Weatherbot: an Automated Skywatcher

Randal C. Nelson
Department of Computer Science
University of Rochester

June 13, 2007

Abstract

We describe a self bootstrapping and adaptive system designed to make observations of an outdoor environment and determine some simple environmental conditions, specifically whether it is day or night, and whether the conditions are clear, cloudy, or some mixture. Doing this so that the system will self-adapt and operate reliably in a variety of locations and environments, robust against changes in seasons, weather, and typical human and non-human disturbances (e.g. streetlights, thunderstorms) is a more complex problem than might first be thought. We describe some of the practical issues, and techniques for dealing with them.

1 Introduction

This paper describes a self bootstrapping and adaptive system that makes observations of an outdoor environment and determines certain environmental conditions, specifically whether it is day or night, and whether the conditions are clear, cloudy, or some mixture. This is more complex than might at first be thought, especially if we want a system that will self-adjust to different hardware and operate reliably in a variety of locations and environments, robust against changes in seasons, weather, and typical human and non-human disturbances (e.g. streetlights, cars, thunderstorms, snowstorms etc.). This is particularly true if one wants to use inexpensive, off-the-shelf sensors, and avoid time consuming, explicit calibration procedures, that may be valid only for a specific sensor in a particular location, in a particular season. We describe some of the difficulties, both generic and specific to the problem, and detail some techniques for dealing with them.

2 Background

Determining day or night for the purposes of controlling outdoor lighting has been performed using simple photosensors for half a century or more. Such mechanisms are adequate for simple tasks where an occasional error has no significant consequences, but have limitations. For example, photosensor control systems turn lights on during daytime storms, or if objects temporarily obscure the sensor. They turn lights off at night if hit by external illumination such as headlights, or in conditions of back-reflection from dense fog or snow. For intelligent systems that may have more complex procedures or temporally planned activities contingent on day and night, errors can be more consequential, and more sophisticated systems are warranted.

*Support for this work provided by NSF Grant ECS-0225413, NSF Grant HLC-0308049, and NSF Grant EIA-0080124
The problem of automatically determining cloud cover from ground imagery has been addressed in several systems, mostly reported in the remote sensing literature. One of the few references in the machine vision literature by Richards and Sullivan in 1992 [1]. They showed that for a small sample of color sky images, Gaussian models in rgb space trained using hand-segmented data, could do a reasonable job marking clear and cloud pixels. Performance was slightly improved by adding dimensions corresponding to texture filter reponses. The system had difficulty distinguishing haze from cloud, and thin cirrus from sky. This is perhaps to be expected. More seriously, the system was not tested on pictures taken a various times of day, or seasons of the year. Given observations we have made concerning the degree to which lighting changes during the course of a day, such a fixed classifier would likely perform poorly if subjected to a broader range of natural conditions. This work does not appear to have been continued, or in fact, ever referenced by any other work.

The remote sensing community has primarily addressed the problem of cloud cover determination from the standpoint of satellite imagery, but there have been a few systems that used ground imagery. The emphasis in these systems has generally been on the optical components, especially methods for obtaining full hemispherical data, and less on sophistication of processing. The best-known project is work at the Marine Physical Laboratory of the Scripps institute, which has had a ground-based sensor program for several decades [3]. The most recent instrument is an impressive device featuring a fisheye lens, a clockwork mechanism for active physical blocking of the sun’s disk, and a high-resolution, high-dynamic range, 16-bit cooled sensor with 4 spectral filters extending into the near IR. This instrument can make color observations even by moonlight. It is not, however cheap. The algorithm for separating cloud from sky is based on the red-blue channel ratio, and does a reasonable job during hours when the sun is not near the horizon, and there is sufficient sunlight or moonlight. Experiments are under way to determine coverage during moonless nights by looking counting visible stars and comparing to constellation maps. The classifier is hand tuned, and requires extensive calibration procedures.

A couple other projects have reported similar devices [2, 4]. All seem to rely on the red/blue channel ratio, use hand-tuned classifiers, and require careful calibration. Issues with the chromatic variation of the lighting during the day are handled by staying well away from sunrise and sunset - the times when such variation is most extreme. No projects we have found attempt to automatically acquire and adapt the required models, or characterize and compensate for the dramatic variation in cloud color found within the hour or so after sunrise and before sunset.

3 Day and Night

Suppose we want to design a visual system living (primarily) in an outdoor environment that knows whether it is day or night. This could be done on the basis of an accurate calendar, clock, knowledge of current latitude and longitude, and an appropriate set of tables. Such systems are high-information, and rather fragile. Or it might be determined on the basis of visual observations of the environment with only approximate measures of elapsed time. It is in this possibility that we are interested.

The first question we should ask in designing such a system is what sorts of variation in the environment it needs to be able to deal with. In the real world, for example, we need to handle the fact that the lengths of day and night periods vary throughout the year. Moreover, within day and night periods, the brightness of a particular scene varies dramatically with the time of day, the amount of cloudiness, and factors such as whether there is snow on the ground. We might also want to handle the possibility that the system might move around in the environment, and thus not always be viewing the same scene. We might want the system to be able to initialize itself and work with no prior information about what brightness levels are associated with day and night, and to adapt if these relations change (e.g. if the system is moved to a new environment, or suffers a modification in a sensor). We probably want to handle the fact that certain short-term phenomena
such as lightning or car headlights at night or brief blocking of the camera lens by moving agents during the day could yield brightness measurements that would be misclassified by any classifier utilizing only instantaneous illumination levels. We might even want the system to recover if it is brought inside and exposed to non-diurnal lighting conditions for a time, or put on a plane and transported to a different time zone.

It turns out that, in non-polar regions, if one makes an instrument that measures the average brightness of the sky (in a watts/m² steradian sense), or even a small overhead patch of it then the range of ”daytime” values encountered from sunrise to sunset does not overlap the range of ”nighttime” values encountered from say 30 minutes after sunset to 30 minutes before sunrise. In fact, a significant gap exists between the ranges. So if we can arrange for our system to be able to take a brightness measurement of the sky whenever it wants to know the daytime condition, then a large chunk of the problem is solved.

One problem with this approach is that a mobile agent may not always be able to see the sky (e.g. it might be in a forest), and even if it can, it might need significant time and smarts to locate it. Another problem is that it is not that easy to get a direct brightness measurement from standard cameras, which usually have several undocumented automatic exposure control mechanisms operating.

However the size of the gap between daytime and nighttime brightnesses suggests that it may not be necessary to actually look at the sky or even correct explicitly for camera autoexposure, as long as the available gain is not significant enough to fully normalize nighttime views (true for all but special military sensors). In fact, since the brightness of objects in the environment depends on an integration over all sources of illumination, which are dominated in daytime by contributions from the sky, it is probably possible to make the discrimination on the basis of some measure of scene illumination as long as sources of artificial nighttime illumination are not too large.

4 Day-Night Model

If we want a single number from an image that provides the most direct information about the overall level of illumination, the most useful quantity would be the image brightness of a piece of material that has white Lambertian reflectance and is maximally exposed. Explicit knowledge of the reflectances of materials in the scene is unlikely to be available, however if the world is Lambertian and uniformly illuminated, then if such a piece of material is visible, it will be the brightest object in the image. In real images, the actual brightest locations are typically due to specular reflections from non-Lambertian objects or direct images of illumination sources such as the sun or artificial lights. The total area of such objects however, is generally relatively small, which suggests that using the brightness of some upper fraction of the image pixels might approach the desired characteristics.

For our system we compute the mean gray-value of brightest the half of the pixels. Experimental evaluation indicates that this is large enough to be insensitive to local non-lambertian effects, and significantly better than using the mean gray value of the entire image. The primary situation in which the benefit is apparent is in brightly lit, highly contrasting scenes, for example outdoor images in direct sunlight, where the dark shadowed areas can drag down the overall mean to a problematic degree.

In environments characterized by day and night periods, we expect the long-term distribution of the image brightness values to exhibit a bimodal distribution. If we want to determine decision parameters automatically, then analysis of this distribution should provide sufficient information. Moreover, if the distribution is maintained dynamically, then the system can be made adaptive to changes in the environment and sensors. In our system the brightness distribution is dynamically maintained by mixing the long-term model with a recent distribution acquired over the last full day-night period, so long as that recent distribution appears to be bimodal and the mixing does not result in the destruction of the bimodality of the accumulated distribution. The last two conditions constitute a ”survival” mechanism that prevents the sys-
tem from destroying useful existing knowledge if unusual, conditions such as introduction into a constantly lighted artificial environment, or failure of a sensor for several days occurs. The combination is done using a "leaky integrator" approach to approximating a running average. \( A_{new} = (1 - \alpha)A_{old} + \alpha A_{recent} \) This update is done during a "rem sleep" period that occurs once per day (at nightfall). Currently \( \alpha \) is set to 0.1, which when coupled with an update rate of once per day is equivalent to a half-life of about a week.

The accumulated distribution data is maintained as an integer histogram of 255 possible illumination values, normalized to sum to a unity equivalent (currently 1,000,000). We use integers rather than floating point values summing to 1.0 to simplify storing the model to a readable text file without loss of precision. From this histogram are derived parametric models of the distribution of brightness during daytime and nighttime periods. Specifically, we describe each by a mean, standard deviation, and a weight representing overall frequency in the distribution. These values are determined by fitting a mixture of Gaussians model to the distribution using the EM algorithm.

Starting points for the day and night models are determined by a heuristic analysis of the distribution. The method employed is to find the overall maximum, take its location as the first peak, and estimate the standard deviation by exploring outward in both directions until a value is found that is less than 1/2 the maximum. The standard deviation is estimated from the distance. This distribution is then removed from the data, and the second Gaussian parameters estimated directly from the residual data. Since spiky data or subsidiary peaks can confuse the first step, we smooth the original distribution with a Gaussian having a standard deviation of 2 brightness levels. It might be thought that since the brightness values are determined by averaging a large number of pixel values, and the histogram is accumulated from a large number of observations, the data would be quite smooth to start with. Due to the discrete nature of the automatic exposure control built into the camera, this is not true, and some mean brightness values occur more frequently than others, especially in dark scenes. This results in an overall histogram that has multiple subsidiary peaks, shown in Figure 1.

![Day-night brightness histogram showing multiple subsidiary peaks due to discrete automatic exposure control.](image)

Figure 1: Day-night brightness histogram showing multiple subsidiary peaks due to discrete automatic exposure control.

Analysis of the brightness data is performed by fitting a mixture of two Gaussians to the distribution. Prior to this process, the distribution is modified to remove the middle .05 of the mass from the distribution.
The reason for this is that, while the brightness values for day and night periods are fairly accurately modeled by a well separated Gaussian distributions, the values observed during the transitional twilight period produce a non-Gaussian tail if grouped with either model. Since the EM algorithm tries to explain all the data, the result is typically that the twilight values get pushed into one model or the other, significantly increasing the variance/standard-deviation and shifting the mean. The .05 value is obtained from prior knowledge that twilight periods constitute about 1/20 of the day. Seeding a third Gaussian model between the two starting points also works, but not as well as the knowledge-based heuristic.

Table 1. shows the mean and standard deviation of the dark and light (presumptively, night and daylight) adaptive gaussian models on various dates. There is some variation in the means, and considerable variation in the standard deviation of the models, especially for the night model. This reflects variation in the short-term climate. For example, cloudy weather, and especially snow increases the brightness and variation at night due to scattered city light. However, there are no systematic instabilities in the adaptive system. None of the values show any tendency to tend towards extremes.

<table>
<thead>
<tr>
<th>Date</th>
<th>day-mean</th>
<th>day-std</th>
<th>day-wt</th>
<th>night-mean</th>
<th>night-std</th>
<th>night-wt</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-08-2005</td>
<td>201.0</td>
<td>12.0</td>
<td>0.56</td>
<td>27.6</td>
<td>6.1</td>
<td>0.44</td>
</tr>
<tr>
<td>05-03-2005</td>
<td>193.1</td>
<td>11.3</td>
<td>0.61</td>
<td>27.9</td>
<td>2.1</td>
<td>0.39</td>
</tr>
<tr>
<td>06-16-2005</td>
<td>194.4</td>
<td>11.4</td>
<td>0.67</td>
<td>17.5</td>
<td>2.1</td>
<td>0.33</td>
</tr>
<tr>
<td>07-01-2005</td>
<td>190.1</td>
<td>12.7</td>
<td>0.68</td>
<td>13.7</td>
<td>4.3</td>
<td>0.32</td>
</tr>
<tr>
<td>11-17-2005</td>
<td>179.1</td>
<td>8.6</td>
<td>0.44</td>
<td>10.5</td>
<td>1.0</td>
<td>0.56</td>
</tr>
<tr>
<td>12-21-2005</td>
<td>185.0</td>
<td>4.9</td>
<td>0.41</td>
<td>19.6</td>
<td>6.8</td>
<td>0.59</td>
</tr>
<tr>
<td>02-24-2006</td>
<td>213.9</td>
<td>10.7</td>
<td>0.47</td>
<td>32.1</td>
<td>12.4</td>
<td>0.53</td>
</tr>
<tr>
<td>03-21-2006</td>
<td>212.0</td>
<td>6.5</td>
<td>0.52</td>
<td>25.4</td>
<td>8.2</td>
<td>0.48</td>
</tr>
<tr>
<td>07-13-2006</td>
<td>205.4</td>
<td>6.8</td>
<td>0.68</td>
<td>13.9</td>
<td>1.0</td>
<td>0.32</td>
</tr>
<tr>
<td>08-30-2006</td>
<td>190.7</td>
<td>14.0</td>
<td>0.61</td>
<td>19.4</td>
<td>3.6</td>
<td>0.39</td>
</tr>
<tr>
<td>10-06-2006</td>
<td>184.8</td>
<td>12.3</td>
<td>0.53</td>
<td>17.4</td>
<td>1.8</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 1: Variation in day and night model parameters over time

5 Day-Night Semantic State

From recent brightness observations and the day-night models derived from the long-term brightness distribution, the system maintains two semantic state variables, and several auxiliary history variables that are used in determining the primary semantic state.

5.1 Daylight State

The first variable, daylight state, is intended to provide a classification of the most recent brightness observation into bright-as-day, dark-as-night or twilight. There are also two auxiliary states. Init, indicates that the system has either made no observation yet, or has not yet established a model. Unknown indicates that the daylight state cannot be otherwise evaluated for some unspecified reason (e.g. the sensor has failed, or the internal models are corrupt).

The bright-as-day, dark-as-night, twilight distinction is made by splitting the interval between the night mean and the day mean into three equal subintervals. Brightness values falling in the central subinterval are classified as twilight. Values above it are classified bright-as-day, and below, dark-as-night. This fixed division works better than one based on the standard deviations of the day and night models. The reason is that that the brightness values for day and night, even well outside of the twilight zone, are non-stationary.
statistics. For example, during the summer, the nighttime brightness is low, and the standard deviation small. The first snowy night of winter, at least in any urban or suburban area, due to efficient scattering of artificial light, suddenly produces brightness values that are several standard deviations from the summer mean. In some sense, an all-night twilight is an accurate description of the new environment, but it is not what we want the system to produce.

5.2 Daytime State

The second variable, daytime state, is intended to indicate whether it is day or night. It takes one of five possible values \{ day, night, confused, init, unknown \}, and is maintained by a more sophisticated state machine than the direct computation that generates daylight state. Transition between the states is made on the basis of the current daylight state value, and a few real-valued variables representing long and short term integrated histories of daylight observations. These serve to introduce hysteresis into the system and prevent oscillation between daytime values due to brightness variation around the transition times.

5.3 Auxiliarly History Variables

We capture long and short term histories of brightness and darkness using a running average of numerical brightness and darkness values associated with each daylight state value. For efficiency, the running averages are computed using a leaky integrator model. We set the time-constant for the short-term averages to reach 90\% of its final value on a step-function input in about 1/2 the estimated length of the twilight transition period (7.5 minutes) and the long-term average, to reach the 90\% level in about the 1/2 length of an average day or night (6 hours).

The numeric brightness and darkness values are determined as follows: dark-as-night has brightness 0.0, darkness 1.0; bright-as-day has brightness 1.0, darkness 0.0. The effective brightness of twilight depends on brightness history. Under normal operation, we want quick transitions from day to night and vice-versa. We achieve this using a temporal contrast mechanism - twilight after it has been bright a while seems darker than if it has been dark, and vice versa. This permits the short-term brightness integrals to approach the transition thresholds during twilight. Specifically, if the long-term brightness average is greater than a priming threshold (currently 0.8) then twilight has brightness .10 and darkness .90. Conversely if the long-term darkness average is greater than the priming threshold twilight has brightness .90 and darkness .10. Otherwise, twilight has brightness .50 and darkness .50. The other daylight state values (init and confused) effectively produce brightness and darkness values of 0. There is no mathematical reason that the brightness and darkness values for a particular daylight condition must sum to 1.0, the fact that they do for most of the conditions is simply convenience.

We also run integrators on the day and night conditions in order to keep track of how long it has been day or night respectively. The time constant on these integrators is set so the integral will reach .90 after six hours. These integrals are reset to 0 on transitions to either day or night. The values are similar to the long-term brightness and darkness averages, but they are changed only when the corresponding daytime state holds and thus are monotonic increasing until reset whereas the brightness and darkness averages retain previous state over transition, and can decrease.

6 Daytime State Transition Function

A graph showing the primary transitions of daytime state state machine is shown in Figure 2. Transitions from states to themselves and the transitions from each state to init on a reset condition are not shown. The conditions on the transition are set up as follows.
In general, we want to transition from day to night only if there has been a long enough day period, and it has now become dark. This is accomplished by requiring that the day integral be high (currently > .90 equivalent to day for more than 6 hours), the short-term dark average be high ( > 0.90 equivalent to dark for more than 7.5 minutes), and the current daylight state be dark-as-night. These fixed values would not work in Alaska, but could easily be set automatically from measurements made the previous day. The transition from night to day is handled analogously. As discussed previously, the use of temporal contrast to prime the short-term integrals during twilight periods avoids the transition delay that would result from direct use of a time-averaged value, while maintaining the advantages of smoothing out short-term variation.

If the conditions for transition from day to night on the short-term dark average and daylight condition are met, but the condition on the daytime integral is not (i.e. if it gets dark before day has lasted long enough) the system goes into the confused state. Similarly for night to day transition conditions. It stays there for the (initially short) period of time (currently about 5 minutes) that it takes a "confusion index" to decay below a threshold, and then enters the unknown daytime state where it tries to determine whether it is day or night again with a short period of observation. (Specifically the system currently waits for the confusion index to decay below 0.5 from an initial value of 1.0 with an with exponential time constant of 10 minutes) A second inconsistent observation will return the system to the confused state, but this time it lasts longer as the decay time constant doubles each time the confused state is entered (up to a limit of about 1/2 a day) This is a form of exponential backoff. The confusion time constant slowly reverts to its initial level, over a period of about a day, so that terrifying the system ultimately has no long-term effects.

The init and unknown states are transitory states, that make observations for a fairly short time in the case of startup, and return from confusion respectively, and then transition to the day or night state based on the short observation period. The system will also enter the init state if no observation has been made for a long period of time (currently 3 hours), either because of some noted failure of the image acquisition process, or the process was somehow suspended for a long time (reset condition).

In the init state, daytime state transitions to day if the short-term brightness average exceeds 0.9. The appropriate long-term brightness average and daytime integrals are set to 1.0 on transition to pretend it has
already been day a long time. Transition to night is handled analogously.

From the unknown state, daytime state transitions to day if the short-term brightness average is > 0.9 and the long-term brightness average is ≥ 0.8 (we want to be quite sure). The long-term averages and the daytime integral values are retained, while the night integral is set to 0. This permits the system to retain memory of previous daytime and remain in an appropriate state in the (most likely) case that some short-term anomalous condition caused entry into the confused state. Transition to night is analogous.

In either state, a persistent twilight condition will cause the system to persist in the state until some change occurs. To prevent indefinite hangout, we could gradually shrink the twilight zone, but this is not currently implemented, as the situation has never come up.

This state-machine has a lot of engineered complexity, and it is natural to ask if this complexity could be reduced. There are certainly principled ways of locking to a signal presumed to be mostly bimodal and periodic that might involve less specified information. However much of the complexity is associated with exception handling. The confused state is desirable, since the system could, e.g., be hit by car headlights for a while at night, or have someone cover its camera during the day, and we would rather not have it making confidant claims about it being day or night in such uncertain environments. If the system is put in an artificial light environment with no regular cycles, we would prefer it to become confused. If we were building some sort of a covert robot agent, for example, it could still creep out during dark periods if it needed to but it would pay to be a bit nervous about it. The desire to handle exceptions robustly makes a lot of the principled methods of learning state transitions (e.g. HMMs or mode-locking loops) not so principled.

7 Output

The system makes an brightness observation periodically. The default is every 20 seconds, and is the point at which the system has been run for the last year. Based on its state, it updates an output file with a description of the current day-night condition. The current output is simple, and determined as follows. If the daylight state value is bright-as-day, the system outputs "day" if the daytime state value is day, and bright-as-day otherwise. If daylight state is dark-as-night, the system outputs "night" if daytime state is night, and "dark-as-night" otherwise. If daylight state is twilight, the system outputs "twilight".

The system also reports on anomalous conditions. If the system observes a dark-as-night condition while daylight state is day and the previous observation was made within 100 seconds and generated bright-as-day, the system reports "sudden dark during the day". An analogous report is made for sudden brightness during the night. If the anomalous brightness condition persists long enough that the system becomes confused it says so, and prints and explanation, e.g. "bright at night, I’m confused”.

8 Day-Night Performance

In almost 1 year of continuous operation, the system has never erroneously classified night as day, or vice versa. No oscillations around twilight have been observed. The period includes thunderstorms producing very dark periods during the day (streetlights came on), and snowstorms producing very bright nights. The adaptive day and night models appear functionally stable, with no evidence of one cluster encroaching on another’s territory, as sometimes happens with adaptive systems. Instances of camera malfunction, sporadically producing black frames did not upset the system’s day/night determination, although it did comment on them.
9 Cloudiness Analysis

We now consider the problem of determining how much of the sky is covered by clouds. With easily available cameras, this determination is only possible during the day, hence the determination described in the previous sections is a prerequisite.

The most obvious approach is to attempt to use color to discriminate clouds from clear sky. Intuition suggests that this should be fairly simple (clear sky is blue) except near sunrise and sunset when there are dramatic red/gold and other color effects. As we shall see, it is not quite so simple, but the basic idea can be made to work reasonable well.

We start by obtaining, periodically during the day, an image of the sky at the widest possible FOV, and bottom about 5 degrees from the horizon. With sufficient resources, we could piece together a hemisphere from multiple images to obtain full sky data, but for now we consider one image at a time. This is not as restricting as it might seem, since cloud patterns move across the sky fairly quickly, so a handful of images several minutes apart produces a pretty good sample of conditions.

The first problem that arises is the sun. For typical cameras, under automatically controlled gain or not, any image containing the sun has so much light bouncing around in it that reasonable separation of clouds and sky cannot be made. The solution is to have the system detect such blinding situations and look away. Because ordinary cameras do not have the dynamic range to distinguish between saturation due to an imaged sun and more mundane causes, an active approach is needed. An initial image is taken with automatic gain control turned on. Then the gain/exposure is explicitly turned down to admit about 50% of the light, and a second image taken. This has the effect of bringing any normally bright regions below saturation. If any saturated regions remain, the size of the largest is compared to an experimentally determined threshold to determine if the sun is in the image. If so, the camera is moved left or right by an amount sufficient to move the sun out of the field of view. This works better than setting the exposure to a minimum and looking for a solar disk because thin clouds often create a situation where the sun is blinding even though a clear disk is not visible.

Once a satisfactory image is obtained, we need to consider the sources of variation in the signal. An analysis of the \((r, g, b)\) histogram data reveals that the principle axis of variation is approximately along the \(r = g = b\) diagonal. This is an intensity variation, and it does not seem strongly correlated with the clear/cloudy classification. Clouds can be both very light, and very dark. Clear sky does not vary quite so much, but the variation over the course of the day, and from near horizon to high elevations is still significant.

Given this, it makes sense to eliminate the intensity dimension, and this can be done by computing the intensity normalized values \(R = r/(r + g + b), G = g/(r + g + b), B = b(r + g + b)\) for all pixels that do not have any saturated band, and that have a least one band above some threshold (e.g. 64/255) The additional conditions avoid poorly conditioned regions of the space where the division might unduly magnify noise, and saturation artifacts. For images taken of the sky under automatic gain control, most pixels satisfy these conditions.

The normalized coordinates are redundant and we consider the \((R, B)\) subset. Looking at the distribution of pixels in \((R, B)\) space from a large collection of sky images with clouds and clear sky, reveals that the new principle axis of variation is approximately along the \(B - R\) axis, that it is several times as large as the secondary axis, and that it correlates well with the clear/cloud distinction. This confirms the intuition that hue would be a good distinguishing characteristic. We thus analyze the one-dimensional statistics of \(B - R\). A more precise axis could be computed automatically using PCA, but there would be little practical difference (in fact just using \(B\) would work almost as well)
10 Clear-Cloudy Model

As in the case for day and night, we model the 1-D $B - R$ distribution using a Gaussian mixture model, where one Gaussian component presumably corresponds to cloudy sky, and another to clear sky.

An initial clear-cloudy model is obtained from a sample of sky imagery that contains significant amounts of both clear and cloudy regions. Generally we use more than one image to obtain a sample. The initial training imagery can either be provided a-priori, or obtained by the system during a startup period after a day-night model has been established.

Once a sample representing a presumably bimodal $B - R$ distribution is provided, a clear-cloudy model is generated as follows. First the distribution is smoothed with a small Gaussian filter (std = 2.0 digitizer units). The eliminates spikes in the distribution due to digitizer posterization effects. Then initial values for the mean and variance of a two component Gaussian mixture model are guessed using the same heuristic employed for the day/night analysis. The EM algorithm is then run to adjust the parameters. Finally, we run a bimodality test on the result. If the mass ratio the two components is greater than a factor of 5 either way, or the peaks are separated by less than 4 digitizer units (on a 0-256 scale), we conclude that the sample was not bimodal, and return with failure.

We now take another step to refine the models. Unlike the case for day and night brightness, a sample of sky imagery generally contains a significant number of pixels that are not clearly sky or cloud. This population arises from hazy regions, edges of clouds, etc. The EM algorithm run with two Gaussian models will typically will assign these pixels to one model or the other, resulting in an overly high variance for that component (which one it goes with depends on irrelevant details of the distribution). To eliminate this problem, we run EM using a triple Gaussian model, with a new model initialized halfway between the two dual-Gaussian peaks with a relatively low weight (.15). This successfully picks up the hazy component, producing much more consistent values for the clear and cloudy components. The model parameters are stored, along with a normalized representation of the generating distribution just as for the day-night case.

Also as with the day-night model, the clear-cloudy model is updated on a daily basis during the REM-sleep period. The update technique is based on a leaky integrator update of the underlying $B - R$ distribution using sky images obtained during the most recent day. The following constraints are observed. First, we require that a minimum number (10) of sky images have been observed that do not look “peculiar” in a sense that will be described later. Second, since there are frequently days which are entirely cloudy (or entirely clear), the daily distribution must pass a bimodality test before any update is attempted. Specifically, the modeling process must succeed on the daily distribution, and the clear and cloudy components computed must both have a weight $> 0.25$. If the daily data distribution fails the bimodality test, we back off to distribution derived from images that individually passed a bimodality test (dual gaussian fit has both components with weight $> 0.25$, and the separation of the peaks is larger than half the model separation), if there were at least 4 such images during the day. This distribution must also pass the bimodality test. Finally, the mixed distribution (.75 old .25 new) must generate a satisfactory bimodal model. If the model generation fails, we just retain the old model.

Figure 3 shows the $B - R$ histogram generated by this process as of late August 2006. It is definitely bimodal, but unlike the day-night brightness, the peaks overlap significantly. On the plus side, there are no subsidiary peaks due to automatic exposure control since normalized color coordinates are used.

Table 2 shows the history of the clear/cloudy model parameters over time. The parameters appear fairly stable; somewhat more so than the day/night model parameters that are similarly maintained. Note that the weights do not sum to 1.0 due to the presence of the third Gaussian model in the mixture.
11 Sky image Analysis

As mentioned above, the system periodically obtains an image of the sky (once every 15 minutes currently). This image is analyzed to make a semantic determination of the sky conditions, and to find proportions of the sky that are clear, cloudy or hazy.

The semantic determination is specified by a state variable **cloudiness state**. It includes states indicating that analysis cannot be carried out for various reasons. These include *unknown* indicating that the system has not even established day or night, *too-dark* indicating that it is night, or too dark to get a good picture of the sky, and *near-dawn* which is returned in pre- or near-sunrise conditions when odd sunrise color effects are likely to be present in the east (which is the direction the camera is constrained to look). A *near-dusk* condition would be employed for a camera that could only look toward the west.

If the sky image is suitable for analysis, then one of the states *clear*, *mostly-clear*, *partly-cloudy*, *mostly-cloudy*, *cloudy*, *hazy* is established. A final state *strange* is set if most of the pixels are saturated or too
The analysis starts by accumulating a histogram of the normalized $B - R$ values for all “good” pixels - that is those that have no saturated band, and at least one band that is not dark ($> 64$ of 256 levels). We then classify these pixels as “clear”, “haze”, or “cloud”. We also accumulate a count of “red” pixels. These are pixels whose $B - R$ value is far to the red side of the cloud component mean. These typically arise in images of sunrises and sunsets. For now we include them as a subset of the cloud pixels. The count is used to make appropriate comments about sunrises and sunsets.

### 11.0.1 Computation of the Blue Shift

Unfortunately, the classification cannot be made simply on the basis of comparing the pixel $B - R$ value to the cloudy and clear component models. The most significant complicating factor is what we call the “blue shift”. This is an atmospheric effect typically occurring within an hour of sunrise and sunset, where cloudy skies can appear very blue. It is noticeable even to humans with their extremely good color constancy abilities. The effect is due to the fact that when the sun is low in the sky the direct illumination of the clouds is oblique, and the illumination of a cloud deck is increasingly due to light scattered from the atmosphere at high altitudes, which is extremely blue. In fact, just after sunset, the $B - R$ value of a cloudy sky can exceed that of a clear sky at noon. Since the light from clouds has a very narrow distribution in the normalized blue-red chromaticity dimension we use for analyzing the sky, this blue shift can cause classification errors when a fixed model is used.

The blue shift primarily affects the mean of the cloud component. It does not change the variance significantly, which remains considerably smaller than the clear sky variance. Clear sky does not suffer much of a blue shift, and the effect is mainly to increase the cloud mean, bringing it closer to the clear mean in near-dusk skies.

The clear-cloudy model describes non-blue-shifted clear and cloud components. In order to accurately classify the pixels, we need to estimate what blue shift, if any is affecting the cloud regions of the image. Note that if it can be established that a certain population of pixels is cloud, then the blue shift can be obtained just by comparing the mean of that population to that of the cloud model component. Consequently, we start by fitting a dual Gaussian model to the $B - R$ values in the image just as in the original model generation process. The result is subjected to a bimodality test requiring that the ratio of the component weights be no more than 5 one way or another, and that the separation of the peaks be at least $1/4$ the separation of the clear and cloud model components.

If the sample is unimodal, we need to decide whether it is due to a clear or cloudy sky. We first recompute the mean and variance for a single mode model. We then step through a number of indications of cloudy skies to make the required determination. We first check the location of the peak. If it is more than 3 clear model standard deviations below the clear model mean, we take the sky to be cloudy, since we have not observed an extreme red-shift of a clear sky. We then check the overall vertical gradient of the image, computed by subtracting the bottom half from the top half. Due to the physics of atmospheric scattering, clear skies are brighter toward the horizon. Thus a unimodal sky that is darker toward the horizon is almost always an indication of a cloudy sky (the converse is not true). Finally we note that overcast conditions have very low chromatic variation. Thus if the standard deviation of the peak is small (less than about 80% of the cloud model $B - R$ std, which is about 5 digitizer units for a 256 level signal for our sensor), we take the sky to be cloudy. Otherwise we take the sky to be clear.

For a bimodal distribution, we first check to see if this is due to a hazy horizon skewing a clear sky signal (this is a frequent occurrence in summer, especially in urban areas). This situation is indicated by a strong overall vertical gradient in the direction of a bright horizon, combined with a lack of any texture that would be produced by non-uniform cloudiness - the other situation that can produce a bright horizon. The latter condition is tested by requiring that the vertical gradient be essentially monotonic (of the same sign) over the
image. This is evaluated by dividing the image into 8 horizontal strips, and computing the difference (local gradient) between all adjacent pairs. We compute a monotonicity index by returning the minimum magnitude strip gradient if the max and min are of the same sign, and the negative of the minimum magnitude if the max and min are of opposite signs. This is positive for globally monotonic gradients, negative for non-monotonic ones, and has well-behaved transition properties.

If a bimodal sky has monotonic gradient, we conclude that the sky is clear. Otherwise we conclude that we have an image of mixed clear and cloudy regions, and proceed as in the model formation process to fit a triple gaussian mixture to the distribution. Again, the intent is to prevent intermediate haze from contaminating one or the other peak. If the peak with the lowest $B - R$ mean is significant (weight greater than 0.1) we take it as representing the clouds. Otherwise we take the mean of the central peak.

So now we either have a clear sky determination, or the $B - R$ value of the cloud peak. If the sky is clear, we return a blue shift of 0. If we have a cloudy peak, we compute a provisional blue shift as the difference between the current cloud peak and the cloud model mean. This is returned subject to the following condition. Recall that large blue shifts occur only close to sunrise and sunset. Prohibiting large blue shifts outside these periods eliminates some potential errors. From empirical observation, we set a daytime window from about 1.5 hours after daybreak, until 1.5 hours before nightfall. Daybreak we know from the daylight observation module. Nightfall we can predict on the basis of the previous day’s observations. Within this daytime window we limit the blue shift to one third of the difference between the model cloud and clear peaks. A softer threshold would probably be appropriate, but we have not collected sufficient statistics to fit a function. In any case, the condition is triggered only rarely by odd conditions during the day (usually direct sun glare off the optical system) that trick the system into thinking that a clear sky has cloud structure in it.

### 11.1 Cloudiness Pixel Classification

Once we have a blue shift value for the clouds, we proceed to classify “good” pixels (the ones not too dark or saturated) as follows. First, determine an upper cloudiness $B - R$ bound. Recall that the cloud model tends to have a very narrow variation. We ensure that the cloudiness class extends at least 3 standard deviations from the blue-shifted cloudiness mean, and no less than 1/3 of the way toward the clear peak. Note that if the blue-shifted cloudy mean exceeds the clear mean, we just get 3 stds.

We then determine a lower clear boundary. We try to allow two standard deviations below the clear model peak, but don’t let the boundary extend more than half way to the shifted cloudy peak. In the rare case that the blue-shifted cloudy peak exceeds the clear model mean, we just set the clear lower boundary to the cloud upper bound plus one digitizer unit (of 256). We note that splitting the interval between the peaks into three equal units works almost as well as the complex condition employing the peak widths.

Finally, we set a “red” boundary well below the range of the cloud model (three cloud standard deviations currently). This condition ensures that virtually no “normal” pixels end up in the “red” class (two standard deviations would get a couple of percent). “Red” pixels actually include yellow, orange, etc. - anything that does not look like blue sky or cloud on the $B - R$ axis.

Pixel classification then proceeds as follows: Any pixel with a $B - R$ value below the red threshold is classified as “red”. Pixels between the red upper bound and the cloud upper bound are classified as “cloud”. Pixels between the cloud upper bound and the clear lower bound are classified as “haze”. And pixels above the clear lower bound are classified as “clear”. Counters are maintained for all the classes, with “red” pixels also contributing to the cloud counter. A $B - R$ histogram is also incremented as the image is processed, with any “red” pixels excluded.

If enough “good” pixels were found (> 1000 currently) we update the daily statistics and then proceed to the semantic classification phase. The daily statistics are incremented with the image $B - R$ histogram if the image is not “peculiar”, or if no cloudiness model yet exists. Currently the requirement is that the blue
shift not be too extreme, specifically, that it be less than 3 cloud model standard deviations. If it is also true that the image seems to be bimodal (both clear and cloud components with weight $> .25\%$, and the peaks separated by more than 1/2 the separation of the model peaks), then the daily bimodal statistics (the fallback distribution) are also incremented with the image $B - R$ histogram.

11.2 Cloudiness Semantic Classification

The semantic classification is based on the relative proportions of clear, cloudy, and hazy pixels. These sum to 1.0, and the partition is most easily described graphically by a classic ternary 3-alloy diagram. This is shown below in Figure 4. Note that even though the “cloudy” concept is the smallest in area, it is the most often observed in Rochester NY.

![Figure 4: Illustration of the zones defining the various semantic cloudiness concepts.](image)

The boundaries are somewhat arbitrary, and chosen for reasonable intuitive performance. They could easily be adjusted to match a formal specification, the main difficulty being that most of the formal meteorological definitions do not involve a hazy component. The basic principles we used to generate the zones are as follows

1. All regions are convex.
2. Clear or cloudy require $> 90\%$ of the image to be interpretable as clear or cloudy.
3. Mostly clear or cloudy require that $> 70\%$ and $< 90\%$ of the image be so interpretable.
4. Partly cloudy is all situations where between 30\% and 70\% of the image is cloudy.
5. Hazy is all situation where $> 50\%$ of the pixels are hazy and

6. For clear and mostly clear classifications, hazy pixels are interpretable as clear if less than $50\%$ of the pixels are classified as hazy.

In addition to the primary classification based on the mixture of pixels types, the system adds occasional high-level comments. If the initial classification is cloudy, but the sun has been directly observed, the system reports “hazy sun with light overcast”. An active search for the sun would make this quite reliable, but with only one window to look out, the system can find the sun less than half the day. The system also comments on “red sky in morning” and “nice sunset” if it finds large numbers of “red” pixels near dawn or dusk.

12 Sky Analysis Performance

13 Cloudiness Analysis

Overall the system performs quite well, and typically generates semantic classifications that seem reasonable to a human observer. The “hazy” category hardly ever occurs, even though it only requires $50\%$ haze pixels. This is due to the dynamic adjustment of the model through the blue-shift mechanism. Much more reliable is the “hazy sun” comment that the system makes when the sky color appears overcast, but the sun has been observed. This actually seems to be consistent with how human observers make the same judgement. A quantitative evaluation is difficult to make, as humans don’t always agree closely on the percentages of sky and clouds making up an image due to fuzzy cloud boundaries. If anything, the system appears to have a slight tendency to overestimate the amount of cloudiness.

We have identified two primary sources of real mistakes. The first arises from sun glare in the early morning, when the system can still generate a high blue-shift estimate. What happens, is that the camera cannot point quite far enough away from the sun to avoid glare in the image; either from internal reflections, or from diffuse light haze near the sun. This can produce a bright zone near the top of a clear sky image, inverting the expected clear-sky gradient, resulting in a determination that the sky is overcast, and an erroneous computation of a high blue shift. In cases where the sun was actually detected, we circumvent the problem by reporting a hazy sun condition, but occasionally the glare is present without the sun having been in the original image. A better strategy would be to actively look for the sun in every sky observation, so that that information is available to the semantic classification process. This would not be too hard to implement, as we know the time since daybreak and can predict the likely location of the sun from previous days’ observations. A similar error occurs occasionally with diffuse jet trails in clear skies that can similarly invert the gradient or monotonicity measures without providing enough of a real cloud signature to be separable from the blue sky.

The second source of error involves mixed cloudiness in highly blue-shifted dusk conditions. In these situations, the $B − R$ values of clear and cloud regions are very close, and the distributions overlap almost completely. Human observers can still distinguish sky from clouds, but the distinction appears to be based on a secondary chromatic characteristic. Subjectively, the clear regions in such conditions are greener than the cloudy ones. This information is not present in our 1-D $B − R$ signal. Retaining the secondary chromatic axis would probably allow the situation to be resolved, but it would significantly increase the complexity of the modeling process to correct an error that occurs quite infrequently.

References

