Cloud Motion as a Calibration Cue

Nathan Jacobs, Mohammad T. Islam, Scott Workman
Computer Science, University of Kentucky
{jacobs,mtis222,smwork3}@cs.uky.edu

Abstract

We propose cloud motion as a natural scene cue that enables geometric calibration of static outdoor cameras. This work introduces several new methods that use observations of an outdoor scene over days and weeks to estimate radial distortion, focal length and geo-orientation. Cloud-based cues provide strong constraints and are an important alternative to methods that require specific forms of static scene geometry or clear sky conditions. Our method makes simple assumptions about cloud motion and builds upon previous work on motion-based and line-based calibration. We show results on real scenes that highlight the effectiveness of our proposed methods.

1. Introduction

Today, cameras are easy to deploy and connect to the Internet. This has led to a profusion of outdoor webcams and a glut of webcam imagery. This data has many promising applications ranging from environmental monitoring [11] to detecting abnormal events [5]. To maximize the value of webcams for such applications, we need to know the location, orientation, focal length and, more generally, the calibration of each camera. This is challenging because often we only have time-stamped imagery from the camera and therefore traditional calibration approaches that rely on calibration targets or multiple views are not appropriate.

Recent interest in webcam imagery has sparked numerous methods for the unattended camera calibration problem, which rely on a variety of cues including sun position [8], time-varying scene appearance [13], moving shadows cast by static scene elements [6], shadows cast by moving clouds [10], sky appearance [12, 15], photometric image formation [20, 14] and dynamic textures [19]. Each of these methods makes assumptions about the scene and then solves for the camera calibration parameters that best fit the observed image data. We take the same approach.

We propose cloud motion as a new cue for static camera calibration. We assume cloud motion is generally a horizontal translation and show how to use video of moving clouds to estimate the radial distortion, horizon line, focal length and geo-orientation of the camera (see Fig. 1). The cloud motion cue is suitable for scenes in which a substantial amount of sky is visible, but does not require any particular static scene structure or direct camera access. We view the cloud motion cue as complementary to previous work that explored other geometric calibration cues. We provide evaluation on real scenes to highlight the strengths and weaknesses of this cue and our methods.

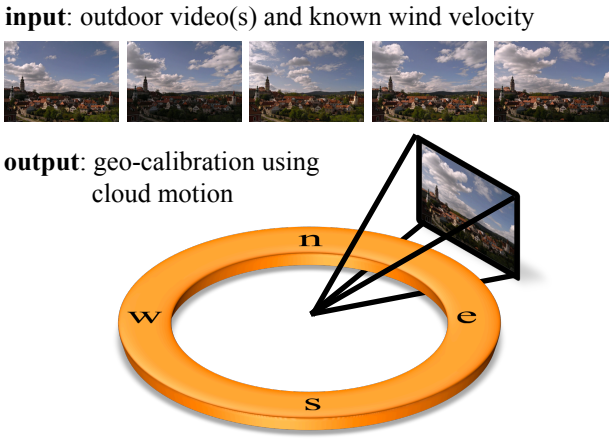


Figure 1: We propose to use natural cues provided by cloud motion to calibrate static outdoor cameras. This will support expanded use of uncalibrated imagery sources, such as webcams, in applications ranging from environmental monitoring to security.

Overview of Approach We begin with a set of videos, each captured from the same static outdoor camera, ideally viewing a significant portion of the sky. We assume that we are given a sky mask, either manually constructed or automatically estimated [16]. We aggregate motion statistics separately for sky regions in each video (Sec. 2). If radial distortion estimation is necessary, we estimate per-pixel flow vectors, fit streamlines, and use existing line-based techniques to estimate distortion parameters (Sec. 2.1). For

each video, we estimate the vanishing point of the cloud motion in a collection of videos captured on different days (Sec. 2.2). We then combine these individual wind estimates in various ways to estimate camera calibration parameters. Given multiple days of video, with different wind directions, we can estimate the horizon line (Sec. 3.1). When the camera location and time-stamp is known for each video, we use publicly available wind velocity data to estimate the focal length of the camera and the pan, tilt and roll of the camera in geographic coordinates (Sec. 3.2).

2. Robustly Estimating Cloud Motion

Key to our method is computing estimates of the motion of the clouds. We explored both sparse and dense methods that compare consecutive frames and found them simultaneously unreliable and slow. Instead, we aggregate information over the entire video before estimating motion. In this section, we provide background on how we aggregate motion information and describe our methods for estimating the radial distortion and the vanishing point of the cloud motion.

We begin with the *brightness constancy constraint* equation, $I_x x_p + I_y y_p + I_t = 0$, which leads to an ill-posed problem with a single constraint and two unknowns per pixel, $\mathbf{p}$. To overcome this, one typically assumes locally constant motion; to estimate the motion, you spatially aggregate the spatio-temporal derivatives and solve for the motion as $[x_p, y_p]^\mathsf{T} = \mathbf{A}_p^{-1}\mathbf{b}_p$, where the structure tensor, $\mathbf{A}$, and the gradient-weighted residual vector, $\mathbf{b}$, are defined as follows:

$$\mathbf{A}_p = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad b_p = \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}.$$

In our case, we assume a unique, constant motion for each pixel for an entire video, therefore we integrate over time instead of space [23]. The six numbers in $\mathbf{A}_p$ and $\mathbf{b}_p$ summarize the motion at each pixel, are very fast to compute and form the foundation for all of our calibration methods. To estimate the average per-pixel optical flow for a single video sequence we can use the equation defined above; we use this for estimating radial distortion and visualization purposes.

2.1. Radial Distortion Estimation

We propose to estimate radial distortion of an outdoor camera using our assumption of translational cloud motion. We compute per-pixel optical flow estimates using the method described in the previous section and smooth them using a Gaussian filter ($\sigma = 7$). We then use an off-the-shelf technique to estimate streamlines [17], which are curves that are tangent to the flow. If our assumption on translational motion holds, this results in a set of image-space curves that are the projection of straight lines in 3D (Fig. 2). We use these in a standard line-based approach for estimating radial distortion [18].

In image regions that violate our motion assumptions, the advection process will often estimate streamlines that are not the projection of 3D lines. To address this, we propose to filter out such lines by scoring each streamline and only using the highest-scoring lines. We assume that the radial distortion is fairly mild; our score is the inverse of the sum-of-squared differences from a quadratic approximation of the streamline. To reduce the number of very short lines, we normalize by the distance between the line endpoints. This approach only requires a single video, but we can combine streamlines from multiple days and use this method to automatically select the best streamlines. We find that this results in a more robust estimate of radial distortion. We demonstrate the effectiveness of this method in Sec. 4.

2.2. Estimating the Cloud Motion Vanishing Point

Our remaining calibration methods require an estimate of the vanishing point of the cloud motion. We found that directly computing optical flow from the structure tensor, as we did in the previous section, resulted in noisy vanishing point estimates, despite attempting numerous approaches to robust estimation. Instead, we adopt the "direct" approach, in which we fit a constrained motion model directly to image derivatives. This avoids the error prone intermediate step of estimating optical flow.

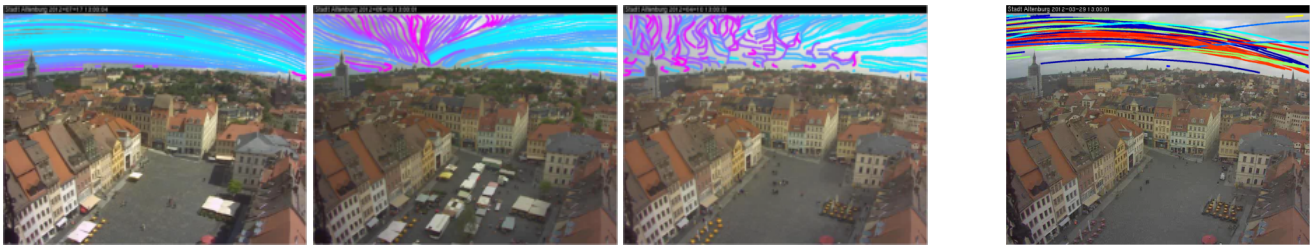
We further constrain the motion at each pixel, $\mathbf{p}$, to be in the direction of the vanishing point such that $[x_p, y_p]^\mathsf{T} = \alpha_p(\mathbf{v} - \mathbf{p})$. This results in the following objective function that depends on the vanishing point location, $\mathbf{v}$, and the scaled, local motion magnitudes, $\{\alpha_p\}$:

$$\begin{aligned} f(\mathbf{v}, \{\alpha_p\}) &= \sum_p ||\alpha_p \mathbf{A}_p(\mathbf{v} - \mathbf{p}) + \mathbf{b}_p||^2 \\ &= \sum_p \alpha_p^2 (\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p^2 (\mathbf{v} - \mathbf{p}) + \\ &\quad 2\alpha_p (\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p{}^\mathsf{T} \mathbf{b}_p + \mathbf{b}_p{}^\mathsf{T} \mathbf{b}_p. \end{aligned} \tag{1}$$

To estimate the vanishing point for a scene, we minimize (1) using a non-linear minimization approach similar to [4], with the key difference being that the camera calibration is unknown. At each iteration we first solve for the optimal value of α_p for each pixel by setting the partial derivative equal to zero:

$$\begin{aligned} \frac{\partial f}{\partial \alpha_p} &= 2\alpha_p (\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p^2 (\mathbf{v} - \mathbf{p}) + 2(\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p{}^\mathsf{T} \mathbf{b}_p \\ \alpha_p^{(t)} &= \frac{-(\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p{}^\mathsf{T} \mathbf{b}_p}{(\mathbf{v} - \mathbf{p})^\mathsf{T} \mathbf{A}_p^2 (\mathbf{v} - \mathbf{p})} \end{aligned} \tag{2}$$

Given that the motion magnitudes, $\{\alpha_p\}$, can be solved for analytically we define $f(\mathbf{v}) = f(\mathbf{v}, \{\alpha_p^{(t)}\})$, where the un-



(a) Example single-video streamlines

(b) Filtered streamlines

Figure 2: Streamline estimation and filtering: (a) streamlines estimated from individual videos, color coded by the score defined in Sec. 2.1. The blue (purple) lines are the higher (lower) scoring streamlines. (b) The highest ranking streamlines across all videos with color representing the video of origin.

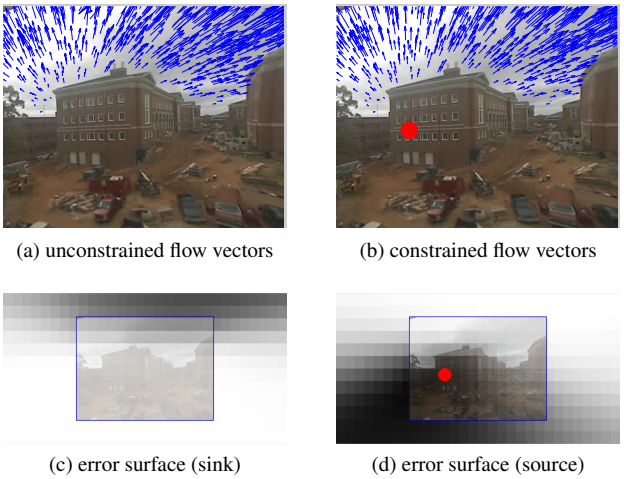
known magnitudes, $\{\alpha_p\}$ are replaced with the optimal values at the corresponding $\mathbf{v}$, subject to the constraint that they result in flows that all point toward or away from the vanishing point. For example, if the vanishing point is a sink, which means the wind is blowing away from the camera, we set $\alpha_p = \max(\alpha_p, 0)$.

We then update the vanishing point location, $\mathbf{v}$, using the gradient of (1):

$$\frac{\partial f}{\partial \mathbf{v}} = \sum_p 2(\alpha_p^{(t)})^2 \mathbf{A}_p^2(\mathbf{v} - \mathbf{p}) + 2\alpha_p^{(t)} \mathbf{A}_p \mathbf{b}_p \qquad (3)$$

We optimize our objective function using gradient descent, with initial conditions chosen by grid sampling the objective function. We find that the objective function is typically very smooth and so the grid sampling can be fairly sparse. Fig. 3 shows an example of the error surface and the resulting set of flows computed from a thirty minute video clip.

Related Work on Constrained Motion Estimation The vanishing point estimation step of our work can be seen as a egomotion problem, except with the camera as the inertial reference frame instead of the scene. Many approaches have been proposed for ego-motion estimation, see [21] for a comparison of several correspondence-based methods. Our vanishing point estimation approach is most similar to work on direct motion estimation [9, 4], which estimates motion directly from image derivatives. In Sec. 3.1, we extend this work to estimate multiple discrete vanishing points, each constrained to the same horizon line. More recently, Sheikh et al. [19] propose two methods for horizon line estimation for translational dynamic textures. The first method assumes planar motion; our method extends this to consider 3D translational motion. The second method assumes that the pattern is a weak-sense stationary space-time process; we attempted to aggregate motion statistics across all videos for a single scene and found that this assumption was rarely satisfied. We believe this is due to biases introduced by prevailing wind directions.



(a) unconstrained flow vectors

(b) constrained flow vectors

(c) error surface (sink)

(d) error surface (source)

Figure 3: Estimated cloud motions: (a) flow vectors estimated independently at each pixel and (b) our globally constrained direct estimation approach. The dot (red) is the estimated vanishing point location. (c,d) For the same scene, false color images (with original image outlined in blue for reference) in which intensity corresponds to the value of our error function (1) for different vanishing points (dark values correspond to low error). In (c) the vanishing point is considered a sink, and in (d) it is considered a source. These error surfaces show that the vanishing point is a source located on the left side of the image.

3. Geometric Calibration Using Cloud Motion

Our assumption that cloud motion is largely translational and horizontal enables us to solve a broad range of camera calibration problems. We begin with an approach for estimating the horizon line.

3.1. Horizon Line Estimation

We jointly estimate image-space vanishing points, $\bar{\mathbf{V}}$, that are consistent with our assumption of horizontal cloud motion. We start with the set of vanishing points estimated on individual days, $\mathbf{V}$, and estimate an optimal set of vanishing points that are constrained to the horizon line, $\bar{\mathbf{V}}$. Since individual vanishing points may be incorrect, we need a principled means of combining vanishing points. At a minimum, we need 2 videos with independent wind directions (not in the same or opposite directions), however, as usual, more videos results in a more reliable horizon line estimate.

We propose to extend our method for single-day vanishing point estimation to multiple days. We jointly optimize over the single-day objective function (1) using gradient descent, subject to the constraint that all points lie on a line. We experimented with various forms of line fitting, both unweighted and with various forms of confidence weighting, and found that re-using our optimization framework is more accurate and more robust to days that violate our assumptions. Directly fitting a line to the estimated vanishing points ignores the relative confidences associated with each vanishing point due to differing cloud conditions. This means that fitting a line directly to the set of vanishing points often fails to find an accurate horizon line, which leads to a cascade of errors in the remaining calibration steps.

We formulate the joint optimization problem as follows:

$$\bar{\mathbf{V}}^* = \arg\min_{\mathbf{V}} F(\mathbf{V}) = \sum_{\mathbf{v}_i \in \mathbf{V}} f(\mathbf{v}_i). \tag{4}$$

subject to the constraint that the vanishing points are collinear. We convert this constrained optimization problem into an unconstrained problem. We use two variables that represent the height of the horizon line at the left and right edge of the image, h_l, h_r, and represent the vanishing points by their distance from the center of the image along the horizon line, ϕ_i. This results in the objective function,

$$F(h_r, h_l, \mathbf{\Phi}) = \arg\min_{\{h_r, h_l, \mathbf{\Phi}\}} \sum_{\mathbf{v}_i \in \mathbf{V}} f(v(h_r, h_l, \phi_i)), \tag{5}$$

which we optimize to solve for $\{h_r^*, h_l^*, \mathbf{\Phi}^*\}$.

We optimize the objective function using gradient descent with initial conditions determined by a line fit directly to the vanishing points, $\mathbf{V}$. The gradient of the vanishing point location, ϕ_i, is computed by projecting the unconstrained vanishing point on to the horizon line, as follows:

$$\frac{\partial F}{\partial \phi_i} = \frac{(h_r - h_l)}{||h_r - h_l||}^T \frac{\partial F}{\partial \mathbf{v}_i} \tag{6}$$

where $(h_r - h_l)$ is a vector along the horizon line. We compute the gradients of the horizon line points by aggregating the normal components of the unconstrained gradients, $\frac{\partial f}{\partial \mathbf{v}_i}$. We compute the new line parameters that would result if we performed the updates specified by the gradients and use the difference from the new line and the current line as the gradient.

3.2. Focal Length and Geo-Orientation Estimation

We use the constrained vanishing points, $\bar{\mathbf{V}}$, to estimate the focal length and geo-orientation of the camera. Many approaches have been proposed for using vanishing points for camera calibration; they typically require mutually orthogonal vanishing points [7] or, more generally, known angles [22]. In our approach, we consider vanishing points as noisy observations of a translational motion with known geo-orientation. We assume the rough geo-location of the camera and the capture time of each video is known. Using this information, we query a weather database to obtain an estimate of the wind velocity, $\mathbf{w}_i$, for each video. If the camera is correctly calibrated, the cloud-motion vanishing points will correspond with the true cloud motion which is largely determined by the wind velocity. We define a misalignment cost function and search for the calibration parameters.

Since we know the horizon line, there are only two unknowns, the camera azimuth, θ, and the focal length, f. We compute the world direction that corresponds to a particular vanishing point, $\mathbf{v}_i$, as $(\mathbf{R}_\theta^{-1}\mathbf{R}_{\phi\psi}^{-1}\mathbf{K}^{-1}\bar{\mathbf{v}}_i)$, where $\mathbf{K} = diag([f, f, 1])$, $\mathbf{R}_{\phi\psi}$ is a rotation matrix that encapsulates camera roll and tilt and is computed directly from the horizon line, and $\mathbf{R}_\theta$ is the rotation matrix that defines the camera azimuth.

We minimize the following error function using grid search to estimate f and θ:

$$\sum_{i=1}^{N} \|\mathbf{w}_i\| - \mathbf{w}_i{}^{\mathsf{T}} \frac{(\mathbf{R}_\theta^{-1}\mathbf{R}_{\phi\psi}^{-1}\mathbf{K}^{-1}\bar{\mathbf{v}}_i)}{\|(\mathbf{R}_\theta^{-1}\mathbf{R}_{\phi\psi}^{-1}\mathbf{K}^{-1}\bar{\mathbf{v}}_i)\|}. \tag{7}$$

The intuition behind this error is that if the vanishing points are perfectly aligned with the wind vectors, the error will be zero, and if they are in exactly the wrong direction, the error will be twice the total magnitude of the wind vectors. Many alternative error metrics are possible; we tried several and found performance to be roughly the same.

4. Evaluation

We evaluate the quantitative and qualitative performance of our proposed calibration approaches and vanishing point estimation method on video from seven outdoor scenes, including six from the LOST dataset [2]. For each scene we have videos captured over several months, with roughly thirty minutes of video each day from each camera. For high-framerate cameras, we resample the videos to 1Hz before computing motion aggregates to reduce numerical is-

sues. Our implementation[1], running on a standard desktop PC, takes between 15 to 30 minutes to fully geo-calibrate each scene.

4.1. Automated Video Filtering

On certain days, individual videos captured "in the wild" may violate our assumption that cloud motion can be modeled as a single translation. We define a metric that we find reliably identifies such videos. This metric is the ratio of two quantities: the numerator is the optimal value of the vanishing point objective function (1) and the denominator is the median of the same function on the grid of samples defined in Sec. 2.2. The intuition behind this metric is that a large improvement in this error ratio suggests that there is a unique vanishing point. We use a threshold of 0.2 to filter out problematic videos and found it to work well in our experiments. While it clearly depends on the local environment of the camera, we found that on average 35 percent of the videos satisfy this criteria. Fig. 4 shows several videos with different error ratio.

4.2. Automatic Radial Distortion Correction

We evaluated our cloud-based distortion estimation method and found that it reliably estimated the distortion on the scenes in our dataset (Fig. 5). Since we do not have easy access to the cameras, we were unable to compare to any ground truth. However, we did compare against manually annotated lines. For each scene, we found five lines in each scene which we believed to be straight and clicked twenty points along each line. We found that the results from our automatically selected lines were qualitatively better than using manually selected lines as input to the same line-based distortion estimation routine. We suspect that our improved results are primarily due to difficulty in selecting truly straight lines and accurately clicking on them.

To better understand the performance of our method, we compare distortion estimates using different numbers of videos to the final calibration result obtained using all videos. For each scene, we compute a distortion estimate based on the first video, then another on the first and second, and so on. Each of these models is compared to the final distortion estimate using the average difference in pixel displacement between the two models. Results in Fig. 5 show that the distortion estimates typically converge in ten videos.

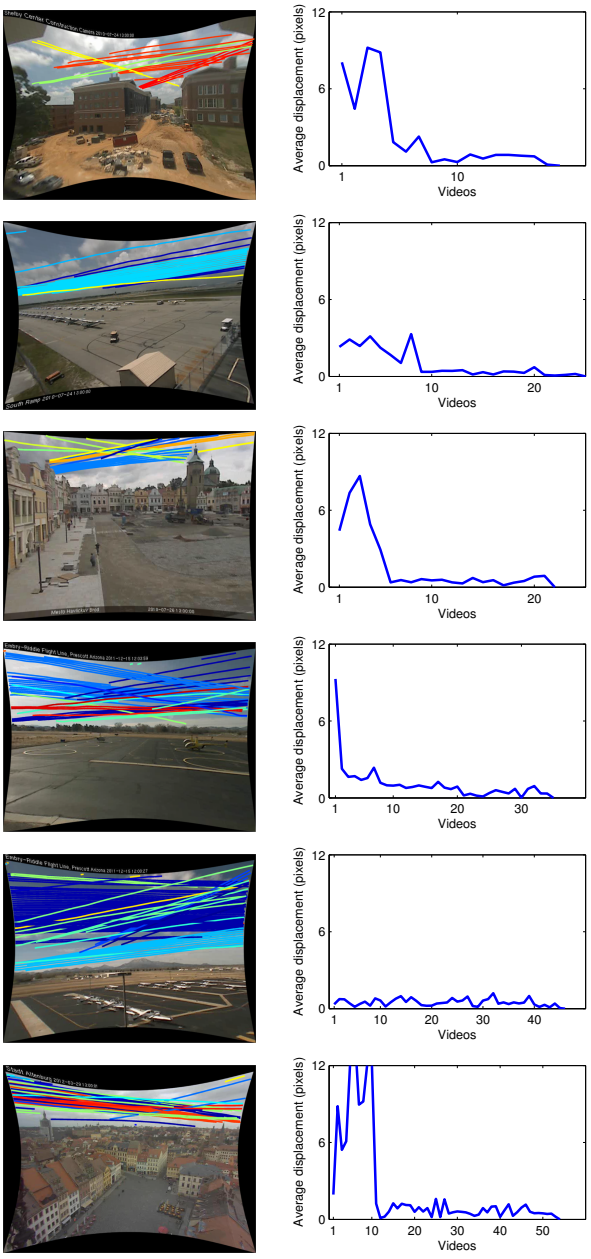


Figure 5: Results of automatic radial distortion correction: (left) Undistorted frames and automatically filtered streamlines, (right) as additional videos are added the distortion estimates converge to the final estimate.

4.3. Vanishing Point Estimation

Using all the pixels from the sky region to find the vanishing point is computationally expensive, so in practice we subsample pixels. We performed an experiment to explore the effect of changing the number of pixels on the resulting vanishing point. For each video in our dataset, we computed the vanishing point for differing numbers of random samples of pixels. We then computed the average error w.r.t. the vanishing point computed using the largest sample size across all videos with 15 random samples for each setting. The results of this experiment (Fig. 6) demonstrate that ten thousand pixels results in average vanishing point estima-

[1] Available at http://cs.uky.edu/~jacobs/projects/clouds2calibration/

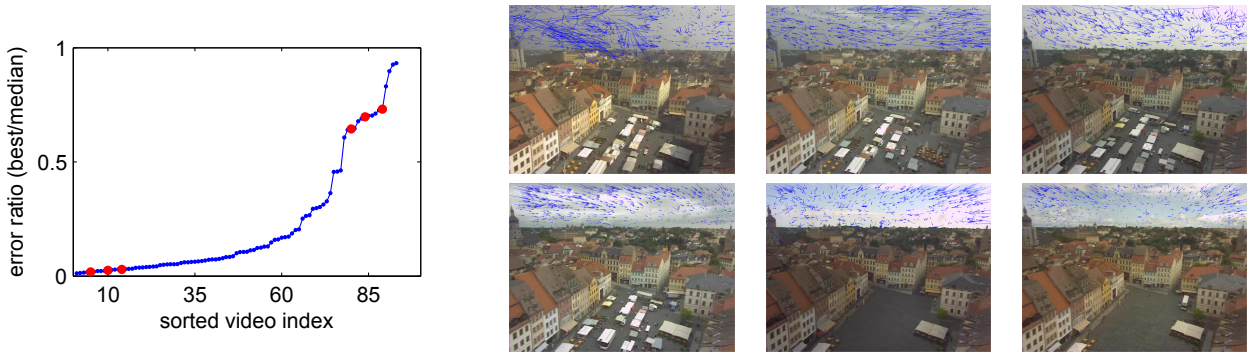


Figure 4: (left) All videos from a single scene sorted by the error ratio defined in Section 4.1. (right) For the six highlighted videos, we show images overlaid with per-pixel optical flow vectors computed from the corresponding video. The top row shows scenes with high error ratios; notice the diversity of optical flow vectors which contrasts with the flows in the bottom row (low error ratio).

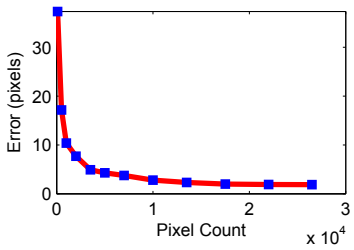


Figure 6: Increasing the number of pixels included in the vanishing point estimation process (Sec. 2.2) increases robustness, but we reach the point of diminishing returns at around ten thousand pixels.

tion error of approximately 3 pixels.

4.4. Horizon Line Estimation

We use the method defined in Sec. 3.1 to estimate the horizon line and a set of constrained vanishing points. The results in Fig. 7 show that even when some of the initial vanishing points are not close to the horizon, our method is able to combine all of the days of data and estimate an accurate horizon line.

4.5. (Geo)-Calibration of Real Scenes

For each of the scenes we consider, we know the geographic location and we manually estimate the orientation and field of view by matching image points to world points using ProjectLive3D [1]. For each video, we know the time-stamp; if possible we use the location and time-stamp to archive upper-level wind data from the FAA[2]; otherwise we use ground-level wind data from Weather Underground[3].

[2] http://aviationweather.gov/products/nws/winds/
[3] http://wunderground.com

Table 1: Quantitative Geo-calibration results.

Scene	FOV (gt, est)	Heading (gt,est)
LOST 26	$68.2°, 71.7°$	$64.1°, 58.2°$
LOST 22	$54.6°, 60.6°$	$-99.3°, -109.1°$
MARK 1	$54.9°, 86.9°$	$0.1°, -10.9°$
LOST 21	$74.7°, 44.6°$	$-138.7°, -141.8°$
LOST 2	$76.0°, 118.5°$	$-102.9°, -116.4°$
LOST 6	$68.4°, 11.4°$	$-164.0°, 160.0°$

For the FAA data, we linearly interpolate between the provided times and location for each elevation and then average across elevations to obtain a single wind velocity estimate for each video. For the Weather Underground data, we remove null values and linearly interpolate in time.

Results of our automated geo-calibration approach are shown visually in Fig. 7 and numerically in Tbl. 1. Based on these results and visual inspection of the error function (7), the cloud motion cue provides a stronger constraint on the camera orientation than on the focal length. For most scenes we have a very good estimate for the heading of the camera, but the focal length of the optimal solution is less reliable. We believe this is largely due to errors in the wind data. Based on manual inspection we see many cases in which the estimated wind data does not agree with the observed motion of the clouds. We believe that including more days of data will minimize this source of error.

5. Conclusion

We have shown that cloud motion provides geometric cues that enable camera calibration. We introduced automated methods for estimating the radial distortion, horizon line, focal length and geo-orientation of a static outdoor camera from video captured over many days. These meth-

ods enable calibration in scenes which do not contain sufficient static geometric information for more traditional calibration techniques, such as orthogonal vanishing points [3] or registration with known geometry [1].

Our work is based on several assumptions about cloud motion that are not always satisfied. We assume clouds move as a 3D translation. To handle such violations, for example when the wind direction changes or two layers of clouds are moving in different directions, we propose to filter out videos based on the uniqueness of the vanishing point. We further assume that cloud motion is in the horizontal plane. While this assumption may be consistently violated in mountainous regions, it was not a significant problem in our dataset.

Acknowledgments We thank Robert Pless, Austin Abrams, Jim Tucek and Joshua Little for collecting the LOST dataset and Jim Knochelmann for collecting the wind velocity metadata. This work was supported by the DARPA Computer Science Study Group program (D11AP00255).

References

[1] A. Abrams, N. Fridrich, N. Jacobs, and R. Pless. Participatory integration of live webcams into gis. In *COM.GEO*, 2010.

[2] A. Abrams, J. Tucek, J. Little, N. Jacobs, and R. Pless. Lost: Longterm observation of scenes (with tracks). In *IEEE Workshop on Applications of Computer Vision*, 2012.

[3] P. Beardsley and D. Murray. Camera calibration using vanishing points. In *British Machine Vision Conference*, 1992.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, 1992.

[5] M. D. Breitenstein, H. Grabner, and L. V. Gool. Hunting nessie—real-time abnormality detection from webcams. In *IEEE Workshop on Visual Surveillance*, 2009.

[6] X. Cao and H. Foroosh. Camera calibration and light source orientation from solar shadows. *Computer Vision and Image Understanding*, 105(1), 2007.

[7] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 1990.

[8] F. Cozman and E. Krotkov. Robot localization using a computer vision sextant. In *IEEE International Conference on Robotics Applications*, volume 1, 1995.

[9] B. K. Horn and E. Weldon Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 1988.

[10] N. Jacobs, B. Bies, and R. Pless. Using cloud shadows to infer scene structure and camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[11] N. Jacobs, W. Burgin, N. Fridrich, A. Abrams, K. Miskell, B. H. Braswell, A. D. Richardson, and R. Pless. The global network of outdoor webcams: Properties and applications. In *ACM SIGSPATIAL*, 2009.

[12] N. Jacobs, N. Roman, and R. Pless. Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In *IEEE Workshop on Applications of Computer Vision*, 2008.

[13] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *IEEE International Conference on Computer Vision*, 2007.

[14] S. J. Kim, J.-M. Frahm, and M. Pollefeys. Radiometric calibration with illumination change for outdoor scene analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[15] J.-F. Lalonde, S. G. Narasimhan, and A. A. Efros. What does the sky tell us about the camera? In *European Conference on Computer Vision*, 2008.

[16] J. Luo and S. Etz. A physical model-based approach to detecting sky in photographic images. *IEEE Transactions on Image Processing*, 11(3), 2002.

[17] T. McLoughlin, R. Laramee, R. Peikert, F. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6), 2010.

[18] B. Prescott and G. McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1), 1997.

[19] Y. Sheikh, N. Haering, and M. Shah. Shape from dynamic texture for planes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

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

[21] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[22] M. Wilczkowiak, E. Boyer, and P. Sturm. Camera calibration and 3d reconstruction from single images using parallelepipeds. In *IEEE International Conference on Computer Vision*, 2001.

[23] J. Wright and R. Pless. Analysis of persistent motion patterns using the 3d structure tensor. In *IEEE Workshop on Applications of Computer Vision*, 2005.

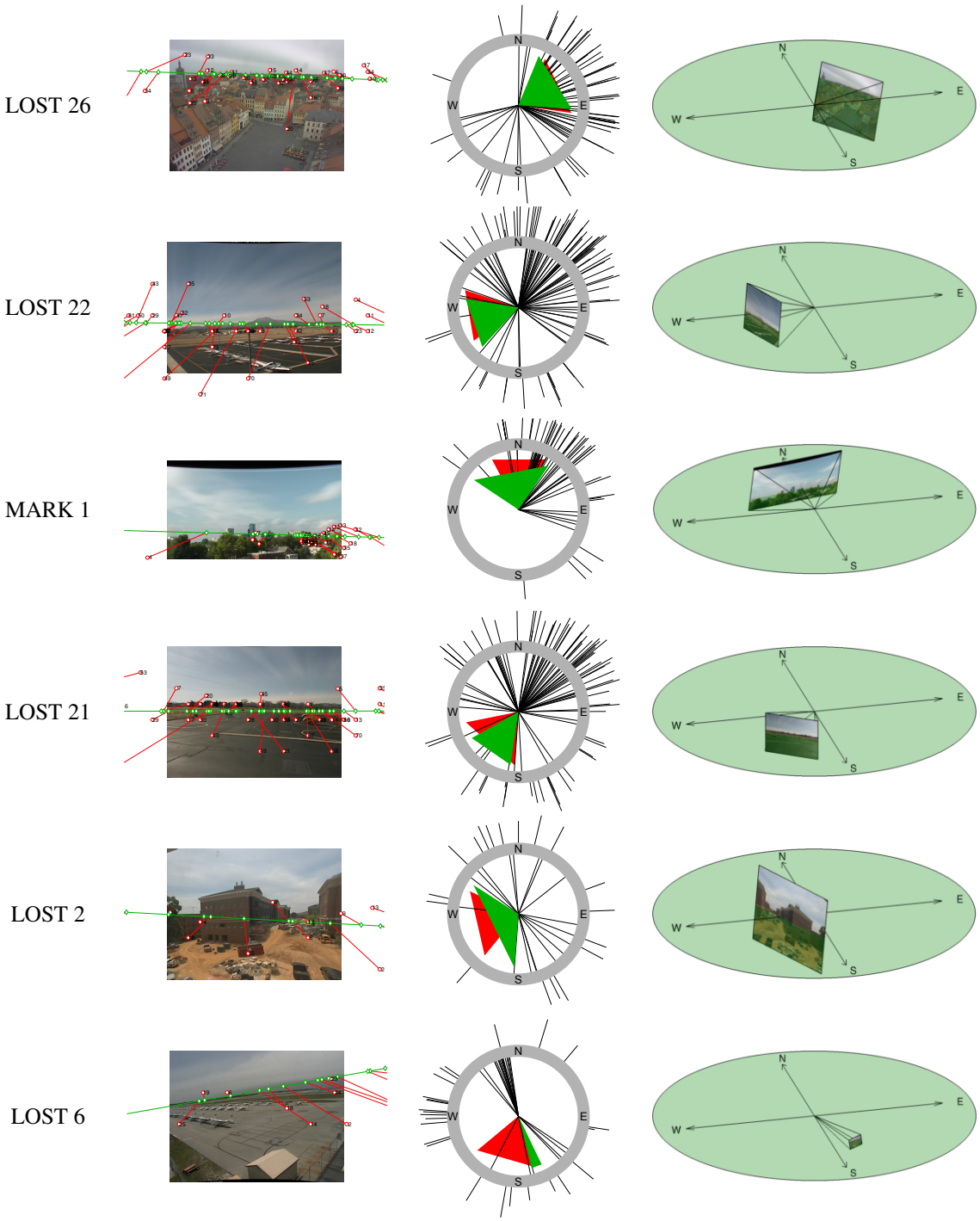


Figure 7: Intrinsic and geo-orientation calibration results on six scenes. (left) For each scene we show a sample image overlaid with the single-video vanishing points (red dots), the estimated horizon line (green) and the final constrained vanishing points (connected to the original by a red line). (middle) We show a visualization of the resulting geo-calibration, viewed from zenith with the camera in the center of a circle labeled with the cardinal directions. Each filled triangle corresponds to either the ground truth (red) or the optimal estimated (green) view frustum. The lines emanating from the center are the world-space rays that correspond to the estimated vanishing points. The lines outside the compass ring represent the wind velocities used for calibration. (right) An alternative view of the calibration results shown in the middle column.