

Article

A Low Cost, Edge Computing, All-Sky Imager for Cloud Tracking and Intra-Hour Irradiance Forecasting

Walter Richardson Jr. ^{1,*}, Hariharan Krishnaswami ², Rolando Vega ³ and Michael Cervantes ⁴

¹ Department of Mathematics, University of Texas at San Antonio, San Antonio, TX 78249, USA

² Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA; Hariharan.Krishnaswami@utsa.edu

³ CPS Energy, San Antonio, TX, 78205, USA; rolveg@yahoo.com

⁴ Merrick & Company, San Antonio, TX 78258, USA; Michael.Cervantes@merrick.com

* Correspondence: Walter.Richardson@utsa.edu; Tel.: +1-210-458-4760

Academic Editor: Tomonobu Senjyu

Received: 31 December 2016; Accepted: 17 March 2017; Published: 23 March 2017

Abstract: With increasing use of photovoltaic (PV) power generation by utilities and their residential customers, the need for accurate intra-hour and day-ahead solar irradiance forecasting has become critical. This paper details the development of a low cost all-sky imaging system and an intra-hour cloud motion prediction methodology that produces minutes-ahead irradiance forecasts. The SkyImager is designed around a Raspberry Pi single board computer (SBC) with a fully programmable, high resolution Pi Camera, housed in a durable all-weather enclosure. Our software is written in *Python 2.7* and utilizes the open source computer vision package *OpenCV*. The SkyImager can be configured for different operational environments and network designs, from a standalone edge computing model to a fully integrated node in a distributed, cloud-computing based micro-grid. Preliminary results are presented using the imager on site at the National Renewable Energy Laboratory (NREL) in Golden, CO, USA during the fall of 2015 under a variety of cloud conditions.

Keywords: solar forecasting; global horizontal irradiance; single board computer; optical flow; cloud motion vectors; ray tracing; micro-grid

1. Introduction

To enable higher renewable energy penetration into the world electric power grid with high reliability, variable resources such as photovoltaic (PV) power plants and wind farms will require development of new technologies. In the case of PV, it will become necessary to accurately forecast the Global Horizontal Irradiance in time and space to manage variability [1]. Novel technologies being developed at the University of Texas at San Antonio will integrate distributed, High Dynamic Range (HDR) sky imaging systems and feature-based cloud advection models to produce accurate geospatial high-resolution irradiance forecasts in the intra-hour time scale. The knowledge of predicted solar irradiance minutes-ahead can allow inverters to intelligently regulate power into the grid, even during intermittent cloudy conditions with attendant ramping events, thereby allowing the utility to satisfy constraints for curtailment, frequency regulation, and active power control. This task involves integration of new hardware for data acquisition and software for image processing, using both physics-based and statistics-based machine learning strategies for intelligent cloud tracking and prediction.

A recent report from the California Independent System Operator Corporation (CAISO) and the North American Electric Reliability Corporation (NERC) [2] states that California has set a target of 33% renewable electric generation in utility portfolios by the end of the decade. Achieving this goal will require changes to existing practices of balancing supply and demand by CAISO. In addition,

as the popularity of residential rooftop solar grows, a better understanding of distributed PV power generation and its interconnection with the environment and load is needed. Improvements in renewable energy forecasting will allow system operators to procure energy and ancillary services in the intra-hour to day-ahead time frames, thereby minimizing costs and improving service to their customers. Satellite and numerical weather prediction are the preferred methods [2] for longer duration solar forecasting (one hour to a few days). In the case of intra-hour or site-specific forecasts, the use of satellite data is uncommon due to the infrequent sampling interval (30–60 min) and the low image resolution, when compared to sky imaging cameras that acquire high resolution images at much shorter intervals (1–15 s).

1.1. UTSA SkyImager

The UTSA SkyImager is the culmination of three years of work by the authors and former graduate students Jaro Nummikoski and Alejandro Camargo. The path was evolutionary and several bifurcations occurred along the way. Initially, it was thought that existing hardware would suffice to meet our solar forecasting needs [3] and our emphasis was on software. When it became apparent that new hardware was needed, the UTSA Macroscope was conceived [4]. While a significant advance in technology, a price tag in the tens of thousands of dollars precluded widespread deployment in a distributed environment. As solar power moves from megawatt generating plants to distributed residential rooftop arrays, there will be an increasing need for active grid management [5] which entails accurate forecasting of the ramps in power output caused by low level cloud movement. In addition, there are islanded micro-grids and situations in Africa and Latin America [6] where solar may well be the dominant power source. There is a pressing need for a *low cost all-sky imager* that provides GHI prediction and micro-grid control [7–9] in real time. Enter the ubiquitous ARM-architecture chip that is found in a cell phone. A single board computer, such as the Raspberry Pi or the Hardkernel Odroid, runs the Linux operating system, takes high-resolution images, allows for a variety of external sensors to be added through the general purpose input-output (GPIO) pins, and can fill a significant role in the Internet of Things (IoT), all for less than \$50. We decided to make the SBC a central theme of the new UTSA SkyImager [10]. The components of the SkyImager described in this paper cost close to \$500, mainly due to the security camera enclosure. However, work is currently underway on a leaner model with a price tag under one hundred dollars.

In this paper, we introduce a novel technology that significantly expands capabilities for forecasting intra-hour GHI and distributed PV power. The primary focus is the newly designed, low cost all-sky UTSA SkyImager (patent pending) shown in Figure 1a. It was deployed at the National Renewable Energy Laboratory in Golden, Colorado, as shown in the satellite photo in Figure 1b. In terms of hardware, the SkyImager is designed around a Raspberry Pi SBC with a fully programmable, high resolution Pi Camera, housed in a durable all-weather enclosure. Our software is written in *Python 2.7* and utilizes the open source computer vision package *OpenCV* [11,12]. The SkyImager can be configured for different operational environments and networks, from a standalone edge computing model to a fully integrated, distributed system utilizing cloud-computing.



Figure 1. (a) UTSA SkyImager deployed on the rooftop of the ESIF building at The National Renewable Energy Laboratory in Golden, CO, USA; (b) satellite photo showing the location of the solar arrays and imager.

1.2. State of the Art in Solar Forecasting

As PV assumes a larger role in the Smart Grid, accurate solar forecasting will require a network of low cost, distributed sensors to acquire and process large amounts of image and weather data as inputs to machine learning algorithms. Existing commercial sky imaging systems often prove to be too costly or are constrained by the proprietary nature of ancillary software licenses. This has led several groups to develop their own sky imaging systems for research purposes in solar forecasting. A large group at UC San Diego [13–16] has done pioneering work in this field. In [17], Coimbra et al. proposed DNI forecasting models that combined data from a Yankee TSI 880 sky imager with a hemispherical mirror and Artificial Neural Network (ANN) optimization schemes. The TSI has high capital and maintenance costs, uses a shadowband mechanism, and requires proprietary software for image acquisition and transfer. The UCSD Sky Imager or USI is described in Yang et al. [18] and captures images using an upward facing charge-coupled device (CCD) image sensor from Panasonic and a 4.5 mm focal length circular fisheye lens. Advantages over the TSI include higher resolution, dynamic range, and lossless PNG compression. The Universität Erlangen-Nürnberg [19] used a five megapixel C-mount camera also equipped with a fisheye lens. Thirions “daemons” algorithm for image registration was implemented for cloud-motion estimation, not unlike the optical flow described later in this paper. The paper by West et al. [20] presents a nice overview of solar forecasting research at CSIRO in Canberra, Australia. They used off-the-shelf IP cameras such as the Mobotix Q24 and the Vivotek FE8172V, and while inexpensive compared to the TSI, the cost is still in the range of 800 euros. They used a dense optical flow algorithm to estimate the movement of each pixel, rather than a feature-tracking strategy. Several groups in China are working on solar irradiance forecasting [21,22], generally with a TSI, but another uses Geostationary Satellite data [23]. Recent developments in GOES-R (geostationary satellite with high spatial and temporal resolution) may well make the satellite approach to minutes-ahead irradiance prediction more attractive.

1.3. Problem Statement and External Inputs

The fundamental problem in short-term solar irradiance forecasting is predicting the location of shadows cast on PV arrays when low level cumulus clouds come between them and the sun. This causes a sudden, large drop in power output, called a “ramp”, which lasts until the cloud moves out of the sun-to-panel path. This process occurs at scales (space and time) that are below the resolution of numerical weather prediction or machine learning approaches that are applicable to day-ahead forecasting. Our approach is physics-based in that it determines the current position of the base of a cumulus cloud, and then extrapolates that position into the future. The model necessarily simplifies reality as it attempts to capture the dominant causes of ramping. Meteorologists classify clouds into three levels (low, middle, high), and we consider only low level clouds. The SkyImager only “sees” to the horizon and cannot predict the arrival of a cold front until it begins to be visible on the horizon.

Moreover, low level stratus and cumulus clouds can have considerable vertical development. They are not static objects, but constantly evolving, growing and dissipating with time. Other approaches are certainly possible. One would be a strategy based upon statistics or machine learning that predicts the grayscale value of each pixel into the future without trying to track individual clouds or assign cause-and-effect. This will form an area of future research.

In order to predict cloud movements and make GHI forecasts, our methodology requires several types of external input data besides the high resolution, 1024×768 pixel images acquired from the Pi camera: current cloud base height (CBH) obtained from ceilometer measurements, true north in image pixel coordinates, zenith angles of the sun position for the particular time of day and year, and specific optical properties of the camera and fish eye lens. CBH refers to the distance from the ground to the base of observed cumulus clouds. These clouds are formed by adiabatic cooling as the air parcel is lifted, and the CBH is the lifting condensation level from meteorology. These heights are obtained from a device called a ceilometer, usually located at a nearby city airport where weather observations are taken at least hourly. Ceilometer data for the NREL site was obtained from the Denver International Airport using the Internet and the Automated Surface Observing System (ASOS) (see Table 1).

Table 1. Sample data from the METAR Observations for Denver International Airport on 27 October 2015. Time is Mountain Standard (7 h behind UTC) and the heights of cloud bases at the Low, Middle, and High Levels are in feet above ground level. Note the very low stratus deck at 15:08 MST that resulted in a special observation and was likely caused by a rain shower. The Denver airport is 36 mi from the NREL site.

Time	Low	Middle	High	Time	Low	Middle	High
12:53	7000	18,000	22,000	15:08	200	7000	12,000
13:53	7000	12,000	22,000	15:53	3000	7000	12,000
14:53	8000	12,000	18,000	16:38	7000	12,000	17,000

Zenith angles are determined from NREL's program SOLPOS, the solar position and intensity calculator, with data calculated at 1-min intervals using NREL's latitude and longitude (39.742, -105.170). This is done at the beginning of each day's data run. Parameters determined by a camera calibration are used to correct for the nonlinear distortion caused by the fish eye lens. The angle of deviation is utilized to rotate and align the image to "True North". Knowing the CBH and zenith angle of the sun, the predicted locations of the cumulus cloud base minutes ahead are used together with ray tracing and a satellite photo of the region of interest surrounding the site, to find the shadows cast on the ground by those clouds. Figure 2 shows the results of these computations for the NREL complex in Golden, CO, USA on the morning of 23 October 2015.



Figure 2. Forecast cloud shadows from the ray-tracing algorithm in the SkyImager for three times on 23 October 2015. Shadow-free areas are light against the background, while the dark regions depict shadows.

2. Materials and Methods

Images are acquired using a Raspberry Pi 2 B (Raspberry Pi Foundation: Cambridge, UK, 2015) equipped with the Pi Camera. Pictures taken in direct sunlight suffer from overexposure in regions

close to the sun and underexposure in areas of dark shade. This is because the complementary metal-oxide semiconductor (CMOS) image sensor (SONY IMX219PQ in the Camera Module V2) lacks the dynamic range of the human eye. To compensate for this, our software incorporates an exposure fusion operation [24,25]. Three images at different exposure times are captured and fused together to produce the HDR image shown in Figure 3a.

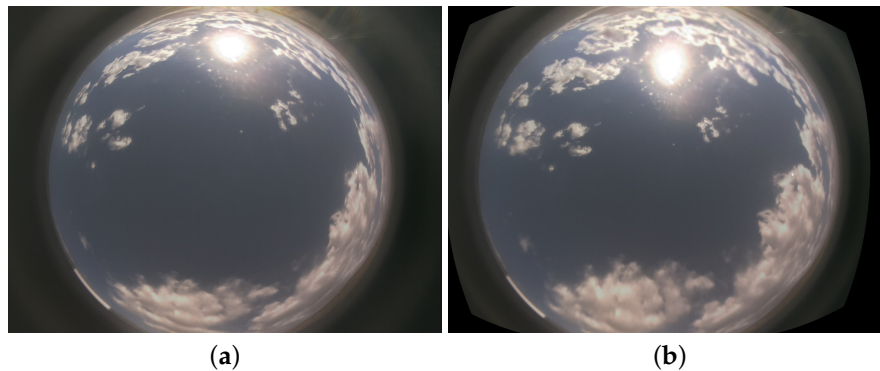


Figure 3. (a) fused raw image captured by the Raspberry Pi Camera on 27 October 2015 at 11:23; (b) transformed image after the distortion caused by the fisheye lens has been removed.

2.1. Functions: Undistort, Crop, Mask, and Segment

The first step in preprocessing a raw image is to remove distortion caused by the fish eye lens. Due to its very wide field-of-view ($>180^\circ$), a fish eye lens produces severe nonlinearities, particularly near its edges. Mathematical transformations can be applied to correct for this distortion, but the problem is similar to that encountered by the cartographer who desires to map the sphere to a plane. One must choose the properties of greatest interest to preserve in the transformation—straight lines, angles, areas, etc.—knowing that others will be sacrificed in the process. The projective transformation model for a pin-hole camera [26] takes a 3D point $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ in world coordinate space and projects it to the point (u, v) in pixel coordinates. The matrix–vector equation describing the transformation is $s \mathbf{m}' = \mathbf{A}[\mathbf{R}|\mathbf{t}]\mathbf{M}'$, which in component form is

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \\ \mathcal{Z} \\ 1 \end{bmatrix}. \quad (1)$$

\mathbf{A} is the “camera matrix” of intrinsic parameters, (c_x, c_y) is the image center, and f_x, f_y are focal lengths in pixels. The following equations are used to correct for radial and tangential distortions introduced by the fisheye lens:

$$x'' = x' \left(1 + k_1 r^2 + k_2 r^4 \right) + 2p_1 x' y' + p_2 (r^2 + 2x'^2), \quad (2)$$

$$y'' = y' \left(1 + k_1 r^2 + k_2 r^4 \right) + p_1 (r^2 + 2y'^2) + 2p_2 x' y'. \quad (3)$$

The *calibrateCamera* function in the Open Computer Vision package was used to perform the calibration of the camera using a standard checkerboard test pattern, which is positioned at various locations in the camera’s field of view. An optimization algorithm is then utilized to find the best fit for a nonlinear model of the distortion produced by the lens. The results are shown in Table 2.

As seen in Figure 3b, the undistorted image contains extraneous pixels outside the circular disk of active pixels—an artifact of the rectangular (not square) area of the CCD chip used to acquire the image. A simple crop produces a square output image with a centered disk of active pixels. A mask is

then applied to eliminate that part of the image close to the horizon that may contain clutter such as buildings, trees, and terrain. For accurate forecasting of cloud motion, it is necessary that the image includes only sky and clouds.

Table 2. Parameters used by the undistort function.

K1	K2	P1	P2	FX	FY	CX	CY
−0.3498788	0.09006071	0.00364227	0.00254462	440	440	478	376

The next step in the image processing sequence is critical: differentiating between clouds and clear sky. On some days, this is relatively easy with air mass cumulus clouds moving through an otherwise clear blue sky as background. At other times, the problem is more complex [27] with multiple cloud layers above the cumulus, such as altostratus and cirrostratus. Because the most important output of the intra-hour solar forecast is to predict cloud movements and ramping, the algorithm focuses on low-level cumulus clouds, which move between the sun and the solar array.

To detect which pixels represent low level clouds, the “Red-to-Blue Ratio” (RBR) method developed at the Scripps Institute of Oceanography [28] is used. It is based upon the different scattering mechanisms for light hitting air molecules versus water vapor in clouds. Clear sky is Rayleigh scattering, which scatters more blue light than red, whereas water molecules in clouds scatter blue and red light almost equally. Details appear in Algorithm 1, in which *rbr* is the red to blue ratio in the range [0, 255] and *red* (*blue*) represents the value of the corresponding color channel in the pixel. The output of this step is a grayscale image with low intensity values representing clear sky and high intensities likely to be clouds. Figure 4a shows the resulting image.

Algorithm 1: Calculating the Red-to-Blue Ratio for determining Cloud vs. Clear Sky

```

1 if red > blue then
2   |  $rbr = 129 + \left( \frac{red/blue}{\max(red/blue)-1} \right) \times 127;$ 
3 if red < blue then
4   |  $rbr = 127 - \left( \frac{red/blue}{1-\min(red/blue)} \right) \times 127;$ 
5 else
6   | rbr = 128;
7
```

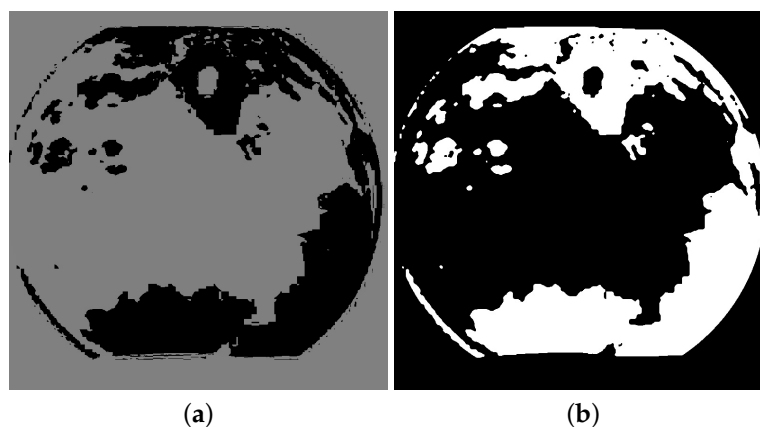


Figure 4. (a) initial transformation into a grayscale image using the Red-to-Blue Ratio that will determine the presence or absence of clouds; (b) binary image in which white represents cloud and black denotes no-clouds.

After calculating the *RBR* for the current image and its clear sky counterpart, the two images are subtracted to isolate the clouds. Although not a perfect cloud detector, the methodology does identify the majority of cumulus clouds and accurately delineates their boundaries. Because the resulting *rbr* images contain jagged edges around cloud boundaries, a median filter [29] is applied to the grayscale image. Finally, thresholding transforms the image from grayscale to binary, with clouds represented by white and clear sky by black. The binary image, ready for cloud motion forecasting, is shown in Figure 4b.

2.2. Geometric Transformation

The physical model of a cumulus cloud used in our forecasting software is an idealization of the actual cloud. First, we assume that the base for all low level clouds is the lifting condensation level or CBH as measured by a ceilometer at the nearest airport. The second assumption is more questionable: the cloud is essentially flat with little or no vertical development. By definition, cumulus clouds are formed by convection, when a parcel of air is lifted, cooled, and water vapor condenses. The process is evolutionary, constantly building and dissipating, but there is always vertical development. The extreme case is a cumulonimbus cloud which may tower many kilometers into the sky. Any system for capturing sky images must necessarily involve projections from 3D objects to the 2D image plane. This can be the perspective projection of a pinhole camera or the equidistant transformation of a fish eye lens. Our current technology does not permit accurate measurement of a cloud's vertical development, only a perspective projection of the entire cloud (note, however, that by using a pair of SkyImagers separated by a reasonable distance, one could obtain depth information as happens in stereo vision). For this reason, a geometric transformation [3] is applied to the binary image. It projects the 2D cloud pixel locations to a physical plane parallel to the tangent plane to the earth, but at a height equal to the CBH. A *Python* scraper program downloads the CBH from the ASOS database [30]. Extreme distortion occurs at the fringes of the lens, so there is a user-defined minimum sky altitude measured from the horizontal. Pixels below this are simply discarded. With these two parameters, the maximum cloud base radius and the linear distance per pixel value on the elevated plane can be calculated. These, in turn, define a coordinate transformation which is applied to the binary image. The result is shown in Figure 5a. For the ray-tracing algorithms described later, the image must then be rotated to "True North" as indicated in Figure 5b.

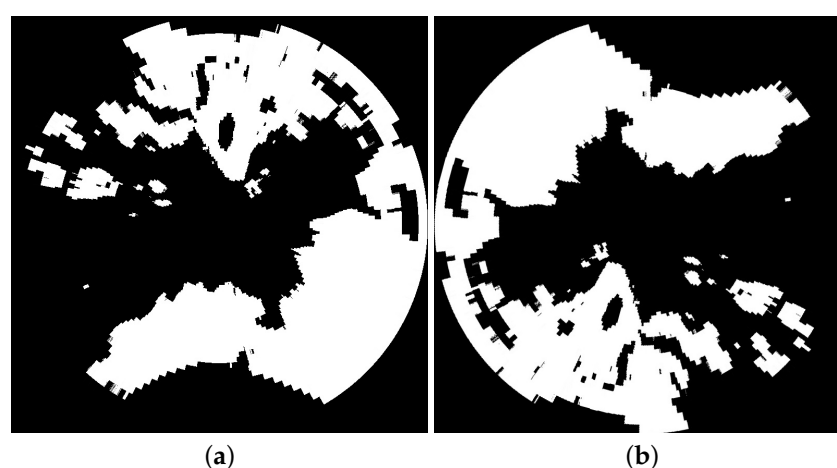


Figure 5. (a) binary image after transformation to CBH; (b) image rotated to True North in preparation for ray-tracing. In the case of the NREL site, this was -195° .

2.3. Persistence and Percentage of Cloud Cover

For minutes-ahead predictions, particularly on days with no clouds or completely overcast conditions, it is hard to beat persistence. The persistence forecasting method is based upon the simple assumption that the current GHI value will not change over the time interval of interest, except for the diurnal changes under clear sky conditions. Hence, the next step in the pipeline is to classify the binary image according to the percentage of white pixels in the disk. The three categories are: clear sky, overcast sky, or moderately cloudy. The percentage of cloud cover for each sky condition are shown in Table 3.

Table 3. Assigning sky condition—clear, overcast, or moderately cloudy.

Sky Condition	Clear Sky	Overcast Sky	Moderately Cloudy
Percentage	<15% White Pixels	>70% White Pixels	15%–70% White Pixels

After classification of the current image, a decision is made as to which forecasting method to use. For clear and overcast sky conditions, our algorithm uses persistence. A moderately cloudy sky condition with 15%–70% cloud cover is likely to indicate the presence of air-mass cumulus clouds. In this case, cloud motion tracking by optical flow is used to predict future cloud locations, and then ray-tracing is employed to forecast shadows and GHI.

2.4. Optical Flow & Cloud Motion Vectors

The term *optical flow* in image processing refers to the apparent motion of the brightness pattern in a time sequence of images. An old, but important problem in computer vision [31], it generally corresponds to the motion field itself: objects moving through the field of view of the camera or the camera itself moving. The goal is to find a pair of functions $u(x, y)$, $v(x, y)$ that represent the x - and y -components of the velocity of the intensity function $E(x, y)$ at pixel location (x, y) . The *optical flow constraint equation* $E_x u + E_y v + E_t = 0$ must be satisfied by u and v , but this is insufficient to determine a unique solution pair. An additional constraint must be added. This is done by incorporating a regularization term that enforces smoothness on the solutions. The goal is to minimize the total energy $e_s + \lambda e_c$ functional

$$e_s + \lambda e_c = \iint \|\nabla u\|^2 + \|\nabla v\|^2 dx dy + \lambda \iint (E_x u + E_y v + E_t)^2 dx dy. \quad (4)$$

The corresponding Euler–Lagrange equations are a coupled pair of elliptic PDEs that must be solved at each time step to obtain u and v . This can be done in several ways including iteratively:

$$\nabla^2 u = \lambda(E_x u + E_y v + E_t)E_x \quad ; \quad \nabla^2 v = \lambda(E_x u + E_y v + E_t)E_y. \quad (5)$$

Tracking the movement of $E(x, y)$ at each pixel location (dense optical flow) is computationally expensive [32], and some form of dimensionality reduction is needed to efficiently track the motion of cumulus clouds. *OpenCV* has several implementations of optical flow, some of which track a sparse set of features in the image, while others track each pixel. In our software, the Lucas–Kanade [33] approach is used as implemented in the function `cv2.calcOpticalFlowPyrLK`, with a prior call to `cv2.goodFeaturesToTrack` that uses the most prominent corners in the image for tracking [34]. Note that this feature extraction strategy implicitly finds information about first and second partial derivatives of the intensity E . Figure 6a–c show the predicted locations of detected corners (indicated by white dots) 5-, 10-, and 15-min ahead in time. Information from repeated calls to the optical flow function allows the computation of cloud motion vectors and extrapolation of cloud boundaries.

1. One feature point in the past image is selected; an attempt is made to find a corresponding nearby feature point in the present image.
2. If a match is found, the pair of feature points is recorded and the velocity vector (direction and magnitude) is calculated.
3. This process is repeated for all feature points in sequence of consecutive images.
4. Velocity vectors generate predicted cloud movement.

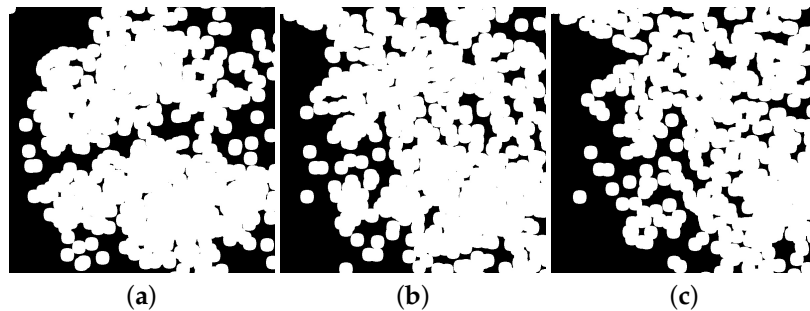


Figure 6. Forecast locations of features detected and tracked using the optical flow algorithm in OpenCV. These will be used to determine the cloud motion vectors. (a) 5-min features forecast on 23 October 2015 made at 11:13 MST; (b) feature locations 10 min into the future; (c) locations 15 min ahead.

2.5. True North Determination and Ray Tracing for GHI Prediction

While predicting cloud movements and the resulting shadows is a critical intermediate step, the ultimate goal is to accurately forecast GHI and the power produced by the PV arrays. The last step in the pipeline is to make a prediction of the GHI in the minutes-ahead time frame. If a shadow is predicted to be over the solar array, then direct sunlight is being blocked and a large drop in DNI will occur. While DHI will also be influenced by the presence of shadows over the surrounding terrain, the dominant term is DNI as shown in the equation $GHI = \cos(\theta)DNI + DHI$.

In order to utilize the predicted cloud locations together with ray tracing to find where shadows of cumulus clouds will be located, it is necessary to know precisely the direction of True North relative to the camera frame of reference. While this can be accomplished physically with a compass during installation of the SkyImager or with a Global Positioning System (GPS) module added to the GPIO pins of the Raspberry Pi, it was convenient to write a program to determine True North just using data gathered from the camera itself. First, a sequence of low exposure images is acquired on a clear sky day and used to plot the trajectory of the sun as determined by the camera. Then, the zenith and azimuth angles of the sun for the same day are calculated from NREL's online SOLPOS program. The Solar Position Algorithm (SPA) calculates the solar zenith and azimuth angles in the period from the years -2000 to 6000 , with uncertainties of ± 0.0003 degrees based on the date, time, and location on Earth [35]. A second trajectory of the computed sun position is plotted in the image reference frame based upon this data. Finally, an optimization routine is employed to determine the degrees of rotation required to align the one curve with the other. The set of images that were acquired at the NREL ESIF facility required a rotation of $\approx -195^\circ$. Finding the true north correction is performed as a calibration step during initial set up of the camera and does not have to be repeated.

Using the predicted locations of cumulus clouds on the horizontal plane at CBH and the precise location of the sun, simple ray tracing provides the locations of cloud shadows cast on the ground. Figure 7a–c show the results of these computations for the entire field-of-view of the camera. After the ray-tracing has been performed, a Region-Of-Interest is selected from each of the three forecasts and used to determine the precise locations of shadows in the vicinity of the ESIF building where the

SkyImager is located. Figure 8a–c show the results of these calculations. They also demonstrate the undesirable aspects of a ray-tracing approach: mathematically, the problem is *ill-posed* with small errors in the input data causing potentially large errors in the resulting output. Note that the DHI is relatively easy to predict, and the extreme variability of the measured GHI is due mainly to the DNI term, which, in turn, is determined by clouds moving through the circumsolar region of the image. However, the methodology proposed does not make explicit use of the pixels in the neighborhood of the sun, but attaches equal weight to all pixels in determining cloud locations. Then, with knowledge of the sun's location and ray-tracing, one attempts to truly make a “point” forecast of the shadows on the solar arrays. The instability of the algorithm explains the results obtained when compared with persistence.

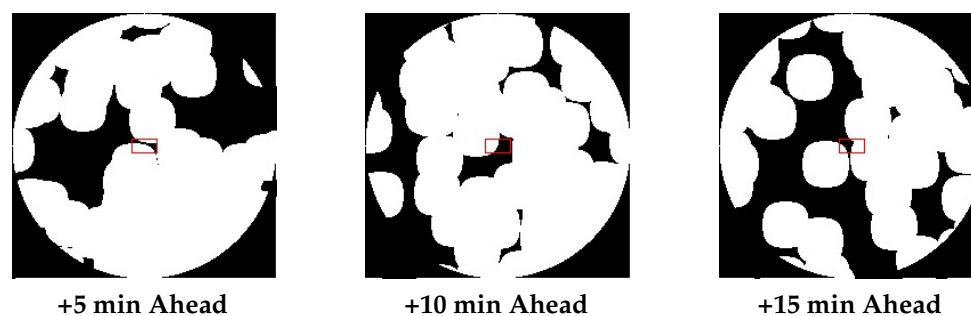


Figure 7. Results of the ray-tracing 5-, 10-, and 15-min ahead, performed on 23 October 2015 at 11:23. Areas in white represent bright sunlight on the ground, while areas in black denote shadows cast by cumulus clouds.

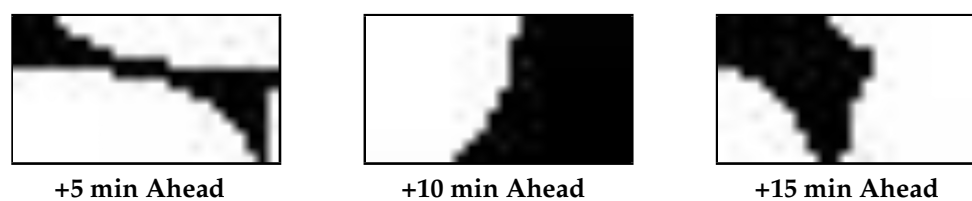


Figure 8. Shadow locations 5-, 10-, and 15-min ahead in the region of interest. The red rectangles in the centers of the images in Figure 7 delimit the region of interest immediately surrounding the camera location.

Once GHI has been forecast, it is usually straightforward to assign a corresponding power produced, which is then fed into the micro-grid management system. Figure 9 shows the functional relationship determined between GHI and power produced for the RSF2 solar generation unit at NREL (a 408 kW commercial rooftop array). GHI values are acquired using one Kipp and Zonen CM22 and four LiCOR 200 pyrhelimeters at different locations throughout the laboratory. After GHI and power forecasts are made, these values are then published to the micro-grid management system using a protocol such as MQTT. Message Queue Telemetry Transport is a publish-subscribe-based “lightweight” messaging protocol for use on top of the TCP/IP internet protocol. It is designed for connections with remote locations where network bandwidth is limited.

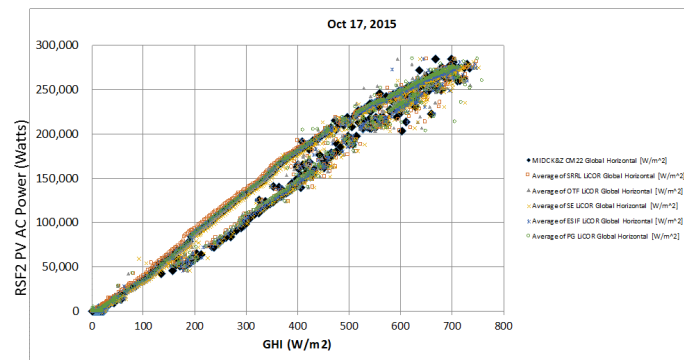


Figure 9. The functional relationship between GHI measured in W/m^2 and PV AC power production in *Watts* for the NREL RSF2 solar generation facility for 17 October 2015.

3. Solar Variability

Almost a terabyte of image data was collected at NREL from 15 October 2015 through 16 April 2016. Preliminary analysis of a small sample of the data is presented here; a more detailed analysis will appear in a forthcoming paper. Figure 10 shows irradiance plots for 12 days beginning with 14 October 2015 and ending with 19 November 2015. Each day's plot shows DNI, GHI, and Direct Horizontal Irradiance (DHI). Although for our purposes GHI is the most important of the three quantities, because it will serve as ground truth for the forecasting algorithms, forecasting the DNI component is the challenging part of the problem. The measured DNI values were recorded using NREL's *CHP1-L* pyrheliometer with units of *Watts/m²*. Initial inspection shows that sky conditions can be grouped into three categories as regards solar variability and ramping events. The first category (**Clear Sky**) consists of predominantly clear sky with little or no cloud cover and no ramping events: 14 October, 2 November, and 3 November. Category 2 (**Overcast**) results from a large cloud mass that obscures the sun during a good part of the day: the afternoons of 27 October and 15 November. Finally, Category 3 (**Moderately Cloudy**) encompasses the remaining days and is characterized by large swings in the DNI values resulting in multiple ramping events. In this manner, we account for the *variability* of the solar forecasting problem [36,37].

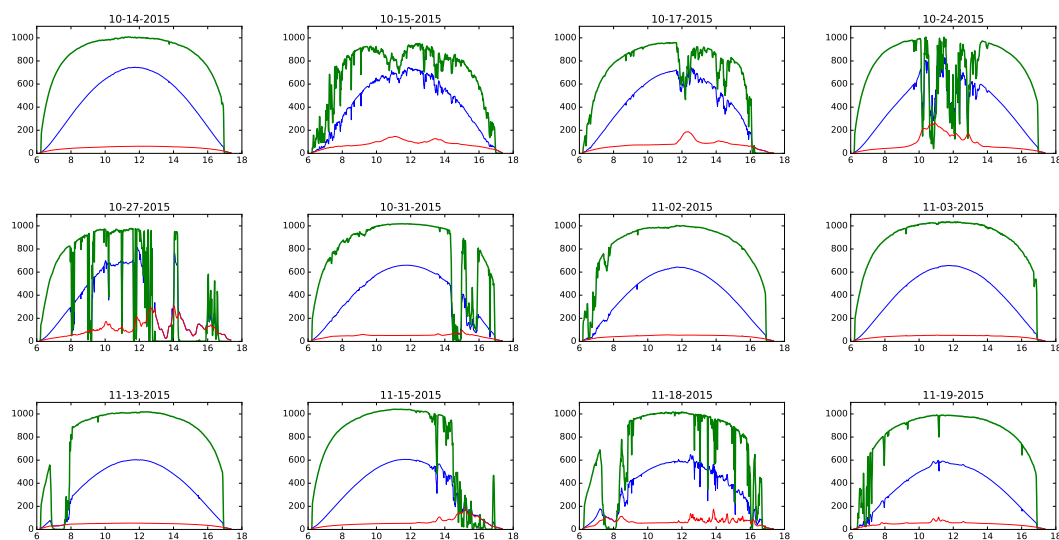


Figure 10. Irradiance values for 12 days in late fall of 2015 measured at the NREL ESIF building in Golden, CO, USA. Green denotes Direct Normal Irradiance, blue is Global Horizontal Irradiance, and red is Diffuse Horizontal Irradiance.

It is instructive to zoom in upon a ramp event and try to correlate the drop in DNI with cloud movement in the image sequence. Unless this is possible visually, there is little chance that machine learning or physics-based approaches will be successful. Consider the observed irradiance values for 27 October 2015. Figure 11 shows a subsequence of the images captured on this day, taken every 2 min beginning at 12:31 (upper left) and ending at 12:54 (lower right). The time lapse begins with no clouds in the circumsolar region, but clouds on the horizon moving rapidly into the region from left to right. By 12:37, clouds have blocked the direct path from the sun to the site, thereby drastically reducing the DNI. At the same time, significant amounts of radiation are still diffusing from the atmosphere so that the GHI decreases briefly to the value of the DHI. By 12:50, those clouds have moved off and DNI recovers.

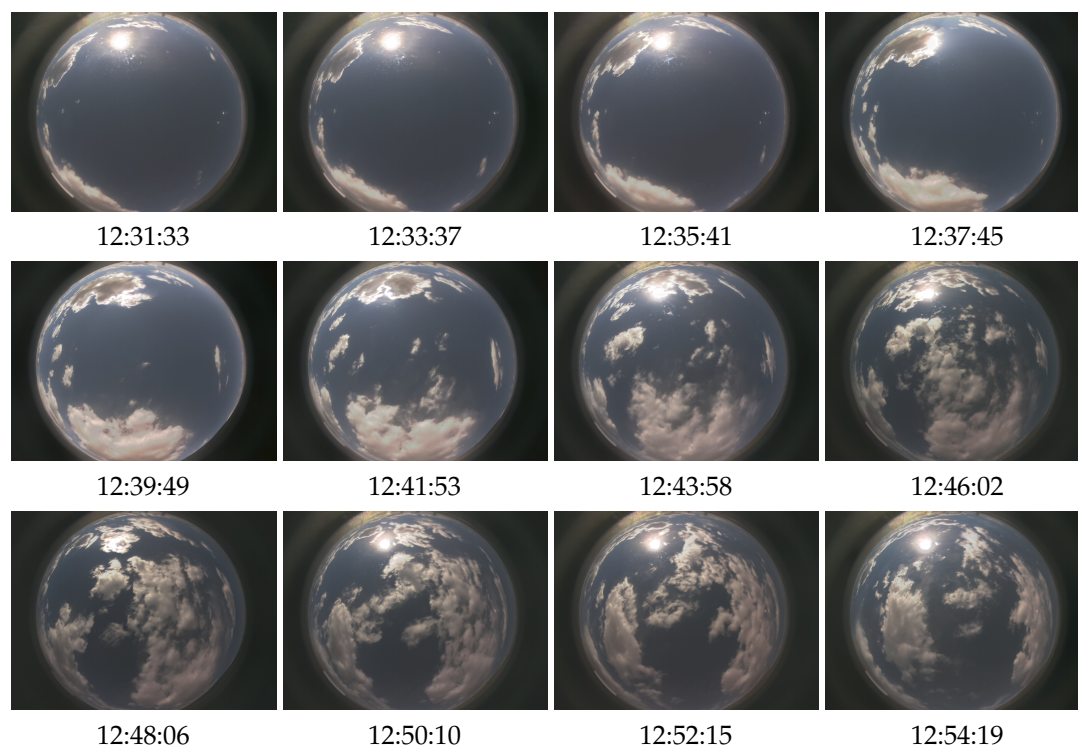


Figure 11. A subsequence of the images captured on 27 October 2015. There are 2 min between the displayed images, beginning at 12:31 (upper left) and ending at 12:54 (lower right). A major ramp event is occasioned by a cloud obscuring the sun.

This evolution of clouds and irradiance is even more striking when the video sequence is viewed, and confirms the fact that pictures from the SkyImager are highly correlated with observed values of the irradiance time series. Moreover, it suggests the camera sensor could be used to measure irradiance as well as forecast it. The blue line in Figure 12 shows the GHI time series on 27 October 2015 and confirms the ramp events occurring during the noon hour. Also plotted are the 10- and 15-min forecasts using both persistence and the camera method. The two track each other closely, since the latter algorithm uses persistence whenever conditions are clear sky or overcast. Only in the case of moderately cloudy conditions is the persistence value modified and then only when a shadow is forecast over the camera site.

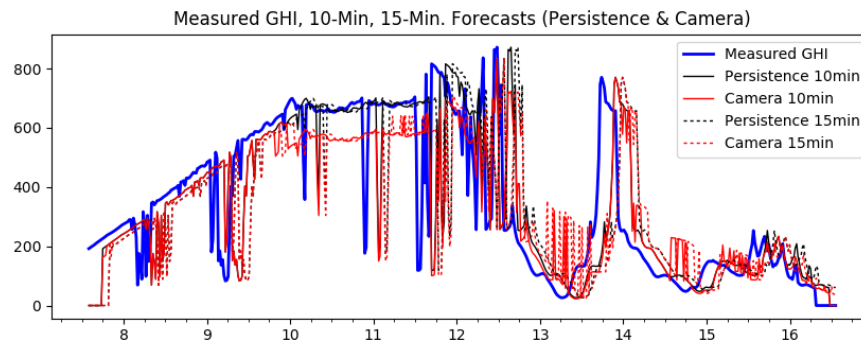


Figure 12. Measured GHI (blue line) for the day of 27 October 2015. Note that the ramp events shown in Figure 11 are clearly indicated by the wildly oscillating GHI values that occur during the noon hour. Also plotted are the 10- and 15-minute forecasts using both persistence (black) and the camera method (red).

4. Discussion

Standard metrics for evaluating the performance of irradiance forecasting algorithms are **MAPE**—Mean Absolute Percentage Error and **RMSE**—Root Mean Square Error:

$$MAPE(\%) = \frac{1}{N} \sum_{j=1}^N \frac{|y_j^M - y_j^P|}{y_j^M} \times 100 \quad RMSE = \left(\frac{1}{N} \sum_{j=1}^N (y_j^M - y_j^P)^2 \right)^{1/2}, \quad (6)$$

where y_j^M is the measured value of a time series, y_j^P is the predicted value, and N is the total number of predictions. The smaller the MAPE, the better the forecast. While there is a stochastic component to the GHI function, the near discontinuities that occur as a cloud moves between the sun and the array, and the ramping in power output that it occasions, make intra-hour prediction of GHI a challenging problem. As in weather forecasting, for short-term prediction, it is difficult to beat *persistence* which simply asserts that whatever the observed value is now, it will persist into the future. This is particularly true for those days that have clear skies with no clouds or completely overcast days: persistence always wins. As described earlier, days were classified as “Clear Sky”, “Overcast Sky”, and “Moderately Cloudy”, and the algorithm only used for the latter category. Persistence was used to forecast on a clear or overcast day (details were given in the Methods Section). If there are no clouds in the region of interest, the Haurwitz model [38,39] is used to calculate clear sky GHI, $G_{clr} = 1098 [\cos \theta_z \exp(-0.057 / \cos \theta_z)]$, where θ is the solar zenith angle. The straight persistence forecast $G(t + \Delta t) = G(t)$ is modified [17] with the equation below that accounts for small changes in θ over the time period Δt :

$$\hat{G}(t + \Delta t) = \frac{G_{clr}(t + \Delta t)}{G_{clr}(t)} \cdot G(t). \quad (7)$$

On moderately cloudy days, our approach assumes that if clouds are predicted to move into the area of the PV arrays, then GHI will be reduced from its current value. If ray-tracing indicates that the solar panels are shaded *at the site*, then GHI should drop dramatically and a ramp event occurs. In Weather Service parlance, this amounts to using a “point” forecast as opposed to an “area” one. In fact, GHI should (and in many cases does) drop to the level of the Diffuse HI. The problem is that the uncertainty in determining the exact position of shadows on the ground is very large. The code reduces the current GHI by a factor of α , where $0 < \alpha < 1$. A value $\alpha = 85\%$ was determined empirically based upon averaging the slopes of the GHI curve when a ramp occurs and the percentage of the GHI that the DNI and DHI constitute, respectively. While this should work for diffuse irradiation, DNI has a strong directional selectivity. Work currently in progress will improve this methodology by weighting heavily the pixels along a line from the center of an image to the sun, as was done in [14].

Table 4 gives a preliminary analysis of data collected at NREL in the fall of 2015, under a variety of cloud conditions from mostly clear sky to overcast as indicated by the Average Cloud Cover column. Two metrics, MAPE and RMSE, are given both for the camera 5-min ahead forecast and for persistence. They were computed for daytime hours, 8:00–16:00 MST, and measured and forecast values every minute were used so that $N = 481$. We note that on a clear day, such as 14 October, the errors are very small and identical for both methods. This is not surprising since, under clear skies, there are no clouds to track and the algorithm falls back on persistence. On 27 October, the conditions were “Moderately Cloudy” with **Avg CC** of 20.5% and **Max CC** of 98%. In this case, the cloud-tracking algorithm cuts in with a resulting MAPE that is slightly lower than persistence. Overall, the results look promising, but they present opportunities for further refinements in the algorithms to improve performance in the future.

Table 4. Error metrics for selected dates in late fall of 2015. **Avg CC** denotes the average cloud cover percentage computed from the images captured by the SkyImager on that day, while **Max CC** denotes the maximum in a single image. **Persist** denotes values computed using persistence, while **Camera** are values determined by the camera algorithm.

Date	Avg CC	Max CC	MAPE Persist	MAPE Camera	RMSE Persist	RMSE Camera
14 October	3.6%	12%	0.533%	0.533%	6.2 W/m ²	6.2 W/m ²
15 October	5.3%	27%	4.54%	5.32%	56 W/m ²	66 W/m ²
17 October	5.9%	42%	5.8%	6.04%	65 W/m ²	67 W/m ²
24 October	7.8%	70%	35.76%	35.66%	214 W/m ²	217 W/m ²
27 October	20.5%	98%	816.27%	793.05%	334 W/m ²	336 W/m ²
31 October	3.3%	20%	54.84%	54.84%	131 W/m ²	131 W/m ²
2 November	2.77%	20%	0.832%	0.832%	11.6 W/m ²	11.6 W/m ²
3 November	2.8%	14%	0.725%	0.725%	9 W/m ²	9 W/m ²
13 November	3.5%	41%	3.7%	3.7%	55 W/m ²	55 W/m ²
15 November	8.3%	92%	75.68%	69.62%	93 W/m ²	92 W/m ²
18 November	5.4%	53%	45.74%	43.48%	140 W/m ²	139 W/m ²
19 November	2.5%	17%	1.53%	1.43%	30 W/m ²	27 W/m ²

5. Lessons Learned and Future Activities

There are five distinct tasks involved in intra-hour solar forecasting for the Smart Grid: development of new hardware, writing software to perform image processing and machine learning, finding the most important variables and input data on which to base the forecasting algorithms, validating the method *offline* on large data sets that extend over several months, and finally implementing the system in real time in a distributed environment as part of an actual micro-grid. Our hope is that other researchers working in the area of solar forecasting will benefit from our experiences.

The work shows that SBC are not just for science fair projects, but can serve as powerful platforms for acquiring and processing data in a research environment. The graphics capabilities of the Raspberry Pi and its on-board camera are superb. Additional horsepower for processing the images in real time can be provided by linking with a second SBC, such as an Odroid C2, although this may be unnecessary as technology continues to improve (e.g., the RPi-3). By far, the most expensive part of the SkyImager was the security camera enclosure, but this approach was used to save time and effort. Using off-the-shelf components for enclosures, heater/fan assemblies, heat sinks, etc., as well as a Wi-Fi connection to the internet, might well bring the cost down below the \$100 mark. This would allow a large number of the devices to be employed in a distributed IoT environment.

We found that modern operating systems such as Raspbian Jessie and Ubuntu give the researcher the full power of software development in the Linux environment on a SBC. Python 2.7 proved to be a convenient, versatile framework not only for writing our programs, but for analyzing and displaying big data, via such packages as *numpy*, *scipy*, and *pandas*. Finally, the Open Computer Vision software *OpenCV 3.1.0* with Python bindings served as a robust development platform for high-level image

processing: reading and writing images to disk, standard image transformations, and calculating optical flow.

Forecasting the ramps in PV power production in the intra-hour time frame is a difficult problem. While there is a strong stochastic component, from the acquisition of the images to the final GHI and power forecasts, the problem is deterministic as well. Large drops in the DNI occur as cumulus clouds move between the sun and the solar arrays. We chose a physics-based approach that predicts GHI based upon the location of cumulus clouds minutes ahead. This task can be subdivided into two problems: (1) accurately predicting cloud movements and (2) predicting drops in DNI occasioned by shadows falling on the solar arrays. The Denver airport is 36 mi from Golden and frequently there are missing entries in the METAR observations, making the CBH value often a reasonable guess. Once the binary cloud images are projected to the CBH level, optical flow algorithms do an excellent job of predicting the apparent movement of intensity levels in the images. Moreover, one can choose between tracking features (Lucas–Kanade) or dense optical flow (Farneback) methods. Analyzing the errors in forecast versus actual cloud positions should be done separately from the GHI predictions.

A ray-tracing approach to locate shadows was chosen early in the design cycle, and while it did not perform better than persistence, we understand the reasons why. The sensitivity in this problem is large: mathematically, the problem is ill-posed and relatively small errors in the future cloud/shadow positions can result in large errors in forecast GHI. Moreover, the focus on low level cumulus clouds both ignores the fact that there can be multiple layers of different cloud types, and necessitates a binary cloud/no-cloud classification with its attendant loss of information. Specifically, converting to binary images before feature extraction and optical flow means that much of the rich information content contained in the original RGB image is lost. Our current research investigates retaining the grayscale images of the red-blue ratio throughout the entire process. More generally, a totally data-driven deep learning approach may prove superior. In the latter case, clouds would not be tracked at all, but, instead, the changing intensity patterns in the red and blue channels of the image sequence would be used with machine learning to forecast GHI. Physics would still be important. Changes in the pixels of the circumsolar region would figure prominently in a DNI calculation, while the remaining pixels would be used to extrapolate the DHI.

For task four, several months of images and observed GHI were collected at NREL. Offline versions of the codes can be used to analyze this data and develop new algorithms such as neural networks that require large computational resources in the training phase. The key to doing this effectively is to find sparse representations of the data-dimensionality reduction is essential. The final task is integration of the technology in a real-time setting, and we have completed the initial work as part of the INTEGRATE project, both at NREL and Ft. Sam Houston, San Antonio, Texas, USA.

Acknowledgments: This research was made possible by the Research Alliance between the San Antonio electric utility *CPS Energy* and the *Texas Sustainable Energy Research Institute* of The University of Texas at San Antonio. The authors thank the *National Renewable Energy Laboratory* for assistance in the deployment of the SkyImager prototype through the NREL INTEGRATE Project, the *Omnetric Group* for their help in integrating the SkyImager into their micro-grid management system, and the *Grid Modernization Group*, *CPS Energy* for their assistance at the Joint Base San Antonio deployment of the SkyImager.

Author Contributions: Rolando Vega conceived of a SkyImager using a single board computer and built the first prototype; Hariharan Krishnaswami provided expertise in PV systems, advanced grid functionalities, ramp forecasting and mentoring of several of his graduate students including one of the co-authors Michael Cervantes; Michael Cervantes wrote the Python code and helped to build and test the SkyImager; Walter Richardson, Jr. provided expertise in image processing and wrote the paper. All four authors collaborated for over a year on this project under the auspices of the Texas Sustainable Energy Research Institute.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

GHI	Global Horizontal Irradiance
DHI	Diffuse Horizontal Irradiance
DNI	Direct Normal Irradiance
NREL	National Renewable Energy Laboratory
CBH	Cloud Base Height
ASOS	Automated Surface Observing System
SBC	Single Board Computer
HDR	High Dynamic Range
MAPE	Mean Absolute Percentage Error
RMSE	Root Mean Square Error

References

1. Eto, J.; Thomas, R. *Computational Needs for the Next Generation Electric Grid Proceedings*; Department of Energy Report lbnl-5105e; Lawrence-Berkeley National Laboratory: Berkeley, CA, USA, 2011.
2. Glassley, W.; Kleissl, J.; van Dam, C.P.; Shiu, H.; Huang, J.; Braun, G.; Holland, R. *California Renewable Energy Forecasting, Resource Data and Mapping*; Appendix A; CAISO-California Independent Systems Operator: Folsom, CA, USA, 2010.
3. Nummikoski, J. Sky-Image Based Intra-Hour Solar Forecasting Using Independent Cloud-Motion Detection and Ray-Tracing Techniques for Cloud Shadow And Irradiance Estimation. Master's Thesis, UTSA, San Antonio, TX, USA, 2013.
4. Camargo, A. Introduction to UTSA-MS Smart Imager and Comparison with TSI Imager Technology. Master's Thesis, UTSA, San Antonio, TX, USA, 2015.
5. Guerrero-Lemus, R.; Martinez-Duart, J. *Renewable Energies and CO₂: Analysis Costs, Environmental Impacts and Technological Trends Lecture Notes in Energy*; Springer: Berlin, Germany, 2013; Volume 3.
6. Guerrero-Lemus, R.; Shephard, L. *Low Carbon Energy in Africa and Latin America: Renewable Technologies, Natural Gas and Nuclear Energy (Lecture Notes in Energy)*, 1st ed.; Springer: Berlin, Germany, 2017.
7. Canizares, C.; Olivares, D.; Mehrizi-Sani, A.; Etermadi, A.; Iravani, R.; Kazerani, M.; Hajimiragha, A.; Gomis-Bellmunt, O.; Saeedifard, M.; Jimenez-Estevéz, G.; et al. Trends in Microgrid Control. *IEEE Trans. Smart Grid* **2014**, *5*, 1905–1919.
8. Ela, E.; Diakov, V.; Ibanez, E.; Heaney, M. *Impacts of Variability and Uncertainty in Solar Photovoltaic Generation at Multiple Timescales*; Technical Report; National Renewable Energy Laboratory: Golden, CO, USA, 2013.
9. Bidram, A.; Davoudi, A.; Balog, R. Control and Circuit Techniques to Mitigate Partial Shading Effects in Photovoltaic Arrays. *IEEE J. Photovolt.* **2012**, *2*, 532–544.
10. Cervantes, M.; Krishnaswami, H.; Richardson, W.; Vega, R. Utilization of Low Cost, Sky-Imaging Technology for Irradiance Forecasting of Distributed Solar Generation. In Proceedings of the IEEE GreenTech Conference, Kansas City, MO, USA, April 2016.
11. Itseez. Open Source Computer Vision Library. Available online: <https://github.com/itseez/opencv> (accessed on 1 June 2015).
12. Itseez. The OpenCV Reference Manual, 2.4.9.0 Edition. 2014. Available online: <http://docs.opencv.org/> (accessed on 23 March 2017).
13. Marquez, R.; Coimbra, C. Proposed Metric for Evaluation of Solar Forecasting Models. *J. Sol. Energy Eng.* **2013**, *135*, 011016.
14. Marquez, R.; Coimbra, C. Intra-hour DNI forecasting based on cloud tracking image analysis. *Sol. Energy* **2013**, *91*, 327–336.
15. Urquhart, B.; Kurtz, B.; Dahlin, E.; Ghonima, M.; Shields, J.; Kleissl, J. Development of a sky imaging system for short-term solar power forecasting. *Atmos. Meas. Tech. Discuss.* **2014**, *7*, 4859–4907.
16. Gohari, S.; Urquhart, B.; Yang, H.; Kurtz, B.; Nguyen, D.; Chow, C.; Ghonima, M.; Kleissl, J. Comparison of solar power output forecasting performance of the Total Sky Imager and the University of California, San Diego Sky Imager. *Energy Procedia* **2014**, *49*, 2340–2350.
17. Chu, Y.; Pedro, H.; Coimbra, C. Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning. *Sol. Energy* **2013**, *98*, 592–603.

18. Yang, H.; Kurtz, B.; Nguyen, D.; Urquhart, B.; Chow, C.; Ghonima, M.; Kleissl, J. Solar irradiance forecasting using a ground-based sky imager developed at UC San Diego. *Sol. Energy* **2014**, *103*, 502–524.
19. Bernecker, D.; Riess, C.; Angelopoulou, E.; Horneegger, J. Continuous short-term irradiance forecasts using sky images. *Sol. Energy* **2014**, *110*, 303–315.
20. West, S.; Rowe, D.; Sayeef, S.; Berry, A. Short-term irradiance forecasting using skycams: Motivation and development. *Sol. Energy* **2014**, *110*, 188–207.
21. Wang, F.; Zhen, Z.; Mi, Z.; Sun, H.; Su, S.; Yang, G. Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting. *Energy Build.* **2014**, *86*, 427–438.
22. Zhu, T.; Wei, H.; Zhang, C.; Zhang, K.; Liu, T. A Local Threshold Algorithm for Cloud Detection on Ground-based Cloud Images. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28 July–30 July 2015.
23. Peng, Z.; Yoo, S.; Yu, D.; Huang, D. Solar Irradiance Forecast System Based on Geostationary Satellite. In Proceedings of the IEEE SmartGridComm 2013 Symposium, Vancouver, BC, Canada, 21–23 October 2013.
24. Mertens, T.; Kautz, J.; Van Reeth, F. Exposure fusion. In *Computer Graphics and Applications, Proceedings of the PG-07 15th Pacific IEEE Conference on Computer Graphics*; IEEE Computer Society: Washington, DC, USA, 2007; pp. 382–390.
25. Traonmilin, Y.; Aguerrebere, C. Simultaneous High Dynamic Range and Superresolution Imaging without Regularization. *SIAM J. Imaging Sci.* **2014**, *7*, 1624–1644.
26. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: New York, NY, USA, 2010.
27. Li, Q.; Lu, W.; Yang, J.; Wang, J. Thin Cloud Detection of All-Sky Images Using Markov Random Fields. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 1545–558.
28. Johnson, R.; Shields, J.; Hering, W. *Automated Visibility & Cloud Cover Measurements with a Solid-State Imaging System: Final Report*; Technical Report; UCSD Scripps Institute of Oceanography, Marine Physical Lab.: La Jolla, CA, USA, 1989.
29. Arias-Castro, E.; Donoho, D. Does median filtering truly preserve edges better than linear filtering? *Ann. Stat.* **2006**, *37*, 1172–2009.
30. Herzmann, D. Mesonet Website. Available online: <https://mesonet.agron.iastate.edu> (accessed on 23 March 2017).
31. Horn, B.K.P. *Robot Vision*; The MIT Press: Cambridge, MA, USA, 1991.
32. Wood-Bradley, P.; Zapata, J.; Pye, J. Cloud tracking with optical flow for short-term solar forecasting. In Proceedings of the 50th Annual Conference, Australian Solar Energy Society, Melbourne, Australia, 6–7 December 2012.
33. Lucas, B.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conferences on Artificial Intelligence*; IJCAI: Vancouver, BC, Canada, 1981; pp. 674–679.
34. Shi, J.; Tomasi, C. Good Features to Track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 1994; pp. 593–600.
35. Reda, I.; Andreas, A. *Solar Position Algorithm for Solar Radiation Applications*; National Renewable Energy Laboratory: Golden, CO, USA, 2003.
36. Voyant, C.; Soubdhan, T.; Lauret, P.; David, M.; Muselli, M. Statistical parameters as a means to a priori assess the accuracy of solar forecasting models. *Energy J.* **2015**, *90*, 671–679.
37. Stein, J.; Hansen, C.; Reno, M. *The Variability Index: A New and Novel Metric for Quantifying Irradiance and PV Output Variability*; Sandia National Laboratories: Albuquerque, NM, USA, 2012.
38. Reno, M.; Hansen, C.; Stein, J. *Global Horizontal Irradiance Clear Sky Models: Implementation and Analysis*; Sandia National Laboratories: Albuquerque, NM, USA, 2012.
39. Dazhi, Y.; Jirutitijaroen, P.; Walsh, W. The Estimation of Clear Sky Global Horizontal Irradiance at the Equator. *Energy Procedia* **2012**, *25*, 141–148.

