Title
Imagers as Sensors: Correlating Plant CO-2 Uptake with Digital Visible-Light Imagery

Authors
Hyman, Josh
Graham, Eric
Hansen, Mark
et al.

Publication Date
2007

DOI
10.1145/1286380.1286387

Peer reviewed

Imagers as sensors: Correlating plant CO_2 uptake with digital visible-light imagery *

Josh Hyman†
josh@cs.ucla.edu

Eric Graham†
egraham@cens.ucla.edu

Mark Hansen†
cocteau@stat.ucla.edu

Deborah Estrin†
destrin@cs.ucla.edu

ABSTRACT

There exist many natural phenomena where direct measurement is either impossible or extremely invasive. To obtain approximate measurements of these phenomena we can build prediction models based on other sensing modalities such as features extracted from data collected by an imager. These models are derived from controlled experiments performed under laboratory conditions, and can then be applied to the associated event in nature. In this paper we explore various different methods for generating such models and discuss their accuracy, robustness, and computational complexity. Given sufficiently computationally simple models, we can eventually push their computation down towards the sensor nodes themselves to reduce the amount of data required to both flow through the network and be stored in a database. The addition of these models turn in-situ imagers into powerful biological sensors, and image databases into useful records of biological activity.

1. INTRODUCTION

Visible-light imagers represent a very powerful and untapped sensing modality. They are typically avoided in traditional sensing applications because they produce large quantities of complex data that cannot be easily interpreted. However, there are many cases where sensing a particular phenomena is either not possible or too invasive given traditional sensors. For example, to measure the rate of CO_2 uptake (photosynthesis) of a plant requires a complex device that attaches to the plant containing an infrared gas analyzer and a mass flow meter. At the moment, these device are quite cumbersome and expensive. Thus, measuring this biological process for a single plant is complex, and for an entire landscape of plants, impossible.

Our goal in this paper is to explain a few techniques to model the rate of CO_2 uptake of drought-tolerant moss, *Tortula princeps*, given a time-series of images, Figure 1, taken of the plant. In general, there is a strong correlation between photosynthetic capacity and spectral properties of leaves [7]. Additionally, there have been few attempts to use basic, easily computed, image features to approximate photosynthesis and carbon gain [9] or plant phenological events [11].

We will generate models of CO_2 uptake using Support Vector Machines (SVMs) as well as Regression Trees. We then compare the different model's predictive accuracy to the error of the in-situ sensors themselves. We are not attempting to produce an optimal model for this task, instead we are trying to show that effective models exist and can be used to predict this phenomena. Further, we consider how the application of these different models, including feature extraction, will perform on sensor nodes with limited computation capacity. As there is no implementation of these models on sensors currently, we analyze the runtime and memory overhead incurred by applying such models. Lastly, we will address how to generalize these models to work in the field and simple techniques to collect images that are more robust to changing lighting conditions.

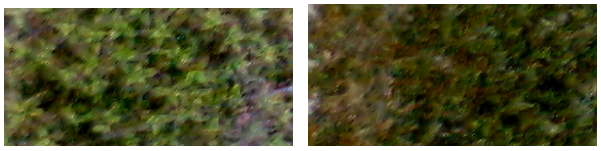
Given an image-based model of CO_2 uptake, we can begin to predict photosynthesis on a much larger scale than is possible to measure traditionally. For example, estimating the CO_2 uptake of an entire forest would now be feasible. This information can then be used to more accurately track the global release of CO_2 into the atmosphere, a nearly intractable problem currently.

Using an in-situ imager to predict difficult-to-measure natural phenomena is the fundamental idea supporting a large class of problems which have only recently been explored. Traditionally, imagers have been in the form of security cameras or other equally immovable devices attached to wired network and power. Data from such devices would be analyzed offline, searching for faces or other well defined objects for example.

Only recently have systems like Cyclops [10] become available to make deploying imagers in sensor networks easier and collecting their data simpler. The NestBox project [1] built a sensor network which attempts to understand the behavior of nesting birds using an imager. This phenomenon can not be currently measured by an existing sensor. Instead, studies of behavior involve manual observation, but increasingly rely on images; scientists watch hours of video to document

*This material based upon work supported by the National Science Foundation under Grant No. CCR-0120778
†Center for Embedded Networked Sensing
University of California, Los Angeles

Proceedings of the 4th International Workshop on Data Management for Sensor Networks (DMSN 07), Vienna, Austria, 2007.
VLDB '07, September 23-28, 2007, Vienna, Austria.



(a) Recently watered moss (b) Drying moss

Figure 1: 1(a) shows *Tortula princeps* with high CO_2 uptake, and 1(b) shows the same individual after it has dried and has a lower CO_2 uptake.

bird habits and how they build a nest, lay eggs, incubate and so on. This project has deployed inexpensive imagers inside bird boxes producing a potentially large amount of medium-quality imagery. They have developed simple models based on custom filters for these images to let scientists estimate the presence or absence of a bird in the nest at any moment, and in turn track rates of occupancy as a function of gestation cycle.

Another example is the use of imagers in the multiscale light mapping project [12] which attempts to observe the duration and pattern of light incident on the forest understory. Standard Photosynthetically Active Radiation (PAR; 400nm - 700nm) light measurement sensors can not be deployed widely enough to allow for complete characterization of the light field. Pixel measurements from in-situ imagers are correlated with incident light measurements taken by either fixed or robotic PAR sensors. These correlations allow them to infer a light field from the imager data, yielding a so-called multiscale analysis.

There have been similar attempts to model natural phenomena using remote sensing imagery. Sensor networks provide the unique challenge of processing the image data on in-situ sensors instead of off-line like conventional remote sensing applications. This requires models which are computationally inexpensive to apply and using minimal storage for the computation as well as the model itself. Further, the sensor network described in this paper takes advantage of inexpensive, readily deployable, visible light sensors that are capable of much higher temporal resolution as compared to a remote sensing platform like a satellite or plane.

Imagers provide a broad, but uncalibrated view of a scene; uncalibrated in the sense that we have to work to relate the observations to some quantity of scientific interest. In many cases we don't need complex computer vision algorithms, but instead can rely on fast filtering of the image data to produce a small number of features. These simple feature extractors, when used together with some form of statistical learning procedure, can transform in-situ imagers into sensors. Further, image databases are becoming nearly ubiquitous in the context of sensor networks as more sensing apps begin to collect images. In fact, an early requirement of systems like SensorBase [4] was the ability to store images effectively. These models have the ability to convert an image database into an untapped chronology of sensed events.

2. DISCERNING A CORRELATION

The rate of photosynthesis in plants is known to be correlated with the amount of chlorophyll and other photosynthesis related pigments in the plant's tissue. Further, the color

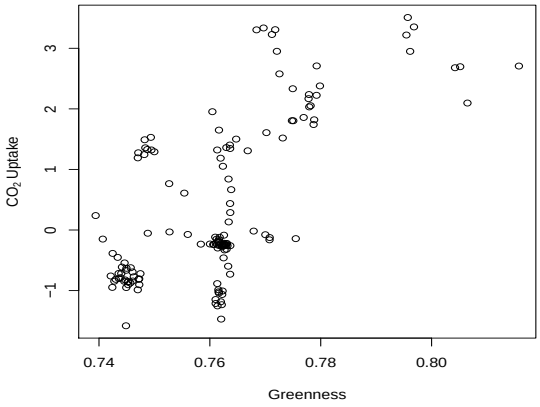


Figure 2: Scatter plot of correlation between the relative greenness image feature and CO_2 uptake

of the plant's leaves, its reflectance in visible wavelengths (400nm - 700nm), is largely influenced by such pigments [7] [8]. The moss we are studying, *Tortula princeps*, goes through natural cycles of drying in the field that are associated with changes in its photosynthetic capacity causing changes in its spectral reflectance. Thus, we concluded that we would be able to predict the rate of CO_2 uptake of this moss using image features alone.

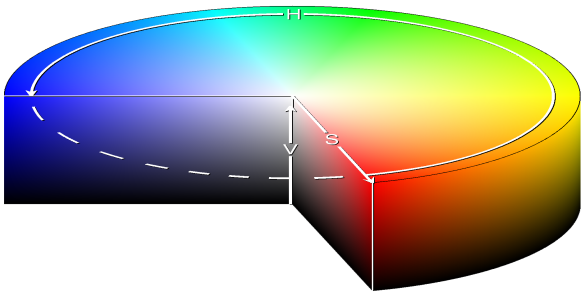
Before trying to predict the CO_2 uptake of the moss, we need to ensure that there is indeed a correlation between image features and the observed sensor values. First, we extract a color HSV (Hue, Saturation, Value) histogram from each image. Then, we compute its quadratic histogram distance [6] to a known green reference histogram, a metric we call relative greenness. This distance metric takes into account the L_2 distance between all pairs of colors in the two histograms, where as simple histogram distance only compares like buckets in a pair of histograms.

As we can see from Figure 2, there is a distinct correlation between the distribution of greenness over time and the sensor value over time. Optimally, this plot would be a straight 45 degree line. However, we see a vertical line near greenness value 0.76 with a cluster around sensor values of -1 and 0 representing many CO_2 values associated with a single greenness value. Still, there was sufficient correlation to prompt further research into a weighted model derived from large number of simple image features.

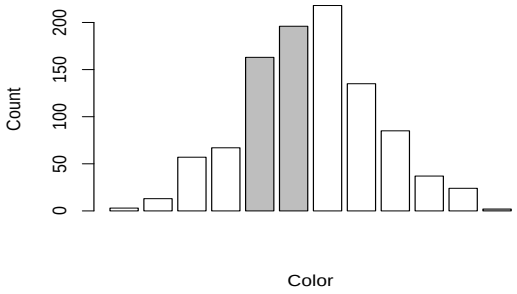
Motivation for a weighted model came as a result of achieving a stronger correlation when we ignored certain buckets of the HSV histogram; specifically buckets containing black, white, or other colors sufficiently far from green. This simple greenness measure is one of a large space of features which we can extract from these images. Fortunately, the use of a weighted model allows us to be a bit less deliberate in our feature selection; our model will heavily weight predictor features that are strongly correlated with our measurements, and lightly weight weakly correlated features.

3. PREDICTING SENSOR VALUES

To predict measured sensor values we need both a set of features and a method to weight these features into a model.



(a) HSV cylinder



(b) Windowed histogram

Figure 3: 3(a) The mapping of RGB colors onto the HSV cylinder [15]. 3(b) A window superimposed over a 1D histogram, the grey boxes represent the window of values being considered in this feature.

To leverage the known correlation between photosynthesis and greenness, we choose various features based on HSV histograms of the images. The features we are extracting are very simple and computationally inexpensive. We believe these simple features to be sufficient because the value we are attempting to predict is highly sensitive to the coloration of the plant. So, we intentionally do not incorporate complex vision related techniques, or even rudimentary measures of texture, because they possess little more expressive power with respect to the model we are building. Additionally, they these features are more expensive to compute, taxing the limited resources available to in-situ sensors.

3.1 Extracting Image Features

Before extracting features from the images, we first converted the images from the RGB (Red, Green, Blue) color space into the HSV (Hue, Saturation, Value) color space. This non-linear transformation, shown in Figure 3(a), nicely separates the notion of color from its intensity. Thus, the wavelength of visible light reflected from the moss is most strongly represented in the Hue direction rather than spread out over all three RGB directions.

Given an image in the HSV color space, we compute a single normalized 3D histogram containing frequencies of all (h, s, v) triples and three normalized 1D histograms for each of the Hue, Saturation, and Value directions individually. Note that a 3D HSV histogram requires approximately 14MB of memory to store - 360 Hue buckets by 100 Saturation buckets by 100 Value buckets by 4 bytes per bucket. To reduce the size of this histogram, we down-sampled each of these dimensions to 100, 10, and 10 respectively, producing a 40KB histogram [13]. Additionally, the "wider" histogram buckets produced by down-sampling reduces the histogram's variance between images potentially increasing the accuracy of the model.

We maintain higher accuracy in the Hue direction because most of the significant color information is contained in that dimension. Given these histograms, we create a set of equal-sized windows spanning a contiguous set of adjacent histogram buckets. The "value" of a given window is the sum of the values in each of the buckets it contains as illustrated in Figure 3(b). This way we can create a large set of image features by varying the window size.

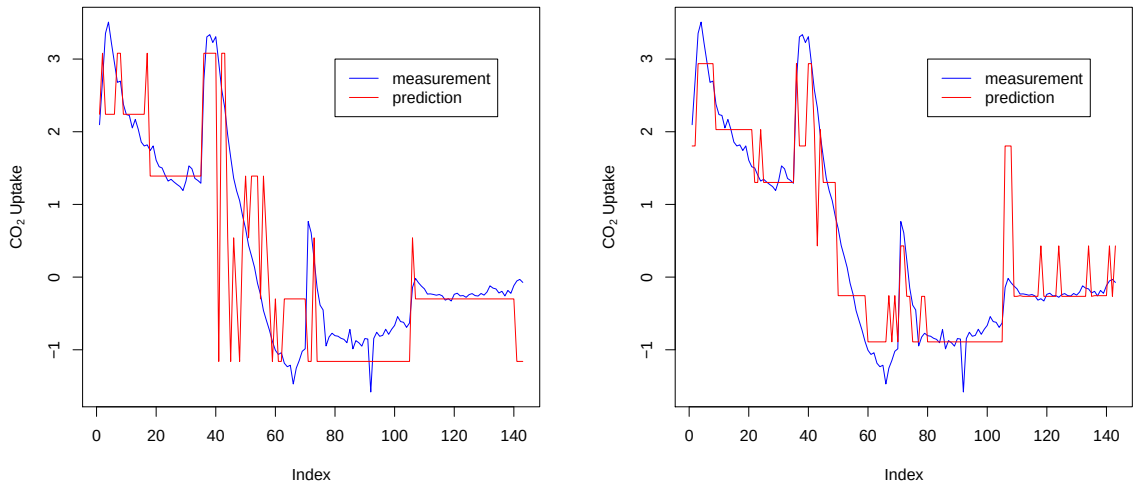
These features only require the computation of an image histogram, an operation linear in the number of pixels. However, computing the value of each feature from the histogram is an operation which is linear in the number of histogram buckets. So, the complexity of computing all features values becomes $O(nb)$ where n is the number of features and b is the number of histogram buckets. To reduce the runtime, we computed a Cumulative Distribution Function (CDF) from the histogram. Given a CDF, we compute the window's value by subtracting the value at the lower end-point of the window from the upper end-point (a constant-time operation) without having to iterate through the entire window's histogram buckets. This makes the complexity of computing all features $O(n)$, allowing us to consider far more features cheaply. This is similar to the integral image technique used to reduce complexity of computing the value of Haar-features from images [14].

Though such an optimization is not too beneficial for computation on a typical workstation, saving such computation on a sensor node is important. Further, once the CDF is computed, there is no longer any need for the image itself. The memory cost for storing an CDF of the image is only $O(b)$ instead of linear in the resolution of the image; this represents a significant memory savings for a constrained sensor. This reclaimed space is more than sufficient for the added memory requirements of applying the models we consider.

3.2 Training and testing methodology

To assess the accuracy of our models, we split our data into a training and testing set. Since we did not have too much data for any given run of the laboratory experiment, our testing set typically had approximately 50 data points. After training, we applied our models to the test set and computed the root mean squared error (RMS Error); an acceptable RMS Error would be approximately 0.5 photosynthetic units. An error of 0.1 photosynthetic units is the limit of current sensing technology because of changes in temperature, moisture, and other environmental effects. Being able to approximate this value with an error of 0.5 photosynthetic units using only a camera is a significant accomplishment. Further, with such accuracy, it will likely be more accurate than other sensing modalities which will have to be estimated using nearby sensors as references.

Instead of using a fixed training and test set to build our model, we could have used boosting. In that setup, we would have trained our model on a randomly selected subset of the data and then iteratively retrained by adding the hard to



(a) Prediction based on 6-class SVM classifier

(b) Prediction based on a regression tree

Figure 4: Plots of CO_2 uptake (measured) and predicted values over time

predict data points to our training set. The iterative growing of test set makes this method most effective when the entire dataset can not be considered simultaneously. However, since each run had limited data, considering the whole dataset was feasible.

Alternatively, we could have used leave-one-out testing where we train on the entire dataset minus one point, and attempt to predict only that single point. This is the other extreme, where we do not have enough data to extract a proper test set. Fortunately, we had sufficient data to extract a test set without compromising our model's accuracy.

3.3 Prediction using classifiers

Predicting the measured values using a classifier required training N SVM based classifiers to each recognize a contiguous range of CO_2 values. To test initially, we chose to train a single classifier on half the range of the measured values. Initially using a linear kernel for the SVM, i.e. no data transformation, and a minimal feature set we were able to achieve an 88% accuracy, 82% percision, and 81% recall on the test set. Here, accuracy is the percentage of correct classification, precision is the percentage of true positive examples with respect to all positive examples, and recall is the percentage of true positive examples which were correctly classified. With the addition of many more features and a cubic kernel, we improved accuracy to 92%, precision to 91%, and recall to 85% on the test set.

These results were sufficient to progress to splitting the measured range into 6 regions and training the associated 6 classifiers. To expand our model to use multiple classifiers, we leveraged the fact that an SVM classifies a point by simply computing the distance between the example and the hyper-surface that separates the two classes being discerned. So, to combine more than one classifier into a discrete prediction, we define the correct classification to be the classifier whose normalized prediction distance is furthest from the hyper-surface in the positive direction. We then predict the value for this example to be the median value of the range the classifier was trained to recognize.

Using this technique, we made this model more robust to varying accuracy, precision, and recall in the set of classifiers being used. This classifier variability stems from the inconsistent number of positive examples (data points in that particular region). Typical accuracy, precision, and recall values of the individual classifiers ranged from approximately 50% to 90%. However, the resulting prediction has an RMS error of 0.74 on the test set and was quite accurate as can be seen in Figure 4(a).

Increasing the number of classifiers to 15 and adding additional features was able to reduce the RMS test error to 0.70, a smaller improvement than anticipated. The expectation was that as the number of classifiers grew, the prediction would approach the measured values, however this was not the case. Analysis of the error showed that it was normally distributed. In fact, the 85th percentile squared error was 0.61, and nearly all squared error value larger than 0.61 occurred for measurements near 1 or greater than 2. We will address why such values cause such large errors in Section 3.6.

Moreover, since we had a limited amount of data, training an ever growing number of classifiers produced diminishing returns as the number of positive examples per classifier approached zero and the computational expense grew. Both the weak prediction performance and poor scaling properties caused us to search for a simpler model which could predict this phenomena.

3.4 Prediction using regression trees

As an alternative to a set of N classifiers, we tried building a regression tree using recursive partitioning. Instead of performing a normal regression, we use a decision tree to partition the data. Then, we perform a regression at leaf nodes in the tree where examples should be more similar in nature [3]. Here, a feature and threshold pair that best

splits the data at a given node is chosen recursively until all of the leaf nodes reach a specified purity or similarity. Thus, as we traverse the tree from the root, nodes contain increasingly similar measurements which obey the feature constraints defined by the tree. To produce a prediction for the set of values at given leaf node, we average all the values at that node. The result of a model trained in this fashion, shown in Figure 4(b), is quite good.

Without significant tuning, we found the RMS prediction error of this model to be 0.49; already much better than the error for an SVM based model. Again, analysis of the error showed it to be relatively normal with the most error occurring on measured values near 1 or greater than 2. Decreasing the splitting threshold of a node and increasing the required purity of terminal nodes only reduced the error slightly at the expense of a larger tree.

In all cases, the regression tree outperformed the SVM classifiers with respect to RMS error. Further, a regression tree based model is better suited for an embedded device because of the brevity of its model definition (only the tree itself is needed) and nominal computational complexity required to apply the model to an example. In contrast, the SVM based models need many large model definition files (one for each classifier), and require performing significantly more math to classify an example. For these reasons, a regression tree based model is superior for this application.

3.5 Other prediction methods

There are various other methods we could have used to build a model of this phenomena. Similar to our use of SVMs would be the application of Support Vector Regression (SVR) [5]. Analogous to the SVM computation, SVRs iteratively compute an increasingly better hyper-surface regression of the data, attempting to minimize a specified loss function at each step. As an extension to our use of regression trees, we could have used random forests [2] to better approximate our measurements. This technique combines multiple regression trees generated from a random subset of the feature space into a single approximation of the system. We will address the use of both of these modeling strategies in future work.

3.6 Hard to predict measurements

The fact that our error distribution is heavily skewed, indicates that our models are likely missing a predictor variable. Interestingly, the largest error occurs around zero valued measurements, an area which was not well distinguished by color as seen in Figure 2. As an attempt to find the missing feature, we produced a large number of features by varying the window size and using larger histograms which were not down-sampled to save space. We concluded that we can't accurately predict values in this range from color features alone.

This result is supported by the biology of this moss. As the plant dries and becomes more brown, its CO_2 uptake decreases and eventually becomes slightly negative. However, there is a period when the moss is almost completely dry when its CO_2 uptake rate becomes even more negative but its color changes only slightly. Also, periods of darkness during the experiment facilitated rehydration of the moss, the 3 pronounced spikes in Figure 3.1. The resulting inconsistent drying rate caused similar colors to represent dissimilar rates of change in CO_2 uptake.

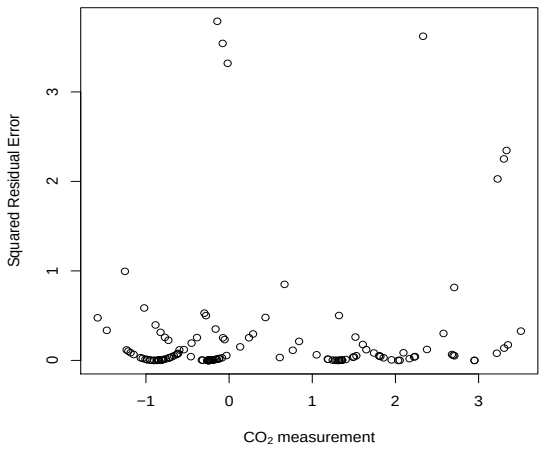


Figure 5: Prediction error of a regression tree based model

Both of these facts help to explain why our error is so pronounced in the region of near zero photosynthesis; the moss can be in two distinct states which are hard to discern using only color. In future work we plan to extract other simple image features, like texture, to see if it can identify differences between these two photosynthetic states to produce a more accurate model.

Another area of poor prediction, as shown by Figure 5, is the region of CO_2 measurements above a photosynthetic rate of 2. The error in this region can be attributed to lack of training examples as opposed to a lack of discerning features. Also, a number parabolic shapes can be seen in the error values. This is due to the fact that our model has a fixed set of possible prediction values. Thus, squared prediction error increases quadratically centered at each possible prediction value.

4. MODEL ROBUSTNESS

Images taken in the laboratory have relatively consistent lighting, whereas field imagers experience scenes that change more radically. To make these models more robust to the lighting conditions, we employed various simple pre-processing techniques to standardize images before we extracted features. First, we tried to use histogram equalization to increase contrast in the images. Unfortunately, this procedure alters each image differently depending on their individual lighting conditions. We expected such non-uniform modifications to adversely effect our model, yet, we found that such a transformation did not change the model in any significant way.

More useful would be to uniformly correct the image's white balance to eliminate changes in lighting. One simple solution would be to include a white reference in the image during data collection to facilitate white balance correction. Another solution would be to get the white balance, or color temperature, from the imager so that we can reverse the effects of auto white balancing to make the resulting images more uniform.

5. FUTURE WORK

The work presented in this paper explores only the first steps required to build robust models for resource-constrained sensors. As mentioned in Section 3.5, there are numerous potential models left unexplored that may be better suited for in-situ sensors. Additionally, to produce more accurate models we can consider searching for a set of optimal features representing the correlation between image histograms and the CO_2 uptake.

Though we considered the cpu and memory impacts of these models, we neglected their affect on power usage. By implementing these models on actual sensors, we will gain insight into how to optimize the models to increase the lifespan of the sensors without sacrificing predictive accuracy.

6. CONCLUSION

In this paper we have outlined how to predict the CO_2 uptake of a moss plant given an image of that plant. We have described different models building strategies, and concluded that regression tree models based on color features are quite effective for this data. Specifically, its prediction performance, brevity of model description, and efficiency of application to new examples make it ideally suited for use on a sensor node. Though regression trees may not be the correct model for every application, the process which we followed and criteria we used to choose between models in this instance are generally applicable.

Combined with models of natural phenomena, imagers become extremely powerful sensors which can measure a wide range of different sensing modalities. Performing image processing closer to the sensor reduces the network load needed to transmit the images and truly transforms the imager into a first class sensor. Moreover, such models unlock sensor data from the confines of the images contained in image databases. Yet, the true power of this type of model lies in estimating measurements of phenomena which cannot be measured by any other means.

7. REFERENCES

[1] S. Ahmadian, T. Ko, S. Coe, M. Rahimi, S. Soatto, M. Hamilton, and D. Estrin. A vision system to infer avian nesting behavior. Technical report, University of California, Los Angeles, 2007.

[2] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

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

[4] K. Chang, N. Yau, M. Hansen, and D. Estrin. Sensorbase. org–a centralized repository to slog sensor network data. *DCOSS/EAWMS Proceedings*, 2006.

[5] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9(9):155–161, 1997.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, et al. Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23–32, 1995.

[7] 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.

[8] E. Graham. *Perspectives in Biophysical Plant Ecophysiology: A Tribute to Park S. Nobel*, chapter The Use of Digital Cameras in Plant Ecology and Ecophysiology. Lulu Press, San Diego, California, U.S.A., 2007.

[9] E. Graham, M. Hamilton, B. Mishler, P. Rundel, and M. Hanson. Use of a Networked Digital Camera to Estimate Net CO2 Uptake of a Desiccation-Tolerant Moss. *International Journal of Plant Sciences*, 167:751–758, 2006.

[10] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: in situ image sensing and interpretation in wireless sensor networks. *International conference on Embedded networked sensor systems*, pages 192–204, 2005.

[11] A. Richardson, J. Jenkins, B. Braswell, D. Hollinger, S. Ollinger, and M. Smith. Use of digital webcam images to track spring green-up in a deciduous broadleaf forest. *Ecosystem Ecology*, 2007.

[12] A. Singh, D. Budzik, W. Chen, M. Batalin, M. Stealey, H. Borgstrom, and W. Kaiser. Multiscale Sensing: A new paradigm for actuated sensing of high frequency dynamic phenomena. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 328–335, 2006.

[13] J. Smith and S. Chang. VisualSEEk: a fully automated content-based image query system. *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, 1997.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of Computer Vision and Pattern Recognition*, 1:511–518, 2001.

[15] Wikipedia. HSV Color Space. *http://en.wikipedia.org/wiki/HSV_color_space*, 2006.