Intrahour Cloud Tracking Based on Optical Flow

Songjie Zhang[1], Zhekang Dong[1, 2], Xinyi Yang[3], Songjian Chai[2], Zhao Xu[2], Donglian Qi[1]

1. College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
E-mail: zsj1993@zju.edu.cn

2. Department of Electrical Engineering, The Hong Kong Polytechnic University Hung Hom, Hong Kong, China
E-mail: eezhaoxu@polyu.edu.hk

3. Department of Atmospheric Science, Zhejiang University, Hangzhou 310027, China
E-mail: 21638021@zju.edu.cn

Abstract: Drifting of the clouds may shade the sun which causes significant power output fluctuations in solar power systems. Forecasting the location of clouds could provide sufficient time to smoothen the power output. A method for identification and tracking cloud pixels from a commercial All-Sky camera is presented. Sky images taken every 5s were processed to determine the optimal threshold of different criterion to identify the six classes of clouds as well as clear-sky pixels. Cloud boundaries are tracked using Lucas-Kanade optical flow estimation method. The time when clouds will shade the sun can be forecasted 30s ahead with an acceptable accuracy. The research results demonstrate that the method presented has the potential for tracking different type of clouds with high variability.

Key Words: Forecasting, cloud tracking, solar energy, optical flow, cloud identification

1 Introduction

Solar power generation is considered an important part of the energy architecture, and it's the largest renewable energy resource available on our planet as well as the source of other resources like wind energy. But solar power is unreliable now due to the influence of stochastic weather conditions. As a matter of fact, it's important to smoothen the PV power output prior to its integration into the grid, for quality purposes [1].

Cloud occlusions are known to be a key contributor to the disturbance [2,3], severe swift power output oscillations are mostly due to sudden changes in solar irradiance caused by cloud occlusions [4]. The influence of clouds on solar irradiance depend on the cloud's volume, height and thickness [5]. Total sky imagers (TSI) and sky cameras equipped with a panoramic fisheye lens are the common equipment used in the forecasting of cloud occlusions.

A few papers have presented short-forecasting methods of cloud occlusions [6]. The work of [7] describes a new online forecasting approach of power production from PV systems, and the method is used to predict hourly values of solar power for horizons of up to 36 h. Jayadevan, Cronin, Brooks, Rodriguez, Koch, Leuthold, and Lonij extracted cloud motion vectors (CMV) from the images of a wide-angle camera mounted on a dual-axis sun-tracker, and utilized the block-motion estimation algorithm to determine the cloud velocity of each image [8]. With the block-motion estimate approach, the horizon of forecasting is up to 10 minutes ahead. Similarly, Chow, Urquhart, Lave, Dominguez, Kleissl, Shields, and Washom have determined cloud velocity and direction up to 5 minutes ahead through a cross-correlation method (CCM) applied to two consecutive images [9]. Later on, Barbieri F et al. have used a probability hypothesis density filter to forecast the time when cumulus clouds will shade the sun, and it achieved an absolute precision of 6 s ahead as well as an acceptable accuracy of 27 s ahead [10]. However, it could only forecast cumulus clouds, for other kinds of clouds, the results would dramatically change. In this work, the optimal threshold of each criteria for different clouds is calculated. Furthermore, based on the Lucas-Kanade optical flow algorithm, a novel cloud tracking method is investigated.

This paper is organized as follows. The obtainment and preprocessing of data are introduced in Section 2. Section 3 described the concept of cloud identification and implementation scheme in detail. Additionally, the Lucas-Kanade optical flow algorithm is presented in Section 4 to estimate the time when the cloud will shade the sun. Finally, we provide the conclusions of this research and suggest possible future investigations.

2 Data

2.1 All-sky camera and associated images

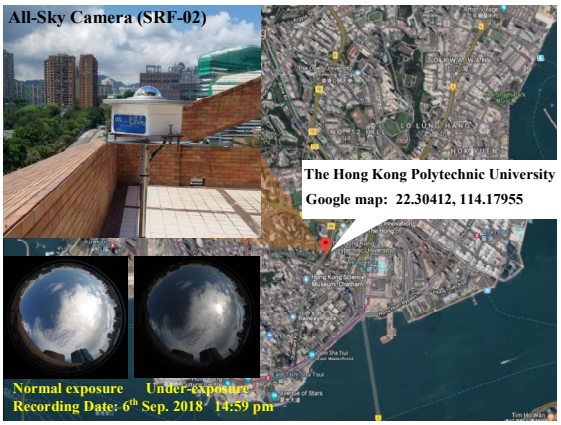


Fig. 1: The All-sky Camera (SRF-02)

The images used in this work were obtained by a commercial All-Sky camera (Model: SRF-02), operating on

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61571394, and the Hong Kong RGC Theme Based Research Scheme under Grants T23-407/13N and T23-701/14N.

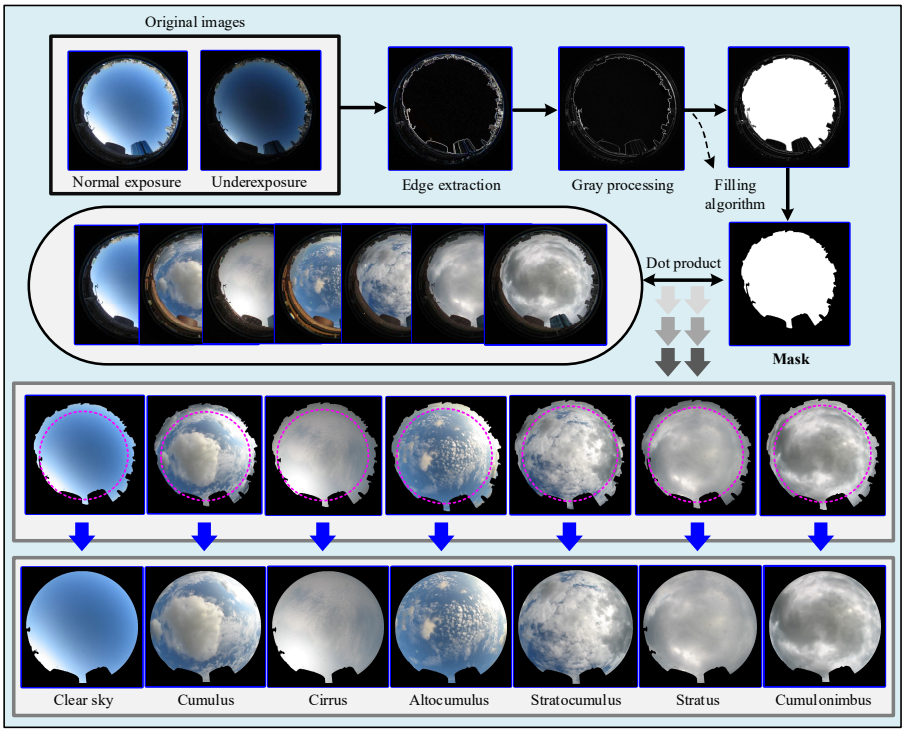


Fig. 2: The flow diagram of the image preprocessing

a flat roof of a laboratory of the Hong Kong Polytechnic University (Latitude: 22.30, Longitude: 114.18), as shown in Fig. 1.

From Fig. 1, the entire system is enclosed in a water- and weather-resistant box with a glass dome pointing to the zenith, which enables it to work during the daytime (6:00am~18:00pm) in any weather condition. Notably, a special full-frame fisheye lens installed in the front of the entrance optics of the camera is responsible for extending the field of view (FOV). The camera is programmed to capture a pair of images every 5 seconds. One is a normal-exposure image, the other is the associated under-exposure image with the same scene. All the images are stored in 24-bit (8 bit for each RGB channel) JPEG format with a maximum resolution of 1600×1200.

2.2 Image processing

In order to eliminate the undesired image information (like the building, roof, lightning rod, etc.), the image preprocessing is necessary. The corresponding flow diagram is demonstrated in Fig. 2.

In Fig. 2, the color edge image is firstly generated using the color image edge extraction algorithm. Then, the corresponding grayscale image can be obtained by the weighted average method. Next, a binary mask image is completed after executing the filling algorithm twice, so that the undesired image information can be removed after the dot product operation. Notably, since the location of the All-Sky camera is fixed and the boundary shapes of the surrounding buildings are relatively unchanged, the mask is a general one, which can be used for all the images captured from the camera. In addition, due to the edge distortion, the zenith angle is set to 70° (It is noted that zenith angle of 70° is recommended for best evaluation in all-sky images based applications). In other words, only the pixels at a FOV≤70° (inside the magenta circle) are included in this work, as shown in the bottom of Fig. 2.

3 Cloud Identification

In order to track the movement of cloud, we should first tell cloudy pixels apart from clear-sky pixels. Colors are usually used to identify the pixels. The color information is mainly divided into three channels, namely R (Red), G (Green) and B (Blue). And through these three channels, we could define a criterion to identify clouds and clear-sky pixels. The most basic criterion is ratio R/B as used by Kazantzidis et al. [11]. Heinle, Macke, and Srivastav used difference B-R, and it performed better with a sufficient set of images [12]. Besides, BRBG [13], BRBGB [14] and MCE [15] are common criterion too. This paper would compare the criterion above and figure out which performs better with our data.

The most difficult task for identification of the cloud is the classification of boundary pixels. Regardless of the method used, the recognition accuracy of cloud identification is error-prone. There are currently several algorithms for distinguishing between cloud pixels and clear sky pixels. Each method has its own optimal threshold value, and the selection of thresholds affects the accuracy of identification.

In this part, we obtained optimal threshold of each criterion for different clouds, and verify whether optimal threshold of each criterion is the same. In order to find the optimal threshold, a reference binary image is needed. With reference to the method in [10], we created a reference binary image of clear-sky (CS) with Photoshop. Meanwhile, the RGB layer of every image is used to obtain a reference binary image of the clear sky CL. Also, the solar disk has been repositioned in a convenient position without any visible sunrays caused by glaring effect.

Fig. 3 illustrates the deviations of results for different types of clouds. It can be observed that RB Ratio has better

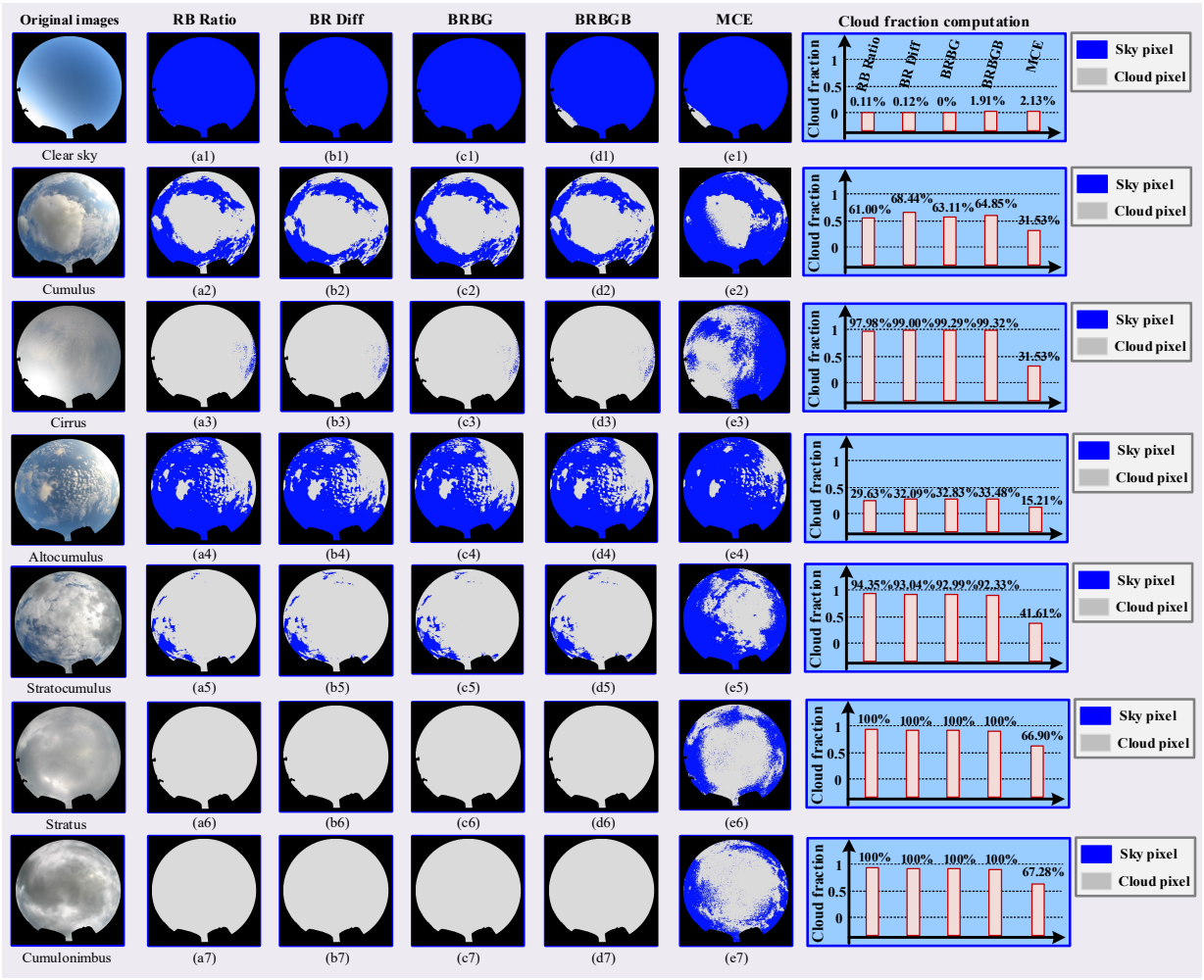


Fig. 3: Examples of the detection results

performance than other criterion for identification of Stratus, while BR Diff has better performance for identification of Cumulus. BRBGB performs better in the identification of Cirrus and MCE performs better in the identification of clear sky. Besides, the most discriminatory type of clouds is Cirrus. The optimal threshold values of different methods have been found as in **Table 1**. It's obvious that the optimal threshold value of each criteria varies significantly. The table also shows that the type of clouds affects the threshold value. So, we came to the conclusion that when comes to different clouds, the optimal threshold value varies. Thus, the cloud classification task is suggested to be executed before cloud pixel identification.

4 Cloud Tracking

After identifying clouds in Section 3, we get the binary image of original image. Now we will track the position and speed of clouds that we detected during Section 3.

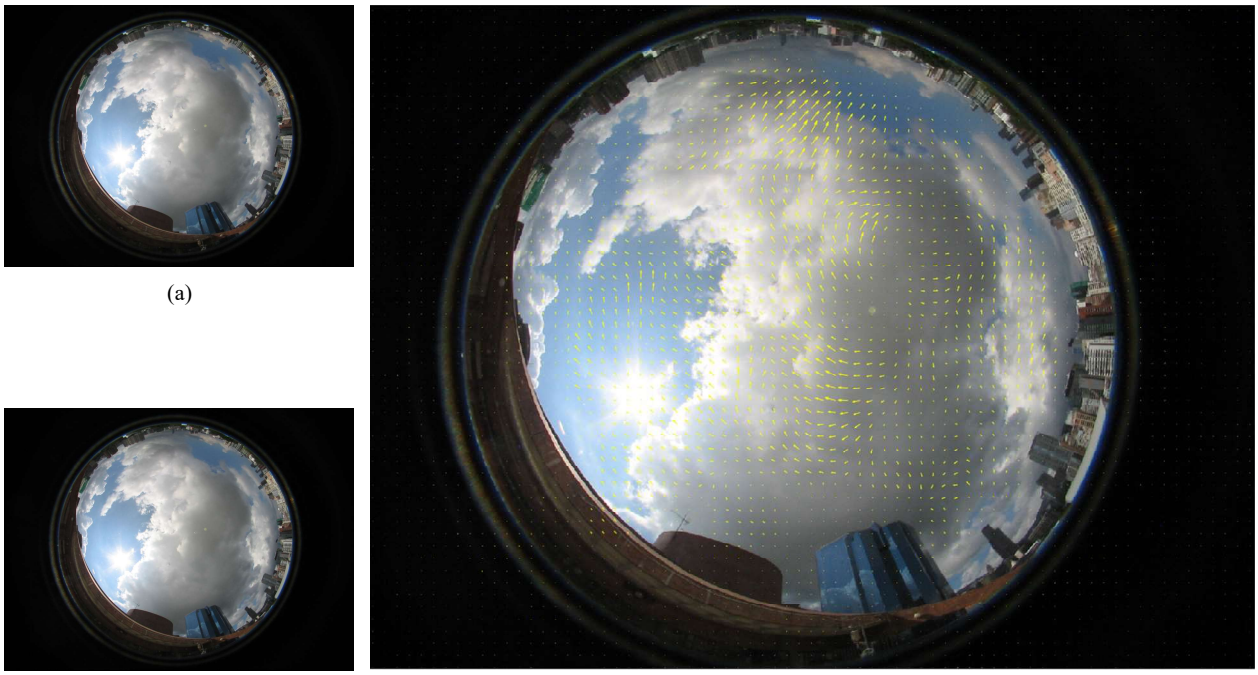
The simplified framework is shown in Fig. 4 where two continuous images captured at time $t=n$ and $t=n+1$ are provided in Fig. 4(a) and Fig. 4(b) respectively. Based on the image correlation information between the two continuous images, we would get the final image (as shown in Fig. 4(c)) with velocity vectors which could forecasting the time of cloud occlusions.

To realize this, we chose the optical flow algorithm to track the flow of cloud pixels, and we will describe the method in this work.

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image [16-20]. The concept of optical flow was first proposed by Gibson in [21]. When observing a moving object, the scene of the object forms a series of continuously changing images on the retina of the human eye. This series of continuously changing information constantly "flows through" the retina as if it were a "flow" of light, that's why it's called optical flow.

Table 1: The optimal threshold values for different methods

	RB Ratio	BR Diff	BRBG		BRBGB		
	Th1	Th2	Th3	Th4	Th5	Th6	Th7
Clear sky	0.84	28.05	1.16	1.14	27.00	7.00	50.00
Cumulus	0.86	30.60	1.18	1.18	26.80	8.00	52.00
Cirrus	0.85	26.14	1.24	1.24	27.30	5.20	51.00
Altocumulus	0.87	25.86	1.17	1.19	27.40	7.00	51.00
Stratocumulus	0.62	45.9	1.32	1.36	30.20	18.00	55.00
Stratus	0.1~0.9	10~50	0.5~2	0.5~2	24~40	5~20	40~56
Cumulonimbus	0.2~0.8	10~50	0.2~5	0.2~5	22~48	3~20	40~58



(a) (b) (c)

Fig. 4: (a) Original image at time $t=n$; (b) Original image at time $t=n+1$; (c) Processed image with velocity vectors

Generally speaking, the optical flow field calculates the moving speed and moving direction of each pixel in each image through a sequence of pictures. For example, when the position of one point in the t-frame is (x, y) and the position in the $(t+1)$-frame is $(x+u, y+v)$, then we can determine the movement of the point is (u, v). And we could estimate its distance from us by the displacement of the point, since objects in the distance move a lot slower than objects in the near.

In order to calculate the position of the point at the $(t+1)$ frame, we should track the pixels through the vision feature. There are many methods of optical flow estimation, the Lucas-Kanade algorithm works best with low displacement of pixels between subsequent images [22, 23].

The LK optical flow method is a gradient-based local parameterized optical flow estimation method. It assumes that the optical flow vector is constant in a spatially-sized neighborhood, and then estimates the optical flow using a weighted least squares method. It calculates the movement of each pixel position between two frames at time t to $t+\Delta t$. Since it is based on the Taylor series of image signals, this method is called differential, which is the use of partial derivatives for spatial and temporal coordinates.

The Lucas-Kanade algorithm is based on three assumptions:

Assumption 1: The brightness is constant. In order to track the object, the gray value of the object should be constant within a short time period. Consider $I(x, y, t)$ and $I(x+dx, y+dy, t+dt)$ are the same pixels in two continuous images, the image constraint equation defined as follows:

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \tag{1}$$

where x and y are the position of the pixel in the image, t denotes the time sequence of the image.

Assumption 2: Time continuous or the displacement is small. In this way, the gray scale can be used to guide the position. Without this assumption, Lucas-Kanade optical flow algorithm could not achieve. And using the Taylor series for the image constraint equation, we can get:

$$I(x+dx, y+dy, t+dt) = I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + H.O.T \tag{2}$$

where $H.O.T$ represents second or higher order. Since the displacement is small, $H.O.T$ could be omitted as follows:

$$I(x+dx, y+dy, t+dt) \approx I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \tag{3}$$

The expression could be abbreviated as follows:

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} \tag{4}$$

Let $I_t=\partial I/\partial t$, $I_x=\partial I/\partial x$, $I_y=\partial I/\partial y$, $V_x=dx/dt$, and $V_y=dy/dt$, Eq. (4) could be rewritten in the form of a matrix as follows:

$$[I_x \quad I_y]\begin{bmatrix} V_x \\ V_y \end{bmatrix} = -I_t \tag{5}$$

Assumption 3: The optical flow in the neighborhood is consistent. Eq. (5) has two unknowns V_x and V_y , so it can't be solved. In order to solve the equation, the third assumption was introduced. Assuming that the optical flow (V_x, V_y) is a constant in a small window of size $m*m$ ($m>1$). Notably, the value of m is set to 45 in this work. Then the following set can be obtained from pixels 1...n:

$$\begin{aligned} I_{x1}V_x + I_{y1}V_y &= -I_{t1} \\ I_{x2}V_x + I_{y2}V_y &= -I_{t2} \\ &\vdots \\ I_{xn}V_x + I_{yn}V_y &= -I_{tn} \end{aligned} \tag{6}$$

where $n=m^2$. For the above set, there are two unknowns while there are more than two equations, which means the equation set is overdetermined. In other words, there is redundancy in the equations.

5 Results

It is noted that the experiments are conducted on a desk-top workstation with Core i7-6700 processor, 16GB DDR4 RAM and Windows 10 OS. Software platforms utilized in this work mainly refer to the Matlab R2018b. And the corresponding experimental results are collected in **Table 2.**

Table 2 Comparison of forecast and actual time

Dataset	Forecast time	Actual time	Error (%)
One	30	35	14.3
Two	30	40	25.0
Three	30	30	0.0
Four	30	25	20.0
Five	30	35	14.3
Six	30	40	25.0

Specifically, **Table 2** shows the results of the forecasting time against the actual time taken for the cloud occlusions as well as the errors of forecasting. We executed the optical flow algorithm on six datasets. Since the camera is programmed to capture images every 5 seconds, the precision of actual time is in units of 5 seconds. For the given six datasets (i.e., Dataset One ~ Dataset Six), the prediction error is less than 25%. The average error of six datasets is 16.4%, which is acceptable under actual circumstances. Also, it has been successfully tested that all the datasets could be effectively processed by optical flow algorithm within a short time period. The error of prediction is not only affected by the type of cloud, but also by the current wind speed. And the error should raise as the forecast time ahead of collision is increased. Thus, the results may varies significantly with different datasets.

6 Conclusion

This paper presented a method for cloud tracking and forecasting utilizing optical flow. Due to the changing shape of clouds, the forecasting of trajectory is difficult to achieve. As long as every pixel is identified to track, the algorithm will be able to predict the trend of cloud shape. Thus, it accurately tracked the movement of the clouds and predicted the trajectory within an acceptable period of time. The algorithm forecasts time of cloud occlusions with 5.3-21.2% error depending on the length of time of the forecast. The longer the forecasting time, the greater the error.

The optimal threshold values of different clouds have been found varies significantly. It means we should preset different threshold value when facing different clouds to achieve the optimal performance.

There have been multiple issues identified that can be addressed with further work. The classification of the cloud is an important issue that may affects the result of cloud identification. And more data would be processed to evaluate the accuracy of the algorithm in the further work.

References

[1] Morjaria M, Anichkov D, Chadliev V, et al. A grid-friendly plant: The role of utility-scale photovoltaic plants in grid stability and reliability. IEEE Power and Energy Magazine, 2014, 12(3): 87-95.

[2] Pfister G, McKenzie R L, Liley J B, et al. Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. Journal of Applied Meteorology, 2003, 42(10): 1421-1434.

[3] Kreuter A, Zangerl M, Schwarzmann M, et al. All-sky imaging: a simple, versatile system for atmospheric research. Applied optics, 2009, 48(6): 1091-1097.

[4] Barnes A K, Balda J C, Escobar-Mejía A, et al. Placement of energy storage coordinated with smart PV inverters. Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES. IEEE, 2012: 1-7.

[5] Heinle A, Macke A, Srivastav A. Automatic cloud classification of whole sky images. Atmospheric Measurement Techniques, 2010, 3(3): 557-567.

[6] Chaturvedi D K, Isha I. Solar Power Forecasting: A Review. International Journal of Computer Applications, 2016, 145(6): 0975-8887.

[7] Bacher P, Madsen H, Nielsen H A. Online short-term solar power forecasting. Solar Energy, 2009, 83(10): 1772-1783.

[8] Jayadevan V T, Rodriguez J J, Lonij V P A, et al. Forecasting solar power intermittency using ground-based cloud imaging. American Solar Energy Society Meeting, Denver, CO, May. 2012: 13-17.

[9] Chow C W, Urquhart B, Lave M, et al. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Solar Energy, 2011, 85(11): 2881-2893.

[10] Barbieri F, Rifflart C, Vo B T, et al. Intrahour Cloud Tracking Based on Probability Hypothesis Density Filtering. IEEE Transactions on Sustainable Energy, 2018, 9(1): 340-349.

[11] Long C N, Sabburg J M, Calbó J, et al. Retrieving cloud characteristics from ground-based daytime color all-sky images. Journal of Atmospheric and Oceanic Technology, 2006, 23(5): 633-652.

[12] Kazantzidis A, Tzoumanikas P, Bais A F, et al. Cloud detection and classification with the use of whole-sky ground-based images. Atmospheric Research, 2012, 113: 80-88.

[13] Antonanzas J, Osorio N, Escobar R, et al. Review of photovoltaic power forecasting[J]. Solar Energy, 2016, 136: 78-111.

[14] Barbieri F, Rajakaruna S, Ghosh A. Very short-term photovoltaic power forecasting with cloud modeling: A review. Renewable and Sustainable Energy Reviews, 2017, 75: 242-263.

[15] Li Q, Lu W, Yang J. A hybrid thresholding algorithm for cloud detection on ground-based color images. Journal of atmospheric and oceanic technology, 2011, 28(10): 1286-1296.

[16] Horn B K P, Schunck B G. Determining optical flow. Artificial intelligence, 1981, 17(1-3): 185-203.

[17] Sun D, Yang X, Liu M Y, et al. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8934-8943.

[18] Sevilla-Lara L, Liao Y, Güney F, et al. On the integration of optical flow and action recognition. German Conference on Pattern Recognition. Springer, Cham, 2018: 281-297.

[19] Hui T W, Tang X, Change Loy C. Liteflownet: A lightweight convolutional neural network for optical flow estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8981-8989.

[20] Wang Y, Yang Y, Yang Z, et al. Occlusion aware unsupervised learning of optical flow. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4884-4893.

[21] Gibson J J. The perception of the visual world. 1950.

[22] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision. 1981.

[23] Wang Z, Yang X. Moving Target Detection and Tracking Based on Pyramid Lucas-Kanade Optical Flow[C]//2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2018: 66-69.