

CLOUD TRACKING FOR SOLAR IRRADIANCE PREDICTION

Ming-Ching Chang*, Yi Yao[†], Guan Li[‡], Yan Tong[§], Peter Tu

GE Global Research Center, Niskayuna, NY USA

ABSTRACT

We propose a video analytic system to segment and track clouds for the purpose of solar irradiance prediction. Ground-based imaging sensors are used to monitor potential sun occlusions for solar irradiance drop prediction, which can be used to assist power ramp control strategies. Sky images are first rectified to remove fisheye artifacts. Cloud pixels are segmented and classified into low-to-high transparencies. Evolution of cloud boundaries are tracked using optical flow. Tracking extrapolation predicts future cloud movements and deformations as well as potential sun occlusions. To accurately estimate solar irradiance drop, we propose a novel scheme based on back-projecting the predicted sun occlusion onto the evolving cloud boundary to count for cloud transparency and irradiance drop estimation. Experimental results show that short-term solar irradiance drop is predictable with reasonable accuracy.

Index Terms— solar irradiance prediction, cloud tracking, optical flow, back projection.

1. INTRODUCTION

A major issue of solar power is its fluctuation due to unstable sunlight irradiance. Cloud occlusions are known to be a key contributor to this effect [1, 2]. The effects of clouds on solar irradiance drop depend on the cloud’s volume, height, and thickness [3]. In this paper, we present a short-term solar forecasting system based on visual tracking of clouds from a ground-based sky imager. By segmentation and tracking cloud pixels, we show that sun occlusion can be estimated for irradiance drop prediction. Our approach can effectively improve solar ramp control [4] and operational cost.

Our video analytic pipeline consists of the following steps (summarized in Fig.1): (i) One or more wide-angle cameras are used to capture large regions of the sky. (ii) The sky image is rectified to remove fisheye distortions, so that cloud movements and deformations can be tracked and predicted using linear projective models. (iii) Cloud pixels are segmented and their transparencies are estimated. (iv) Horn/Schunck optical flow is applied to calculate the motion field of the cloud

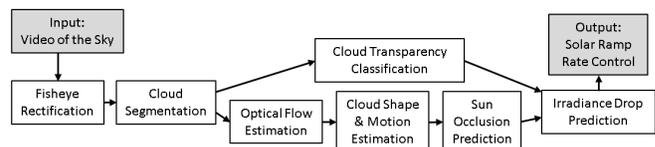


Fig. 1. Proposed pipeline of visual cloud tracking for solar prediction.

boundary. (v) Cloud deformations and movements are predicted via extrapolation of the motion field. (vi) Occlusion of the sun is predicted along with irradiance drop estimation based on estimates of cloud transparency.

The main contribution of this paper is a video analytic system focused on cloud tracking and sun-occlusion analysis for solar irradiance prediction. The system integrates a wide range of practical computer vision techniques including camera calibration, image rectification, segmentation, classification, optical flow, tracking and prediction. We propose a novel back-projection method for a *quantitative* prediction of solar occlusion by considering both the cloud motion field and cloud transparencies.

Related Works: There exists extensive research on sky imagers and cloud tracking for solar forecasting [5]. Analysis typically focuses on cloud type classification, thickness, transparency, velocity, height, and relative positions *w.r.t.* the sun. Most existing methods are based on simple image processing methods to determine cloud regions. Relatively little effort was been devoted to developing an end-to-end solution for solar occlusion prediction. The UCSD Total Sky Imager [6, 4, 7] segmented and determined cloud type by classifying sky pixels (clear sky, optically thin and thick clouds) using the RGB red-blue ratio. Threshold parameters were chosen using manually annotated images. Cloud motion vectors were calculated using cross-correlation between consecutive images, where only rough global motion is estimated. The work of [8] employs a similar pipeline of cloud pixel segmentation and Lucas-Kanade optical flow [9] for sun occlusion estimation up to 40 seconds. In contrast, we perform a probabilistic classification of cloud types (in §2), and we estimate cloud motion using globally refined Horn/Schunck optical flow [10] (in §3) to generate a longer prediction (as shown in §5).

2. SKY IMAGER SETUP, CLOUD SEGMENTATION AND TRANSPARENCY ANALYSIS

We first describe our sky imager hardware configuration. To avoid camera over-exposure due to direct observation of the sun, an effective way for **sun position estimation** or **block-**

*Ming-Ching Chang is currently with the Electrical and Computer Engineering Department, University at Albany, SUNY, NY, USA.

[†]Yi Yao is currently with Sarnoff Corporation, NJ, USA.

[‡]Guan Li is currently with Zillow Group, Seattle, USA

[§]Yan Tong is currently with Univ. of South Carolina.



Fig. 2. (a) Input wide-angle fisheye and the view after rectification. (b) Calibration using a standard checkerboard pattern using OCamCalib [11], where main steps include corner extraction, re-projection error analysis, and the recovery of camera parameters and checkerboard positions in 3D.

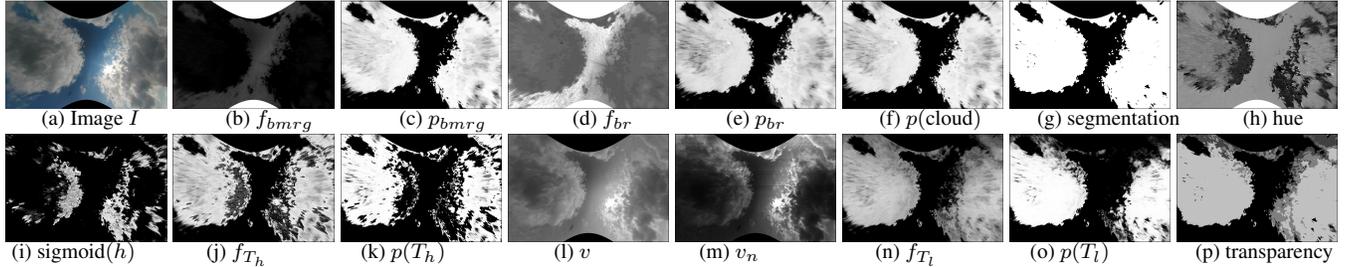


Fig. 3. Cloud segmentation and transparency classification. (a) Input sky image. (b) $f_{bmr_g} = b - \min(r, g)$. (c) $p_{bmr_g} = 1 - \text{sigmoid}(f_{bmr_g})$. (d) $f_{br} = \frac{b}{r}$ with intensity scaling by multiplication of 100 and visualized in $[0, 255]$. (e) $p_{br} = 1 - \text{sigmoid}(f_{br})$. (f) Cloud mask $p(\text{cloud})$. (g) Cloud segmentation mask after thresholding and morphological filtering. (h) Hue map h . (i) $\text{sigmoid}(h)$. (j) High-transparency feature $f_{T_h} = p(\text{cloud}) - [1 - \text{sigmoid}(h)]$. (k) High-transparency map $p(T_h) = \text{sigmoid}(f_{T_h})$. (l) Intensity map v . (m) Normalized intensity map v_n . (n) Low-transparency feature $f_{T_l} = p(\text{cloud}) - [1 - \text{sigmoid}(v_n)]$. (o) Low-transparency map $p(T_l) = \text{sigmoid}(f_{T_l})$. (p) Final cloud transparency label map.

ing is required. The UCSD system [4, 7] uses a mechanical moving stripe to physically block the sun. In contrast, we exploit a software solution by calculating the sun location in the image view based on the Solar Position Algorithm (SPA) [12] and camera calibration. Specifically, given the current UTC time and the site location in terms of global *longitude* and *latitude*, we follow Jean Meeus’s algorithm [13] detailed in [12] to precisely calculate the solar *zenith* and *azimuth* angles, and convert the sun location to pixel coordinates in our sky image using calibrated camera (intrinsic and extrinsic) parameters. Fig.4c depicts an example result, where the two green ellipses indicate the sun mask and an enlarged over-exposed region. We denote S^τ the sun mask at time τ for later use in occlusion estimation.

We use a wide angle lens to ensure full sun coverage throughout the day. Thus **fish-eye rectification** is performed, such that linear velocity models can be assumed for cloud motion prediction. We use the OCamCalib [11] to calculate the rectification from a set of checker board images, where both the *radial* and *de-centering* distortions are estimated [14, 15] via nonlinear optimization. Camera intrinsic and extrinsic parameters are derived by minimizing the re-projection errors of the image grids. Fig.2 summarizes the main steps.

We describe an empirical rule-based, probabilistic approach to per-frame **cloud pixel analysis** for joint segmentation and classification. Sensor-dependent parameters are empirically determined at the optics setup stage. We first determine whether a pixel belongs to the cloud or the background (sky or other). For a cloud pixel I_x , we define a pixel-wise *classification function* $T(I_x)$ that classifies I_x into four types: *high-* (T_h), *mid-* (T_m), and *low-* (T_l) of *transparencies* and the *background* (T_b). Given an input image I , the RGB and HSV for a pixel I_x are denoted as $I_x = (r, g, b)$

and $I_x = [h, s, v]$ respectively, where $0 \leq r, g, b, v \leq 255$, $0 \leq h \leq 360$, and $0 \leq s \leq 1$.

In the literature [1, 16, 2, 3], the *red/blue ratio* ($\frac{r}{b}$) [4, 1, 16, 2] and *red-blue difference* ($r - b$) [3] are often used for **cloud segmentation**, since clear sky usually scatters more blue than red light, and the clouds scatter both blue and red light almost equally. However, we found this simple approach did not perform well when faced with typical environmental variations. Instead, we adopt the probabilistic *logistic sigmoid* for robust classification. Specifically, we define two features based on (i) the difference between the blue and the minimum of red and green, $f_{bmr_g} = b - \min(r, g)$, and (ii) flip the red/blue ratio, $f_{br} = \frac{b}{r}$. Both features are mapped to a normalized range of $[0, 1]$ using a sigmoid function, $p_{bmr_g} = \frac{1}{1 + \exp(-\frac{f_{bmr_g} - \mu_{bmr_g}}{\sigma_{bmr_g}})}$, where μ_{bmr_g} and σ_{bmr_g} represent the mean and standard deviation of cloud pixels, respectively. $p_{br} = \frac{1}{1 + \exp(-\frac{f_{br} - \mu_{br}}{\sigma_{br}})}$ is calculated similarly. Our soft cloud segmentation is derived as: $p(\text{cloud}) = \sqrt{p_{bmr_g} \cdot p_{br}}$. Thresholding followed by morphological filtering results in the final segmentation map, see Fig.3a-g.

Determination of high-transparency clouds: We observed that both the RGB and hue h of a transparent cloud are close to that of clear sky, while the cloud can still be robustly segmented, as in Fig.3a,h. We thus intersect the cloud mask and the hue map to identify the transparent clouds. Specifically, the difference between p_{bmr_g} and the sigmoid of hue $p_h = \frac{1}{1 + \exp(-\frac{h - \mu_h}{\sigma_h})}$ represents a useful feature. Finally, a high-transparency cloud feature is defined as $f(T_h) = p_{bmr_g} - p_h$.

Determination of low-transparency clouds: We observed that low-transparency clouds are typically dark and thick, and hence have lower pixel brightness as in Fig.3a,l.

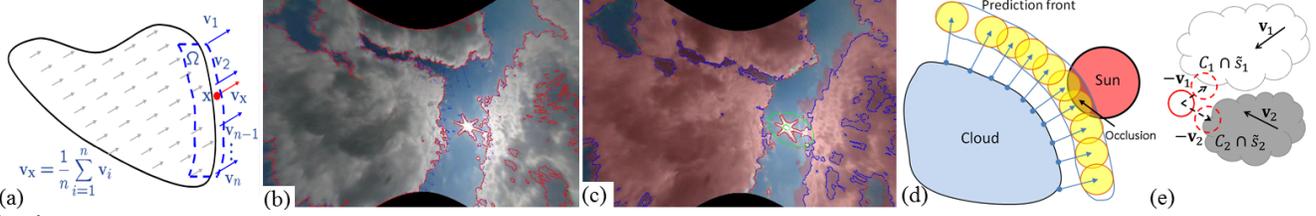


Fig. 4. (a) Velocity estimation by averaging out the boundary optical flow field. (b) An example velocity estimation map \mathcal{B}^t at time t . Red curves outline the detected cloud region. Blue arrows indicate estimated velocity field on the cloud boundaries. The length of the arrows is proportional to the magnitude of the estimated velocities. (c) Cloud motion prediction mask \hat{C}^τ at a future time $\tau = t + \Delta t$. (d,e) sun occlusion calculation using back-projection (see text).

We use the difference between cloud map and the brightness v as a feature for this type of cloud. We first normalize the v map by $v_n = \frac{v - v_{\min}}{v_{\max} - v_{\min}}$ as in Fig.3m. We then calculate $p_{v_n} = \frac{1}{1 + \exp(-\frac{v_n - \mu_{v_n}}{\sigma_{v_n}})}$. A low-transparency feature is defined as $f(T_l) = p_{bmr} - p_{v_n}$.

Finally, the probability of high- $p(T_h)$ and low- $p(T_l)$ transparencies can be calculated by normalizing $f(T_h)$ and $f(T_l)$ to the range of $[0, 1]$ using additional sigmoids. After further morphological filtering, the remaining cloud pixels belong to the mid-transparency T_m . Fig.3 illustrates our complete rule-based cloud segmentation pipeline. For each input image I , connected component labeling is performed to determine the set of cloud regions $\{C_i\}_{i=1}^b \in I$, where b is the total number of regions.

3. CLOUD MOTION ESTIMATION

After each cloud region C_i is defined, such regions can be visually tracked based on appearance and shape matching. Since cloud evolution is typically non-rigid and can split and merge, we dynamically track the boundary B_i of each cloud C_i using Horn/Schunck *optical flow* [10]. The displacement is calculated for fine pixel grids between consecutive frames, thus the raw estimated velocity is mostly noisy. To this end, we **aggregate velocity estimation** from a larger boundary region for robust estimation as in Fig.4a. Specifically, for each cloud boundary point of interest $I_x \in B_i$ (red dot in Fig.4a), we determine the velocity $\mathbf{v}(I_x)$ by calculating the mean velocity of all points $\{x\}$ in the boundary strip Ω_x around I_x , i.e., $\mathbf{v}(I_x) = \frac{\sum_{x \in \Omega_x} \mathbf{v}(I_x)}{|\Omega_x|}$, where $|\Omega_x|$ denotes strip size n . Fig.4b shows an example of the resulting velocity field.

For each cloud boundary B_i at the t -th frame, given the estimated *boundary velocity field* $\mathcal{V}_i^t = \{\mathbf{v}(I_x^t)\}$ of B_i , the next step is to **predict cloud boundary movements** in the future $\tau = t + \Delta t$ frame, where the *prediction interval* Δt is a pre-specified system parameter. Our goal is to estimate the probability of any pixel \hat{I}_x^τ in the τ -th frame belonging to a cloud boundary. We add the $\hat{\cdot}$ symbol (e.g., \hat{I}) to indicate a predicted future quantity throughout this paper. We define a *label function* $L(\hat{I}_x^\tau)$, which takes value of 1 if the pixel \hat{I}_x^τ at time τ belongs to a cloud boundary and 0 otherwise. The cloud boundary is predicted by intersecting all individual predictions, where each individual prediction $p(L(\hat{I}_x^\tau) = 1 | \mathbf{v}(I_x^t))$ follows a Gaussian distribution from extrapolating the velocity $\mathbf{v}(I_x^t)$ with increasing variance:

$$p(L(\hat{I}_x^\tau) = 1 | \mathcal{V}^t) = 1 - \prod_{x \in I} \left[1 - p(L(\hat{I}_x^\tau) = 1 | \mathbf{v}(I_x^t)) \right].$$

Specifically, the Gaussian mean μ is derived from the velocity magnitude and the prediction interval, following a simple linear motion model $\mu(\mathbf{v}(\hat{I}_x^\tau)) = I_x^t + \mathbf{v}(I_x^t)\Delta t$. The Gaussian variance grows as the prediction interval Δt increases. The intersection of Gaussian prediction boundary comprises a probabilistic map indicating the probability of any given pixel \hat{I}_x^τ at the future frame τ belonging to a predicted cloud boundary. Thresholding this map yields the predicted cloud boundary mask $L(\hat{I}_x^\tau)$. Fig.4d depicts an illustrative example. Connected component interior filling is applied to produce the set of predicted cloud regions $\{\hat{C}_i^\tau\}_{i=1}^b$.

We experimented with pre-recorded videos, so that we could verify the cloud motion prediction $\{\hat{C}_i^\tau\}_{i=1}^b$ overlaid on top of the true future cloud image I^τ . Fig.4c illustrates an example result, where each predicted cloud region \hat{C}_i^τ is highlighted in light red. The groundtruth cloud boundary B^τ in the future frame are visualized in two colors. Blue indicates successful predictions where the groundtruth boundary falls within the predicted mask, and solid red indicates failure predictions. Observe that the majority of the successful prediction pixels, indicating a high prediction accuracy.

4. SOLAR OCCLUSION PREDICTION

Given the set of predicted cloud regions $\{\hat{C}_i^\tau\}_{i=1}^b$ and the obtained sun mask S^τ , we can **predict sun-cloud occlusion** as well as estimate the amount of the anticipated solar irradiance drop. We use a *flag function* $F(\tau)$ to indicate the prediction of an occlusion event that will cause solar irradiance drop, i.e., $F(\tau) = 1$ suggests a sun occlusion and 0 otherwise. A naive solution is to count the total sun-cloud intersection pixels or area $A^\tau = \sum_{i=1}^b |\hat{C}_i^\tau \cap S^\tau|$ followed by a applying a threshold θ_A , i.e., $F(\tau) = 1$ if $A^\tau > \theta_A$ and 0 otherwise. We observe that in practice the calculation of A^τ is very inaccurate. Simple thresholding of A^τ results in excessive switches of the flag function, thus resulting in false triggering of events.

To obtain a more robust solar occlusion estimation, we enforce the constraint that $F(\tau)$ can only switch values when the above thresholding condition $A^\tau > \theta_A$ is accumulated for more than sufficient number of consecutive frames. To this end, we introduce two additional *accumulation variables* c_- and c_+ and apply two respective accumulation thresholds θ_{c_+} and θ_{c_-} for robust prediction of sun occlusion events:

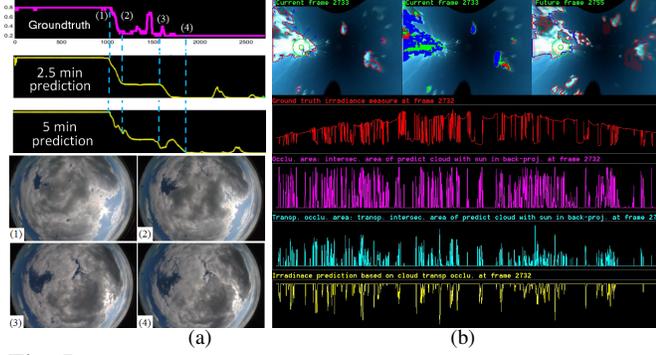


Fig. 5. Evaluations. Comparison between the recorded groundtruth and predicted irradiance measures for (a) a 40-minute sequence at two predictions of 2.5 minutes ($\Delta t = 150s$) and 5 minutes ($\Delta t = 300s$), where (1-4) depicts observed images. (b) a 7.5-hour sequence recorded at 0.1 FPS. The first view shows the input image with detected cloud boundary and the optical flow motion field. The second view shows the cloud segmentation with estimated transparency classes. The third view shows the future frame, the calculated sun mask and the predicted cloud boundary. Red curve depicts ground truth. Pink curve depicts the interaction-area based occlusion. Cyan curve depicts the interaction-area considering cloud transparency. Yellow curve depicts the sun irradiance drop estimation curve.

- If $A^T \geq \theta_A$, increase c_+ by one and reset $c_- = 0$.
- If $A^T < \theta_A$, increase c_- by one and reset $c_+ = 0$.
- If $c_+ > \theta_{c_+}$, set $F(t) = 1$ and reset $c_+ = 0$.
- If $c_- > \theta_{c_-}$, set $F(t) = 0$ and reset $c_- = 0$.

Solar irradiance drop prediction: The above accumulated intersection-area based sun occlusion estimation can be improved by considering cloud transparency that better correlates with sun irradiance drop. Recall that each cloud pixel is classified into three categories: high-, medium-, and low-transparency. For each category, we assign a fixed weighted amount of irradiance drop when a sun occlusion occurs. We define such type-specific irradiance drop estimations as:

$$\mathcal{I}^r(I_x) = \begin{cases} 1 & \text{if } T(I_x) = T_h \\ 2 & \text{if } T(I_x) = T_m \\ 3 & \text{if } T(I_x) = T_l \end{cases} \quad (1)$$

The predicted irradiance drop is the averaged over all pixel-wise estimations $\mathcal{I}^r = \frac{\sum_i \sum_{I_x \in S \cap \hat{C}_i^T} \mathcal{I}^r(I_x)}{\sum_i |S \cap \hat{C}_i^T|}$.

4.1. Estimate solar irradiance drop using back-projection

Note that the above intersection-area based sun occlusion estimation only considers a single cloud occluder event. Observe in Fig.4e that multiple cloud occluders or “frontiers” can occur at the same time, and overlapping occlusions need to be count individually. Also, the dynamic nature of cloud motion makes it hard to extrapolate the transparency estimation. To this end, we propose a novel algorithm to **back-project** the predicted sun occlusion to the cloud boundary for occlusion reasoning. In this way occlusions can be accurately and quantitatively accumulated for irradiance drop estimation.

For each detected cloud region C_i , an average speed $\mathbf{v}(C_i)$ is computed based on all the velocities of the boundary pixels that are sufficiently close to the sun image (*i.e.* $<$ threshold d_s). Let I_s denote the sun center pixel.

$$\mathbf{v}(C_i) = \frac{\sum_{\forall I_x \in B_i \text{ and } |I_x - I_s| \leq d_s} \mathbf{v}(I_x)}{\sum_{\forall I_x \in B_i \text{ and } |I_x - I_s| \leq d_s} 1} \quad (2)$$

The sun center is moved to $I_s^r = I_s - \Delta t \mathbf{v}(C_i)$ in order to generate a back-projected sun region \tilde{S}_i . If \tilde{S}_i intersects with cloud region C_i , the irradiance decrease caused by C_i is given by $\mathcal{I}_i^r = \frac{\sum_{I_x \in \tilde{S}_i \cap C_i} \mathcal{I}^r(I_x)}{|\tilde{S}_i \cap C_i|}$. Considering all detected cloud regions, we have $\mathcal{I}^r = \frac{\sum_i \sum_{I_x \in \tilde{S}_i \cap C_i} \mathcal{I}^r(I_x)}{\sum_i |\tilde{S}_i \cap C_i|}$. Fig.4d,e illustrates the back-projection sun occlusion calculation.

Our method estimates the irradiance drops (*i.e.*, “changes” of irradiance) over time (instead of an absolute value). We shift the resulting irradiance measure to zero using clear sky *i.e.* when the sun is not obscured at all.

5. EXPERIMENTAL VALIDATION

We implemented the described system in C++ on a fan-less embedded industrial computer box, which are deployed next to GE’s solar testing site at Schenectady, NY. The pipeline can run up to 10 FPS, however 0.1 FPS is sufficient for our solar forecasting application. We collected sky video sequences using a Nikon camera with resolution 1936×1296 . We also use an irradiance sensor to collect groundtruth simultaneously at a frequency of one sampler per second. All sigmoid parameters for cloud segmentation and classification in §2 are determined empirically, *e.g.*, $\mu_{bmr} = 15$, $\sigma_{bmr} = 5$, $\mu_{br} = 1.0$, $\sigma_{br} = 0.3$, $\mu_h = 0.2$, $\sigma_h = 0.05$, $\mu_{vn} = 0.9$, $\sigma_{vn} = 0.2$. We tested our algorithm with two prediction intervals $\Delta t = 150s$ and $\Delta t = 300s$ with identical parameter settings. Fig.5a reports results on a sequence contains 1426 images (45 minutes), where images are taken every 2 minutes. At time (1), the cloud on the left begins to occlude the sun, whereas at time (2) the cloud completely occludes the sun. As the clouds keep moving toward right, the darker cloud begins to block the sun at time (3). The sun is fully covered at time (4). We define the per-frame *cloud boundary prediction accuracy* $\mathcal{A} = \frac{|\sum_i \hat{C}_i^T \cap \sum_i B_i^T|}{|\sum_i B_i^T|}$ for qualitative evaluation. For a video, the average of \mathcal{A} ’s over all frames are used. We obtained $\mathcal{A} = 0.85$ for $\Delta t = 150s$, and $\mathcal{A} = 0.86$ for $\Delta t = 300s$, which demonstrated the effectiveness of our method. Fig. 5b reports results on a longer sequence about 7.6 hours at 0.1 FPS. Observe how the prediction curves align and correlate with the groundtruth irradiance curves. Note that our resulting irradiance prediction curve (yellow) can be further scaled by the overall irradiance of a clear sky from measures throughout a day, to yield a long-term irradiance evaluation that better matches the (red) groundtruth envelope curve. The **Bird Clear Sky Model** [17] authored by Richard Bird serves well for such scaling purpose.

6. CONCLUSIONS

We present a video analytic system to segment and track clouds for short-term solar irradiance drop prediction. **Future work** includes finer classification of cloud types and transparencies.

7. REFERENCES

- [1] G. Pfister, R. L. McKenzie, J. B. Lileya, A. Thomasa, B. W. Forganb, and C. N. Long, “Cloud coverage based on all-sky imaging and its impact on surface solar irradiance,” *Journal of Applied Meteorology*, vol. 42, no. 10, pp. 1421–1434, 2003.
- [2] A. Kreuter, M. Zangerl, M. Schwarzmann, and M. Blumthaler, “All-sky imaging: a simple, versatile system for atmospheric research,” *Applied Optics*, vol. 48, no. 6, pp. 1091–1097, 2009.
- [3] A. Heinle, A. Macke, and A. Srivastav, “Automatic cloud classification of whole sky images,” *Atmospheric Measurement Techniques*, vol. 3, no. 3, pp. 557–567, 2010.
- [4] M. S. Ghonima, B. Urquhart, C. W. Chow, J. E. Shields, A. Cazorla, and J. Kleissl, “A method for cloud detection and opacity classification based on ground based sky imagery,” *Atmos. Meas. Tech. Discuss.*, vol. 5, pp. 4535–4569, 2012.
- [5] D. K. Chaturvedi and Isha, “Solar power forecasting: A review,” *International Journal of Computer Applications*, vol. 145, no. 6, 2016.
- [6] Chi Wai Chow, Serge Belongie, and Jan Kleissl, “Cloud motion and stability estimation for intra-hour solar forecasting,” *Solar Energy*, vol. 115, pp. 645–655, 2015.
- [7] Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom, “Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed,” vol. 85, no. 11, 2011.
- [8] Philip Wood-Bradley, Jose Zapata, and John Pye, “Cloud tracking with optical flow for short-term solar forecasting,” in *Australia and New Zealand Solar Energy Society Conference*, 2012.
- [9] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.
- [10] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [11] D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easy calibrating omnidirectional cameras,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 297–310, 2006.
- [12] Ibrahim Reda and Afshin Adnreas, “Solar position algorithm for solar radiation applications,” Tech. Rep., NREL, <http://www.nrel.gov/docs/fy08osti/34302.pdf>, 2008.
- [13] J. Meeus, *Astronomical Algorithms*, Willmann-Bell Inc., Richmond, Virginia, USA, 2 edition, 1998.
- [14] Z. Zhang, “On the epipolar geometry between two images with lens distortion,” *International Conference on Pattern Recognition*, vol. 1, pp. 407–411, 1996.
- [15] D.C. Brown, “Decentering distortion of lenses,” *Photogrammetric Engineering*, vol. 7, pp. 444–462, 1966.
- [16] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, “Retrieving cloud characteristics from ground-based daytime color all-sky images,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, 2006.
- [17] R. E. Bird and R. L. Hulstrom, “Simplified clear sky model for direct and diffuse insolation on horizontal surfaces,” *Technical Report No. SERI/TR-642-761*, Golden, CO: Solar Energy Research Institute, 1981.