University of California
Los Angeles

Imagers as Sensors:

Using Visible Light Images to Measure Natural Phenomena

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Joshua Mark Hyman

2010

The dissertation of Joshua Mark Hyman is approved.

Mani Srivastava

Stefano Soatto

Mark Hansen

Deborah Estrin, Committee Chair

University of California, Los Angeles

2010

To my Mom . . .

who — among so many other things —

nurtured my curiosity.

—

To all the people who encouraged and

supported me along the way, especially

Dad, Jenn, Bube, Zadie, and Caitlin.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Completing this dissertation would not have been possible without the encouragement and thoughtful guidance of so many people.

I am forever grateful to my adviser, Deborah Estrin, who encouraged me to attend graduate school and who afforded me the opportunity to pursue my degree while working in industry. Her pragmatic advice and unwavering faith in me helped me see my dissertation through to completion. Quite simply, I could not have completed this work without her.

Mark Hansen, my co-adviser, taught me how to learn from data, extracting the meaning from noise. He pushed me to defend my assertions, helping me to better understand the problems at hand and more generally making me into a better researcher. Always encouraging, he helped me persevere through the toughest times. His help and guidance were invaluable.

Eric Graham's collaboration made much of my work possible. He was instrumental in identifying biologically relevant problems that would be useful to solve. He helped me preform experiments to evaluate my work both at James Reserve and on campus. I truly appreciate all of his help.

I want to thank all my lab mates from CENS (Center for Embedded Networked Sensing) who made my experience in graduate school so enjoyable. Tom Schoellhammer, though you were on your way out while I was just getting started, you took me under your wing and were always willing to lend a hand. You continue to be an excellent roll model and a great friend. Teresa Ko, your especially insightful comments really helped me to better understand my own work and the way it should be presented. I truly enjoyed working with you. Thanks to Martin Lukac, Brian Fulkerson, Andrew Parker, Sasank Reddy, Nithya Ramanathan,

Min Mun, and Donnie Kim for all of your feedback along the way.

To all the people at Google who helped me balance work and school, thank you. Greg Billock, your sage advice and general good humor made the whole process easier. Thomas Williams, you gave me a chance to be an engineer while in school, all the while nurturing my career and giving me the opportunity to succeed. To all my teammates, especially John Fisher, Steve Gerding, Rob Nelson, Joe White, Joel Ingram, and Dustin Boswell, you shouldered the burden of my time constraints, I cannot thank you enough for helping to make this possible.

Finally, I want to thank my family. Mom, Dad, Jenn, Bube, Zadie, Uncle Louis, Aunt Sharon, Shelley, Jason, and Erich, you have always been there for me, always encouraging, always confident in my abilities even when I was not. I cannot express in words how much you all mean to me. Caitlin, you supported me when I needed you most, I am forever grateful.

Thank you all.

Vita

1983	Born, Los Angeles, California, USA.
2005	B.S. Computer Science and Engineering, University of California, Los Angeles.
2008	M.S. Computer Science, University of California, Los Angeles.
2010	Ph.D. Computer Science, University of California, Los Angeles.
2005–present	Senior Software Engineer, Google Inc., Santa Monica, Ca.

Publications

J. Hyman, M. Hansen, and D. Estrin. "Estimating the Spectral Reflectance of Natural Imagery Using Color Image Features," *W*orkshop on Applications, Systems, and Algorithms for Image Sensing, 2008.

J. Hyman, E. Graham, M. Hansen, and D. Estrin. "Imagers as sensors: Correlating plant CO2 uptake with digital visible-light imagery," *W*orkshop on Data Management in Sensor Networks, 2007.

S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. "Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons From a DietSense Prototype," *W*orkshop on Embedded Networked Sensors, 2007.

Abstract of the Dissertation

Imagers as Sensors:

Using Visible Light Images to Measure Natural Phenomena

by

Joshua Mark Hyman

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2010

Professor Deborah Estrin, Chair

There exist many natural phenomena where direct measurement is either impossible or extremely invasive. The signals that biologists wish to measure about these phenomena have a variety of characteristic forms. We consider three specific forms: continuous signals (like CO_2 flux), discrete signals (like pollinator presence on a flower), or discrete spatio-temporal signals (like pollinators visiting a field of flowers). We propose using imagers as sensors by constructing a set of template procedures that uses images to obtain estimates of such phenomena.

These procedures, composed of state-of-the-art computer vision, image processing, and statistical learning and sampling algorithms, are evaluated in the context of specific applications and shown to be general through their limited assumptions. We describe various methodologies that can be used to isolate changes in the subject from changes in the local environment, making existing algorithms robust to field conditions present in real applications. Finally, we rigorously define the proposed procedures and evaluate their accuracy on real data gathered in

the field, augmented by simulation when required. Our goal is to influence future sensing system design through the identification of mechanisms that regularize the input to these procedures, making subsequent processing simpler.

For each form of signal we consider, we apply our approach to a specific application. Our procedure for predicting continuous signals is applied to the prediction of CO_2 flux from a moss plant, measurements that would otherwise require encasing the plant in an air-tight box. We consider pollinator occupancy of a flower, data that would otherwise be collected manually by humans in the field, as a representative instance of discrete signals. Scaling pollinator occupancy measurement to an entire field of flowers is the application we consider when evaluating our procedure for collecting discrete spatial temporal signals.

CHAPTER 1

Introduction

There are many important natural phenomena that traditional sensors cannot measure directly. For example, accurately measuring a plant's rate of photosynthesis (release or absorption of CO_2) requires encasing part or all of the plant in an air tight chamber. Then, the air entering and leaving the chamber are compared to measure instantaneous CO_2 flux. Such measurement is error-prone and must be frequently calibrated, making long term deployment difficult. Additionally, such an apparatus is clearly too bulky and invasive to be used in a field environment. Another similar example is counting pollinator visitations to a field of flowers. Though this could be sensed by the change in capacitance of a flower's pedals when a bee is present, such dense sampling is clearly impossible at any reasonable scale.

Visible-light imagers represent a very powerful and untapped sensing modality. Imagers are the missing input required to accurately model natural phenomena when direct measurement is difficult. Images are typically avoided in traditional sensing applications because they produce large quantities of uncalibrated data. The form of calibration required for an imager-based ecological sensor is unlike that of typical sensors; there is no conveniently accessible reference that can be used to calibrate an imager used as a CO_2 sensor, for example. We aim to use state-of-the-art computer vision, image processing, and statistical learning algorithms to build template imager calibration procedures intended for

estimating both continuous and discrete phenomena (Figure 1.1). First, image features must be extracted that are both domain relevant and immune to changing field conditions. Second, these features are used to model the signal of interest as measured in a controlled laboratory environment or a simulation based on human labeled data from the field. We will produce specific instantiations of these template procedures, evaluating them in the context of a particular biological applications.

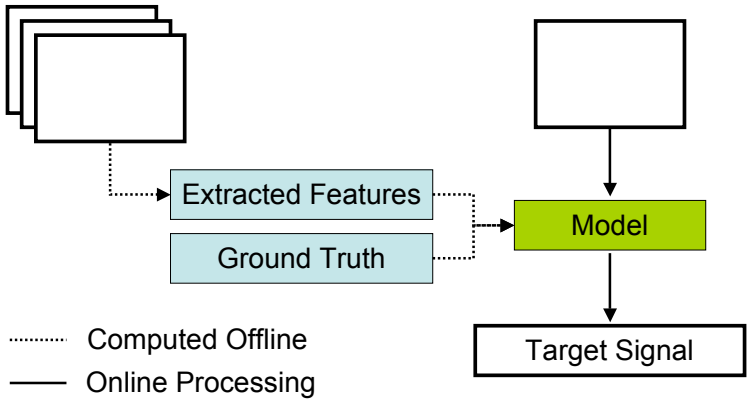


Figure 1.1: This diagram shows the two parts of imager calibration. First, field condition invariant image features are extracted from images. Second, laboratory data is acquired as ground truth for off-line the signal modeling. Together, both elements are used to calibrate the imager.

Within this framework, we endeavor to estimate three different types of signals, each requiring a different approach. Continuous signals, like plant photosynthetic respiration, require that we model the signal of interest as it changes through time from features directly or indirectly extracted from the images themselves. Discrete signals, like pollinator occupancy, require that we apply a more traditional computer vision approach to detect the presence of target objects within the scene. Finally, discrete spatio-temporal signals, like pollinator activity over an entire patch of flowers, require that we employ a sampling based approach to gather summary statistics, like event density, since direct measurement of the

entire phenomena would require too many cameras.

1.1 Properties of an Imager

An imager is any device that can capture and record the spatial distribution of light reflected from a scene. These devices typically have a number of CCD (charge coupled device) or CMOS (complementary metal-oxide semiconductor) sensors, each with a filter that allows them to be most sensitive to a particular range of wavelengths [28]. The image captured by the imager is composed from the incident illumination as it reflects off of the objects in the scene and modulated by the sensitivity of these sensors in the camera. For this work, we ignore second-order camera effects, like lens distortion, assuming that they are uniform across the image and have little influence.

We consider three classes of imagers for use as sensors: stationary still cameras, stationary video cameras, and pan-tilt-zoom video cameras. We define a video camera to be a camera capable of capturing frames with a rate of at least 1Hz. We plan to use the still camera to obtain repeat images of the same scene, but the period between exposures will be far larger, upwards of 10 minutes. The choice of camera is tightly coupled to the type of signal the phenomena of interest suggests; the finer the required measurement granularity, the higher the frame rate we must capture the imagery. The finer the required spatial resolution, the higher the density of sensors required in the camera. Further, if we desire to capture the signal over a larger region than the field of view for the camera, we must employ a camera that can be actuated to cover the area of interest.

1.2 Imager-Based Sensing Applications

There is a large class of sensing applications that can make use of imagers calibrated to estimate continuous, discrete, or discrete spatio-temporal signals. We define the subset of applications we consider using a series of suggestive questions. Though this list is not exhaustive, it describes the application characteristics leveraged by our process.

Is an imager the most natural sensor for the phenomena?

For many applications, such as detecting birds flying past an imager [54] or counting the number of eggs in a nest [53], an imager is the most natural sensor of the phenomena. Alternatively, the target signal could be logically encoded in image features that are not easily discerned by a human. Thus, using a traditional vision approach, like object detection or image segmentation, is a non-starter. This criteria defines whether the solution will employ mostly computer vision techniques, estimating a discrete signal, or statistical and signal processing techniques, estimating a continuous signal.

In this work, we consider applications of both forms; specifically, applications for which an imager is *not* the most natural sensor and applications for which an imager is the *only* reasonable sensor due to the density or duration of sensing required. When approaching applications where an imager is not the most natural approach, we consider the space of solutions that require what we call *applied vision*. This entails applying the physics-based modeling of image formation, as developed by the vision community, to help calibrate the imager and finally model natural phenomena. For applications that fundamentally require an imager because of their density

or extent, we apply traditional vision algorithms simplified by assumptions reasonable for the target application.

Is ground truth data available for field imagery?

The modeling of these systems can be greatly simplified if the ground truth data is acquired through field measurement. Though this is sometimes possible, we consider the more general case where such field measurement is not possible or requires significant human effort to label the gathered data. If ground truth data were easy to acquire, there would likely be no need for the use of an imager. Thus, having minimal or no ground truth data for field imagery is simply an intrinsic characteristic of these applications.

In these cases, laboratory experiments or short field collection combined with simulation must function as surrogates for extended data collection in the field. As such, they must be shown to sufficiently capture the space of important inputs to the ecologically process. The resulting model must be evaluated using properties of the ecological system, to place a bound on prediction error. For example, previous work [76] attempting to measure the rate of plant photosynthesis used the fact that plant growth (as measured by leaf area) is related to the integral of carbon uptake over time (the result of photosynthesis).

Which spectrum of light is measured by the imager?

Though all frequencies of light from infrared to ultra-violet are of some ecological interest, the visible range has been found to be particularly useful for measuring many phenomena [34] [50] [36] [73] [22]. In this work we confine our measurement to the visible range. This has the additional benefit of allowing for the use of commercially available digital imagers.

Current digital imagers use CCD or CMOS sensors that are most sensitive to light in the visible range, 400nm – 700nm, since their dynamic range is bandwidth limited by various filters [98]. Intrinsically, these sensors have a dynamic range that extends beyond the visible range into near-infrared as well as ultra-violet and could, in principle, be used as a sensor for those spectra as well.

Event detection or process estimation?

Interest in ecological phenomena broadly falls into two categories: event detection and process estimation. In this work, we focus on process estimation, which allows us to take advantage of a variety of application-dependent simplifying assumptions. For example, these processes typically have memory, which implies that the target signal is continuous. This suggests that the time dimension of the model's input can be used to reduce the prediction error.

In Chapter 2, we focus on the measurement of a plant's CO_2 flux, a continuous signal that is a function of the imager's output. In Chapters 3 and 4, we consider the estimation of discrete signals, such as the presence of a pollinator on a flower. It is important to note that we are still modeling a process through a sequences of discrete events. We are not attempting to detect novel anomalies, such as suspicious bag in a crowded airport terminal. Instead, we use the temporal continuity of sequential images from a video sequence, to identify the continuous motion of a particular foreground subject.

Which image features are extracted?

There are many features that could be extracted from images. Some fea-

tures, like texture, are somewhat independent of lighting by nature. Other features, like color, intensity, or radiance are significantly affected by the incident illumination. We consider a variety of potential image features, each relevant to a particular form of estimate. For continuous signal estimation, we consider both chromaticity and the relative spectral reflectance of the subject over the visible range. These features have been widely used in biological applications for everything from soil versus vegetation classification [69] to detecting the presence of clouds [94]. For discrete signal estimation, we again consider color based features, but this time in the form of foreground and background models.

Modeling a single- or multi-valued signals?

Ecological processes can be defined by a set of measurable responses to their environmental input, some of which may be dependent on one another. We focus on applications that are interested in single-valued signals that are partially dependent on various easily-measurable environmental inputs. In Chapter 4, we extend our analysis to consider applications that require summary statistics of a discrete signal to be predicted over a 2-dimensional area. Though we consider an area of larger extent, we are still functionally computing a single valued signal.

Vantage point: in-situ or remote imaging?

Remote imaging, from satellites or planes, has produced excellent insight into large scale ecosystem processes [104]. However, even with high resolution imagers, single image pixels may represent tens or hundreds of meters. As a result, their predictions are necessarily general as acquiring ground truth to calibrate more specific these measurements is difficult. We choose to focus on local measurements acquired from in-situ imagers because they

have the potential to be more accurately calibrated using laboratory experiments. The template procedures we define for imager calibration targeted at in-situ imagers is similar in spirit to processing techniques used to derive meaning from remote imagery, but necessarily differs in practice. An ecosystem leveraging the strengths of both remote and in-situ imagers would be quite fruitful as data acquired from in-situ imager deployments can then be used by the remote sensing community as ground truth to further refine their predictions. Inversely, phenomena sensed by remote imagers could direct the deployment of more dense measurement by in-situ imagers.

Are domain-relevant sensors co-located with the imager?

Ecological processes are affected by a variety of different inputs, some more easily measured than others. For example, the rate of plant photosynthesis is known to be affected by the availability of temperature, light, and moisture [76]. We can easily measure temperature and light using traditional sensors that can be deployed in the form of micro-meteorological stations. Further, by measuring these signals both in the field and during laboratory experiments, we can more easily reason about the model's accuracy. For these reasons, we choose to focus on applications that have meaningful co-located sensors in field deployments when attempting to produce continuous signals. Such co-located sensors carry slightly less importance when attempting to estimate discrete signals.

What is the expected sample frequency?

Depending on the application, ecologists can sparsely sample environments of interest (at a monthly or weekly frequency) for long durations [76], or densely sample for a short durations [5]. As a result, models based on these data collection efforts cannot make temporally dense predictions over long

period of time. We attempt to collect data at least multiple times an hour, and sometimes multiple times a second. This allows us to produce accurate predictions based on these data having a maximum resolution of hours or even as low as minutes depending on the application. Domain scientists can use this data to study an entirely new set of phenomena that occur on time scales that could not be captured previously. For example, we can measure the effect of a summer rain event on moss photosynthesis, or the pollination activity occurring to a patch of flowers. These events are known to be important, but have traditionally been incredibly time-consuming to measure.

1.3 Application Driven Innovation

Building image-based ecological sensors is a driving force for innovation in both sensor networking and computer vision. Traditionally, sensor networking has always endeavored to solve real application problems and innovate by adapting best-of-breed algorithms to the specific task at hand. Similarly, our approach to building an image-based sensor leverages the best available vision algorithms and innovates in areas where those techniques perform poorly. We develop each template procedure in the context of a particular application and discuss the minimal set of requirements of future applications that wish to instantiate our approach. We anticipate further innovation through the reuse of this process for different ecological applications.

For each of the template procedures we propose, we consider a single motivating application while carefully defining our assumptions to maintain generalizability within the class of applications we target. The study of the drought-tolerant moss *Tortula princeps* requires the estimation of its CO_2 uptake over time, a

continuous signal. This moss has the biologically interesting ability to hibernate when conditions are not favorable for its growth. Ecologists are curious as to why it is not more prevalent in dry climates for which it seems well suited. This particular problem lends itself nicely to the use of imagers because it has been shown [102] [34] that a plant's photosynthetic respiration is related to its spectral reflectance. Further, this and other moss can be reliably modeled in the laboratory and are quite representative of other higher-order plant species [79]. Previous work [76] showed that the photosynthetic respiration of plants would remain at its maximum if it weren't limited by the ambient temperature, the availability light, or moisture. Though temperature and light sensors can easily be deployed, moisture measurements are far more complex. Simple thermocouples on the surface of the plant are insufficient. Instead the plant must be destructively measured by removing it from its habitat and its weight compared against a reference dry weight. Thus, an imager is an ideal choice of sensor, for continuous measurement in the field.

The study of pollinator behavior and plant-pollinator interaction requires the collection of pollinator occupancy data for individual flower over long periods of time, a discrete signal [9]. Typically, summary statistics such as the number, duration, and species of pollinator visitations are used to understand everything from the effect of invasive plants on pollinator's behavior [5] to the evolutionary pressures pollinators place on flower diversification [20]. Temporally short data collection periods have hindered the predictive abilities of these studies due to the extraordinary cost of manual data acquisition. We aim to augment existing data collection efforts through automatic efforts to reliably capture these discrete events.

We also aim to enable data collection efforts previously unavailable to domain

scientists by scaling our discrete signal approach to cover large spatial regions. For example, Fontaine et. al. [30] studied how the density of bumblebees in a given region affects their choice in flowers. In this study's current form, data is manually collected necessitating a very short collection period and a limited collection region. We can expand the duration and area studied by collecting the required density estimates using an actuated camera. This approach enables more complex studies that would have been previously impossible.

1.4 Contributions

- **Application evaluated image-based sensor template procedures:** Define a set of procedures to correlate images to biological and ecological signals of interest using a series best-of-breed computer vision, image processing, and statistical learning algorithms. We will evaluate the prediction accuracy of this procedure in the context of specific applications, showing how to leverage intrinsic properties of that particular instantiation of the process. Using the procedures that we developed, we test our ability to measure photosynthesis of a drought-tolerant moss, *Tortula princeps*, to help ecologists understand its habitat requirements and long-term grown trends. Additionally, we test our ability to measure both pollinator occupancy for single flowers and density of occupancy for an entire field, to help biologists better understand pollinator behavior and plant-pollinator interactions.

- **Field-robust algorithms and methodology:** We have found that many of the best-of-breed algorithms make unacceptable assumptions and require modification. For example, we can not directly extract image features from field imagery because changing natural illumination has a significant effect

on the image's appearance unless these effects are reversed by ambient illumination detection and compensation. Similarly, background models fail when considering natural scenes with significant foreground motion without the addition of image registration through the introduction of an easily identified region of interest. More generally, we articulate a methodology for making algorithms robust to the field conditions present in real-world applications.

The rest of this dissertation is organized as follows. Chapter 2 considers using imagers to estimate continuous signals. A template procedure for predicting discrete signals is presented in Chapter 3. This approach is scaled to predict discrete signals over entire regions in Chapter 4. Finally, we describe avenues of future work in Chapter 5 and draw conclusions in Chapter 6.

CHAPTER 2

Predicting Continuous Signals

In this chapter, we discuss the prediction of continuous signals from imagery of natural scenes. Though specific sensors may exist to measure these signals, they require either destructive modification of the environment or brittle equipment, making long term deployment difficult. Instead, we propose using a properly calibrated imager to predict these continuous signals. To motivate our analysis, we consider a representative continuous signal: the measurable bi-product of moss photosynthesis, CO_2 flux. We leverage the fact that this signal has is strongly correlated with the color of the plant in addition to being temporally smooth. Such an imager-based sensing system allows scientists to acquire high temporal resolution field measurements of plant photosynthesis that could not be acquired previously.

2.1 Introduction

Calibrating an imager for use as a sensor requires two fundamental steps: device calibration and target signal modeling. The process we propose, depicted in Figure 2.1, is constructed from a series of models that eventually produce the signal of ecological interest. We choose this configuration both because it is suggested by the physical model of image formation and because it allows us to easily reuse existing algorithms. The output of the device calibration stage is

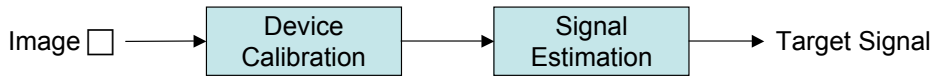


Figure 2.1: The process we propose consists of the two logical parts depicted here: device calibration and signal estimation.

either illumination invariant image features of the relative spectral reflectance of the subject. The choice is largely application specific and typically depends on the availability of ground truth data; in some cases, it may be difficult to collect accurate relative spectral reflectance data while capturing the signal of interest.

The first step in device calibration is the extraction useful features from the available images. In order to perform meaningful feature extraction, we must account for the spectral response characteristics of the CCD (charge coupled device) or CMOS (complementary metal-oxide semiconductor) sensor as well as the spectral power distribution (SPD) of the incident light. The general form of this calibration, known as color constancy [66], has traditionally been difficult. Various computer vision applications, such as object recognition and image segmentation, would benefit if such calibration could be performed accurately in general. In our applications, we are free to fix the location of the observer (relative to the subject) as well as the subject itself. In particular, we can produce accurate models of both the changing incident illumination and subject's spectral reflectance. These simplifying assumptions make this specific instantiation of the color constancy problem more tractable.

Once invariant image features have been extracted, they must be correlated to the signal of interest. Deriving such a correlation requires the construction of a model based on experimentally acquired data from imagers, and perhaps co-located traditional sensors. In the case of photosynthesis measurement, temperature, PAR (photosynthetically active radiation), and rainfall sensors are par-

ticularly useful. Including traditional sensors, in addition to the imager, has two important benefits. First and foremost, we can use sensing modalities that are correlated with the phenomena to increase prediction accuracy. Second, by incorporating field-deployable sensors into the model, we can more easily reason about the model's accuracy under field conditions.

Once devised, we must evaluate the prediction accuracy of the model. This is somewhat complicated by the fact that ground truth data is unlikely to be available in the field. By using a combination of laboratory experimentation, simulation, internal consistency checks, and other environmental cues we can leverage domain relevant information to evaluate our results. The design of the laboratory experiments is especially important. We must capture environmental signals easily measured in the field so as to corroborate the laboratory findings.

The state-of-the-art vision algorithms we use to form the various models were formulated independently. Some are formulated in a regression context, and others are formulated in a Bayesian context. Additionally, they make varying physical assumptions about the lighting, subject, and camera. As a result, their combination in our framework a bit awkward. In future work, we intend to sort out this inconsistency, placing all the stages of our procedure on consistent theoretical ground.

The remainder of the chapter is organized as follows. Section 2.2 discusses illumination modeling and estimation as well as reflectance modeling and estimation. Section 2.3 discusses the procedure for estimating a target signal through a spectrum of potential mechanisms. An overview of the proposed procedure is described in Section 2.4 and Section 2.5 evaluates this procedure in the context of an application. Work related to our approach is discussed in Section 2.6 and conclusions are drawn in Section 2.7.

2.2 Device Calibration

The purpose of device calibration is to undo the effect of changing environmental conditions on the image formation process. In particular, we would like to accurately reconstruct the relative spectral reflectance of the subject given color features extracted from an image. Formally, image formation is composed of three components: the spectral power distribution (SPD) of the incident light $E(\lambda)$, the relative spectral reflectance of the surface $S(\lambda)$, and the spectral response of the imaging device's sensor $R(\lambda)$. There are two types of spectral reflectance. Light that reflects directly off the surface is known as interface reflectance, usually seen as the spectral highlight off of a glossy surface. Light that enters the surface and interacts with colorant particles is known as body reflectance [108]. Assuming the surface is matte or Lambertian, having only body reflection, the response of the imager's *kth* sensor to a *(lighting, surface)* pair over the spectral range w is defined by Equation 2.1.

$$r_k = \int_w E(\lambda)S(\lambda)R_k(\lambda)d\lambda \tag{2.1}$$

For typical visible light imagers, $w = (400\text{nm}, 70\text{nm})$ specifying the visible range, and $k = 3$ corresponding to the red, green, and blue sensors in the imager. Since common commercial imagers intend for their output to be consumed by humans, having only three color sensors is reasonable; human color vision was determined to be a 3-dimensional space by color matching experiments [26]. That is, the use of three orthogonal sensors can represent most[1] of the gamut of human

[1]Any basis defined by human-visible colors cannot represent the entire gamut of human vision using positive coefficients. This fact can be described geometrically. The projection of the 3-dimensional human color gamut onto a plane of uniform brightness (chromaticity space) results in a convex polygon [39]. In this plane, the basis functions are represented by points, and the space of all colors represented by their linear combination using positive coefficients is a triangle. There is no triangle composed of points within a convex polygon that contain all points within that polygon.

color vision. However, we are not interested the resulting human-perceived color; instead, we are interested in $S(\lambda)$, the relative spectral reflectance of the matte surface contained in the image, and the lighting corrected image.

This formulation is a bit simplistic. In particular it doesn't capture second-order effects attributed to the camera's lens, shutter speed, and aperture. We assume that the lens' distortion is uniform across the image and that the shutter speed an aperture are set such that the sensor is not saturated. An effect we can't ignore is JPEG image compression [48]. This compression algorithm is both lossy and has a spacial component, considering multiple adjacent pixels at a time. We consider the effects of JPEG compression on this model in Section 2.5.2.

2.2.1 Modeling Illumination and Relative Spectral Reflectance

We build a 3-dimensional linear model for the surface reflectance of the subject using principle component analysis (PCA) [86]; this results in a set of basis functions $\mathbf{B}$ and their weights w. We can write this in matrix notation (Equation 2.2) if we discretize the spectral range into n bins; $\mathbf{B}$ is a $n \times 3$ matrix, w is a 3×1 weight vector, and $\hat{S}(\lambda)$ is a $n \times 1$ vector that estimates of the surface's spectral reflectance. Since we are considering outdoor ecological applications, we can apply previous work [49] that has similarly defined a 3-dimensional linear model for daylight (Equation 2.3) using PCA.

$$\hat{S}(\lambda) \approx \mathbf{B_s} w_s \tag{2.2}$$

$$\hat{E}(\lambda) \approx \mathbf{B_e} w_e \tag{2.3}$$

Initially, we build the lighting and reflectance models independently. In the future, we intend to build these models iteratively because simply modeling each independently is sub-optimal [70]. In particular, the reflectance model can be

designed to best fit the areas of most change when illuminated by different relevant spectra. Similarly, the measured spectral sensitivity of the imager can be incorporated to reduce the model's emphasis on wavelengths for which the imager has minimal sensitivity.

Once we have models for illumination and relative reflectance, we must mitigate the effect of the camera's shutter speed and aperture on $R_k(\lambda)$. Changing the shutter speed and aperture results in the image being under- or over-exposed. We assume that this effect is uniform across the sensor and that the sensor is never completely saturated (avoiding the loss of information). By using 2-dimensional chromaticity coordinates, instead of the raw 3-dimensional color coordinates, we can compensate for this uniform change in brightness. The chromaticity space is the projection of the 3-dimensional color space onto a plane of uniform brightness, and thus mitigates the effects of exposure. The chromaticity space we choose is the x and y dimensions of the xyY color space as defined by CIE [44].

$$\begin{aligned} r &\approx \hat{E}(\lambda)\hat{S}(\lambda)^T R(\lambda) \\ r &\approx (\mathbf{B_e} w_e)(\mathbf{B_s} w_s)^T \mathbf{R} \end{aligned} \tag{2.4}$$

Our resulting model for image formation (Equation 2.4), has six unknowns: the w_e and w_s weight vectors. As described, this system is under constrained since we only have two equations as defined by the chromaticity coordinates produced from the three sensors available in commercial imagers. Thus, we must estimate both w_e and w_s using the distribution of chromaticity coordinates present in the image. We proceed by estimating these values in sequence. First, we estimate w_e to produce the illuminant's spectra. Then, we transform the image to place it under a reference illuminant. From this "registered" image, we estimate w_s resulting in $\hat{S}(\lambda)$, an estimate of the subject's spectral reflectance. We assume that the same camera is used to produce all of the analyzed images, and thus the

effect of $R_k(\lambda)$ on the final pixel value is constant across all images. Thus, we need only compensate for $E(\lambda)$ when creating the registered image.

2.2.2 Estimating Incident Lighting

There are a number of lighting estimation techniques suggested by the literature [3], each making different assumptions about the lighting and subject present in the image. Since our applications may have a fixed set of possible illuminants (for example, a subset of daylight illuminants) and typically have a single subject, we would like leverage that information during lighting estimation. Depending on the nature of the application we can use either the Color by Correlation [29] algorithm or the Gamut Mapping algorithm [31] [27] [2]. The Color by Correlation algorithm assumes knowledge of both the subject as well as all possible illuminations. As a result, it is only capable of predicting illuminations that it has "seen" before. In contrast, the Gamut Mapping algorithm only assumes knowledge of image's subject. Consequently, it can predict an infinite set of possible illuminants.

By leveraging more application specific information, the Color by Correlation algorithm has been shown to slightly out-perform [40] the Gamut Mapping algorithm (both easily out-perform other more simplistic algorithms). Thus, the trade-off between these two algorithms is simply generality versus accuracy. If the set of possible illuminants can be defined, the Color by Correlation algorithm is superior. If not, we must turn to the Gamut Mapping algorithm. Since the choice of algorithm is application dependent, we present a short explanation of each here.

Color by Correlation

The Color by Correlation algorithm computes a correlation matrix representing the probability that given illuminant was present in a particular image. Each column of the matrix is a possible illuminant, and each row is the probability that a particular chromaticity coordinate would be observed for surfaces under that particular illumination. Since chromaticity can take on any real value in the range $[0, 1]$, the space is quantized to make building a correlation matrix feasible.

Producing the log-likelihood is a simple application of Bayes' rule. For a given illuminant E and a given set of observed chromaticities C_{im}, Equation 2.5 defines the probability that E was the illuminant for C_{im}. If we assume that the prior probabilities for E and C_{im} are uniform, all illuminations and surfaces are equally likely, then Equation 2.5 simplifies to Equation 2.6.

$$Pr(E|C_{im}) = \frac{Pr(C_{im}|E)Pr(E)}{Pr(C_{im})} \tag{2.5}$$

$$Pr(E|C_{im}) \propto Pr(C_{im}|E) \tag{2.6}$$

Further, we note that $Pr(C_{im}|E)$ is simply the product of the probability of observing each chromaticity c (Equation 2.7). Finally, if we take the logarithm of both sides (Equation 2.8), we get the same value as produced by multiplying the correlation matrix to a particular image's binary chromaticity vector. The binary chromaticity vector of an image is 1 for every value that is present in the image, and 0 elsewhere.

$$Pr(E|C_{im}) \propto \prod_{\forall c \in C_{im}} Pr(c|E) \tag{2.7}$$

$$log(Pr(E|C_{im})) \propto \sum_{\forall c \in C_{im}} log(Pr(c|E)) \tag{2.8}$$

There are two major shortcomings of this algorithm as suggested by Barnard et. al. [3]; both are related to the assumption that the set of possible illuminants

is fixed. First, the set of observed chromaticity coordinates may suggest that none of the illuminants are possible. Second, the algorithm cannot predict a mixture of known illuminants. To solve the first problem, they suggest smoothing the frequency distribution of the chromaticity coordinates using a Gaussian filter. However, this still requires us to train the algorithm using an illuminant set that has complete coverage of all possible illuminants. As we suggested earlier, if such a set cannot meaningfully be produced, then the Gamut Mapping algorithm is a better choice for lighting estimation.

Gamut Mapping

The Gamut Mapping algorithm assumes a set of known surfaces defined by the convex hull of their combined color gamut under a known illuminant. However, it makes no assumption about the set of possible illuminants to which those surfaces may be subjected. In this context, the gamut is defined to be the set of all color coordinates that can be produced by the given surfaces, under a given lighting, with a given camera [31]. More recent approaches [27] measure this gamut in chromaticity space making it more robust to illumination intensity. This algorithm attempts to derive a transformation (or change of basis) to map the observed gamut under unknown illumination to the measured gamut under a known illumination.

$$E_{\text{ref}} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} E_{\text{measured}} \tag{2.9}$$

These transforms, as represented by their diagonal matrices (Equation 2.9), define the change in whitepoint between the reference illumination and the unknown illumination. The whitepoint of an illuminant is the color coordinate for a

pure white Lambertian surface as viewed under that illuminant. Thus, this transform is equivalent to determining the properties of the unknown light source with respect to the reference. In general, lighting transformation matrices (as represented by Equation 2.4) are not purely diagonal. However, *von Kries coefficient law* tells us that the diagonal values are most influential [47].

Unfortunately, there is no unique transform because we incorrectly assumed that the unknown illuminant's gamut was *equal* to the measured gamut. In fact, the measured gamut is a proper subset of the unknown gamut and we have only one sample image's gamut under that illuminant. This causes there to be several transforms that map the unknown illuminant to the reference illuminant. How to chose the "best" transform from this set has been contested in the literature. All solutions involve choosing the "average" solution, which has slightly different meaning depending on the exact problem formulation [2].

Since the Gamut Mapping algorithm can potentially produce any white point as output, it is clearly more general than the Color by Correlation algorithm. However, it makes the assumption that the camera's sensors are sufficiently narrow bandwidth that Equation 2.1 can be simplified to Equation 2.10. That is, they can be modeled as impulse functions at some wavelength λ_k, typically the center wavelength of the camera's sensor.

$$r_k = E(\lambda_k)S(\lambda_k) \tag{2.10}$$

This assumption is clearly not true of typical cameras. A technique known as sensor sharpening [4] attempts to map a camera's wide bandwidth sensors to narrow bandwidth (sharpened) sensors. Additionally, von Kries coefficient law is also somewhat unrealistic. However, it has been shown [28] [111] that for "reasonable" illuminants (such as daylight), it appears to hold.

2.2.3 Changing Illumination

After we've estimated the lighting present in a given image, we must transform the images to be under some reference illuminant. We call this operation *re-lighting* the image. Since we are considering outdoor phenomena, we choose D_{65} [77] as the reference illumination; D_{65} is an approximation of daylight defined by CIE [44]. Producing a re-lighting transform when using the Gamut Mapping lighting estimation algorithm is trivial: we simply compute the reference gamut from images illuminated by a D_{65} source and use the resulting diagonal lighting transform.

Building a re-lighting transform from the output of the Color by Correlation algorithm requires that we produce a lighting transformation matrix. Like Gamut Mapping, we assume that the camera's sensors are impulse functions at the sensor's center wavelength. We can define the lighting transformation T_{light}, in terms of Equation 2.10, as Equation 2.11.

$$\begin{bmatrix} E_1(\lambda_R)S(\lambda_R) \\ E_1(\lambda_G)S(\lambda_G) \\ E_1(\lambda_B)S(\lambda_B) \end{bmatrix} = T_{\text{light}} \begin{bmatrix} E_2(\lambda_R)S(\lambda_R) \\ E_2(\lambda_G)S(\lambda_G) \\ E_2(\lambda_B)S(\lambda_B) \end{bmatrix}$$

$$T_{\text{light}} = \begin{bmatrix} E_1(\lambda_R)/E_2(\lambda_R) & 0 & 0 \\ 0 & E_1(\lambda_G)/E_2(\lambda_G) & 0 \\ 0 & 0 & E_1(\lambda_B)/E_2(\lambda_B) \end{bmatrix} \tag{2.11}$$

We choose these center wavelengths to be $\lambda_R = 620nm$, $\lambda_G = 530nm$, and $\lambda_B = 450nm$; these are close to the center wavelength of the sensors on typical digital cameras [28]. The Color by Correlation algorithm produces $E_2(\lambda)$ and we have already assumed that $E_1(\lambda)$ is the standard D_{65} illuminant. To re-light the image, we need only compute the diagonal lighting matrix and then transform each of the image's pixels individually.

This formulation only works if we specified $E_1(\lambda_k)$ and $E_2(\lambda_k)$ in absolute terms. However, the spectral power distribution of an illumination is typically normalized such that $E(\lambda_{560}) = 100$ (as is the case for the D_{65} specification). This has the effect of multiplying T_{light} by β as defined in Equation 2.12.

$$\beta = \frac{E_2(\lambda_{560})}{E_1(\lambda_{560})} \cdot 100 \tag{2.12}$$

The β term can be factored out of the resulting transformed image if we use chromaticity coordinates instead of absolute color coordinates. This is intuitively true since chromaticity coordinates are designed to be independent of brightness, the effect for which β compensates. Further, using chromaticity coordinates is a reasonable requirement as we have already leveraged chromaticity coordinates to produce a brightness invariant image for lighting estimation.

2.3 Modeling the Target Signal

There are a spectrum of available mechanism we could use to predict the target signal from the data we have collected. At one extreme, we can estimate the relative spectral reflectance of the subject, since we believe that to be correlated to the target signal, and the model the signal from that estimate. At the other extreme, we can model the target signal directly from image features after the lighting transformation has accounted for varying incident illumination. Clearly, there is a variety of other intermediate approaches where both the image features and the relative spectral reflectance can be modeled together.

Both of these approaches have merit. Using relative spectral reflectance as an intermediate is beneficial since we can measure it in the field, allowing a form of model validation not otherwise available. However, measuring this quantity in the lab may be prohibitively difficult since we would have to disturb the measurement

of the target signal itself. We can mitigate this concern by using the image features directly, but this then requires us to produce an alternate method to assess the validity of our results.

The correct approach is an application specific choice. If the data are easy to collect and the phenomena is related to relative spectral reflectance, using an RSR-based model is likely the correct approach. For our application, however, such data collection is not possible since we would loose all precision in our CO_2 measurement in the lab. Thus, we are forced to use a less structured approach based on the image features directly. What follows is a more detailed description of both approaches. Further, in Section 2.5.3 we evaluate the RSR-based approach's ability to predict RSR in the field, and we evaluate the image feature based approach's ability to correctly estimate the target signal.

2.3.1 Signal Estimation Through Relative Spectral Reflectance

To perform signal estimate based on the relative spectral reflectance of the specimen, we must first compute that reflectance. This requires estimating the weights w_s for the relative spectral reflectance basis functions B_s (see Equation 2.2) derived by using PCA. Unlike lighting estimation, however, we have less insight into the relationship between relative spectral reflectance and the chromaticity coordinates. Accordingly, we choose to estimate the parameters of our relative spectral reflectance model using non-linear regression. The input to this non-linear regression will be the 2-dimensional chromaticity coordinates. Similar to the Color by Correlation algorithm, we quantize the chromaticity space into $n \times n$ bins, using each as feature in our predictive model. These features are stable between images since we previously corrected for changes in illumination using the re-lighting transform.

Our previous work [42] showed that using this technique produced reasonable results for laboratory data. The dataset used in that work had consistent illumination since all images were taken under controlled laboratory lighting. As a result, it did not require the images to be chromatically registered using a relighting transform. Instead of using the x and y dimensions of the xyY color space, that work used the H and S dimensions of the HSV color space. Like xyY, HSV is a deformation of the RGB color space that extracts the brightness (the V dimension) from the chromaticity (the H and S dimensions). Unlike the formulation described here, the target signal was directly modeled from the quantized chromaticity.

Data about the relationship between the relative spectral reflectance, other co-located sensor, and the signal of interest can then be gathered in a laboratory experiment. Such experiments, typically suggested by the domain science, must be sufficiently realistic such that derived models have predictive power in the field. Similar to spectral reflectance estimation, we have little insight into how the spectral reflectance and corrected chromaticity relate to the target signal.

We suggest that this relationship be modeled by non-linear regression; specifically polynomial multivariate adaptive regression splines (PolyMARS) [33]. We suggest PolyMARS rather than regression trees or other non-parametric approaches since it produces continuous values as opposed to the discrete values represented by the regression tree's leaf nodes.

2.3.2 Signal Estimation Through Direct Feature Modeling

To estimate the signal from the chromaticity features directly, we again turn to non-parametric non-linear regression in the form of PolyMARS [33]. Further, we attempt to leverage domain-specific structure present in the application. For the

application we consider, we know, from studies of drought tolerant moss [85], that there are biological changes that occur as the moss prepares for a drought. Since these changes only reverse in the presence of water, it is clear that there we must model the moss as a system that has some limited memory of its previous states.

To model this memory, we provide N temporally shifted version of each input feature to the model for consideration. This has the effect of allowing the model to view the previous state of the system when constructing the current state. However, this approach has the potential to significantly increase the size of the input data for our model. To prevent this feature explosion, we first model the system without any temporally shifted features, only adding temporally shifted versions of features chosen by the first modeling pass.

Incorporating time in this fashion would appear to make our model sensitive to deviations in the sampling rate or the phenomena's rate of progression. Since we are in control of the deployment, we can ensure that the sampling rate is the same as that used to train the model. To better account for the changes in the phenomena's rate of progression, we can leverage the fact that these change only serve to stretch or shrink an characteristic curve. The characteristic shape of the temporal component can be captured using a varying coefficient model [38]. Here the temporal component is modeled as a polynomial using our various input features as the coefficients. We show in Section 2.5.3 that such an approach is not required to get sufficiently accurate predictions. As a result, we leave this to future work.

2.3.3 Model Validation

For the ecological systems we consider, there is no meaningful way to capture ground truth in the field. As a result, we must validate the predictions of our model through other means. This requirement can be partitioned into two types of validation. First, we would like to ensure that the magnitude of individual predicted measurements are accurate. Second, we want to ensure that the process we are modeling progresses at a reasonable rate through time. Any validation procedure is likely to be highly application specific. However, there are certain best-practices that can be applied in general.

Previous ecological work [76] that studied CO_2 uptake in other plants encountered a similar problem. Their solution was to relate their predictions to an expectation of plant growth. In laboratory experiments, they correlated their model's estimates with an increase or decrease in leaf count. This approach takes advantage of domain specific information to validate the model: a net CO_2 gain should result in more leaves and a net CO_2 loss should result in fewer leaves. Validating the model simply required measuring leaf count in the field and comparing to the model's prediction. An extension of this approach could ensure internal consistency of the model by measuring net CO_2 gain during a period where no leaves were lost or created. During these periods, we expect there to be a net zero gain in CO_2.

In general, this metric suggests that easily observable characteristics of the system (either visual cues or other deployed sensors) can be used to validate prediction accuracy. This seems tautological: if such metrics existed, we would use them to help model the system. However, we are interested in the absolute instantaneous value of the signal, and this type of metric essentially measures the integral of that signal's value over time.

This form of validation metric attempts to remove absolute error from our predictions but contains no time component. As suggested earlier, the processes we would like to model have memory and produce continuous signals over time. Factoring in the time component again requires application specific cues. For example, for measuring photosynthesis in moss, we know that such activity only happens while the moss is hydrated. The moss become hydrated briefly at dawn (from morning dew) and after a rain. Further, the duration of hydration mostly depends on the ambient air temperature and relative humidity. Thus, we can use the time of day or quantity of rain in addition to temperature and humidity to estimate the duration of active photosynthesis. Again, this estimates an orthogonal signal but allows us to evaluate the accuracy of our model.

A final approach is to build more realistic (less controlled) laboratory experiments. This requires that we be able to measure the signal of interest in simulated field conditions, which may not always be possible. Following the moss example, we could perform the photosynthesis measurement outside under natural light with uncontrolled (but representative) temperature and humidity. This type of experiment would provide us with measurements of the target signal that can be directly compared to the predicted values.

2.4 System Overview and Generalizability

The process we have described is shown in Figure 2.2. It makes a few specific assumptions about the subject, the camera, and the application itself. We assume that the subject of the image can be modeled as a Lambertian surface, it is matte with no spectral highlights. Though not a very restrictive assumption, we assume that the illumination and reflection can be accurately modeled; this is the case for daylight and most natural surfaces. The specifics of the lighting-

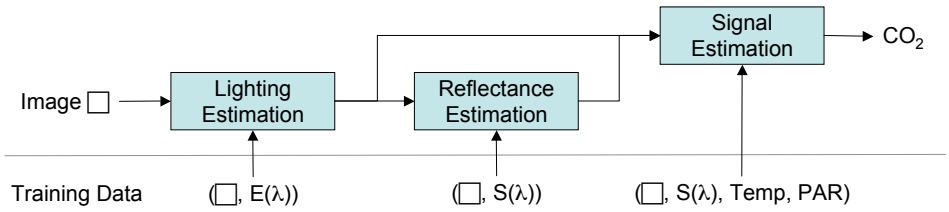


Figure 2.2: A graphical representation of Equation 2.4 with the addition of signal estimation. The models (boxes on the top row) are trained using the data depicted below each. These data are experimentally acquired. Note, we can train signal estimation with or without reflectance estimation. In this particular instantiation of the process, directly predict CO_2 uptake from image features.

related assumptions made by the illumination estimation algorithms we employ were discussed earlier. We assume all images were taken with the same camera and that the shutter speed and aperture were adjusted to avoid saturating the sensor (as is typically the case for most modern cameras). We ignore second-order camera effects, like lens distortion, assuming that they are uniform across the image and have little influence. Finally, we expect there to be other in-situ ecological sensors available that can be used both for modeling and model validation.

Our process is derived directly from the physical model of image formation, and is broken into stages. First, we estimate the lighting present in the scene using either the Color by Correlation or Gamut Mapping algorithm. Given the lighting, we can perform a change of basis to place the scene under a reference illuminant. Next, we predict the relative spectral reflectance of the surface in the transformed image. Finally, using co-located sensors and either the predicted relative spectral reflectance or directly extracted image features, we estimate the target signal using non-linear regression.

The current formulation suggests that the parameters of the lighting, reflectance, and target signal models be computed in sequence. An alternate approach would be to estimate the parameters of all three models at once. Such

(a) Hydrating the moss (at 16:45) (b) Moss actively photosynthesizing (at 16:50)

Figure 2.3: Moss at James Reserve during July 2008 after a long dry period. (a) shows us hydrating the moss and it beginning to photosynthesize in the moist areas. After only 5 minutes, much of the moss is green and photosynthesizing as seen in (b).

an approach could conceivably do equally well, but discards seemingly important information. Namely, it doesn't explicitly attempt to account for the predicted lighting, possibly causing more error in reflectance and target signal estimation. Yet, such an approach is an interesting generalization since it is able to apply domain knowledge, in the form of the individual models, without any additional supervision. We intend to evaluate this alternate approach in future work.

2.5 Application-driven evaluation: a case study

As suggested earlier, estimating the photosynthesis of *Tortula princeps*, a drought tolerant moss, is an example where imagers can become very useful sensors. Previous work [76] has produced monthly estimates of photosynthesis for plants in the field. Using field-based imagers, we can easily produce hourly photosynthesis estimates. High temporal resolution is of particular interest for this application since this moss can begin photosynthesizing mere minutes after becoming hydrated after a long dry spell (Figure 2.3).

Previous work suggests that relative spectral reflectance as well as the color of a plant is related to that plant's photosynthesis and overall CO_2 uptake [102]. This intuitively makes sense since greener plants are rich in chlorophyll, a reactive photo-pigment involved in carbon uptake [34]. Thus, we expect that the light reflected from an actively photosynthesizing plant would be related to the relative spectral reflectance of the chlorophyll molecule. In the following sections we endeavor to calibrate an imager to measure the relative spectral reflectance of this moss. We will then model the CO_2 uptake of the moss directly from image features, showing its accuracy on both lab data and under simulated field conditions.

The goal of this ecological study is to determine the effect of short summer rain events on the moss' ability to survive. Ecologists hypothesize that short summer rain events are detrimental to the moss because it causes the moss to expend more carbon than is is able to uptake. Thus, it has been suggested that this moss is capable of surviving long hot summers as long as there is minimal rain during those periods. Using the estimates provided by our imager-based sensor, we can test this hypothesis.

2.5.1 Experimental Setup

This experiment attempts to performs two related functions. First, we aim to model the relative spectral reflectance of the moss as it drys over the course of a day. Second, we aim to show we can use a camera to accurately estimate the relative spectral reflectance of a fixed subject under realistic (and changing) natural illumination. We acquired a number of moss samples from the James Reserve, seen in Figure 2.3. We hydrated the moss and allowed it to dry for approximately 6 hours, from 12pm until 6pm. We collected samples of the illumination, moss'

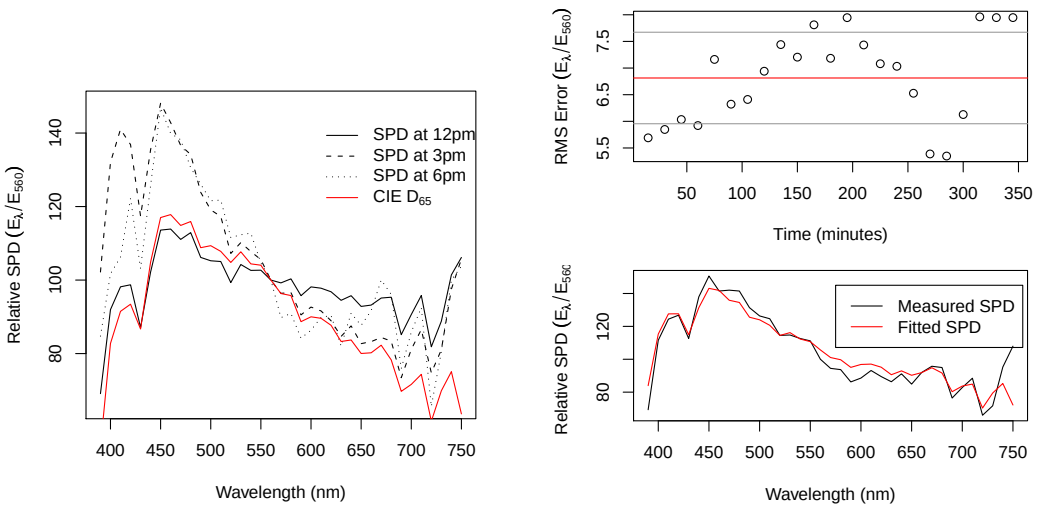
relative spectral reflectance, and images containing the moss and MacBeth Color Checker with an interval of 15 minutes. In total, 23 samples were collected. As mentioned earlier, there is a temporal component to the model validation. By allowing the moss to dry over a period, we attempt to include those temporal variation in our training data.

In order to measure both the incident illumination as well as the plant's relative spectral reflectance, we used a spectroradiometer (Licor 1800 [59]). To measure the absolute spectral power distribution of the incident illumination, we calibrated the response of the spectroradiometer using a reference tungsten illuminant, similar to the CIE A reference [44]. Similarly, the spectral reflectance of the plant was measured with respect to same tungsten illuminant. Samples of both the plant and the incident illumination were taken at 2nm increments from 390nm to 750nm.

Images of moss were taken using two standard consumer-grade cameras with their auto white-balance settings turned off. We used a Canon EOS 450D [14] to capture 10MP images in both RAW format and JPEG format; this camera represents a relatively high-end imager. Additionally, we used a Pentax Optio S5z [83] to capture 5MP images in JPEG format; this camera represents a lower-end imager. Each image contained both the moss sample as well the MacBeth Color Checker reference; this chart contains 24 color swatches of known relative spectral reflectance.

2.5.2 Imager Calibration

We verified that the illumination we measured using the spectroradiometer was reasonable by comparing it to the CIE standard D_{65} illuminant as seen in Figure 2.4(a). Each of these spectra have been normalized such that $E(\lambda_{560}) = 100$.



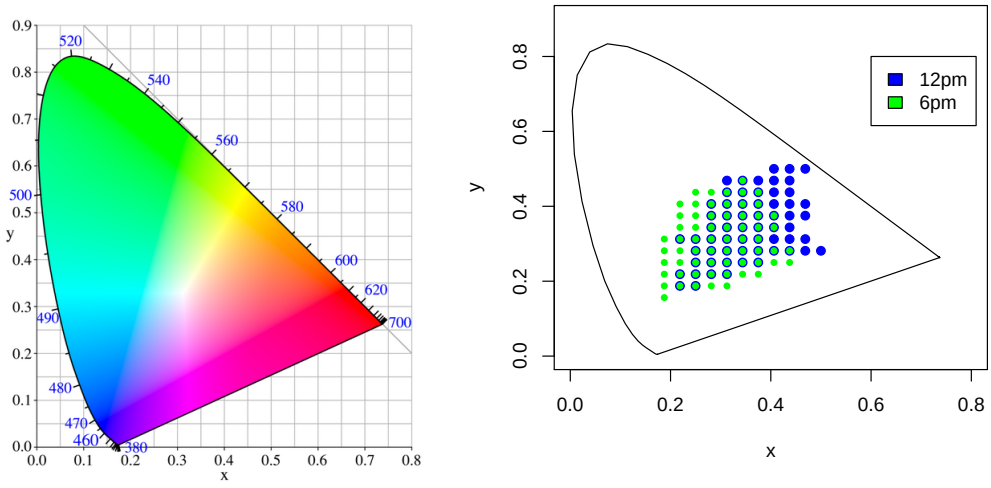
(a) Measured Illumination

(b) Illumination model accuracy

Figure 2.4: The spectral power distribution (SPD) of the illumination measured during the course of the experiment as well as the CIE standard D_{65} illuminant is shown in (a). The accuracy of the daylight model built by Judd et. al. [49] for our measured illuminants is plotted against time in (b), the red line is the mean and the gray lines are the first standard deviation. Below, we show the fit for the sample with the largest RMS error (the 21st sample at 315 minutes).

The CIE standard D_{65} illuminant is an approximation of daylight as measured in the northern hemisphere. We see that our measured spectra have the same characteristic shape as D_{65} although they are slightly bluer late in the day; they contain more power in the 400nm–500nm range than D_{65}. This similarity suggests that our measurements are producing reasonable spectra.

Using the daylight model derived by Judd et. al. [49], we computed the weights (w_e) of the basis functions (B_e), as defined in Equation 2.3, for our measured illuminants. As shown in Figure 2.4(b), the RMS error of this model does follow some time dependent trend through the course of the day. Initially, this might suggest that the model is missing some relevant information. However, we see that the fit for the example with the largest absolute RMS error still is quite good. This further confirms that our measurements are accurate. As we see in



(a) CIE Chromaticity Diagram (b) Chromaticity shift of the MacBeth Color Chart

Figure 2.5: The standard chromaticity diagram shown in (a) [110], as defined by CIE [44]. (b) shows the chromaticity shift of the MacBeth Color Checker over the course of 6 hours illuminated by daylight. Both figures are shown in the xyY color space.

Figure 2.4(a) the lighting measurements are all quite similar in form. As a result, we choose to use the Color by Correlation algorithm described in Section 2.2.2. This algorithm requires that we convert the RGB color coordinates of our images into discretized chromaticity coordinates. We choose to use x and y dimensions of the xyY color space, as defined by CIE; the chromaticity gamut defined by this chromaticity space is shown in Figure 2.5(a).

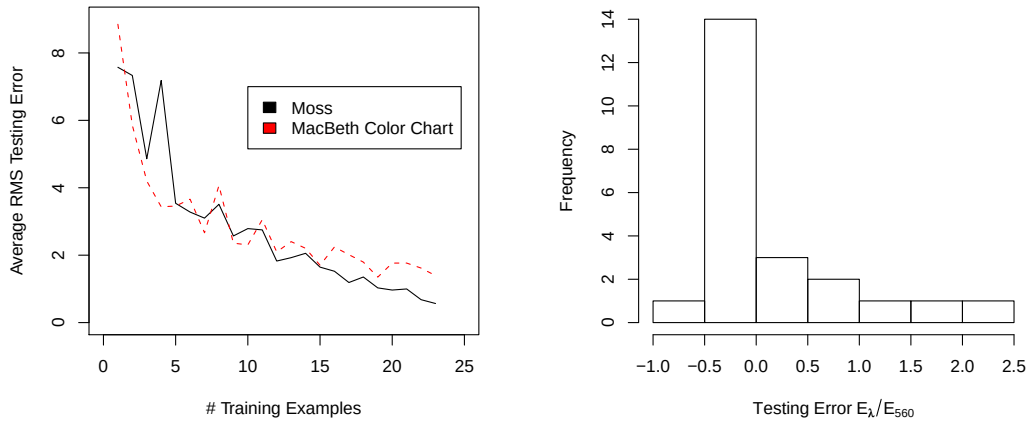
To demonstrate the color shift caused by lighting, we plot the discretized chromaticity coordinates of the MacBeth Color Chart from the first and final samples of the experiment in Figure 2.5(b). Here we have chosen to partition the xy plane into a grid of 32×32 discrete chromaticity coordinates, as suggested by [29]. Each dot represents a discrete coordinate that contains at least one pixel. Since the image's subject is fixed, the shift in color can only be attributed to

change in illumination. This is the effect we are attempting to remove.

The 2-dimensional chromaticity distributions of the sampled images are stored in matrices that we convert into row-major ordered vectors. Each vector is normalized by the number of pixels in the image and associated with the illumination measured using the spectroradiometer; these become the columns in the correlation matrix. This is a slightly simpler formulation of the Color by Correlation algorithm because we need not burden the model with the chromaticity distributions of other subjects under the same lighting; we have only one subject. To produce the log-likelihood that example image has been illuminated by particular illumination (Equation 2.8), we simply multiply the correlation matrix by the binary chromaticity vector. Recall, this binary vector is 1 for all chromaticity coordinates found in the example image and 0 elsewhere.

The training set for the Color by Correlation algorithm is selected at random from the set of experimentally obtained samples. We hand segmented the images from both cameras into an images containing only the moss and images only containing the MacBeth Color Checker. Figure 2.6(a) shows average RMS residual error between the predicted illumination and the measured illumination as a function of the training set size. Interestingly, for large training set sizes ($n >= 16$), the moss images had a slightly lower error than the images containing the MacBeth Color Chart. This is odd because the moss' reflectance is changing over time, where as the chart's reflectance is constant. In these cases approximately 70% of samples were used for training, so we believe this is simply an effect of over-training the model.

Though not shown, the model trained using both raw and JPEG images taken from the Canon camera produced similar residual error. This interesting result shows that JPEG compression has a minimal effect on the accuracy of the Color



(a) Error with changing training set size (b) Residual error with 12 training examples

Figure 2.6: Error of the Color by Correlation model derived from images of moss and the MacBeth Color Checker taken with the Pentax Optio S5z camera under varying illumination. (a) shows the reduction in error as the size of the training set increases. (b) shows the histogram of testing error on moss examples for a training set size of 12 images.

by Correlation algorithm when applied to these data. We can understand why by considering how JPEG compression works. First, it converts the image into the $YCbCr$ color space, which has two chromaticity dimensions and one brightness dimension similar to the xyY color space we used to train our model. Next, it computes the discrete 2-dimensional cosine transform of each 8×8 pixel block in the image. This produces the spatial frequency of colors within a given image block. Leveraging the fact that humans are more sensitive to lower frequency variations in color and brightness, JPEG compression discards some information about the high frequency signals, retaining most information about the lower frequency signals [48]. For both the moss and the MacBeth Color Chart, the spacial frequency in all three color dimensions is relatively low. This suggests that JPEG compression would have minimal effect on the chromaticity-based

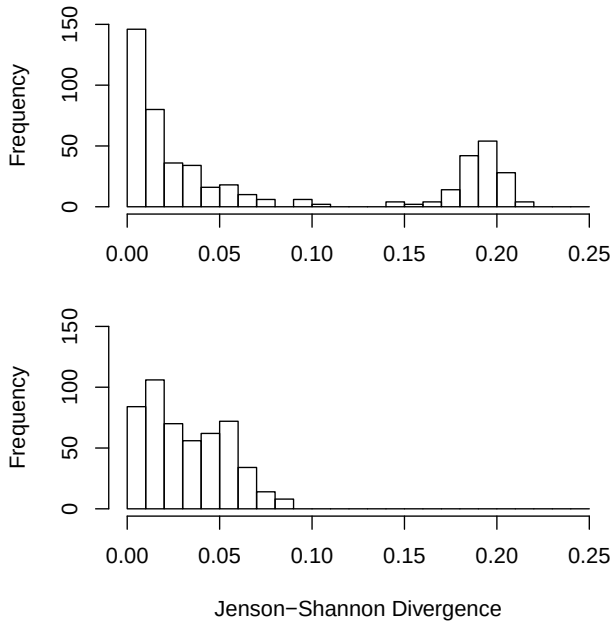


Figure 2.7: The Jenson-Shannon Divergence, before (top) and after (bottom) re-lighting, of all pairs of images containing the MacBeth Color Chart under varying daylight illumination. Optimally, all divergences would be zero after the lighting transformation.

signals we are using to build our model.

Once we obtain an accurate estimate of the image's lighting we can correct for that illuminant using T_{light} (see Equation 2.11). We test this transform on our segmented images containing the MacBeth Color Chart because its spectral reflectance doesn't change (unlike the moss). To visualize the results of this transform, we choose to compute the 2-dimensional Jenson-Shannon Divergence (a symmetrical version of the Kullback-Leibler divergence [55]) of the discretized chromaticity coordinates. We compute this divergence for all pairs of examples and expect the divergences to small since the subjects are identical.

Histograms of these divergences are shown in Figure 2.7. As we can see, the lighting transformation compresses the distributions of divergences towards zero as expected. An unfortunate consequence is that it has also increased the

variance among the previously well clustered examples. We hypothesize that this is caused by inherent error in estimating our sensors as impulse functions, a poor choice of center wavelengths, or color alterations resulting from JPEG image compression. Recall, we previously assumed that the camera's sensors were impulse functions (responding to a single center wavelength) such that we could compute a diagonal lighting transformation. In future work, we intend to try develop another algorithm, perhaps based on sensor sharpening, to further compress this divergence. Any improvement at this stage in our processing will improve the accuracy of our predictions.

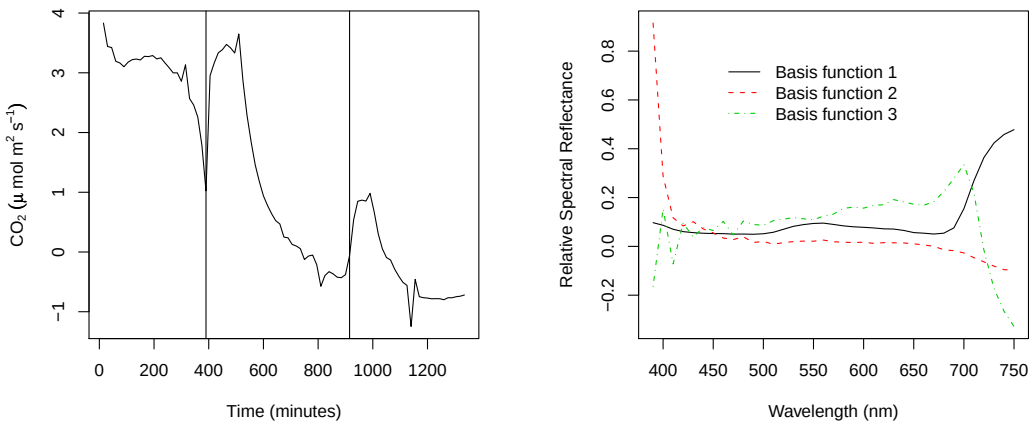
2.5.3 Estimating the Target Signal

Prior ecological analysis [36] of this moss has produced very detailed measurements of CO_2 uptake in the laboratory. The moss was placed in a chamber under controlled temperature and lighting conditions. Ambient air was drawn through a series of tubes such that some passed through the chamber as a sample, and the rest was unaltered as a control. Using an infrared gas analyzer [60], the air that had passed through the chamber was compared to the control to compute the relative increase or decrease of CO_2 in the air by volume. The instantaneous CO_2 uptake of the moss ranged between -1 μmol m^{-2} s^{-1} and 6 μmol m^{-2} s^{-1} with an error of $\pm$ 0.5 μmol m^{-2} s^{-1}. Through the experiment, a florescent light was turned on and off at 12 hour intervals to simulate day and night. During the light periods, the moss was kept at 15 °C; during the dark periods, the moss was kept at 10 °C. In addition to the CO_2 and temperature measurements, PAR measurements and images of each sample were collected at 10 minute intervals. The experiment captured the moss' progression from hydrated to dry (known as a dry down) over the course of a eight days.

In addition to a series of moss dry downs, other measurements were taken to produce an environmental productivity index (EPI) [76] for this moss plant. This index posits that there are three factors that limit a plant from gaining carbon at its maximal rate: availability of light, availability of moisture, and suitable temperature. Further, it suggests that independently measuring the effect of each dimension on the plant's respiration is sufficient to reconstruct the plant's behavior in the field. For example, temperature was varied from -1 °C to 34 °C while the moss was kept moist and well lit. Then, by simply multiplying the limiting factors together we can predict the approximate percentage reduction in CO_2 uptake as compared to the moss' maximum absorption.

The problem with using this model in the field is the lack of information about the moss' moisture levels; temperature and PAR are easily measured. As a surrogate, we intend to use the moss' relative spectral reflectance, which has been shown to be correlated with moisture as well as CO_2 uptake [34]. Thus, to leverage the data previously collected, we must compute the relative spectral reflectance of the moss in the images during the dry down. Unfortunately, if we were to apply our spectral reflectance model to compute the relative spectral reflectance for our existing datasets, we would not know if they aligned with the actual spectral reflectance values, and could not appropriately validate our results.

As discussed earlier, there are a spectrum of approaches that can be used to estimate the target signals. Since we do not have the necessary data to leverage relative spectral reflectance to model the target signal, we instead focus on direct estimation using features of the lighting corrected images. Still, we first consider the accuracy of our relative spectral reflectance estimation, since it is a compelling approach that may be usable in other applications.. Subsequently, we discuss the



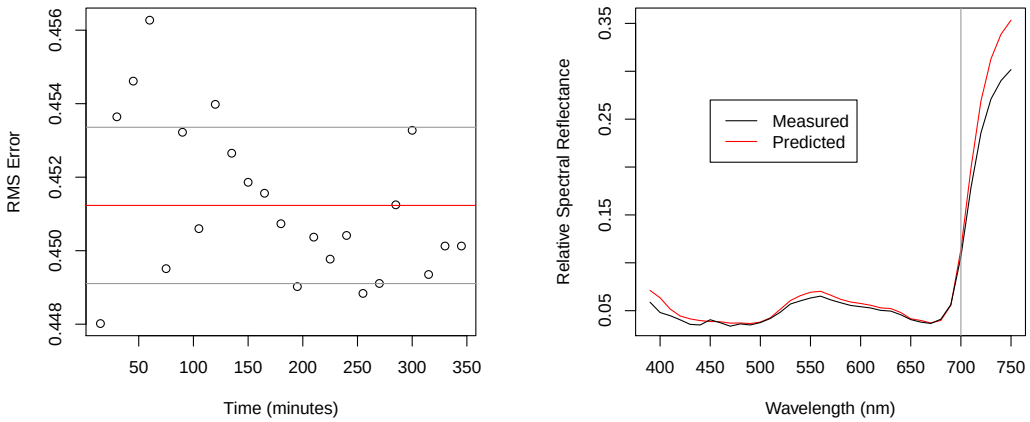
(a) CO_2 update over time

(b) RSR Basis Functions

Figure 2.8: The CO_2 response of a drying moss is shown in (a). The vertical lines represent a discontinuity in the graph where data wasn't collected for 12hrs while the moss was not exposed to light. The basis functions (b), as determined by functional PCA for the relative spectral reflectance of the moss as it dries over time.

target signal estimation accuracy of a model based solely on image features,

From the work done by Graham et. al. [36], we have six datasets, each acquired as a different moss sample dried over time. An example of the CO_2 uptake of the moss during the light periods is shown in Figure 2.8(a). The training data for our model is a set of data points randomly sampled from five of the moss sequences; in all cases, no more than 50% of the data points from any given sequences are selected. Our model is then tested on the sixth sequence in its entirety. We then perform cross-validation by rotating the training and testing sets. The our evaluation metric is the average RMS error for all possible rotations. The RMS error of the laboratory instruments is approximately 0.1 parts per million (ppm) [42], and biologists feel that 0.5 ppm is acceptable for this application.



(a) Spectral reflectance model error

(b) Predicted RSR for observation 4

Figure 2.9: The RMS residual error of the spectral reflectance predicted by our procedure is shown in (a); the red line is the average error and the gray lines are the first standard deviation. In (b) we show the predicted spectral reflectance of the observation with the largest error (observation 4 at time 60 minutes).

Estimating Relative Spectral Reflectance

After the images have been transformed we must predict the parameters of the relative spectral reflectance model (shown in Figure 2.8(b)). We have chosen to use only the first three basis functions for our model because they account for 99.96% of the variance contained in the moss' measured relative spectral reflectance. The first basis function, plotted in black, represents the average spectral reflectance across all samples. The second and third basis functions, plotted in red and green respectively, show the type of variation seen. In particular, we see that there is significant variation in the blue (400nm – 450nm) and red (675nm – 750nm) parts of the spectrum. We expect some variation near 400nm because it is near the minimum wavelength our spectroradiometer can measure. It is not clear what caused the variation around 700nm. We suspect it was due to drift in the spectroradiometer's sensors during the course of the experiment.

Given this model, we must predict weights of these basis functions (w_s from

Equation 2.4). We do this by training three regression-tree based models, one for each parameter, using the 2-dimensional chromaticity coordinates from the images previously registered by re-lighting. We trained this estimation model 12 samples, the same value which produced reasonable results for the lighting estimation. The RMS residual error of this prediction is shown in Figure 2.9(a). We can see that there is no meaningful spatial or temporal pattern in the error, suggesting the model captures most of the variation in the data. However, the magnitude of the error is rather large and we see some rather significant outliers. In comparison, the best possible values for w_s produce a mean RMS residual error of 0.0214, approximately 20 times smaller than the error produced by the spectral reflectance estimation model.

To better understand this error we plot the measured and estimated spectral reflectance for the largest outlier, sample 4 occurring at 60 minutes. As we can see in Figure 2.9(b), the fit is quite good. The vast majority of the error comes from wavelengths greater than 700nm. This error is somewhat expected since it is present in the model's basis functions as well as the original measurements. However, without evaluating the effect on the estimation of the target signal (CO_2 uptake), it is impossible to tell if this prediction accuracy is sufficient for our application. As suggested earlier, the error of this prediction can be reduced by improving the accuracy of the re-lighting transform.

A complete treatment of relative spectral reflectance estimation for use with natural imagery can be found in our prior work [43]. Since, we are unable to collect relative spectral reflectance data in the lab because it would disturb the measurement of the target signals, we have no ground truth data with which to train our model. We continue our analysis by considering the prediction of CO_2 directly from image features.

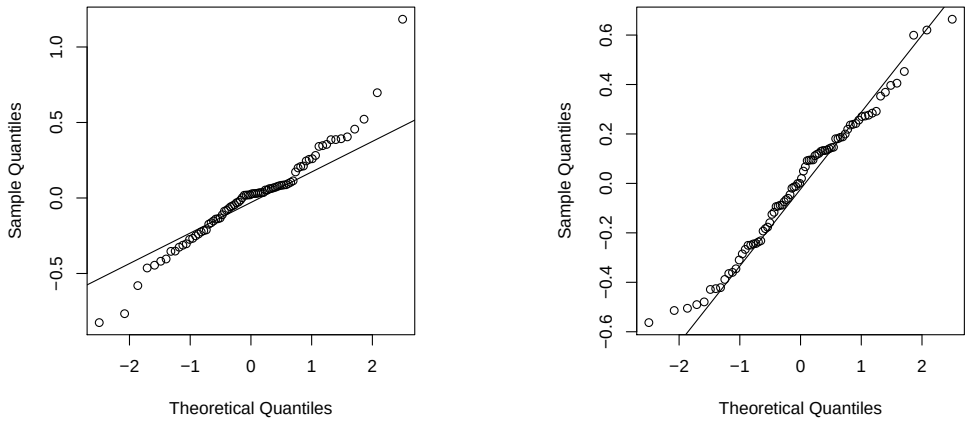
Feature	x	y	Approx. Wavelength (nm)
363	0.34375	0.375	555
364	0.375	0.375	580
395	0.34375	0.40625	545
428	0.375	0.4375	545
525	0.40625	0.53125	540

Table 2.1: The top five chromaticity features selected for modeling CO_2 along with their chromaticity coordinates and approximate wavelengths.

Using Image Features Directly

Instead of modeling the target signal indirectly through relative spectral reflectance, we choose to model the target signal directly from chromaticity features of the images. Similar to lighting estimation, we discretized the x and y dimensions chromaticity space into 32×32 buckets, each bucket representing a 0.031-unit square area of chromaticity space. The value at each of these chromaticity buckets represents the percentage of pixels whose chromaticity falls into this bucket. We then clean the data by squashing buckets containing less than 1% of the pixels to zero and only considering a single significant figure after the decimal place. This has the effect of reducing the number of anomalies present that would otherwise distract the regression algorithm.

We then trained a regression tree based model on the chromaticity features, predicting CO_2 as the response. Though this model performed poorly, it suggested that only a small number of chromaticity features, shown in Table 2.1, significantly contributed to the model. The wavelength value shown is derived from the point on the gamut boundary nearest to each feature bucket. The location of



(a) Regression Tree-based Model Residuals (b) MARS-based Model Residuals

Figure 2.11: The QQ-plot of the Regression Tree-base model's residuals show a significant deflection from normal, indicating a poor fit (a). In contrast, the QQ-plot of the PolyMARS-based model's residuals show alignment with normal, (b).

these features relative to the xy gamut is shown in Figure 2.10. Training a PolyMARS model on the same input data produced either identical features or features immediately adjacent in chromaticity space.

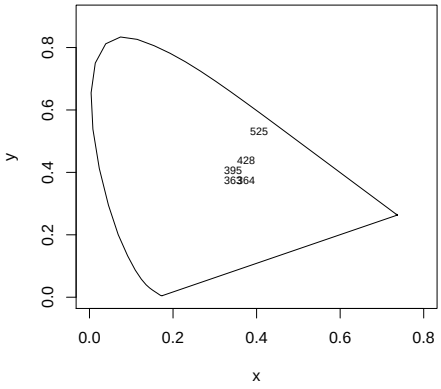


Figure 2.10: The features selected by our regression tree-based model.

To incorporate the memory present in this process, we created 10 temporally shifted versions of each of the features identified in Table 2.1; since data points were separated by 15 minutes, this resulted in a maximum temporal shift of 150 minutes. Building a new model based on these features reduced the model's average RMS error from 0.355 ppm to 0.303 ppm. Using a regression tree is suboptimal since the data are contin-

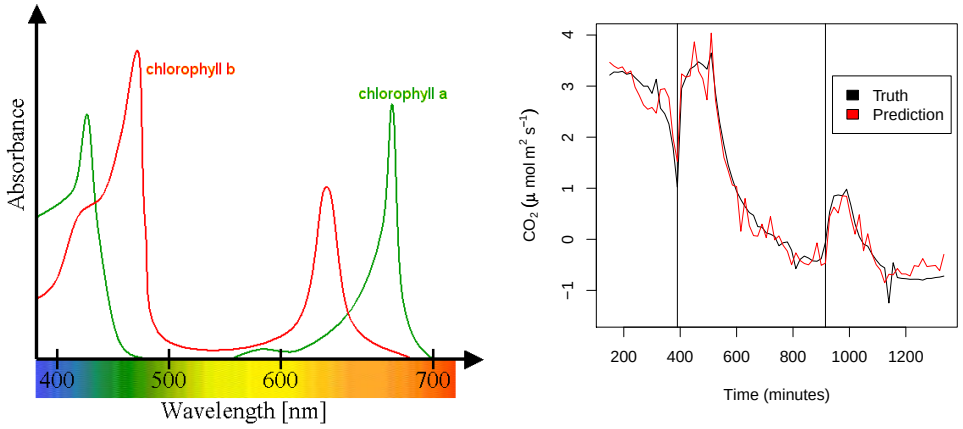
uous, and the regression tree can only predict discrete values. This is shown most obviously by the large tail in the QQ-plot of the regression tree's residuals, Figure 2.11(a). After replacing our model with PolyMARS, we further reduced the average RMS error to 0.279 ppm and reduced the heavy tails on the fit's residuals, Figure 2.11(b).

The PolyMARS-based model chose the features shown in Table 2.2; these features are shown in the order they are chosen by the model. This order roughly correlates to the significance of each feature's contribution to the model. We notice that all chosen features fall within 545nm - 580nm; this nicely matches with the spectral absorption of Chlorophyll B show in Figure 2.12(a). The biology of the moss suggests that the amount of active Chlorophyll A and B present in the plant should correlate with its CO_2 uptake [85]. As a result, we believe that this model has captured the essence of the moss' behavior.

Feature	Temporal Shift
428	none
428	-120 min
364	-120 min
525	-15 min
364	-90 min

Table 2.2: The final features chosen for modeling CO_2 and their associated temporal shift. Interestingly, only three unique features contributed to the final model.

The CO_2 uptake values predicted by our model on testing data is shown in Figure 2.12(b). The prediction (red) closely follows the ground truth data (black); since we are using PolyMARS, we are able to produce much smoother results in comparison to the discrete values produced by a regression tree based model. There are two regions where we have prediction errors: at the end of the first light period (around 300 minutes), and the end of the final light period. The initial errors are caused by slight variations in the color of the different moss as the are wet the first time; some take on a slightly deeper green. These variations

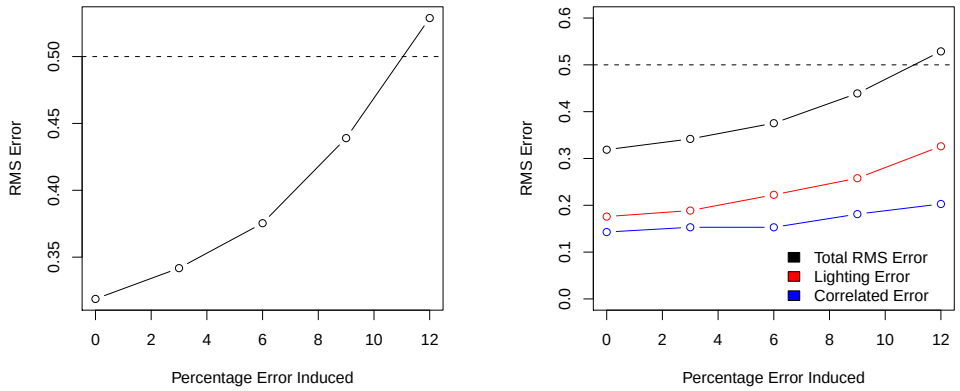


(a) Chlorophyll Spectral Absorption (b) Example Prediction

Figure 2.12: The absorption spectra of Chlorophyll A and B [109] shown in (a) nicely aligns with the features chosen in our model; that is, our model chooses features that represent wavelengths that Chlorophyll reflects. An example CO_2 prediction on testing data is shown in (b).

are naturally damped as the layer of soil below the moss absorbs the water and slowly rehydrates the moss during the second and third light periods. When the moss is nearly dry and preparing for drought, it ceases to change color even though it is still emitting CO_2. Thus, we are unable to identify this state using color alone.

Finally we test our entire procedure, end-to-end, under simulated natural lighting. Each lab experiment was conducted over the course of several days. We take the gathered imagery and transform it using Equation 2.11, applying a sequence of actual natural illuminants measured in our field experiment. To account for the error introduced by our illumination model, we apply Gaussian noise of varying intensity to evaluate it's effect our CO_2 prediction accuracy. Perturbing the spectra using Gaussian noise is a reasonable reflection of reality since atmospheric effects will result in such noise as measured by Judd et. al.



(a) Simulated Lighting Error (b) Error Breakdown

Figure 2.13: The RMS error measured when simulated lighting and images are perturbed by Gaussian noise of varying intensity, (a). The contribution of the lighting and correlated error to the total error is shown in (b). The dashed line in both figures is the error bound given by domain scientists.

[49]. Still the effect of this lighting change is nearly invertible since we are using an idealized lighting transformation. To add additional error, we compute the feature covariance matrix of moss in each dataset emitting or absorbing a similar amount of CO_2. We then apply correlated noise based on this matrix to perturb the data in a non-invertible fashion likely to be present in field imagery. Finally, following the procedure we've defined: we estimate the lighting, re-light the scene to be under a fixed illuminant (here D65), and estimate the CO_2 using a model trained on the lab data and imagery.

As we increase the amount of noise applied to the lighting and imagery, we start to see our procedure break down and the RMS error of our model rise, Figure 2.13(a). These issues stem from three places: incorrectly predicting the scene's illumination, poorly inverting the altered lighting that is present, and the model poorly compensating for the slightly altered appearance of the moss.

The least significant of these issues is the effect of correlated noise as seen in Figure 2.13(b). Since this the model was trained on five replicates, we expect that most of the variation in appearance is captured. As a result, we see that the error due to correlated noise alone is consistently less than error caused by lighting effects.

The more significant source of error was the noise applied to the incident illumination. As the magnitude of the lighting error increased, the more frequently we mispredicted the scene's illumination. Even when we did correctly predict the illumination, we would correct it with the model's illuminant that did not contain the added noise. Regardless of the cause, the error is caused by poor re-lighting resulted in a slight shift in chromaticity space of the extracted features, adversely effecting CO_2 prediction of our model.

We expect that our daylight model may induce at most a 7% error into the predicted scene lighting, as seen in Figure 2.4(b); recall, the SPD is normalized to 100 at $\lambda = 650$, though it does not have mean 100. When we apply noise resulting in 9% error, we are able to predict CO_2 with an RMS error of 0.439 ppm, which is below the domain-scientist required 0.5 ppm.

2.6 Related Work

There are a number of fields that have done work related to our image-based prediction of continuous signals. For the types of biological applications we consider, much of the work has been from the fields of agricultural engineering and environmental monitoring. Attempts to use satellite based remote imaging also have many similar characteristics, though their field-of-view is much larger and the signals they attempt to estimate are less directly tied to individual plant phe-

nomena. Here, we discuss contributions from those communities that are relevant to our work.

Agricultural Engineering

Our proposed procedure has many characteristics in common with research in the agricultural engineering field. This research attempts to use images to monitor crop health, increasing yield by detecting problems quickly. For example, detecting weed growth in crop fields has long been a problem for the agriculture industry. A review of recent literature [12] suggests that there has been a shift from using remote imagers to in-situ imagers. For example, remote imagers, typically in aircraft, have success detected weeds when the patches are dense and uniform in color, they have trouble detecting small patches because of their low resolution. In contrast, in-situ mobile imaging devices have had more success detecting smaller patches of weeds growing amongst the crops. Slaughter et. al. [96] approaches this classification problem from first principles. Like our formulation, they use the physical model of image formation in an attempt to build lighting invariant color features. Additionally, they used shape and texture features as suggested by their application.

Techniques for dealing with natural lighting conditions from a machine vision perspective are discussed in a series of works by Marchant et al. [69] [68] [67]. Set in an agricultural context, they attempt to modify existing vision algorithms to better distinguish soil from vegetation. Similar to our procedure, they choose to use the distribution of sensor values as input to their model since the possible subjects have very different spectral reflectance characteristics. In their case, the model was a binary classifier since they were interested in a binary signal. They approximate the spectra of daylight using an idealized black-body radiator

and approximate the spectral sensitivity of the camera sensors as impulse functions. A transformation was constructed, based on a ratio of sensor responses and other factors, which rendered their images sufficiently independent of changing illumination. Finally, they showed that this transform effectively separated field imagery of soil and vegetation. Unlike our goal, this and the previous system attempt to build binary classifiers of the image's subject. However, the techniques they use are applicable to our process formulation.

Environmental Monitoring

The goal of our work is similar in to much of the research in the environmental monitoring field. Unlike our work, however, typical research in that discipline performs rudimentary analysis of the images, ignoring the physical models of image formation. A representative work is the environmental monitoring system described by Crimmins et. al. [22]. It attempts to measure relative vegetation coverage by producing a "greenness" signal from a sequence of images. This signal is simply computed using the difference between the mean value of the color channels. They show that even this simple image feature tracks the increase in plant coverage over a three month period relatively well. However, this feature began to loose stability once the image became shaded by the canopy's growth. In fact, the system that inspired our work [36] used a similarly simple feature (average red-to-green ratio) to predict moss CO_2 uptake. However, they were only able to accurately predict relatively large values of CO_2 uptake. They posited that for small values of the target signal, there was less variation in the image and thus a simple average ratio was insufficient.

Remote Sensing

The remote sensing community approaches this sensing problem from a signal processing perspective. Common practice in these works has been to devise an image feature that is linearly related to the signal of interest. To some extent, this is the inverse of the environmental monitoring field's approach; instead of hypothesizing a feature using domain knowledge and measuring the correlation, they derive a feature which is defined to be well correlated. For example, the Dark Green Color Index (DGCI) [50] attempts to measure the species and health of commercial turf grass fields. The Damage Sensitive Spectral Index (DSSI) [73] tries to measure the damage to a wheat crop caused by weather or insects. By far, the most common feature is the Normalized Difference Vegetative Index (NDVI) [89] and its two close derivatives, the Soil-Adjusted Vegetation Index (SAVI) [41] and the Atmospherically Resistant Vegetation Index (ARVI) [51]. These indexes attempt to measure how much live, green vegetation is present in an image. Our work tries to strike a balance between the data-driven approach of the remote sensing community and the theory-driven approach of the environmental monitoring community. We do this by imparting structure to the procedure rooted in theory while allowing the models to adapt to the data.

Similar hybrid approaches are seen in the remote sensing literature. For example, the satellite-based Multiangle Imaging SpectoRadiometer (MISR) [61] attempts to detect the presence of clouds and cloud thickness using visible light and infrared imagery. They use image features based on radiance (the reflective characteristic of the subject) because they know that there is a significant difference between in energy reflected by land and the energy reflected by clouds. More recent analysis [94] of data produced by that satellite produced binary classifiers based on SVM to identify features that were most useful when attempting to

distinguish ice sheets from clouds; both of which have very similar radiance.

Like MISR, the river morphology measurement system developed by Legleiter et. al. [57] takes a hybrid approach to feature selection. The purpose of this system was to derive the depth of a river channel from visible-light imagery. Informed by the application, they choose to extract the log-ratio of color band pairs. These pairs were selected such that one band had much greater attenuation in water than the other. As a result, the log-ratio of these values is sensitive to the river's depth and less effected by suspended sediment. A model, calibrated against field measurements, was then derived by linearly transforming the log-ratio.

Our work leverages the best that these distinct communities have to offer, producing a image processing toolkit suitable for in-situ imagers. The features we select are domain relevant but the phenomena are modeled using non-linear techniques. The sensing system we propose takes advantage of inexpensive, readily deployable, visible light sensors. Compared to satellite- or plane-based remote imagers, they have much higher temporal and spatial resolution. Unlike the trend in the agricultural engineering community, we believe that remote and in-situ imagery can easily work harmoniously to measure natural phenomena. For example, a combined approach where data from a remote sensing applications can trigger the deployment of localized in-situ imagers would be mutually beneficial. This takes advantage of the significant strengths of each technology: the large coverage area of remote imagers, and the higher spatial resolution of in-situ imagers.

2.7 Conclusion

Driven by a motivating application, we have developed a template procedure for estimating a continuous ecologically-relevant signal, CO_2 uptake from a moss plant. At a high level, we compute the illumination present in the scene, we then re-light the imagery to be under a reference illuminant, and compute the target signal either by way of the subject's relative spectral reflectance or directly from features of the relit imagery. Though this process requires a number of assumptions of the application and environment, we have shown them to be both reasonable and common to many applications. The key innovation of this procedure is the automatic detection and reversal of ambient illumination to regularize the imagery for further processing. When evaluated under simulated natural conditions, we have shown that we can estimate the CO_2 uptake of the moss plant with an error less than 0.5ppm, making these estimate useful to domain scientists.

CHAPTER 3

Predicting Discrete Signals

In this chapter we discuss the prediction of discrete signals from video of natural scenes. We would like to detect the presence of somewhat rare, novel objects near specific, user-defined regions of interest. We leverage the highly correlated nature of temporally adjacent frames within the video sequence to both reinforce potential detections and discount potential mis-detections of novel objects. Sequences of frames thought to contain such novel objects (the target) would then be provided to the domain scientist for further study. Such a system allows domain scientists to analyze orders-of-magnitude more video data in this summarized form as compared with unaided, human analysis of the complete video.

3.1 Introduction

In a general sense, we are attempting to perform anomaly detection [16] over sequences of images. The obvious approach drawn from computer vision is object detection. Object detection considers a single image, attempting to detect if a specific object is present [81]. Simply performing object detection neglects two key aspects of our problem: we are trying to detect novel object of any form, and we are operating on video rather than unrelated still images. Though we are not necessarily interested in the location of the target within a given frame or the character of the target itself, we need a way to leverage the correlation between

frames. Thus, we choose to model our approach after object tracking [112], using techniques optimized for the specific application of interest.

The class of intended applications is characterized by novelty of the target relative to the background and the expected motion of the target. A variety of salient image features roughly correspond to object novelty; features such as color, intensity, shape, texture, and movement. These features are borne from human perception of novel objects present within cluttered scenes [45]. The expected distribution of these feature values for the target and the background helps define both which features are useful for discriminating the target from the background, and whether our modeling efforts should be focused on the target or the background. For the duration of this chapter, we focus on the requirements of a specific application: pollinator visitation. For this application, we wish detect bees as they visit an individual flower; we are not trying to detect the bee flying through the scene, just its presence on the flower itself. Though we leverage application specific characteristics to build our procedure, we argue that these characteristics are common to a variety of biologically relevant applications.

We expect that the target is not a particularly novel object relative a cluttered background; it has a poorly defined shape, elliptical from most orientations, and has little defining color or texture. Thus, we are forced to introduce a region of interest (ROI) defined by both it's scientific interest and its ability to make the target appear novel. For example, a bee does not present any novel features when compared to the cluttered background of a bush, Figure 3.1(a). However, when the bee approaches a flower, it stands out against the typically vivid colors of the flower's pedals, Figure 3.1(b). The ROI itself is expected to be easily identified relative to the background; in this example the flower is clearly visible against the cluttered natural background in Figure 3.1(a). Properly identifying the ROI has

(a) Cluttered Scene (b) ROI with target present

Figure 3.1: Against a complex and cluttered background (a), even a human observer would have trouble identifying the target. However, when restricting our view to only a important region of interest (ROI) (b), the target stands out more visibly.

the side-effect of registering the image in the face of global scene motion relative to the camera. Further, restricting our search to the ROI alone discards other objects and motion that would only distract our automated detection.

Unfortunately, even after we restrict our search to the ROI, the target is still particularly difficult to recognize because of its general lack of distinguishing features. So, instead of simply using static visual ques present in a single frame, we leverage the temporal continuity of the video sequence to identify the target using both appearance and motion. The motion of the target within the scene is expected to be erratic and very quick. In order to effectively capture it's motion globally within the scene would require a frame rate much faster than 30Hz. However, when the target approaches the ROI, we expect it's motion to become far more predictable and slow significantly. This slower motion we expect to be easily captured at 30Hz and below, speeds typically available in consumer-grade video cameras today. In our application, the pollinator moves quickly throughout the scene and isn't easily tracked, even by humans. However, once on a flower, the pollinator's motion slows significantly.

The choice of camera limits the features available for our analysis. For example, it is difficult to capture smooth motion with less than a 30Hz frame rate, and it is difficult to model texture at very low resolution. We require that the camera used is able to capture frames with at least 640x480 resolution at a minimum of 1Hz, preferably 20Hz or 30Hz. The target, when present, should occupy approximately 1% of the frame or more. Thus, the target should be at least 40x40 pixels in size.

For these applications, we focus on tracking single targets as they move about near the ROI. Though this approach can be extended to multiple targets, see Section 3.3.3, we choose to avoid the added complexity since such situations are atypical for these applications. Additionally, we anticipate only offline processing of these data; the data is collected in the field and analyzed after the fact, there is no attempt made to feedback incremental results into the sensing system itself. We consider an online system, allowing for camera actuation based feedback, in Chapter 4.

The class of sensing application we describe encompasses a wide range of biologically relevant studies. This work is immediately applicable to the study of the pollination behavior of bees in the presence of invasive plant species [5] or the study of weed seed predation by insects [75]. In these cases, an obvious ROI (here a flower or weed seed) serves to localize our search for an elusive target. Even still, this approach is appropriate for detecting novel objects or motion near any automatically extracted ROI; potential applications range from sensing movements of individually tagged fluorescing bacterial cells [91] or the detection of near-microscopic marine animals [24] to the behavioral monitoring of nocturnal fish [82] or the detection of large, migratory, wild animals [18]. In all of these cases, the target itself is sufficiently novel compared to the background that an

ROI is not needed to limit our search.

The rest of this chapter is structured as follows: Section 3.2 discusses the specific motivating application we will consider throughout our analysis. An overview of the proposed procedure is outlined in Section 3.3 and described in the following sections. A thorough evaluation of this procedure in the context of the specific application is presented in Section 3.4. Related work is discussed in Section 3.5, and conclusions are drawn in Section 3.6.

3.2 Motivating Application

The study of pollinators behavior and its effect on the local flora is a multi-disciplinary field spanning botany, horticulture, entomology, and ecology. At the core of many studies in this field is the measurement of flower occupancy by pollinators [9]. Typically, scientists are interested in the number, duration, and species of pollinator visitations to a given species of flower in the presence of other flower species. From these data they can begin to draw conclusions about the health of both the plant and pollinator populations in a given ecosystem [5].

For example, the effect of invasive plant species on the mutualistic networks that are extremely important to the diversity and stability of the plant and pollinator communities is studied by Bartomeus et.al. [5]. They seek to attribute diminished pollinator visitation to native plant species to the introduction of various invasive plants. Cooley et.al. [20] study whether flower color and shape diversification, and subsequent potential for pollinator preference, may lead to reproductive isolation among morphs of a particular flower species.

Both of these studies require the long-term collection of pollinator occupancy data. These studies define pollinator visitation to be one or more pollinators

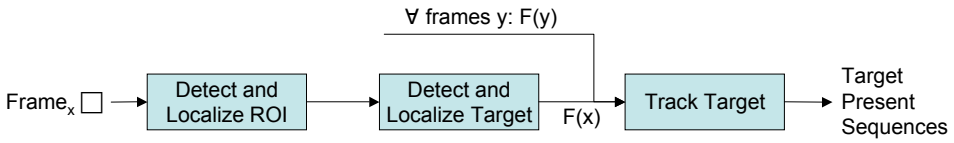


Figure 3.2: The process we propose consists of the three logical parts depicted here: detecting and localizing the region of interest (ROI), detecting and localizing the target occluding the ROI, and tracking the target across multiple sequential frames using some set of features $F(x)$ derived from image$_x$. The output of this procedure is a set of contiguous image sequences believed to contain the target of interest.

present on a particular flower. Currently, these data are acquired manually by having a human monitor the flower, watching for pollinator visitations [20]. This presents significant spatial and temporal limitations on data collection. The collection procedure outlined in these representative studies suggest that data can only be collected for 30 minutes durations and can only cover a small area much less than 1m^2. The procedure we propose can extend the duration of data collection for a small cluster of near-by flowers to the entirety of a day. We consider increasing the spatial extent of the data collection in Chapter 4.

3.3 Procedure

We propose a multi-stage procedure to process the incoming video imagery and produce occupancy data that can be consumed by scientists. This procedure is designed to be tailored to the specific application of interest, though the procedure itself is general. Further, we attempt to minimize the user input required to extract the biological signal of interest.

The procedure, as depicted in Figure 3.2, roughly consists of three independent pieces. For each frame, we must first automatically localize the ROI specified by the user. This effectively discards a large fraction of each frame that is unin-

teresting, by definition, and would only serve to confuse further processing. Next, we detect and localize the target as it occludes the ROI. We consider both modeling the target itself, by way of its visual appearance or its motion, and modeling the background, looking for occlusions that imply a foreground object. Finally, using the likelihood and location of a potential match produced by the target detection, we follow the target's motion over time looking for discontinuities since the target's motion is expected to be smooth; this leverages the inherent temporal correlation between frames to discard false detections.

This particular segmentation of the problem is suggested by computer vision literature. Detecting the ROI and limiting our further processing to that region is a form of translational image registration [35]. Registration is a clearly separable operation that serves to provide stable data to the following operations; forcing subsequent stages to detect and compensate for such motion would only serve to further complicate their requirements. Separating target detection from target tracking is also a common approach in the literature [112]. Many tracking algorithms simply presuppose that target locations are known [106] while others suggest specific, but independent, detection methods [7].

Though separate in principle, these stages clearly have significant interaction; later stages must compensate for any errors made in earlier stages. Many statistical approaches exist to model these interactions globally using constructs like joint likelihood filters, joint probabilistic data association, or multiple hypothesis filters [95]. These approaches attempt to simultaneously detect and track objects, effectively feeding back trajectory information to aid detection; more details can be found in Section 3.5. We assert that such complex approaches are unnecessary for our application. Instead, we propose that incremental improvement of each stage with minimal compensation effort in following stages produces sufficiently

accurate results for our applications, as shown in Section 3.4.

Humans are particularly adept at detecting specific interesting objects and noticing motion in a complex scene. Though computer vision attempts to mimic the abilities of human vision, it's capabilities are still far from comparable. Thus, we attempt to define heuristics of potential applications required for successful detection and tracking. First, the ROI should be obvious to an untrained human when only given approximately 250ms to view the scene, and it should occupy at least a 150px square area (approximately 10% of a 640x480 image). Second, the target should be obvious to an untrained human when viewing frames of the ROI progressing at 1Hz. Finally, the data should be captured at a minimum of 1Hz for background modeling and a minimum of 20Hz is required for foreground modeling.

The solution we propose is very similar to those in the literature. What differentiates our approach is its simplicity, ease of debugging failures, and is application aware formulation. Our procedure is designed to solve a specific class of biologically relevant data collection problems, allowing it to leverage application specific characteristics to improve performance.

3.3.1 Detect and Localize the Region of Interest

When we consider the natural imagery depicted in Figure 3.3, it is clear that spotting a bee against the cluttered background is a near impossible challenge. However, spotting a bee against the brightly colored flower is more feasible. This foreground object, which we define to be the region of interest (ROI), is characterized by it's distinct visual difference from the background clutter and movement that is unrelated to the final detection problem (here locating a pollinator). Further, we expect that the ROI is present in all frames of the video sequences. Our

Figure 3.3: We define the flower to be the region of interest. Considering only this object allows us discard a significant fraction of the frame that we deem to be uninteresting.

(a) Frame A (b) Frame B (c) Frame C

Figure 3.4: Three near-by frames in the image sequence depict the significant motion of the region of interest (the flower).

goal is to automatically detect, localize, and crop this ROI in every frame in the image sequence. This operation allows us to discard a large fraction of the image would only serve to confound our later detection problem. Finally, it has the added benefit of reducing the computational expense of future analysis by significantly reducing the size of the image.

Figure 3.4 illustrates the motion of the ROI over within a single image sequence; significant foreground and background motion is present. The foreground motion can be characterized as translation within the 2D viewing plane, thus we consider this to be analogous to translational registration [35]; we are simply attempting to find the appropriate translation of the ROI that results in sequential images being aligned and cropped appropriately. The magnitude of spatial displacement of the ROI between adjacent frames is modeled as a Normal distribution (Equation 3.1).

$$\begin{aligned} x_{t+1} &= N(\mu = 0, \sigma_x) + x_t \\ y_{t+1} &= N(\mu = 0, \sigma_y) + y_t \end{aligned} \tag{3.1}$$

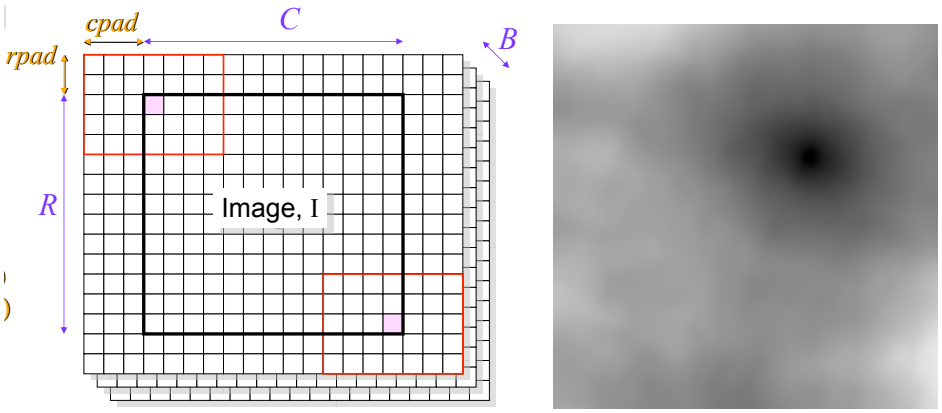
Using this formulation, we choose to apply a technique called template matching [13] to detect and localize the ROI based on a template image representing the ROI. In principle, we can use this formulation to decrease the computational

complexity of template matching by a constant factor in the average case. However, worst case complexity would be unchanged. Instead, we use this formulation to rank potential matches found in the example image. Potential ROI matches closer to the location of ROI in the previous frame are preferred over more distant potential matches. This helps us avoid transient aberrations that may otherwise reduce our accuracy. A similar formulation is used to track the foreground target in Section 3.3.3.

Though not required for the considered applications, we could apply the Hough transform [1] or a Scale-Invariant Feature Transform (SIFT) [62] to this template image to account for 2D in-plane rotation. As its name implies, SIFT can also tolerate changes in scale, and using more template images containing the ROI from various poses, it can also account for 3D out-of-plane rotation.

The user typically acquires the template image by cropping the ROI from the first frame of the image sequence. Given this template image, which is expected to be a sub-image fully or partially contained within a given example image, the template matching algorithm computes some similarity measure between the template and the example image for for all possible translations of the template image depicted in Figure 3.5(a). The similarity measure, parameterized by the location of the template image relative to the example image, is computed for all aligned pixels and applied to each color channel. The maximum likelihood estimate for the correct alignment of template and example image is located at point of maximum similarity.

The template is never exactly, pixel for pixel, present in the example image. The example image always includes some distortion due to lighting or lens effects. Instead, we model the instance of the template present in the image as 0-mean



(a) Template Matching (b) Example Match

Figure 3.5: A graphical depiction of template matching is shown in (a). Here the red rectangle represents the template, and the black rectangle (size RxC with B color bands) represents the example image. The image is padded with empty pixels so that all possible template translation can be attempted (Figure from [84]). An example of a template matched against an example image plotted as a heat map is illustrated in (b); darker implies more similar, and the dark spot in the upper right represents the flower in the image.

Gaussian noise added to the template (Equation 3.2).

$$T_I(x, y) = N(\mu, \sigma) + T(x, y) \tag{3.2}$$

We expect the similarity measure, or match value, to have two distinct regimes corresponding to complete mis-alignment and near-alignment. When there is no alignment, we expect the match value to be small, implying dissimilarity. This can be seen in the random fluctuations in the similarity metric depicted in Figure 3.5(b). Once the template is nearly aligned, we expect the match value to be large, implying similarity. It is important to note that we can only consider this similarity metric to imply likelihood when the template is nearly aligned [13]. When the template is sufficiently novel in comparison to the majority of the example image, simple thresholding can differentiate these two regimes.

The simplest implementation of template matching uses convolution. By defi-

nition, this implies that the similarity measure is the inner product, or unnormalized cross-correlation, of the color values at corresponding pixels (Equation 3.3). The best template alignment $\hat{\theta}$ is located at maximum value of S (Equation 3.4).

$$S = T * I$$

$$S(x', y') = \sum_{x',y'} T(x', y') \cdot I(x + x', y + y') \tag{3.3}$$

$$\hat{\theta} = \underset{x,y}{\operatorname{argmax}}\ S(x, y) \tag{3.4}$$

Template matching using convolution can be efficiently implemented using Fast Fourier Transforms as shown in Equation 3.5 [74].

$$S = \mathcal{F}^{-1}(\mathcal{F}(T) \cdot \mathcal{F}(I)) \tag{3.5}$$

This process is performed on each color channel individually and summed. The main draw back of this approach is the fact that matches are not normalized. As a result, variation in image intensity significantly affects the match value.

Obvious alternatives to using unnormalized cross-correlation as a similarity measure are normalized cross-correlation (Equation 3.6)

$$S(x, y) = \frac{\sum_{x',y'} T(x', y') \cdot I(x + x', y + y')}{\sqrt{\sum_{x',y'} T(x', y') \cdot \sum_{x',y'} I(x + x', y + y')}} \tag{3.6}$$

and normalized Euclidean (L2) distance (Equation 3.7).

$$S(x, y) = \frac{\sum_{x',y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \tag{3.7}$$

Normalized cross-correlation can compensate for intensity changes in the example image. Having a cross-correlation term in addition to L1-distance, using Euclidean distance to measure similarity can compensate for both intensity changes and color shifts. Both of these similarity measures can be implemented nearly as efficiently as Fast Fourier Transforms [58].

3.3.2 Detect and Localize the Target

After the ROI has been localized, we must focus on detecting the target of interest within the identified region. There are a variety of potential methods for detecting and localizing the target; for example, we can: model the target directly, search for novel motion within the region, or model the background identifying non-background objects as the foreground target. Regardless of approach, our goal is to produce the likelihood and (x, y) coordinates of the single most probable possible match. Recall, the input to this stage is the localized ROI and the output match for a sequences of frames will be analyzed by the tracking phase to extract sequences containing the target (Figure 3.2).

Model the Foreground

The direct approach is to model the target object in question. However, we cannot assume that defining features for the targets of interest are available for required the class of applications we consider, in this case bees (see Figure 3.6). For example, bees have minimal novelty in texture, color, or shape. Worse, there are many other object in the background clutter surrounding the ROI that look quite similar to the target itself. For example, a leaf in partial shadow can have a very similar appearance to the bee itself. Thus, directly using an approach like template matching is bound to fail. Even more complex approaches are foiled by the lack of distinct features.

Search for Novel Motion

Instead of directly modeling the foreground, we can apply an algorithm like optical flow [6] to identify motion vectors within the scene. With these motion vectors

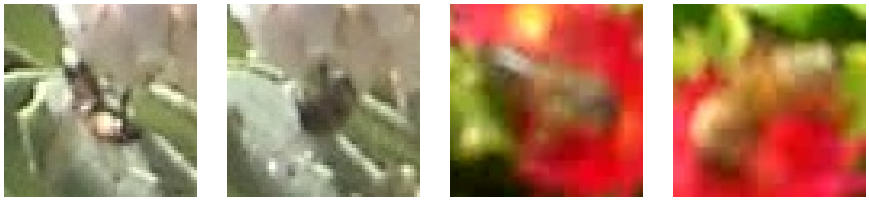
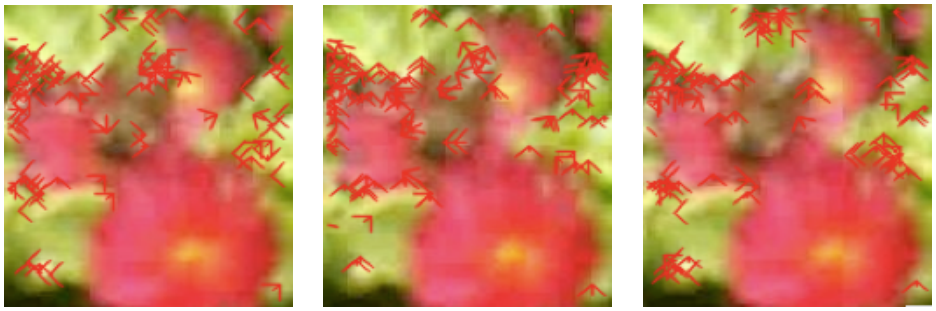


Figure 3.6: Four frames containing a bee perched on the region of interest. They are generally uniform in color and texture. They have multiple possible poses, though all are conical in nature.

in hand, we can identify motion that is counter to the global or local average direction and magnitude of motion. There are a variety of optical flow algorithm present in computer vision literature; we chose to use the Lukas-Kanade method [63] because of its assumption of locally consistent flow and general popularity within the field of computer vision.

The algorithm progresses by first choosing a set of features from pairs of temporally adjacent images and the solving an overdetermined system of linear equations defining their potential motion. Typically, the image features are chosen by finding corners in the image after edge detection has been performed [6]. Then, various features are discarded based on their novelty and the density of features present in a given region; this helps improve feature coverage of the image while limiting the total number of features tracked. Finally, the features are efficiently matched conditioned by similarity and locality constraints [10].

To use this approach requires relatively high frame-rate data, greater than 20Hz, so that frame-to-frame displacement of the target is small. This requirement stems from locality of motion constraints within the definition of the optical flow algorithm. In order to improve accuracy and robustness, most optical flow algorithms prefer motion vectors that result in local, relatively small magnitude, motion vectors rather than larger magnitude motion vectors [64]. If the frame



(a) (b) (c)

Figure 3.7: Three frames from a 20Hz video with optical flow vectors overlaid. (a) flow vectors associated with feature on the bee itself appear to have novel motion relative to the background. (b) background flow vectors have little globally directed motion, so identifying novel motion would be difficult. (c) no features local to the bee were chosen by the feature selection algorithm.

rate were lower, the distance moved by the subject would produce motion vectors whose magnitude would be too large to correctly identify under these constraints. Further, optical flow is somewhat susceptible to movement of the camera. Though this movement can be incorporated into model of global or local average motion, the magnitude of the motion can drown out the relatively small motion expected of the target.

Using this approach provided mixed results for our data. In Figure 3.7(a) we can see that many features are chosen on the bee, and they appear to be novel relative to the generally left-oriented motion of the rest of the frame. However, there are a few important failure modes: multiple regions of novel motion, poorly defined global or local motion, and missing target-related motion. In Figure 3.7(b), the general motion is somewhat poorly defined, but still, the target's motion is somewhat unique. However, there is also another region of unique motion to the left of the target. Finally, in Figure 3.7(c), no features of the target itself were tracked, resulting in misidentification of novel motion; allowing more features to be tracked in this instance only served to confound the direction of

global motion.

Model the Background

The final method models the background and assumes that any occlusions are interesting foreground objects. There are a variety of background subtraction algorithms present in the literature. We chose to use the algorithm presented by Ko et. al. [54] because it explicitly takes into account so-called camouflaged foreground objects, those whose color distributions appear to be similar to that of the background.

The background is modeled as a set of color histograms $p_{ij}(x)$, each representing a square region of the image of size $4c^2$ centered around the pixel located at the ith row and jth column of the image with value $x \in \mathbb{R}^3$. The same image patch from T frames are combined into the final estimate of the background histogram for the given patch (Equation 3.8). The feature vector x, which represents the color 3-tuple at a given pixel, is quantized to better approximate the true density.

$$p_{ij}(x) = \frac{1}{|S|} \sum_{s \in S} \delta(s - x)$$
$$S = x_t(a, b) \mid |a - i| < c, |b - j| < c, 0 \leq t \leq T \tag{3.8}$$

Using all spatially local pixels in the surrounding region helps to make the background model resilient to movement in the background itself. Leveraging the temporal extent of the dataset helps tolerate changes in the background over time caused by changing lighting or other natural effects. We chose T images randomly sampled from the image sequence, without knowledge of the target's presence. This provides a more representative sample of possible backgrounds in comparison to T sequential images.

Figure 3.8: Background subtraction can easily model the background in this image sequence since it is relatively stable and the foreground object is sufficiently novel in comparison. The target seen in an image from the sequence (left) is easily visible in the difference image (right).

Simpler background subtraction algorithms attempt to classify each pixel in a test image against the background distribution to determine membership [25]. To deal with camouflaged targets, Ko et. al. compare the histograms of image patches in the test image, $q_{ij}(x)$, to those of the background model, $p_{ij}(x)$, using the Bhattacharyya distance [8] (Equation 3.9). The distance, d, falls in the range $[0, 1]$ where larger values imply greater similarity.

$$d = \int_X \sqrt{p_{ij}(x) \cdot q_{ij}(x)} dx \tag{3.9}$$

The result of performing background subtraction is a difference image, where each gray scale pixel in the image represents the Bhattacharyya distance between the region about that pixel and the corresponding region in the background model. In some cases where the background is sufficiently simple, the target immediately stands out against the background, as seen in Figure 3.8. In these cases, a simple threshold for the distance and blob detection will easily detect and localize the target.

However, in most cases with cluttered natural scenery as the background, natural change in the background's appearance creates significant noise in the

Figure 3.9: Background subtraction has more difficulty modeling cluttered natural scenery (left) where light flecks and other transient effects cause increased noise in the difference image (right). Thus, we use template matching to identify the target in the difference image (red square).

difference image (Figure 3.9). In these cases, such simplistic approaches will find many targets resulting from transient aberrations. Instead, we again use template matching to detect regions of the appropriate intensity and shape. The user defines a single template by selecting the appropriate region of a difference image that contains the target, in much the same way that the ROI is identified.

The single best match value found by template matching along with its (x, y) coordinates are emitted to the next stage that tracks these potential targets over time. It is important to note that no attempt is made to classify if a given match is in fact a foreground object or a background aberration. False positives are identified and handled appropriately during tracking.

3.3.3 Track the Target over Time

With the potential target localized, we focus on separating the matches that are in fact representative of the target's presence from those that are simply noise

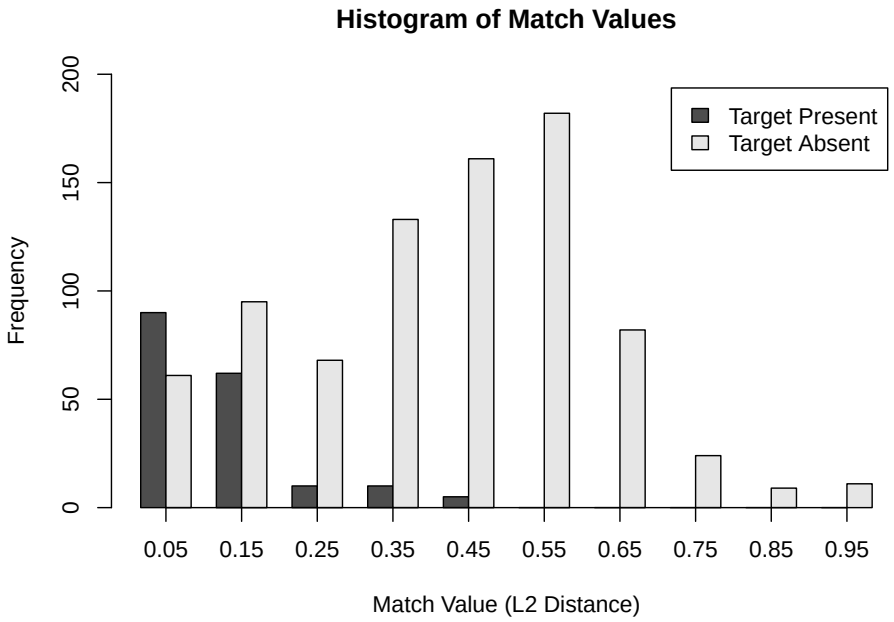


Figure 3.10: Distribution of match values when target is present and absent. The two distributions overlap significantly, foiling any naive classification based on match value alone.

found in the difference image. Our goal is to produce a set of disjoint image subsequences that where the target is present within the ROI.

This simplest approach would be to build a binary classifier to distinguish between target and not-target using the match value. Such a classifier, whether built by SVM [105] or a simpler technique like decision trees [11], would simply try to separate the classes by some sort of boundary in feature space. When the background model is accurate and there is minimal background motion, such classification is sufficient. Unfortunately, as the background model begins to break down in the presence of significant background motion, the distribution of match values is not separable as seen in Figure 3.10. Any attempt to build a classifier around this feature will either render many false positives or false negatives if it is tuned for recall or precision respectively.

To surmount this difficulty, we leverage the fact that frames are not independent, but are temporally correlated. We assume that motion of the target follows some simple physical process once it enters the ROI. That is, its motion must be smooth in time and space; it cannot be erratic. In our application, we expect the bee to land on the flower and proceeds to walk around. If we capture imagery of this motion with sufficient sampling frequency, we expect it to be smooth[1]. Since we have no *a priori* knowledge of the process, we simply model the displacement of the target frame-to-frame as a normal distribution (Equation 3.10).

$$\begin{aligned} x_{i+1} &= N(\mu = 0, \sigma_x) + x_i \\ y_{i+1} &= N(\mu = 0, \sigma_y) + y_i \end{aligned} \tag{3.10}$$

This approach is quite similar to object tracking when treating the object being tracked as a single point [112]. In our instantiation of object tracking, we only consider object translation in the 2D viewing plane. We expect that the detection algorithm to compensate for rotation and deformation of the target, correctly detecting and localizing the target if the target is in fact present in the frame. When no target is present, its behavior is undefined, and it may emit random values.

Recall, the features emitted from object detection for the ith image are the match value m_i and the match location l_i. As noted above, we model the deflection of the target between frames as a normal random variable and compute σ_x and σ_y directly using maximum likelihood estimation. We model the match value when the target is present as a Gamma distribution Γ, because of its heavy tail, and directly compute its α and β parameters using maximum likelihood estimation.

[1]Section 3.4.3 shows that a frame-rate of 1Hz is sufficient for the target's motion in our application to appear smooth

We now try to find all contiguous sequences of images S_i from the ordered set of frames F, such that $S_i \cap S_j = \phi \; \forall \, i \neq j$, where S_i obeys the following constraints:

$$\begin{aligned}
\frac{1}{|S_i|} \sum_{s \in S_i} m_s &< 90\text{th percentile}(\Gamma) \\
max(m_s) \, \forall s \in S_i &< 99\text{th percentile}(\Gamma) \\
\frac{1}{|S_i|} \sum_{s \in S_i} \|l_s - l_{s-1}\| &< 90\text{th percentile}(N) \\
max(\|l_s - l_{s-1}\|) \, \forall s \in S_i &< 99\text{th percentile}(N)
\end{aligned} \tag{3.11}$$

These constraints are intentionally set conservatively so that our output will be tuned for recall rather than precision. We make this trade-off since domain scientists would rather the results contain false positives, which they can quickly ignore upon visual inspection, instead of incorrectly omitting false negatives. An optimal solution for this global constraint satisfaction problem is possible, but it is polynomial in complexity since all possible contiguous subsequences must be considered.

Instead, we attempt to greedily grow sequences around seed frames where $m_s < \text{median}(\Gamma)$. We set the seed constraint such that at least one frame from each sequence is represented. Though this will likely identify frames that do not actually contain the target, we prefer recall. This approach of choosing seed frames is a departure from traditional object tracking formulations. Most object tracking algorithms expect that at least one frame is labeled with the correct location of the object [112] or that optical flow can reveal the initial correspondence between the first two frames [87]. Instead, we approximate this knowledge with seed frames and compensate for false positives since we cannot expect to have such information.

Once seed frames are chosen, the sequence is grown one frame at a time subject to the constraints; the next frame to consider alternates between the frame temporally before the start of the sequence and the frame just after the end of the sequence. Sequences are grown until no frames can be added without breaking the constraints. Since this algorithm is prone to prematurely ending sequences because of outliers in the detection output (detection errors), we merge sequences that are separated by less than τ_s seconds. To reduce false positives introduced by invalid seeds, we drop sequences lasting less than τ_d seconds. For our application, we choose $\tau_s = 1$ seconds to allow for one frame detection errors at 1Hz and $\tau_d = 2$ seconds because the minimum interesting dwell time of a pollinator is 2 seconds [9].

This formulation is an extension of previous work by Sethi et. al. [92] and Rangarajan et. al. [87]. Sethi et. al. introduce a *smoothness of motion* constraint that they model in terms of estimated inertia of the tracked object. Rangarajan et. al. further impose a penalty based on *proximal uniformity*, asserting that the target shouldn't move too far in a short period of time (given sufficiently high sampling rate) in addition to having smooth motion. In both formulations, detection is considered separately from tracking, where the detected object is characterized as a point only defined by its (x, y) coordinates. Both propose greedy algorithms for their respective constraint satisfaction problems.

3.3.4 Considering Multiple Potential Targets

The approach described here could be extended to consider multiple potential targets, essentially allowing more information to pass from object detection to object tracking. We modify the detection algorithm to emit all targets within five percent of the best match. The locations of these matches are then clustered and

one target from each cluster is passed to the tracking algorithm. Like optical flow feature selection, we prune potential matches, only allowing K matches within a radius R. For our applications we are seeking clusters that are about twice as large as the target itself; so, for a 40px square target, we see clusters with $R = 40px$. From each of these clusters we select the most likely target ordered by match values.

The tracking algorithm could then be modified to use these extra data to account for detection failures, choosing only one detected target from each frame. We apply the same constraints as defined by Equation 3.11. However, instead of only considering the single match provided by the detection algorithm, we greedily choose the match that minimizes these constraints. This approach attempts to correct for oscillations in the detection algorithm when two or more nearly equally likely targets are present. We show through evaluation that this modification is not required to achieve sufficient accuracy. Still, we intend to peruse this modification in future work.

Alternatively, we can relax the single target constraint and consider each potential detection as a target and track each over time subject to our existing formulation; a similar approach is taken by Shafique et. al. [93] where they model the point targets as nodes in a digraph and attempt to find the minimal paths between frames subject to constraints. However, the class of applications we consider are only interested in target presence rather than exact count or quantity of targets present at once. As a result, we ignore multiple target tracking for the purposes of this work, deferring its consideration to future work.

3.3.5 Generalizability

During the course of our procedure description, we have assumed certain characteristics of the application and shown why they are reasonable when considering pollinator visitation. These assertions primarily concerned properties of the target (both is appearance and its motion) and properties of the region of interest.

The assumptions that we made about the target necessitated the introduction of a region of interest. Specifically, we assert that the target is not novel compared to a cluttered natural background and that its motion is quick and erratic. This is clearly true of insects, but is not necessarily true of larger animals, like deer, or simpler natural background, like under water. In these cases, slight simplifications can be applied to our approach. In both cases we can remove intermediate registration provided by ROI detection and localization, modeling the target directly for obvious targets, or modeling the background directly for simpler backgrounds. However, the ROI may still be of use if has an affect on the phenomena. In the case of pollinators, we assert that the ROI reduces complex, erratic movement of the target to a simple predictable motion. We believe this to be true for many biologically relevant studies like the study of foraging behavior in black bears [71] where they authors tracked the bear's foraging on human in developed areas. Though the bear don't likely move that quickly in the open, they certainly move much less when feeding.

With the need for an ROI established, we began to make certain assumptions about its form. The most important characteristic of the ROI is its visibility amongst the background clutter. This is almost certainly true in the general case, be it a particular food containing bin or even a marine formation [24], regions of interest are defined by their unique appearance relative to the background. Further, we require that the ROI be present in all frames of the video sequence.

Figure 3.11: The rig used to capture the *IcePlant1* and *IcePlant2* datasets in the Los Angeles Basin.

In most cases, we feel that it is reasonable for camera system deployed to maintain this invariant. In cases where the ROI has significant motion this assumption may begin to break down. Finally, we assert that the ROI can be represented by a single template image. When the ROI is fixed relative to the camera, this is likely the case. If not, we believe that more complex modeling can replace our simple template matching approach to identify the ROI in these scenes, though we leave this to future work.

3.4 Evaluation

Biologists are interested in the pollination behavior of bees, specifically the quantity and duration of visits to a particular flower [5]. For this application, the

Data Set	Frame Rate	Duration	Frames	Targets	Motion
IcePlant1	20Hz	5 min	6000	127	camera (small)
IcePlant2	20Hz	10 min	12000	904	camera (small)
Manzanita1	0.5Hz	10 min	350	15	wind (large)
Manzanita2	0.5Hz	30 min	1000	182	wind (large)
Manzanita3	0.5Hz	300 min	9000	0	wind (large)

Table 3.1: Details about the collected pollinator datasets. Targets refers to the number of frames that contain the target foreground object.

region of interest is the flower itself and the target we are trying to track is the bee, when present. As output, we would like to provide biologists with a summary of the pollinator events that took place over a given period of time. This summary will include sequences of frames that contain the target, with the target identified, as well as summary statistics like dwell time of the pollinator on the flower.

To perform this evaluation, we collected a number of different datasets with a variety of characteristics. Two datasets were collected in the Los Angeles Basin of *Aptenia cordifolia*, a species of ice plant. These data were collected at 20Hz for five and ten minutes (we will refer to these data as *IcePlant1* and *IcePlant2* respectively), see Figure 3.11. Another two datasets were collected at James Reserve of *Arctostaphylos pringlei*, a species of Manzanita shrub. These data were collected at 0.5Hz for 10 minutes, 30 minutes, and 5 hours (we will refer to these data as *Manzanita1*, *Manzanita2*, *Manzanita3* respectively). Details about these datasets can be found in Table 3.1. All imagery were collected with 640x480 resolution. We limit our evaluation to these temporally short datasets since we require a human to manually inspect all frames and label ground truth. However, the procedure we have developed can run for much longer.

Figure 3.12: The illumination present throughout the Manzanita datasets varied significantly. Here we show the visual difference between direct (left) and indirect (right) illumination.

The *IcePlant* datasets had somewhat limited background motion due to the rigid nature of the plant. Most of the motion in these datasets was induced by camera instability during collection. In contrast, the *Manzanita* datasets had significant natural background motion due to wind; the same wind also has some effect on the mounted camera. The foreground in the *Manzanita* datasets also saw significant changes in illumination (see Figure 3.12), oscillating between direct and indirect sunlight somewhat randomly.

A pollinator was considered to be present if it occluded any part of the flower, regardless if it was contained completely within the boundaries of the ROI. All frames that contained a pollinator were labeled as such, though the exact location of the pollinator was not computed. Recall, knowledge of the pollinator's location is not required by the domain scientists since they have plan to visually inspect all output. Even though this information is computed as a byproduct during detection and tracking, we don't collect ground truth or evaluate our ability to

localize the target since it is not of interest to the domain scientist.

3.4.1 Region of Interest Detection

When identifying the region of interest (ROI), our goal is to limit further computation to the ROI alone. Practically, this requires cropping the input image to the boundaries of the ROI. For this to be effective, the detected ROI must fully contain the semantic ROI, in this case the flower. Thus, we define our criteria to be:

$$Accuracy = 1 - \frac{Misses}{Total\,Frames} \tag{3.12}$$

where a miss is a frame where the computed ROI does not contain the entirety of the semantic ROI.

To obtain the template image for each dataset, we manually crop the ROI from the first frame of the dataset. We evaluate the effectiveness of the various similarity measures on each of the collected data sets in Table 3.2. First, we see that *IcePlant1* and *IcePlant2* have very simple background that make template

Accuracy (Misses)	IcePlant1	IcePlant2	Manzanita1	Manzanita2
Cross-Correlation	100% (0)	100% (0)	86.6% (47)	96.1% (42)
Normalized Cross-Corr	100% (0)	99.3% (2)	98.2% (6)	99.6% (4)
L2 Distance	100% (0)	100% (0)	98.2% (6)	99.8% (2)
Normalized L2 Distance	100% (0)	100% (0)	98.5% (5)[a]	99.9% (1)

[a] The ROI is either partially or completed out-of-frame in 4 frames.

Table 3.2: The accuracy of various distance metrics for each of the tested datasets is shown here along with the absolute number of misses. We define a miss to be any localization that does not completely contain the region of interest (ROI). Typical failure modes result in partially cropped ROIs, which will result in poor performance later in the process.

Figure 3.13: Example frames illustrating the visual effect caused by the change in natural illumination that occurs during the day. These images were captured at 2pm (left), 4pm (middle), and 6:30pm (right).

	Hour 1	Hour 2	Hour 3	Hour 4	Hour 5
Accuracy (Misses)	100% (0)	100% (0)	98.8% (12)[b]	100% (0)	80% (200)

[b] The ROI is either partially or completed out-of-frame in 6 frames.

Table 3.3: The accuracy of computing the ROI in the presence of changing natural illumination from 2pm until 7pm.

matching successful across the board. However, for the Manzanita datasets, we see that the simplest approach, unnormalized cross-correlation, is not very effective. The best distance measure is shown to be normalized Euclidean distance.

To measure the resilience to changing lighting conditions, we captured a 5 hours dataset from James Reserve of the same Manzanita shrub, from 2pm to 7pm (dataset *Manzanita3*); example frames from this sequences can be seen in Figure 3.13 and results in Table 3.3. During the first 4 hours of sampling, there were only 12 misses, 6 of which were caused by the semantic ROI being out-of-frame due to wind. Only after 6:30pm, when the sun was nearly down and the frame nearly black, did we start to see a significant fraction of misses.

When operating on images with 640x480 resolution, our implementation of template matching (based on OpenCV [80]) can process a single image in approximately 200ms on a 2.4GHz Intel Core 2 Duo. Though we don't require this

system to process data as fast as it is captured, this processing latency is more than sufficient to process 1Hz data online.

3.4.2 Target Detection

The goal of the target detection algorithm is to correctly localize the target in each frame. Given the way we have defined the algorithm, a potential target will be found in each frame even if no target is present. Recall, we defer the separation of invalid targets from from valid targets until we have temporal information during tracking. Our detection algorithm requires us to train both a background model and a template matching model. We choose to train the template matching model using only one example to ease the burden on the user.

The background model, however, must be trained with multiple representative background frames. Instead of requiring the user to identify specific frames that do not contain the target, we choose to train on N randomly selected frames without replacement from the image sequence, again reducing the burden on the user. Like ROI detection, we choose to evaluate the algorithm's effectiveness by measuring its accuracy as defined in Equation 3.12.

We first evaluate this procedure on the *IcePlant* datasets. For these data, we down-sample the 20Hz data to 1Hz to achieve parity with the *Manzanita* datasets. Recall, for these datasets, there is limited background motion and the target occludes nearly the entire flower. Thus, we expect that relatively

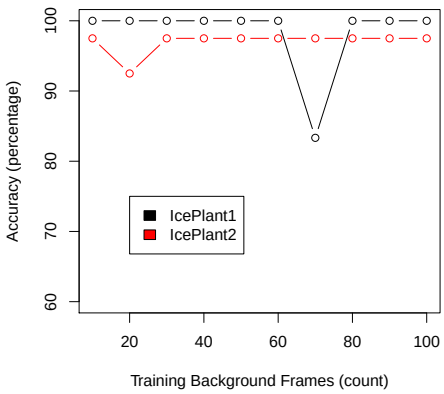
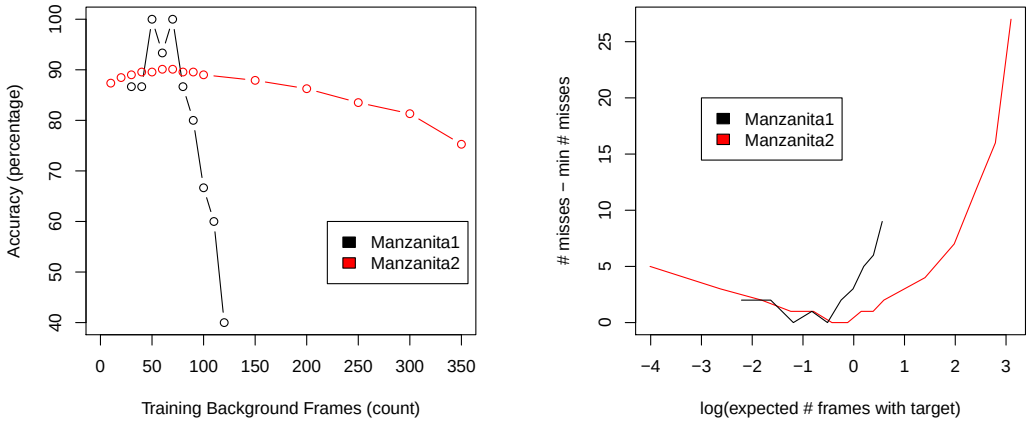


Figure 3.14: The accuracy of the detection algorithm when attempting to localize the foreground target on the *IcePlant* datasets.

few frames are required to train the background model, as depicted in Figure 3.14. Since there are so few frames with the target present for the *IcePlant1* dataset, any miss is exaggerated as seen with 70 training frames. At 1Hz, only 6 frames contained the target, so a single miss resulted in a 16% drop in accuracy. For the *IcePlant2* dataset, there was a single frame that could not be correctly identified in any of the experimental setups resulting in an accuracy of 97.5%. Background subtraction correctly identified the region where the foreground bee occluded the background flower, but the region was not large enough to trigger a template match since the bee itself was occluded by other background elements.

These data are not particularly interesting because they don't sufficiently stress the algorithm. The result of the algorithm when applied to the *Manzanita* datasets better illuminates potential flaws in the approach. In Figure 3.15(a) we see the accuracy of the algorithm as we vary the number of training frames for both the *Manzanita1* and *Manzanita2* data sets. The maximum accuracy of *Manzanita1* is 100% with 50 or 70 training frames, and 90.1% with 60 or 70 training frames for *Manzanita2*. We see that the accuracy of this algorithm does not strictly increase as more training frames are added. This is expected since the frames are chosen randomly and frames including the target will be inevitably included as the number of training frames increases.

The inflection point in the algorithm's accuracy is after including about 70 example background frames for both datasets. This is somewhat unexpected given the fact that the *Manzanita1* dataset is about a third the size of the *Manzanita2* dataset (see Table 3.1). To understand this oddity, we define a new quantity, sensitivity, to be the number of additional misses incurred as the expected number of training frames containing a target $E(x)$, increases with the number of training frames x. Note, that $E(x)$ can be computed in the straight forward manner even



(a) Detection Accuracy

(b) Sensitivity to Example Frames

Figure 3.15: These plots illustrate the (a) accuracy and (b) sensitivity of the proposed detection algorithm. For both data sets training on 70 random images is optimal, and that the *Manzanita1* is more sensitive to the target presence in the example background images.

though the expected distribution of targets across the training set is not uniform. Since we are randomly sampling frames, the effect of the bursty distribution of targets only effects second-order statistics.

In Figure 3.15(b) we see that the *Manzanita1* dataset is far more sensitive to the target being present in the training examples. This sensitivity stems from the fact that the target tends to appear in the same location on the ROI in *Manzanita1* (near the bottom of the flower in the center), whereas its location is more evenly distributed around the perimeter of the ROI in *Manzanita2* as seen in Figure 3.16. When the background subtraction algorithm incorporates a training frame containing a target into the background model, it will have a significant negative effect on the detection of any other target in a nearby region. Since the targets in the *Manzanita1* dataset are spatially clustered, this effect is amplified.

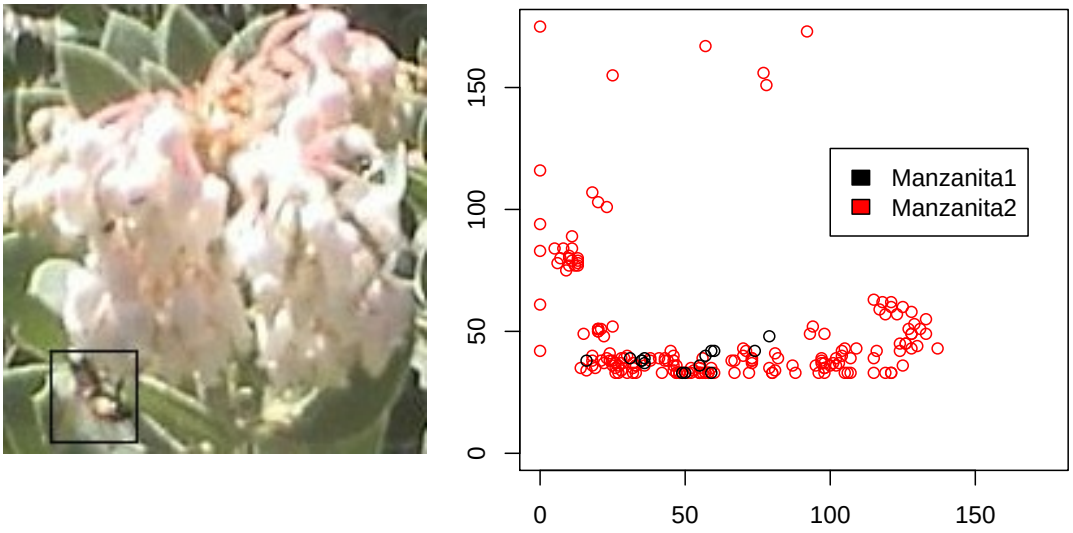


Figure 3.16: The ground truth location of the upper-left corner of the target's bounding box with respect to the region of interest for datasets *Manzanita1* and *Manzanita2*.

When the detection algorithm did fail to localize the target, it failed due to an excess of noise in the difference image produced by background subtraction. These events typically occurred when the flower was significantly displaced from its resting location due to wind. In these cases, our assumption of pure planar motion begin to break down, and the background model begins to poorly account for the out-of-plane rotation of the flower. An example localization success and failure are depicted in Figure 3.17.

When considering only frames that actually contain the target, these failures have two different types of temporal characteristics: many sequential failures and random single frame failures. When there is significant background noise due to a major displacement event, failures are likely to be highly correlated, resulting in many sequential failures. Alternatively, during minor displacement events, some small amount of noise is introduced and single frame failure are more common. Random single frame failures can typically be accounted for during tracking, whereas sequential failures cannot be recovered.

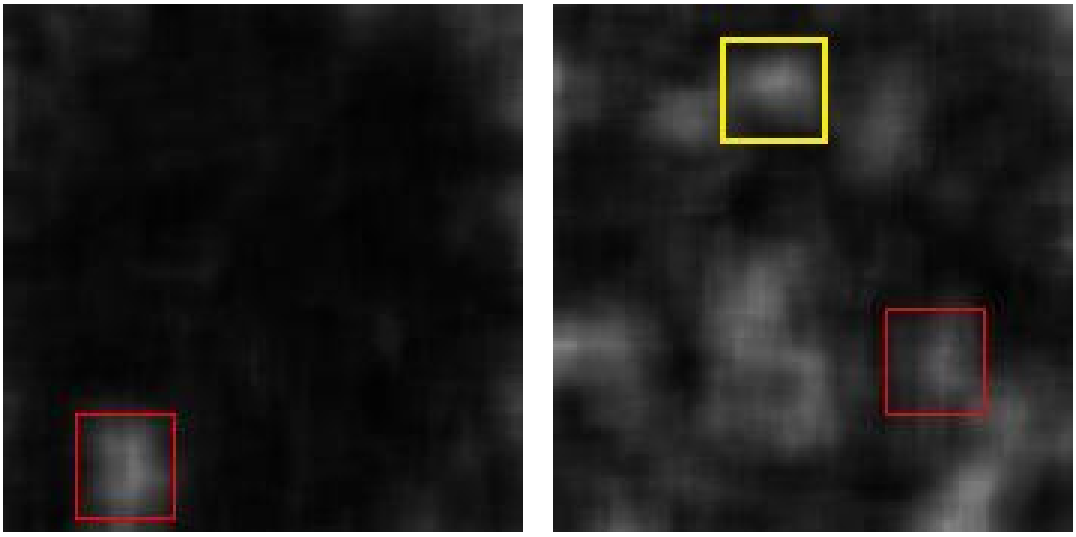


Figure 3.17: Example target localization superimposed on the difference image produced by background subtraction (red box). When successful, background subtraction typically produces limited noise (left). Failures typically arise because of excess noise (right); here the correct location is denoted by the yellow box.

In order to determine the number of training frames for a future instantiation of this procedure, we suggesting repeating the same measurement was have performed. In general terms, having fewer than 20 training frames is unlikely to be sufficient for all but the simplest scenes. Conversely, having more than 150 training frames is likely to over-train the background model, leading to the failures seen in the *Manzanita1* dataset. A reasonable place to start is 70 training frames, as was found to be optimal for all four tested datasets.

3.4.3 Target Tracking

Though we are tracking the target through time, we are not interested in its location but rather its presence. As a result, we evaluate our tracking algorithm as though it were a classifier, measuring precision and recall. Optimally, we would like to have approximately 90% precision and 90% recall. Since a human will

eventually be analyzing the image sequences thought to contain the target, we are willing to sacrifice precision for recall; it is more important for the domain scientist to be presented all interesting frames even if there are more false positives.

As described in Section 3.3.3, the tracking algorithm has a number of parameters. There are three that constrain the allowable match values within the sequence; these can be derived from quantiles a Gamma distribution fit to the match values of frames where the target is known to be present. Another pair of parameters constrain the allowable displacement of the target between adjacent frames; these are computed from a Normal distribution fit to displacements between adjacent frames containing the target. In practice, the values of parameters constraining displacement can be fixed *a priori*. When the target is not in frame, the location reported by the detection algorithm varies wildly. As a result, a simple threshold is sufficient to differentiate between this random motion, and the methodical motion of the target.

We first consider the success of this algorithm on the *IcePlant* datasets. For the purposes of this evaluation, we trained the detection algorithm using 20 random frames; as we saw in Figure 3.14, nearly any number of training frames produces the same detection results. In Figure 3.18, we see that for *IcePlant1* we either get 0% or 100% precision and recall when varying the model parameters.

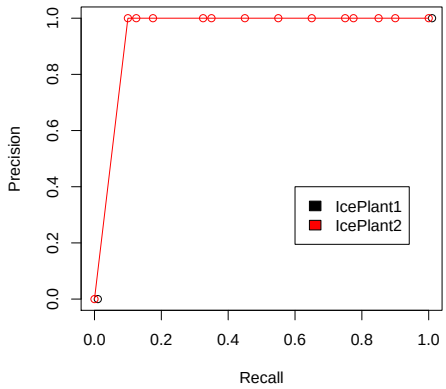
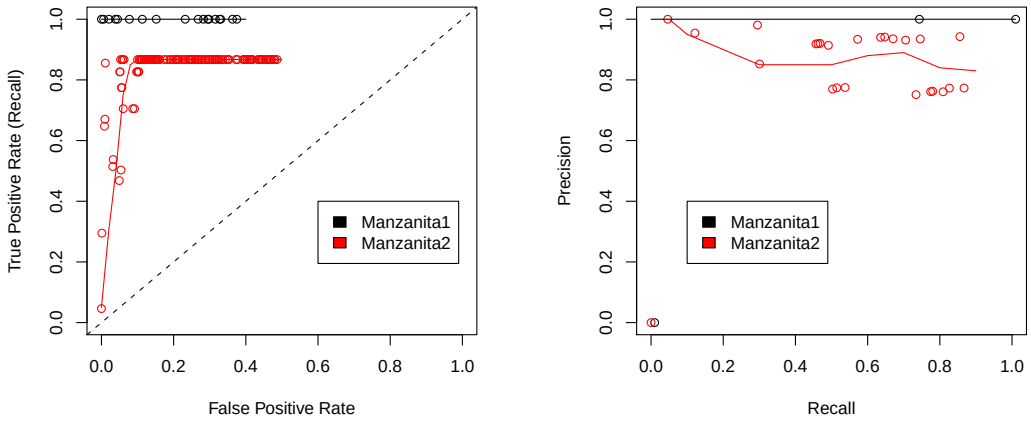


Figure 3.18: The precision and recall of the tracking algorithm when attempting to identify the foreground target on the *IcePlant* datasets.

For *IcePlant2* we achieve 100% precision for a wide range of recall values up to and including 100% recall. Using maximum-likelihood to fit a Gamma dis-



(a) ROC Curve

(b) Precision Recall Curve

Figure 3.19: These plots illustrate the effectiveness of the tracking algorithm. (a) shows the ROC curve and (b) shows the precision recall curve for the *Manzanita* datasets. Note, not all true/false positive values are possible. The achievable values are defined by characteristics of the data and the tracking algorithm.

tribution to the match values when the target was present we got $\Gamma_{\text{iceplant}}(\alpha = 1.51, \beta = 0.26)$. We acquire the parameters for our tracking algorithm by computing the empirical quantiles of Γ_{iceplant} as specified by Equation 3.11. The tracking algorithm produces 100% precision and recall when using these parameters.

We saw previously that the detection accuracy for *IcePlant2* was 97.5%, as it suffered a single failure due to the target being occluded by the background (see Figure 3.14). Since our tracking algorithm has 100% recall, this implies that it properly dealt with a single frame detection error. Since this detection error occurred in the middle of a sequence, the tracking algorithm created two adjacent sequences separated by a single frame. We then greedily merged the two sequences, forming a larger sequence and enveloping the detection error.

Next we consider the *Manzanita* datasets, which had significantly more unrelated motion and color variation. For the purposes of this analysis, we trained

Data Set	Detection Accuracy	Tracking Precision	Tracking Recall
IcePlant1	100%	100%	100%
IcePlant2	97.5%	100%	100%
Manzanita1	100%	100%	100%
Manzanita2	90.1%	96.8%	85.8%

Table 3.4: Summary of results when our procedure is applied to the various datasets.

the detection algorithm with 70 random frames; this value produced the highest accuracy for both the *Manzanita* datasets. For *Manzanita1*, we computed $\Gamma_{\text{Manzanita1}}(\alpha = 7.37, \beta = 0.01)$ as the maximum likelihood Γ. Applying the parameters as described above, we were able to achieve 100% precision and recall as shown in Figure 3.19. This was somewhat expected given that the detection algorithm was able to locate the targets with 100% accuracy.

Like the other datasets, we compute $\Gamma_{\text{Manzanita2}}(\alpha = 2.22, \beta = 0.05)$ from match value of frames with targets present in the *Manzanita2* dataset. Configuring the algorithm with the parameters derived from $\Gamma_{\text{Manzanita2}}$ results in 96.8% precision 85.8% recall. This is within 2% of the precision and 0.5% of the recall of the best possible parameters found by brute force.

On average, 90% of frames in a given sequence were correctly returned to the user, but in the worse case, this dropped to 45% for a single sequence. Further, our 85.8% recall on the *Manzanita2* dataset is less than the 90.1% detection accuracy. These failures are due to a few representative types of errors: a) dropped subsequences, b) dropped sequences prefix or suffix, and c) added sequences. Type (a) an (b) failures are caused by the detection oscillating between two potential targets in the frame; one is the correct target, the other is noise. This sort of oscillation isn't apparent in the accuracy measurement of the detection

algorithm. Type (c) errors happen when the noise found in the difference image is not random, but instead a systematic problem detecting some novel part of the background. To the detection algorithm, this error appears to be a foreground object and it triggers the tracking algorithm as a result.

It seems that each of these types of errors are equally likely, as measured by intentionally using bad parameter values to increase the errors. A summary of the results of our approach on the four datasets are shown in Table 3.4.

3.5 Related Work

Our work encompasses a number of different subfields within computer vision. In particular we perform object detection of localize the region of interest and locate a potential target present on or near that ROI, and then use object tracking to find sequences of frames that contain the target of interest. The literature presents a number of different approaches to each of these problems. We provide a short summary of related approaches and applications.

Object Detection

There are a variety of mechanisms useful for performing object detection with in images. When considering video imagery, the three primary approaches distill into detecting the foreground object through direct modeling, detecting the object by modeling the background and searching for occlusions, and searching for salient image features likely to be the foreground object. The simplest form of foreground object modeling is template matching [13]. This approach requires near matches of the object to be present in the image, an assumption reasonable in some situations like our detection of the region of interest. An alternate ap-

proach based on a cascade of features passed to a classifier is proposed by Viola et. al. [107]. This approach has been shown to work quite well for detecting objects, like faces, that share common structure. Neither of these approaches work well for detecting bees because they have poorly defined visual features with little structure.

An alternate approach simply searches for the region in the image that would most likely grab the attention of a human. This type of approach is compelling since the bee is visible to a human when the video is played back. Itti et. al. [46] [45] develop a procedure that produces a saliency map, where each location in the map represents the interestingness of that location computed through abrupt changes in color, intensity, or orientation. This approach has been successfully used detect small sea life passing by underwater observatories [24] [17]. This approach has trouble with our application since the bee we are trying to detect is somewhat less apparent that the flower. As a result, the algorithm will detect sun flecks on the flower and surrounding plant, while ignoring bee itself.

The approach we settled upon for our work was modeling the background and asserting that non-background objects were the foreground object of interest. There are a variety of background subtraction approaches that leverage both parametric and non-parametric descriptions of the background. Harville et. al. [37] use an explicit mixture of Gaussian approach to model the background at individual pixels, adapting that model over time. Instead of explicitly fitting a Gaussian, Ko et. al. [54] build an empirically distribution of pixel values in the background, comparing this to the distribution taken from an example image using the Bhattacharyya distance [8]. Such a non-parametric approach has had success in both our application and others [25] that attempt to localize foreground object against a natural background.

Motion Detection

An alternative to directly detecting the object itself is detecting characteristic motion within the scene that likely represents the object. This is a more specific version of the saliency approach, usually only considering simple features like edges and corners [64]. A large variety of such so-called optical flow techniques are presented in the literature. In general, they attempt to extract a set of representative image features, and locate their approximate location is subsequent images [6]. It is important to note that performing such analysis requires very high-temporal resolution data to minimize the actual change between images. This approach had little success on our data since relatively few features that were selected were actually part of the target, and the target's motion was small with respect to the natural motion present in the rest of the scene.

Object Tracking

Once targets have been identified, we used object tracking to help discard false positives, requiring a sequence of detections to obey certain constraints. Similar approaches largely fit into two categories: deterministic approaches attempting to minimize some energy function representing the object's motion and statistical approaches that try to propagate the uncertainty in detection into the tracking algorithm. Shafique et. al. [93] suggest a deterministic approach using a non-iterative greedy approach to find the minimum energy path linking target detections across multiple frames. Sand et. al. [90] tracks far more points for much longer, framing their work as an extension of optical flow for use in point tracking.

The deterministic approaches separate detection from tracking [21]. A class of statistical approaches, called particle filters [88], attempts to integrate detection

and tracking by feeding tracking results back into the detection to help improve precision and recall. One representative approach to such feedback is work by Tao et. al. [97] where they attempt to track how various objects, generalized as arbitrary layers, move around the scene. Each pixel is to a layer and this assignment is allowed to change as the layers are tracked and individual pixels are found to be moving counter to the dominant motion of the layer. A similar form of feedback is present in work by Betke et. al. [7] where a recursive Bayesian filter [95] is fed data from both the object detection algorithm itself as well as the predicted tracks of those objects based on previous motion. Considering these tracks help to rule out mis-detections, a pattern we successfully applied to improve the precision of our procedure.

3.6 Conclusion

We have developed a procedure that is able to gather significantly more data than was previously available by manual collection. Through application driven evaluation, we have shown that our methodology can produce results useful for many biological studies, specifically those related to pollinator visitation. The key innovation of this procedure is the automatic identification and localization of a biologically relevant region of interest. By automatically cropping this region from the larger frame we both register the images for and discard segments of the images that would only serve to distract further processing. Though our evaluation is specific to a particular set of studies, we have defined a larger class of applications that could leverage this approach.

Using this data collection methodology, biologists can gather detailed data about somewhat rare, novel events that could not be captured previously without significant expense.

CHAPTER 4

Predicting Discrete Spatio-Temporal Signals

In the previous chapter, we developed a procedure to extract frames containing a novel object occluding a region of interested from a lengthy video sequence. Though this procedure is directly useful to existing biological studies, it still has the same scaling problems present in existing methods: more resources (now in the form of cameras rather than people) must be spent to gather data over a large area containing many independent regions of interest.

In this chapter we propose an extension to allow such spatio-temporal data collection. Attempting to capture all events with far fewer cameras than regions of interest is clearly impossible. Instead, we attempt to measure density of events per unit time though sampling, which is a useful summary statistic that can be used to further a variety of biological studies.

4.1 Introduction

Existing biological studies attempt to monitor some fixed area, searching for novel objects near regions of interest, using minimally invasive instrumentation. Similar to monitoring single regions of interest as discussed in Chapter 3, current techniques require a human to monitor the phenomena either in-person or on video after the fact. For example, studies that monitor plant-pollinator interactions typically can monitor less than 5 approximately $1m^2$ sites for about an

Figure 4.1: A $1m^2$ patch of flowers for which we'd like to collect pollinator density. The flowers (regions of interest) are easily identified by a human observer.

hour [20], or monitor more than 50 approximately $0.5m^2$ sites for only a few minutes each [5]. These approaches essentially result in a non-random sample of the events that is then used to draw conclusions about the phenomena in question. Our goal is to construct a procedure that can monitor a $1m^2$ site for an entire day to capture the density of pollinator visitation.

The class of target applications has a variety of important characteristics that we must exploit to scale these data collection efforts. These characteristics fall into three categories: attributes of the regions of interest, flexibility of the instrumentation, and properties of the event being sampled. First, regions of interest (ROIs) must be novel with respect to the rest of the scene and easily visible when viewing the entire scene. For example, the $1m^2$ patch seen in Figure 4.1 contains 14 flowers that are easily visible from this perspective. We allow these ROIs to move about some rest position during the course of data collection, but we expect them not to occlude one another. Though we could complicate our approach to tolerate occlusions, there are many scenes, such as Figure 4.1, that do not require

the additional algorithmic complexity.

Detecting events occurring over the entire patch would require particularly high resolution imagery. Our previous work, discussed in Chapter 3, required that the ROI occupy a 150px-square area. If we assume the ROI is 1-2in across, we would need imagery with a resolution of approximately 4500x4500 pixels to cover a $1m^2$ area. Currently, such imagery can only be acquired by professional digital cameras. Further, if we wish to scale to areas larger than $1m^2$, we have no hope of capturing high enough resolution imagery even with the best digital cameras available. Thus, we must instrument the environment with one or more pan-tilt-zoom cameras that we can actuate to produce a narrower and higher resolution view of an individual ROI.

We anticipate that the various ROIs have a non-uniform probability event occurrence. For example, in the case of pollinator visitation, certain flowers are more likely to attract pollinators than others. Though event duration may vary, we expect that there is a temporal correlation between events; the probability of an event occurring a short time in the future is higher if an event has occurred recently. This is a property of optimal foraging theory [65], which pollinators are known to implement [9].

Given these properties of the application, we strive to build a more scalable approach to gathering data using pan-tilt-zoom cameras to capture video imagery. Simply using random sampling of regions of interest in the given site neglects a key property of our problem: events are know to occur with non-uniform density. Instead, we propose using adaptive stratified sampling [101] to direct the motion of the *in-situ* cameras, each stratum being a particular automatically identified region of interest. This sampling procedure allows us to focus our limited resources and adapt to the clustered nature of events, producing an unbiased estimate of

events per unit time. Domain scientists can then leverage the resulting density estimate to draw conclusions about the phenomena under study.

The rest of this chapter is structured as follows: motivating biological applications are discussed in Section 4.2. The proposed procedure is described in Section 4.3. We discuss the experimental setup used to collect a representative dataset in Section 4.4 and describe the simulation we built to iteratively refine and evaluate our approach. Section 4.5 contains a thorough evaluation of our approach. Related work is discussed in Section 4.6, and conclusions are drawn in Section 4.7.

4.2 Motivating Application

Pollinator studies typically focus on either the behavior of the pollinator itself, or the effect the pollinator has on its environment. Both of these types of studies can benefit from long-term density estimates over a large area. For example, Fontaine et. al. [30] studied how density of bumblebees in a given region affects their choice in flowers. In particular, will a higher density of bees result in a more flower species being visited? Data for the existing study was collected in the field by a human observing individual bees and flowers. Though aided by software [78], which provided a simple interface for recording individual events, a trained observer was required to identify the events.

Bartomeus et. al. focus on the effect pollinators have on their local environment [5]. They postulate that invasive species are more attractive to pollinators than indigenous species, resulting in the further spread of the invasive plants. Data collection for this study required humans to manually count pollination events over many small regions for a short period of time. The density of polli-

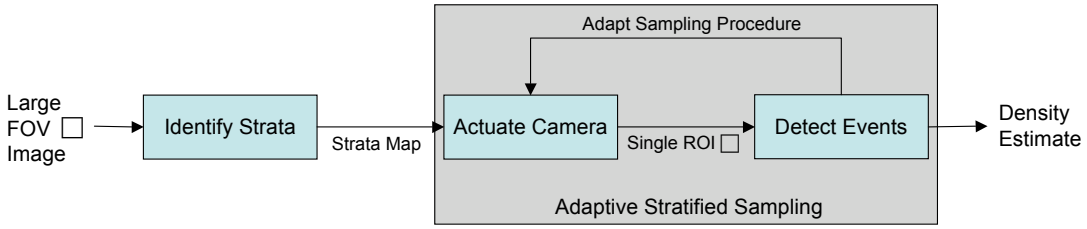


Figure 4.2: The process we propose consists of the three logical parts depicted here: detecting and localizing the various regions of interest (ROI) in a large field of view (FOV) image, actuating the camera to focus on a signal ROI, and detect visitation events on that ROI adapting the sampling procedure as necessary. The output is the temporal density of visitation events to the set of ROIs.

nation events between patches with and without invasive plants were compared to draw conclusions.

To increase the predictive power of these studies requires long-term data collection over a larger area. The procedure we propose can use one or more cameras to instrument a given region, allowing biologists to collect pollinator density data over approximately 1m^2 patches for multiple days. Further, we can instrument a number of independent regions to enable biologists to compare pollinator density across related regions. Alternatively, we can measure the density of pollinator visitation to different flowers within a single region, using multiple cameras trained on the same patch.

4.3 Procedure

We define a multi-stage sampling procedure to estimate the temporal density of novel events occurring near several regions of interest over some predefined area. Like other procedures we have discussed, this procedure can be tailored to the specific application of interest; the application we will examine is pollinator visitation density to a field of flowers.

The procedure is composed of three distinct parts as depicted in Figure 4.2. First, from images with large spatial extent, we detect the approximate locations of the various regions of interest in the scene. Second, we actuate the camera to zoom in on each region in succession, collecting higher resolution imagery with a narrower field of view. Finally, we detect the presence of novel objects on the ROI under current study, and feedback presence information to adapt the sampling methodology to best capture the phenomena.

Prior to executing this process, we must train the procedure on the current arrangement of the scene. This requires us to capture some number of training images to be used to correctly localize the regions of interest as well as train our sampling and detection algorithms. The duration that can be sampled without retraining the process varies between applications. For the pollination studies we consider, the process has to be retrained daily as flowers bloom and wither over time.

4.3.1 Assumptions

In order to make this problem tractable, we make a series of realistic assumptions about the character of both the regions of interest and the phenomena. Most importantly, for the applications we consider, we are only interested in localizing events that occur near one of the regions of interest present in the scene. For example, we only are interested in pollinator visitation to flowers, and are not concerned with pollinators flying about the scene. In order to limit the amount of data we require humans to label, we assume that all regions of interest look roughly the same. This way, a user supplying a single example is sufficient to identify all regions of interest. When this is not a realistic assumption, like when viewing a large flower from different perspectives, we allow the user to manually

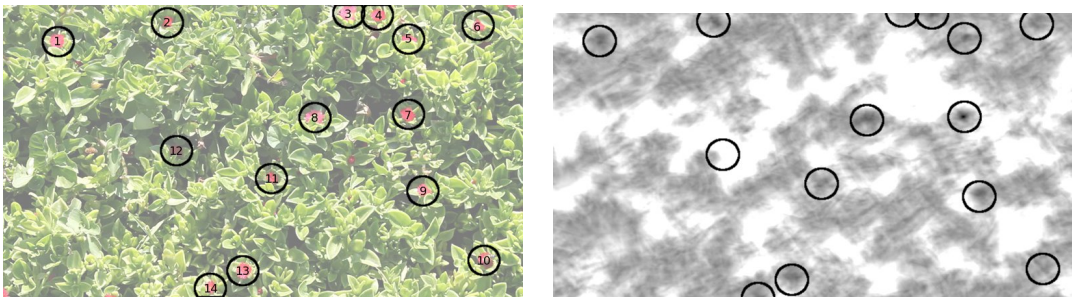
identify each region from the training imagery.

As mentioned earlier, we also assume that not all regions of interest are equally interesting implying that each region of interest has a different probability of an event occurring. Additionally, we expect this probability to be stable over some period of time T so that we can successfully adapt our sampling procedure accordingly. We expect that T is much larger than the amount of time required to sample each region of interest in succession. In the case of flowers, this duration is related to the lifetime of a flower [9]; in most cases more than a day.

Our assumptions about the interestingness of individual regions of interest to the phenomena are valid for a variety of applications, and specifically for pollinator behavior. Pollinators have been shown to obey optimal foraging theory [23] [9]. This theory, originally stated by MacArthur et. al [65], when applied to pollinators states that a pollinator chooses flowers that appear to be the most rewarding, they limit the effort expended on finding new rewarding flowers, they dwell on flowers that are in fact rewarding, and they ignore flowers found to be not rewarding. Since we expect that all pollinators have a similar definition of reward [56], this theory implies that each flower has individual probability of visitation proportional to its reward, and that this probability is stable while the reward persists.

4.3.2 Identifying Regions of Interest

The regions that we would like to sample are the particular regions of interest in the image; in the case of the pollination studies, the various flowers. Our goal is to detect and localize these regions of interest reliably so that we can later actuate our camera to gather higher resolution imagery of individual flowers. Like our previous formulation of ROI detection in Section 3.3.1, we anticipate that the



(a) Entire Scene (b) Template Similarity

Figure 4.3: The detection and localization of ROIs seen in the entire scene (a) is shown in the template similarity image (b). We can see that not all regions of interest can be localized, especially those that are significantly occluded.

ROI movement can be approximated by 0-mean Gaussian motion about some representative location (x_c, y_c).

$$\begin{aligned} x_i &= N(\mu = 0, \sigma_x) + x_c \\ y_i &= N(\mu = 0, \sigma_y) + y_c \end{aligned} \tag{4.1}$$

This movement is due to natural effects, like wind, that perturb the scene causing random and unpredictable motion. Thus, our goal is to identify this characteristic location by inspecting some sequence of training images.

To detect and localize the set of regions of interest present in the image, we have the user crop a single region of interest from a representative image. We then use template matching with a Euclidean distance similarity measure (as described in Section 3.3.1) to identify regions. Using only a single template image takes advantage of our assumption that all regions of interest look similar, and thus will respond strongly to the single extracted template image. The result is a number of distinct regions with high similarity as seen in Figure 4.3. In this visualization, darker implies more similar; the various regions were manually annotated in accordance with the actual flower locations.

In order to produce actual characteristic locations, we cluster all points within the 5th-percentile of the maximum similarity. Since we are using Euclidean distance to measure similarity, we use the 5th-percentile of the empirical distribution; if we were using cross-correlation, we'd use the 95th-percentile. All contiguous regions smaller than 5% of the template image's size are discarded, removing noise that could incorrectly generate nonexistent regions. To find a representative location for the individual regions, we extract regions from a number of representative training images. The identified locations are clustered, producing an average location.

This approach works quite well for many scenes. However, for regions that aren't sufficiently similar, like large flowers viewed with different orientations, using a single template image will likely fail. To surmount this problem, we can use multiple template images extracted from a representative frame by the user. We again use template matching to locate matching regions. However, instead of searching for all regions that respond strongly, we look for the single strongest response for each template. This is analogous to detecting and localizing a single flower described in Section 3.3.1. Though this modified approach requires more work to deploy, it will result in more reliable identification of the regions.

4.3.3 Adaptive Stratified Sampling

The simplest approach to sampling the phenomena in question would be to randomly sample the regions of interest. However, this does not exploit the fact that each region of interest has as unique and independent likelihood of visitation. Since we expect significant variation in the frequency of events across the regions, we choose to use an approach called adaptive stratified sampling. Here we defined each region of interest to be a stratum, or sub-population with re-

spect to all regions of interest present in the field of view. We must estimate the interestingness I_s of each stratum s and devote n_s samples to that region, such that $I_s \propto n_s$. That is, the more interesting the region, the larger fraction of total samples N are gathered while focused on that region. To estimate I_s we could employ an epoch based approach; we randomly sample each stratum for some fraction of the epoch to estimate I_s and then sample the phenomena proportional to I_s for the rest of the epoch. However, this approach wastes resources and is cumbersome.

Instead, we apply a specific instantiation of adaptive stratified sampling developed by Thompson et. al. [101] that produces a model-unbiased estimator of density without directly computing I_s without the need for sampling epochs. Again, we define an individual region of interest as a stratum and additionally define a sample as the dwell time of a camera focused on that stratum. Thus, our sampling algorithm progresses through each stratum sampling for some period of time τ. If an event is detected within that stratum during the camera's dwell time, the sample duration is increased to 2τ. The result of this sampling procedure is the number of events that were observed $\hat{Y}$ during the entire sampling period T; or simply an estimate of the density of events $\hat{D} = \hat{Y}/T$.

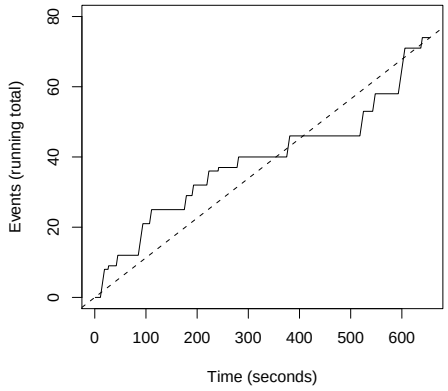


Figure 4.4: The running total of pollination events over time to a particular flower. We see that it follows a characteristic rate $\lambda = 0.113$.

Since we are no longer sampling randomly, we must show that the resulting model produces an unbiased estimator for the density D. We assume that each stratum s can be modeled as a Poisson process [72] with intensity λ_s and that it will

be sampled for a duration $n_s\tau$. Thus, conditional on λ_s and n_s, the number of observed events $\hat{Y}_s$ from stratum s for the duration of sampling is Poisson($\lambda_s n_s$). Figure 4.4 shows pollination events occurring to a single flower over time. Modeled as a Poisson process, it has a characteristics rate of $\lambda_s = 0.113$. Finally, the overall density is the sum of independent Poisson processes, which is also Poisson with intensity λ.

We are not interested in λ_s, the rate of events at a single stratum, we instead would like to show that $E[(\hat{D} - D)^2] = 0$, making $\hat{D}$ an unbiased estimator of the phenomena's density across all strata given this sampling procedure. From the Poisson assumptions we have:

$$E[\hat{D}_s|\lambda_s, n_s] = E\left[\frac{\hat{Y}_s}{n_s\tau}|\lambda_s, n_s\right] = \lambda_s \tag{4.2}$$

$$E[D_s|\lambda_s] = E\left[\frac{Y_s}{N_s\tau}|\lambda_s\right] = \lambda_s \tag{4.3}$$

So,

$$E[(\hat{D}_s - D_s)^2] = E\left[\frac{\lambda_s}{\tau}\left(\frac{1}{n_s} - \frac{1}{N_s}\right)\right] \tag{4.4}$$

$$E[(\hat{D} - D)^2] = \sum_s E\left[\frac{\lambda_s}{\tau}\left(\frac{1}{n_s} - \frac{1}{N_s}\right)\right] \tag{4.5}$$

Since $E[\bar{Y}_s|\lambda_s, n_s] = \lambda_s/(n_s\tau)$, an unbiased estimate of $E[(\hat{D} - D)^2]$ is:

$$\sum_s \bar{Y}_s n_s \left(\frac{1}{n_s} - \frac{1}{N_s}\right) \tag{4.6}$$

A rigorous analysis of this sampling methodology, showing that it produces an unbiased density estimate, can be found in [101].

Using adaptive stratified sampling to produce a density estimate requires we only learn a single parameter, the sample duration τ. We know that τ must be much larger that the minimum amount of time the PTZ camera takes to

pan and focus on a new target. From experimentation, this as found to be approximately 4 seconds. Also, τ should not be too large, or it would would ignore temporally coincident phenomena, accidentally introducing bias into the sampling procedure. In order to learn τ, we add a training phase to the beginning of our deployment when a human observer records events occurring to the stratum. Given these representative data, we run the procedure in simulation to determine the appropriate value for τ (more details about the simulation can be found in Section 4.4.2; an evaluation of our training methodology can be found in Section 4.5.2).

4.3.4 Detecting Events

The detection of events occurring nears regions of interest follows directly from our work in Chapter 3. However, this approach was not intended to work on-line, which is required for this application. Recall, that all parts of that method are on-line compatible except for the final tracking portion. The tracking algorithm chooses all frames in the video that meet the seed criteria, and tries to grow sequences around those frames. We modify this to look for a seed frame while we are sampling from a given stratum. If one is found, we greedily grow the sequence around that seed until it either runs into the bounds of the sampled sequence or exceeds the sequence criteria. This modification does not change any of the properties of the detection algorithm described earlier, so we expect it to perform similarly.

The training requirements of that procedure are an instance of the ROI cropped from a single frame, approximately 50 frames to build a background model, an instance of the event cropped from a difference image, and a few labeled images to derive the distributions of match values and frame-to-frame

target displacement. The exact number of frames required to train the background model on the ROI varies based on the complexity and motion of that ROI. Section 3.4.2 explains how to choose the number of required frames.

We intend to acquire these data at the inception of the deployment while we compute τ. The camera will be trained on each stratum in sequence for sufficiently long to gather frames for a background model and capture an event. A human would then be required to annotate these acquired data so the distributions for match value and frame-to-frame target displacement can be computed. Additionally, we must account for individual stratum appearing, changing their appearance, or disappearing; for example, a flower can blossom, change color, or whither during the course of a long deployment. To compensate for these effects, we also plan to retrain the background model at the start of each period of sampling; say, the beginning of each day. This would only require focusing on each stratum long enough to acquire approximately 50 frames.

4.3.5 Using Multiple Cameras

This procedure can be easily extended to use multiple cameras. The extra cameras can be trained on independent, adjacent, or identical fields of view. In all cases, the procedure we describe would actuate each camera independently. If all of the cameras are trained on the same field of view, regions of interest within that field of view could be divided amongst the available cameras. In this way, the field of view could be more densely sampled producing a more accurate density estimate.

Each of these methods for leveraging extra cameras have specific usages in the target applications. For example, collecting data for independent or adjacent fields of view is applicable to the work of Bartomeus et. al. where they are study-

ing separate environments containing a varying number of invasive plant species [5]. Work by Fontaine et. al. can leverage overlaid cameras, each tuned for a particular species of flower in the field of view [30]. This way, pollinator visitation density estimates can be acquired on a per flower species basis to determine if bumblebee's choice of flower is affected by density of pollinators in the region.

4.3.6 Generalizability

While describing our procedure, we have made a series of assumptions and explained how they are valid for the particular application of interest. These assumptions fall into two categories: requirements of the regions of interest within the scene, and characteristics of the novel event. We argue that they are generally applicable, such that this procedure can be reused for a variety of other applications.

We have required all regions of interest to be relatively similar in appearance. We have proposed an approach that uses multiple regions templates to tolerate the difference in appearance, and will show that it performs significantly better than using only a single template. Additionally, we require that the various regions of interest do not occlude one another. For many applications, we believe that there is some vantage point that would allow this assumption to hold. When this is impossible, we could attempt to track the regions of interest over time using an object tracking approach that tolerates occlusion. For example, we could use template matching to localize the target in each frame. Then instead of clustering between frames to find an average location, we could track those points over time, accounting for occlusions similar to work by Sethi et. al. [92]; we leave this modification to future work.

Fundamental to our sampling approach is the temporal correlation between

events; both the fact that an event occurring at time t implies a higher likelihood of an event at time $t+1$ and that event probability at a given stratum is stable over some time T. As mentioned earlier, this is true for pollinator visitation, but we argue that it is also applicable to many other applications. Specifically, this assumptions holds for any animal visitation phenomena that obeys optimal foraging theory [65], like the grazing of deer [52] or the foraging of black bears [71].

4.4 Experimental Setup

In principle, we could have taken an implementation of our approach into the field and attempted to estimate the density of a particular phenomena. In our case, that would entail taking a pan-tilt-zoom (PTZ) camera into the field where we could capture pollinator visitation to flowers in some patch of ground.

Unfortunately, it would be difficult to thoroughly evaluate its effectiveness for a variety of reasons. First, we'd have no ground truth data to compare our estimate against. Second, it would be difficult to evaluate the effect of changing τ, the minimum time spent sampling an individual stratum. Third, iterating on the design and reproducing the evaluation would be complex due to changing field conditions between iterations. Finally, any data collection efforts would be subject to the availability of the phenomena in the field. In the case of pollinator behavior, this is tightly coupled with the blooming of local flowers, a transient event. To surmount these difficulties, we collected high-resolution imagery and simulated camera actuation and subsequent sampling.

Figure 4.5: An example frame from a dataset acquired to evaluate our approach. Here, the individual stratum are identified and labeled.

4.4.1 Data Acquisition

The set of biological studies driving this work wish to gather data about pollinator visitation density. Thus, to evaluate our procedure, we collect video of pollinator visiting a patch of flowers; this patch of flowers is in fact the same patch as the *IcePlant* data sets collected in Chapter 3. We chose to capture high-resolution video of this patch of ground using an HD video camera [15]. This video was captured at 1920x1080 at 20Hz and spanned an entire $1m^2$ patch of flowers.

Using this video, we can emulate a 640x480 PTZ camera with 3x zoom by simply scaling and cropping the frames appropriately. This allows us to have ground truth data for the entire scene so we can properly evaluate our approach.

Data Set	Strata	Day
IcePlant1	14	0
IcePlant2	10	1
IcePlant3	13	2

Table 4.1: The datasets, collected at 10am on three sequential days, were each 15 minutes in length, but differed in the number of strata present in the scene.

To gather this ground truth, each frame was manually segmented into strata and each (frame, strata) pair was manually annotated when a pollinator was present. Figure 4.5 shows a single frame from the *IcePlant1* dataset with the stratum identified. Three datasets, listed in Table 4.1 were acquired and analyzed in this fashion. Each dataset had a duration of 15 minutes. Though the study could have lasted much longer, we limit our analysis to 15 minutes so that we can more easily inspect the ground truth data.

4.4.2 Camera Simulation

With these datasets in hand, we faithfully simulated the entire system. Not only does this simulation help us experiment with the sampling algorithm, it will also be used to train the sampling algorithm for future deployments. The main components of our simulation were the camera's actuation and the target detection. To reproduce the camera in simulation, we experimentally measured the amount of time it takes to pan and focus on a new target when programatically actuated. The exact duration depends on the actuation required, but the motion was typically completed in 4 seconds.

This simulation frees us to experiment with the value of τ as well as the number of cameras to deploy. We can easily vary τ and measure the affect it has on the density estimate. Similarly, we can divide the field of view into sections, one per virtual camera, and see how many cameras are needed to improve our density estimate.

To reproduce the target detection, we implemented the various failure modes described in Section 3.4.3. The simulation reads the human labeled, ground truth data and randomly chooses to permute it, inducing error. Each permutation emulates either removed subsequence error or an added sequence error, the two

characteristic forms of error present in our discrete signal estimation algorithm (Section 3.4.3). We will use this mechanism to induce detection errors comparable to those seen in our previous experiments. Since we saw approximately 3 times more recall errors than precision errors, we will induce 3 remove-type errors to every one add-type error.

4.5 Evaluation

For our evaluation, we test our the accuracy of our procedure when gathering pollinator visitation density for a single patch. For an actual biological study, multiple independent patches would be monitored. However, we only need to consider a single patch since the phenomena occurring at each are independent [5].

For these applications, the biologists would like to compare density estimates. There is no particular error bounds required by the domain scientists, but any significant error reduces there ability to draw statistically significant conclusions. Existing studies [5] have measured pollinator visitation density to between 5% and 15% error. Thus, we strive to measure the density of events per unit time to within 10% of its actual value. Our analysis is split into three separate parts: evaluating our ability to reliably localize the strata, the effectiveness of the training methodology used to tune dwell time for a particular stratum, and an assessment of the benefit gained from deploying multiple cameras to a single patch.

4.5.1 Localizing Strata

Correctly locating the strata is crucial to the success of our sampling. If the strata are incorrectly identified, it will induce error in our density estimate by

	Single Template		Multi-Template	
	Precision	Recall	Precision	Recall
IcePlant1	66.7%	85.7%	99.8%	99.8%
IcePlant2	61.5%	80.0%	99.3%	99.3%
IcePlant3	70.5%	92.3%	99.6%	99.6%

Table 4.2: The precision and recall of the two approaches for locating strata over the course of the entire video sequence. Though using multiple templates requires more user input, it is clearly worth the cost.

either having the camera focus on regions that don't contain an actual ROI or by ignoring regions that actually do contain an ROI. We have proposed two similar approaches; one requires the user locate a single representative region, and the other requires the user identify all regions of interest. Regardless, we evaluate the results using the same criteria in both cases: precision and recall of region recognition and localization. We further require that we can correctly identify this regions for the duration of the video captured. A region is said to be correctly localized if the bounding box identified completely contains the region of interest.

An example result of applying our procedure using a single template to a frame from the *IcePlant1* dataset can be seen in Figure 4.6. All of the black-colored blobs are regions thought to contain a stratum, where as the black circles identify regions that are actually flowers. As we can plainly see, this approach produces a number of false positives while not even producing 100% recall. Instead, if we use one template image per stratum,

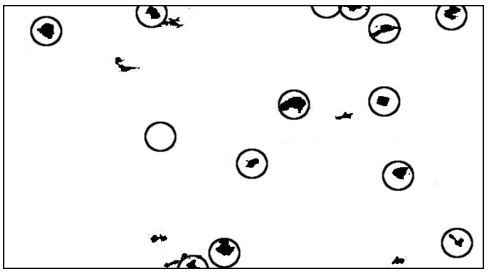


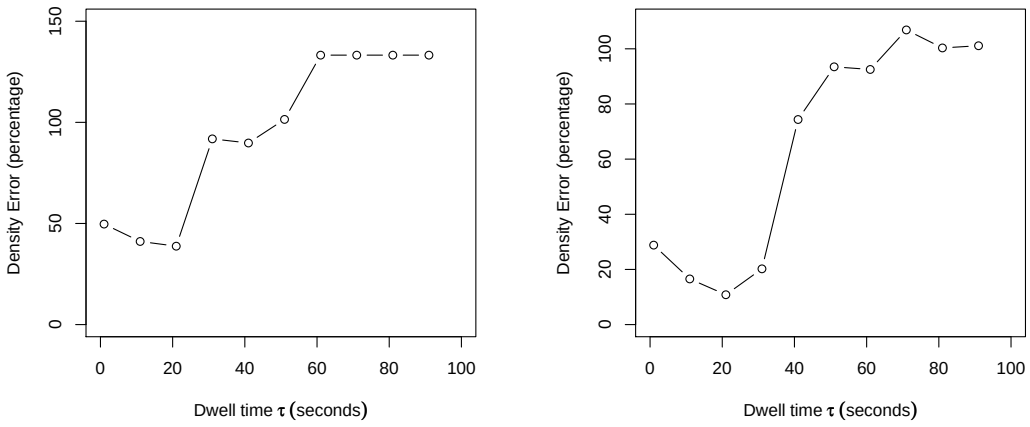
Figure 4.6: The localization of strata using a single template for the *IcePlant1* dataset. Any pixel whose similarity is within the 5th-percentile of the maximum similarity is represented in black. The circles represent the actual location of the strata.

we produce exactly only possible localization for each stratum, and can correctly identify all 14 strata with almost no error (Table 4.2). Since a single region is identified for each template, precision and recall values for each dataset are identical since each localization failure will affect them identically. Although it can become a maintenance burden and is more time consuming to have the user identify all strata present once per deployment, the resulting precision and recall of identification are certainly worth the added expense.

Once we identify potential strata in a single frame, we then need to cluster these regions across frames, computing an average location. We keep adding frames until the average location of the region stabilizes. For the three datasets we've captured, the motion of the individual flowers plus the motion of the camera was so minor that 5 frames was sufficient to stabilize our location estimates. When more motion is present, like the *Manzanita* datasets evaluated in Section 3.4, we expect that an order of magnitude more frames will be needed to stabilize these estimates. Still, even if we have to capture 50 frames of training data, it is a similar number of training frames we will require to train the sampling model for camera actuation (see below) or the background subtraction model for event detection (see Section 3.4.2).

4.5.2 Sampling Model Parameters

There is one fundamental tuning parameter for our process, the time constant τ. This constant defines the initial duration that a given stratum is sampled before moving to the next. If this value is too small, we will spend too much time panning between stratum and not enough time sampling the phenomena. If the value is too large, we will bias our estimate toward a single stratum, ignoring the globally occurring phenomena.



(a) Training Error

(b) Full Dataset Error

Figure 4.7: The percentage error in our density estimate as we vary the dwell time on a single strata. The training error (a) was computed by only considering first 1/6 of the data set. When we consider the entire dataset (b) we can compute the optimal dwell time these data.

To compute τ, we simulate the phenomena and the sampling procedure on the initial 2 minutes of data. As discussed in Section 4.3.3, this period would be part of the initial training period where labeled data would be available. Thus, we try all possible values of τ between the 5 seconds (the minimum suggested by the pan duration) and 100 seconds. As expected, we see that neither small nor large values of τ are appropriate, Figure 4.7(a). The values displayed here represent a window of 10 values of τ surrounding the data point. We average because a particular value of τ may do particularly well on the training data if it transitions at exactly the right time by chance. Over-fitting in this way would reduce the generalizabilty of the τ chosen.

In this case, the optimal value of τ is approximately 20 seconds, resulting in a 19.53% error in density (0.095 events/sec measured versus 0.068 events/sec actual). When we apply this value of τ to the entire dataset, we find that we get a

10.82% error in density (0.056 event/sec measured versus 0.062 events/sec actual). Figure 4.7(b), shows that $\tau = 20$ is the optimal value when considering the entire dataset. This means that our training methodology produced an appropriate choice of τ for the measured phenomena.

It is important to note, that the value of τ is dependent on the phenomena but not specific to the strata currently present in the scene. That is, as the strata change, say new flowers bloom and old flowers whither, we can still apply the same values of τ for our analysis [56]. Using $\tau = 20$ seconds, we computed the density of pollinator visitation for the remaining two datasets, *IcePlant2* and *IcePlant2*. We see in Table 4.3 that the resulting error in our density estimate remains stable. This implies both that the phenomena of pollinator visitation is relatively stable and that our sampling methodology is able to repeatably capture an accurate density estimate.

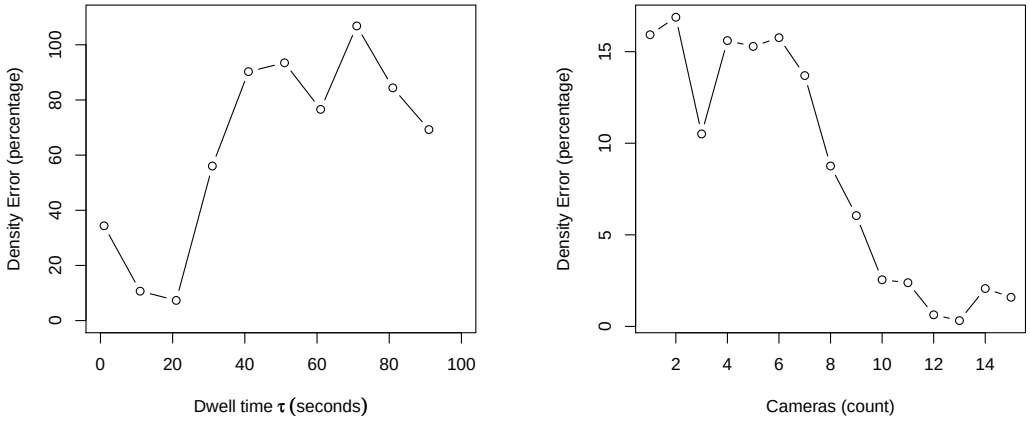
Day	Strata	Density Error
0	14	10.82%
1	10	11.43%
2	13	11.17%

Table 4.3: The percentage error in our density estimate using $\tau = 20$ seconds on the three datasets. We see that the error is relatively consistent event though the number of strata changed.

4.5.3 Utility of Added Cameras

If we wish to further reduce the error in our density estimates, we can employ more cameras to more densely sample the phenomena. Clearly, having more cameras than stratum is wasteful, but the utility of each additional camera is not immediately clear. In particular, the utility is likely tied to the phenomena; if particular flowers are never visited by pollinators, there is no reason to devote a camera to sample that flower.

For our particular deployment, the utility of additional cameras on the



(a) Density Estimate Error

(b) Added Camera Utility

Figure 4.9: We approximately double the number of strata by combining the *IcePlant1* and *IcePlant2* datasets in simulation. (a) The percentage error in our density estimate as we vary the dwell time on a single strata. (b) The utility of added cameras on this simulated data with $\tau = 20$.

IcePlant1 dataset can be seen in Figure 4.8. As expected, as the number of cameras increase, the error approaches zero. Even with fewer than half as many cameras as strata (6 cameras and 14 strata), we can achieve less than 4% error in our density estimates. Depending on the specific requirements of the biological study, deploying 6x more cameras may be worth the added expense.

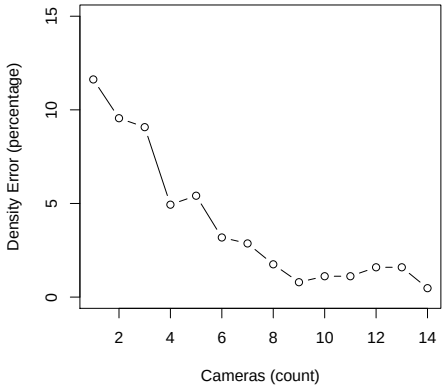


Figure 4.8: The absolute error between the actual density and the measured density decreases as the number of cameras increase.

We can further stress our procedure by simulating a patch with approximately double the number of flowers by simulating a dataset containing both the *IcePlant1* and *IcePlant2* datasets side by side. We see that $\tau = 20$ is again the

optimal value for this simulated dataset containing 24 strata (Figure 4.9(a)), further reinforcing that this value is characteristic of the phenomena itself. When adding more cameras under these conditions, we find that it again takes about half as many cameras as strata (11 cameras and 24 strata) to reduce to about 5% (Figure 4.9(b)). Interestingly, we notice that the progression to this state is somewhat erratic. This is an amplified version of similar behavior we see when considering fewer strata. As we add more cameras, their utility does not become pronounced until each camera has fewer than 5 strata to sample. Thus, we assert that we can sample a large region containing upwards of 20 strata with density estimate accuracy sufficient for biologists, but to reduce the error to below 5% each camera must be assigned fewer than 5 strata. Still, it is important to note that this procedure can produce arbitrarily accurate density estimates by devoting more resources to sampling the phenomena.

4.6 Related Work

There are a number of fields that have produced work related to the procedure we describe. Here, we discuss related sampling approaches and we have already examined approaches related to event detection in Section 3.5.

As discussed earlier, the adapting our sampling approach to the phenomena can be achieved using either a one-phase or two-phase approach. These phases represent when information is collected about the location and density of the phenomena, and when the output data is actually collected. The two-phase approach measure properties about the phenomena in the first phase, defining a plan of action for the second phase. In contrast the one-phase approach continually adapts, lengthening the time available to collect the actual data. A review of such one- and two-phase approaches to adaptive stratified sampling is presented

by Turk et. al. [103]. Their key observation is that no signal sampling strategy is appropriate for all applications, as support by numerous studies and simulations. Thus, we only choose to consider sampling approaches from the literature that attempt to sample data for functionally similar applications.

Two-Phase Sampling

One related application is trawling for fish to determine their density in a particular region of ocean [32]. Like our applications, thee fish cluster in certain areas that are not known *a priori* and must be located in order to effectively determine their density. These regions are analogous to our flowers that can be visually detected. Still, even though we now have a hint as the probably locations of the events, we have no understanding of which regions are likely to attract more pollinators. For the fish trawling study, the employ a two phase approach; first locating the regions with large quantities of fish and subsequently returning to sample those regions more densely. They found that this straight forward approach produced decent results both in simulation and in practice.

An extension of this two-phase approach is presented by Conroy et. al. [19]. Instead of simply using the values acquired by the first phase to inform the sampling in the second phase, they model the phenomena measured in the first phase using a Bayesian formulation. Using this formulation, they can propagate the error in the initial phase into the second phase, placing more accurate error bounds on the final estimate.

One-Phase Sampling

The one-phase sampling approach developed by Thompson et. al. [100] [101] forms the basis of our work. Similar to [32], the application considered is trawling

for marine life, in this case shrimp. They propose a one-phase approach that updates the trawl length based on the catch in the current trawl. We adapted this to length the sample duration of a particular flower when a target was found on that flower. This takes advantage of the characteristic patchiness of both the marine life and pollinator visitation phenomena. The approach of updating the trawl length is shown to produce an unbiased estimator of the density of events even though the sampling procedure is being updated on-line.

In more recent work, Thompson et. al. [99] extend their one-phase approach to better deal with phenomena where the spatial component can be expressed as a graph, the social graph when considering the spread of diseases. Weighted links in the graph and information about neighbors are used overcome minimal evidence that would otherwise limit sampling of particular nodes. This work can be applied to our application if we consider adjacent flowers in a region to represent a graph whose edges are weight by the likelihood of a pollinator passing from one flower to a given neighbor. However, after having analyzed the data, we found little correlation between events on flowers, leading us to believe that the phenomena occurring on each flower is independent. Still, in when this independence does not exist, this graph-based approach has the potential to compensate for the dependent effects.

4.7 Conclusion

Using imagers to increase the data collection coverage of novel events occurring to regions of interest is clearly valuable to scientists. Using pollinator visitation as the driving application, we have shown that the procedure we've developed can successfully produce accurate density estimates for short, somewhat rare events. We are able to leverage our previous work to detect events by actuating cameras to

gather higher resolution, narrower field of view imagery of the individual regions of interest. The key innovation of our procedure is the novel use of adaptive stratified sampling to we focus our limited resources on regions likely to be visited. Finally, we show that by devoting more resources to the sampling effort, we can decrease the error in our density estimate.

Having this tool at their disposal, biologists can devise more elaborate studies than are currently possible due to the volume of data collection required.

CHAPTER 5

Future Work

There are a variety of future directions for our work, the most important of which are the distillation a reusable toolkit for each of the three template procedures, and further deployments to collect more biologically relevant data and evaluate the robustness of these approaches. There are also a set of algorithmic improvements we have suggested that may be required for these future deployments.

As the software currently exists, it is not yet packaged for broad use. Though all the procedures are automated and have reasonable computational properties, they have minimal documentation, require system knowledge to debug failures, required user workflow (e.g. labeling imagery or template extraction) is not well described, and evaluation mechanisms are not centralized. To polish the implementation, we plan to build a web interface for interacting with the software, which would run locally on the user's computer. It would walk the user through labeling or inputting training data, showing both training and testing results in graphical form using the same metrics we have previously used to evaluate these procedures.

The long term impact of our work will be seen through its deployment in future image-based sensing systems and its influence on the design of related systems. The goal of these deployments is to collect biologically useful data that will directly be used in a biological study. Example applications we describe are carefully modeled after existing studies, but we plan to directly apply of

our procedures to perform previously intractable biological studies. An initial candidate for our continuous signal procedure is the inference of CO_2 flux from the years of imaging data collected by MossCam [36] at James Reserve. For our discrete signal procedure, we plan to perform a pollinator studies to better understand how optimal foraging theory [65] applies to bees.

Driven by the additional requirements of future deployments, we may need to consider various algorithmic improvements to our template procedures, though their high-level form will remain unchanged. When estimating continuous signals we can improve our lighting model by jointly considering the possible illuminations and relative spectral reflectance to limit our modeling effort to the possible visual appearances of the subject. For discrete signals, we can improve our tracking approach to handle multiple targets. Additionally, we can improve the training of our background model by iteratively training the model by removing training frames contained in the final detection output (since they likely contain the target), and continuing until the output converges. Further algorithmic improvements will no doubt present themselves as our work is extended into new application domains.

A final extension to our work is the design of additional template procedures for new types of biological signals. For these systems, the primary goal would be to isolate changes the particular target of interest from changes to the local environment; emulating our approach to extract color in the presence of changing lighting or the region of interest in the presence of significant natural motion. Through this isolation, new field robust template procedures can be constructed.

CHAPTER 6

Conclusion

In this dissertation, we have discussed three template procedures that can be used to build image-based sensor for continuous, discrete, and discrete spatio-temporal biological phenomena. Each of these signals are extracted from natural processes that evolve somewhat predictably over time, as opposed to the detection of discrete, rare, and novel events. Leveraging signal-specific mechanisms, we isolate the changes in subject of interest from changes in the background environment, making the procedures more resilient to changing field conditions. We have shown how to effectively model the target signals using a combination of training data from both the field and the laboratory. Finally, we have evaluated our template procedures in the context of specific applications and argued for the procedure's generalizability due to their limited assumptions.

We consider the CO_2 flux from a drought-tolerant moss *Tortula princeps* as an application for predicting continuous signals. With little more than the color of the subject available as input, the template procedure for continuous signals relies heavily on color image features. Using color features directly is brittle due to the effect of changing illumination. Thus, we have developed an approach to predict and compensate for the natural changes in the incident illumination present in the scene, producing stable image features that represent changes to the moss alone. Through the combination of laboratory measurements of CO_2 flux and field measurements of lighting, we have shown in simulation that our

model can predict CO_2 flux to 0.439ppm, which is within the 0.5ppm error bounds acceptable to domain scientists.

Extracting frames from a video sequence that depict the presence of pollinators on a flower was the application used to evaluate the template procedure for discrete signals. Natural motion in the scene, typically caused by wind, causes significant variation in the appearance of the background and the location of the foreground (the flower and more importantly the pollinator). To discard much of the confounding motion in the scene, we automatically extracted a region of interest (ROI), the flower, from the scene. This approach has two beneficial effects: First, the phenomena dictates that the natural motion of the target becomes much slower and predictable when it approaches the ROI. Second, registering the image in this way significantly simplifies further processing such as background modeling and target motion modeling. Using the procedure we describe, we are able to achieve at worst 90% extraction precision and 80% extraction recall, in many cases achiving perfect precision and recall.

Scaling the previously mentioned pollinator application to estimate the density of pollinator presence events over an entire field of flowers, rather than a single flower, is an example of a discrete spatio-temporal signals. Video imagery of the entire scene contains a mix of regions of interest (the flowers) and regions that can be safely discarded (the surrounding leaves and ground). We automatically identify the regions of interest and consider each an independent stratum when applying an adaptive sampling procedure to preferentially sample regions that are seeing more activity. Isolating the regions in this way allows us to both reuse the discrete signal template procedure when considering a single stratum and produce a model unbiased estimator of event density across all strata. This approach yields a density estimate with approximately 10% error, which can be

further reduced by using additional cameras.

In addition to the template procedures we have discussed, our primary contribution is the consistent structure of these procedures that makes them robust to field conditions: isolate the target signal from the naturally induced background noise, be it lighting, motion, or other confounding effects. Then, directly model the signal from relevant image features. We believe that this form is common to all image-based sensor deployments and have discussed the applicability of this approach to three specific types of biological signals.

References

[1] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[2] K. Barnard. *Practicle Color Constancy*. PhD thesis, 2000.

[3] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms. Part I: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–984, 2002.

[4] K. Barnard, F. Ciurea, and B. Funt. Sensor sharpening for computational color constancy. *Journal of the Optical Society of America A*, 18(11):2728–2743, 2001.

[5] I. Bartomeus, M. Vila, and L. Santamaria. Contrasting effects of invasive plants in plant–pollinator networks. *Oecologia*, 155(4):761–770, 2008.

[6] S. Beauchemin and J. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[7] M. Betke, D. Hirsh, A. Bagchi, N. Hristov, N. Makris, and T. Kunz. Tracking large variable numbers of objects in clutter. 2007.

[8] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35(99-109):4, 1943.

[9] J. Biernaskie, S. Walker, and R. Gegear. Bumblebees Learn to Forage like Bayesians. *The American Naturalist*, 174(3), 2009.

[10] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. *Intel Corporation, Microprocessor Research Labs*, 2000.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall, 1984.

[12] R. B. Brown and S. D. Noble. Site-specific weed management: sensing requirements – what do we need to see? *Weed Science*, 53(2):252–258, 2005.

[13] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*, chapter 3 and 4. Wiley, 2009.

[14] Canon. Canon 450d. http://www.usa.canon.com/consumer/controller?act=ModelInfoAct&fcategoryid=139&modelid=16303.

[15] Canon. Canon 500d. http://www.usa.canon.com/consumer/controller?act=ModelInfoAct&fcategoryid=139&modelid=18385.

[16] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[17] D. Cline, D. Edgington, and J. Mariette. An automated visual event detection system for cabled observatory video. *IEEE Oceans*, 2007.

[18] R. Colwell, G. Brehm, C. Cardelus, A. Gilman, and J. Longino. Global warming, elevational range shifts, and lowland biotic attrition in the wet tropics. *Science*, 322(5899):258, 2008.

[19] M. Conroy, J. Runge, R. Barker, M. Schofield, and C. Fonnesbeck. Efficient estimation of abundance for patchily distributed populations via two-phase, adaptive sampling. *Ecology*, 89(12):3362–3370, 2008.

[20] A. Cooley, G. Carvallo, and J. Willis. Is Floral Diversification Associated with Pollinator Divergence? Flower Shape, Flower Colour and Pollinator Preference in Chilean Mimulus. *Annals of Botany*, 2008.

[21] I. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[22] M. A. Crimmins and T. M. Crimmins. Monitoring Plant Phenology Using Digital Repeat Photography. *Environmental Management*, 41:949–958, 2008.

[23] M. Cruzan. Pollinator behavior, 2007. http://web.pdx.edu/~cruzan/PollinatorBehavior.pdf.

[24] D. Edgington, D. Walther, K. Salamy, M. Risi, R. Sherlock, and C. Koch. Automated event detection in underwater video. *IEEE Oceans*, 2003.

[25] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[26] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics*, chapter 35: Color Vision. Addison-Wesley, 1963.

[27] G. D. Finlayson. Color in Perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1034–1038, 1996.

[28] G. D. Finlayson and S. D. Hordley. Color constancy at a pixel. *Journal of the Optical Society of America A*, 18(2):253–264, 2001.

[29] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Colour by correlation: a simple, unifying approach to colour constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1209–1221, 2001.

[30] C. Fontaine, C. Collin, and I. Dajoz. Generalist foraging of pollinators: diet expansion at high density. *Journal of Ecology*, 96(5):1002–1010, 2008.

[31] D. Forsynth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–36, 1990.

[32] R. Francis. An adaptive strategy for stratified random trawl surveys. *New Zealand Journal of Marine and Freshwater Research*, 18(1):59–71, 1984.

[33] J. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–141, 1991.

[34] A. Gitelson, Y. Gritz, and M. Merzlyak. Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves. *Journal of Plant Physiology*, 160(3):271–282, 2003.

[35] A. A. Goshtasby. *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*. Wiley, 2005.

[36] E. A. Graham, M. P. Hamilton, B. D. Mishler, P. W. Rundel, and M. H. Hansen. Use of a Networked Digital Camera to Estimate Net CO_2 Uptake of a Desiccation-Tolerant Moss. *International Journal of Plant Sciences*, 167:751–758, 2006.

[37] M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. *Lecture notes in computer science*, pages 543–560, 2002.

[38] T. Hastie and R. Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society*, pages 757–796, 1993.

[39] F. Hill. *Computer Graphics Using OpenGL*, chapter 12: Color Theory. Prentice Hall, second edition, 2001.

[40] S. D. Hordley and G. D. Finlayson. Reevaluation of color constancy algorithm performance. *Journal of the Optical Society of America A*, 23(5):1008–1020, 2006.

[41] A. Huett. A soil-adjusted vegitation index (SAVI). *Remote Sensing of Environment*, 25:53–70, 1988.

[42] J. Hyman, E. Graham, M. Hansen, and D. Estrin. Imagers as sensors: Correlating plant CO_2 uptake with digital visible-light imagery. *Workshop on Data Management in Sensor Networks*, 2007.

[43] J. Hyman, M. Hansen, and D. Estrin. Estimating the Spectral Reflectance of Natural Imagery Using Color Image Features. *Workshop on Applications, Systems, and Algorithms for Image Sensing*, 2008.

[44] International Commission on Illumination. Cie. `http://www.cie.co.at`.

[45] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 631–637, Washington, DC, USA, 2005. IEEE Computer Society.

[46] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[47] H. Ives. The relation between the color of the illuminant and the color of the illuminated object. *Color Research and Application*, 20:70–70, 1995.

[48] Joint Photographic Experts Group. Jpeg compression. `http://www.w3.org/Graphics/JPEG/itu-t81.pdf`.

[49] D. Judd, D. MacAdam, G. Wyszecki, et al. Spectral distribution of typical daylight as a function of correlated color temperature. *Journal of the Optical Society of America*, 54(8):1031–1040, 1964.

[50] D. Karcher and M. Richardson. Quantifying Turfgrass Color Using Digital Image Analysis. *Journal of Crop Science*, 43(3):943, 2003.

[51] Y. Kaufman and D. Tanre. Atmospherically resistant vegetation index (ARVI) for EOS-MODIS. *IEEE Transactions on Geoscience and Remote Sensing*, 30(2):261–270, 1992.

[52] J. Kie, C. Evans, E. Loft, and J. Menke. Foraging behavior by mule deer: the influence of cattle grazing. *The Journal of Wildlife Management*, 55(4):665–674, 1991.

[53] T. Ko, S. Ahmadian, J. Hicks, M. Rahimi, D. Estrin, S. Soatto, S. Coe, and M. P. Hamilton. Heartbeat of a Nest: Using imagers as biological sensors. *ACM Transaction on Sensor Networks*, Vol. 6 Iss. 3, 2010.

[54] T. Ko, S. Soatto, and D. Estrin. Background Subtraction with Distributions. *European Conference on Computer Vision*, 2008.

[55] S. Kullback. *Information Theory and Statistics*. Wiley, 1959.

[56] D. Lefebvre, J. Pierre, Y. Outreman, and J. Pierre. Patch departure rules in Bumblebees: evidence of a decremental motivational mechanism. *Behavioral Ecology and Sociobiology*, 61(11):1707–1715, 2007.

[57] C. J. Legleiter, D. A. Roberts, A. Marcus, and M. A. Fonstad. Passive optical remote sensing of reiver channel morphology and in-stream habitat: Physical basis and feasibility. *Remote Sensing of the Environment*, 93:493–510, 2004.

[58] J. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, 10:120–123, 1995.

[59] Licor Biosciences. Licor 1800. `http://www.licor.com/env/Support/discontinued/li1800.jsp`.

[60] Licor Biosciences. Licor 6262. `http://www.licor.com/env/Products/GasAnalyzers/li6262/6262.jsp`.

[61] Y. Liu, J. Sarnat, B. Coull, P. Koutrakis, and D. Jacob. Validation of MISR Aerosol Optical Thickness Measurements Using AERONET Observations over the Contiguous United States. *Journal of Geophysical. Research*, 2003.

[62] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Corfu, Greece, 1999.

[63] B. Lucas. *Generalized image matching by the method of differences*. PhD thesis, 1985.

[64] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679. Citeseer, 1981.

[65] R. MacArthur and E. Pianka. On optimal use of a patchy environment. *The American Naturalist*, 100(916):603, 1966.

[66] L. T. Maloney and B. A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, 1987.

[67] J. Marchant and C. Onyango. Shadow-invariant classification for scenes illuminated by daylight. *Journal of the Optical Society of America A*, 17(11):1952–1961, 2000.

[68] J. Marchant and C. Onyango. Spectral invariance under daylight illumination changes. *Journal of the Optical Society of America*, 19(5):840–848, 2002.

[69] J. Marchant, N. Tillett, and C. Onyango. Dealing with Color Changes Caused by Natural Illumination in Outdoor Machine Vision. *Cybernetics and Systems*, 35(1):19–33, 2004.

[70] D. H. Marimont and B. A. Wandell. Linear models of surface and illuminant spectra. *Journal of the Optical Society of America A*, 9(11):1905–1913, 1992.

[71] R. Mazur and V. Seher. Socially learned foraging behaviour in wild black bears, Ursus americanus. *Animal Behaviour*, 75(4):1503–1508, 2008.

[72] M. I. Miller and D. L. Snyder. *Random Point Processes in Time and Space*. Springer-Verlag New York Inc, 2nd edition, 1991.

[73] M. Mirik, G. M. Jr., S. Kassymzhanova-Mirik, N. Elliott, V. Catana, D. Jones, and R. Bowling. Using digital image analysis and spectral reflectance data to quantify damage by greenbug (Hemitera: Aphididae) in winter wheat. *Computers and Electronics in Agriculture*, 51:86–98, 2005.

[74] S. K. Mitra and J. F. Kaiser. *Handbook for Digital Signal Processing*. Wiley, 1993.

[75] S. Navntoft, S. Wratten, K. Kristensen, and P. Esbjerg. Weed seed predation in organic and conventional fields. *Biological Control*, 49(1):11–16, 2009.

[76] P. Nobel. PAR, Water, and Temperature Limitations on the Productivity of Culitvated Agave Forcroydes (Henequen). *Journal of Applied Ecology*, pages 157–173, 1985.

[77] O. Noboru and A. R. Robertson. *Colorimetry*, chapter 3.9: Standard and Supplementary Illuminants. Wiley, 2005.

[78] L. Noldus. The Observer: a software system for collection and analysis of observational data. *Behavior research methods, instruments & computers*, 23(3):415–429, 1991.

[79] M. J. Oliver, J. Velten, and A. J. Wood. Bryophytes as experimental models for the study of environmental stress tolerance: *Tortula ruralis* and desiccation-tolerance in mosses. *Plant Ecology*, pages 73–84, 2000.

[80] OpenCV. `http://opencv.willowgarage.com`.

[81] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, pages 555–562, 1998.

[82] B. Patullo, G. Jolley-Rogers, and D. Macmillan. Video tracking in the extreme: Video analysis for nocturnal underwater animal movement. *Behavior Research Methods*, 39(4):783, 2007.

[83] Pentax. Pentax s5z. `http://www.pentaximaging.com/files/manual/OptioS5z_web.pdf`.

[84] A. Peters. *Lectures on Image Processing*, chapter 7: Convolution. archive.org, 2007.

[85] M. C. Proctor, M. J. Oliver, A. J. Wood, P. Alper, L. R. Stark, N. L. Cleavitt, and B. D. Mishler. Desiccation-tolerance in bryophytes: a review. *The Bryologist*, 110(4):595–621, 2007.

[86] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer, 1997.

[87] K. Rangarajan and M. Shah. Establishing motion correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91.*, pages 103–108, 1991.

[88] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[89] J. Rouse, R. Hass, J. Schell, and D. Deering. Monitoring the Vernal Advancement and Retrogradation (Greenwave Effect) of Natural Vegetation. *Proceedings of the Remote Sensing Center*, 1974.

[90] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[91] I. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology*, 151(2):182–195, 2005.

[92] I. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on pattern analysis and machine intelligence*, 9(1):56–73, 1987.

[93] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 27(1):51–65, 2005.

[94] T. Shi, E. Clothiaux, B. Yu, A. Braverman, and D. Groff. Detection of daytime arctic clouds using misr and modis data. *Remote Sensing of Environment*, 2006.

[95] D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley-Interscience, 2006.

[96] D. Slaughter, D. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1):63–78, 2008.

[97] H. Tao, H. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 75–89, 2002.

[98] A. J. Theuwissen. *Solid-State Imaging with Charge-Coupled Devices*. Kluwer Academic Print on Demand, 1995.

[99] S. Thompson. Adaptive web sampling. *Biometrics*, 62(4):1224–1234, 2006.

[100] S. Thompson, F. Ramsey, and G. Seber. An adaptive procedure for sampling animal populations. *Biometrics*, 48(4):1195–1199, 1992.

[101] S. Thompson, G. Seber, and O. Okogbaa. *Adaptive sampling*, chapter 1 and 7. Wiley New York, 1996.

[102] Z. Tuba, Z. Csintalan, and M. Proctor. Photosynthetic response of a moss, *Tortula ruralisi*, ssp. *ruralis*, and the lichens *Cladonia convolta* and *C. furcata* to water deficit and short periods of desiccation, and their ecophysiological significance: a baseline study at present-day CO_2 concentration. *New Phytologist*, pages 353–361, 1996.

[103] P. Turk and J. Borkowski. A review of adaptive cluster sampling: 1990–2003. *Environmental and Ecological Statistics*, 12(1):55–94, 2005.

[104] S. L. Ustin, D. A. Roberts, J. A. Gamon, G. P. Asner, and R. O. Green. Using Imaging Spectroscopy to Study Ecosystem Processes and Properties. *BioScience*, 54(6):523–534, 2004.

[105] V. Vapnik and S. Kotz. *Estimation of dependences based on empirical data*. Springer-Verlag New York Inc, 2006.

[106] C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[107] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume 1, pages 511–518, 2001.

[108] B. A. Wandell. *Foundations of vision*. Sinauer Associates, 1995.

[109] Wikipedia. Chlorophyll spectra. `http://en.wikipedia.org/wiki/Chlorophyll`.

[110] Wikipedia. Cie 1931 color space. `http://en.wikipedia.org/wiki/CIE_1931`.

[111] J. Worthey and M. Brill. Heuristic analysis of von Kries color constancy. *Journal of the Optical Society of America A*, 3(10):1708–1712, 1986.

[112] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.