SHORT TERM SOLAR IRRADIANCE PREDICTION MODEL USING GROUND BASED SKY IMAGING

Kaumadee Shashikala Samarakoon

(139518U)

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Electrical Engineering

Department of Electrical Engineering

University of Moratuwa
Sri Lanka

March 2017

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: .... Date: 17/03/2017

(K.S.Samarakoon)

The above candidate has carried out research for the Masters Dissertation under my supervision.

UOM Verified Signature

Signature of the supervisc.. ....ate 17/03/2017

(Dr. Asanka Rodrigo)

ACKNOWLEDGEMENT

First and foremost, I would like to pay my sincere gratitude to my project supervisor Dr. Asanka Rodrigo for his continuous guidance and support throughout the research to develop the result in to a fruitful outcome.

Secondly I would like to thank the postgraduate research coordinator of department, Dr. Upuli Jayatunga for the supervision and management of research evaluation.

I would like to a pay my sincerest appreciation towards the academic staff of Department of Electrical Engineering for their valuable feedbacks and constructive comments during the progress reviews.

I pay my special gratitude to the final year project group 16 of batch '10 and project group 06 of batch '11 for their continuous support and coordination throughout the research.

Finally, I would like to thank the non-academic staff of Department of Electrical Engineering for their support during the whole post graduate course to make this endeavor a success.

ABSTRACT

With the growing energy demands and depleting fossil fuel deposits world is moving towards renewable energy. Solar energy has become a focus on this transition, especially in tropical countries where solar power harvesting is economical. The major disadvantage of solar power is its intermittent behavior due to cloud movement. Several researches are ongoing to predict the solar power in different temporal and spatial resolutions as a solution to this drawback.

This research discusses on a short term solar prediction model using total sky images captured with a specifically developed hardware. A MATLAB software model is developed using image processing techniques to track and predict cloud movement throughout the visible span. The cloud movement behavior is mapped with the output current of a fixed solar panel to develop the prediction algorithm. The developed software model is capable of predicting the solar panel output on real time total sky images with a 10 second temporal resolution. Additionally, the model parameters can be fine-tuned to predict the solar irradiance for a different location.

Key words: solar prediction, image processing, renewable energy, solar power, MATLAB

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
ARMA	Auto Regressive Moving Average
CEB	Ceylon Electricity Board
FL	Fuzzy Logic
GHI	Global Horizontal Irradiation
IPP	Independent Power Producer
MOS	Model Output Static
NDFD	National Digital Forecast Database
NREL	National Renewable Energy Laboratory
NWP	Numerical Weather Prediction
PV	Photo Voltaic
RGB	Red Green Blue
RMSE	Root Mean Square Error
SLSEA	Sri Lanka Sustainable Energy Authority
SPA	Solar Position Algorithm
TDNN	Time Delay Neural Network
TSI	Total Sky Imager
USB	Universal Serial Bus
UTC	Universal Time Coordinated
YES	Yankee Environmental Systems

CHAPTER 01

INTRODUCTION

The world energy demand is growing continuously with the population growth and technology advancements. Renewable energy has become the global trend with the fossil fuel deposit depletion around the world and environmental constraints like global warming. Solar power has become a leading trend in the transition to renewable energy resources.

1.1 Sri Lankan Energy Context

Sri Lankan electricity generation is mainly based on thermal and hydro resources. After full utilization of all the possible hydro sites, Sri Lanka had to go for other renewable energy resources to cater the expanding demands. Increase of the utilization of thermal power has economic constraints since Sri Lanka does not have fossil fuel deposits and cost of fuel import is higher. Also thermal power has environmental constraints which add on to environmental pollution.

In the movement of CEB (Ceylon Electricity Board) electricity generation from conventional to non-conventional energy, solar power is considered as one of options to reach for as a tropical country.

National energy policy has set a target to obtain 20% of the total electricity generation in country from renewable sources by 2020. In order to achieve the target several encouraging actions are taken like giving permits to Independent Power Producers (IPP), net metering facility, etc.

1.2 Solar Energy in Sri Lanka

Sri Lanka has an average solar capacity of 4.0 - 4.5 kWh/m^2/day in flat zones while the high plains are also rich with an average of 2.5 - 3.5 kWh/m^2/day.

Sri Lanka Sustainable Energy Authority (SLSEA) has installed the first solar park in Sri Lanka at Hambantota with a capacity of 1237 kW. Also CEB has introduced net metering facility in 2007 to integrate more renewable energy to Sri Lankan power grid. The number of roof mounted solar panels is increasing with the net metering concept introduced to Sri Lankan electricity consumers.

1.3 Research Motivation

Solar energy is a freely available variable source of energy. When integrated in to an electricity grid, this intermittent behaviour negatively affects the system stability. Due to this constraint the integrated percentage of solar energy is also limited.

If an accurate prediction mechanism can be developed, solar energy integration percentage can be increased and it will be an economical and environmental saving.

1.4 Project Objectives

Project scope is to design and develop a solar irradiance prediction model using ground based sky imagery using MATLAB software.

1.5 Project Overview

Under the project work, the sky images captured from the developed hardware are studied, processed and analyzed. Then a prediction model is developed with MATLAB software and tested with the real irradiation data recorded with a solar panel. Finally, the model is fine-tuned and validated with real data.

CHAPTER 02

LITERATURE REVIEW

2.1 Solar Power Prediction Techniques

With the increase of solar power integration to power grids, an accurate prediction on generation is required to maintain the system health and quality. The cloud cover and cloud movement has a direct effect on solar plant performance. When it comes to large scale solar plant integration, effect of this intermittent behaviour is threatening to the system stability.

Solar irradiance prediction gives out a critical input about the performance of the plant at various time horizons. Those forecasts are applied in system control, utility management, energy market and bidding strategies on decision making. Hour- ahead and day - ahead are the most commonly used temporal horizons in forecasts for long term decision making.

When it comes to the grid operating, the required time granularity of forecasting becomes smaller. It is required to predict the plant performance ahead for few minutes or seconds in order to maintain the system balance.

Forecast methods are being researched on various geographic horizons also. When considering small scale rooftop solar plants dispersed in an area, a prediction of higher spatial resolution is important to get a coarse estimate on upcoming electricity generation from solar panels.

There are several solar prediction techniques developing under research. Solar power prediction methods are generally characterized as physical or statistical, however in practice the line between these approaches is blurred. Physical approaches explicitly model, physical atmospheric phenomenon as part of the irradiance prediction using Numerical Weather Prediction (NWP) models or sky images. Statistical approaches predict irradiance from training and statistically derived values. For an example, a physical approach may use developing cloud vector based forecasts via interpolation of recent consecutive sky images and a statistical approach may use current and historical power output alone to predict future output.

NWP model outputs can also be fed into statistical models to improve the forecast accuracy but often it is the requirement where physical NWP models have to be developed and run at higher spatial and temporal resolutions than the general NWP models. The solar plant simulation component of a forecasting system is often a separate module that uses the required meteorological forecasts as inputs, and can also be in the form of a physical model with known specifications that can be modeled or a statistically-derived one. Solar forecast providers in practice draw from multiple forecasting methods to tailor solar plant power predictions to end user needs.

A persistence forecast is often used as the benchmark for assessing the skill of various solar plant power prediction models. A persistence forecast simply extrapolates current plant output into the future, with adjustments based on changing sun angles. The solar plant simulation component is required to predict solar plant power production using a set of mathematical equations that combines solar resource information with solar system specifications to produce power output predictions.

2.2 Numerical Weather Prediction

Numerical Weather Prediction uses the power of computers to make weather forecasts using complex computer programs run on supercomputers that generate predictions of many atmospheric variables such as temperature, pressure, wind and rainfall. Three dimensional models of the atmosphere and oceans are used to predict the weather based on current weather conditions. Initialization is the term used to describe the process of entering observation data into the model to generate initial conditions. A number of global and regional forecast models are run in different countries worldwide, using current weather observations relayed from devices known as radiosondes on weather balloons, radar, weather satellites and ground station measurements as inputs. NWP model runs are typically initiated two to four times per day, for example at 0, 6, 12 and 18UTC (Coordinated Universal Time). In order to limit computational requirements, global NWP models have relatively coarse resolutions with grid spacing in the order of 40 km to 90 km. Regional NWP models cover a limited geographical area with higher resolution, which attempt to account for local terrain and weather phenomena in more detail than global models. The weather forecasts that Sri Lanka rely on daily are the results of one or more of these NWP models.

2.3 Cloud Imagery Based Prediction

Based on cloud imagery, a wide range of prediction algorithms were developed. Basically the techniques are categorized in to two methods based on the camera location, whether it is on a satellite or on ground. These methods reduce the prediction error of NWP methods over shorter time horizons up to 6 hours.

2.3.1 Total Sky Imaging(TSI) Approach

TSI based forecasting is used to predict solar plant power output in real time up to 30 minutes ahead through advanced image processing and cloud tracking techniques. Due to the limited view capabilities, TSI devices are unable to detect clouds that will impact a site beyond the 30-minute time horizon. TSI devices take an overhead image of the surrounding sky which provides the images that are then processed to generate a forecast. In general, this approach assumes the opacity, direction and velocity of movement of the clouds in the future is consistent with the initial conditions observed through the TSI device. Irradiance is predicted based on current cloud shadows and then the cloud shadows are relocated forward in time based on cloud velocity and direction to generate the forecast. A total sky imager device, depending on cloud height, can deliver an image for 5 - 10 square miles under cloudy sky conditions and 15 square miles in totally clear sky conditions. Therefore, a single imaging device can be used to cover the area of a multi-MW solar farm, but several TSI devices would be needed to predict for a multi-hundred MW farm.

Solar power predictions using a TSI based system are the best technique to predict short term ramps for individual solar power plants. However, the need for managing ramps on a large grid system decreases with geographic distribution. Thus, total sky imagers are specifically applicable to island regions, small balancing authorities and to independent solar power producers that must meet certain ramp rate parameters. Given that TSI devices are not ubiquitous, cloud images obtained from geostationary satellites provide an alternative way to obtain cloud images, which are available for the entire globe with lower spatial resolution and longer sampling rates than TSI images.

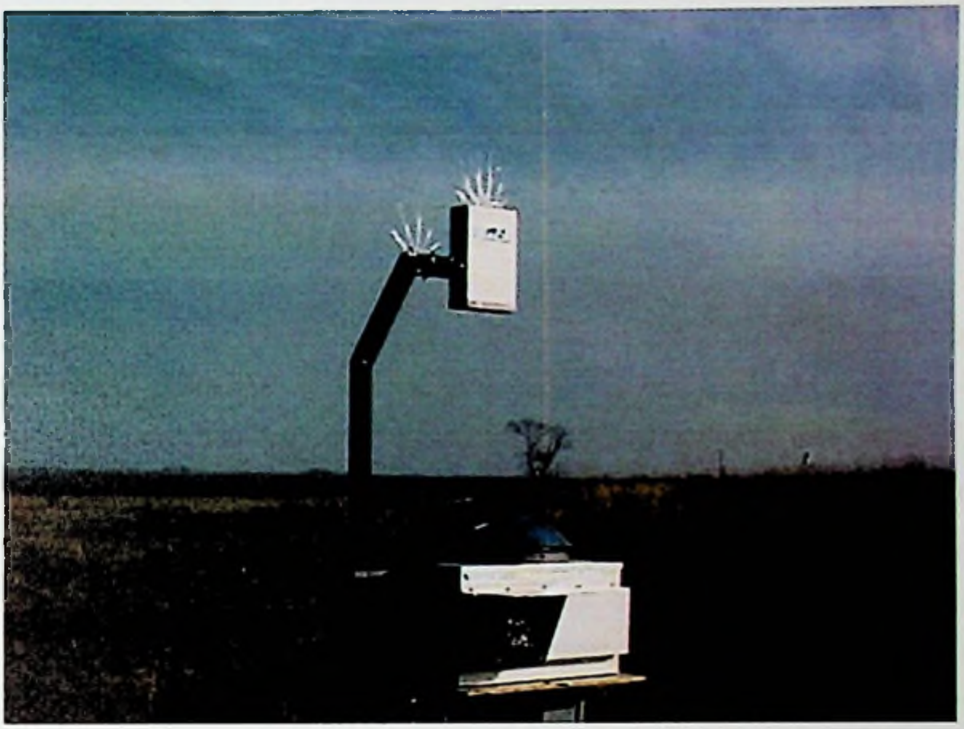

Figure 2.1: Total sky imager device (TSI 880) developed by Yankee Environmental Systems (YES) Incorporated

The total sky imager (TSI) manufactured by Yankee Environmental Systems Incorporated consists of a spherical mirror and a downward pointing camera. Images are taken every 30 seconds when the sun is above an elevation angle of three degrees. The camera provides images that are 640 by 480 pixels and the mirror occupies 420 by 420 pixels. The image output setting by the TSI data actuation software are default and cannot be changed. Images are 24-bit compressed, jpg type. The compression process induces a small loss of information in the image. Additionally, the system has an automatic gain adjustment to provide a larger dynamic range for the 8 bits on each of the red, green and blue channels. The adjustment process causes the intensity histogram of each channel to redistribute and thus the relative spectral composition of the images to fluctuate slightly with incoming signal strength. To determine sky cover obstructed by the camera arm and the shadow band, image masks are generated, edges of the masks are identified, and pixel values of the edge region are used to interpolate the region within.

2.3.2 Satellite Cloud Image Approach

The satellite cloud motion vector approach is conceptually similar to the total sky image method where cloud patterns are detected through the use of visible or infrared images from satellite based sensors and cameras flying overhead.

Clouds reflect light into the satellite leading to detection and the ability to calculate the amount of light transmitted through the cloud, which generates cloud images with the opacity characterized that are processed in a similar manner to cloud images generated using TSI devices. The direction and speed of clouds can be assessed using subsequent satellite images to predict future impacts on plane of array irradiance for a particular solar plant site. The lower spatial and higher temporal resolution for satellite images causes satellite forecasts to be less accurate than sky imagery on intra-hour time scales, but extensive comparisons or combinations of the two approaches have not been conducted. The satellite imagery technique for solar power predictions is considered the best forecasting technique from 1 to 6 hours in advance of the forecast time horizon.

The advantage of satellite imagery compared to the TSI is that a much larger spatial scale of cloud patterns can be detected and that high quality satellite imagery is continuously available for the entire world. The cloud index can be calculated accurately from the reflectance measured by the satellite. This cloud index method is a mature approach that has been used extensively in solar resource mapping. Cloud motion vectors can be determined from subsequent satellite images. Assuming cloud features do not change between two images, cloud speed is computed by finding the same features in a successive image. Hamill and Nehrkorn has applied the same method and showed the forecast with a backward trajectory technique outperformed the persistence forecast in their research. Also Hammer has developed a statistical method based on conditional probabilities to predict cloud cover and solar radiation up to 2 hours ahead.

The advantages of satellite based solar power level prediction are the spatial resolution of geostationary satellite images is 1 km or which is much less than ground based sky images. Hence with the exception of large convective clouds, most clouds cannot be detected and located directly. It can only conclude that clouds have to be located somewhere within the pixel. Compared to TSI this has disadvantages also. The time frequency, download time and processing of the images is slower than that of the sky imager, which means the forecast cannot be updated as frequently. The lack of high spatial and temporal resolution in the satellite image data reduces the performance of the satellite based approach relative to the sky imager method for very short look ahead times. However, the much larger area of coverage means that the motion of the cloud field can be projected forward over longer time periods. Satellite forecasts have been shown to outperform Numerical Weather Prediction (NWP) models for short-term forecasts. Perez has concluded that for forecasts up to 5 hours ahead satellite derived cloud motion based forecasting leads to a significant improvement over National Digital Forecast Database (NDFD) forecasts on his research publication. For 1 hour forecasts the results from persistence and satellite derived cloud motion are found to be on par, probably due to satellite's navigation and parallax uncertainties, which tend to mitigate for longer times.

2.4 Statistical Models Based Prediction

Several studies for forecasting solar irradiance in different time scales have appeared recently based on time series models, Artificial Neural Networks (ANNs), Fuzzy Logic (FL) and hybrid system such as ANFIS (Adaptive Neuro-Fuzzy Inference System), ANN-wavelet, and ANN-genetic algorithms. Many different models are used to predict the solar radiation time series, like the classic Autoregressive (AR) model, the Autoregressive and Moving Average (ARMA) model and Markov Chains. Furthermore, adaptive methods such as the Time Delay Neural Network (TDNN), which has been proven to be reliable in predicting the future trend of a time series, are also being used to solve this problem.

2.4.1 ARMA Model

The Autoregressive Moving Average (ARMA) model is usually applied to auto correlated time series data. This model is a great tool for understanding and predicting the future value of a specified time series. ARMA is based on two parts named as Auto Regressive (AR) part and Moving Average (MA) part. Also this model is usually referred to as ARMA (p, q). In this p and q are the order of AR and MA respectively. The popularity of the ARMA model is its ability to extract useful statistical properties and the adoption of the well-known Box–Jenkins methodology. ARMA models are very flexible since they can represent several different types of time series by using different order. It has been proved to be competent in prediction when there is an underlying linear correlation structure lying in the time series. One major requirement for ARMA model is that the time series must be stationary.

2.4.2 ARIMA Techniques

The ARIMA (Auto-Regressive Integrated Moving Average) techniques are reference estimators in the prediction of global radiation field. It is a stochastic process coupling autoregressive component (AR) to a moving average component (MA). ARIMA models allow treating non stationary series. Reikard has applied a regression in log to the inputs of the ARIMA models to predict the solar radiation. He has compared ARIMA models with other forecast methods such as ANN. At the 24 hour horizon, his statement was that the ARIMA model captures the sharp transitions in irradiance associated with the diurnal cycle more accurately than other methods.

2.4.3 Artificial Neural Networks (ANNs)

As an alternative to conventional approaches, artificial neural networks (ANNs) have been successfully applied for solar radiation estimation. Artificial neural networks recognize patterns in data and have been successfully applied to solar forecasting. Using training data, ANNs have been developed to reduce relative RMSE (root mean square error) or rRMSE of daily average GHI by as much as 15% when compared to 12–18 hour ahead NWP forecasts.

2.5 Hybrid Models

A persistence forecast is the most basic statistical approach to predict the future output from a solar plant. In this method past time series data is used to forecast solar plant power output in the future with minor adjustments based on the sun position in the sky. Predicting solar plant production using purely statistical methods is not generally part of modern solar power prediction systems. However, hybrid approaches make use of advanced statistical techniques to correct for known deficiencies associated with different forecasting methods through adjustments for model biases or automated learning techniques. Model Output Statistic (MOS), which is post processing model, uses statistical correlations between observed weather elements and climatological data, satellite retrievals, or modeled parameters to obtain localized statistical correction functions. This allows for the enhancement of low resolution data by considering local effects including topographic shading, or for correcting systematic deviations of a numerical model, satellite retrievals or ground sensors. One of the major disadvantages of statistical methods is the requirement for accurate historic datasets used to develop statistical correlations separately for each location. This means that MOS based forecasts are not immediately available for larger areas or for locations without prior measurements, such as most non-urban solar power plants.

Purely statistical approaches to solar power predictions do not attempt to simulate future prediction using irradiance and module temperature. The starting point is a training dataset that contains a variety of inputs that could include NWP outputs, ground station or satellite data, historic solar plant production data etc. The dataset is used to train models such as autoregressive or artificial intelligence models that produce a forecast of solar plant output over a specified future timeframe based on past inputs available at the time when the model is run. The line between physical and statistical methods is blurred in today's modern solar power prediction systems. Best practices adopt a hybrid approach whereby physical models produce irradiance and module temperature forecasts used in a solar plant simulation model, the output of which is then subject to statistical post processing to improve accuracy. Past solar plant output is an important source of data to train MOS post processing models thereby improving future forecast skill. The best statistical approaches make use of the knowledge from physical models to select input variables used to train a statistical model.

2.6 Comparison Between Techniques

The simplest stochastic learning technique is the persistence forecast which is based on current or recent PV (Photo Voltaic) power plant or radiometer output and extrapolated to account for changing sun angles. Persistence forecast accuracy decreases strongly with forecast duration as cloudiness changes from the current state.

Total sky imagery can be used to forecast from real time up to 10 to 30 minutes ahead by applying image processing and cloud tracking techniques to sky photographs. The published methods assume persistence in the opacity, direction, and velocity of movement of the clouds. Irradiance is predicted for the current cloud shadow and then the cloud shadow is moved forward in time based on cloud velocity and direction. For satellite imagery, similar methods as in total sky imagery are applied. Clouds reflect light into the satellite leading to detection and the ability to calculate the amount of light transmitted through the cloud. The higher spatial and temporal resolution probably causes satellite forecasts to be less accurate than sky imagery on intra hour time scales, but extensive comparisons or combinations of the two approaches have not been conducted. Satellite imagery is commonly considered the best forecasting technique up to 5 hour forecast range.

CHAPTER 03

TOTAL SKY IMAGING HARWARE

In order to obtain sky imagery to analyze for prediction, a total sky image recording hardware needed to be developed. Two final year project groups have developed their own total sky image capturing hardware setups with real time power and image data logging. The collected data using both hardware are used to develop and validate the prediction software model.

3.1 Hardware Setup 01

This setup is developed by a final year project group of department of Electrical Engineering, UoM of batch '10 supervised under Dr. Asanka Rodrigo.

For the image capturing methodology a Logitech® c270 web camera, which is commercially available is used mounted with a fish eye lens of 160° angle of view. Video capture capability of camera is up to 1280 x 720 pixels. The communication between the image capturing system and the processing unit is carried out by USB (Universal Serial Bus) communication provided by the manufacturer of the web camera. By this means the image data is sent continuously to the processing unit during the operation.

If there is a high illuminance level of light falling on the camera lens then there is a glare occurring on the image and the sensor of the camera gets damaged by the excess energy of light waves. When the camera turns towards the sky, inevitably the lens of the camera is exposed to the direct sun light and it is at the risk of getting damaged. Therefore, the direct sunlight falling on the lens must be avoided while allowing the system to capture sharp images of the sky.

The major threat to the camera which is the exposure to the direct sunlight has been taken care of by mean of a solar tracking device covering the sun as seen by the camera. A casing made out of Perspex is used as a cover preventing rain, dirt and dust and insects damaging the circuitry.

A light-weighted, opaque material has to be selected as the shadowing object. Since the sun is located millions of kilometers away from the earth, the size of the shadowing object has no significant impact on the shadow produced on the camera lens but it should not cover up a large proportion of the image which may cause to lose cloud data for analysis. Therefore, a Styrofoam sphere which is large enough just to cover up the sun as seen from the camera is used.

The shadowing object is held by an arm which is capable of moving freely with respect to the solar position. A thin copper wire is used initially as the arm. This prevents covering up a considerable area of the image by the arm. The length of the arm is taken as it equals to the curvature of a quadrant of a sphere so that the shadowing object is kept exactly at the top of the sphere when the sun comes to the zenith of the celestial sphere.

Figure 3.1: Total sky imaging hardware setup 01

Figure 3.2: A sky image taken without camera protection

Figure 3.3: A sky image taken with camera protection

The shadowing object follows the solar path so that it is in line with the sun at any given time blocking the direct sun light. To track the sun movement, Solar Position Algorithm (SPA) is used which is a well-established method in finding solar path in two coordinates on celestial sphere developed by NREL (National Renewable

Energy Laboratory) in America. The control of the arm of the camera protection system is done by two servo motors which are used with an arm connected to each motor to rotate in 3-Dimentional space.

The collected data are used to fine tune and validate the software model.

3.2 Hardware Setup 02

This setup is developed by a final year project group of department of Electrical Engineering, UoM of batch '11 supervised under Dr. Asanka Rodrigo.

Based on the hardware 01 setup the second hardware is developed to enhance the features and as a solution to address certain drawbacks of the previous hardware setup.

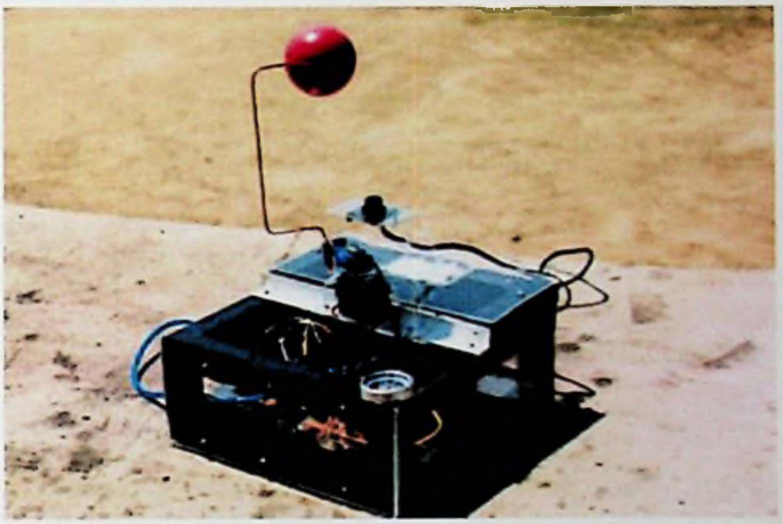

Figure 3.4: Total sky imaging hardware setup 02

In the previously developed hardware 01, a webcam with a separately attached fish eye lens has been used. But captured images were in poor quality due to the separately mounted fisheye lens was not attached in a proper manner. To eliminate this drawback a web camera with an inbuilt fish eye lens has been used for new hardware. The viewing angle of this new camera exceeds 180°.

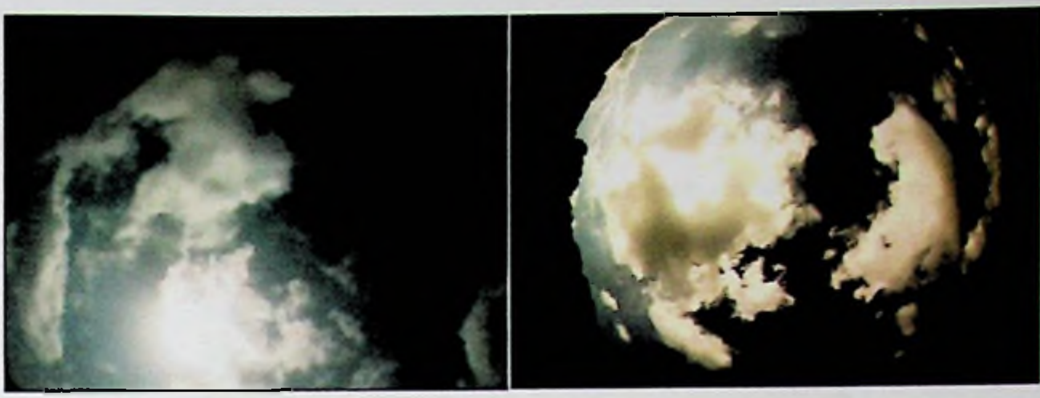

Figure 3.5: Image captured from hardware 01 (left) and image captured from hardware 02 (right)

Before using the fish eye lens mounted web camera images to process in prediction software, a calibration of the camera had to be done. Since the image has a full coverage of 180 °, the camera calibration methods can be applied in order to convert the image in to real world coordinates. The following are the identified distortions that are associated with the fish eye lens web camera.

1) Radial distortion
2) Tangential distortion

In the fish eye lens web camera which is used, the tangential distortion was absent and therefore the radial distortion parameter calculation were only required.

Optical sensors in the camera are very sensitive to the direct sunlight and hence there is a high possibility of damage to the camera when the camera is kept under the direct sunlight. Also the direct exposure to the sunlight causes a sun glare which can result a low contrast distorted image. Therefore to prevent those unwanted affects a mechanism to cover the direct sun light is necessary. One another purpose of constructing a camera protection system is to detect location of sun in image plane. Two methods of sun shading are tested under hardware 02.

1) Single axis tracker
2) Two axis tracker

In the single axis tracker that has been developed a black colored curved Perspex arm is been rotated according to the azimuth angle. The major drawback of this mechanism was the need of extra effort to detect location of sun in image plane. Therefore, this method of camera protection has been discarded and a two-axis tracker has been developed.

Figure 3.6: Single axis tracker

Figure 3.7: Two axis tracker

The sun location is easily identified with the red colour sun tracker. The SPA algorithm used in hardware 02 also same which is developed by NREL.

CHAPTER 04

DEVELOPMENT OF SOFTWARE MODEL

The software model is developed using MATLAB software with the performance of a CORE i3 processor laptop computer on Windows platform.

4.1 Preparation of Input Video Data

The recorded videos of hardware 01 are used directly in the software model with the parallel output data of a 5W polycrystalline solar panel. Since the captured range of the web camera is around 150°, the recorded images can be used in prediction algorithm directly without any prior image calibration. However, the pre-processing stage filters out the distortion to a certain extent although the resulting images are not aligned with the real world coordinated perfectly. The video quality of the web camera is 1280 x 720 pixels and 15 frames per second. Considering the data feeding gap as 10 seconds, each 150^{th} frame is extracted as image processing feed.

The recorded videos of hardware 02 are covering more than 180°. Hence the recorded videos had to be converted in to real world coordinate scale by removing the distortion. This process is named as camera calibration which should result unique parameters to a camera-lens unit and a certain angle. The video quality is 640 x 480 pixels and 30 frames per second. Considering the data feeding gap as 10 seconds, each 300^{th} frame is extracted as image processing feed.

4.1.1 Camera Calibration Process

In the used fish eye lens web camera, the tangential distortion was absent and therefore the radial distortion parameter calculation was only performed.

The theory behind the camera calibration is explained below. If (X, Y) are two pixel points in the distorted image, the calibrated pixel point or corrected pixel point can be derived to be as follows,

$$X_{CORRECTED} = X\,(1+k1r^2+k2r^4+k3r^6)$$

$$Y_{CORRECTED} = Y\,(1+k1r^2+k2r^4+k3r^6)$$

Where k1, k2, k3 are distortion coefficients and r is radius. Therefore, when determined above k1, k2, k3, the distortion coefficients, the matrix can be constructed and can be given as an input to the software as the distortion coefficient matrix. Also the following conversion matrix is determined and used in camera calibration process.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$$

c_x, c_y : optical centers expressed in pixel coordinates

f_x , f_y : camera focal lengths

X,Y,W : distorted points

x,y,w : undistorted points

The above parameters of the distortion coefficient matrix and camera matrix were determined using the captures of a 9x6 classical black and white chess board. Initially 25 snapshots are required to build the equations and derive matrix parameters.

Figure 4.1: Raw image (left) and calibrated image (right)

4.2 Cloud Identification

One of the basic sky image processing concepts in cloud identification is using the strength of red and blue channels. Clear sky looks bluer due to the molecular scattering of shorter wavelength light waves. Contrastingly clouds look whiter due to the scattering of visible range wavelength light waves more evenly. This difference is used as the way to separate clouds in sky images in many literatures.

Image is saved in MATLAB software as an RGB (Red Green Blue) matrix in digital format. Hence the different colour channels can be directly separated in the matrix. Under literature, two major methods are used to separate clouds on sky background.

1) RB ratio criterion
2) R-B (difference) criterion

Red to blue colour ratio criterion is the most basic and initially used method. The secondly developed R-B (difference) criterion which uses the difference of the two colour channels outperforms the results of RB ratio criterion. A.Kazanzidis and the group has used R-B (difference) criterion and recorded a higher accuracy level compared with RB ratio criterion.

Figure 4.2: Original image

Figure 4.3: RB ratio image

In R-B (difference) criterion, (Blue value – Red value) pattern is selected according to figure 4.4 and used to fine tune the weighting factors of the two channels. A sample of factor fine tuning is shown in figure 4.5. The factors were fine-tuned as follows.

Blue colour : 1

Red colour : 1.02

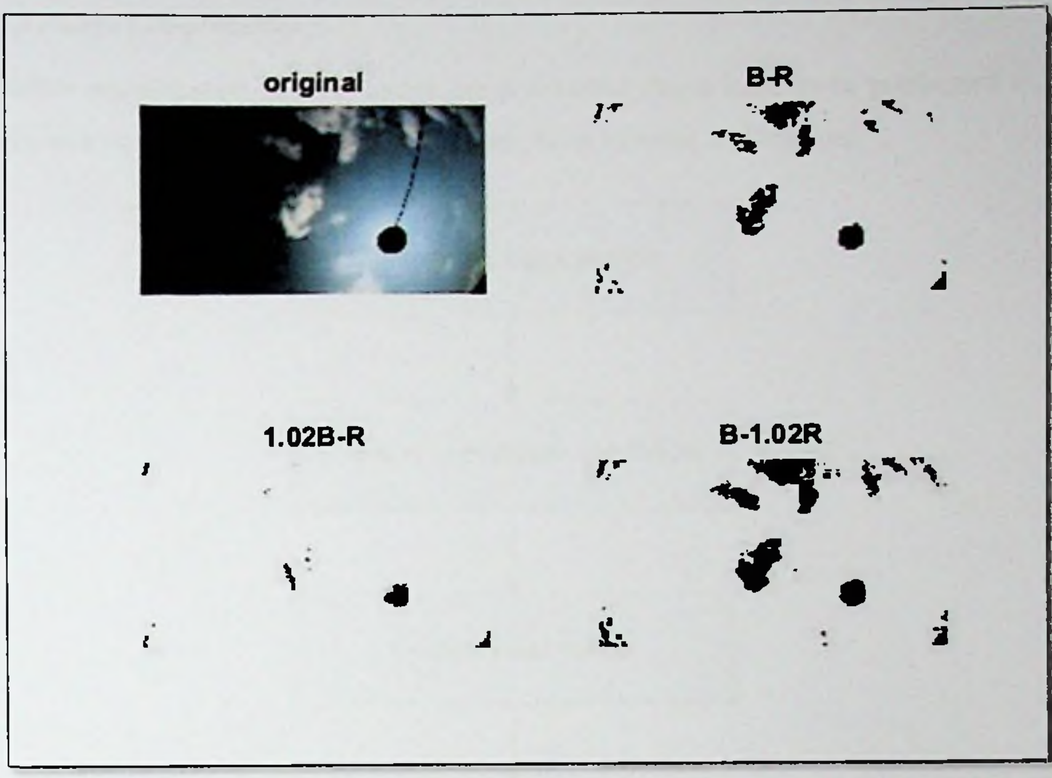

Figure 4.4: Fine tuning the equation format of Red and Blue channels

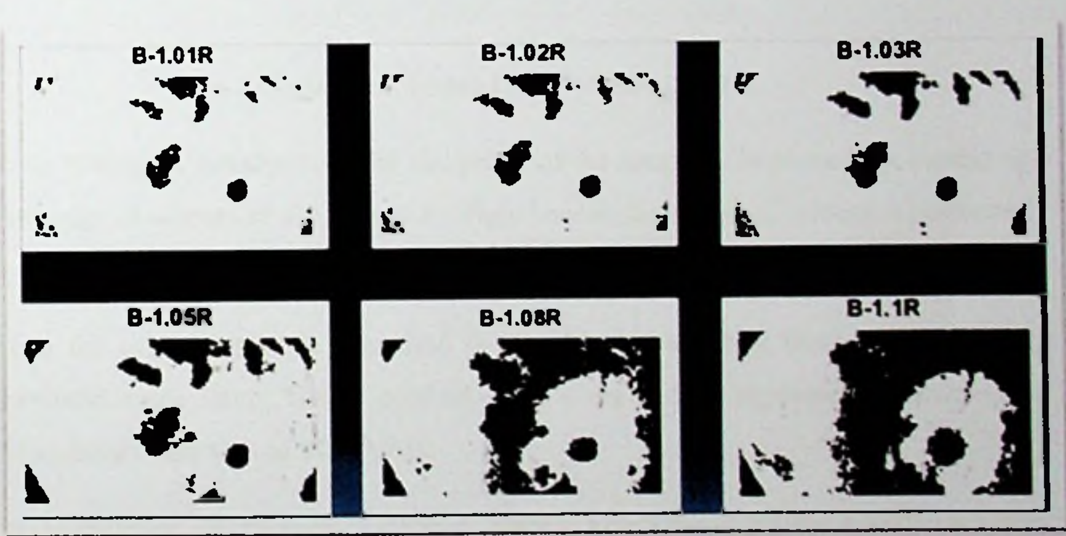

Figure 4.5: Fine tuning the weighting factors for Red and Blue channels

The result of this process is a single matrix image. And it is modified through the image pre-processing stages.

4.3 Image Pre-processing

Before segmentation, several image pre-processing stages have to be performed to improve the clearness of the image reducing level of noise and blurness.

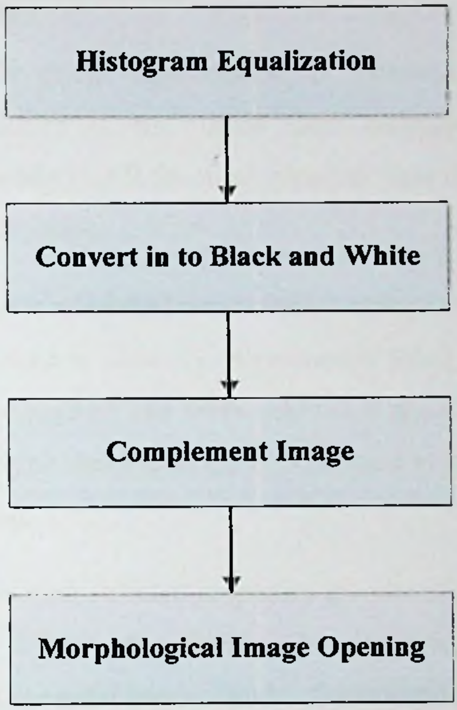

Figure 4.6: Image Pre-processing stages

With histogram equalization, the sharpness of the image is improved by increasing the range of colours of the image. In Digital scale, the range of colours is converted in to 0-255 range as the maximum.

Then the colour image is converted in to a black and white image, deciding the threshold value using Otsu's method. Otsu's method is implemented inside the 'Graythresh' function of MATLAB.

Next the image complement is taken in order to select cloud regions in colours since black colour in digital black and white image represents 0 and white colour represents 255.

Morphological image opening is an image conditioning and noise removal stage where very small isolated regions will be filtered out. This method effectively filters out noises like 'salt and pepper'. An erosion process is performed followed by a dilation using a disk of radius of 10 pixels as the structuring element. This stage clears a large amount of noise which prepares the image to identify cloud regions.

4.4 Image Segmentation

The algorithm used for image segmentation is 'connected component labeling algorithm' developed based on the 'union find' technique founded by Robert Sedgewick in 1998. The MATLAB function 'bwlabel' uses this technique to identify homogeneous areas in an image.

The result of the process is a labelled matrix with a unique label value for each area. After that a filtering is done to filter off unnecessarily small clouds where the effect on solar irradiation is negligible. The areas below 500 square pixels are filtered off. Then again an image segmentation process is performed to label the remaining areas with a new set of labeling.

The number of areas and their selected properties are identified and saved in an array under the function 'regionprops'. Using the Centroid, Area, EquiDiameter, MajorAxisLength, MinorAxisLength and BoundingBox properties further comparisons are performed.

4.5 Cloud Movement Identification

With the identified areas of the above section the software model has to find the sun shade and the clods separately for tracking process.

4.5.1 Identifying Solar Tracking Shade

Since no image data are available for night time, to position the sun at the beginning of the day the solar position is manually entered to the system. This can be pre-programed to the software model since the initial sun position changes slightly day by day.

In the images of hardware 01, the sun shade is identified using the properties of 'Centroid', 'Area' and 'EquiDiameter'. In hardware 02, since the sunshade is totally red colour and no other red colour objects are in the image, the red colour channel is filtered out and identified as the sunshade location.

Consecutive sunshade areas are identified using the properties of the initial sun tracking shade. When searching for matching clouds for each area, this sunshade area should be skipped.

4.5.2 Identifying Separate Clouds

For the prediction process, images of 10 seconds gap are fed to the software. To identify the same cloud in consecutive images for tracking purpose the properties of cloud regions are compared with each area. Among the properties available under 'regionprops' function, 'Centroid', 'Area', 'MajorAxisLength' and 'BoundingBox' are used to identify the same cloud in each image frame. The cloud region with nearest properties is matched to the initial cloud area. If a matching cloud is not identified, that area is saved and compared with the next frame. If a match is not found in that frame also that cloud data are discarded.

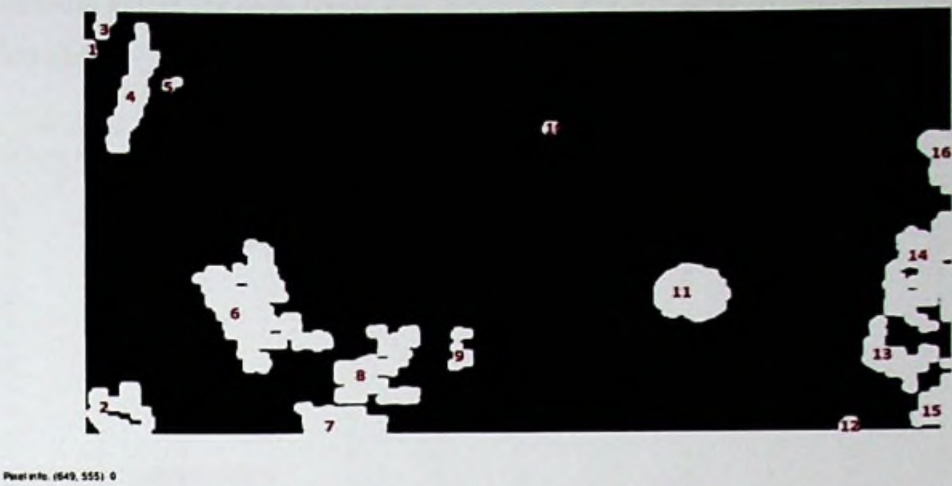

Figure 4.7: Cloud identification – frame 01

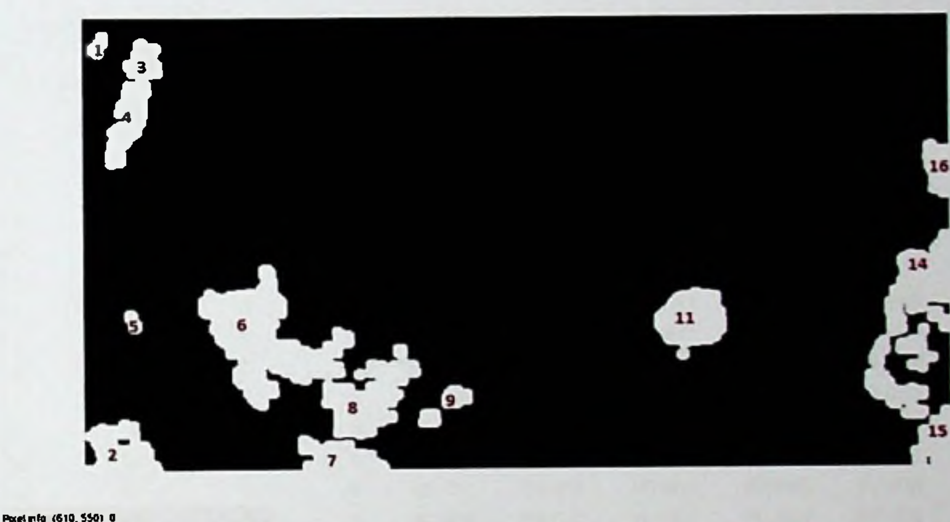

Figure 4.8: Cloud identification – frame 02

4.6 Cloud Movement Prediction and Cloud Tracking

To predict the next position of an identified cloud, its velocities along X and Y axis are calculated in pixels per second. The average velocities of the centroid of cloud are considered and predicted the X and Y coordinates of next centroid value.

The labels of matched regions for each cloud region are saved in reference array and updated with each image frame. Also cloud movement prediction is calculated as centroid values for each frame and saved in prediction array to refer to predict solar irradiance.

Reference_array

62x100 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	0	6	1	2	3	4	0	0	0	0	0	0
3	1	4	2	3	4	0	0	1	5	0	0	0
4	2	5	1	2	3	0	0	0	0	4	0	0
5	3	4	2	3	4	0	0	0	0	0	1	5
6	4	4	1	2	3	0	0	0	0	0	0	0
7	5	4	1	2	3	0	0	0	0	0	0	0
8	6	4	1	2	3	0	0	0	0	0	0	0
9	7	4	1	2	3	0	0	0	0	0	0	0
10	8	4	1	2	3	0	0	0	0	0	0	0
11	9	4	1	2	3	0	0	0	0	0	0	0
12	10	6	1	2	0	0	0	0	0	0	0	0
13	11	3	0	1	0	0	0	0	0	0	0	0
14	12	3	0	1	0	0	0	0	0	0	0	0

Figure 4.9: Sample of reference array

Prediction_array_index

62x16 double

	1	2	3	4	5	6	7	8	9
1	0	0	0	71.8474	140.1162	368.1529	541.3177	899.4787	483.3856
2	1	0	0	75.8921	132.5814	369.8566	543.7962	896.4967	486.8147
3	2	0	0	100.1604	87.3726	380.0791	558.6675	878.6051	507.3893
4	3	0	0	124.4287	42.1638	390.3015	573.5389	860.7134	527.9639
5	4	0	0	148.6970	-3.0451	400.5239	588.4102	842.8217	548.5385
6	5	0	0	172.9654	-48.2539	410.7464	603.2815	824.9300	569.1131
7	6	0	0	197.2337	-93.4627	420.9688	618.1529	807.0383	589.6877
8	7	0	0	221.5020	-138.6716	431.1913	633.0242	789.1466	610.2623
9	8	0	0	245.7703	-183.8804	441.4137	647.8955	771.2549	630.8369
10	9	0	0	270.0386	-229.0892	451.6362	662.7669	753.3632	651.4115
11	10	0	0	294.3069	-274.2980	461.8586	677.6382	735.4715	671.9861
12	11	0	0	318.5752	-319.5069	472.0811	692.5095	717.5798	692.5607
13	12	0	0	342.8435	-364.7157	482.3035	707.3809	699.6881	713.1353
14	13	0	0	367.1118	-409.9245	492.5260	722.2522	681.7964	733.7099
15	14	0	0	391.3801	-455.1334	502.7484	737.1235	663.9048	754.2844
16	15	0	0	415.6484	-500.3422	512.9709	751.9948	646.0131	774.8590
17	16	0	0	439.9167	-545.5510	523.1933	766.8662	628.1214	795.4336

Figure 4.10: Sample of prediction array

4.7 Solar Coverage Approximation

The amount of solar coverage with each cloud is approximated separately with the developed algorithm. Using the calculated velocities for X and Y directions of each cloud the solar covering time and duration are calculated separately for each cloud.

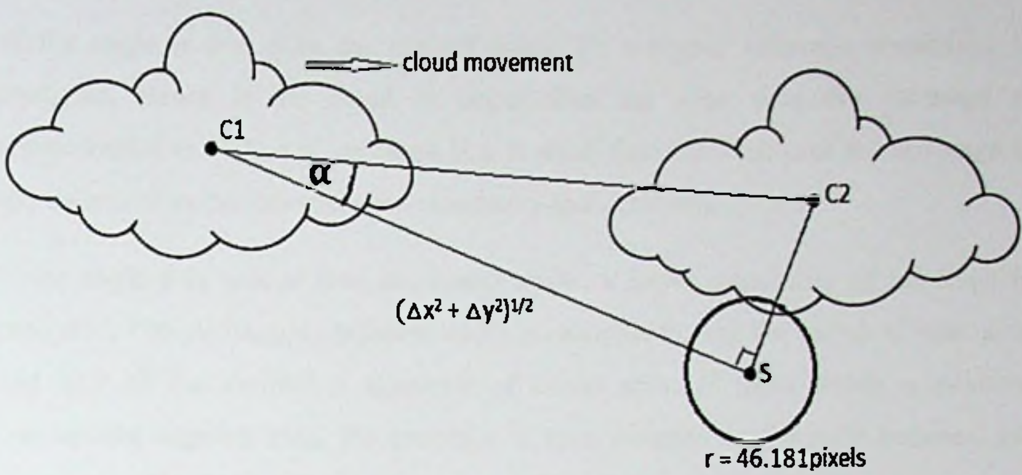

Figure 4.11: Cloud cover approximation algorithm sketch

- $C2S = Tan(Alpha)\,(\Delta x^2 + \Delta y^2)^{1/2}$
- Width of the overlapping area = (Equivalent Diameter/2 + r) – C2S

Direction of cloud position vector from sun shade and the direction of velocity vector of cloud centroid are shown in the figure and used in the algorithm.

Difference between the 2 vectors is calculated as Delta _alpha and Cutoff Angle is approximated as 7° for the studied data sets which were captured over University of Moratuwa. The coverage approximation algorithm by any cloud with the use of Delta_Alpha variable is mentioned below.

If (Delta _alpha < Cutoff Angle)

If (Cloud_size > Solar_area)

Coverage = 1

else Coverage = Cloud_size / Solar_area

If (Delta _alpha > Cutoff Angle)

If (overlapping segment width > 0)

Coverage = overlapping segment width /Solar_area Diameter

Angle α can be calculated using the C_1C_2 and C_1S distances.

If the angle is less than the cut off angle 7°, a higher coverage possibility is predicted. Hence if the cloud is larger than the solar area the coverage is approximated as 1. And if the cloud size is small than the solar area the coverage is approximated as the ratio between cloud area and solar area.

If the angle α is greater than the cutoff angle, a lower possibility of coverage is predicted. The overlapping segment width is calculated using the radius of solar area and half of the equivalent diameter of cloud area. If there exists a positive overlapping segment area, the coverage is approximated as the ratio between the segment width and the solar area diameter.

This calculated coverage is used as cloud coverage in prediction algorithm.

4.7.1 Cloud Factor

The total amount of clouds in sky other than the direct sun covering clouds affects the solar irradiation indirectly. To count this effect on prediction the variable 'cloud factor' is calculated for each image frame.

$$\text{Cloud Factor} = \frac{\text{Cloud pixels in frame}}{\text{Total pixels in frame} - \text{Sunshade area}}$$

4.8 Prediction Algorithm

A 5W polycrystalline solar panel is used to record real time power data with the sky videos. Hence the prediction algorithm is developed to predict the current output of the same solar panel considering the maximum current output of the panel as 0.22A. The other variables used in the prediction equation are cloud cover and cloud factor.

$$I_{predicted} = I_{max} - (K_1 \times \text{Cloud Cover}) - (K_2 \times \text{Cloud Factor}) + K_3$$

The constants are fine-tuned as follows.

$$K_1 = 0.45 \; ; K_2 = 1.67$$

If a cloud covers sun, $K_3 = 0.69$; else $K_3 = 0.48$

CHAPTER 05

DISCUSSION

5.1 Methodology

The methodology used in this research to predict solar irradiance level is based on image processing of total sky images. Under the group of researching methodologies for solar prediction, this method is a leading approach to get a short time horizon prediction around a lower spatial span.

The data source used is a hardware module located at University of Moratuwa which covers total visible sky range at the location of (6.7969° N, 79.9018° E). The total sky images collected through the module are undergone through several pre-processing stages and fed to the prediction methodology which is based on individual cloud tracking. The movement of each cloud is tracked throughout the span irrespective of its effect on irradiation. The individual cloud effects are approximated with the prediction algorithm and the overall prediction is given as the result with a 10 second interval.

The prediction outcome of this methodology is the electric current output of a fixed solar panel which is also located at the same place with the hardware module. The solar panel behavior is assumed to be constant throughout the prediction span.

5.2 Comparison with Other Methodology

Among different solar prediction methods in research, there are several sky imagery methods developed. The two main sky imagery methods are satellite imagery and ground based sky imagery. Satellite imagery are used in a different prediction method compared with ground based imagery methods. Moving camera and fish eye lens camera are the two types of cameras used in ground based hardware setups which are used to collect cloud data of sky for prediction purposes.

The prediction methodology calculations used in moving camera methods are more sophisticated and complex since the coordinates of the image with respect to the real world coordinates are being changed continuously. The extracted images show more details since only a focused area of sky is captured. As the details over a small spatial horizon available, the temporal horizon of the prediction is comparatively lower.

The fixed camera which captures the total sky has few advantages over the moving camera. Since the camera is still the image coordinates are same as the real world coordinates. Hence the calculation complexity is comparatively lower. Also as a moving mechanism is not required for camera, the development and maintenance cost of the hardware are less. The details captured in the image for a small area is lesser in fixed camera since the captured area is higher resulting a lesser prediction accuracy compared with moving camera prediction method.

The unique features of the developed model are; it is specifically developed for the UoM coordinates with a low cost hardware setup compared with high end total sky imagers and the model can be adjusted and fine-tuned for different locations. This model is developed considering the cloud types and patterns of a tropical country like Sri Lanka.

5.3 Results and Validation of Model

The model is based on the tracking of individual clouds and integrating in to the prediction result. Therefore, individual cloud tracking accuracy directly affects the prediction result. The obtained accuracy is shown in next section.

5.3.1 Prediction and Tracking Accuracy of a Cloud

The graph shown in figure 5.01 shows the accuracy of movement prediction of a single cloud throughout the visible span through images. The graph is plotted over the X axis vs Y axis of the MATLAB image plane using centroid coordinates of the cloud at each consecutive frame. The actual cloud position is estimated using MATLAB image processing techniques, which gives the centroid coordinates of the identified cloud region.

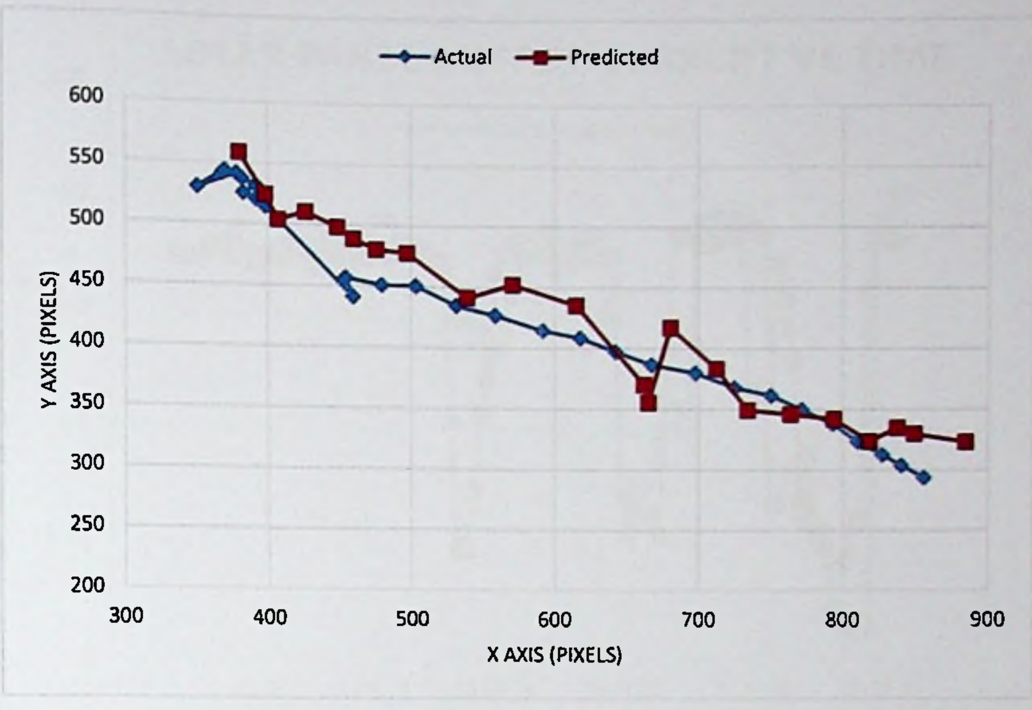

Figure 5.01: Accuracy of a single cloud movement prediction

5.3.2 Validation of The Prediction Model

The developed prediction algorithm is validated using the sky image captures of hardware 01 and hardware 02 parallel with the solar panel power output data. Graphs shown in figure 5.02 and figure 5.03 shows the solar panel output current prediction accuracy of the developed model with a set of data of hardware 01 and hardware 02 respectively.

The visible valleys in graphs are due to clouds which flow over solar area while the other area prediction is performed with cloud factor data.

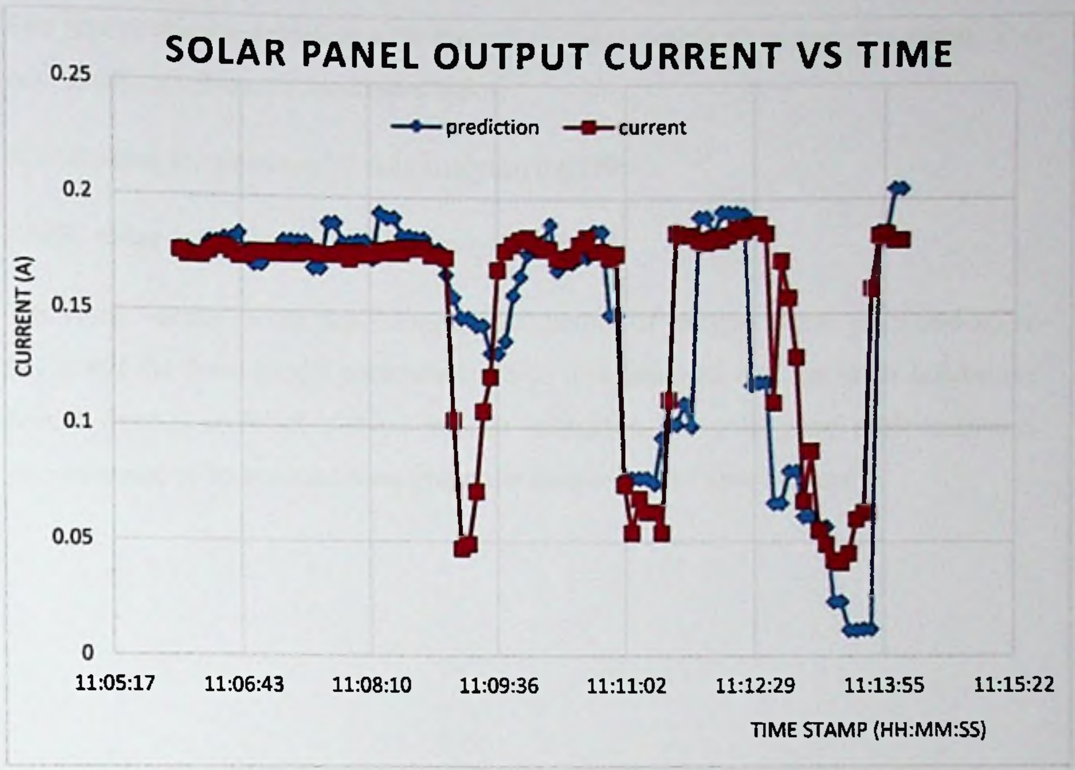

Figure 5.02: Predicted and actual solar panel output current using hardware 01images

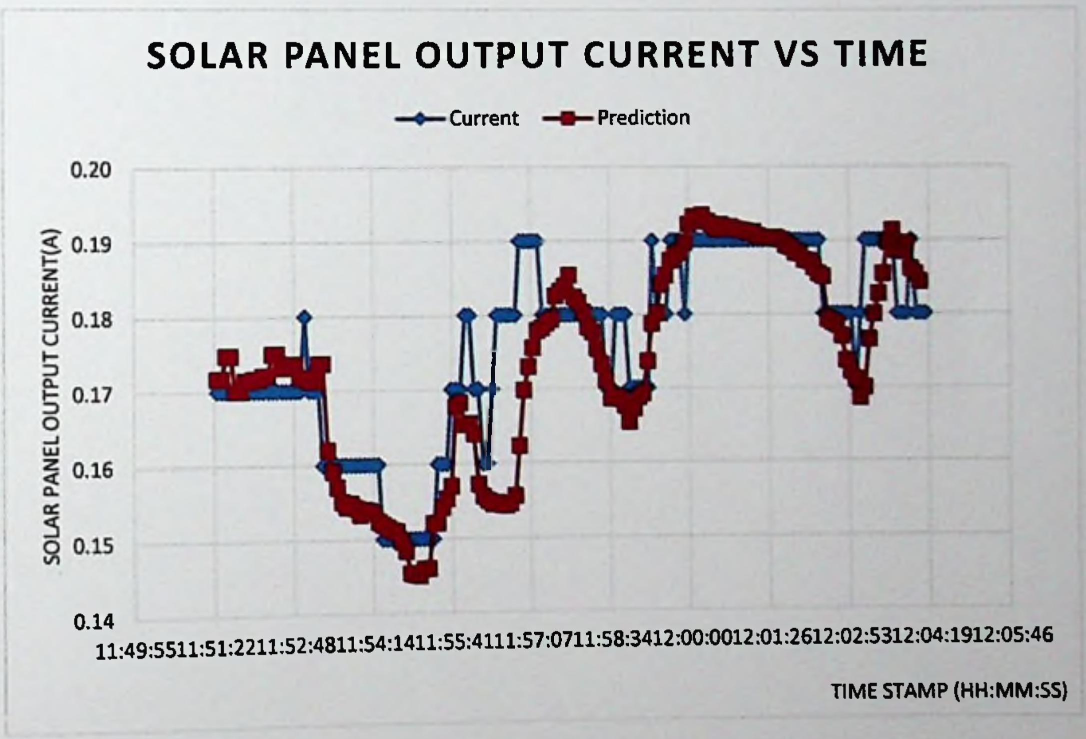

Figure 5.03:Predicted and actual solar panel output current using hardware 02 images

The results show a deviation with the actual value within an acceptable range. The error analysis values are mentioned below.

RMSE value for hardware 01 data analysis: 0.0359

RMSE value for hardware 02 data analysis: 0.1024

The result variations are depending on the quality of images where glare and noise occur and the fixed model parameters where it is assumed that the other conditions remain fixed or no effect of them on solar irradiation. The solar panel performance is also assumed to be constant throughout the data collected time horizon.

CHAPTER 06

CONCLUSIONS

This software model is developed as a location specified model, to the coordinates of University of Moratuwa (6.7969° N, 79.9018° E). The prediction results can be used real time for a solar farm at the location to maintain and improve the power system balance and stability, whether it is a standalone system or a grid connected farm.

6.1 Model Limitations

The prediction of solar irradiance is based on the output imagery of the developed hardware by university students. The performance of the hardware are not up to the extreme end performance like commercial total sky imagers. The coverage and image quality variations effect the accuracy of results.

The obtained results of the image processing model are approximately accurate for 10 seconds of duration. As a short term prediction model, these results can be used in immediate system control decisions. The temporal resolution can be improved with higher quality images and further developed prediction algorithm.

6.2 Future Research and Applications

The hardware can be developed in to a fully automated system with a mini computer installed on board to output the prediction results real-time. Also the processed data can be transferred to a main system or a data logging system through a wireless communication module for further analysis as an improvement.

The model can be improved with integration of ground specific variables like temperature, wind speed which affects the output of a solar panel. Additionally, software constants can be separately fine-tuned for different months based on the monsoon effect for the location on each month.

Also the integrated hardware and software module can be installed in other locations and modify the prediction algorithm for those locations. To apply the developed algorithm in to a different location, the constants of the model should be fine-tuned using collected data of the location.

Also if a different type and size of a solar panel is used instead of 5W polycrystalline panel, the variable I_{max} will be a different value. Hence the model has to be adjusted by fine-tuning the constants with location specific solar irradiance data. The solar panel performance is assumed to be constant throughout the considered time span.

A solar prediction model running with a hardware is very useful in integrating a solar farm to a grid system or using standalone. This developed model can be used to predict the energy harvest short-term, where electricity generation and consumption can be mapped real-time.

REFERENCES

[1] V.T.Jayadevan, V.P.A. Lonij, "Forecasting solar power intermittency using ground based cloud imaging", *Proceedings of American Solar Energy Society Meeting* (2012)

[2] V. Kostylev, A. Pavlovski, "Solar power forecasting performance - towards industry standards", *Workshop on the integration of solar, 2011*

[3] C.Cornaro, F Bucci, M. Pierro, F.Del Frate, S. Peronaci, A. Taravat, "solar radiation forecast using neural networks for the prediction of grid connected pv plants energy production (dsp project)", *Weather Intelligence for Renewable Energies conference*

[4] S.Sendanayake, M.P.Miguntanna, M.T.R.Jayasinghe, " Estimating incident solar radiation in tropical islands with short term weather data" , *European Scientific Journal January 2014*

[5] C. N. Long, D. W. Slater T. Tooman, "Total Sky Imager Model 880 Status and Testing Results", November 2001

[6] T. Hashimoto, Y. Nagakura, "Prediction of output power variation of solar power plant by image measurement of cloud movement", *Journal of Advanced Research in Physics 2(2), 021107 (2011)*

[7] Long, C.N., Sabburg, J.M., Calbé, J., Pagès, D., 2006. Retrieving cloud characteristics from ground - based day time color all - sky images . J.Atmos.OceanicTechnol.23,633–652)

[8] A. Kazanzidis, P. Tsoumanikas,A.F.Bais,F.Photopaulos,G. Economou, "Cloud detection and classification with the whole sky ground based images", *Atmospheric research 2012*

[9] Yankee Enviromental Systems,inc. (2006) TSI-880 automatic total sky imager,[Online]. Available: http://www.vesinc.com/products/data/tsi880/.

[10] A.S. Rodrigo, K.H.E.Perera, H.M.S. Priyadharshana, V.G.C. Priyanka, R.A.R.A. Ranasinghe, "Cloud images capturing system for solar power level prediction," *Moratuwa Engineering Research Conference (MERCon), April 7-8, 2015.*

APPENDIX A

MATLAB code of the software model

```
clear
im1=imread('output2\frame-150.jpg');%read first image
imd=double(im1);
imr=imd(:,:,1);
imb=imd(:,:,2);
imdiff=imb-1.02*imr;% this const to be fine tuned
imh=histeq(imdiff);
L=graythresh(imh);
imbw=im2bw(imh,L); % convert image in to B&W
imc=imcomplement(imbw);

seo1 = strel('disk',10);
isp1=imopen(imc,seo1);

label1=bwlabel(isp1);
R = regionprops(isp1,'Area'); % preparing area filter
ind = find([R.Area] >= 500);

filteredIm1 = ismember(label1,ind); %applying area filter
processedIm1 = bwlabel(filteredIm1);
%identify sun shade1, give centroid limits
im1labels = max(max(processedIm1))
%get max of columns and get max of them
stats1 =
regionprops(logical(filteredIm1),'Centroid','Area','MajorAxisL
ength','MinorAxisLength','EquivDiameter');
sunshade1=[0,0,0,0];%Centroid(1),Centroid(2),Area,label
for k = 1:im1labels
    data = stats1(k)
    if(abs(data.Area-6700)<3000)%area limit for initial
sunshade
        if(abs(1-
(data.MajorAxisLength/data.MinorAxisLength))<0.5)
            if((abs((data.Centroid(1)-900))<5) &&
(abs((data.Centroid(2)-500))<20))%decided from centroid data
of given video
                if(sunshade1==[0,0,0,0])%if no sunshade1 so
far set this

sunshade1=[data.Centroid(1),data.Centroid(2),data.Area,k];
                elseif(abs(data.Area-6700)< abs(sunshade1(3)-
6700))%check which data nearest to 6700

sunshade1=[data.Centroid(1),data.Centroid(2),data.Area,k];
                end
            end
        end
```

```
    end
end
index = 1;
Frame_No = 150;
Reference_array(1,:) = 1:100;% an array to keep track of the
same cloud while labels are changing, 2nd row filled with
im1labels
%define the no. of columns with a column index
Reference_array(2:62,1) = 0:60;%filling index column up to
1hour
Reference_array(2,2:im1labels+1)= 1:im1labels;%2nd row up to
im1 labels
Reference_array(2,2) = sunshade1(4);%writing 1st sunshade
label at row2
Reference_array(2,sunshade1(4)+1) = 1;%giving a place to
replaced label no.01
last_column_index = im1labels+1;%keep track of the last
column, +1
missed_clouds = 0;
pre_missed_clouds = 0;
total_missed_clouds = 0;
%-------------------------------------------------------------
-
while(index<300)% no of frames to go
    if(index>1) %if it is not the first round
      im1 = im2;
      im1labels = im2labels;
      stats1 = stats2;
      sunshade1 = sunshade2;
    end
Frame_No = Frame_No + 450;%frame after 30 sec
Address = strcat('output2\frame-',int2str(Frame_No),'.jpg');
im2=imread(Address);% read next image
%im2 processing
imd=double(im2);
imr=imd(:,:,1);
imb=imd(:,:,2);
imdiff=imb-1.02*imr;
imh=histeq(imdiff);% histogram equalization
L=graythresh(imh);
imbw=im2bw(imh,L); % convert image in to B&W
imc=imcomplement(imbw);

seo1 = strel('disk',10);
isp1=imopen(imc,seo1);

label1=bwlabel(isp1);
R = regionprops(isp1,'Area'); % preparing area filter
ind = find([R.Area] >= 500);

filteredIm2 = ismember(label1,ind); %applying area filter
processedIm2 = bwlabel(filteredIm2);
```

```
%imshow(filteredIm2);
im2labels = max(max(processedIm2))%get max of columns and get
max of them
stats2 =
regionprops(logical(filteredIm2),'Centroid','Area','MajorAxisL
ength','MinorAxisLength','EquivDiameter');
%above labeled matrix converted in to a logical matrix to save
memory
%identify sun shade2
sunshade2=[0,0,0,0];%Centroid(1),Centroid(2),Area,label
cloud_factor = sum(imc(:))/(1280*720-6700);
range1 =
strcat('A',int2str(uint16(index*5)),':A',int2str(uint16(index*
5)));
xlswrite('Prediction.xlsx',cloud_factor,2,range1);
for m = 1:im2labels % to find sunshade2
    data = stats2(m);
    if(abs(sunshade1(1)-data.Centroid(1))<50) &&
(abs(sunshade1(2)-data.Centroid(2))<50)
        if(abs(data.Area-6700)<3000)
         if(sunshade2==[0,0,0,0])
sunshade2=[data.Centroid(1),data.Centroid(2),data.Area,m];
            elseif(abs(data.Area-6700)< abs(sunshade2(3)-
6700))%compare area to find best match

sunshade2=[data.Centroid(1),data.Centroid(2),data.Area,m];
         end
       end
    end
end
if(index==1)
    Reference_array(3,sunshade1(4)) = sunshade2(4);%selecting
reference column for sunshade
    Reference_array(index+2,2) = sunshade2(4);%writing
sunshade value2 in reference array column2
else  for row_two_value = 1:last_column_index% if 26 columns
exist
        if(Reference_array(2,row_two_value)== sunshade1(4))
            Reference_array(3,row_two_value)= sunshade2(4);
            break;
        end
      end
end
 matched_array = true([1 im2labels]);%definining a logic array
to filter selected clouds
matched_array(uint8(sunshade2(4)))=0;%skipping 2nd sunshade
from matching algo
matched_cloud = [0,0,0,0,0];% define an array for matched
centroid, Centroid(1),Centroid(2),Area,label,eqD
%maxrow3column = 0;
  for k = 1:im1labels %image1 index
```

```
    if (k == sunshade1(4)) %skipping sunshade1 from searching
algo
        continue;
    end
    a=matched_cloud(4);%a gets a value if a cloud from im2 is
matched to previous k
    if((k~=1) && (a~=0))
        matched_array(uint8(a))=0;
    end %skiping selected clouds, k not required, not
 applicable at first round
    matched_cloud = [0,0,0,0,0];% resetting array for matched
centroid, Centroid(1),Centroid(2),Area,label,eqD
    velocity = 0;
    coverX = 0;
    coverY = 0;
    time_to_cover_sun = 0;
    covering_duration = 0;
    x = stats1(k).Centroid;%image1 data
for m = 1:im2labels % to match a m for the current k
 if(matched_array(m) == 1)%if no cloud is matched for m so far
        y = stats2(m).Centroid;%image2 data
        if (abs((y(1)-x(1)))<100 && abs((y(2)-x(2)))<100)%
check centroid values
            if(abs(stats2(m).Area-stats1(k).Area) <
2700)%check area
                if(abs(stats2(m).MajorAxisLength-
stats1(k).MajorAxisLength) < 500)%check shape
                    if(matched_cloud == [0,0,0,0,0])
                        matched_cloud =
[y(1),y(2),stats2(m).Area,m,stats2(m).EquivDiameter];
                    elseif (abs(stats2(m).Area-
stats1(k).Area)<(abs(matched_cloud(3)-stats1(k).Area)))%select
lowest area difference
                        matched_cloud =
[y(1),y(2),stats2(m).Area,m,stats2(m).EquivDiameter];

                    end
                  end
                end
            end
        end
    end
for n = 2:last_column_index % to write matched labels to ref.
array
        if (Reference_array(index+1,n)==k)% finding ref.column
related to current k label
            Reference_array(index+2,n) = matched_cloud(4);
            if maxrow3column < row2column
                maxrow3column = row2column;
            end
            break;
        end
```

```
    end
  end
    % count total missed clouds
    for n = 2:last_column_index
        if (Reference_array(index+2,n) == 0)
          total_missed_clouds = total_missed_clouds+1;
        end
    end
    missed_clouds = total_missed_clouds -
pre_missed_clouds;%calc missed_clouds for this frame
  pre_missed_clouds = total_missed_clouds;%updating pre missed
clouds

    matched_array = true([1 im2labels]);%redefinining the
logic array to filter missed clouds
for n = 1:im2labels % to find not matching labels// this
loop's result is same as current value
       for m = 2:last_column_index % searching through all the
used columns
           if(Reference_array(index+2,m) == n)
               matched_array(1,n) = 0; %making matched labels
zero
           end
       end
end
    %writing missed_cloud values to ref array
    for x = 1:im2labels
        if(matched_array(1,x) == 1)   %selecting missed values
          Reference_array(index+2,last_column_index+1) = x;
%writing missed values
          last_column_index = last_column_index + 1;%increment
last_column_index
        end
    end
%     Reference_array
% eval(sprintf('Prediction_array_%i =
zeros(62,last_column_index*2);',index));
    Prediction_array_index = zeros(62,last_column_index*2+1);
Prediction_array_index(1:62,1) = 0:61;%index column

    %calc velocity
    for x = 2: last_column_index %column range in ref array
        if(Reference_array(index+1,x) ~= 0 &&
Reference_array(index+2,x) ~= 0)%there exists a matched cloud
in lower row
            %xlswrite('ref_array.xlsx', Reference_array);%to
check ref array values
            %stats1(Reference_array(index+1,x));
          Centroid1X =
stats1(Reference_array(index+1,x)).Centroid(1);
          Centroid1Y =
stats1(Reference_array(index+1,x)).Centroid(2);
```

```
%           EqDiameter1 =
stats1(Reference_array(3,x)).EquivDiameter;
          Centroid2X =
stats2(Reference_array(index+2,x)).Centroid(1);
          Centroid2Y =
stats2(Reference_array(index+2,x)).Centroid(2);
          EqDiameter2 =
stats2(Reference_array(index+2,x)).EquivDiameter;

         Vx = (Centroid2X-Centroid1X)/0.5% if +ve moving right
(v per min)
         Vy = (Centroid2Y-Centroid1Y)/0.5% if +ve moving down

         % predict next cloud positions each minute
         Prediction_array_index(1,2*(x-1))= Centroid1X; % if x
=2, these are sunshade values
         Prediction_array_index(1,2*x-1)= Centroid1Y;%given x
range starts from 2
         Prediction_array_index(2,2*(x-1))=
Centroid2X;%therefore x replaced with (x-1)at x centroid
column logic
         Prediction_array_index(2,2*x-1)= Centroid2Y;

            for minute = 3:60
                Prediction_array_index(minute,2*(x-1)) =
Centroid2X + Vx; %x value to array
                Centroid2X = Centroid2X + Vx;%updating
centroid variable
                Prediction_array_index(minute,2*x-1) =
Centroid2Y + Vy; %y
                Centroid2Y = Centroid2Y + Vy;
            end
         % predict solar coverage
if(Vx>0 && (sunshade2(1)>(Centroid2X)-EqDiameter2/2))% (x -
cloud radius)
                coverX = 1;
            elseif (Vx<0 &&
(sunshade2(1)<(stats2(Reference_array(index+2,x)).Centroid(1)+
stats2(Reference_array(index+2,x)).EquivDiameter/2)))
                coverX = 1;
            end
            if(Vy>0 &&
(sunshade2(2)>(stats2(Reference_array(index+2,x)).Centroid(2)-
stats2(Reference_array(index+2,x)).EquivDiameter/2)))
                coverY = 1;
            elseif (Vy<0 &&
(sunshade2(2)<(stats2(Reference_array(index+2,x)).Centroid(2)+
stats2(Reference_array(index+2,x)).EquivDiameter/2)))
                coverY = 1;
            end

        end
```

```
    end
     Vx = (matched_cloud(1)-x(1))/5;% if -ve moving right
    Vy = (matched_cloud(2)-x(2))/5;% if -ve moving down
    predict solar coverage
    if(Vx>0 && (sunshade2(1)>(matched_cloud(1)-
matched_cloud(4)/2)))% (x -cloud radius)
        coverX = 1;
    elseif (Vx<0 &&
(sunshade2(1)<(matched_cloud(1)+matched_cloud(4)/2)))
        coverX = 1;
    end
    if(Vy>0 && (sunshade2(2)>(matched_cloud(2)-
matched_cloud(4)/2)))
        coverY = 1;
    elseif (Vy<0 &&
(sunshade2(2)<(matched_cloud(2)+matched_cloud(4)/2)))
        coverY = 1;
    end
    calc covering time & duration
    if((coverX == 1) && (coverY == 1))
        Tx = abs((sunshade1(1)-matched_cloud(1)-
(matched_cloud(4)/2))/Vx);%sun considered as a point compared
with cloud size
        Ty = abs((sunshade1(2)-matched_cloud(2)-
(matched_cloud(4)/2))/Vy);
        Dx =
abs((stats2(matched_cloud(4)).EquivDiameter+92.362)/Vx);
%pi*r^2 = 6700
        Dy =
abs((stats2(matched_cloud(4)).EquivDiameter+92.362)/Vy);
        % checking overlap
        if(Tx<Ty)% finding which axis covers first
            R1 = Ty;
        else R1 = Tx;
        end
        if((Tx+Dx)<(Ty+Dy))% finding which axis stops covering
first
            R2 = Tx+Dx;
        else R2 = Ty+Dy;
        end
        if(R1<R2)
            R1 = R1 + (index + 1)*5;%considering 5s rounds,
image 1 is out of this index, so +1
            R2 = R2 + (index + 1)*5;% remove +1 its wrong

        %predict covering percentage
        VT_alpha = Vy/Vx;
        PT_alpha = (sunshade2(2)-
matched_cloud(2))/(sunshade2(1)-matched_cloud(1));
        Delta_alpha = abs(atand(PT_alpha)-atand(VT_alpha));
        covering_percentage = 0;
            if(matched_cloud(3)>6700 && Delta_alpha<7 )
```

```
                covering_percentage = 1;
            elseif(matched_cloud(3)<6700 && Delta_alpha<7 )
                covering_percentage =matched_cloud(3) /6700;
            elseif( Delta_alpha>7 )
                if(matched_cloud(5)/2+46.181-
tan(Delta_alpha)*((sunshade2(1)-
matched_cloud(1))^2+(sunshade2(2)-matched_cloud(2))^2)^0.5<0)
                covering_percentage = 0;
                else covering_percentage
=((matched_cloud(5)/2+46.181-tan(Delta_alpha)*((sunshade2(1)-
matched_cloud(1))^2+(sunshade2(2)-
matched_cloud(2))^2)^0.5))/(2*46.181);
                end
elseif(matched_cloud(3)<6700 && Delta_alpha>7 )
                covering_percentage =0.3;
            else covering_percentage = 0.5;
            end
            write to excel sheet
            range =
strcat('F',int2str(uint16(R1)),':F',int2str(uint16(R2)));
            [rows,columns] = size(Prediction_array_index);
            if columns>256
                error('column limit exceeded');
            end

            range =
strcat('A1',':',char2str(lastcolumn),':',int2str(count));

xlswrite('Prediction.xlsx',covering_percentage,range)
              xlswrite('cloud_prediction.xlsx',
Reference_array,index);%excelsheet, array name, sheet no.
                  range = strcat('A',int2str(index));
              xlswrite('cloud_prediction_final.xlsx',
Prediction_array_index,1,range);
              Prediction_array_index;
      end
    end
end

index = index + 1;
  %Reference_array;
end
%subplot(2,1,1);imshow(processedIm1);title('1');
%subplot(2,1,2);imshow(processedIm2
```