# Toward Fully Automatic Geo-Location and Geo-Orientation of Static Outdoor Cameras

Nathan Jacobs, Nathaniel Roman, and Robert Pless
Department of Computer Science and Engineering
Washington University, St. Louis, MO, USA
{jacobsn,ngr1,pless}@cse.wustl.edu

## Abstract

*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 rely on access to accurately time-stamped images captured over time from cameras that do not move. To support these algorithms we also offer a method of camera viewpoint change detection, or "tamper detection", which determines if a camera has moved in the challenging case when images are only captured every half hour. These algorithms are tested on a subset of the AMOS (Archive of Many Outdoor Scenes) database, and we present preliminary results that highlight the promise of these approaches.*

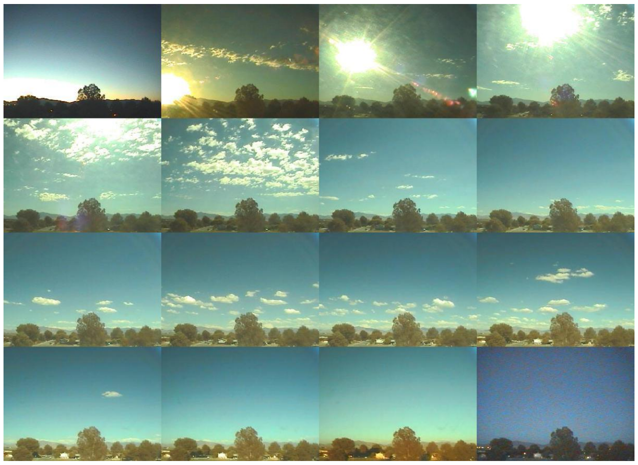


Figure 1. We show how to geo-locate and geo-orient a static camera using natural cues from the diurnal cycle.

## 1. Introduction

There is a large and growing global imaging network that comprises tens of thousands of cameras connected to the Internet. Some of these cameras are part of large, coherent sensor networks [1, 2], while a broader and more diverse set can be discovered by searching the Internet [3]. To integrate these cameras into a useful imaging system, one basic requirement is to determine the location and calibration of these cameras. Since there are many cameras, and since cameras may be replaced, moved, changed and re-purposed at the whim of whomever owns the camera, it is important that this process is automatic.

A recent study has found that there are important similarities in images taken by nearly all static outdoor cameras. In particular, the PCA decomposition of images from each camera creates image components (which are scene dependent), and coefficients (whose daily pattern of variation are nearly independent of the scene) [9]. In this work we propose to use these regularities to support partial camera calibration. Additionally, since these statistics rely on the fact that the camera does not move, we also present a novel, robust, and efficient mechanism for the rather prosaic task of determining whether a camera has moved (as opposed to becoming partially occluded by a spider, seeing an unusually bright sunset, etc.). This problem is challenging when there is a need for high accuracy and when there is a long gap (tens of minutes) between consecutive images, as is often then case in remote sensing image capture.

The camera calibration approach is based on three steps. First, we find a sequence of images where the camera has not moved. Second, we implement a recently reported algorithm to compute a "canonical day" decomposition (similar to a PCA decomposition, but consistent across multiple cameras). This decomposition supports automatic camera geo-location, as well as robust and automatic sky segmentation. Third, we use a graphics tool which creates a full hemispherical sky intensity map for a given geo-location and time of day. An efficient optimization process searches to find 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.

This preliminary work has several limitations. First, we optimize only over the camera orientation and the field of view, rather than the complete calibration matrix. Thus, our algorithm is more "camera orientation" rather than camera calibration. Second, we use the recently published AMOS dataset [9] which includes camera location information for some cameras, but camera calibration and camera orientation for none. Thus, we provide qualitative results for just a few cameras for which the scene provided cues that allow us to estimate the true camera orientation by hand.

## 2. 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 case where there are feature correspondences between multiple cameras [11, 4]. Larger scale camera localization is less well studied, but has been addressed by matching features in the image to features computed from a digital elevation map [18, 7, 17].

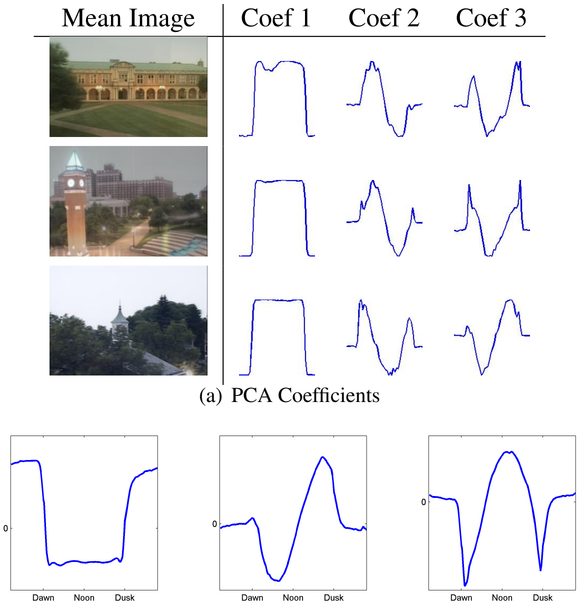
In distributed camera networks, distributed calibration algorithms have been proposed and implemented [8, 12], and different types of correspondence cues such as object tracks [13], and timing correlations of when objects enter and exit the camera views [19]. These allow inference of camera locations when the camera fields of view do not overlap, but require that the cameras are nearby and see the same objects.

Aside from work to be discussed later [10], previous work on global camera geo-location using natural scene variations is based on explicit measurements of the sun position [6], which was followed by work in computing absolute camera orientation [20]. These techniques require the sun to be in the field of view and accurate camera calibration to determine the angle of the sun.

## 3. Consistent Natural Variations for Sky Segmentation

The consistent causes of image variations in static outdoor cameras are the diurnal cycle and the weather. Recently, it was found that even if cameras view different scenes, there are consistent patterns to how these images vary over time. In particular, the PCA decomposition of images from each camera creates image components (which are scene dependent), and coefficients (whose daily pattern of variation are nearly independent of the scene) [9]. Creating methods that use these coefficients to annotate cameras and scenes eliminates any camera- specific feature specification.

The data from each camera can be summarized as a data matrix $I \in \mathbb{R}^{p \times T}$ where each column is an image of $p$ pixels of the same scene at time $t$. Singular Value Decomposition decomposes this matrix as $I = U \Sigma V^{\mathsf{T}}$, where the



(a) PCA Coefficients

(b) Coef 1

(c) Coef 2

(d) Coef 3

Figure 2. The first three canonical component coefficients learned from the AMOS dataset.

columns of $U$ are the principal components and the columns of $V$ (we use the first three columns for all experiments) are the time-series of principal component coefficients. A large scale statistical study of 538 natural outdoor scenes [9] finds that the matrix of components $U$ and singular value matrix $\Sigma$ are scene dependent but the matrix of coefficients $V$ is much less so. Figure 2 shows examples of coefficient trajectories (i.e, the leading columns of $V$) for a single day for several different cameras.

This previous work determined a diurnal "canonical coefficient basis" $V'$ which is consistent across all outdoor cameras and remains nearly optimal at encoding scene variations. Here we use this natural diurnal basis to construct the corresponding diurnal components for each image data set. In particular, we decompose the images for each camera $k$ as $I_k = U_k \Sigma_k {V'}^{\mathsf{T}}$. By forcing each camera to use the same coefficients over the day, the components are readily comparable. This allows the use of a simple threshold (and the same threshold for all cameras) to classify pixels as sky when its value in this first diurnal component is much stronger than its value in subsequent components. While segmenting sky is not a particularly challenging task, it is important that it is robust and automatic. For the purpose of detecting camera orientation, it is only necessary that the set of pixels labeled sky are correct — if some pixels are sky but are labeled otherwise this results only in a slight loss of data. In any case, our initial result are encouraging; Figure 3 shows our pixel classification on four representative cameras.

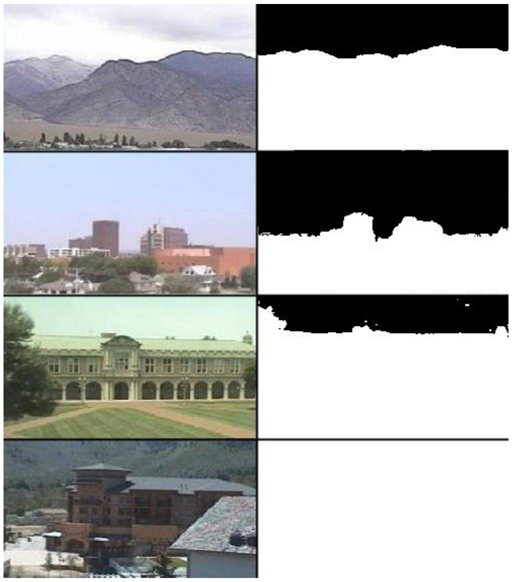


Figure 3. Automatic segmentation of the sky based on canonical components.

## 4. Camera Motion Detection (Tamper Detection)

Detecting a change of camera viewpoint is a critical preprocessing step to our geo-localization and geo-orientation algorithms. They both require images from the same viewpoint. In our problem domain, very low-frame rate image sequences, existing methods of detecting camera tampering [16] are insufficient. The key challenge in detecting camera viewpoint change for very low-frame rate video is that a useful algorithm must be invariant to many factors, such as weather, lighting, and activity in the scene.

Mutual information provides an efficient metric for the difference between image frames that is, at least, partially invariant to these factors. However, changes between frames, especially around dawn and dusk, may be too drastic to have high mutual information between consecutive frames due to the extremely low frame rate (1 frame per 30 minutes) of the data. To minimize the effect of such lighting changes, we do not directly compare consecutive images; instead, we compare each image with an image captured at the same time on the previous day.

The mutual information between pairs of images captured at the same time on consecutive days is highly dependent on the time of day the images were captured. For example, images captured during the night are often dominated by noise and therefore have low mutual information. To reduce the number of false positives we normalize the current mutual information by the expected mutual information learned from previous days. We normalize by treating the mutual information as a Gaussian random variable conditioned on the time of day and computing the z-score $z_t$ of the mutual information of the new image pair. We estimate the distribution parameters using the most recent 10 days of images.

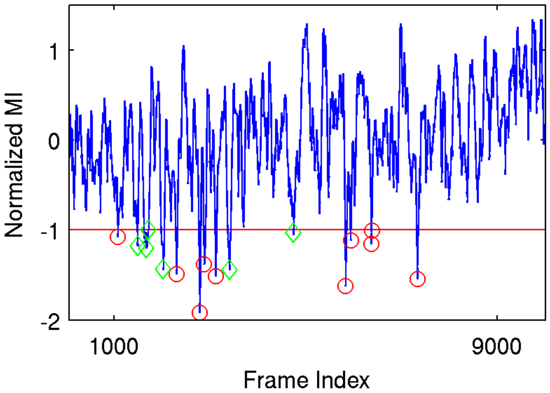


Figure 4. The normalized mutual information score from each image to an image from the same time the previous day, for 10000 images taken over approximately 200 days. For image indices whose score falls below the -1 threshold SIFT and RANSAC are used to compute a homography with the previous frame. When this motion is small, the index is marked with a circle and considered not to have moved, when the homography corresponds to a non-trivial motion, the index is marked with a diamond and corresponds to a viewpoint shift.

We compute a moving average $\bar{z}_t$ of the z-scores for an entire day to further reduce the chance that a spurious low z-score will result in a possible camera shift. We note that if a viewpoint shift occurs the z-scores will be low for a full day following the change. The smoothing process will not hide true viewpoint shifts. We consider a smoothed z-score $\bar{z}_t$ below -1 (i.e., low mutual information) to reflect a candidate viewpoint shift. After non-minima suppression the remaining candidate image pairs are passed to a more computationally expensive image alignment process to determine if they are true viewpoint shifts.

We compute SIFT features on each candidate image and the immediately preceding frame. We then use RANSAC to compute the homography between the image pair. If the difference between the homography and the identity transform is small (i.e., it corresponds to a maximum pixel motion of less than 2 pixels) then we do not consider this candidate pair a true viewpoint shift, otherwise we do. Note that if the image change is dramatic, such as having no overlap between the two images, the chance of the best RANSAC result being similar to the identity matrix is negligible.

Viewpoint shift detection results generated using this procedure are presented in Figure 4. Tests were performed on a set of 10000 images from 200 days from AMOS camera 4. This set of images contains significant non-motion-related image changes (e.g., snow and stray reflected light). Examples of successes and failures are shown in Figure 5.

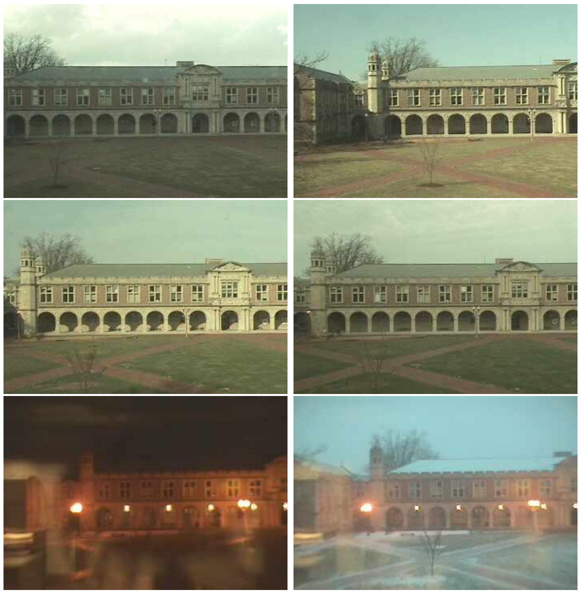
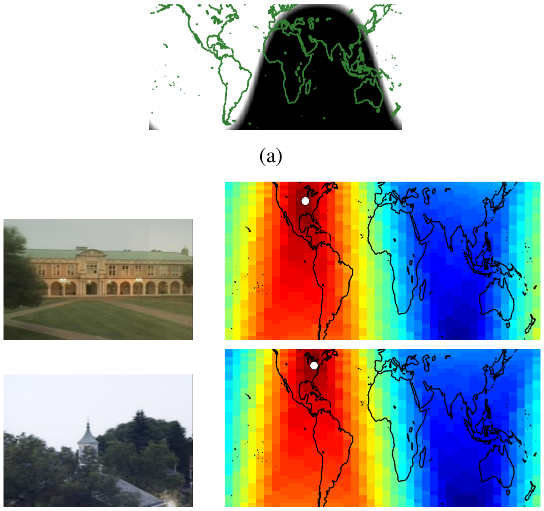


Figure 5. (top) Example pair of subsequent images where the algorithm correctly detected a viewpoint shift. (middle) Example pair of images where the normalized mutual information score was low, but the SIFT feature matching indicated that there was no camera motion. (bottom) this pair of images is representative of the failures of this algorithm observed in these 10000 frames. These were incorrectly marked as a viewpoint shift, in part because the reflection off the window through which the camera views the scene.

This algorithm is intended to detect viewpoint changes in cameras that are mostly static. A limitation of the algorithm presented is that if more than one viewpoint change takes place during a single day, then it will most likely only detect one of the changes. We feel that this is not a significant hindrance because any camera that moves that often is not sufficiently passive to work with our geo-localization and geo-orientation algorithms.

## 5. Camera Localization

Our method for camera orientation requires an estimate of the geo-location of the camera. We implement a recently reported method for camera localization [10]. This algorithm creates a synthetic daylight map over the entire earth, where intensities correspond to the amount of sunlight (see Figure 6 for 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. These synthetic daylight maps are created for the times when each image is captured. The correlation between the synthetic images and the first canonical coefficient trajectory is computed. Examples of correlation maps generated using this method are shown in Figure 6. A related method which uses satellite imagery instead of the synthetic daylight imagery was reported to give better accuracy [10], but we found fewer large errors in correlations with synthetic daylight images.



(a)

Figure 6. A single synthetic day-night satellite image (top), and the correlation of the first canonical PCA coefficient with pixels from a sequence of such images. The region with the highest correlation corresponds to the location of the camera (white dot).

## 6. Camera Orientation

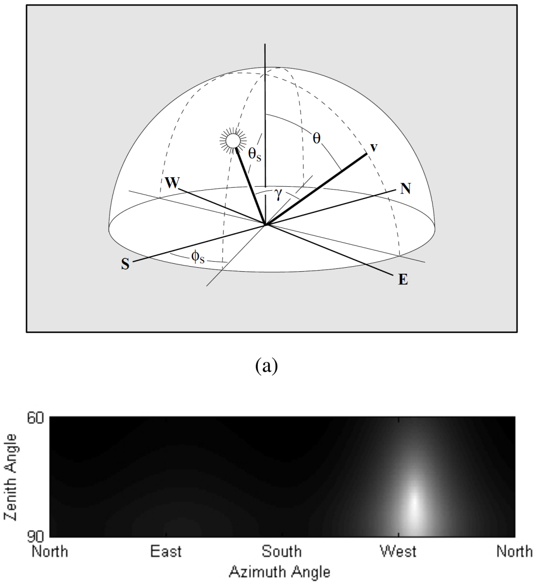
Given the location of the camera and time-stamped images from the camera it is possible to geo-orient the camera. We determine the azimuth angle (degrees eastward from the north) by comparing the sky pixels (see Section 3) of many images from the camera with predictions of sky luminance based on an analytical model.

We choose the simple analytical model of sky luminance [14]) that has been adopted as an International Commission on Illumination (CIE) standard [5]. The model predicts the luminance $L_v$ of a sky position on a clear day:

$$L_v = L_z \frac{(0.91 + 10e^{-3\gamma} + 0.45\, cos^2\gamma)G(\theta)}{(0.91 + 10e^{-3\theta_s} + 0.45\, cos^2\theta_s)G(0)} \tag{1}$$

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 7. The sun position is specified by the azimuth $\theta_s$ (degrees eastward from north) and zenith angles $\phi_s$ (angle from vertical). The sky direction is specified by the zenith angle $\theta$ and the angular distance $\gamma$ 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 [15] using



(a)

(b) Synthetic Luminance Map

Figure 7. (a) An illustration of the parameters of the analytical sky luminance model described in Section 6 (Figure from [14]). (b) An example synthetic sky luminance map generated by the analytical model.

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 1.

For each hypothesized focal length, we re-sample the image to a size so that each pixel corresponds to a subtended angle of 0.3 degrees (the same as the generated the 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 with a given sky position. This score needs to be computed and averaged for images at several different times of day. 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.

To evaluate this method we estimated the orientation of a small set of cameras (from the AMOS data set, cameras 4, 330, 652. An image from Camera 4 is shown in this paper in Fig 5, images from camera 652 are shown Figure 1). The results are within 5 degrees in azimuth angles from handcrafted estimates based on explicit reasoning about the sun position and matching image features to satellite imagery.

Although automatic techniques for determining if images from static cameras have clear skies exist [9], 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 8 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 limited to images that don't show the sun (Figure 8, second from the top).

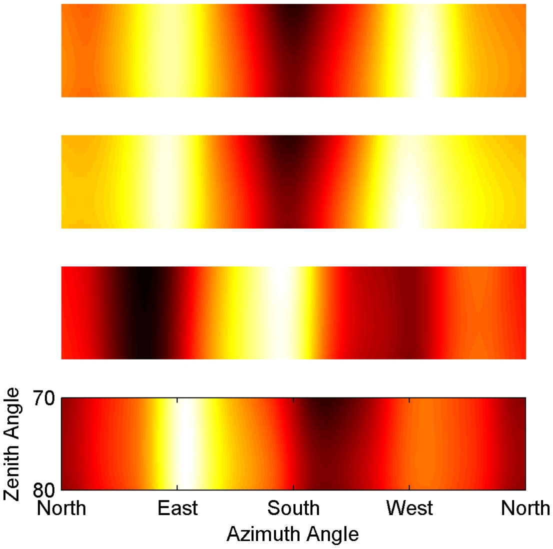


Figure 8. These images show the scores obtained by our camera orientation estimation procedure. The color reflects how likely the camera is pointing in a given direction (white means more likely). The top two images are for the same eastward facing camera. The top was estimated using images with the 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. The bottom image sees a large portion of the sky and directly views the sun.

To simplify and make efficient our optimization, we make several assumptions about the camera calibration. First, we assume that the camera has square pixels, no radial distortion, and the principle point occurs in the middle of the image. Furthermore, we recognize that mapping the image (as a rectangle) onto the sky-luminance image (calculated in azimuth zenith coordinates) is geometrically incorrect, Since most images have only a slow gradient across the image, we believe the speed benefits of computing the NCC score with a convolution outweighed the loss in accuracy, although this is a subject of further investigation.

In addition to geometric calibration the radiometric calibration is also important. The cameras we use are not under our control, therefore we do not have access to the radiometric calibration. The use of NCC to compute similarity between sky pixels and orientations provides some robustness to automatic exposure correction. Full knowledge of the radiometric calibration would improve the accuracy of our results.

## 7. Discussion

This paper presents preliminary results for an algorithm that computes the orientation of an outdoor camera based solely on image data. This includes an algorithm for camera tamper detection (detecting changes in image orientation), which is a necessary precursor to algorithms that exploit regularities in scene appearance over long time periods. When cameras are static, the proposed algorithm in this paper and recent previous work suggest it is possible to robustly geo-locate a camera to within 50 miles and geo-orient a camera to within about 5 degrees.

Within the domain of geo-orienting a camera, existing data sets do not yet support quantitative evaluation of these algorithms. Further research should also be directed to the stronger geometric constraints available from image gradients (rather the whole image cross-correlation). These may provide enough data to support effective estimation of the focal length, and may be naturally combined to simultaneously solve for geo-location and geo-orientation.

## References

[1] http://weatherbugmedia.com/.

[2] http://www.hazecam.net/.

[3] http://www.opentopia.com/hiddencam.php.

[4] P. Baker and Y. Aloimonos. Calibration of a multi-camera network. In *Omnivis 2003: Omnidirectional Vision and Camera Networks*, 2003.

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

[6] F. Cozman and E. 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.

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

[8] D. Devarajan, R. J. Radke, and H. Chung. Distributed metric calibration of ad hoc camera networks. *TOSN*, 2(3):380–403, 2006.

[9] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis,MN, June 2007.

[10] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *Proc. IEEE International Conference on Computer Vision*, Rio De Janiero, Brazil, Oct. 2007.

[11] J. Jannotti and J. Mao. Distributed calibration of smart cameras. In *Workshop on Distributed Smart Cameras*, 2006.

[12] W. Mantzel, H. Choi, and R. Baraniuk. Distributed camera network localization. In *Proc. Signals, Systems and Computers*, volume 2, 2004.

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

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

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

[16] E. Ribnick, S. Atev, O. Masoud, N. Papanikolopoulos, and R. Voyles. Real-time detection of camera tampering. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Washington, DC, USA, 2006. IEEE Computer Society.

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

[18] W. Thompson, T. Henderson, T. Colvin, L. Dick, and C. Valiquette. Vision-based localization. In *ARPA Image Understanding Workshop*, pages 491–498, Washington D.C., 1993.

[19] K. Tieu, G. Dalley, and W. E. L. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Proc. IEEE International Conference on Computer Vision*, pages 1842–1849, 2005.

[20] A. Trebi-Ollennu, T. Huntsberger, Y. Cheng, E. T. Baumgartner, B. Kennedy, and P. Schenker. Design and analysis of a sun sensor for planetary rover absolute heading detection. *IEEE Trans. on Robotics and Automation*, 17(6), 2001.