frames. Note that this step is not shown in Figure 5, as it is performed across frames.

In the remainder of this section, we review the above steps in detail.

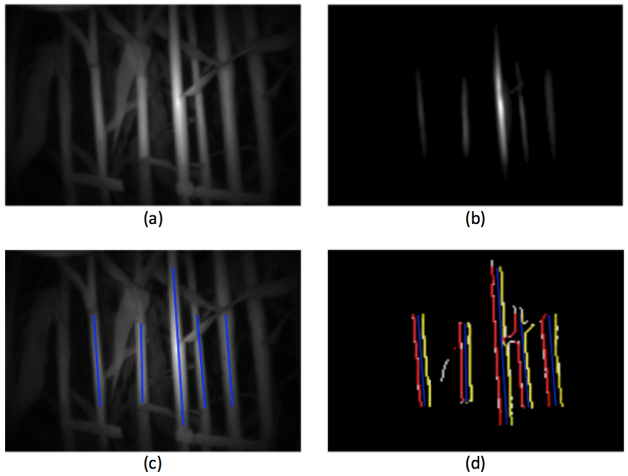


Figure 5: Width estimation flowchart. (a) Initial IR image, (b) Frangi Filter output, Section 3.1 (c) Hough Transform outputted lines, Section 3.2, (d) Edge and width processing output, Section 3.3

3.1 Frangi Filter

The raw infrared images of the sorghum bases are filled with stems as well as leaves and offshoots. To detect and measure the stems, it is necessary to transform the image domain to one in which the stems have increased prominence. Geometrically, stems are more tubular than other plant structures, so a vesselness filter, in this case a Frangi filter, is used to highlight them. This difference in visibility can be seen in the output of the Frangi Filter in Figure 5b, from the input in Figure 5a.

The Frangi Filter selectively accentuates tubular or blob-like structures in an image. To do so, it uses the eigenvalues of the image Hessian matrix at each point. The image Hessian is computed by smoothing the image and then taking its second derivatives, or gradients. An efficient way to compute the second derivatives is through convolution with the second derivative of a two dimensional Gaussian. Finally, the eigenvalues of the 2×2 gradient matrices at each pixel are computed. The ratio of the eigenvalues is used to generate a measure of the vesselness or "tubular-ness" at each pixel [13]. Recall that the two eigenvectors computed at each pixel indicate the directions of greatest local intensity change, while the eigenvalues indicate the magnitude of intensity change. If the ratio of the eigenvalues at a given pixel is approximately one, then the pixel is part of a blob-like, or circular, structure, as the intensity changes are uniform in both directions. If the ratio of the eigenvalues is far from one however, e.g. $\lambda_1 >> \lambda_2$, then the pixel is part of a tube-like structure. Stems are tube-like in the images, as they are narrow, roughly vertical stripes of brightness on a darker background.

Since we compute the image Hessian matrix after convolution with a Gaussian, choosing the σ, or width, of the Gaussian filter allows us to select the width of tubes or blobs to accentuate. For the first stem image captured in a sequence, we can only guess the width of the stems we are looking for, and a therefore a large range of σ values must be scanned over. In subsequent frames the average estimated stem width so far can be used to narrow our guess for the appropriate σ.

3.2 Hough Transform

In the Frangi filter output in Figure 5b, the stems are easily discernible by the human eye. The Frangi filtered image has bright, vertical bands at the original stem locations. A Hough Transform is used to detect these bands. The popular Hough Transform uses a voting procedure to find various shapes, especially lines. Each pixel votes on candidate shape sizes and locations it can belong too, and the final shape is chosen based on the highest votes. The line-detecting Hough Transform parameterizes the set of all possible lines in an image by their perpendicular distance from the upper left hand corner of the image, ρ, and their angle from horizontal, θ, as seen in Figure 6. This parameterization is equivalent to $\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$.

Once the image is converted into this paramter space, we utilize the prior knowledge that stems are approximately vertical. This allows us to eliminate all candidate lines that are more horizontal, for example all lines with θ outside of a small region around 0^o. In this work, we use $-15^o < \theta < 15^o$, but these bounds can be adjusted depending on the crop being measured and the terrain characteristics.

After filtering the Hough parameter space of more horizontal lines, the line with the most votes is declared to be a stem. However, each image contains multiple stems. Rather than continuing to search the Hough parameter space for additional stem lines, we revert back to the initial infrared image space. The declared stem is removed from the image by removing all pixels within an appropriate region around the estimated stem line. The

size of the region is chosen based on the estimated stem width. Once the stem is removed, the Hough procedure is repeated to find the next stem. The procedure is repeated a preset number of times corresponding to the maximum number of stems expected within a frame. This limit can also be chosen adaptively, based on the number of stems observed in the previous frame. The Hough transform iterations stop before the limit if the remaining lines in the image are too faint. At the end of the iterative Hough processing step, the locations of the stems in the frame are known, and each stem is described by a line segment running along it, as seen in Figure 5c.

3.3 Edge and Width Processing

The next step is to extract the widths of the located stem by finding the edges of the detected stems in the original infrared image. An edge detection filter is used to detect the stem edges. Then, for each stem, the algorithm iterates vertically along the Hough detected stem line and finds the width at every point. The width at a given point along the stem line is the minimum horizontal distance until an edge is encountered on either side of the line. This edge detection process can be seen in Figure 5d, with the blue line denoting the found stem, the red dots the detected left side of the stem, and the yellow dots the detected right side of the stem. This produces a noisy array of measurements of the stem's width as a function of position along the stem line, or equivalently the stem height. We refer to this unfiltered, noisy array of width measurements as $W_i(y)$, the initial measured width of the stem at height y. This process is visualized in Figure 7, where we start with a detected stem and its estimated edges in Figure 7a and measure the width of the plant at each point along its stem yielding $W_i(y)$ in Figure 7b.

To produce a less noisy, lower variance measure of the width of the stem as a function of height, the measurements are smoothed by taking a sliding window of length K. At each point, the K measurements in the sliding window are sorted by value, and the smallest $\frac{1}{8}$ and largest $\frac{5}{8}$ of measurements are discarded. The motivation for these thresholds is to discard excessively small outliers, while focusing on the smaller measurements in general, as we wish to measure the width of the actual stem, and not the width of the stem plus a leaf branching off from it. We choose K, the number of points to aggregate, to be proportional to the stem width, to ensure the stem is not occluded by leaves which cause overestimates of stem width. We refer to this processed, array of

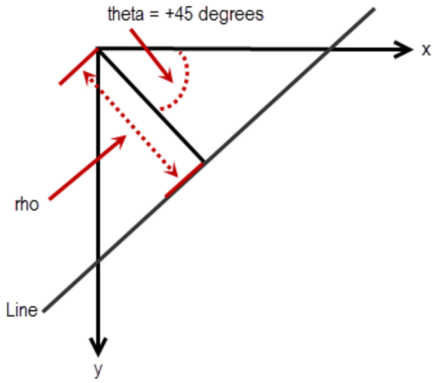


Figure 6: Line finding Hough Transform parameterization

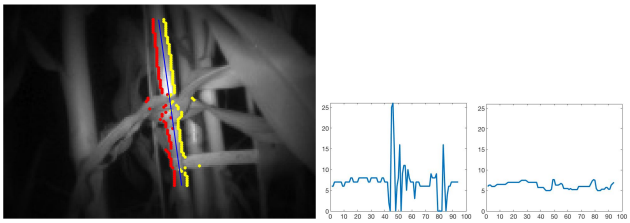

(a) (b) (c)

Figure 7: (a) Initial Image with edges detected, (b) $W_i(y)$ unadjusted widths, (c) $W_p(y)$ adjusted widths. X axis represents distance from top of plant, Y axis the width in pixels

width measurements as $W_p(y)$, the estimated width of the stem at height y. An example of this processing can be seen above, transforming $W_i(y)$ shown in Figure 7b to $W_p(y)$ in Figure 7c.

The final width value assigned to a stem in a given frame is computed from the smoothed and filtered measurement array $W_p(y)$. In $W_p(y)$, we consider the measurements corresponding to the spatial bottom half of the plant. Within this subset, we discard outliers while maintaining focus on the smallest measurements by discarding the lowest $\frac{1}{4}$ and highest $\frac{1}{2}$ of measurements value-wise. We compute the average of these measurements, and use this outputted value as our estimated width for the stem in the current frame.

3.4 Confidence

In addition to computing a width measurement for each stem, we compute a confidence measure that indicates our certainty that the detected stem actually corresponds to a real stem rather than a leaf or some other plant feature, as well as our confidence in the resulting width estimate. We generate the confidence measure by combining several different metrics:

1. *Maximum number of consecutive errored width measurements in* $W_i(y)$. Recall that $W_i(y)$ is the unfiltered array of width measurements found along the stem line as described in Section 3.3. An attempt to find the width measurement at a given point along the stem line is said to have errored if an edge can not be found within $1.5 \times$ (last stem width in pixels) on either side. If a long sequence of consecutive measurements have errored, the stem line detected with the Hough transform is not likely to lie along the actual stem or has passed outside the stem bounds. We make this feature's contribution to the confidence measure proportional to the square of the maximum number of errored measurements to achieve this penalization.
2. *Total number of errored width measurements in* $W_i(y)$. This feature captures the fact that a stem's width estimate is suspect if a large number of the stem's width measurements have errored.
3. *Number of errored width measurements in* $W_i(y)$*the bottom half of the initial measurements array.* Width measurements for the lower portion of the stem are used to compute the final width estimate, as we are interested in the width of the base of the sorghum plant. Sorghum is traditionally modeled as having a stem resembling a truncated cone, and thus it is desirable to measure the stem width close to the ground. Therefore, errored measurements for this region of the stem are considered more problematic and hence given additional

weight when incorporated into the confidence measure.

4. *Integral of derivative of $W_i(y)$, initial width measurements array, squared.* If the stem line found during the Hough transform step truly lies along a stem, even the initial unsmoothed measurement array $W_i(y)$ should be fairly smooth. The values in $W_i(y)$ should vary gradually, as actual stems do not have rapidly varying widths. If the stem line does not lie along an actual stem, or the stem edge in the image is weak and noisy the width estimates along the stem line have high local variance. Such measurements are likely inaccurate estimates of stem width. Thus, we use as a feature $\sum_y (\frac{d}{dy} W_i(y))^2$. This feature aims to detect high local variance. A measurement array with many jumps in width values has derivative values with large magnitudes. The feature sums the derivative values so more jumps across the measurement array are reflected in this measure.
5. *Integral of derivative of bottom half of $W_i(y)$.* As mentioned before, the quality of width measurements for the bottom half of the stem are particularly important. Therefore, the previous confidence feature applied to the bottom half of the stem is given additional weight in the final confidence measure.
6. *Number of votes in Hough Parameter Space* The final feature used to quantify our confidence in the stem is the number of votes the line corresponding to the estimated location of the stem received in the Hough parameter space, or how many pixels in the Frangi - filtered image this line accommodates. This is to say, we are more confident in stems that pass through regions we identify as more vessel-like. In order to make this feature similar to the others (higher value indicates lower confidence), we take the inverse of the intensity in the Hough parameter space.

To compute an overall confidence value for a stem width measurement in a given frame, we take a linear combination of these features. We adjusted the coefficients to somewhat standardize the impact of each of these features, yet give the more important features i.e., 1,3, and 5, a higher weight. This confidence measure means that we do not trust measurements with a high outputted value from this linear combination.

3.5 Tracking

So far, we have discussed how we estimate the width of a stem from a single frame. However, to avoid double-counting of stems per plot and to refine width estimates for individual stems, it is necessary to track and correspond stems through frames. We use a Hidden Markov Model of the system as shown in Figure 8, in which the estimated locations of the stems in each frame are noisy readings of the true positions of stems moving with some variable velocity. More formally, the hidden states f_i are the true positions of the robot and the location and width of all the stems in frame i, and the observed states d_i are the captured infrared images, derived approximate velocity, and estimated stem locations frame i.

To estimate the velocity for a frame, we estimate the translation between the two images by detecting and corresponding features between adjacent frames. If we fail to find enough features to estimate the translation, such as in the case of occlusion, we use a momentum based model, updating our previous velocity

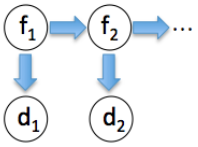
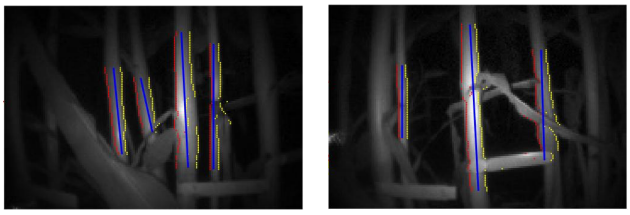


Figure 8: Underlying Hidden Markov Model, f_i are the current visible stem locations and widths, d_i the estimated stem locations and widths, and the estimated robot velocity

by factoring in the computed translation of stems between adjacent frames. This entails examining all stems found in the current frame, and computing the horizontal distance their midpoint traveled from the previous frame.

In order to track the stems, we maintain a running list of stems visible in the current frame, removing stems from the list when they reach the edge of the image frame or exhibit low-confidence width measurements in many consecutive frames. The latter case suggests that the stem is a false detection, or occluded for too long to continue tracking. When a new frame is processed, all candidate stems in the frame are found using the technique described in Sections 3.1 and 3.2. The candidate stems are greedily matched with the stems in the running list from the previous frame based on a similarity function. Our similarity function compares the midpoint of a stem in the current frame with the projected midpoint of the stem from the previous frame based on the estimated robot velocity, as well as the difference between the two stems' angles.



(a) (b)

Figure 9: Reasons for using tracking: (a) erroneous stem edge detection, (b) accurate width markings / estimates

Using this confidence metric, we can filter measurements over numerous frames to obtain a lower variance, higher accuracy measurement. While in general, perfectly head on measurement as in Figure 9b yield an accurate estimate, even with this confidence measure, it is sometimes difficult to detect poor measurements, as in Figure 9a, where the measurements are smooth with no errors, except for two discontinuities where the detected right edge is found in the middle of the stem. This difficulty in determining whether a stem is properly measured is alleviated by tracking, as with estimates over multiple frames, we can better determine the outliers / errored measurements such as Figure 9a, as we will have more measurements such as in Figure 9b to compare it to. Thus, while tracking does somewhat improve accuracy, its main function is to lower the variance of width estimation by filtering, as described in Section 3.3, over more frames' estimated widths.

3.6 System Tradeoffs

We use a low-resolution, 176×120 pixel infrared camera, as isolating stems in RGB color images is difficult and unreliable due to shadows and a similarly colored background. The 2.5D depth data is essential to convert from measurements in pixels in the image, to measurements in centimeters. Knowledge of the distance to the stem, and the cameras cm/pixel ratio at a given distance makes this computation possible.

To make this phenotyping methodology accessible to farmers and biologists, we opted for a less expensive but lower resolution 2.5D depth camera. To overcome low resolution constraints, we correspond many measurements of one plant to obtain an accurate final width estimate. Our TOF camera has a field of view and spatial resolution such that for a point $x\ cm$ from the camera, one pixel in the image corresponds to a $\frac{x}{100}cm \times \frac{x}{100}cm$ square in the 3D world. Thus, when the camera is $30cm$ away from the plant it has a resolution of $0.3cm$ / pixel, which means that $1.8cm$ wide stems are approximately 6 pixels wide in the image. Consequently, an error of 1 pixel translates to 15% error. As such, many width measurements of the same plant are necessary to average out errors and achieve sub pixel accuracy. More specifically, this averaging for sub pixel accuracy is achieved through two separate steps described below:

1. For a given plant in a given frame, many width measurements are found along the length of the plant stem. These measurements are processed into one final width estimate for the plant in the given frame, as explained in Section 3.3. As the camera's resolution is 176×120 pixels, this step averages over $\sim 20-30$ values.
2. To offset erroneous measurements and partial occlusion, stems are tracked through all the frames in which they appear. Since each stem appears in around 150-200 frames with the robot moving at approximately 0.07 m/s, and the camera capturing footage at $45fps$, this leads to averaging over several width estimates for a single plant.

4 Height Estimation

This sections covers the height estimation algorithms that take as input images captured by the disparity camera mounted on the mast of the autonomous robot. In section 4.1 we begin by describing the method of image-based height estimation for a row of closely spaced plants. In 4.2, we describe techniques to deal with particular image regions and artifacts that can lead to errors in height estimates. Further, in section 4.4, we describe a tracking algorithm that can be used to refine height estimates across frames while avoiding double counting. Tracking is especially important for height estimation in a field setting, because occlusion, lighting variation, wind, and other field conditions lead to a nonzero variance in the height estimates of a single plant across multiple frames. By averaging estimates over multiple frames, we aim reduce the variance and more accurately estimate the true plant height.

4.1 System Overview

Visualize a tall crop row of sorghum or similar looking corn in a field. One way to estimate the plant heights is to draw a continuous contour line along the top edge of the crop row. We call this line a top contour. The basic principle of our height estimation algorithm is to find the top contour from the sensor images, as shown in Figure 10. We then generate a set of height estimates from the "continuous" top contour using two alternative sampling methods. The first sampling method, suitable for young sorghum, involves segmenting individual plants in the captured frames. It is described in Section 4.3.1. The second sampling method is used later in the growth cycle when stems and leaves make it impossible to segment individual plants. We refer to it as a "blind" sampling technique and describe it in Section 4.3.2. Regardless of the sampling method used, the top contour and plant heights are always extracted from each frame independently as individual frames are captured. Afterwards, a tracking algorithm combines the height estimates of the same plant from different frames to refine the estimates for each plant and avoid double-counting. An

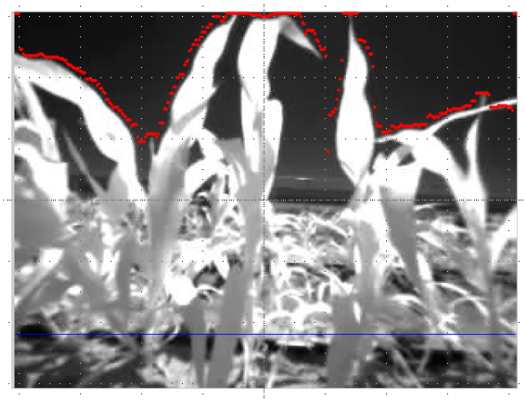


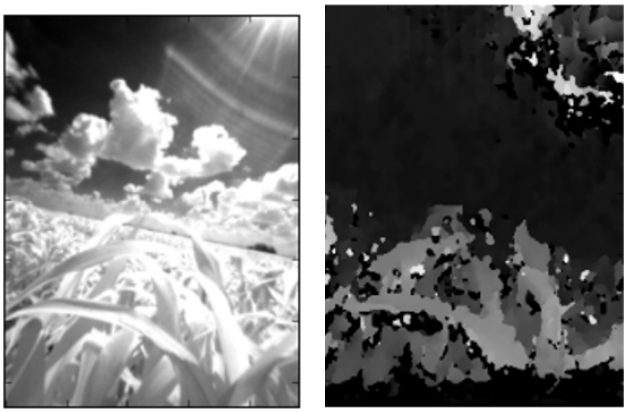
Figure 10: Intensity image with detected top contour in red. The line of the ground is shown in blue.

assumption of the top contour detection method is that the top contour of the desired crop row is visible in the captured camera frames. This requires the camera to be mounted such that there is always line of sight to the tops of the sorghum plants. Sorghum strains can grow very tall and also have high variability in height, making it difficult to ensure line of sight and keep *all* plants within the FOV. Even though we use a camera with a large FOV to counter this, the leafy canopy of sorghum plants and their height variation can still make this requirement difficult to meet at all times.

4.2 Top Contour Detection

The height estimation algorithm uses two input images with the same dimensions from the stereo sensor: the computed disparity image, and one of the two IR intensity images as shown in Figure 4b. The main step in top contour detection is to segment the disparity image using k-means, with k=2. Recall that a pixels intensity value in the disparity image is proportional to the distance from the camera of the real-world object captured at that pixel. Running k-means on the disparity image clusters the pixels by depth. For k=2, the resulting clusters can be interpreted as a foreground cluster and a background cluster. This is shown in Figure 12a which is the result of k-means clustering on the disparity image in Figure 11b. Essentially, k-means binarizes the disparity image, segmenting the region containing the front-most row of plants from the sky and background. The advantage of using k-means is that a hard threshold for segmentation is avoided. The desired top contour is essentially the top edge of the foreground

cluster. However, the disparity images computed by the stereo sensor tend to have high-error in the sky and sun region, particularly if the sun itself is visible in the frame. This is due to high IR intensities from direct sunlight and erroneous matching during the depth computation. Therefore, portions of the sky are mistakenly computed as being close to the camera, when they should be maximally distant.



(a) (b)

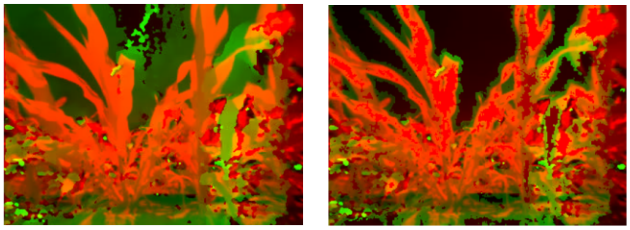
Figure 11: Corresponding (a) intensity and (b) disparity images captured by stereo camera show error region due to sun.

Considering the intensity image in Figure 11a and the disparity image in Figure 11b, we can see that the front-most plants appear as lighter regions in the disparity image because they are correctly computed as being closer to the camera. However, we also see a light region in the top right corner of the disparity image corresponding to the sun in the intensity image. In cases such as this, after running k-means, portions of the sky and sun are erroneously lumped in the foreground cluster. These portions must be removed before the top contour is extracted. To remove regions of the sky that are mistakenly classified as part of plants, we use the visual smoothness of the sky region. To deal with sun regions that are mistakenly classified as plants in the captured images, we use a connected components based method.

4.2.1 Sky Filtration To address the problem of sky pixels erroneously classified as foreground during k-means segmentation of the disparity image, the intensity image is used to create a *sky-filter*-a mask covering the sky region in the images-that removes these sky pixels. The sky is more easily distinguishable in the intensity image because it has less intensity variation and fewer small structures than the plants. The sky is also darker, but as the plants contain shadows it is best to avoid segmenting the sky based purely on pixel intensity. Rather, segmentation uses the eigenvalues of the image Hessian, or the image structure matrix [9]. Recall that the image structure matrix for a single pixel is given by:

$$H = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial I}{\partial y^2} \end{bmatrix}$$

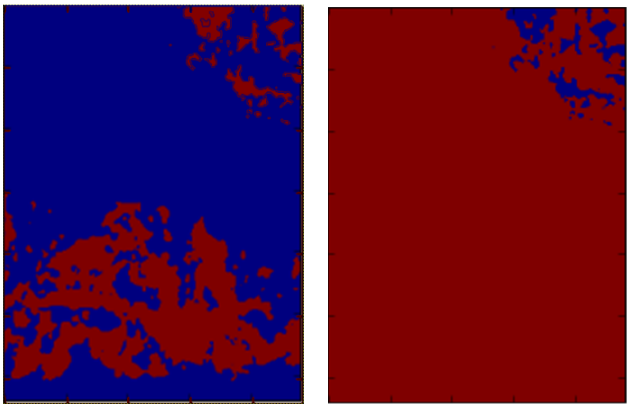
To determine the local behavior of an image, we average H element-wise over several neighboring pixels. The dominant eigenvector of H, denoted by $\vec{v_1}$, gives the dominant direction of local intensity change in the image. Near an edge, $\vec{v_1}$ will point towards the edge. The dominant eigenvalue , denoted by λ_1, corresponds to the magnitude of the local intensity change, and is large if there is an edge present. In a "smooth" region, the dominant eigenvalues are small. By thresholding on the magnitude of the dominant eigenvalue of the structure matrix H of the intensity image, we generate the sky-filter. Figure 12 shows an overlay of the disparity and intensity images that demonstrates the impact of applying the sky-filter. Brighter green corresponds to lower computed depth. We can see that erroneous green regions in the sky in Figure 12a are removed by the sky-filter in Figure 12b.



(a) (b)

Figure 12: Overlay of intensity in red and disparity in green (a) before and (b) after sky filter application.

4.2.2 Sun Removal When the disk of the sun is visible in the intensity and disparity images, it is erroneously classified as "foreground". The sun is not removed by the sky filtration process because it has ray and ringing structures that lead to larger Hessian eigenvalues. A separate connected-components analysis is used to remove the sun. We begin sun removal by re-using the binary image produced by running k-means on the disparity image.



(a) (b)

Figure 13: Steps of sun removal through connected components. (a)Binary image produced through k-means with k=2 on disparity image. (b) Connected components classifies blue pixels as being in sun region

For the disparity image in Figure 11, this produces the binary image shown in Figure 13a where blue corresponds to pixels classified as background, and red corresponds to pixels classified as front-row plants. The red region in the top right

corner of the image marks sun pixels erroneously grouped with front row plants. We additionally run k-means with k=2 on the intensity image to segment it into a set of bright and dark pixels. Since the sun is both bright and erroneously classified as foreground, connected components is run on the foreground pixels from the disparity image and the bright pixels from the intensity image [10],[8]. As the sun is usually at the top of the image, connected components which are contained in the top half of the images are classified as being part of the sun. Running connected components on the binarized image from Figure 13a and using the described classification requirement on the resulting components produces the image in Figure 13b. Here the blue pixels are part of connected components that were classified as being part of the sun. Such components determined from the disparity and intensity images are combined to create a mask that removes the sun region.

Thus, we take the top edge of the processed set of "foreground" pixels in the disparity image as an accurate top contour. The top contour is essentially a continuous line in the image, but we desire a set of discrete height measurements for each plant. Therefore, we sample the top contour to determine the plant heights.

4.3 Top Contour Sampling

We propose two approaches to sampling the top contour to produce a set of discrete plant heights suited to different periods of sorghum growth. In section 4.3.1 we describe an algorithm for the early part of the growth cycle, when the full plant height is visible in the camera's FOV. This algorithm infers the individual plants' structures from the disparity image and generates a bounding box for each plant. This bounding box can then be used to sample the top contour and assign a height estimate for each plant. During the later part of the growth cycle, the height and leaf density makes it impossible to exactly segment individual plants in images. A "blind" sampling technique for this situation is described in Section 4.3.2.

4.3.1 Top Contour Sampling for Early Growth The early growth algorithm begins by iterating over every point in the extracted top contour. For each point, it considers a set of line segments starting from this point and reaching the bottom of the image. Each line has a different angle from vertical, ranging from $-\psi$ to ψ. The standard deviation of the pixel intensities along each line is computed and the line with the minimum standard deviation is selected as the "best" line for that point on the top contour. Figure 14 illustrates how the best line for one top contour point is found. We can see that the chosen best line for the given top contour point runs along the stem the most of all candidate lines. Figure 15b depicts the results of this step on the example image shown in 15a. Each blue line is the chosen "best" line for one point in the top contour. Together, all the blue lines compose the set of "best fit lines" for the disparity image. Intuitively this method captures the fact that lines running along a plant have less deviation in depth than lines that run over multiple plants and across gaps between plant stems and leaves.

In Figure 15b, we can see that the best fit lines roughly reconstruct the individual plant structures because the lines tend to angle such that they clump into plants. In a single plant, as one moves left to right across the image, the angle from vertical of the chosen lines transitions from positive to negative where positive angle is clockwise. Therefore, the angles of the best fit lines can be used to segment the individual plants. Figure 15c is a plot of best fit line angles. Individual plants can be segmented by finding the large discontinuities from negative to positive in the sequence of best fit line angles. To locate the discontinuities, we apply a median filter to the angle data, take the difference of consecutive data points, and then find the locations of the peaks. The positions of the peaks result in the columns in the image that separate individual plants. To further improve results, it is useful to set a limit on the maximum number of peaks and the minimum distance between peaks. In the case of our sorghum data, this is possible because the sorghum seeds are planted by machine, and tend to have relatively consistent spacing. Figure 15d displays an example segmentation result. Note that once we have segmented the plants, it is easy to assign a height estimate to each one by sampling the top contour once within each plant bounding box. We assign each plant the height of the highest top contour point within its bounding box.

4.3.2 Top Contour Sampling for Later Growth During the later part of the sorghum growth cycle, the height of the plants and the density of leaf growth makes individual plant segmentation impossible. Instead, the top contour is found as before, but the height is estimated "blindly" without exactly segmenting individual plants. This is done by regularly sampling the top contour such that the distance between samples is the estimated inter-plant distance as shown in Figure 23. This is useful for estimating the plant height statistics in a single plot.

4.4 Tracking

Due to high camera frame rate of $30 fps$, a single plant appears in multiple frames. Tracking allows us to leverage this multiplicity to refine height estimates across multiple captures. It is also necessary to reduce skewing of the returned histogram of height estimates due to double counting. For height estimation, our tracking algorithm consists of 2D image registration. This

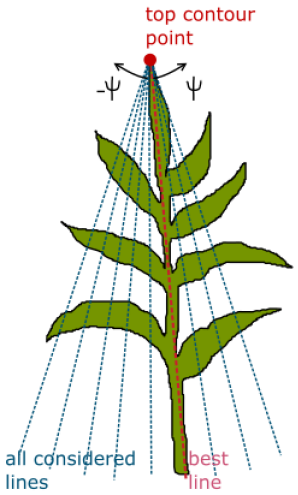


Figure 14: Schematic of best fit line for individual plant segmentation

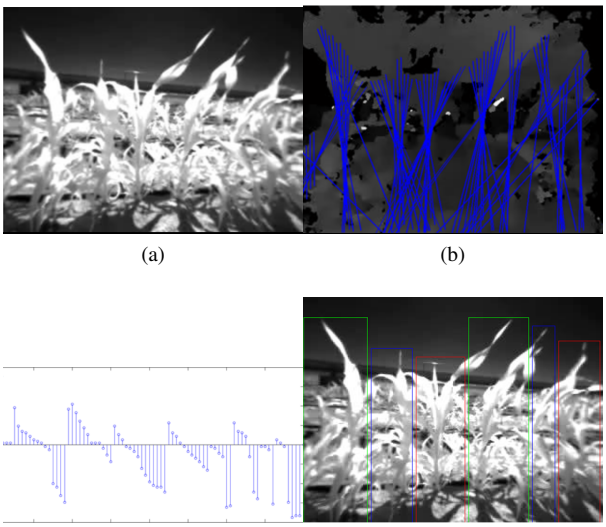


Figure 15: Segmentation steps: (a) Original Intensity Image; (b) Disparity Image with overlaid best fit lines; (c) Angles of best fit lines; (d) Segmentation result showing determined plant bounding boxes overlaid on intensity image

method finds the peak in the cross-correlation of two images to determine the shift between them [11].

In the case of blind segmentation used during the later growth period, two consecutive frames are registered, and the resulting registration shift is used to determine which plants have been seen before and which are newly encountered so that the heights can be updated appropriately. This step is simple because in blind segmentation, heights are extracted by sampling the top contour at points with a fixed period. Heights measured in the current frame can be crudely but simply corresponded to heights measured in the previous frame by shifting the sampling points in the current frame by the registration shift between the current and previous frame. For the case of individual plant segmentation used on early growth sorghum, the registration step is more difficult because segmented plants must be matched between frames. This matching is made complex by the fact that exact segmentation results may differ between frames, as a plant could be erroneously detected or erroneously undetected.

Consider two consecutive frames f_1 and f_2 in which the plants have been individually segmented, as shown in the schematic of Figure 16. Assume that frame f_2 is captured immediately after f_1. Suppose there are n plants detected in f_1 and m plants detected in f_2, where $n = 4$ and $m = 5$ in the schematic. The bounding boxes of the plants in frames f_1 are denoted by $b_1^{(1)}, b_2^{(1)}, ..., b_n^{(1)}$ where plant i is bounded between pixels $b_i^{(1)}$ and $b_{i+1}^{(1)}$ along the image x-axis, which is shown to have a length of 120 in the schematic of Figure 16. Similarly, plant boundaries in frame f_2 are denoted $b_1^{(2)}, b_2^{(2)}, ..., b_m^{(2)}$. For example, in the schematic, $b_1^{(1)} = 0$ and $b_2^{(1)} = 30$, so plant 1 in frame f_1 is bounded between pixels 0 and 30. Image registration is used to find the shift, s_x, between the two frames f_1 and f_2. Then, given the bounding boxes of the segmented plants in f_1 we compute the registration-shifted bounding boxes of f_1 plants as:

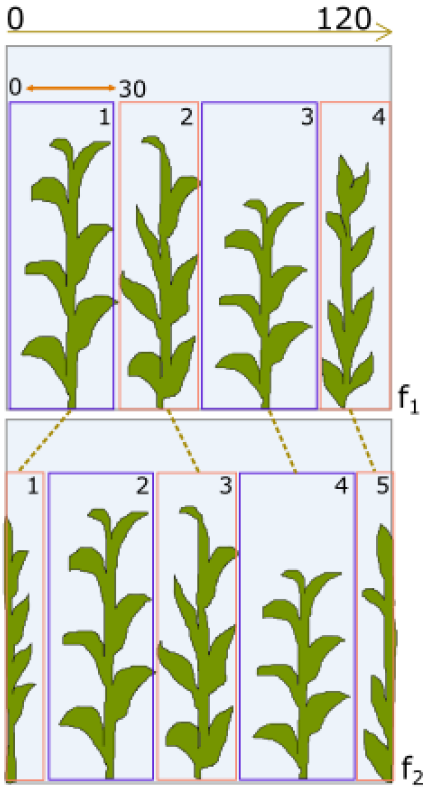


Figure 16: Schematic of matching between plants in consecutive frames f_1 and f_2 for tracking. Plant bounding boxes are shown in blue & yellow. Note the erroneous matching of first plant.

$$b_1^{(1)'}, b_2^{(1)'}, ..., b_n^{(1)'} = b_1^{(2)} + s_x, b_2^{(2)} + s_x, ..., b_m^{(2)} + s_x$$

These are the boundaries of plants in frame f_1 shifted by registration shift s_x which can be viewed as the "estimated" locations of the plants segmented in frame f_1 in frame f_2. Let (I,J) represent the set of pairs describing the matching between the plants in f_1 and f_2. The number of pairs in (I,J) is $min(m,n)$. In the schematic, the set of pairs (I,J) is $[(1,1),(2,3),(3,4),(4,5)]$. The element $(1,1)$ in this set, for example, means that plant 1 in frame f_1 is matched to plant 1 in frame f_2. The cost of a particular pair (i,j) within the set of pairs (I,J) is defined as the horizontal overlap between the registration-shifted bounding box of plant i from frame f_1 with the bounding box of plant j in frame f_2.

This cost can be written as:

$$c_{i,j} = overlap((b_i^{(1)'}, b_{(i+1)}^{(1)'}), (b_j^{(2)}, b_{(j+1)}^{(2)}))$$

Suppose, for simplicity of illustration, that for the frames f_1 and f_2 in the schematic of Figure 16, the shift between the frames is erroneously computed as 0. The cost of the pairing $(1,1)$, denoted by $c_{1,1}$ can be visualized as in Figure 17. The shaded yellow region is the overlap between the bounding boxes of the two plants. The total cost of the full set of pairings (I,J) is the sum of the costs of each member pair:

$$cost[(I,J)] \triangleq \sum_{(i,j)\in(I,J)} c_{i,j}$$

The chosen pairing set, $(I,J)^*$ is the one with maximum cost from all possible pairing sets.

$$(I,J)^* = argmax_{(I,J)} \sum_{(i,j)\in(I,J)} c_{i,j}$$

The chosen pairing set is used to determine which new plants have entered the camera FOV between frames f_1 and f_2 and which plants in f_2 were already encountered in f_1. This information is then used to update plant heights appropriately.

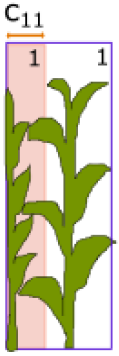


Figure 17: Overlap between matched plants for tracking

4.5 Actual Height Computation

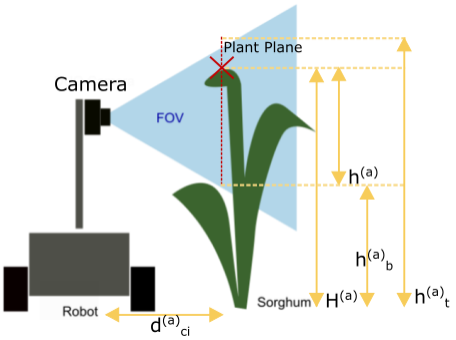


Figure 18: Geometry of real height computation.

Finally, the top contour pixel positions in the image must be assigned a height above the ground in meters. This is a geometric computation that uses the camera's FOV and height above the ground, the image dimensions, and the plant distances from the image plane to determine the metric height of each image pixel. In the following elaboration, quantities labeled with a superscript i are measured in pixels, and correspond to distances in the captured image. Quantities labeled with a superscript a are measured in meters and correspond to distances in the actual world. We assume that the camera height is constant and the camera is angled such that its optical axis is perpendicular to the ground, which implies that the ground is also flat. The schematic of Figure 18 illustrates the geometry of this computation for a top contour point marked with a red 'x'. The dotted red line is the plant plane. This plane is perpendicular to the camera's optical axis and contains the top-contour point. The distance from the camera lens to the plant plane, labeled $d_{ci}^{(a)}$, is important for converting the height from pixels to meters of the point labeled 'x'. To make this conversion, we assume that all objects captured in the image lie on the same plane, namely the plant plane. With this assumption, without going into details, it is clear that we can use the value of $d_{ci}^{(a)}$, the camera height, and the angle of the FOV, to compute the height in meters of the bottom-most pixel in the image, and the top-most pixel in the image, labeled $h_b^{(a)}$ and $h_t^{(a)}$ respectively. We denote the height in pixels of the top contour point 'x' from the bottom of the image as $h^{(i)}$ and the full height of the image in pixels as $y^{(i)}$. We can compute $h^{(a)}$, the height of the top-contour point in meters above the height of the bottom-most pixel in meters as:

$$h^{(a)} = h^{(i)} \times \frac{h_t^{(a)} - h_b^{(a)}}{y^{(i)}}$$

Here, we are essentially assigning each image pixel a height in meters, by dividing the full height of the image in meters, $h_t^{(a)} - h_b^{(a)}$, by the number of pixels $y^{(i)}$. $d_{ci}^{(a)}$ can be kept constant, based on the inter-row distance of the field, or it can be found from the stereo camera's depth computation. Then the full height of the plant in meters denoted $H^{(a)}$ is computed as:

$$H^{(a)} = h^{(a)} + h_b^{(a)}$$

5 Results

We report the results of our height and width estimation techniques on planted sorghum. The data for the reported experiments was gathered by a human controlled robot in fields of sorghum, with rows spaced 0.75m apart, and 3m long ranges with 50 plants per plot range. The robot was driving at approximately 0.07 m/s, yielding around 2000 frames at 45 fps. 10 plants were tagged and ground truthed per range. In the following, we report average percent error and average absolute percent error for height and stem width estimates. We define these metrics as follows. Suppose the estimated value of either the height or stem width estimate for plant i is e_i and the real value is r_i. For N plants, the average percent error E is computed as:

$$E = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{(e_i - r_i)}{r_i}$$

The average absolute percent error A is computed as:

$$A = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{|(e_i - r_i)|}{r_i}$$

5.1 Width Estimation

To test our width estimation algorithm, we tracked ground truthed plants through 16 frames, and computed the width based on the measurements over these 16 frames. We did this instead of looking over measurements over the range, as this made it easier to visually correspond which stem

Figure 19: Sorghum stem: note widths above the node, and at center of internode

we measured with the ground truthed stem. Note that ground truth measurements were taken a centimeter or two above the stem nodes rather than at the stem's narrowest point, while our algorithm estimates the narrowest width of the stem, technically referred to as the "middle of the internode". This difference can be seen in Figure 19. Manually examining the collected frames, we observed that the difference between the width measurements at the ground truthed stem position and the stem's narrowest point was consistently close to one pixel, yielding an an average error of -16%, as seen in Table 1. We thus adjusted our outputted stem widths by one pixel to compensate for this discrepancy, yielding results in better agreement with the ground truth, as can be seen in Table 2.

Range	Average % Error	Average Absolute % Error
Range 1	-8	15
Range 2	-15	17
Range 3	-24	24
Range 4	-16	16
Range 5	-12	25
Range 6	-16	17
Range 7	-20	20
Average	-16	19

Table 1: Unadjusted Width Estimation Results

Range	Average % Error	Average Absolute % Error
Range 1	5	13
Range 2	1	11
Range 3	-12	14
Range 4	-3	6
Range 5	5	24
Range 6	-3	10
Range 7	-6	10
Average	-1.8	13

Table 2: Adjusted Width Estimation Results

Figure 20 is a plot of the relative width estimate error as a function of ground truth stem width for 70 plants. Clearly, the relative error of the estimates tends to decrease as the actual stem width increases. This can partially be attributed to the fact that we manually added a fixed one pixel to each width estimate to compensate for the difference between the ground truth measurement position and the width estimation position along the plant stem. While 1 pixel seems to be an adequate compensation for our purpose, it is intuitively obvious why this constant compensation would lead to the downward error trend of Figure 20. The difference in stem width between the estimation position at the middle of the internode and the ground truth measurement position just above the node should not be constant but should scale with the width of the plant.

Recall that our algorithm tracks a single plant across multiple frames and averages over the width estimates in each frame to generate a final estimate for the plant. Figure 21 shows the width estimate in each frame for a single plant stem over the 192 consecutive frames it was detected in. The black line marks the final width estimate for the particular stem, while the red lines

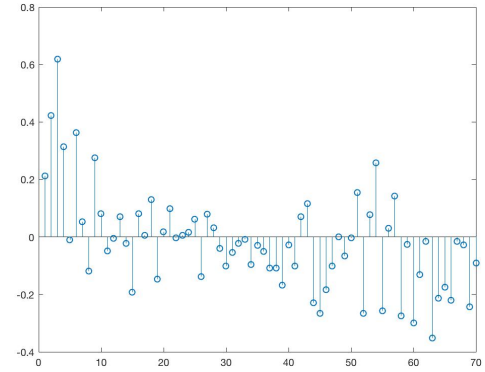


Figure 20: Percent error sorted by increasing width

mark $\pm 15\%$ error boundaries around the final estimate. We can see that, overall, the estimates over the frames are closely clustered, with only one region of high deviation, corresponding to frames in which the stem was partially occluded. Such highly deviant estimates are discarded in the computation of the final stem width, as our algorithm removes the largest and smallest 10% of estimates over the frames and averages over the remaining ones.

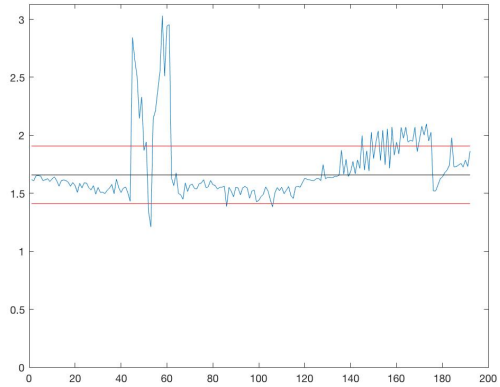


Figure 21: Stem Width over frames

In addition to individual plant width estimates, we also generated a histogram of the plant widths over the entire plot row. Out of the 204 stems our algorithm detected in the row, we discarded all those that appeared in fewer than 50 frames as false detections, leaving 75 stems remaining. For each of these stems, we ignored outlier width measurements, then averaged the remaining ones. The results can be seen below in Figure 22. While we do not have a ground truth histogram, the 10 ground truth measurements for that plot averaged to 1.83cm, and our histogram has a mean of 1.65cm, yielding -9.8% error.

5.2 Height Estimation

To test our height estimation algorithm, we ran it on still images of sorghum with one image captured per plant and videos captured while moving past rows of sorghum. The height estimates were compared to ground truth values. In the case of video, each video captures a full row of plants. In the case of still frames, images were captured of a few plants per row. The number of plants is noted in Table 3. The error was computed between the average estimated height of the full row, and the average of the ground truth measurements taken in that row. Knowing the camera mounting height is vital to our algorithm, so the reported re-

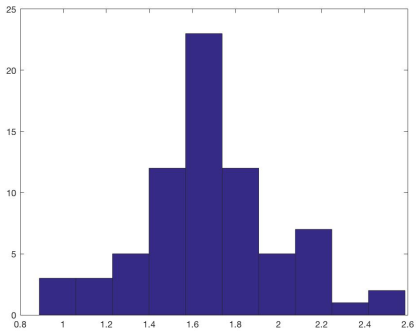


Figure 22: Histogram of stem width estimates in cm over one row

sults are only for cases where we are certain about the camera height during image collection. The results are rounded as we were not interested in tenths of a percent error.

Range	# Plants	Avg % Error	Avg Absolute % Error
Range 1	7	9	22
Range 2	8	2	24
Range 3	7	15	19
Range 4	8	-4	30
Range 5	5	-6	12
Range 6	8	-12	22
Range 7	5	-5	20
Range 8	6	16	25
Range 9	9	11	30
Average	63 (total)	2.9	22.7

Table 3: Results on Still Frames

Range	Average % Error	Average Absolute % Error
Range 1	17	17
Range 2	-15	21
Range 3	-6	12
Range 4	-8	14
Range 5	-7	13
Average	-3.8	15.4

Table 4: Results on Video

Note that the results as expressed by the average absolute error metric on video in Table 4 are more accurate than those on the still frames in Table 3. This is because the video captures more plants. Also each plant is captured in multiple frames in the video and tracking allows height estimates to be refined across these multiple captures. Figure 23 visualizes a typical result of this algorithm. In two consecutive frames, we can see the detected "top-contour" in black. The vertical black lines demarcate the bounds of individual plants, and each plant is numbered in yellow at the top of the frame. The bounds are equal width because blind segmentation, as described in section 4.3.2, has been used. The yellow dots are the points in the top contour chosen as the height for each plant. The tracking algorithm will match the plants in the frames and update heights appropriately.

Finally, Figure 24 is an example of a histogram of height estimates as generated by our algorithm. The estimates are determined from video frames of 45 sorghum plants in the same row. The mean of the estimates was compared to the ground truth

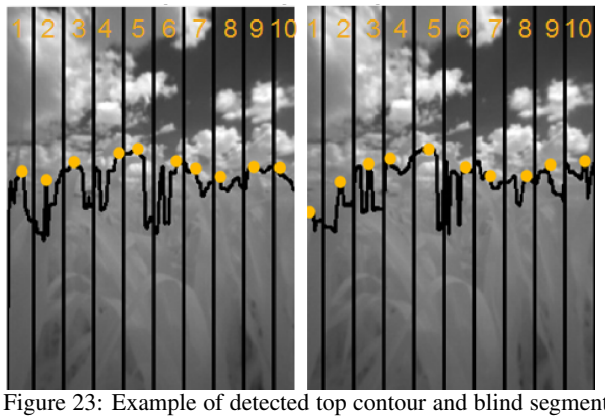


Figure 23: Example of detected top contour and blind segmentation results

mean for 10 plants in the same row. The average absolute error was 14%.

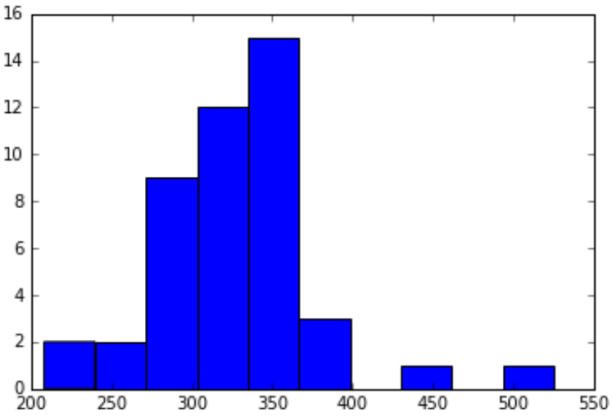


Figure 24: Histogram of height estimates in cm

5.3 Future Work

There are many promising future directions to be considered. The first would be to generalize the method presented here for sorghum to other plant varieties. Our algorithms work well for sorghum, and can be applied to corn, which looks and grows similarly. However, more significant modifications would need to be made for very different crops, such as miscanthus, though the overall algorithmic techniques are still useful. As part of algorithm generalization, many thresholds used in our algorithm were tuned for our data and field conditions, and could be generalized and made more adaptive.

An algorithmic avenue for future work is the development of a temporal Hough Transform. While we currently use the Hough Transform to detect stems, and then track them separately by maximizing our matching accuracy between frames, a temporal Hough transform would take a broader view and detect and track stems simultaneously. This would involve estimating stem locations and trajectories over the entire range rather than frame by frame. If we assume the robot is traveling at a constant velocity, to find a single stem line the temporal Hough transform would optimize over a 3 dimensional space of ρ, θ, and t where t is the time at which the stem enters the frame. In practice, the robot's velocity is not constant, and the velocity assumption needs to be

relaxed. To achieve this, we could perform a standard $2D$ ($\rho \times \theta$) Hough transform on each image, and tie the transforms together frame by frame to generate a $3D$ array ($\rho \times \theta \times t$). At this point, rather than simply finding the candidate line with the maximum votes in the $2D$ array, we would try to find a path connecting peaks in adjacent frames, while maximizing the probability of each $2D$ Hough transform in a manner similar to the Viterbi Algorithm.

As mentioned earlier, an RGB camera alone is insufficient, as it is unable to detect stems in a frame, and still needs the distance to the plant to convert pixels to real world measurements. However, standard RGB cameras are appealing for our method because a high resolution RGB camera costs much less than a high resolution TOF camera. Thus, a cost effective solution for accurate width estimation could involve detecting stems in low resolution infrared TOF images, corresponding the stem locations to locations in higher resolution RGB images, and estimating widths in the RGB domain. A simpler yet more costly method could entail using an additional infrared camera with higher resolution, but without depth capabilities. With this configuration, one could perform all detection and processing on this higher resolution infrared image, and simply correspond frames with the time of flight camera to determine depth to relate estimates in pixels to cm.

There are many opportunities for future work to improve the height estimation algorithm. As emphasized before, height estimation for sorghum is difficult due to large absolute heights and variability of height between varieties. The thick leaf canopy during later growth obstructs line of sight to the plant tips for any camera that is not mounted high enough. Furthermore, the variability in sorghum heights often exceeds the FOV of the camera, so while the tops of some plants are visible, others are not. It is impractical to address this by manually and continually adjusting the robot mast height. Even an adequately large FOV camera may not be able to view all the plant tops. Alternative camera configurations, such as a high mounted, downward pointing camera could reduce this issue. However, mounting a camera at great height on a mast while maintaining stability and minimizing motion is difficult in itself. Another possibility would be a camera overhanging the crop row and oriented to point directly downwards. This would simplify the configuration geometry, but the mounting problem would remain. New algorithms would need to be developed to segment plants in images collected in such a configuration.

A final avenue for future work is developing a method for motion compensation in height estimation. Our height estimation algorithm relies heavily on accurately knowing the camera height. As the camera is attached to a tall mast on a moving robot, maintaining a constant height can be difficult. The fields of sorghum we tested in had extremely flat ground, but camera height issues would be exacerbated in rougher terrain. Using data from an IMU on the robot and corresponding this with captured images could reduce estimation error.

Acknowledgments

The information, data, and 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-AR0000598. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. We thank Joshua Peschel and Sierra Young for building and configuring the robot, and providing field data. We'd also like to thank the entire TERRA-MEPP team, and Jihui Jin in particular, for their guidance and support.

References

[1] van der Heijden, Gerie, Yu Song, Graham Horgan, Gerrit Polder, Anja Dieleman, Marco Bink, Alain Palloix, Fred van Eeuwijk, and Chris Glasbey. . "SPICY: towards automated phenotyping of large pepper plants in the greenhouse." Functional Plant Biology 39.11 (2012): 870-877.

[2] Mohammed Amean, Z., T. Low, C. McCarthy, and N. Hancock. "Automatic plant branch segmentation and classification using vesselness measure." Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2013). Australasian Robotics and Automation Association, 2013.

[3] Pound, Michael P., Andrew P. French, Erik H. Murchie, and Tony P. Pridmore. "Automated recovery of three-dimensional models of plant shoots from multiple color images." Plant physiology 166.4 (2014): 1688-1698.

[4] Noordam, J. C., J. Hemming, C. Van Heerde, F. B. T. F. Golbach, R. Van Soest, and E. Wekking. "Automated rose cutting in greenhouses with 3D vision and robotics: analysis of 3D vision techniques for stem detection." International Conference on Sustainable Greenhouse Systems-Greensys2004 691. 2004.

[5] Holmgren, Johan, Mats Nilsson, and Hkan Olsson. "Estimation of tree height and stem volume on plots using airborne laser scanning." Forest Science 49.3 (2003): 419-428.

[6] Kaartinen, H., Hyypp, J., Yu, X., Vastaranta, M., Hyypp, H., Kukko, A., Holopainen, M., Heipke, C., Hirschmugl, M., Morsdorf, F. and Nsset, E. "An international comparison of individual tree detection and extraction using airborne laser scanning." Remote Sensing 4.4 (2012): 950-974.

[7] Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007

[8] Christophe Fiorio and Jens Gustedt, Two linear time Union-Find strategies for image processing, Theoretical Computer Science 154 (1996), pp. 165-181.

[9] Gordon, Geoff. "Detecting Edges and Other Image Features." Edges, Corners, Colors, Etc. Carnegie Mellon University, 16 June 2004. Web. 15 Aug. 2016.

[10] Kensheng Wu, Ekow Otoo and Arie Shoshani, Optimizing connected component labeling algorithms, Paper LBNL-56864, 2005, Lawrence Berkeley National Laboratory (University of California), http://repositories.cdlib.org/lbnl/LBNL-56864

[11] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup, Efficient subpixel image registration algorithms, Optics Letters 33, 156-158 (2008).

[12] Minervini, Massimo, Hanno Scharr, and Sotirios A. Tsaftaris. "Image Analysis: The New Bottleneck in Plant Phenotyping [Applications Corner]." IEEE signal processing magazine 32.4 (2015): 126-131.

[13] Frangi, Alejandro F. "Three-dimensional model-based analysis of vascular and cardiac images." (2001).

[14] Canny, John. "A computational approach to edge detection." IEEE Transactions on pattern analysis and machine intelligence 6 (1986): 679-698.

Author Biography

Tavor Baharav is currently pursuing a BS in Electrical Engineering and Computer Science and a minor in Applied Mathematics at the University of California Berkeley (expected graduation date: 2018). His current research focuses on active learning approaches for computing sparse Walsh Hadamard Transforms.

Mohini Bariya is currently pursuing a PhD in Electrical Engineering at the University of California Berkeley. She received her bachelors degree in Computer Science from UC Berkeley in 2016. She is interested in signal and image processing and is motivated by problems in agriculture and energy. Her current research is on algorithms for the electric grid.

Avideh Zakhor joined the faculty at UC Berkeley in 1988 where she is currently the Qualcomm Professor of Electrical Engineering and Computer Sciences. Her areas of interest include theories and applications of signal, image and video processing, 3D computer vision, and multimedia networking. She has won a number of best paper awards, holds 6 U.S. patents, and is the co-author of three books with her students.

Prof. Zakhor received the B. S. degree from California Institute of Technology, Pasadena, and the S. M. and Ph. D. degrees from Massachusetts Institute of Technology, Cambridge, all in electrical engineering, in 1983, 1985, and 1987 respectively. She was a General Motors scholar from 1982 to 1983, was a Hertz fellow from 1984 to 1988, received the Presidential Young Investigators (PYI) award, and Office of Naval Research (ONR) young investigator award in 1992. In 2001, she was elected as IEEE fellow and received the Okawa Prize in 2004.

She co-founded OPC technology in 1996, which was acquired by Mentor Graphics (Nasdaq: MENT) in 1998, UrbanScan Inc. in 2005 which was acquired by Google in 2007, and Indoor Reality in 2015.