WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Computer Science and Engineering

Dissertation Examination Committee:
Robert Pless, Chair
Gruia-Catalin Roman
Yixin Chen
Victor Gruev
Joseph A. O'Sullivan
Mladen Victor Wickerhauser

DISCOVERING, LOCALIZING, CALIBRATING, AND USING THOUSANDS OF OUTDOOR WEBCAMS

by

Nathan Bradley Jacobs

A dissertation presented to the
Graduate School of Arts and Sciences
of Washington University in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2010
Saint Louis, Missouri

ABSTRACT OF THE DISSERTATION

Discovering, Localizing, Calibrating, and Using Thousands of Outdoor Webcams

by

Nathan Bradley Jacobs

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2010

Research Advisor: Professor Robert Pless

The web has an enormous collection of live cameras that capture images of roads, beaches, cities, mountains, buildings, and forests. With an appropriate foundation, this massively distributed, scalable, and already existing network of cameras could be used as a new global sensor. The sheer number of cameras and their broad spatial distribution prompts new questions in computer vision: How can you automatically geo-locate each camera? How can you learn the 3D structure of the scene? How can you correct the color measurements from the camera? What environmental properties can you extract?

We address these questions using models of the geometry and statistics of natural outdoor scenes, and with an understanding of how image appearance varies due to transient objects, the weather, the time of day, and the season. We show algorithms to calibrate cameras and annotate scenes that formalize this understanding in classical linear and nonlinear optimization frameworks. Our work uses images captured at long temporal scales, ranging from minutes to years; often these images are from a database

of images we created by capturing images from 1000 webcams every half hour. By exploring natural cues capable of working over such long time scales on a broad range of scenes, we deepen our understanding of the interplay between geographic location, time, and the causes of image appearance change.

Acknowledgments

My friends, colleagues, and family have supported me during my graduate studies; and some have supported me for years before that. I would like to take the opportunity to thank them.

To my advisor, Dr. Robert Pless, I owe a debt of gratitude. His wisdom, willingness to listen, and unshakable enthusiasm have been inspirational. I hope to return the favor someday to future generations of scientists. I would also like to acknowledge the other members of my committee, Dr. Catalin Roman, Dr. Yixin Chen, Dr. Viktor Gruev, Dr. Joseph A. O'Sullivan, and Dr. Victor Wickerhauser, who have offered their time and expertise to assist in the evaluation of my research.

For the past 5 years, my colleagues in the Media & Machines lab have provided me with an intellectually stimulating work environment. I would first like to thank Sasakthi Abeysinghe, my long-time officemate, for always cheering me on. Thanks to Michael Dixon, my long-time lunchmate, for being a thoughtful critic. Thanks to Dr. Richard Souvenir for being a good role model and an insightful researcher. I would also like to thank a few current and former M & M lab graduate students, including Dr. Rob Glaubius, Manfred Georg, Ross Sowell, Fritz Heckel, Tom Erez, Stephen Schuh, Doug Few, Lu Liu, Nuzhet Atay, and Austin Abrams. These individuals have been both friends and colleagues, always willing to advise, critique, or just listen. I would also like to thank the members of the webcam team that contributed significantly to this work: Brian Bies, Walker Burgin, Nick Fridrich, Kylia Miskell, Nathaniel Roman, David Ross, Scott Satkin, and Richard Speyer. Finally, I thank the M & M lab faculty, especially Dr. Cindy Grimm, Dr. Tao Ju, Dr. Caitlin Kelleher, and Dr. Bill Smart, for their countless nuggets of wisdom.

I probably don't fully appreciate how much work it takes to keep a department running, but I have nothing but praise for the assistance and support I have been given by the staff of the computer science department. For this, my special thanks go to Kelli Eckman, Myrna Harbison, Madeline Hawkins, and Sharon Matlock. They managed to keep me from stepping into countless traps and mud holes through the years.

Finally, I would like to thank my family and, with great pride, dedicate this dissertation to them. To my parents, Brad and Linda, I owe thanks for giving me the confidence to take a risk. To my sister, Fresa, I owe thanks for her willingness to chat at odd hours while I waited for convergence. To my children, Zane and Lia, I give thanks every day for the smile on my face. And, to my amazing wife, Julie, whose love and support made this possible, I simply say thank you.

Nathan Bradley Jacobs

Washington University in Saint Louis
May 2010

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Thousands of cameras exist that allow the current image to be retrieved via the Internet. Web-attached cameras, also known as webcams, are placed for a variety of reasons. For example, they might be used to provide evidence of traffic congestion to commuters or to share the natural beauty of the scene with others. Images from these cameras are a vast untapped resource of information about the world and the way it changes over time. Figure 1.1 shows example images from three of the thousands of webcams that can be found online.

Outdoor cameras and camera networks are used as important sensors for a wide variety of problems, including measuring plant growth [64], surveying animal populations [41], monitoring coastal erosion [68], and security [79]. Unfortunately, camera networks are expensive to design, deploy, and maintain, effectively limiting the number of projects able to install a dedicated camera network. A report by the Heinz Center [17] lists several information gaps that could be partially filled using land-based cameras: assessing the amount of particulate matter in the air, assessing plant growth, and determining changes in exercises habits of people (e.g., are more people running this year?).

The webcam network is inherently more scalable than a camera network designed for a specific purpose because the monetary cost of camera deployment and maintenance is paid for by others. Our cost in adding additional cameras to the network is only in discovering and archiving images. The cost for a novel study is in selecting a set of cameras, selecting a set of images, developing algorithms specific to a task, and processing the selected images.

(a) (b) (c)

Figure 1.1: Outdoor webcams are placed by many individuals for many reasons. We propose to use these cameras for measuring the world. (a) This camera viewing the Pebble Beach Golf Club could be used to determine the surf conditions and the tidal phase. (b) This camera could provide information about the current weather conditions in St. Louis, MO. (c) We also show how natural scene variations can help determine which of these cameras is located in New York City and which is in Las Vegas.

Environmental properties directly affect the images we collect from outdoor webcams: whether it is cloudy or sunny is visible by the presence of shadows; wind speed and direction is visible in smoke, flags, or close-up views of trees; particulate density is reflected in haziness and the color spectrum during sunset. We explore techniques to automatically extract such environmental properties from long sequences of webcam images. This allows the webcams *already* installed across the earth to act as generic sensors to improve our understanding of local weather patterns.

What can we measure using images from webcams? We want to provide a theoretical and technical foundation for answering this question. We consider measuring temporal signals and non-temporal scene properties. Examples of temporal signals that could be measured using webcams include wind velocity, relative humidity, tidal phase, snow cover, and seasonal plant growth. Examples of non-temporal properties include geometric scene structure (e.g., buildings and trees), functional scene structure (e.g., walking paths), and the geographic location of the camera.

The primary challenge in using webcams to make measurements of specific phenomena is that, unlike specialized sensors or camera networks designed for a specific purpose, webcams are not carefully selected, calibrated, or placed to make measurement easy.

How can we overcome this challenge to measure temporal signals of interest in a particular application? Our approach is to control for the variations that are not of interests (e.g., whether it is cloudy or not) by explicitly modeling the complex interactions of the underlying factors that determine image appearance. In the next section, we describe the conceptual model we use to model this complexity.

1.1 Generative Model of Outdoor Scenes

Understanding the relationships between these causes of change and the appearance of an image is essential to the use of webcams as sensors. Some relationships are easy to observe; for example, the relationship between the time of day and the image appearance. Others are more challenging to observe; for example, the relationship between the number of people in the scene and the weather conditions (e.g., fewer people are on the beach when it is cold, therefore temperature affects image appearance).

We describe the dependence of images on latent factors in the world as a sparse network that we call the geo-temporal image formation model (GIFM). The model describes the high-level relationships between natural scene variations and the resulting images captured by the camera. Figure 1.2 provides a pictorial representation of a simple version of this model in which each circle represents a distinct factor in the image formation process, and edges represent relationships between these factors. For example, the weather conditions depend on the geolocation of the camera (i.e., $P(\text{weather}|\text{geolocation}) \neq P(\text{weather})$). In addition to specifying dependencies, the model implicitly specifies factors that we deem independent. For example, the model currently shows that the imaging noise of the camera is independent of time. While a complete version of a geo-temporal image formation model would be significantly more complex, we use this model primarily as a conceptual foundation.

We focus on isolating small portions of this model by constructing datasets that explicitly condition on or marginalize over the other variables. For example, in work on depth estimation, we control for long-term variations (e.g., seasonal changes) by only considering images captured over a few hours, and we marginalize over variables we are unable to easily control for such as the *transient objects*.

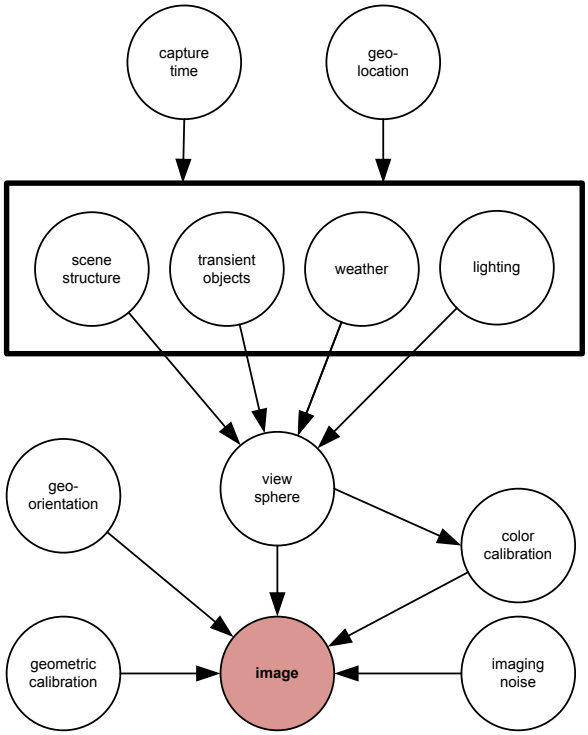


Figure 1.2: Our conceptual model of the geo-temporal context that underlies how images are formed from outdoor cameras. The edges represent a conditional relationship between variables. For example, the edge from the time variable to the rectangle means that all variables inside the rectangle are directly conditioned on the time (e.g., the lighting from the sun depends on the time of day).

1.2 Contributions

The conceptual and technical contributions of this dissertation are:

- We create and describe a large dataset of images captured from webcams. This work extends upon earlier work [49] to catalog a single outdoor scene for a year. To our knowledge, our dataset is the largest such ever assembled. Constructing this dataset and making it publicly available represents a considerable effort and a substantial contribution to the field of computer vision.
- We describe a user interface for browsing a large number of images from a single webcam. We show how the known time and geographic location can be used to create a free source of training data and a method for filtering the images.
- We use our webcam dataset to characterize properties of the global network of outdoor webcams. We consider both low-level statistics, such as the average image file size, and higher-level concepts, such as whether or not the scene includes a tree.
- We advocate for the use of webcams for science. As part of this we define a set of calibration problems and present several example applications. One application is motivated by the immediate needs of a plant physiologist in validating satellite estimates of the spring-onset of leaf growth on deciduous trees. We show that the high spatial resolution captured by webcams is valuable in distinguishing between multiple tree species.
- We formulate a geo-temporal image formation model and show how it can be used to reason about the underlying causes of change in outdoor scenes. We show that simple second-order analysis can highlight changes at scales ranging from hours to seasons. We show how jointly considering the variations across a large number of cameras leads to a consistent decomposition, which we use to label pixels with colors that relate to surface orientation. We also show how the image formation model can be used, coupled with our understanding of the underlying causes of image appearance variations, to geo-calibrate the webcams.
- We describe our method for automated camera localization, which uses coarse scale natural changes (the sun and the weather) to determine the location of

an outdoor camera. We show robust methods of extracting image features related to these underlying causes and further show how to use these features to estimate the location of a camera. We evaluate these methods on a large number of webcams from our dataset.

- Further refining the camera calibration, we create a new method of estimating the geo-orientation of an outdoor camera. We use the sky as a calibration pattern, thereby eliminating the need for physical access to the camera.

- We show how the shadows cast by clouds can be used to estimate the geometric structure of the scene. We define two brand-new cues, one spatial and one temporal, that are based on cloud shadows. For the spatial cue, we show how to adapt the non-metric multidimensional scaling algorithm to the task of depth map estimation. For the temporal cue, we define the set of linear constraints it creates, along with a one-dimensional ambiguity, and describe our method for combining these constraints with the spatial cue to resolve the ambiguity. We qualitatively and quantitatively evaluate these methods on real videos of outdoor scenes.

Chapter 2

Outdoor Webcams and Webcam Image Properties

We make the argument that the outdoor webcams distributed across the Internet are an important, underutilized imaging resource. To support this argument, this chapter provides answers to the following fundamental questions about this resource:

- How do you find webcam URLs?
- What is the geographic location of these webcams?
- Is it possible to collect images from these cameras on a large scale?
- What are the properties of the cameras, the scenes they view, and the images they capture?

2.1 The Global Network of Outdoor Webcams

The web contains an enormous collection of cameras that provide live images of a wide variety of scenes. This section explores properties of this vast, already existing camera network that we call the global network of outdoor webcams (GNOW). We begin with the basic problem of identifying a list of URLs that point to images captured by an outdoor webcam.

2.1.1 Discovering Webcam URLs

The first challenge in using the GNOW is finding URLs that point to webcams. Our strategy for finding URLs involves merging lists from webcam aggregators and explicit searches for cameras in places such as national parks, state departments of transportation, and similar queries targeting each country around the world. Many of the cameras we have discovered come from web-sites that contain lists of large numbers of webcams that either consist of cameras that they explicitly own (e.g., the Weatherbug camera network [82]), or cameras that individuals register to be part of a collective (e.g., the Weather Underground webcam registry [81]). Additionally, we use a collection of Google searches for unusual terms, such as "inurl:axis-cgi/jpg", that are primarily used in default webpages generated by webcams.

Using this manual method for roughly 300 person-hours, a group of undergraduate and graduate students (and one hardy associate professor) found 16 112 webcam URLs that produce different live images. To our knowledge, this is the largest list of webcam URLs ever assembled. We believe that while it is only small fraction of the substantially larger GNOW, it is sufficiently large to be representative of the whole. The remainder of this section describes our work to understand the spatial distribution of these cameras, the properties of the images they capture and the scenes they view.

2.1.2 The Spatial Distribution of Outdoor Webcams

Obtaining accurate location estimates for the cameras is critical for our goal of measuring environmental properties. Therefore, we use a combination of automatic and manual techniques (see Section 3.2 for details) to determine the geographic locations of the physical cameras represented by the URLs we have discovered.

Figure 2.1 shows the geographic locations we estimate for these cameras. The distribution shown in this plot demonstrates, as one would expect, that there are many more cameras in heavily industrialized regions, especially in Europe and North America, than in more rural or remote areas. The results also show regions that have far

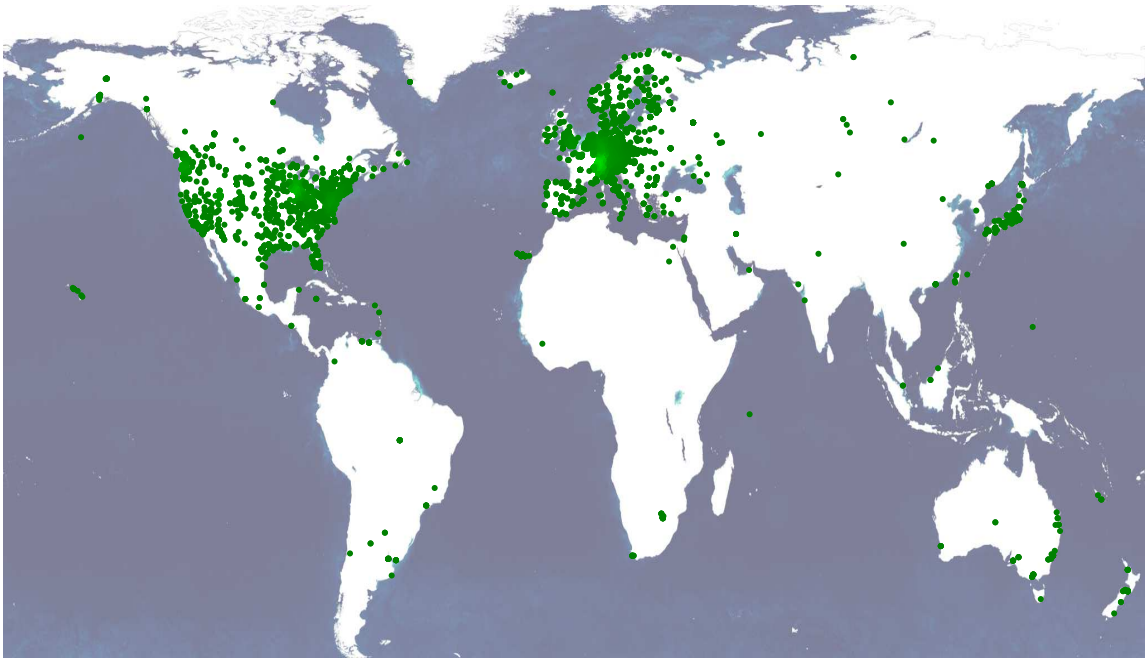


Figure 2.1: The location of tens of thousands of outdoor webcams are shown above as green dots, with lighter green dots corresponding to geographic regions with a higher density of cameras. While it is likely that our exclusive use of the English language to search for webcams URLs biases the result somewhat, we think this distribution mirrors the true distribution of webcams.

fewer webcams than we expected, especially France and Asia. We suspect this bias is partly because Internet searches were conducted in English.

This result demonstrates that the GNOW covers a substantial portion of the globe with cameras. In the remainder of this section, we explore the properties of the images captured by these cameras and the scenes they view.

2.1.3 Webcam Image-File and Scene Properties

In this section, we consider both low-level properties of the images, such as file size, and high-level properties of the scene, such as whether or not a mountain is visible. This analysis highlights the broad range of cameras and scenes in the GNOW.

We begin by describing low-level properties of individual webcam image files and the sequence of images generated by a webcam. Figure 2.2 shows the distribution of file sizes and image dimensions that reflects the fact that most webcams provide small,

highly compressed images. In order to understand the distribution of temporal refresh rates, we estimate the refresh rate for a set of 500 randomly selected cameras using a standard method [7]. The distribution in Figure 2.2(c) reflects the fact that many webcams are configured to capture new images every 5 minutes.

To begin to characterize statistics of the scenes viewed by this set of cameras, we manually estimated the minimum and maximum distance of objects in the scene from the camera for a randomly chosen subset of 300 cameras. We grouped our estimates into the following intervals: 1–10 meters, 10–100 meters, 100–1000 meters, and greater than 1000 meters. Most cameras capture images of objects both near and far; this is highlighted in Figure 2.2(d), where the cumulative distribution functions for min- and max-depth show that 80% of the scenes have an object within 10 meters of the camera, and only 20% of the scenes do not contain an object more than 100 meters away.

Additionally, we manually labeled the scenes imaged by 1300 randomly sampled cameras. We tagged each scene based on several characteristics: if it was outdoors and/or it contained a road, trees, buildings, or substantial sky or water (where we define 'substantial' to mean 'at least a fifth of the picture'). Figure 2.3 shows specific examples of this labeling and gives some global statistics. This type of manual labeling is especially helpful for selecting a subset of cameras for a particular measurement task (see Section 2.2.1).

2.2 AMOS: Archive of Many Outdoor Scenes

Thus far, we have described properties of the GNOW; we now describe the Archive of Many Outdoor Scenes (AMOS) dataset, which we have constructed by archiving images from a subset of GNOW cameras. We began collecting imagery in March 2006 and currently have more than 50 million images from more than 1000 cameras.

The cameras in the dataset were selected by a group of graduate and undergraduate students using a standard web search engine. Images from each camera are captured several times per hour using a custom web archiver that ignores duplicate images and records the capture time. The images from all cameras are 24-bit JPEG files that

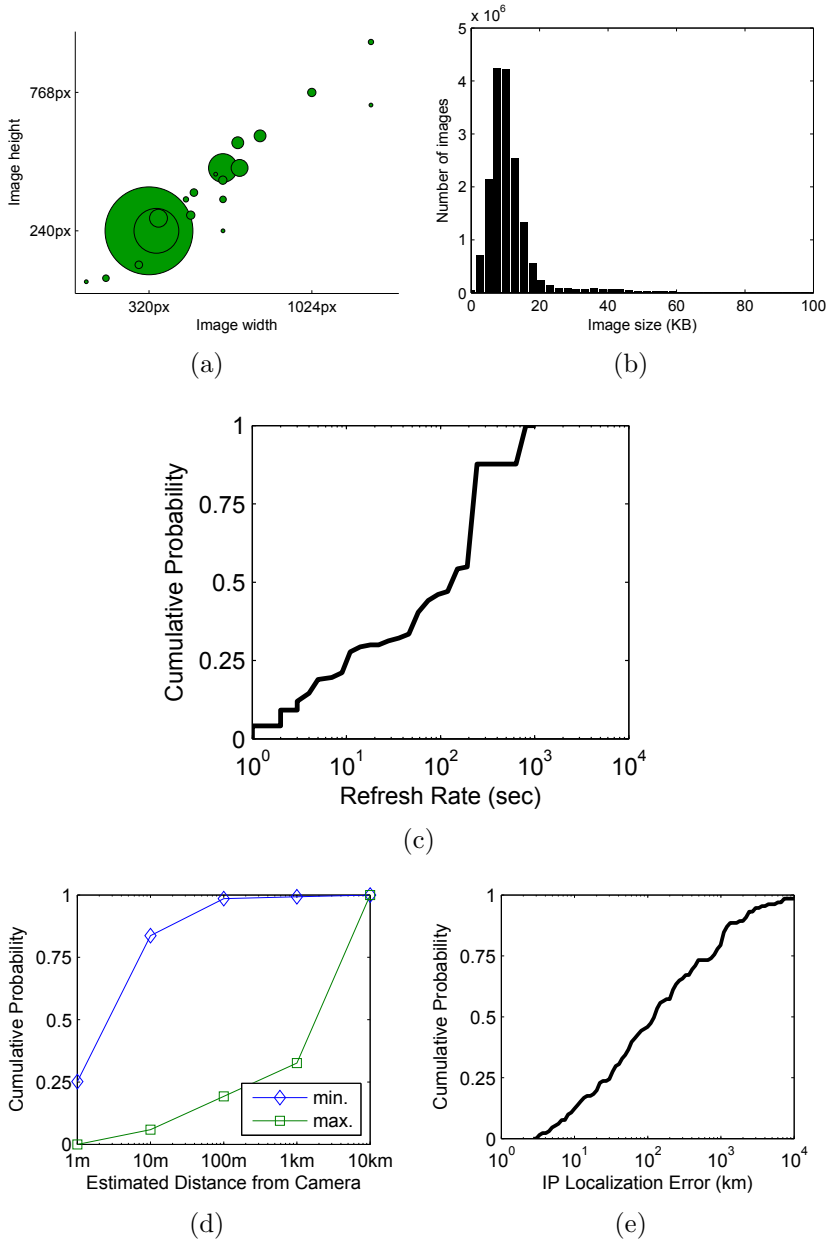
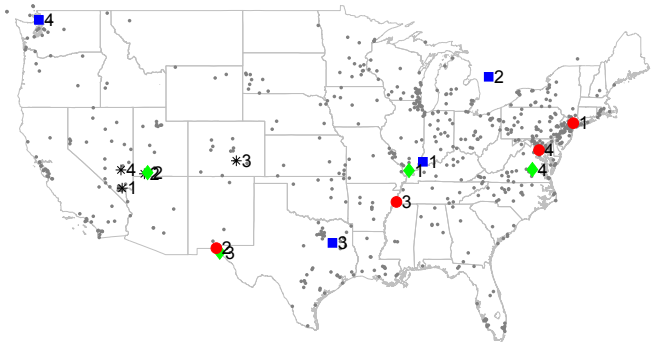


Figure 2.2: (a) The distribution of image sizes measured in pixels. Each circle is centered at an image size with area proportional to the number of cameras. (b) The distribution of image sizes in kilobytes. (c) The cumulative distribution of refresh rates of webcams. Note the large number of cameras that refresh every 5 minutes. (d) The cumulative density function of the minimum depth of an object in the scene (blue line, near the top) and the maximum depth of an object in the scene (green line, near the bottom). Most cameras see objects both in the near field (10m or less) and far field (at least 1 km). (e) The cumulative distribution of localization errors of IP-addressed based localization. The median error is 111km.



(a)

Tags	Percentage
Outside	95
Road	50
Trees	67
Buildings	60
Sky	68
Water	22

(b)

(c) contains a mountain (black star)

(d) contains natural water and a building (blue square)

(e) contains a building but no trees (red circle)

(f) contains trees but no building (green diamond)

Figure 2.3: (a) A map of the subset of webcams manually inspected for scene labeling. (b) Global statistics of the labels. For example, of the webcams we manually labeled, 67% contained a tree. (c–f) Four example images from cameras chosen based on the presence or absence of a set of manual metadata tags. The locations of each of these images is presented on the map. Notice, for example, that all the scenes that view mountains are located in Utah, Nevada, or Colorado.

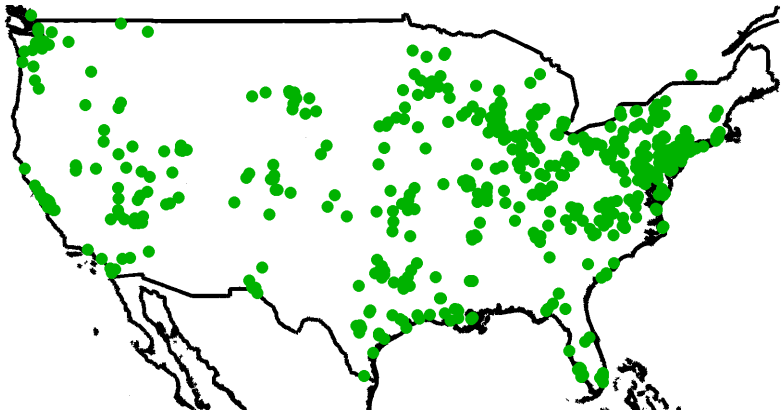


Figure 2.4: A scatter plot of the locations of cameras in the AMOS dataset.

vary in size from 316×240 to 2048×1536, with the majority measuring 320×240. In addition to a large amount of image data, each camera is assigned latitude and longitude coordinates. The majority of cameras in this archive are located in the continental United States, Figure 2.4 shows the locations of these cameras.

This dataset is unique in that it contains significantly more scenes and a longer duration than in previous datasets [49] captured from static outdoor cameras. The remainder of this section describes our work to improve the usefulness of the data by adding novel visualizations and manual annotations.

Related Work: Outdoor Image Datasets

Given the prevalence of learning-based methods and the need for empirical validation, the creation of datasets is an important part of computer vision. Most datasets focus on providing data for specific tasks, such as scene categorization [40] or face recognition [58]. Narasimhan et al. [49] created a dataset consisting of images captured every hour for one year from a single urban scene. The camera was carefully calibrated, and the images come with high-quality ground truth. The AMOS dataset (2) differs in that it contains images from many cameras over a longer time period.

Significant efforts to collect and annotate large numbers of webcams have been undertaken. Notably, Google Maps now features a "webcam layer" that organizes live webcam feeds from approximately 13 000 webcams. Other large collections of webcam URLs [82, 81, 54] have been created, and many of these cameras are geo-located.

However, these collections are not as spatially dense as the cameras we have discovered (see Section 2.1.1), are not being systematically archived, and to our knowledge are not yet being used to explicitly infer geographic information or for environmental monitoring.

The creation of large, labeled image datasets is a challenging effort and represents a significant contribution to the community. Recent examples include datasets of many small images [77], with labeled objects [66], and of labeled faces [25]. Each of these datasets fills a niche by providing different types of labeled images. More similar to the AMOS dataset [28] is one with many images, and associated metadata, from a single carefully controlled static camera [49]. The AMOS dataset is unique in providing time-stamped images from many cameras around the world. Recently, construction of a webcam image dataset, similar to the AMOS dataset, began [38]. The construction of this new dataset highlights the research interest in large datasets of outdoor imagery. To date, no other dataset simultaneously provides the broad range of geographic locations and the long temporal duration that characterize the AMOS dataset.

Given the vast number of images in the AMOS dataset—more than 50 million as of January 2010—it is often challenging to find a subset of images that are suitable for a particular algorithm evaluation. Compact summaries can enable rapid browsing of a large collection of images. One area of previous work is on image-based summaries of a video; see [51] for a survey. Another interesting approach uses a short video clip to summarize the activity in a much longer video [61]. To our knowledge, all the previous work is designed to work with high frame-rate video. We present visualizations that highlight the geographic nature and long temporal duration while simultaneously handling a very low frame rate and very long duration dataset.

2.2.1 Browsing Webcams and Webcam Imagery

In working with a large archive of images from many webcams, we find that visualization tools are critical for debugging, updating, and maintaining the capture system. This section describes how we visualize the AMOS dataset in the form of a web site.

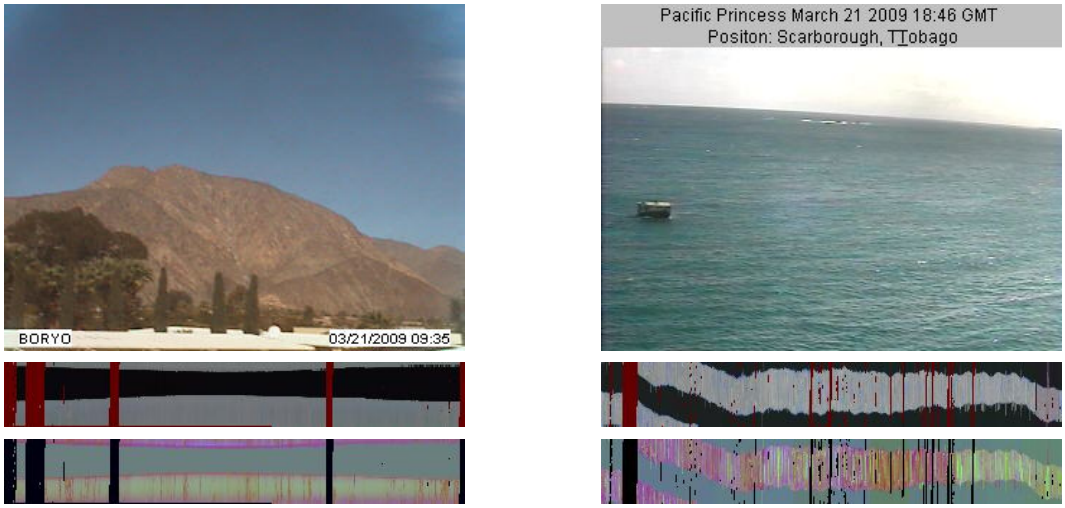
The AMOS website [1] supports browsing to find cameras as well as images relevant to particular tasks.

Currently, we use a two-layer web interface to display the dataset. First, a page shows the current image of every camera in the data set, or a subset based on keyword/tag filtering as shown in Figure 2.3. This filtering and browsing interface is important for determining how many cameras may support a particular task, and for determining whether cameras are broken or not delivering images (a common occurrence because the cameras are controlled by third parties and are therefore supported and maintained with varying degrees of attention). Second, each camera has a dedicated page which includes all known meta-information (e.g., tags, geo-location, geo-orientation, and internal calibration), as well as an interface that supports searching for a specific image from a camera.

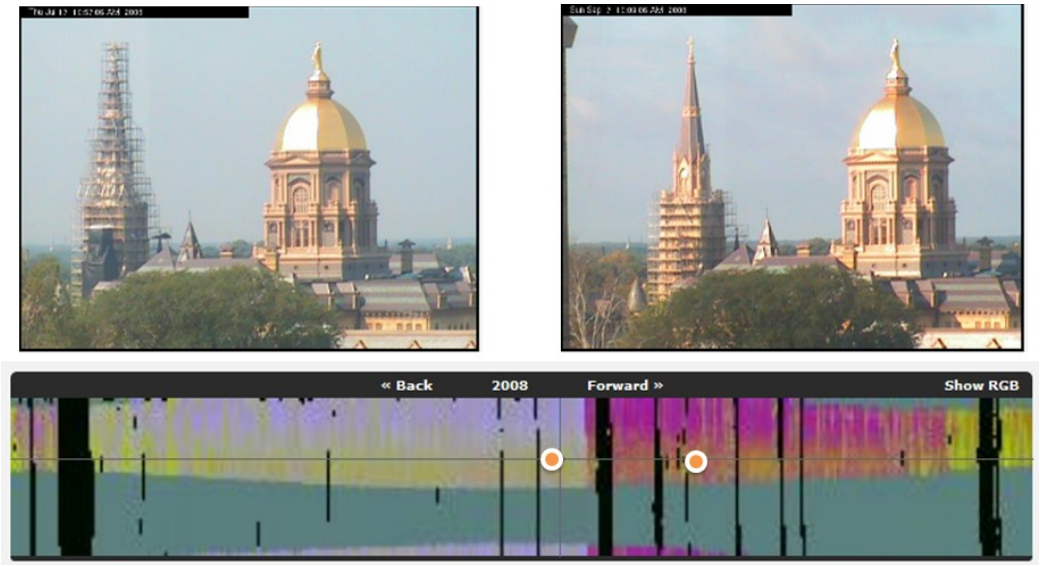
Searching for a specific image from a camera is done using a summary of the image appearance over the course of each year. The first instance of this yearly summary is an image indexed by time of year (on the x-axis) and time of day (on the y-axis). At each pixel, this image shows the mean color of the entire image captured at that time on that day. Figure 2.5(a) shows two example cameras and the annual summaries of those cameras for 2008. This interface makes it very easy to see when day (gray), night (black), and missing images (dark red) occur at a camera. For example, the left image shows the nighttime growing shorter during the middle of the summer. The right side of Figure 2.5(a) shows the unusual circumstance of a camera for which the night time seems to drift throughout the year; this camera is mounted on the bridge of a Princess Cruises ship that circumnavigated the globe in 2008.

The web interface allows the user to click a pixel on this summary image, and shows the image taken on that day, closest to the selected time. This gives an intuitive way to view, for example, a large set of images taken near dawn, by selectively clicking along the day-night interface. Additionally, keyboard interfaces allow moving to the image taken at the same time the previous day, or moving forward and backward within a day, to give time-lapse movies at different time resolutions.

However, the two summary visualizations shown immediately below the images in Figure 2.5(a) are less informative than one would like. Since many cameras perform both contrast equalization and color balancing in order to produce reasonable pictures



(a)



(b)

Figure 2.5: Examples of summary images we use to visualize a year of webcam images. (a) Images from two cameras in our database with the corresponding RGB- and PCA-based annual summary images. The right camera is on a cruise ship that circumnavigated the globe during 2008; this causes nighttime to "wrap" during the year. (b) An example where the PCA-based summary image highlights a small change in the camera viewpoint; the dots on the summary image correspond to the images (above) that show a small viewpoint shift. The time when the shift occurs corresponds to the summary image changing from yellow/blue to purple.

at all times of the day, this summary image often shows little more than a clear indication of when daytime is, and other changes such as shifts in camera viewpoint or changes in scene color may not be visible.

A more abstract but informative visualization can be achieved by performing principal component analysis (PCA) on the set of images, which we compute incrementally using Brand's [5] algorithm. The set of images from a camera $\{I_1, I_2, \ldots, I_k\}$ is approximated as the linear combination of the mean image μ and of three basis images $\{b_1, b_2, b_3\}$ so that for each image i, $I_i \approx \mu + \alpha_{i,1} b_1 + \alpha_{i,2} b_2 + \alpha_{i,1} b_3$. The vector of coefficients $(\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3})$ gives a compact description of image i relative to the overall variation in images seen at that camera. We normalize these coefficients and use them to define the RGB channels of a *false-color* summary image.

Both forms of this summary visualization make it simple to determine if cameras have captured data at particular times of year and allow rapid navigation through the large image dataset. The PCA-based summary visualization is ideal for highlighting camera movement. The bottom of each part of Figure 2.5 shows this visualization for three different cameras. In particular, this highlights the consistency of the daily variations of the desert scene, the inconsistency throughout the year of the view from the cruise ship, and the slight change in orientation of the image of the golden rotunda.

2.2.2 Manual Object Labeling

We have worked to increase the diversity of the annotations available for image browsing, beyond location, time, and crude scene categories, by incorporating an object annotation tool [66]. This web-based tool enables users to label regions of the image that correspond to objects in the world, such as trees, roads, and buildings. This form of annotation is easy to create and could be used to support additional means of browsing the dataset.

In addition to aiding dataset navigation, the localized object annotations are useful for learning-based methods and algorithm evaluation in computer vision. These methods often require large amounts of training data to support learning of complicated classification and regression models. Using AMOS as a source for training data

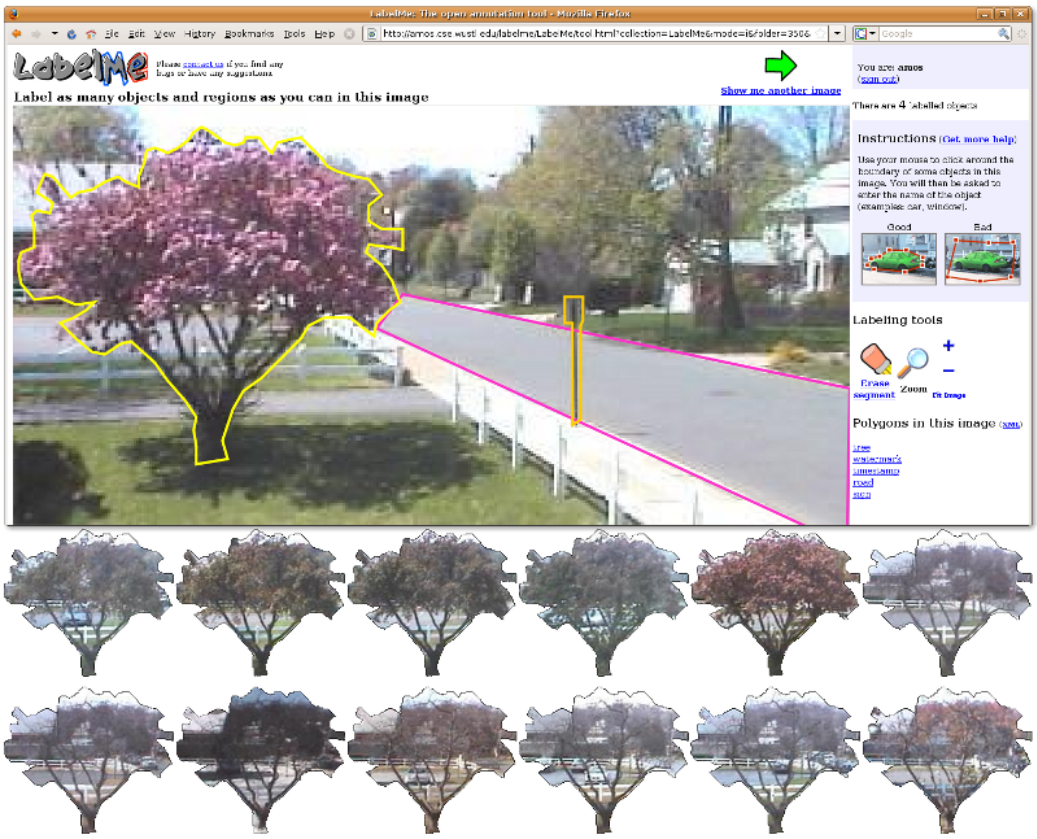
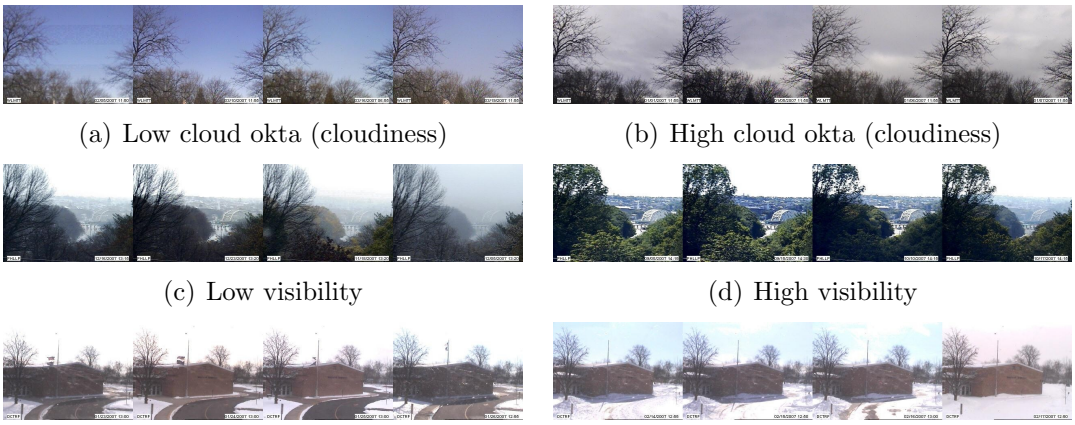


Figure 2.6: We use the LabelMe annotation tool [66] to rapidly annotate many images from a webcam. The annotations from a single image extended through time during periods without camera motion.



(a) Low cloud okta (cloudiness)

(b) High cloud okta (cloudiness)

(c) Low visibility

(d) High visibility

(e) Low snow depth

(f) High snow depth

Figure 2.7: Automatic image labels can be created by spatially and temporally registering webcam images to weather reports. Above are montages of webcam images that correspond to extreme readings for a variety of weather properties.

is intriguing because it addresses one of the significant challenges in creating training datasets: the time-consuming process of labeling images. We make the observation that when the camera is static, object annotations from a single frame can be extended through time. Using the web-based image annotation tool, it is possible to annotate static scene elements and obtain views of the same scene element in many weather conditions and seasons. In some cases, it is possible to obtain many thousands of pictures of the same scene element. Figure 2.6 shows an example of one such annotation extended through time.

2.2.3 Automatic Scene and Image Labeling

One advantage of having an accurate location estimate for a camera is that it facilitates integration with existing GIS databases. This enables, for example, the rich annotations present in these databases to be transferred to the cameras. Also, these annotations can determine if a camera is in a dense urban area or farm-land or if it is more likely viewing a road than a river.

In addition to cameras, individual images can be automatically labeled. To demonstrate this, we spatially and temporally registered sequences of webcam images to historical weather readings from the National Climatic Data Center [50]. Figure 2.7

shows the type of filtering possible when local weather readings are registered to webcam images. As with the manual scene labeling described in the previous section, this labeling provides an interesting, additional source of training data for computer vision algorithms. At a more fundamental level, it supports future work in exploring the use of geographic location as a contextual cue to aid in image understanding.

The following section uses the AMOS dataset to better understand the properties of the set of images captured over a long period of time from a static outdoor camera.

2.3 Statistics of Outdoor Scenes

The statistical properties of images captured from a single static outdoor scene are significantly different from those of image datasets composed of images captured from many different viewpoints. Work on the statistics of natural images has primarily explored the latter case and ignored the former. However, understanding these statistical properties is fundamental to working in surveillance and, more generally, outdoor scene understanding. This section describes an empirical evaluation of second-order statistical properties of static outdoor video using images drawn from the the AMOS dataset.

We initially follow the methods and approach of work characterizing the statistics of arbitrary natural image patches and windows of short video clips. But for video taken from a single viewpoint, the same analytic tools find much more specific statistical correlations. These correlations relate to important scene features; for example, image regions that share geometric features such as surface normal and depth have correlated responses to lighting changes. Clustering of appearance changes [34] and explicit modeling of the physics of scattering media [48] have shown impressive results on segmenting scene structure and weather patterns of long sequences of images from a static camera [49]. We claim that these structures are available in data from static cameras without complicated algorithms or physical modeling, using only principal component analysis over time scales of days, weeks, and months. Furthermore, static cameras show surprisingly similar types of variation that can be unified into a canonical decomposition. This supports the automatic annotation, in any static camera, of the scene structure at a pixel location.

Related Work: Image Statistics

Studies of natural images have considered second-order statistics through the PCA decomposition [18], and, more recently, using higher order statistics and Independent Components Analysis (ICA). When ICA is applied to natural image patches to find optimal sparse codes, it produces basis images that appear very similar to receptive fields in the visual cortex (for example [53, 80]). However, these statistics are only computed for relatively small patch sizes because in natural images, there are only weak correlations between pixels that are far apart.

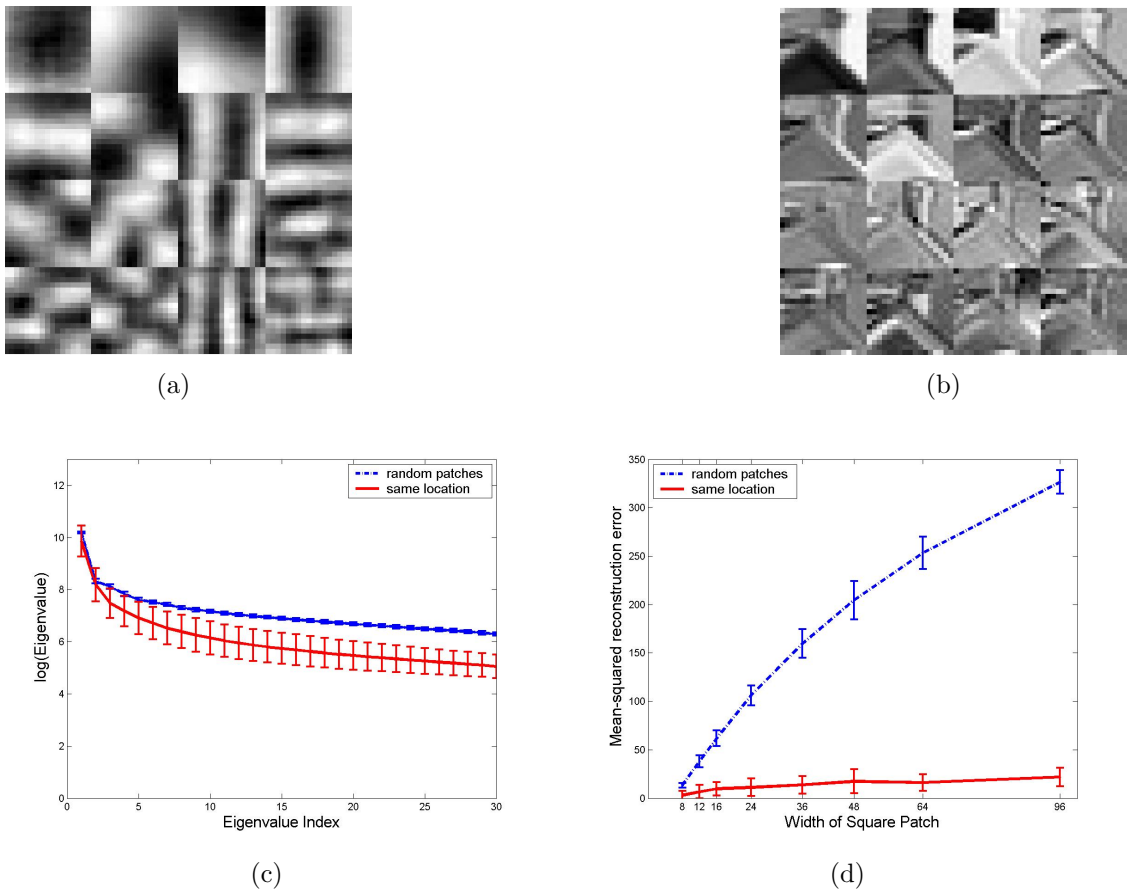
In replicating these studies on image patches taken from the same location in a static camera, we find empirical evidence (see Section 2.3.1) that location-specific bases are much more informative than patch bases developed on generic natural image patches. These bases reflect stronger correlations between distant pixels, and that these strong correlations exist over extremely large patches.

Correlations between multiple images have been studied for natural video where the dominant cause of image change is camera motion. In this case, small space-time patches have non-separable spatial and temporal correlations [13], and in optimal sparse codes designed for such time-varying natural imagery, nearly all of the basis functions code for motion [52]. These are also studied largely on small patches, as correlation between pixels decrease with longer spatial and temporal distances.

2.3.1 Natural vs. Location-Specific Statistics

We compare the second-order statistics of natural image patches and location-specific patches. One goal of this is to discover how much benefit there is to making a representational basis that is specialized to a particular location and to measure how this benefit scales with patch size. For all basis computations in this work we use the singular value decomposition (SVD) if memory permits; otherwise, use incremental SVD [5].

In this section, we characterize the singular values of the SVD and reconstruction error for varying patch sizes and linear basis functions. For each camera in the AMOS



(a) (b) (c) (d)

Figure 2.8: The covariant structure of multiple patches from the same location in a scene is much more informative than the covariant structure of arbitrary patches in natural scenes. This is apparent from the structure of the first 16 principal components, comparing (a) patches taken at random locations from an image sequence, and (b) patches taken from the same location in 2000 images from a static camera. This effect holds over many components and at many scales, (c) the singular values of the SVD are significantly lower (note this is a log-plot), and (d) the mean-squared error of the pixel intensity reconstruction stays nearly constant as the patch size increases.

database, we have approximately 2000 images taken during October and November. Location-specific statistics are created by randomly choosing a patch location, and for each image collecting the pixel values of that patch in a column vector I_j. We collect these vectors in a matrix $\mathbf{I} = I_1, \ldots, I_n$ and compute the SVD $\mathbf{I} = U\Sigma V^\mathsf{T}$ (since we first subtract the mean, the columns of U are PCA component images and the columns of V are the PCA coefficients). We compute the reconstruction error for 200 randomly selected patch locations to determine the mean and standard deviation. The natural (non-location specific) image statistics are computed by selecting one patch from a random location (uniformly across the image) in each image of the scene. This naturally enforces the goals of using the same number of patches, and of sampling from images throughout the day. This is repeated 200 times to determine the mean and standard deviations of the singular values and reconstruction error.

The results are shown in Figure 2.8; at the top are example results from (a) natural image patches and (b) location-specific patches. The principal components of one set of 2000 natural image patches resemble a 2D frequency decomposition, as has been widely reported (see, for example [67]). These components look qualitatively similar throughout different repetitions. In contrast, the principal components for the location specific patches are drastically different from the natural components and between repetitions because they reflect the structure of the scene in view at that location.

For a fixed patch size, the difference in the magnitude of singular values remains large out to as many values as we have computed, Figure 2.8 (bottom left) shows the mean and variance of the singular values for a 16×16 patch. The differences become even more dramatic for larger patch sizes. Using a fixed number of components (30), Figure 2.8 (bottom right) reports the mean and standard deviation of the mean-squared reconstruction error. This reconstruction error grows very slowly as a function of patch size, because most variations in appearance from a fixed camera are caused by lighting changes that affect large parts of the scene. Since the reconstruction error remains small for large patches, the remainder of our analysis considers principal components across the entire image.

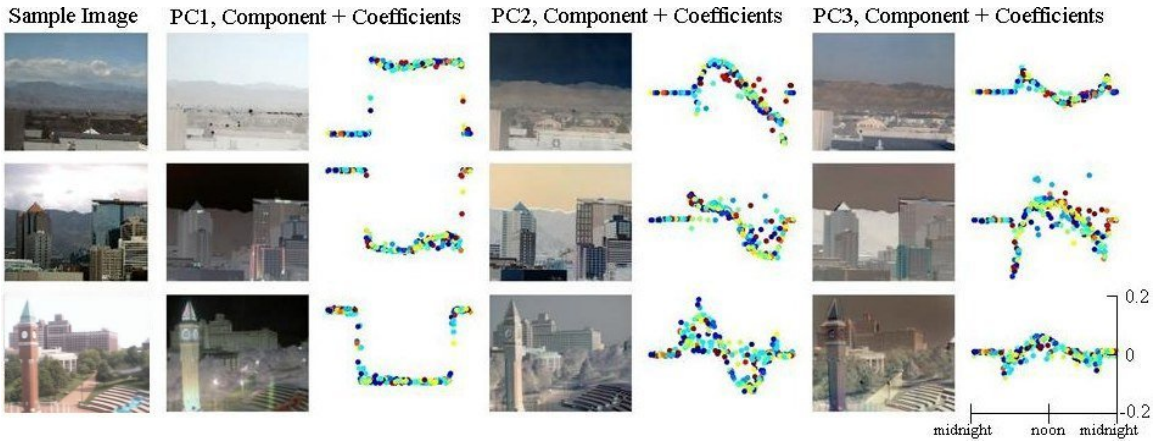


Figure 2.9: The most significant principal components of outdoor video captured by a static camera are often dependent on the time of day. This figure shows a sample image, and the top three PCA component/coefficient pairs for three cameras. The PCA components are normalized to range from 0 to 255 so they can be displayed as standard images. The x-axis of the PCA coefficient plots is determined by the time of day the corresponding image was captured, and the points are color-coded by the day.

2.3.2 Daily Variations of Outdoor Scenes

Not only are the second-order statistics of static cameras interesting over large spatial scales, similar structures can be seen across multiple cameras. In this section, we support this claim with examples from the AMOS dataset.

We first take a 2000-frame sequence, with times distributed over a week, and compute both the principal components and the coefficients used to linearly reconstruct each image. Figure 2.9 shows for three cameras, a sample image, the first three principal components and the coefficient values plotted as a function of their time of day (color-coded by which day). These coefficients are strongly correlated with time of day and are surprisingly similar between cameras. Changes from day to day are also visible due to the color-coding. Next, we isolate changes due to time of day by constructing a set of average images.

To isolate changes due to the time of day, we construct an *average day*. For each camera, this consists of a set of 48 images, each estimated by averaging all images captured over the course of a week for a particular half-hour portion of the day. We then perform a PCA decomposition of the *average day*. Figure 2.10 shows coefficient trajectories from several cameras (i.e., three leading columns of V) for an *average*

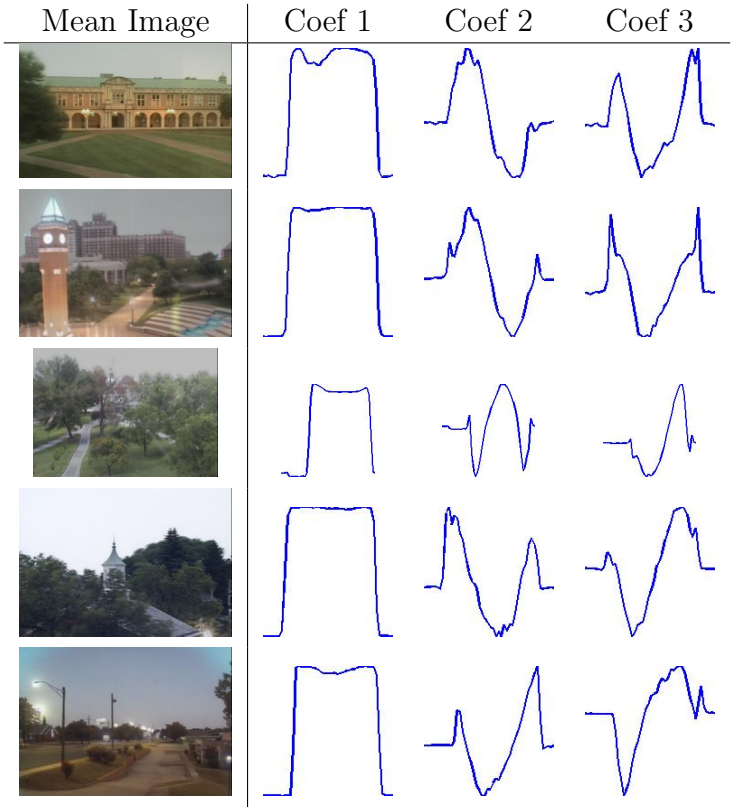


Figure 2.10: The principal component coefficients of static images of outdoor scenes have consistent patterns. This figure shows the mean image and plots of the first three PCA coefficients for one day for several camera. The x-axis of each plot is the time of day and the y-axis is the coefficient value. The coefficients for different cameras are similar despite imaging very different scenes.

day. We find that a short duration—one week—is sufficient to reduce the impact of seasonal and daily variations.

In a study of 538 cameras, we find that, for the vast majority, the leading PCA component of the *average day* encodes the difference between day and night. Differences in dawn and dusk time due to geolocation or natural seasonal variation cause the times of the sharp transitions to change. The second and third components have coefficient trajectories that indicate differences due to the sun's position. The scene-specific components highlight appearance changes between the sun facing east and west, and differences between dawn and dusk and the middle of the day. The ordering of these components is not fixed; we address this in the following section.

2.3.3 Consistent Temporal Variations

In this section, we explore temporal variations due to the time of day. To isolate variations due to transient phenomena such as weather and moving objects, we construct an *average day*, a set of 48 average images, one for each half-hour of the day, from all of the images from the month of June 2006.

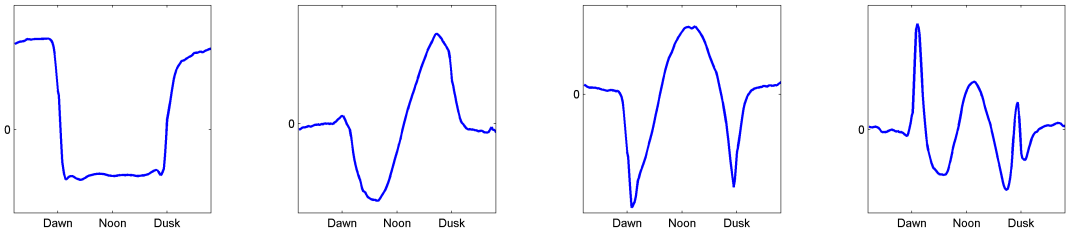
The SVD of this set of images highlights that while the the principal components are strongly dependent on the scene, the coefficient matrices V of different cameras are surprisingly similar. Figure 2.10 shows the first four principal component coefficients of several cameras, plotting columns of the coefficient matrix V_i, which, by construction of our image set, corresponds to time of day.

The primary differences between cameras in the coefficient trajectories are due to three factors: a shift due to local dawn/dusk time, a column permutation due to the relative strength of different types of variation in the scene, and inversion due to the non-uniqueness of the SVD. The remainder of this section describes a method of eliminating these and other, minor, differences between the coefficients.

First, we temporally align the coefficient matrices, V_i, by considering the first column as a function of time and computing the extrema of its derivative (this corresponds to finding the rise and fall of coefficient one in Figure 2.10). All coefficient matrices are then linearly interpolated to have the same number of coefficients before dawn, during the day, and after dusk. For cameras with known latitude and longitude, the standard deviation of our estimates when compared to standard civil twilight on June 15, 2006, was 19 minutes. This is a reasonable error value, since there is only one image for every 30 minutes.

The remaining variation is in the order and sign of the columns of the coefficient matrices. Starting from temporally aligned coefficient matrices V_i, we solve for a coefficient matrix $\bar{V}$ that is a solution to the following problem

$$\begin{array}{rl} \text{minimize} & \sum_{i=1}^{n} \max_{p \in \mathbf{P}} \left\| \bar{V}^{\mathrm{T}} p V_i \right\|_{\mathsf{F}} \\ \text{subject to} & \bar{V}^{\mathsf{T}} \bar{V} = \mathbf{I} \end{array} \tag{2.1}$$



(a) Coef 1 (b) Coef 2 (c) Coef 3 (d) Coef 4

Figure 2.11: The first four canonical component coefficients learned from the AMOS dataset.

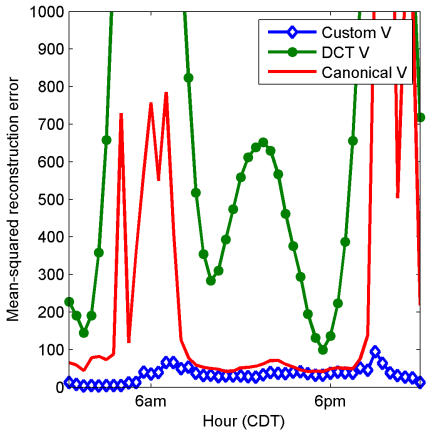


Figure 2.12: Average reconstruction error of half-hour average images for all cameras using three components.

where $\mathbf{P}$ is the set of generalized permutation matrices with entry values in the set $\{0, 1, -1\}$ and only one non-zero entry in each row and column. Figure 2.11 shows the first four canonical coefficients learned from a randomly selected set of 145 coefficient matrices.

Using $\bar{V}$, we can now decompose images from any camera in a way that facilitates image and scene understanding. Given images $\mathbf{I_j}$ from camera j, we can solve linearly for an orthogonal matrix of canonical components $\bar{U_j}$ and a diagonal matrix of weights $\bar{\Sigma_j}$ that are the solution to

$$\mathbf{I_j} = \bar{U_j}\bar{\Sigma_j}\bar{V_j}^{\mathsf{T}} \tag{2.2}$$

where $\bar{V_j}$ is $\bar{V}$ temporally aligned to this camera.

The canonical-day decomposition is not as good, in terms of the squared error, at reconstructing images as the camera-specific SVD for the same number of components, but it is better than a generic low frequency decomposition (using DCT coefficients in place of V) by a factor of two. Figure 2.12 shows the reconstruction error by time of day. The average reconstruction errors for V_i, $\bar{V}_i$, and a discrete cosine transform matrix (commonly used in compression), are respectively 32.6, 302.2, and 866.7. The reconstruction errors using the canonical-day coefficients $\bar{V}_i$ are significantly lower than when using the DCT matrix V_d. The results also show that when using the canonical components most of the error occurs near dawn and dusk and that the components are fairly accurate during the day.

Image Labeling Using the Canonical-Day Decomposition

Using the canonical-day decomposition, we can label individual images from the scene. Given any image I_j taken at time t from the scene, we project it onto the canonical-day components to obtain a vector of weights $c_j = \bar{U}_i^{\mathsf{T}} I_j$. These weights are then compared to the corresponding values, based on time of day, of the canonical-day coefficients. As an example, Figure 2.13 shows a scatter plot of images colored by $c = |\bar{V}_i(t,2)\bar{\Sigma}_i(2,2)| - |\bar{U}_i(:,2)^{\mathsf{T}} I_j|$. This measure correlates with the cloudiness of the current image.

Pixel Labeling Using the Canonical-Day Decomposition

The canonical components that result from Equation 2.1 have a consistent meaning across all cameras because the coefficient matrices are shared. This allows one annotation scheme to be applied to all cameras. To highlight this capability, we solved for the canonical components of 12 scenes and created a false-color image for each scene. We create a false-color image whose three color channels, RGB, are the third, second, and the negative of the first canonical components; this order was chosen so that strongly negative parts of the first canonical component are blue, mimicking the sky. This false-color image strongly correlates with scene structure; example images are shown in Figure 2.14. This results in a rough color-coding of the scene that separates

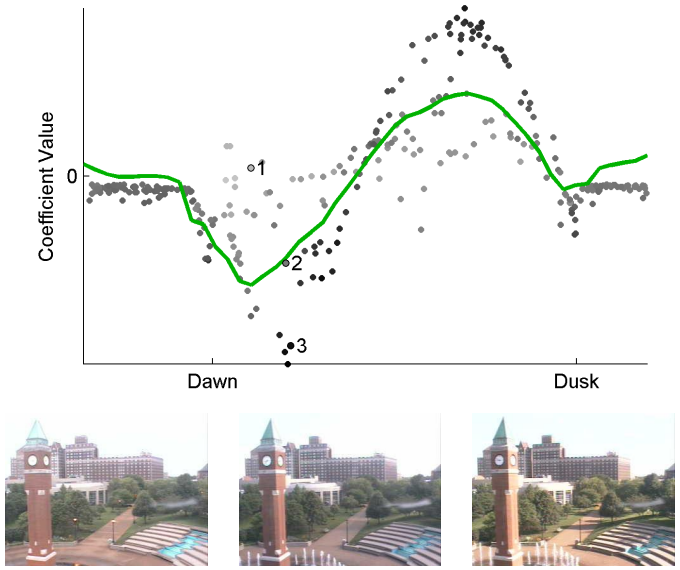


Figure 2.13: Using canonical-day decomposition to determine weather conditions. The plot shows two types of coefficients for a set of images from a single camera. The solid line represents the values of the second canonical-day coefficient (automatically aligned with dawn and dusk). The dots represent individual images from the camera with the x-value corresponding to the time when the image was captured and the y-value equal to the projection of the image onto the second canonical-day component. The dots are colored based on a function described in Section 2.3.3.

trees and horizontal surfaces (light green) from eastward facing walls (orange/red) from westward facing wall and sky (blue).

2.3.4 Variations at Longer Time Scales

We have shown that consistent patterns of daily variation occur across many cameras viewing a broad range of scenes. In this section, we explore longer time scales and find significant variations due to weather conditions, human activity, and the change of seasons.

We begin by examining variations that occur from day to day. In order to reduce effects due to the time of day, we only look at images from one hour of the day. We create a set of 30 images for each camera, one for each day of June 2006, by averaging all images captured between 12:00 pm and 1:00 pm on each day. We then decompose this set of images using the SVD. Empirically, there is less regular temporal structure in this basis than in the basis of half-hour average images, but the first component

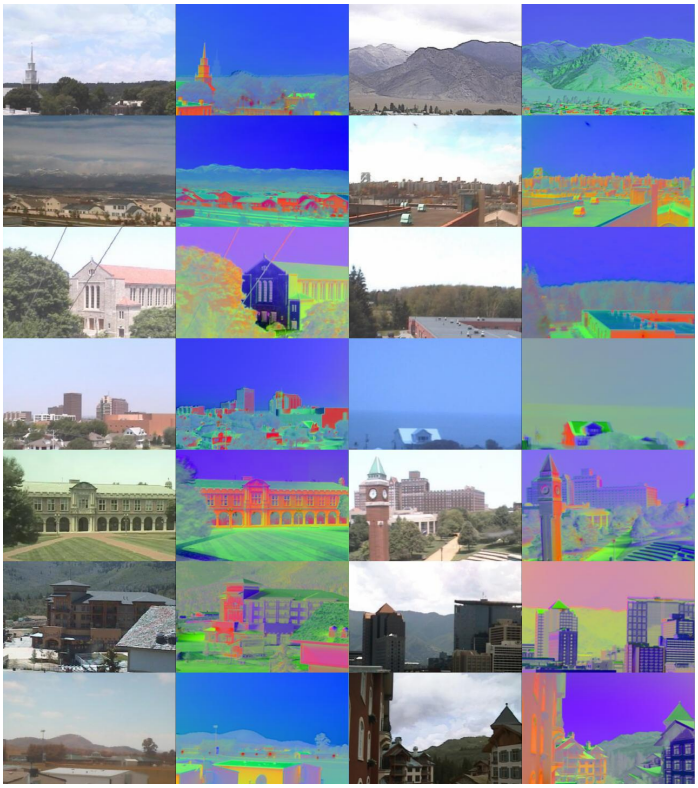


Figure 2.14: The components of the canonical-day decomposition code for lighting variations. The above shows a collection of pairs of an example image from a camera and a false color image made from the first three components of the canonical-day decomposition. The colors indicate sky (light blue), trees (light green), eastward facing wall (orange), westward facing wall (blue).

Figure 2.15: Images of day-to-day variations, described in Section 2.3.4, organized by the value of the first principal component coefficient. This value changes from day to day and is often dependent on the weather; occasionally it is dependent on human causes. The first principal component of the scene on the bottom is dependent on the presence of cars in the parking lot.

often has interesting structure. While the first component often is related to whether or not it is cloudy, it is occasionally caused by human activity (see Figure 2.15 for examples).

We now turn to variations that occur at the scale of many months. Variations of this type include changes in shadow positions (due to changes in the zenith angle of the sun), changes in weather conditions (e.g., snow on the ground), and changes in plants (e.g., the presence or absence of leaves on deciduous trees). To reduce the effect of short-term causes, we create a set of average images for each camera that includes primarily long-term variations. To do this, we divide the year into 15-day intervals and create an image for each interval. The image is the average of all images within

(a)

(b)

Figure 2.16: The PCA decomposition of an outdoor scene can also highlight seasonal variations. (a) A subset of 15-day-interval average images we constructed to reduce the impact of non-seasonal variations. (b) The first principal component (left) encodes primarily for the presence of trees, the second (right) for different types of trees.

the interval captured between 12:00 pm and 1:00 pm. While the PCA decompositions of these sets of images are often highly structured (see Figure 2.16 for an example), we find that they are much less consistent across cameras than the daily variations in Section 2.3.2.

2.3.5 Summary

We have found that image sets from static cameras have strong correlations over large spatial and temporal extents. The principal components of these data sets, and

their temporally aligned variants, are useful because they can be compared between cameras and provide simple and automated tools to extract scene structure. We believe that understanding long-scale spatial and temporal correlations in static video sequences is vital to better understanding the statistics of natural imagery. It may also directly affect the compression and transmission of surveillance video and maintenance of surveillance background models.

Chapter 3

Unattended Camera Calibration

Automating tools for geo-locating and geo-orienting static cameras is a key step in creating a useful global imaging network from cameras attached to the Internet. We present algorithms for partial camera calibration that use collections of time-stamped images as input. While the problems of determining the location and orientation of a camera have been extensively studied, in the webcam domain these problems require novel solutions. Traditional methods rely upon large camera motions, manually manipulated calibration objects, or overlapping fields-of-view. We present camera calibration methods that work with cameras that have little or often no motion and that capture low-resolution images. And, importantly, the methods do not require physical access to the camera.

3.1 Related Work

Camera calibration has a long history in the computer vision literature. Finding the extrinsic calibration (the positions and orientations) of cameras in a network has been extensively studied in the cases where there are feature correspondences between multiple cameras [31, 2].

Within the deployment of more distributed camera networks, distributed versions of these calibration problems have been proposed [12, 42]. Various cues, in addition to feature correspondences, have been proposed to define camera topologies or approximate relative camera positions based on object tracks [43] or statistical correlation of when objects enter and exit the camera views [76]. Both methods allow inference of

camera locations when the camera fields of view do not overlap, although the cameras must be close enough so that objects appear or disappear between cameras, and there is a low entropy distribution of differences between departure times (from one camera) and arrival times (in another camera).

Geo-calibration is less studied but has been addressed by matching features in the image to features computed from a digital elevation map [74, 10, 70]. Early work on using natural variations for camera geo-calibration was based on explicit measurements of the sun position [9], which was followed by work in computing absolute camera orientation [78]. These techniques require the sun to be in the field of view and accurate camera calibration.

Many algorithms have been developed to infer scene and camera information using fixed-view time lapses. Examples include a methods for clustering pixels based on the surface orientation [34], for factoring a scene into components based on illumination properties [73], for obtaining the camera orientation and location [30, 29, 72, 39] and for automatically estimating the time-varying camera response function [33].

3.2 Camera Localization

This section explores the following localization challenge: Given static cameras that are widely distributed in a natural environment with no known landmarks, no ability to affect the environment, and perhaps no overlapping sensing areas ("fields of view"), discover the positions of the cameras. This is an important problem with webcams because the geographic position is often not provided and manual human localization is time consuming.

We first consider localizing the cameras using a commonly used database that provides estimates of the geographic location of an IP address. We show that this method, while simple and fast, has limitations and is often highly inaccurate.

We conclude with a unified method that uses two distinct cues to the geographic location of the camera. The key point in the problem definition is that we consider

real, natural environments. Sensing data from natural environments has useful properties for localization. First, variations in natural environments happen at many time scales, examples include changes due to daylight, weather patterns, and seasons. Second, because these phenomena are spatially localized, over a long period of time the time-course of these variations is unique to a particular geographic location.

3.2.1 IP Address Lookup

Our first method for estimating the location of the camera is based entirely on the network address of the website hosting the images the webcam captures. The process is straightforward: First translate the webcam URL into an IP address, and then query the IPInfoDB geolocation database [26] to obtain an approximate camera location.

To better understand the accuracy of IP-based webcam localization, we performed an intensive study on a randomly selected set of 200 cameras. We then attempted to manually localize these cameras guided by contextual cues and visual alignment of image features with satellite imagery. This resulted in a set of 138 cameras for which we were confident that our location estimate was within 100 meters of the true location. Comparison of the IP-based estimates with manually generated ground-truth camera locations (see Figure 2.2(e)) shows that there is significant error in the IP-address-based location estimates. In fact, half of the cameras have a localization error greater than 111km.

In reviewing the sources of error of this method, we find two primary causes. First is the inherent difficulty in mapping the geographic location of every IP address: the database is sometimes incorrect. This failure mode will likely be addressed as geolocation databases improve over time. Second, the geolocation estimate may be accurate for the webcam URL, but the server hosting the webcam images might not be geographically near the webcam. This is a failure mode that is unlikely to be eliminated by advances in IP-geolocation databases. This motivates the use of methods that use alternative cues to determine the location of a webcam.

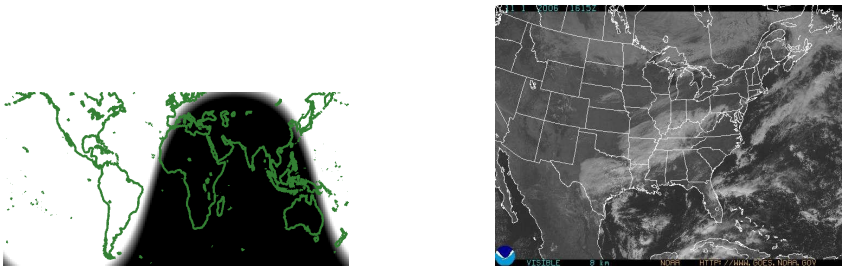
3.2.2 Time-lapse Camera Localization

We describe a method for estimating the location of a camera using natural image appearance variations. We show that the variations in ground-based cameras can be related to variations in geo-registered satellite imagery, and that this can be used to estimate the location of the camera. Since the mapping from satellite image coordinates to a global coordinate system is known, the localization problem reduces to determining which pixel in the satellite image is the most likely location of the camera. We use straightforward statistical techniques and show that this is possible using only a small number of principal component coefficients of images from the camera and satellite images taken at the same time. Notice that we are being intentionally vague about the specific type of satellite image. In the results, we show examples with a visible-light and a synthetically generated day/night satellite image.

We find that by using natural temporal variations that cause large-scale image changes, as discovered by PCA, our methods gives an accurate estimate of the camera location when other methods are likely to fail; specifically, it is robust to imaging distortion and works well even with a limited field of view.

For a collection of T different time points, we find the satellite images and camera images taken closest to each time point. The geo-registered satellite images are combined into a matrix of $S \in \mathbb{R}^{p \times T}$, where each column is an image. The camera image data is decomposed using incremental SVD [5] to approximate the first k PCA components of the camera images. The corresponding coefficients define the matrix $V \in \mathbb{R}^{T \times k}$, where each column is the time-course of one coefficient from the camera we are attempting to localize.

For a given camera, we compute the correlation score of each pixel in the satellite image. This score is defined as the correlation of the individual pixel time-series signal (the rows of S which encode how that pixel of the satellite varies through time), and a signal constructed as a projection of the PCA coefficient matrix V. We construct this projection as the linear combination of the rows of V^{T} that is closest to the satellite pixel signal in the least-squares sense. This score can be computed for all pixels at once as:



(a) (b)

Figure 3.1: Examples of geo-registered images that we use to localize cameras. (a) A synthetic satellite image in which intensity corresponds to the amount of sunlight. (b) A visible-light image from a geostationary satellite.

$$diag(S(SV(V^{\mathsf{T}}V)^{-1}V^{\mathsf{T}})^{\mathsf{T}})$$

Allowing for the pixel to correlate with a linear combination of the PCA coefficients provides robustness to the ordering of these PCA coefficients. Computing this score for every pixel yields a false-color satellite image in which pixel intensities correspond to the temporal similarity of the pixel to the camera. Examples of these images for two types of satellite images are shown in figures 3.4 and 3.2.

All experiments were performed on the AMOS dataset described in Chapter 2. To enable quantitative evaluation, we selected the cameras with known latitude and longitude coordinates. Cameras that moved (including rotation or zoom) during the two testing time frames (April 2006, February and March of 2007) were rejected from the dataset.

3.2.3 Localization Using a Synthetic Daylight Map

As a baseline for comparison, we consider the case in which no satellite coverage is available. We use the algorithm described above without modification on a synthetic daylight map where intensities correspond to the amount of sunlight (Figure 3.1(a) shows an example). These images are generated by thresholding the solar zenith angle z for a given time and location. Pixels intensities are as follows: black if $z > 100$, white if $z < 90$, and varying linearly between the thresholds. Examples of correlation

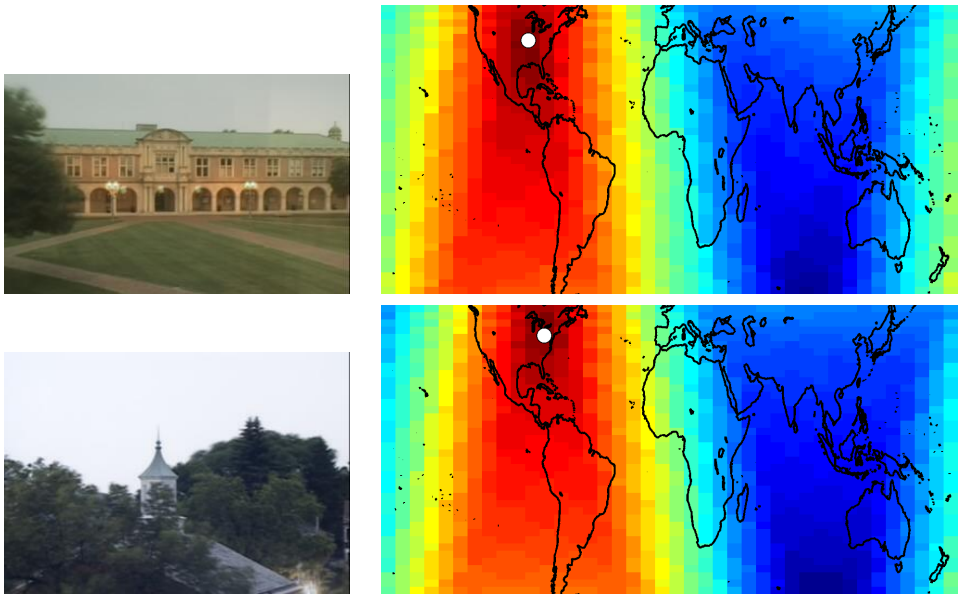


Figure 3.2: The correlation of the first PCA coefficient with pixels from the synthetic daylight map shown in Figure 3.1(a). The region with the highest correlation corresponds to the location of the camera (white dot).

maps generated from this dataset are shown in Figure 3.2. This method gives very similar results to an algorithm that specifically searches for dawn and dusk in the image data and uses the length of the day and the dawn time to calculate position.

3.2.4 Localization Using Visible Satellite Images

We now present results of localizing cameras using images from the NASA Geostationary Operational Environmental Satellite [14]. See Figure 3.1(b) for an example image from the satellite dataset. We tested on two 300-image datasets from two satellite views: one of the Maryland area and one of the Pennsylvania area. We find that by using visible satellite images, our algorithm localizes most cameras within 50 miles of the known location. Figure 3.5 shows a histogram of errors in the predicted locations. Figure 3.3 shows the actual position and our estimates for cameras in Pennsylvania. The mean localization error over all cameras is 44.6 miles; this is skewed by dramatic errors in a few cameras and dropping the eight outliers reduces the mean to 23.78 miles. A major source of error for the outliers is incorrect time stamps. This occurs when a camera does not capture a new image for each request

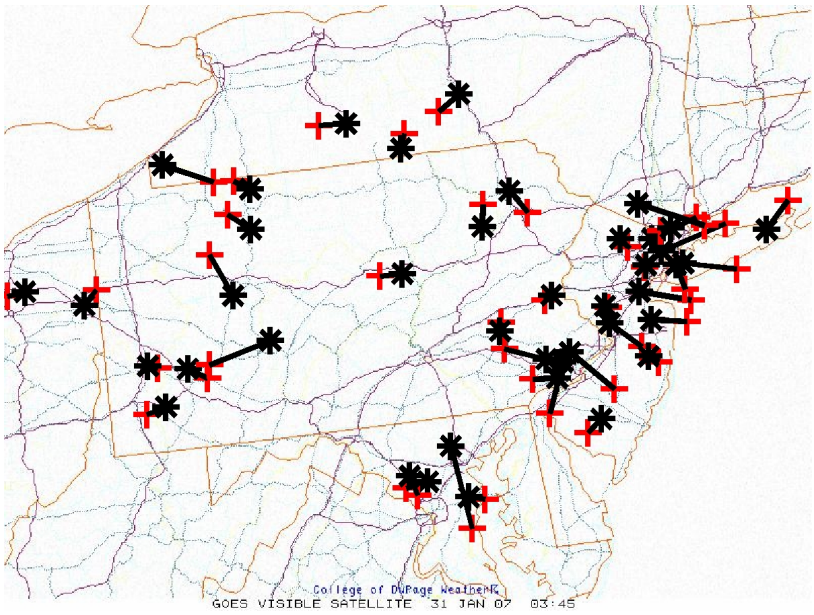


Figure 3.3: A comparison of the estimated (red crosses) and actual locations (black stars) for the Pennsylvania-area cameras.

but instead captures an image on a fixed schedule; many webcams only capture a new image every 15 minutes. For these cameras, our capture program records an archive time that is, on average, 7.5 minutes after the correct time. This delay leads to a bias of about 99 miles $= q2\pi\cos(40\,^\circ\,\text{latitude})3\,963$ miles, for a camera in Pennsylvania, where $q = 7.5/(60 \times 24)$ is the fraction of a day elapsed.

3.3 Estimating Camera Geo-Orientation

Our approach to estimating the geo-orientation of a camera consists of three steps. First, we find a sequence of images where the camera has not moved. Second, we label pixels that image the sky, either manually or automatically using the canonical-day decomposition [29]. Third, we use a graphics rendering method that creates a full hemispherical sky-intensity map for a given geo-location and time of day. An efficient optimization process finds the orientation of the camera relative to this hemisphere, which maximizes the correlation between the sky pixels in the image and the predicted sky intensity.

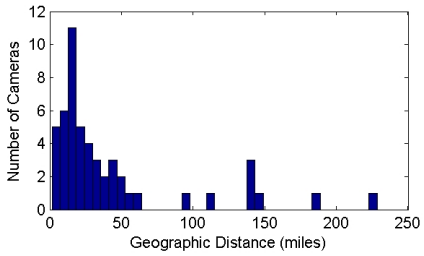
Figure 3.4: Three example results that use visible satellite imagery to estimate the camera location. (left) Example images from three of the 538 static webcams that have been logged over the last year. (right) Correlation maps with satellite imagery; a measure of the temporal similarity of the camera variations to satellite pixel variations is color coded in red. The cross shows the maximum correlation point, while the star shows the known GPS coordinate. Note that our method can estimate the location of the camera without directly viewing the sky (bottom).

3.3.1 Estimating Orientation Using an Approximate Analytical Model

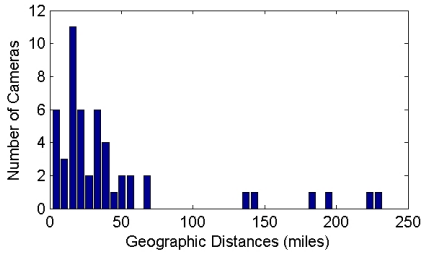
We choose the simple analytical model of sky luminance [60]) that has been adopted as an International Commission on Illumination (CIE) standard [8]. The model predicts the luminance L_v of a sky position on a clear day:

$$L_v = L_z \frac{(0.91 + 10e^{-3\gamma} + 0.45\,cos^2\,\gamma)G(\theta)}{(0.91 + 10e^{-3\theta_s} + 0.45\,cos^2\,\theta_s)G(0)} \tag{3.1}$$

where $G(\theta) = 1 - e^{-0.32/cos\theta}$. The parameters specify the angular positions of the sun and sky relative to the camera and are illustrated in Figure 3.6. The sun position is specified by the azimuth θ_s (degrees eastward from north) and zenith angles ϕ_s (angle



(a) using visible-satellite images (Pennsylvania area)



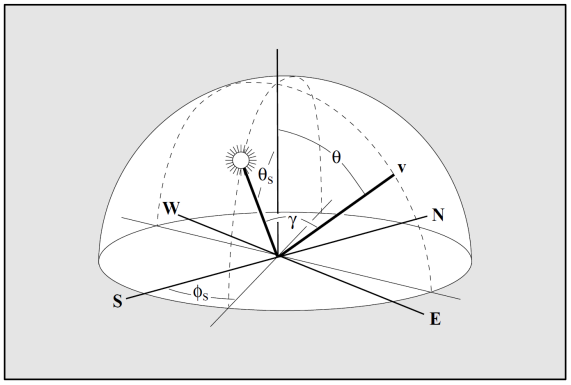
(b) using visible-satellite images (Maryland area)

Figure 3.5: The distribution of errors in geo-location estimates obtained using visible-satellite imagery for a sets of cameras in the Pennsylvania and Maryland areas.

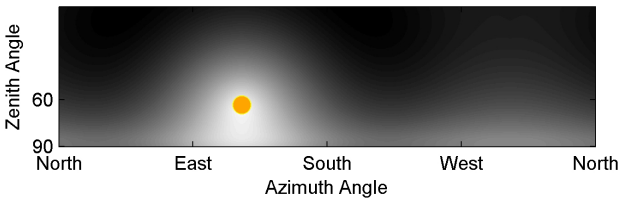
from vertical). The sky direction is specified by the zenith angle θ and the angular distance γ from the sun direction.

Given a time-stamped image, we generate a synthetic sky-luminance map for comparison. The synthetic sky-luminance map, an array where every element corresponds to the luminance L_v of a sky direction, is generated as follows. We first compute the direction of the sun [62] using the camera location and an image time stamp. Next, given the sun direction, we generate a sky-luminance map (azimuth range $[0, 360)$, sampled every 0.3 degrees, and zenith range $[60, 90]$, sampled every 0.3 degrees) using Equation 3.1.

To simplify the calibration problem, we make several assumptions about the camera. We assume that the camera has square pixels, no radial distortion, and no roll, and the principle point occurs in the middle of the image. This reduces the calibration problem to finding the optimal pan, tilt, and focal length parameters.



(a)



(b) Synthetic Luminance Map

Figure 3.6: (a) An illustration of the parameters of the analytical sky-luminance model described in Section 3.3 (Figure from [60]). (b) An example of a synthetic sky-luminance map generated by the analytical model with the sun (orange circle) added for clarity.

For each hypothesized focal length, we re-sample the webcam images to a size such that each pixel corresponds to a subtended angle of 0.3 degrees (the same as the generated sky-luminance image). We then create a correlation map using normalized cross-correlation (NCC) to score every possible potential location of the image on the sky map. This gives a score between -1 and 1 of how similar the sky pixels from the image are compared to a given sky position. We compute this score for images at several different times of day and average the results. We use the azimuth angle of the sky-luminance map element with the maximum NCC score (averaged over many images) as the estimate of the camera orientation.

3.3.2 Experimental Evaluation

To evaluate this method, we estimated the orientation of three cameras from the AMOS data set (specifically, AMOS cameras 4, 330, and 652). The results are within 5 degrees in azimuth angles from manually estimated estimates based on explicit reasoning about the sun position and matching image features to satellite imagery.

Although automatic techniques exist for determining whether images from static cameras have clear skies [28], we manually chose a single day with mostly clear skies. For each camera, approximately 30 images are compared to a corresponding synthetic sky map. Figure 3.7 shows the average NCC scores computed over all azimuth and zenith values. Because the sun provides such a strong cue whether or not its explicit position is calculated, we compare results for camera 4 (top), and camera 4 exclusive of images that directly image the sun (Figure 3.7, second from the top). Although in both cases the method correctly estimates that the camera is westward facing the eastward direction, perhaps surprisingly, also has a high score. This happens because NCC normalizes both images and ignores the absolute intensity, an implication that, ignoring intensity, east and west must appear similar. As can be seen in Figure 3.6 when the sun is near the horizon in the morning, the opposite direction along the horizon has a similarly shaped bright region. Unless the sky pixels from the webcam encompass both of these regions, NCC will give both a similar score. This limitation in NCC motivates the exploration of alternative methods of using this basic cue for camera calibration.

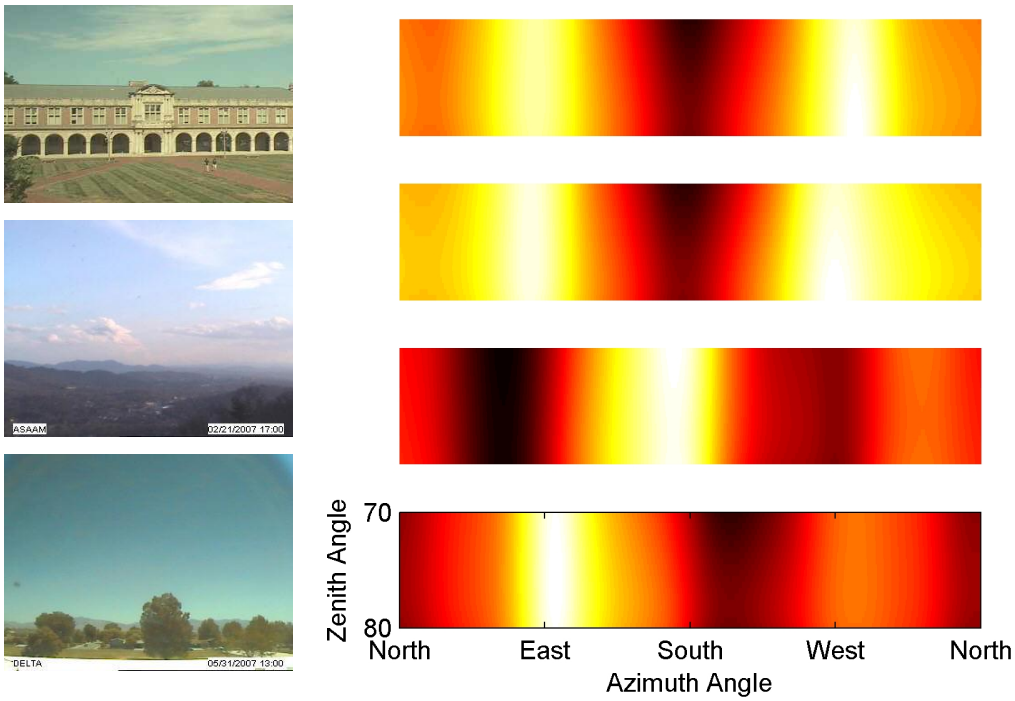


Figure 3.7: (left) Images of three scenes used to evaluate our orientation estimation algorithm. (right) False-color images show the scores obtained by our camera orientation estimation procedure. The color reflects how likely the camera is pointing in a given direction (white means more likely). The top two images are for the same eastward facing camera (top-left). The top was estimated using images with the sun in the field of view and the bottom without those images. The second from the bottom is a southward facing camera that never directly views the sun (middle-left). The bottom image sees a large portion of the sky and directly views the sun (bottom-left).

In our initial work, we hypothesized that stronger constraints were available for more complete camera calibration [29]. In work that followed our initial publication [39], our hypothesis was shown to be correct. The authors propose a method that uses a more direct optimization formulation but similarly uses an analytical model of the sky to solve for the zenith angle, azimuth angle, and focal length of a camera. Results using this method on three AMOS cameras show a maximum error for focal length of 3.3%, zenith angle of 3.5 °, and azimuth angle of 2.6 °.

Chapter 4

Using Cloud Shadows to Infer Scene Structure and Camera Calibration

> *Everything is related to everything else, but near things are more related than distant things.*
>
> Waldo Tobler

We explore the use of clouds as a form of structured lighting to capture the 3D structure of outdoor scenes observed over time from a static camera. We derive two cues that relate 3D distances to changes in pixel intensity due to clouds shadows. The first cue is primarily spatial, works with low frame-rate time lapses and supports estimating focal length and scene structure, up to a scale ambiguity. The second cue depends on cloud motion and has a more complex, but still linear, ambiguity. We describe a method that uses the spatial cue to estimate a depth map (one example of this is shown in Figure 4.1) and a method that combines both cues. Results on time lapses of several outdoor scenes show that these cues enable estimating scene geometry and camera focal length.

Although clouds are among the dominant features of outdoor scenes, they are usually treated as noise by visual inference algorithms. However, the shadows they cast on the ground over time give novel cues for inferring 3D scene models. The basic insight is that there is a relationship between the time series of intensity at two pixels and

the distance between the imaged scene points. We describe two cues, one spatial and one temporal, that further refine this relationship. We also present algorithms that use these cues to estimate a depth map.

The first cue is purely spatial; it ignores the temporal ordering of the imagery and does not require a consistent wind velocity. We begin by considering that if the relationship between pixel time-series correlation and 3D distance is known, then there is a simple problem: Given an image and the 3D distance between every pair of scene points, find the 3D model of the scene that is consistent with the camera geometry and the distance constraints. However, the relationship between correlation and distance is unknown and depends on the scene and the type of clouds in the scene. We present a method that simultaneously solves for the relationship between distance and correlation and for a corresponding 3D scene model.

The second cue requires higher frame-rate video and the ability to estimate the temporal offset between a pair of pixel-intensity time series. This temporal delay, coupled with knowledge of the wind velocity, allows us to define a set of linear constraints on the scene geometry. With these constraints, there is a very clean geometric problem: Given an image and the distance between every pair of pixels projected onto the wind direction, solve for a 3D scene structure that is consistent with the projected distances and the camera geometry.

Our work falls into the broad body of work that aims to use natural variations as calibration cues; each of these methods makes certain assumptions. For example, we require weather conditions in which it is possible to, mostly, isolate the intensity variations due to clouds from other sources of change. The methods we describe are a valuable addition to the emerging toolbox of automated outdoor-camera calibration techniques.

4.1 Related Work

Stochastic Models of Cloud Shapes The structure of clouds has been investigated both as an example of natural images that follow the power law and within

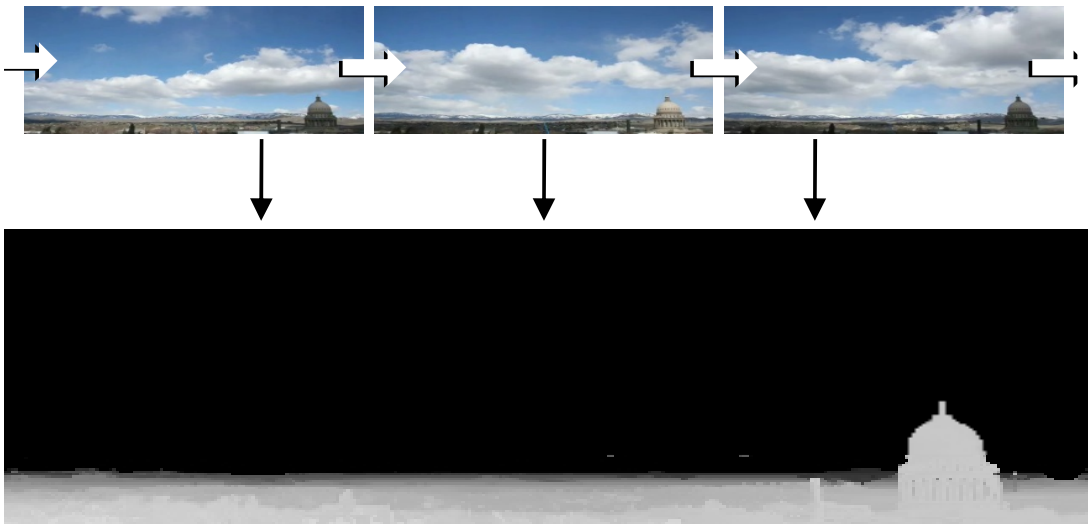
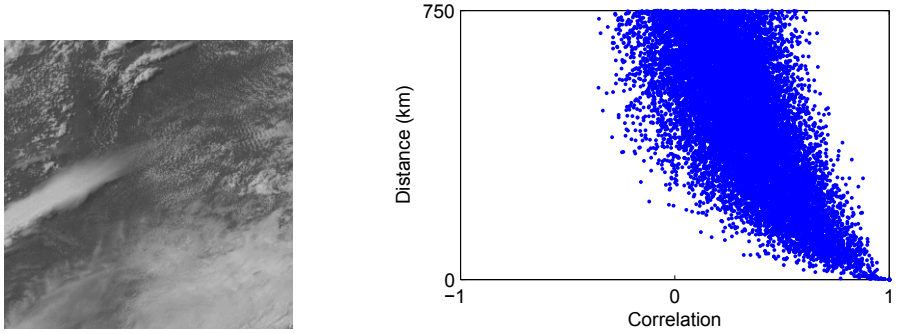


Figure 4.1: Clouds and cloud shadows are significant sources of variation in video of outdoor scenes. In this work, we explore using cloud shadows as a cue to infer scene structure and camera geometry. The depth map was created using the methods described in Section 4.3.1

the atmospheric sciences community. Natural images of clouds often exhibit structure where the expected correlation between two pixels is a function of the inverse of their distance [6], and where, furthermore, there is a scale invariance that may be characterized by a power law (with the ensemble spatial frequency amplitude spectra ranging from $f^{-0.9}$ to f^{-2} [3]). These trends have been validated for cloud cover patterns, with empirical studies demonstrating that the 2D auto-correlation is typically isotropic [71], but that the relationship of spatial-correlation to distance varies for different types of clouds (e.g. cumulus vs. cirrus clouds) [75]. This motivates our use of a non-parametric representation of the correlation-to-distance function.

Shadows in Video Surveillance For video surveillance applications, clouds are considered an unwanted source of image appearance variation. Background models explicitly designed to capture variation due to clouds include the classical adaptive mixture model [69] and subspace methods [45]. Farther removed from our application, object detection/recognition is disturbed by cast shadows because they can change the apparent shape and cause nearby objects to be merged. Several algorithms seek



(a) (b)

Figure 4.2: The correlation-to-distance relationship between sample points has a similar form at many scales. Here we show long-range correlations due to clouds in a sequence of satellite images. (a) A single 1 km-scale image from the visible-light band of the GOES-12 satellite. (b) The relationship between correlation and distance for a set of 90 satellite images captured during the summer months of 2008. Notice that the expected value of distance is a monotonically decreasing function of correlation and that the variance in the conditional distribution is much lower at closer distances.

to minimize these effects, using a variety of approaches [59], including separating brightness and color changes [23].

Geometry and Location Estimation Using Natural Variations Within the field of remote sensing, shadows have long been used to estimate the height of ground structures from aerial or satellite imagery [11]. Recent work in analysis of time-lapse video from a fixed location have used changing lighting directions to cluster points with similar surface normals [35]. Other work has used known changes in the sun illumination direction to extract surface normal of scene patches [73], define constraints on radiometric camera calibration [33, 72], and estimate camera geo-location [72]. Work on the AMOS dataset of time-lapse imagery demonstrates consistent diurnal variations across most outdoor cameras and simple methods for automated classification of images as *cloudy* or *sunny* [28]. This supports methods that estimate the geo-location of a camera, either by finding the maximally correlated location (through time) in a satellite view, or interpolating with respect to a set of cameras with known positions [30]. The recently created database of "webcam clip-art" includes camera calibration parameters to facilitate applications to illumination and appearance transfer across scenes [38].

4.2 Structural Cues Created by Cloud Shadows

The image of cloud shadows passing through a scene depends upon the camera and scene geometry. Here we describe two properties of outdoor-scene time lapses that depend on cloud shadows, are easy to measure, and, as we show in Section 4.3, can be used to infer camera and scene geometry.

4.2.1 Geographic Location Similarity

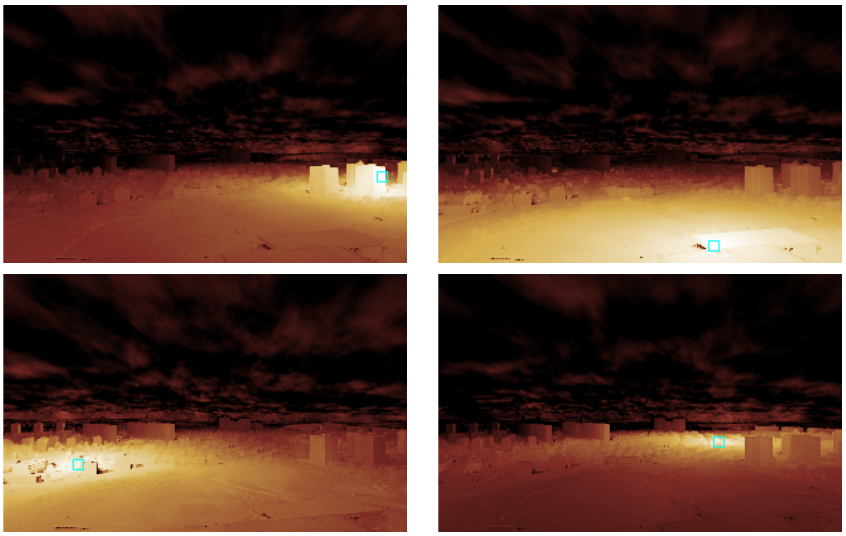
The closer two points are in the world to one another the more likely they are to be covered by the shadow of the same cloud. Thus, for a static outdoor camera, the time series of pixel intensities are usually more similar for scene points that are close than for those that are far apart.

We begin by considering the correlations that arise between pixels in satellite imagery. The statistical properties of this approximately orthographic view are similar to the spatial properties of the cloud shadows cast onto the ground. We empirically show the relationship between correlation and distance for a small dataset of visible-light satellite images (all captured at noon on different days during the summer of 2008). The scatter plot in Figure 4.2, in which each point represents a pair of pixels, shows that the correlation of the pixel intensities is clearly related to the distance between the pixels. Additionally, it shows that the expected value of distance is a monotonically decreasing function of correlation.

This relationship also holds at a much finer scale. To show this, we compute correlation between pixels in a time-lapse video captured by a static outdoor camera on a partly cloud day. Since we do not know the actual 3D distances between points, we cannot generate a scatter plot as in the satellite example. Instead, Figure 4.3 shows examples of correlation maps generated by selecting one landmark pixel and comparing it to all others. The false-color images, colored by the correlation between a pair of pixels, clearly show that correlation is related to distance.

We note that different similarity measures between pairs of pixels could be used (and, in some cases, would likely work much better). We choose correlation because

(a)



(b) Landmark-Pixel Correlation Maps

Figure 4.3: (a) A frame from a time-lapse video of an outdoor scene captured on a partly cloudy day. (b) False-color images colored by the correlation of the time series of the highlighted landmark pixel with all other pixels in the image.

it is simple to compute online and works well in many scenes. Our work does not preclude the use of more sophisticated similarity metrics that explicitly reason about the presence of shadows using, for example, color cues. In Section 4.3, we show how to infer the focal length of the camera and a distance map of the scene using correlation maps as input.

4.2.2 Temporal Delay Due to Cloud Motion

As clouds pass over a scene, each scene point exhibits a sequence of light and shadow. In the direction of the wind, these time series are very similar but are temporally

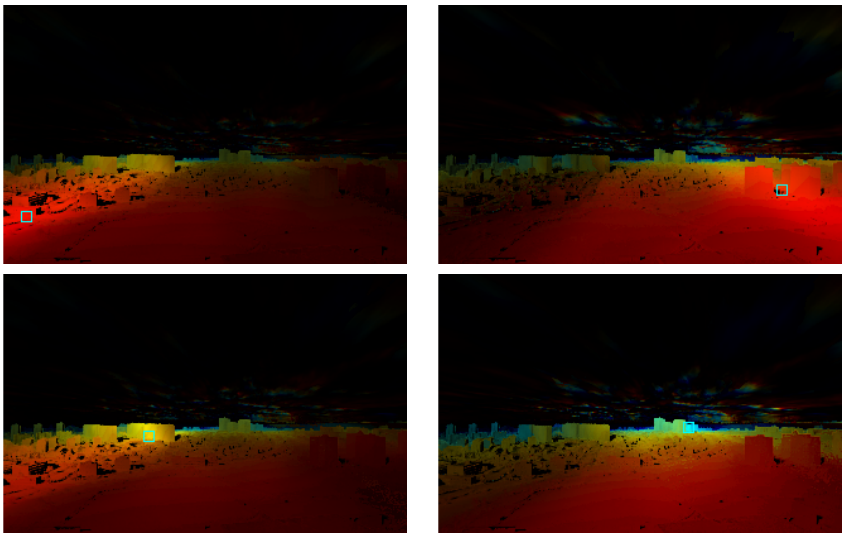


Figure 4.4: False-color images with colors based on the temporal delay between a landmark pixel and all other pixels in the scene (using the scene from Figure 4.3). The hue of each pixel is determined by the delay; the value is determined by the confidence in the delay (low intensity regions are low confidence).

offset relative to the geographic distance between the points (see Figure 4.5). Also, for short distances perpendicular to the wind direction, we expect to see zero temporal delay. As in the previous cue, we expect correlation, after accounting for delay, to decrease with distance due to changing cloud shapes or, different clouds altogether if we move far enough perpendicular to the wind direction.

Our method for estimating the temporal offset between the time series of a pair of pixels consists of two phases. First we use cross-correlation to select the integral offset that gives the maximum correlation. Then we obtain a final estimate by finding the maxima of a quadratic model fit to the correlation values around the integer offset. We use the correlation of the temporally aligned signals as a confidence measure (e.g., low correlation means low confidence in the temporal offset estimate).

Figure 4.4 shows examples of false-color images constructed by combining the estimated delay and the temporally aligned correlation for every pixel, relative to a single landmark pixel. The motion of the clouds in this scene is nearly parallel with the optical axis, so the temporal delays are roughly equal horizontally across the image (i.e., perpendicular to the wind direction), but the correlations quickly decrease as distance from the pixel increases (i.e., different clouds are passing over those points).

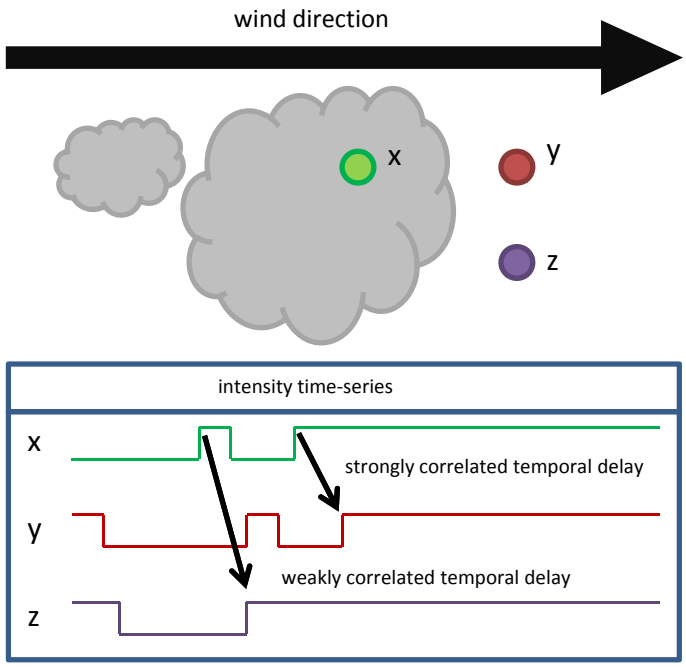


Figure 4.5: The time-series of variation in light intensity induced by cloud shadows is dependent on the direction of motion of the wind, the shape of the clouds, and the geographic position of the scene points.

Orthogonally, the correlations are relatively higher in the direction of the wind, but the delay changes rapidly.

4.3 Using Clouds to Infer Scene Structure

The dependence of correlation upon distance and the temporal delay induced by cloud shadow motion are both strong cues to the geometric structure of outdoor scenes. In this section, we describe several methods that use these cues to infer a depth map and simplified camera geometry.

We assume a simplified pinhole camera model. Assuming a focal length, f, a point, $R_i = (X, Y, Z)$, in the world projects to an image location, expressed in normalized homogeneous coordinates as $r_i = (\frac{Xf}{Z}, \frac{Yf}{Z}, 1)$. For each pixel, i, the imaged 3D point, R_i, can be expressed as $R_i = \alpha_i r_i$ with depth, α_i. We define the 3D distance between two points as $d_{ij} = ||R_i - R_j||$. Note that the use of 3D distances is not technically correct; it should take into account the location of the sun. Consider, for example,

that any two scene points in-line with the sun vector see the same cloud shadows and will therefore have similar time series. In our experiments, we handle this by modifying d_{ij} by projecting the points, along the sun direction vector, to the ground plane prior to computing the distance; if the sun vector is unknown, we project points straight down. This gives distances that are meaningful with respect to time-series similarities induced by cloud shadows. Note that this creates a point ambiguity where the depth of a pixel ray that is parallel to the sun vector is unconstrained.

4.3.1 Estimating Scene Structure Using Pairwise Correlation

In outdoor scenes, there is a strong relationship between correlation, ρ_{ij}, and 3D distance, d_{ij}, between the imaged scene points. In this section, we show how to estimate a depth map, $\mathbf{a} = \alpha_1, \ldots, \alpha_n$, for an outdoor scene using this relationship. The challenge with estimating the scene structure given the pairwise correlations is the unknown conditional relationship between correlation and distance between scene points, $E(d_{ij}|\rho_{ij})$. In other words, we do not know what the distance between a pair of points should be for a given value of correlation; this mapping depends on, among other factors, the type of clouds passing overhead.

We assume that the geographic correlation function (GCF) with respect to a single scene point is geographically isotropic (i.e., if you could view the correlation map of the scene from zenith, the iso-contours would be circular and the expected value of correlation would monotonically decrease with distance from the landmark pixel). This implies that for the correct scene geometry, the distribution of correlation at a given distance is relatively low variance. We use this to define an error function for evaluating possible depth maps. More formally, for a good depth map, $\mathbf{a}$, we expect that the following value to be small:

$$V(\mathbf{a}) = \int \mathrm{Var}(d(\mathbf{a})|\rho) d\rho. \tag{4.1}$$

In a real scene, if the clouds have an isotropic GCF, then the shadows cast by the clouds will likely have an anisotropic GCF unless the sun is directly overhead. Consider, for example, the elliptical shadow cast by a sphere onto the ground plane. In

this work, we ignore this effect and expect the shadows to have an isotropic GCF regardless of the sun position. This is equivalent to modeling the cloud layer as having height zero.

Overview

We use Non-Metric Multidimensional Scaling (NMDS) [36, 37] to simultaneously solve for $E(d_{ij}|\rho_{ij})$ and the depth map, $\mathbf{a}$. Like classical Multidimensional Scaling (MDS), NMDS solves for point locations given pairwise relationships between points. Unlike MDS, NMDS does not expect the input relationships to correspond to distances; instead, the input is only required to have a monotonic relationship to distance. Since we assume that distance is a monotonically decreasing function of correlation, we can use the NMDS framework to solve for this mapping.

In our application NMDS works, from a high-level, as follows. First we initialize a planar depth map (see Section 4.3.1). Then we iterate through the following steps:

- determine d_{ij} for the current depth map,
- estimate the mapping from distance to correlation, $E(d_{ij}|\rho_{ij})$ (see Section 4.3.1),
- use the pairwise correlation, ρ_{ij}, and $E(d_{ij}|\rho_{ij})$ to compute a pairwise distance estimate,
- update the depth map to better fit the estimated distances (see Section 4.3.1).

We now describe the three components of this procedure in greater detail.

Initialization

Here we describe a method for initializing a depth map that makes the assumption that the scene is planar. We solve for the camera focal length, f, and external orientation parameters, θ_x and θ_z, that minimize the variance of the correlation-to-distance mapping. More formally, we choose parameters that minimize Equation 4.1,

$$\min_{f,\theta_x,\theta_z} V(f, \theta_x, \theta_z). \tag{4.2}$$

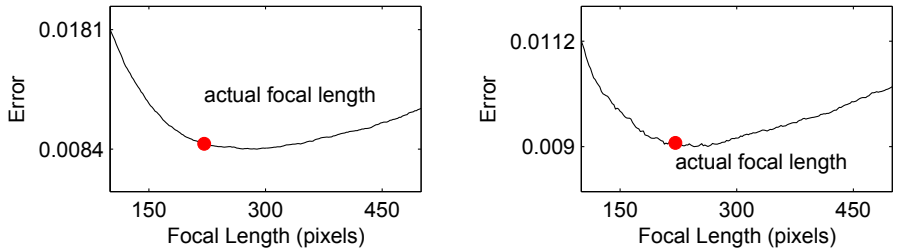


Figure 4.6: The error in different focal length values in our ground-plane-based initialization method for two scenes. The red points correspond to the value of focal length provided by the camera in the image EXIF tags.

Note that in this case the 3D distances, d_{ij}, are a function of the three parameters, which together with a ground plane assumption imply a depth map. We exhaustively search over a reasonable range of parameters and choose the setting that minimizes the objective function.

Figure 4.6 shows the value of the error function defined above with respect to focal length for two scenes. We find that the estimated focal length is close to the ground truth value in both cases. We use the planar depth map to provide an initial estimate for the mapping from correlation to distance (the distance from the camera to the ground plane along each pixel ray). See Figure 4.8 for two examples of initial depth maps discovered using this method. This initial depth map is used to initialize the correlation-to-distance mapping, $E(d_{ij}|\rho_{ij})$.

Estimating Pairwise Distance Given Correlation

This section describes our model of the monotonic mapping from correlation to distance, $E(d_{ij}|\rho_{ij})$. Many simple parametric models could be used to fill this requirement but they impose restrictions on the mapping, which can lead to substantial artifacts in the depth map. Instead, we choose a non-parametric model that makes the following minimal assumptions on the form of the mapping:

- $\mathrm{E}(d_{ij}|\rho_{ij} = 1) = 0$, when the correlation is one the expected distance is zero,
- $\mathrm{E}(d_{ij}|\rho) \geq E(d_{ij}|\rho+\epsilon)$, expected distance is a monotonically decreasing function of correlation

These assumptions follow naturally from empirical studies on the spatial statistics of real clouds [71]. While these statistics are not present in all time-lapse videos, we leave for future work the task of determining which videos have the appropriate statistics.

We use the non-parametric regression method known as monotonic regression [36], to solve for a piecewise linear mapping from correlation to distance while respecting the constraints described above. The first step is choosing an optimal set of expected pairwise distances, $\hat{\mathbf{d}}$, for a fixed set control points uniformly sampled along the correlation axis (we use 100 control points). We then use a linear program to solve for values of $\hat{\mathbf{d}}$ that minimize $\sum \left| \hat{\mathbf{d}}_{\mathrm{Bin}(\rho_{ij})} - d_{ij} \right|$ relative to the distances, d_{ij}, implied by current scene model (initially a plane). Given the control point locations and corresponding optimal distance values, we use linear interpolation to estimate the expected value of distance for a given correlation.

Examples of the correlation-to-distance mapping, $E(d_{ij}|\rho_{ij})$, are shown in Figure 4.8. Note that the expected values are reasonable when compared to the sample points and that they would be difficult to model with a single, well-justified parametric model. We use this regression model to define the expected distance between a pair of points, and we use this expected distance as input into the depth map improvement step described in the following section.

Translating Pairwise Distances Into Depths

We use $E(d_{ij}|\rho_{ij})$, defined in the previous section, to estimate a distance matrix. We pass this distance matrix as input to a nonlinear optimization-based MDS [37] procedure to translate estimated distances into 3D point locations. We augment MDS to respect the constraint that the 3D point locations must lie along rays defined by the camera geometry. We fix the focal length to the value estimated in the initialization step.

The error (*stress*) function for MDS is as follows:

$$S(\mathbf{a}) = \frac{1}{2} \sum_{i,j} w_{ij} (d_{ij} - E(d|\rho_{ij}))^2 \tag{4.3}$$

where the weights, w_{ij}, are an increasing function of the correlation, ρ_{ij}. In other words, we expect the distance estimates from high-correlation pairs to be more accurate than those of lower-correlation pairs. In this work, we use $w_{ij} = \rho_{ij}^2$ for $0 \leq \rho_{ij}$ and $w_{ij} = 0$ for $\rho_{ij} < 0$. Recall that the 3D distance, d_{ij}, between imaged scene points is a function of the depths, $\mathbf{a}$, along pixel rays. We minimize the *stress* function with respect to the depths using the trust region method, constrained so that $\mathbf{a} \geq 0$. We perform several descent iterations for a given distance matrix before re-estimating the correlation-to-distance mapping, $E(d_{ij}|\rho_{ij})$, using the updated point locations. Additionally, we constrain the average of the estimated pairwise distances to remain constant to avoid the trivial, zero-depth solution.

We use a straightforward application of the chain rule to compute the gradient and to form a diagonal approximation of the Hessian; we show them here for completeness. We first define the error in an individual pair of distances as $e_{ij} = d_{ij} - E(d|\rho_{ij})$. We also note that we can describe the pairwise distances, in terms of inner products between pixel rays, as $d_{ij}^2 = \alpha_i^2 r_{ii} + \alpha_j^2 r_{jj} - 2\alpha_i \alpha_j r_{ij}$, where $r_{ij} = r_i{}^{\mathsf{T}} r_j$ is an inner product of pixel rays. This allows us to write the gradient,

$$\frac{\partial S}{\partial \alpha_i} = \sum_j \frac{w_{ij} e_{ij} (\alpha_i r_{ii} - \alpha_j r_{ij})}{d_{ij}}, \tag{4.4}$$

and the Hessian,

$$\frac{\partial^2 S}{\partial \alpha_i^2} = \sum_j w_{ij} \left\{ \frac{e_{ij} r_{ii}}{d_{ij}} + \frac{(\alpha_i r_{ii} - \alpha_j r_{ij})^2}{d_{ij}^2} - \frac{e_{ij} (\alpha_i r_{ii} - \alpha_j r_{ij})^2}{d_{ij}^3} \right\}. \tag{4.5}$$

Ideally, we would use all pairs of pixels when minimizing the *stress* function. We find that using a much smaller number yields excellent results and is substantially less resource-intensive (we typically use around 100 randomly selected landmark pixels for a 320×240 image). In our Matlab implementation, the complete depth estimation procedure, including the ground-plane based initialization, typically requires several minutes to complete.

This algorithm is essentially a projectively constrained variant of the NMDS [36] algorithm. It is well known that NMDS is subject to local minima, which can lead to suboptimal depth maps. This has not been a significant problem for depth estimation,

but understanding this is an interesting area for future work. The majority of errors we see in the final depth maps are caused by erroneously high correlations for distant pixel pairs. Frequent causes of this problem are insufficient imagery for estimating the correlation, large sun motions, which cause higher correlations between surfaces with similar normals, and automatic camera exposure correction, which causes shadowed pixels to be highly correlated across the image.

4.3.2 Estimating Scene Structure Using Temporal Delay in Cloudiness Signal

The motion of clouds due to wind causes nearby pixels to have similar but temporally offset intensity time series. Together, these temporal offsets, $\Delta_{t(i,j)}$, give constraints on scene geometry. Section 4.2.2 shows examples of these temporal offsets.

Let W be a 3D wind vector that we assume it is fixed for the duration of the video. A pair of points in the world, R_i, R_j, that are in-line with the wind satisfy the linear constraint $R_i - R_j = W\Delta_{t(i,j)}$, where $\Delta t(i,j)$ is the time is takes for the wind (and therefore the clouds) to travel from point R_j to point R_i. However, the algorithm in Section 4.2.2 can often compute the temporal offset between pixels not exactly in-line with the wind. We generalize the constraint to account for this by projecting the displacement of the 3D points onto the wind direction, $\hat{W} = W/||W||$:

$$\hat{W}^\top(R_i - R_j) = \hat{W}^\top W\Delta_{t(i,j)}. \tag{4.6}$$

Based on the simplified camera imaging model, each pixel corresponds to a known direction, so the 3D point position, R_i, can be written as a depth, α_i, along the ray, r_i. Explicitly showing this constraint in terms of the unknown depths we find:

$$\hat{W}^\top(\alpha_i r_i - \alpha_j r_j) = \hat{W}^\top W\Delta_{t(i,j)}, \tag{4.7}$$

$$\alpha_i\hat{W}^\top r_i - \alpha_j\hat{W}^\top r_j = \hat{W}^\top W\Delta_{t(i,j)} \tag{4.8}$$

This set of constraint defines a linear system,

$$\mathbf{Ma} = \mathbf{\Delta}, \tag{4.9}$$

where $\mathbf{a}$ is a vector of the (unknown) depth values, α_i, for each pixel, the rows of $\mathbf{M}$ contain two non-zero entries of the form $(\hat{W}^\top r_i, -\hat{W}^\top r_j)$, and $\mathbf{\Delta}$ contains the scaled temporal delays between pixels.

The constraint on depth due to temporal delay has an ambiguity. In all cases, the matrix $\mathbf{M}$ has a null space of dimension at least one. This is visible from the structure of $\mathbf{M}$: adding any multiple of $\alpha' = (\frac{1}{\hat{W}^\top r_1}, \frac{1}{\hat{W}^\top r_2}, \ldots)$ to the depth map, $\mathbf{a}$, does not change the left-hand side of Equation 4.8. The next section describes how we overcome this ambiguity.

4.3.3 Combining Temporal Delay and Spatial Correlation

The two cues we describe have ambiguities—the scale ambiguity for the spatial cue and the null space ambiguity for the temporal cue—that prevent metric interpretation of the generated depth maps. Combining the two cues allows us to simultaneously remove both ambiguities and makes possible future work on metric scene estimation. We propose the following simple method.

Starting with the constraints defined by the temporal cue, we solve for a feasible depth map, $\mathbf{a}$, using a standard non-negative least squares solver. We then consider the set of solutions of the form $\mathbf{a} + k\alpha'$ and search over values of k to find a *good* depth map. While many criteria exist for evaluating a depth map, we focus on combining the two cues we have described to remove this ambiguity. As with the spatial cue, we make the assumption that correlation is geographically isotropic. This motivates us to use the error function defined in Equation 4.2 to evaluate the different depth maps. The only difference is that we now search over the null space as opposed to the focal length and orientation parameters. In Section 4.4.2, we show results that demonstrate that depth maps with low error function values are more plausible than those with error function values.

4.4 Evaluation

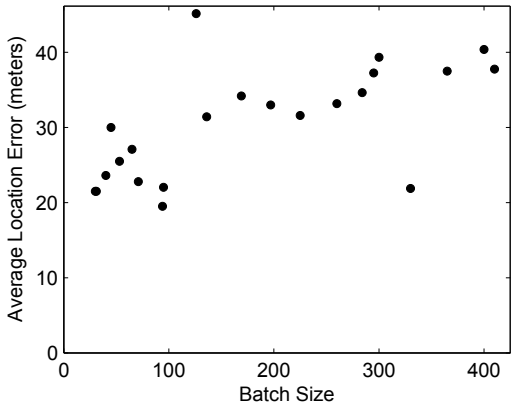
We demonstrate depth estimation on several outdoor scenes. In all examples, we resize the original images to be 320 pixels wide and assume that the sky has been manually masked off. In some cases, shadow regions are masked using automatic filtering methods based on thresholding the variance of the individual pixel time series.

4.4.1 Depth from Correlation

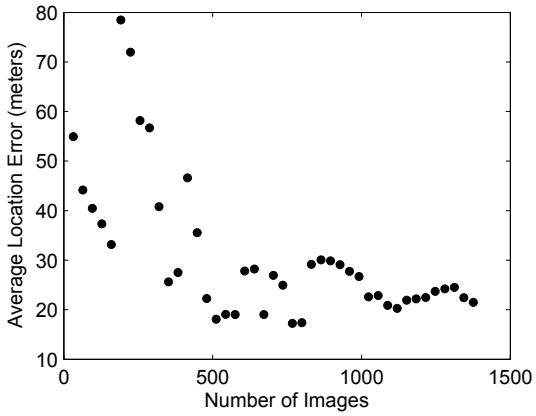
We show depth maps generated using the method described in Section 4.3.1. As input, we provide correlations between 100 randomly selected pixels and all other pixels in the scene; in both cases we omit sky pixels. Examples of these correlation maps can be seen in Figure 4.3.

We first evaluate our methods on a time lapse captured over three hours with a frame rate of one image every five seconds. Naïvely computing correlation on the entire video sequence yields a low-quality correlation map due to long-term and spatially broad changes caused by the sun motion and melting snow on fields in the near ground. Computing correlations over short temporal windows and then averaging these correlations removes these artifacts. Figure 4.8 shows the depth map estimated from this scene and the correlation-to-distance mapping we estimate as part of the optimization.

Quantitative evaluation of the errors in this scene find an average per-pixel error of 20 meters in the estimates of the 3D point locations (relative to hand-clicked corresponding points). This represents 2% error, relative to the scale of the scene, in the position estimates. Figure 4.7 shows how this error varies as we change the way we process the image data. In the first experiment, we fix the number of images but vary the window size. We find that as window size increases, the error increases; this is due to longer-term changes such as the sun motion being a dominant cause of correlation. These longer-term changes invalidate our assumption of a geographically isotropic correlation function. In the second experiment, we fix the window size (at the window size that gave minimum error in the previous experiment) and vary the



(a)

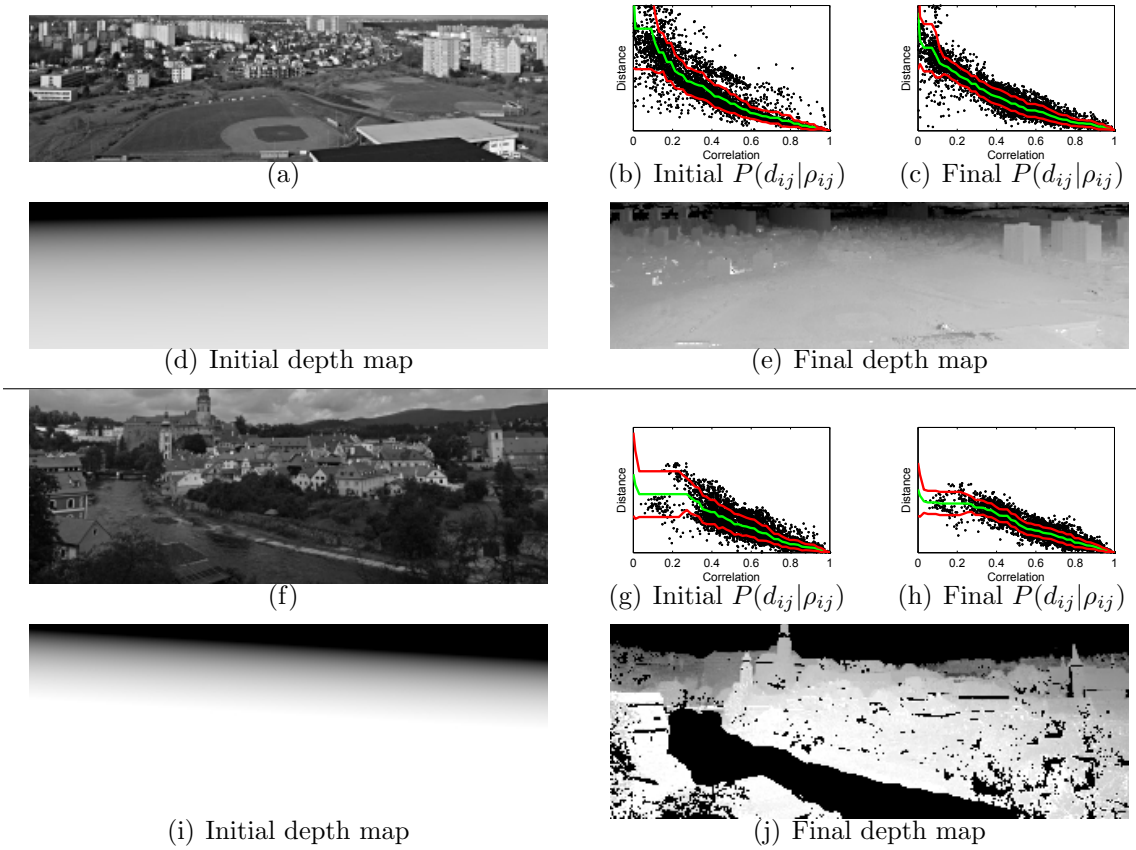


(b)

Figure 4.7: Quantitative evaluation of the scene maps generated using NMDS with the spatial cue. The results show (a) that increasing the window size used for computing correlations increases the error and (b) using more images in estimating the correlations gives lower error.

number of images used to compute correlation. We find that, as expected, using more windows, and hence more images, results in position estimates with lower error.

The second time lapse consists of 600 images captured over 50 minutes. Figure 4.8 shows the depth map estimated from this scene and the correlation-to-distance mapping we estimate as part of the optimization. The river and sky were manually masked while the shadow regions were automatically masked by removing pixels with low-variance time series.



(a) (b) Initial $P(d_{ij}|\rho_{ij})$ (c) Final $P(d_{ij}|\rho_{ij})$

(d) Initial depth map (e) Final depth map

(f) (g) Initial $P(d_{ij}|\rho_{ij})$ (h) Final $P(d_{ij}|\rho_{ij})$

(i) Initial depth map (j) Final depth map

Figure 4.8: Examples of depth maps estimated using our NMDS-based method using correlations between pairs of pixels. The correlation to distance mappings at the optimal solution are clearly lower variance than those of the initial planar depth map.

A final example of using the spatial cue to estimate a depth map is shown in Figure 4.1. This time lapse demonstrates that NMDS is able to recover from significant errors in the initial depth map; for example, the initial depth estimate of the rotunda was incorrect by several kilometers.

We emphasize that in these examples we perform no post-processing to improve the appearance of the generated depth maps. The optimization is based solely on geometric constraints on the camera geometry and the expectation that the correlation-to-distance mapping is geographically isotropic.

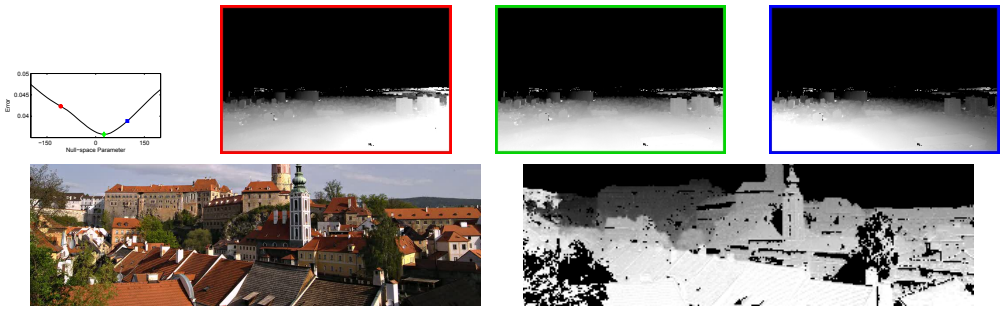


Figure 4.9: (top) A plot of the error function for differing depth maps created using the procedure described in Section 4.3.3. The plot highlights the smooth nature of this objective function for depth maps generated for different values of the null space parameter. The depth map generated at the optimal null space parameter is significantly more plausible than the others. (bottom) A cropped frame and a corresponding depth map generated for another scene using the same procedure.

4.4.2 Depth from Combining Temporal Delay and Spatial Correlation

Figure 4.9 shows the depth map generated by the method described in Section 4.3.3. Note that to reduce memory usage, we discard constraints for pixel pairs, ij, whose temporally aligned correlation is less than a threshold (we use threshold 0.85). The top row of the figure show results on a previously described scene. This result demonstrates that higher values of the error function lead to lower quality depth maps. For the second scene 200 frames of a time lapse (captured one frame every five seconds) were used to estimate a delay map. This delay map is translated into a depth map using the combined inference procedure.

4.5 Conclusion

We presented two novel cues, both due to cloud shadows, that are useful for estimating scene and camera geometry. The first cue, based on spatial correlation, leads to a natural formulation as a Non-Metric Multidimensional scaling problem. The second cue, based on temporal delay in cloud signals, defines a set of linear constraints on scene depth that may enable metric depth estimates. These cues are unique in that they can work when other methods of inferring scene structure and camera geometry have difficulties. They require no camera motion, no haze or fog, no sun motion, and

no moving people or cars. We also demonstrated how to combine these cues to obtain improved results. This work adds to the growing literature on using natural scene variations to calibrate cameras and extract scene information.

Chapter 5

Using Webcams for Science

Our work on automatically discovering, calibrating, and geo-locating outdoor cameras brings us closer to the goal of using webcams as scientific instruments. This chapter gives several "proof of concept" uses the webcam network presents for measuring environmental properties.

The geo-temporal image formation model that organizes much of our discussion highlights both challenges and opportunities of webcam imaging. Image understanding is challenging because of the highly structured variability in the measured signal, such as variability due to objects in the scene, weather and atmospheric effects, and specular reflections. Our previous chapters have sought to minimize these variations, in order to, for example, simplify inference of camera locations or orientations. However, the fact that images vary due to so many causes is also a strength. With proper analysis from the now-calibrated cameras, it is possible to infer the presence or degree of a wide range of natural phenomena.

A broad range of environmental properties can be seen in webcam images. Some of these possibilities have been noted within the context of repeat photography, as shown in Table 5.1. Here we explore techniques to automatically extract such environmental properties from long sequence of webcam images. This allows the webcams *already* installed across the earth to act as generic sensors to improve our understanding of local weather patterns and variations.

We begin with a simple, scientifically important example for using the webcam network for estimating the timing of the spring onset of leaf growth on deciduous trees. The remainder of the chapter shows linear methods for estimating more complicated

Observed in Landscape Pictures	**Value to Environmental Monitoring**
Plants • Species (Type and Size) • Leaf Cover (Amount and Color) • Flowers and Fruit (Amount) • Timing of Events	• Ground truth/verify analysis of satellite products used in research and operations • Track invasive plants • Monitor plant response to changes in local, regional, global environmental conditions, and determine important field sites • Support local to international phenology networks
Land Surface • Type • Erosion	• Measure erosion rates • Measure snow depth • Measure location of glaciers
Water Levels • Tides • Rivers and Streams • Lakes, Ponds, and Puddles	• Monitor flooding response to rain events • Expand water level monitoring network
Sky • Clouds, Sun Location, Sky Color, and Visibility	• Verify cloud analysis using satellite data • Expand visibility network
Buildings and Development • Houses • Factories • Roads	• Long-term monitoring of land cover, including identifying lawn cover, using satellite imagery

Table 5.1: Some of the natural changes visible in images captured by static cameras (adapted from the Measuring Vegetation Health Project [44])

signals and for inferring satellite views from a large number of webcams. Together these "proof of concept" applications support the view that the global network of outdoor webcams is a promising resource for gathering scientific data about the natural world.

5.1 Related Work

There is a long history of using camera networks to monitor environmental changes and social behaviors. Cameras have been used to measure phenomena including waves on the ocean [20], the onset of spring leaf growth [63], and the relationship between leaf growth and carbon dioxide concentrations [83]. Often these tasks are approached with the motivation of validating satellite remote sensing observations but sometimes the cameras are used to provide novel information. One problem faced in these fields is that they are all working independently to set up their own monitoring stations. There is a great opportunity for sharing installation and maintenance costs that has yet to be taken advantage of.

Notable examples that use large dedicated camera networks include the Argus Imaging System [21] with 30 locations and 120 cameras that explicitly focuses on coastal monitoring. Cameras within the Argus network and similar cameras set up on an ad-hoc basis for individual experiments have been used to quantify density of use of beach space [46], the use of beaches as a function of weather [32], and trends both in beach usage and beach erosion [16]. Another large, dedicated camera network is the Haze Cam Pollution Visibility Camera Network [19]. In this case, the cameras are placed near measurement systems for air pollution and other meteorological data, but the images are primarily used to provide the public a visual awareness of the effects of air pollution. To our knowledge, these cameras have not been systematically used to provide additional quantitative measurements to augment the explicit pollution or meteorological data, but recent work has validated that similar cameras have high correlation with explicit measurements of atmospheric visibility, based both on ground [86], and satellite measurements [85, 84].

Additional work has focused on phenology, the study of the effects of seasonal climate variations on long-term changes in plants and animals. Recent studies have shown

that phenology is a robust indicator of climate change effects on natural systems; for example, earlier budburst and flowering by plants have been documented in response to recent warming trends. Improved monitoring of vegetation phenology is viewed as an important means of documenting biological responses to a changing world [55]. New and inexpensive monitoring technologies are resulting in a dramatic shift in the way that phenological data are now being collected [47]; already, several networks based around color digital imagery (e.g., "PhenoCam" [56] and the "Phenological Eyes Network" [57]) have been established to monitor phenology on a regional scale. Previous studies have provided solid evidence that both qualitative and quantitative information about seasonal changes in the condition and state of vegetation canopies can be extracted from webcam images [65, 64].

Recent work has begun exploring the use of the global network of outdoor webcams for science. Bradley et al. [4] demonstrate that some of the necessary image processing can be performed using a web-based interface. Recent work by Graham et al. [15] demonstrates, similar to our example in the next section, that a large collection of webcams can be used to estimate temporal properties of leaf growth on trees.

5.2 Using Webcams to Estimate Spring Leaf Growth

We consider an application of webcams for science that provides robust tools to quantify the timing and rate of the "spring onset" of leaf growth on trees. In many regions of the world, the timing of spring onset has advanced at between 2 and 5 days per decade over the last 30 years [55], and the length of the growing season is an important factor in controlling primary productivity and hence carbon sequestration. Our analysis here expands on the ongoing efforts of the PhenoCam project [56, 64], in which a smaller number of dedicated, high-resolution (1296×960 pixel) cameras were deployed specifically for this purpose at forest research sites in the northeastern U.S. While there are additional challenges in working with a much larger set of cameras for which the camera settings and internal processing algorithms are unknown, results presented here show that the spring green-up signal is visible in many cameras not dedicated to this monitoring task.

We describe a method for estimating the timing of spring leaf development from webcam images. Importantly, this method does not require a co-located sensor or ground observations of vegetation phenology, and human input is minimal. We use the simple "relative greenness" signal [65] and show that it can be extended to many of the cameras in the AMOS dataset. The relative greenness, $g/(r+g+b)$, is defined as the average of the green color channel divided by the sum of all color channels.

We begin by selecting a set of cameras with a significant number of trees in the field of view. For each camera, we extract a set of images (at most one for each day) captured around noon for the first 275 days of 2008. We manually draw a polygon around the trees; since the cameras are static, or registered post-capture, only one polygon must be drawn. We then compute the average greenness value of the tree region for each image. In order to characterize the timing of spring leaf growth, we fit a 4-parameter sigmoid model [65],

$$g(t) = a + \frac{b}{1 + \exp(c - dt)} \tag{5.1}$$

where t is the day of year, to the greenness signal. Note that c/d corresponds to the day of the year of the vertical midpoint of the model.

5.2.1 Correcting for Automatic Color Balancing

Some cameras in the dataset automatically adjust the color balance to respond to changing illumination conditions (due, for example, to clouds, solar elevation, and aerosols). This causes problems because the colors measured by the camera vary even when the underlying color of the scene does not change. To compensate for this automatic color balancing, we use scene elements such as buildings or street signs, (whose true color we assume to be constant over time) as an ad-hoc color standard. We then solve for the linear color-axis scaling, which maintains the color of the color standard, and apply this scaling to the entire image to create a color-balanced image.

Figure 5.1 shows the raw and color-corrected greenness signals (using the method from Section 5.2.1) and the estimated sigmoidal model for a single camera. In addition, the figure contains a montage of three images for manual inspection. The images in

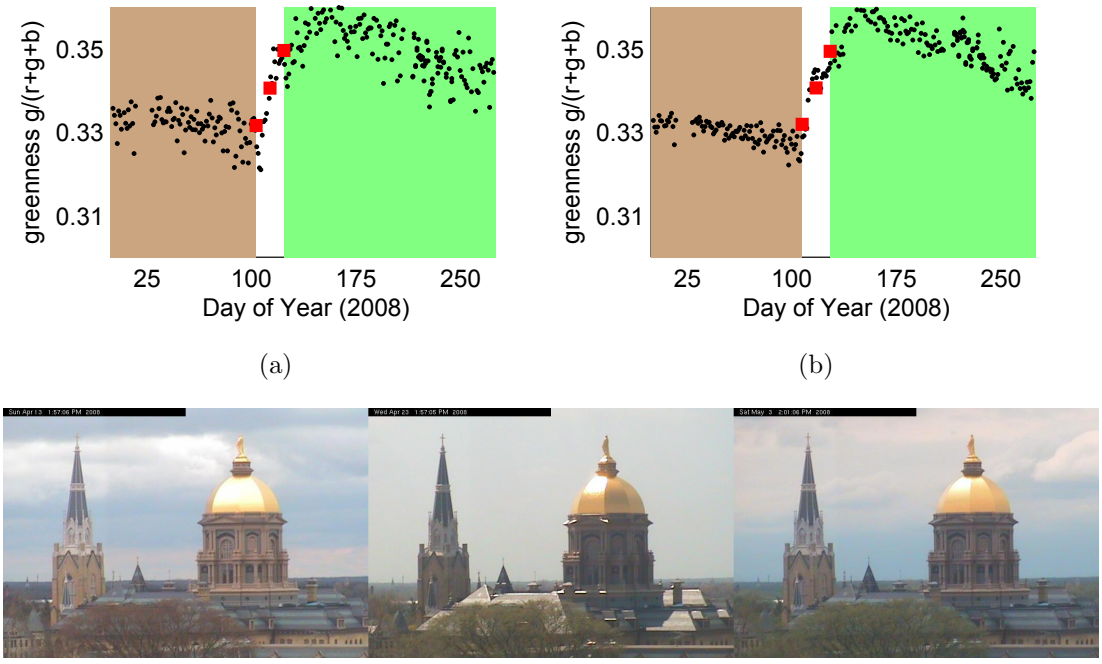


Figure 5.1: Estimating spring leaf growth using an outdoor webcam. (a) The raw value of greenness over time (black dots) and a partitioning of the year based on the presence/absence of leaves. (b) Results for the same set of images after correcting for different lighting conditions and the automatic color balancing algorithm in the camera. We use the methods described in Chapter 5.2.1 with the rotunda as a color standard. The color correction reduces the local variance of the greenness score but, in this case, does not significantly impact the estimated onset of spring leaf growth. (c) Three images, each corresponding to a red square marker in (b), to verify the model fit. In this scene, the trees are in the foreground below the building. This highlights that only a small region of trees is needed to obtain an estimate of spring green-up time.

the montage are selected by first determining the vertical mid-point, $\hat{t}$, of the sigmoid function. The images selected for the montage are the images closest to $\hat{t}-10$ days, $\hat{t}$, and $\hat{t}+10$ days. More results, as well as a color-coded map, are shown in Figure 5.2. The map shows, as expected, a slight linear correlation between latitude and the "spring onset" [22]. The highlighted montages show that the estimated dates are accurate.

5.2.2 Inferring 'Spring Onset' in Multiple Tree Species

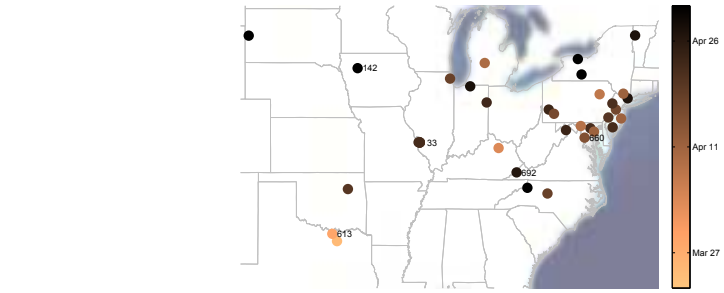
In some webcam images, temporal variations in the average greenness signal are due to the presence of multiple tree species. This same problem occurs in satellite imagery, but unlike satellite imagery, webcams allow us to address the problem. In fact, it is possible to factor the average greenness signal into components due to multiple tree species.

Our approach is to fit a mixture-of-sigmoids model,

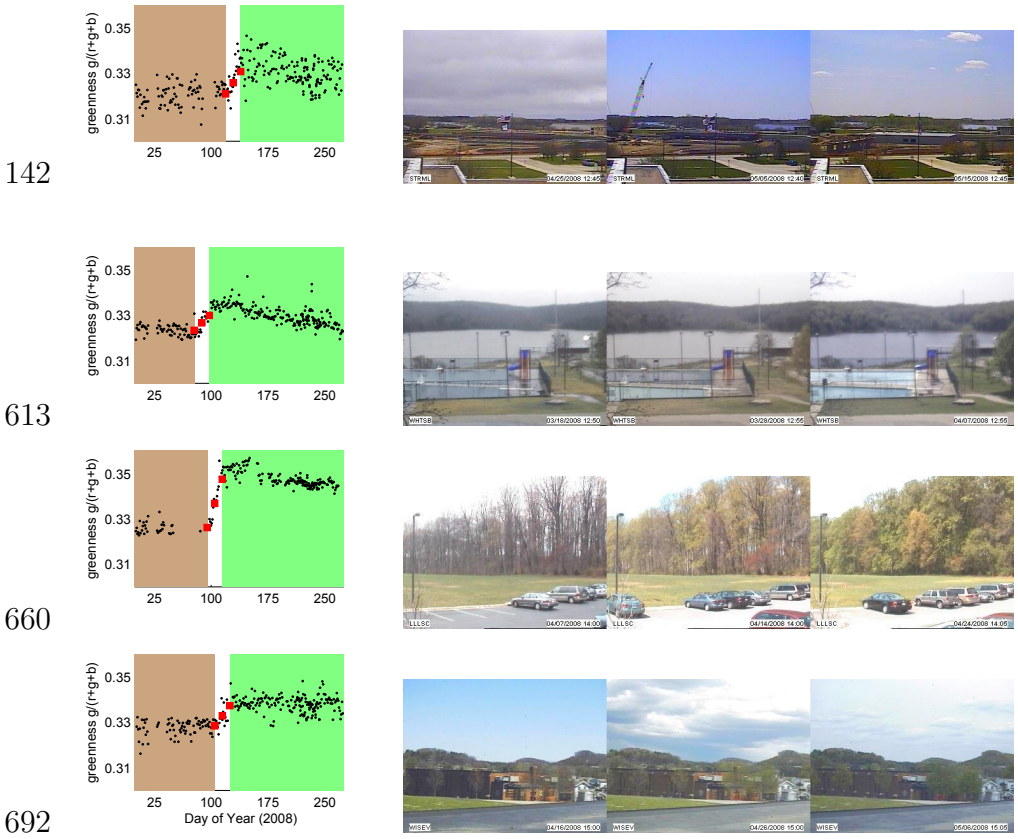
$$g(t) = a + \frac{b}{1+\exp(c-dt)} + \frac{e}{1+\exp(f-gt)},$$

to the greenness signal (we use Levenberg-Marquardt to fit the model). Figure 5.3 shows the result of fitting this model to the average greenness signal from a camera that views multiple tree species. The time-series shows that the new function is a more accurate model of the data (i.e., the extra sigmoid allows the model to fit the small rise that occurs roughly 20 days before the main rise).

The coefficients of mixture-of-sigmoids model helps to segment the image into regions that correspond to the individual mixture components. To obtain the segmentation, shown in Figure 5.3, we first fit two single-sigmoid models, one for each component in the mixture model, separately to each pixel's greenness signal. Each new model has the same form as Equation (5.1) except two parameters, c and d, are held fixed to the values from the corresponding mixture component (these correspond to the horizontal shift and stretch of the sigmoid). For each pixel, the model with the lowest mean-squared error is chosen as the correct model and the pixel is labeled accordingly. This segmentation breaks the scene into the two types of trees in the field of view.



(a)



(b)

Figure 5.2: Determining the onset of spring leaf growth using webcams. (a) A scatter plot of the locations of webcams used in the experiment (points colors correspond to the midpoint of spring leaf growth and are determined by a sigmoidal model of greenness). (b) (left) The greenness signal of the webcams and the corresponding leaf-growth transitions determined by the sigmoidal model. (right) The images that correspond to the red square markers in the plot in the left column. Careful inspection reveals that our model correctly finds the transition between no-leaves and leaves. Note that this visual inspection is difficult, and in fact often impossible, with satellite imagery.

(a) (b)

Figure 5.3: Using webcam images for phenological monitoring has advantages in operating at a much finer spatial and temporal resolution than satellite imagery. Here we show that the higher spatial resolution enables the distinction between trees with different spring leaf growth rates. (a) The value of the greenness signal (black dots) for a camera viewing a clock-tower in a plaza. The thin (red) curve is the value of a single-sigmoid model fit to the data. The thick (green) curve is the value of a mixture-of-sigmoids model fit to the data. (b) A montage of images captured by the camera. The first image (left) has a color overlay that corresponds to the component of the mixture-of-sigmoids model that best fits the time series of the underlying pixel. The other images provide evidence that the segmentation is meaningful (the purple regions grow leaves earlier than the red regions).

These results offer exciting possibilities for low-cost automated monitoring of vegetation phenology around the world. There are numerous potential applications of the resulting data streams [47], including real-time phenological forecasting to improve natural resource management—in particular, agriculture and forestry—and human health (e.g., the dispersal of allergenic pollen), as well as validation and improvement of algorithms for extracting phenological information from satellite remote sensing data.

5.3 Using Webcams as Environmental Sensors

The phenology application in the previous section shows that the global network of outdoor cameras can be used to generate useful scientific information. In this section, we explore linear methods that enable estimation of more complicated signals than the average color. We consider two weather signals for our driving examples: wind velocity and water vapor pressure. These two signals present unique challenges and opportunities. The effect of wind velocity is limited to locations in the scene that are affected by wind (e.g., flags and vegetation), while the effect of water vapor pressure on the scene may result in broad, but subtle, changes to the image.

The challenge with both is that images may vary less because of the variations related to the signal of interest and more because of other causes. Thus we explore efficient mechanisms to learn features that are invariant to other scene variations, as well as tools to measure features and fit distributions to measurements conditioned on known causes. Our method assumes the availability of images and nearby weather data with corresponding time stamps.

We consider a time series of images, $I_1, I_2, \ldots I_n$, captured from a camera with a known geographic location, and a synchronized time series of wind velocity estimates, $Y_1, Y_2, \ldots Y_n$, captured from a nearby weather station. Canonical correlation analysis [24] (CCA) is a tool for finding correlations between a pair of synchronized multi-variate time signals. Applied to this problem, it finds a projection vector, A, and a projection vector, B, that maximizes the correlation between the scalar values AI_t and BY_t, over all time steps, t. Then, given a new image, I_{t+1}, we can predict the projected version of the wind velocity signal as: $BY_{t+1} \approx AI_{t+1}$. We find that both the A and the B matrices relate to interesting features of the scene.

We now consider two examples to evaluate our method. As input, we use images from the AMOS dataset and weather data from the Historical Weather Data Archives (HDWA) of the National Oceanic and Atmospheric Administration (NOAA). We use the ground truth location of the camera to find the location of the nearest weather station and use the provided web interface to download the desired data. In both cases, we solve for PCA and CCA projections using 200 images captured once per day at noon for two months and evaluate on images captured once per day at noon from the following two weeks.

The first example is in predicting wind velocity. The output of CCA is a pair of matrices, A and B, that approximates a linear projection of the wind velocity. Figure 5.4 shows results including the linear image projection found by our method. This projection clearly highlights the orientation of the flag. The plot shows the first dimension that CCA predicts from both the webcam images and the weather data for our test data. The prediction of the second dimension (not shown) is much less accurate, which means that for this scene our method is able to predict only one of two components of the wind velocity. This result is not surprising, because the image of the flag in the scene would be essentially the same if the wind was blowing

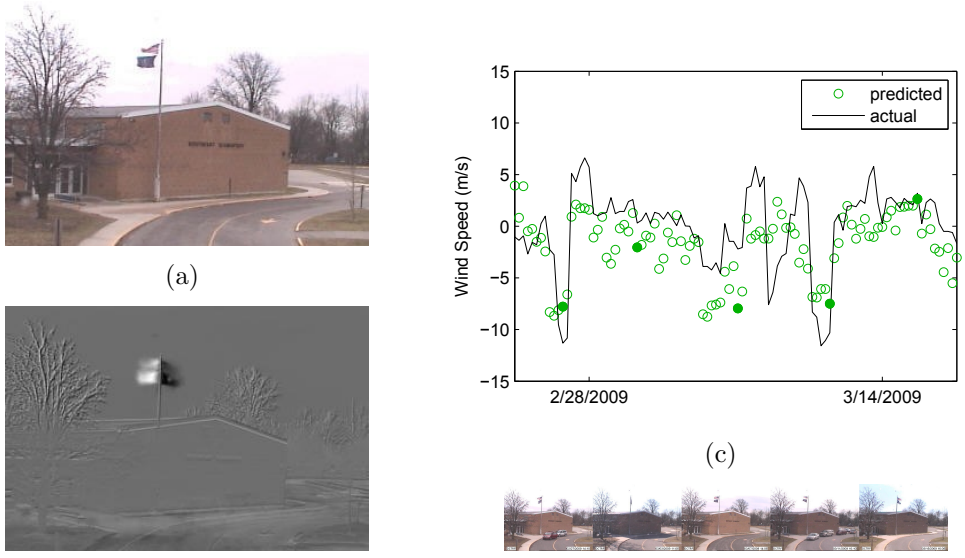


Figure 5.4: An example of predicting wind speed from webcam images. An example image (a) and the CCA projection (b) used to linearly predict the wind speed from a webcam image. (c) Predicted wind speed values and corresponding ground truth. (d) A montage in which each image corresponds to a filled marker in the plot above.

towards or away from the camera. In Figure 5.5, we show the relationship of the CCA projection vector and the geographic structure of the scene. We find that the wind velocity projection vector is, as expected, perpendicular to the viewing direction of the camera.

As a second example, we use a different scene and attempt to predict the water vapor pressure, the contribution of water vapor to the total atmospheric pressure (we note that since water vapor pressure is a scalar the CCA-based method is equivalent to linear regression). Using the method exactly as described for predicting wind velocity in the previous example gives a very low correlation prediction; in other words, no linear projection of a webcam image is capable of predicting water vapor pressure. We find that replacing the original images with the corresponding gradient magnitude image achieves much better results. Figure 5.6 shows water vapor pressure prediction results using the gradient magnitude images. The results indicate that water vapor pressure is strongly related to differences in gradient magnitudes in the scene, because clouds in the sky lead to lower contrast shadows.

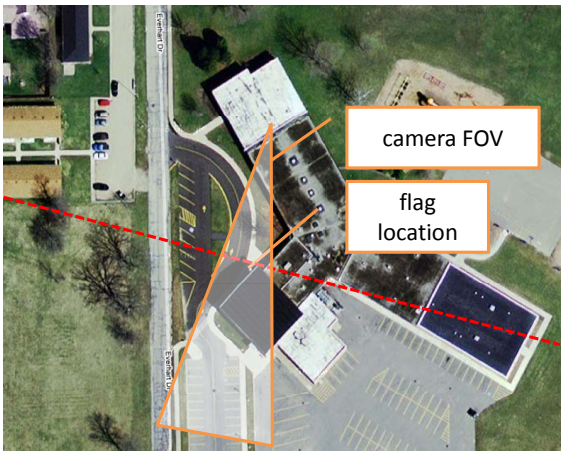
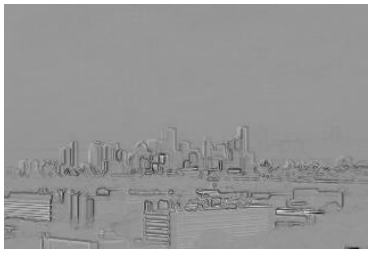
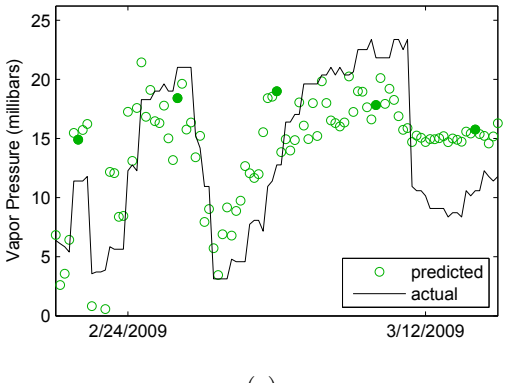


Figure 5.5: An image from Google Maps of the area surrounding the camera. The camera FOV was manually estimated by visually aligning scene elements with the satellite view. The dashed line (red) is the CCA projection axis defined as B in Section 5.3. This image confirms that, as one would expect, our method is best able to predict wind direction when the wind is approximately perpendicular to the principal axis of the camera.

(a)



(b)



(c)

(d)

Figure 5.6: An example of predicting water vapor pressure (a meteorological quantity) from webcam images. (a) An example image of the scene. (b) The projection of the gradient magnitude used to linearly predict the water vapor pressure. (c) Predicted water vapor pressure values and corresponding ground truth. (d) Each image corresponds to a filled marker in the plot above. Inspection of the images revealed that the poor prediction accuracy for the period following March 12, 2009, was due to heavy fog and subsequent water on the optics.

We demonstrated that a generic supervised learning technique can automatically learn to estimate the relationship between a time lapse of images and a time-varying weather signal (in this case, wind velocity) [27]. The supervised setting, while limited to situations in which a collocated sensor is available, demonstrates that extracting a variety of environmental properties is possible. A side benefit is that the models trained in this fashion often show interesting relationships to the calibration of the camera; in this case, we find a relationship between a model for predicting wind velocity and the geo-orientation of the camera.

5.4 Generating Satellite Images from Many Webcams

In Chapter 3, we solve for camera locations by finding the maximum correlation between the time series of webcam images and the time series of pixel intensities of a geo-registered satellite image. This leads us to consider the reverse question: Could a collection of widely distributed cameras allow us to predict an unknown satellite image? In this section, we demonstrate the ability to construct visible satellite images.

We take the supervised approach by using regularized linear regression to learn a mapping from a set of webcam images to a single satellite image. Each training example consists of a satellite image, $S(t)$, and a set of webcam images, $I_{c,t}$, taken at the same time, t. We first reduce the dimensionality of webcam images, $I_{c,t}$, separately at each camera using PCA and use the first k PCA coefficients as predictors (the results shown use $k = 3$) in the regression model.

To learn the regression model, we construct a matrix of T satellite images, $S \in \mathbb{R}^{p \times T}$, where each column is a satellite image. The webcam data is summarized as a matrix of PCA coefficients, $V \in \mathbb{R}^{T \times k}$, where each row contains the first k PCA coefficients for all cameras for images captured at given time. We then solve for the set of coefficients, F, using ridge regression, $F = SV(V^{\mathsf{T}}V + \lambda \mathbf{I})^{-1}$, with a regularization constant, $\lambda = .01$. Using F we can predict an unseen satellite image from a set of camera PCA coefficients, V_t^{T}, by multiplying by the coefficient matrix, FV_t^{T}.

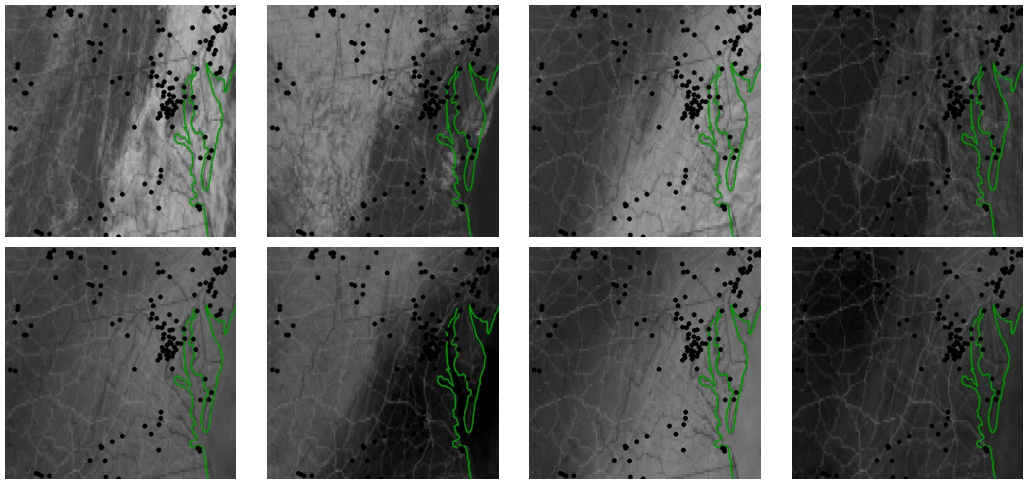


Figure 5.7: Satellite images predicted from webcam images. (top) Ground-truth visible-satellite images from the Washington, D.C. area. (bottom) Synthetic satellite images created with a linear regression model that uses the time-stamped PCA coefficients of webcams (located at black dots) as predictors.

We evaluated this method using a set of 1700 visible satellite images from four consecutive months and 42 webcams in the Maryland/Virginia area (the set shown in Figure 3.5). We use 1400 of these satellite images to define the linear regression model. Figure 5.7 shows that prediction of satellite images from web camera images is feasible using these methods.

When these images are scaled so that black is 0 and white is 1, the average per-pixel variance in the original images is 0.0183. Reproducing these images (from the training set) gives and an average per-pixel SSD error of 0.0070. Thus, using a linear model with input from 42 cameras explains 62% of the image variation. This limited success highlights the challenges of reproducing the satellite images using a linear model. Testing this regression model on the validation data, we find an SSD error of 0.0082, explaining 55% of the image variation. We foresee significant decreases in reconstruction error with the use of more sophisticated image representations and estimation methods that use temporal constraints to combine information from webcam images captured at different times.

Chapter 6

Discussion

Our work was motivated by the potential of using the global network of outdoor webcams (GNOW) as an environmental imaging resource. We focused on defining the global network of outdoor webcams, discovering natural temporal variations in outdoor webcam scenes, calibrating cameras in the network, estimating scene geometry, and exploring potential scientific applications. As part of this we have defined several new computer vision problems with corresponding novel solutions.

We have considered the underlying causes of image appearance variation through a model we call the geo-temporal image formation model (GIFM). A better understanding of the GIFM is of fundamental importance to the field of computer vision. We developed algorithms that focus on portions of this model for problems including: estimating a depth map, estimating the geolocation of the camera, and labeling pixels based on surface orientation.

We showed that it is possible to use the GNOW as a scientific imaging resource in the context of a real application: estimating the onset of leaf growth on trees. Beyond this application, there are many other areas that may benefit from this resource. Example applications include: monitoring coastal erosion and changes in glaciers, measuring atmospheric visibility, tracking plant growth, and detailed analysis of cloud cover. Our work in discovering, archiving, calibrating, and understanding webcam images lays an important foundation for these and other future scientific uses.

The algorithms we developed and insights gained from the GIFM will likely apply to other domains. For example, video surveillance system are often designed to characterize the transient objects within a scene, but the appearance of those transient

objects depends on the 3D scene structure and the lighting. A formal approach to characterizing the GIFM may allow surveillance systems to reason more explicitly about occlusions, shading of the object, and shadows moving in the scene. The automated methods we provide to characterize 3D scene structure may also decrease the amount of manual effort necessary to deploy a camera and solve for these elements of the GIFM.

An important area for future work is to further refine our methods to support additional scientific imaging applications. It is likely that a particular application will require improved calibration or signal extraction methods. These applications may also motivate novel calibration problems. More generally, this will lead to the need for more accurate models of the natural phenomena that generate images and for automated algorithms that simultaneously consider larger portions of the GIFM.

References

[1] Archive of Many Outdoor Scenes, website. http://amos.cse.wustl.edu/.

[2] Patrick Baker and Yiannis Aloimonos. Calibration of a multicamera network. In *Omnivis 2003: Omnidirectional Vision and Camera Networks*, 2003.

[3] Vincent A. Billock. Neural acclimation to 1/f spatial frequency spectra in natural images transduced by the human visual system. *Phys. D*, 137(3-4):379–391, 2000.

[4] E. Bradley, D. Roberts, and C. Still. Design of an image analysis website for phenological and meteorological monitoring. *Environmental Modelling and Software*, 2009.

[5] Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings European Conference on Computer Vision (ECCV)*, 2002.

[6] G. J. Burton and Ian R. Moorhead. Color and spatial structure in natural scenes. *Applied Optics*, 26(1):157–170, 1987.

[7] Junghoo Cho and Hector Garcia-Molina. Estimating frequency of change. *ACM Trans. Internet Technology*, 3(3):256–290, 2003.

[8] CIE Technical Committee. Spatial distribution of daylight - luminance distributions of various reference skies. Technical Report CIE-110-1994, International Commission on Illumination, 1994.

[9] Fabio Cozman and Eric Krotkov. Robot localization using a computer vision sextant. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 106–111, Nagoya, Japan, May 1995.

[10] Fabio Cozman and Eric Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1997.

[11] Paul M. Dare. Shadow analysis in high-resolution satellite imagery of urban areas. *Photogrammetric Engineering and Remote Sensing*, 71(2):169–177, 2005.

[12] Dhanya Devarajan, Richard J. Radke, and Haeyong Chung. Distributed metric calibration of ad hoc camera networks. *TOSN*, 2(3):380–403, 2006.

[13] Dawei W. Dong and Joseph J. Atick. Statistics of natural time-varying images. *Network: Computation in Neural Systems*, pages 345–358, 1995.

[14] Geostationary Satellite Server. http://www.goes.noaa.gov/.

[15] Eric A. Graham, Erin C. Riordan, Eric M. Yuen, John Hicks, Eric Wang, Deborah Estrin, and Philip W. Rundel. Leveraging internet-connected cameras to create a transcontinental plant phenology monitoring system. Poster presentation at 94th Ecological Society of America (ESA) Annual Meeting, August 2009.

[16] Jorge Guillén, Antonio García-Olivares, Elena Ojeda, Andres Osorio, Oscar Chic, and Raul González. Long-term quantification of beach users using video monitoring. *Journal of Coastal Research*, 24(6):1612–1619, 2008.

[17] H. John Heinz III Center for Science, Economics, and the Environment, The. *The State of the Nation's Ecosystems 2008: Measuring the Land, Waters, and Living Resources of The United States*. Island Press, September 2008.

[18] P. Hancock, R. Bradley, and L. Smith. The principal components of natural images. *Network*, 3:61–70, 1992.

[19] Haze Cam Pollution Visibility Camera Network. http://www.hazecam.net/.

[20] R. Holman, J. Stanley, and T. Ozkan-Haller. Applying video sensor networks to nearshore environment monitoring. *Pervasive Computing, IEEE*, 2(4):14–21, Oct.-Dec. 2003.

[21] Rob Holman, John Stanley, and Tuba Ozkan-Haller. Applying video sensor networks to nearshore environment monitoring. *IEEE Pervasive Computing*, 2(4):14–21, 2003.

[22] A. D. Hopkins. Periodical events and natural law as guides to agricultural research and practice. *Monthly Weather Review. Supplement No. 9*, pages 1–42, 1918.

[23] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV FRAME-RATE Workshop*, 1999.

[24] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[25] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report CS TR-07-49, University of Massachusetts, Amherst, 2007.

[26] IPInfoDB. http://www.ipinfodb.com/.

[27] Nathan Jacobs, Walker Burgin, Richard Speyer, David Ross, and Robert Pless. Adventures in archiving and using three years of webcam images. In *Proceedings IEEE CVPR Workshop on Internet Vision*, 2009.

[28] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[29] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In *IEEE Workshop on Applications of Computer Vision (WACV)*, January 2008. (oral presentation).

[30] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *IEEE International Conference on Computer Vision (ICCV)*, October 2007.

[31] John Jannotti and Jie Mao. Distributed calibration of smart cameras. In *Workshop on Distributed Smart Cameras*, 2006.

[32] Martina Kammler and Gerald Schernewski. Spatial and temporal analysis of beach tourism using webcam and aerial photographs. In G. Schernewski and N. Löser, editors, *Managing the Baltic Sea: Coastline Reports*. Costal and Marine Union (EUCC), 2004.

[33] Seon Joo Kim, J.-M. Frahm, and M. Pollefeys. Radiometric calibration with illumination change for outdoor scene analysis. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[34] Sanjeev J. Koppal and Srinivasa G. Narasimhan. Clustering appearance for scene analysis. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[35] Sanjeev J. Koppal and Srinivasa G. Narasimhan. Appearance derivatives for isonormal clustering of scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1375–1385, 2009.

[36] J. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2), June 1964.

[37] J. B. Kruskal and M. Wish. Multidimensional scaling. *Sage University Paper series on Quantitative Application in the Social Sciences*, pages 07–011, 1978.

[38] Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 28(5), December 2009.

[39] Jean-François Lalonde, Srinivasa G. Narasimhan, and Alexei A. Efros. What does the sky tell us about the camera? In *Proceedings European Conference on Computer Vision (ECCV)*, 2008.

[40] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:2169–2178, 2006.

[41] Robert A. Long, Paula MacKay, Justina Ray, and William Zielinski, editors. *Noninvasive Survey Methods for Carnivores*. Island Press, May 2008.

[42] William Mantzel, Hyeokho Choi, and Richard Baraniuk. Distributed camera network localization. In *Proc. Signals, Systems and Computers*, volume 2, 2004.

[43] Dimitri Marinakis and Gregory Dudek. Topology inference for a vision-based sensor network. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 121–128, 2005.

[44] Measuring Vegetation Health Project. http://mvh.sr.unh.edu/.

[45] Anurag Mittal, Antoine Monnet, and Nikos Paragios. Scene modeling and change detection in dynamic scenes: A subspace approach. *Computer Vision and Image Understanding*, 113(1):63 – 79, 2009.

[46] A. Moreno. The role of weather in beach recreation—a case study using webcam images. In A Matzarakis, C R de Freitas, and D Scott, editors, *Developments in Tourism Climatology*. International Society of Biometeorology: Commission on Climate, Tourism and Recreation, 2007.

[47] Jeffrey T. Morisette, Andrew D. Richardson, Alan K. Knapp, Jeremy I. Fisher, Eric A. Graham, John Abatzoglou, Bruce E. Wilson, David D. Breshears, Geoffrey M. Henebry, Jonathan M. Hanes, and Liang Liang. Tracking the rhythm of the seasons in the face of global change: phenological research in the 21st century. *Frontiers in Ecology and the Environment*, 7(5):253–260, 2009.

[48] Srinivasa G. Narasimhan and Shree K. Nayar. Shedding light on the weather. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[49] Srinivasa G Narasimhan, Chi Wang, and Shree K Nayar. All the images of an outdoor scene. In *Proceedings European Conference on Computer Vision (ECCV)*, 2002.

[50] National Climatic Data Center. http://www.ncdc.noaa.gov/.

[51] J. Oh, Q. Wen, J. Lee, and S. Hwang. Video abstraction. *Video Data Management and Information Retrieval*, pages 321–346, 2004.

[52] Bruno A. Olshausen. Learning sparse, overcomplete representations of time-varying natural images. In *ICIP (1)*, pages 41–44, 2003.

[53] Bruno A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Network*, 7(2):333–340, 1996.

[54] Opentopia Webcams. http://www.opentopia.com/hiddencam.php.

[55] M.L. Parry, O.F. Canziani, J.P. Palutikof, P.J. van der Linden, and C.E. Hanson, editors. *Climate Change 2007: Impacts, Adaptation and Vulnerability*. Cambridge University Press, 2007.

[56] PhenoCam. http://phenocam.unh.edu/.

[57] Phenological Eyes Network. http://pen.agbi.tsukuba.ac.jp/.

[58] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:947–954 vol. 1, June 2005.

[59] Andrea Prati, Ivana Mikic, Mohan M. Trivedi, and Rita Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003.

[60] Arcot J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. In *Proceedings ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 91–100, New York, NY, USA, 1999.

[61] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, Oct. 2007.

[62] I. Reda and A. Andreas. Solar position algorithm for solar radiation application. Technical Report NREL/TP-560-34302, National Renewable Energy Laboratory, 2003.

[63] Andrew Richardson, Julian Jenkins, Bobby Braswell, David Hollinger, Scott Ollinger, and Marie-Louise Smith. Use of digital webcam images to track spring green-up in a deciduous broadleaf forest. *Oecologia*, 152(2):323–334, May 2007.

[64] Andrew D. Richardson, Bobby H. Braswell, David Y. Hollinger, Julian P. Jenkins, and Scott V. Ollinger. Near-surface remote sensing of spatial and temporal variation in canopy phenology. *Ecological Applications*, 19(6):1417–1428, 2009.

[65] Andrew D. Richardson, Julian P.Jenkins, Bobby H. Braswell, David Y. Hollinger, Scott V. Ollinger, and Marie-Louise Smith. Use of digital webcam images to track spring green-up in a deciduous broadleaf forest. *Oecologia*, 152(2), 2007.

[66] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008.

[67] Eero P. Simoncelli and Bruno A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

[68] MWJ Smit, SGJ Aarninkhof, Kathelijne Mariken Wijnberg, M González, KS Kingston, HN Southgate, BG Ruessink, RA Holman, E Siegle, M Davidson, et al. The role of video imagery in predicting daily to monthly coastal evolution. *Coastal engineering*, 54(6-7):539–553, 2007.

[69] Chris Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2246–2252, 1999.

[70] Fridtjof Stein and Gerard Medioni. Map-based localization using the panoramic horizon. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Nice, France, 1992.

[71] Chen-Hui Sun and Lawrence R. Thorne. Inferring spatial cloud statistics from limited field-of-view, zenith observations. In *Proceedings of the Fifth Atmospheric Radiation Measurements (ARM) Science Team Meeting*, pages 331–334. U.S. Department of Energy, 2000.

[72] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[73] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics*, 26(3), 2007.

[74] William Thompson, Thomas Henderson, Thomas Colvin, Lisa Dick, and Carolyn Valiquette. Vision-based localization. In *ARPA Image Understanding Workshop*, pages 491–498, Washington D.C., 1993.

[75] Lawrence R. Thorne, Kenneth Buch, Chen-Hui Sun, and Carl Diegert. Data and image fusion for geometrical cloud characterization. Technical Report SAND97-9252, Sandia National Laboratories, 1997.

[76] Kinh Tieu, Gerald Dalley, and W. Eric L. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, pages 1842–1849, 2005.

[77] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007.

[78] Ashitey Trebi-Ollennu, Terry Huntsberger, Yang Cheng, E. T. Baumgartner, Brett Kennedy, and Paul Schenker. Design and analysis of a sun sensor for planetary rover absolute heading detection. *IEEE Trans. on Robotics and Automation*, 17(6), 2001.

[79] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings* -, 152(2):192–204, April 2005.

[80] J. H. van Hateren and D. L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London, Series B*, 265:2315–2320, 1998.

[81] Weather Underground. http://www.wunderground.com/.

[82] Weatherbug Inc. http://weatherbugmedia.com/.

[83] Lisa Wingate, Andrew D. Richardson, Jake F. Weltzin, Kenlo N. Nasahara, and John Grace. Keeping an eye on the carbon balance: linking canopy development and net ecosystem exchange using a webcam network. *Fluxletter*, 1(2):14–17, 2008.

[84] C. J. Wong, M. Z. MatJafri, K. Abdullah, and H. S. Lim. Temporal and spatial air quality monitoring using internet surveillance camera and alos satellite image. In *Proceedings IEEE Aerospace Conference*, 2009.

[85] C. J. Wong, M. Z. MatJafri, K. Abdullah, H. S. Lim, and K. L. Low. Temporal air quality monitoring using internet surveillance camera and alos satellite image. In *Proceedings IEEE Geoscience and Remote Sensing Symposium*, 2007.

[86] Ling Xie, Alex Chiu, and Shawn Newsam. Estimating atmospheric visibility using general-purpose cameras. In *Proceedings International Symposium on Advances in Visual Computing*, pages 356–367, 2008.

Vita

Nathan Bradley Jacobs

Date of Birth June 9, 1977

Place of Birth Sycamore, Missouri

Degrees B.S. Summa Cum Laude, Computer Science, December 1999
Ph.D. Computer Science and Engineering, May 2010

Professional Societies Institute of Electrical and Electronics Engineers
Association for Computing Machines
Society for Industrial and Applied Mathematics

Publications

Nathan Jacobs, Stephen Schuh, and Robert Pless. Compressive sensing and differential image motion estimation. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), March 2010.

Michael Dixon, Nathan Jacobs, and Robert Pless. An efficient system for vehicle tracking in multi-camera networks. In ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), September 2009.

Nathan Jacobs, Walker Burgin, Nick Fridrich, Austin Abrams, Kylia Miskell, Bobby H. Braswell, Andrew D. Richardson, and Robert Pless. The global network of outdoor webcams: Properties and applications. In ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL GIS), November 2009.

Nathan Jacobs, Walker Burgin, Richard Speyer, David Ross, and Robert Pless. Adventures in archiving and using three years of webcam images. In IEEE CVPR Workshop on Internet Vision, June 2009.

Nathan Jacobs, Michael Dixon, Scott Satkin, and Robert Pless. Efficient tracking of many objects in structured environments. In IEEE ICCV Workshop on Visual Surveillance, October 2009.

Nathan Jacobs, Richard Souvenir, and Robert Pless. The global webcam imaging network. In Applied Imagery Pattern Recognition Workshop (AIPR), 2009.

Robert Pless, Nathan Jacobs, Michael Dixon, Ralph Hartley, Patrick Baker, Derek Brock, Nick Cassimatis, and Dennis Perzanowski. Persistence and tracking: Putting vehicles and trajectories in context. In Applied Imagery Pattern Recognition Workshop (AIPR), 2009.

Nathan Jacobs, Michael Dixon, and Robert Pless. Location-specific transition distributions for tracking. In IEEE Workshop on Motion and Video Computing (WMVC), January 2008.

Nathan Jacobs, Nathaniel Roman, and Robert Pless. Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In IEEE Workshop on Applications of Computer Vision (WACV), January 2008.

Nathan Jacobs and Robert Pless. Shape background modeling : The shape of things that came. In IEEE Workshop on Motion and Video Computing (WMVC), February 2007.

Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2007.

Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In IEEE International Conference on Computer Vision (ICCV), October 2007.

Michael Dixon, Nathan Jacobs, and Robert Pless. Finding minimal parameterizations of cylindrical image manifolds. In IEEE CVPR Workshop on Perceptual Organization in Computer Vision (POCV), June 2006.

Nathan Jacobs and Robert Pless. Real-time constant memory visual summaries for surveillance. In ACM International Workshop on Visual Surveillance and Sensory Networks (VSSN), October 2006.

May 2010

Using Thousands of Outdoor Webcams, Jacobs, Ph.D. 2010