Cloud tracking with optical flow for short-term solar forecasting

Philip Wood-Bradley, **José Zapata, John Pye**

Solar Thermal Group, Australian National University, Canberra, Australia

Corresponding author: John Pye – john.pye@anu.edu.au

1. Abstract

A method for tracking and predicting cloud movement using ground based sky imagery is presented. Sequences of partial sky images, with each image taken one second apart with a size of 640 by 480 pixels, were processed to determine the time taken for clouds to reach a user defined region in the image or the Sun. The clouds were first identified by segmenting the image based on the difference between the blue and red colour channels, producing a binary detection image. Good features to track were then located in the image and tracked utilising the Lucas-Kanade method for optical flow. From the trajectory of the tracked features and the binary detection image, cloud signals were generated. The trajectory of the individual features were used to determine the risky cloud signals (signals that pass over the user defined region or Sun). Time to collision estimates were produced based on merging these risky cloud signals. Estimates of times up to 40 seconds were achieved with error in the estimate increasing when the estimated time is larger. The method presented has the potential for tracking clouds travelling in different directions and at different velocities.

Keywords: forecasting, cloud, solar resource, cloud tracking, transients, solar-thermal energy, ANU.

2. Introduction

Clouds introduce the greatest uncertainties in predicting solar irradiance. They scatter light, greatly reducing direct normal irradiance (Cazorla, et al, 2008). The ability to predict the movement of clouds to forecast the solar irradiance is essential for the effective operation of many solar applications such as solar thermal systems, photovoltaic systems, and grid regulation (Mathiesen & Kleissl, 2011; Martínez López, et al, 2002).

A method for short term solar and cloud forecasting has been presented by the University of California, San Diego, USA (UCSD). The team computed the intra-hour forecast using whole sky images from a ground based whole sky camera[1]. The method assumes that cloud movement is homogeneous and produces a vector describing the movement (Chow, et al, 2011). The analysis of clouds through ground-based sky imagery has been investigated for many uses. Some of these uses are, classification of cloud types (Kazantzidis, et al, 2012), cloud cover estimation (Cazorla, et al, 2008), cloud detection (Li, et al, 2012), identifying cloud characteristics (Long, et al, 2006), feature extraction (Calbó & Sabburg, 2008), solar irradiance measuring (Tomson, 2010), and short term solar forecasting (Chow, et al, 2011). For the real world uses of these applications it is beneficial for large areas of sky, if not the whole sky, to be captured.

The method presented in this paper is a technique for tracking and predicting cloud movement using ground-based sky imagery and optical flow[2] for the use in short term solar forecasting. The method has the potential to provide added functionality in terms of multi-directional and velocity tracking. There is also the desire to build local expertise in this needed area.

3. Method

3.1. Experimental Setup

A laptop webcam, specifically a *Lenovo Easycamera*, was used as the ground-based sky camera. The images acquired were partial sky images with a field of view of approximately 60 degrees in both the horizontal and vertical directions. Partial sky images were used due to the ease of implementation and affordability. Full-sky imaging is intended for the future, but for the purpose of algorithm development only a narrow field of view was necessary. The images had a resolution of 640 by 480 pixels and were stored in 24 bit JPEG format.

---

1 A camera based on the ground that has a field of view of 180 degree by 360 degree such that the whole sky can be captured.

2 Optical flow is the measurement of the movement of an object in an image relative to the image.

The images were processed using the programming language *Python*, incorporating the *OpenCV*[3] library and the *NumPy* scientific computing package. *Python* is advantageous due to its functionality with images and matrices.

3.2. Cloud Detection

The clouds were distinguished from the sky by analysing the difference between the blue and red colour channels of each pixel in the sequence of images. This characteristic was chosen due to its ease of implementation and relative accuracy for circumsolar regions. The difference for each pixel was compared to a threshold to segment the images. The threshold was determined by the user based on the condition of the sky in the sequence of images being analysed. Pixels with difference values above the threshold were set to zero and below to 255 representing sky and cloud respectively. The value of 255 corresponds to maximum intensity with respect to images, resulting in a white pixel. The resultant binary detection image, Figure 1, is utilised in forecasting the cloud movement.

Figure 1. Acquired image (Left); binary detection image (Right)

3.3. Tracking Cloud Movement

Cloud tracking was the second task, undertaken after cloud detection. The movement of the clouds through multiple subsequent images must be tracked. To do this optical flow was implemented. This is due to the characteristics of optical flow being well suited to cloud movement. The Lucas-Kanade algorithm for optical flow works best with low displacement of pixels between subsequent images (Lucas & Kanade, 1981). This is ideal as individual clouds tend to move in a constant direction and at a constant speed. The sequences of images were taken with a frequency of one frame a second to allow for smooth cloud movement in the images.

The first step when tracking clouds is to identify features in an image that have characteristics ideal for tracking. These features are located on high gradients of colours or intensities of the pixels and are usually located on the corners of objects in an image. The most prominent features on sky images are the corners located on the edges between the cloud and the sky. These corners are the first to be identified by the function. This is useful due to the edges of the clouds being of most importance in terms of solar forecasting. To further emphasise the edges of interest, the single channel image of the difference between the blue and red colour channels (BR Difference Image) of the image was used, shown in Figure 2. Using this image as the input reduces features being found inside clouds. A function from the *OpenCV* library, g*oodFeaturesToTrack(),* was used to locate suitable features to track by analysing the corner quality at each pixel in the image. It returned the locations of the features with the most prominent corners (OpenCV, 2012) (Shi & Tomasi, 1994).

3 OpenCV is an open-source software library containing specialised functions for computer vision applications.

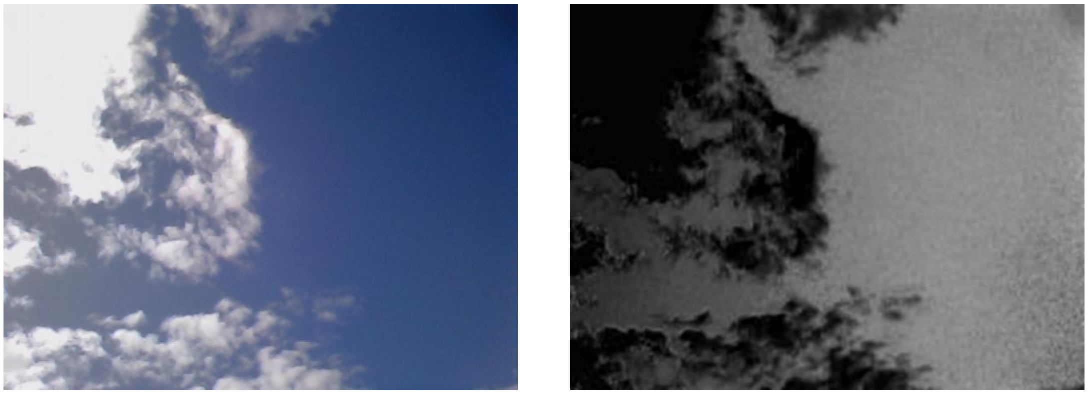

Figure 2. Original image (Left); B-R difference image scaled to 255 (right)

The features identified previously are used as inputs into the optical flow tracking algorithm. Optical flow is the measurement of the motion of objects in an image, or image velocity, relative to the observer (Barron et al, 1994; Lucas & Kanade, 1981). Optical flow approximates the motion field in an image from the patterns in the intensity of the image. The features identified in the image are matched against others in the subsequent image from the sequence of images. The highest probable match is returned and used in calculating the motion. The BR difference image was also used as the input to the optical flow algorithm.

The Lucas-Kanade method assumes that there is even local flow around the neighbourhood of pixels around the feature and that the displacement of the feature is small (Lucas & Kanade, 1981). The method weights the pixels more heavily that are closer to the centre pixel of the neighbourhood of pixels (Barron et al., 1994).

To implement the optical flow the *OpenCV* function *calcOpticalFlowPyrLK()* was utilised. The function finds the location of the feature being tracked in the next image and stores the previous location. A line was drawn between each of the previous feature locations to depict this and can be seen in Figure 3. The features located in the sky on the right hand side of the left image have occurred due to noise and are obviously not suitable features to track. These features were removed manually.

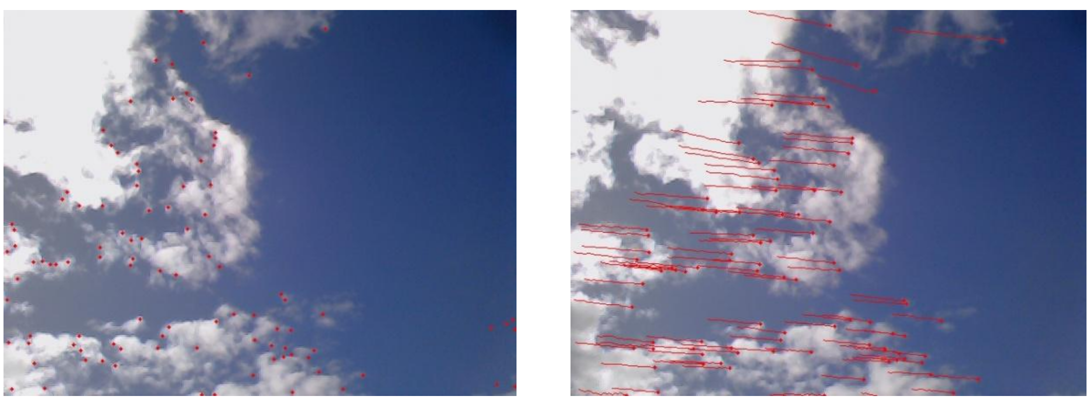

Figure 3. Original feature locations (left); Line drawn between previous feature locations (right).

3.4. Forecasting Cloud Movement

To forecast the movement, the linear regression line through all the previous locations of the each feature was calculated. The velocity of each feature was found by translating the initial and final feature location onto the regression line and dividing the distance that the feature had travelled along the line by the number of images that have been processed (the number of previous locations stored). This velocity had units of pixels per second due to the images being acquired at one frame per second. An important factor is that the velocity it calculated for each feature. This means that clouds moving in multiple directions and at different speeds can be forecast.

The binary detection image generated in the cloud detection phase was then incorporated. Each point along the regression line of the feature was given a value corresponding to the same point in the binary detection image. This is shown in Figure 4, with values of 255 relating to cloud and zero to sky.

The regression lines that intercepted the Sun or a user defined region, as shown in Figure 5, were classified as having the highest risk. The corresponding cloud signals were merged and the times taken for the clouds to reach the regions were calculated based on the pixel displacement and the velocity derived earlier, shown in Figure 6.

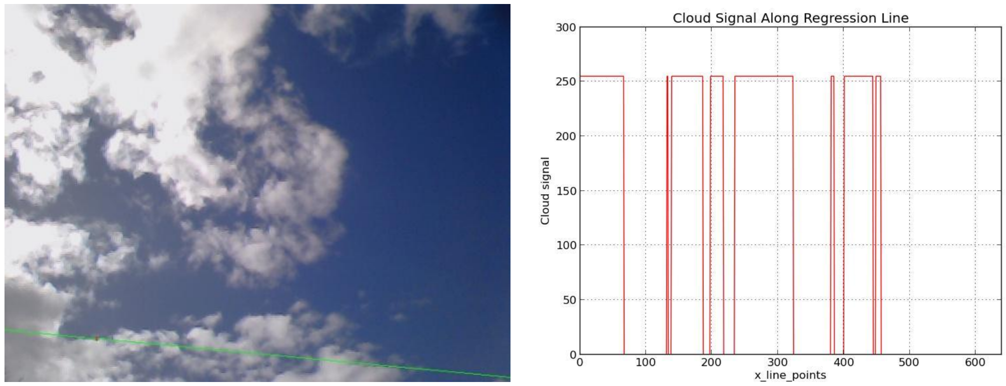

Figure 4. Regression line of a single feature (left) and corresponding cloud signal (right)

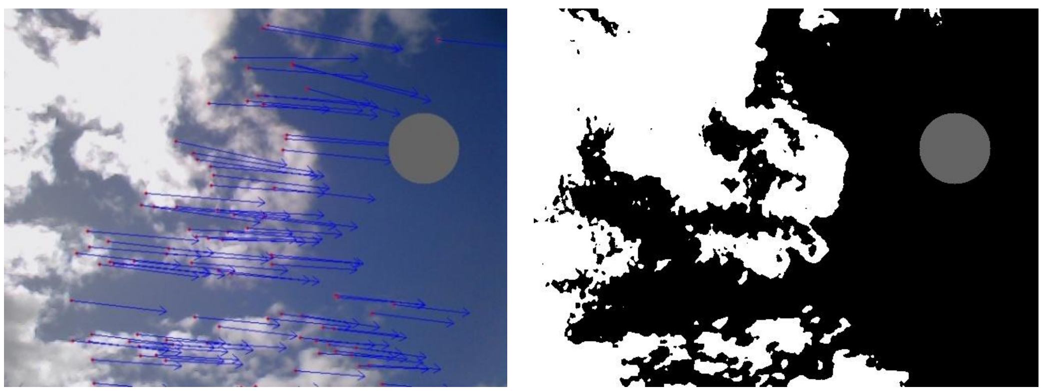

Figure 5. Image with velocity vectors and user defined region (left); detection image with user-defined region (right);

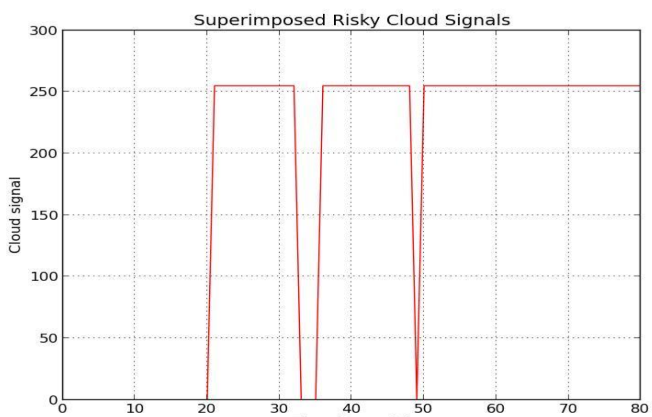

Figure 6. Superimposed risky cloud signals graph, with time to collision estimates

4. Results

Table 1 shows the results of the forecast time against the actual time taken for the clouds to collide with the user defined region. Dataset One had a smoother cloud velocity and a larger set of images, which increased the amount of previous points in the tracked features. This meant that the trajectory lines of the features were resistant to change causing a consistent output resulting in smaller errors associated with data set one. Dataset Two had fast clouds, meaning the forecast time for 40 and 30 seconds could not be recorded due to the features not being tracked long enough to add stability.

For Dataset One the 20 second forecast was one second off with 5.3% error. The 30 second forecast was 3 seconds off resulting in 11.1% error. The error should reduce as the forecast time ahead of collision is decreased. More data must be processed to evaluate the error in the algorithm further. The UCSD team obtained errors of around 6% for 30 second forecasts increasing to 23-30% for five minute forecasts (Chow, et al., 2011). Due to the limitations for the camera and setup, forecasts of this size were unachievable for this paper.

Table 1. Comparison of forecast and actual time to shading of sun by cloud, using the optical flow model.

Dataset	Forecast time (seconds)	Actual time (seconds)	Error (%)
One	40	33	21.2%
Two	40	-	-
One	30	27	11.1%
Two	30	-	-
One	20	19	5.3%
Two	20	23	13.0%
One	10	12	16.7%
Two	10	12	16.7%

5. Conclusions and Further Work

5.1. Conclusions

This paper presented a method for cloud tracking and forecasting utilising optical flow. It accurately tracked the movement of the clouds and predicted the trajectory. The use of the Lucas-Kanade method for optical flow is well suited to cloud tracking as the movement of clouds is small between images. Due to multiple features being tracked, the algorithm could work on clouds that have different velocities or that are moving in different directions. As long as there are enough features identified to track, the algorithm will be able to identify the risky clouds. The algorithm produced time to collision estimates with 5.3-21.2% error depending on the length of time of the forecast. The longer the forecast a larger error is present.

The algorithm used to identify the features performed exceptionally. It found features predominantly along the cloud to sky edge improving the tracking accuracy. Utilising the BR difference image emphasised the edges of interests reducing the number of stray points.

5.2. Further Work

There have been multiple issues identified that can be addressed with further work. Firstly the Sun is a major component of the project and is causing issues with detection. Two ways that could deal with the Sun are undertaking a computer vision solution or a hardware solution. A computer vision solution could consist of classifying the Sun and segmenting the detection image into three components, sky, cloud and Sun. A hardware solution could be blocking the sun from view by placing an obstructing disc in-between the sun and the camera reducing its effects on the image (Cazorla, et al., 2008).

Secondly, bad features are identified and more features need to be identified as the algorithm runs. To fix this issue the bad features must be identified and removed without disrupting any of the good features. Additionally new features must be able to be found as new clouds enter the field of view. A possible approach to this issue would be to remove features that do not have a velocity or features that are moving erratically. New features could be found depending on the time. For instance, every 20 seconds find more good features to track. An approach to reduce the identification of bad features could be to use the binary detection image as a mask when searching for the features. The search would be conducted around the edges of clouds within a margin such that central cloud or sky features are not identified.

Testing is required for clouds moving in different directions and/or at different velocities. The algorithm has the potential for dealing with these clouds due to the tracking method. Investigation into the rate of expansion or disappearing of clouds could be instigated by using optical flow. This could add accuracy to the solar forecasting. In saying this, more testing is required in general, with more varying data sets. A camera system would assist with acquiring data sets. The best camera system would take whole sky imagery allowing for large forecasts.

Another area that could be investigated is identifying different levels of risky cloud signals. Features that have a trajectory through different solar and circumsolar regions could have different risks associated with them. This would allow for more informed decisions, based off the solar forecast, to be made.

Currently the analysis of cloud tracking and forecasting takes time depending on the number of features identified. With 1-10 features being tracked the processing time is small, at around 5 images per second. When over 100 features are being tracked the processing time is approximately one image per second meaning that real time forecasts is currently not achievable with the computer used. Optimisation of the algorithms is required to reduce the processing time.

Other stages of further work involve incorporating real-time solar data and imagery allowing for accurate solar forecasts. This is an exciting goal for the future of this project.

Bibliography

Feature detection – OpenCV v2.4.3 documentation (03 Nov 2012). Retrieved 4 Dec 2012 from http://docs.opencv.org/modules/imgproc/doc/feature_detection.html

Barron, J. L., Fleet, D. J., & Beauchemin, S. (1994). Performance of Optical Flow Techniques. *International Journal of Computer Vision*.

Calbó, J., & Sabburg, J. (2008). Feature Extraction from Whole-Sky Ground-Based Images for Cloud-Type Recognition. *Journal of Atmospheric and Oceanic Technology, 25*(1), 3-9, 11-14.

Cazorla, A., Olmo, F. J., & Alados-Arboledas, L. (2008, January). Development of a Sky Imager for Cloud Cover Assessment. *Optical Society of America, 25*(1), 29-39.

Chow, C., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., et al. (2011). Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Solar Energy, 85*, 2881-2893.

Kazantzidis, A., Tzoumanikas, P., Bais, A., Fotopoulos, S., & Economou, G. (2012). Cloud detection and classification with the use of whole-sky ground-based images. *Atmospheric Research*, 80-88.

Li, Q., Lu, W., Yang, J., & Wang, J. (2012). Thin Cloud Detection of All-Sky Images Using Markov Random Fields. *IEEE Geoscience and Remote Sensing Letters, 9*(3), 417-421.

Long, C. N., Sabburg, J. M., Calbó, J., & Pagès, D. (2006). Retrieving Cloud Characteristics from Ground-Based Daytime Color All-Sky Images. *Journal of Atmospheric and Oceanic Technology, 23*(5), 633-652.

Lucas, B., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereovision. *Imaging Understanding Workshop, pp. 121-130.* Pittsburgh.

Martínez López, M., Vargas, M., & Rubio, F. R. (2002). *Vision-Based System for the Safe Operation of a Solar Power Tower Plant.* Berlin: Springer-Verlag.

Mathiesen, P., & Kleissl, J. (2011). Evaluation of numerical weather prediction for intra-day solar forecasting in the continental United States. *Solar Energy, 85*, 967-977.

Shi, J., & Tomasi, C. (1994). Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition.* Seattle.

Tomson, T. (2010). Fast Dynamic Processes of Solar Radiation. *Solar Energy, 84*, 318-323.