Use of Computer Vision to Track Thin Body Motion with the Application of Tracking Passion Plant Vine Tendrils

Joshua N. Moser

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Masters of Science in Mechanical Engineering

Alfred L. Wicks, Chair
A. Lynn Abbott
Alan Asbeck

April 24, 2018
Blacksburg, Virginia

Keywords: Computer Vision, Motion Tracking, Passion Plant, Vine Tendril, Kalman Filter
Copyright 2018, Joshua N. Moser
Use of Computer Vision to Track Thin Body Motion with the Application of Tracking Passion Plant Vine Tendrils

Joshua N. Moser

Academic Abstract

This research focuses on developing an algorithm set to track the vine tendril motion of a passiflora incarnate, commonly referred to as the passion fruit plant, to facilitate research into if there is a correlation between plant motion and plant health. An evaluation was done of clustering based color segmentation with a focus on K-means, feature / texture segmentation utilizing Scale Invariant Feature Transforms (SIFT), and temporal based segmentation using Gaussian Mixture Model Background Subtraction to segment out the tendril in each video frame. Morphological image processing methods, such as dilation and connected component analysis, were used to clean up the segmentation results to give an estimate of the vine tendril’s location at each frame. Kalman filtering was then used to track the tendril’s location through the different frames dealing with large jumps in tendril location, cases where the tendril remained stationary between frames, and cases where there was error in the segmentation process. The resulting algorithm set was successful at tracking the tendril during times when the tendril had large jumps in position and it almost always succeeded in keeping track of the tendril during errors in the segmentation due to lack of tendril motion. The few cases that were not successful were evaluated and suggestions were made to resolve these issues in future data collection.
Use of Computer Vision to Track Thin Body Motion with the Application of Tracking Passion Plant Vine Tendrils

Joshua N. Moser

General Audience Abstract

This research focused on developing an algorithm sequence that could find the tendril of a passiflora incarnate, commonly referred to as the passion fruit plant, in a single frame of a video and then track that tendril through the different frames in the video. Having the ability to track a plant tendril through a video allows biologists to research if there is a link between the amount a plant moves and the plant’s health. The algorithms evaluated for finding the plant in the image used color, features and motion to try and distinguish the tendril from the rest of the image. After the tendril was found, a tracking algorithm that combined a prediction from a model for the tendril’s location with the measured location was used to deal with noise and errors in the measurement. It was found that using the motion based algorithm worked the best to find the tendril (with the addition of some image processing to remove noise). This combined with the tracking algorithm allowed for the tendril to be successfully tracked through the different frames with one exception. Future work and recommendations were made to deal with this exception.
Acknowledgments
I would like to take the time to thank all those who have made it possible for me to make it to this point in my graduate studies. First, I would like to thank Dr. Wicks for advising me during this research and teaching me how to become a researcher in addition to an Engineer.

I would also like to thank the rest of my committee: Dr. Abbott for all the input he provided in the field of image processing and Dr. Asbeck who supported my interest in a nontraditional topic for the field of Mechanical Engineering.

To my parents, Jim and Dena Moser, I would like to say thank you for all of the principles you taught me and the support you and my brother Ryan, have given me during my graduate studies.

I would also like to thank all of those in the Mechatronics lab and my friends and colleges outside of my academic studies who supported me and were there to listen and provide feedback for many different ideas and discussion topics.

Last but not least, I also owe debt of graduate to God. I would not be who I am or where I am today without Him.
# Contents

## 1 Introduction & Background

1.1 Motivation ................................................. 1  
1.2 Proposed Solution ....................................... 2  
  1.2.1 Image Processing ..................................... 2  
  1.2.2 Tracking ............................................. 2  
1.3 Plant Motion ............................................. 3  
1.4 Thesis Architecture ..................................... 3  
  1.4.1 Introduction & Background ......................... 3  
  1.4.2 Theory .............................................. 4  
  1.4.3 Experiment .......................................... 4  
  1.4.4 Results ............................................. 4  
  1.4.5 Conclusions & Future Work ....................... 4  

## 2 Theory .................................................. 6  

2.1 Object Segmentation ..................................... 6  
  2.1.1 Color Based .......................................... 7  
  2.1.2 Feature-Based ...................................... 14  
  2.1.3 Temporal Based ..................................... 15  


2.2 Image Processing .................................................. 17
  2.2.1 Morphological ................................................. 17
  2.2.2 Median Blur .................................................. 22
2.3 Object Tracking .................................................. 22
  2.3.1 Kalman Filter ................................................ 23

3 Experiment ......................................................... 25
  3.1 Plant Type Choice .............................................. 25
  3.2 Hardware Setup ............................................... 26
  3.3 Data ............................................................ 26
  3.4 Segmentation .................................................... 29
    3.4.1 Color Based .............................................. 30
    3.4.2 Feature Based ............................................ 33
    3.4.3 Temporal Based .......................................... 33
  3.5 Processing Segmented Image .................................. 34
    3.5.1 Median Blur .............................................. 34
    3.5.2 Dilation .................................................. 35
    3.5.3 Connected Componnets .................................. 36
  3.6 Tracking ........................................................ 37
    3.6.1 Motion Model ............................................. 38
    3.6.2 Kalman Filter ............................................ 39

4 Results ............................................................ 41
  4.1 Plant Size .................................................... 41
  4.2 Segmentation .................................................. 43
4.2.1 Color Based ......................................................... 43
4.2.2 Temporal Based .................................................. 47
4.3 Processing ................................................................. 49
  4.3.1 Median Blur .................................................... 49
  4.3.2 Dilation .......................................................... 50
  4.3.3 Connected Components ................................. 50
4.4 Tracking ................................................................. 51
  4.4.1 Kalman Filter .................................................. 51
  4.4.2 Motion Logic .................................................. 51
  4.4.3 Results Review ............................................... 54

5 Conclusions & Future Work ........................................ 58
  5.1 Conclusions ......................................................... 58
    5.1.1 Process Pipeline Summary ............................ 58
    5.1.2 Goal Summary ........................................... 63
  5.2 Future Work ....................................................... 64
    5.2.1 Recommendations ....................................... 64
    5.2.2 Final Remarks ........................................... 65

Bibliography ................................................................. 66
List of Figures

2.1 RGB colorspace\[1]  ................................................................. 7
2.2 HSV colorspace\[2]  ................................................................. 8
2.3 L* a* b* colorspace\[3]  ............................................................. 9
2.4 K-means Example  ................................................................. 12
2.5 Orientations for local points calculated and scaled.[4] ............... 15
2.6 Connectedness  ................................................................. 18
2.7 Dilation Example  ................................................................. 19
2.8 CCL Two Pass Example  ........................................................... 21
2.9 Example of Median Filter in 2D\[5] ............................................. 22
3.1 One motion cycle of plant tendril  ........................................... 29
3.2 Channels in RGB image  ....................................................... 30
3.3 Channels in HSV image  ....................................................... 30
3.4 Channels in L* a* b* image  .................................................. 31
3.5 K-means RGB results  ....................................................... 32
3.6 K-means HSV results  ....................................................... 32
3.7 K-means L* a* b* results  .................................................. 33
3.8 Channels in RGB image  ....................................................... 34
3.9 Median blur results  ....................................................... 35
List of Tables

2.1 Time Update Equations .................................................. 24
2.2 Measurement Update Equations ........................................ 24
3.1 Silhouetted Index Results for Colorspace ....................... 31
3.2 Time Update Equations Incorporating Model Parameters .... 39
3.3 Measurement Update Equations Incorporating Model Parameters ...... 40
Chapter 1

Introduction & Background

1.1 Motivation

Plant health is an area of interest for biologists, specifically when it comes to determining ways to detect when a plant is in the early stages of leaving a healthy state. A current approach under research by biologists for determining plant health is to evaluate the energy expenditure of the plant. The hypothesis is that a plant that is suffering from an ailment will redirect energy into fighting that ailment. It is proposed that, by watching a plant’s motion when it is healthy, a baseline for its motion can be established which will allow for the detection of a change in the motion due a change in energy allocation due to an ailment. This would alert researchers of a problem before traditional symptoms such as wilting or discoloration took place. This leads to the biology researcher needing to evaluate hours or days worth of video footage of plant motion to determine if there is a correlation between plant health and motion. If a correlation is found, there will need to be some way to generate a model of the motion for the baseline of healthy plant motion so any deviations from the normal motion can be quantified. This is where a research problem for the engineer arises. How can a system be developed to observe the video footage of plant motion so that an analysis of the motion can be done autonomously? The software / algorithm facet of this question is the motivation for this thesis.
1.2 Proposed Solution

To address the problem of following plant motion through the frames in a video sequence, the proposed solution, and the goal of this thesis, is as follows. Use image processing techniques to locate and recognize the portion of the plant that is moving and is of interest to the researcher. Once it has been recognized, approaches from tracking methodology will be used to reduce the error in the image processing. This will allow for the algorithm set to keep track of the moving portion of the plant through a sequence of frames in a video despite noise or pauses in plant motion.

1.2.1 Image Processing

Image processing is a type of digital signal processing that is applied to two-dimensional information in the form of a matrix. Each element reflects the value of a finite number of bits. These bits represent the amount of light captured by the imaging sensor at that point in the field of view. The primary techniques in the image processing field that will be used are the algorithms for image segmentation.

Image Segmentation

Image segmentation is the processes of dividing up an image based on content properties. These properties can be color based, shape based, feature based and more. For the problem of tracking part of a plant in an image sequence, image segmentation is important because it will be used in the process of distinguishing the plant from the background of each image frame. It can also be used to try and classify different portion of the plant from one another. An approach to this that will be used in this thesis is clustering which will be discussed in the theory section of this document.

1.2.2 Tracking

Tracking can be defined as following or watching the path of something or someone. This is often a task for visual systems such as cameras, that will record an object moving. The location of the object in the image, the measurement, is not always known with complete
certainty because of noise in the image (data). This is due to errors converting the light into digital matrix values or from uncertainty introduced in the segmenting process between frames. Because of this a statistical combination of measured data and future location prediction based off of a motion model is used to determine a more accurate estimation of an object’s actual location in an image. The algorithm used in this thesis to accomplish this will be the Kalman filter.

1.3 Plant Motion

There are two general types of plant movements. The first category is tropism. This category of movements are directed growth due to external stimuli. An example of this is a plant growing towards light (phototropism) or away from the ground (gravitropism). The direction is a result of the stimuli. The second category, nastic, are movements that are also caused by stimuli. However, the motion resulting form the stimuli can be in an direction independent of the stimuli. An example of this would be plants that curl up at night. The type of plant motion that is of interest for this study is circumnutational motion, a type of nastic motion. Circumnutation motion is a circular or helical movement of a plant stem, tendril or leaf. This type of motion is of interest because of its repetitive nature. After the tools have been developed in this thesis to track the plant motion, this type of motion (circular and repetitive) will make it easier for biologists to run a series of controlled tests to try and determine the a correlation between plant motion and plant health.

1.4 Thesis Architecture

1.4.1 Introduction & Background

Chapter 1 of this document covers the Introduction and Background for this Thesis. First is the motivation section, where the reason for this research is discussed and the problem that this thesis is addressing is described. This chapter then gives a brief, high level, view of the proposed solution to the problem. Following this is a section on plant motion describing different types of motion and describing the type that will be studied in this thesis. Finally, this chapter discuss the architecture of the thesis and gives an overview description of what
each chapter will cover.

1.4.2 Theory

The Theory chapter, chapter 2, will discuss algorithms used to carry out the proposed solution given in chapter 1. This will start out discussing the object segmentation algorithms where it will show the theory for color transformations and K-means clustering, Scale Invariant Feature Transform (SIFT) for the feature based segmentation, and Gaussian Mixture Models for the temporal based object segmentation. Next, the image processing techniques such as filtering, morphological operations and connected components will be discussed since they will be used to clean up the results of the segmentation and identify the tendril in the image. Finally, the theory behind the Kalman filter will be explained for the use of tracking the tendril through image frames.

1.4.3 Experiment

The Experiment chapter, chapter 3, focuses on the specifics of the experiment such as the hardware configuration used to collect the data and the plant that was used as the subject for data acquisition. Then, the reasoning behind, implementation sequence, and preliminary outputs of the algorithms described in the Theory chapter will discussed.

1.4.4 Results

In the Results chapter, chapter 4, a holistic view of the results from each step in the algorithm will be given for each point in the process pipeline. This will include explanations for why the results take the from seen from the algorithm outputs.

1.4.5 Conclusions & Future Work

In the final chapter, chapter 5, a summary of the algorithm sequence will be given and the results will all be tied together to show how they meet the goal given in chapter 1. Recommendations will also be made for future work. This will include recommendations for
an improved set up for the experiment based off the results found in this thesis. This chapter will also cover alternative algorithms that could be used in the future to potentially improve the results or add a higher level or robustness to noise or occlusion.
Chapter 2

Theory

In this chapter, the algorithms used to process the data will be discussed in detail. First, different colorspaces and how they are used will be explained. Next, the theory of K-means clustering will be explained. Following that, the theory behind the Scale Invariant Feature Transform will be briefly discussed (since it does not play a major role in the process). The theory behind the Gaussian Mixture Model, specificity in the context of background subtraction, will then described. Following this, morphological operations such as dilation and connected components will be discussed. Median filtering will also be described for the use of noise reduction. Finally, the theory behind the Kalman filter will be discussed in the context of tracking in images.

2.1 Object Segmentation

As mentioned in the previous chapter, to track an object through an image sequence, first that object must be separated from the rest of the image. This can be done through several different approaches. Three of these approaches: clustering, feature extraction, and background subtraction, will be discussed in the next few subsections.
2.1.1 Color Based

As the name implies, color-based segmentation relies on using differences in the color (pixel values) to distinguish between objects in an image, allowing the desired object to be segmented from the rest of the image. One of the common approaches for doing this is clustering.[11] When using color-based clustering it is important to first talk about the different colorspace that exist because each colorspace represents the information differently. These differences can be leveraged to help get the desired content from the image (in the case of this thesis the plant tendril).

Colorsapce

The standard representation for non-monochrome images is the R (red), G (green), Blue (B) colorspace.[12] This is because the RGB values in an image represent the three intensities from the red, green, and blue filters in a typical imaging device (figure 2.1 gives an example of the RGB colorspace).

From the RGB colorspace, other colorspaces can be derived using linear and nonlinear transformations.[12] For example, the HSV (hue, saturation, value) colorspace allows a color to be quantified by the color / shade (a value representing this is in the hue channel), the
purity of the color (a value for the saturation / intensity), and the brightness (the value). This allows for easy evaluation of overall brightness of a color. Figure 2.2 shows a representation of the HSV colorspace. To convert from the RGB colorspace to the HSV colorspace, each of the channels must first be mapped from their default range of $0 \rightarrow 255$ to $0 \rightarrow 1$. This is done by normalizing each of the channels by 255. These new representations of RGB are noted as $R’G’B’$. Two parameters, shown in equation 2.1 are then calculated using $R’G’B’$. These two parameters can then be used to calculate values for the H, S and V channels (equations 2.2, 2.3, & 2.4 respectively).

$$MAX = \max(R’, G’, B’)$$
$$MIN = \min(R’, G’, B’)$$

$$H = \begin{cases} 
0, & \text{if } R’ = G’ = B’ \\
60^\circ \times \left(0 + \frac{G’-B’}{MAX-MIN}\right), & \text{if } MAX = R’ \\
60^\circ \times \left(2 + \frac{B’-R’}{MAX-MIN}\right), & \text{if } MAX = G’ \\
60^\circ \times \left(4 + \frac{R’-G’}{MAX-MIN}\right), & \text{if } MAX = B’ 
\end{cases}$$

$$S = \begin{cases} 
0, & \text{if } R’ = G’ = B’ \\
\frac{MAX-MIN}{MAX}, & \text{else}
\end{cases}$$
A colorspace is chosen based off what image processing is going to be done. For many image processing applications, including segmentation, some kind of distance measurement (such as euclidean) is done to compare how similar two colors are. The $L^*a^*b^*$ space (figure 2.3), created by the Commission Internationale de l’Eclairage, was designed to achieve a representation of colors that conform to a human’s perception of color.\textsuperscript{16} In other words, the extent of differences in colors that humans see will be represented in the same way with the $L^*a^*b^*$ colorspace using a luminance channel ($L^*$) and two color channels ($a^*$ and $b^*$).

\begin{equation}
V = MAX
\end{equation}

Figure 2.3: $L^*a^*b^*$ colorspace\textsuperscript{3}.

To convert from the RGB to the $L^*a^*b^*$ colorspace the RGB channels have to be transformed into the CIE tristimulus values.\textsuperscript{17} This is done using the equation set in 2.5. After this, the
conversion to the L*a*b* colorspace can be accomplished using the equation set 2.6.18

\[
X = 0.4303R + 0.3416G + 0.1784B \\
Y = 0.2219R + 0.7068G + 0.1784B \\
Z = 0.0202R + 0.1296G + 0.9393B
\]

(2.5)

\[
L^* = 116 \left[ f\left(\frac{Y}{Y_0}\right) - 16 \right] \\
a^* = 500 \left[ f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right] \\
b^* = 200 \left[ f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right]
\]

(2.6)

where

\[
f(q) = \sqrt[3]{q} \quad q > 0.008856 \\
f(q) = 7.787q + \frac{16}{116} \quad q \leq 0.008856
\]

\[
X_0 = 0.950456, \; Y_0 = 1.000000, \; Z_0 = 1.088754
\]

Sometimes, instead of representing an image as a three channel color image, it is desirable to represent it as a single-channeled image (called a grayscale image) based on the intensities of the three color channels. This is done by weighting each of the RGB channels and summing them to get a single value representing each pixel. The standard weighting for image processing for each of the channels is shown in equation 2.7.19 To get a binary image from a single channel image, a threshold operation is done where any value above the threshold is given a value of 1 and anything below the threshold is given a value of 0.

\[
Gray = 0.3R + 0.59G + 0.11B
\]

(2.7)

Clustering

Clustering, as the name suggests, is used to group like things in images. There are two common methods for color-based clustering in computer vision / image processing, K-means
and mean shift with K-means being the most widely used and the one used in this research.[11]

Let’s say that \( X = \{x_i : i = 1, ..., n\} \) where \( x_i \) is a data point (in the case of this thesis, a pixel). K-means will cluster the points in the set \( X \) to their closest cluster center (sometimes referred to as centroids) from the set of cluster centers, \( C = \{c_k : k = 1, ..., K\} \). Each data point is given a label according to the center it is closest to. For a given cluster \( (k) \) we can define these labels \((l)\) as the set \((J)\) shown in equation 2.8. This equation can be understood as all the data points, \( q \), that are in the \( k^{th} \) cluster make up the set \( J_k \).

\[
J_k = \{i \in q : l(i) = k\} \quad (2.8)
\]

The K-means clustering algorithm has four main steps, seen below.[20][21][22] A visual representation of this algorithm can be seen in figure 2.4.[23]

1. Choose number of clusters, \( K \), and initialize starting location for the centroids / centers \((c_{k=1...K})\).

2. Compare each pixel against each cluster centroid using euclidean distance measurement and assign it to the cluster that it matches the closest (equation 2.9).

\[
l(i) = \arg\min_{k \in K} (\|x_i - c_k\|_2) \quad (2.9)
\]

3. After each pixel has been assigned a cluster, average all of the pixels assigned to a given cluster and set the resulting average as the cluster’s new centroid (equation 2.10).

\[
c_k = \mu_k = \frac{\sum_{i \in J_k} x_i}{\sum_{i \in J_k} 1} \quad (2.10)
\]

4. Repeat steps 2 & 3 until pixels and centroids converge to constant identities and value, that is, minimize equation 2.11.

\[
E(C) = \sum_{k=1}^{K} \sum_{i \in J_k} (\|x_i - \mu_k\|^2) \quad (2.11)
\]
There are many ways to evaluate how well the data was clustered. However, they generally fall into one of two groups; external criteria and internal validation. External criteria are those that use knowledge that is not intrinsic to the data to evaluate the clustering. An example of this would be using user specified intuition about the data being clustered. Internal criteria use the metrics inherent to the data set, that is, those being evaluated in the clustering. For this research internal criteria methods will need to be used since the algorithm is being applied to data where no ground truth is known to compare against. There are many different index to evaluate clustering with internal criteria. Rendón et al. list many of the common ones and does a comparison between them to evaluate which ones work best for K-means clustering. They found that the silhouette index (SIL) and the Davies-Bouldin index (DB) both preformed the best (and equally well) over all of the other standard indices.
evaluated.\textsuperscript{[24]} Due to this the silhouette index was chosen to evaluate the clustering done in this thesis.

The silhouette index can be thought of as a measure of group compactness and the separation between groups.\textsuperscript{[7] \textsuperscript{[26]}} For a data set \(X = \{x_i : i = 1, \ldots, n\}\) grouped into a set of clusters, \(C = \{c_k : k = 1, \ldots, K\}\) a silhouette width (value), \(S(x_i)\), is assigned to each data point, \(x_j\), (where \(j\) is the data point from set \(X\) being evaluated) in a given cluster \((c_k)\).

\[
S(x_j) = \frac{b(x_j) - a(x_j)}{\max(a(x_j), b(x_j))} \quad (2.12)
\]

Equation 2.12 shows how the silhouette value is calculated for a given data point. In equation 2.12, \(a(x_j)\) is the within-cluster mean distance (defined in equation 2.13). In equation 2.13 \(n_k\) is the number of data points in cluster \(c_k\). \(Y_j\) is defined as the set of all the data points in the \(k^{th}\) cluster, \(c_k\), except for the point being evaluated \((x_j)\). Mathematically this can be expressed as \(Y_j = \{X | X \in c_k \land x_i \neq x_j\}\). The function \(d()\) is the distance function used in the clustering method, in this case euclidean distance.

\[
a(x_j) = \frac{1}{n_k - 1} \sum_{Y_j} d(x_j, Y_j) \quad (2.13)
\]

In equation 2.12, \(b(x_j)\) (equation 2.14) is the minimum average distance between the point being evaluated, \(x_j\), and all of the points not in the \(k^{th}\) cluster (represented by \(Z_j\)). \(Z_j\) can be mathematically defined as \(Z_j = \{X | X \notin c_k\}\). The number of elements in \(Z_j\) is represented by \(n_z\).

\[
b(x_j) = \min \delta(x_j, Z_j) \quad (2.14)
\]

where

\[
\delta(x_j, Z_j) = \frac{1}{n_z} \sum_{\substack{Z_j \in C \\ Z_j \notin c_k}} d(x_j, Z_j)
\]

This leads to the silhouette width for a given cluster \(k\) to be defined as \(S(c_k)\) seen in equation 2.15 which, in turn, allows for a silhouette index for the all of the clusters and therefore the
entire clustering analysis (defined by SIL) seen in equation 2.16. The SIL score can take a value between -1 and 1 where 1 reflects the ideal clustering of the data. This can be understood by remembering what the enumerator of equation 2.12 represents. For the SIL score to be close to 1 \( a(x_j) \ll b(x_j) \) meaning that the grouping of a cluster is much smaller then the lowest average distance between clusters indicating that the clustering of the data points represented their separability well. Using this same logic the further away from 1 the SIL score goes the less well the clustering represents the separability in the data.

\[
S(c_k) = \frac{1}{n_k} \sum_{X \in c_k} S(X) 
\]

\[
SIL = \frac{1}{K} \sum_{c_k \in C} S(c_k) 
\]  

\[2.16\]

2.1.2 Feature-Based

Feature based segmentation relies on features to determine differences between objects in an image. This approach allows characteristics of an object, such as texture, to be used as the guide for segmentation instead of relying on color information. One of the most common feature descriptors and one that was used for this experiment is the Scale Invariant Feature Transform (SIFT). It is a powerful feature because, as the name suggests, the feature is invariant for scale and rotation.\[27\] The process of generating a SIFT descriptor can be broken down into 4 steps (seen below).\[28\] Because this method is not usable for the data in this thesis (due to reasons that will be demonstrated in future chapters) just a brief overview is given of the method so that enough information is present to understand why the method is not usable. The key take away is that the keypoint (feature) is generated based off gradients in the image. If there are not any gradients in the object trying to be segmented, a feature based analysis will be of no use. An example of the keypoint descriptor that results from the process can seen in figure 2.5. Each of the small arrows in figure 2.5 are gradients from the pixels of the object of interest. These are used to generate an overall gradient direction and magnitude for that pixel patch. No gradients on the object would mean no small arrows in figure 2.5, making it impossible to generate the feature.

1. Scale space extrema detection
2. Key point localization

3. Orientation assignment

4. Key point descriptor

![Orientations for local points calculated and scaled.][4]

### 2.1.3 Temporal Based

A temporal based approach to segmentation uses information across time instead of just using the spatial analysis at a fixed point in time. In the context of an image sequence, the temporal element is present in the progression of frames. The higher the frame rate, the less time there will be between each frame.

A primary method to image based temporal segmentation is background subtraction, which operates on the principle of subtracting out the non-changing parts of the image (referred to as background) leaving only the moving object (referred to as the foreground) in an image sequence.[29] There are several ways to implement this concept; simple frame subtraction, running Gaussian average, temporal median filter and mixture of Gaussians.[30] For this process, the mixture of Gaussians method is chosen because it has low memory consumption.
and a low rate of complexity.\textsuperscript{29} This method’s formulation is described in the work of Zivkovic et al.\textsuperscript{31,32} It works by building a statistical model for each pixel in the background using a mixture of gaussian models. When a moving object enters the image sequence, the pixels representing the object will not fit the model for the background built in the previous frames and will be classified as foreground. For a given time \( t \), each pixel is defined as \( \vec{x}(t) \) (for the three color channel values in each pixel). The set of training data \( X_T \), contains all of the pixels for a period of time \( T \), and is defined as \( X_T = \{x^{(t)}, ..., x^{(t-T)}\} \). As new frames are input the last pixels in the set are removed (this is noted by \( X_T \) containing only data from \( (t) \) to \( (t-T) \)). Since \( X_T \) contains samples of background pixels (BG) and foreground pixels (FG) the probability of the estimated density is shown in equation 2.17.

\[
\hat{p}(\vec{x}|X_T, BG + FG) = \sum_{m=1}^{M} \hat{\pi}_m \cdot N(\vec{x}; \hat{\mu}_m, \hat{\sigma}^2_m I) 
\]  

(2.17)

In equation 2.17, \( M \) is the number of Guassian components, \( I \) is the identity matrix, \( \hat{\mu}_m \) and \( \hat{\sigma}^2_m \) are the \( m \)-th estimate of the mean and variance for the \( m \)-th Gaussian components. \( \hat{\pi}_m \) is the \( m \)-th estimate of the mixing weight. This weight represents the portion of data accounted for by that component in the Gaussian model.\textsuperscript{29} when a new data sample arrives at time \( t \), the parameters \( \hat{\pi}_m, \hat{\mu}_m, \) and \( \hat{\sigma}^2_m \) update according to equations 2.18, 2.19, and 2.20 respectively.

\[
\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) 
\]

(2.18)

\[
\hat{\mu}_m \leftarrow \hat{\mu}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m 
\]

(2.19)

\[
\hat{\sigma}^2_m \leftarrow \hat{\sigma}^2_m + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T \vec{\delta}_m - \hat{\sigma}^2_m) 
\]

(2.20)

where \( \vec{\delta}_m = \vec{x}^{(t)} - \hat{\mu}_m \)

In the update equations (2.18, 2.19, and 2.20) the constant \( \alpha \) is used limit old data’s influence. Effectively \( \alpha = \frac{1}{T} \), which creates an exponentially decaying wight envelope. The \( o_m^{(t)} \) is referred to as an “ownership” label. It is set to 1 for “close” components (defined as the smallest Mahalanobis distance between the sample and a component) and set to 0 for all the others. If there are no “close” components a new one is generated. When the maximum number of components is reached the one with the smallest weight is discarded (\( \hat{\pi}_m \)).
2.2 Image Processing

2.2.1 Morphological

According to Parker’s textbook "Algorithms for Image Processing and Computer Vision 2nd ed." [33] a Morphological operation is one that is related to the shape and structure of an object. Parker goes on to explain that the mathematics behind the morphological algorithms is related to simple set theory. The pixels in a given image can be viewed as a set where the two dimensional locations of the pixels provide the shape. Basic operations based on pattern matching can be applied to the image’s pixel set.

Dilation

One such operation is dilation. It can be represented with set theory as equation 2.21 which is understood as dilation being the union of all translations (positions) given by the structuring element (filter). [33] The practical implementation of this results in a structuring element of either four connectivity or eight connectivity (Figures 2.6a & 2.6b respectively) being passed over the image. If the center of the filter is on a pixel with a high (white) response it will convert the value of each of the pixels under the rest of the filter to high as well.

\[ A \oplus B = \bigcup_{b \in B} (A)_b \] (2.21)
An example of this can be seen in figure 3.10. Figure 2.7a shows the starting position in the dilation process. In figure 2.7b, the center of the filter is on a white pixel. Therefore, it sets all of the pixels under the filter as white (see figure 2.7c). This process continues as the filter sweeps across the image (figure 2.7d). The dilation result can be seen in figure 2.7e.
Figure 2.7: Dilation Example

(a) Dilation begins.\cite{35}

(b) Light center so fills in structure.\cite{35}

(c) Light center so continues to fill in structure.\cite{35}

(d) Dark center so does not fill in structure.\cite{35}

(e) Dilation complete.\cite{35}
Connected Components

In computer vision applications it is often necessary to determine separate objects in an image. While this can be done with color evaluation, there are often applications where objects need to be distinguished between even if they are the same color (this is the case with this experiment since most of the plant and some of the background in the images all appear to be very close in color). This is done by evaluating what pixels are connected to another within the same object. A common method of doing this is using an algorithms family called connected component labelling (CCL). The CCL method requires a binary image where the foreground pixels are given a label of 1 and the background pixels are given a label of 0. According to Hernandez-Belmonte et al.\cite{36} there are three main groups in this family of algorithms.

- Multi-pass algorithm, where multiple scans of the image are done
- Two-pass algorithm, the image is scanned only twice
- Tracing-type algorithms, uses an implicit method

The two-pass algorithm is what was used in this thesis. In the first pass, a structure element like those used in figure 2.6 can be used. It is swept over the binary image from left to right, top to bottom. Each pixel is compared with those in the structure element. If a pixel that is part of the foreground enters the structure element and it is the only foreground pixel in the structure element, it will be given a new label. If a foreground pixel enters the structure, element and there is already a foreground pixel in the structure element then the new pixel will be given the label of the foreground pixel already in the structure element. However, if there are two foreground pixels in the structure element with different labels, then a note is made in an equivalence list that the label of the two foreground pixels are the same and the new foreground pixel is assigned the label of the smallest label between the two. In the second pass all the labels marked as the same object in the equivalence list are given a final label that identifies them all as the same object. An example of what this might look like can be seen in figure 2.8.
Figure 2.8: CCL Two Pass Example
2.2.2 Median Blur

Often there is noise in an image that needs to be filtered out. The median filter is used because it is much more robust with respect to the presence of a noise spike than something like a mean filter.\cite{38} The median filter can be expressed with equation 2.22. Figure 2.9 shows an example of what this would look like in the context of an image filter. In this figure the highlighted blue value of 96 would replace the highlighted blue value of 255.

\[
MedianFilter(x_1, \ldots, x_N) = MEDIAN(\|x_1\|^2, \ldots, \|x_N\|^2)
\]  

(2.22)

2.3 Object Tracking

Once an object has been detected, it is necessary to track it through a series of frames. One way of tracking an object through time (which for this thesis is through image frames) is to use a statistically combined motion model with measured data. This will be done using a Kalman filter.
2.3.1 Kalman Filter

Welch et al.\cite{Welch1995} explains that the Kalman filter implements a predictor-corrector estimator that minimizes a converge error. The general Kalman filter does this by using a linear stochastic difference equation to relate the state at a previous time step to the current one while also modeling the “noise” in the system to factor in possible error. The state estimation equation takes the form seen in equation 2.23, where $\mathbf{x}$ is a vector of the states. In this research, the states are the position of the centroid in a polar coordinate from. The details on how the Kalman filter is set up with respect to this model will be discussed in chapter 3. The matrices $\mathbf{A}$ and $\mathbf{B}$ represent the state transformation matrix and the input transformation matrix, respectively. The state transformation matrix is used to describe how the states transform between the different time steps, represented by $k$. The input transformation matrix relates the control inputs of the system to the state variables. Process noise is added to the system by modeling it as the matrix $\mathbf{W}$.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{W}_{k-1}$$  \hspace{1cm} (2.23)

The measurements for the system can be modeled in the form of equation 2.24. In this equation, the vector $\mathbf{z}$ represents the measurement. Each measurement is related to the states with matrix $\mathbf{H}$. Similar to the process noise, measurement noise is added to the model using matrix $\mathbf{V}$.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{V}_k$$  \hspace{1cm} (2.24)

The Kalman filter uses both a priori and a posteriori knowledge to estimate the states. The notation for a priori knowledge being used is a superscript minus sign over the variables. Tables 2.1 & 2.2 hold the equation form of the Kalman filter. The equations are separated into two groups; the time update set and the measurement update set. The time update set make a theoretical prediction of what the state will be in the current time step based off how the state transforms through time and the control inputs given (equation 2.25). The covariance error matrix is also theoretically predicted based on the previous covariance error and the process noise (equation 2.26). Here it needs to be noted the process noise is now being represented by the matrix $\mathbf{Q}$ because it is can be considered constant across the time steps.
Table 2.1: Time Update Equations

\[
\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{2.25}
\]

\[
P_k^- = AP_{k-1}A^T + Q \tag{2.26}
\]

One of the key elements in the Kalman filter is the Kalman gain, \( K \). Effectively, it is a weighting factor that takes into account how confident the filter is in the a priori prediction vs the measurement. Equation 2.28 shows how it is implemented. If the gain is low, the measurement is not well trusted, and the a posteriori prediction is heavily based on the a priori prediction. However, if the measurement is trusted, its information will be incorporated into the a posteriori prediction. The Kalman gain is calculated each time step using equation 2.27. Similar to the process noise, the measurement noise is now represented by \( R \) since it can be held constant across the time index. Finally, the a posteriori covariance matrix is solved for based on the Kalman gain and a priori covariance matrix prediction (equation 2.29).

Table 2.2: Measurement Update Equations

\[
K_k = A P_k^- H (H P_k^- H^T + R)^{-1} \tag{2.27}
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \tag{2.28}
\]

\[
P_k = (I - K_k H)P_k^- \tag{2.29}
\]

To use the Kalman filter, the different transformation matrices are determined based on a state space model and the noise matrices are specified. As mentioned above, the details on these parameters for this research are discussed in chapter 3. After the transformation and noise matrices are specific the time and measurement equations are used to track the states for each time step. For each time / update step (k) the time update equations are used first to make the prediction, and the measurement equations are used to correct that prediction according to the measurement. This cycle repeats as the k increments.
Chapter 3

Experiment

This chapter will cover the setup up of the data collection rig, the details about the data process pipeline, and the specific parameters for the algorithms in the pipeline. The first element discussed will be the choice of plant for the thesis. Next, the hardware set up will be described for the experiments done in this thesis. After that, a description of the data used for the research in this thesis will be given. Next, details about the process pipeline will be discussed starting with the segmentation. This will include results from the color-based K-means clustering, an explanation why feature-based methods were not possible for this data, and temporal based segmentation using Gaussian Mixture Model background subtraction. Following the segmentation section, image processing will be discussed. This will consist of dilation and connected components analysis as well as median filtering. Finally, the details of the tracking will be discussed. This will consist of a description of the dynamics model for the tendril as well as defining the parameters for the Kalman filter.

3.1 Plant Type Choice

The Passiflora incarnanta (often referred to as the passion plant) was used for this research because it is a plant native to the Virginia area that uses a tendril to help it grow by aiding in climbing and support. As mentioned in chapter 1, the desired type of motion to study is circumnutation due to its repetitive nature. The passion plant’s tendril exhibits this type of motion while it seeks to find something to climb. Another reason this plant was chosen is because of the rate at which it seeks to find something to climb. It is exhibits a lot of
motion in a relatively short amount of time (it has been observed rotating its tendril $360^\circ$) in an hour. This makes it a good candidate for the study of plant motion’s correlation to plant health. The more active the plant, the easier it will be to determine a baseline for the typical motion of said plant. If large portions of inactivity were normal for a plant, it would make it harder to distinguish between normal behavior of the plant and its behavior during an ailment.

3.2 Hardware Setup

To develop and test algorithms it is important to be able to keep a vine tendril in frame through its whole motion cycle with as little occlusion as possible. The hardware setup up and plant size was not a variable that could be changed since the data was not taken specifically for this thesis. However, for the sake of future work, an evaluation of using both a large and a small passion plant was done. To evaluate which plant was better for observing motion a camera was mounted above each of the passion plants. One was fully mature and growing many vines, each up to 10 or 15 feet long. The other was a young passion plant that was not growing any extra vines. This entire plant only stood about 1 ft tall in the pot. The results of this evaluation will be discussed in Chapter 4. The data used in the rest of this thesis was obtained by the Spatially and Mechanically Accurate Robotic Table (SMART) at the Institute for Advanced Learning and research (IALR) in Danville Virginia. The camera used to capture the data was a Logitech C920 HD Pro using a resolution of $640 \times 480$ at a frame rate of 1 frame every 30 seconds ($0.033 Hz$).

3.3 Data

An example of a series of images for this data set is shown in figure 3.1. The plant tendril moves in a somewhat circular manner as it is trying to find something to encircle. As can be seen in subfigures 3.1b $\rightarrow$ 3.1e the tendril swings out to the side of the plant. This will be the most effective portion of the motion cycle to track the tendril because it is not completely over the rest of the plant and it is moving at its fastest velocity.

The portion of the cycle depicted in subfigures 3.1g & 3.1h will prove the most difficult since, from the camera’s perspective, the plant tendril turns into almost a single point instead of a
2D body in the image. There are additional potential challenges presented in this data. The background of the images are very noisy with many gradients and color variations. This will potentially create difficulties trying to use shape, feature, and color based algorithms. The best example of this can be seen in subfigure 3.1d. Notice how the tendril would be hard to distinguish against the silver gray background about 10° counterclockwise of its current position. This is amplified by the fact that there are not many pixels on the plant tendril. Because of this, the veins in the leaves and some of the small leaves at the end of the plant look very similar to the passion plant vine tendril (see subfigure 3.1h).

It is worth noting that while this data set is being used to develop the algorithms, specific aspects potentially unique to this data set will not be exploited whenever possible. An example of this would be the knowledge that the tendril in this data set always has a clockwise type motion when it is moving. However, the tracking will not use this knowledge outright to only look for the next position of the tendril along that motion path. This will help ensure that algorithms that show promise for this data set will potentially show success for other similar types of motion among plant tendrils.
(c) Continued clockwise rotation
(d) Continued clockwise rotation

(e) Clockwise rotation slowing
(f) Beginning to pass overhead of plant
3.4 Segmentation

The general goal for the image processing part of data processing is to get the vine tendril identified from everything else in the image. However, to make this easier the first step is to try and segment out the whole plant (minus the tendril) from the rest of the image and then to try and recognize the tendril from the background.
3.4.1 Color Based

The first step to segmenting out the plant from the rest of the image is to look at the images in different colors to see if any of the colors aid in the segmentation process. In addition to the RGB colorspace, the HSV colorspace, and L*a*b* colorspace are used. Figures 3.2 → 3.4 give an example of how the information is represented differently for the three colors.

![Red Channel](image1.png) ![Green Channel](image2.png) ![Blue Channel](image3.png)

Figure 3.2: Channels in RGB image

![Hue Channel](image4.png) ![Saturation Channel](image5.png) ![Value Channel](image6.png)

Figure 3.3: Channels in HSV image
To verify the decision to use one specific colorspace over the other two, the K-means clustering algorithm was run on the RGB, HSV, and L* a* b* colorspaces and the best result from each was chosen to compare against the others. It is important to note that for this thesis best will be defined based off of two criteria. The first is how well the data can be separated (clustered) in the colorspaces, quantified using the silhouette index described in the theory chapter. The second is how well the parts of the image of interest are segmented from the rest of the image. This is determined from inspection by looking at a representation of the image in which all of the data points are given the color of the cluster they belong in. The results from the silhouette analysis can be seen in table 3.1.

<table>
<thead>
<tr>
<th>Number of Clusters</th>
<th>RGB</th>
<th>HSV</th>
<th>L* a* b*</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.57</td>
<td>0.41</td>
<td>0.57</td>
</tr>
<tr>
<td>3</td>
<td>0.46</td>
<td>0.36</td>
<td>0.49</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.24</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>0.34</td>
<td>0.25</td>
<td>0.38</td>
</tr>
<tr>
<td>6</td>
<td>0.34</td>
<td>0.24</td>
<td>0.38</td>
</tr>
<tr>
<td>7</td>
<td>0.33</td>
<td>0.24</td>
<td>0.36</td>
</tr>
<tr>
<td>8</td>
<td>0.33</td>
<td>0.17</td>
<td>0.34</td>
</tr>
<tr>
<td>9</td>
<td>0.30</td>
<td>0.14</td>
<td>0.32</td>
</tr>
<tr>
<td>10</td>
<td>0.31</td>
<td>0.16</td>
<td>0.31</td>
</tr>
<tr>
<td>11</td>
<td>0.28</td>
<td>0.15</td>
<td>0.29</td>
</tr>
<tr>
<td>12</td>
<td>0.28</td>
<td>0.14</td>
<td>0.26</td>
</tr>
</tbody>
</table>

While RGB space was run with from 2 through 12 centers as table 3.1 indicates after 5
centers the portion of the images where so fractured between centers that the segmentation complete failed. Because of this only centers 3 through 5 are displayed in figure 3.5. The HSV and L*a*b* spaces also run from 2 centers to 12 centers as well and a sampling of the output can be seen in figures 3.6 & 3.7 respectively.

![RGB 3 centers](image)
(a) RGB 3 centers

![RGB 4 centers](image)
(b) RGB 4 centers

![RGB 5 centers](image)
(c) RGB 5 centers

Figure 3.5: K-means RGB results

![HSV 3 centers](image)
(a) HSV 3 centers

![HSV 5 centers](image)
(b) HSV 5 centers

![HSV 10 centers](image)
(c) HSV 10 centers

Figure 3.6: K-means HSV results
3.4.2 Feature Based

A feature based approach using SIFT was considered for the next step in segmenting to try to separate the tendril from the background based off of the texture on the tendril. A preliminary evaluation of what kind of texture information is present on the tendril was done by taking the derivative in the column and row directions of the images, respectively. The column derivative was calculated by convolving the Sobel operator in equation 3.1 with the grayscale data image. Similarly the row derivative can be calculated with the operator in equation 3.2. The Sobel operator was chosen because it is relatively insensitive to noise.

\[
S_{\text{col}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.1)
\]

\[
S_{\text{row}} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.2)
\]

3.4.3 Temporal Based

To segment out the tendril from the background a temporal motion based approach was taken. A Gaussian Mixture Model (GMM) was used to incorporate the technique of background subtraction without all the noise from a straight subtraction between frames. Despite
this, there was still too much noise when the tendril’s speed did not exceed that of the rest of the plant by a great enough threshold. To mitigate this, a subtraction of two different Gaussian Mixture Models was used. The GMM of the previous frame was subtracted from the GMM of the current frame. An example of what this looks like can be seen in figure 3.8. Subfigure 3.8a shows the current frame’s GMM model, subfigure 3.8b shows the previous frame’s GMM and subfigure 3.8c shows the difference between them. Note that most of the leaf motion has been subtracted out in 3.8c.

![Figure 3.8: Channels in RGB image](image)

### 3.5 Processing Segmented Image

After the segmentation process is completed, work still needs to be done to reduce the noise (non-tendril pixels) still in the image caused by pixels of the leaf being classified as foreground (rightly so since they are slightly moving). This is done using median blur, dilation, and connected components analysis.

#### 3.5.1 Median Blur

The first step in removing this noise is to blur the image with a median blur. The size of the window convolved over the image was experimented with, and the result of 3 pixel by 3 pixel window was chosen because it gave the best results. It was large enough to get rid of a majority of the speckled noise but it was not too large such that it removed the thin portion of the vine tendril. Figure 3.9 shows examples of the data before the median blur filter and after with filter sizes of 3×3 and 5×5.
3.5.2 Dilation

After the blurring is completed by the median filter to remove some of the background noise, a dilation process is done to try and connect the different portions of the vine tendril together. It is important for the next steps in the processing portion of the algorithm sequence that the tendril is a large binary body in the resulting image. However, too much dilation might cause the noise to be combined with the tendril. Figure 3.10 shows the results of different dilation kernels (subfigure 3.10d).
3.5.3 Connected Components

Once the dilation is complete, the assumption is made that the largest binary object in the image is the vine tendril. This assumption is the reason the dilation step had to be done (to grow the size of the tendril). However, even with the dilation step, this assumption is not always valid. Dealing with these instances is addressed later in the tracking section. To determine individual bodies in the binary image, connected components analysis is used. The body with the largest area in the foreground of the image is selected as the tendril and the centroid of this body is used as the point to describe its location. Figure 3.11 shows some results from this process. Subfigure 3.11a → 3.11c shows examples of the data before connected components analysis was done, the individual body identifications, and the largest body selected (respectively). Subfigures 3.11d → 3.11f are the same with a second data frame example.
3.6 Tracking

As stated earlier, the previous portion of the algorithm sequence usually preforms well at extracting the tendril from the images. However, there are times the tendril is not moving for a portion in the video sequence. Because of this, not as much or none of the tendril will be classified as foreground in the GMM subtraction. Since the goal of this thesis is to develop a set of algorithms to determine if a plant is moving this is not a problem as long as the algorithm set can reacquire the plant after it starts moving again. To accomplish this type of tracking, a Kalman filter is used. It was chosen instead of a particle filter because the system is not multimodal. Since the motion of the tendril will be modeled in such a way (described later in this section) that prevents it from being nonlinear, an Extended Kalman filter is also not required.
3.6.1 Motion Model

As mentioned previously, the model that was chosen for the tendril is one that prevents it from being nonlinear. Since the tendril motion is mostly rotational about an approximate center point in the image frame, the tendril can be modeled in polar space as a vector rotating, $\theta$, with a changing length, $r$. A graphic of this can be seen in figure 3.12.

The angle measurement is straightforward. As shown in figure 3.12, it is the measure of the angle between the positive horizontal axis and the vector to the tendril’s centroid in a counterclockwise direction. The distance measurement, $r$, is the measurement from the center of rotation (which remains fixed through the frames) to the centroid of the detected tendril body. In addition to the nature of the tendril’s motion, particularly the random angular velocity in the circular rotation, it is not possible to model the dynamics in a deterministic form. Due to artifacts from the image processing and motion of the tendril out of the image plane, the distance to the centroid also varies randomly. To model this, both the change in angular position and the change in radial distance to the centroid are sampled from normal distributions with means of 0 and standard deviations of $\frac{\pi}{80}$ rad ($2.25^o$) and 2 pixels respectively (determined experimentally). The equation formulation of this motion model can be seen in equations 3.3 & 3.4.

Figure 3.12: Polar space and sign convention for tendril motion model
\[ \theta_t = \theta_{t-1} + \theta_{\Delta t} \quad (3.3) \]
\[ r_t = r_{t-1} + r_{\Delta t} \quad (3.4) \]

where
\[ \theta_{\Delta t} \sim \text{Norm}(0, \frac{\pi}{80}) \]
\[ r_{\Delta t} \sim \text{Norm}(0, 2) \]

### 3.6.2 Kalman Filter

#### Implementation

The theory behind the Kalman filter is discussed in chapter 2. The equations given for the general Kalman filter can be simplified given the model being used for the tendril motion. Since the model of motion described above only consists of two independent states and the model to convert a previous time step prediction to the present one is just the addition of some offset, the state transition matrix, \( A \), and the input matrix \( B \), are both the identity matrix. Table 3.2 shows the time update equations in a form incorporating the model information. Equation 2.25 takes the form of equation 3.5. Experimentally the process noise covariance, \( Q \), was found to be 0.1 for both state dimensions. Given this, equation 2.26 becomes equation 3.6.

**Table 3.2: Time Update Equations Incorporating Model Parameters**

\[
\begin{bmatrix}
\hat{\theta}_k^- \\
\hat{r}_k^-
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
\hat{\theta}_{k-1}^- \\
\hat{r}_{k-1}^-
\end{bmatrix} +
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
\theta_{\Delta t} \\
r_{\Delta t}
\end{bmatrix}
\]  

(3.5)

\[
P_k^- =
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
P_{\theta_{k-1}}^- \\
P_{r_{k-1}}^-
\end{bmatrix} +
\begin{bmatrix}
1 & 0
\end{bmatrix}^T
\begin{bmatrix}
0.1 & 0 \\
0 & 0.1
\end{bmatrix}
\]  

(3.6)

In a similar manner, the measurement update equations can be parameterized as well (see table 3.3). Because the states are being observed / measured directly, the observation model, \( H \), is just the identity matrix. The measurement noise covariance was modeled as \( \frac{\pi}{50} \) (3.6°)
and 1 pixel because of the magnitude of the fluctuations in both the rotational and radial dimensions respectively. This leads to equations 2.27 $\rightarrow$ 2.29 taking the form of equations 3.7 $\rightarrow$ 3.9.

Table 3.3: Measurement Update Equations Incorporating Model Parameters

| $K_k = \begin{bmatrix} P_{\theta_k}^- & 0 \\ 0 & P_{r_k}^- \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{\theta_k}^- & 0 \\ 0 & P_{r_k}^- \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \frac{\pi}{50} & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \right) \right)$ (3.7)

| $\hat{\theta}_k = \hat{\theta}_k + \begin{bmatrix} K_{\theta_k} & 0 \\ 0 & K_{r_k} \end{bmatrix} \left( \begin{bmatrix} Z_{\theta_k} \\ Z_{r_k} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\theta}_k^- \\ \hat{r}_k^- \end{bmatrix} \right)$ (3.8)

| $P_k = \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_{\theta_k} & 0 \\ 0 & K_{r_k} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} P_{\theta_k}^- & 0 \\ 0 & P_{r_k}^- \end{bmatrix}$ (3.9)

Motion Logic

As was mentioned earlier, there are some times where the largest binary object from the connected components analysis is not the tendril. This means that the Kalman filter will be given measurement information that does not represent the tendril’s location. Because the primary time this takes place is when then tendril is not moving, the Kalman filter will converge to something other then the tendril. When the tendril starts moving again, the Kalman filter will reacquire the tendril. However, this would make it so that the lack of motion in the tendril would not be detected since the Kalman filter would just track something else when the tendril stopped. To mitigate this, the error was calculated between the measurement and the predicted location of the tendril for both the radial and rotation dimensions. If the error is past a threshold, then the measurement is considered bad data and the previous measurement is given to the filter instead. So that this does not effect the Kalman gain / change the balance of the filters trust ratio between the measurement and the motion model, the previous location prediction is also given to the Kalman filter. This effectively pauses the filter until the error measurements are back within tolerance. The exact values for the error thresholds were determined experimentally to be 0.85 radians (approximately 50°) for the rotational error and 100 pixels for the radial error.
Chapter 4

Results

This chapter will cover the results from the experiment section. It will start by discussing the results from the experiment to determine the best plant size to evaluate. Next, it will summarize the best colorspace. Following this, the results from the feature base experiment will be discussed. Next, the results from the temporal based segmentation will be evaluated. Following this, the results from the 2nd part of the process pipeline will be discussed. This will be composed of the median blur results for noise reduction, dilation results, and then connected components analysis to determine the tendril in the image. Next, the third portion of the process pipeline, tracking, will be discussed. Finally, a summary of the results will be given before leading into the conclusion chapter.

4.1 Plant Size

As described in the previous chapter, a test was done to see if a larger or smaller passion plant was better to observe for this experiment from the perspective of collecting data. The smaller plant (under 12 inches) could fit completely under the camera and the whole plant could be evaluated from above. This made it much easier to view the tendril without it being occluded by the leaves of the plant. An example of these two views can be seen in figure 4.1. The smaller passion plant allowed for all of the motion to remain in the frame without having to restrict the motion of the plant. As can be seen in figure 4.1b the larger plant’s vine had to be restrained to keep the motion of the tendril in the camera’s field of view. A consequence of this restraint was that the tendril’s motion was potentially inhibited.
Even without the restraint on the vine, the larger passion plant’s growth lead to instances were the tendril would be next to another object. Because of this, the tendril’s motion was altered when it struck an object. All of these elements lead to the conclusion the smaller plant was indeed the better specimen for a study attempting to correlate plant motion to plant health through energy expenditure.

(a) Smaller passion plant.

(b) Larger passion plant

Figure 4.1: Plant sizes
4.2 Segmentation

4.2.1 Color Based

Colorspace Transformation

As mentioned in chapter 3, the data was transformed into the L\*a\*b\* colorspace since it provided a more successful space to separate the plant from the background while still showing a difference between the tendril and the rest of the plant. Figure 4.2 gives an example of the data in each of the colorspaces.

K-means

The best results from the K-means clustering algorithm, seen in figure 4.3 verified that the L\*a\*b\* colorspace was the best out of the three to aid in the segmentation. The RGB colorspace does not even come close to segmenting out the part of the plant that is of interest from the rest of the image (see subfigure 4.3c). The HSV colorspace, subfigure 4.3b does well at grouping most of the plant into one cluster but it also includes part of the soil and a little bit of the background of the image in the segmentation. However, it has the lowest silhouette score of all of the three colorspaces (as seen in table 3.1 in a previous chapter). The L\*a\*b\* result from the clustering, subfigure 4.3a, resulted in a plant well segmented from the background. It is interesting to note that the clustering does not include the tendril in with the plant. This is what is desired because it captures the information of the difference in tendril and plant color that a person can see in the RGB image. The tendril does appear to be the same color as many parts of the background of the image. However, this was addressed with the temporal based segmentation. It is important to note that since the silhouette scores were generally not good (in the 0.35 out of 1 range) for all of the colorspaces clustering in general is not a good way to try and segment out the data. This means that a non-color based method will be used and therefore the specific color spaces used will be of less importance.
Figure 4.2: Data in different color spaces

(a) Data in $L^*a^*b^*$ colorspace

(b) Data in HSV colorspace

(c) Data in RGB colorspace
Figure 4.3: Best clustering results of the three colorspace

(a) $L^*a^*b^*$ Kmeans with 10 Centers

(b) HSV Kmeans with 6 Centers

(c) RGB Kmeans with 5 Centers
Feature Based

The results of these Sobel filters (derivatives) can be seen in figures 4.4 and 4.5. They show that there is no feature content on the tendril to use as identification with the SIFT feature descriptors. A patch of the leaf is included in the figure to give an example of an object with texture. The red rectangle on subfigures 4.4a & 4.5a shows where the leaf patch is located and the orange rectangle shows where the tendril patch is taken from. The derivative patches of the leaf have a fair amount of texture response (subfigures 4.4c & 4.5c) but the tendril patches do not have any gradient on the surfaces (subfigures 4.4d & 4.5d). This means that the feature based method is not a good approach to try and segment out the the tendril from the rest of the background of the image. A different approach will need to be used (namely Gaussian Mixture Model (GMM) background subtraction)

Figure 4.4: Example of derivative in column (left to right) direction
4.2.2 Temporal Based

Figures 4.6 → 4.9 show the results of the GMM difference subtraction. The $L^*a^*b^*$ colorspace was used since it had lower background fluctuation (which effected the output classification) and for its ability to described the differences in color described earlier in this chapter. Even with the $L^*a^*b^*$ colorspace there is still noise present due to other parts of the image moving such as the leaves around the tendrils connection point to the plant. Notice the difference in the results at each of the different locations in the motion cycle. These differences are linked to how much the vine tendril and leaves were moving (since that is what is being used to statistically build the classification criteria). The brighter / larger responses are from instances where portions of the plant were moving more between frames. The slimmer results are due to just slight motion between the frames.

Figure 4.5: Example of derivative in column (top to bottom) direction
Figure 4.6: Gaussian mixture model beginning of motion cycle

Figure 4.7: Gaussian mixture model 1/4 through motion cycle

Figure 4.8: Gaussian mixture model 2/4 through motion cycle
4.3 Processing

4.3.1 Median Blur

Figure 4.10 shows the results from the median blur. Notice how most of the noise (response that is not part of the vine tendril) in subfigure 4.10a is no longer present in subfigure 4.10b. It does not remove all of the noise, but it does remove enough to allow the dilation step to effectively amplify / magnify the tendril in the image.

Figure 4.9: Gaussian mixture model 3/4 through motion cycle

Figure 4.10: Median blur example
4.3.2 Dilation

The results of the dilation can be seen in figure 4.11. Note that the tendril in this image is broken into two different bodies (subfigure 4.11a). After the dilation process the two tendril portions have been connected together (subfigure 4.11b). Figure 4.11 also gives a good example of why the median filter in the previous section was used. Here it was not able to remove all of the noise. The reader can see that if the noise had not been reduced at all before the dilation was done it would make the result of the dilation much more likely to connect portions of noise together or to the tendril pixels causing problems with the connected components step.

4.3.3 Connected Components

The results from the connected components analysis can be see in figure 4.12. The colors in subfigure 4.12a represent the different bodies found. Due to the color spread and the number of bodies found, a few of the bodies appear to have very similar color. This is an artifact of the number of bodies to number of distinct colors and is only a problem when displaying the results since the different colors just to display what distinct bodies were found. The take away is that the tendril has clearly been identified out from the rest of the bodies as a separate, distinct object in the binary image. Subfigure 4.12b shows the largest foreground body selected with all of the other bodies removed.
4.4 Tracking

4.4.1 Kalman Filter

The results of the Kalman filter tracking can be seen in figure 4.13. The gold dot on the image represents the center of rotation. It remains constant through the frame sequence. The red dot is the measured centroid location from the largest binary object (such as the one in subfigure 4.12b). The light blue dot shows the motion model's prediction of where the centroid of the tendril should be and the dark blue shows the location from the Kalman state estimation. The data snapshots in figure 4.13 demonstrate that even when the motion model is not able to accurately predict the location the Kalman filter is able to reconverge to the correct location.

4.4.2 Motion Logic

As discussed in chapter 3, there were times that the measurement was wrong, that is, it did not even detect the tendril as the largest binary body in the image. Because of this, logic was put into place around the Kalman filter algorithm to pause the Kalman filter while the false measurements were taking place. Figure 4.14 shows two examples of the false measurements taking place. The first, seen in subfigure 4.14a was resolved with an error threshold of for the radial direction of 100 pixels. Subfigures 4.15a & 4.15b shows the difference between
(a) Kalman initialization

(b) Kalman first converging
Figure 4.13: Kalman filter tracking

(c) Kalman large motions

(d) Kalman reconverging
when the error check is in place vs without it. The second example of error, subfigure 4.14c was resolved with a rotation error threshold of 0.85 radians (approximately 50\(^\circ\)), shown in subfigures 4.15c & 4.15d).

It is important to note that implementing the rotational error threshold does cause problems at one place in the data. Figure 4.16 shows the point where the vine tendril passes directly over the plant and jumps from an angle in the range 0\(^\circ\) < \(\theta\) < 60\(^\circ\) to an angle in the range 300\(^\circ\) < \(\theta\) < 360\(^\circ\). Because of this the Kalman filter will freeze and permanently loose the tendril since the tendril does not return to this location until a full rotation has passed at which point the error repeats. If the rotational error check is removed the filter can continue to track the tendril through this portion of the motion. However, there is more error in other portions of the motion cycle when the rotational error threshold is removed. This problem can potentially be resolved with a higher frame rate in the data collection which will be discussed in chapter 5 along with other recommendations.

### 4.4.3 Results Review

After the analysis is complete it was found that the L\(^*\)a\(^*\)b\(^*\) colorspace best represented the color differences that are perceived in the original data images so it was used to represent the data in the rest of the processing. To segment out the tendril from the rest of the image, motion (temporal) background subtraction based segmentation in the form of a Gaussian Mixture Model background subtraction was used. It successful removed most of the data in the image that was not the tendril. To remove some of the noise in the image after the GMM segmentation a median blur filter was utilized. To reconstruct any breaks in the tendril a dilation process was used before a connected component analysis was used to determine the tendril body in the image. Finally, a Kalman filter was used to track the tendril through different frames and handle cases where the tendril stopped moving or was misidentified in the segmentation and processing steps.
Figure 4.14: Kalman filter false measurement examples
Figure 4.15: Kalman filter false measurement correction
Figure 4.16: Error that defeats $\theta$ threshold
Chapter 5

Conclusions & Future Work

5.1 Conclusions

This chapter will give a summary of the process pipeline used to analyze the data, summarize how the goals defined at the beginning of the thesis were satisfied, give recommendations modifications to the experimental setup, and make future work suggestions.

5.1.1 Process Pipeline Summary

To summarize the overall data analysis process, figure 5.1 shows each of the steps applied to the same image frame. The first part of the pipeline is to convert the original data image (subfigure 5.1a) to the L* a* b* colorspace (subfigure 5.1b) since it better represents the difference in color between the tendril and the rest of the plant. Next, the Gaussian Mixture Module background subtraction process is done on the L* a* b* to statistically classify the foreground (changing parts) of the image (subfigure 5.1c). Following that, noise is removed using a median filter (subfigure 5.1d). After the noise is reduced, dilation was done to amplify the tendril’s pixel count to remove any breaks between the tendril components (when possible) in the image (subfigure 5.1e). Next, a connected component analysis of the data was done to find unique bodies in the image (subfigure 5.1f). The largest body was extracted and its centroid was determined (subfigure 5.1g). Finally, a Kalman filter is used to predict if the data being measured is the tendril or if it is a false measurement, that is, one that is of noise rather than the tendril (subfigure 5.1h). It then make a prediction for the
actual location of the tendril.
(c) Find GMM difference

(d) Median blur
Figure 5.1: Summary of process pipeline
5.1.2 Goal Summary

As stated in chapter 1, the goal of this thesis was to use image processing techniques to locate the portion of the plant that is moving and of interest and then track it moving through frames in a video sequence. Below, a summary is given how these goals were met.

Locate Tendril Goal

The goal of locating the moving part of the plant that is of interested was accomplished with the ability to segment out the tendril from the rest of the image (see subfigure 5.1g). Due to the motion of the tendril it was possible to extract it from the rest of the image using statistical pixel classification and image processing techniques described throughout this thesis. While there is some error that occurs with the process in a few of the data frames, it is important to note that these errors occur when the tendril is not moving. This means that as long as the tracking algorithm can keep a good approximation of the tendril location at the stationary point while the errors are occurring, it will not affect the analysis of how the plant is moving.

Track Tendril Goal

The tracking portion of the goal was satisfied by using a Kalman filter to track the location of the plant tendril despite the errors that arose in locating the tendril in some of the frames. There are only a couple instances (see the end of chapter 4) where the Kalman filter loses track of where the tendril is located. However, even when this occurs it is only a few frames until the filter can re-acquire the tendril’s location. This means despite the error in tracking in these instances it will not greatly affect the overall tracking of the tendril motion since the tendril’s change in location is not overly significant between a few frames (since a few frames only represent a few minutes in time). The future work section will speak to how this can be further improved to reduce the amount of frames the tendril is not correctly tracked.
5.2 Future Work

5.2.1 Recommendations

Experimental Setup

To increase the ability to segment out the plant from the background it would be beneficial to make the background behind the plant a uniform color (preferably black). As shown in the results from the color based clustering in chapter 4, there were instances where color could not be used to separate the plant from the background because of the color variations in the background (this can be seen in figure 4.3). Removing these variations would open up the option of using more color-based segmentation which could, in turn, be using in conjunction with the methods used in this thesis to increase the ability to segment out the tendril.

Another improvement would be to have more pixels on the plant tendril. This would take the form of a different camera, one that allowed for higher resolution. Some algorithm sets were not feasible because there were too few pixels on the tendril.

A third improvement would be the frequency at which the data was acquired. An increased framerate would allow for fewer jumps in tendril motion. As discussed in this thesis, that would allow for the error tolerances with the Kalman filter prediction to be tighter. This would reduce the number of errors that are not currently corrected (discussed in chapter 4).

Additional Algorithms

The Condensation algorithm uses a particle filter to track while implementing information about the shape of the object being tracked.\[44\] It would be interesting to see how taking into account the shape of the tendril could assist in the tracking. Another thing to look at would be to track multiple tendrils in the same image. To do this, a multiple body tracking algorithm would need to be researched and implemented.

As discussed in chapter 3, the change for the tendril’s motion was drawn from normal distributions. Another area of future work would be to determine if there is a better way to predict the motion. One possibility would be to use the optical flow of the tendril to get a better model of the change through the previous frames. This, in conjunction with Markov chain prediction, could potentially improve accuracy in the modeling of the tendril
dynamics.

It would also be interesting to apply machine learning to try and detect the tendril in the image. Since it is not known what kind of feature space should be used, it would be good to try and train a Convolutional Neural Network to recognize the tendril. The algorithm set described in this thesis could be used to generate the training data to train the machine learning.

**Additional Plants**

A natural question to follow up this research is what other types of plants could this process be applied too? The answer to that is related to how the algorithm sequence works. For this process to work on other plants a few things would have to be true. First, the portion of the plant to be tracked would have to move more then the rest of the plant since the part to be tracked is located through a GMM (motion based) method if segmentation. A second requirement is that the motion of the plant part being track moves mostly in the 2D field of view plane of the camera without be occluded. If it moves to much out of plane the kalman filter may fail to track it. If these two conditions are met it is probable that this algorithm set can be used to track a portion of a plant moving in an frame sequence.

### 5.2.2 Final Remarks

This thesis presents a promising way to segment out and track the motion of a vine tendril from a Passiflora incarnanta for the purpose of evaluating the energy usage being expended by the plant. This provides a way for video data to be evaluated in the research of correlating plant health to plant energy expenditure.
Bibliography


