Cloudmaps from Static Ground-View Video

Nathan Jacobs[a,*], Scott Workman[a], Richard Souvenir[b]

[a] *Department of Computer Science, University of Kentucky, Lexington, KY, 40506*
[b] *Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC, 28223*

Abstract

Cloud shadows dramatically affect the appearance of outdoor scenes. We describe three approaches that use video of cloud shadows to estimate a cloudmap, a spatio-temporal function that represents the clouds passing over the scene. Two of the methods make assumptions about the camera and/or scene geometry. The third method uses techniques from manifold learning and does not require such assumptions. None of the methods require directly viewing the clouds, but instead use the pattern of intensity changes caused by the cloud shadows. An accurate estimate of the cloudmap has potential applications in solar power estimation and forecasting, surveillance, and graphics. We present a quantitative evaluation of our methods on synthetic scenes and show qualitative results on real scenes. We also demonstrate the use of a cloudmap for foreground object detection and video editing.

Keywords: image formation; time-lapse; clouds; lighting estimation; solar forecasting; scene factorization

1. Introduction

Clouds are a significant factor in determining the available solar energy and, consequently, have a significant impact on processes ranging from plant growth [1], solar power generation [2] and climate change [3]. For such applications, the most common method for assessing available solar energy relies on point source samples using specially designed solar radiation sensors. This can lead to inaccurate estimates, especially if the sensor is located far from the object of study. Vision-based methods have the potential to extend the coverage of such sensors, thereby increasing the accuracy of solar radiation estimates. However, recent vision-based methods for outdoor scene understanding either explicitly eliminate images captured on cloudy days [4, 5, 6] or only estimate a single scalar cloudiness parameter per frame [7, 8, 9].

The most prominent approaches for estimating cloud cover rely on sky cameras [10, 11, 12, 13, 14], which are imaging systems setup to capture a full view of the sky from a single location. In such systems the sun is often in the field of view, which causes artifacts. Therefore, these cameras often incorporate moving physical barriers to block the sun. The main drawback of these approaches is the cost associated with purchasing, maintaining, and deploying the equipment. Additionally, limitations inherent to a single viewpoint are unavoidable. A single viewpoint reduces the value of a sky camera when the clouds are near the ground or when they are vertically thick. In both cases, it is difficult to estimate the amount of cloud occlusion anywhere except directly toward the sun, which, in most setups, is blocked.

While sky cameras are rare, surveillance cameras and webcams are ubiquitous. These types of cameras are inexpensive to purchase, and often view very little sky but a wide area on the ground. We propose to use such cameras to estimate a *cloudmap*, a time-varying 3D function, defined in world coordinates, that describes the clouds passing over a scene (see Fig. 1). Our methods take advantage of shadows cast by moving clouds to simultaneously estimate a cloudmap and a mapping between image pixels and cloudmap coordinates. We estimate the amount of sunlight attenuation at each pixel in each frame and combine these individ-

[*] Corresponding author
Email addresses: jacobs@cs.uky.edu (Nathan Jacobs), scott@cs.uky.edu (Scott Workman), souvenir@uncc.edu (Richard Souvenir)

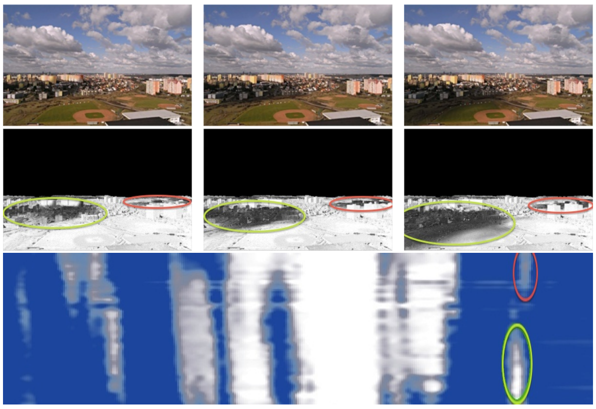


Figure 1: Given a video from an outdoor static camera, we estimate the sunlight attenuation due to clouds (middle) by analyzing the time series of intensity changes in pixels from the ground. We introduce several methods, which do not require directly viewing the clouds in the sky, for combining these time series into a cloudmap (bottom), a geo-temporal function that describes the the shape and thickness of set of clouds that passed over the scene. This cloudmap summarizes ∼25 minutes of video and the clouds corresponding to the visible shadows in the image are outlined.

ual pixel estimates into a globally coherent model.

Given a video of an outdoor scene, we first extract a scalar time series for each pixel describing sunlight attenuation, i.e., cloudiness. For each pair of pixels, we estimate the temporal delay between the corresponding time series, and then denoise the time series using the assumption that the time series of other pixels, directly in-line with the cloud motion direction, will be similar. We then estimate the scene model and cloud motion direction using one of three methods. Finally, the denoised cloudiness time series, temporal delays, scene model, and cloud motion direction are used to estimate a cloudmap which is a time-varying 3D function. If required for an application, we render the cloudmap in 2D by making some assumptions about how the cloudmap changes.

This paper makes several contributions: three techniques for estimating cloudmaps from outdoor video, techniques for denoising and rendering cloudmaps, and a method for estimating the cloud motion direction from a video with known geometry. We present quantitative and qualitative results on a variety of scenes, as well as show several proof of concept applications.

2. Related Work

Images have been used to estimate cloud cover for a variety of purposes, including image retrieval [15], graphics [16], weather estimation [8] and solar power forecasting [2, 17]. This paper extends our previous work [18], which was, to our knowledge, the first work to attempt to use ground shadows to estimate a cloud layer. Here, we describe work on several related problems.

2.1. Outdoor Video Understanding

Recent work has shown the benefits of explicitly reasoning about the underlying causes of outdoor appearance variations. For example, color changes due to sun motion are strong cues to scene shape and albedo [4, 19, 5, 6] and transient clouds and their shadows are strong cues to scene geometry [20, 21] and camera calibration [22]. Outdoor videos have also been used to estimate dynamic scene properties, including low-dimensional cloudiness models [7, 8, 9]. Our work is the first to construct highly detailed cloudiness models from video of an outdoor scene.

2.2. Shadows in Outdoor Scenes

Methods for detecting and handling shadows appear in several settings. Most similar to our work is in the surveillance domain [23, 24] where it is important to reduce the number of false-positive detections. These methods focus on modeling individual pixel variations and do not explicitly model the motion of the clouds in the scene. More sophisticated methods, such as [25], can detect shadows in individual frames, but are computationally intensive and are less robust because they do not take advantage of the available temporal information. Our approach takes advantage of the image time series to obtain accurate per-pixel, per-frame attenuation estimates and explicitly fits a model that accounts for cloud motion.

2.3. Estimating Outdoor Illumination

Estimating a cloudmap is a special case of the more general problem of estimating illumination conditions. Recent work in several areas has attempted to solve this problem from a single image of an outdoor scene. This is a challenging line of research, with many inherent ambiguities.

Lalonde et al. [26] estimate illumination conditions from a single outdoor image and obtain an estimate of the sun direction, but only consider three cloudiness states: clear, partly cloudy, and overcast. Li et al. [27] address the problem of estimating the degree to which each sky pixel in an image is occluded by clouds by directly observing the sky. They use a Gaussian mixture model to represent a set of simple color features combined with a Markov Random field for enforcing spatial coherence. Peng et al. [16] address the same problem, but incorporate a physically based sky appearance model [28]. Veikherman et al. [14] propose a tomographic approach to building a 3D model of clouds from imagery captured by a network of sky cameras. This approach is complementary to ours because it relies on a different imaging geometry.

Murdock et al. [8] propose a method for estimating cloud cover from a collection of ground-based cameras. The authors use many images and corresponding pixel intensities from existing satellite images to train camera-specific regression models that predict the cloudiness given a single image from the webcam. The scalar cloudiness estimates from simultaneously captured images from many webcams are interpolated to estimate a synthetic satellite image. A key difference in our approach is our generative model, which is necessary because satellite imagery of sufficient resolution is not available and, even if it was, would require significant registration effort. Also, we address a different problem; rather than a single scalar cloudiness value per image, we estimate a spatio-temporal function, the cloudmap, from a video. Additionally, our methods provide information about scene geometry that may be useful for other applications.

3. Estimating and Visualizing Cloudmaps

A cloudmap is a spatio-temporal function, $C(x, y, z, t) \in [0, 1]$, ranging from 'no direct sunlight' to 'full sunlight'. It defines how clouds attenuate the sunlight for every point in the scene and thus has a dramatic affect on the appearance of the scene. The effects of cloudmaps are evident in the sky, as clouds that attenuate sunlight, and on the ground, as cloud shadows due to the attenuation. In this work, we focus on the analysis of cloud shadows on the ground. The remainder of this section introduces the foundational generative model and methods we use to estimate and visualize a cloudmap.

3.1. Cloudmap Estimation

Given a video captured by a static outdoor camera, we compute the per-pixel sunlight attenuation, map pixel locations to world locations, and then interpolate to fill occlusions. In this section, we describe our approach to estimating a cloudmap when the mapping from each pixel, p, to a world location, (x_p, y_p, z_p), is known.

3.1.1. Geometric Image Formation Model

The attenuation of a scene point, (x, y, z), is determined by the clouds between the scene point and the sun. We assume that sunlight is parallel, therefore we have a single sun direction, s_t, for a scene at a particular time. For each scene point, (x, y, z), we define the sunlight attenuation as $C(x, y, z, t) = \min(1, \int c(\alpha s_{tx} - x, \alpha s_{ty} - y, \alpha s_{tz} - z), t)\,\mathrm{d}\alpha$, where $c(x, y, z, t)$ represents the infinitesimal sunlight attenuation of the atmosphere, and the integration over α represents the accumulation of attenuation as we move from the sun, through the atmosphere, toward the scene point, (x, y, z).

In this work, we assume that clouds are above the static scene elements and the camera, which is the case unless there is fog or there are scene elements, such as tall buildings or mountains, that reach the clouds. This means that the attenuation value of scene points does not change as we move along the sun direction. Therefore, we can represent the sunlight attenuation as a time-varying function, $\bar{C}(u, v, t)$, where (u, v) is a location on the ground plane. This eliminates the redundancy in $C(x, y, z, t)$. If the world location, (x_p, y_p, z_p), that is imaged by a pixel, p, is known, and it projects to the ground plane location, (u_p, v_p), then the attenuation of the pixel at time, t, is equivalently, $\tilde{C}(p, t) = C(x_p, y_p, z_p, t) = \bar{C}(u_p, v_p, t)$.

3.1.2. Estimating Per-Pixel Cloudiness

The brightness, $I(p, t)$, of a pixel, p, at time, t, is a function of static scene geometry, time-varying lighting conditions and camera properties, such as the focal length, location and orientation. Recent approaches to outdoor photometric stereo [6, 29] attempt to fit explicit models of albedo and surface orientation. However, these approaches explicitly filter out days with cloudy conditions and require multiple days of video to be effective. We describe two simple strategies for estimating the per-pixel cloudiness, one that works well for short videos (i.e., minutes) and one that works well for longer video (i.e., hours).

For short video, we assume the sun is stationary and define the following simple image formation model:

$$I(p,t) \approx \theta_u(p)\tilde{C}(p,t) + \theta_l(p). \tag{1}$$

In this model, θ_u is the maximum observed intensity for each pixel over the full video, and θ_l is the minimum. We select pixels not under cast shadows and assume they are illuminated by the full sun and the full cloud cover at least once. Our estimate of the sunlight attenuation time series for each pixel is:

$$\tilde{C}(p,t) = \frac{I(p,t) - \theta_l(p)}{\theta_u(p)}. \tag{2}$$

For longer videos, in which sun motion is more significant, the min/max method defined above leads to cloudiness estimates that are affected by surface orientation. To address this, we estimate quadratic upper and lower bounds, which correspond to the appearance of the scene with full sun and full cloud cover, respectively. First, we solve for a concave quadratic upper bound, $f(\Theta_u, t) = \theta_{u,2}t^2 + \theta_{u,1}t + \theta_{u,0}$, for the intensity time series, $I(p,t)$, at each pixel as follows:

$$\begin{aligned} \min_{\Theta_u} \quad & \sum_t \{f(\Theta_u,t) - I(p,t)\}^2 \\ \text{s.t.} \quad & f(\Theta_u,t) \geq I(p,t),\ \theta_{u,2} \leq 0 \end{aligned} \tag{3}$$

with optimal values $\Theta_u^* = (\theta_{u,2}^*, \theta_{u,1}^*, \theta_{u,0}^*)$. We then solve for the lower bound with the additional constraint that the lower bound should be less concave than the upper bound:

$$\begin{aligned} \min_{\Theta_l} \quad & \sum_t \{f(\Theta_l,t) - I(p,t)\}^2 \\ \text{s.t.} \quad & f(\Theta_l,t) \leq I(p,t),\ \theta_{u,2}^* \leq \theta_{l,2} \leq 0 \end{aligned} \tag{4}$$

with optimal values $\Theta_l^* = (\theta_{l,2}^*, \theta_{l,1}^*, \theta_{l,0}^*)$. The time series of cloudiness estimates for each pixel is:

$$\tilde{C}(p,t) = \frac{I(p,t) - f(\Theta_l^*,t)}{f(\Theta_u^*,t) - f(\Theta_l^*,t)}. \tag{5}$$

Fig. 2 shows an example comparing the cloudiness estimates from the min/max model to the quadratic bounds model for two pixels with different surface normals. For the left (blue) pixel, the full-sun intensity increases during the video as the sun more directly illuminates the surface. The opposite is true of the right (red) pixel. The min/max model does not compensate for this, but the quadratic model does. However, for shorter videos, the min/max model may be preferred because the quadratic model may overfit.

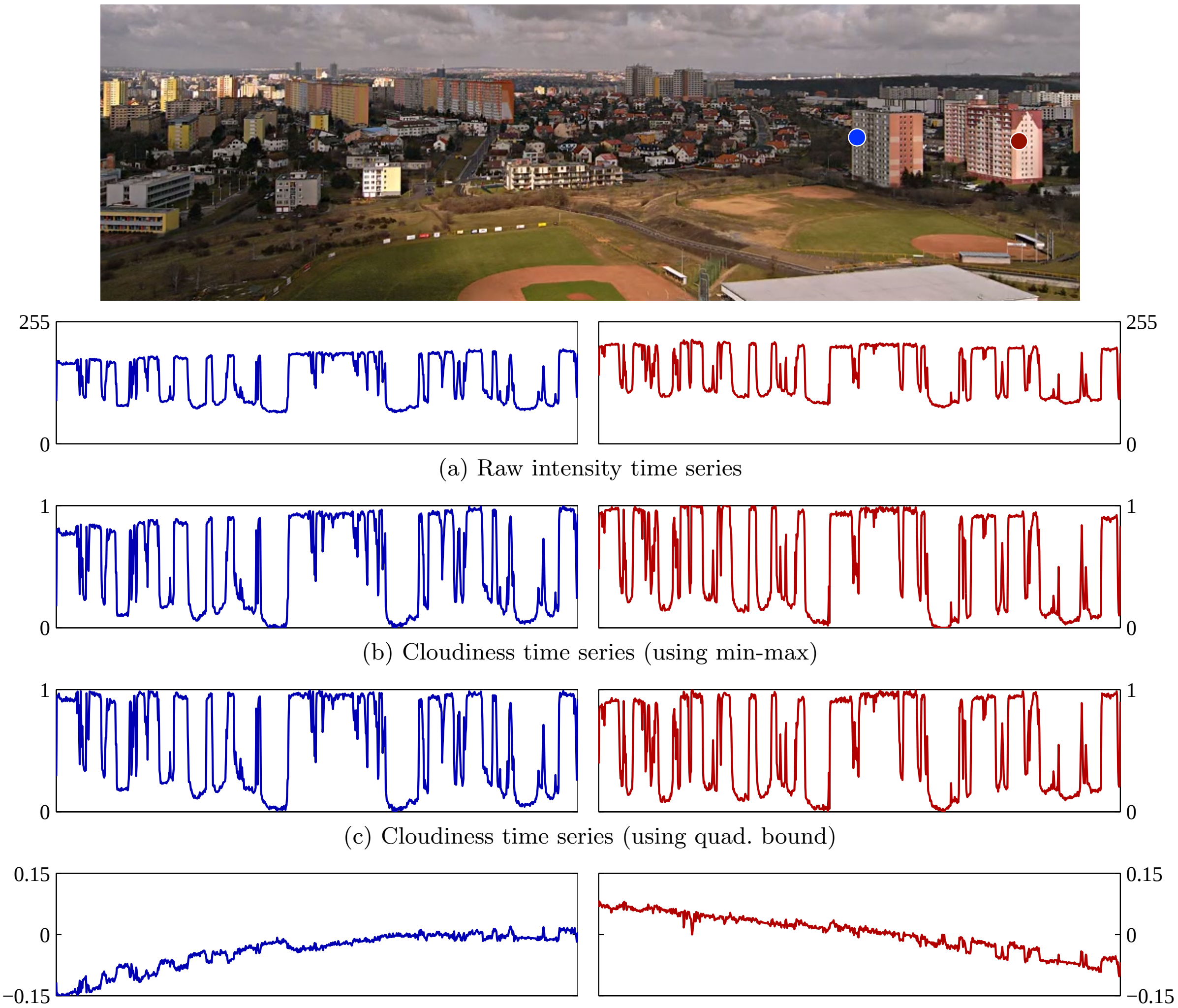
3.1.3. Estimating Temporal Delay

Appearance changes caused by cloud shadows provide information regarding the relative locations of points in the scene. Consider two pixels, p and q, that image scene points which are directly in-line with the cloud motion velocity, $(w_x, w_y, 0)$. If p images (x_p, y_p, z_p), then q images $(x_p + w_x\delta_{pq}, y_p + w_y\delta_{pq}, z_p)$, where the temporal delay, δ_{pq}, is the amount of time it takes the cloud shadows to travel from p to q. When the temporal delay, δ_{pq}, is small, the patterns of intensity changes due to the motion of clouds at these two points is similar. That is $\tilde{C}(p,t) = C(x_p, y_p, z_p, t) \approx C(x + w_x\delta_{pq}, y + w_y\delta_{pq}, z_p, t+\delta_{pq}) = \tilde{C}(q,t)$. In general, this similarity decreases as the temporal delay, δ_{pq}, increases, because cloud shapes evolve over time. This same basic relationship holds if clouds are not directly in-line with the cloud motion. However, the similarity will decrease as the displacement orthogonal to the cloud motion increases, because different clouds are passing over the points.

There are many approaches [30, 31] for estimating the temporal delay, δ_{pq}, between two time series, $I(p,t)$ and $I(q,t)$. We compute the delay using a two-stage process, first described by Workman et al. [21]. To initialize, we select the integer delay offset that maximizes the correlation between the two time series (in our implementation we use the time series of the green color channel, because it had less spatial noise than the intensity in the videos we use for evaluation). We then refine this estimate, to obtain sub-frame accuracy, by choosing the temporal offset, δ, that maximizes the correlation between the two time series, $\max_\delta \mathbf{corr}(I(p,t), I(q,t+\delta))$, using local iterative search (golden section search with parabolic interpolation) with δ constrained to be within one frame of the integer estimate. The result is an estimate of the temporal delay, δ_{pq}, and the correlation of the time series, $\rho_{pq} = \mathbf{corr}(I(p,t), I(q,t+\delta_{pq}))$, after compensating for the delay.

3.1.4. Removing Noise Due to Transient Objects

The assumption of our generative model is that all appearance changes in the scene are due to cloud shadows. If the scene contains changes that are not caused by clouds, such as small moving objects and plants swaying in the wind, this violates our assumptions and can lead to inaccurate cloudmaps. We propose to filter out appearance changes that are not due to cloud shadows using an approach



(a) Raw intensity time series

(b) Cloudiness time series (using min-max)

(c) Cloudiness time series (using quad. bound)

(d) Difference between min-max and quad. cloudiness time series

Figure 2: For the two marked pixels (blue and red), each row of plots shows (a) the raw intensity time series for one hour of video, the cloudiness estimated using the (b) min/max and (c) quadratic models, and (d) the differences between the two estimates. The quadratic model is able to compensate for brightness changes due to sun motion.

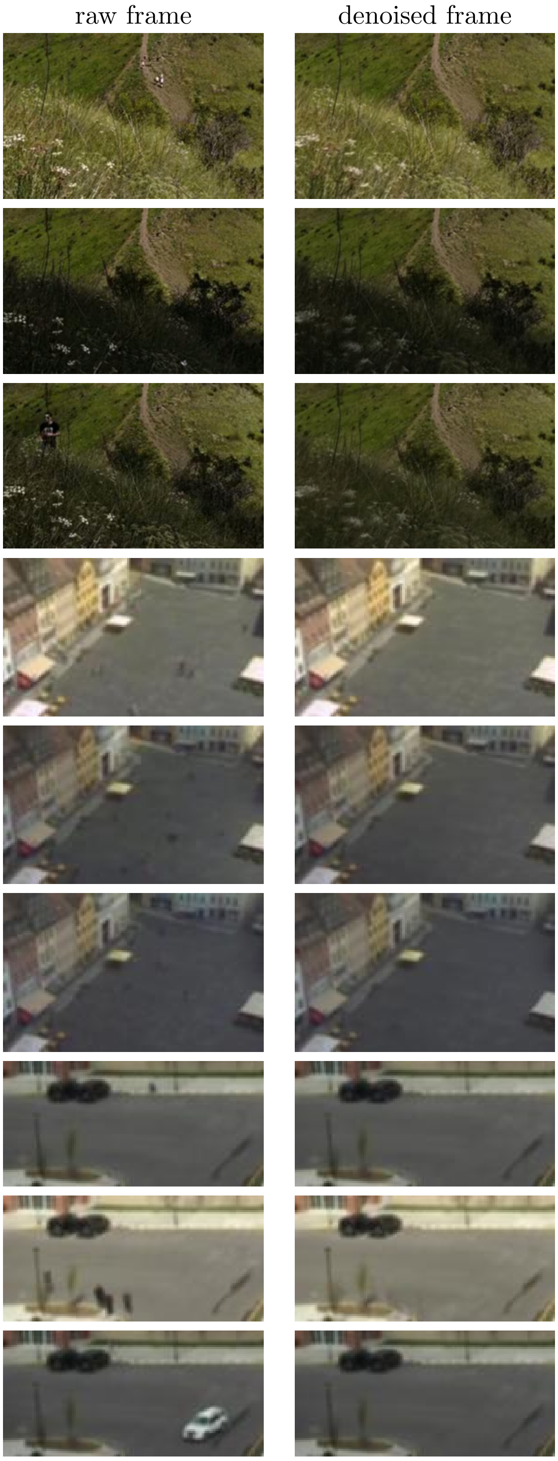


Figure 3: (left) Example frames from three different outdoor scenes. (right) Corresponding frames in which local appearance changes have been removed, but global changes are preserved, using the method described in Sec. 3.1.4.

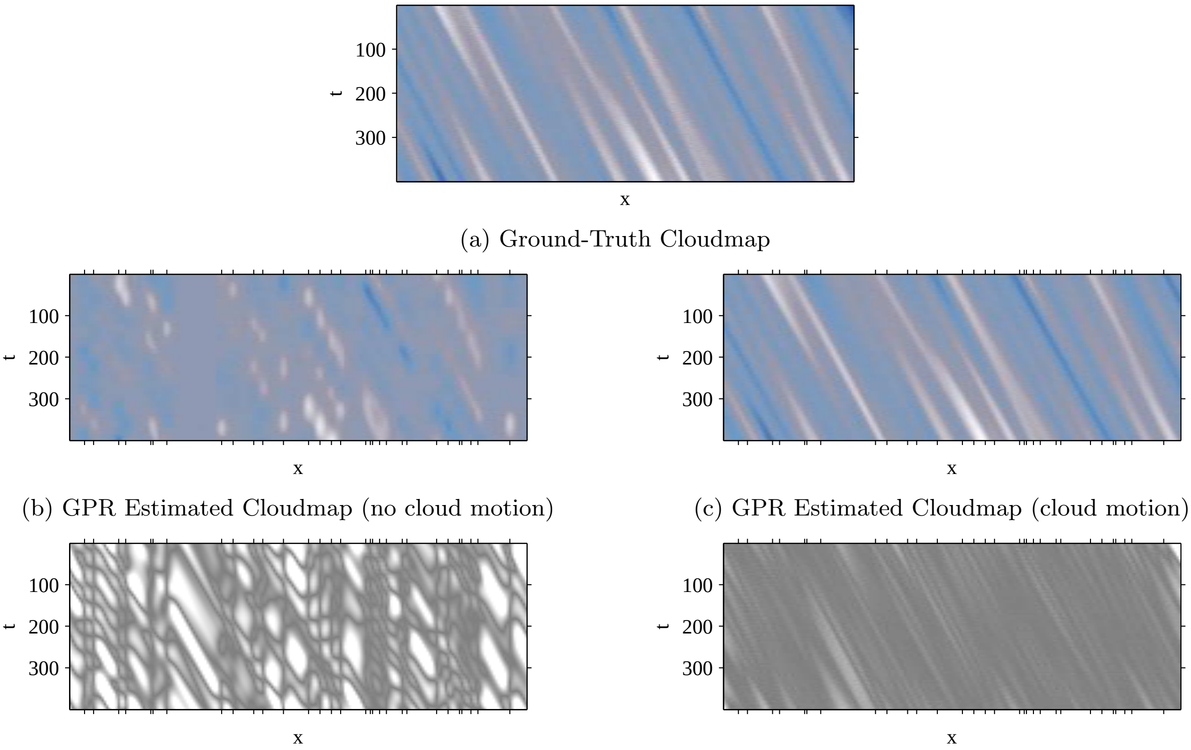
similar to manifold denoising [32]. The intuition behind our approach is that changes due to clouds have a much larger spatial extent than those due to moving objects and plants. In practice, this assumption is reasonable; clouds that cast significant shadows are generally more than a hundred meters across, and often much larger.

For a given pixel, we filter out the small changes by linearly reconstructing its time series. As a basis, we use a collection of temporally aligned time series that have high delay-corrected correlation, ρ_{pq}, but are not close in estimated 2D coordinates or image coordinates. This last restriction is important because it means that localized changes will not be in the basis used to reconstruct the time series, and hence such changes will be minimized. We use the same procedure, with different basis time series for each pixel in the video, to create a final filtered (cloud-only) video. See Fig. 3 for examples of video frames with localized appearance changes removed. In Sec. 6.1, we show how this technique can be applied to detecting foreground objects.

3.2. Cloudmap Interpolation and Visualization

Unless the scene is planar, the complete cloudmap for a region is not viewable due to scene occlusions. However, since the cloudmap is often highly structured due to cloud motion, it is possible to fill in missing values. Many methods have been proposed for such problems; we use a Gaussian process regression model [33] to predict the cloudiness. As training data for the regression model, we use cloudiness estimates, $\bar{C}(u_p, v_p, t)$, as targets and motion-adjusted location parameters, $(u_p - w_u t, v_p - w_v t, t)$, as predictors. We then fit covariance and noise model parameters using nonlinear optimization. By adjusting for cloud motion, we can use an isotropic squared exponential kernel to obtain accurate predictions. The weights of the regression model are related to how quickly the clouds change in space and in time. Given the model parameters, the Gaussian process can be queried to estimate any missing cloudiness values, and obtain the confidence in the cloudiness estimate.

Fig. 4 shows an example of this process, with and without cloud motion, applied to synthetic data. The ground-truth cloudmap (Fig. 4a) was generated by randomly sampling from a Gaussian process model with a linear covariance function that incorporates a cloud motion velocity parameter. We randomly sample 30 collinear spatial locations, extract all cloudiness values from the ground-truth



(a) Ground-Truth Cloudmap

(b) GPR Estimated Cloudmap (no cloud motion)

(c) GPR Estimated Cloudmap (cloud motion)

(d) Estimated Cloudmap Error (no cloud motion)

(e) Estimated Cloudmap Error (cloud motion)

Figure 4: (a) A 1D+T slice of a synthetic 2D+T cloud layer. (b,c) Cloudmaps reconstructed using Gaussian Process Regression with and without a known cloud motion direction. Training samples, $(x_i, t_i, C(x_i, t_i))$, are the time series of randomly selected locations, shown as tick marks along the x-axis. (d,e) The absolute prediction error for each method. Incorporating the cloud motion direction significantly improves the prediction accuracy.

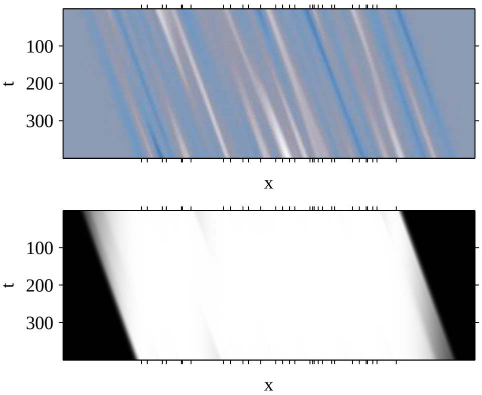


Figure 5: An example of a spatially extrapolated cloudmap (top) reconstructed from sparse spatial samples (x-axis tick marks) and the corresponding error map (bottom) in which white means zero prediction error. Our process for converting a 2D+T cloudmap to a 1D+T cloudmap visualization is analogous to extracting a row from the extended cloudmap.

cloudmap, and use gradient descent to fit the parameters of two different models: one with a simple linear covariance function and the other with a linear covariance function that compensates for cloud motion. Since the points are collinear, we can visualize the result as a 1D+T image. From this simple example we can see that incorporating cloud motion in the model is critical for accurate cloudmap estimates. Fig. 5 demonstrates how we can extrapolate beyond the spatial sampling area.

Since it is difficult to visualize a 2D+T cloudmap, we present a simple method for rendering a 2D+T cloudmap as a 2D image. While there is potential for information loss, such as when a cloud changes shape, this view is useful for visualization and has potential applications to graphics (Sec. 6.2). We use a similar regression model as above. Essentially, the clouds serve as an inertial reference frame; a model is fit to the points and queried to obtain a 2D cloudmap. Following our 1D+T example (Fig. 5) this is equivalent to choosing a single time step and rendering all spatial locations.

4. Estimating Scene Models

We present three new methods for estimating scene shape and cloud motion direction from video. These methods take as input the cloudiness time series, temporal delay estimates and delay-corrected correlations for a subset of the scene pixels from the ground. The first method assumes known scene geometry, or uses an existing method (e.g., [20]) to explicitly estimate the geometry (Sec. 4.1). The second method requires a known camera model and assumes a planar world (Sec. 4.2). The third method uses manifold learning techniques to directly estimate the scene layout and is suitable when little is known about the camera calibration and orientation (Sec. 4.3).

4.1. Known-Geometry

When the scene geometry is known, the only remaining unknown is the cloud motion direction. Given the geometry in an east-north coordinate frame, we solve for the cloud motion direction vector, using brute force search, that results in the maximum correlation between the measured temporal delay and the distance projected onto the cloud motion vector for all pairs of pixels.

When the scene geometry is not known in advance, which is the case for all real scenes used in this work, we use an existing method for scene shape estimation from cloud shadows [20] (ProjN-MDS). The basic idea is to use nonlinear optimization to solve for the scene geometry given a video captured on a partly cloudy day. While this approach can work well, this method for shape estimation is computationally intensive and subject to local minima. In addition, it either requires strong assumptions about the camera geometry or accurate camera calibration. Both of these are significant limitations when working with publicly available outdoor webcams.

Given the scene geometry and an estimate of the cloud motion direction, we can use the cloudmap denoising methods and rendering methods described in the previous section.

4.2. Planar Model

This method converts temporal delay estimates, δ_{pq}, and delay-corrected correlations, ρ_{pq}, into an estimate of the cloud motion velocity, $\vec{w}$, and the tilt, θ, of the camera that yield a scene structure that best matches the temporal delay estimates, δ_{pq}. This method assumes a known focal length, no camera roll, and a planar world.

For real-world points directly in line with the cloud motion, the temporal delay, δ_{pq}, is linearly related to the real-world distance, $||\mathbf{x_p} - \mathbf{x_q}|| = ||\vec{w}|||\delta_{pq}|$ where $\mathbf{x_i} = (u_i, v_i)$ (i.e. distance equals rate times time). Since the temporal delay is signed, we have that $\mathbf{x_p} - \mathbf{x_q} = \vec{w}\delta_{pq}$. This relationship can be generalized to include pixels not directly in line with the cloud motion direction by projecting the vector between the two points onto the cloud motion vector:

$$\vec{w}^{\mathsf{T}}(\mathbf{x_p} - \mathbf{x_q}) = \vec{w}^{\mathsf{T}}\vec{w}\delta_{pq}. \tag{6}$$

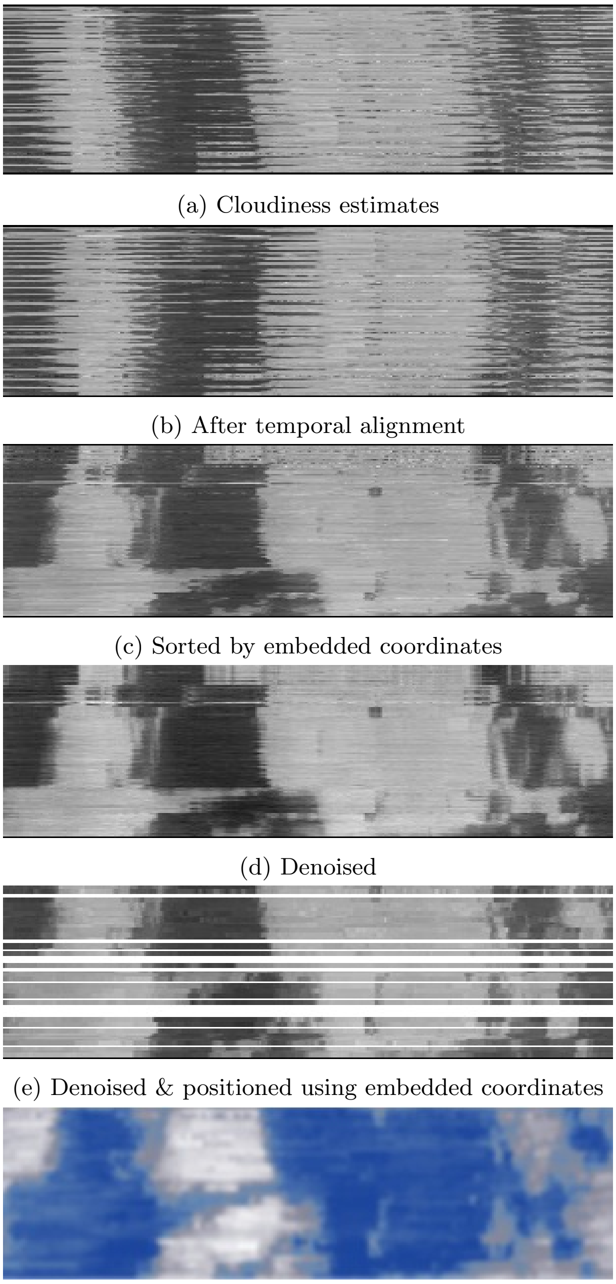
Given the external rotation matrix, R_θ, and camera matrix, $K = diag(f, f, 1)$, each pixel, p, maps to a 3D ray $R_\theta^{-1}K^{-1}p$. Incorporating the planar world assumption, we are left to solve for the intersection point, $\mathbf{x_p}$, with the ground plane. For a given scene, the projected distance between these points should satisfy Eqn. (6), which leads to the following cost function:

$$E(\vec{w}, \theta) = \sum_{pq} \left| \vec{w}^{\mathsf{T}}(\mathbf{x_p} - \mathbf{x_q}) - \vec{w}^{\mathsf{T}}\vec{w}\delta_{pq} \right| .$$

We optimize for $\vec{w}$ and θ using a grid search over horizon lines and cloud motion directions. To calculate the error for a particular setting, we compute world locations projected onto the cloud motion direction vector. The cloud motion magnitude is estimated by using robust linear regression to predict the temporal delay from the projected distances. We explicitly filter out pixel pairs with low correlation ($\rho_{pq} < .9$ for all experiments) and horizon line estimates in the bottom eighth of the image. The result of this optimization is a set of planar world coordinates, X, and estimated cloud motion velocity vector, $\vec{w}$. This is used an input for estimating a cloudmap.

4.3. Direct

In this section, we present a data-driven method that makes few geometric assumptions about the scene. We estimate the spatial scene layout, X, by learning a 2D embedding from the cloudiness time series data using nonlinear dimensionality reduction. Following a common approach in manifold learning [34, 35], we apply multiple distance metrics to decompose the set of cloudiness time series into orthogonal sources of variation: relative spatial location both parallel and orthogonal to the direction of cloud motion, in our case. In this section, we describe our two-stage procedure for estimating X: first, solve for the coordinate that corresponds to the cloud motion velocity vector, $\vec{w}$, and then solve for the coordinate along the orthogonal axis. Fig. 6 illustrates our method.



(a) Cloudiness estimates

(b) After temporal alignment

(c) Sorted by embedded coordinates

(d) Denoised

(e) Denoised & positioned using embedded coordinates

(f) Final cloudmap

Figure 6: Overview of our approach for cloudmap estimation using the direct method. (a-e) The intensity values represent the estimated cloudiness, the horizontal coordinate is time. (f) The final interpolated cloudmap visualization (the black-to-white gray scale colormap replaced with a white-to-blue colormap that we use in subsequent cloudmap visualizations.

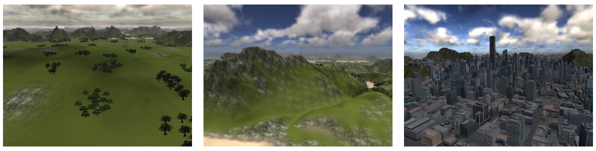


Figure 7: Example frames from the simulated scenes (Prairie, Hills, City) used for quantitative evaluation.

4.3.1. Estimating the Cloud Motion Direction Coordinate

We combine multiple pairwise temporal delays into a single global temporal delay estimate for each pixel and use this directly as the first coordinate in our embedding. For pixels representing locations roughly in line with the cloud motion direction, the delay estimates are highly correlated with spatial distance. These pairwise delays can then be generalized to a global low-dimensional embedding using linear (e.g., MDS) or nonlinear (e.g., ISOMAP [36]) dimensionality reduction methods. Unfortunately, this approach fails in the case of data points not aligned with the cloud motion direction because the temporal delay is unreliable. To overcome this problem, we propose a variant to the neighborhood selection problem in dimensionality reduction. Rather than the neighborhood consisting of *nearest* neighbors as measured by the base distance metric (in our case, temporal delay), the trusted distances are those with the highest delay-corrected correlation, ρ_{pq}.

Here, we employ a variant of ϵ-ball selection: a pixel q is considered a neighbor, $q \in N(p)$, of a pixel p if $1 - \rho_{pq} < \epsilon$. In our experiments, we selected ϵ to be the value corresponding to the top 10^{th} percentile of sorted correlations for each pixel. That is, pixels that are highly correlated after correcting for temporal delay are assumed have reliable temporal delay estimates. To obtain our first coordinate, we minimize the following linear objective function with respect to the global delay estimates, t_i, for each pixel:

$$\sum_{p,q \forall p \in N(q)} \rho_{pq}(t_q - t_p - \delta_{pq})^2. \tag{7}$$

For this coordinate, a unit step corresponds to the average distance traveled by a cloud in a single frame. Therefore, if the true cloud velocity were known this would be a metric coordinate.

4.3.2. Estimating the Ortho-Velocity Coordinate

To complete the 2D spatial embedding, we solve for the coordinate in the direction orthogonal to the cloud motion direction, o_i. We first temporally align the pixel time series using the global delay estimates, t_i, and then apply ISOMAP [36], which embeds points in a low-dimensional Euclidean space by preserving the geodesic pairwise distances of the points in original space. In order to estimate the (unknown) geodesic distances, distances are calculated between points in a trusted neighborhood and generalized into geodesic distances using an all-pairs shortest-path algorithm. Unlike the case with the cloud motion coordinate, the trusted neighborhood in this case is based on the nearest neighbors. Typically, the Euclidean distance metric is used, but other distance measures have been shown to lead to a more accurate embedding of the original data. Given the delay-corrected correlation, ρ_{pq}, between pixels p and q, our distance metric is $d(p, q) = \sqrt{(1 - \rho_{pq})}$. At this stage, we have an estimate of the 2D location, $x_i = (t_i, o_i)$ of each pixel and an implicit estimate of the cloud motion direction, $\vec{w} = (1, 0)$.

4.4. Discussion

The three algorithms presented in this section each estimate a cloudmap from a video, but are best suited for different situations. When an accurate 3D scene model is available, the known-geometry method is most appropriate as it will be more accurate than the planar method and is simpler than the direct method. Such geometry can be estimated automatically from video data using a variety of techniques, including cloud shadow motion [20, 21], but these methods require more video (often from multiple days) to obtain a high-quality scene model. If the camera will be used repeatedly to estimate a cloudmap, it may be beneficial to obtain a 3D scene model using other methods, such as structure from motion or laser scanning, as errors in the scene model can introduce artifacts in the resulting cloudmap.

Under the circumstance that the scene is known to be planar, the planar model is advantageous as it can estimate an accurate scene model with a limited amount of data. Finally, if the scene is not planar and a 3D scene model is unavailable, the direct estimation method should be applied as it avoids the step of estimating a 3D model and does not introduce artifacts for non-planar scenes.

5. Evaluation

Since ground-truth scene geometry or precise weather information (e.g., cloud motion direction) are not available for many videos, we take a multifaceted approach to evaluation, using a mix of visual and quantitative measures on real and synthetic videos. As a pre-processing step, we construct a mask to ignore regions in the sky and always in shadow and then use low-variance sampling to select a subset of ground pixels to use as landmarks. Using more landmark pixels results in more accurate results but requires additional computation. We empirically determined that $k = 200$ landmark pixels provides a good tradeoff between accuracy and efficiency and use this value in all experiments.

5.1. Datasets

We use two datasets to evaluate our methods, all of which are available at http://cs.uky.edu/~jacobs/data. For quantitative evaluation of cloudmap extraction, we generated simulated scenes using Unity Pro[1] and the Nuaj' weather simulator plugin[2]. These tools are typically used for rendering high fidelity environments for video games. For our experiments, we modeled three environments similar to scenes commonly captured by outdoor webcams. Fig. 7 shows example frames from the Prairie, Hills, and City scenes. Using the weather simulator, moving clouds (and the associated cast shadows) were introduced into each scene and the rendered scene was recorded. Each video was captured at 800 × 600 resolution and contains roughly 3,500 frames. This corresponds to about 30 minutes of video captured at 2 fps. For qualitative evaluation we use a collection of videos downloaded from Internet video sharing sites [20]. The videos in this dataset represent a wide variety of scenes, cloud conditions, and video quality levels.

5.2. Quantitative Evaluation on Synthetic Data

Fig. 8 shows a visual comparison of small sections of the extracted cloud maps. Each image can be interpreted like Fig. 6f, the only difference is the process by which it was generated. The top row represents the ground-truth clouds passing over the scene, extracted from the simulator. The next three rows show the cloudmaps extracted by each of the three methods. The methods perform similarly on the City scene. However, on the Prairie and Hills scenes the known-geometry method has noticeable banding artifacts which are due to errors in its estimate of the scene geometry. The planar method shows similar artifacts for the (non-planar) Hills scene. Tbl. 1 shows the correlation, computed by converting each image into a vector, between ground-truth and estimated cloudmaps for the methods for each scene. These results are in line with the visual results, showing the strength of the planar approach on the flat Prairie scene and the benefit of the direct approach on the Hills scene.

Table 1: Correlation between ground-truth and estimated cloudmap for different methods and scenes.

	Prairie	Hills	City
Known-Geo	0.805	0.721	0.746
Planar	0.825	0.773	0.751
Direct	0.813	0.844	0.746

5.3. Qualitative Evaluation on Real-World Scenes

Fig. 9 shows the cloudmaps extracted using the planar (top), known-geometry (middle) and direct (bottom) methods, where white indicates full attenuation and dark blue is unattenuated. Similar to the synthetic results, depending on how well the scene matches the assumptions of the approach, the methods extract plausible cloudmaps in a variety of scene types. In general, the direct method generates coordinates with fewer artifacts. For example, in the third row, the known-geometry coordinates exhibit artifacts in the region corresponding to the row of trees. Fig. 10 shows illustrative examples of embeddings created using the known-geometry and direct methods. The embeddings capture the horizontal spatial layout of the scene; we visualize them by constructing two false-color images, one for each embedding coordinate. The first coordinate corresponds to the direction the clouds are moving and the second is in the orthogonal direction. These examples highlight that our algorithms are clearly extracting geometric information about that scene.

6. Applications

We give two proof-of-concept examples that demonstrate the potential of cloudmaps for applications in surveillance and graphics.

[1] http://unity3d.com/unity/
[2] http://www.nuaj.net/

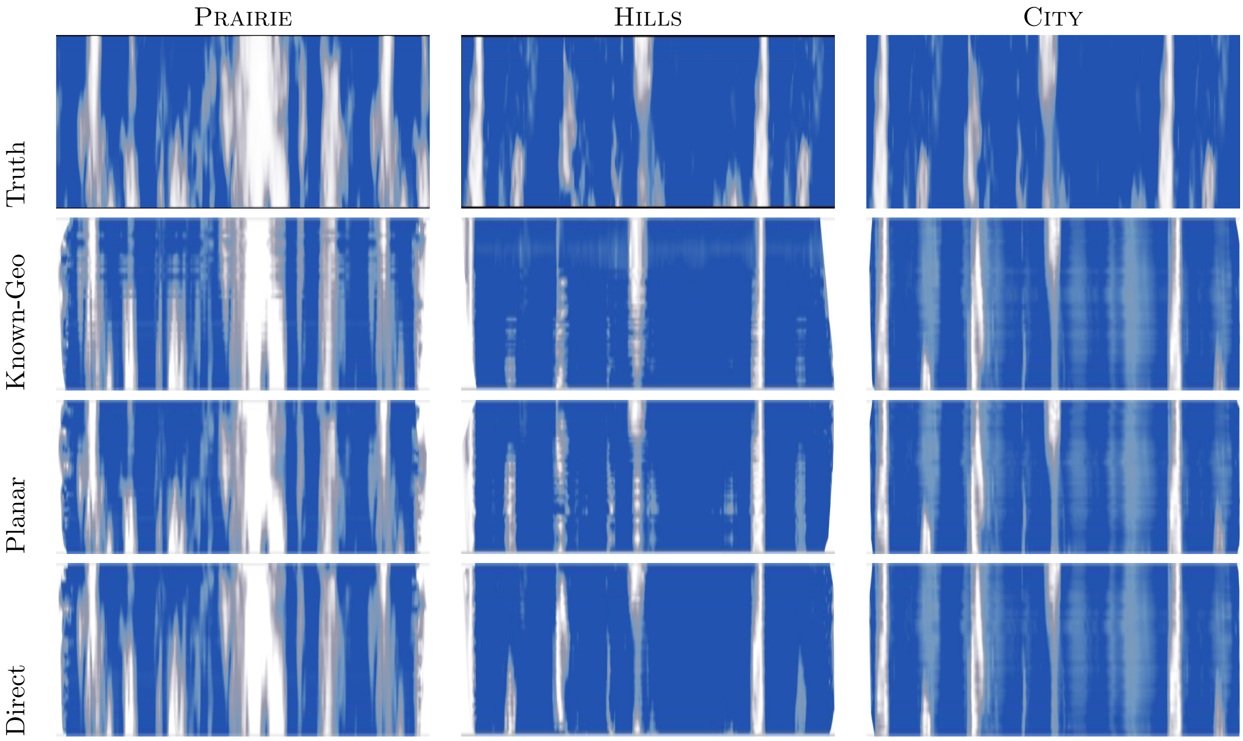


Figure 8: Cloudmaps extracted from simulated scenes (Fig. 7) and rendered as false-color images. Each column shows a segment of the ground-truth cloud pattern (top) and the extracted cloudmaps using the known-geometry method (Alg. 1, Sec. 4.1), the planar model (Alg. 2, Sec. 4.2) and the direct method (Alg. 3, Sec. 4.3). For each image, the blue regions were free of clouds and white regions had clouds that fully shaded the sun.

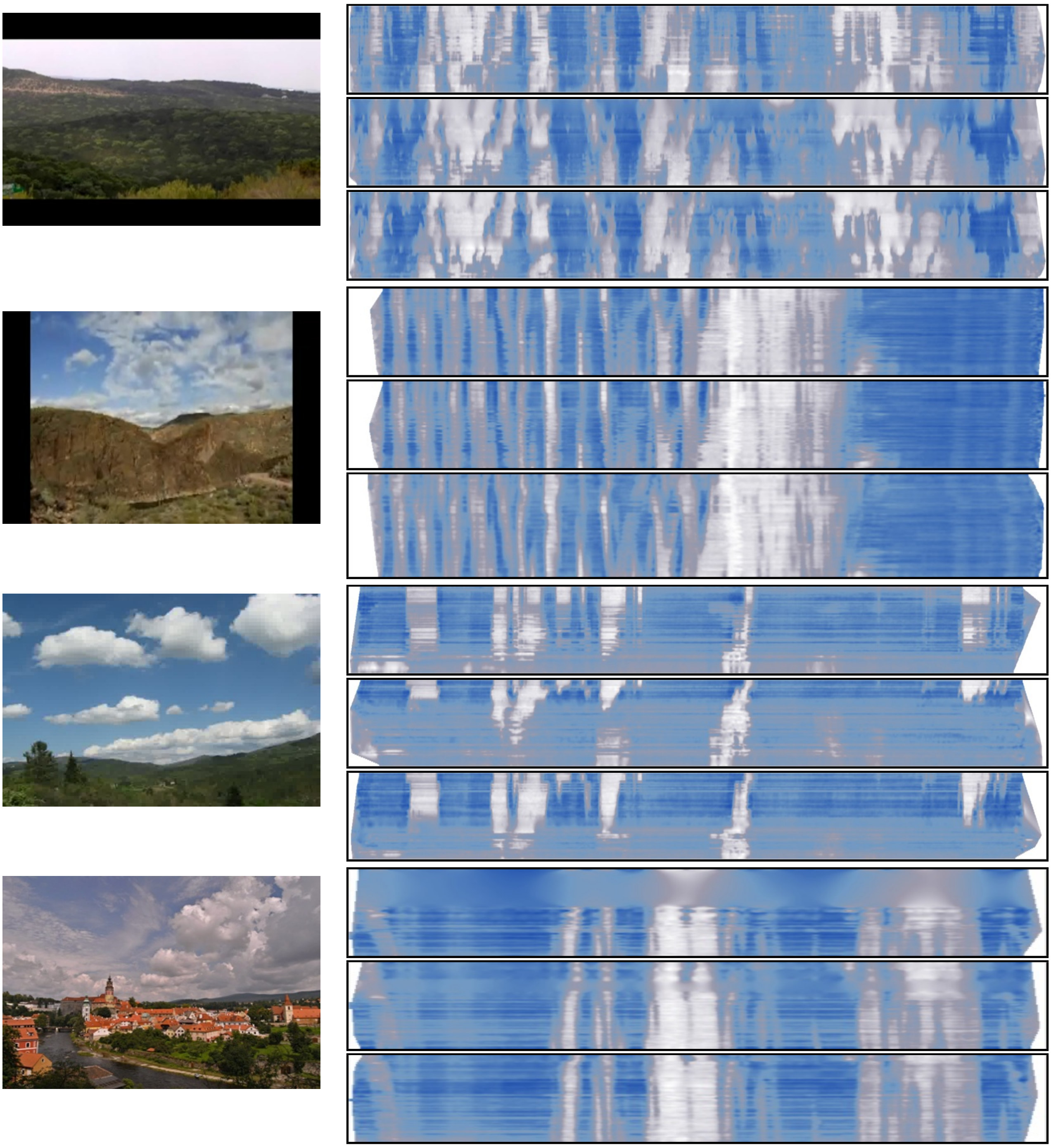


Figure 9: Cloudmaps created using the planar (top), known-geometry (middle) and direct method (bottom). The x-axis corresponds to the cloud motion direction.

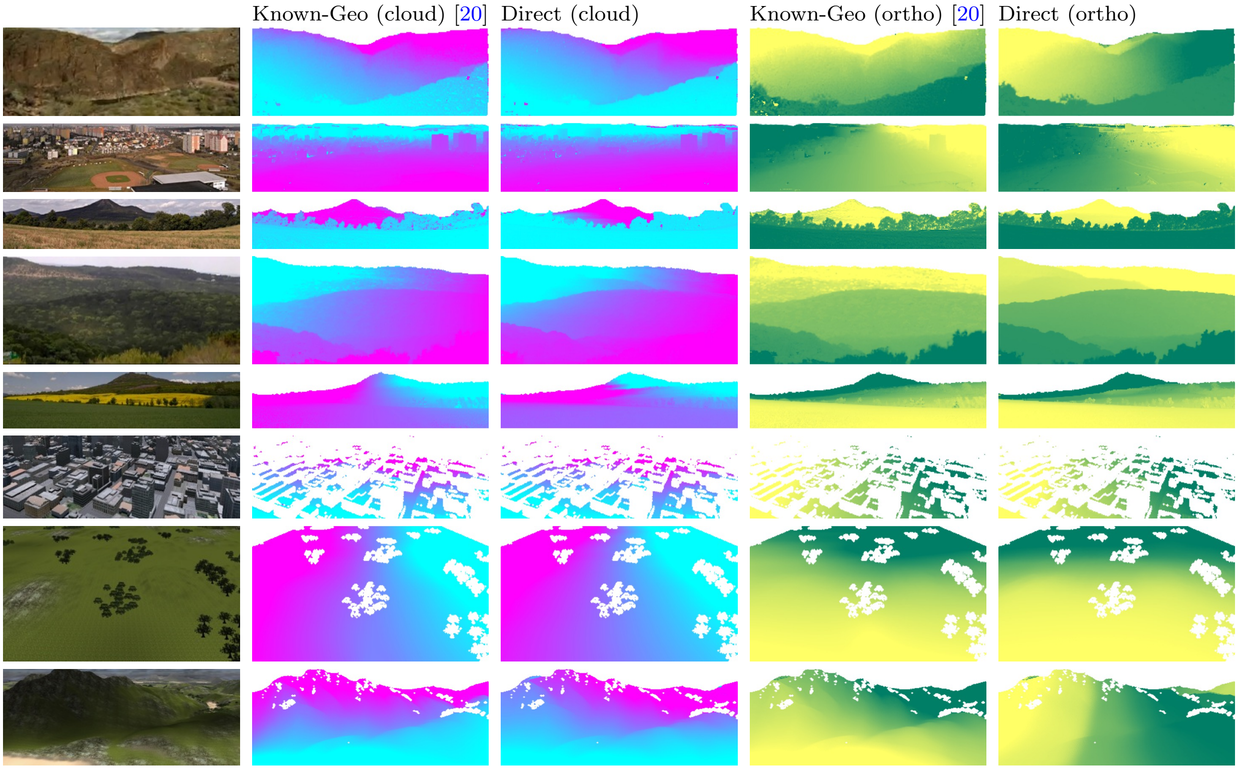


Figure 10: False-color images that represent scene shape. For each scene, we show the coordinates that correspond to the cloud motion direction (cloud) and the orthogonal direction (ortho). In each, the white pixels correspond to masked out regions and the colormap is scaled to range from the 10th to the 90th percentile of the underlying data.

6.1. *Background Modeling*

Partly cloudy conditions are challenging for color-based background modeling systems. A frequent type of error is false positive detections due to intensity changes caused by cloud shadows. To overcome this, adaptive mixture model (AMM) approaches [37] learn the various color modes in the data. However, such approaches can give false negative detections when objects pass in front of surfaces with different intensities because they learn to ignore intensity changes. We propose to remove the impact of clouds first and learn a background model of the "declouded" videos.

Given a video, $\{I(t)\}$, we first use the manifold denoising process described in Sec. 3.1.4 to generate a new video, $\{\hat{I}(t)\}$, without small transient objects. We then construct a declouded video in which each frame is $I(t) - (\hat{I}(t) + \bar{I})$, where $\bar{I} = E[I(t)]$ is the average image of the video. This results in a video showing only changes due to small objects, not clouds. Fig. 11 shows an example from one video that we use as input to further processing.

For qualitative evaluation, we train two different AMMs: one on the raw video and one on the declouded video. Results, shown in Fig. 12, demonstrate that declouding the video before learning a background model can improve the set of detections returned. These improvements include a reduction in the number of false positives, due to rapid appearance changes from cloud shadows, and false negatives, due to mixture components with high variance.

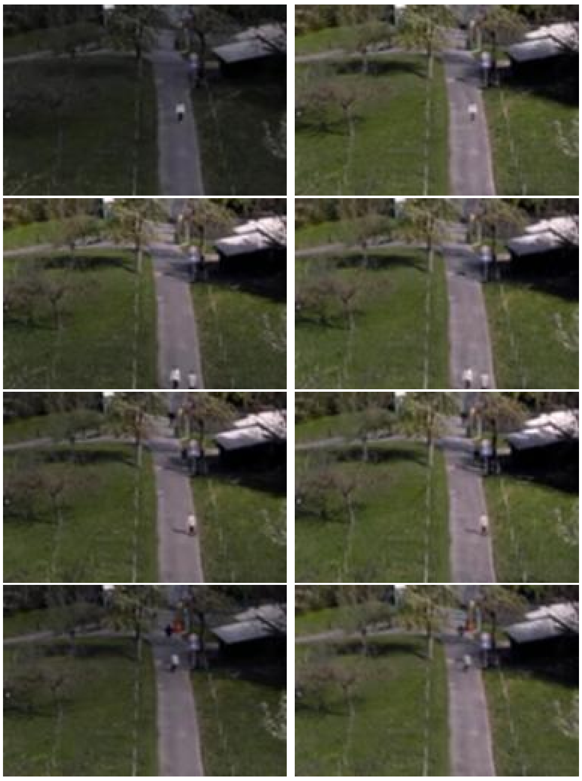
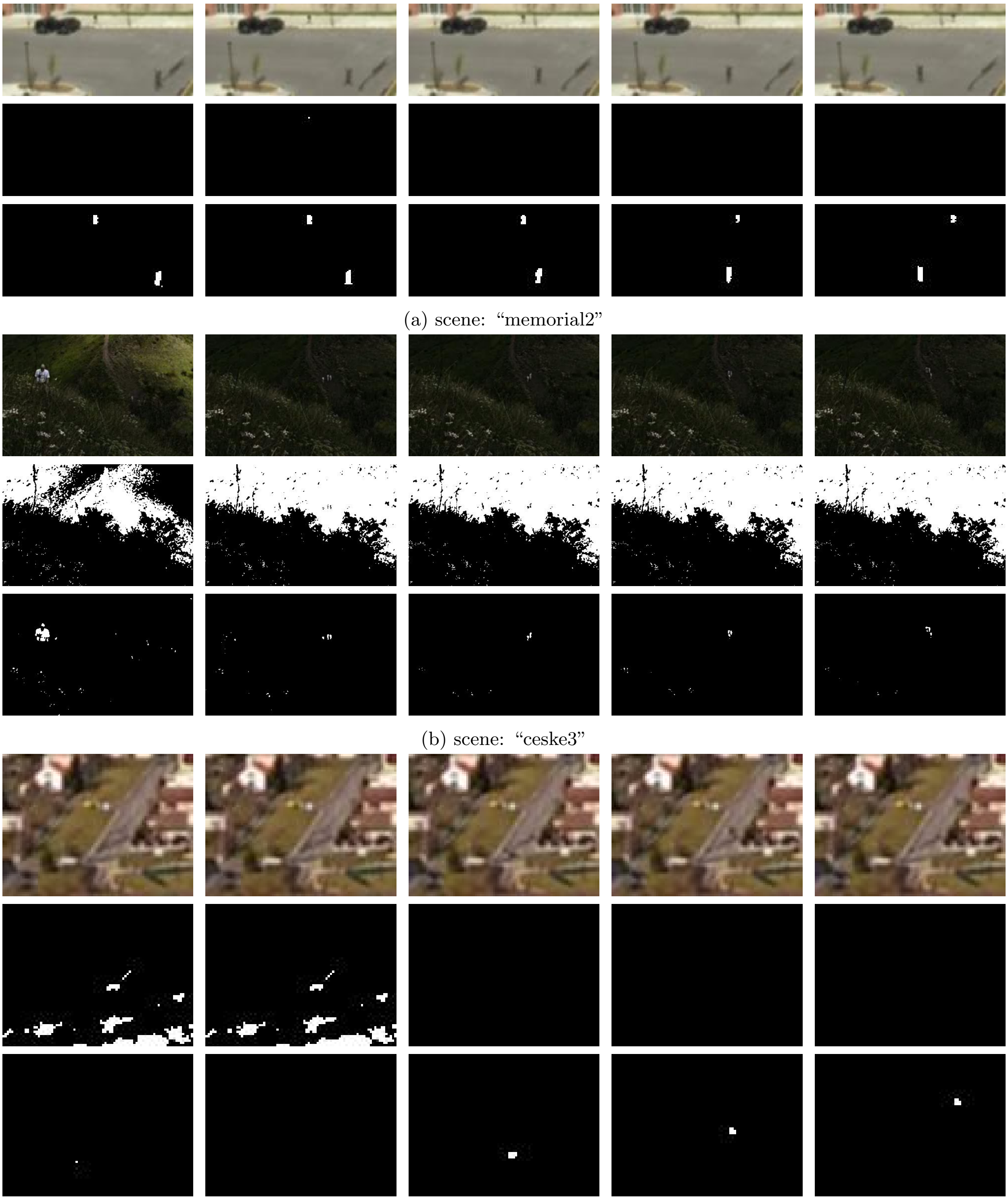


Figure 11: Raw frames from an outdoor scene (left) and the corresponding "declouded" frame (right).

6.2. *Scene Relighting*

The appearance of a scene can be changed by directly editing the 2D rendering of the cloudmap. After fitting the model and estimating and editing the cloudmap, the steps for rendering the scene are as follows: extend the landmark pixel location estimates to all pixels, for each time step lookup the location in the cloudmap, then re-render the scene using the scene generative model. To extend the location estimates, we solve a large linear system of the form (7) to estimate the coordinate in the cloud-motion direction and use Nadaraya-Watson kernel regression (with a Gaussian kernel) for the coordinate orthogonal to the cloud motion direction. Fig. 13 shows the results of introducing artificial clouds into a real scene using the min/max image formation model. For this example, we manually created the new cloudmap (Fig. 13a) using an image editing tool (white regions correspond to thick

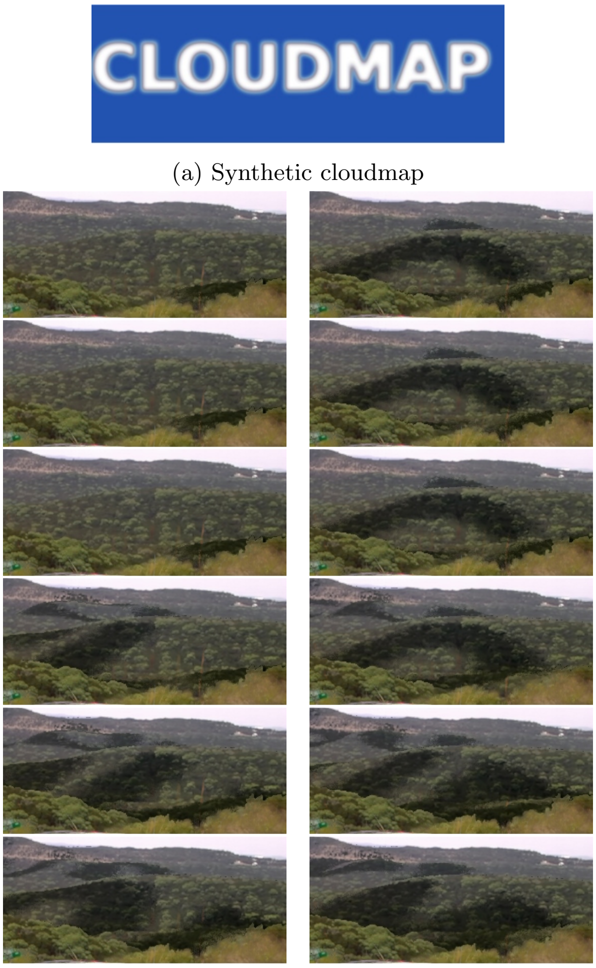


(a) scene: “memorial2”

(b) scene: “ceske3”

(c) scene: “praha4”

Figure 12: Example foreground object detections generated by an AMM on three outdoor scenes. For each, the top row is the raw frame, the second row are detections from the raw video and the bottom are detections from the “declouded” video.



(a) Synthetic cloudmap

(b) Simulated clouds

Figure 13: A simple example of the use of a cloudmap for a graphics application. A synthetic cloudmap (a) is used to relight the scene (b).

clouds that fully block the sun, blue regions correspond to clear skies). This straightforward demonstration of cloudmaps is meant to be indicative of many new applications that are possible with scene models that incorporate information, such as cloud shadows, which had previously been considered as a nuisance in outdoor scene modeling.

7. Conclusion

We introduced the concept of a cloudmap and described several methods for estimating them from video of an outdoor scene captured by a static camera. This work is a step toward the goal of enabling software systems to better understand the local weather conditions. Our approach does not require a specialized sky camera; a simple surveillance camera is sufficient. Numerous applications are possible using this basic framework including: measuring available solar energy, mapping local cloud motion patterns, background modeling for surveillance, and scene relighting for computer graphics.

Our approach uses the scene as a large distributed imaging system for understanding the sky. This imaging system has a unique two-stage geometry. The first stage, in which light from the sun is projected onto the scene, is essentially a large orthographic camera, because solar light rays are almost parallel by the time they reach the Earth. If we could distribute pixels around the scene, we could directly measure the intensity of this light incident on each scene point and obtain a map of the attenuation layer of the clouds. In the second stage, we use a single central camera to measure the reflected light off of each scene element. The main challenge we addressed is that the scene geometry is unknown and the spatial sampling usually has gaps (unless the scene is planar).

This work is an example of a recent trend [38, 39, 40] exploring the use of unusual imaging geometries to measure natural phenomena, often labeled as compressive sensing. We think that this approach has the potential to significantly improve upon the performance, and reduce the cost, as compared to traditional projective sensors.

Acknowledgments

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA CSSG D11AP00255) and the National Science Foundation (EPSCoR 0918856). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of the Department of Defense or the U.S. Government.

References

[1] G. Szeicz, Solar radiation for plant growth, Journal of Applied Ecology (1974) 617–636.

[2] R. Marquez, C. F. Coimbra, Intra-hour DNI forecasting based on cloud tracking image analysis, Solar Energy 91 (2013) 327–336.

[3] R. Eastman, S. G. Warren, A 39-yr survey of cloud changes from land stations worldwide 1971–2009: Long-term trends, relation to aerosols, and expansion of the

tropical belt., Journal of Climate 26 (4) (2013) 1286–1303.

[4] K. Sunkavalli, W. Matusik, H. Pfister, S. Rusinkiewicz, Factored time-lapse video, ACM Transactions on Graphics (Proc. SIGGRAPH) 26 (3).

[5] F. Langguth, J. Ackermann, S. Fuhrmann, M. Goesele, Photometric stereo for outdoor webcams, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[6] A. Abrams, C. Hawley, R. Pless, Heliometric stereo: Shape from sun position, in: European Conference on Computer Vision, 2012.

[7] L. Shen, P. Tan, Photometric stereo and weather estimation using internet images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[8] C. Murdock, N. Jacobs, R. Pless, Webcam2satellite: Estimating cloud maps from webcam imagery, in: IEEE Workshop on Applications of Computer Vision, 2013.

[9] M. Islam, N. Jacobs, H. Wu, R. Souvenir, Images+weather: Collection, validation, and refinement, in: CVPR Workshop on Ground Truth, 2013.

[10] S. A. Ackerman, S. K. Cox, Comparison of satellite and all-sky camera estimates of cloud cover during gate, Journal of Applied Meteorology 20 (5) (1981) 581–587.

[11] C. N. Long, J. Sabburg, J. Calbó, et al., Retrieving cloud characteristics from ground-based daytime color all-sky images., Journal of Atmospheric & Oceanic Technology 23 (5).

[12] A. A. Silva, M. P. de Souza Echer, Ground-based measurements of local cloud cover, Meteorology and Atmospheric Physics 120 (3-4) (2013) 201–212.

[13] A. Kazantzidis, P. Tzoumanikas, A. Bais, S. Fotopoulos, G. Economou, Cloud detection and classification with the use of whole-sky ground-based images, Atmospheric Research 113 (2012) 80–88.

[14] D. Veikherman, A. Aides, Y. Y. Schechner, A. Levis, Clouds in the cloud, in: ACCV, Springer, 2014, pp. 659–674.

[15] L. Tao, L. Yuan, J. Sun, Skyfinder: attribute-based sky image search, ACM Transactions on Graphics 28 (3) (2009) 68:1–68:5.

[16] K.-C. Peng, T. Chen, Incorporating cloud distribution in sky representation, in: IEEE International Conference on Computer Vision, 2013.

[17] Y. Chu, H. T. C. Pedro, C. F. M. Coimbra, Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning, Solar Energy 98 (2013) 592–603.

[18] N. Jacobs, J. King, D. Bowers, R. Souvenir, Estimating Cloud Maps from Outdoor Image Sequences, in: IEEE Winter Conference on Applications of Computer Vision, 2014.

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

[20] N. Jacobs, A. Abrams, R. Pless, Two cloud-based cues for estimating scene structure and camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence (2013) 2526–2538.

[21] S. Workman, R. Souvenir, N. Jacobs, Scene Shape Estimation from Multiple Partly Cloudy Days, Computer Vision and Image Understanding (2015) 116–129.

[22] N. Jacobs, M. Islam, S. Workman, Cloud motion as a calibration cue, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[23] A. Prati, I. Mikic, M. M. Trivedi, R. Cucchiara, Detecting moving shadows: Algorithms and evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 918–923.

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

[25] J.-F. Lalonde, A. A. Efros, S. G. Narasimhan, Detecting ground shadows in outdoor consumer photographs, in: European Conference on Computer Vision, 2010.

[26] J.-F. Lalonde, A. A. Efros, S. G. Narasimhan, Estimating natural illumination from a single outdoor image, in: IEEE International Conference on Computer Vision, 2009.

[27] Q. Li, W. Lu, J. Yang, J. Wang, Thin cloud detection of all-sky images using markov random fields, IEEE Geoscience and Remote Sensing Letters 9 (3) (2012) 417–421.

[28] N. Igawa, Y. Koga, T. Matsuzawa, H. Nakamura, Models of sky radiance distribution and sky luminance distribution, Solar Energy 77 (2) (2004) 137–157.

[29] J. Ackermann, F. Langguth, S. Fuhrmann, M. Goesele, Photometric stereo for outdoor webcams, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[30] S. Bjorklund, L. Ljung, A review of time-delay estimation techniques, in: IEEE Conference on Decision and Control, Vol. 3, 2003, pp. 2502–2507.

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

[32] M. Hein, M. Maier, Manifold denoising, in: Advances in Neural Information Processing Systems, 2006.

[33] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005.

[34] A. Elgammal, C. Lee, Separating style and content on a nonlinear manifold, in: IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[35] R. Souvenir, R. Pless, Isomap and nonparametric models of image deformation, in: IEEE Workshop on Applications of Computer Vision, 2005.

[36] J. Tenenbaum, V. De Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[37] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 1999.

[38] R. G. Baraniuk, E. Candès, R. Nowak, M. Vetterli, Compressive sampling, IEEE Signal Processing Magazine 25 (2) (2008) 12–13.

[39] R. Fergus, A. Torralba, W. T. Freeman, Random lens imaging, Tech. rep., Massachusetts Institute of Technology: Computer Science and Artificial Intelligence Lab (Sep. 2006).

[40] A. Liutkus, D. Martina, S. Popoff, G. Chardon, O. Katz, G. Lerosey, S. Gigan, L. Daudet, I. Carron, Imaging with nature: Compressive imaging using a multiply scattering medium, Scientific Reports 4.