# Two Cloud-Based Cues for Estimating Scene Structure and Camera Calibration

Nathan Jacobs, *Member, IEEE,* Austin Abrams, *Member, IEEE,* and Robert Pless, *Member, IEEE*

**Abstract**—We describe algorithms that use cloud shadows as a form of stochastically structured light to support 3D scene geometry estimation. Taking video captured from a static outdoor camera as input, we use the relationship of the time series of intensity values between pairs of pixels as the primary input to our algorithms. We describe two cues that relate the 3D distance between a pair of points to the pair of intensity time series. The first cue results from the fact that two pixels that are nearby in the world are more likely to be under a cloud at the same time than two distant clouds. We describe methods for using this cue to estimate focal length and scene structure. The second cue is based on the motion of shadow clouds across the scene; this cue results in a set of linear constraints on scene structure. These constraints have an inherent ambiguity, which we show how to overcome by combining the cloud motion cue with the spatial cue. We evaluate our method on several time lapses of real outdoor scenes.

**Index Terms**—time lapse, depth map, non-metric multidimensional scaling, image formation, shape from shadows, clouds

## 1 Introduction

Although clouds are among the dominant features of outdoor scenes, with few exceptions visual inference algorithms treat their effects on the scene as noise. However, the shadows they cast on the ground over time give novel cues for inferring 3D scene models. Clouds are one instantiation of the first law of geography, due to Waldo Tobler:*"Everything is related to everything else, but near things are more related than distant things."* In a sense, we are applying this law to the problem of estimating a depth map from time-lapse imagery. The basic insight is that there is a relationship between the time series of intensity at two pixels and the distance between the imaged scene points. We describe two cues and present algorithms that use these cues to estimate a depth map.

For the first cue, we compute the temporal correlation between pairs of pixels and observe that if the relationship between this correlation and 3D distance is known then there is a simple problem: Given an image and the 3D distance between every pair of scene points, find the 3D model of the scene that is consistent with the camera geometry and the distance constraints. However, the relationship between correlation and distance is unknown because it depends on the scene and the type of clouds in the scene. Thus, we derive and present a method that simultaneously solves for the relationship between distance and correlation and for a corresponding 3D scene model.

The second cue requires higher frame-rate video and the ability to estimate the temporal delay between a pair of pixel-intensity time series. This temporal delay, coupled with knowledge of the wind velocity, allows us to define a set of linear constraints on the scene geometry. These constraints also define a clean geometric problem: Given an image and the distance between every pair of pixels projected onto the wind direction, solve for a 3D scene structure that is consistent with the projected distances and the camera geometry.

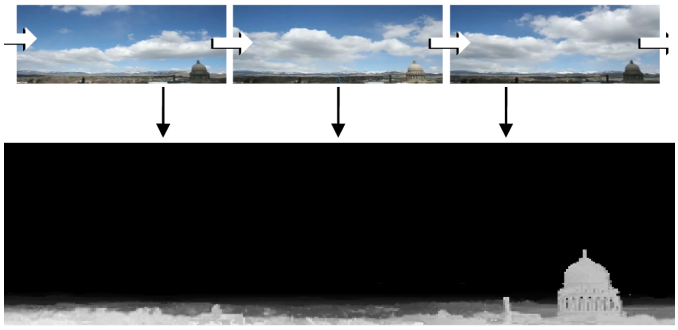


Fig. 1

Our work falls into the broad body of work that aims to use natural variations as calibration cues and each of these methods makes certain assumptions. For example, we require weather conditions in which we can isolate the intensity variations due to clouds from other sources of change. The methods we describe are a valuable addition to the emerging toolbox of automated outdoor-camera calibration techniques.

### 1.1 Related Work

#### 1.1.1 Stochastic Models of Cloud Shapes

The structure of clouds has been investigated both as an example of natural images that follow the power


- *N. Jacobs is with the Department of Computer Science, University of Kentucky, Lexington, KY, 40506. E-mail: jacobs@cs.uky.edu*
- *A. Abrams and R. Pless are with the Department of Computer Science, Washington University in St. Louis, MO, 63130.*

law and within the atmospheric sciences community. Natural images of clouds often have structure where the expected correlation between two pixels is a function of the inverse of their distance [1]. Furthermore, there is a scale invariance that may be characterized by a power law (with the ensemble spatial frequency amplitude spectra ranging from $f^{-0.9}$ to $f^{-2}$ [2]). These trends have been validated for cloud cover patterns, with empirical studies demonstrating that the 2D auto-correlation is typically isotropic [3], but that the relationship of spatial-correlation to distance varies for different types of clouds (for example, cumulus vs. cirrus clouds) [4]. This motivates our use of a non-parametric representation of the correlation-to-distance function.

#### 1.1.2 Shadows in Video Surveillance

For video surveillance applications, clouds are considered an unwanted source of image appearance variation. Background models explicitly designed to capture variation due to clouds include the classical adaptive mixture model [5] and subspace methods [6]. Farther removed from our application, object detection/recognition is disturbed by cast shadows because they can change the apparent shape and cause nearby objects to be merged. Several algorithms seek to minimize these effects, using a variety of approaches [7], including separating brightness and color changes [8].

#### 1.1.3 Geometry and Location Estimation Using Natural Variations

Within the field of remote sensing, shadows have long been used to estimate the height of ground structures from aerial or satellite imagery [9]. Recent work in analysis of time-lapse video from a fixed location have used changing lighting directions to cluster points with similar surface normals [10]. Other work has used known changes in the sun illumination direction to extract surface normal of scene patches [11], define constraints on radiometric camera calibration [12], [13], and estimate camera geo-location [13]. Work on the AMOS (Archive of Many Outdoor Scenes) dataset of time-lapse imagery demonstrates consistent diurnal variations across most outdoor cameras and simple methods for automated classification of images as *cloudy* or *sunny* [14]. This supports methods that estimate the geo-location of a camera, either by finding the maximally correlated location (through time) in a satellite view, or interpolating from a set of cameras with known positions [15]. The recently created database of "webcam clip-art" includes camera calibration parameters to enable applications such as illumination and appearance transfer across scenes [16].

### 1.2 Overview

We introduce two new cues for depth estimation for static cameras. In addition, we describe several algorithms that use these cues to estimate properties of the camera and the scene. Sec. 2 describes the cues: one based on geospatial proximity and one based on cloud motion. In Sec. 3 we describe several distinct algorithms that use these cues to estimate the focal length of the camera and a depth map for the scene.

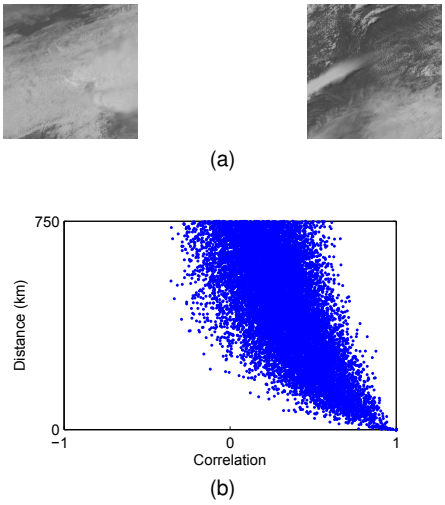



Fig. 2

### 1.3 Assumptions

The algorithms we present exploit properties of natural scenes that are captured by time-lapse videos. However, they depend on several assumptions about the camera, properties of the clouds in the scene, and for the second algorithm, knowledge about wind-speeds and geo-orientation of the camera. The camera is assumed to be completely static so that a pixel is observing the same scene location for the duration of the video. Also, we assume the sky pixels and mirrored surfaces (such as building windows and water) have been masked off and are not considered.

We also make assumptions about the structure of clouds in the scene. First, we assume that the correlation between two pixels is a monotonically decreasing function of the distance between the scene points they observe. We solve for the form of this function, but assume it is constant across the image. On scenes with geographic features that impact the distribution of clouds, such as coastlines or mountains, this assumption may fail.

For the second algorithm, we make use of temporal delay patterns arising from wind-blown clouds. Our approach to using this cue assumes that the wind speed and direction is constant over the course of the time-lapse, and that there are not multiple layers of clouds that have different directions of motion. The geo-orientation of the camera (the pan-angle) of the camera and the wind direction are assumed to be known. Finally, if the wind velocity is known, this can be used to solve to scale ambiguity. Otherwise, in all cases, the 3D model has an unknown scale.

## 2 Structural Cues Created by Cloud Shadows

The image of cloud shadows passing through a scene depends upon the camera and scene geometry. Here we describe two properties of outdoor-scene time lapses that depend on cloud shadows, are easy to measure, and, as we show in Sec. 3, can be used to infer camera and scene geometry.

### 2.1 Geographic Location Similarity

The closer two points are in the world, the more likely they are to be covered by the shadow of the same cloud. Thus, for a static outdoor camera, the time series of pixel intensities are usually more similar for scene points that are close than for those that are far. This is a compelling cue because it does not require a high framerate video stream.

We begin by considering the correlations that arise between pixels in satellite imagery. The statistical properties of this approximately orthographic view are similar to the spatial properties of the cloud shadows cast onto the ground. We empirically show the relationship between correlation and distance for a small dataset of visible-light satellite images (all captured at noon on different days during the summer of 2008). The scatter plot in Figure 2, in which each point represents a pair of pixels, shows that the correlation of the pixel intensities is clearly related to the distance between the pixels. Furthermore, the expected value of distance is a monotonically decreasing function of correlation.

This relationship also holds at a much finer scale. To show this, we compute correlation between pixels in a time-lapse video captured by a static outdoor camera on a partly cloud day. Since we do not know the actual 3D distances between points we cannot generate a scatter plot as in the satellite example. Instead, Figure 6 shows examples of correlation maps generated by selecting one landmark pixel and comparing it to all others. The false-color images, colored by the correlation between a pair of pixels, clearly show that correlation is related to distance.

We use correlation as a similarity measure because it is simple to compute online and works well in many scenes. In longer videos we compute correlation over many short temporal windows and then average these to compute the final score (see Figure 3). This similarity measure reduces the effect of long-range correlations due sun motion caused by objects with similar surface normals [10]. Our approach does not preclude the use of more sophisticated similarity metrics that explicitly reason about cloud shadows using, for example, color cues.

In Sec. 3.1, we show how to use these correlation maps, which reflect the relationship between correlation (or some other similarity measure) and distance, to infer the focal length of the camera and a distance map of the scene. In the following section we introduce an additional cue that is based on the motion of the cloud layer.

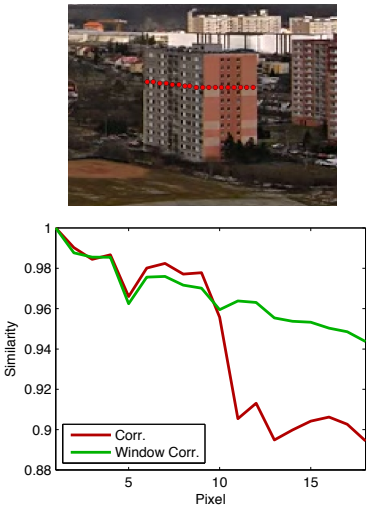



Fig. 3

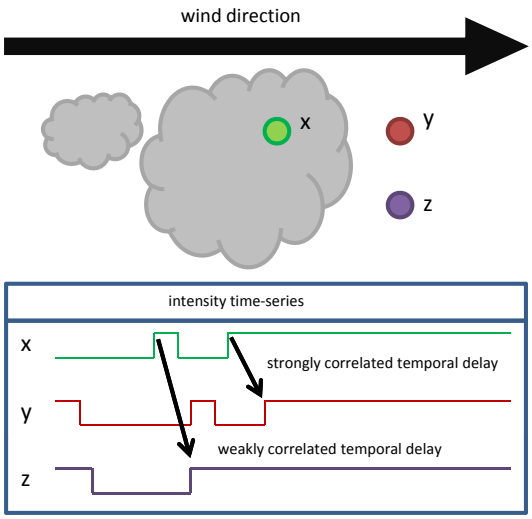



Fig. 4

### 2.2 Temporal Delay Due to Cloud Motion

As a cloud passes over a scene, each pixel varies due to the cloud's attenuation of sunlight. These variations result in a time series of pixel intensities that depend, in part, on the clouds. In the direction of the wind these time series are very similar but temporally offset in proportion to the geographic distance between the points (see Figure 4). Also, for short distances perpendicular to the wind direction we expect to see zero temporal delay. We expect correlation, after accounting for delay, to decrease with distance due to changing cloud shapes or, different clouds altogether if we move far enough perpendicular to the wind direction. In Sec. 3.2, we

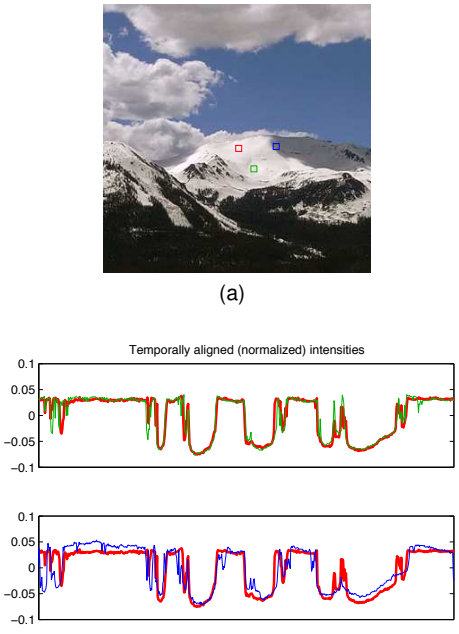



Fig. 5

formalize this as a set of linear constraints relating scene structure to the temporal delay giving the best correlation in the time series at pairs of pixels. These constraints can be sparse and are weighted by the correlation, so slow cloud changes over time are incorporated naturally into the optimization. The remainder of this section describes how we compute the temporal offset and shows examples of estimated offsets in real scenes.

Our method for estimating the temporal offset between the time series of a pair of pixels consists of a coarse alignment followed by a refinement stage. First we use cross-correlation to select the integer temporal offset, $\hat{\Delta}_t$, for which the signals are maximally correlated. Then we refine this estimate to the sub-frame level, $\Delta_t$, by finding the maxima of a quadratic model of the correlation values around the maximal integer offset. We use the correlation of the temporally aligned signals as a confidence measure, for example, low correlation means low confidence in the temporal offset estimate.

Figure 5 shows the result of this estimation procedure for three pixels from a mountain scene. After temporal alignment, the pixels directly in line with the wind are more similar than those that are not in line with the wind.

Figure 6 shows examples of false-color images constructed by combining the estimated delay and the temporally aligned correlation for every pixel, relative to a single landmark pixel. The motion of the clouds in this scene is nearly parallel with the optical axis, so the temporal delays are roughly equal horizontally across the image (perpendicular to the wind direction) but the correlations quickly decrease as distance from the pixel increases (different clouds are passing over those points). Orthogonally, the correlations are relatively higher in the direction of the wind but the delay changes rapidly.

## 3 Using Clouds to Infer Scene Structure

The dependence of correlation upon distance and the temporal delay induced by cloud shadow motion are both strong cues to the geometric structure of outdoor scenes. In this section, we describe several methods that use these cues to infer a depth map and simplified camera geometry.

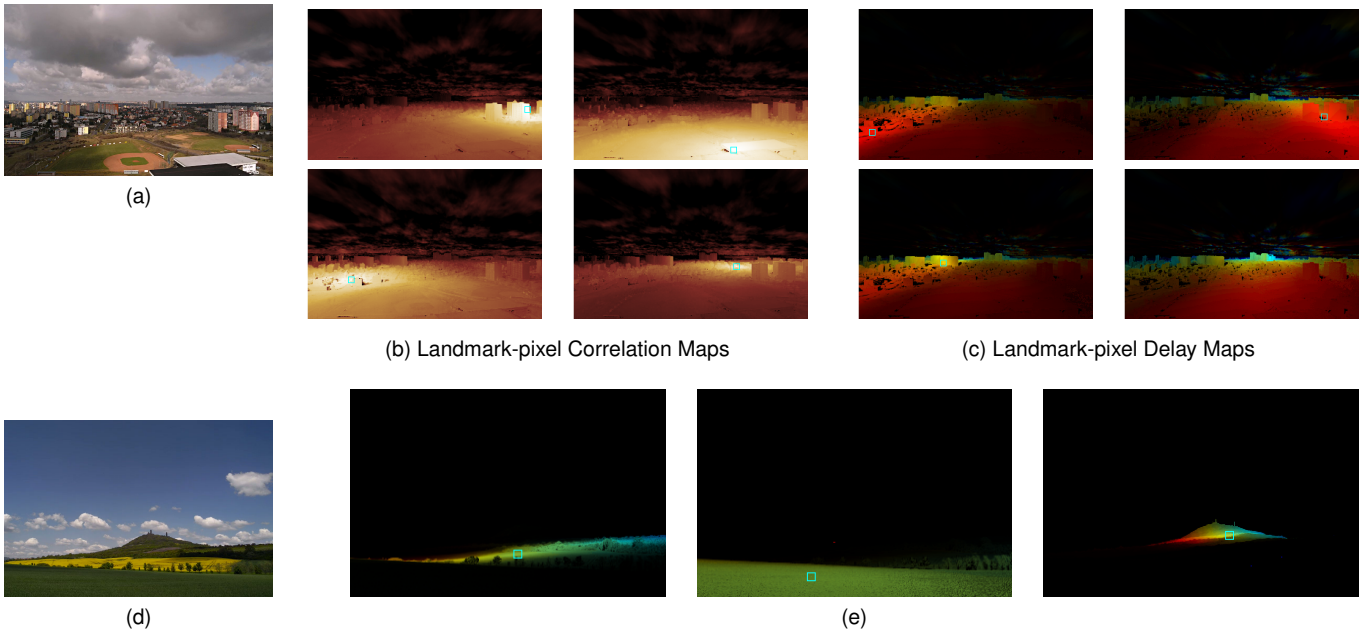
We assume a simplified pinhole camera model. Assuming a focal length, $f$, a point, $R_i = (X, Y, Z)$, in the world projects to an image location, expressed in normalized homogeneous coordinates as $r_i = (\frac{Xf}{Z}, \frac{Yf}{Z}, 1)$. For each pixel, $i$, the imaged 3D point, $R_i$, can be expressed as $R_i = \alpha_i r_i$ with depth, $\alpha_i$. Thus, the 3D distance between two points is $d_{ij} = ||R_i - R_j||$. But, for our purposes the use of 3D distances is not technically correct. Consider, for example, that any two scene points in-line with the sun vector see the same cloud shadows and will therefore have similar time series. In our experiments, we solve this problem by modifying $d_{ij}$ by projecting the points, along the sun direction vector, to the ground plane before computing the distance (if the sun vector is unknown we project points straight down). This gives distances that are related to time-series similarities induced by cloud shadows. A side effect of this projection is that it creates a point ambiguity where the depth of a pixel ray that is parallel to the sun vector is unconstrained.

### 3.1 Estimating Scene Structure Using Pairwise Correlation

In outdoor scenes under partly cloudy conditions there is a strong relationship between correlation, $\rho_{ij}$, and 3D distance, $d_{ij}$, between the imaged scene points. In this section, we show how to estimate the focal length of a camera and a depth map, $\mathbf{a} = \alpha_1, \ldots, \alpha_n$, for an outdoor scene using this relationship. Our approach has two main parts: we first create a planar approximation of the scene that gives the most consistent mapping from distance to correlation, we then use this to initialize a non-metric multidimensional scaling approach [18] that solves for the form of the correlation-to-distance mapping and the complete depth map that is most consistent with the measured correlations.

#### 3.1.1 *Estimating Focal Length and Horizon Line*

We solve for the camera focal length, $f$, and external orientation parameters, $\theta_x$ (tilt) and $\theta_z$ (roll) in a local east-north-up coordinate system, using a maximum likelihood method based on the relationship between



(a)

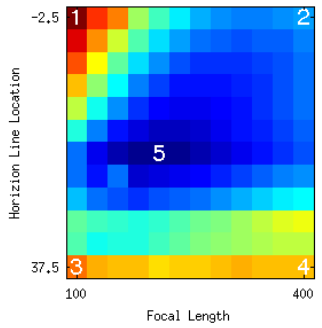
(b) Landmark-pixel Correlation Maps

(c) Landmark-pixel Delay Maps

(d)

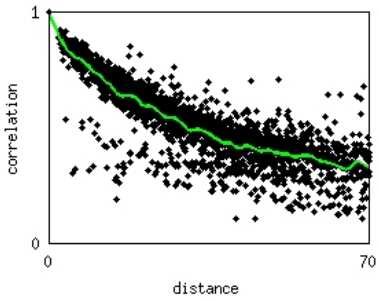
(e)

Fig. 6

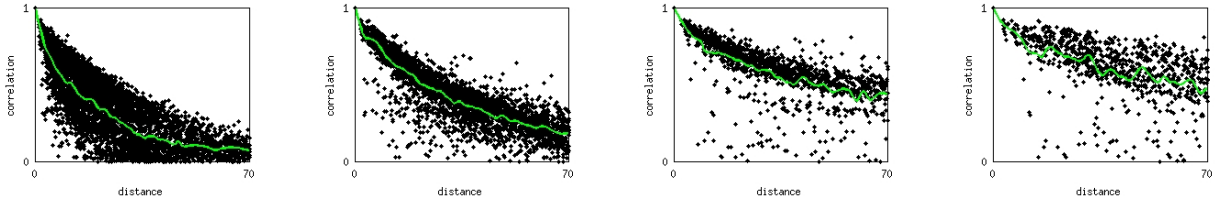



(a) planar initialization error function

(b) horizon lines for labeled points in (a), 1,2 (top), 5 (middle), and 3,4 (bottom)




(c) optimal $d_{ij} \rightarrow \rho_{ij}$ (scene 5)




(d) suboptimal $d_{ij} \rightarrow \rho_{ij}$ (scenes 1–4)

Fig. 7

correlation and distance[1]. See Figure 7 for an example of this procedure applied to a scene. In previous work [20], we sought parameters that made the mapping from correlation to distance most consistent. However, this gives solutions biased to have the scene depths be smaller and is very sensitive to points near the horizon. By optimizing over the consistency of the mapping from distance to correlation, we remove both of these problems with a small increase in complexity.

Under the assumption that the scene is a plane, these parameters define distances to all points in the scene, and therefore distances between all points in the scene. The idea of the approach is that there should be a consistent mapping between distances between points in the scene and correlations between time-sequences observed at those pixels; we search over the parameters, $(f, \theta_x, \theta_z)$, to make this mapping as consistent as possible. In particular, we search to find a mapping from distances to correlations that maximizes the likelihood of the observed correlations given the distances.

Assuming the camera is a fixed distance above a flat plane, the parameters, $(f, \theta_x, \theta_z)$, define a distance to every point in the scene, and therefore a distance, $d_{i,j}$, *between* every pair of pixels, $(i, j)$, in the scene. We define $\rho^*(d_{i,j}) = E[\rho|d]$ as the expected correlation between those pixels, given their pairwise distance (as determined by the flat plane assumption and camera parameters). We simultaneously estimate the function $\rho^*$ and the pairwise distances, $d_{i,j}$, using a probabilistic approach in which the conditional distribution of sample correlations, $P(\rho_{i,j}|\rho^*(d_{i,j}))$, is modeled as the distribution of the sample correlation, $\rho_{i,j}$, given the true correlation for bivariate normal distribution [21]. We express the entire optimization process as a maximum likelihood estimate as follows:

$$\max_{f,\theta_x,\theta_z,\rho^*} \prod P(\rho_{i,j}|\rho^*(d_{i,j})). \quad (1)$$

The probability $P(\rho_{i,j}|\rho^*(d_{i,j}))$ quantifies the likelihood of the sample correlation, $\rho_{i,j}$, given the (estimated) true correlation. To complete the optimization, we exhaustively search over a reasonable range of parameters, $(f, \theta_x, \theta_z)$, solving for $\rho^*$ at each setting, and choose the setting that maximizes the likelihood.

Our approach to estimating $\rho^*$, which maps distances to correlations as $\rho^*(d) = E[\rho|d]$, is based on samples of $(d_{i,j}, \rho_{i,j})$. Recall that the sample correlations, $\rho_{i,j}$, are measured from the imagery and the distances, $d_{i,j}$, are computed from the camera parameters, $(f, \theta_x, \theta_z)$. We define $\rho^*$ using a spline curve that ranges from the smallest to largest pairwise distance and which has a value between -1 and 1. The standard approach to fitting a spline to samples $(d_{i,j}, \rho_{i,j})$ using least squared error is optimal under the assumption that $\rho_{i,j}$ is a random variable with a Gaussian distribution. Our initial experiments using this method gave poor estimates of $\rho^*$ which we attribute to the mismatch between the distribution of the sample correlation, $\rho$, and the Gaussian distribution.

We solve for the spline curve that estimates $E[\rho|d]$ by using sampling to minimize (1). For each pair, $(d_{i,j}, \rho_{i,j})$, we construct thirty additional pairs, $\{(d_{i,j}, \rho_1), (d_{i,j}, \rho_2), \ldots\}$, by sampling from the distribution of the sample correlation for a bivariate normal distribution, using $\rho_{i,j}$ as the population correlation and sample size 60. We then use standard least squares to fit a spline curve through these samples. We found that this method gives significantly improved estimates of $\rho^*$ compared to directly fitting the spline to the original data. The expanded set of points more accurately reflects the increased uncertainty in estimating low correlations vs high correlations and allows us to use standard spline fitting code. Evaluating this spline curve defines our mapping from distances to correlations, $\rho^*(d)$.

We find that this optimization procedure is sufficient for the small number of parameters needed for a planar scene. Figure 10 shows the results of this method in computing the initial (planar) depth for two examples. Sec. 4.1 gives an experimental evaluation of this method. However, it does not work well for estimating a full depth map because of the high-dimensionality of the parameter space (a depth value for every pixel). This required an alternative method for optimizing for the depth map; in the following section we describe an efficient method that uses iterative local descent. This enables all depths to be simultaneously updated resulting in much faster convergence. We use the planar depth maps found using this method to initialize the method in the following section.

#### 3.1.2 Depth Map Estimation Method Overview

We use Non-metric Multidimensional Scaling (NMDS) [22], [18] to simultaneously solve for $E[d_{ij}|\rho_{ij}]$ and the depth map, $\mathbf{a}$. Like classical Multidimensional Scaling (MDS), NMDS solves for point locations given pairwise relationships between points. Unlike MDS, NMDS does not expect the input relationships to correspond to distances, instead the input is only required to have a monotonic relationship to distance. Since we assume that distance is a monotonically decreasing function of correlation, we can use the NMDS framework to solve for this mapping.

In our application NMDS works, from a high-level, as follows. First we initialize the solution with a planar depth map (we describe a method for estimating an initial depth map in Sec. 3.1.1). Given an initial depth map, we iterate through the following steps:

1) determine the 3D distance between scene points, $d_{ij}$, implied by the current depth map,
2) estimate the mapping from distance to correlation, $E[d_{ij}|\rho_{ij}]$ (see Sec. 3.1.3),
3) use the pairwise correlation, $\rho_{ij}$, and $E[d_{ij}|\rho_{ij}]$ to compute a pairwise distance estimate,

1. The pan angle is needed to geo-reference the scene, but is unnecessary to compute the distance between points in the scene. If needed, as in the depth from wind velocity cue, we assume that pan (degrees from north) is provided by the user, or some other estimation method, such as [19].

4) update the depth map to better fit the estimated distances (see Sec. 3.1.4).

We now describe the two major components of this procedure in greater detail.

#### 3.1.3 Estimating Pairwise Distance Given Correlation

This section describes our model of the monotonic mapping from correlation to distance, $E[d_{ij}|\rho_{ij}]$. Many simple parametric models could be used to fill this requirement but they impose restrictions on the mapping which can lead to substantial artifacts in the depth map. Instead we choose a non-parametric model that makes the following minimal assumptions on the form of the mapping:

- $\mathrm{E}[d_{ij}|\rho_{ij} = 1] = 0$, when the correlation is one the expected distance is zero,
- $\mathrm{E}[d_{ij}|\rho] \geq E[d_{ij}|\rho + \epsilon]$, expected distance is a monotonically decreasing function of correlation

These assumptions follow naturally from empirical studies on the spatial statistics of real clouds [3]. We present results on scenes that violate these assumptions in Sec. 4.

We use the non-parametric regression method known as monotonic regression [22] to solve for a piecewise linear mapping from correlation to distance while respecting the constraints described above. Control points, $\hat{\rho} = \{\hat{\rho}_1, \ldots, \hat{\rho}_k\}$, are uniformly sampled along the correlation axis, with the mapping from sample correlation values to the closest control point defined by $c(\rho) \in \{1, \ldots, k\}$ ($k = 100$ in all experiments). We use linear programming to estimate expected pairwise distances, $\hat{\mathbf{d}} = \{\hat{d}_1, \ldots, \hat{d}_k\}$, that satisfy the maximal correlation and monotonicity constraints defined above while minimizing $\sum \left| \hat{\mathbf{d}}_{\mathrm{c}(\rho_{ij})} - d_{ij} \right|$ relative to the distances, $d_{ij}$, implied by current scene model (initially a plane). Given the control point locations, $\hat{\rho}$, and optimal distance values, $\hat{\mathbf{d}}$, we use linear interpolation to estimate the expected value of distance for a given correlation.

Figure 10 shows examples of the correlation-to-distance mapping, $E[d_{ij}|\rho_{ij}]$. The expected values are reasonable when compared to the sample points and would be difficult to model with a single, well-justified parametric model. We use this regression model to define the expected distance between a pair of points with a given correlation, and we use this expected distance as input into the depth map improvement step described in the following section.

#### 3.1.4 Translating Pairwise Distances Into Depths

We use $E[d_{ij}|\rho_{ij}]$, defined in the previous section, to estimate a distance matrix that reflects 3D distances between imaged scene points. We then use Multidimensional Scaling (MDS) [18] to translate estimated distances into 3D point locations. Because imaged points must lie along a particular pixel rays, we augment MDS with the constraint that the 3D point locations must fit with the camera geometry. This algorithm is essentially a projectively constrained variant of the Non-metric Multidimensional Scaling (NMDS) [22] algorithm.

The error (*stress*) function for MDS is as follows:

$$S(\mathbf{a}) = \sum_{i,j} w_{ij}(d_{ij} - E[d|\rho_{ij}])^2 \tag{2}$$

where the weights, $w_{ij}$, are an increasing function of the correlation, $\rho_{ij}$. In other words, we expect the distance estimates from high-correlation pairs to be more accurate than those of lower-correlation pairs. In this work, we use $w_{ij} = \rho_{ij}^2$ for $0 \leq \rho_{ij}$ and $w_{ij} = 0$ for $\rho_{ij} < 0$. Recall that the 3D distance, $d_{ij}$, between imaged scene points is a function of the depths, $\mathbf{a}$, along pixel rays. Following the standard iterative minimization approach [18], we minimize the *stress* function using gradient descent [22] with respect to the depths using the trust region method, constrained so that $\mathbf{a} \geq 0$. We also constrain the average of the estimated pairwise distances to remain constant to avoid the trivial, zero-depth solution. We use a straightforward application of the chain rule to compute the gradient with respect to $\mathbf{a}$ and to form a diagonal approximation of the Hessian. We do several descent iterations for a given distance matrix before re-estimating the correlation-to-distance mapping, $E[d_{ij}|\rho_{ij}]$, using the updated point locations.

Ideally we would use all pairs of pixels when minimizing the *stress* function. Unfortunately, maintaining the full correlation and distance matrices in memory between all pairs of pixels is unreasonable for all but the smallest images (a $320 \times 240$ image would require storing several $76800 \times 76800$ matrices). To overcome this we select a set of landmark pixels (in all experiments we select 100 landmark pixels using k-means) and only consider the subset of rows of our correlation and distance matrices that correspond to the landmark pixels. While we are using landmark pixels we are still optimizing over the depths for all pixels. We find that using landmark pixels significantly reduces computation time and memory consumption with minimal impact on the resulting depth maps. In our Matlab implementation the complete depth estimation procedure, including the ground-plane based initialization, typically requires several minutes to complete.

### 3.2 Estimating Scene Structure Using Temporal Delay in Cloudiness Signal

The motion of clouds due to wind causes nearby pixels to have similar but temporally offset intensity time series. Together these temporal offsets, $\Delta_{t(i,j)}$, give constraints on scene geometry. Sec. 2.2 shows examples of these temporal offsets.

Let $W$ be a 3D wind vector which we assume it is fixed for the duration of the video. A pair of points in the world, $R_i, R_j$, that are in-line with the wind satisfy the linear constraint $R_i - R_j = W\Delta_{t(i,j)}$ where $\Delta_{t(i,j)}$ is the time is takes for the wind (and therefore the clouds) to travel from point $R_j$ to point $R_i$. However, the algorithm

in Sec. 2.2 can often compute the temporal offset between pixels not exactly in-line with the wind. We generalize the constraint to account for this by projecting the displacement of the 3D points onto the wind direction, $\hat{W} = W/||W||$:

$$\hat{W}^\top (R_i - R_j) = \hat{W}^\top W \Delta_{t(i,j)}. \quad (3)$$

Based on the simplified camera imaging model, each pixel corresponds to a known direction, so the 3D point position, $R_i$, can be written as a depth, $\alpha_i$, along the ray, $r_i$. This gives the following set of constraints on the unknown depths:

$$\hat{W}^\top (\alpha_i r_i - \alpha_j r_j) = \hat{W}^\top W \Delta_{t(i,j)}, \quad (4)$$

$$\alpha_i \hat{W}^\top r_i - \alpha_j \hat{W}^\top r_j = \hat{W}^\top W \Delta_{t(i,j)} \quad (5)$$

This set of constraint defines a linear system,

$$\mathbf{M}\mathbf{a} = \mathbf{\Delta}, \quad (6)$$

where $\mathbf{a}$ is a vector of the (unknown) depth values, $\alpha_i$, for each pixel, the rows of $\mathbf{M}$ contain two non-zero entries of the form $(\hat{W}^\top r_i, -\hat{W}^\top r_j)$, and $\mathbf{\Delta}$ contains the scaled temporal delays between pixels.

The constraint on depth due to temporal delay has an ambiguity. In all cases, the matrix $\mathbf{M}$ has a null space of dimension at least one. This is visible from the structure of $\mathbf{M}$, adding any multiple of $\alpha' = (\frac{1}{\hat{W}^\top r_1}, \frac{1}{\hat{W}^\top r_2}, \ldots)$ to the depth map, $\mathbf{a}$, does not change the left hand side of Equation 5. The next section describes how we overcome this ambiguity.

### 3.3 Combining Temporal Delay and Spatial Correlation

The two cues we describe have ambiguities, a scale ambiguity for the spatial cue and a null space ambiguity for the temporal cue, that prevent metric interpretation of the generated depth maps. Combining the two cues allows us to simultaneously remove both ambiguities and makes possible metric scene estimation. We propose the following method.

Starting with the constraints defined by the temporal cue, we solve for a feasible depth map, $\mathbf{a}$, using a standard non-negative least squares solver. We then consider the set of solutions of the form $\mathbf{a} + k\alpha'$, and search over values of $k$ to find a *good* depth map. While many criteria exist for evaluating a depth map we focus on combining the two cues we have described to remove this ambiguity. As with the spatial cue, we make the assumption that correlation is geographically isotropic. This motivates us to use the error function defined in Equation 1 to evaluate the different depth maps. The only difference is that we now search over the null space as opposed to the focal length and orientation parameters. In Sec. 4.3, we show results that demonstrate that depth maps with low error function values are more plausible than those with higher error function values.

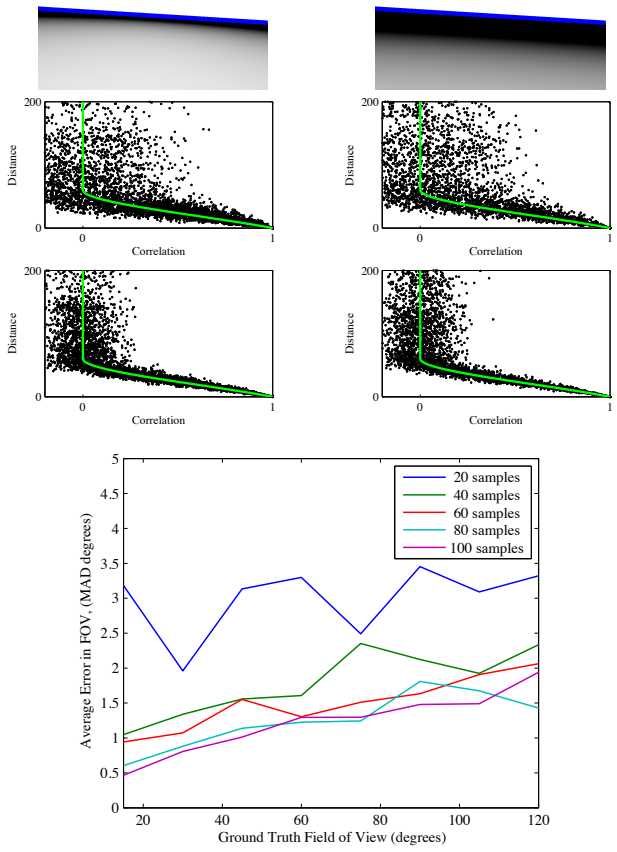



Fig. 8


Fig. 9

## 4 Evaluation

We describe our evaluation of various components of our approach and show results on end-to-end depth estimation on many outdoor scenes.

### 4.1 Focal Length from Correlation

This section presents an evaluation of our focal length estimation procedure on simulated and real data. In both cases we are evaluating the effect of noise: in the simulated case we vary the amount of noise in the correlation estimates and in the real-data case we are implicitly testing the accuracy of our method when the planar model is violated.

#### 4.1.1 Simulated Data

In this section, we evaluate the robustness of our initialization method to noise in the correlation readings. Our goal is to better understand the amount to which the accuracy of the results depends on the accuracy of the correlation estimate. First, we create a simulated scene, with a known focal length, image size, distance-to-correlation function and a randomly selected horizon line (see Figure 8). Given this, we know the actual distance between each pair of points. Then, we use the known distance to create simulated correlations between pairs of points, by sampling from a bivariate Gaussian with correlation determined by the distance-to-correlation function. This gives us a set of correlation values which we use as input to our initialization procedure. Figure 8 shows simulated correlations and quantitative evaluation of this method. The results demonstrate that increasing the accuracy of our correlation estimate, by increasing the number of samples, increases the accuracy of our field-of-view estimates. For the case of 100 samples, we obtain estimates of the field-of-view with with a mean-absolute deviation (MAD) of less than $2^\circ$.

#### 4.1.2 Real Data

Here we estimate the focal length of the two scenes shown in Figure 10. In Figure 9 we hold the camera orientation parameters fixed and vary the focal length to show how the error function, Equation (1), changes. The error function is minimized very near the true focal length. While these scenes are not planar, we find that the estimated focal length is close to the ground truth value in both cases.

### 4.2 Depth from Correlation

To evaluate the end-to-end performance of the depth from correlation algorithm (Sec. 3.1) across a broad range of representative videos, we collected a diverse set of videos [23]; most were found by searching on video sharing sites for videos that were (a) available to download, (b) were captured by a static camera on a partly cloudy day, and (c) did not include large fractions of the images with significant non-cloud sources of appearance change (e.g. traffic and pedestrians). In total, thirty-six videos were judged to fit these criteria, and we evaluated the performance of the depth-from-correlation algorithm on each. We found that for about half of the scenes our method estimated depth maps with minimal apparent artifacts. For most scenes we used correlation as a time-series similarity metric; for the two scenes in Figure 10 we used windowed correlation with a temporal window of approximately five minutes due to significant sun motion.

The remainder of this section discusses successes and failures to provide a better understanding of our method. In all real-data examples, we resize the original images to be 320 pixels wide and manually mask off the sky. We emphasize that in these examples we do no post-processing to improve the appearance of the generated depth maps. The optimization is based solely on geometric constraints on the camera geometry.

#### 4.2.1 Case Studies: Successes

Figure 10 shows the depth map and the correlation-to-distance mapping for two scenes. The first time lapse (top) was captured over three hours with pictures captured every five seconds. Naïvely computing correlation on the entire video sequence yields a low quality correlation map due to long term and spatially broad changes caused by the sun motion and melting snow on fields in the near ground. Computing correlations over short temporal windows and then averaging these correlations removed these artifacts. The second time lapse (bottom) consists of 600 images captured over 50 minutes. In addition to the sky, the river was manually masked and the shadow regions were automatically masked by removing low-variance pixels. Figure 1 contains another example of using the spatial cue to estimate a depth map and Figure 13 shows seven additional examples of successful depth maps with intermediate results, generated using our NMDS-based method.

#### 4.2.2 Case Studies: Impact of Violated Assumptions

We show six examples of failures that result from violation of the assumptions described in Sec. 1.3. We describe the specific examples below, but in general the results show that when correlations are not geographically isotropic our method generates low quality depth maps. However, even in these cases we often estimate reasonable depth maps for the parts of the scene where the assumptions hold. The main causes of errors we see are regions that have significant appearance variations that do not depend on cloud shadows (e.g. cast and attached shadows and occlusions from moving objects).

Figure 14 shows examples of depth maps that have significant errors. The first scene (top) fails in two distinct parts of the scene. In the bottom left, the depth estimates are noisy because that region almost always in shadow. In the top right, a distant hill is estimated to be close to the camera due to spurious high correlations due to the short duration of the video. The second scene has a poor quality depth map in the distance due to points that move in and out of shadow as a group (e.g. rows of trees). In the third scene, the correlation is dominated by surface normals and shadows leading to difficulty in estimating an initial correlation to distance mapping. In the fourth scene, the vertical parts of the bridge are well estimated but the remainder (i.e. in shadow under the bridge and on the water) fails. In the fifth scene, the distant hilltop is estimated to be close to the camera because there are numerous low-flying clouds that pass in front of the hill which increase the correlation. Since high-correlation pixels tend to be close together they are estimated to be close to the camera since they are widely spaced in the image. And finally, in the sixth scene

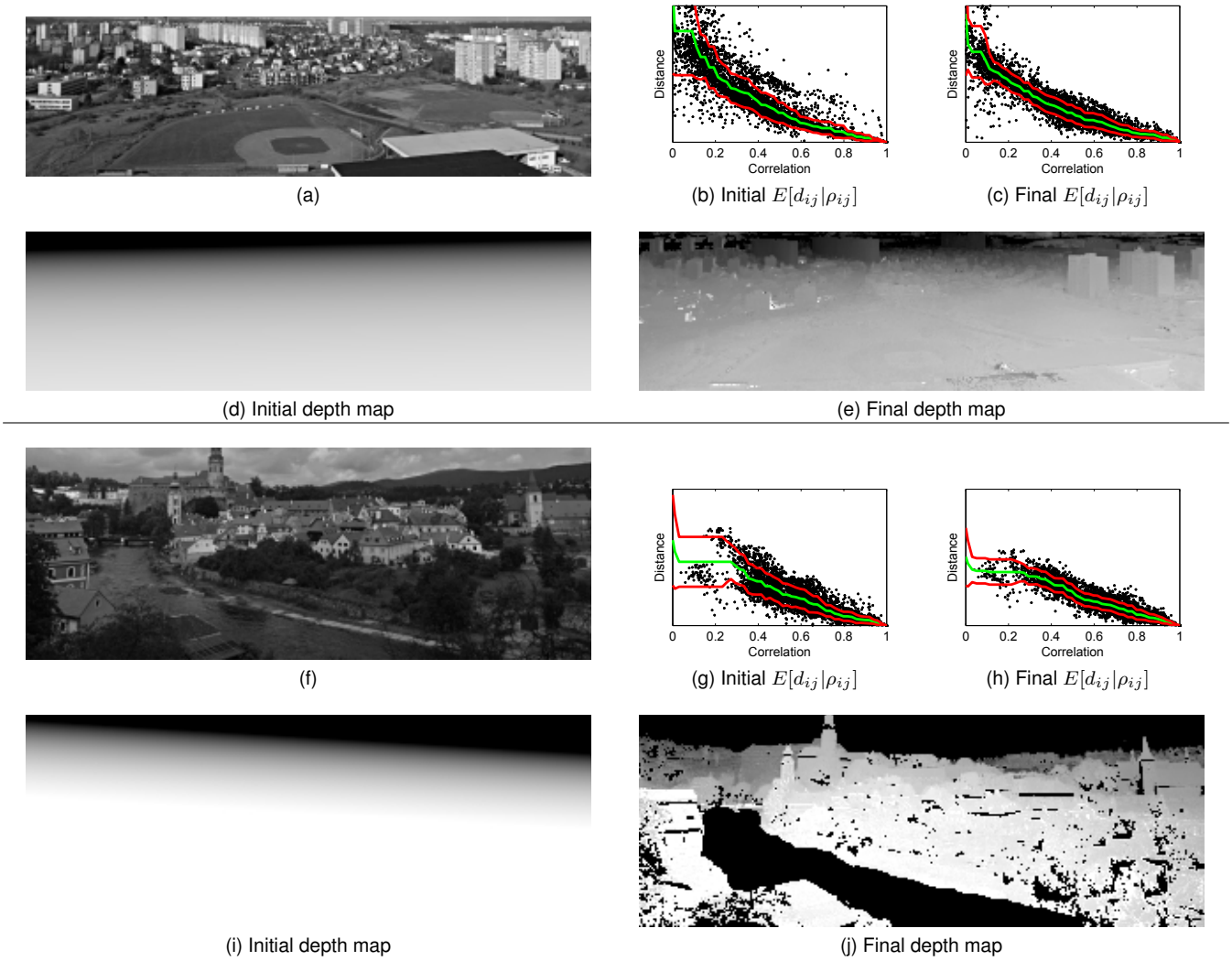



Fig. 10

(bottom), a few points in the distance with spurious high correlations, due to the short duration of the video, are pulled very close to the camera.

The quality of depth maps we generate depend on the ability to compute a geographically isotropic similarity function. We have shown several examples where correlation is unable to achieve this and gives poor quality depth maps. However, even in these cases the method often estimates reasonable depth maps for large parts of the scene.

#### 4.2.3 *Quantitative Evaluation*

We performed limited quantitative analysis on one scene for which we could align the scene with a satellite image. We clicked on 15 points scattered across different locations in the scene and found their corresponding points in the satellite image. These points varied from roughly 100 to 1000 meters from the camera. We find an average per-pixel error of 20 meters in the estimates of the 3D point locations (relative to the hand-clicked corresponding points) after a rigid alignment to geolocate the camera. This represents 2% error, relative to the scale of the scene, in the position estimates. Figure 11 shows how this error varies as we change the way we process

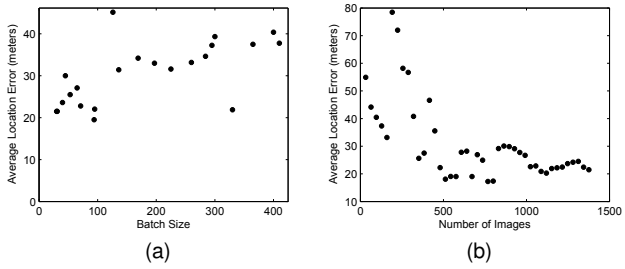



Fig. 11

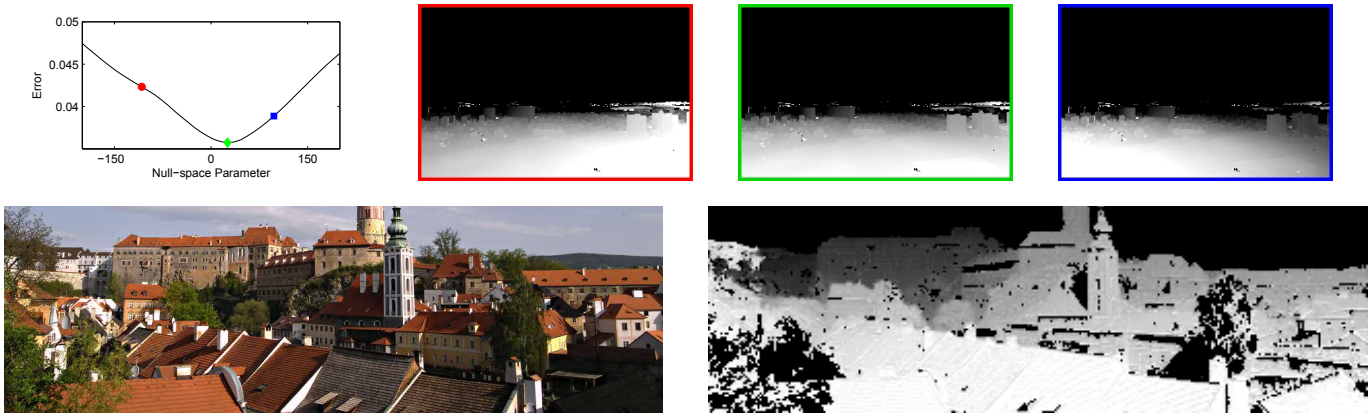



Fig. 12

the image data.

To assess the sensitivity to changes in parameter settings we performed two experiments. In the first, we fix the number of images but vary the temporal window size. We find that as window size increases, the error generally increases; this is due to the increased impact of appearance changes that violate our assumptions, such as the sun motion. In the second experiment, we fix the temporal window size (at the window size that gave minimum error in the previous experiment) and vary the number of images used to compute correlation. We find that, as expected, using more images, results in position estimates with lower error.

### 4.3 Depth from Combining Temporal Delay and Spatial Correlation

Figure 12 shows a depth map generated by the method described in Sec. 3.3. To reduce memory usage we discard constraints for pixel pairs, $ij$, whose temporally aligned correlation is less than a threshold (we use threshold 0.85). The top row of the figure show results on a previously described scene. This result demonstrates that higher values of the error function lead to lower quality depth maps. For the second scene, two-hundred frames of a time lapse, with one frame every five seconds, were used to estimate a delay map. We again use the combined inference procedure to estimate a depth map.

## 5 Conclusion

We presented two novel cues, both due to cloud shadows, that are useful for estimating scene and camera geometry. The first cue, based on spatial correlation, leads to a natural formulation as a Non-metric Multidimensional scaling problem. The second cue, based on temporal delay in cloud signals, defines a set of linear constraints on scene depth that may enable metric depth estimates. These cues are unique in that they can work when other methods of inferring scene structure and camera geometry have difficulties. They require no camera motion, no haze or fog, no sun motion, and no moving people or cars. We also demonstrated how to combine these cues to obtain improved results. This work adds to the growing literature on using natural scene variations to calibrate cameras and extract scene information.

## Acknowledgments

We gratefully acknowledge the support of NSF CAREER grant (IIS-0546383) and DARPA grant (D11AP00255) which partially supported this work.

## References

[1] G. J. Burton and I. R. Moorhead, "Color and spatial structure in natural scenes," *Applied Optics*, 1987.

[2] V. A. Billock, "Neural acclimation to 1/f spatial frequency spectra in natural images transduced by the human visual system," *Physica D: Nonlinear Phenomena*, 2000.

[3] C.-H. Sun and L. R. Thorne, "Inferring spatial cloud statistics from limited field-of-view, zenith observations," in *Atmospheric Radiation Measurements (ARM) Science Team Meeting*, 2000.

[4] L. R. Thorne, K. Buch, C.-H. Sun, and C. Diegert, "Data and image fusion for geometrical cloud characterization," Sandia National Laboratories, Tech. Rep. SAND97-9252, 1997.

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

[6] A. Mittal, A. Monnet, and N. Paragios, "Scene modeling and change detection in dynamic scenes: A subspace approach," *Computer Vision and Image Understanding*, 2009.

[7] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

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

[9] P. M. Dare, "Shadow analysis in high-resolution satellite imagery of urban areas," *Photogrammetric Eng. and Remote Sensing*, 2005.

[10] S. J. Koppal and S. G. Narasimhan, "Appearance derivatives for isonormal clustering of scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

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

[12] 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.
[13] 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.
[14] N. Jacobs, N. Roman, and R. Pless, "Consistent temporal variations in many outdoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
[15] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless, "Geolocating static cameras," in *IEEE International Conference on Computer Vision*, 2007.
[16] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Webcam clip art: Appearance and illuminant transfer from time-lapse sequences," *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 2009.
[17] S. Lovejoy and D. Schertzer, "Generalized scale invariance in the atmosphere," *Water Resources Research*, 1985.
[18] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd ed. Springer, 2005.
[19] 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.
[20] 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.
[21] J. Kenney and E. Keeping, *Mathematics of statistics*. Van Nostrand, 1954, vol. 3.
[22] J. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, 1964.
[23] http://cs.uky.edu/~jacobs/projects/shape-from-clouds/.

**Nathan Jacobs** graduated from the Univ. of Missouri in 1999 with a BS in Computer Science and completed his Ph.D. in Computer Science at Washington Univ. in St. Louis in 2010. He is currently an Assistant Professor of Computer Science at the Univ. of Kentucky. His research area is computer vision, with a focus on algorithms for widely distributed cameras, object tracking, environmental monitoring and surveillance.

**Austin Abrams** graduated from Truman State University in 2009 with a BS in Computer Science. He is currently a Ph.D. student in Computer Science at Washington University in St. Louis, with a research focus in computer vision using long-term time-lapse imagery.

**Robert Pless** graduated from Cornell University in 1994 with a BS in Computer Science and completed his Ph.D. in Computer Science at the University of Maryland in 2000. He has since been on the Computer Science faculty at Washington University in St. Louis where he founded the Media and Machines Laboratory and serves as Assistant Director of the Center for Security Technologies. His research interests focus on understanding motion in video, with applications to surveillance and medical imaging.

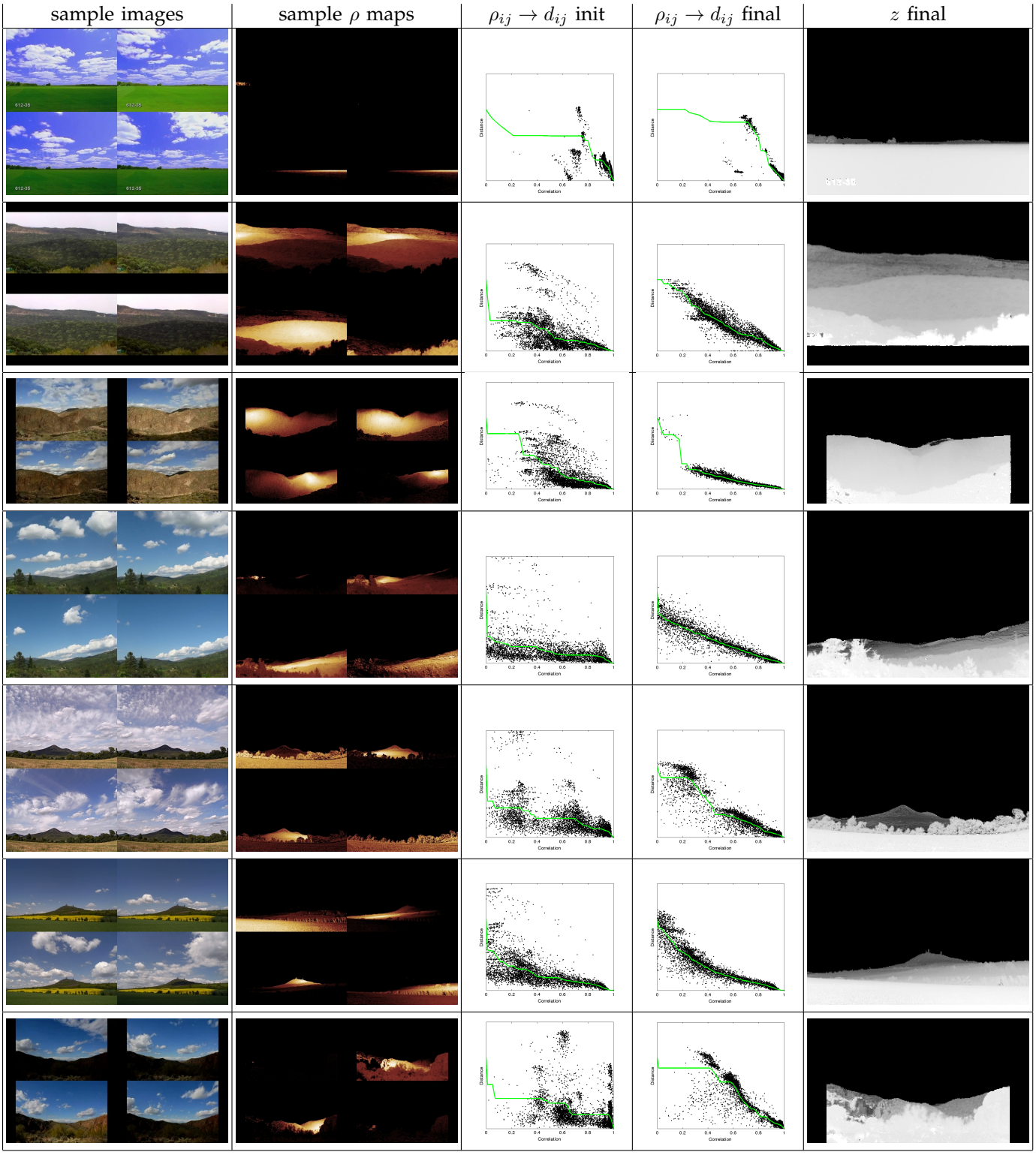

sample images
sample $\rho$ maps
$\rho_{ij} \rightarrow d_{ij}$ init
$\rho_{ij} \rightarrow d_{ij}$ final
$z$ final
Distance
Correlation

Fig. 13

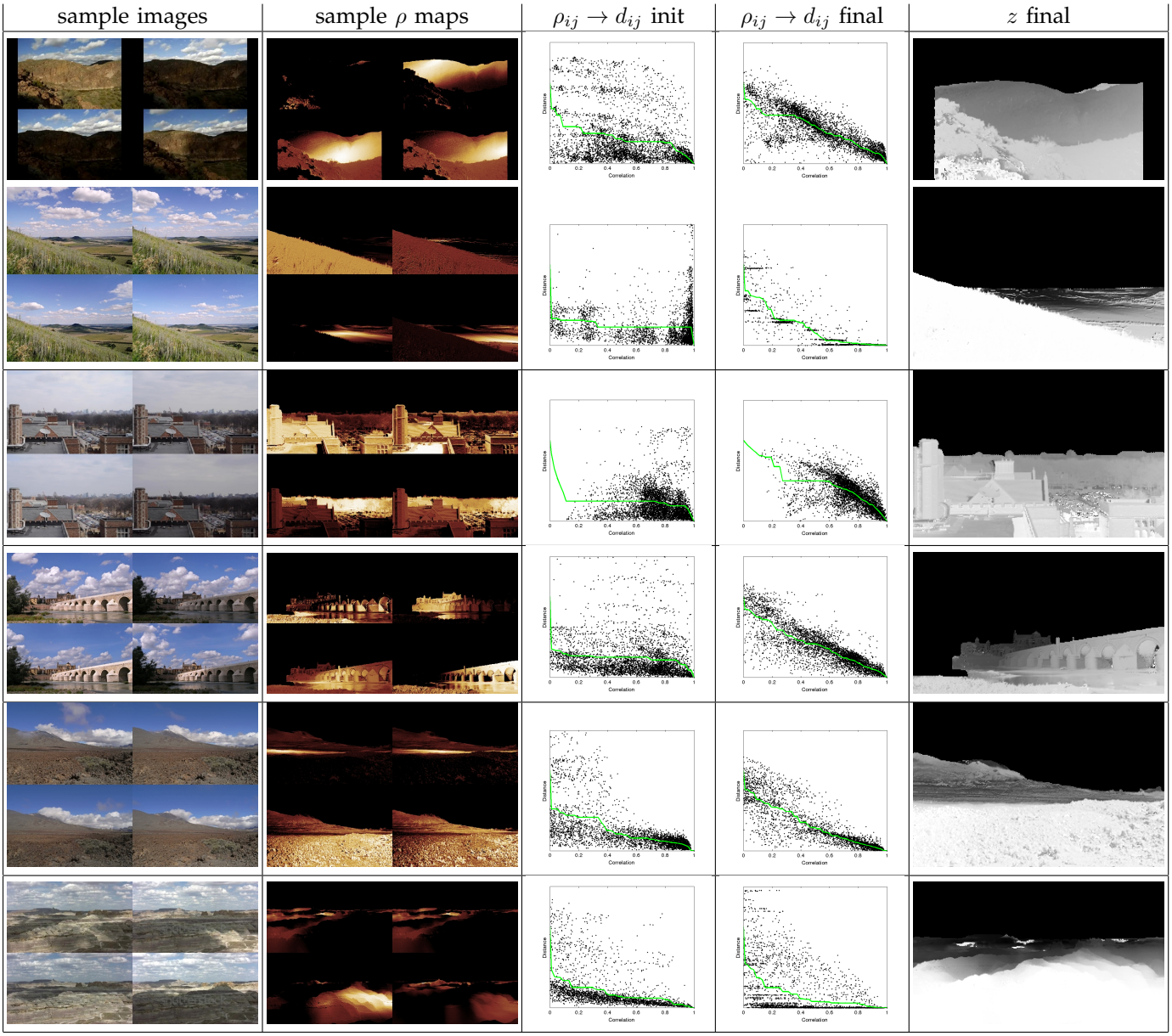

sample images
sample $\rho$ maps
$\rho_{ij} \rightarrow d_{ij}$ init
$\rho_{ij} \rightarrow d_{ij}$ final
$z$ final
Distance
Correlation

Fig. 14