Scene Understanding using Clouds

Scott Workman, University of Kentucky
Advisor: Nathan Jacobs

Abstract

Clouds are among the dominant features of outdoor scenes, yet most vision algorithms treat their effects on the scene as noise. However, the motion of clouds and appearance changes from the shadows they cast provide strong constraints on both camera and scene geometry. We introduce methods that use observations of an outdoor scene over days and weeks as input for estimating camera calibration and scene geometry. Cloud-based cues are an important alternative to methods that require specific forms of static scene geometry or clear sky conditions.

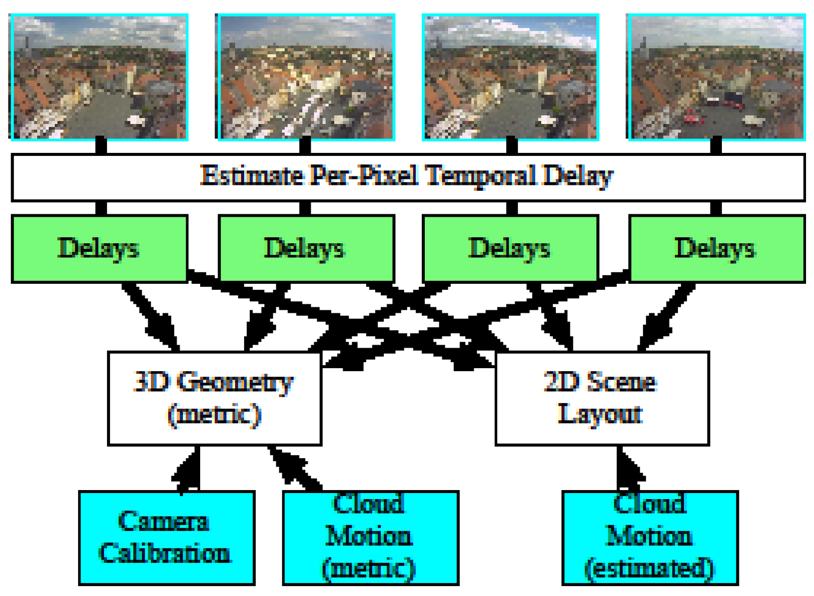
Overview:

- Clouds cause dramatic scene changes.
- They are considered a nuisance by most vision algorithms.
- We use clouds for estimating camera calibration and scene geometry.

Image by Martin Setvak

Scene Geometry from Several Partly Cloudy Days

Summary: The shadows cast by moving clouds provide direct constraints on scene depth. We use simple geometric constraints based on appearances changes caused by moving clouds shadows and show how combining constraints from multiple days removes the ambiguity inherent to only a single day.



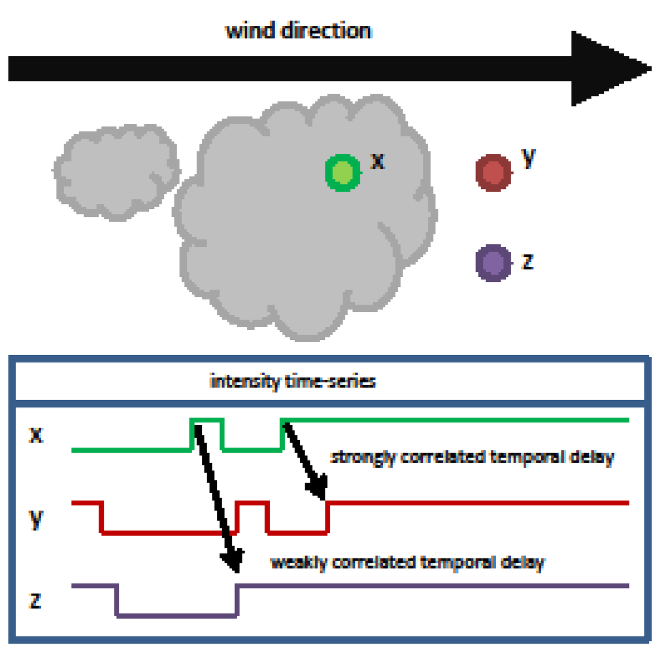
Estimating temporal delay:

The delay constraint describes the relationship between two 3D points inline with the cloud motion vector.

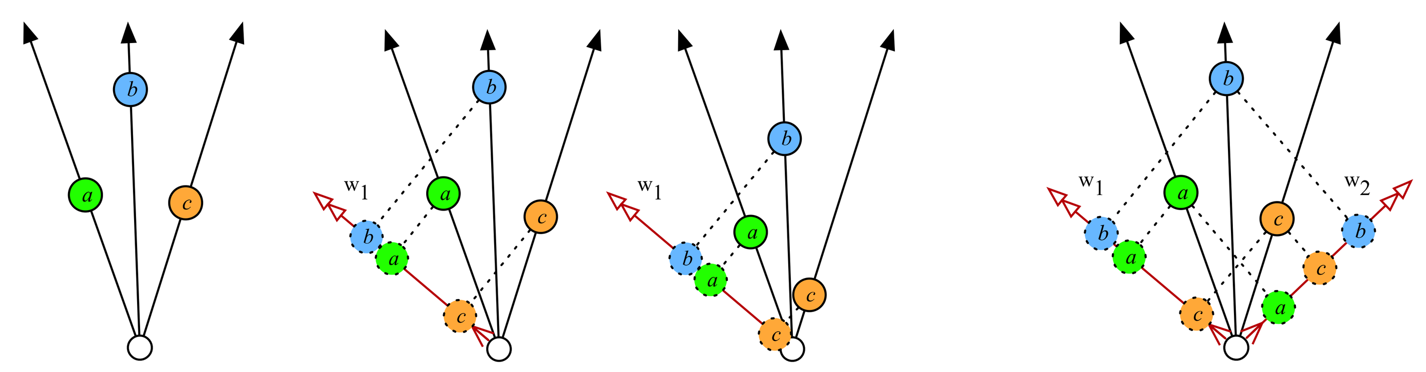
$$y - x = w\lambda$$

For two points not directly in line with the wind

$$w^T(z - x) = w^T w\lambda$$



Ambiguity: delay constraints from a single day have an inherent ambiguity



True scene geometry

With one day of data, many possible solutions that satisfy a set of measured delays

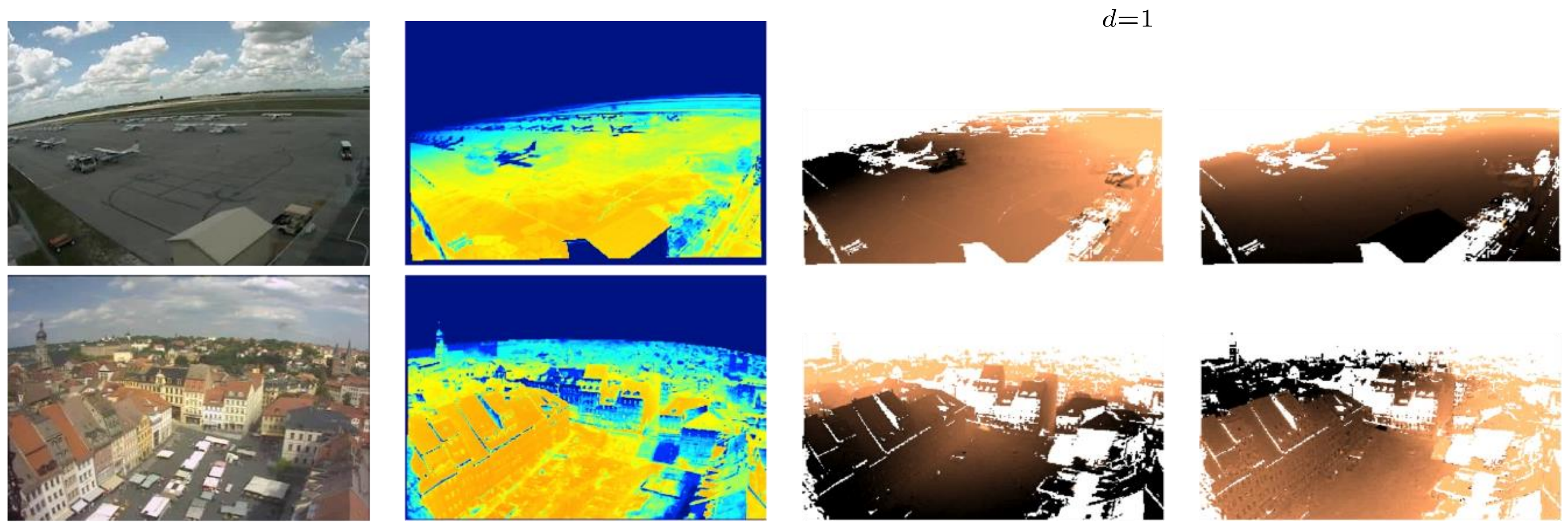
With an additional set of delays we get a unique solution.

Multiday Objective Function:

For a single pair of points: $w_d^T(x_p - x_r) = \lambda_{drp} w_d^T w_d$

For all pairs of points in one video (one day): $w_d^T X H = w_d^T w_d \Lambda_d$

Full objective function over multiple days: $f(X, \{w_d\}) = \sum_{d=1}^{D} \|w_d^T X H - w_d^T w_d \Lambda_d\|_2^2$



Cloud Motion as a Calibration Cue

Summary: We estimate the focal length, horizon line, and geo-orientation of a static outdoor camera by estimating cloud motion and relating it to local weather observations.

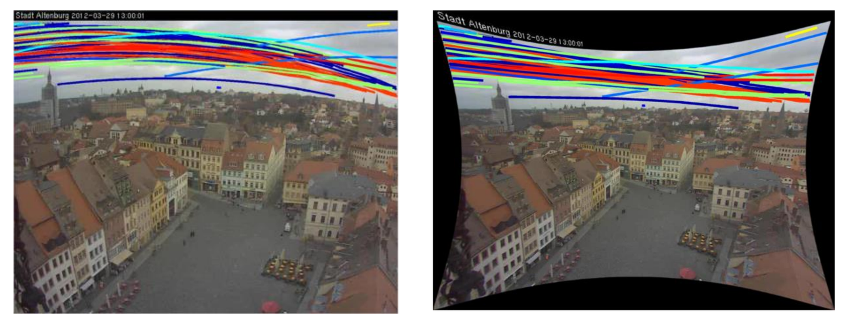
Approach:

1. For many videos, estimate cloud motion using differential optical flow.

$$I_x x_p + I_y y_p + I_t = 0 \quad [x_p, y_p]^T = \mathbf{A}_p^{-1}\mathbf{b}_p$$

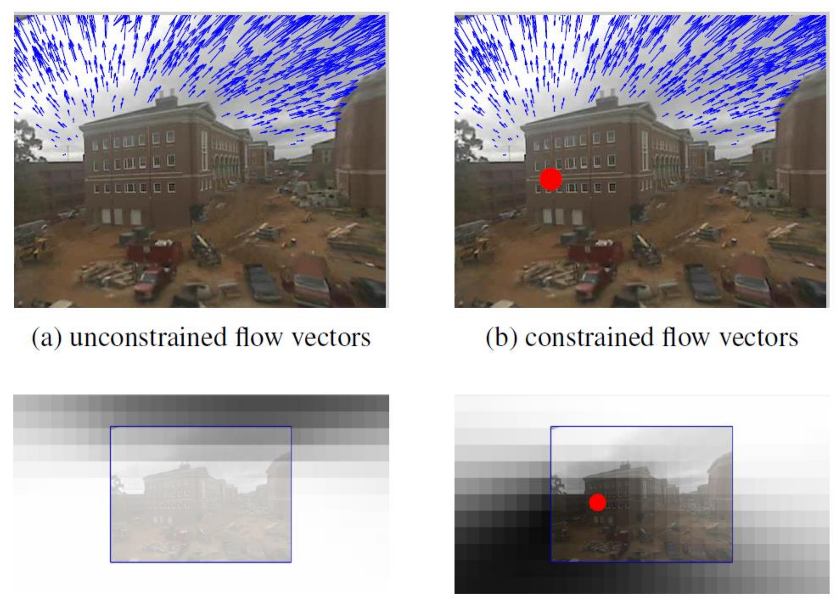
$$\mathbf{A}_p = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad b_p = \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

2. (if necessary) Remove radial distortion.



3. Estimate vanishing point of cloud motion in each video.
   - Constrain motion at each pixel to be toward the vanishing point.

$$f(\mathbf{v}, \{\alpha_p\}) = \sum_p \|\alpha_p \mathbf{A}_p(\mathbf{v} - \mathbf{p}) + \mathbf{b}_p\|^2 = \sum_p \alpha_p^2 (\mathbf{v} - \mathbf{p})^T \mathbf{A}_p^2 (\mathbf{v} - \mathbf{p}) + 2\alpha_p (\mathbf{v} - \mathbf{p})^T \mathbf{A}_p{}^T \mathbf{b}_p + \mathbf{b}_p{}^T \mathbf{b}_p$$



(a) unconstrained flow vectors (b) constrained flow vectors

(c) error surface (sink) (d) error surface (source)

4. Given two days (but more helps) with independent cloud motions: estimate focal length, horizon line and geo-orientation.

Estimate horizon line:

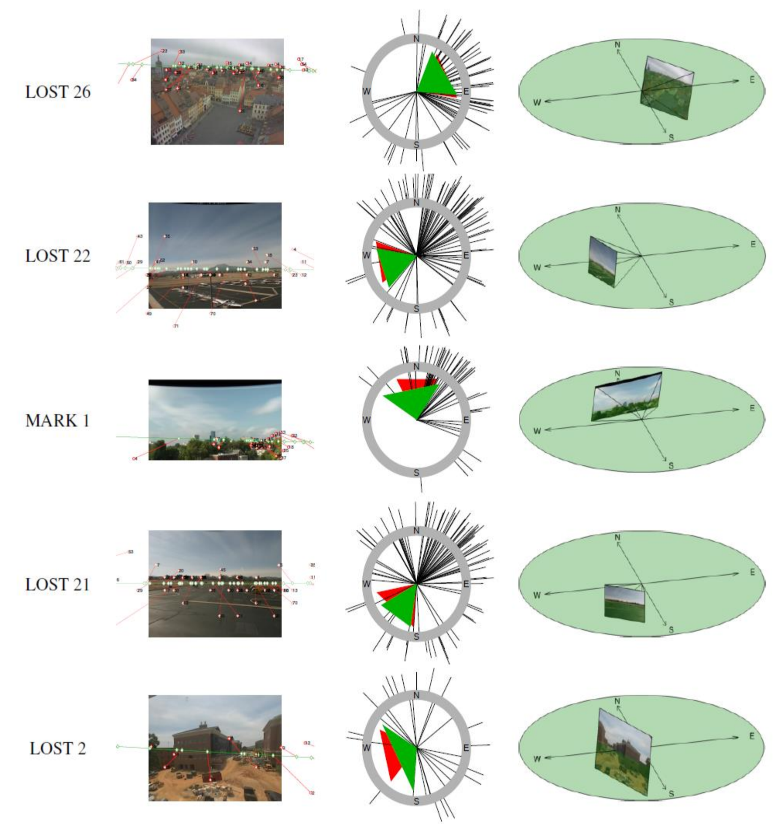
Idea: Cloud motion is horizontal and driven by the wind, so vanishing points should lie on the horizon line.

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

Estimate focal length and orientation:

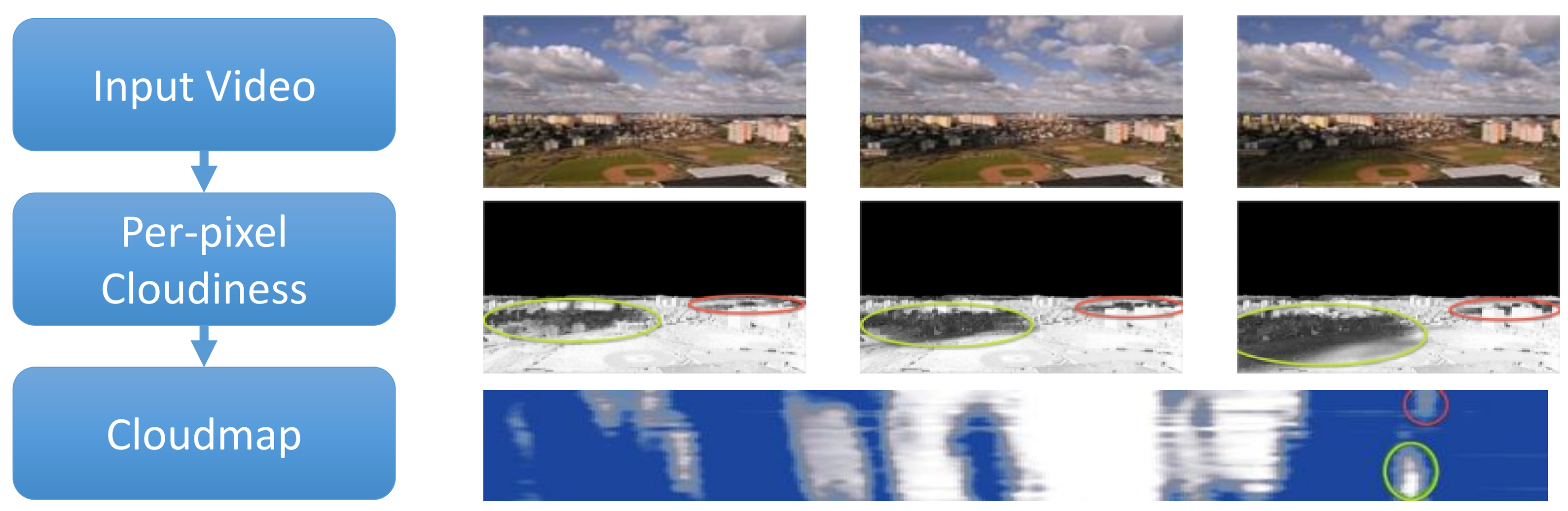
Idea: vanishing points of cloud motion should agree with local weather reports of wind directions.

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



Estimating Cloudmaps from Outdoor Video

Problem statement: Given a video of an outdoor scene, estimate the layer of clouds that passed over the scene.



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 sunlight.

Estimating per-pixel cloudiness: how much sunlight is illuminating each imaged scene point in each video frame.

sun color — cloudmap — shadow color

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

where ρ^1 and ρ^0 are max/min brightness respectively.