Estimating Cloudmaps from Outdoor Image Sequences

Nathan Jacobs and Joshua King
University of Kentucky
{jacobs,jski223}@cs.uky.edu

Daniel Bowers and Richard Souvenir
University of North Carolina at Charlotte
{dbower13,souvenir}@uncc.edu

Abstract

Cloud shadows dramatically affect the appearance of outdoor scenes. We describe two approaches that use video of cloud shadows to estimate a cloudmap, a spatio-temporal function that represents the clouds passing over the scene. Our first method makes strong assumptions about the camera geometry and estimates the cloud motion direction. Our second method uses techniques from manifold learning and does not require known geometry. Neither method requires directly viewing the clouds, but instead uses the pattern of intensity changes caused by the cloud shadows. We show renderings of cloudmaps extracted using both methods from videos of real outdoor scenes as well as quantitative results on synthetic datasets. An accurate estimate of the cloudmap has potential applications in surveillance and graphics, as well as scientific studies that depend on solar radiation.

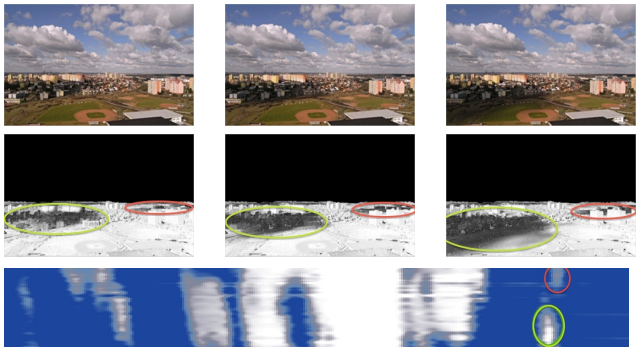


Figure 1: We estimate the sunlight attenuation due to clouds (middle) for video frames (top) to extract a cloudmap (bottom), a geo-temporal function that describes the set of clouds that passed over the scene. This cloudmap summarizes ∼25 minutes of video and the clouds corresponding to the visible shadows in the image are outlined.

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 to solar power generation. For such applications, current methods for assessing available solar energy rely 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 [17, 11, 1] or only estimate a single scalar cloudiness parameter per frame [15, 12, 6].

We propose to use outdoor cameras to estimate a *cloudmap*, a time-varying 2D function, defined in world coordinates, that describes the clouds passing over a scene (see Fig. 1). Our approach uses video of an outdoor scene to simultaneously estimate a cloudmap and a mapping between image pixels and cloudmap coordinates. This paper makes several contributions: two 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

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 [17, 18, 11, 1] and transient clouds and their shadows are strong cues to scene geometry [7] and camera calibration [9]. Outdoor videos have also been used to estimate dynamic scene properties, such as low-dimensional cloudiness models [15, 12, 6]. Our work is the first to construct highly detailed cloudiness models from video of an outdoor scene.

Shadows in outdoor scenes Methods for detecting and handling shadows appear in several settings. Most similar to our work is in the surveillance domain [13, 5] 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 [10], 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 fit a model that accounts for cloud motion.

Estimating cloud cover Murdock et al. address the problem of estimating cloud cover from a collection of ground-based cameras [12]. 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 timestep, we estimate a detailed, time-varying, cloudmap. Additionally, our methods provide additional information about scene geometry that may be useful for other applications.

3. Defining and Visualizing Cloudmaps

A cloudmap is a spatio-temporal function, $C(x, y, t) \in [0, 1]$, ranging from 'no direct sunlight' to 'full sunlight', which defines how clouds attenuate the sunlight and thus has a dramatic affect on the appearance of locations in the scene. Cloudmaps are evident in the sky (clouds that affect sunlight attenuation) and on the ground (cloud shadows are the result of sunlight attenuation). In this work, we focus on the analysis of cloud shadows on the ground, as most applications require geo-localized estimates of solar energy. 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 sun-light 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, is known.

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 [1, 2] 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 define models for the pixel intensity time-series for both short (i.e., minutes) and long (i.e., hours) video.

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

$$I(p, t) \approx \rho^1(p) C(x_p, y_p, t) + \rho^0(p). \tag{1}$$

In this model, $\rho^1(\rho^0)$ are the maximum (minimum) observed intensity over the full video. 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:

$$C(x_p, y_p, t) = \frac{I(p, t) - \rho^0(p)}{\rho^1(p)}. \tag{2}$$

For longer videos, the min/max method defined above leads to cloudiness estimates that are affected by the 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 at each pixel as follows:

$$\begin{aligned} \min_{\Theta_u} \quad & \sum_t (f(t; \Theta_u) - I(p, t))^2 \\ \text{s.t.} \quad & f(t; \Theta_u) \geq I(p, t),\ \theta_{u2} \leq 0 \end{aligned} \tag{3}$$

where $f(t; \Theta) = \theta_2 t^2 + \theta_1 t + \theta_0$. We then solve for the lower bound:

$$\begin{aligned} \min_{\Theta_l} \quad & \sum_t (f(t; \Theta_l) - I(p, t))^2 \\ \text{s.t.} \quad & f(t; \Theta_l) \leq I(p, t),\ \theta^*_{u2} \leq \theta_{l2} \leq 0 \end{aligned} \tag{4}$$

with the additional constraint that the lower bound should be less concave than the upper bound. In this model, the albedo terms ρ^0, ρ^1 are subsumed in the upper and lower bounds; the cloudiness estimate time-series for each pixel is:

$$C(x_p, y_p, t) = \frac{I(p, t) - f(t; \Theta_{lp})}{f(t; \Theta_{up}) - f(t; \Theta_{lp})}. \tag{5}$$

Fig. 2 shows an example comparing the cloudiness estimate from min/max model to the quadratic bounds model for two pixels with different surface normals.

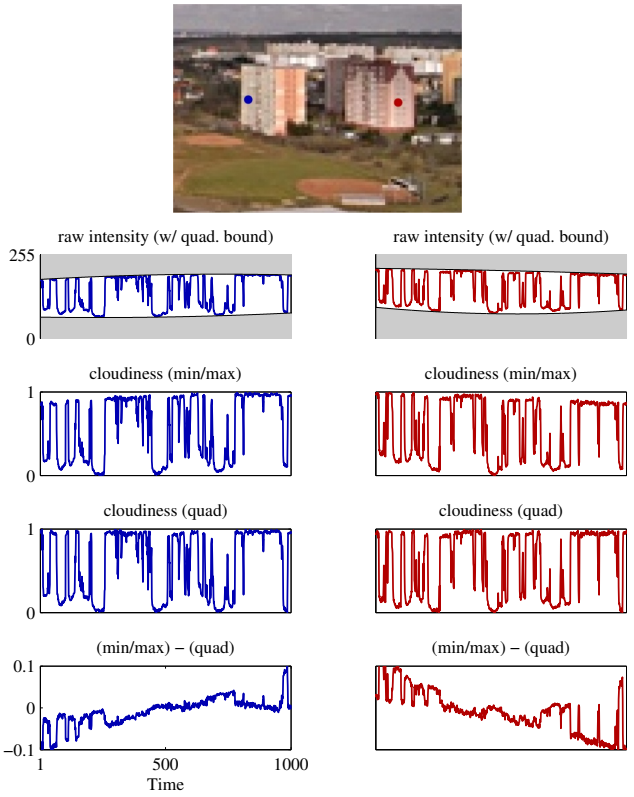


Figure 2: For the two marked pixels, each row of plots show (1) the raw intensity time-series for one hour of video, the cloudiness estimated using the (2) min/max and (3) quadratic models, and (4) the differences between the two estimates. The quadratic model is able to compensate for sun motion over time.

Estimating temporal delay Locations directly in-line with the cloud motion often have very similar patterns of intensity changes, temporally offset due to the motion of clouds. That is, $C(x, y, t) \approx C(x + w_x \Delta t, y + w_y \Delta t, t + \Delta t)$ where (w_x, w_y) is the cloud motion velocity. In the direction orthogonal to the cloud motion, points are affected by a different set of clouds, but, in general, the time series of nearby pixels are still very similar, i.e., $0 \approx \sum_t (C(x, y, t) - C(x - w_y, y + w_x, t))^2$.

Temporal delay, δ_{pq}, is computed between the cloudiness time series of two pixels, p, q, using a two-stage process [8]. The first stage measures delay using pairwise cross correlation to obtain an initial, integer estimate of δ_{pq} and correlation, ρ_{pq}. For sub-frame accuracy, this estimate is refined using a quadratic model that estimates correlation, $\hat{\rho}_{pq}$, for delays in a small window around the original estimate, δ_{pq}. The final delay maximizes:

$$\frac{(\hat{\rho}_{pq} - \rho_{pq})^2}{(\hat{\rho}^*(\rho) - \rho_{pq})^2} \tag{6}$$

where $\hat{\rho}^*(\rho)$ is a normalization constant, based on the maximum value of $\hat{\rho}$ in small window (one time-step) of the original estimate.

Removing noise due to transient objects Denoising the cloudiness time series can remove artifacts caused by localized, non-cloud related appearance changes, such as moving plants or vehicles. Our approach, similar to manifold denoising [4], relies on the manifold structure of the time series and aims to filter out causes of appearance change that are purely local in nature, but retain those related to the clouds. To accomplish this, we linearly reconstruct the time series for each pixel using a collection of temporally aligned time series that have high delay-corrected correlation, $\hat{\rho}_{pq}$, but are not close in estimated 2D coordinates or image coordinates, so that localized changes will not be in the basis used to reconstruct the time series, and hence these changes will be minimized. In Sec. 6.1, we show how this technique can be applied to detecting foreground objects.

Cloudmap interpolation and 2D rendering Unless the scene is planar, the complete cloudmap for a region is not completely 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 [14] to predict the cloudiness. As training data for the regression model, we use cloudiness estimates, $C(x_p, y_p, t)$, as targets and motion-adjusted location parameters, $(x_p - w_x t, y_p - w_y t, t)$, as predictors. We then fit covariance and noise model parameters using nonlinear optimization. By adjusting for cloud motion, we can use a simple isotropic squared exponential kernel to obtain accurate predictions. The weights of 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.

Since it is difficult to visualize a 3D cloudmap, we present a simple method for rendering a 3D 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, but replace the 3D predictors with the 2D predictors, $(x_{pt}, y_{pt}) = (x_p, y_p) - (tw_x, tw_y)$. Essentially, the clouds serve as an inertial reference frame; a model is fit to the points and queried to obtain a 2D cloudmap.

4. Estimating Scene Models

We present two new methods for estimating scene shape and cloud motion direction from video. Both take as in-

put the cloudiness time series, temporal delay estimates and delay-corrected correlations for a subset of the scene pixels. The first method assumes known scene geometry, or uses an existing method [7] to explicitly estimate the geometry. The second method uses manifold learning techniques and is suitable when little is known about the camera calibration and orientation.

4.1. Known-Geometry Cloudmap Estimation

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 distance projected onto the cloud motion vector for all pairs of pixels.

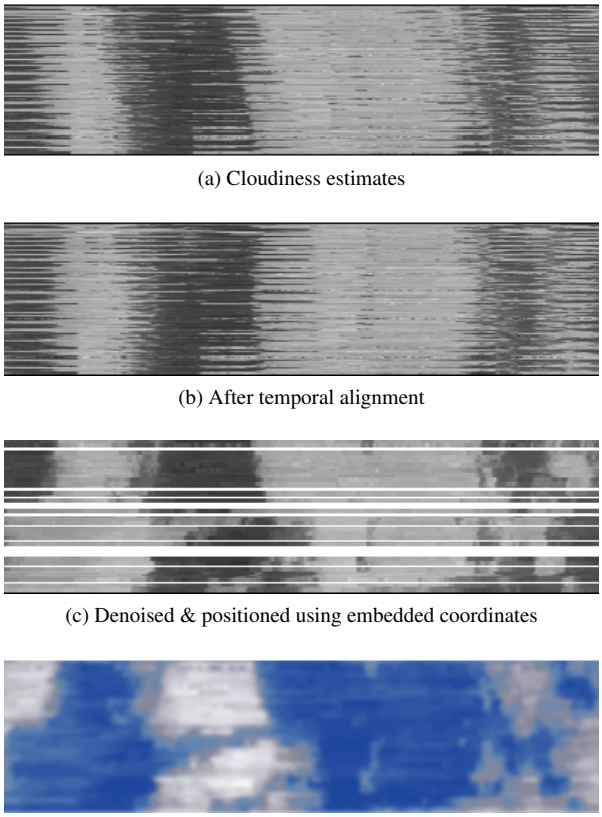
When the scene geometry is not known in advance, which is the case for all real scenes used in this work, we use an existing method for scene shape estimation from cloud shadows [7] (ProjNMDS). The basic idea is to use nonlinear optimization to solve for the scene geometry given a video captured on a partly cloudy day. While this method was not originally intended to produce cloudmaps, we adapt it for this application by adding the cloud motion direction estimate. Given this, we can use the cloud map denoising methods and rendering methods described in the previous section. While this approach can work well, the 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. In the next section, we describe a method that is simpler to use in practice and directly estimates the cloudmap.

4.2. Geometry-Free Cloudmap Estimation

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 [3, 16], 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. 3 illustrates our method.

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 out 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 [19]) 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 the base distance metric (in our case, temporal delay), the trusted distances are those with the highest delay-corrected correlation, $\hat{\rho}_{pq}$.

Here, we employ a variant of ϵ-ball selection: a pixel q is considered a neighbor of a pixel p if $1 - \hat{\rho}_{pq} < \epsilon$. In our experiments, we selected ϵ to be the value corresponding to the top 10^{th} percentile of sorted correlations for each



(a) Cloudiness estimates

(b) After temporal alignment

(c) Denoised & positioned using embedded coordinates

(d) Final cloudmap

Figure 3: Overview of our approach for cloudmap estimation. In each image, the intensity (color) values represent the estimated cloudiness, and the horizontal coordinate is time.

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:

$$\sum_{p,q \forall p \in N(q)} \hat{\rho}_{pq}(t_q - t_p - \delta_{pq})^2. \quad (7)$$

In this first 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.

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 [19], 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, $\hat{\rho}_{pq}$, between pixels p and q, our distance metric is $d(p,q) = \sqrt{(1-\hat{\rho}_{pq})}$. 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, $\vec{w} = (1, 0)$.

5. Evaluation

Since ground-truth scene geometry or precise weather information (e.g., cloud motion direction) are not available for many videos, for evaluation, we take a multi-faceted approach 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.

Datasets We use two datasets to evaluate our methods. 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. 4 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 roughly corresponds to 30 minutes of video captured at 2 fps. For qualitative evaluation we use a collection of videos downloaded from Internet video sharing sites [7]. The videos in this dataset represent a wide variety of scenes, cloud conditions and video quality levels.

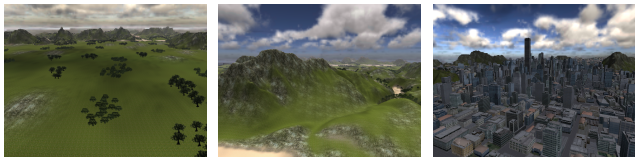


Figure 4: Example frames from the simulated scenes (PRAIRIE, HILLS, CITY) used for quantitative evaluation.

	Prairie	Hills	City
ProjNMDS	0.805	0.721	0.746
Geo-free	0.813	0.844	0.746

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

Quantitative Evaluation on Synthetic Data Fig. 5 shows a visual comparison of small sections of the extracted cloud maps. The top row represents the ground truth cloudmap passing over the scene, extracted from the simulator. The middle and bottom rows show the cloudmaps extracted by the ProjNMDS and geometry-free methods, respectively. Both methods perform similarly on the CITY scene. However, on the PRAIRIE and HILLS scenes the ProjNMDS method has noticeable banding artifacts which are due to errors in its estimates of scene geometry. Table 1 shows the correlation between ground-truth and estimated cloudmaps for both methods and all scenes. These results confirm that the geometry-free method gives results that are closer to the ground truth on the PRAIRIE and HILLS scenes.

Qualitative Evaluation on Real-World Scenes Fig. 7 show the cloudmaps extracted using the ProjNMDS (top)

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

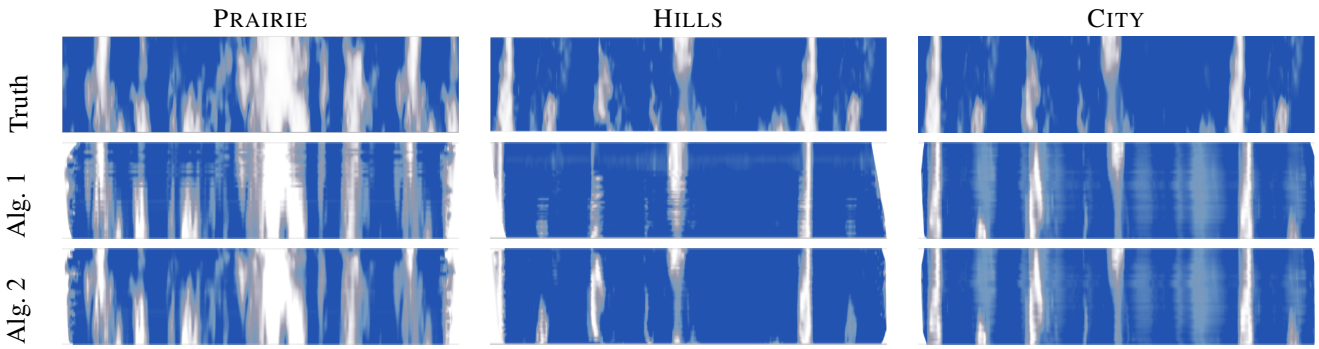


Figure 5: Cloudmaps extracted from simulated scenes. Each row shows a segment of the ground truth cloud pattern (top) and the extracted cloudmaps using ProjNMDS (Alg. 1, Sec 4.1) and the geometry-free method (Alg. 2, Sec 4.2).

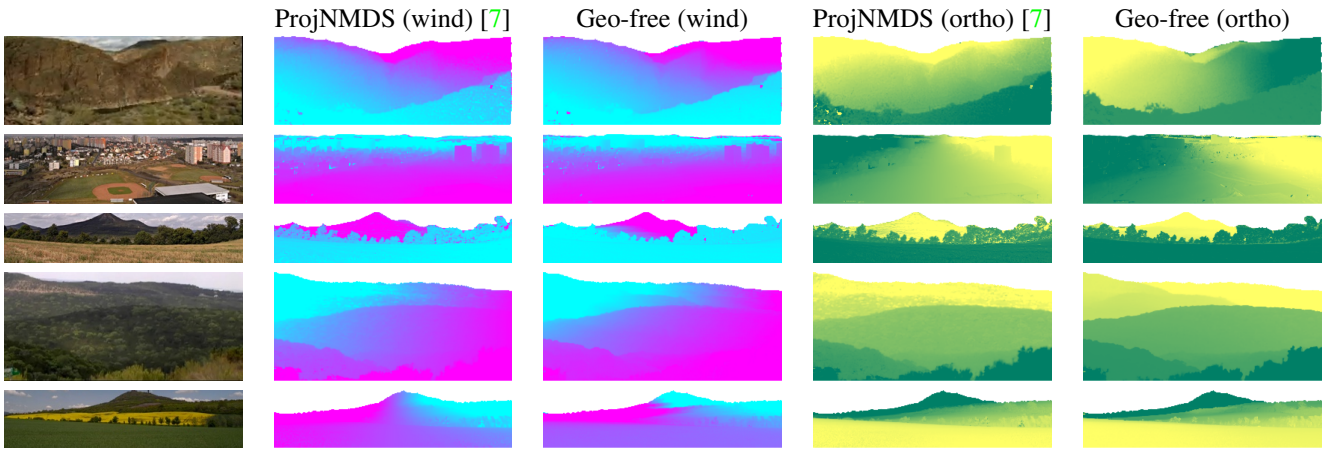


Figure 6: False-color images that represent scene shape. For each scene, we show the coordinates that correspond to the cloud motion direction (wind) and the orthogonal direction (ortho), with white regions that correspond to sky.

and geometry-free (bottom) approaches (where white is fully attenuated and dark blue is unattenuated). Both methods extract plausible cloudmaps in a variety of scene types. Fig. 6 shows an illustrative example of an embedding created using both methods. From this we can see that the geometry-free method generates coordinates with fewer artifacts. For example, in the third row, the ProjNMDS coordinates show artifacts in the region corresponding to the row of trees.

6. Applications

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

6.1. Background Modeling

Transient objects in a scene are removed by fitting the geometry-free model and using the aligned cloudiness time series to predict pixel intensities. To predict, a linear regression model is fit from cloudiness to pixel intensity for each pixel in the scene. Cloudiness time series are chosen that are similar, after compensating for temporal delay, but not close in image space. This last restriction is important, because it prevents small transient objects from being included in the predictor basis, and hence causes them to be left out of the predictions. This gives a model that converts cloudiness time series to pixel intensities for every pixel in the scene that can be used to predict the "object-free" background image that includes appearance changes due to clouds. Fig. 8 shows an example of using this approach for background subtraction in an outdoor scene.

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 timestep lookup the location in the cloudmap, then re-render the scene using the scene generative model. To extend the loca-

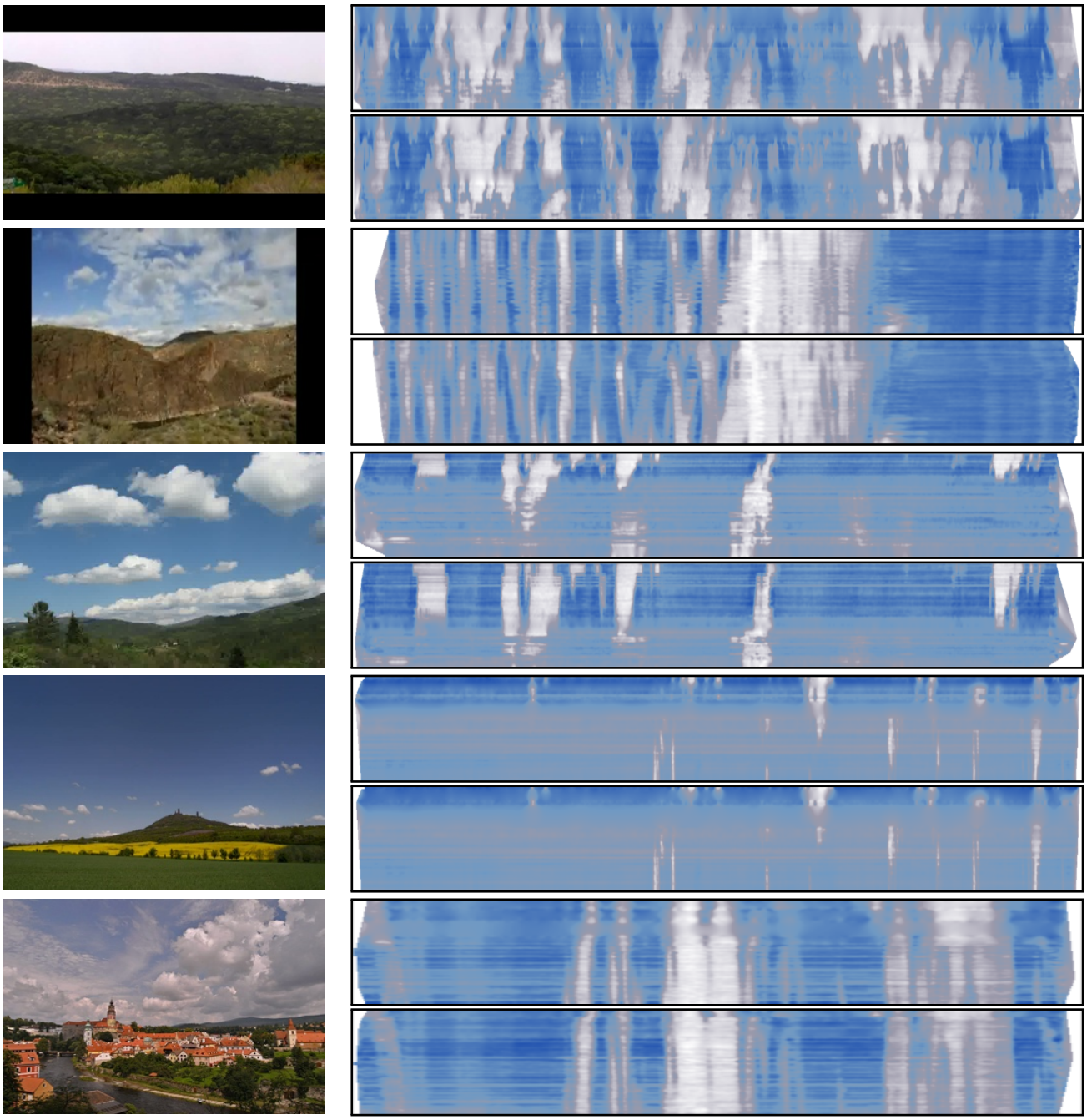


Figure 7: Cloudmaps created using the ProjNMDS (top) and geometry-free method (bottom). The x-axis corresponds to the cloud motion direction.

tion estimates, we solve a large linear system of the form (6) to estimate the cloud-motion coordinates and use locally weighted linear regression to estimate the orthogonal coordinates. Fig. 9 shows the results of introducing artificial clouds into a real scene.

7. Conclusion

We introduced the concept of a cloudmap and described several methods for estimating it from video. Numerous applications are possible using this basic framework including: cloud measurement, cloud editing, image relighting and temporal super-resolution. We believe this is an im-

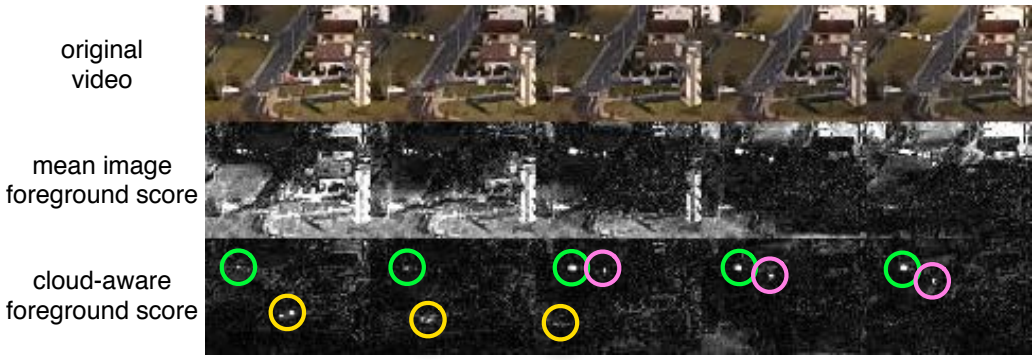


Figure 8: The top row shows frames from a video of a road and a few small buildings. Background subtraction using a mean image (middle) gives high foreground score (white) for appearance changes due to cloud shadows. The bottom row shows cloudmap-aware background subtraction where two vehicles and one pedestrian are now detected in the foreground.

(a) Synthetic cloudmap

(b) Simulated clouds

Figure 9: A simple example of the use of a cloudmap for a graphics application. (a) A synthetic cloudmap. (b) Relighting the scene using the synthetic cloudmap.

portant step toward constructing outdoor scene models that can incorporate a wide variety of weather phenomena.

Acknowledgements

This research was supported by DARPA CSSG D11AP00255 and NSF EPSCoR 0918856.

References

[1] A. Abrams, C. Hawley, and R. Pless. Heliometric stereo: Shape from sun position. In *ECCV*, 2012.

[2] J. Ackermann, F. Langguth, S. Fuhrmann, and M. Goesele. Photometric stereo for outdoor webcams. In *CVPR*, 2012.

[3] A. Elgammal and C. Lee. Separating style and content on a nonlinear manifold. In *CVPR*, 2004.

[4] M. Hein and M. Maier. Manifold denoising. In *NIPS*, 2006.

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

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

[7] N. Jacobs, A. Abrams, and R. Pless. Two cloud-based cues for estimating scene structure and camera calibration. *PAMI*, 2013.

[8] N. Jacobs, B. Bies, and R. Pless. Using cloud shadows to infer scene structure and camera calibration. In *CVPR*, 2010.

[9] N. Jacobs, M. Islam, and S. Workman. Cloud motion as a calibration cue. In *CVPR*, 2013.

[10] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010.

[11] F. Langguth, J. Ackermann, S. Fuhrmann, and M. Goesele. Photometric stereo for outdoor webcams. In *CVPR*, 2012.

[12] C. Murdock, N. Jacobs, and R. Pless. Webcam2satellite: Estimating cloud maps from webcam imagery. In *WACV*, 2013.

[13] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *PAMI*, 2003.

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

[15] L. Shen and P. Tan. Photometric stereo and weather estimation using internet images. In *CVPR*, 2009.

[16] R. Souvenir and R. Pless. Isomap and nonparametric models of image deformation. In *WACV*, 2005.

[17] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2007.

[18] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? In *CVPR*, 2008.

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