Snakes in a Plant: 3D Reconstruction of Foliage using Tethered Active Contours

Patrick A. Plonski, Volkan Isler

June 15, 2018
Snakes in a Plant: 3D Reconstruction of Foliage using Tethered Active Contours

Patrick A. Plonski and Volkan Isler

Abstract—We present a novel method for reconstructing 3D models of foliage from a pair of images. Leaves are especially challenging objects to reconstruct in natural settings because of the lack of distinct features. We present a novel method for simultaneously growing contours to detect leaf boundaries. We then compute the transformation between the leaves to generate disparity maps. We demonstrate the effectiveness of our method on challenging instances where standard reconstruction methods fail.

I. INTRODUCTION

As the population of the world increases, while the resources to support us remain fixed, new techniques for monitoring crops are due to take on growing importance. New technologies in robotics and sensors allow increasingly dense monitoring of test plots, but merely taking measurements is insufficient to obtain the full value of the sensor: It is also crucial to develop new methods of processing the measurements.

Some important agricultural properties of all plants are the sizes, positions, and orientations of their leaves. From these quantities the plant health and yield can be estimated. Leaf detection has been specifically used for phenotyping in controlled indoor environments, however existing methods require direct depth sensors. If leaves can be detected using only a camera, both indoor and outdoor monitoring will become feasible at lower price points.

Leaves have specific properties that make it challenging for standard stereo vision methods to distinguish them. Leaves are highly self-similar in texture, which makes unique local feature matches hard to come by. They are also highly self-similar in color, which inhibits superpixel-type segmentation. However, when a leaf partially occludes another leaf, there is usually a change in light intensity at the boundary because the two surfaces have different angles. Therefore, shape-based detectors that find leaf outlines in an edge map are a natural choice for the domain.

Shape-based visual leaf detection is not easy. In addition to edges at leaf boundaries, there are also intra-leaf edges of similar magnitude. Existing shape-based methods of visual leaf detection use only one view at a time, and require that a single leaf be clearly visible against a background. These methods are unsuitable for the overlapping leaves of dense foliage. However, by using multiple views, there is much less ambiguity in which edges define leaf boundaries. Our contribution in this work is an extension of active contour leaf detection to multiple simultaneous views, for dense foliage. By combining shape methods with 3d triangulation, we do better than using either method by itself.

Our contribution:
- An extension of active contour leaf detection to multiple simultaneous views
- A method to use these matched active contours to estimate foliage geometry
- Demonstrated on different plants in different conditions

II. RELATED WORK

In this section we will present the methods in the literature that are used to detect leaves. The method proposed by Bradley et al. [1] is the first designed for multi view reconstruction of the 3d positions of leaves in dense foliage. Their first step is to compute a set of triangulated SIFT features to use as a base for reconstructing the surface. Then they compute a disparity map using the sparse features and Normalized Cross Correlation, and repeatedly fit a scanned 3d leaf template to the surface. Finding the initial pointcloud can be very challenging, depending on the type of images and the leaf species. We found that frequently there will be many leaves that have no matched features on them. Another drawback of their method is that they require a very strong prior belief on the 3d model of a leaf. Other methods that start with a general 3d reconstruction and fit leaves to it include Quan et al. [9], who start with a pointcloud computed using the approach presented in [7]. Then they segment individual leaves using a graph cut algorithm that attempts to splits pixels that have changes in depth or color. Teng et al. [10], uses the same graph cut technique on a depth map constructed through optical flow between two close images. These methods share the disadvantage that when the leaves lack distinct visual features in their interiors, initial 3d reconstruction is challenging. As our proposed method only uses leaf outlines, it performs well in such situations.

There is recent interest in the problem of tracing the outline of leaves in a single image for the purposes of leaf classification, given color or shape priors. Wang et al. [12] use a watershed method that relies on the initial color segmentation leaving clear silhouettes that can be separated. Manh et al. [8] and Cerutti et al. [2], [3] use shape models for leaf polygons. Using gradient descent, the leaf polygons are moved to lie near edges, and near boundaries of the foreground; thus a good initial color segmentation is again required. Our proposed method uses multiple views to resolve the ambiguity that arises when leaves are too dense.
for background subtraction to provide useful silhouettes for each one.

Existing work on detecting leaves for phenotyping uses depth sensors to estimate the 3d geometry of the entire domain as the first step. Van der Heijden et al. [11] perform initial segmentation using a graph cut method on a depthmap constructed using Time Of Flight distance sensors. Then the closest leaves to the camera are individually selected using a region growing method that avoided crossing strong edges in light intensity or depth. Chene et al. [4] similarly rely on direct depth measurements, specifically from a Kinect sensor. LIDAR-type sensors are very expensive, and Kinect-like RGB-d systems have low resolution and poor performance in outdoor setups; this motivates the search for a method to extract leaf geometry using cameras only.

Previous techniques of 3d reconstruction generally start with a weak model of what is to be reconstructed. In this work we start with a strong model of what an image of a leaf looks like, and leverage it to improve the accuracy of leaf estimation and foliage reconstruction from images.

III. PROBLEM FORMULATION

Here we formally state the problem of foliage reconstruction.

Given two images from equivalent cameras of the same patch of foliage from different viewing angles, find the leaves that are clearly visible and unoccluded from both angles, and reconstruct the foliage up to a scale.

To solve this challenging problem we make some assumptions about the leaves in the images. We require that they can be represented as compact planes, and that the translation between the camera views is small enough so that the ordering of leaves on the projective plane does not change. We also assume the camera intrinsic parameters are known, and that the transformation between the camera frames can be detected (for instance, through standard structure from motion techniques from shared features detected in the background). If the camera intrinsic parameters are not known then the stereo disparities can be computed but the leaves cannot be accurately triangulated.

In contrast with previous work, we allow a larger translation between camera frames (as we do not use optical flow), we do not require that more than a single point on a single leaf be matched between the images, and we do not require the use of a range sensor.

IV. APPROACH

In this section we present our method for finding leaves from two uncalibrated images. There are six main steps, which are presented in order. First color segmentation is applied to eliminate parts of the images that cannot be leaves. Then the images are rectified to aid matching later. Next a separate set of snakes is estimated for the leaves in each rectified image. Next the leaves are matched, and tethered snakes are initialized and grown in the images based on the
matches. Finally, the positions in which the tethered snakes come to rest are used to estimate the 3d leaf positions.

A. Background Subtraction

The first step is to remove out the parts of the images that are clearly not parts of leaves. We achieve this by applying thresholds in the CIE L*a*b* colorspace. In this space colors are represented with a lightness $L$ and two color parameters $a$ and $b$. The $a$ dimension represents the balance between green and red, while the $b$ dimension represents the balance between yellow and blue. For our segmentation we ignore $L$ and apply a threshold to both color dimensions. To enforce a minimum size on leaves, we also apply thresholds to the image after a Gaussian blur operation, where the size of the blur is selected based on the minimum leaf we wanted to detect. Formally, if a pixel had color $\{L, a, b\}$ and the same pixel in the blurred image has color $\{L', a', b'\}$, we consider it to be in the foreground if $(a < \alpha_1) \land (b > \beta_1) \land (a' < \alpha_2) \land (b' > \beta_2)$, given parameters $\alpha_1, \beta_1, \alpha_2,$ and $\beta_2$ which should be selected or learned based on the color of the plant.

B. Stereo Rectification

To aid matching, the next step is stereo rectification. We closely follow the method in Hartley and Zisserman.
chapter 11 [5] for uncalibrated stereo rectification. We use SIFT features to compute the fundamental matrix, and use RANSAC to reject outliers features that are far away from the best homography describing their transformation between frames. This relies on the fact that the camera translation is small compared with the change in depth between the closest and the farthest features. By assumption, we know this to be true for the foreground. Since the background is generally farther away, we can assume the property holds for it as well. To align the rectified frames, we minimize the least square distance between the features classified as foreground, according to background subtraction. This only requires a small number of representative foreground features.

C. Myopic Active Contours

Kass et al. [6] first developed the concept of active contours, or “snakes”, to find shapes in images. A snake consists of nodes in an image joined by link line segments. The method of active contours uses gradient descent to minimize an energy function for a snake which is the sum of two components: an external component which represents how well the snake matches the image (usually be penalizing distance from edges or features of interest), and an internal component which represents how well the snake matches a predefined model (usually by penalizing abrupt changes in curvature or node spacing).

For leaf detection we use an external energy function that depends on a normalized map of Sobel edge intensity, so that, in the color-segmented foreground, mean edge intensity is 0 and standard deviation is 1. The energy cost for each pixel on the boundary of the leaf is linear to the negation of edge intensity; thus strong edges are sought and flat, featureless regions are avoided. To this edge energy we add a penalty term for each pixel that is outside the thresholded foreground.

We designed our internal energy function to induce the snake to grow outward until it encounters strong edges, while remaining compact and regular, if possible. The energy cost is positively linear to the perimeter, and negatively linear to the radius. The balance between the perimeter and radius cost parameters controls how far from a circle the snake is allowed to deform to. We also enforce consistent node spacing (consistent link length) with a quadratic term the penalizes deviation from the mean.

Snake evolution is performed by visiting each node in sequence and picking the best pixel in the neighborhood for the node to move to, given the edge map, the positions of the neighboring nodes, the mean link length, and the center point of the snake region (which is used to compute the radius). Evolution is terminated when there are no local improvements that can be made.

Since optimization follows the gradient, it is crucial to initialize each snake in the attraction basin of a leaf. We also want to find as many leaves as possible. To this end, we define seed points based on the edge map, and an assumed minimum leaf size. As we expect leaves to have weak edges, it would be reasonable to search for leaf centers by detecting positions with the smallest amount of nearby edge weight. However, in practice these positions are often in the middle of leaf clusters, and it is ambiguous where the initialized snake should grow to. Therefore, we define a function $g$ which is the Sobel edge intensity map for the foreground, convolved with a Gaussian filter of scale $2\sigma_t$, minus twice the map convolved with a Gaussian filter of scale $\sigma_t$. When $\sigma_t$ is the size in the image of the smallest leaf, the function $g$ is a bandpass filter which reaches its maximum value in the center of small leaves, or near the the sharp edges the define boundaries of larger leaves. By initializing the snakes near sharp edges, we improve the likelihood that the snakes grow in the appropriate direction.

Our method for growing snakes that trace all leaves in a particular image is the following: Compute seed points $S$ as the local maxima of $g$ that take positive value. Select the seed point from $S$ with highest value, and initialize a snake surrounding that point at a size smaller than the smallest leaf. Allow the snake evolve to minimize its energy. When it has reached a local minima, remove all seed points from $S$ that are inside the snake, and if there is at least one seed left in $S$, initialize a new snake with the highest one remaining.

D. Leaf Matching

When we have the snake sets for each view, now we need to match the snakes across the views so that we can initialize our tethered snakes. For matching purposes we parameterize each snake by its centroid and magnitude, weighting pixels for each snake in a manner which is inversely proportional to how many snakes the pixel is contained in. Then we match each snake to its nearest neighbor in the other view, where distance is defined from similarity in size, centroid disparity along the epipolar line, and centroid disparity perpendicular to the epipolar line. After rejecting as outliers all those snakes which have no neighbor within a threshold distance, we initialize the tethered active contours of the next step based on the weighted means of the connected components in the bipartite graph of matches. The initial sizes of tethered contours are equal for each view, and set based on sums of the magnitudes of the initial snakes, scaled down to ensure that the initial contour remains inside the leaf it is supposed to represent.

E. Tethered Active Contours

Growing the tethered active contours is much the same as growing the myopic ones. A tethered active contour is just a set of nodes for each image, together with an affine transformation between the nodes in the images. Because the images are rectified, we restrict the transformation to consist of a translation and a scaling, both along the epipolar line. This affine transformation approximates the leaves as planar, and small compared with their distance to the camera. The tethering is achieved with an added energy term for squared distance from the transformation of the corresponding node in the other view. The transformation is recomputed to minimize tethering energy after each node has been adjusted once.
F. 3D Reconstruction

Finally, given final positions of tethered contours, the disparity can be directly computed from the transformation. From disparity we obtain depth.

V. EXPERIMENTS

In this section we demonstrate the suitability of our method for detecting and reconstructing leaves.

We tested our method with two different plants. The images were taken outdoors with the camera on a Samsung Galaxy S3. We could not obtain a ground truth disparity map but the estimated disparity looks qualitatively accurate, as shown in the figures. Occasionally the tethered snakes would grow to contain more than one leaf, particularly when there was a many-to-one matching after the myopic step, however, the tethered snakes were in general much more accurate at tracing leaf boundaries. This justifies the use of multiple views for active contour leaf tracking.

VI. CONCLUSION

The problem of using cameras to reconstruct the 3d position of leaves if challenging, because the self-similarity of leaves inhibits standard stereo vision techniques. In this work
we have presented a method of using tethered active contours to perform this reconstruction. We have demonstrated that the method performs well in practice. Because our method is very different from existing reconstruction methods, it can perhaps be combined with them to form a very robust estimator, in situations when some of the leaves have unique features. Additional further areas in which the algorithm can be improved include allowing partial myopic leaves to match, and growing multiple active contours at once so that boundaries between leaves can be directly contested by the snakes to determine which one leaf is more likely to contain a particular pixel.

ACKNOWLEDGMENT

This material is based upon work supported by the MN-DRIVE initiative.

REFERENCES