

**FORECASTING SOLAR POWER INTERMITTENCY
USING GROUND-BASED SKY IMAGING**

by

Vijai T. Jayadevan

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
In the Graduate College
THE UNIVERSITY OF ARIZONA

2 0 1 3

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Jeffrey J. Rodriguez
Associate Professor of
Electrical and Computer Engineering

Date

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Jeffrey J. Rodriguez, for all the guidance I received from him, and for teaching me how to do good research. I would also like to thank Dr. Alexander D. Cronin for entrusting me with this project. His appreciation of even the simplest task I did, kept me inspired to get better. I am grateful to Dr. Ali Bilgin for accepting my request for being a part of my committee and for his helpful suggestions. I am also grateful to Dr. Vincent P. A. Lonij, who helped me with my every step in the initial phases of the project. My lab mates, Sundaresh Ram, Adria E. Brooks, Ramaprasad Kulkarni, Dani DellaGiustina, Deanna Lewis and Kevin Duong deserve many thanks for their friendship and help during my graduate studies. I would like to thank my parents, brother and sister, for their unconditional love, encouragement, patience, and support which has always kept me going. Finally, thanks to the Tucson Electric Power for funding my research assistantship.

To my parents, brother and sister
whose love and encouragement
made this possible

TABLE OF CONTENTS

LIST OF FIGURES	7
LIST OF TABLES	9
ABSTRACT	10
1 INTRODUCTION	11
2 IMAGING SETUP	14
2.1 Overview	14
2.2 Sun tracker	15
2.3 A note on image quality	16
3 IMAGE PREPROCESSING	21
4 CLOUD DETECTION	23
4.1 Introduction	23
4.2 A contrast-enhancing feature for cloud detection	23
4.2.1 Cloud detection methods in the literature	24
4.2.2 Normalized saturation/value ratio	27
4.2.3 Results	31
4.2.4 Conclusion	36
4.3 Context-aware cloud detection in sky images using conditional random fields	36
4.3.1 Proposed method	38
4.3.2 Results	45
4.3.3 Conclusion	48
5 MOTION ESTIMATION	50
5.1 Introduction	50

5.2	Choosing a motion estimation algorithm	50
5.3	Median filtering and motion vector transformation	53
5.4	Removing the horizon	56
6	ADVECTION AND POWER PREDICTION	61
6.1	Advection	61
6.2	Power forecasting	65
7	CONCLUSION AND FUTURE WORK	73
7.1	Conclusion	73
7.2	Future work	74
8	PUBLICATION LIST	76
	REFERENCES	77

LIST OF FIGURES

2.1 The sun-tracking camera at TEP solar test yard.	18
2.2 A sample image acquired using the sun-tracking camera.	19
2.3 Sky image captured using a) the GeoVision camera and b) a smart-phone camera.	20
 3.1 Image before normalization and after normalization	 22
4.1 Non-linear mapping between S/V ratio and NSV ratio.	25
4.2 Histograms in different feature spaces	28
4.3 Surface plots and corresponding segmentation results for NSV and NBR ratios	29
4.4 Segmentation results for sun-tracking camera images	33
4.5 Segmentation results for general sky images	34
4.6 Example illustrating the recall/precision tradeoff between NSV ratio and NBR ratio	35
4.7 Normalized Histograms in various feature spaces.	37
4.8 Comparison of the class conditional distribution and the Logistic function for NBR ratio	39
4.9 Figure illustrating how the interaction potential is defined.	41
4.10 Segmentation results for NSV and NBR ratios using adaptive thresholding scheme	46
4.11 Segmentation results demonstrating the importance of using spatial information in the CRF model	46
4.12 Segmentation results demonstrating the use of mean shift segmentation in dealing with noise	46
4.13 Segmentation results obtained using the CRF model for some images from the test database.	49
 5.1 Bar plot of PSNR values for the various motion estimation algorithms	 53
5.2 Bar plot of accuracy of advected images for the BME algorithms . .	54
5.3 Motion vectors before and after median filtering	58

5.4	Real-world reference frame	59
5.5	Motion vectors before and after motion vector transformation	60
6.1	Forward advection	63
6.2	Problems with forward advection	64
6.3	Five-minute-ahead power prediction plot 1	66
6.4	Five-minute-ahead power prediction plot 2	68
6.5	Ten and fifteen-minutes-ahead power prediction	69
6.6	Image demonstrating a reason for wrong forecast	71

LIST OF TABLES

4.1	Precision, Recall, and Accuracy (with 95% Confidence Intervals)	31
4.2	Accuracy, Precision and Recall Values with 99.9% Confidence Interval for Adaptive Thresholding Schemes (ATS) and CRF Based Method, Evaluated on Image Set C	44
4.3	Accuracy, Precision and Recall Values with 98% Confidence Interval for Fixed Thresholding Scheme and CRF Based Method, Evaluated on Image Set D	44
6.1	Root-Mean-Squared Error (Watts) for the Forecast Made on Four Days	70

ABSTRACT

Solar power utilization at the utility-scale is one of the grand challenges identified by the National Science Foundation. The intermittent output of solar power plants due to passing clouds and nighttime is a major problem. Intermittency limits the adoption of solar power by utility companies and industry because they require reliable, predictable power generation. Methods used to counter the problem of intermittency, such as interconnecting geographically dispersed photovoltaic (PV) systems, using dispatchable spinning reserves, energy storage and smart grids require accurate forecasting of PV power output for safe and efficient operation. For instance, utility operators require a ten-minute warning to bring spinning reserves online. Forecasting at multiple timescales is valuable for utility operators and plant owners. Day-ahead forecasts are needed to better determine pricing in the energy market. Hour-ahead and shorter time-scale forecasts are valuable for electric grid operators to schedule spinning reserves. Image analysis algorithms applied to images acquired using ground-based sky imaging (GBSI) systems have shown promise in making intra-hour power forecasts, but typically have limited reliability. This thesis aims at developing novel image analysis techniques to produce reliable intra-hour power forecasts. We describe the setup of a GBSI system at the Tucson Electric Power solar power test yard, which unlike most other such systems tracks the sun. The various modules involved in producing a power forecast, namely, cloud detection, motion estimation and cloud advection are described in detail. A new feature for enhancing the contrast between cloud and sky to aid cloud detection is proposed. A novel conditional random field based method, which attains a very high accuracy for cloud detection, is presented. Finally, the utility of the forecasts is demonstrated using standard metrics used in the literature.

CHAPTER 1

INTRODUCTION

Photovoltaic installations have grown exponentially over the last decade [1]. The growth was driven by various factors such as the plunging cost of solar power due to technological advancements, low carbon footprint of solar, increase in fossil fuel prices and government incentives in support of solar. In 2012, I had the opportunity to attend the keynote speech at the World Renewable Energy Forum by the Energy Secretary of the United States Department of Energy, Steven Chu. Dr. Chu started off by telling the audience how the environmental concerns led him to venture into clean energy. The Nobel laureate stressed the importance of not missing out on an opportunity to bring about a clean energy revolution in the United States, thus leading to a sustainable future. Turning to renewable sources of energy for sustainability is not confined to the US, but is a worldwide phenomenon today. Thanks to these concerns and the technological advancements, the trend of rising solar energy share is expected to continue, leading to an increase in solar energy penetration year after year.

However, the intermittent nature of solar power makes it difficult to integrate it into the electricity grid. Intermittency due to passing clouds is one of the major reasons for solar resource variability [2]. In fact this variability has limited the adoption of solar energy. Intermittency will become a bigger issue as the solar power penetration grows. Discovering new ways to compensate for intermittency will accelerate the adoption of solar power.

Several techniques exist to counter the problem of intermittency, including interconnecting geographically dispersed photovoltaic (PV) systems, using dispatchable spinning reserves, energy storage, and smart grids. All these methods require accurate forecasting of PV power output for safe and efficient operation. Industrial-scale loads that can be dynamically controlled by a solar-aware smart grid, such

as water-pumps or chillers, have a finite turn-on time; i.e., they start consuming power before they produce useful work. For this reason, the decision to turn on or off a load or source requires an accurate forecast of the timing and duration of a cloud event.

To comprehensively deal with intermittency, forecasting at different time scales is required [3]. While day-ahead forecasts are needed to better determine pricing in the energy market, hour-ahead and shorter time-scale forecasts are valuable for electric grid operators to schedule spinning reserves in order to offset mismatch between power production and demand. Previous studies have demonstrated that PV intermittency due to clouds can be forecast using numerical weather prediction (NWP) models, analysis of satellite imagery, and the use of a ground-based sky imaging (GBSI) systems [4–6]. NWP models are capable of forecasting clouds several days ahead; however, the cloud arrival time is only accurate to several hours [5, 7–9]. Analysis of cloud motion in satellite imagery performs better than NWP forecasts up to forecast horizons of 3–4 hours [4, 5]. But intra-hour forecasts from satellite images is challenging because of coarse spatial resolution. Hence, image analysis techniques applied to images acquired by GBSI systems have been explored for the task of intra-hour solar power forecasting [6, 10–12].

A complete framework for short-term power forecasts should ideally include a GBSI system to capture images and algorithms for cloud detection, motion estimation and cloud advection (predicting the future position of clouds based on current velocity). This work describes the development of such a framework. It includes a new GBSI system that we developed, which is unique in the respect that it tracks the sun. Such a setup has its own advantages like higher resolution around the sun (i.e., more pixels per cloud if we keep the clouds near the center of the image instead of near the border of the image) during most times of the day and ease of occluding the sun to prevent over-exposure. The imaging setup is described in chapter 2. The acquired images need some preprocessing to remove unwanted regions and to correct for intensity distortions, as described in chapter 3. The next step is to perform

cloud detection on these preprocessed images. Simple thresholding schemes (both fixed and adaptive thresholding) have been extensively applied for cloud detection. Various features have been suggested in the literature for these thresholding schemes, like the red to blue component ratio of an RGB image. However, those features did not produce satisfactory results on our images. Hence, we propose a new feature, the normalized saturation to value (NSV) ratio, which will enhance the contrast between sky and cloud and thus aid cloud detection. This feature especially suits adaptive thresholding schemes like cross entropy minimization. The proposed new feature is discussed in chapter 4. Though we were able to produce significant improvement in segmentation accuracy by using the new feature, simple thresholding schemes did not produce satisfactory results. Therefore, we introduce a new segmentation scheme based on conditional random fields (CRF). Taking the spatial context into consideration was the key to obtaining a good segmentation. The context-aware segmentation scheme based on a CRF model was able to produce highly accurate segmentation. This method too is discussed in chapter 4. In chapter 5, the algorithm used for motion estimation is presented. Block motion estimation, usually employed in video encoding, was used to estimate the cloud velocities. Once proper estimates of the velocities are obtained, cloud advection is performed. The process of estimating the future positions of the clouds using their current velocities is referred to as advection and is described in chapter 6. Since the estimated velocity vectors of different clouds in the same image are usually different, advecting the clouds forward using multiple motion vectors would produce holes in the advected image. This problem is overcome by following a method similar to the backward ray tracing approach in computer graphics. Chapter 6 includes a discussion on this topic. Finally, in chapter 7 the results for prediction of photovoltaic power plant output are presented.

CHAPTER 2

IMAGING SETUP

2.1 Overview

To understand how our imaging setup is different from others, it is appropriate to present a very brief overview of other existing ground-based sky imaging (GBSI) systems. One of the most well-known GBSI systems is the Whole Sky Imager (WSI) developed by the Scripps Institution of Oceanography at the University of California, San Diego [13]. This sophisticated instrument, with features including high quality optics with options to perform spectral filtering and temperature controlled charge-coupled devices (CCD), can be used to make high quality scientific observations. But due to the same reasons, it has a high cost which puts it out of reach of many researchers. The Total Sky Imager (TSI) developed by Yankee Environmental Systems, Inc. is another popular GBSI system which is commercially available for purchase. Apart from these, researchers around the world have developed their own cost-effective GBSI systems [14–17]. Almost all of these GBSI systems are stationary systems which use either an upward-looking camera fitted with a fish-eye lens or a camera looking down onto a curved mirror to obtain a complete view of the sky (i.e., field of view close to 180°). We, on the other hand, use a totally different imaging setup in which a fish-eye camera mounted on an equatorial mount is used to track the sun. The advantage of this setup is higher resolution around the sun in comparison to other imaging setups. Since we are mainly interested in solar power forecasting, a higher resolution around the sun is preferred. Since the position of the sun is fixed in every image, blocking the sun in order to avoid saturation of the image is easier in comparison to other imaging setups.

2.2 Sun tracker

The development of the sun-tracking camera imaging and image analysis system is a joint effort between the University of Arizona Photovoltaics (UAPV) laboratory in the Dept. of Physics, and the Signal and Image Laboratory (SaIL) in the Dept. of Electrical and Computer Engineering. The sun-tracking camera is located at the Tucson Electric Power (TEP) Solar Test Yard. The equatorial mount on which the camera is mounted, has two stepper motors that provides precise control over where the camera is pointed. One of the easiest ways to set up a tracker using this mount is to point one of the axes of the mount towards the celestial north and then rotate one of the stepper motors (the right ascension motor) at rate equal to the Earth's rate of rotation. This was the way the tracker was set up initially, with control software designed using LabView (National Instruments Corp.) for controlling the right ascension motor. But under this setup, the tracker needed regular manual adjustments because the declination angle of the sun changes over time. To overcome this issue, the old LabVIEW code was replaced by a new code. The new program made use of a solar positioning algorithm [18] to track the sun with greater accuracy. Given the location (latitude, longitude and altitude), date and time (with the UTC offset), the solar positioning algorithm outputs the solar azimuth and zenith angles. The LabVIEW code assumes that the camera is pointed at the sun when the program begins to execute. It then retrieves the solar zenith and azimuth angles at regular intervals (every 2 seconds) and calculates the change in those angles from the previous values. Based on the change in those values, the number of steps required to be taken by the stepper motors (both declination and right ascension motors) is then calculated. To calculate the number of steps required, careful calibration was previously done. The code then uses a LabJack interface (LabJack Corp.) to instruct the stepper motors to take the required number of steps. With this setup, the tracker is able to track the sun throughout the day. After sunset, the program computes the solar azimuth and zenith angles for sunrise of the next day. It then follows the same process; i.e., it calculates the

number of steps required to be taken to reset to the sunrise position and resets during night. Fig. 2.1 shows the sun-tracking camera at the TEP test yard.

The camera is placed in a weatherproof box with the lens of the camera facing a glass pane face. The glass pane is approximately 1 cm away from the lens. A circular rubber disc of approximate diameter 1 cm, glued on to the external surface of the glass pane, serves as an occlusion blocking direct sunlight from hitting the lens. As shown in Fig. 2.2 the area in the image that needs to be occluded prior to automated image analysis is much smaller compared to other approaches. The dark circle at the center of the image is the result of an image preprocessing step in which the area around the occlusion blocking the sun is removed. A better way to design an occlusion would be to suspend a spherical object (like a ball) further away in front of the camera. The further away the occluding object the better it would protect the camera from overexposure. The camera used is a GeoVision outdoor security camera. Movies of duration 5 minutes are recorded in MPEG-4 format at a frame rate of one frame/sec. Videos are recorded covering the entire day from approximately sunrise till sunset. Frames are then extracted from these movies for analysis. The image resolution is 1024×1280 pixels and 8 bits/pixel gray scale.

2.3 A note on image quality

The images captured using the GeoVision camera look dull in comparison to other general sky images (Fig. 2.3). This raises a question on the quality regarding the optics used within the imaging system. A higher quality camera may improve the quality of the images acquired. A DSLR camera was previously used in the sun-tracking system, but had to be replaced with the GeoVision outdoor security camera because two DSLR cameras broke down within a span of one month, probably due to the heat in Tucson. The camera in the sun tracker is placed within a weather-proof box. This box has a glass pane facing the camera lens. The purity

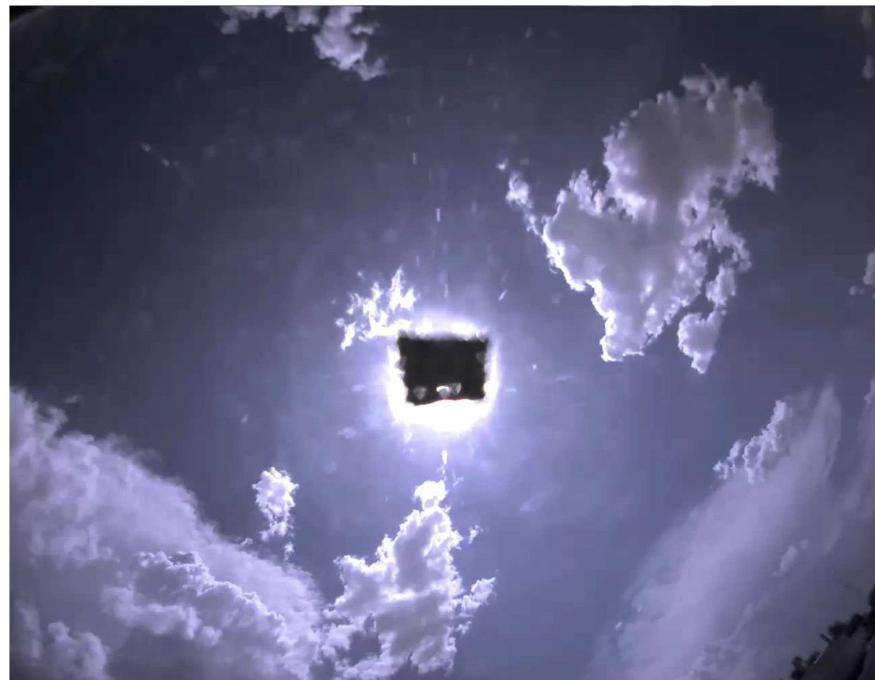
of the material of the glass pane may be another reason affecting the image quality. Using a glass pane made of high quality glass would help in dealing with the issue. Since the camera is enclosed within the box, the amount of light entering the imaging system is curtailed. This seems to be a third reason for the dull images produced by cameras when used in the sun-tracker setup. As a solution the protective box can be made of glass on all the sides to ensure that the camera receives enough light to capture better pictures. A fourth reason could be the gradual accumulation of dust particles on the glass pane with the passage of time. Regular cleaning of the protective box (at least once a day) will help in dealing with the dust accumulation problem.



Figure 2.1: The sun-tracking camera at TEP solar test yard.



Figure 2.2: A sample image acquired using the sun-tracking camera.



a)



b)

Figure 2.3: Sky image captured using a) the GeoVision camera and b) a smart-phone camera.

CHAPTER 3

IMAGE PREPROCESSING

The acquired images need to be corrected for intensity distortion before they can be processed. Fig. 3.1 shows the results of performing image normalization. Fig. 3.1a shows that the intensity or brightness decreases monotonically towards the corners. To compensate for this, we use a clear sky library of images. The idea is to infer the degree of intensity normalization required at each pixel position by using the clear sky image as a reference. The normalization process involves the following steps:

1. Find a clear sky model for the image to be processed from the clear sky library. The clear sky model chosen is a clear sky image captured almost at the same time as the image to be processed and is from the nearest possible date. For testing purposes, we have manually chosen the clear sky model for each image.
2. Convert the image from RGB to HSV color space. For a clear sky image, the hue and saturation components stay constant but the value drops as we move away from the center. Hence, compensating for the decrease in value component removes the artifact.
3. Pick a reference intensity value from the clear sky model. The term intensity is used here to refer to the V component of the image in HSV space. The mean intensity value of all pixels lying within an annular shaped region around the center is chosen as the reference intensity, V_{Ref} . The inner radius was chosen to be 300 pixels, and the outer radius was fixed at 325 pixels. These values were chosen to avoid regions very close to the center which are over-exposed due to the presence of the sun and also to avoid regions too far away from the center which might be too dark.



Figure 3.1: Image a) before normalization and b) after normalization.

4. Capture the variation in the value component of the clear sky image in the HSV space. Specifically we fit a polynomial surface of degree 4 in each direction to the value component by minimizing the mean squared error. Since we are operating in a discrete space, the fitted surface can be represented as a 2D matrix $\hat{V}(i,j)$.
5. Derive a normalization factor for each pixel. V_{Ref} is the uniform reference intensity that we expect the whole sky region to have, so

$$\lambda(i,j) = V_{Ref}/\hat{V}(i,j)$$

is the scaling factor required to normalize each pixel.

6. Multiply the image to be normalized with $\lambda(i,j)$ to obtain the normalized image.

Apart from this, a circular region at the center of the image (of radius 100 pixels) and the regions below the horizon are removed in the preprocessing step. In order to determine the location of the horizon in the image, motion vector transformation equations (described in chapter 5) are used. We also compensate for the fish eye lens effect in this process. This is important as the horizon is not a straight line but a curve. Further details are provided in chapter 5.

CHAPTER 4

CLOUD DETECTION

4.1 Introduction

The Earth’s climate and energy balance is influenced to a great extent by clouds. Clouds are also responsible for the largest uncertainties in climate change predictions [19]. Hence, making reliable cloud observations is an important problem which is not confined to just solar power forecasting research. Historically, human observers were employed to record cloud observations at weather stations. But due to reliability and cost issues, the use of automated systems was explored for this task. Long et al. [15] reported that ground-based sky imaging (GBSI) systems together with automated image processing algorithms could complement or even replace human observers for the task of recording cloud observations. In the next section we first present a contrast-enhancing ratio which is suitable for adaptive thresholding schemes. Though it improves cloud detection accuracy significantly, the accuracy levels are still below desired levels. Therefore, in the subsequent section we introduce a conditional random field (CRF) based method for cloud detection which gives very high accuracy.

4.2 A contrast-enhancing feature for cloud detection

As mentioned in chapter 2, research groups around the world employ different camera systems to make cloud observations. Images of the same scene captured by two different camera systems can vary in details. A general feature used for cloud detection should be able to produce sufficient and consistent contrast between cloud and sky pixels for images captured using different imaging systems. However, we found that the various features proposed in the literature are unable

to produce satisfactory results for cloud detection for our images. The main reason is the insufficient contrast between cloud and sky when those features were used for cloud detection. Therefore, we propose a new feature that we call the normalized saturation/value (NSV) ratio, which performs well on a broad range of images.

In the following sections we discuss the various features and algorithms proposed in the literature for cloud detection, define the new feature proposed and explain the reasoning behind the choice, and provide results and details regarding the data set used for testing the performance of different features for cloud detection.

4.2.1 Cloud detection methods in the literature

Any cloud detection scheme has two important components: the feature used to distinguish between cloud and sky pixels and the detection algorithm that uses this feature. The basic philosophy behind most schemes is to devise a feature that discriminates between cloud and sky pixels and then threshold this feature to detect cloud pixels. When fixed thresholds are used, the thresholds are empirically determined using training data. For instance, empirically determined thresholds for the red to blue component (RB) ratio [13, 20] have been applied extensively to distinguish cloud pixels from sky pixels. Souza-Echer et al. [17] used a fixed threshold on the saturation value of the pixels in the IHS color space to detect clouds. Neto et al. [21] used multidimensional Euclidean geometric distance (EGD) and Bayesian methods to classify sky and cloud patterns based on the observation that sky and cloud patterns occupy different loci on the RGB color space. Recently, Ghonima et al. [10] measured cloud opacity as the difference in RB ratio between the pixel to be classified and the corresponding pixel in the clear sky library. They described a method to compensate for the variations in aerosol optical depth by using a haze correction factor derived with the help of a clear sky library.

Yamashita et al. [22] and Li et al. [23] proposed the use of normalized blue to red component (NBR) ratio (Yamashita et al. called it sky index) for cloud

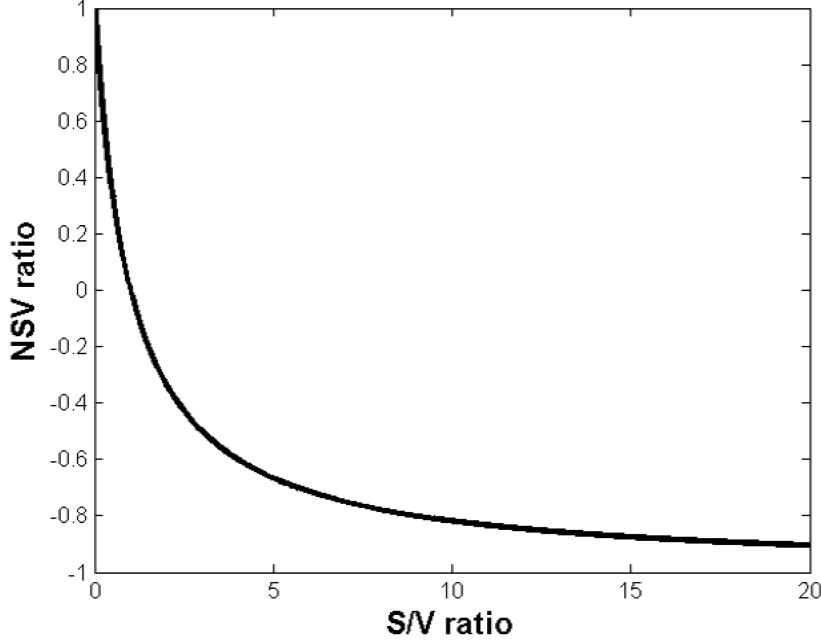


Figure 4.1: Non-linear mapping between S/V ratio and NSV ratio.

classification and detection. NBR ratio is defined as

$$\beta_{\text{NBR}} = (B - R)/(B + R)$$

where B and R represent the blue and red component intensities of the pixel, respectively. Li et al. also proposed a hybrid algorithm combining fixed and adaptive thresholding schemes. If the standard deviation of NBR ratio is below a threshold a fixed threshold determined statistically from training data is used for cloud detection. Otherwise, an adaptive thresholding algorithm based on cross-entropy minimization [24] is applied to obtain the threshold for the image.

The cross-entropy minimization used by Li et al. [23] depends on two conditions being satisfied in order to perform well. First, there should be minimal overlap (preferably no overlap) between the sky and cloud pixels in the feature space, as this is a necessary condition for the existence of an appropriate global threshold. Second, the intra-class variance should be much less in comparison to the inter-class variance. To understand this condition, we need to look a little deeper into

the cross-entropy minimization technique. When Li and Lee [24] proposed cross-entropy minimization, they treated it as an image distribution reconstruction problem. The image itself is represented as a function $f : N \times N \rightarrow G$, where N is the set of non-negative integers and G is the set of integer gray levels. The aim is to find a segmentation function $g : N \times N \rightarrow S$, where $S = \{\mu_1, \mu_2\} \in \Re^+ \times \Re^+$ where \Re^+ is the set of real positive numbers. The segmented image is constructed as

$$g(x, y) = \begin{cases} \mu_1, & f(x, y) < t \\ \mu_2, & f(x, y) \geq t \end{cases}$$

Here, t is the threshold and μ_1 and μ_2 are the levels assigned to pixels belonging to the two classes (i.e., cloud and sky). The problem can be defined as finding an optimal g (or finding an optimal set of parameters μ_1, μ_2, t) given some constraint. The constraint applied is that of intensity conservation, i.e., in both classes, the total feature value (i.e., the sum of feature values of pixels in each class) in the reconstructed image and the observed image should be identical. The basic idea is to find a g that leads to a reconstructed distribution of feature values that closely follows the observed distribution of feature values. As Li and Lee show in their paper, Otsu's algorithm minimizes the mean squared error between f and g , whereas their method minimizes cross-entropy to obtain the best fit. cross-entropy is a measure of the information theoretic distance between two distributions. If P and Q are two discrete distributions, say $P = \{p_1, p_2, p_3, \dots, p_N\}$ and $Q = \{q_1, q_2, q_3, \dots, q_N\}$, then the cross-entropy between P and Q is given by

$$D(Q, P) = \sum_{k=1}^N q_k \log_2 \left(\frac{q_k}{p_k} \right)$$

The process of finding a threshold is equivalent to fitting a binary surface to the data distribution, where the best fit is found by minimizing either the mean squared error or cross-entropy. If the intra-class variance is not much less than the inter-class variance, the fitted surface may not provide a good segmentation. In the next section, we define the NSV ratio and explain why the NSV ratio performs better on our images based on the two conditions mentioned above.

4.2.2 Normalized saturation/value ratio

The NSV ratio is defined as

$$\gamma_{\text{NSV}} = (1 - \lambda)/(1 + \lambda)$$

where

$$\lambda = S/V$$

V is the value component and S is the saturation component in the HSV color space. An RGB image can be converted to HSV color space by using the formulas given in [25]. While hue in the HSV color space defines the color itself, saturation represents the degree of difference between a color and gray and value is an indication of the brightness of the color. Clouds, which consist of water and ice particles, can scatter different wavelengths of light and hence appear white. Gas molecules in clear sky predominantly scatter blue wavelengths and hence they appear blue. In terms of saturation, clear sky is highly saturated with a minimal amount of white, whereas clouds have low saturation. On the other hand, clouds often appear brighter relative to sky and hence have a larger V component in the HSV space compared to sky. Therefore the S/V ratio enhances the contrast between cloud and sky pixels. The normalization step actually performs a nonlinear compression of the S/V ratio values which results in a compact representation. Fig. 4.1 shows the non-linear mapping between S/V ratio and NSV ratio. As Li et al. pointed out, the normalization step helps in dealing with outliers. For instance, the S/V ratio for sky pixels, which usually is less than 2, at times shoots up to values greater than 15. These outliers can lead to a higher intra-class variance, which could negatively impact the cross-entropy minimization process. The normalization step helps in keeping the intra-class variance low.

As stated above, the first condition that needs to be satisfied for any global thresholding scheme to perform well is that there should be minimal overlap between the two classes in the feature space in which the algorithm operates. As Fig. 4.2 shows, there is considerable overlap between cloud and sky in feature space

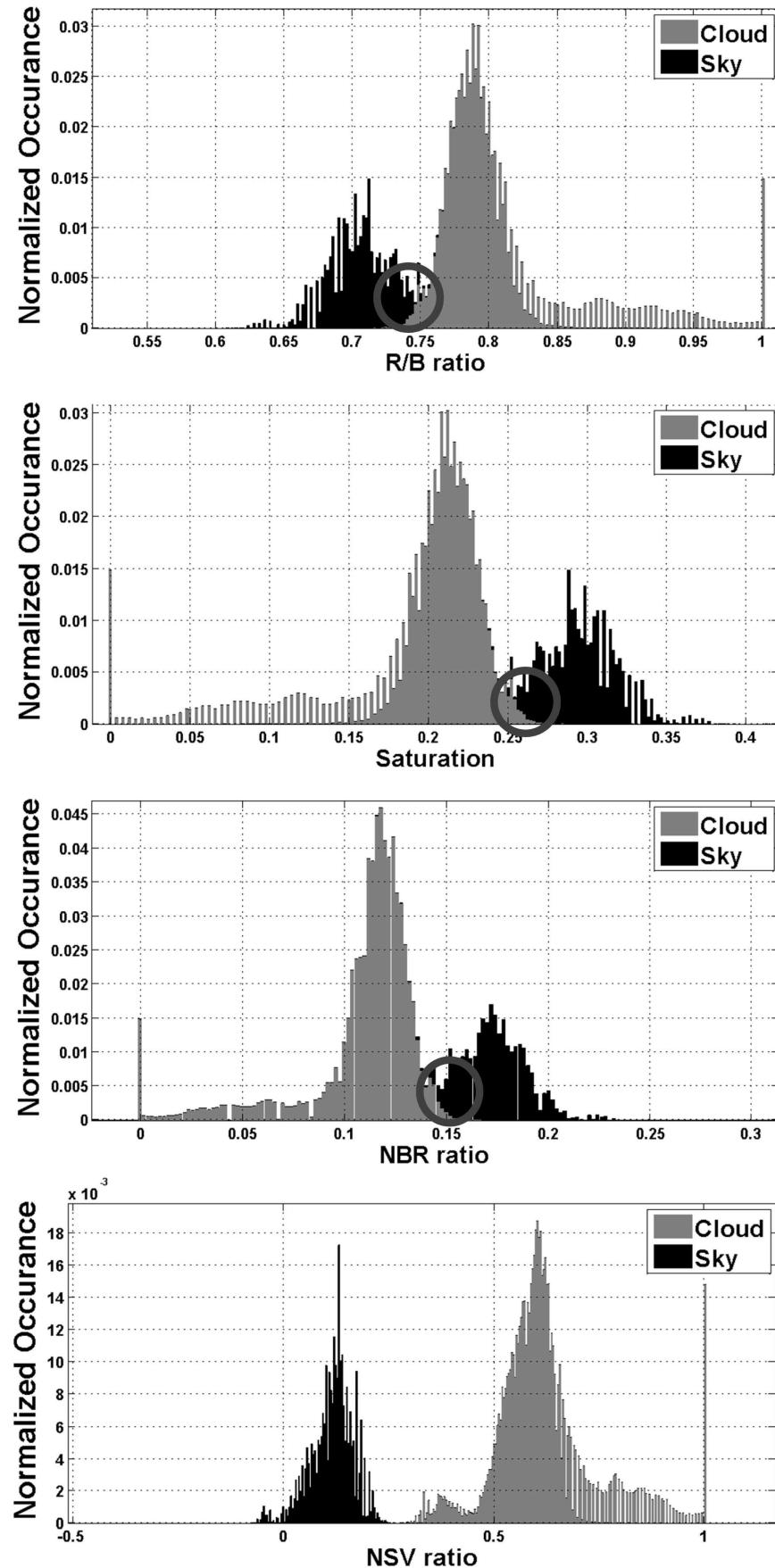


Figure 4.2: Histograms in different feature spaces for the image from Fig. 2.2.

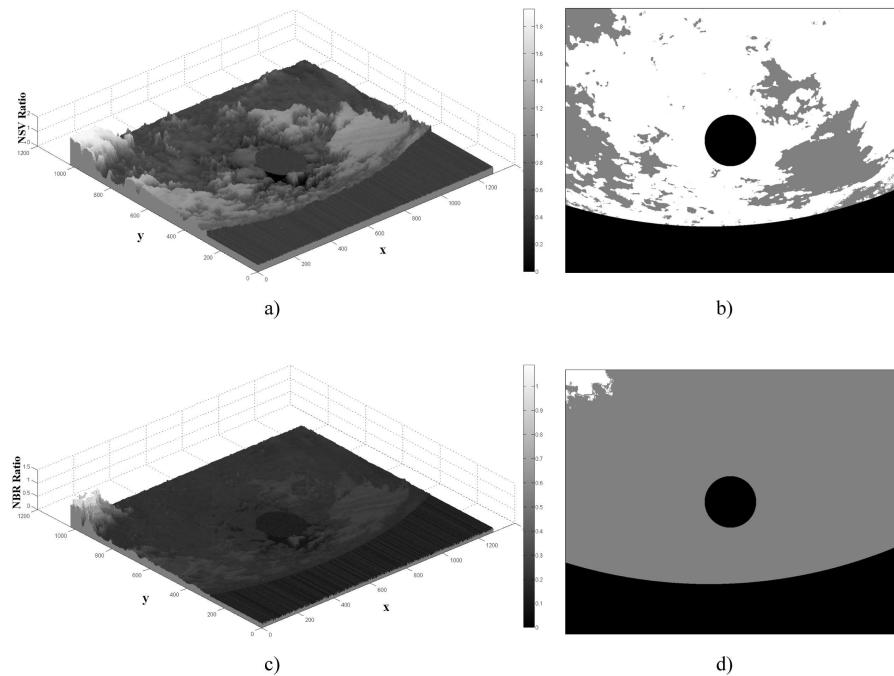


Figure 4.3: Surface plot of a) 1 - NSV ratio and (b) the corresponding segmentation (white color represents clouds and gray represents sky) for the image from Fig. 2.2. Surface plot of c) NBR ratio and d) the corresponding segmentation.

when the RB ratio or saturation in HSV color space or NBR ratio is used as the feature. The overlapping regions are marked with circles. On the other hand, the NSV ratio provides a stronger separation between the two classes, which indicates the existence of a fixed threshold that can separate the two classes well. The second condition necessary for the cross-entropy minimization algorithm to produce good results was that the inter-class variance should be much higher in comparison to the intra-class variance. As shown in Fig. 2.2, the image has a dark patch of sky on the top left corner due to overcast conditions. This is not an isolated occurrence, and different regions belonging to the same class can have different intensities especially during overcast conditions. In the surface plot of NBR ratio (Fig. 4.3), this causes the formation of a high peak around that region, which is distinctly separate from the rest of the surface. As a result of this, when the algorithm tries to minimize the (binary surface) fit error (cross-entropy or mean squared error), it results in a wrong segmentation. I.e., in this case the best fit does not correspond to the best segmentation. In the case of NSV ratio, the two classes occupy ranges of values which are separate enough to produce a good segmentation.

Though NSV ratio serves as a good contrast-enhancing feature that can be employed in adaptive thresholding schemes such as cross-entropy minimization, it has some drawbacks that need to be discussed. One such drawback is that it cannot be used in a fixed thresholding scheme. In Li's hybrid scheme, fixed thresholding is applied when there is just one class of object in the image, i.e., images which represent either clear sky or completely overcast conditions. Clear sky images are much brighter than completely cloudy images. Since the V component of the HSV space is involved, the NSV ratio depends on the brightness of the images. Although the NSV ratio of clouds is usually higher than that of sky, since the clear sky images are brighter, the range of NSV ratio values occupied by clear sky images and completely cloudy images overlap. Hence, in the case of NSV ratio, a global fixed threshold usually does not exist. Another drawback of the NSV ratio is that it cannot be used for cloud detection in images taken during times close to sunrise or sunset. The sky in such images would look less saturated, so the NSV ratio

Table 4.1: Precision, Recall, and Accuracy (with 95% Confidence Intervals)

		NSV Ratio	NBR Ratio
Set A	Precision	0.7839 ± 0.088	0.7045 ± 0.117
	Recall	0.9324 ± 0.046	0.9237 ± 0.080
	Accuracy	0.8250 ± 0.053	0.7127 ± 0.102
Set B	Precision	0.9080 ± 0.085	0.9762 ± 0.041
	Recall	0.8174 ± 0.072	0.7467 ± 0.122
	Accuracy	0.8434 ± 0.066	0.8508 ± 0.068
Overall	Precision	0.8460 ± 0.093	0.8403 ± 0.117
	Recall	0.8749 ± 0.068	0.8353 ± 0.114
	Accuracy	0.8342 ± 0.059	0.7817 ± 0.094

would not be able to produce the contrast (between cloud and sky pixels) required to apply adaptive thresholding. However, for solar power forecasting, times close to sunrise and sunset are not of much interest; hence, this is not a serious drawback.

4.2.3 Results

The minimum cross-entropy thresholding scheme is applied to segment the images. The adaptive thresholding scheme is applied to both NSV and NBR ratio images to compare performance. Multiple datasets of images are employed. A total of 28 images were used for performance evaluation. Among these, 14 images were acquired using our sun tracking camera, which we hereafter refer to as Set A. Images obtained using different camera systems, including images chosen randomly from the Internet, constitute the rest of the dataset, which we refer to as Set B. All the major cloud types (cirriform, stratiform and cumuliform) are represented in the both the datasets. However, clear sky images and completely cloudy images are not included in the datasets because, as discussed in section 4.3, fixed thresholds usually does not exist in the case of NSV ratio.

Accuracy, precision and recall computed based on the confusion matrix are employed as metrics for performance evaluation [26]. The metrics are defined as

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

where TP (true positive) is the number of cloud pixels classified correctly, TN (true negative) is the number of sky pixels classified correctly, FP (false positive) is the number of cloud pixels that got misclassified as sky pixels and FN (false negative) is the number of sky pixels which got misclassified as cloud pixels.

Table 1 provides a comparison of the performance of NSV ratio and NBR ratio on the two image sets. Accuracy, precision and recall with their 95% confidence intervals are reported. The confidence intervals were calculated using a t-distribution. For Set A, though the point estimate of accuracy is significantly higher for NSV than for NBR, the confidence intervals overlap at 95% confidence level. However, at 85% confidence level, the accuracy of NSV (0.8250 ± 0.037) and NBR (0.7127 ± 0.072) do not overlap. Hence, we can only be 85% confident that NSV is better than NBR for images in Set A. However, for Set B the point estimate of accuracy for NBR is slightly better than for NSV. But since the 95% confidence intervals overlap substantially, there is no clear winner for Set B.

It is noteworthy that the recall values for the NSV usually appear greater than for NBR. This is evident in Fig. 4.6, where we can see that NSV detects more clouds than NBR. In solar power forecasting the cost of missing (not detecting) a cloud, i.e. a false negative, is likely to be higher than the cost of a false positive. Hence, a higher value of Recall would be preferred.

As stated before, the NSV ratio has some drawbacks because of its dependence on the brightness/intensity of the image. But at the same time it has proven to be a very useful feature suitable for adaptive thresholding schemes. Similarly there is a lot of useful information in the blue and red components of the RGB image,

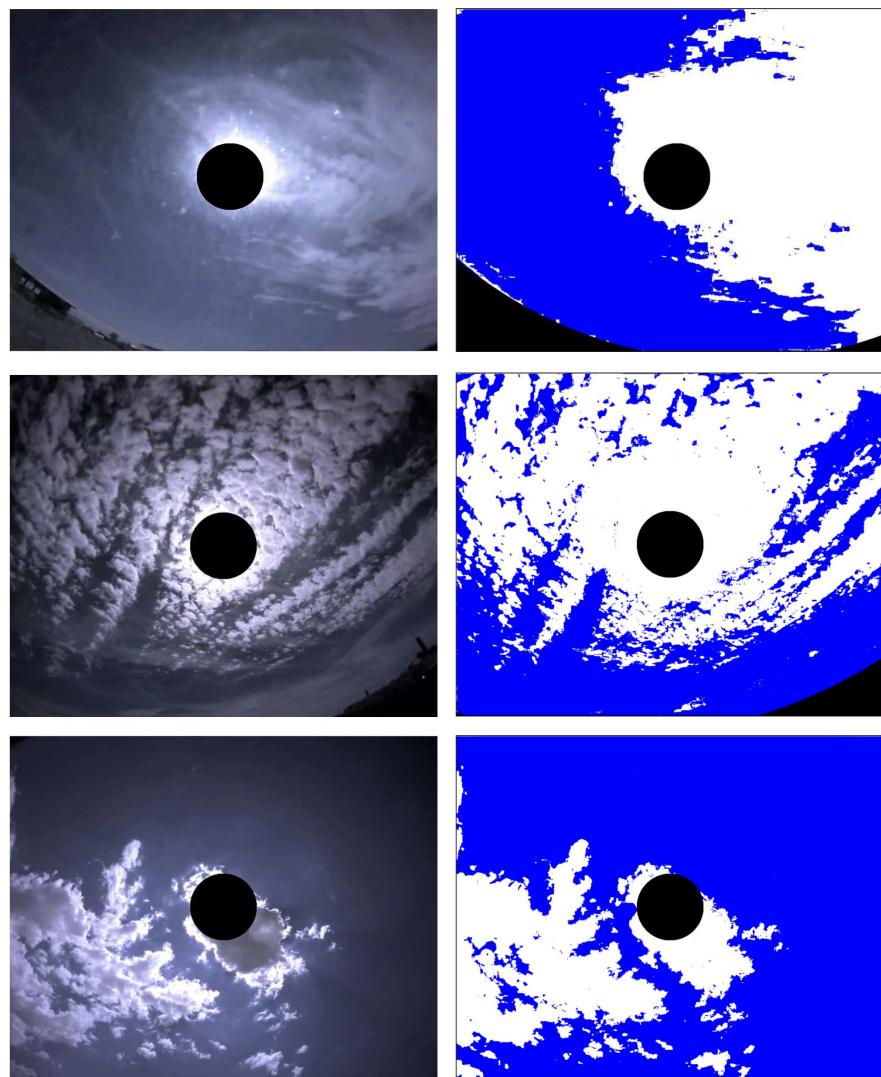


Figure 4.4: Segmentation results using NSV ratio for some images captured using the sun tracking camera.

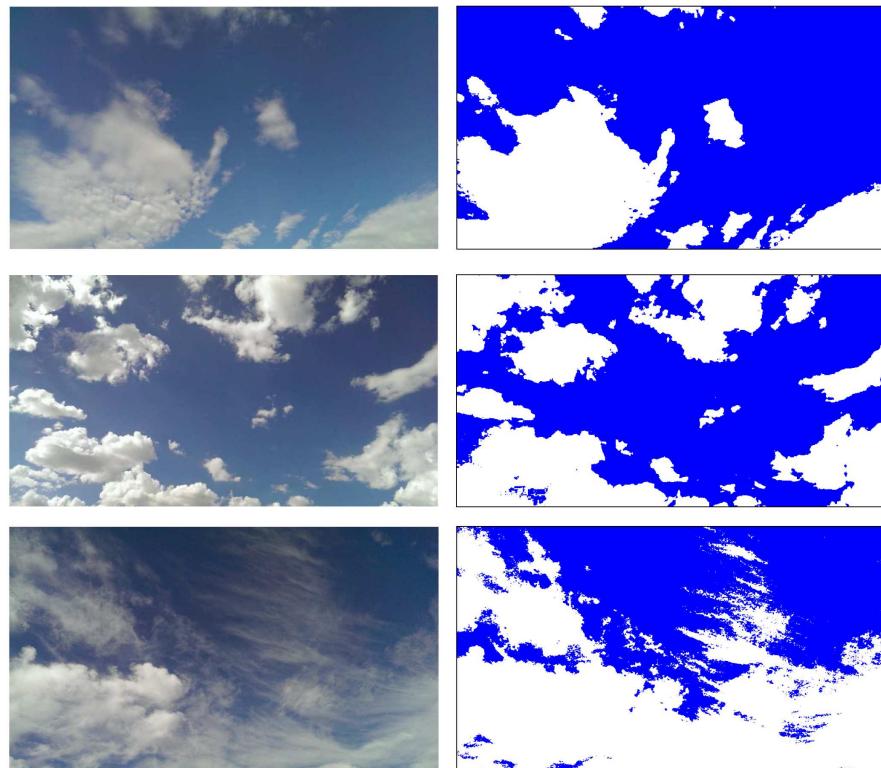


Figure 4.5: Example sky images (left) and the corresponding segmentation using NSV ratio (right).

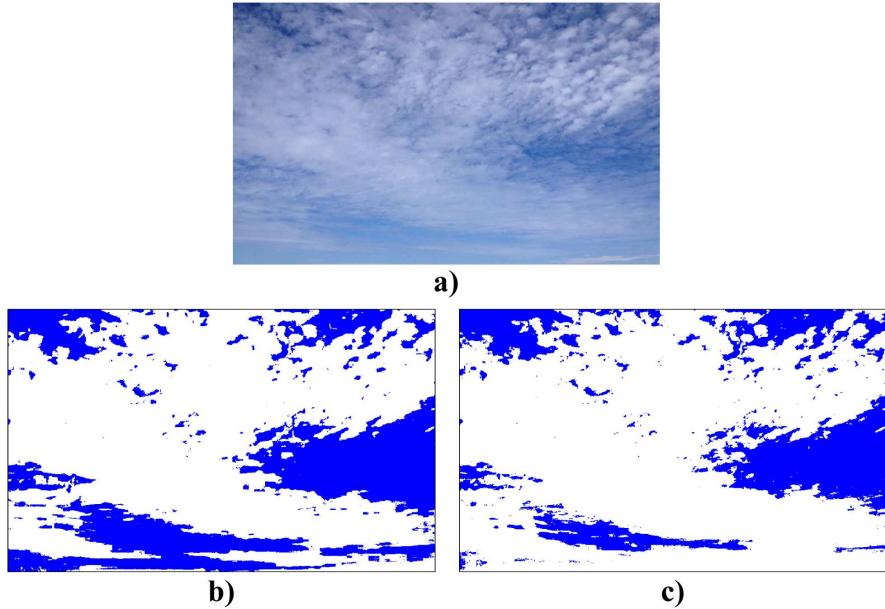


Figure 4.6: An example illustrating the recall/precision tradeoff between NSV ratio and NBR ratio. a) Original image, b) segmentation result (NBR ratio), c) segmentation result (NSV ratio).

though it alone might not be sufficient to classify a broad range of images. Hence, it would be a good idea to explore methods using both NBR and NSV ratios for classification.

Another problem that needs to be addressed is the detection of clouds near the sun. As can be seen in Fig. 4.4, the region around the sun looks overexposed and hence is detected as cloud no matter whether we use NBR or NSV as the feature. One way to deal with this issue is to use motion information, which can be derived by comparing adjacent frames in a sequence of images (explained in the following chapter). The basic idea is that the results of the cloud detection algorithm and the results of motion estimation should concur. This is, however, not applicable in cases where there is just a single image and not a sequence. But for our application, i.e. solar power prediction, cloud detection near the sun is not a major issue because the effect of overexposure is usually confined to a radius of 150 pixels around the sun at the center of the image. We care more about the

clouds away from the sun, and in those regions there is minimal or no effect due to the sun.

4.2.4 Conclusion

We have introduced a new feature which performs well on a broader class of images acquired using different camera systems. The useful properties of this feature were demonstrated, and its drawbacks were pointed out. A comparison of the performance of the NSV ratio with the well-known NBR ratio was provided. The NSV ratio provides improved accuracy and significantly higher recall than the NBR ratio for the images in Set A captured from a single camera, while still achieving comparable performance for the more general collection of images in Set B.

4.3 Context-aware cloud detection in sky images using conditional random fields

In the previous section we showed that the NSV ratio could serve as a contrast enhancing feature suitable for adaptive thresholding. As discussed in the previous section and [23] and as depicted in the Fig. 4.7 there is considerable overlap between cloud and sky pixels in the various feature spaces. The histograms shown in Fig. 4.7 correspond to a set of images used as a training set and not a single image. These histograms depict the problem associated with a fixed thresholding scheme. Whatever fixed threshold is picked, it will inevitably misclassify some pixels.

On the other hand in the case of adaptive thresholding it is only necessary to ensure that the classes do not overlap in the feature space on an image by image basis. In previous section though we showed that the NSV ratio provides a strong separation between the two classes (cloud and sky) in the feature space for many images, we also mentioned that this is not always true. The dependency of NSV ratio on the value component of the pixels (in the HSV space) causes

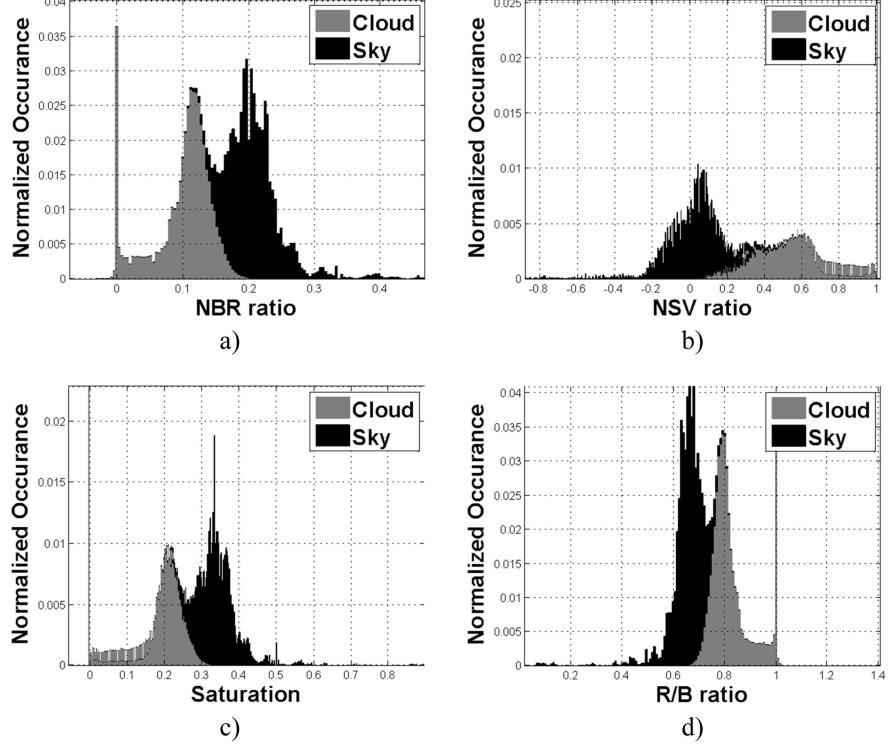


Figure 4.7: Normalized Histograms in various feature spaces.

the NSV ratio for some dark clouds to overlap with that of sky pixels. We also showed that an adaptive thresholding scheme such as cross-entropy minimization might pick a wrong threshold if the intra-class variance is high, which is a likely scenario in the case of sky images. Minimum cross-entropy thresholding and Otsu's thresholding can be viewed as a surface fitting problem, and the best fit may not always correspond to the best segmentation as we showed in the previous section.

To overcome these issues, we propose a cloud detection scheme based on the CRF framework. We explain how this model addresses each of the issues mentioned above. Quantitative and qualitative results are presented, and possible extensions to this work are discussed towards the end.

4.3.1 Proposed method

Probabilistic graphical models like Markov random fields (MRF) and conditional random fields (CRF) have been extensively applied for the task of contextual image segmentation [27–33]. Li et al. [30] proposed an MRF model for detecting thin clouds. Though MRF models have proven to be successful for computer vision tasks like segmentation and image denoising, they have some drawbacks. Being a generative model, MRF based approaches model the joint density $P(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = \{x_i\}_{i \in S}$ represents the input image data, where S is the set of all sites. A site can correspond to a single pixel or a group of pixels (region). And $\mathbf{y} = \{y_i\}_{i \in S}$, $y_i \in L$ represents the class labels corresponding to the sites (\mathbf{y} is often referred to as a label configuration) and L is the set of all possible labels/classes. In the MAP (maximum a posteriori) MRF framework, Bayes’ rule is employed to derive the posterior probability of the labels given the data:

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})} = \frac{P(\mathbf{x}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{x})} \propto P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$$

In many cases such as in the case of cloud detection \mathbf{x} is always observed (i.e. during training as well as while performing inference or classifying) and hence it is not necessary to model $P(\mathbf{x})$ which might not be a simple function. Another drawback of MRF modeling is that often for tractability the likelihood $P(\mathbf{x}|\mathbf{y})$ is assumed to have a factorized form, i.e. $P(\mathbf{x}|\mathbf{y}) = \prod_{i \in S} p(x_i|y_i)$. This assumption is too restrictive as complex relationships usually exist between data at neighboring sites. And finally the MRF model imposes label consistency (modeled as the prior $P(\mathbf{y})$) uniformly over the entire image without taking into consideration the observed data.

CRF models, on the other hand, are discriminative models which directly model $P(\mathbf{y}|\mathbf{x})$, the probability of a label configuration given the data. The CRF model therefore does not attempt to model the distribution of input data $P(\mathbf{x})$. The conditional independence assumption of the likelihood model given the labels is relaxed in the case of CRFs. CRFs also allow us to model data-dependent label

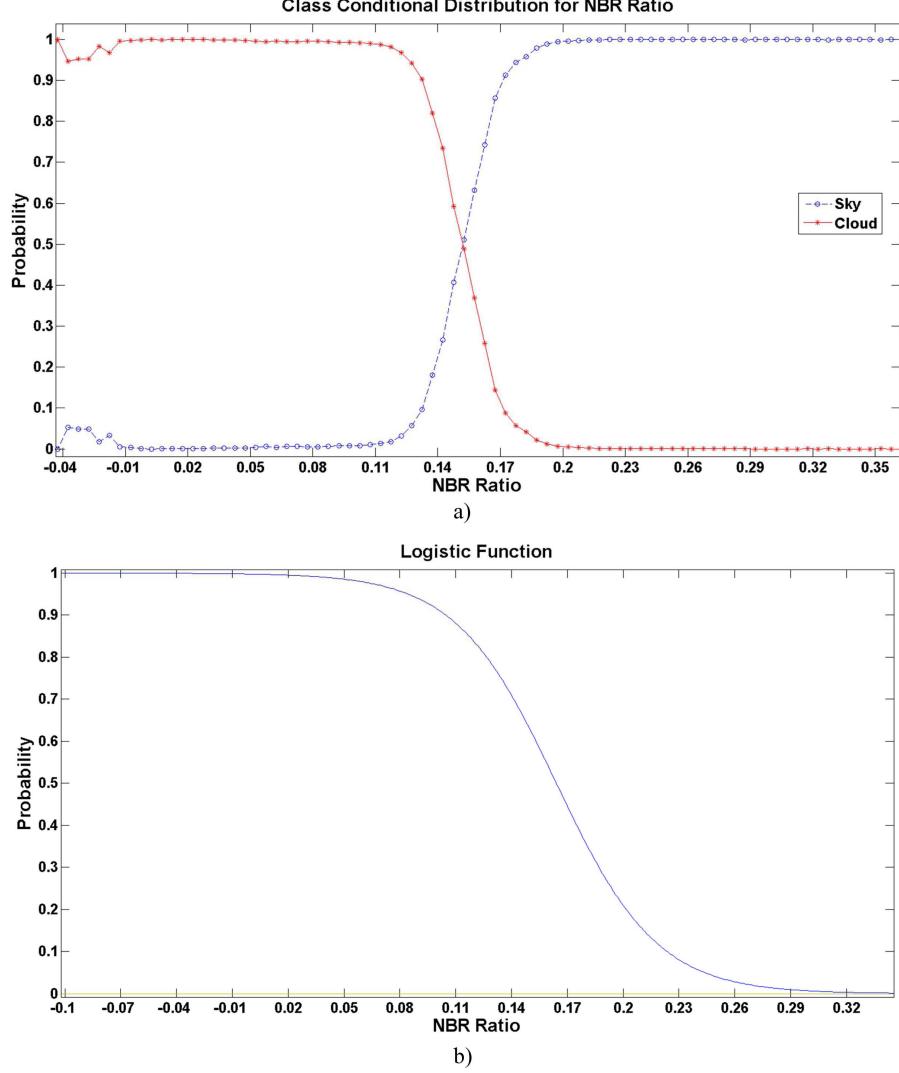


Figure 4.8: a) Class conditional distribution for NBR ratio obtained using training data and b) Logistic function, ψ , with parameters $\alpha_0 = 6.072$ and $\alpha_1 = -37.001$ estimated using training data.

interactions as the clique potentials, or the interaction potentials are functions of both the labels and the data. This property, as we will see, plays a crucial role in cloud detection. Due to the above-mentioned factors, CFRs outperform MRF models for various computer vision tasks including image segmentation [32, 34].

CRF Model

As mentioned before, we combine a discriminative classifier and a higher order clique potential in a CRF framework. Again $\mathbf{x} = \{x_i\}_{i \in S}$ represents the input image data, where S is the set of all sites. We divide the image into homogeneous regions using mean shift clustering [35], hence each region defines a site. The labels corresponding to the sites are $\mathbf{y} = \{y_i\}_{i \in S}$, $y_i \in L$ with $L = \{0, 1\}$ where 0 is the label for sky and 1 is the label for cloud. Following the discriminative random field approach [32], assuming \mathbf{y} to obey the Markov property conditioned on the data \mathbf{x} (i.e., $P(y_i|\mathbf{x}, y_{S-\{i\}}) = P(y_i|\mathbf{x}, y_{N_i})$, where N_i represents the neighborhood of site i), and using the Hammersley Clifford theorem [36] the posterior distribution of the label configuration is defined as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{i \in S} \psi(x_i) + \beta \sum_{i \in S} \sum_{j \in N_i} \phi(\mathbf{x}, y_i, y_j) \right) \quad (4.1)$$

where

$$\psi(x_i) = \frac{\exp(\alpha_0 + \alpha_1 K_i)}{1 + \exp(\alpha_0 + \alpha_1 K_i)} \quad (4.2)$$

$$\phi(\mathbf{x}, y_i, y_j) = \begin{cases} (V_s - V_i), & V_s = \bar{V}_j | y_j = 0 \text{ when } y_i = 0 \\ (V_i - V_c), & V_c = \bar{V}_j | y_j = 1 \text{ when } y_i = 1 \end{cases} \quad (4.3)$$

and

$$Z = \sum_{\mathbf{y}} \exp \left(\sum_{i \in S} \psi(x_i) + \beta \sum_{i \in S} \sum_{j \in N_i} \phi(\mathbf{x}, y_i, y_j) \right) \quad (4.4)$$

K_i and V_i represent the NBR ratio and NSV ratio at site i , respectively; α_0 , α_1 and β are the parameters that need to be estimated. Z is the partition function and serves as a normalization factor. It is computed by summing the model over all possible configurations of \mathbf{y} .

Both the NBR and NSV ratios are utilized in our CRF model. As can be seen in Fig. 4.8, the class conditional density for cloud is almost one for all NBR

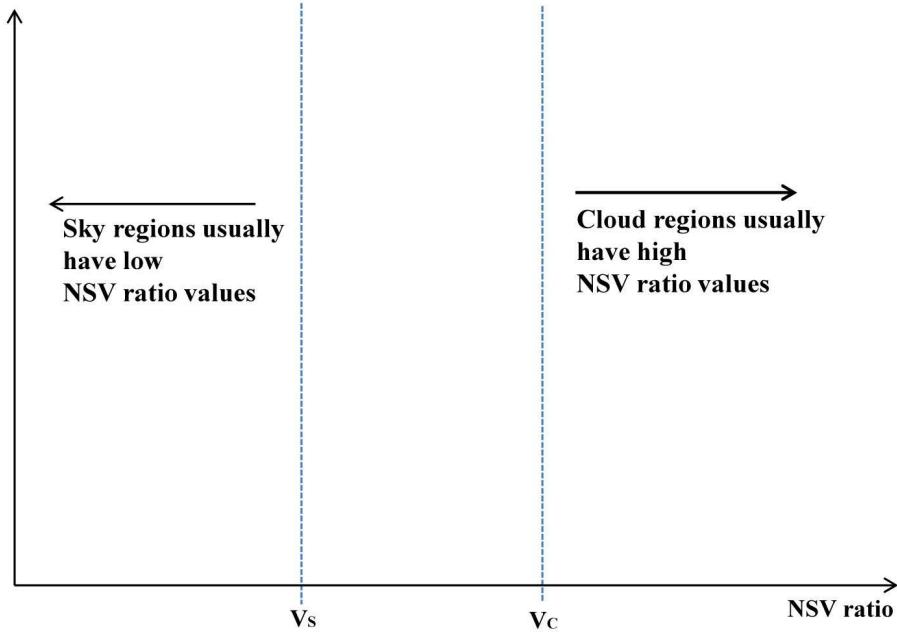


Figure 4.9: Figure illustrating how the interaction potential is defined.

values below 0.11, and it is almost zero for all values above 0.2. It is mainly in the range [0.11, 0.20] that reliance on NBR alone would result in misclassification, and outside this range we can make cloud detection decisions almost unambiguously. Thus, there are a considerable number of regions which we can classify as cloud or sky with very high confidence. And for the remaining regions where there are ambiguities, we can look at the spatial context to make decisions. As we showed in the previous section, the NSV ratio exhibits enhanced contrast between cloud and sky, so it could serve as an effective, local feature for testing spatial consistency. But due to the dependency of this ratio on the value component of the HSV space of the image, it makes more sense to use the NBR ratio as a global feature.

This idea can be modeled very well using a CRF framework. The function ψ is nothing but a logistic regression classifier. This classifier gives a well calibrated probability value indicating whether the site under consideration is cloud or sky just based on the NBR ratio at that site. As shown in Fig. 4.8, the logistic function derived from the training data is a smoother version of the class conditional density for the cloud. In the DRF framework terminology [32] the logistic function is the

association potential. Now in order to take the spatial context into account, we define the interaction potential function ϕ . Before defining the interaction potential we need to define the neighborhood N_i for the site i . In our case, we define N_i as all regions which are within a 200-pixel radius of the centroid of site i (the site i is not part of the neighborhood). It is not necessary that a region be entirely within this radius to be considered a neighbor; even if it is partially within this range, it will be considered a neighbor. The interaction potential can now be understood with the help of Fig. 4.9. V_S and V_C are the average NSV values of sky neighbors and cloud neighbors, respectively. Clouds usually have higher NSV values in comparison to sky regions, hence $V_C > V_S$. The interaction potential has been defined in such a manner that V_S and V_C serve as reference points. I.e., the farther the NSV ratio of a site is to the left of V_S the higher the probability that it is a sky region, and the farther it is to the right of V_C the higher the probability that it is cloud. If the NSV ratio of a site is in between V_S and V_C , then the interaction potential is negative, and the magnitude will depend on its proximity to the two reference points.

As stated earlier, the mean shift clustering algorithm [35] is used to form homogeneous regions in an image. The parameters including range bandwidth, color bandwidth, minimum number of pixels in a region, etc., are set manually to produce the best qualitative results. Defining the CRF model over regions instead of pixels has two main advantages. First, it would help in speeding up the inference algorithm (the algorithm used to find the most probable labeling configuration given a new image) and secondly it will help combat noise as we will demonstrate in the results section.

Parameter Estimation and Inference

The parameters in CRF are usually estimated using maximum likelihood estimation (MLE) [34,37,38]. MLE finds the parameters that maximize the conditional likelihood of the true labels given the training data. Stochastic gradient descent (SGD) is usually employed to find the MLE estimates [34]. In SGD the partial

derivatives with respect to each parameter are evaluated. These partial derivatives involve the computation of the expected value of the feature function (for e.g. interaction potential in our case) over all possible label configurations [37, 38]. Since this computation is intractable, the expectation is approximated by coming up with an approximation for $P(\mathbf{y}|\mathbf{x}; \theta)$. Here, θ represents the parameter set $\{\alpha_0, \alpha_1, \beta\}$. Note that $P(\mathbf{y}|\mathbf{x}; \theta)$ is the probability density over all possible label configurations given an image and the parameters, and it needs to be calculated for each image. Loopy belief propagation, Markov chain Monte Carlo (MCMC) and Contrastive divergence are some methods used for approximate training [37, 38]. Being approximations, these methods may not estimate the parameters well [31]. Furthermore, for methods like belief propagation, the time complexity is exponential in the size of the largest clique. And in our case, since the CRF model involves higher order cliques, the largest clique might have a size of 80, hence loopy belief propagation cannot be employed.

An alternative, practical approach is piecewise training [31, 33, 39]. In piecewise training each piece of the CRF model is learned independently. In our case, parameters for association potential and interaction potential are learned independently. The training methodology is similar to that of [33]. Our training data set consists of eight manually labeled images. We first estimate the association potential parameters α_0 and α_1 using four images from the training set. We use the R project for statistical computing [40] to estimate the parameters. The `glm` function in the stats core package is used for this purpose. This function uses the iteratively reweighted least squares (IWLS) method to obtain the fit. Once the parameters for association potential are estimated, we estimate the parameter for the interaction potential by keeping the association potential parameters constant. The second half of the training dataset (four images) which was not used to train the association potential is used for this purpose. The interaction potential parameter, β , which minimized the pixel wise classification error (i.e. the total number of misclassified sky pixels and cloud pixels) was picked.

Table 4.2: Accuracy, Precision and Recall Values with 99.9% Confidence Interval for Adaptive Thresholding Schemes (ATS) and CRF Based Method, Evaluated on Image Set C

	ATS (NSV ratio)	ATS (NBR ratio)	CRF
Accuracy	0.7979 \pm 0.0963	0.7173 \pm 0.1654	0.9346 \pm 0.0269
Precision	0.7782 \pm 0.1552	0.7324 \pm 0.1954	0.9561 \pm 0.0560
Recall	0.9047 \pm 0.0838	0.9022 \pm 0.1196	0.9022 \pm 0.0471

Table 4.3: Accuracy, Precision and Recall Values with 98% Confidence Interval for Fixed Thresholding Scheme and CRF Based Method, Evaluated on Image Set D

	Fixed Thresholding	CRF
Accuracy	0.8892 \pm 0.0359	0.9436 \pm 0.0181
Precision	0.9420 \pm 0.0378	0.9597 \pm 0.0341
Recall	0.8236 \pm 0.0540	0.9095 \pm 0.0309

Once we have estimated all the parameters, we need an inference algorithm which would find the most probable labeling configuration given an image (data) and the parameters. I.e., we want to find \mathbf{y}^* such that

$$\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}; \theta)$$

Due to the intractability of exact inference, we use the local search algorithm, iterative conditional modes (ICM), proposed by Besag [41]. ICM is an iterative procedure that maximizes the local conditional probability. For each site in an image, we find

$$y_i = \operatorname{argmax}_{y_i} P(y_i|y_j, \mathbf{x})$$

This procedure is iterated until convergence. ICM requires an initial labeling configuration to begin with. The output of the logistic regression classifier (the association potential ψ) is used as the initial guess.

During training and inference, there may be situations where a particular site has only one class of sites around it, such that all its neighbors are cloud or all are sky. In such scenarios we use the average sky and cloud NSV values of the entire image instead of the corresponding values of the neighbors. In order to calculate the average sky and cloud NSV values of the entire image, the labeling configuration from the previous ICM iteration is used.

4.3.2 Results

In order to evaluate the performance of the proposed method, two separate test datasets were used. One of these datasets (Set C) contains 22 images and was used to compare the performance of the CRF-based method with the minimum-cross-entropy method proposed by Li et al. [23]. Li’s method uses a hybrid scheme combining fixed thresholding and adaptive thresholding. The decision to use an adaptive threshold or fixed threshold is based on the standard deviation of the NBR ratio. But we observed that this decision rule does not always work well because some clear sky and completely cloudy images have a standard deviation greater than that of some images containing both sky and cloud. Hence, for a fair comparison we have only included images containing both sky and cloud in the first dataset. The second dataset (Set D) contains 26 images in total and was used to compare the performance of the proposed CRF method with that of fixed thresholding proposed by Long et al. [15]. This data set contains all the 22 images in Set C with an addition of two completely cloudy and two clear sky images.

Again, we use accuracy, precision and recall computed based on the confusion matrix as metrics for performance evaluation [26].

Table 4.2 shows the results for two adaptive thresholding schemes and the CRF

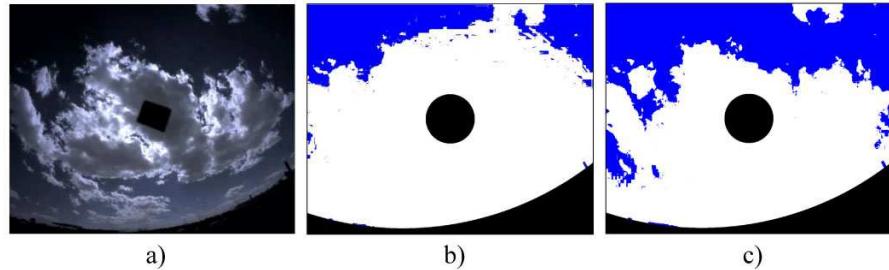


Figure 4.10: a) Original image and segmentation results using adaptive thresholding with b) NBR Ratio and c) NSV Ratio as features.

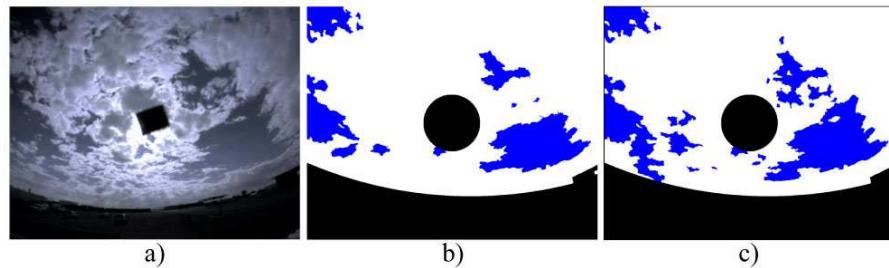


Figure 4.11: a) Original image and segmentation results using the CRF model b) with $\beta = 0$ and c) with $\beta = 0.95$ (the estimated value).

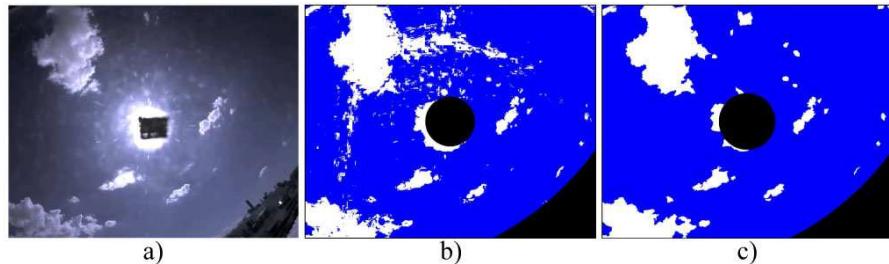


Figure 4.12: a) Original image, b) pixel-wise classification result of logistic regression and c) segmentation result using the CRF model with $\beta = 0$.

method using Set C. Both adaptive thresholding schemes are based on cross-entropy minimization, the difference being that one uses NSV as the feature whereas the other uses NBR. The accuracy of the CRF-based method is much higher in comparison with the other two methods, and the accuracy values do not overlap even at 99.9% confidence. The NSV thresholding performed better than NBR threshold-

ing. The reason for the poor performance of adaptive thresholding is that the best fit does not always correspond to the best segmentation/classification , especially when there is high intra-class variance. For instance, as shown in Fig. 4.10, the sky region at the top portion of the image is dark in comparison to the sky region at the bottom of the image. And when the adaptive thresholding algorithm tries to fit a binary surface that minimizes the cross-entropy, it ends up misclassifying the entire bottom portion of the image.

The results of comparing the performance of our method with the fixed thresholding scheme is provided in Table 4.3. Though the fixed thresholding scheme does better than adaptive thresholding, the CRF method provides considerable improvement in accuracy. The accuracy values do not overlap at 98% confidence level, so we can be 98% confident that the CRF method is better. Here, taking the spatial context into consideration leads to better accuracy. Fig. 4.11 shows that the inclusion of interaction potential helps in correcting some mistakes made in segmentation using the association potential alone. Looking at the local context is really helpful in regions where the histograms of sky and cloud regions overlap in the feature space.

Forming regions using mean shift segmentation and then performing region-based classification helps in dealing with noise. Fig. 4.12 shows that using logistic regression (association potential) to classify the image on a pixel-by-pixel basis yields noisier results than classifying regions formed by mean shift segmentation. Hence, defining the CRF model over regions not only speeds up computation but also helps in combating noise. However, in cases where a cloud has an unusual shape, segmentation may fail to find the proper regions. But for most cases, a region based approach works well. Fig. 4.13 shows the segmentation results for some images obtained using our CRF method

4.3.3 Conclusion

We have presented a CRF model for cloud detection on ground-based sky images which outperforms Li's adaptive thresholding algorithm and Long's fixed thresholding algorithm on our images. Though our model uses only two features (NSV ratio and NBR ratio) it is very flexible, and more features (if found useful) can be added very easily. For instance, adding more features into the association potential is trivial. Similarly, the interaction potential could be replaced by one or more features that may help take the local context into account in a better way. Since each imaging system is different, it is possible to find features different from what we have used which are better suited for different imaging setups. But our aim here is to demonstrate that the idea of combining a discriminative classifier with a higher-order clique potential in a CRF framework is a really powerful scheme for cloud detection.

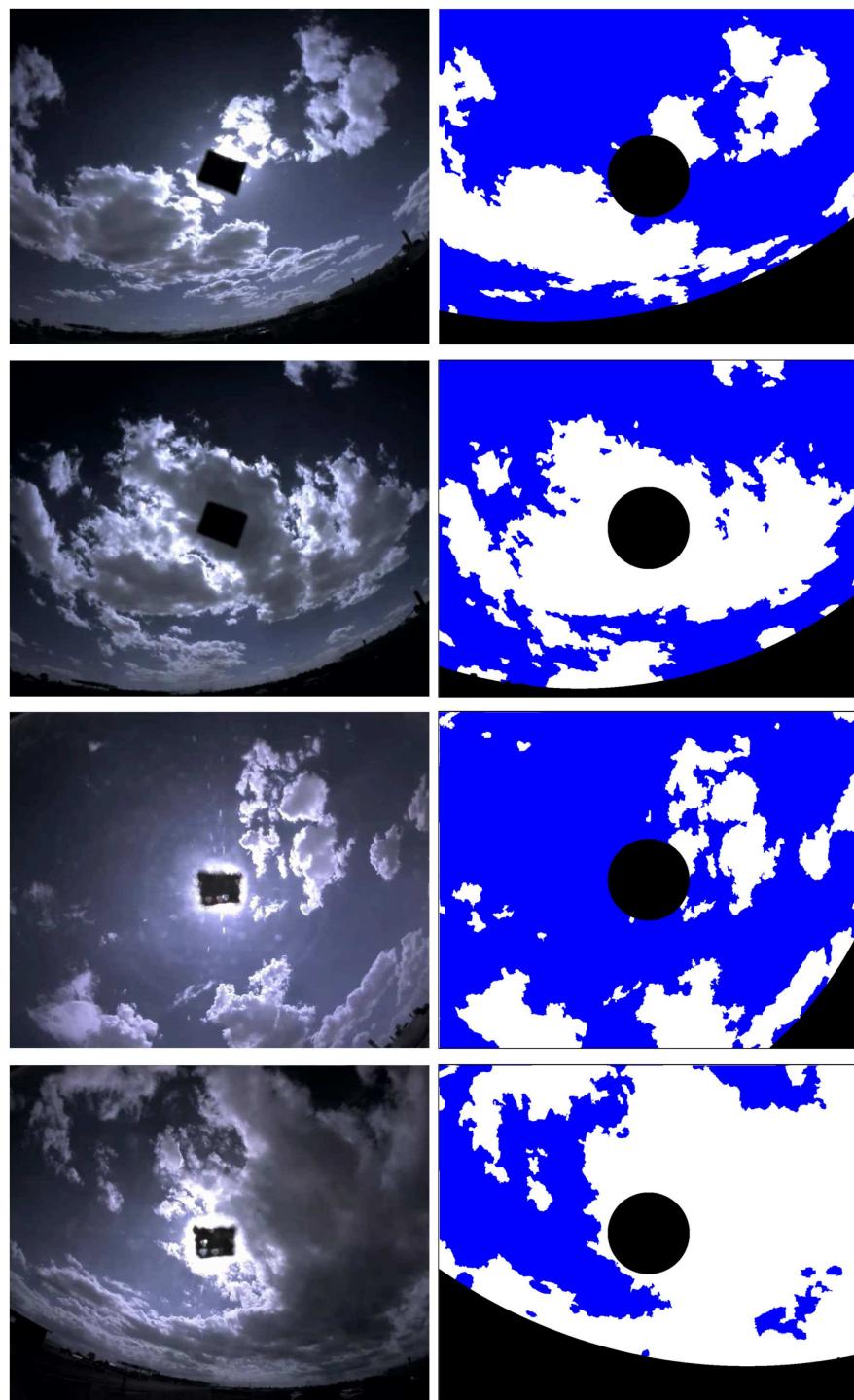


Figure 4.13: Segmentation results obtained using the CRF model for some images from the test database.

CHAPTER 5

MOTION ESTIMATION

5.1 Introduction

Block-matching motion estimation (BME) and optical flow are two common motion estimation algorithms. BME is widely used in motion-compensated video coding. BME algorithms exploit the fact that adjacent frames in video often have redundant information due to the frame rate exceeding the amount of motion. Optical flow, on the other hand, is a method to estimate local motion based on local derivatives. To determine the optical flow, the motion constraint equation is first derived assuming that the motion from frame to frame is small. Then to solve the aperture problem (one constraint and two unknowns), either Lucas-Kanade method (which assumes that the flow is constant in a local neighborhood) or the Horn-Schunk method (which assumes a smooth flow field globally) is employed.

5.2 Choosing a motion estimation algorithm

In order to find a motion estimation method suitable for sky images, three BME algorithms and an optical flow implementation were evaluated. The three BME algorithms evaluated were adaptive road pattern search (ARPS) [42], diamond search (DS) [43] and hexagon-based search (HS) [44]. All these algorithms come under the category of fast BME algorithms as they avoid an exhaustive search and thus save computation time. These algorithms are the standard BME algorithms proven to work well on a variety of images. Execution speed and ease of implementation was also considered while selecting them. The basic idea behind all these algorithms is to partition each video frame into non-overlapping blocks and then compare these blocks with a previous (reference) frame to determine a

motion vector for each block. A coarse search is first done to identify the most probable neighborhood for finding a match, and then a fine search is performed by zooming into the neighborhood found. To measure the dissimilarity of blocks, HS uses mean of absolute difference (MAD) whereas DS and ARPS employ sum of absolute difference (SAD) between blocks. The main difference between HS and DS is in the pattern used for searching. Diamond search uses a diamond pattern around the center of the block, whereas HS uses a hexagon-based pattern. The ARPS, on the other hand, not only uses a different pattern (a rood pattern) but also uses an adaptive size for the pattern by looking at the previously estimated neighborhood motion vectors.

To evaluate the performance of these algorithms, we consider various effective frame rates (i.e., various temporal separations between the current frame and the reference frame) for computation of the motion vectors. In addition, for the BME techniques, we consider various block sizes for partitioning the image. (The optical flow techniques compute the motion for all the pixels rather than blocks.)

Peak signal-to-noise ratio (PSNR) is the most commonly employed metric to evaluate the performance of BME algorithms. To compute the PSNR, the motion-compensated image has to be created from the reference image. I.e., each block in the current frame is replaced with the best matching block from the reference frame to construct the compensated image. Next the mean squared error (MSE) is computed:

$$\text{MSE} = \frac{1}{NM} \sum_{i=1}^M \sum_{j=1}^N (I_{\text{comp}}(i, j) - I_{\text{cur}}(i, j))^2$$

where M and N are the dimensions of the image respectively, I_{comp} represents the compensated image, and I_{cur} represents the current image. The PSNR can then be calculated as

$$\text{PSNR} = 10 \log_{10} \left(\frac{M_C^2}{\text{MSE}} \right) \text{ dB}$$

where M_C is the maximum intensity (gray scale value) in the image, i.e. 255.

The PSNR values were calculated for each algorithm using some randomly

selected sky images. Frame rates of 1, 5, 10 and 15 and block sizes of 25, 50, 75 and 100 were used. Fig 5.1 shows the evaluation results. The optical flow method was the least impressive and the most time consuming. The longer computation time can be understood because the motion vectors have to be evaluated for all the pixels. The results of the optical flow method do not depend on the block size but are repeated in Fig 5.1 for the sake of completeness. The PSNR for optical flow corresponding to block size of 75 appears different from the results for other block sizes because the image was re-sized to fit an integral number of blocks and so for block size 75 the image size was different. Though the other three methods seem to fare well in comparison to optical flow, the HS algorithm turns out to be the best among the lot. Except for the combination of block size 25 and frame rate 15, the PSNR of the HS algorithm was greater than or equal to the PSNR of other methods. The PSNR decreases with increasing frame rate, which is understandable because clouds are not rigid bodies and change shape with time. Therefore, a higher frame rate should lead to a lower PSNR. For the same reason, Fig 5.1 does not tell us much about optimal frame rates or block sizes. In order to find good values for these parameters, a task-based evaluation was performed. The final aim of motion estimation is to be able to estimate the future position of the clouds from the current velocities. Hence, advected images were created using the velocities estimated by the BME algorithms (optical flow was not included in this evaluation as it did not perform well in the PSNR analysis). In the advection step, the clouds are advected to their future positions (this is explained in detail in the next chapter). The basic idea is to create a future image of the clouds based on the velocity estimates and then compare that image to the actual image at that future time. In the experiment automatically segmented clouds were advected to their positions five minutes into the future and then the accuracy was calculated by comparing the advected segmented image to the actual future image (manually segmented). Again, the accuracy was calculated based on the confusion matrix as before (see section 4.2.3). Fig 5.2 shows the bar plot of the accuracy. The HS algorithm yet again fares better than others and is consistent.

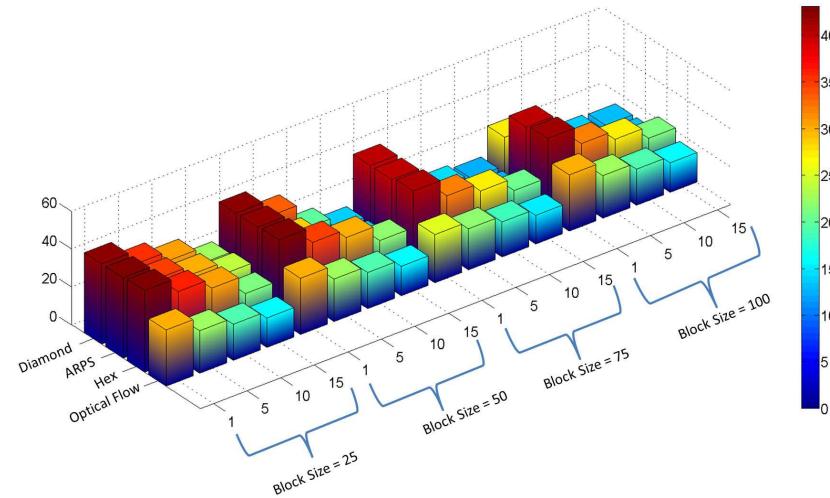


Figure 5.1: Bar plot of PSNR values (in dB) for the various algorithms. The color indicates the height of the bar, and the numbers along the horizontal axis (1, 5, 10 and 15) indicates frame rates (in seconds). The block size indicates the side length of the square blocks (in pixels) into which the frame is divided.

Another interesting observation is that the frame rate does not seem to matter much in terms of accuracy. The block sizes 50 and 75 seem to do consistently well. One thing to keep in mind here is that the clouds could change shape significantly during a time period of five minutes, and the advection step does not take that into consideration. This may raise questions on the usefulness of advection accuracy as an evaluation metric. But since the PSNR for lower frame rates would be higher as mentioned earlier, task-based evaluation of the motion vector estimates was performed to aid the process of choosing an optimal frame rate for sky images. In the advection scheme explained later, multiple frame rates ranging from 4 to 12 are used and the block size is set to 50.

5.3 Median filtering and motion vector transformation

To deal with the noisy outliers in the motion vectors estimated by the HS algorithm, median filtering is employed. For simplicity and computational speed,

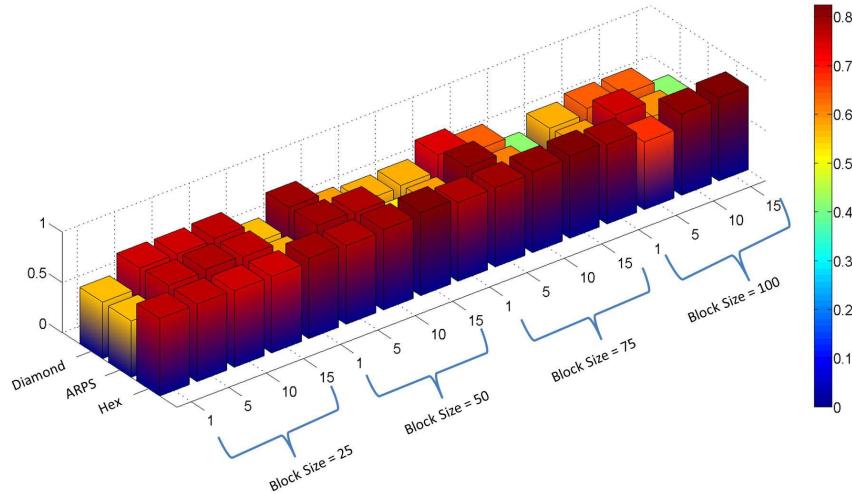


Figure 5.2: Bar plot of accuracy of advected images for the BME algorithms.

we applied median filtering to the x and y components of the velocity vector independently. The vector median filter is an alternative technique that could later be explored. Fig 5.3 shows the results of performing median filtering.

After median filtering the motion vectors have to be transformed from pixels/sec to meters/sec. Clouds far away on the horizon have a smaller angular velocity than clouds overhead as shown in Fig. 5.5. We must correct this in order to obtain accurate forecasts. In order to compensate for this, we convert pixel coordinates of clouds to 3D position vectors in the real-world reference frame (Fig. 5.4). To do this, we first define a camera reference frame (right handed) such that V_{CX} and V_{CY} correspond to right and up in the image respectively and V_{CZ} points out of the image plane. For each pixel we then determine a vector in the camera reference frame of unit length that corresponds to the direction from the camera to the object being imaged at that pixel. For example, a cloud at coordinates (x,y) in an image would correspond to a vector $\vec{V}_{\text{Cloud}}^C = (x, y, f)/|x, y, f|$, where f is a constant determined by the imaging system (roughly equal to focal length/pixel size). Note that the superscript indicates the reference frame, C for the camera reference frame, R for the real-world reference frame.

Next, we determine a representation of \vec{V}_{Cloud}^C in the real-world reference frame (\vec{V}_{Cloud}^R). Specifically, we must find a matrix \mathbf{T} such that in the real-world frame

$$\mathbf{T} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \vec{V}_{\text{CX}}^R$$

$$\mathbf{T} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \vec{V}_{\text{CY}}^R$$

$$\mathbf{T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \vec{V}_{\text{CZ}}^R$$

where \vec{V}_{CX}^R , \vec{V}_{CY}^R , \vec{V}_{CZ}^R are the coordinate vectors of the camera reference frame (V_{CX} , V_{CY} and V_{CZ}) represented in real-world coordinates. These three equations can be written as

$$\mathbf{T} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} V_{\text{CX}1}^R & V_{\text{CY}1}^R & V_{\text{CZ}1}^R \\ V_{\text{CX}2}^R & V_{\text{CY}2}^R & V_{\text{CZ}2}^R \\ V_{\text{CX}3}^R & V_{\text{CY}3}^R & V_{\text{CZ}3}^R \end{bmatrix}$$

What remains is to determine \vec{V}_{CX}^R , \vec{V}_{CY}^R and \vec{V}_{CZ}^R which are the representations of V_{CX} , V_{CY} , and V_{CZ} in real-world coordinates.

In the real-world reference frame, the positive X direction (V_{RX}) points towards west, the positive Y direction (V_{RY}) points towards south and the positive Z direction (V_{RZ}) points vertically up with respect to the ground (Fig. 5.4). \vec{V}_{CZ}^R is a known vector if we know the direction of the sun at any given instant of time. We use a solar position algorithm [18] to determine the zenith and azimuth angle of the sun. Once we know the zenith and azimuth angles, \vec{V}_{CZ}^R can be computed as

$$\vec{V}_{\text{CZ}}^R = [-\sin \phi \sin \theta, -\sin \theta \cos \phi, \cos \theta]$$

where θ = zenith angle (angle from the vertical) in radians and ϕ = azimuth angle (eastward from the north) in radians (see Fig. 5.4). The camera is mounted on the tracker such that \vec{V}_{CX}^R is perpendicular to both \vec{V}_{CZ}^R and \vec{V}_{CN}^R ; therefore, \vec{V}_{CX}^R is given by

$$\vec{V}_{CX}^R = \vec{V}_{CN}^R \times \vec{V}_{CZ}^R / |\vec{V}_{CN}^R \times \vec{V}_{CZ}^R|$$

where \times indicates cross product. \vec{V}_{CY}^R is perpendicular to both \vec{V}_{CX}^R and \vec{V}_{CZ}^R and is therefore given by

$$\vec{V}_{CY}^R = \vec{V}_{CZ}^R \times \vec{V}_{CX}^R$$

The cloud vector in the real-world reference frame is now given by

$$\vec{V}_{Cloud}^R = \mathbf{T} \vec{V}_{Cloud}^C$$

The transformation matrix \mathbf{T} is a unitary matrix corresponding to a rotation. This means \vec{V}_{Cloud}^R still has unit length. We convert this unit length vector to a cloud position vector by multiplying by a constant such that the height of the cloud \vec{V}_{Cloud}^R is equal to 3000 meters, which we assume to be the typical cloud height. In this way, we can now estimate actual cloud velocity and forecast cloud events. I.e., the motion vector in pixels/sec would tell us the future position of a block (we are computing motion block by block as explained before) in the image. We can use the above transformation to convert the current and future positions of the block in the camera reference frame into their real-world coordinates, which would give us an estimate of the velocity of the cloud blocks in meters/sec. We pick a typical cloud height to get motion vectors with physically intuitive values. The result of motion vector transformation is shown in Fig. 5.5.

5.4 Removing the horizon

As mentioned in chapter 3, the horizon needs to be removed from the images and the motion vector transformation equations can be used for this purpose. Removing the horizon is important because, if we know where the horizon is, we

can avoid doing any processing steps in those regions. It is also helpful in the advection step when we do back-tracking. The idea here, is to estimate the 3D real-world coordinates corresponding to the image pixels and then find the pixel locations where the z-coordinate (V_{RZ} in Fig. 5.4), or height above ground level, becomes negative.. The horizon corresponds to the points where the z-coordinate changes sign. We also compensate for the fisheye lens effect in this process. This is important as the horizon is not a straight line but a curve. To compensate for the fisheye lens effect, stereographic projection equation is used.

$$r = 2f \tan(\theta/2)$$

where r is the distance of the pixel from the center of the image in camera reference frame, f is the focal length and θ is the angle in radians between the actual point being imaged and the optical axis of the lens. If (x_p, y_p) are the coordinates of a pixel in the camera reference frame (origin is the center of the image), we can compute $r = \sqrt{x_p^2 + y_p^2}$. Now θ can be computed using the projection equation defined above. The actual distance value of the point r_{new} is given by $r_{new} = f \tan \theta$. We can now use the initial angular position of the pixel (i.e., $\arctan(y_p/x_p)$) and the new distance value to compute the compensated pixel coordinates.

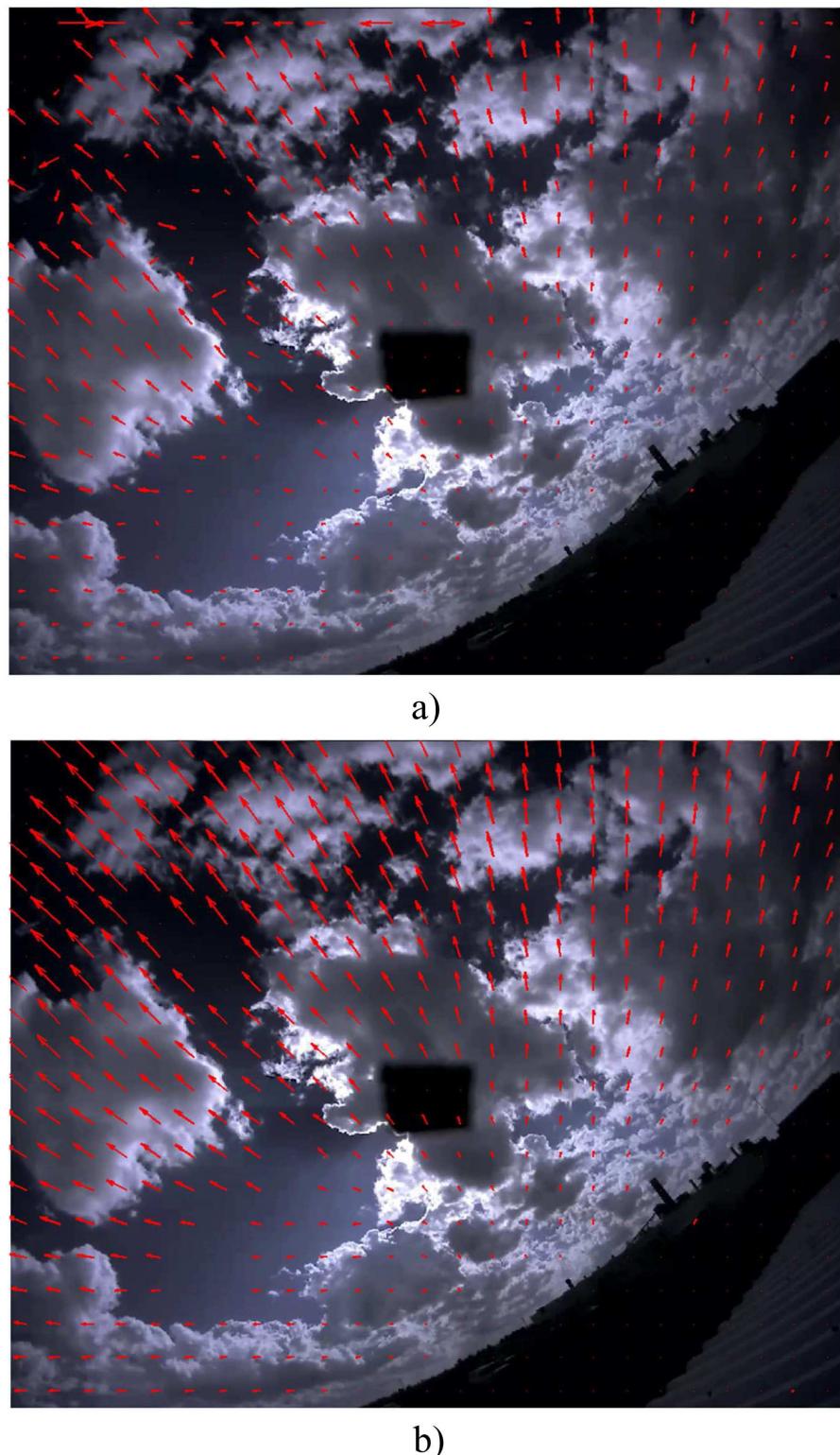


Figure 5.3: a) Motion vectors before median filtering and b) after median filtering. The lengths of the displayed motion vectors are normalized separately for the two images.

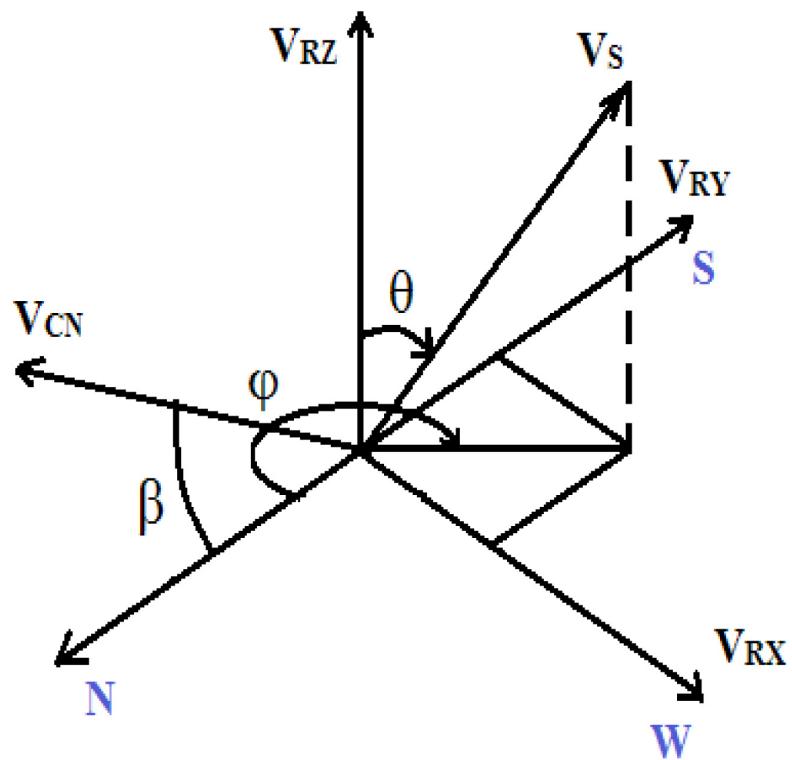
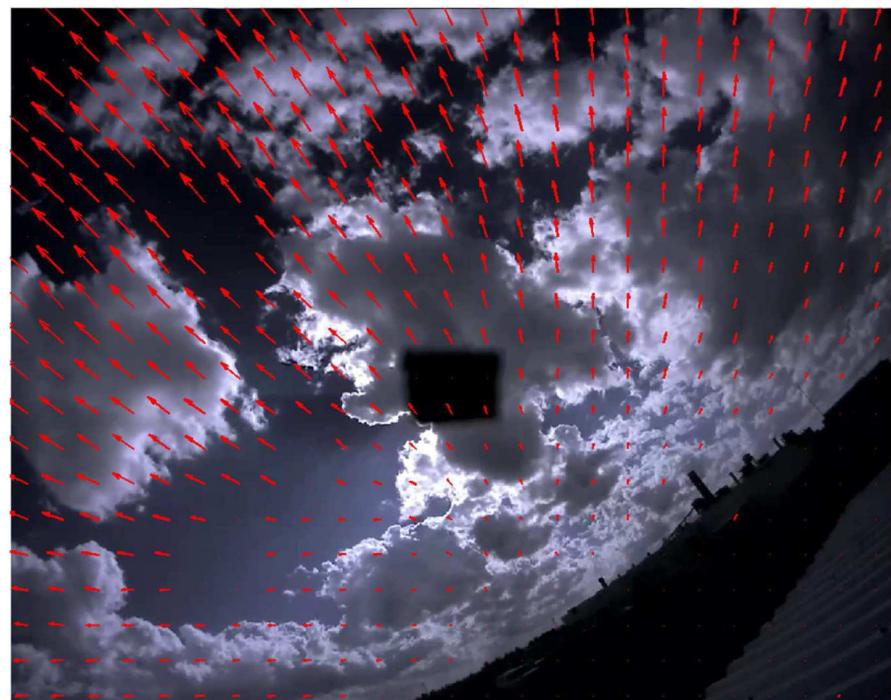
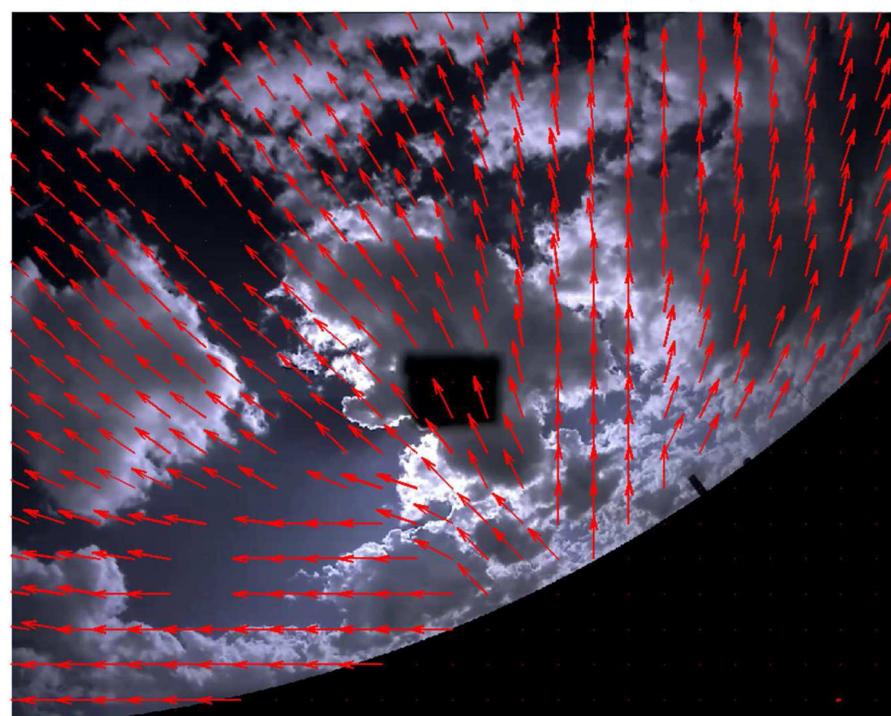


Figure 5.4: Real-world reference frame (V_{RX} , V_{RY} , V_{RZ}). V_{CN} represents the Earth's axis of rotation, pointing towards the celestial north. V_S points towards the sun; θ = zenith angle (angle from the vertical) in radians and ϕ = azimuth angle (eastward from the north) in radians.



a)



b)

Figure 5.5: a) Motion vectors before and b) after motion vector transformation.

CHAPTER 6

ADVECTION AND POWER PREDICTION

6.1 Advection

The process of finding the future position of the clouds, once the clouds have been detected and their motion estimated, is commonly referred to as advection. The simplest way to perform this task is to use the velocity of the block estimated in pixels/sec by the block-matching motion estimation algorithm, to calculate the future position of the blocks. But this approach is too simplistic. Since we know that the clouds move as cohesive unit, this type of advection is not logical. In order to address these issues we perform advection in the real-world reference frame (Fig. 5.4). The advantage of doing so is twofold. One is that when we transform from camera reference frame to real-world reference frame, we convert the motion vectors from pixels/sec to meters/sec, thus compensating for the difference in angular velocity between clouds overhead and clouds far away. The second advantage is that it would be more reasonable to assume that all clouds are moving with a single velocity in real world than in an image. Though this assumption might seem too restrictive, it has been used by previous researchers and was reported to be quite effective [6, 12].

Advection in real-world reference frame can be achieved in multiple ways. The most straight forward way of accomplishing advection is to do the following. Convert the current position of the blocks (the center of the block represents the position of the block) into its real-world coordinates using the transformation described in the previous chapter (motion vector transformation). Next, use the motion vectors in meters/sec and the forecast horizon (the time duration into the future for which we are making predictions) to compute the new positions (coordinates) of the block in real-world reference frame. Convert the real-world coordinates back

into camera coordinates using the transformation equations described earlier. In this manner, we would be able to construct the future frame. One problem with this scheme is that it would leave gaps in the constructed image which needs to be filled. As shown in Fig. 6.1, we can use a simple nearest neighbor interpolation to fill in the gaps at the center. There is no need to interpolate the entire image as we care only about what happens at the center, around the sun. Since the clouds are moving towards the top-right corner in the example in Fig. 6.1 and because we do not yet know what follows the clouds at the extreme left, the regions near the left edge of the frame appear empty. Though this scheme seems to be effective in creating future frames, depending on the perspective of the camera (or in other words the time of day) and depending on the motion of the clouds, the blocks might end up at positions that leave huge gaps in between them. This problem is illustrated in Fig. 6.2. In such a scenario, performing a nearest neighbor interpolation does not make sense. A smarter interpolation scheme has to be devised to overcome this problem. Alternatively, we can do the reverse of the previous procedure to construct the future frame. I.e., for each pixel in the future frame, we can trace back the pixel in the current frame which is likely to end up in the position of the future frame pixel based on the velocity vectors. This way, we can avoid gaps in the constructed frame. While doing so, care has to be taken to consider the fact that the camera will have a different perspective five or ten minutes into future. Hence, two different transformation matrices need to be employed while converting back and forth between real-world and camera reference frames. A problem might occur in this scheme of advection when the pixel in the current frame to which a pixel in the future frame gets mapped is in the center of the image. Since a circular portion of radius 100 pixels is removed from the center of the image in the preprocessing step, we do not have information as to whether the pixels in that region are cloud or sky. To overcome this issue, we first interpolate (nearest neighbor) the cloud map of the current frame to fill in the gap at the center. Fig. 6.2 d) shows the results for this scheme. It must be kept in mind that clouds, being non-rigid bodies, change shape during the 5 minute duration. Hence, predicting the exact

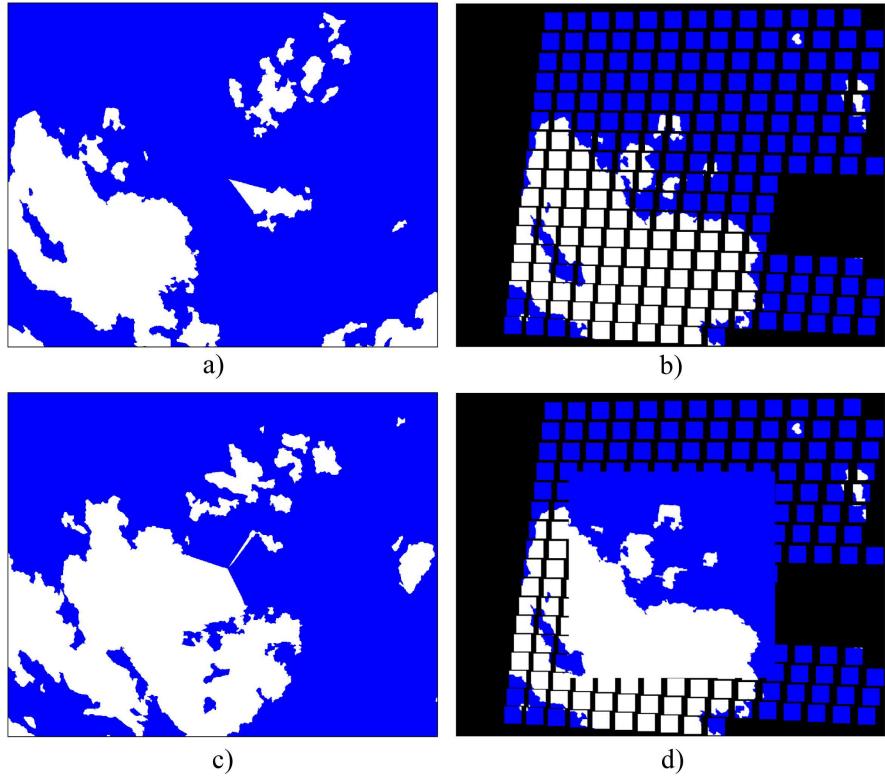


Figure 6.1: a) Original cloud map (segmented image), b) image obtained after 5 minute advection using the first scheme (forward), c) actual cloud map after 5 minutes and d) advected cloud map after nearest neighbor interpolation in the center of the image.

cloud map after five minutes would require more sophisticated approaches which take the actual physics behind the process into consideration. It is also possible for clouds to appear and disappear within the forecast horizon and this often leads to errors in the forecasts. Another issue that needs to be dealt with in both advection schemes, is choosing a representative motion vector from the available list of motion vectors. Since we employ block-matching motion estimation, each block would give us a motion vector. We perform a connected component analysis on the cloud map, to figure out the various connected cloud entities in the image. Here, we have a choice of choosing one motion vector per cloud or choosing a single motion vector for all the clouds. We choose the geometric median of the motion vectors of blocks

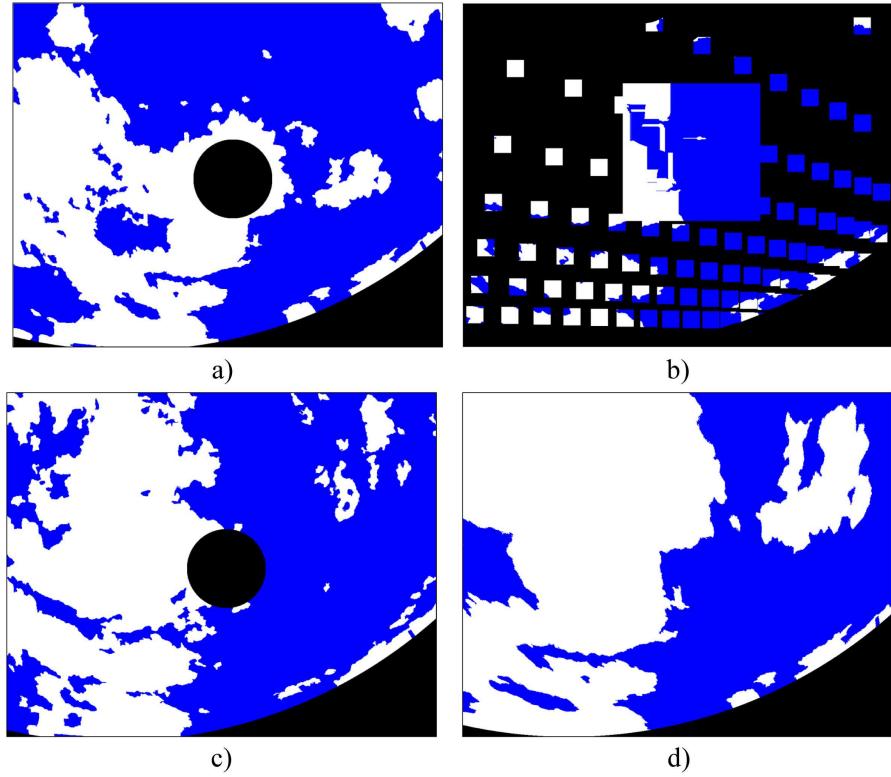


Figure 6.2: a) Original cloud map (segmented image), b) image obtained after 5 minute advection and interpolation using the first scheme (forward), c) actual cloud map after 5 minutes and d) advected cloud map using the second scheme (backward).

belonging to that cloud to represent individual cloud velocities and the geometric median of the motion vectors of all cloud blocks to assign a single motion vector to all the clouds. The motion vectors estimated would have a few outliers due to noise and also due to the fact that a change in shape of the cloud would be perceived as motion. But it is reasonable to think that a majority of the motion vectors would represent the actual motion of the cloud. Hence choosing the median motion vector to represent cloud motion would be appropriate. After initial analysis, it was decided to choose a single motion vector for all clouds as this turned out to be a more reliable option. The details regarding this analysis are provided in the next section.

6.2 Power forecasting

Once the cloud advection step is completed, the next step is to make power forecasts. At TEP solar test yard, where the camera is located, power measurements from various solar panels are recorded with 1 second precision. Therefore, the moment we observe a cloud passing beneath the sun in the video recorded by the camera, the power values recorded at TEP will also show a drop. I.e. the power values are in synchronization with what we see in the camera. Hence, these power measurements can be employed to measure the accuracy of the power forecasts. In order to make forecasts, a 50 square pixel region at the center of the image is considered. This square represents the position of the sun. The system currently produces binary forecasts only. To predict the power at a future point in time the cloud map advected to that point in time is considered. If more than half (50%) of the central square region is covered with clouds in the advected cloud map a power level of 20% of ideal day value at that point in time is predicted. otherwise, a power value equal to the ideal day power value is predicted. The ideal day power value is the power value recorded on a clear sky day closest to the day for which forecasts are made. One prediction is made every 15 seconds. While making predictions, 3 sets of motion vectors are considered. The first set of motion vectors is obtained by using the current frame (for which prediction has to be made) and a frame recorded four seconds before the current frame as inputs to the block-matching motion estimation algorithm. For the second and third sets of motion vectors, frame rates of 6 and 8 are used. The purpose of doing this is to make the system robust to noise in the estimated motion vectors. Hence every 15 seconds, when a prediction has to be made, three different advected cloud maps are created based on the three sets of motion vectors. Each cloud map would give its decision and in the case of a conflict, the majority decision is considered. Fig. 6.3 and Fig. 6.4 shows the one-minute average, five minutes ahead power forecasts for four days. Fig. 6.5 shows ten and fifteen minutes ahead, one-minute average power forecasts for December 31, 2012. It must be noted that the forecast is not run for

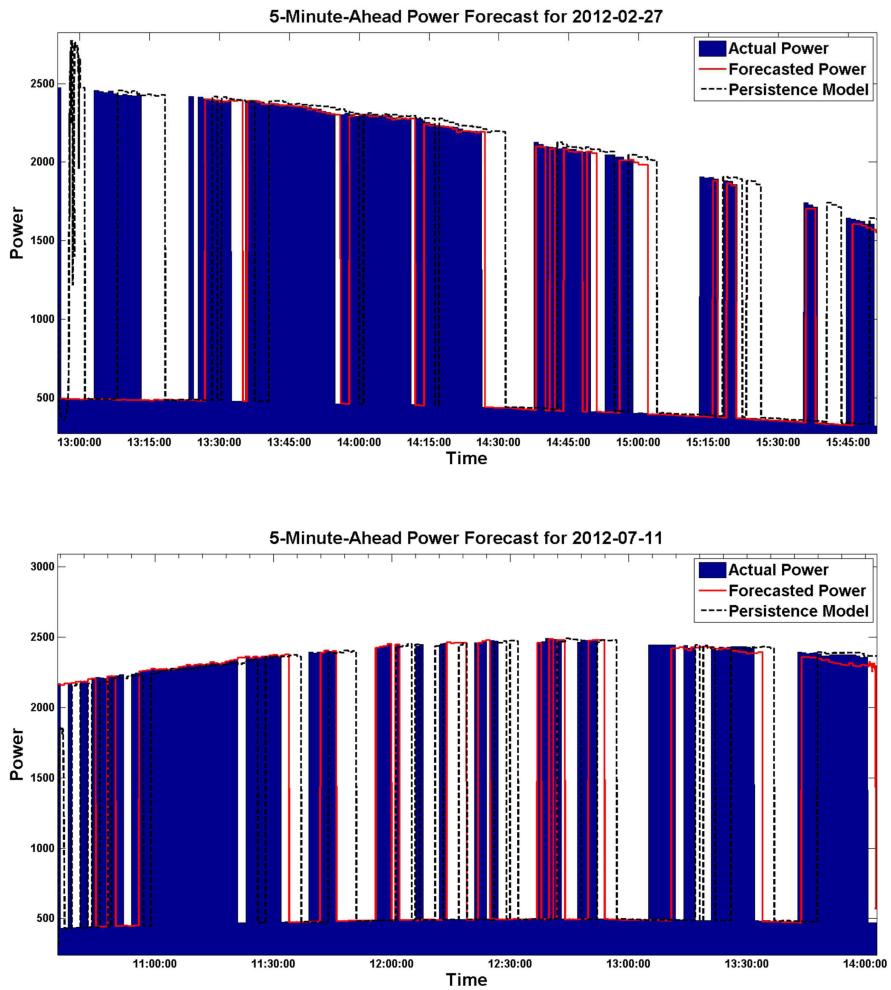


Figure 6.3: Five-minute-ahead one-min average power prediction for February 27 and July 11, 2012.

the entire day but for an average three hours per day, during times when there was some cloud activity. As mentioned above, since one prediction is made every 15 seconds, there would be four predictions per minute. The mean value of these predictions are computed, and if that mean is above 75% of the ideal day value, then the forecast for that minute is set to the ideal day value for that time. On the other hand, if the mean value turns out to be smaller, then the forecast for that minute is set to 20% of the ideal day value (which corresponds to an 80% drop). In a desert region like Tucson, the diffuse component of the solar radiation usually varies from 15% to 30% [45]. Therefore, when the sun is blocked by the clouds, the power from the solar panel does not fall to zero. The dotted line in the plot is the persistence model. The persistence model is the simplest forecasting model which assumes that whatever condition is prevailing now will be the condition at the time for which the forecast is made, hence the name persistence. I.e., if the forecast horizon is say five minutes, and if the sky is clear now, the persistence model would predict that it is going to be clear five minutes from now. Though it would seem to be a model easy to beat, it is very accurate for short-term forecasts. The reason is that the climate varies slowly and so there is a high probability that the condition now would remain the same five minutes from now. Hence, the persistence model is employed widely in the literature to compare the accuracy of the forecasts [6, 12]. Therefore, we compare the accuracy of our system to the persistence model. Table 6.1 reports the comparison results. Except for the 15-minute-ahead forecast on July 11, 2012, our forecasts have lower root-mean-squared error (RMSE) than the persistence model. Though better than persistence model, the system do make some mistakes. One of the reasons for wrong forecasts is the dynamic behavior of the clouds. As mentioned before, clouds are not rigid bodies and can drastically change shape within minutes. Clouds often disappear just before hitting the sun, and at times they appear from nowhere. This becomes a bigger issue for longer forecast horizons. The system as of now is not equipped with capabilities to deal with such things and hence will invariably make some mistakes. Another issue that leads to inaccurate forecasts is the rule used to decide whether it is cloudy or

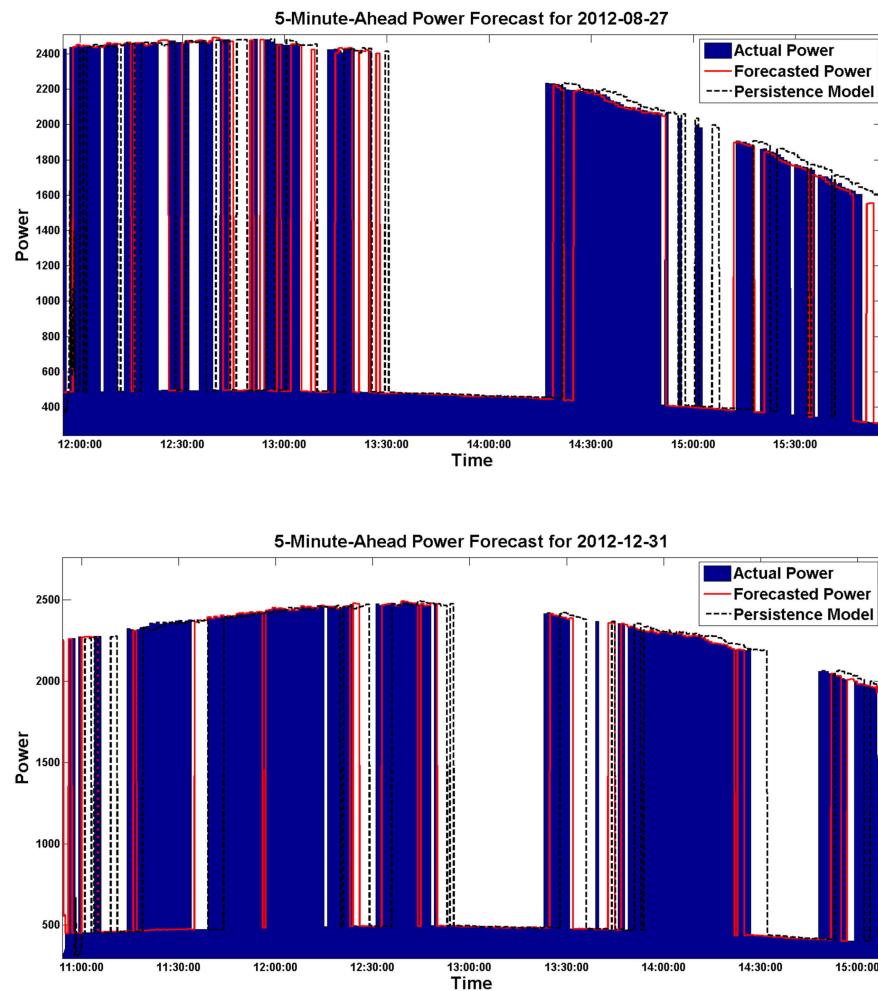


Figure 6.4: Five-minute-ahead one-min average power prediction for August 27 and December 31, 2012.

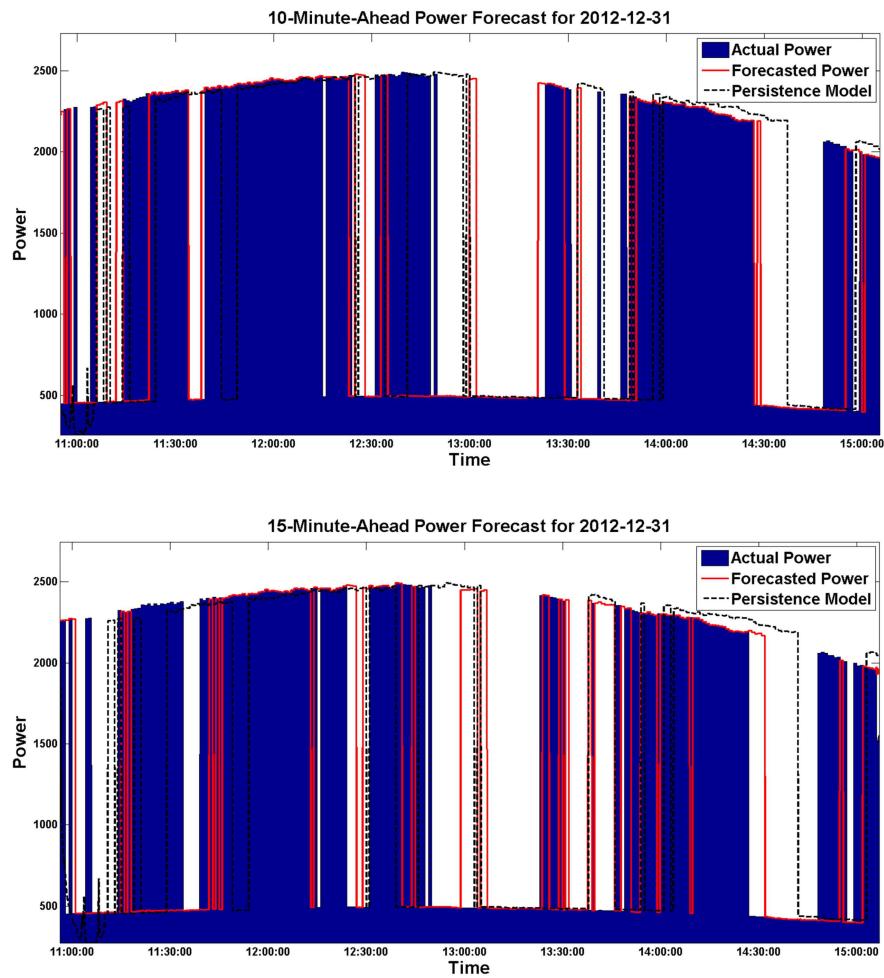


Figure 6.5: Ten and fifteen-minutes-ahead one-min average power prediction for December 31, 2012.

Table 6.1: Root-Mean-Squared Error (Watts) for the Forecast Made on Four Days

	Forecast Horizon	RMSE (Our Forecast)	RMSE (Persistence)
February 27, 2012	5	860.57	1085.77
	10	1051.17	1227.25
	15	1067.31	1185.48
July 11, 2012	5	966.76	1340.87
	10	1158.68	1306.20
	15	1292.30	1225.03
August 27, 2012	5	903.31	1011.91
	10	942.96	1147.20
	15	1037.13	1136.35
December 31, 2012	5	890.31	997.62
	10	935.35	1151.07
	15	1096.57	1263.67



Figure 6.6: Though the central region around the sun is totally covered with clouds, sun shines forth through them.

clear. The rule as stated before is to look whether the 50 pixel square region at the center is at least 50% cloudy or not. Many a times clouds could be seen filling regions in and around this box, but without obstructing the sunlight (see Fig. 6.6). This is the reason why the algorithm wrongly predicted low power values around between 13:00 Hrs and 13:15 Hrs on February 27, 2012 (See Fig. 6.3). This again was the reason for predicting the fall in power values around 14:45 Hrs on the same day when there was no actual drop in values around that time. Another area of concern is the process of choosing a single motion vector. The chosen motion vector serves as an approximation to the real cloud motion and might therefore cause inaccuracies in predicting rise or fall of power. As mentioned earlier, we decided to choose a single motion vector for all clouds because it outperformed persistence model on all the four days. While the multiple motion vector option could not beat the persistence model on one of the days. At the same time the multiple motion

vector option generated lower RMSE values than the single motion vector option for one of the days. Therefore, the multiple motion vector approach has potential to improve forecasts. However, extensive testing could not be performed due to lack of time. This could be taken up as a future work. Sometimes, the tracker fails to keep the sun perfectly centered in the image. Such tracker errors will lead to wrong forecasts. Since the motion vectors are transformed from pixels/sec to meters/sec, even a small offset in the position of the sun from the center of the image can lead to large errors in the estimated motion vectors. Therefore, it is important to keep the tracker error free.

Run time is another issue with the system. It takes approximately one minute to make one prediction on an Intel Core i5 machine with 8 GB RAM. The segmentation (cloud detection) step takes close to 45 seconds to complete, the motion estimation step takes approximately 5 seconds (for three sets of motion vectors), the advection and interpolation steps take close to 13 seconds and another 2 seconds for other miscellaneous steps. Real-time forecasting is not possible with this implementation in MATLAB. However, the system can be used for real-time forecasting if one prediction is made every minute, rather than every fifteen seconds. Implementing the system in C or C++ can significantly improve the performance.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this Thesis we presented a complete framework capable of producing intra-hour solar power forecasts. In chapter 2 we described the imaging set up pointing out the fact that we use an imaging set up very different from other such systems. Next, we described the image preprocessing steps needed to normalize the image to compensate for the intensity distortion. In the fourth chapter the very important step of cloud detection was discussed. We introduced a new feature for cloud detection, especially suited for adaptive thresholding schemes, which was found to be useful for a broad range of images captured using different imaging systems. We also introduced a novel conditional random field (CRF) based cloud detection algorithm. The main idea here was to present a powerful framework to take spatial context into account. The features we used may not be the optimal features for other imaging systems, but the framework is very generic and replacing the features we employed with some other feature is trivial. Chapter 5 dealt with the problem of finding motion vectors. In this chapter, the analysis results of comparing different motion estimation algorithms were presented. The process of choosing an algorithm and the necessary parameters that suited sky images were presented. The motion vector transformation equations used to convert the cloud velocities from pixels/sec to meters/sec were described in this chapter. In chapter 6 the advection scheme and power forecasting method was presented. The utility of back tracing advection scheme was demonstrated. Finally, the power forecasting scheme was described. The superiority of the forecasts produced by our method in comparison to the persistence model was established. Though better than persistence model, the system does make mistakes while forecasting. The reasons for making such mistakes

were pointed out. The problem of performance was also discussed. In short, this thesis aimed at setting up a framework for intra-hour forecasting, making contributions to sky image analysis in the process, much of which was accomplished.

7.2 Future work

Improving the computational speed of the system, so as to enable live forecasting, should be a top priority in future work. Most of the previous work in automated cloud tracking has involved methods like k-means clustering to estimate a single motion vector for all the clouds. The use of multiple motion vectors has shown promise and therefore its utility needs to be investigated. A closely related problem is decoupling the actual motion of clouds and the apparent motion due to their change of shape in the motion estimation step. The advantage of doing so is twofold. Firstly, it would lead to a more accurate estimate of cloud velocity which would directly translate to higher accuracy in forecasting. Secondly, the perceived motion due to change in shape could tell us a lot about the dynamic behavior of the clouds. I.e., it can tell us whether the cloud is growing or shrinking, or tell us the likelihood that this cloud will make it to the sun without disappearing, etc. Another improvement would be to perform temporal median filtering of the motion vectors in order to reduce error. Similarly, the historical information regarding cloud detection can also be utilized to improve the cloud detection step. During the preprocessing step, a circular region of radius 100 pixels is removed from the center. And later, during the advection step, this region at the center is filled by using a nearest neighbor interpolation scheme. Better interpolation schemes are needed to deal with the problem of missing information at the center around the sun. Cloud detection in the central region is difficult, but the clouds that end up in these regions would have been out of that region at some point in time. Hence, keeping track of clouds entering and leaving that region would help us avoid interpolation altogether. The relationship between the area covered by the clouds at the center of the image (around the sun) and drop in power needs

to be investigated, which would hopefully lead to a way to quantify the forecasted power more precisely, rather than having just a binary forecast.

CHAPTER 8

PUBLICATION LIST

Listed below are the publications prepared as part of this thesis.

1. V. T. Jayadevan, J. J. Rodriguez, V. P. A Lonij, and A. D. Cronin, “Forecasting solar power intermittency using ground-based cloud imaging”, In *Proc. 41st ASES Annual Conference*, vol. 3, 2012, pp. 21002106.
2. V. P. A. Lonij, V. T. Jayadevan, A. E. Brooks, J. J. Rodriguez, K. Koch, M. Leuthold, A. D. Cronin, “Forecasts of PV power output using power measurements of 80 residential PV installs”, In *38th IEEE Photovoltaic Specialists Conference (PVSC)*, 2012, pp. 3300-3305.
3. V. T. Jayadevan, J. J. Rodriguez, and A. D. Cronin, “A new contrast enhancing feature for cloud detection in ground based sky images”, To be submitted to the *Journal of Atmospheric and Oceanic Technology*.
4. V. T. Jayadevan, J. J. Rodriguez, and A. D. Cronin, “Context-aware cloud detection in sky images using conditional random fields”, To be submitted to the *IEEE Transactions on Geoscience and Remote Sensing*

REFERENCES

- [1] G. R. Timilsina, L. Kurdegashvili, and P. A. Narbel, “Solar energy: Markets, economics and policies,” *Renewable and Sustainable Energy Reviews*, vol. 16, no. 1, pp. 449 – 465, 2012.
- [2] R. H. Inman, H. T. Pedro, and C. F. Coimbra, “Solar forecasting methods for renewable energy integration,” *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535 – 576, 2013.
- [3] W. Pickard and D. Abbott, “Addressing the intermittency challenge: Massive energy storage in a sustainable future [scanning the issue],” *Proceedings of the IEEE*, vol. 100, no. 2, pp. 317–321, 2012.
- [4] E. Lorenz, D. Heinemann, H. Wickramarathne, H. Beyer, and S. Bofinger, “Forecast of ensemble power production by grid-connected PV systems,” in *20th European PV Conference*, 2007, pp. 3–9.
- [5] R. Perez, S. Kivalov, J. Schlemmer, K. Hemker Jr, D. Renné, and T. E. Hoff, “Validation of short and medium term operational solar radiation forecasts in the US,” *Solar Energy*, vol. 84, no. 12, pp. 2161–2172, 2010.
- [6] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom, “Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed,” *Solar Energy*, vol. 85, no. 11, pp. 2881–2893, 2011.
- [7] J. Remund, R. Perez, and E. Lorenz, “Comparison of solar radiation forecasts for the USA,” in *Proc. of the 23rd European PV Conference*, 2008, pp. 1–9.
- [8] E. Lorenz, J. Remund, S. C. Müller, W. Traunmüller, G. Steinmauer, D. Pozo, J. Antonio, V. L. F. Ruiz-Arias, L. Ramirez, M. G. Romeo *et al.*, “Benchmarking of different approaches to forecast solar irradiance,” in *24th European Photovoltaic Solar Energy Conference*, 2009, pp. 1–10.
- [9] P. Mathiesen and J. Kleissl, “Evaluation of numerical weather prediction for intra-day solar forecasting in the continental United States,” *Solar Energy*, vol. 85, no. 5, pp. 967–977, 2011.
- [10] M. Ghonima, B. Urquhart, C. Chow, J. Shields, A. Cazorla, and J. Kleissl, “A method for cloud detection and opacity classification based on ground based sky imagery,” *Atmospheric Measurement Techniques Discussions*, vol. 5, no. 4, pp. 4535–4569, 2012.

- [11] V. T. Jayadevan, J. J. Rodriguez, V. P. Lonij, and A. D. Cronin, "Forecasting solar power intermittency using ground-based cloud imaging," in *Proc. 41st ASES Annual Conference*, vol. 3, 2012, pp. 2100–2106.
- [12] R. Marquez and C. F. Coimbra, "Intra-hour DNI forecasting based on cloud tracking image analysis," *Solar Energy*, vol. 91, no. 0, pp. 327 – 336, 2013.
- [13] J. Shields, M. Karr, T. Tooman, D. Sowle, and S. Moore, "The whole sky imager—a year of progress," in *Eighth Atmospheric Radiation Measurement (ARM) Science Team Meeting, Tucson, Arizona*, 1998, pp. 23–27.
- [14] G. Pfister, R. McKenzie, J. Liley, A. Thomas, B. Forgan, and C. N. Long, "Cloud coverage based on all-sky imaging and its impact on surface solar irradiance," *Journal of Applied Meteorology*, vol. 42, no. 10, pp. 1421–1434, 2003.
- [15] C. N. Long, J. Sabburg, J. Calbó, and D. Pages, "Retrieving cloud characteristics from ground-based daytime color all-sky images," *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, 2006.
- [16] D. Pagès, J. Calbó, C. Long, J. González, and J. Badosa, "Comparison of several ground-based cloud detection techniques," *Extended Abstracts, European Geophysical Society XXVII General Assembly*, pp. 26–28, 2002.
- [17] M. Souza-Echer, E. Pereira, L. Bins, and M. Andrade, "A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera," *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 3, pp. 437–447, 2006.
- [18] "Solar position algorithm for solar radiation applications," *Solar Energy*, vol. 76, no. 5, pp. 577 – 589, 2004.
- [19] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, M. Tignor, and H. Miller, *Climate Change 2007: The physical science basis: Contribution of Working Group I to the fourth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2007.
- [20] J. Shields, R. Johnson, and T. Koehler, "Automated whole sky imaging systems for cloud field assessment," in *Fourth Symposium on Global Change Studies of the American Meteorological Society, Boston, Massachusetts*, 1993.
- [21] S. L. Mantelli Neto, A. von Wangenheim, E. B. Pereira, and E. Comunello, "The use of Euclidean geometric distance on RGB color space for the classification of sky and cloud patterns," *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 9, pp. 1504–1517, 2010.

- [22] M. Yamashita, M. Yoshimura, and T. Nakashizuka, “Cloud cover estimation using multitemporal hemisphere imageries,” in *Proc. XXth Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS04)*. Citeseer, 2004, pp. 818–821.
- [23] Q. Li, W. Lu, and J. Yang, “A hybrid thresholding algorithm for cloud detection on ground-based color images,” *Journal of atmospheric and oceanic technology*, vol. 28, no. 10, pp. 1286–1296, 2011.
- [24] C. H. Li and C. Lee, “Minimum cross entropy thresholding,” *Pattern Recognition*, vol. 26, no. 4, pp. 617–625, 1993.
- [25] A. R. Smith, “Color gamut transform pairs,” in *ACM Siggraph Computer Graphics*, vol. 12, no. 3. ACM, 1978, pp. 12–19.
- [26] R. Kohavi and F. Provost, “Glossary of terms,” *Machine Learning*, vol. 30, no. 2-3, pp. 271–274, 1998.
- [27] D. Panjwani and G. Healey, “Markov random field models for unsupervised segmentation of textured color images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 939–954, 1995.
- [28] K. Held, E. Kops, B. Krause, W. Wells, R. Kikinis, and H. Muller-Gartner, “Markov random field segmentation of brain mr images,” *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 878–886, 1997.
- [29] F. Melgani and S. Serpico, “A markov random field approach to spatio-temporal contextual image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 11, pp. 2478–2487, 2003.
- [30] Q. Li, W. Lu, J. Yang, and J. Z. Wang, “Thin cloud detection of all-sky images using markov random fields,” *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 417–421, 2012.
- [31] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in *Computer Vision-ECCV 2006*. Springer, 2006, pp. 1–15.
- [32] S. Kumar and M. Hebert, “Discriminative random fields: a discriminative framework for contextual interaction in classification,” in *Proceedings. Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1150–1157.
- [33] P. Kohli, L. Ladick, and P. Torr, “Robust higher order potentials for enforcing label consistency,” *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

- [34] X. He, R. Zemel, and M. Carreira-Perpindn, “Multiscale conditional random fields for image labeling,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. II–695–II–702 Vol.2.
- [35] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [36] S. Z. Li, *Markov random field modeling in image analysis*. Springer, 2009.
- [37] C. Sutton and A. McCallum, “An introduction to conditional random fields for relational learning,” *Introduction to statistical relational learning*, vol. 93, pp. 142–146, 2007.
- [38] C. Elkan, “Log-linear models and conditional random fields,” *Tutorial notes at CIKM*, vol. 8, 2008.
- [39] C. Sutton and A. McCallum, “Piecewise training for undirected models,” in *Proceedings. Twenty-First Annual Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 568–575.
- [40] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2012, ISBN 3-900051-07-0. [Online]. Available: <http://www.R-project.org/>
- [41] J. Besag, “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 259–302, 1986.
- [42] Y. Nie and K.-K. Ma, “Adaptive rood pattern search for fast block-matching motion estimation,” *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1442–1449, 2002.
- [43] J. Y. Tham, S. Ranganath, M. Ranganath, and A. Kassim, “A novel unrestricted center-biased diamond search algorithm for block motion estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 369–377, 1998.
- [44] C. Zhu, X. Lin, and L.-P. Chau, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349–355, 2002.
- [45] J. Krupa, “Energy from the desert: Very large scale photovoltaic power-state of the art and into the future,” 2013.