**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**2D AND 3D IMAGE ANALYSIS AND ITS APPLICATION
TO SKY/CLOUD IMAGING**

SOUMYABRATA DEV

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2017

2D AND 3D IMAGE ANALYSIS AND ITS APPLICATION
TO SKY/CLOUD IMAGING

SOUMYABRATA DEV

2017

2D AND 3D IMAGE ANALYSIS AND ITS APPLICATION TO SKY/CLOUD IMAGING

SOUMYABRATA DEV

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2017

Acknowledgment

I would like to offer sincere gratitude to my two wonderful supervisors—Prof. Lee Yee Hui & Dr. Stefan Winkler. Thanks Prof. Lee for admitting me into your research group, and for introducing me into the wonderful world of research and academics. You are a passionate professor, a dedicated researcher and a wonderful friend—sincere thanks for believing in me always, especially during the formative years of my PhD and course work. A huge thanks to Stefan for all your support and guidance throughout my PhD—this thesis is in its current form because of the numerous discussions and regular meetings with you. Thank you for introducing me into the world of *computer vision*, for meticulous reading and editing the papers, and for guiding me throughout this journey.

I have been extremely fortunate to be a visiting Doctoral Assistant at Audiovisual Communication Laboratory (LCAV), at École Polytechnique Fédérale de Lausanne, Switzerland, under the supervision of Prof. Martin Vetterli and Dr. Adam Scholefield, during my PhD. Thank you Martin for this wonderful opportunity—it was my honor to be a part of LCAV and interact with the brilliant minds. I sincerely appreciate your guidance throughout the period, and always inspiring me to be a better researcher. A special shout-out to Adam for your time and continuous support, for all the white-board discussions at *BC 366*, and for providing me all the confidence.

Friends have been an integral part of this PhD journey, and I am truly indebted to them for their love and support. I would like to thank—Prof. J. T. Ong, Dr. Meng Yusong, Dr. Lakshmi Sutha, Ma Ting, Mao Xiao Hong, Yeo Jun Xiang, Ian Bajaj, Mi Siya, Shilpa Manandhar (thanks for the warm friendship and companionship) and Yuan Feng (thanks for the wonderful times). A warm thanks to Ong Kee Wang,

Darko Jokic & Charles Wong for assisting to build the mechanical models of our sky cameras. And of course, Florian Savoy—a true engineer and a wonderful friend; thanks for all the co-authorships, *nuits blanches* before the conference deadlines and for all our technical discussions. *Merci Florian!*

I would also like to thank all my friends I met during my internship at Lausanne. Thanks to Alireza Ghasemi, Mitra Fatemi, Runwei Zhang for being a wonderful office-mate, Niranjana Thanikachalam for all the guidance and friendship, Gilles Baechler for the friendship (and running sessions by the lake), Marta Martinez, and of course, Julien Reichel for being an amazing host. Special thanks to Robin Scheibler for introducing me to *python*, for the numerous discussions over coffee, and for being a wonderful friend. A huge thanks to Radmila Popovic for all the love and support. And finally, special thanks to my dearest friend Miranda Kreković for all the good times spent at Lausanne and beyond—I sincerely appreciate and respect it. *Puno hvala Miranda!*

Thank you Subhrajit, Sumanta, Zishan, Binit, Sana, Ankush, Deval, Sushant and my wonderful flatmates—Jithin, Dharendra, Pravesh & Raunaq (who were no less than my brothers) for all the love and support throughout this journey. I’m sure I have missed mentioning the names of many who were beside me at all the happy and sad times. Singapore is beautiful because of your presence!

I would also like to thank my thesis advisory committee members Prof. Wang Gang and Prof. Cai Jianfei for advising this thesis work. Also, thanks to my PhD oral examination panel members, Prof. Bi Guoan, Prof. Mohammed Yakoob Siyal and Prof. Anamitra Makur for taking their respective time in reading this thesis, and providing valuable feedback on the same. Sincere thanks to the anonymous internal and external PhD thesis examiners, who were amongst the first to closely examine the thesis and provide useful feedback.

I would also like to acknowledge the financial support from Nanyang Technological University Singapore and Ministry of Defence (MINDEF) Singapore; and administrative support from Graduate Programme Office, School of Electrical and Electronic Engineering. Thanks Christina Wong for your continuous help and support, and your infectious laughter definitely made the experience more enjoyable.

Special thanks to Tay TingFang for assisting me throughout the process of my thesis examination (and for acceding to all my technical and logistical requests for my thesis examination).

Thanks to Dr. John Hammersley, Shelly Miller & Dr. Lim Lian Tze from Overleaf for sharing the vision of open science with the community. This entire thesis was written in *Overleaf*; and it was my pleasure to spread the *Overleaf* joy amongst the NTU community, as an advisor.

Finally, I would like to offer my sincere thanks and gratitude to my parents Ashoka Dev & Bidhu Bhusan Dev, my brother Bedabrata Dev and my sister-in-law Rupsa Banerjee. Thank you for your unconditional love and support and for the innumerable sacrifices. Special thanks to my mother who inspired me to pursue a doctoral degree at the very outset!

Abstract

With the ubiquity of powerful digital cameras, there has been a paradigm shift in understanding and analyzing any scene. The central theme of this thesis is to use conventional cameras to capture images of a scene and derive essential information about its content. In particular, we focus our attention on images captured by ground-based sky cameras. Ground-based sky cameras, popularly known as Whole Sky Imagers (WSIs), are now extensively used by remote sensing analysts. They are useful in variety of applications: signal attenuation analysis, local weather prediction, or solar and renewable energy forecasting. They complement conventional weather satellite images with higher temporal and spatial resolutions. The WSIs capture images of the earth’s atmosphere in regular intervals of time. In this thesis, we analyze this huge amount of captured image data, and propose several algorithms and methodologies for various applications.

We designed and built three different versions of ground-based sky cameras, which are efficient, low-cost, and highly flexible. Our custom-built WSIs are equipped with a high resolution fish-eye lens that captures the images of the sky scene at user defined intervals of time. Our imagers are modular with easily configurable hardware and software, offer a high image resolution (5184×3456 pixels), and cost less than US\$3K to build.¹ Furthermore, they give us the flexibility to explore the use of High Dynamic Range Imaging (HDRI) techniques to capture the wide range of luminosity of the sky scene.

With the vast image data collected from these ground-based sky cameras, extract-

¹In comparison, the popular commercial TSI-880 model sold by Yankee Environmental Systems costs around US\$30-35K, and has a far inferior image resolution of only 352×288 pixels.

ing valuable information viz. cloud coverage can be challenging. Cloud segmentation of the images into cloud vs. sky regions is one of the fundamental problems, as clouds are non-rigid, and have a wide variety of shapes, sizes, colors or textures. In this thesis, we present a supervised segmentation framework for ground-based sky/cloud images based on a systematic analysis of different color spaces and components. Unlike other state-of-the-art cloud segmentation methods, our proposed approach is entirely learning-based and does not require any manually defined parameters. As part of this effort, we released a large segmentation database of annotated sky/cloud images to the research community. In addition to cloud segmentation, we also propose a framework for automatic cloud type recognition that classifies sky/cloud image patches efficiently into various cloud types. Our proposed algorithm systematically integrates both *color* and *texture* cues of cloud, and has a good multi-class classification accuracy. We also released a large-scale cloud categories database for further benchmarking purposes.

In most cases of atmospheric study, collocated weather stations are present alongside these sky cameras. Unlike solar pyranometers and other regular meteorological sensors, ground-based sky images have additional information about the continuous evolution of cloud over time. We use our cloud/sky images to propose a solar radiation estimation model, that can accurately capture the short-term fluctuations of solar irradiance. We also use these sequence of images to estimate cloud motion fields, and thereby predict future locations of cloud with a lead time of a few minutes. These solar estimation and cloud forecasting techniques can be applied in the field of solar energy generation and forecasting.

Finally, we use our ground-based sky cameras in a stereo-setup for accurate reconstruction of cloud-base height. We propose a consistent and optimal point localization algorithm for noisy camera poses and error-free matching between image frames. A rigorous analysis of such localization problems provides better insights in cloud-base height reconstruction.

Ground-based whole sky cameras have opened up new opportunities for monitoring the earth’s atmosphere for a range of applications. In this thesis, we have identified a few research gaps and addressed some important problems in the field

of cloud imaging. We believe that the techniques and algorithms developed in this thesis will assist the remote sensing community with a better understanding of clouds and associated phenomena.

Contents

Acknowledgment	iii
Abstract	vi
Contents	xii
List of Figures	xix
List of Tables	xxi
Abbreviations	xxiv
1 Introduction	1
1.1 Background and Motivation	1
1.2 Thesis Outline & Main Contributions	5
2 Literature Review	7
2.1 Ground-based Imaging Instrument	7
2.2 Image Segmentation	9
2.3 Image Type Recognition	11
2.4 Meteorological Sensors & Cameras	12
2.5 Stereo Reconstruction	14
2.6 Summary	15

3	Whole Sky Imager	16
3.1	Introduction	16
3.2	Outline of Our Contribution	17
3.3	General Design of WAHRISIS	17
3.4	WAHRISIS Design	19
3.4.1	Sun-blocker based WAHRISIS Model	20
3.4.2	Actively-cooled WAHRISIS Model	22
3.4.3	Passively-cooled WAHRISIS Model	23
3.5	WAHRISIS Calibration	23
3.5.1	White Balancing	24
3.5.2	Geometric Calibration	26
3.5.3	Vignetting Correction	32
3.6	High Dynamic Range Imaging	34
3.6.1	HDR Image Generation	36
3.6.2	Introducing HDRI techniques in sky cameras	37
3.7	Conclusions	40
4	Color Space Analysis for Cloud Detection	41
4.1	Introduction	41
4.2	Outline of Our Contribution	42
4.3	Color Models for Sky/Cloud Images	42
4.4	Approach and Tools	43
4.4.1	Color Distribution	43
4.4.2	Principal Component Analysis	44
4.4.3	Discrimination Threshold	45
4.4.4	Clustering	46
4.4.5	Kullback-Leibler Divergence	46
4.5	Rough Set Based Color Channel Selection	47
4.5.1	Rough Set Theory	47
4.5.2	Proposed Rough-Set Based Selection Method	49
4.6	Experiments	51

4.6.1	Dataset	51
4.6.2	Distribution Bimodality	52
4.6.3	Principal Component Analysis	52
4.6.4	Rough-set Based Relevance Scores	55
4.6.5	Cloud Classification Performance	55
4.6.6	Benchmarking	57
4.7	Discussion	58
4.8	Conclusions	60
5	Color-based Image Segmentation	61
5.1	Introduction	61
5.2	Outline of Our Contribution	62
5.3	Probabilistic Segmentation	62
5.4	Sky/Cloud Image Databases	65
5.4.1	HYTA Database	65
5.4.2	SWIMSEG Database	66
5.5	Segmentation Results	68
5.5.1	Choice of Color Channels	71
5.5.2	Effect of Database Characteristics	73
5.5.3	Benchmarking	74
5.6	High-Dynamic-Range Imaging for Cloud Segmentation	80
5.6.1	Cloud Segmentation Using HDRI	81
5.6.2	Experimental Evaluation	85
5.7	Conclusions	87
6	Texture-based Image Classification	90
6.1	Cloud Classification	90
6.2	Outline of Our Contribution	92
6.3	Cloud Features	93
6.3.1	Statistical Features (SF)	93
6.3.2	Spatial Gray Level Dependence Matrices (SGLDM)	94

6.3.3	Gray Level Difference Statistics (GLDS)	94
6.3.4	Fractal Feature (FF)	95
6.3.5	Fourier Magnitude (FM)	95
6.3.6	Heinle Feature (HF)	95
6.3.7	Local Binary Pattern (LBP)	95
6.3.8	Scale Invariant Feature Transform (SIFT)	96
6.4	Proposed Classification Approach	96
6.4.1	Feature Descriptor	98
6.4.2	Discriminative Model	98
6.4.3	Classification Metric	99
6.5	Experimental Evaluations	100
6.5.1	Evaluation Metric	100
6.5.2	SWIMCAT Database	101
6.6	Classification Results	103
6.6.1	Cloud Features	103
6.6.2	Optimal Parameter Selection	105
6.6.3	Benchmarking Methods	108
6.7	Conclusion	109
7	Weather Parameters Estimation from Images	111
7.1	Introduction	112
7.2	Outline of Our Contribution	113
7.3	Data Collection	114
7.3.1	Whole Sky Imager (WSI)	115
7.3.2	Weather Station	116
7.4	Modeling Solar Irradiance	118
7.4.1	Cosine-weighted hemispheric sampling	118
7.4.2	Relative Luminance Calculation	120
7.4.3	Modeling Irradiance from Luminance Values	121
7.4.4	Performance Comparison and Validation	123
7.4.5	Discussion	127

7.5	Detecting Rainfall Onset from Sky Cameras	130
7.5.1	Impact of Precipitation	130
7.5.2	Methodology & Results	132
7.5.3	Summary	135
7.6	Short-term Cloud Tracking Using Sky Images	136
7.6.1	Methodology	136
7.6.2	Results & Discussions	140
7.6.3	Summary	141
7.7	Conclusion	141
8	3D Scene Analysis Using Multiple Cameras	143
8.1	Outline of Our Contribution	144
8.2	Introduction of Localization Problem	144
8.3	Problem Set-up	146
8.4	SHARP: A Consistent and Optimal Localization Algorithm	149
8.5	SHARP with Noisy Camera Poses	150
8.5.1	Motivation & Insights	151
8.5.2	Geometric Constraints	156
8.6	Proposed Algorithm	158
8.6.1	Noise Model	158
8.6.2	Proposed Approach	160
8.6.3	Computation of Proposed Approach	160
8.7	Experimental Evaluations	163
8.7.1	Effect of Noise in View Angle	164
8.7.2	Effect of Noise in X-component of Camera Positions	164
8.7.3	Effect of Noise in Y-component of Camera Positions	165
8.8	Application to Ground-based Sky Cameras	166
8.8.1	Workflow	167
8.8.2	Validation	171
8.9	Conclusion	173

9 Conclusion & Future Work	175
9.1 Conclusion	175
9.2 Future Work	177
A Calculating Sun Position from Earth’s Surface	181
B Calculating the Position of Sun Blocker	182
C Derivation of Normalization Factor	184
D Computation of Geometrical Bounds in a Multi-camera Setup	185
Notations	206
Author’s Publications	209

List of Figures

1.1	Illustration of cloud attenuation in satellite communication links. . .	2
3.1	General design of WAHRISIS.	18
3.2	Server-client setup in WAHRISIS.	19
3.3	Schematic drawing of WAHRISIS.	21
3.4	Actively-cooled WAHRISIS design along with its different components.	23
3.5	Passively-cooled WAHRISIS design along with its different components.	24
3.6	Captured images using automatic and custom white balance settings.	25
3.7	Example of chromatic aberration and calibration function	26
3.8	Illustration to show the advantages of color-channel-specific calibration as compared to grayscale calibration.	27
3.9	Schematic representation of object space and image plane	28
3.10	Sample of the checker-board images used for the calibration	29
3.11	Result of the calibration process compared to the various theoretical projection models.	30
3.12	Re-projection errors as a function of the distance to the image center.	31
3.13	Image captured inside the integrating sphere.	33
3.14	Luminance as a function of the distance from the image center. . . .	34
3.15	Correction coefficients in red and scatter plot of all the values in blue.	35
3.16	Example of three LDR images captured by sky camera.	35
3.17	Response curve of the HDRI system, computed from three LDR images with different exposure settings.	36
3.18	Tone-mapped image computed with the pictures shown in Fig. 3.16 .	37

3.19	Effect of sun glare with different strategies for sun occlusion.	39
4.1	Visual representation of 16 color channels for a sample sky/cloud image.	44
4.2	Pictorial illustration of different rough-set terminology.	48
4.3	Sample images from the HYTA database along with corresponding sky/cloud segmentation ground truths.	51
4.4	Distribution of the variance across the principal components for all images of the HYTA database.	53
4.5	Distribution of triangular area enclosed by two given color channels projected on the plane spanned by the first and second principal components.	54
4.6	Average accuracy and F-score values for 16 color channels across all image of HYTA database.	56
4.7	Scatter plot between average accuracy and different benchmarking measures, while ranking the different color channels.	58
4.8	F-score values of 16 color channels for 1D and 2D clustering cases. . .	59
5.1	Illustration to show vignetting- and color-correction.	67
5.2	Generation of undistorted images using ray-tracing.	68
5.3	Sample images from the SWIMSEG database along with its corresponding ground truths.	69
5.4	Distribution of images in SWIMSEG according to various characteristics.	70
5.5	Binary classification performance for different color channels.	72
5.6	Distribution of average F-score value as a function of various characteristics of the images in SWIMSEG.	74
5.7	ROC curves of our proposed segmentation approach for HYTA and SWIMSEG databases.	77
5.8	Segmentation results of sample images from HYTA and SWIMSEG databases for different benchmarking algorithms.	79
5.9	Proposed HDRSeg cloud segmentation approach.	81
5.10	Illustrative example to show the ratio image and probabilistic segmented image.	83

5.11	Illustrating explaining the proposed HDRSeg framework.	85
6.1	Proposed texton-based framework for cloud type recognition.	99
6.2	Various cloud classes of SWIMCAT database.	102
6.3	Process to generate image patches of fixed dimension 125×125	103
6.4	Visual representation of various cloud features in lower dimensional subspace.	104
6.5	Confusion matrix of cloud classification results.	110
7.1	Impact of clouds on direct solar irradiance.	115
7.2	Measured solar radiation, and its deviation from the clear-sky model over a day.	117
7.3	Cosine weighted hemispheric sampling process used to select the pixels used for solar irradiance estimation.	119
7.4	Proposed linear regressor model to estimate solar irradiance from sky images.	122
7.5	Solar radiation and image luminance follows each other closely over the entire period of the day.	124
7.6	Histogram of difference between estimated and actual solar radiation. We observe that most of the data are concentrated in the 0-bin. . . .	124
7.7	Effect of the percentage of training and testing images on the RMSE of proposed model.	127
7.8	Comparison of several solar estimation models amongst each other. .	127
7.9	Scatter plot between measured solar radiation and estimated solar radiation for the benchmarking algorithms.	128
7.10	Distribution of F-number of the WAHRSIS images that are used to derive the proposed model.	129
7.11	Effect of the onset of rainfall on the measured luminance from sky camera.	131
7.12	Trend of Clearness Luminance Index (I) w.r.t. proximity of rainfall onset.	133

7.13	CDF plots of Clearness Luminance Index of images within and outside ± 15 minutes of rain event.	134
7.14	Operating Characteristics (OC) curve generated by varying the critical clearness luminance index (I_c).	135
7.15	Ratio channels of two successive image frames.	137
7.16	Horizontal and vertical translation of pixels between two successive sky/cloud image frames.	138
7.17	Prediction of sky/cloud image for a sample day with a lead time of 2 and 4 minutes.	139
7.18	Comparison of prediction accuracy percentage with different lead times.	140
8.1	Guarding a non-convex shape art gallery from an intruder $\mathbf{P}$	145
8.2	Localizing an intruder $\mathbf{P}$ in <i>any</i> room type using infinite resolution cameras.	146
8.3	Region of interest is an <i>area</i> instead of a particular point, when we deal with finite resolution pixel camera.	147
8.4	Acquisition system with one camera, mentioning the various camera pose notations.	148
8.5	Acquisition system with multiple cameras, mentioning the various camera pose notations.	149
8.6	Illustration of half-space intersections to generate a closed convex polygon in a multiple-camera setup.	150
8.7	Illustration of a consistent region when the camera poses are noiseless.	151
8.8	Illustration of a non-consistent region when the camera poses are noisy.	151
8.9	Illustration of a single camera with finite pixel resolutions.	152
8.10	Distribution of active region with noiseless camera view angle.	153
8.11	Illustration of a single camera with noisy view angle.	154
8.12	Computation of upper bound on the view angle.	154
8.13	Computation of lower bound on the view angle.	155
8.14	Distribution of active region with noise only in view angle.	155

8.15	Depicting 6 distinct scenarios in $\mathbb{R}^2$, when the noise is present in camera locations.	156
8.16	Determination of upper bound of view angle for the rotation of camera.	157
8.17	Determination of lower bound of view angle for the rotation of camera.	158
8.18	Experimental set-up with three cameras having random camera poses.	164
8.19	Effect on re-projection error with varying noise in view angle.	165
8.20	Effect on re-projection error with varying noise in X co-ordinate of the camera positions.	166
8.21	Effect on re-projection error with varying noise in Y co-ordinate of the camera positions.	167
8.22	Images captured by the pair of sky imagers.	168
8.23	Illustration of the undistortion process of an image taken with our fish-eye lens of WAHRIS.	168
8.24	Illustration of the rectification process of the fish-eye effect.	170
8.25	Image pair before and after rectification.	170
8.26	Illustration of the computed disparity map.	171
8.27	Illustration of the computation of a point in 3D.	172
8.28	3D point cloud computed for a pair of images from the sequence.	172
8.29	Image pair from the computer-generated sequence.	173
8.30	Illustration of 3D point cloud for an image pair computer from the computer-generated sequence.	173
9.1	Illustration to show the benefits of near-IR capture in cloud imaging.	178
B.1	Representation of WAHRIS with the lengths and angles required for the computation of the motor angles.	183
D.1	Determination of upper bound for the translation of camera in X-direction (Scenario II).	186
D.2	Determination of lower bound for the translation of camera in X-direction (Scenario II).	186

D.3	Determination of upper bound for the translation of camera in X-direction (Scenario III).	187
D.4	Determination of upper bound for the translation of camera in X-direction (Scenario III).	188
D.5	Probability is 0 for the translation of camera in X-direction (Scenario IV).	189
D.6	Probability is 0 for the translation of camera in X-direction (Scenario V).	189
D.7	Determination of lower bound for the translation of camera in X-direction (Scenario VI).	190
D.8	Determination of upper bound for the translation of camera in X-direction (Scenario VI).	190
D.9	Determination of lower bound for the translation of camera in Y-direction (Scenario II).	191
D.10	Determination of upper bound for the translation of camera in Y-direction (Scenario II).	191
D.11	Probability of the point C is 0 for the translation of camera in Y-direction (Scenario III).	192
D.12	Probability of the point C is 0 for the translation of camera in Y-direction (Scenario IV).	192
D.13	Determination of lower bound for the translation of camera in Y-direction (Scenario V).	193
D.14	Determination of upper bound for the translation of camera in Y-direction (Scenario V).	193
D.15	Determination of upper bound for the translation of camera in Y-direction (Scenario VI).	194
D.16	Determination of lower bound for the translation of camera in Y-direction (Scenario VI).	194
D.17	Determination of lower bound of view angle for the rotation of camera (Scenario II).	195

D.18 Determination of upper bound of view angle for the rotation of camera (Scenario II).	195
D.19 Determination of lower bound of view angle for the rotation of camera (Scenario III).	196
D.20 Determination of upper bound of view angle for the rotation of camera (Scenario III).	196
D.21 Probability of the point C is 0 for the rotation of camera (Scenario IV).	197
D.22 Determination of upper bound of view angle for the rotation of camera (Scenario V).	197
D.23 Determination of lower bound of view angle for the rotation of camera (Scenario V).	197
D.24 Determination of upper bound of view angle for the rotation of camera (Scenario VI).	198
D.25 Determination of lower bound of view angle for the rotation of camera (Scenario VI).	198

List of Tables

2.1	Overview of various ground-based whole sky imagers and their intended applications.	9
3.1	Cost analysis for all the components of WAHRISIS.	22
3.2	Comparison of the different WAHRISIS models.	40
4.1	Color spaces and components used for analysis.	43
4.2	PBI values for all color channels.	52
4.3	Percentage of variance captured across different principal components for HYTA database.	52
4.4	Average relevance value across all images of the HYTA database. . . .	55
4.5	Correlation of cloud classification accuracy with ranking scores obtained using different methods.	57
5.1	Number of training and test images for HYTA and SWIMSEG databases.	71
5.2	Performance evaluation of different benchmarking algorithms.	80
5.3	Comparison of HDRSeg segmentation algorithms with other benchmarking algorithms.	87
6.1	Extensive evaluation of different parameters in our proposed cloud classification approach.	106
6.2	Various sets of filters in the filter bank of our proposed cloud classification approach.	107
6.3	Benchmarking of various cloud classification algorithms.	109

7.1	Benchmarking of our proposed approach with other solar radiation estimation models.	126
-----	---	-----

Abbreviations

AEB	Automatic Exposure Bracketing
BOW	Bag-Of-Words
CDF	Cumulative Distribution Plot
CLAHE	Contrast-Limited Adaptive Histogram Equalization
colorHIST	Color Histogram
CRE	Cloud Radiative Effect
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DR	Dynamic Range
dSIFT	Dense Scale Invariant Feature Transform
EV	Exposure Value
FF	Fractal Feature
FFT	Fast Fourier Transform
FM	Fourier Magnitude
FN	False Negative
FP	False Positive
GEOSS	Global Earth Observation System of Systems
GHI	Global Horizontal Irradiance
GLCM	Gray-Level Co-occurrence Matrix
GLDS	Gray Level Difference Statistics
GPS	Global Positioning System
HDR	High Dynamic Range
HDRI	High Dynamic Range Imaging
HF	Heinle Feature

HYTA	Hybrid Thresholding Algorithm
ICESat	Ice, Cloud, and land Elevation Satellite
ICLTP	Illumination-invariant Completed Local Ternary Pattern
IR	Infra-Red
KL	Kullback-Leibler
LBP	Local Binary Patterns
LCAV	Audiovisual Communications Laboratory
LDR	Low Dynamic Range
LED	Light-Emitting Diode
LiDAR	Light Detection And Ranging
MODIS	Moderate-Resolution Imaging Spectroradiometer
MSS	Meteorological Service Singapore
NASA	National Aeronautics and Space Administration
NIR	Near-Infra-Red
OC	Operating Characteristics
PBI	Pearson's Bimodality Index
PCA	Principal Component Analysis
PLS	Partial Least Squares
PV	Photo Voltaic
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
SDK	Software Development Kit
SGLDM	Spatial Gray Level Dependence Matrices
SHARP	Sequential Half space Aggregation for Reconstructing a Point
SIFT	Scale Invariant Feature Transform
SLIC	Simple Linear Iterative Clustering
SVM	Support Vector Machine
SWIMCAT	Singapore Whole sky IMaging CATegories database
SWIMSEG	Singapore Whole sky IMaging SEGmentation database
TN	True Negative
TP	True Positive

USB	Universal Serial Bus
WAHRSIS	Wide Angle High Resolution Sky Imaging System
WMO	World Meteorological Organization
WSI	Whole Sky Imager
YES	Yankee Environmental Systems

Chapter 1

Introduction

1.1 Background and Motivation

The study of clouds in the earth’s atmosphere is becoming popular amongst the remote sensing community for a variety of applications and domains—aviation, weather prediction, and solar energy. Furthermore, ground-to-satellite or ground-to-air communication signals suffer from substantial attenuation because of clouds in the atmosphere. Rain, clouds, atmospheric particles, and water vapor along the signal path affect communication links in various ways.

Sky pictures are commonly used in the analysis and prediction of signal attenuation due to clouds. Imaging the cloud in the atmosphere can be performed in different ways. Satellite imagery [1,2] and aerial photographs [3] are popular in particular for large-scale surveys; airborne light detection and ranging (LiDAR) data are extensively used for aerial surveys [4]. However, these techniques rarely provide sufficient temporal and/or spatial resolution for localized and short-term cloud analysis over a particular area. This is where ground-based whole sky imagers (WSIs) offer a compelling alternative [5]. The images obtained from these devices provide high-resolution data about local cloud formation, movement, and other atmospheric phenomena.

With the rapid advancement in photogrammetric techniques, high resolution cam-

eras are now extensively used for almost all applications. The existence of ubiquitous cameras around us generate a huge amount of data, which can be effectively used to solve most of the fundamental problems in the field of remote sensing. Extracting valuable information from the huge amount of image data by detecting and analyzing the various entities in these images is challenging. In this thesis, we attempt to answer some of the fundamental questions in the cloud imaging by connecting the remote sensing and computer vision communities.

Figure 1.1: Attenuation: The effects of the atmosphere on signal transmitted from geostationary satellites. The rain, cloud and the other atmospheric particles contributes a noise signature that ultimately degrades the fidelity of the satellite signals. Image is archived from NASA, Glenn Research Center.

Our objective is to use whole sky imagers to provide an analysis of the effect of clouds in signal strength attenuation during satellite and military communications. Satellite communication links suffer from appreciable attenuation in the space to ground communication because of the rain, cloud, atmospheric particles, water vapor particles in its signal path [6, 7]. Figure 1.1 shows a satellite slant path which is affected by propagation impairments. The images obtained from the whole sky imagers will help us to detect and identify different type of clouds, analyze the

small-scale movement which will subsequently help us in deriving a model for signal attenuation of satellite and air-to-ground communication links.

We aim to understand the effects of clouds on communications using ground-based whole sky imagers. These ground-based devices capture the entire sky at regular intervals of time. The resulting images are of higher resolution than what can be obtained from satellites, and the upwards pointing nature of the camera makes it easy to capture low-lying clouds.

The objectives of this thesis can be clearly defined as:

- To design and build low-cost and high image resolution sky cameras.
- To identify the best color channels for cloud detection.
- To propose a color-based image segmentation algorithm.
- To classify cloud types into various categories, as identified by World Meteorological Organization (WMO).
- To estimate various weather parameters using ground-based sky imagers.
- To analyze point localization error in a multi-camera set-up and to localize cloud base using stereo cameras.

The current approaches for such problems lack a systematic analysis, proper benchmarking in public databases and better performance. Therefore, we have identified these problems and propose the following solutions for these various tasks.

We present the design of the first low-cost, high image resolution sky camera designed for tropical countries like Singapore. We name our imagers **WAHR**SIS, that stands for **W**ide **A**ngle **H**igh **R**esolution **S**ky **I**maging **S**ystem. We present three models of WAHRSIS.

Using several statistical tools viz. Principal Component Analysis (PCA), bimodality analysis and clustering techniques, we present a structured and systematic analysis of the various existing color channels. We propose a rough-set based color channel selection method that accurately estimates the efficiency of different color channels

for cloud detection. Our method is the first that employs rough-set based techniques to solve this problem of color channel selection. Such rough-set methods provide us with mathematical tools to define the *approximate* ground-truth maps of sky/cloud images.

We introduce the first large-scale, publicly available sky/cloud image database with manually annotated segmentation masks. We refer to this database as **SWIM-SEG**: Singapore **W**hole sky **I**Maging **SEG**mentation database. Unlike the conventional binary segmentation algorithms, we propose a probabilistic segmentation method. In our method, instead of *hard* labeling, we assign a *probability* to a pixel to belong to cloud category. We also benchmark with the state-of-the-art cloud segmentation algorithms.

We also propose a robust cloud classification framework for identifying and recognizing cloud types in tropical regions like Singapore. We propose a filter-bank based cloud classification framework that systematically integrates both *color* and *texture* cues of sky/cloud images, and benchmark with other existing algorithms. We also release the first large-scale cloud categorization database to the research community. We refer to this database as **SWIMCAT**, that stands for **S**ingapore **W**hole sky **I**Maging **C**ATegories database.

We propose a model to estimate the instantaneous solar radiation from the corresponding ground-based sky/cloud images. Extensive benchmarking w.r.t. other existing solar estimation algorithms show the efficacy of our approach. We also propose a method to detect the onset of precipitation from sky/cloud images. Finally, we present a methodology to perform short-term cloud forecasts using optical flow techniques.

Finally, we present a localization algorithm in a multiple-camera setup, under the assumption of error-free matching and noisy camera poses. We benchmark our algorithm with other state-of-the-art point localization algorithms. We present a framework for estimating the cloud-base height, using a pair of sky cameras.

1.2 Thesis Outline & Main Contributions

Chapter 1: Introduction We discuss the objectives and motivation of the thesis in this chapter. We also describe the outline of the thesis, and briefly highlight the main contributions which were offered throughout this thesis.

Chapter 2: Literature Review In this chapter, we provide a thorough literature review of the various aspects of this thesis. We give a general account of the current state-of-the-art practices, and also aim to identify a few research gaps in the current literature.

Chapter 3: Whole Sky Imager The central theme of this thesis involves the use of high-resolution, ground-based sky imagers for continuous monitoring of the earth’s atmosphere. Unlike conventional satellite images, ground-based sky cameras have high temporal and spatial resolutions. In this chapter, we discuss the methodology for spatial-, color- and luminance- calibration for our custom-built, low-cost sky imagers. Moreover, the images captured by regular sky cameras are often saturated because of the direct glare from the sun. In this chapter, we show how High Dynamic Range Imaging (HDRI) techniques can solve this problem of over-saturation.

Chapter 4: Color Spaces and Components Our ground-based imagers called WAHRISIS (cf. Chapter 3) captures the images of the atmosphere at regular intervals of time, and stores the captured images in a server. The first task of cloud analysis is to accurately detect the cloud pixels in sky/cloud images. This is an extremely challenging task, as clouds are non-rigid and do not have specific shape or size. Therefore, *color* is used as the most discriminatory feature for cloud detection. This chapter deals with a structured review of the various color spaces and components that are generally used in cloud analysis.

Chapter 5: Color-based Image Segmentation A structured and systematic analysis of the various color channels of sky/cloud images assist us in identifying the most discriminatory color channels. In this chapter, we use the identified discriminatory color channel and discuss the approach of our color-based image segmentation algorithm.

Chapter 6: Texture-based Image Classification One of the important

tasks after successful cloud detection, is to identify and categorize the different cloud types. In this chapter, we deal with multi-class classification of sky/cloud image patches. We perform a systematic analysis of various image features that were designed specifically for this task.

Chapter 7: Weather Parameters Estimation from Images In the previous chapters, we discussed our proposed methodology to perform color-based segmentation and texture-based classification on sky/cloud images. In this chapter, we discuss about the impact of clouds on solar and renewable energy generation. In tropical regions like Singapore, the weather phenomenon is highly localized and clouds often block the sun. This greatly impacts the incoming solar radiation falling on the earth's surface. Therefore, the generated solar energy drops drastically in a short span of time. Such scenarios are not conducive for Photo Voltaic (PV) power generation. It is extremely difficult to model these sharp variations of the incoming solar radiation. In this chapter, we analyze the impact of clouds on such incoming solar radiation and also discuss our current works on cloud tracking and detecting rainfall onset.

Chapter 8: 3D Scene Analysis Using Multiple Cameras The earlier chapters deal with the study and analysis of images, captured from a single camera. In this chapter, we extend our discussion to the task of scene analysis using multiple cameras. The first part of this chapter deals with the introduction of a novel localization algorithm in the setup of multiple cameras, having finite pixel dimension and noisy camera poses. The remaining part of this chapter discusses a practical application of point localization task. We use scene flow techniques on stereo images captured by ground-based sky cameras to localize cloud mass.

Chapter 9: Conclusion & Future Work We conclude the thesis in this chapter. Future works using ground-based sky camera are also discussed. Following this chapter, we provide a detailed list of the various notations that were used throughout this thesis, for easy reference.

Chapter 2

Literature Review

In this chapter, we provide a thorough literature review of the various aspects of this thesis. As mentioned in the previous chapter, this thesis involves the use of camera(s) to understand a scene better, and learn important cues of its surroundings. In most cases, we apply our proposed algorithms and frameworks in the context of cloud imaging using ground-based whole sky imagers. In this chapter, we describe the existing works of this particular field. We attempt to identify a few existing research gaps in the literature and bridge most of them by proposing state-of-the-art algorithms and best practices.

2.1 Ground-based Imaging Instrument

Satellite images are commonly used to monitor the earth and analyze its various phenomena. They provide remote sensing analysts with accurate information about various earth's events. Satellite images are available in different spatial and temporal resolutions and also across various ranges of the electromagnetic spectrum, including visible, near- and far-infrared regions. For example, multi-temporal satellite images are extensively used for monitoring forest canopy changes [8] or evaluating sea ice concentrations [9].

The presence of clouds plays a very important role in the analysis of satellite

images. NASA’s Ice, Cloud, and land Elevation Satellite (ICESat) has demonstrated that 70% of the world’s atmosphere is covered with clouds [10]. Therefore, there has been renewed interest amongst the remote sensing community to further study clouds and their effects on the earth.

Satellite images are a good starting point for monitoring the earth’s atmosphere. However, they have either high temporal resolution (e.g. geostationary satellites) or high spatial resolution (e.g. low-orbit satellites), but never both. In many applications like solar energy production [11], local weather prediction, tracking contrails at high altitudes [12], studying aerosol properties [13], attenuation of communication signals [14, 15], we need data with high spatial *and* temporal resolution. This is why ground-based sky imagers have become popular and are now widely used in such applications. The ready availability of high-resolution cameras at a low cost facilitated the development of various models of sky imagers. These sky cameras are popularly known as Whole Sky Imagers (WSIs), as they capture the whole sky scene with its wide-angled lens. In geographical small countries like Singapore, ground-based whole sky imagers offer remote sensing analysts a fantastic alternative, as compared to satellite images. These sky cameras have high temporal and spatial resolutions. However, conventional sky cameras are expensive, have limited flexibilities and low image resolutions. We address this in Chapter 3, and present custom-built sky cameras.

A WSI consists of an imaging system placed inside a weather-proof enclosure that captures the sky at user-defined intervals. A number of WSI models have been developed over the years.

A commercial WSI (TSI-440, TSI-880) manufactured by Yankee Environmental Systems (YES) is used by many researchers [20, 24, 27]. Owing to the high cost and limited flexibility of commercial sky imagers, many research groups have built their own WSI models [16–19, 21–23, 26]. For example, the Scripps Institution of Oceanography at the University of California San Diego has been developing and using WSIs as part of their work for many years [28]. Table 2.1 provides an overview of the types of ground-based sky cameras used by various organizations around the world, and their primary applications.

Application	Organization	Country	WSI Model
Air traffic control [16]	Campbell Scientific Ltd.	United Kingdom	IR NEC TS9230
Cloud characterization [17]	Atmospheric Physics Group	Spain	GFAT All-sky imager
Cloud classification [18]	Brazilian Institute for Space Research	Brazil	TSI-440
Cloud classification [19]	Laboratory of Atmospheric Physics	Greece	Canon IXUS II with FOV 180°
Cloud macrophysical properties [20]	Pacific Northwest National Laboratory	United States	Hemispheric Sky Imager
Cloud track wind data monitoring [21]	Laboratoire de Météorologie Dynamique	France	Nikon D100 with FOV 63°
Convection [22]	Creighton University	United States	Digital Camera
Radiation balance [23]	Lindenberg Meteorological Observatory	Germany	VIS/NIR 7
Solar power forecasting [24]	Solar Resource Assessment & Forecasting Laboratory	United States	TSI-440
Weather monitoring [25]	Pacific Northwest National Laboratory	United States	TSI-880
Weather reporting [26]	Ecole Polytechnique Fédérale de Lausanne	Switzerland	Panorama Camera

Table 2.1: Overview of various ground-based whole sky imagers and their intended applications.

These ground-based sky cameras capture images at regular intervals of time, and stores them in local or remote server. The captured sky/cloud images are subsequently analyzed for various applications. The segmentation of clouds from these images is one of the fundamental tasks—existing works on cloud detection are discussed in the subsequent Section 2.2 of this chapter.

2.2 Image Segmentation

Segmentation is one of the first steps in sky/cloud image analysis. It remains a challenging task because of the non-rigid, feature-less, and poorly-defined structure of clouds. Also, the shape of clouds also changes continuously over time. Thus, classical image segmentation approaches based on shape priors [29] are not suitable. Furthermore, the wide range of lighting conditions (direct sunlight to completely covered skies) adds to the difficulty.

As color is the most discriminating feature in sky/cloud images, most works in the literature use color for cloud segmentation. Long et al. [20] showed that the ratio of red and blue channels from RGB color space is a good candidate for segmentation and tuned corresponding thresholds to create binary masks. Calbó and Sabburg [30] use the same (R/B) ratio to derive statistical features (mean, standard deviation, entropy etc.) of the clouds and subsequently classify the sky/cloud images into different cloud types. Heinle et al. [31] exploited the difference of red and blue channels for successful detection and subsequent labeling of pixels. Liu et al. [32] also used the difference of red and blue channels in their superpixel-based cloud segmentation framework. Souza et al. [27] used the Saturation (S) channel for calculating cloud coverage. Mantelli-Neto et al. [18] investigated the locus of cloud pixels in the RGB color model. Li et al. [33] proposed cloud detection using an adaptive threshold technique in the normalized blue/red channel. Yuan et al. [34] proposed a cloud detection framework using superpixel classification of image features.

In these existing methods of the current literature for cloud segmentation, the selection of color models and channels has not been studied systematically. Many existing approaches [30–33, 35] use combinations of red and blue channels, which is a sensible choice, because the sky is predominantly blue due to the Rayleigh scattering of light at shorter wavelengths. However, we are not aware of any experimental analysis presented regarding the efficacy of these color channels in sky/cloud image segmentation. Furthermore, all of the above methods rely on manually-defined parameters and case-based decisions for segmentation. These make the methods somewhat ad-hoc and prone to errors. Finally, most of them assign binary labels by design, which further reduces their flexibility and robustness. We address this limitation in Chapter 4, and propose a probabilistic color-based image segmentation algorithm in Chapter 5.

The successful detection of clouds in sky/cloud images is also helpful for its classification into various cloud types. Such cloud-type recognition tasks are generally done manually. However, a few approaches have been recently developed for such tasks—we discuss them in the next Section 2.3 of this chapter.

2.3 Image Type Recognition

For varied applications viz. aviation, weather prediction, and solar energy, clouds are continuously monitored by experts from meteorological institutes. In most cases, cloud observation is done manually by the experts. Cloud experts from the meteorological institutes observe these clouds to better understand the atmospheric phenomenon; and classify the clouds into various categories as identified by World Meteorological Organization (WMO) [36]. They report the different cloud parameters such as cloud type, cloud base height, cloud coverage regularly. These observations are performed manually by cloud experts who have received extensive training in this field. Though accurate, such manual observations are highly expensive and cumbersome. Therefore, it is necessary to employ automatic cloud classification algorithms which are systematic and less expensive as compared to its human counterpart.

Automatic cloud type recognition is done primarily from satellite or ground-based sky cameras. Traditionally, satellite images were mostly used for such tasks. However, these images suffer from low temporal and spatial resolutions, and thus cannot be used in geographically small countries like Singapore. Cloud formation and the weather conditions are highly localized in such countries. Therefore, we use images from WSIs for automatic cloud classification task.

The WMO recommends a genera-based classification [36], which defines how to classify cloud patches into various categories based on their shape, structure, transparency, arrangement inside the cloud, texture, color, and height of their base. For example, cumulus clouds are puffy in shape and are regularly patterned; stratus clouds are mostly featureless, and are characterized by their transparent, whitish veil appearance; nimbostratus clouds are rain clouds, having thick dark appearance; and so on. Although clouds do not exhibit *strong* image-based features, we can still exploit the color and texture of the cloud to extract discriminatory cues.

In the literature, several features were developed for automatic cloud type recognition. Some of the earlier methods used co-occurrence and auto-correlation matrices [37], Fourier transformation in classifying pre-defined sky conditions [30]. Heinle et al. [31] extracts several statistical features from pre-defined sky conditions and

uses k-nearest neighbor classifier during the classification step. In order to exploit the textural information of the clouds, classical texture descriptors such as Scale-Invariant Feature Transform (SIFT) [38] and Local Binary Patterns (LBP) [39] have been also exploited. Considering simplicity and efficiency, LBP and its variants have been used extensively [40, 41]. However, they cannot be an ideal technique in cloud classification task because of the inherent varying illumination conditions and dependency on camera parameters. Liu et al. [42] has proposed a descriptor called Illumination-invariant Completed Local Ternary Pattern (ICLTP) for cloud classification tasks.

Most of the existing approaches in the literature use a combination of color and texture features, which make sense, as different cloud types can be categorized mainly based on these discriminatory cues. However, there is no rigorous experimental analysis on the efficacy of these features for such multi-class classification framework. We provide a systematic evaluation of different cloud features in Chapter 6, and propose a complete framework for cloud type recognition in Singapore.

In most cases, collocated meteorological sensors are also present with sky cameras. Such scenario provides us ample opportunities to establish the connections between them. Weather sensors are point-devices that collect data for a particular point on the earth's surface. On the other hand, sky cameras equipped with wide-angled lenses provide us more information about the evolution of cloud. This motivates us to identify a few research problems and motivate us to develop techniques for a few intended applications—we will discuss such related works in the next Section 2.4 of this chapter.

2.4 Meteorological Sensors & Cameras

In addition to ground-based sky cameras, the other meteorological sensors measuring solar radiation, precipitation, wind speed are used in tandem to provide a holistic analysis of various phenomena in earth's atmosphere. Ground-based meteorological sensors are mainly point devices that measure and record the various weather parameters at a particular location on earth's surface. However, collocated sky cameras

can provide us more information about the evolution of clouds over a period of time. Therefore, it helps us to use these sky camera images to estimate and/or predict some of the weather parameters. It greatly helps the remote sensing analysts for application areas such as solar energy generation and study of cloud attenuation.

Firstly, meteorological sensors and cameras were extensively used in the field of solar energy generation. These sensors provide useful information about clouds—it is necessary as clouds have a significant impact on solar energy generation. The clouds intermittently block the sun and significantly reduce the solar irradiance reaching solar panels. A short-term forecast of the solar irradiance is needed for the grid operators to mitigate the effects of a power generation ramp-down. The daily variations in temperature, pressure and humidity, coupled with clear-sky model of solar radiation were used in the literature to estimate the instantaneous solar radiation. Several attempts have been done to estimate the solar radiation from general meteorological measurements viz. temperature, humidity and precipitation [43–46]. These existing models aim to provide global solar radiation using different sensors. Recently, with the development of low-cost photogrammetric techniques, sky cameras are being deployed for such purposes. Alonso-Montesinos and Batlles used sky cameras to quantify the total solar radiation [47]. However, it could not model the sharp short-term variations of solar radiation.

Secondly, sky cameras can also be used to identify the onset of precipitation. In the literature, rainfall onset is studied mainly using long-term climate data. Rainfall measurements from weather stations are analyzed to detect hidden patterns, and subsequently used to predict the onset of precipitation [48]. Precipitation radar is also used to measure tropical rainfall with high accuracy [49]. However, these studies are based on long-term rain forecasting. Recently, sky cameras have been used to estimate the clear sky intensity distribution [50]. To the best of our knowledge, no prior work have used sky cameras to detect rainfall onset.

And finally, sky cameras are regularly used to detect and predict future location of clouds. Cloud tracking is an essential task in the field of solar energy generation and forecasting. Traditionally, cloud tracking has been performed from satellite images for accurate weather prediction. Pioneer work of tracking involved detecting

optical flow patterns in satellite images [51]. These flow patterns are useful to detect the evolution of several weather patterns. This can be seen in [52], where Sieglaff et al. fused data from satellites images, radar, and numerical models to understand the evolution of convective clouds. Very recently, estimation of cloud motion from sky cameras has been used in solar irradiance forecasting [53, 54]. This seamless cooperation of sensor data from various sources helps in creating robust frameworks and best practices for numerous applications. This is comparatively a new development in remote sensing, as most of the existing works are based on long-term climate statistics. We aim to complement such studies and establish new methodologies using sky cameras—more details are discussed later in Chapter 7 of this thesis.

Such application areas of solar energy forecasting also require a fair estimate of base height of cloud mass, along with predicting its future location. Most of the existing works use a pair of (or multiple) sky cameras for reconstructing cloud base. We discuss these existing works in the last Section 2.5 of this chapter.

2.5 Stereo Reconstruction

Estimating the precise localization of clouds in the atmosphere is required for several applications. The energy production of solar panels greatly depends on solar irradiance. Due to its intermittency, operators need a precise short-term forecast of the cloud coverage above solar panels to take preventive actions before a ramp-down. Similarly, air-to-ground or air-to-air communications through the atmosphere suffer from attenuation due to rain, clouds, atmospheric particles, and water vapor [6]. Accurate information about cloud formations along the signal path is key to better understanding these phenomena.

Several works in the literature have tried to address this task. Allmen and Kegelmeyer [55] compute cloud base height from a pair of whole sky imagers using correlation of images rectified by a pseudo-cartesian transformation and optical flow fields along successive images. Seiz et al. [56] match points extracted with Förstner and Harris operators from two different cameras with fish-eye lenses. They use least-squares matching and introduce a hierarchical pyramid-based approach based.

Kassianov et al. [57] compute the overlapping area between two images from a pair of ground-based imagers using a merit function. They derive the height of the clouds from the size of this overlapping area, which results in one value per image pair. Peng et al. [58] use three total sky imagers for 3D cloud detection and tracking. They assume clouds have only planar motion vectors and define an objective function to be maximized based on both motion tracking and height estimation.

The concept of 3D scene flow is also widely used for such purposes. It was first introduced by Vedula et al. [59]. Popular methods for computing scene flow include Huguet and Devernay [60], who couple optical flow and dense stereo matching via partial differential equations; Li and Sclaroff [61], who introduce probability distributions for optical flow and disparity; and Cech et al. [62], who reduce the computational complexity using seed growing.

Reconstructing the base height of a cloud mass using stereo cameras is a challenging task. Conventional stereo reconstruction algorithms do not work well in these cases, as clouds are generally non-rigid and do not exhibit strong features. In our study, we propose a framework based on 3D scene flow technique to estimate cloud base height. In Chapter 8 of this thesis, we also analyze the localization performance in a multi-camera setup with finite pixel-width dimension.

2.6 Summary

In this chapter, we have presented an overview of the existing works of cloud imaging for various remote sensing applications. The detection of clouds in the sky/cloud images is one of the fundamental tasks in this field. Furthermore, for the purpose of weather monitoring and other applications, it is also necessary to classify the clouds into various types. Other tasks such as correlating images with weather sensors and reconstructing cloud-base height are necessary to understand the different phenomenon of the earth's atmosphere. In this chapter, we have identified the shortcomings in the existing methods for these specific tasks. In the following chapters of this thesis, we will explore the contributions of our work in extensive details.

Chapter 3

Whole Sky Imager

3.1 Introduction

Whole sky imagers are becoming popular amongst the research community for a variety of applications and domains, such as aviation, weather prediction, and solar energy. The resulting images are of higher resolution than what can be obtained from satellites. Furthermore, the upwards pointing nature of the camera makes it easy to capture low-altitude clouds. Therefore, they provide a useful complement to satellite images.

Several models of automatic whole sky imagers have been developed at the Scripps Institute of Oceanography, University of California, San Diego, with cloud detection in mind [63]. Yankee Environmental Systems commercialize the *TSI-880*, which is used by many institutions [20, 27]. Those imagers provide a good starting point for cloud analysis, but their design and fabrication is controlled by a company, making specialized adaptations hard or impossible. Besides, those devices are expensive (around 30,000-35,000 US\$ for the *TSI-880*) and use very low image resolution (352×288) capturing devices. Those limitations encouraged us to build our own.

In this chapter, we propose three novel designs of a whole sky imager called **WAHRSIS** (Wide Angle High-Resolution Sky Imaging System). The first version of WAHRSIS consists of a sun-blocker that reduces the sun glare in the captured

images. It uses mostly off-the-shelf components which are available on the market for an overall price of around 2,500 US\$. A simple overall design allows a home-made mounting of the various parts. The resulting images are of high resolution and can be used for various applications. In addition to this sun-blocked based design of WAHRIS, we also propose two additional compact models of WAHRIS. These compact models use High-Dynamic-Range-Imaging (HDRI) techniques to reduce sun glare in captured images. The designs of these sky cameras are specifically catered to hot and humid weather conditions of Singapore. The hermetically sealed enclosure needs to operate at permissible temperature and humidity levels for proper functioning of the several electronic items, housed inside the enclosure.

3.2 Outline of Our Contribution

The main novel contributions of this chapter include:

- Design of low-cost, flexible and high resolution ground-based sky cameras;
- Framework for spatial-, color- and luminance- calibration of sky cameras;

The rest of this chapter is organized as follows. We describe the general design of our imaging infrastructure in Section 3.3. Section 3.4 describes the mechanical designs of its three models: sun-blocker based model, actively-cooled model and passively-cooled model. We describe the various calibration processes in Section 3.5. Finally, the uses of HDRI imaging techniques are introduced in Section 3.6. Section 3.7 concludes the chapter.

3.3 General Design of WAHRIS

Our WAHRIS system essentially consists of a hermetically sealed box, a workstation and an imaging system. The sealed box is necessary to protect its different internal components from the outdoor weather conditions. It consists of a temperature and humidity control module, that helps in maintaining the internal working conditions of

WAHRIS. It also comprises a micro-computer that helps in controlling the various components of the sky imager. Different models of WAHRIS have various sun-glare control mechanisms. The entire sky imaging system is connected to a remote server, that helps in archiving the captured images. A general system design of WAHRIS is shown in Fig. 3.1.

Figure 3.1: General design of WAHRIS. The camera, workstation and the sealed mechanical box is used in real-time image acquisition, whereas offline image processing and visualization are performed in the remote server.

The WAHRIS system captures images at regular intervals of time. This creates enormous amount of image data. Therefore, a centralized and an efficient storage system is required. We have thus setup a server for this purpose, as shown in Fig. 3.2. It currently stores the sky images, as well as the weather station, radiosonde, human observation, ceilometer and weather radar measurements. Three WAHRIS models and three weather stations are located at Nanyang Technological University, which are sending their measurements to this server in real time. The radiosonde data is retrieved twice a day from the website of the *Department of Atmospheric Science*

(University of Wyoming).¹

Figure 3.2: Server and client setup for data storage and visualization.

3.4 WAHSIS Design

In this section, we describe the mechanical designs of three WAHSISs. The sun-blocker based WAHSIS model consists of a sun-tracking blob that tracks the position of sun over the period of the entire day. This helps in reducing the glare of the sun in the captured images. However, most part of the captured images get occluded by the sun-blocker. The other models of WAHSIS do not use a sun-blocker in its design. Instead, it uses HDRI techniques to reduce sun-glare in the images. More details on the advantages of HDRI can be found in Section 3.6. The design of the various models of the sky cameras are kept as simple as possible. This is because we use off-the-shelf components to build the sky imagers, and it can be easily replicated by other enthusiasts, as a part of engineering design project ^{2 3}.

¹<http://weather.uwyo.edu/upperair/sounding.html>.

²The design of our sky camera is recently featured in ‘H. Godrich, Students’ Design Project Series: Sharing Experiences [SP Education], IEEE Signal Processing Magazine, Jan 2017’ as a part of student project series.

³A tutorial to build the WAHSIS Sky Imager with automated cloud coverage estimation is available online at <https://github.com/Soumyabrata/DIY-sky-imager>.

3.4.1 Sun-blocker based WAHRSIS Model

Components

Figure 3.3 shows a drawing of the sun-blocker based WAHRSIS. Its main components are the following:

- The *sun-blocker* occludes the light coming directly from the sun and thus reduces the glare in the circumsolar region of the captured image. It is composed of a main arm, which is rotated around the outside of the box by a motor. A polystyrene ball is fixed on top of a stem, which is attached to the main arm via a second motor rotating along the other axis. Details about estimating the position of sun at a particular location on earth are described in Appendix A.
- An *Arduino board* controls the two motors of the sun blocker. The algorithm is briefly explained in the Appendix B.
- *Outer casing*: A hermetically-sealed box prevents moisture and dirt from affecting the internal components. It contains a transparent dome at the top in which the fish-eye lens of the camera is positioned.
- The *camera and lens* is discussed in more detail in Section 3.4.1 below.

The device operates autonomously. However, some periodic manual maintenance interventions may be needed occasionally, for example to prevent the wear and tear of the motor gears. We have currently built and installed this on the roof-top of our university building. Table 3.1 details all the components and their respective prices at the time of purchase.

Imaging System

The imaging system of WAHRSIS consists of a *Canon EOS Rebel T3i* (a.k.a. *EOS 600D*) camera body and a *Sigma 4.5mm F2.8 EX DC HSM* Circular Fisheye Lens with a field of view of 180 degrees. It captures images with a resolution of 5184×3456 pixels. Due to the lens design and sensor size, the entire scene is captured within

Figure 3.3: Schematic drawing of WAHRSIS.

a circle with a diameter of approximately 2950 pixels (see Figure 3.6). A custom program running on the built-in laptop uses the *Canon Digital Camera SDK* to control the camera. Images can be automatically captured at regular time intervals, and the camera settings can be adjusted as needed.

The sensor of a typical digital camera absorbs near-infrared light quite effectively, so much that there is an infra-red blocking filter behind the lens. We had this filter physically removed and replaced by a piece of glass, which passes all wavelengths. The reasoning is that near-infrared light is less susceptible to attenuation due to haze [64], which is a common phenomenon in the atmosphere and especially around large cities. Haze consists of small particles in suspension in the air. Those have a scattering effect on the light whose incidence is modeled by Rayleigh’s law, meaning that the intensity of the scattered light $E_s \sim E_0/\Lambda^4$, where E_0 is the intensity of the emitted light, and Λ the wavelength of the light. We can thus reduce this effect by considering longer wavelengths, such as the ones in near-infrared. This model

Items	Cost (in US\$)
Arduino Mega	90
Arduino Mega 2560-control board	60
Bipolar 5.18:1 Planetary Gearbox Stepper	25
Bipolar 99.55:1 Planetary Gearbox Stepper	35
Stepper Motor Driver	20
Real Time Clock	15
Big Easy Stepper Motor Driver	30
12 V DC Power Supply	50
Base Hard plastic or Acrylic (Water-proof housing)	120
Plastic Dome	25
Metal Arm	50
Sun Blocker	5
Cables and accessories	40
Sigma 4.5 mm F2.8 EX DC HSM Circular Fisheye Lens	950
Canon EOS Rebel T3i (600D) camera body	360
Alteration for near-infrared sensitivity	300
Built-in laptop	350
Total Cost	2525

Table 3.1: Cost analysis for all the components of WAHRIS.

is however only valid for particles whose size is smaller than $\Lambda/10$. Most cloud consist of larger particles; their resulting scattering effect obeys Mie's law, which is wavelength-independent. For this reason, we expect our modified camera to provide us sharper images of cloud formations.

The sun-blocker-based WAHRIS model provides us high resolution images of the sky scene. However, a significant portion of the captured image is occluded by the sun-blocker. In the subsequent sections, we present two models of WAHRIS, that do not use sun-blocker in their designs. Instead, it uses HDRI techniques to reduce sun-glare in the captured images. More details about HDRI can be found later in Section 3.6 of this chapter.

3.4.2 Actively-cooled WAHRIS Model

The actively-cooled WAHRIS model consists of a hermetically-sealed plastic box with sunlight-reflecting paint on it, as shown in Fig. 3.4. A thermoelectric Peltier cooler is placed at the bottom of the casing. When electricity is provided, the side facing inside the box gets cold while the outside face gets hot. Fans are placed at both sides to dissipate the cold or heat produced. The on-board computer controls the cooler operation using temperature and humidity sensors.

Figure 3.4: Actively-cooled WAHRIS design. (a) DSLR camera, (b) dome, (c) ODRUID board, (d) casing, (e) thermoelectric cooler with fans.

3.4.3 Passively-cooled WAHRIS Model

The passively-cooled WAHRIS model consists of a ventilated casing made of galvanized steel, as shown in Fig. 3.5. This material reflects the sun light well and prevents rusting and corrosion. Ventilation grids are placed on each side, allowing ambient air to flow inside while preventing rainwater from entering. A metallic sheet is placed a few centimeters on top of the box, along with the transparent dome for the camera lens. The air in between acts as insulator, which protects the case from being heated by direct sunlight. We refer this model as passively-cooled model, as we use only ambient air and ventilation to maintain its internal temperature and humidity, without using any external cooling devices.

3.5 WAHRIS Calibration

The images captured by WAHRIS are of high resolution and capture the full sky hemisphere. However, the scene appears distorted by the fish-eye lens, and the colors are not rendered correctly due to the alteration for near-infrared sensitivity. Various calibration stages are thus required ⁴. We discuss geometric calibration,

⁴The source code of the various calibration processes for our sky camera WAHRIS is available online at <https://github.com/Soumyabrata/WAHRIS>.

Figure 3.5: Passively-cooled WAHRSIS design. (a) Dome, (b) shelter, (c) air vent cover, (d) casing.

white balancing and vignetting correction for our WAHRSIS models.

3.5.1 White Balancing

The camera of sun-blocker based WAHRSIS is sensitive to near-infrared light and thus sees beyond the visible spectrum. Color calibration in the traditional sense is less meaningful for our camera, because it is sensitive to both visible and near-infrared light. However, white balancing is still necessary, as we will explain here.

The sensitivity of the various channels depends on the Bayer filter in front of the sensor. It is known that the red pixels show more near-infrared leakage than the blue or green ones. The automatic white balance settings were engineered for a non modified camera. As a result, images captured under an automatic white balance mode appear reddish, as shown in Fig. 3.6a. Fortunately, our camera also provides a custom mode. It relies on a known white patch in the scene, for which the camera computes the incident light intensity quantization parameters to render this patch white in the captured picture. The resulting image after custom white balancing

looks visually plausible, as is shown in Fig. 3.6b.

Figure 3.6: Captured images using automatic and custom white balance settings.

Without white balancing, the red channel of our camera is prone to over-saturation and color clipping due to the additional near-infrared light reaching the sensor. The red channel of Fig. 3.6a contains 24.5% of clipped values (i.e. where $R = 255$), whereas the same channel in Fig. 3.6b contains only 0.7% of them. This shows the importance of the custom white balancing, which manages to compensate quite well for the near-infrared alteration.

The other models of WAHRIS do not use near-infrared sensitive lenses in its design. Therefore, we perform no color calibration on the actively-cooled- and passively-cooled- model of WAHRIS.

Chromatic Aberration

The refractive index of lenses is wavelength-dependent. This means that light rays from different parts of the spectrum converge at different points on the sensor. Chromatic aberration refers to the color distortions introduced to the image as a result of this phenomenon. Since the camera of sun-blocker based WAHRIS is sensitive to near-infrared, a larger part of the light spectrum is captured by the sensor, making the device more prone to this issue. Figure 3.7a shows chromatic aberration on an image captured by the camera.

The calibration process described in the previous section uses grayscale images.

(a) Example of chromatic aberration. Notice the green and magenta artifacts at the edges of the squares.

(b) Difference between the calibration function of each color channel and the grayscale calibration function.

Figure 3.7: Example of chromatic aberration and calibration function

In order to obtain more accurate calibration functions, we apply the same process on each of the red, green, and blue channels individually. As shown in Fig. 3.7b, we observe small differences in the resulting calibration functions, especially for the red channel. Visually, there is a striking difference in the color of the building for grayscale- and color-channel-specific- based calibration method. We show it in Fig. 3.8.

3.5.2 Geometric Calibration

A geometric calibration of the capturing device needs to be performed in order to obtain a precise model of the distortion introduced by the imaging system, which in our case includes the fish-eye lens and the dome [65]. This consists of determining the intrinsic parameters of the camera, which relates the pixel coordinates with their 3D correspondences in a camera coordinate system. On the other hand, extrinsic parameters express the translations and rotations required to use other 3D world coordinate systems, such as the ones span by the checkerboard borders. Intrinsic parameters give all the information needed to relate each pixel of a resulting image with the azimuth and elevation angles of the incident light ray and vice-versa, which is essential to be able to measure the physical extent of clouds.

Figure 3.8: Section of an image taken with WAHRIS that was rectified using grayscale (left) and color-channel-specific (right) calibration. The chromatic aberration in the right image is much less noticeable.

Fisheye Lens Calibration Background

The modeling of a typical (non-fisheye) camera uses the pinhole model. However, this is not applicable to our case because of the very wide capturing angle. A schematic representation of the refraction of an incident ray in a fisheye lens is shown in Fig. 3.9. Most common approaches relate the incident angle (β) with the radius on the image plane (r) and the focal length ($\bar{f}$), assuming that the process is independent of the azimuth angle.

Several theoretical models relating those two values exist:

- Stereographic projection: $r = 2\bar{f} \tan(\beta/2)$
- Equidistance projection: $r = \bar{f}\beta$
- Equisolid angle projection: $r = 2\bar{f} \sin(\beta/2)$
- Orthogonal projection: $r = \bar{f} \sin(\beta)$
- Perspective projection: $r = \bar{f} \tan(\beta)$

Fisheye lens manufacturers usually claim to obey one of those, but those equations cannot be exactly matched in practice. However, they can be approximated by a

Figure 3.9: Schematic representation of object space and image plane polynomial due to the Taylor series theory. Some of the calibration techniques make use of this, as it easily allows a slight deviation from the projection equation. Two polynomial functions can also be used to cope with possible radial and tangential distortions caused by the lens.

Most calibration techniques use a checker-board with a known pattern. Shah et al. [66] introduce a calibration method where such a polynomial is used. Furthermore, they model radial and tangential distortions. The optical center of the camera has to be found using a low power laser beam, which is cumbersome in practice. Bakstein et al. [67] use a spherical retina model, but do not provide any extrinsic parameter estimation technique and use cylindrical patterns. Sturm et al. [68] introduce a very generic model, where a camera is described by the coordinates of a set of rays and a mapping between those and the image pixels. However the authors experienced difficulties with a fish-eye lens, especially on the side view. Kannala et al. [69] use polynomials to describe the calibration and the radial and tangential distortions, but they also assume that the positions of the pattern on the checker-board are known. Finally, Scaramuzza et al. [70] provide an algorithm estimating both intrinsic camera parameters and the extrinsic parameters describing the positions of the checker-board. They provide a full *MATLAB* implementation of their method, making it

the easiest to use. We have chosen to use this method and enhanced it for our needs.

Calibration Method

The calibration toolbox proposed by Scaramuzza et al. [70] takes as input a number of images containing a known checker-board pattern. The first step is corner detection, which is performed by the toolbox. User interaction is required when the automatic detection algorithm fails. Those points are then used to find the parameters by a least-squares linear minimization, followed by a non-linear refinement with a maximum likelihood criterion. The toolbox uses a polynomial to model the ray refraction as well as a 2×2 matrix transformation and a translation to cope with small axis misalignment and the digitizing process. The intrinsic parameters thus consist of both the coefficients of the polynomial as well as the values of the matrix and the translation parameters, which describe the position of the center of the image.

Figure 3.10 shows a sample of the images we used for the calibration.

Figure 3.10: Sample of the checker-board images used for the calibration

The toolbox also outputs the re-projection errors, i.e. the difference between the original point location computed during the corner detection and the one resulting from the estimated parameters. This is done on the same images as the ones used for the parameter estimation. However we noticed that the returned errors do not necessarily relate the quality of the calibration, as the re-projection sometimes looks more accurate than the corner detection or the user input in estimating the coordinates of a grid corner. Furthermore, we would like to estimate a general error which does not only rely on the location of the checker-boards in the images used for the fitting. We also observed that the choice of the input images has a significant influence on the output of the algorithm. We thus modified the calibration process in the

following way: we randomly split our captured checker-board images into a training set of 10 images and a validation set of 6 images. We compute the parameters using the training set and the re-projection error using the validation set. We repeat this process 50 times with different randomized splits and finally consider the parameters leading to the smallest error on the validation set as the most accurate one. In this way we can reduce the uncertainty of the whole process.

Figure 3.11: Result of the calibration process compared to the various theoretical projection models.

Our fish-eye lens (*Sigma 4.5mm F2.8 EX DC HSM*) was designed by the manufacturer to follow the equisolid projection for visible light, but may be less accurate for the near-infrared spectrum. There is also a transparent dome on top of the camera lens, which may have an additional refracting effect on the incident rays. The toolbox models the calibration function as a polynomial and can thus incorporate these effects when fitting the coefficients. We have set the maximum degree of the polynomial to 4, as advised by the authors of the toolbox. We indeed noticed a bigger re-projection error with smaller degrees. However when the maximum degree is set to a higher number, the values of the coefficients of the degrees higher than 4 are almost zero.

Figure 3.12: Re-projection errors as a function of the distance to the image center (crosses: raw data; line: moving average).

Figure 3.11a shows the result of the resulting calibration compared to the various theoretical projection models. Figure 3.11b shows the distance in pixels between our estimated calibration parameters and the various projection models. We see that we reach a distance of more than 16 pixels for angles around 40 degrees. We also find that the maximum viewing angle of the whole device is 178.7 degrees, slightly less than hemispheric.

Figure 3.12 shows the re-projection error for all the points of the grids in the validation set. A moving average with a window of 50 pixels is also shown. The average re-projection error for our model is 3.87 pixels. We can see a minor increase of the error with distance from the image center (and thus the elevation angle of the incident ray).

3.5.3 Vignetting Correction

Vignetting refers to a darkening toward the corners of the image which is introduced during the capturing process. It has several origins. Natural vignetting is due to the incident light rays reaching the camera sensors with varying angles. It is commonly modeled by the cosine fourth law of the angle at which the light ray strikes the image sensor. The camera of WAHRISIS is particularly prone to this effect due to its wide angle. Optical vignetting describes the shading caused by the lens cylinder itself, blocking part of the off-axis incident light. This is aperture-dependant ⁵.

In most cases, it is difficult to model such vignetting function using the cosine fourth law. It is however approximated by a higher-order polynomial function with respect to the radial distance of the fish eye image. We denote this radial distance as r , as introduced previously in Section 3.5.2. In a perfectly calibrated imaging system, the image center coincides exactly with vignetting center. However, in practice, there is a mismatch between the two.

Suppose the vignetting luminance of a pixel in a fish-eye lens image is denoted by L_w . In [71], d'Angelo parameterized this luminance L_w with respect to radial distance r using a sixth-order polynomial:

$$L_w = \epsilon_1 r^6 + \epsilon_2 r^4 + \epsilon_3 r^2 + 1$$

,

where $\epsilon_1, \epsilon_2, \dots$ are the polynomial fitting coefficients.

In order to analyze such vignetting phenomenon in our WAHRISIS system, we use an integrating sphere for our experiments. It consists of a large sphere, whose interior surface is made of a diffuse white reflective coating, resulting in a uniform light source over the whole surface. A Light-Emitting Diode (LED) light source as well as the camera are placed inside. Figure 3.13 shows an image captured inside this sphere. Notice the higher brightness at the center. This image has been taken with the biggest available aperture ($\bar{f}/2.8$) and constitutes the worst case with regards to vignetting. Smaller aperture values will result in less significant distortions.

⁵<http://toothwalker.org/optics/vignetting.html>

Figure 3.13: Image captured inside the integrating sphere.

We plot the luminance value of the pixels in the fish-eye lens (L_w) with respect to the radius (r) of the image center ⁶. Figure 3.14 shows the luminance of the pixels inside the lens circle as a function of the distance from the image center.

We observe that the luminance of an uniformly-lit scene is perceived differently at different radial distances of the image. The luminance is highest near the center, and it radially drops off with increasing distance. We fit a second-degree polynomial function between the two. We obtain the fitting equation as:

$$L_w = \epsilon_1 r^2 + \epsilon_2 r + \epsilon_3$$

,

where $\epsilon_1 = -1.365 \times 10^{-5}$, $\epsilon_2 = -0.004006$ and $\epsilon_3 = 235.8$ are the corresponding polynomial fitting constants.

In order to correct the luminance values, we normalize the obtained luminance values and then take their inverse. We fit a moving average and use this result as a radius dependent correction coefficient, as shown in Fig. 3.15. We repeat this process for each aperture setting.

In this section, we discussed the various calibration processes that we perform in our WAHRIS system. In the subsequent section, we will see how HDRI techniques

⁶For the sake of simplicity, we assume that the vignetting center and image center coincides in our system.

Figure 3.14: Luminance as a function of the distance from the image center. The red line is the fitting equation $L_w = -1.365 \times 10^{-5} r^2 - 0.004006 r + 235.8$.

can help in reducing sun-glare in (two of) our sky-camera models, and significantly simplify post-processing techniques.

3.6 High Dynamic Range Imaging

The conventional cameras cannot capture and render a wide range of the luminosity of a scene. On a typical sunny and clear day, it is difficult to capture the entire luminance range of the sky scene using a conventional camera. The region around sun has a luminous intensity several orders of magnitude higher than other parts of the scene. The ratio between the largest and the smallest luminance value of a scene is referred to as its Dynamic Range (DR). Typical cameras can only capture a Low Dynamic Range (LDR) image. Therefore, we use the exposure bracketing option of our cameras to capture three pictures at 0, -2 and -4 exposure value (EV) in quick succession. The aperture remains fixed across all captured images, while the shutter speed automatically adapts to match the appropriate exposure. Figure 3.16 shows

Figure 3.15: Correction coefficients in red and scatter plot of all the values in blue.
 an example of the captured images at varying exposure levels.

Figure 3.16: Example of three images captured by the camera at varying exposure levels and shutter speeds. (a) 0 EV, 1/400 sec. (b) -2 EV, 1/1600 sec. (c) -4 EV, 1/4000 sec.

These LDR images are then fused together to generate a High-Dynamic-Range (HDR) image. A detailed discussion on the generation of HDR image is found in the subsequent section.

3.6.1 HDR Image Generation

The different LDR images can be fused together into a single, high-dynamic-range radiance map. As their exposure times are known, we can compute the camera response curve of the imaging system using samples obtained from the different LDR images⁷. We use Debevec and Malik’s algorithm [72] to recover HDR radiance map from LDR images. This is shown in Fig. 3.17. The dynamic range of the resulting HDR radiance map is 15 bits, which represents a significant improvement over the 8-bit LDR images.

Figure 3.17: Response curve of the HDRI system, computed from three LDR images with different exposure settings.

This generated HDR radiance map cannot be viewed directly on a conventional LDR display. Therefore, we tone-map the HDR radiance map into a conventional 8-bit LDR image using contrast-limited adaptive histogram equalization (CLAHE) [73]. We use CLAHE for this application because it involves a logarithmic transformation of the higher DR radiance map to a lower 8-bit display. This produces a perceptually

⁷The source code to generate the camera response curve and other HDR imaging techniques are available online at <https://github.com/Soumyabrata/HDRCaptures>.

recognizable image, because response of human eye to light is also logarithmic. This operation compresses the dynamic range by preserving the details in the image. Figure 3.18 shows an example of a tone-mapped image.

Figure 3.18: Tone-mapped image computed with the pictures shown in Fig. 3.16

3.6.2 Introducing HDRI techniques in sky cameras

The actively-cooled and the passively-cooled WAHRIS models do not have a physical sun blocking mechanism. Instead, we use HDRI techniques to reduce the over-exposure resulting from sun glare. This avoids occlusions from the sun-blocker and its support structure in the captured images (which significantly complicate subsequent computer vision tasks), while at the same time keeping the number of saturated pixels in the circumsolar region to a minimum.

The imaging system of these WAHRIS models consists of the same imaging system as compared to our sun-blocker based WAHRIS model. These imagers use an ODROID-C1 single board computer; with its 1.5 GHz processor and 1 GB RAM, this board is powerful enough to perform on-board image processing tasks.

We perform the HDR fusion and tone-mapping on a server that collects the images from all our WSIs. Since we have three imagers capturing one frame every 2 minutes, we need algorithms which are computationally efficient. In that context, we use the GPU implementation of Akyüz [74]. It relies on the OpenGL API to perform the entire HDR pipeline on the GPU. Computing a 18 megapixels HDR image and

its tone-mapped version takes less than 7 seconds, compared to a few minutes with standard CPU algorithms.

For storage efficiency, we use the *JPEG-XT* format [75] to compress our HDR images⁸. *JPEG-XT* is an extension of *JPEG* currently being developed for HDR images. The tone-mapped version is stored as a standard *JPEG* image, while the inverse tone-mapping operation is added to the file as metadata. This allows for the HDR radiance map to be decoded. Using *JPEG-XT* at a quality level of 90%, we obtain a file size of about 8 MB. The *JPEG-XT* format is used to store the HDR images. An important advantage of *JPEG-XT* format is that it is backward compatible to the popular *JPEG* compression technique. It consists of a base layer that is compatible with the legacy systems and an extension layer that provides the full dynamic range. This represents a significant improvement compared to the common *RGBE* format, which uses 50MB per image by storing every pixel with one byte per color channel and one byte as a shared exponent.

We use such HDR imaging techniques instead of using RAW images. There are compelling trade-offs in choosing either HDR or RAW images. HDR images uses a set of Low Dynamic Range (LDR) images to compute the HDR radiance map; and we can access the 0EV LDR image which is needed for subsequent analysis. The size of an LDR image is approximately 2-3MB per image. On the other hand, the entire dynamic range of the sky scene can be captured using a single RAW image. However, the size of a conventional RAW image is approximately 18MB per image. Therefore, because of operational issues, we have chosen to use exposure bracketed LDR images, instead of capturing in RAW mode. In the future designs, we can consider using only RAW mode while capturing the sky/cloud images. In the upcoming Chapter 5 of this thesis, we discuss how HDRI techniques can improve cloud segmentation algorithms, especially around circumsolar region.

As a final comparison, we compare the images obtained from the different models of our sky camera. Figure 3.19(c) shows the resulting image, along with those obtained from a sky imager with a mechanical sun blocker and a normal single-exposure

⁸We use the reference software source code available at <https://jpeg.org/jpegxt/software.html>

image. In order to visualize the efficacy of our proposed approach, we zoom-in on the circumsolar region in each of the three images to look at the amount of detail visible around the sun. Although the sun-blocker is effective at reducing sun glare, the occlusion is significant. Without a sun-blocker, the number of saturated pixels is very high in a normal single-exposure image. Fusing multiple exposures into an HDR image is indeed a good solution for minimizing the effects of over-exposure.

Figure 3.19: Effect of sun glare with different strategies. The white rectangle outlines the circumsolar region, which is enlarged in the bottom row. Saturated pixels are shown in pink.

Unlike our sky imagers, the commercial sky imagers have a low image resolution of 352×288 . Moreover, the total cost is high too (US\$30-35K). Also, no HDRI techniques are present in the commercial imager. We employ HDRI techniques in

our imager, that helps us in building more compact versions of sky cameras. A more detailed discussion on HDRI techniques, and how they solve cloud segmentation problems in circumsolar region will be discussed later in this thesis in Chapter 5. Most of our works are performed in LDR images, unless otherwise stated.

3.7 Conclusions

In this chapter, we have presented WAHRIS - a new custom-built whole sky imager. Its advantages are low cost and simple design. We presented three models of WAHRIS: sun-blocker based model, actively-cooled model and passively-cooled model. We described its components and the calibration of its imaging system. Our imager provides high resolution images of the whole sky hemisphere, which can be useful for many various applications, as indicated in the subsequent chapters.

Unlike our sun-blocker based model, the remaining two models does not use a mechanical sun blocker. Instead, they rely on the fusion of multiple exposures to create a high-dynamic-range image and minimize over-saturation of pixels near the circumsolar region. Table 3.2 summarizes the various characteristics of the different models of WAHRIS.

WAHRIS Properties	Sun-blocker model	Actively-cooled model	Passively-cooled model
Glare reduction method	Sun blocker	HDRI	HDRI
Occlusion severity	Moderate	Nil	Nil
Weatherability	Limited	Good	Good
Portability	Limited	Good	Good
Image resolution	High	High	High
Total cost (in US\$)	\$2525	\$1895	\$1769

Table 3.2: Comparison of the different WAHRIS models.

We use the captured images for sky/cloud segmentation and classification into different cloud types. Moreover, we deploy multiple sky cameras in order to estimate cloud base height.

Chapter 4

Color Space Analysis for Cloud Detection

4.1 Introduction

The most fundamental problem in cloud imaging is the successful detection of clouds in the captured sky/cloud images. The detection of clouds from sky images is challenging as clouds do not possess any definite structure, contour, shape, or size. As a result, color has been used as the predominant feature for sky/cloud segmentation. Numerous techniques based on different color models and spectral wavelengths have been proposed in the literature to solve this problem. However, the selection of color models and channels in the existing algorithms seem ad-hoc in manner, without systematic analysis or comparison. This chapter aims to address this issue by providing a structured review and evaluation of color models for the segmentation of sky/cloud images. For this purpose, we make use of several tools and techniques. Our comprehensive analysis is able to clearly identify the most suitable color models and components for the analysis of cloud/sky images. The analysis in this chapter is limited to Low-Dynamic-Range (LDR) images only.

4.2 Outline of Our Contribution

The main novel contributions of this chapter include:

- Structured and systematic analysis of various color channels in sky/cloud images using various statistical tools such as color distribution characteristics, principal component analysis (PCA), rough-set based feature selection and clustering methods;
- Proposal of a rough-set based method that can accurately assess the efficiency of different color channels for cloud segmentation. To the best of our knowledge, our proposed approach is the first that uses rough set theory for color channel selection in visible-light images.

The rest of this chapter is organized as follows. Section 4.3 describes the different color spaces and components that are used for detection of clouds in ground-based sky/cloud images. We introduce the different statistical tools in Section 4.4 that are exploited to identify the most discriminatory color channels. In Section 4.5, we propose a rough-set based color channel selection method. Experimental results are discussed and analyzed in Section 4.6. Finally, Section 4.8 concludes this chapter.

4.3 Color Models for Sky/Cloud Images

In the literature, several techniques based on different color models have been proposed for sky/cloud segmentation. Long et al. [20] use the ratio of red and blue channels (R/B) to detect clouds using appropriate threshold values. Calbó and Sabburg [30] use the same (R/B) ratio to derive statistical features (mean, standard deviation, entropy etc.) of the clouds and subsequently classify the sky/cloud images into different cloud types. Heinle et al. [31] utilize a k-nearest-neighbor classifier using the difference of red and blue channels ($R - B$) to classify cloud types. Souza-Echer et al. [27] choose saturation for the estimation of cloud coverage. Mantelli-Neto et al. [18] classify clouds by exploiting the locus of pixels in the RGB color model.

Most recently, Li et al. use a normalized difference of blue and red channels ($\frac{B-R}{B+R}$ [33]) for cloud detection.

The choices of color models employed in existing literature are based on empirical observations about the color distributions of cloud and sky pixels. However, there exists no systematic analysis to determine the most suitable color channel(s) for this purpose. The purpose of this chapter is to present a unified approach in the analysis of sky/cloud images using different color channels and to identify the color channels with the best results ¹.

We consider the following color spaces and components for analysis (see Table 4.1): RGB , HSV , YIQ , CIE $L^*a^*b^*$, three forms of red-blue combinations (R/B , $R - B$, $\frac{B-R}{B+R}$), and chroma $C = \max(R, G, B) - \min(R, G, B)$. In addition to the color channels c_{1-9} and c_{13-16} used in the existing literature, we also include c_{10-12} and c_{16} , as separating chromatic and achromatic information may prove beneficial for sky/cloud image segmentation. All the color channels are normalized.

c_1	R	c_4	H	c_7	Y	c_{10}	L^*	c_{13}	R/B	c_{16}	C
c_2	G	c_5	S	c_8	I	c_{11}	a^*	c_{14}	$R - B$		
c_3	B	c_6	V	c_9	Q	c_{12}	b^*	c_{15}	$\frac{B-R}{B+R}$		

Table 4.1: Color spaces and components used for analysis.

Figure 4.1 illustrates the 16 color channels of a sample sky/cloud image.

4.4 Approach and Tools

4.4.1 Color Distribution

Since we are essentially trying to distinguish between two classes of pixels (sky and clouds), a color model with a bimodal distribution can facilitate this task. Color components with a high degree of bimodality are not only good candidates for determining the number of clusters in a sample image, but can also be used to decide

¹The source code of our color channel analysis on sky/cloud images is available online at <https://github.com/Soumyabrata/color-channels>.

Figure 4.1: Sample image and segmentation mask from the HYTA database [33], together with the 16 color channels from Table 4.1 (the color maps for c_4 , c_5 , c_9 , c_{15} , and c_{16} are inverted such that cloud pixels always have a lighter shade than sky pixels).

whether or not the sample image requires segmentation in the first place. Pearson’s Bimodality Index (PBI) is a popular statistic to evaluate the bimodal behavior quantitatively [76]. It is defined as:

$$\text{PBI} = b_2 - b_1, \quad (4.1)$$

where b_2 is the kurtosis and b_1 is the square of skewness. A PBI value close to 1 indicates a highly bimodal distribution.

4.4.2 Principal Component Analysis

We use the Principal Component Analysis (PCA) to (a) check the correlation and similarity between different color components, and (b) determine those color components that capture the greatest variance.

The PCA is computed as follows. Let us assume a sample image $\mathbf{X}_i$ of dimension $m \times n$ pixels from a set of N images ($i = 1, \dots, N$). Each color channel of the sample image is reshaped into a straight vector $\mathbf{c}_j$ of dimensions $mn \times 1$. These column vectors are stacked alongside each other to form a matrix $\hat{\mathbf{X}}_i$ of dimensions $mn \times 16$:

$$\hat{\mathbf{X}}_i = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_j, \dots, \mathbf{c}_{16}]; j = 1, \dots, 16 \quad (4.2)$$

The ranges of these 16 color channels are different and need to be normalized so that no color channel is under- or over-represented in the PCA analysis. Each of the

16 color channels is normalized using its mean and standard deviation across all the images in the dataset, thereby generating the new image representation $\ddot{\mathbf{X}}_i$ with zero mean and unit variance:

$$\ddot{\mathbf{X}}_i = \left[\frac{\mathbf{c}_1 - \bar{c}_1}{\sigma_{c_1}}, \frac{\mathbf{c}_2 - \bar{c}_2}{\sigma_{c_2}}, \dots, \frac{\mathbf{c}_j - \bar{c}_j}{\sigma_{c_j}}, \dots, \frac{\mathbf{c}_{16} - \bar{c}_{16}}{\sigma_{c_{16}}} \right], \quad (4.3)$$

where $\bar{c}_j$ and σ_{c_j} are the mean value and standard deviation of the j -th color channel.

Subsequently the covariance matrix $\mathbf{M}_i$ is computed for each of the $\ddot{\mathbf{X}}_i$. Let the eigenvector $\mathbf{e}_{ij}$ and eigenvalue λ_{ij} ($j = 1, \dots, 16$) be obtained from the eigenvalue decomposition of the matrix $\mathbf{M}_i$. For the i^{th} image, the eigenvectors $\mathbf{e}_{ij}$ corresponding to the largest eigenvalues λ_{ij} encompass an orthonormal vector subspace. The eigenvectors and eigenvalues are calculated to project the multi-dimensional vector space into lower dimensions and to reveal the underlying relationship amongst them.

4.4.3 Discrimination Threshold

In addition to principal component analysis, we also use the Receiver Operating Characteristics (ROC) curve [77, 78] to identify the important color components in sky/cloud image segmentation. The ROC curve represents the fraction of true positives and false positives samples in the sky/cloud binary classification problem for varying values of discrimination threshold. The area between this ROC curve and the random classifier slope is widely used as a measure for the discriminatory performance in a classification framework.

We consider the set of 16 color components and compute the area under the ROC curve for each, which is subsequently used to identify the best color components for sky/cloud segmentation.

4.4.4 Clustering

For clustering the sky/cloud images, we employ the fuzzy c-means algorithm [79, 80] to assign probabilities of cloud detection to the set of pixels of the input image. The algorithm for the effective segmentation of clouds from the sky/cloud images employs the minimization of the following objective function:

$$J = \sum_{n_c=1}^2 \sum_{l=1}^{mn} p(\mathbf{x}_f)^\tau d(\mathbf{x}_f, \mathbf{v}_r), \quad (4.4)$$

where τ is called the fuzziness index which controls the degree of fuzziness during the clustering process, and n_c is the number of clusters. In our experiments, we set $\tau = 2$ and $n_c = 2$ clusters (binary segmentation). The term $d(\mathbf{x}_f, \mathbf{v}_{n_c})$ denotes the 2D Euclidean norm between the input feature vector $\mathbf{x}_f$ and the cluster centers $\mathbf{v}_{n_c}$. Both $\mathbf{v}_1$ and $\mathbf{v}_2$ are vectors of dimension o , where o can take any positive integer number.

4.4.5 Kullback-Leibler Divergence

Information-theoretic measures like Kullback-Leibler (KL) distance or divergence are also widely used for band selection purposes, especially in hyperspectral imaging [81]. This distance can be interpreted as the amount of dissimilarity of the color channel from the ground truth. In other words, a higher distance indicates an unfavorable color channel for cloud segmentation. Therefore, we also compute the KL-divergence of the color channels from the binary ground truth images.

In this section, we introduced the various statistical tools for a systematic analysis of color channels. However, these techniques had certain shortcomings with regards to ranking color channels according to their relevance for cloud detection. In the subsequent section, we will propose a *rough-set* based approach that provides better results.

4.5 Rough Set Based Color Channel Selection

Classical rough sets [82] are an approximation of conventional sets in set theory. In a scenario where it is difficult to define the boundaries of a conventional set, rough set theory provides a set of mathematical tools to define them in an approximate way. It facilitates an objective analysis in a data-driven system which is vague, uncertain and incomplete. Here we first explain the related terminologies of rough sets, and subsequently define the use of rough sets for visible-light images.

4.5.1 Rough Set Theory

In rough sets, information is expressed in the form of a *decision table*. We define a decision table $\mathcal{L}$ such that each row represents an observation, and each column is an attribute from attribute set $\mathcal{A}$. This non-empty set of observations is usually referred to as the *universe* $\mathcal{U}$. Formally, for each entry in the decision table, we define the function f that maps attribute $\mathcal{A}$ to value domain $\mathcal{V}$, $f : \mathcal{U} \times \mathcal{A} \rightarrow \mathcal{V}$.

Any reduct $\mathcal{P}$ from the set of attributes $\mathcal{A}$ satisfies the indiscernibility (or equivalence) relation $\text{IND}(\mathcal{P})$. For any reduct $\mathcal{P} \in \mathcal{A}$, the $\mathcal{P}$ -indiscernibility relation is defined as:

$$\text{IND}(\mathcal{P}) = \{(x_s, x_t) \in \mathcal{U}^2 \mid \forall a \in \mathcal{P}, f(x_s, a) = f(x_t, a)\}, \quad (4.5)$$

where x_s and x_t are two observations from the universe $\mathcal{U}$, and a is an element from set $\mathcal{P}$. This indicates that x_s and x_t are indiscernable based on the attribute $\mathcal{P}$, as the value function f assigns both x_s and x_t to the same value. This partition of $\mathcal{U}$ generated by $\text{IND}(\mathcal{P})$ is denoted as:

$$\mathcal{U}/\text{IND}(\mathcal{P}) = \{[x_s]_{\mathcal{P}} \mid x_s \in \mathcal{U}\}. \quad (4.6)$$

Let $\mathcal{X}$ be a set of observations from the universe $\mathcal{U}$. Rough set theory asks the

question: how can we express this conventional set $\mathcal{X}$, using only the information in attribute set $\mathcal{P}$? In general, there is no precise answer, and therefore approximations are generated. They are defined by their corresponding $\mathcal{P}$ -lower- and $\mathcal{P}$ -upper-approximations as:

$$\underline{\mathcal{P}}(\mathcal{X}) = \cup\{[x_s] \mid [x_s] \subseteq \mathcal{X}\}, \quad (4.7a)$$

$$\overline{\mathcal{P}}(\mathcal{X}) = \cup\{[x_s] \mid [x_s] \cap \mathcal{X} \neq \emptyset\}. \quad (4.7b)$$

The observations in the lower approximation set $\underline{\mathcal{P}}(\mathcal{X})$ are the definite members of $\mathcal{X}$, also called the *positive* region $\text{POS}(\mathcal{X})$. On the other hand, $\overline{\mathcal{P}}(\mathcal{X})$ represents the upper approximation of the set. It denotes the possible members of $\mathcal{X}$, based on the knowledge in the decision table. We illustrate this in Fig.4.2.

Figure 4.2: Illustration of a typical rough set, which approximates a conventional set (depicted in blue). Each individual grid depicts a partition of the universe generated by an equivalence relation. The union of all such partitions indicated with solid green borders (definite members) represents the *lower approximation*; while the dotted red borders (possible members) represent the *upper approximation*.

Let us assume that the attribute set $\mathcal{A}$ consists of both condition and decision attributes $\mathcal{C}$ and $\mathcal{D}$ respectively, such that $\mathcal{A} = \mathcal{C} \cup \mathcal{D}$. The *relevance* criterion is defined as the dependence between $\mathcal{C}$ and $\mathcal{D}$, and can be expressed as:

$$\gamma_{\mathcal{C}}(\mathcal{D}) = \frac{|\text{POS}_{\mathcal{C}}(\mathcal{D})|}{|\mathcal{U}|}, \quad (4.8)$$

where $|\cdot|$ denotes the cardinality of the set. This dependence value is an important measure to identify the most discriminate attribute from set $\mathcal{A}$. The value of $\gamma_{\mathcal{C}}(\mathcal{D})$ ranges between 0 and 1, where 0 indicates independence and 1 indicates $\mathcal{D}$ fully depends on $\mathcal{C}$.

4.5.2 Proposed Rough-Set Based Selection Method

We now describe the problem of color channel selection using rough set terminology and propose our algorithm for this purpose². The main benefit of using rough set theory is that it provides a systematic method to approximate the segmentation ground truth, with the highest degree of approximation. Moreover, no prior information about the data is needed for the analysis.

We consider 16 color channels in our analysis, as shown in Table 4.1 and illustrated in Fig. 4.1. We utilize rough set theory to identify the most *informative* color channel(s) (a.k.a. reducts) from these 16.

Suppose that $\mathcal{U}_i$ is a non-empty finite universe of pixel observations for a single sky/cloud image $\mathbf{X}_i$ from the image dataset $\mathcal{T} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$. We assume that $\mathbf{Y}_i$ is the binary ground-truth image of the sample image $\mathbf{X}_i$. We define $\mathcal{A}_i$ as the set of its corresponding vectorized color channels $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{16}\}$ for a particular image $\mathbf{X}_i$, along with the vectorized decision attribute $\mathbf{y}_i$. These 16 color channels are called condition attribute sets, and one ground truth vectorized image is the decision attribute set from the family of attributes $\mathcal{A}_i$.

²The source code of our proposed rough-set based algorithm is available online at <https://github.com/Soumyabrata/rough-sets>.

We define a decision table $\mathcal{L}_i$ such that each row represents a pixel, and each column represents an attribute. Thus, for a single image $\mathbf{X}_i$, the function f assigns a value v_i^{kj} in the value domain to each variable-attribute pair (q_k, a_j) , where q_k is the k -th pixel value of the image and a_z is the z -th attribute from the set $\mathcal{A}_i$. For the sake of brevity, we will drop the index i in the subsequent discussions of this chapter.

Our objective is to characterize the ground-truth image $\mathbf{y}$ from the knowledge of the reduct $\mathcal{P}$. We define the reduct $\mathcal{P}$ as the subset of condition attribute sets $\mathcal{C}$.

We are interested in identifying the most discriminative color channel that is strongly dependent on the sky/cloud decision attribute $\mathbf{y}$. This dependence on a particular color channel is measured by the corresponding *relevance* criterion of the color channels. Color channels with a high relevance value are better candidates for sky/cloud segmentation.

In analogy to Eq. (4.8), we define the relevance criterion γ_j for each color channel $\mathbf{c}_j$, which indicates the dependence between ground truth $\mathbf{y}$ and color channels:

$$\gamma_j = \frac{|\text{POS}(\mathbf{y})|}{|\mathcal{U}|}. \quad (4.9)$$

For image $\mathbf{X}_i \in \mathbb{R}^{m \times n}$, we generate the decision table $\mathcal{L}_i \in \mathbb{R}^{mn \times 17}$. Each row of the decision table $\mathcal{L}_i$ corresponds to a pixel. The first 16 columns represent the color channels, and the 17th column corresponds to the ground truth label. We generate 16 distinct partitions from this decision table using the knowledge of the ground truth labels. Subsequently, the corresponding lower approximations of 16 color channels are generated using Eq. (4.7). In fact, this lower-approximation set is the union of all partitions (generated by the equivalence relation) that are possible members of the *cloud* label. Next, we compute the relevance value γ_j of all color channels for the image $\mathbf{X}_i$ using Eq. (4.9).

We perform this for all images of dataset $\mathcal{T}$. Finally, we compute the average relevance $\bar{\gamma}_j$ of each color channel across all the images. The color channel with the maximum average relevance value $\bar{\gamma}_j$ is the best amongst all color channels under consideration.

4.6 Experiments

4.6.1 Dataset

In order to analyze the various statistical approaches and check the efficacy of our proposed color channel selection algorithm, we conduct experiments on a publicly available sky/cloud image database called HYTA [33].

It consists of 32 distinct images with different sky/cloud conditions along with their binary ground-truth segmentation masks. These images are collected from two main sources, namely ground-based sky cameras located at Beijing and Conghua, China. These sky cameras were developed by the Institute of Atmospheric Sounding at the Chinese Academy of Meteorological Sciences. The obtained images are stored in *JPEG* format. No radiometric calibration data are available for this dataset. Additionally, several images found on the Internet from other cameras and locations are included. The ground truth segmentation masks of each of the images are generated using a Voronoi polygonal region generator. The images in HYTA³ are of various dimensions; the average is 682×512 pixels. A few sample images of HYTA along with their ground truth are shown in Fig. 4.3.

Figure 4.3: Sample images from the HYTA database (top row) along with corresponding sky/cloud segmentation ground truth (bottom row).

³In this thesis, the dataset name and its associated algorithm are referred as HYTA interchangeably.

4.6.2 Distribution Bimodality

The segmentation of input image into two classes (sky and clouds) becomes easier for those color channels which exhibit higher bimodality. The bimodal behavior of a color channel for the concatenated distribution is measured using PBI as described in Section 4.4.1. The results are summarized in Table 4.2. Out of the 16 color channels, c_5 (S), c_1 (R), and c_{13} (R/B) have the lowest PBI and thus exhibit the most bimodal distributions, indicating that the segmentation of images into two distinct classes should work well in these color channels.

c_1	2.24	c_4	3.11	c_7	2.71	c_{10}	2.86	c_{13}	2.27	c_{16}	4.27
c_2	2.83	c_5	1.94	c_8	3.98	c_{11}	8.85	c_{14}	4.43		
c_3	3.25	c_6	3.26	c_9	5.96	c_{12}	4.59	c_{15}	2.92		

Table 4.2: PBI values for all color channels. The most bimodal channels are highlighted in bold.

4.6.3 Principal Component Analysis

The distribution of the amount of variance captured by the principal components is shown in Fig. 4.4. The combination of first and second principal components capture a large majority (95.4%) of variance in the input data across the individual images, and over 90% for the entire database.

Table 4.3 shows the variance captured by each principal component for the concatenated distributions. We observe that the first two principal components capture a large part of the variance across all the images, namely 85.6% of the total variance.

Database	1 st	2 nd	3 rd	4 th	5 th	6 th
HYTA	60.8%	24.8%	7.86%	5.11%	0.84%	0.36%

Table 4.3: Percentage of variance captured across different principal components for HYTA database.

In order to find the most significant vector for the detection of clouds, the contributing effect of the individual variables on the primary principal component axis

Figure 4.4: Distribution of the variance across the principal components for all images of the HYTA database. The separate bar on the right shows the variance distribution for the concatenation of all images of the entire database.

is analyzed. Every color channel has different loading factors on the most important principal component axis, and consequently each of them has a different contribution to the common variance captured in the corresponding data matrix $\ddot{\mathbf{X}}_i$. These individual loading factors are essentially the projections of the input vectors (color channels) on the principal components. The significant color channels in terms of relative contribution to the first principal component axis are c_{13} (R/B), c_{15} ($\frac{B-R}{B+R}$), and c_5 (S).

Extending our analysis to finding the most significant pair of color channels, we need to understand a pair's cumulative contribution to the common variance. The sum of the squares of the corresponding loading factors for different color channels in the concatenated distribution represents the cumulative common variance captured by the pair. The pairs that have the highest cumulative loading on the first principal component are c_{13} - c_{15} , c_5 - c_{13} , and c_5 - c_{15} .

Another aspect to consider is the orthogonality between the individual color channels. For this purpose, we compute the triangular area enclosed by two given input

vectors (color channels) projected on the plane spanned by the first and second principal components. There are $\binom{16}{2} = 120$ combinations of two color channels. The triangular area for all unique cases is shown in Fig. 4.5, where the brightness of each square represents the area captured by the corresponding pair. As expected, certain color channel pairs (the dark patches) are highly correlated, for example the green channel (c_2) with most luma channels (c_7, c_{10}), the various R - B combinations (c_{13-15}) with one another, and so on.

Figure 4.5: Distribution of area AOB for all input vector pairs, where OA and OB are two color channels. The brightness of each square represents the area captured by the corresponding pair.

4.6.4 Rough-set Based Relevance Scores

In our proposed rough set based method, we measure the dependency of each color channel and report its corresponding average relevance value $\bar{\gamma}$ across all images of HYTA. Table 4.4 summarizes the results. We observe that certain color channels viz. c_{15} ($\frac{B-R}{B+R}$), c_{13} ($\frac{R}{B}$), c_5 (S) have higher relevance scores as compared to others, making these color channels favorable candidates for sky/cloud image segmentation. On the other hand, color channels c_{11} (a^*) and c_4 (H) have low relevance scores, indicating that these color channels contribute less to the decision attribute for (non-) cloud pixels. Therefore, the latter color channels are not conducive for sky/cloud segmentation.

c_1	0.70	c_4	0.46	c_7	0.72	c_{10}	0.66	c_{13}	0.84	c_{16}	0.66
c_2	0.66	c_5	0.82	c_8	0.78	c_{11}	0.33	c_{14}	0.69		
c_3	0.58	c_6	0.58	c_9	0.69	c_{12}	0.61	c_{15}	0.84		

Table 4.4: Average relevance value across all images of the HYTA database. The most relevant color channels are highlighted in bold.

4.6.5 Cloud Classification Performance

In this section, we evaluate the discriminative power of each of the 16 color channels with respect to the cloud classification task. The performance results of the 16 color channels will help us to benchmark our proposed rough-set based method and other statistical benchmarking measures.

We follow a supervised learning approach and train a Support Vector Machine (SVM) to validate our findings. We use each of the 16 color channels separately as candidate feature vectors and train 16 different SVMs. The trained SVMs are then used for sky/cloud pixel classification in order to check the efficacy of respective color channels in the classification task. We randomly segregate the HYTA dataset into 15 training images and 17 test images.

For an objective evaluation of our algorithm, we report the F-score and Accuracy. Suppose TP , FP , TN , and FN denote the True Positives, False Posi-

tives, True Negatives, and False Negatives, respectively, in this binary classification task. Accuracy is defined as the ratio of pixels that are correctly classified, $(TP + TN)/(TP + TN + FP + FN)$. The F-score, a popular metric in a binary classification problems, is defined as the harmonic mean of Precision = $TP/(TP + FP)$ and Recall = $TP/(TP + FN)$.

Figure 4.6 shows the binary classification results for each of the 16 color channels, computed over 50 different random selections of training and test sets.

Figure 4.6: (a) Accuracy and (b) F-scores for 16 color channels for all images in the HYTA database. For each box, the central red line indicates the median, the top and bottom edges correspond to 25th and 75th percentiles, and the whiskers represent the extreme data points.

We observe from Fig. 4.6 that color channels c_{15} ($\frac{B-R}{B+R}$) and c_5 (S) have relatively higher accuracy and smaller variation as compared to the other color channels. Conversely, color channels c_{11} (a^*) and c_4 (H) score poorly in terms of accuracy. Therefore, each color channel has different discriminative power to classify sky and cloud pixels from ground-based images. This ranking of color channels in terms of their classification performance serves as the ground-truth to verify the efficacy of different color channel selection algorithms.

4.6.6 Benchmarking

In this section, we will benchmark our proposed rough-set based method and other color channel selection methods with the accuracy measures of the color channels.

We check the correlation of the average accuracy scores for all the color channels with the normalized scores obtained from different approaches. Table 4.5 shows the correlation coefficients of these approaches with the average classification accuracy of the color channels. Our proposed method using rough sets achieves the highest correlation.

Methods	Correlation (r)
Proposed approach	0.84 ($\uparrow$)
Bimodality	-0.58 ($\downarrow$)
Loading factors	0.57 ($\uparrow$)
ROC curve	0.78 ($\uparrow$)
Kullback-Leibler divergence	0.12 ($\downarrow$)

Table 4.5: Correlation of cloud classification accuracy with ranking scores obtained using different methods. The $\uparrow$ (or $\downarrow$) indicates if higher (or lower) magnitude signifies better performance.

Figure 4.7 shows the respective scatter plots. The correlation coefficient is highest ($r = 0.84$) for our proposed algorithm. From Fig. 4.7(a), we can clearly see that color channels c_5 (S) and c_{15} ($\frac{B-R}{B+R}$) are better candidates for sky/cloud classification. Similarly, color channels c_{11} and c_4 with lower relevance scores are poor candidates. Figure 4.7(b) also reveals that c_5 and c_{15} are conducive color channels as their PBI values are closer to 1. Similar results can be drawn from Fig. 4.7(c), where favorable color channels c_5 and c_{15} have comparatively lower KL-divergence values, and color channel c_4 has the highest KL-distance. However, the order of ranking for the other color channels is poor, leading to a low correlation value ($r = 0.12$).

From these results, we observe that the relevance criterion has the highest correlation with the cloud segmentation accuracy among all methods. Therefore we conclude that our proposed rough set based color channel selection algorithm is useful to rank color channels and identify suitable ones for image segmentation. We

Figure 4.7: Scatter plot between average accuracy and (a) relevance ($r = 0.84$), (b) bimodality ($r = -0.57$), and (c) Kullback-Leibler distance ($r = 0.12$) for all 16 color channels (cf. Table 7.1). Our proposed approach using relevance scores achieves the highest correlation when ranking the color channels with respect to their segmentation performance.

also note that certain color channels viz. c_5 and c_{15} always perform best in all three benchmarking methods. Conversely, color channels c_4 and c_{11} rank lower than others. Visual inspection of these color channels confirms these findings (cf. Fig. 4.1).

4.7 Discussion

We discussed the performance of the different statistical tools and our proposed rough-set based selection method; and identified the most appropriate color channels for cloud detection. However, we used a *single* color channel in our analysis - and identified the color channels that are conducive for detection. In this section, we explore the possibilities of using multiple channels for this purpose.

As described in Section 4.4.4, we perform fuzzy c-means clustering using the varied existing color channels. There are 16 color channels for 1D clustering. In a similar manner, there are $\binom{16}{2}$ unique color channel combinations for 2D-clustering. Their performance is illustrated in Fig. 4.8 in an upper-triangular grid, where the intensity of a particular grid denotes the F-score of that particular color channel pair. Similarly, the diagonal elements represent the F-scores of the corresponding 1D color channels. Since the ground truth provided by the HYTA database are binary images consisting of two classes, namely sky and cloud, we convert the probabilistic

Figure 4.8: F-Scores for all unique cases of 1D (along the diagonal) and 2D clustering, represented on a brightness scale. The highest F-Scores for 1D (dotted frames) and 2D cases (bold frames) are indicated numerically.

distribution of cloud pixels $p(\mathbf{x}_f)$ (cf. Section 4.4.4) to a binary image by thresholding, i.e. a pixel having a probability of cloud $> 50\%$ is assigned to a cloud pixel and others to sky pixels.

The most promising color channels for 1D-clustering are c_{13} (R/B), c_{15} ($\frac{B-R}{B+R}$), and c_5 (S) with high precision, recall, and F-scores. For 2D, the best combinations include c_5 - c_8 , c_8 - c_{13} , and c_8 - c_{15} (c_8 is the in-phase component of the YIQ color model and it represents essentially another variety of red-blue difference), with several other close contenders. However, there is hardly any performance gain from using two color components rather than just one. Therefore, for the task of cloud detection, a single color channel (if chosen carefully) can have optimal performance, in comparison to computationally-expensive multiple color channels.

4.8 Conclusions

In this chapter, we have presented a systematic analysis of color spaces and components for detection of clouds in sky/cloud images. We analyzed using a variety of statistical tools and methods: distribution bimodality, PCA, clustering and KL divergence. Experimental evaluation with a cloud segmentation database yields consistent results across analysis methods. Additionally, we proposed a rough-set based method that provides the best correlation results with the performance measures of the color channels. Also, we observed that clustering results obtained using the best 1D color channels or 2D color channel pairs (including red-blue combinations R/B and $\frac{B-R}{B+R}$, saturation S , or in-phase component I) are on par with each other. Future works involve the use of these identified color channels for several tasks.

Chapter 5

Color-based Image Segmentation

5.1 Introduction

Segmentation is one of the first steps in sky/cloud image analysis. It remains a challenging task because of the non-rigid, feature-less, and poorly-defined structure of clouds, whose shape also changes continuously over time. Thus, classical image segmentation approaches based on shape priors [29] are not suitable. Furthermore, the wide range of lighting conditions (direct sunlight to completely covered skies) adds to the difficulty. Therefore, in most cases, color is used as the most discriminatory cue. In the previous Chapter 4, we have provided a systematic analysis of the various color spaces and components. In one of our earlier works [83], we learnt the semantics of sky/cloud images for an automatic annotation of pixels with three class labels—sky, thin clouds and thick clouds. However, the segmentation results were not promising for thin clouds. Therefore, based on the findings of the previous chapter, we propose a probabilistic color-based image segmentation algorithm in this chapter.

5.2 Outline of Our Contribution

The motivation of this chapter is to propose a robust framework for color-based cloud segmentation under any illumination conditions, including a systematic analysis of color channels. The framework is based on partial least squares (PLS) regression and provides a straightforward, parameter-free supervised segmentation method. We show that our approach is robust and offers a superior performance across two different databases as compared to current state-of-the-art algorithms. Furthermore, it allows annotating each pixel with a degree of *belongingness* to the sky or cloud category, instead of the usual binary labeling.

The main novel contributions of this chapter compared to earlier works include:

- Introduction of a large public sky/cloud image database with segmentation masks;
- Extensive evaluation of color components and selection of appropriate color channels on two different sky/cloud image databases;
- Robust learning-based framework for sky/cloud segmentation that outperforms existing methods.

The rest of this chapter is organized as follows. Section 5.3 discusses our proposed supervised probabilistic segmentation framework. The sky/cloud databases used for evaluation, including our newly introduced database, are presented in Section 5.4. Section 5.5 presents the experimental evaluation of the segmentation framework. We extend our discussion to High Dynamic Range (HDR) images in Section 5.6. Section 5.7 concludes this chapter.

5.3 Probabilistic Segmentation

Using statistical tools, we identify the most discriminative color channels for cloud segmentation. Out of 16 color channels, the best performing color channels may be subsequently used as discriminatory feature $\mathbf{X}_i^f$ in the segmentation framework.

Our objective is to predict the label of a pixel as either sky or cloud, given a set of training images with manually annotated ground truths. We formulate this task of sky/cloud image segmentation in a probabilistic manner using PLS regression. Both Principal Component Analysis (PCA) and PLS are based on the same underlying motivation, namely to extract components that accounts for maximum variation in input data. Unlike alpha matting, we do not assume that a particular pixel is a linear combination of two classes – *sky* and *cloud*.

We start with the assumption that the sky/cloud segmentation problem is dependent on a large number of explanatory independent variables (color channels in this case). Using dimensionality reduction techniques like PCA, we observe the degree of correlation amongst the variables and the amount of variance captured. PCA provides this information based only on the independent input color channels and project the data into new sets of orthogonal axes. In PLS however, both the input color channels and the output labels (sky/cloud) are considered. Using a set of manually defined ground truth images, we model the relationship between the input features and the output label. In the testing stage, we use this estimated model to predict the output labels of the pixels of a given test image.

Suppose, for a sample $\mathbf{X}_i \in \mathbb{R}^{mn \times n}$, the feature vector is denoted by $\mathbf{X}_i^f \in \mathbb{R}^{mn \times k}$ and the corresponding binary ground-truth image as $\mathbf{Y}_i \in \mathbb{R}^{mn \times 1}$. The feature vector $\mathbf{X}_i^f$ consists of k color channels from the set of 16 color channels. We decompose the feature vector $\mathbf{X}_i^f$ and labels $\mathbf{Y}_i$ as

$$\mathbf{X}_i^f = \mathbf{T}_i \mathbf{P}_i^T + \mathbf{E}_i, \quad (5.1)$$

$$\mathbf{Y}_i = \mathbf{U}_i \mathbf{Q}_i^T + \mathbf{F}_i, \quad (5.2)$$

where $\mathbf{T}_i \in \mathbb{R}^{mn \times p}$ and $\mathbf{U}_i \in \mathbb{R}^{mn \times p}$ are p extracted latent matrices. $\mathbf{P}_i \in \mathbb{R}^{k \times p}$ and $\mathbf{Q}_i \in \mathbb{R}^{1 \times p}$ are the loading matrices, and $\mathbf{E}_i \in \mathbb{R}^{mn \times k}$ and $\mathbf{F}_i \in \mathbb{R}^{mn \times 1}$ are the residual matrices. Henceforth, for the sake of brevity and without loss of generality we drop the suffix i in the notations.

In the generalized partial least squares regression sense, we decompose $\mathbf{X}^f$ and $\mathbf{Y}$ so as to maximize the covariance between $\mathbf{T}$ and $\mathbf{U}$ [84]. Under the assumption

that the input matrix $\mathbf{X}^f$ can be expressed as a linear combination of a few latent variables, we can assume the existence of a linear relationship between $\mathbf{T}$ and $\mathbf{U}$ such that

$$\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{H}, \quad (5.3)$$

where $\mathbf{D} \in \mathbb{R}^{p \times p}$ is a diagonal matrix and $\mathbf{H}$ is the residual matrix. Combining Eqs. (5.2) and (5.3), we get

$$\mathbf{Y} = \mathbf{T}\mathbf{D}\mathbf{Q}^T + (\mathbf{H}\mathbf{Q}^T + \mathbf{F}), \quad (5.4)$$

$$\mathbf{Y} = \mathbf{T}\mathbf{C}^T + \mathbf{F}^*, \quad (5.5)$$

where $\mathbf{C}^T$ is the matrix of regression coefficients and $\mathbf{F}^*$ is the residual matrix. We extract score vectors $\mathbf{t}$ and $\mathbf{u}$ from predictors and response variables respectively, and express as:

$$\mathbf{t} = \mathbf{X}\mathbf{w}, \quad (5.6)$$

$$\mathbf{u} = \mathbf{Y}\mathbf{c}, \quad (5.7)$$

where $\mathbf{w}$ and $\mathbf{c}$ are the corresponding weight vectors of $\mathbf{X}$ and $\mathbf{Y}$. These score vectors are necessary to define the response and predictor variables in the PLS setting. $\mathbf{W}$ represents a set of orthogonal basis vectors, such that $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p)$. From [85], we can express the regression coefficient matrix as

$$\mathbf{T} = \mathbf{X}\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}. \quad (5.8)$$

Using Eqs. (5.4) and (5.8), we get

$$\mathbf{Y} = \mathbf{X}\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{C}^T + \mathbf{F}^*, \quad (5.9)$$

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{F}^*, \quad (5.10)$$

where $\mathbf{B} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{C}^T$ represents the matrix of regression coefficients.

In our proposed approach, we select a set of training images from the database and obtain the regression coefficient matrix $\mathbf{B}$ as per Eq. (5.10). In the testing stage, we obtain the corresponding segmentation result as

$$\mathbf{Y}_{\text{test}} = \mathbf{X}_{\text{test}}\mathbf{B} \quad (5.11)$$

by using the obtained regression coefficient matrix $\mathbf{B}$. However, because of the residual matrices and the inherent imperfections in modeling, the values of $\mathbf{Y}_{\text{test}}$ are not discrete in nature. We do not employ any thresholding at this stage, but rather normalize these values to the range $[0, 1]$ ($0 = \text{sky}$, $1 = \text{cloud}$). These normalized values provide a probabilistic indication of the *belongingness* of a particular pixel to a specific class (i.e. cloud or sky) instead of a binary decision.

One of the most important pre-requisites for rigorous evaluation of any segmentation algorithm is an annotated dataset of suitable images. Most research groups working on sky/cloud segmentation use their own proprietary images that are not publicly available. In the subsequent section, we will introduce a new segmentation database that is released to the research community for future benchmarking purposes.

5.4 Sky/Cloud Image Databases

In this section, we briefly mention the currently available database containing ground truth images and describe the need for a more comprehensive and consistent database.

5.4.1 HYTA Database

Till date, the only publicly available sky/cloud image dataset annotated with segmentation masks was the Hybrid Thresholding Algorithm (HYTA) database¹ by Li et al. [33]. More details of HYTA dataset were already provided in our earlier Chapter 4.

¹Thanks to Prof. Qingyong Li, Professor of Beijing Jiaotong University for providing the dataset.

5.4.2 SWIMSEG Database

Because of the dearth of public databases for sky/cloud images, we created SWIMSEG, the **S**ingapore **W**hole Sky **I**Maging **S**EGmentation Database.² It consists of 1013 images that were captured by a custom-designed ground-based sky imager called Wide Angle High Resolution Sky Imaging System (WAHRISIS), developed and deployed at Nanyang Technological University in Singapore [5]. The imaging system of our ground-based sky camera consists of a Canon EOS Rebel T3i (a.k.a. EOS 600D) camera body and a Sigma 4.5mm F2.8 EX DC HSM Circular Fish-eye Lens with a field of view of 180 degrees. We discussed the various models of our sky cameras earlier in Chapter 3.

The images captured by this system suffer from distortions in illumination and color, which need to be corrected. An example is shown in Fig. 5.1. First, because of a phenomenon called vignetting, the images are comparatively darker at the edges as compared to the center. Vignetting occurs because the incident light rays reach the camera sensor with varying angles and is particularly noticeable for fish-eye lenses. We analyze vignetting in our imaging system using an integrating sphere [5] and perform the necessary correction in the image, the result of which is shown in Fig. 5.1(b).

Second, the colors in the captured images vary due to the different weather conditions and capturing hours of the images and often do not correspond to the actual colors in the scene. We use a color checkerboard with 18 different color patches as a reference, whose exact $L^*a^*b^*$ values are specified by the manufacturer. Using the mapping between captured and actual color, we thereby obtain the color-corrected image, which is shown in Fig. 5.1(c).

Finally, the images from the database are undistorted using the geometric calibration function, which models the effect of the lens. This function relates each pixel of the captured image to the azimuth and elevation angle of the corresponding incident light ray. Using that information, we can project the image onto a hemisphere whose center is the focal point. This is shown in Fig. 5.2(a). We generate

² The SWIMSEG database can be downloaded from <http://vintage.winklerbros.net/swimseg.html>.

Figure 5.1: Image correction w.r.t. illumination and color.

undistorted pictures using a ray tracing approach by considering an imaginary camera (with a standard lens) at center of the hemisphere, which points towards a user defined direction with a given azimuth and an elevation angle. In order to simulate this camera, we consider an output image with dimension 600×600 . Each output image corresponds to a viewing angle of 62° . The rays passing through each pixel will intersect the hemisphere and then converge towards the center. The value of a pixel is then equal to the color of the hemisphere at its intersection point. We perform this process at varying azimuth angles for a particular angle of elevation to generate several undistorted images. Figure 5.2 shows the output image as well as the lines corresponding to its borders and diagonals on the original image.

The corresponding sky/cloud segmentation masks of the images were created in consultation with cloud experts from the Singapore Meteorological Services. A few representative sample images from this database are shown in Fig. 5.3.

SWIMSEG consists of 1013 patches that were handpicked carefully from images taken over the period from October 2013 till July 2015 in Singapore. In the selection process, different parameters (viz. azimuth and elevation angle of sun, time of day, cloud coverage percentage, camera parameters) were considered. The camera elevation angle is set to three distinct angles (36° , 45° , and 60°), whereas the azimuth angle is sampled at various angles around the unit hemisphere. The camera is set to auto mode while capturing pictures, and we observe that the settings tend to remain similar across all images. This is primarily because weather conditions in Singapore

Figure 5.2: Generation of undistorted images using ray-tracing. (a) Projection of the image on a hemisphere; (b) Input image with borders and diagonals of the output image; (c) Undistorted output image.

remain fairly similar throughout the year.

Figure 5.4 characterizes the images in the dataset according to the time of day they were taken, cloud coverage, and distance from the sun. The sun rises and sets at around the same time throughout the year in Singapore, namely at 7am and 7pm local time. The number of clear sky patches is relatively low as compared to moderately-cloudy or overcast conditions. This is because Singapore experiences a tropical climate in which clear-sky conditions are rare. Although most of the image patches are sampled at a certain distance from the sun to avoid saturation effects, we still include a substantial number of images close to sun.

5.5 Segmentation Results

In the previous Section, we identified suitable candidate color components for segmentation, such as c_5 (saturation) and various blue-red ratios (c_{13} , c_{14} , c_{15}). These color channels are expected to perform better than others. We now evaluate the segmentation results for all 16 color channels, by individually considering them as the discriminatory feature vector $\mathbf{X}^f \in \mathbb{R}^{mn \times 1}$ for segmentation. We showed previously [86] that there is hardly any performance improvement when using more than 1 color channel in the feature matrix $\mathbf{X}_f$.

We perform extensive experiments of our proposed approach on the HYTA and

Figure 5.3: Sample images from the SWIMSEG database (top row), along with corresponding sky/cloud segmentation ground truth (bottom row) and percentage of cloud coverage. SWIMSEG databases. Each of the images in the two datasets are expressed in the form of $\hat{\mathbf{X}}_i$, as shown in Eq. (4.2). The corresponding feature vector $\mathbf{X}_i^f$ is computed for both training and testing stages. The training images are used to calculate the matrix of regression coefficients $\mathbf{B}$ as per Eq. (5.10). In our experiments, we use `plsregress`, the MATLAB implementation of the SIMPLS algorithm [84] for this. We set $p = k = 1$, as we use a single color channel for evaluation. During the testing phase, $\mathbf{X}_{\text{test}}$ of the testing image is computed. Using the matrix of regression coefficients $\mathbf{B}$, the corresponding $\mathbf{Y}_{\text{test}}$ is computed.

In the ideal scenario, the value of $\mathbf{Y}_{\text{test}}$ is either 0 or 1. However, in the actual implementation, the values are continuous in nature. Suppose we denote the obtained values of $\mathbf{Y}_{\text{test}}$ as $\{v_1, v_2, \dots, v_{mn}\}$. Let θ be the function that normalizes the obtained values of $\mathbf{Y}_{\text{test}}$ to the interval $[0, 1]$:

$$\theta(v_i) = \frac{v_i - \min v_i}{\max v_i - \min v_i}.$$

We perform *soft* thresholding by normalizing these values in the range $[0, 1]$. Therefore, this normalized value indicates the belongingness of a pixel to belong to the cloud (or sky) category.

Figure 5.4: Distribution of images in SWIMSEG according to various characteristics.

The objective evaluation is based on the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) samples in a binary image. We report Precision, Recall, F-score, and Misclassification rate for the different benchmarking algorithms, which are defined as follows:

$$\begin{aligned} \text{Precision} &= TP / (TP + FP), \\ \text{Recall} &= TP / (TP + FN), \\ \text{F-score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned}$$

$$\text{Misclassification rate} = \frac{FP + FN}{TP + TN + FP + FN}.$$

5.5.1 Choice of Color Channels

As described earlier, the segmentation performance is dependent on the choice of color channels. In order to understand the effect of color channels, we use the individual color channels to constitute the feature vector $\mathbf{X}^f$ in this experiment. We choose training and test set sizes as per Table 5.1.

Database	Training Set	Test Set	Total
HYTA	17	15	32
SWIMSEG	500	513	1013

Table 5.1: Number of training and test images for HYTA and SWIMSEG databases.

We compute F-scores (the harmonic mean of precision and recall) to characterize the segmentation performance of each color channel. We perform a 5-fold and 10-fold cross validation in HYTA and SWIMSEG datasets, respectively. In each of these experiments, 1 fold is used for testing and the remaining folds are used in the training stage. Figure 5.5 shows the results for HYTA and SWIMSEG datasets.

(a) HYTA

(b) SWIMSEG

Figure 5.5: Binary classification performance for different color channels.

The classification performance for individual color channels follows the same trend

in both HYTA and SWIMSEG dataset. Color channels c_5 and the red-blue ratio channels $c_{13,15}$ perform best, which is consistent with what we found in the earlier Chapter 4. Based on these results, we conduct the subsequent evaluations with c_{15} as the feature vector $\mathbf{X}^f$.

5.5.2 Effect of Database Characteristics

We study the effect of the different characteristics of the database on the obtained F-score values of our proposed segmentation algorithm by checking whether the obtained segmentation results are biased towards a particular set of image parameters, specifically time of day, cloud coverage, and distance from the sun. We present the distribution of F-score values across these parameters in Fig. 5.6.

The average F-score is slightly high in the morning and late afternoon, while it drops during mid-day. This is because the images captured during mid-day have a higher tendency to get over-exposed because of the presence of direct sun. On the other hand, the images captured at the morning and evening hours of the day receive slant sun-rays, and contain more clouds. The variation of the F-score values w.r.t. the percentage of cloud in an image is shown in Fig. 5.6(b). We observe that the performance is marginally better for higher percentage of cloud coverage in an image. Finally, we note that there is no clear relationship of the distance of the image patch from the sun on the obtained F-score values, which is illustrated in Fig. 5.6(c).

Figure 5.6: Distribution of average F-score value as a function of various characteristics of the images in SWIMSEG.

5.5.3 Benchmarking

We benchmark our proposed approach with current state-of-the-art cloud detection algorithms: Li et al. [33], Souza et al. [27], Long et al. [20], and Mantelli-Neto et al. [18]. We also compare against another popular image segmentation algorithm viz.

SLIC superpixels [87].

Li et al. [33] use HYTA that combines both adaptive and fixed threshold approach. Souza et al. [27] use the saturation component of the intensity, hue, and saturation (*IHS*) color model in defining appropriate thresholds to obtain the final segmentation mask. Long et al. [20] use the ratio of red and blue channels to model the molecular scattering in the atmosphere. Mantelli-Neto et al. [18] use multi-dimensional Euclidean geometric distance and Bayesian approaches to identify sky and cloud pixels.

The popular superpixel segmentation Simple Linear Iterative Clustering (SLIC) divides the input image into a number of superpixels. We combine it with the DBSCAN algorithm [88] to cluster the superpixels and subsequently categorize them into sky/cloud based on the average ratio of red and blue of a particular superpixel.

In addition to these state-of-the-art cloud segmentation algorithms, we also compare our proposed algorithm with other typical machine-learning based methods. We extract discriminative features from image, and use them in a classifier model. In our case, we employ Support Vector Machine (SVM) classifier along with various image features.

We use several feature descriptors viz. Local Binary Patterns (LBP) [89], Color Histogram (colorHIST) [90], and Dense Scale Invariant Feature Transform (dSIFT) [38] for extracting discriminatory features from sky/cloud images. Quite recently, Yuan et al. [34] used Bag-Of-Words (BOW) model to model cloud- and non-cloud- satellite images from dSIFT local features. Yuan et al. [34] over segments a satellite image into sub-images using SLIC superpixel generation method. Image descriptors are thereby extracted from these sub-patches using dense SIFT extractor. Subsequently, the BOW model is used to model the cloud- and non-cloud images and the corresponding codebook is generated. A Support Vector Machine (SVM) classifier is finally used for the super-pixel classification into cloud- and non-cloud- parts [34]. In addition to these features, we also use texture descriptors (using Schmid filter [91]) along with BOW and SVM as a benchmarking cloud segmentation algorithm. Furthermore, we also use the raw gray-scale (GRAY) color channel values as discriminatory feature in feature extraction stage. These features are used in training a SVM classifier. In

the testing stage, these descriptors are again computed for the test images, and the trained SVM classifier classifies them to either sky or cloud pixels. Therefore, these different image features together with SVM classifier form a collection of machine-learning based sky/cloud classification algorithms.

In our proposed approach, we assign a membership value for the cloud category to each pixel. However, as the available ground-truth masks are binary, we convert our probabilistic map into a binary map by simple thresholding in order to perform the objective evaluations. The threshold value can be any value in the closed interval $[0,1]$; a lower value indicates a higher tendency for a pixel to classify as *cloud* and a higher value indicates a higher tendency to classify as *cloud* pixel. Figure 5.7 shows the ROC curve for our proposed algorithm by changing the threshold value in the closed interval $[0,1]$ for both HYTA and SWIMSEG databases. 20 trials were performed for each threshold value, with training and testing sets of images chosen randomly for each trial. The subsequent results were obtained with a threshold of 0.5.

(a) HYTA

(b) SWIMSEG

Figure 5.7: ROC curves of our proposed algorithm, including error bars in X- and Y-directions for (a) HYTA and (b) SWIMSEG datasets.

Figure 5.8 shows a few sample images from the HYTA and SWIMSEG databases together with the segmentation ground truth and the corresponding output images obtained by our proposed approach and other existing methods. Most of the approaches work quite well when there is a clear contrast between sky and cloud regions, as is often the case for the HYTA database. The method of Mantelli-Neto et al. [18] fails on images that contain non-uniform illumination of cloud mass from sunlight. The detection of thin clouds is the most problematic: Long et al. [20] and SLIC+DBSCAN methods classify the entire image patch as cloud in such cases. The method of Li et al. [33] fails in cases where the boundary between sky and cloud is ill-defined. We observe from the last two examples in Fig. 5.8 that Li et al. over-segments the corresponding input images. Except for Souza et al. [27], most of the approaches fail for images of the SWIMSEG dataset with generally less well-defined sky/cloud separation. Similarly, the other machine-learning based methods viz. GRAY+SVM and dSIFT+BOW+SVM [34] perform poorly for the wide range of images. This is because the Bag-of-words model obtained for cloud- and non-cloud- superpixels fails to discriminate between the superpixels in the test images. The results obtained by our PLS-based approach show that it works well across diverse illumination conditions and for both databases.

The simulations were conducted in Matlab on a 64-bit Ubuntu 14.04 LTS workstation, Intel i5 CPU @2.67GHz. In terms of computation time, our proposed method takes an average of 1.31s and 1.89s for a single image from the HYTA and SWIMSEG databases, respectively. In the training stage, HYTA requires 21.5s and SWIMSEG required 1018.6s for computing the regression coefficient matrix $\mathbf{B}$. The computation times of other benchmarking algorithm are provided in Table 5.2.

Table 5.2 provides an objective evaluation of our proposed approach with other state-of-the-art algorithms for all images in the testing set of the two databases. Existing algorithms require extensive tuning using various manually-defined thresholds, and achieve either high precision or high recall in our evaluation, but not both. Long et al. and Mantelli-Neto et al. have high recall values. However, they fare poorly in the corresponding precision values. Souza et al. on the other hand has a high precision value, but low recall. These existing methods often suffer from under- or

Figure 5.8: Sample images from HYTA (left 3 columns) and SWIMSEG (right 3 columns) databases, along with the corresponding ground truth and the sky/cloud segmentation obtained using various methods. The *blue* and *red* in the colormap indicate *sky* and *cloud* respectively.

over-identification of cloud pixels as they are threshold-based, and a single threshold may not work well for all types of sky/cloud images. Our proposed approach performs best for both HYTA and SWIMSEG databases on the basis of F-scores, highlighting the effectiveness of PLS-based learning method.

	Methods	Precision	Recall	F-score	Misclassification rate	Time [s]
HYTA	Li et al.	0.89	0.82	0.81	0.12	1.34
	Souza et al.	0.88	0.67	0.65	0.15	1.33
	Long et al.	0.64	0.97	0.71	0.26	1.22
	Mantelli-Neto et al.	0.54	0.97	0.63	0.44	1.43
	SLIC + DBSCAN	0.65	0.83	0.60	0.32	5.10
	GRAY + SVM	0.89	0.58	0.67	0.31	2.44
	LBP + SVM	0.80	0.65	0.72	0.22	4.29
	ColorHIST + SVM	0.75	0.74	0.75	0.21	2.47
	dSIFT + BOW + SVM	0.61	0.66	0.62	0.38	4.91
	Texture + BOW + SVM	0.82	0.62	0.69	0.25	2.60
	Proposed Method	0.94	0.80	0.85	0.10	1.31
SWIMSEG	Li et al.	0.90	0.86	0.88	0.11	2.06
	Souza et al.	0.95	0.76	0.81	0.14	2.04
	Long et al.	0.71	0.98	0.80	0.23	1.83
	Mantelli-Neto et al.	0.70	0.97	0.79	0.24	2.16
	SLIC + DBSCAN	0.72	0.79	0.65	0.33	5.39
	GRAY + SVM	0.87	0.56	0.64	0.33	2.61
	LBP + SVM	0.62	0.65	0.63	0.36	4.73
	ColorHIST + SVM	0.81	0.64	0.66	0.31	2.63
	dSIFT + BOW + SVM	0.65	0.88	0.72	0.28	5.04
	Texture + BOW + SVM	0.82	0.71	0.70	0.31	2.74
	Proposed Method	0.92	0.90	0.90	0.09	1.89

Table 5.2: Performance evaluation using binary ground truth of HYTA and SWIMSEG databases. The best performance according to each criterion is indicated in bold. All experiments are evaluated on the same set of testing images from the HYTA and SWIMSEG databases. We also list the computation time for a single image for all algorithms.

5.6 High-Dynamic-Range Imaging for Cloud Segmentation

While there have been several studies analyzing clouds and their features from WSI images [20, 27, 32, 33, 86], most of them avoid the circumsolar region, because cap-

turing the details in this area is a non-trivial task. In this section, we show how to improve segmentation results by capturing a larger dynamic range of the sky using High-Dynamic-Range Imaging (HDRI) techniques. We then introduce HDRSeg, a graph-cut based segmentation algorithm that uses the HDR radiance map for accurate segmentation of sky/cloud images. Figure 5.9 briefly summarizes our proposed approach.

Figure 5.9: Proposed HDRSeg cloud segmentation approach. (a) High- (b) Mid- (c) Low-exposure Low-Dynamic-Range (LDR) images; (d) Tonemapped High-Dynamic-Range (HDR) image; (e) Binary output using HDRSeg. Saturated pixels in all images are shown in pink. The number of saturated pixels is significantly reduced in the tonemapped HDR image, without compromising on the fine cloud details.

The motivation is to propose the use of HDR imaging for cloud segmentation in ground-based sky cameras. We have already detailed the process of image capture and storage in our sky camera system an earlier Chapter 3 of this thesis. We showed that using HDR images for cloud imaging significantly reduces the amount of saturated pixels in an image, and is therefore an efficient manner to capture the circumsolar region. Furthermore, HDR imaging generally provides better segmentation results, as compared to LDR images, regardless of the segmentation method used.

5.6.1 Cloud Segmentation Using HDRI

We propose a graph-based segmentation algorithm called HDRSeg, that formulates the sky/cloud segmentation task as a graph-partitioning problem. Graph-cut for cloud segmentation was proposed earlier by Liu et al. [92]. However, Liu et al. used conventional LDR images in their segmentation framework and generated seeds

using Otsu threshold [93]. Also, they did not consider the circumsolar regions in their evaluation.

Notations

Suppose that the low-, mid- and high-exposure LDR sky/cloud images are represented by $\mathbf{Z}_i^L$, $\mathbf{Z}_i^M$ and $\mathbf{Z}_i^H$ respectively, $i = 1, \dots, N_h$, where N_h is the total number of HDR sets in the dataset. Without any loss of generality, $\mathbf{Z}_i$ denotes either low-, mid- or high-exposure LDR image in the subsequent sections. Each of these LDR images are *RGB* color images, $\mathbf{Z}_i \in \mathbb{R}^{m \times n \times 3}$, having a dimension $m \times n$ for each *R*, *G* and *B* channels. We generate the HDR radiance map $\mathcal{H}_i \in \mathbb{R}^{m \times n \times 3}$ from the set of three LDR images $\mathbf{Z}_i^L$, $\mathbf{Z}_i^M$ and $\mathbf{Z}_i^H$, as described in Section 3.6.1.

Let $\tilde{p}$ be a sample pixel in the image $\mathbf{Z}_i$. We aim to assign labels to each of the pixels $\tilde{p}$, as either *cloud* or *sky*. We denote the label as $L_{\tilde{p}}$, where $L_{\tilde{p}} = 1$ or 0 if $\tilde{p}$ is a *cloud* or *sky* pixel, respectively. We model this task as a graph-based discrete labeling problem, wherein we represent $\mathbf{Z}_i$ as a graph, comprising a set of nodes and edges.

Generating Seeds

Graph-cut based segmentation algorithms [94] require the user to initially label a few pixels as 'foreground' and 'background'. We refer to these prior labeled pixels as *seeds*. The process of generating seeds is generally done manually before partitioning the graph into two sub-graphs. In HDRSeg, we automatically generate these initial seeds by assigning a few pixels in the sky/cloud image with *sky* and *cloud* labels.

In our previous Chapter 4, we have performed a systematic analysis of the existing color spaces and components, and observed that $(B - R)/(B + R)$ is one of the most discriminating color channels for cloud segmentation (B and R indicate the blue and red channels of the image). Furthermore, in this current chapter, we have proposed a probabilistic approach for sky/cloud image segmentation, instead of the conventional binary approach. In this probabilistic approach, each pixel is assigned a *soft* membership to belong to *cloud* category, instead of a *hard* membership.

To illustrate these concepts, consider the example shown in Fig. 5.10. Figure 5.10(a) shows a sample LDR sky/cloud image $\mathbf{Z}_i$. We extract the $(B-R)/(B+R)$ color channel from this LDR image and show it in Fig. 5.10(b). A fuzzy c-means clustering on this extracted color channel yields the probabilistic output image, shown in Fig. 5.10(c). It denotes the confidence level of each pixel to belong to the cloud category. We assume that for a given pixel, the sum of membership values for sky and cloud category is unity.

Figure 5.10: Illustrative example to demonstrate the probability of a pixel belonging to the ‘cloud’ category. (a) Sample sky/cloud image; (b) $(B - R)/(B + R)$ color channel image; (c) Probabilistic output image using the method described in Section 5.3.

In HDRSeg, we apply fuzzy c-means clustering to the $(B - R)/(B + R)$ ratio channel of the HDR radiance map $\mathcal{H}_i$ to estimate the probability of a pixel belonging to the cloud category. We denote the ratio channel as $\tilde{\mathcal{R}}_i$. The membership values obtained by clustering provide us a mechanism in assigning the seeds with a degree of confidence. We assign these initial seeds for HDRSeg as follows: pixels having membership $>\tilde{\alpha}$ and $<(1 - \tilde{\alpha})$ are labeled as cloud and sky respectively. The value of $\tilde{\alpha}$ is a constant and is set experimentally (more on this below).

Partitioning the HDR Graph

The segmentation framework of HDRSeg employs a graph-based image segmentation approach, wherein we represent the ratio-image $\tilde{\mathcal{R}}_i \in \mathbb{R}^{m \times n}$ as a set of nodes and edges. Each edge of the graph is given a corresponding weight that measures the

dissimilarity between two pixels. We follow the work of Boykov and Jolly [94] and try to minimize the segmentation score E :

$$E = \sum_{\tilde{p}} A_{\tilde{p}} + \mu \sum_{(\tilde{p}, \tilde{q}) \in \tilde{\mathcal{N}}; A_{\tilde{p}} \neq A_{\tilde{q}}} B_{\tilde{p}, \tilde{q}}, \quad (5.12)$$

where $A_{\tilde{p}}$ denotes the data cost for an individual pixel $\tilde{p}$, $B_{\tilde{p}, \tilde{q}}$ denotes the interaction cost between two neighboring pixels $\tilde{p}$ and $\tilde{q}$ in a small neighborhood $\tilde{\mathcal{N}}$, and μ is the weighting parameter.

The complete proposed HDR segmentation algorithm can be summarized as follows:

Algorithm 1 HDRSeg

Require: LDR sky/cloud images with varying EVs.

Ensure: Segmented image.

- 1: Create HDR radiance map $\mathcal{H}_i$ from bracketted set of LDR images;
 - 2: Extract $(B - R)/(B + R)$ ratio channel $\tilde{\mathcal{R}}_i$ from HDR radiance map $\mathcal{H}_i$;
 - 3: Generate initial seeds from extracted ratio channel for image segmentation;
 - 4: Partition the ratio channel $\tilde{\mathcal{R}}_i$ into two subgraphs using the generated seeds;
 - 5: **return** Binary segmented image.
-

We illustrate the complete HDRSeg segmentation framework in Fig. 5.11. Figure 5.11(a) represent the three sample LDR images $\mathbf{Z}_i^L$, $\mathbf{Z}_i^M$ and $\mathbf{Z}_i^H$ respectively captured at $\{-4, -2, 0\}$ EV settings. We generate the corresponding HDR radiance map $\mathcal{H}_i$ from these LDR images. A tone-mapped version of $\mathcal{H}_i$ is shown in Fig. 5.11(b), for visualization purposes. We extract the $(B - R)/(B + R)$ ratio channel from $\mathcal{H}_i$, and generate the initial cloud and sky seeds marked in *green* and *red* color respectively, as shown in Fig. 5.11(c). The binary output image after graph cut is shown in Fig. 5.11(d).

Figure 5.11: Proposed HDRSeg framework for sky/cloud segmentation. (a) Exposure bracketed $\{-4, -2, 0\}$ Exposure Value (EV) LDR images; (b) High Dynamic Range (HDR) image tone-mapped for visualization purpose; (c) Image with initial seeds, where *cloud* seeds and *sky* seeds are represented in *green* and *red* respectively, and undecided pixels are represented in *yellow*; (d) Binary sky/cloud segmentation result. The saturated pixels in (b), (c) and (d) are masked in pink.

5.6.2 Experimental Evaluation

HDR Sky/Cloud Image Database

Currently there are no available HDR image datasets for sky/cloud images. We therefore consider 26 sets of HDR captures, which comprise a total of 78 LDR images. Each HDR capture is based on three LDR images (low-, mid- and high-) which were captured in Automatic Exposure Bracketing (AEB) mode of our camera. These high-quality HDR images are captured with our sky imagers, which we cropped to remove the occlusions by buildings or antennas near the horizon line. The average dimension of the captured LDR images is of 1800×1600 pixels. They were captured at various times of the day, under different weather conditions. The corresponding ground truth images were manually generated in consultation with experts from the Meteorological Service Singapore (MSS). We perform a detailed evaluation of several cloud segmentation algorithms on this newly created dataset.

Results

HDR imaging is an effective technique for cloud observation. It helps us better image the circumsolar region, without saturating the neighboring pixels. To illustrate this fact, we calculate the number of saturated pixels in the low-, med- and high- LDR images. We also calculate the number of saturated pixels (if any) in tonemapped images, in the dataset. Using our HDR techniques, we observe that the tonemapped

images have 87.2 times fewer saturated pixels, as compared to the high-exposure LDR images. Similarly, it has 9.5 times and 2.0 times fewer saturated pixels with respect to mid- and low-exposure LDR images.

In addition to containing fewer saturated pixels, HDR imaging also helps in improved cloud segmentation performance, regardless of the techniques used, as we demonstrate in the following. Cloud segmentation is essentially a binary classification problem, wherein we classify each pixel as either sky or cloud labels. We report the Precision, Recall and Error values for the different cloud segmentation methods. A detailed definition of these metrics are already provided earlier in Section 5.5.

For evaluation purposes, we benchmark HDRSeg with existing cloud segmentation algorithms. We consider Li et al. [33], Long et al. [20], Souza et al. [27] and Mantelli-Neto et al. [18]. All these existing algorithms, which were briefly described in Chapter 2, are designed for conventional LDR images.

We designed our proposed HDRSeg segmentation algorithm on the HDR radiance maps. We extract the $(B - R)/(B + R)$ channel from these high-dynamic radiance maps, and assign the initial seeds for *cloud* and *sky* pixels. The value of $\tilde{\alpha}$ determines the amount of initial seeds in HDRSeg. We empirically set the value of $\tilde{\alpha}$ to 0.98. We set a high value of $\tilde{\alpha}$, because it corresponds to higher confidence in assigning the correct labels.

Since existing cloud segmentation algorithms are designed for conventional LDR images, we evaluate these current cloud segmentation algorithms on the mid-exposure LDR images as well as the tonemapped images. Our proposed HDRSeg algorithm is the only one designed to make use of the full HDR radiance maps. However, for the sake of comparison, we also evaluate for mid-exposure LDR and tonemapped images. The detailed evaluation results of HDRSeg, along with the other cloud segmentation algorithms, are shown in Table 5.3.

From Table 5.3, we observe that HDR imaging improves the cloud segmentation performance, irrespective of the methods used. We observe that most of the benchmark algorithms (except for Li et al.) have a better performance with tonemapped images, as compared to mid-exposure LDR image. This is because a tonemapped version exhibits fewer saturated pixels and clearer contrast between sky and cloud,

Methods	Using LDR image			Using Tonemapped image			Using HDR Luminance map		
	Precision	Recall	Error [%]	Precision	Recall	Error [%]	Precision	Recall	Error [%]
Long et al.	0.68	0.99	31.77	0.72	0.99	27.31 (↑)	-	-	-
Souza et al.	0.76	0.99	21.80	0.79	0.95	20.98 (↑)	-	-	-
Mantelli-Neto et al.	0.69	1.0	31.33	0.84	1.0	15.41 (↑)	-	-	-
Li et al.	0.90	0.90	11.93	0.79	0.92	24.45 (↓)	-	-	-
HDRSeg	0.77	0.98	21.12	0.81	0.95	18.93 (↑)	0.95	0.88	10.44 (↑)

Table 5.3: The average scores across all the images are reported for the various benchmark methods. The *arrow* inside the parenthesis indicates the error performance of the algorithm as compared to its corresponding LDR version. We indicate increase (and decrease) in performance with ↑ (and ↓) respectively, when compared with LDR image. The best performance according to each criterion is indicated in bold.

as compared to the corresponding LDR image. However, HDRSeg using the entire HDR luminance map achieves the best error performance of 10.44% across all the methods.

Most of the other algorithms are biased towards a higher recall value (tendency to over-estimate cloud cover), as compared to precision. These existing algorithms are based on a set of thresholds, either fixed or adaptive, and are therefore more prone to high error percentage. However, HDRSeg uses the entire dynamic range of sky/cloud scenes to make a more informed decision in classifying a pixel as cloud or sky.

5.7 Conclusions

There are three primary advantages of our proposed segmentation approach compared to other cloud detection algorithms.

First, our cloud segmentation framework is not based on any pre-defined assumptions about color spaces and does not place any restrictions on the type of input images. We systematically compare different color channels and identify the most suitable ones. We also explain the reason for their better performance based on rigorous statistical evaluations in two datasets.

Second, many existing cloud segmentation algorithms rely on a set of thresholds, conditions, and/or parameters that are manually defined for a particular sky/cloud image database. Our proposed cloud segmentation approach is entirely learning-based and thus provides a systematic solution to training for a given database.

Third, conventional algorithms provide a binary output image from the input sky/cloud image. Although these binary images are informative in most cases, they lack flexibility and robustness. We have no indication of the effectiveness of the thresholding of the input image. In reality, because of the nature of clouds and cloud images, it is undoubtedly better to employ a *soft* thresholding approach. Our proposed approach achieves a probabilistic classification of cloud pixels. This is very informative as it provides a general sense of *belongingness* of pixels to the cloud class. However, as only binary ground-truth images are available, we convert these probability maps into binary images for performing a quantitative evaluation of our algorithm. In our future work, we plan to extend this analysis by creating *probabilistic ground truth* images, where the ground truth of the input images is generated by aggregating annotations from multiple experts.

Finally, from the extensive experiments on the two datasets (cf. Table 5.2), we observe that the performance with images from the SWIMSEG database is generally better than for HYTA, even though the behavior of color channels is similar in both (i.e. the same color channels do well for both SWIMSEG and HYTA). We believe this is because the images in SWIMSEG were captured with a camera that has been calibrated for color, illumination, and geometry [5].

Naturally, many challenges remain, for example: How do different weather conditions affect the classification performance? The weather in Singapore is relatively constant in terms of temperature and humidity, with little variation throughout the year. Usually it is either partly cloudy, or rainy (in which case the sky will be completely overcast, making segmentation unnecessary). As a result, our database is not suitable for investigating this question. Completely overcast conditions can be dealt with by simple pre-processing of the image before segmentation, e.g. by calculating the number of clusters, as we have done in our previous work [83].

In this chapter, we have proposed a probabilistic approach using PLS-based re-

gression for the segmentation of ground-based sky/cloud images. Our approach is entirely learning-based and does not require any manually-defined thresholds, conditions, or parameters at any stage of the algorithm. We also release an extensive sky/cloud image database captured with a calibrated ground-based camera that has been annotated with ground-truth segmentation masks. The conventional segmentation techniques are not catered for the circumsolar region, as those particular images are completely saturated. In the later part of this chapter, we solved the cloud segmentation problem for images around the sun using HDRI techniques.

Our future work will include the annotation of a database with probabilistic ground-truth segmentation maps. In the following chapters, we will go beyond segmentation, and devise algorithms to classify clouds into different types [83]. Also, we will provide theoretical studies in estimating cloud base height for multi-camera setups.

Chapter 6

Texture-based Image Classification

In this chapter, we discuss our work on multi-class image classification that is based on an efficient integration of *color* and *texture* cues. We present our proposed classification method to efficiently recognize cloud types from Low Dynamic Range (LDR) sky/cloud image patches. Multi-class classification of cloud types from images obtained by ground-based sky cameras is a challenging task. This is because clouds are generally featureless, and conventional machine learning techniques cannot be directly applied on them. Several feature descriptors are therefore developed to extract the most discriminatory cloud features. In this chapter, we systematically study the several features that were developed for cloud classification task. Also, we propose a new filter-bank based classification algorithm that efficiently extracts color and texture information from sky/cloud images, and surpasses the performance of these existing approaches.

6.1 Cloud Classification

The classification of clouds into various types is an important step because individual cloud types have different properties. The most common scheme is the genera-based cloud classification proposed by the World Meteorological Organization (WMO) [36]. It consists of 10 categories. They can be sorted in 3 basic groups based on their

shapes:

- Cumulus: heap clouds (cumulus, stratocumulus, altocumulus, cirrocumulus and cumulonimbus)
- Stratus: layer clouds (stratus, nimbostratus, altostratus and cirrostratus)
- Cirrus: hair-like/feathery clouds (cirrus, cirrocumulus and cirrostratus)

Each genus can be further subdivided into several species and varieties, according to their shape and structure, or their transparency and arrangement inside the cloud. In our specific application of analyzing cloud attenuation in satellite path links [14], we however do not assume such precise categorization. This is because, occurrence of the all types of cloud is not significant in our tropical climate. In tropical countries like Singapore, we do not witness all the various kinds of cloud types. Furthermore, it is not possible to classify into various cloud categories, based on visual inspection only. Moreover, they are too precise for our measurement techniques and our objectives. Therefore, we contrive the 10 different cloud types into 5 categories. More details about the 5 categories will be discussed later in Section 6.5 of this chapter.

Cloud classification is challenging because clouds are non-rigid in nature and do not have definite structure, color, texture or contour. The inherent difficulty in such problems is that the appearance of an object is heavily dependent on the illumination conditions, camera settings and the other ambient conditions. In the literature, several approaches based on the concept of filter responses and filter banks have been widely used. Gabor filters, Laplacian-of-Gaussian and other wavelet form filters are used in such applications. Moreover, several statistical features viz. mean, standard deviation and entropy of color channels are successfully used for cloud type recognition. In this chapter, we attempt to analyze these features from a machine-learning framework. We explore: which features have higher inter-class separability in the feature space, and are thus good candidates for classification?

Conventional feature extraction techniques fail in cloud type recognition task, as clouds are generally *fuzzy* in nature, and do not exhibit any strong features. Therefore, in most cases, color and texture plays the role of dominant cue features.

However, a seamless integration of color into a texture classification framework is still an open question. In this chapter, we attempt to solve the question: how can *color* and *texture* cues be combined for an efficient cloud type recognition task?

6.2 Outline of Our Contribution

In this chapter, we discuss our proposed methodology to efficiently integrate *texture* and *color* cues for robust image classification. We evaluate the efficacy of several cloud features that were specifically designed for cloud type recognition. A systematic selection of color channel and texture parameters of cloud images is an important step for better classification results. We propose a filter-bank based cloud classification framework, that is competitive with respect to the state-of-the-art algorithms. We release the first large-scale cloud categorization database called SWIMCAT to the research community. SWIMCAT stands for Singapore Whole sky IMaging CATegories Database, containing 784 images categorized in 5 distinct categories: clear sky, patterned clouds, thick dark clouds, thick white clouds, and veil clouds. In this chapter, we will present a thorough analysis of the proposed classification algorithm for the task of cloud type recognition.

The main contributions of this chapter include:

- A novel cloud classification framework which is extensively studied and tested in cloud images obtained in a tropical region like Singapore.
- Extensive evaluation of different cloud features in a large-scale cloud category database.
- Introduction of the first publicly-available, large-scale cloud category database called SWIMCAT, for further benchmark analysis.

The rest of this chapter is organized as follows. In Section 6.3, we analyze the different existing features that were developed for the task of cloud classification. We visualize these features, and identify the shortcomings of the existing methods. In this chapter, we perform a systematic analysis of the *color* and *texture* cues of

sky/cloud images, and propose a complete framework in Section 6.4 for cloud type recognition. Evaluation metrics and the first publicly available cloud categorization database are introduced in Section 6.5. We discuss our classification results in Section 6.6. Finally, Section 6.7 concludes the chapter.

6.3 Cloud Features

In this section, we will briefly go through the several features that were designed for effective cloud type classification. Suppose that a sample sky/cloud image is represented by $\mathbf{Z}_i \in \mathbb{R}^{m \times n \times 3}, i = 1, 2, \dots, N$ which is an *RGB* image in a dataset consisting of N images. We assume that there are a total of $\bar{p}$ classes, and each category consists of $\bar{q}$ sky/cloud images. The corresponding *R, G, B* images of $\mathbf{X}_i$ are represented by $\mathbf{Z}_i^R, \mathbf{Z}_i^G$ and $\mathbf{Z}_i^B$ respectively. Heinle et al. [31] uses a combination of these channels to design discriminatory features. Most of the other features were designed on the corresponding gray-scale image $\bar{\mathbf{Z}}_i$. For the sake of brevity and without the loss of generality, we will drop the index i in our subsequent discussions of this chapter.

6.3.1 Statistical Features (SF)

Statistical Features (SF) were widely used by [39, 95]. It consists of calculating several statistical measures from an image such as mean, median, standard deviation, skewness and kurtosis. These features provide a overall sense of the color of the image and its variation. For example, the gray-scale version of a thick dark cloud will be *darker* than a corresponding clear sky patch. The vectorized form of the gray-scale image $\bar{\mathbf{Z}}$ is represented as $\bar{\mathbf{z}}$. We derive the statistical features on the vectorized image. We calculate the mean, median, standard deviation, skewness and kurtosis on the vectorized image and represent it as $\bar{\mathbf{z}}_m, \bar{\mathbf{z}}_d, \bar{\mathbf{z}}_v, \bar{\mathbf{z}}_w$ and $\bar{\mathbf{z}}_k$ respectively. The SF feature is formed by concatenation all these measures together to form a 5-dimensional feature vector $[\bar{\mathbf{z}}_m, \bar{\mathbf{z}}_d, \bar{\mathbf{z}}_v, \bar{\mathbf{z}}_w, \bar{\mathbf{z}}_k]$ from a single gray-scale image $\bar{\mathbf{Z}}$. The SF do not take into account the spatial dependencies of the pixel values.

6.3.2 Spatial Gray Level Dependence Matrices (SGLDM)

Haralick et al. in [96] proposed a texture feature that takes into account the spatial dependency of the gray-scale pixel values in an image. This feature indicates the probability that two pixels have the same gray-level values for a specific distance in a given orientation. We represent this in the form of a matrix called Gray-Level Co-occurrence Matrix (GLCM). We calculate the corresponding 8-level quantized image of the gray-scale cloud image $\bar{\mathbf{Z}}$, and measure the spatial dependencies of two pixels with the same quantized value. For a particular pixel-of-interest in $\bar{\mathbf{Z}}$, we check the corresponding pixels which are oriented at angles of 0° , 45° , 90° and 135° . This helps in distinguishing clouds having a regular pattern, as compared to other cloud types. Once GLCMs are calculated for the four different orientations, we calculate 13 statistical measures for each of these orientations. We compute the angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, and two information measures of correlation. The corresponding means and ranges of these 13 statistical measures are computed across the 4 directions. We concatenate the mean values of these 13 measures and represent it as SGLDM_m . The corresponding feature with range values is called SGLDM_r .

6.3.3 Gray Level Difference Statistics (GLDS)

Gray Level Difference Statistics (GLDS) is a subset feature from SGLDM. It is developed by Weszka et al. [97] and has been originally used for terrain classification. It is now used for the task of cloud classification as well. We compute the gray-level co-occurrence matrix for the four orientations of 0° , 45° , 90° and 135° . The means of these four matrices are calculated, and we denote the 4-dimensional feature vector as GLDS.

6.3.4 Fractal Feature (FF)

Wu et al. in [98] developed a multi-resolution feature descriptor that provides good classification accuracy for ultrasonic liver images, as compared to other textural features. This feature describes the overall *roughness* in an image, which is based on fractal geometry in an image. This fractal dimension is represented using the Hurst exponent. We use the gray-scale cloud image $\bar{\mathbf{Z}}$, and compute three multi-resolution versions at ratios 1, 1/2 and 1/3. The Hurst exponents are calculated for these 3 versions, and we generate a 3-dimensional fractal feature vector.

6.3.5 Fourier Magnitude (FM)

The fourier transform is one of the conventional techniques to study the patterns in an image. In this case, we first generate the Fast Fourier Transform (FFT) of gray-scale cloud image $\bar{\mathbf{Z}}$. The FFT output contains both the magnitude and phase component. We use the magnitude component as the Fourier Magnitude (FM) feature vector for the image.

6.3.6 Heinle Feature (HF)

Heinle in [31] proposed a 12-dimensional feature for the task of cloud type recognition. It uses all the red, green and blue color channels, $\mathbf{Z}^R$, $\mathbf{Z}^G$ and $\mathbf{Z}^B$. Heinle Feature (HF) computes the following: the average of red and blue channel, standard deviation and skewness of blue channel, the average difference between red and green, red and blue, and green and blue. It also generates the GLCM matrix from the blue channel, and extracts the energy, contrast, homogeneity and entropy from it. The last dimension of the 12-dimensional feature vector is the cloud cover, which is generated from the blue channel.

6.3.7 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is a popular feature descriptor technique that is widely used in the computer vision community. It captures the distribution of gray-scale

pixels in 8-point neighborhood of a pixel value. LBP is a popular technique for feature extraction as it is invariant to illumination changes. Liu and Zhang in [99] used LBP in ground-based cloud classification. We use the LBP pattern of gray-scale cloud image $\bar{\mathbf{Z}}$ for our evaluation.

6.3.8 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) is another popular technique developed in computer vision community [38]. It generates a collection of descriptors from an image that are invariant to an affine rotation, translation and scaling. It is also robust to illumination changes. SIFT has been used in numerous applications. Wu et al. in [100] has extracted SIFT features from ground-based images to identify 4 distinct cloud conditions. For our analysis, we compute the SIFT descriptor on the gray-scale cloud image $\bar{\mathbf{Z}}$. Each descriptor is a 128-dimensional vector, and subsequently all descriptors are concatenated together to form a single feature vector.

These features have been used extensively for cloud type recognition at several occasions. However, there is no unified study on all these 9 cloud features. In a conventional classification framework, these feature descriptors are generated and used as an input in a classifier model (SVM, KNN etc.). In this chapter, we ask this question: does these *designed* cloud features help in increasing the inter-class separability, leading to an increased classification accuracy?

6.4 Proposed Classification Approach

The existing cloud features are useful in identifying the color spaces and textural features for cloud type recognition. However, the classification results using these features are low. In this section, we propose a cloud classification framework that integrates color and texture cues efficiently to provide higher classification accuracy. Our approach is based on the original Varma and Zisserman's texton approach [101].

In the original texton approach [101], the gray-scale input images are first convolved with a set of filters from a filter bank. These responses are clustered using

k-means clustering to generate a set of cluster centers, which are called *textons*. The process is repeated separately for all the classes. The generated textons from each classes are combined together to form a *texton dictionary*. The individual classes are finally modeled using a histogram distribution of the individual textons from the dictionary. In our proposed cloud classification methodology, we modify the original texton-based method primarily in two distinct ways.

1. We generate the texton dictionary by exploiting the inherent information across categories.
2. Instead of a gray-scale image, we incorporate color into the classification framework in a systematic manner.

Firstly, in the original approach, the set of textons are generated individually for each categories. In most cases, the differences between images across several categories are faint. This results in the generation of *similar* textons in the dictionary, leading to a higher mismatch in the model creation stage. The motivation of any classification algorithm is to generate appropriate discriminative models for each categories. In our proposed approach, we aggregate the filter responses of the training images across all the categories. We use conventional k-means clustering on this concatenated filter response instead of individual classes' filter responses. The generated cluster centers are the modified textons. Subsequently, discriminative histogram for each category is generated. Our method uses the minimum number of textons and help in a compact and relevant representation of the input data space.

Secondly, the original texton-based approach works on gray-scale image. Several methods have been introduced in the literature that exploits both color and texture in a single framework. One of the popular extensions to gray-scale texton-based classification is to apply the methodology separately to different color channels and components. The responses are then integrated into representative histograms of different categories. However, no clear analysis has been carried out which color channels are the most representative candidates in this approach? Usually, an educated guess is taken on what color spaces seem to appropriately represent the different

features ably. In [102], Sun and He generate the composite feature vector by combining both filter response and RGB color vector. Burghouts and Geusebroek in [103] have treated color as a separate entity and added it as a post-processing step. In our proposed approach for cloud categorization task, we systematically choose the color channel that is best suited for this task.

6.4.1 Feature Descriptor

In our earlier Chapter 4, we identified the color channels that are optimum for cloud type recognition. For the sample image $\mathbf{Z}_i \in \mathbb{R}^{m \times n \times 3}$, we extract the particular channel and refer as $\mathcal{R} \in \mathbb{R}^{m \times n}$. More details on this ratio channel can be found later in this chapter in Section 6.6.2. We convolve this color channel with the set of filters $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{n_f}$ from the filter bank.

$$\mathbf{Y} = (\mathcal{F}_1 * \mathcal{R}(x, y), \dots, \mathcal{F}_{n_f} * \mathcal{R}(x, y)); \forall (x, y) \in \mathbb{R}$$

This convolution operation is executed for all $\bar{q}$ images in the category. Also, the same is replicated across all the $\bar{p}$ categories. We obtain the concatenated matrix by aggregating the filter responses of the images across all categories. This aggregated matrix is the combined filter response; and we denote it by $\mathbf{Z}_\Delta \in \mathbb{R}^{m \times n \times n_f \times \bar{q} \times \bar{p}}$.

6.4.2 Discriminative Model

The filter responses for mn pixels in the stacked filter response $\mathbf{Z}_\Delta \in \mathbb{R}^{m \times n \times n_f \times \bar{q} \times \bar{p}}$ are highly correlated with each other. We perform a general k-means clustering process in this stacked filter response matrix in order to find its subspace representation. The n_t generated cluster centers are referred as the modified-textons in our classification framework. We illustrate this process in Fig. 6.1.

The frequency of occurrence of textons is a distinguishing feature for each category. For an individual image in each category, we compare the filter response at each pixel with every textons from the texton dictionary. Out of the n_t textons, the one having the least 2D Euclidean norm is mapped to the particular pixel. Thereby,

Figure 6.1: Texton generation in our proposed cloud classification approach. Sky/cloud images from individual categories are convolved with filter bank. The generated filter responses are aggregated across the categories. The textons are generated from the k-means cluster of the concatenated filter response.

we generate a texton map from each of the individual images of a category. The histogram generated from the texton map give us an indication of the frequency distribution of the textons in a single image. We average the $\bar{q}$ images in a category to determine the histogram $h_j^{\text{class}}, \hat{j} = 1, 2, \dots, \bar{p}$.

6.4.3 Classification Metric

The testing stage of our classification framework involves the assignment of a category to each of the testing image. We generate the test histogram h_j^{test} using the textons in the texton dictionary. The generated histogram h_j^{test} is compared with each of the $\hat{j}$ model histograms one-by-one using χ^2 statistic. This distance metric is shown in Eq. 6.1.

$$\chi^2 = \frac{1}{2} \sum \frac{(h_j^{\text{class}} - h_j^{\text{test}})^2}{(h_j^{\text{class}} + h_j^{\text{test}})^2} \quad (6.1)$$

This distance metric describes how unlikely a particular distribution is drawn from a given distribution. The category having the least χ^2 distance is the assigned category for the test image. In our case, the number of bins is fixed for training and test histograms. Therefore, we do not suffer from any disadvantages in using this bin-by-bin dissimilarity metric.

6.5 Experimental Evaluations

6.5.1 Evaluation Metric

Let TP , TN , FP , FN denote true positive, true negative, false positive and false negative respectively in a binary classification task. The performance of a general binary classification task is measured using precision and recall. They are described as $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$ respectively.

However, in our task of cloud type recognition, we need an evaluation metric that is applicable for multi-class classification problem. Therefore, we report the following metrics to evaluate the performance: a) Average accuracy, b) Macro- precision and recall, c) Micro- precision and recall.

Average accuracy

One of the most common metrics is accuracy, that represents the ratio of correctly matched classification results. We report the average accuracy value across the $\bar{p}$ categories. It is defined as: $\text{Accuracy} = \frac{1}{\bar{p}} \sum_{\hat{j}=1}^{\bar{p}} \frac{TP_{\hat{j}} + TN_{\hat{j}}}{TP_{\hat{j}} + FP_{\hat{j}} + TN_{\hat{j}} + FN_{\hat{j}}}$.

Macro- averaged values

Macro- averaged values provide an indication of the general performance of the classification framework. It provides equal weight to each categories. Macro- precision and recall for our multi-class classification task are described as $P_{macro} = \frac{1}{\bar{p}} \sum_{\hat{j}=1}^{\bar{p}} \frac{TP_{\hat{j}}}{TP_{\hat{j}} + FP_{\hat{j}}}$ and $R_{macro} = \frac{1}{\bar{p}} \sum_{\hat{j}=1}^{\bar{p}} \frac{TP_{\hat{j}}}{TP_{\hat{j}} + FN_{\hat{j}}}$ respectively.

Micro- averaged values

Micro- average values, on the other hand, provide equal weights to per-image classification performance. Micro- precision and recall for our multi-class classification task are described as $P_{micro} = \frac{\sum_{\hat{j}=1}^{\bar{p}} TP_{\hat{j}}}{\sum_{\hat{j}=1}^{\bar{p}} TP_{\hat{j}} + \sum_{\hat{j}=1}^{\bar{p}} FP_{\hat{j}}}$ and $R_{micro} = \frac{\sum_{\hat{j}=1}^{\bar{p}} TP_{\hat{j}}}{\sum_{\hat{j}=1}^{\bar{p}} TP_{\hat{j}} + \sum_{\hat{j}=1}^{\bar{p}} FN_{\hat{j}}}$ respectively.

These evaluation metrics should be evaluated on a multi-class categorization database, along with the other state-of-the-art algorithms. Unfortunately, most of the existing works in the literature use proprietary images that are not publicly available. In the next section, we will discuss more on this drawback.

6.5.2 SWIMCAT Database

While there has been extensive work on sky/cloud images image analysis captured by ground-based cameras by a number of research groups around the world, public benchmarking databases in this domain are rare. We therefore created our own database for this purpose, which we call **SWIMCAT** (Singapore **W**hole-sky **I**maging **C**ATegories database)¹. This is the first of its kind in the application of automatic cloud classification.

Image acquisition

SWIMCAT contains images captured using WAHRSIS, a calibrated ground-based whole sky imager, which was designed by us [5]. We selected a total of 784 patches

¹The SWIMCAT database is available for download from <http://vintage.winklerbros.net/swimcat.html>.

comprising 5 cloud categories from images that were captured in Singapore over the period January 2013 to May 2014. The 5 categories of clear sky, patterned clouds, thick dark clouds, thick white clouds, and veil clouds are defined based on visual features of sky/cloud conditions, in consultation with experts from the Singapore Meteorological Services. All image patches are of dimension 125×125 pixels. A representative image from each category is shown in Fig. 6.2.

Figure 6.2: (a) Clear sky (b) Patterned clouds (c) Thick dark clouds (d) Thick white clouds (e) Veil clouds

Patch generation

The images captured by WAHRIS are distorted owing to the fish eye lens of its imaging system. We model its fish eye lens using the geometric calibration function. By virtue of this calibration function, we can successfully relate every pixel of the captured image to the corresponding azimuth and elevation angle of the incident ray. We use ray tracing approach to generate undistorted cropped images. The camera is assumed to lie in the center of an hemisphere. We use a constant view angle of 15° to generate output images having a fixed dimension 125×125 , as illustrated in Fig. 6.3. This process is continued for varying sets of azimuth and elevation angles.

This process of generating image patches is performed for several input images. Out of the generated image patches, a few images contain occlusion because of the neighboring buildings and surroundings. These patches are discarded from the database.

Figure 6.3: Process to generate image patches of fixed dimension 125×125 .

6.6 Classification Results

In this section, we will provide a detailed analysis of the existing cloud features on SWIMCAT dataset. Our proposed approach based on texture-based classification is also evaluated on this dataset. We identify the most discriminatory cloud features in this task. However, the multi-class accuracy results using existing features is not high. We, thereby, provide the results of our proposed approach.

6.6.1 Cloud Features

The objectives of extracting features from sky/cloud images is to represent the various classes in the most discriminatory manner. A higher inter-class separation in the feature space representation indicate good candidates for classification.

We extract the features on all the 784 images of the dataset. The features SF, SGLDM_m, SGLDM_r, GLDS, FF, FM, LBP and SIFT are extracted from the gray-scale versions of the SWIMCAT dataset. The HF features are generated from a combination of red and blue channels of the image. These existing features are of different dimensions, and encode the discriminatory cues in the feature space. The dimensions of these feature vectors are high, and it is not possible to visualize them

in their original feature space.

Dimensionality reduction techniques are therefore, extensively used to project the higher dimension vectors into a lower dimension subspace. We use the Principal Component Analysis (PCA) to perform the dimensionality reduction of our feature vectors. PCA aims to transform the original feature space into distinct orthogonal axes, such that these new axes capture the maximum variance of the input data.

Figure 6.4: Each data point represents an image in the SWIMCAT dataset, and each color represents individual classes. The five categories: clear sky, patterned clouds, thick dark clouds, thick white clouds and veil clouds are represented in red, green, blue, cyan and magenta markers.

Figure 6.2 shows the dimensionality reduction results for the different features. The SF feature separates the various classes comparatively well. This is because

the variations in color between sky, dark clouds and white clouds are quite distinct. However, there is an overlap between patterned clouds and white clouds, as their average color of the two classes are quite similar. The SGLDM_m , SGLDM_r features are based on GLCM matrices, and we observe that there is a high correlation between the features of different classes. The patterned clouds stand out in SGLDM_r , as they have a distinctive texture pattern. The GLDS feature is based on GLCM matrix, and we observe the same observation in its representation. The patterned clouds (represented in green) have clear separation with respect to the other cloud classes. The FF perform poorly, as the roughness is not a discriminatory feature cue between the 5 cloud classes. We also observe that, in FM, the variation is observed mainly along the first principal axis. On the other hand, the HF uses a combination of red and blue channels' statistics, and cloud coverage measure to discriminate between the individual classes. The conventional computer vision techniques of LBP and SIFT fail to separate the different cloud classes.

6.6.2 Optimal Parameter Selection

Our proposed texton-based approach is dependent on a number of parameters viz. color channel, set of filters in the filter bank, dimensions of the filters, and the number of textons needed in the clustering process. In this section, we will provide a detailed evaluation of these parameters for our proposed approach.

Color Channel

Based on the results of an earlier analysis of color channels we conducted in Chapter 4, we check the performance of Saturation (S) channel of the HSV color space, various common red-blue combinations (R/B , $R-B$, $\frac{B-R}{B+R}$), and chroma $C = \max(R, G, B) - \min(R, G, B)$. All other parameters are kept fixed (Filter set 3, dimension 49×49 , 30 textons). From Table 6.1, it is observed that most of the color channels work well for sky, patterned clouds and thick white clouds because of their distinctive texture and color. Thick dark clouds have a low saturation component and are therefore correctly classified by color channel S and R/B . The best performing color channel

R/B is selected for our final classification task.

		Sky	Pattern Clouds	Thick Dark Clouds	Thick White Clouds	Veil Clouds	Average	P_{macro}	R_{macro}	P_{micro}	R_{micro}
Channels	S	0.96	1.00	0.82	0.93	0.82	0.91	0.91	0.91	0.87	0.84
	R/B	1.00	1.00	0.98	1.00	0.69	0.93	0.94	0.93	1.00	0.75
	$R - B$	0.93	1.00	0.09	1.00	0.91	0.79	0.79	0.79	0.92	0.89
	$\frac{B-R}{B+R}$	0.76	0.98	0.38	1.00	0.60	0.74	0.78	0.74	0.97	0.65
	C	0.91	1.00	0.38	1.00	0.91	0.84	0.87	0.84	0.95	0.90
Filter sets	Set 1	1.00	1.00	0.49	0.78	0.89	0.83	0.86	0.83	0.86	0.88
	Set 2	0.98	1.00	0.60	0.78	0.89	0.85	0.87	0.85	0.88	0.88
	Set 3	1.00	1.00	0.93	1.00	0.70	0.92	0.93	0.92	0.96	0.76
	Set 4	1.00	1.00	0.53	0.78	0.89	0.84	0.86	0.84	0.88	0.88
	Set 5	0.98	1.00	0.47	0.78	0.89	0.82	0.85	0.82	0.86	0.88
Filter dimension	19×19	0.98	0.93	0.47	0.78	0.91	0.81	0.85	0.81	0.95	0.90
	29×29	1.00	0.78	0.51	0.78	0.67	0.75	0.78	0.75	0.79	0.69
	39×39	1.00	0.98	0.38	0.76	0.82	0.79	0.83	0.79	0.97	0.82
	49×49	1.00	1.00	0.97	1.00	0.68	0.93	0.94	0.93	0.97	0.75
	59×59	0.98	1.00	0.64	0.76	0.89	0.85	0.87	0.85	0.91	0.89
	69×69	0.98	0.84	0.64	0.58	0.87	0.78	0.79	0.78	0.69	0.85
	79×79	0.98	0.71	0.58	0.51	0.84	0.72	0.75	0.72	0.59	0.82
	89×89	0.98	0.71	0.69	0.47	0.87	0.74	0.76	0.74	0.57	0.85
	99×99	1.00	0.69	0.58	0.24	0.87	0.68	0.68	0.68	0.52	0.84
Number of textons	10	1.00	0.44	0.78	0.76	0.47	0.69	0.69	0.69	0.53	0.56
	20	0.98	0.98	0.42	0.8	0.82	0.80	0.84	0.80	0.95	0.82
	30	1.00	1.00	0.97	1.00	0.68	0.93	0.94	0.93	1.00	0.75
	40	1.00	1.00	0.62	0.78	0.91	0.86	0.87	0.86	0.91	0.91
	50	1.00	1.00	0.56	0.80	0.91	0.85	0.87	0.85	0.93	0.91
	60	0.98	1.00	0.58	0.82	0.89	0.85	0.88	0.85	0.91	0.89
	70	0.98	1.00	0.56	0.82	0.89	0.85	0.87	0.85	0.91	0.88
	80	0.98	1.00	0.60	0.80	0.89	0.85	0.87	0.85	0.93	0.89
	90	0.98	1.00	0.58	0.82	0.89	0.85	0.87	0.85	0.93	0.89
	100	0.98	1.00	0.60	0.82	0.91	0.86	0.88	0.86	0.93	0.91

Table 6.1: Performance evaluation of sky/cloud recognition rates for different color channels, filter sets, filter dimensions, and number of textons. The best performing parameters are highlighted.

Filter Set

For our experiments, we use different sets of S-filters that aim to capture the majority of frequency components, and subsequently choose the most suitable filter set. Table 6.2 shows the $(\hat{\sigma}, \hat{\tau})$ pairs for various scenarios, where $\hat{\sigma}$ denotes the scale of the filter in the filter set and $\hat{\tau}$ represents the number of cycles inside the gaussian envelope of the filter set. The individual filters are L_1 -normalized before convolution with the input images.

$(\hat{\sigma}, \hat{\tau})$	Set 1	Set 2	Set 3	Set 4	Set 5
(0.5,0)		✓	✓	✓	✓
(2,1)	✓	✓	✓	✓	✓
(4,1)	✓	✓	✓	✓	✓
(4,2)	✓	✓	✓	✓	✓
(6,1)	✓	✓	✓	✓	✓
(6,2)	✓	✓	✓	✓	✓
(6,3)	✓	✓	✓	✓	✓
(8,1)	✓	✓	✓	✓	✓
(8,2)	✓	✓	✓	✓	✓
(8,3)	✓	✓	✓		✓
(10,1)	✓	✓	✓	✓	
(10,2)	✓	✓	✓	✓	
(10,3)	✓	✓	✓	✓	
(10,4)	✓	✓	✓		
(12,1)			✓		
(12,2)			✓		
(12,3)			✓		
(12,4)			✓		

Table 6.2: The $(\hat{\sigma}, \hat{\tau})$ values for different sets. Set 1 represents the original S-Filter set.

Table 6.1 shows the effect of different filter sets (σ, τ) on the final results. The other parameters are kept constant (Color channel R/B , Filter dimension 49×49 , 30 textons). The DC component in the filter set is required to capture the low frequencies of the image. An absence of the DC component particularly affects the performance for thick dark clouds, as such images contain pre-dominantly low frequency components. The optimal filter Set 3 corresponds to a scenario that contains DC component and higher values of both σ and τ .

Filter Dimension

As the SWIMCAT database consists of images with fixed dimension 125×125 , the optimum filter size needs to be tuned accordingly. As before, we keep the other parameters fixed (Color channel R/B , Filter set 3, 30 textons). From Table 6.1, we see that the average classification accuracy across different categories is poor for small filter sizes, as they fail to capture the predominant essence of the cloud textures. Also, at large filter size, the boundary effects reduce accuracy. The best performance is achieved with an intermediate filter size of 49×49 pixels, which is

thus a good choice for the final classification task.

Number of Textons

Table 6.1 shows the effect of the number of textons on the classification performance. The other parameters are kept fixed (Color channel R/B , Filter Set 3, dimension 49×49). The results are poor when the number of textons is small. However, beyond 30 textons, there is no significant performance gain in the classification accuracy. Thus, we select the number of textons as 30 to avoid overfitting.

6.6.3 Benchmarking Methods

In order to understand the efficacy of these features, we provide an objective evaluation of these benchmarking features. We use these individual features to train a Support Vector Machine (SVM) classifier. The trained model is used in the evaluation of the remaining testing images.

The entire SWIMCAT dataset is divided into two distinct sets - training and testing sets. We choose 40 distinct images from each category in a random fashion, to create the training set. The remaining images from each category are considered as the testing set.

We train individual SVM models for each of the features. The generated multi-class SVM is used to classify the images in the testing set. As described in Section 6.5, we evaluate the accuracy, P_{macro} , R_{macro} , P_{micro} and R_{micro} of the images. The averaged value across all the testing images is reported in Table 6.3.

As expected, the conventional image-based feature extractor viz. LBP and SIFT have a poor classification accuracy. The HF+SVM performs comparatively well with an accuracy of 0.83, as compared to other feature-based techniques. These observations are also evident from the various scatter plots in Fig. 6.4. The confusion matrix generated using HF+SVM is shown in Fig. 6.5(a). The clear sky and patterned clouds are classified quite accurately. However, the thick white and veil clouds are mis-classified with each other. This is also confirmed from Fig. 6.4, that *cyan* and *magenta* overlap with each other. We also benchmark our proposed texton-based

Methods	Accuracy	P_{macro}	R_{macro}	P_{micro}	R_{micro}
SF+SVM	0.53	0.63	0.53	0.85	0.41
SGLDM _m +SVM	0.65	0.67	0.83	0.56	0.67
SGLDM _r +SVM	0.57	0.57	0.57	0.45	0.45
GLDS+SVM	0.50	0.54	0.50	0.63	0.36
FF+SVM	0.31	0.40	0.31	1.00	0.24
MI+SVM	0.64	0.65	0.64	0.53	0.94
HF+SVM	0.83	0.83	0.83	0.85	0.64
LBP+SVM	0.27	0.41	0.27	0.07	0.93
SIFT+SVM	0.31	0.40	0.31	1.00	0.24
Original texton approach	0.78	0.78	0.78	0.70	0.85
Proposed approach (using GRAY color channel)	0.75	0.73	0.75	0.77	0.71
Proposed approach (using R/B ratio channel)	0.95	0.95	0.95	1.00	0.79

Table 6.3: Cloud type classification results of our proposed approach along with other benchmarking algorithms. We use Support Vector Machine (SVM) classifier, along with several feature descriptors. The best performances according to each criterion are depicted in bold.

approach on these images. We obtain the best accuracy of 0.95, amongst all the methods. We use the R/B color channel, because this channel is most appropriate amongst various color spaces and components [104]. The corresponding confusion matrix is shown in Fig. 6.5(b). All classes except veil clouds are classified accurately. Our proposed approach has a limitation that it obtains a low accuracy of 0.76 for veil clouds. This is primarily, because veil clouds are very difficult to detect, and are often mis-interpreted as clear sky.

6.7 Conclusion

In this chapter, we have provided a systematic analysis of the various features that were developed for the task of cloud type recognition. We observed that conventional machine-learning based feature extraction techniques do not generalize well in this task. Amongst the various features, Heinle features perform the best as it exploits the discriminatory cues of red and blue color channels, and also considers the cloud

Figure 6.5: Confusion matrix across 5 categories generated from (a) HF+SVM and (b) Proposed approach.

coverage as one of the parameters. However, the best performance is obtained from our proposed texton-based approach that uses optimized filter bank and color channel for optimal performance. We have presented an effective way to combine both color and texture features for accurate cloud classification results. Future works include the extension of this proposed texton-based approach to other remote sensing applications.

Chapter 7

Weather Parameters Estimation from Images

In the previous chapters, we have described our proposed frameworks for cloud segmentation and cloud-type recognition. In this chapter, we extend our analysis of sky/cloud images to the field of solar and renewable energy. Here, we describe how clouds impact the generation of solar energy; and we propose a model to estimate the solar irradiance using whole sky imagers. We derive a model for estimating the solar irradiance using pictures taken by those imagers. Unlike pyranometers, these sky images contain information about the cloud coverage and can be used to derive cloud movement. An accurate estimation of the solar irradiance using solely those images is thus a first step towards short-term solar energy generation forecasting, as cloud movement can also be derived from them. We derive and validate our model using pyranometers co-located with our whole sky imagers. Our method shows a significant improvement in estimating strong short-term variations. As intended applications, we also discuss our current ongoing works on cloud tracking and detecting rainfall onset from sky images.

7.1 Introduction

Localized and short-term forecasting of cloud movements is an on-going research topic. Optical flow techniques can be used to generate forecasted images using two anterior frames. Our proposed method is thus a first step towards solar irradiance forecasting, as the input data used to estimate the irradiance is the same as the one used to forecast the sky condition.

The accurate estimation of solar energy is a challenging task, as clouds greatly impact the total irradiance received on the earth's surface in an intermittent fashion. In the event of clouds covering the sun for a short time, there is a sharp decline of the produced solar energy. Therefore, it is of utmost importance to model the incoming solar radiation accurately. In this chapter, we ask this fundamental question: in what ways can the rapid fluctuations of the solar irradiance be best captured? We use ground-based sky cameras to estimate the solar irradiance.

The analysis of clouds and several other atmospheric phenomenon is traditionally done using satellite images. The most widely used satellite data is from Moderate-resolution Imaging Spectroradiometer (MODIS), which is on board the Terra and Aqua satellites. They provide a large-scale view of the cloud dynamics and various atmospheric phenomena. The data from this satellite on-board instruments are usually available only twice in a day. This is useful for a macro-analysis of cloud formation at a particular location on the earth's surface. One of the illustrative examples of such satellite data is the HelioClim-1 database from Global Earth Observation System of Systems (GEOSS). It provides hourly and daily average of surface solar radiation received at ground level [105]. Ouarda et al. in [106] assessed the solar irradiance from six thermal channels obtained from SEVIRI instrument. However these information are temporal and spatial averages. Solar energy applications requires knowledge of the solar irradiance at specific locations and at every time through the day. Therefore, images obtained from satellite are not conducive for such analysis, especially in geographical small countries like Singapore where the cloud formation is highly localized. We discussed the advantages of using sky cameras in detail earlier in Chapter 3 of this thesis.

7.2 Outline of Our Contribution

In this chapter, we use images obtained from Whole Sky Imagers (WSIs) to accurately model the fluctuations of the solar radiation ¹. There are several advantages of using a WSI to estimate solar irradiance, instead of using a pyranometer. Common weather stations generally use a solar sensor that measures the total solar irradiance. It is a point-source device providing information for a particular location. We do not have access to the cloud macrophysical properties, and its evolution over time. On the other hand, the wide-angle view of ground-based sky camera provide us extensive information of the sky. We can track a cloud mass over successive image frames, and also predict its future location. In this chapter, we attempt to solve the fundamental problem of modeling solar irradiance from sky images. This will also help in solar energy forecasting, which is useful in photovoltaic (PV) systems [107].

The main contributions of this chapter compared to existing works in the literature include:

- A robust framework to accurately estimate and track the rapid fluctuations of solar irradiance;
- A proposal of a solar irradiance model using ground-based sky camera images, with an extensive benchmarking of our proposed model with other solar irradiance estimation models;
- A methodology to use whole sky images to detect the onset of precipitation; and
- A proposed approach to perform short-term prediction of localized cloud motion using ground-based sky images.

The rest of the chapter is organized as follows. In Section 7.3, we briefly discuss our experimental setup comprising whole sky imagers and weather stations. Our

¹The source code of our proposed methodology to estimate solar irradiance from circumso-lar region of the sky/cloud images is available online at <https://github.com/Soumyabrata/solar-irradiance-estimation>.

proposed model of estimating the solar radiations along with its rapid fluctuations are described in detail in Section 7.4. In Section 7.4.4, we benchmark our proposed methodology with other existing solar estimation models. However, our proposed method suffers from a few disadvantages. We identify those drawbacks and suggest solutions to address them; these are discussed in detail in Section 7.4.5. In this chapter, we also identify two intended applications of using whole sky images in the field of solar energy generation. In Section 7.5, we discuss how the luminance of whole sky images can be used to indicate the onset of precipitation. Finally, in Section 7.6, we discuss our methodology of using optical flow techniques to perform short-term cloud forecasting. Section 7.7 concludes this chapter.

7.3 Data Collection

Our experimental setup consists of weather stations and ground-based WSIs. These devices are collocated at the rooftop of our university building (1.34°N, 103.68°E). These devices continuously capture the various meteorological data, and archive them for subsequent analysis.

The pyranometer reads the instantaneous solar irradiance reaching the sensor at any instant of time. This constitutes both the direct and diffuse components of solar radiation. On a typical day, the solar radiation can fluctuate significantly in a short span of time, which is quite different from the ideal cosine response. If we examine the solar radiation readings on the 7th of December 2015 (cf. Fig. 7.1), we observe that the recorded solar radiation falls drastically from 758 Watt/m² at 10:30, to 283 Watt/m² at 10:32, and then rises again to 714 Watt/m² at 10:34. We observe two rapid fluctuations within a span of a few minutes.

The images captured using Wide Angle High Resolution Sky Imaging System (WAHRIS) at the corresponding time instants are shown in Fig. 7.1. It becomes clear that the fluctuation in solar radiation is due to clouds obscuring the sun from view. Assuming the diffuse component of solar radiation to be almost constant at these three instants of the day, the fluctuations in solar radiation recording are primarily because of the direct irradiance component. Therefore, the clouds in the

Figure 7.1: Consecutive images captured over an interval of 4 minutes on 7 December, 2015. The amount of total solar radiation recorded by the weather station is shown in parentheses. The sun is obscured by clouds at 10:32, resulting in a sudden dip in solar radiation.

vicinity of sun have the highest impact on recorded solar radiation.

In [108], we study the impact of clouds on total solar irradiance reaching the earth’s surface. We define the difference between measured solar irradiance and clear-sky irradiance as the Cloud Radiative Effect (CRE). We analyze the relationship between measured solar irradiance and computed cloud coverage value. Extensive results on a set of 6365 images show that higher cloud coverage leads to larger cloud radiative effect ².

7.3.1 Whole Sky Imager (WSI)

Commercial WSIs are available in the market. However, those imagers have high cost, low image resolution, and less flexibility in operation. In Chapter 3, we have discussed the design of our custom-built, low-cost, and high-resolution sky imagers. We refer these imagers as WAHRIS, that stands for Wide Angle High Resolution Sky Imaging System [5]. Over the years, we have built several versions of WAHRIS [5, 109]. They are now deployed at several rooftops of our university campus, capturing images of the sky at intervals of 2 minutes. We use these images to estimate the total solar

²The source code of our study on cloud radiative effect using sky cameras is available online at <https://github.com/Soumyabrata/cloud-radiative-effect>.

radiation.

7.3.2 Weather Station

In addition to the sky imagers, we have also installed collocated weather stations. We use *Davis Instruments 7440 Weather Vantage Pro* for our recordings. It measures rainfall, total solar radiation, temperature and pressure at intervals of 1 minute. The resolution of the tipping-bucket rain gauge is 0.2 mm/tip.

It also includes a solar pyranometer measuring the total solar irradiance flux density in Watt/m². This consists of both direct and diffused solar irradiance component. The solar sensor integrates the solar irradiance across all angles, and provide the net solar irradiance. On a clear day with no occluding clouds, the solar sensor ideally follows a typical cosine response. The solar sensor reading is highest during noon when the incident angle of sun rays is at the minimum, whilst the reading is low during morning and evening hours.

The solar radiation on a clear sky can be modeled using the solar zenith angle, and earth's eccentricity. Several clear sky models have been developed for various regions. The best clear-sky model for Singapore is provided by Yang et al. [110]. The clear-sky Global Horizontal Irradiance (GHI) G_c is modeled as:

$$G_c = 0.8277 E_k I_{sc} (\cos \psi)^{1.3644} e^{-0.0013 \times (90 - \psi)}, \quad (7.1)$$

where E_k is the eccentricity correction factor for earth, I_{sc} is the solar irradiance constant (1366.1 Watt/m²), and ψ is the solar zenith angle (measured in degrees). The factor E_k is calculated as:

$$E_k = 1.00011 + 0.034221 \cos(\Gamma) + 0.001280 \sin(\Gamma) + \\ 0.000719 \cos(2\Gamma) + 0.000077 \sin(2\Gamma),$$

where $\Gamma = 2\pi(d_n - 1)/365$ is the day angle (measured in radians) and d_n is the day number of the year.

As an illustration, we show the clear-sky radiation for the 1st of September 2016 in Fig. 7.2. The actual solar radiance measured by our weather station is also plotted. We also show the deviation of the measured solar radiation from the clear-sky model. We observe that there are extremely rapid fluctuations in the measured readings. These rapid fluctuations are caused by the incoming clouds that obstruct the sun from direct view. Such information about the cloud profile and its formation cannot be obtained from a point-source solar recording. Therefore, we aim to model these rapid fluctuations in the measured solar radiation from wide-angle images captured by our sky cameras.

Figure 7.2: Solar radiation measurements on the 1st of September 2016. Note the rapid fluctuations of high magnitude in the measured solar radiation recording.

7.4 Modeling Solar Irradiance

This section details our model for computing the solar irradiance from images captured by a whole sky imager. We sample pixels using a cosine weighted hemispheric sampling to simulate the behavior of a pyranometer based on the fisheye camera lens. We then compute the relative luminance using the image capturing parameters, after gamma correction. We finally derive a linear regressor to scale the computed luminance to match measured irradiance values.

7.4.1 Cosine-weighted hemispheric sampling

The behavior of our fisheye lens with focal length $\bar{f}$ is modeled by the equisolid equation $r = 2\bar{f} \sin(\beta/2)$, relating the distance (r) of any pixel from the center of the image to its incident light ray elevation angle (β). This allows to project a captured image on a unit hemisphere, as shown in Fig. 7.3a.

The solar irradiance is composed of a direct component relating the sun light reaching the earth without interference, as well as diffuse and reflected components. Given the high resolution of our images, we consider randomly sampled pixel locations on the hemisphere as input to the luminance computation. We follow a cosine weighted hemispheric distribution function, the center of which is at the location of the sun. This is because clouds in the circumsolar region have the highest impact on the total solar irradiance received on the earth's surface. We provide more emphasis to the clouds around the sun, as compared to those near the horizon.

The first step is to compute the sampled locations from the top of the unit hemispheric dome. Each of the locations are computed as follows, using two random floating points R_1 and R_2 as input, where ($0 \leq R_1, R_2 \leq 1$):

$$\phi = 2\pi R_1, \beta = \arccos(\sqrt{R_2})$$

Figure 7.3: Cosine weighted hemispheric sampling process used to select the pixels used for solar irradiance estimation.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin(\beta) \cdot \cos(\phi) \\ \sin(\beta) \cdot \sin(\phi) \\ \cos(\beta) \end{bmatrix} \quad (7.2)$$

This is represented in Fig. 7.3b.

The second step is to detect the location of the sun using a thresholding method. This is needed to align the center of the previously computed distribution (i.e. top of the hemispheric dome) to the actual sun location in the unit sphere. We choose a threshold of 240 in the red channel R of the RGB captured image, and compute the centroid of the largest area above the threshold ³. We then compute the rotation matrix transforming the z -axis unit vector to the unit vector pointing towards the sky. We apply this rotation to all the sampled points, resulting in Fig. 7.3c.

³We experimentally calculated this threshold and the process is described in [111]

This means that the amount of sampled points in a region of the hemisphere is proportional to the cosine of the angle between the sun direction and the direction to that region. We experimentally concluded that this achieves a good balance between all irradiance components. We thus consider the pixel values of 5000 points sampled using this method as input for the irradiance estimation.

7.4.2 Relative Luminance Calculation

For each of the $\bar{s}$ sampled pixels in the *RGB* image, we compute its brightness value using the following formula ⁴:

$$Y_{\bar{s}} = 0.2126 \cdot R_{\bar{s}} + 0.7152 \cdot G_{\bar{s}} + 0.0722 \cdot B_{\bar{s}}$$

The JPEG compression format encodes images after applying a gamma correction. This non-linearity mimics the behavior of the human eye. This needs to be reversed in order to compute the luminance. In our case, we use a non-linear correction factor of 2.2, which is most commonly used in imaging systems [112]. We thus apply the following formula, assuming pixel values normalized between 0 and 255:

$$Y'_{\bar{s}} = 255(Y_{\bar{s}}/255)^{2.2}$$

Unlike the conventional approach, we perform this non-linear correction stage directly on the brightness values, instead of employing to the individual red-, green- and blue- channels. We then average the pixel values across all the $\bar{s}$ sampled points in the image, and denote it by $\mathcal{N}$. This pixel value given by $\mathcal{N} = (1/t_s) \sum_{i=1}^{t_s} Y'_{\bar{s}}$, denotes the average luminance value of the sampled points from the image, where t_s is the total number of sampled points in the hemispheric sampling process.

However, each image of the sky camera is captured with varying camera parameters viz. ISO, F-number and shutter speed. These camera parameters can be read from the image metadata, and are useful to estimate the scene luminance. The amount of brightness of the sampled points $\mathcal{N}$, is proportional to the number of

⁴following SMPTE Recommended Practice 177

photons hitting the camera sensor. This relationship between scene luminance and pixel brightness [113] can be modeled using the camera parameters as:

$$\mathcal{N} = K_c \left(\frac{e_t \cdot I_s}{f_s^2} \right) \mathcal{L}_s$$

where $\mathcal{N}$ is the pixel value, K_c is a calibration constant, e_t the exposure time in seconds, f_s the aperture number, I_s the ISO sensitivity and $\mathcal{L}_s$ the luminance of the scene.

We can thus compute the relative luminance $\mathcal{L}_r$ using the following:

$$\mathcal{L}_r = \mathcal{N} \left(\frac{f_s^2}{e_t \cdot I_s} \right)$$

7.4.3 Modeling Irradiance from Luminance Values

Using our hemispheric sampling and relative luminance computation, we therefore have one relative luminance value $\mathcal{L}_r$ per image. We propose our model using this relative luminance value to estimate the solar radiation. The usual sunrise time in Singapore is between 6:40 am and 7:05 am, and sunset time is approximately between 6:50 pm and 7:10 pm⁵. Therefore, we consider images captured in the time interval of 7:00 am till 7:00 pm. We use our ground-based whole sky images captured during the time period from January 2016 till August 2016 to model the solar radiation. The solar irradiance is computed as the flux of radiant energy per unit area normal to the direction of flow. The first step in estimating irradiance from the luminance is thus to cosine weight it according to its direction of flow. We weight our measurements according to the solar zenith angle ψ .

The modeled luminance $\mathcal{L}_m$ is expressed as:

$$\mathcal{L}_m = \mathcal{L}_r(\cos \psi)$$

Let us assume that the actual solar radiation recorded by the weather station be $\mathcal{S}$. We check the nearest weather station measurement, for all the images captured by

⁵<http://www.nea.gov.sg/weather-climate/forecasts>

WAHRSIS between April 2016 till December 2016. Figure 7.4 shows the scatter plot between the image luminance and solar radiation. The majority of the data follows a linear relationship between the two. However, it deviates from linearity for higher values of luminance. This is mainly because of the fact that the mapping between scene luminance and obtained pixel value in the camera sensor becomes non-linear for large luminances. A more detailed discussion on this is provided in Section 7.4.5.

Figure 7.4: Model of the solar radiation using the image luminance computed with our proposed framework. We observe that it deviates from linearity at higher luminance values.

We use a linear regressor to model the measured solar radiation $\mathcal{S}$ from the image luminance $\mathcal{L}_m$. This is based on our assumption that the mapping from scene luminance to pixel values in the captured image is linear. Thus, we model solar radiation as: $\mathcal{S} = a \times \mathcal{L}_m + b$. The values of a and b are derived as 0.0138 and -39.896 respectively for our data. Therefore, our proposed model for estimating solar irradiance is:

$$\mathcal{S} = 0.0138 \times \mathcal{L}_m - 39.896. \quad (7.3)$$

This model is derived specifically for equatorial region like Singapore, and the regression constants are based on our WAHRSIS sky imaging system. However, these values need to be fine-tuned while applying our methodology for other regions and different imaging system.

7.4.4 Performance Comparison and Validation

In this section, we evaluate the accuracy of our proposed model. The model is derived based on WAHRSIS images captured from January to August 2016. We also use these images to evaluate the accuracy of our proposed model. Furthermore, we benchmark our algorithm with other existing solar radiation estimation models.

Evaluation

One of the main advantages of our approach is that all rapid fluctuations of solar radiation can be accurately tracked from the image luminance. We illustrate this by providing the measured solar readings of 01-Sep-2016 in Fig. 7.5. The clear-sky model follows a cosine response and is shown in black; whereas the measured solar recordings is shown in red. We normalize our computed luminance in a manner that matches the measured solar readings. We multiply each data points of the luminance with a conversion factor, such that the distance between corresponding inter-samples of luminance and weather station is minimized (cf. Appendix C for details). We observe that our computed luminance from the whole-sky image and the measured solar radiation closely follows each other. We emphasize here that it is an important contribution to successfully track the rapid solar fluctuations. Unlike other solar estimation models based on meteorological sensor data, our proposed model can successfully estimate the *peaks* and *troughs* of solar readings accurately.

Using our proposed methodology, we compute the luminance of all the captured

Figure 7.5: We show the measured weather station data (in red), and the clear sky radiation (in black) as on 01-Sep-2016. We observe that the image luminance normalized w.r.t. the measured solar radiation, follows the measured readings closely. The sampling interval between two measurements is 2 minutes.

images. We use Eq. 7.3, and estimate the corresponding solar radiation values. The actual ones are recorded in the collocated weather station. These recordings serve as the ground-truth measurements. Figure 7.6 shows the histogram of difference between the estimated and actual solar radiation. We observe that the estimated solar radiation do not deviate much from the actual solar radiation. It is clear that 47.9% of data points are concentrated in the range $[-100, +100]$ Watt/m².

Figure 7.6: Histogram of difference between estimated and actual solar radiation. We observe that most of the data are concentrated in the 0-bin.

Benchmarking Techniques

We benchmark our proposed approach with other existing solar estimation models. To the best of our knowledge, currently, there are no proposed models to estimate short-term fluctuations of solar irradiance from ground-based images. However, most remote sensing analysts have been using other meteorological sensor data eg. daily temperature, humidity, rainfall and dew point temperature to estimate daily solar irradiance. One of the pioneer work was done by Hargreaves and Samani [43], who proposed a model based on daily temperature variations. Donatelli and Campbell [44] improved the model by including clear sky transitivity as one of the factors. On the other hand, Bristow and Campbell [45] also proposed a new model of solar radiation estimation, by including the atmospheric transmission coefficient. Subsequently, Hunt et al. [46] showed that the solar estimation model can be further improved by incorporating the daily precipitation data in the model. We benchmark our proposed approach with these different existing models.

We calculate the Root Mean Square Error (RMSE) of the estimated solar radiation and Spearman's rank correlation coefficient as the evaluation metrics. The RMSE of an estimation algorithm represents the standard deviation of the actual and estimated solar radiation values. Table 7.1 shows the RMSE values of our proposed algorithm with the other existing benchmarking algorithms. Our proposed approach performs the best. We also evaluate the spearman correlation coefficients of the different benchmarking algorithms, since this is a non-parametric measure to find the relationship between measured and estimated solar radiation. This does not assume that the underlying dataset are derived from a normal distribution. We report the correlation values in Table 7.1. Our proposed approach has also the highest correlation amongst all methods.

Table 7.1 explains the results where the training and testing set of images are identical, and all images are considered for evaluation. We are also interested to check if our proposed model can generalize well with random samples of our captured sky camera images. We choose a random selection of images as the training set, and fit our linear regressor on these selected training images. The RMSE values

Methods	RMSE (Watt/m ²)	Correlation
Proposed approach	178.27	0.86
Hargreaves and Samani	982.35	0.67
Bristow and Campbell	318.07	0.68
Donatelli and Campbell	324.48	0.67
Hunt et al.	922.66	0.65

Table 7.1: Benchmarking of our proposed approach with other solar radiation estimation models. All correlation values have p-value equal to 0.

are then calculated on these training images. We perform this analysis for varying percentage of training images. Each experiment is performed 100 times to remove any selection bias. Figure 7.7(a) shows the results on training images. We observe that the variation of the RMSE values gradually decreases, as we increase the number of training images. Moreover, we also check the variation of RMSE values when the testing images are not identical as training images. Once we choose a random selection of images as training set, the remaining images are considered as the testing set. We show the RMSE results on such images in Fig 7.7(b). As expected, the variation of RMSE values increases with higher percentage of training images. The linear regressor model overfits the data, and provides higher variation in the error when tested on a fewer testing images. However, the average RMSE does not vary much in all cases. Therefore, our proposed model is free from selection bias, and generalizes well with random selection of training and testing images.

As a final comparison, we illustrate the various benchmarking models in Fig. 7.8. Unfortunately, most of these algorithms fail to capture the short-term variations of the actual solar radiation.

We represent the scatter plot between the measured solar radiation and estimated solar radiation for the different benchmarking algorithms in Fig. 7.9. We observe that there is no strong correlation for most of these existing algorithms. This is because meteorological sensor data alone, with no cloud information cannot determine the sharp fluctuations of the solar radiation. This is an important limitation of these models, which we attempt to address in this chapter. Our model based on sky images have additional information about cloud movement and its evolution, which is the fundamental reason behind rapid solar radiation fluctuations. In our proposed model,

Figure 7.7: Effect of the percentage of training images on RMSE values. The lower and upper end of each box represents the 25th and 75th percentile of the data, and the red line represents the median value. Each experiment is conducted 100 times with a random choice of training and testing sets.

most of these short-term variations are captured properly as shown in Fig. 7.5.

7.4.5 Discussion

Our proposed approach can estimate the solar radiation accurately with the least root mean square error, as compared to other models. The main advantage of our approach is that it can be used on predicted images as well, opening the potential for short term solar irradiance forecasting, which is needed in the solar energy field.

Figure 7.8: Comparison amongst different benchmarking solar estimation models, along with clear sky model and measured solar radiation on 01-Sep-2016. We observe most of the existing algorithms fail to capture the rapid fluctuations of the measured solar radiation.

Figure 7.9: Scatter plot between measured solar radiation and estimated solar radiation for the benchmarking algorithms.

However, it suffers from a few drawbacks. In this section, we highlight them and suggest techniques to address them. Firstly, we use *JPEG* images instead of uncompressed *RAW* images for the computation of scene luminance. The *JPEG* compression algorithm introduces non-linearities in the pixel values and our proposed model thus deviates from a linear relationship. In Fig. 7.4, we have demonstrated the obtained relationship between actual solar radiation and the estimated image luminance from our sky camera images. We observe that the relationship is linear for low luminance values (or low solar radiation). However, it deviates from non-linearity at higher luminance values (or higher solar radiation). We can generate more consistent results by using only *RAW* format images. Nevertheless, we still use *JPEG* images, as *JPEG* images have a significantly smaller size and less perceptible

distortion in image quality, as compared to other lossy compression techniques. We capture images at an interval of 2 minutes – this amounts to an incredibly huge dataset of RAW images. Therefore, we stick to capture images in JPEG mode because of operational use. This assumption is practical from an operational point of view. On the other hand, uncompressed *RAW* images have large file size and it is impossible to capture and store *RAW* images at short capturing intervals due to the induced latency.

Secondly, our captured images have a wide range of camera settings with varying shutter speed, ISO and aperture values. This is disadvantageous because the relationship between pixel value and camera aperture value becomes non-linear for larger F-numbers. The relationship deviates from linearity above F 4.0 [113]. Figure 7.10 depicts the wide range of F-numbers in the captured images used in deriving our proposed model. We observe that a significant percentage of images have large F-numbers, where the non-linearity sets in. In our future work, we intend to use an integrating sphere to calibrate our camera to varying levels of scene brightness.

Figure 7.10: Distribution of F-number of the WAHRIS images that are used to derive the proposed model.

Hence, we are able to estimate the sharp short-term variations of solar radiation,

which significantly improves the state-of-the-art. This approach is of interest in the solar energy field, because these variations cause a sudden decrease in the electricity generation from solar panels. Short-term predictions of such ramp-downs are needed to maintain the stability of the power grid. Combining our solar irradiance estimation approach with cloud movement tracking in the input images could ultimately lead to better irradiance predictions. This approach of using whole sky imagers in the field of solar radiation has other applications as well.

A correct estimation of solar radiation from sky images can also assist us in devising an alternative manner to detect the onset of precipitation. This is because, *dark clouds* (clouds with low luminance value) generally occur before a rainfall event, and is a good indication of rainfall onset. We discuss this in detail in the subsequent Section 7.5.

7.5 Detecting Rainfall Onset from Sky Cameras

In this section, we discuss how sky images can be used effectively for the detection of rainfall onset. These images contain additional information about cloud coverage and movement and are therefore useful for accurate rainfall nowcast. We validate our results using rain gauge measurement recordings and achieve an accuracy of 89% for correct detection of rainfall onset.

7.5.1 Impact of Precipitation

We observed that the luminance of the image in the circumsolar region (region around the sun) and the measured solar radiation from the weather station are correlated. Therefore, we can define the empirical luminance of the clear sky from the theoretical clear-sky model (cf. Eq. 7.3), based on the relationship between solar irradiance and image luminance. We define this estimated luminance of a cropped image around the sun as the *clear-sky luminance*, denoted by $\mathcal{L}_c$. In other words, it measures the luminous intensity per second around the sun observed by our camera sensor, in the event of a clear sky.

In addition to clear-sky luminance, we also calculate the luminance of the whole sky image denoted by $\mathcal{L}_m$. We crop a square of dimension 2000×2000 pixels from the image to exclude the occlusions and neighboring buildings. The average luminance of the cropped image is $\mathcal{L}_m$.

Figure 7.11 shows the luminance value of the images captured across different times of the day for a typical day in December. We also calculate the clear-sky luminance and plot it in Fig. 7.11.

Figure 7.11: We illustrate the measured luminance and clear-sky luminance for the 11th of December 2015. The primary y-axis shows the normalized luminance value, and the secondary y-axis measures the rainfall rate (measured in mm/hour). We also show the corresponding sky images captured by our sky camera at three distinct times – before, during and after the rain event.

We observe that the measured luminance value sharply decreases with the onset of rainfall. The luminance continues to remain low during the entire duration of the rainfall. It gradually increases after the precipitation event. We verify this by referring to the corresponding sky images captured by our WSI at different time instants. The sun is visible in the first image captured at 12:54 (before the rain event), when we record a normalized luminance of 0.7. The second image captured at 14:07 (during the rain event) has dark clouds, with a much lower normalized luminance of 0.02. Finally, the third image captured at 16:00 (after the rain event)

shows a clearer sky condition, and the normalized luminance is 0.58. Therefore, we observe that the onset of precipitation reduces the luminance measured in the sky images compared to the clear-sky luminance.

7.5.2 Methodology & Results

In this section, we describe our methodology to detect precipitation from sky camera images. The measured luminance value deviates from the ideal luminance value (corresponding to clear sky) during a rainfall event. We use this deviation from clear sky luminance as a measure to indicate rainfall onset.

We define Clearness Luminance Index (I) as the ratio of measured luminance ($\mathcal{L}_m$) value to the clear-sky luminance ($\mathcal{L}_c$) value, $I = \mathcal{L}_m / \mathcal{L}_c$.

In case of overcast condition, the obtained Clearness Luminance Index I is low. On the other hand, I equals unity during ideal clear-sky cases. We evaluate the values of I for a longer period of time, namely the entire month of December 2015. Figure 7.12 shows the relationship between index I of the captured images and its corresponding distance from the nearest rain event (measured in minutes).

From Fig. 7.12, we observe that the index value gradually decreases as it approaches a rain event, and reaches a minimum during rainfall. Post rain event, the clearness luminance index gradually increases with the passage of time. The trend roughly follows a V-shaped curve.

Critical Clearness Luminance Index

In this section, we define a threshold in I to detect the onset of rainfall as the Critical Clearness Luminance Index (I_c). The criterion for rainfall detection proposed here is as follows: if the measured index I decreases below the critical I_c , there is an indication of precipitation.

We consider all the images captured in December 2015. We use rainfall recorded by the collocated weather station as ground truth measurements. We classify each of the weather station measurements into *rain* and *non-rain* events, based on the rain gauge measurements. We also check the corresponding WSI images captured at the

Figure 7.12: Trend of Clearness Luminance Index (I) before and after precipitation measured for all days in December 2015. The value of I gradually decreases as it approaches rainfall onset, and then gradually increases again.

same time, and calculate their index I . The entire set of images in our dataset are segregated into two categories. The first category comprises those images that are within ± 15 minutes of a rain event, including the images captured during the rain event. The rest of the images are considered in the second category.

Figure 7.13 plots the Cumulative Distribution Plot (CDF) of clearness luminance index of images within and outside the time window of ± 15 minutes.

The motivation of classifying the images into two categories (within and outside ± 15 minutes interval) is to analyze the impact of precipitation on the image luminance $\mathcal{L}_m$. From Fig. 7.13, we observe that images within ± 15 minutes interval of rain event have a much lower luminance index than those images further from a rain event. This is intuitive as dark cumulus clouds appear before a rainfall, and they often persist for a brief period after the rainfall event. Furthermore, dark clouds are generally not present on non-rainy days.

Figure 7.13: CDF plots of Clearness Luminance Index of images within and outside ± 15 minutes of rain event. Each point in the graph denotes the percentage of time the clearness index I is less than its corresponding value.

Operating Characteristics (OC) Curve

In this section, we are interested in determining the critical clearness luminance index (I_c). For this purpose, we calculate the operating characteristics (OC), which shows the percentage of time for images within ± 15 minutes of rain event against those outside ± 15 minutes of rain event. We vary the value of I_c from 0.01 to 0.2, in steps of 0.01. The resulting OC curve is shown in Fig. 7.14.

We observe a sharp transition at $I_c = 0.08$. This is generally referred as the *elbow* of the OC curve, and is chosen as the operating point as it has the best performance. We check the corresponding percentage of time for this critical threshold (I_c) of 0.08. We observe that the images within ± 15 minutes of rain events have a clearness index below the critical threshold for 89.41% of time. On the other hand, only 13.13% of time images outside the time window have a clearness index below the critical

Figure 7.14: Operating Characteristics (OC) curve generated by varying the critical clearness luminance index (I_c). The *elbow* in the plot (shown in red) is our chosen I_c .

threshold. Therefore, $I_c = 0.08$ can be used as the threshold for the clearness index for estimating the onset of rainfall. However, this threshold is derived for a tropical region like Singapore. This value may need to be adjusted when applying this model to other regions or using different sky imagers.

7.5.3 Summary

In this section, we have discussed how ground-based sky cameras can be effectively used to detect rainfall onset. We defined a Clearness Luminance Index that measures the deviation of estimated luminance from the clear-sky luminance. We have proposed a critical clearness luminance index that indicates the onset of precipitation. Extensive results on a month of sky images and weather station recordings show the efficacy of our proposed approach. In the future, we plan to verify these results over longer periods.

7.6 Short-term Cloud Tracking Using Sky Images

In addition to detecting rainfall onset, we also use sky images to perform short-term forecasting of cloud images. This is useful for solar energy forecasting. We use two successive frames to compute optical flow and predict the future location of clouds. We achieve good prediction accuracy for a lead time of up to 5 minutes.

7.6.1 Methodology

In this section, we first give a general formulation of the optical flow technique and then describe how it is applied to our problem.

Optical Flow Formulation

Optical flow is based on the brightness constancy constraint, which states that pixels of an image sequence do not change value, but only shift position over time. Let's define a pixel intensity at image coordinates (x, y) and time t_m by $\mathbf{X}(x, y, t_m)$. Under this condition, this intensity would have moved by $(\Delta x, \Delta y)$ after a time Δt_m :

$$\mathbf{X}(x, y, t_m) = \mathbf{X}(x + \Delta x, y + \Delta y, t_m + \Delta t_m).$$

Using a first order Taylor series expansion, the optical flow equations can be derived [114]:

$$\frac{\partial \mathbf{X}}{\partial x} u_x + \frac{\partial \mathbf{X}}{\partial y} u_y + \frac{\partial \mathbf{X}}{\partial t_m} = 0,$$

where (u_x, u_y) is the optical flow (or velocity), and $\frac{\partial \mathbf{X}}{\partial x}$, $\frac{\partial \mathbf{X}}{\partial y}$ and $\frac{\partial \mathbf{X}}{\partial t_m}$ are the derivatives of the image in the x , y , and t_m dimensions.

This equation cannot be solved analytically, as it has two unknowns. This is known as the *aperture problem*. In order to solve it, two main approaches exist:

- Local methods state that the optical flow vectors are constant within some neighborhood and thus increase the number of equations to solve for the same optical flow vector. A typical example is the Lucas-Kanade method [115].

- Global methods assume that the optical flow vector distribution should be smooth across the spatial and temporal axes. They minimize a global energy function in order to reduce large optical flow gradients. A famous algorithm is the Horn-Schunck method [116].

We use the implementation ⁶ from [117], which is based on a combination of both local and global approaches, following the method proposed in [118].

Cloud Tracking

We use two image frames taken at times $t_m - 2$ minutes and t_m , and compute the translational vectors in both x and y direction of the images, using the method mentioned above. We rely on the assumption that clouds do not significantly change between image frames. The optical flow vectors estimate the direction and orientation of the moving clouds, assuming an affine transformation between two frames. The algorithm provides a dense result, i.e. a velocity vector is associated to every pixel coordinate in the input image.

Figure 7.15: Ratio channels of two successive image frames.

Since we are interested in tracking the detailed cloud shape, it is important to use a color space which provide a good separation of clouds and sky. We use a variant

⁶Available at <https://people.csail.mit.edu/celiu/OpticalFlow/>

of the ratio of red and blue color channels of the image. In an earlier chapter, we have analyzed various color channels and concluded that $(B - R)/(B + R)$ is the most discriminatory color channel, where B and R indicate the blue and red color channels respectively.

As an illustration, we show the ratio channel of two successive image frames in Fig. 7.15. We observe that there is a clear contrast in this channel. These ratio channels are used to estimate the flow field of clouds.

Using the images in Fig. 7.15, we obtain the horizontal and vertical translations of each of the pixels, which can be combined into a vector field. We show them in Fig. 7.16, plotted in terms of pixels/minute. This provides an idea about the relative speed and direction of each of the pixels between the two frames.

Figure 7.16: Horizontal and vertical translation of pixels between Frame 1 and Frame 2. The color map represents the speed in pixels/minute units.

Cloud Motion Prediction

We can now use the above information to predict future frames. The vector fields are applied individually to each of the red, green, and blue channels of the image.

Clouds have an ill-defined shape, and can change their shape and size very quickly. However, even though the underlying assumptions do not exactly match this reality, they work well to track clouds for short lead times, as is the case in many other

Figure 7.17: Prediction of sky/cloud image on 16 April 2015 with a lead time of 2 and 4 minutes. The prediction accuracy is 83.44% for a lead time of 2 minutes, and 72.37% for a lead time of 4 minutes. The binary images are generated using our sky/cloud segmentation algorithm described in Chapter 5.

optical flow problems.

For higher lead times, we use the actual frame at time t_m and predicted frame at time $t_m + 2$ minutes to predict the frame at time $t_m + 4$ minutes. Similarly, we use the predicted frame at time $t_m + 2$ minutes and the predicted frame at $t_m + 4$ minutes, to predict the frame at time $t_m + 6$ minutes, and so on. At every stage, we use the previous two frames (actual or predicted), to compute the subsequent frame. This works well under the assumption that the clouds do not *significantly* change their shape and location for the given lead time.

7.6.2 Results & Discussions

We now evaluate the forecasting accuracy of our methodology. For this purpose, we compute the binary sky/cloud image of the forecasted image using our cloud detection algorithm as described in Chapter 5. We then compare it with the binary image computed from the original image. The accuracy is then calculated as the percentage of correctly classified pixels (sky or cloud) in the predicted binary image, as compared to the actual binary image.

We compute the prediction accuracy of our cloud tracking algorithm for a typical day in April 2015 and present our results for different lead times. Figure 7.18 shows the performance of our cloud tracking algorithm for different lead times. We observe that prediction accuracy is good for short lead times, but gradually decreases as lead times become larger. This makes sense as clouds generally move quite fast, and can change shape between image frames. Furthermore, the error obtained in the intermediate forecast image is cascaded to future images.

Figure 7.18: Comparison of prediction accuracy percentage with different lead times.

We provide a few illustrative examples of our prediction accuracy upto 4 minutes. Figure 7.17 shows the output of our algorithm with a lead time of upto 4 minutes, in

intervals of 2 minutes. We use the frame at time $t_m - 2$ minutes and the frame at time t as the input images of our proposed approach to predict the frame at time $t_m + 2$ minutes. We show the actual and predicted images along with their corresponding binary images. The accuracy achieved with our algorithm is 83.44%. We then use the frame at t_m minutes along with the predicted image at $t_m + 2$ minutes to predict the frame at $t_m + 4$ minutes. The accuracy is still good (around 72.37%) given the higher lead time. However, we observe for higher lead times, the predicted image gets progressively more distorted as compared to its actual image, and artifacts appear. This happens because the error incurred in the previous lead times gets cascaded to future frames, as clouds significantly changes in shape over a relatively short period of time. After 10 minutes, the clouds have moved significantly, and there is little correlation between frames. These predictions are only based on images capturing a small area of the sky, and better long-term predictions would only possible at a larger scale and a lower level of detail.

7.6.3 Summary

In this section, we have discussed about our methodology to track cloud movement across successive image frames from sky cameras. It is based on optical flow and performs well for lead times of a few minutes. The accuracy gradually decreases for larger lead times. Our proposed approach is intended mainly for the short-term prediction of cloud movements, as we perform a localized analysis of cloud motion. We also show how the luminance of images captured by our sky camera can be used to detect the onset of precipitation. In our future work, we plan to use other meteorological data, such as wind sensors, to further increase the prediction accuracy.

7.7 Conclusion

In this chapter, we presented a method for estimating solar irradiance using the luminance of images taken by a whole sky imager. We are able to estimate the sharp short-term variations, which significantly improves the state-of-the-art. This ap-

proach is of interest in the solar energy field, because these variations cause a sudden decrease in the electricity generation from solar panels. Short-term predictions of such ramp-downs are needed to maintain the stability of the power grid. Combining our solar irradiance estimation approach with cloud movement tracking in the input images could ultimately lead to better irradiance predictions.

Chapter 8

3D Scene Analysis Using Multiple Cameras

In the earlier chapters, we used a single ground-based sky camera for an extensive analysis of the earth’s atmosphere. We used our custom-built sky cameras to perform segmentation, classification, estimation and prediction of meteorological parameters etc. In this chapter, we analyze the localization error of *cloud mass* in a setup consisting of multiple whole sky imagers. To better understand the localization error in a more general imaging context, we use simulated camera models in our study. We assume our cameras have a finite pixel resolution, which are kept in a multi-camera setup. Using concepts drawn from computational geometry, we provide bounds on the localization of a point in a multi-camera setup. In the later part of this chapter, we show how multi cameras (i.e. several whole sky imagers) can help in accurate estimation of cloud base height.

Point localization in multi-camera setups ¹ has been widely studied in computer vision. In the first part of this chapter, we explain our work on consistent and optimal point localization algorithm, for a multi-camera setup. We formulate this point localization task as a gradient-ascent optimization function, for maximizing

¹A part of the work presented in this chapter was done during my doctoral internship in Audio-visual Communications (LCAV) Laboratory, École Polytechnique Fédérale de Lausanne, in collaboration with Prof. Martin Vetterli, Dr. Adam Scholefield and Alireza Ghasemi.

the objective function under computational geometric constraints. Experimental results verify the efficacy of our approach as compared to the current state-of-the-art localization algorithms. In the later part of this chapter, we show how point localization of cloud mass is performed in a network of sky cameras.

8.1 Outline of Our Contribution

The main contributions of this chapter compared to existing works in the literature include:

- We propose a new localization algorithm in a multi-camera setup, where the camera poses are noisy and feature matching are noiseless; and
- We describe a method to localize a cloud mass using a pair of sky cameras.

The rest of the chapter is organized as follows. In Section 8.2, we introduce the problem of localization in multi-camera setup. Section 8.3 discusses the setup and introduces the various notations that will be used throughout this chapter. We describe a recently introduced point localization algorithm for noiseless camera poses in Section 8.4. We propose a modified algorithm for noisy camera poses in Section 8.5. The detailed computation of our proposed algorithm is described in Section 8.6 and its experimental results are presented in Section 8.7. The remaining part of this chapter illustrates an application of point localization in the context of cloud imaging. Section 8.8 discusses this. Finally, Section 8.9 concludes this chapter.

8.2 Introduction of Localization Problem

The classical *Art Gallery Problem* is a widely known problem in the computer vision community. We illustrate this problem in Fig. 8.1. The objective of this problem is to *guard* an art gallery room (either convex or non-convex type) from an intruder, denoted by $\mathbf{p}$. Every point (*intruder*) $\mathbf{p}$ inside polygon is guarded by set of cameras

Figure 8.1: Guarding a non-convex shape art gallery from an intruder $\mathbf{P}$.

(guards) $\mathbf{S}$ iff $\forall q \in \mathbf{S}$ s.t. $\mathbf{pq} \ni \text{polygon}$. This problem has been widely studied in the literature for different room types.

An interesting extension of this art gallery problem is the *Point Localization Problem*. This involves localizing the *intruder*, in addition to guarding the *art gallery* from various intruders. Considering cameras with infinite resolution, we need *at least* two cameras to localize the intruder. We illustrate this in Fig. 8.2. The intruder $\mathbf{P}$ can be localized by the two cameras, as shown in the figure if the intruder do not lie in the line joining the two cameras. If the intruder is *seen* by the two cameras, the exact position and orientation of the intruder can be estimated via triangulation. However, in practice, we do not have infinite resolution cameras. Therefore, the intruder can no longer be localized via triangulation in the line of sight of the camera with the highest accuracy.

In the case of finite-resolution cameras, the intruder can therefore be localized in a *region of interest*, instead of a particular point. This region of interest is generated by the intersecting half-planes of the activated pixels ² in the two cameras. We illustrate this scenario in Fig. 8.3.

²Activated pixel denotes the corresponding pixel in the finite-resolution camera that contains the corresponding feature-point of the localized object.

Figure 8.2: Localizing an intruder $\mathbf{P}$ in *any* room type using infinite resolution cameras.

From Fig. 8.3, we also observe that the consistent region for localizing the intruder progressively reduces with the addition of new cameras, assuming that the intruder can be captured in the new cameras as well. The new consistent region is estimated by the progressive intersection of the half-spaces.

8.3 Problem Set-up

In this section, we will formally introduce the problem setup and also explain the various notations. These notations will be used throughout this chapter and will be consistent throughout this chapter, unless specified otherwise.

For the sake of simplicity, we consider the case of localizing a point $\mathcal{I}$ in a 2-dimensional search space, $\hat{\mathcal{S}} \in \mathbb{R}^2$. The analysis can be easily extended to 3-dimensional search space as well. Let us consider that we are interested to localize the point $\mathcal{I}$, as shown in Fig. 8.4. The figure describes the acquisition system with one finite-resolution camera. The camera location is denoted by $(\hat{x}^1, \hat{y}^1)$, and its view angle as Θ^1 . Therefore, the camera pose is referred as $(\hat{x}^1, \hat{y}^1, \Theta^1)$, which collectively defines the camera location and the view angle. All these measurements are denoted with respect to any global co-ordinate system. In the illustration, we denote the finite-resolution camera to have 4 equal-width pixels, with 5 corresponding light

Figure 8.3: Region of interest is an *area* instead of a particular point, when we deal with finite resolution pixel camera.

rays³. All these rays have viewing angles of $\theta_1, \theta_2, \dots, \theta_5$, with respect to the X-axis. Without any loss of generality, we consider the corresponding feature point of point $\mathcal{I}$ is observed in the fourth pixel of the camera. This pixel is referred as the *active pixel*, throughout this chapter. The half-spaces corresponding to the active pixel make angles θ_4 and θ_5 respectively with the X-axis.

This problem set-up can also be extended to that with multiple cameras. We illustrate this in Fig. 8.5. The illustration shows a set-up consisting of two cameras. Following a similar naming convention, the second camera has the pose $(\hat{x}^2, \hat{y}^2, \Theta^2)$. Also, the corresponding feature of the point $\mathcal{I}$ is observed in the first pixel of the camera.

This set-up can be therefore generalized for $\tilde{n}$ cameras, with random camera poses. The pose of the $\tilde{n}$ -th camera is denoted by $(\hat{x}^{\tilde{n}}, \hat{y}^{\tilde{n}}, \Theta^{\tilde{n}})$.

³We consider a pin-hole camera model

Figure 8.4: Acquisition system with one camera, mentioning the various camera pose notations.

For subsequent calculations, we also define *offset angle* of all the rays with respect to its corresponding view angles. These offset angles are the difference of angles between the light rays and the camera view angle ⁴. Mathematically speaking, it can be defined as follows:

$$\theta_1 = \delta_1 + \Theta$$

$$\theta_2 = \delta_2 + \Theta$$

$$\dots$$

$$\theta_5 = \delta_5 + \Theta$$

For the sake of brevity, we will drop the index $\hat{i}$ for the $\hat{i}$ -th camera in most of our discussions.

⁴Camera View Angle can also be defined as the angle made by the central pixel with the global X-axis.

Figure 8.5: Acquisition system with multiple cameras, mentioning the various camera pose notations.

8.4 SHARP: A Consistent and Optimal Localization Algorithm

For cameras having noiseless camera poses and noiseless feature matching, a recent point localization algorithm called **SHARP** was proposed by Ghasemi et al. [119]. **SHARP** stands for **S**equential **H**alf space **A**ggregation for **R**econstructing a **P**oint. It consists in the generation of half spaces from the light rays corresponding to the active pixel. Subsequently, the convex polygon is generated with these intersection of half spaces. This process is continued till all the cameras in the set-up are added in the system. These half-space intersections generate a closed convex polygon. Assuming that the point is equally likely to be present inside the polygon, SHARP returns the centroid of the polygon as the optimal estimate. We have illustrated this concept in Fig. 8.6.

The current state-of-the-art localization algorithms consists in minimizing the re-projection error of the point to the center of the active pixel, under different constraints. We explain this by illustrating the simplest case of point reconstruction with single camera, as shown in Fig. 8.4.

The estimate $\mathbf{p}^*$, for the actual point $\mathbf{p}$ is calculated as an error minimization

Figure 8.6: Illustration of half-space intersections to generate a closed convex polygon in a multiple-camera setup.

problem. Let us suppose the pixel center is denoted by $\mathbf{q}$, and the re-projection function to the image plane is defined as $\Phi(\mathbf{p}')$. The estimate is calculated as shown in Eq. 8.1, under different norm constraints (viz. $\ell_1, \ell_2, \ell_\infty$).

$$\mathbf{p}^* = \arg \min_{\mathbf{p}'} \|\mathbf{q} - \Phi(\mathbf{p}')\|_{\ell_1, \ell_2, \ell_\infty} \quad (8.1)$$

8.5 SHARP with Noisy Camera Poses

In the case of noiseless camera poses, half spaces corresponding to the active pixels of the different cameras form a consistent region. We show this in Fig.8.7. Three cameras having finite-resolution pixels are stationed in $\mathbb{R}^2$ with random camera poses. With noiseless camera poses, the probability of localizing the point is unity anywhere inside the consistent region (cf. Fig. 8.7(b)). SHARP returns the centroid of this region as the optimal estimate.

However, in the event of noisy camera poses, a consistent region can no longer

Figure 8.7: A consistent region is formed when the camera poses are noiseless.

be computed. We illustrate this in Fig. 8.8. In this chapter, we propose a modified SHARP algorithm that localizes the point with high accuracy in the event of noisy camera poses.

Figure 8.8: Consistent region can no longer be formed with noisy camera poses.

8.5.1 Motivation & Insights

We explain the insights behind our proposed approach with a simple example. Consider a single camera (CAM 1) with 4 equal-width pixels in $\mathbb{R}^2$, as shown in Fig. 8.9.

We are interested to localize the point P , shown in *red*. The fourth pixel of the camera is the corresponding active pixel.

Figure 8.9: Illustration of a single camera with finite pixel resolutions.

Considering that the feature-point matching is correct, the point P can lie anywhere between the two half spaces defined by the active pixels. Therefore, we obtain a uniform probability of unity for the distribution of active region, as shown in Fig. 8.10(b). This holds true for noiseless camera poses.

However, it no longer holds for noisy camera poses. As an example, we consider noise only in view angle of the camera. We illustrate this in Fig. 8.11, where the view angle of the camera CAM 1 is noisy. In this case, the probability of point P (which is outside the region enclosed between two half-spaces) to lie in active region is *non-zero*. Assuming a gaussian noise model, we are interested to compute the distribution of active region in $\mathbb{R}^2$ for this illustration.

We assume the view angle follows a gaussian distribution with mean value $\bar{\Theta}$ and variance $\sigma_{\bar{\Theta}}^2$.

$$\Theta \sim \mathcal{N}(\bar{\Theta}, \sigma_{\bar{\Theta}}^2) \quad (8.2)$$

Figure 8.10: Distribution of active region with noiseless camera view angle.

We are interested to compute the two limits for which point $\mathbf{p}$ (which makes an angle θ with X-axis) fall in the active region. We illustrate the case of upper limit in Fig. 8.12. Point $\mathbf{p}$ can be inside active region when CAM 1 is rotated in the clock-wise direction by a specific amount as shown in Fig. 8.12.

We calculate this angle as:

$$\begin{aligned}
 \theta'_4 &= \theta_4 - (\theta_4 - \theta) \\
 \implies \theta'_4 &= \theta \\
 \implies \Theta + \delta_4 &= \theta \\
 \implies \Theta &= \theta - \delta_4
 \end{aligned} \tag{8.3}$$

Therefore, camera view angle ($= \Theta$) equal to $\theta - \delta_4$ corresponds to the upper limit of the view angle for which the point $\mathbf{p}$ lie inside the active region.

Similarly, we compute the lower bound. We illustrate it in Fig. 8.13. In this case, CAM 1 needs to be rotated further such that the new position of θ_5 coincides with θ .

Figure 8.11: Single camera with noisy view angle. There exists a finite non-zero probability for the point P to lie in active region.

Figure 8.12: Computation of upper bound on the view angle.

We calculate this angle as:

$$\begin{aligned}
 \theta'_4 &= \theta_4 - (\theta_5 - \theta) \\
 \implies \theta'_4 &= \theta_4 - \theta_5 + \theta \\
 \implies \Theta + \delta_4 &= \theta_4 - \theta_5 + \theta \\
 \implies \Theta &= \theta_4 - \theta_5 + \theta - \delta_4
 \end{aligned} \tag{8.4}$$

Therefore, camera view angle ($= \Theta$) equal to $\theta_4 - \theta_5 + \theta - \delta_4$ corresponds to the lower limit of the view angle for which the point P lie inside the active region.

Figure 8.13: Computation of lower bound on the view angle.

For a particular noise level in camera view angle (fixed variance σ_{Θ}^2), we can thus calculate the probability of each point in $\mathbb{R}^2$ to fall in active region. We show the distribution of active region in Fig. 8.14.

Figure 8.14: Distribution of active region with noise only in view angle.

Assuming gaussian noise, this distribution is generated by calculating the normal cumulative probability defined over appropriate *lower* and *upper* bounds. We will formally define our proposed algorithm in Section 8.6.

8.5.2 Geometric Constraints

In the illustration in Fig. 8.11, we explain the scenario where noise is present only in view angle. However, noise can also be present in the camera positions (x- and y-coordinate in $\mathbb{R}^2$ plane). In this section, we will compute the geometrical limits for three distinct cases: noise in view angle, X- and Y- coordinate of the camera.

It should be noted here that the geometrical limits depend on the position of point P in $\mathbb{R}^2$. We divide this search space into 6 distinct scenarios, as depicted in Fig. 8.15.

Figure 8.15: Depicting 6 distinct scenarios in $\mathbb{R}^2$, when the noise is present in camera locations.

Noise in view angle

We calculate the bounds in the view angle, assuming that the camera locations are noiseless.

Upper Bound

We show the computation for Scenario I.

Figure 8.16: Determination of upper bound of view angle for the rotation of camera (Scenario I).

$$\begin{aligned}
 \theta'_5 &= \theta_5 + (\theta - \theta_4) \\
 \Theta + \delta_5 &= \theta_5 + (\theta - \theta_4) \\
 \implies \Theta &= \theta_5 + (\theta - \theta_4) - \delta_5
 \end{aligned} \tag{8.5}$$

Lower Bound

$$\begin{aligned}
 \theta'_4 &= \theta_4 - (\theta_5 - \theta) \\
 \Theta + \delta_4 &= \theta_4 - (\theta_5 - \theta) \\
 \implies \Theta &= \theta_4 - \theta_5 + \theta - \delta_4
 \end{aligned} \tag{8.6}$$

Therefore, the lower and upper geometrical limits are $+(\theta_4 - \theta_5 + \theta - \delta_4)$ and

Figure 8.17: Determination of lower bound of view angle for the rotation of camera (Scenario I).

$+(\theta_5 + (\theta - \theta_4) - \delta_5)$ ⁵. Also note that these bounds are restricted to all points lying in Scenario I⁶.

8.6 Proposed Algorithm

In this section, we will define our proposed localization algorithm.

8.6.1 Noise Model

We assume that the poses for the different cameras in the multi-camera system assume a gaussian distribution. Let us assume that $\Theta^{\hat{i}}$ is the view angle of $\hat{i}$ -th camera. And $\Delta \hat{x}_c^{\hat{i}}$ and $\Delta \hat{y}_c^{\hat{i}}$ are the translations of the $\hat{i}$ -th camera in X- and Y-directions respectively. Without any loss of generality, we will drop the suffix $\hat{i}$ in our discussions.

We assume that the view angle Θ follows a normal distribution with mean $\bar{\Theta}$ and variance σ_{Θ}^2 .

⁵Please note the positive and negative conventions. Rotations in the clockwise direction are positive, while the counter-clockwise rotations are referred as negative.

⁶Calculation related to Scenario II-VI are provided in Appendix D.

$$\Theta \sim \mathcal{N}(\bar{\Theta}, \sigma_{\Theta}^2) \quad (8.7)$$

Similarly, $\Delta \hat{x}_c$ and $\Delta \hat{y}_c$ follow a gaussian distribution with mean 0 and variance $\sigma_{\Delta \hat{x}_c}^2$ and $\sigma_{\Delta \hat{y}_c}^2$ respectively.

$$\begin{aligned} \Delta \hat{x}_c &\sim \mathcal{N}(0, \sigma_{\Delta \hat{x}_c}^2) \\ \Delta \hat{y}_c &\sim \mathcal{N}(0, \sigma_{\Delta \hat{y}_c}^2) \end{aligned} \quad (8.8)$$

Considering independent and identically distributed random variables, noise in each of the three components: view angle, X- and Y- coordinate can be treated separately.

However, we combine them in a single model by defining a multivariate gaussian distribution. We define this distribution for any position $(\hat{x}, \hat{y})$ in $\mathbb{R}^2$ as:

$$\Psi(\hat{x}, \hat{y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (8.9)$$

, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix. They are given by:

$$\boldsymbol{\mu} = [\mu_{\Theta}, \mu_{\Delta \hat{x}}, \mu_{\Delta \hat{y}}] \quad (8.10)$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{\Theta}^2 & \sigma_{\Theta} \sigma_{\Delta \hat{x}_c} & \sigma_{\Theta} \sigma_{\Delta \hat{y}_c} \\ \sigma_{\Delta \hat{x}_c} \sigma_{\Theta} & \sigma_{\Delta \hat{x}_c}^2 & \sigma_{\Delta \hat{x}_c} \sigma_{\Delta \hat{y}_c} \\ \sigma_{\Delta \hat{y}_c} \sigma_{\Theta} & \sigma_{\Delta \hat{y}_c} \sigma_{\Delta \hat{x}_c} & \sigma_{\Delta \hat{y}_c}^2 \end{pmatrix} \quad (8.11)$$

8.6.2 Proposed Approach

Our proposed approach returns the point in the search space $\mathbb{R}^2$ having the highest value. We observe the noisy view angle, and set it to μ_Θ as shown in Eq. 8.10. The other parameters $\mu_{\Delta\hat{x}}$ and $\mu_{\Delta\hat{y}}$ are set to 0. We compute the **multivariate normal cumulative probability** evaluated over the *4-dimensional hypercube* defined with *appropriate* lower and upper limits, for all the cameras in the system. We compute the product of these values for all the cameras, and return the point having the highest measure.

We outline our algorithm in Algorithm 2.

Algorithm 2 Point localization algorithm with noisy camera poses

Require: Set of parameters for noise in view angle, X- and Y- component i.e. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

Ensure: Estimated point position in $\mathbb{R}^2$.

- 1: **loop** For all points in search space $\mathbb{R}^2$
 - 2: **loop** For all $\hat{i}$ cameras in multi-camera system
 - 3: Observe a noisy realization for a camera in multi-camera system.
 - 4: Set μ_Θ as the noisy view angle for the camera.
 - 5: Rotate the half-spaces such that these half-spaces lie in 1st quadrant, as shown in Fig. 8.15.
 - 6: Compute the **multivariate normal cumulative probability** evaluated over the *4-dimensional hypercube* defined with *appropriate* lower and upper limits, denoted by $p^{\hat{i}}$.
 - 7: **end loop**
 - 8: **end loop**
 - 9: Compute the product of all the multivariate normal cumulative probability values, i.e. $p = p^1 * p^2 * \dots * p^{\hat{i}}$.
 - 10: **return** point in $\mathbb{R}^2$ with maximum value, i.e. $(\hat{x}, \hat{y}) = \underset{\hat{x}, \hat{y}}{\operatorname{argmax}} p$.
-

8.6.3 Computation of Proposed Approach

In this section, we will describe the process of computing the multivariate normal cumulative probability values. We define the geometrical bounds $\forall (\hat{x}, \hat{y}) \in \mathbb{R}^2$ as

$\hat{a}(\hat{x}, \hat{y})$ and $\hat{b}(\hat{x}, \hat{y})$. Let us assume that the multivariate normal distribution for view angle, X- and Y- component be $f(\Theta)$, $f(\Delta\hat{x})$ and $f(\Delta\hat{y})$. We define the multivariate normal cumulative probability for a single camera as $F(\hat{x}, \hat{y})$.

For a single camera, we define as,

$$F(\hat{x}, \hat{y}) = \int_{\hat{a}(\hat{x}, \hat{y})}^{\hat{b}(\hat{x}, \hat{y})} f(\Theta, \Delta\hat{x}, \Delta\hat{y}) d\Theta d\Delta\hat{x} d\Delta\hat{y} \forall \hat{x}, \hat{y} \in \hat{\mathcal{S}},$$

where it is integrated over the geometrical bounds $\hat{a}(\hat{x}, \hat{y})$ and $\hat{b}(\hat{x}, \hat{y})$ for all the components.

Assuming that noises are i.i.d. over these three components, we can re-define $F(\hat{x}, \hat{y})$ as follows:

$$F(\hat{x}, \hat{y}) = \int_{\hat{a}(\hat{x}, \hat{y})}^{\hat{b}(\hat{x}, \hat{y})} f(\Theta) d\Theta * \int_{\hat{a}(\hat{x}, \hat{y})}^{\hat{b}(\hat{x}, \hat{y})} f(\Delta\hat{x}) d\Delta\hat{x} * \int_{\hat{a}(\hat{x}, \hat{y})}^{\hat{b}(\hat{x}, \hat{y})} f(\Delta\hat{y}) d\Delta\hat{y}$$

$$F(\hat{x}, \hat{y}) = F_{\Theta}(\hat{x}, \hat{y}) * F_{\Delta\hat{x}}(\hat{x}, \hat{y}) * F_{\Delta\hat{y}}(\hat{x}, \hat{y})$$

We are interested to compute $\text{argmax} F(\hat{x}, \hat{y})$, and localize the point $(\hat{x}, \hat{y})$ in $\mathbb{R}^2$. We employ a simple gradient ascent algorithm for the same; we state it in Algorithm 3. We define a learning rate η , and compute the gradient $F'(\mathbf{t})$ ⁷. Thus, each point $\mathbf{t} \in \mathbb{R}^2$ is updated accordingly.

Our objective function $F(\hat{x}, \hat{y})$ has no closed form expression; however an exact expression exists for $F'(\hat{x}, \hat{y})$.

Let us now show the computation of gradient for view angle. We know,

$$F_{\Theta}(\hat{x}, \hat{y}) = \int_{\hat{a}(\hat{x}, \hat{y})}^{\hat{b}(\hat{x}, \hat{y})} f_{\Theta}(\hat{x}, \hat{y}) d\hat{x} d\hat{y}, \forall \hat{x}, \hat{y} \in \hat{\mathcal{S}},$$

We define the gradient $F'_{\Theta}(\hat{x}, \hat{y})$ over $\hat{x}$ and $\hat{y}$ as

$$F'_{\Theta}(\hat{x}, \hat{y}) = \left[\frac{\partial F_{\Theta}(\hat{x}, \hat{y})}{\partial \hat{x}} \quad \frac{\partial F_{\Theta}(\hat{x}, \hat{y})}{\partial \hat{y}} \right]$$

⁷ $F'(\mathbf{t})$ has different forms depending on the location of point $\mathbf{t}$ in search space $\hat{\mathcal{S}}$.

Algorithm 3 Gradient ascent algorithm for the computation of $F(\hat{x}, \hat{y})$.

Require: Noisy camera parameters for all the $\hat{i}$ cameras.

Ensure: Estimated point location $\mathcal{I}, (\hat{x}, \hat{y}) \in \mathbb{R}^2$

- 1: Choose an initial point $\mathbf{t} \in \mathbb{R}^2$, in search space $\hat{\mathcal{S}}$, and learning rate η
- 2: **loop** Until *convergence* is achieved.
- 3: Compute $F'(\mathbf{t})$ at point $\mathbf{t}$.
- 4: Update new location,

$$\mathbf{t}^u := \mathbf{t} + \eta F'(\mathbf{t})$$

5: **end loop**

6: Converged optimal point, $\mathbf{t}^* = \mathbf{t}^u$.

7: **return** point location $\mathcal{I}$, as indicated by $\mathbf{t}^*$.

The partial derivatives are computed using Leibniz integral rule.

$$\frac{\partial F_{\Theta}(\hat{x}, \hat{y})}{\partial \hat{x}} = f_{\Theta}(\hat{b}(\hat{x}, \hat{y})) \frac{\partial f_{\Theta}(\hat{b}(\hat{x}, \hat{y}))}{\partial \hat{x}} - f_{\Theta}(\hat{a}(\hat{x}, \hat{y})) \frac{\partial f_{\Theta}(\hat{a}(\hat{x}, \hat{y}))}{\partial \hat{x}}$$

$$\frac{\partial F_{\Theta}(\hat{x}, \hat{y})}{\partial \hat{y}} = f_{\Theta}(\hat{b}(\hat{x}, \hat{y})) \frac{\partial f_{\Theta}(\hat{b}(\hat{x}, \hat{y}))}{\partial \hat{y}} - f_{\Theta}(\hat{a}(\hat{x}, \hat{y})) \frac{\partial f_{\Theta}(\hat{a}(\hat{x}, \hat{y}))}{\partial \hat{y}}$$

We compute these partial derivatives separately for all 6 different scenarios. Let us check for Scenario 1.

For scenario 1, the lower- and upper- bounds are given by $\hat{a}(\hat{x}, \hat{y}) = +(\theta_4 - \theta_5 + \theta - \delta_4)$ and $\hat{b}(\hat{x}, \hat{y}) = +(\theta_5 + (\theta - \theta_4) - \delta_5)$ respectively. All quantities except θ are constant, and then their corresponding derivatives are zero. The quantity θ can be expressed in terms of $\hat{x}$ and $\hat{y}$, as follows:

$$\theta = \tan^{-1} \frac{(\hat{y} - \hat{y}_c)}{(\hat{x} - \hat{x}_c)}$$

Therefore, $\frac{\partial f_{\Theta}}{\partial \hat{x}}$ and $\frac{\partial f_{\Theta}}{\partial \hat{y}}$ are given by

$$\frac{\partial f_{\Theta}}{\partial \hat{x}} = \frac{\hat{y}_c - \hat{y}}{(\hat{x} - \hat{x}_c)^2 + (\hat{y} - \hat{y}_c)^2}$$

$$\frac{\partial f_{\Theta}}{\partial \hat{y}} = \frac{(\hat{x} - \hat{x}_c)^2}{(\hat{x} - \hat{x}_c)^2 + (\hat{y} - \hat{y}_c)^2}$$

Similarly, the gradient over the remaining two X- and Y- components $F'_{\Delta\hat{x}}(\hat{x}, \hat{y})$ and $F'_{\Delta\hat{y}}(\hat{x}, \hat{y})$ can be defined accordingly.

In a general setting having multiple cameras in the setup, our objective function takes the form as,

$$G(\hat{x}, \hat{y}) = F_1(\hat{x}, \hat{y})F_2(\hat{x}, \hat{y}) \dots F_{\tilde{n}}(\hat{x}, \hat{y})$$

Similarly, the gradient $F'(\hat{x}, \hat{y})$ can be generalized to $\tilde{n}$ cameras as follows:

$$\begin{aligned} G'(\hat{x}, \hat{y}) = & F'_1(\hat{x}, \hat{y})F_2(\hat{x}, \hat{y}) \dots F_{\tilde{n}}(\hat{x}, \hat{y}) \\ & + F_1(\hat{x}, \hat{y})F'_2(\hat{x}, \hat{y}) \dots F_{\tilde{n}}(\hat{x}, \hat{y}) \\ & + \dots \\ & + F_1(\hat{x}, \hat{y})F_2(\hat{x}, \hat{y}) \dots F'_{\tilde{n}}(\hat{x}, \hat{y}) \end{aligned}$$

8.7 Experimental Evaluations

In order to assess the accuracy of the modified SHARP algorithm, we conduct extensive experiments. We compare the re-projection error of the proposed algorithm with the other state-of-the-art algorithms. We have briefly described the various benchmarking algorithms in Section 8.4.

We consider a three-camera setup for our experimental evaluations. The scenario is depicted in Fig. 8.18. Each camera has a random position and view-angle in the 2-dimensional search space, $\hat{\mathcal{S}} \in \mathbb{R}^2$. Suppose that the actual position of the localized point is denoted by *red* dot. We are interested to localize this point with highest accuracy in case of noisy camera poses.

In case of noiseless scenario, a consistent and convex polygon is generated by the intersection of corresponding half-spaces. Considering a uniform distribution in the probability of localizing the point, the centroid of the convex polygon provides the optimal estimate. However, a consistent region can no longer be computed in case of noisy camera poses.

In order to understand the effect of noise in the re-projection error, we experiment

Figure 8.18: Experimental set-up with three cameras having random camera poses. The three cameras are represented as CAM 1, CAM 2 and CAM 3 respectively. The actual location of the localized point is denoted in *red* dot.

for various scenarios.

8.7.1 Effect of Noise in View Angle

We vary the noise in the view angle, by keeping the noise in camera locations fixed. We set the camera location noise as $\sigma_{\Delta\hat{x}} = 0.1$ and $\sigma_{\Delta\hat{y}} = 0.1$. We should note that noise is present in all the three cameras. Using the modified SHARP algorithm, we compute the location having the highest probability in $\mathbb{R}^2$. We perform this experiment for 100 different trials, and report the average re-projection error. We show the results in Fig. 8.19. Clearly, the proposed algorithm works best as compared to ℓ_1 -, ℓ_2 - and ℓ_{inf} - norm minimization techniques. Moreover, with increasing noise, the performance of the other algorithms continues to degrade. Our proposed approach has a stable performance at higher noise levels too.

8.7.2 Effect of Noise in X-component of Camera Positions

We also analyze the effect of noise in X- component of camera position on the average re-projection error. We vary this noise, by keeping the noise in the other components

Figure 8.19: Effect on re-projection error with varying noise in view angle.

fixed. We set the noise in view angle as $\sigma_{\Theta} = 1$ and in the Y-component as $\sigma_{\Delta\hat{y}} = 0.1$. We calculate the average re-projection error for varying amount of noise in X-component. Each score is averaged over 100 different experiment trials. Figure 8.20 reports the results. It is clear that our proposed approach works best as compared to other algorithms.

8.7.3 Effect of Noise in Y-component of Camera Positions

We perform similar experiments for varying amount of noise in Y-component of camera positions for the 3 cameras. We set the noise in other components fixed: $\sigma_{\Theta} = 1$ and $\sigma_{\Delta\hat{x}} = 0.1$, and vary the noise in Y-component of camera locations. We perform 100 experiments for a single noise setting. Figure 8.21 shows the results for varying amount of noise. It is clear that our proposed approach works best as compared to the benchmark methods.

In this section, we have proposed a point localization algorithm with noisy camera poses. This has been done in 2-D under arbitrary camera poses with finite resolutions and perfect feature matching. We formalize this as a gradient ascent algorithm, and derive closed-form expressions for the objective function. Extensive experimental

Figure 8.20: Effect on re-projection error with varying noise in X co-ordinate of the camera positions.

results prove the efficacy of our proposed approach as compared to other state-of-the-art point localization algorithms. In the subsequent section, we will discuss such localization task in the context of cloud imaging. We present our current work on localizing a cloud mass using a stereo-setup of sky cameras.

8.8 Application to Ground-based Sky Cameras

In this section, we present an illustration of localization algorithm in a network of multiple ground-based sky cameras. We present a framework for cloud base height estimation using two such imagers. Our method is based on stereoscopic scene flow. We demonstrate the feasibility of our approach and use computer-generated images with controlled cloud height to validate the accuracy of our method.

Localizing clouds in 3D space is a challenging task. Clouds usually do not exhibit precise boundaries and may not have distinctive textures, which makes matching two images difficult. However, clouds evolve slowly, and successive frames look similar. We present a method to estimate the cloud base height from a succession of images captured by ground-based imagers with a hemispherical view of the sky. We use

Figure 8.21: Effect on re-projection error with varying noise in Y co-ordinate of the camera positions.

stereoscopic scene flow, which is defined as the three-dimensional motion field of points in the world [59]. This allows us to retrieve the 3D location as well as the motion of the clouds, by incorporating past frames in the computation. Although we focus on the 3D location (i.e. cloud base height) in this work, the motion information can be used as input for cloud movement prediction.

8.8.1 Workflow

This section describes the workflow of our proposed approach. It takes as input a sequence of images taken simultaneously by two sky imagers, such as those shown in Fig. 8.22. They were taken by a pair of our custom-designed whole sky imagers [5, 109], which we call Wide-Angle High-Resolution Sky Imaging System (WAHRISIS). More details of these imaging system were already discussed earlier in Chapter 3 of the thesis. The two imagers are placed approximately 95 meters apart on rooftops of the Nanyang Technological University in Singapore.

Figure 8.22: Images captured by the pair of sky imagers.

Undistortion

The first step consists of removing the distortion due to the fish-eye lens in the images. The projection behavior of the lens used in our imagers follows the equisolid angle mapping function, $r = 2\bar{f}\sin(\beta/2)$, where r is the distance from the image center, β is the elevation angle of the incident light ray, and $\bar{f}$ is the focal length. More details about different projection functions are described in Chapter 3. We thus can retrieve the incident light ray of every pixel in the image. The position along that ray is the value which needs to be computed by triangulation to find the position of the cloud in 3D. Fig. 8.23 shows the original image and the pixel values projected on a hemisphere.

Figure 8.23: Illustration of the undistortion process of an image taken with our fish-eye lens (bottom). The color value of each incident light ray is projected on a hemisphere (middle). The image plane (top) is computed as the value of each light ray going through the pixel at its location (as indicated at the corners of the image).

We use a ray-tracing approach to generate undistorted images. Considering the camera at the origin, we place a virtual plane at a user-defined height. We then compute the pixel values of the image by looking for the incident light ray that intersects the plane at the pixel location, as shown in Fig. 8.23. There is usually no ray going through an exact pixel location, so we interpolate using the nearest rays.

In order to map this image to real world coordinates, we give each pixel a width of one meter. We then define the height of the plane to be at 400 meters, which translates to a viewing angle of 103 degrees.

Rectification

We use the real world units defined above to place the image plane of the second imager with respect to the first one. In practice, the positions of the imagers can be retrieved by Global Positioning System (GPS). Stereo vision algorithms typically compute the disparity between the left and right images along one axis only in order to reduce the complexity. For this, the input images need to lie on a common image plane. The process of computing such images is known as rectification.

We incorporate this step into the undistortion process described above. We apply 3D rotations on the virtual image plane before computing the pixel values, as shown in Fig. 8.24. Two rotations are needed. The first one is applied around the z axis and relates the difference of longitude and latitude between the imagers, quantified with the azimuth angle ω_1 . The second rotation is applied around the rotated y axis and relates the difference in altitude between the imagers, which is quantified with the elevation angle ω_2 .

Figure 8.25 shows both pairs of undistorted images before and after rectification. Notice how the disparity between the images is only present along the horizontal axis in Fig. 8.25(b).

Disparity Map

The next step is to compute the disparity map between the sequence of pairs of undistorted, rectified images. We use the three-dimensional scene flow algorithm

Figure 8.24: Illustration of the rectification process. The two coloured dots represent the two imagers with their respective image planes. The disparity between the images only appears along one axis after rectification.

Figure 8.25: Image pair before and after rectification.

proposed by Cech et al. [62], which jointly estimates a disparity map and optical flows between stereo image frames.

The strength of the scene flow approach is the inclusion of past frames into the computation of the disparity map. While clouds do not have many distinct features to track, they evolve slowly. Successive frames taken at appropriate intervals indeed do not exhibit large differences, and past frames can thus complement the information provided in the current frame.

Figure 8.26 shows the disparity map computed from the pair of images shown in Fig. 8.25(b). Each pixel value of the disparity map indicates the shift to add to the horizontal image coordinate to obtain the position of the pixel capturing the same physical point in the other image. Note that the algorithm does not define values

over the whole image, as some parts do not exhibit sufficient texture or do not look similar in the two images due to the different viewpoints.

Figure 8.26: Disparity map for the image pair in Fig. 8.25(b).

3D Point Cloud

The last step consists of mapping the disparity map to real world 3D coordinates. For every defined value of the disparity map, we compute the position of that point on both rectified images in the image plane, as shown by the red and blue dots in Fig. 8.27. We then consider the two rays going through those points and the device position. Those two rays intersect at the real world location of the captured point, shown in green in Fig. 8.27.

Figure 8.28 shows the resulting point cloud for an image pair from the sequence. We distinguish two different clusters: the first is at an altitude of about 700 meters and corresponds to the cumulus cloud at the bottom right of the input image, while the second represents the higher cloud layer appearing at around 3000 meters.

8.8.2 Validation

Since it is very difficult to measure the actual 3D location and shape of clouds, we use computer-generated images with a user-defined cloud bottom altitude for validating

Figure 8.27: Illustration of the computation of a point in 3D.

Figure 8.28: 3D point cloud computed for a pair of images from the sequence.

our approach. We use *Blender*,⁸ an open-source 3D computer graphics program, for creating images using a procedural cloud shader and a model of the fish-eye lens of WAHRISIS. While being simpler than the reality, these images do have the advantage that the positions of the imagers and clouds can be user-defined.

For this experiment, we created a sequence of 21 frames with a cloud moving over the two cameras. The average cloud bottom altitude is 500 meters, and the distance between the cameras is 400 meters. Fig. 8.29 shows an example of such an image pair.

Figure 8.30 shows the resulting 3D point cloud for an image pair from the computer-generated sequence. It matches with the ground truth created by Blender.

⁸ <http://www.blender.org/>

Figure 8.29: Image pair from the computer-generated sequence.

Figure 8.30: 3D point cloud for an image pair computer from the computer-generated sequence. The bounding boxes in which Blender generates the cloud are shown in green. The two black dots indicate the position of the imagers.

Averaging over the entire sequence, 82.4% of the reconstructed 3D points lie inside the bounding box in which the cloud is generated.

In this section, we have discussed how 3D scene flow techniques can be used to estimate cloud base height in a multiple-camera setting. We have introduced a workflow to a common scene flow algorithm in this context, which we validated with computer generated images.

8.9 Conclusion

In this chapter, we have proposed a point localization algorithm in a multiple-camera setting, under the assumption of error-free feature matching and noisy camera poses. Our modified SHARP algorithm has the best performance as compared to the other

benchmarking algorithms. In the later part of this chapter, we demonstrate a practical example of point localization task—localizing cloud mass in the 3D space using a pair of ground-based sky cameras. Our theoretical analysis on point localization in multi-camera setup serves us an attempt to understand the accuracy of the cloud base height estimation. In practice, point localization in the 3D space is a challenging task, as it depends on a number of important factors: error-free feature matching and tracking, accurate estimation of camera poses and dense feature correspondence between stereo images. The task gets more challenging in cloud base height estimation, as clouds are generally featureless and feature matching between two images gets more challenging.

Chapter 9

Conclusion & Future Work

9.1 Conclusion

In this thesis, we have discussed the wide range of our contributions in remote sensing for varied applications. We mainly focused our attention on the analysis of clouds in satellite communication links. As Singapore is a geographically small land region, conventional techniques using satellite images are not conducive for study. The images obtained from satellites suffer from poor spatial and temporal resolutions. Therefore, we have presented a new paradigm in understanding the earth’s atmosphere—using ground-based sky cameras to better analyze clouds for different applications. These cameras, popularly known as Whole Sky Imagers (WSIs) offer a compelling alternative. In this thesis, we have presented several models of our custom-built sky cameras. Our designed sky camera models have lower cost, better image resolution, and higher flexibility as compared to the commercially available sky cameras.

The sky cameras capture the sky/cloud scene at regular intervals of time, and archives the captured images for further analysis. One of the fundamental problems in cloud imaging is the *successful* detection of cloud pixels in these images. This is a non-trivial task, as clouds are non-rigid in shape, and do not conform to any particular shape or size. The *color* of clouds is generally used as the most discrimi-

natory feature. In this thesis, we have provided a systematic analysis of the different color channels that are frequently used for cloud recognition. We have identified the *best* color channels that can be used for subsequent analysis. Furthermore, the conventional sky/cloud image segmentation algorithms are generally binary in nature, i.e. each pixel is assigned either *sky* or *cloud* label. In this thesis, we have proposed a probabilistic cloud segmentation approach, wherein each pixel is assigned a probability to be categorized in cloud category.

For the purpose of regular weather monitoring, the various meteorological institutes around the world regularly report cloud types in its meteorological reports. Such classifications are generally done manually, which is of course, cumbersome and time-consuming. In this thesis, we attempt to provide a complete framework for a multi-class classification of cloud types. Our framework has the best classification performance as compared to the other benchmarking algorithms. We systematically integrate the *color* and *texture* cues of cloud for better performance.

In addition to ground-based sky cameras, different meteorological sensors continually record the various weather parameters of the earth's atmosphere. However, these devices are point-measurement devices, i.e. these devices only record measurements for a particular point location of the earth's atmosphere. Sky cameras equipped with wide-angle lenses, on the other hand, provide remote sensing analysts better insights on the evolution of clouds over a period of time. In this thesis, we have analyzed the information from sky cameras and other sensors to provide deeper insights in solar energy generation. We have also used sky cameras to predict future location of clouds and to detect the onset of precipitation. These applications greatly help in the field of solar and renewable energy forecasting.

Finally, we discussed our work on estimating the base height of a cloud mass, using a pair of (or multiple) sky cameras. We used stereoscopic scene flow techniques to estimate cloud-base height, and verified its accuracy using computer-generated images. We have also provided deeper insights in the localization performance of an *intruder*¹ in a multi-camera setup with noisy camera poses.

¹This is in reference to the famous art-gallery problem as described in Chapter 8.

9.2 Future Work

This thesis explored varied aspects of cloud imaging using ground-based sky cameras. While it solved most of the fundamental problems, it also helped us in generating future research proposals. In this section, we will briefly mention some of the intended future works.

In this thesis, we described the design of three models of ground-based sky cameras. These imagers consists of a Digital Single-Lens Reflex (DSLR) camera equipped with a fish-eye lens. We are now interested to design a miniature model using a regular smart-phone. With the rapid development in the imaging system of hand-phones, it seems a plausible alternative to perform our task of cloud imaging using smart-phones. The smart-phone application will be involved in capturing images at regular intervals of time, and will be sent to the server for archival and further processing.

The conventional approach consists in segmenting the cloud image into binary labels—sky and cloud. In this thesis, we have provided a probabilistic segmentation approach wherein we employ a *soft* thresholding approach. In the future work, we aim to generate the ground-truth database with probabilistic labels. We will perform the manual labeling via crowd-sourcing experiments, wherein several cloud experts will manually label the pixels. This will help us to assign a probability ground-truth map for the sky/cloud images.

In this thesis, we deal primarily with daytime images captured by our sky camera. However, nighttime cloud imaging is also essential in certain applications, such as continuous weather analysis and satellite communication. Nighttime sky/cloud images are darker and noisier, and thus harder to analyze. Currently, we use a superpixel-based segmentation technique ² to detect cloud in nighttime images. In the future, we plan to extend our cloud categorization and cloud localization for nighttime images as well.

In addition to these visible-light imaging techniques, we are also interested to explore further in Near-Infra-Red (NIR) techniques. Such imaging techniques are

²The source code of our nighttime cloud segmentation is available online at <https://github.com/Soumyabrata/nighttime-imaging>.

very important in our current application and has immense potential in capturing sharper and clearer images of sky/cloud hemisphere. Our current work on cloud detection and its classification can be extended to these near-infrared cloud images for better results. Fredembach and Süssstrunk demonstrated in their recent work [120] that the haze disappears in near-infrared capture of an outdoor scene as compared to an image captured by a conventional RGB image. This is because of a physical phenomenon called Rayleigh scattering. Small atmospheric particles present in the air scatter incident light with varying degree. Because of Rayleigh scattering, the component of light having the least wavelength gets scattered the most; and thus provides a bluish color to the sky. However, near-infrared is less scattered as compared to its visible counterparts and renders the sky darker. Clouds however do not change its behavior of light scattering because of its bigger particle size which is referred as Mie scattering.

Figure 9.1: On the left, a conventional RGB image. On the right, a near-IR capture of the same scene. The haze has disappeared, revealing a sharper, cleaner, picture [120].

On top of ground-based imaging, satellite images also provide us interesting perspectives of the upper layer of cloud. In this thesis, our focus was not intended on satellite images as its spatial and temporal resolution were not conducive for our application of cloud attenuation and solar energy forecasting. These high resolution requirements are very strong in satellite imaging. No mission is generally designed for monitoring of small areas, during airborne sensing. However, satellite such as MODIS can offer us interesting insights on the cloud top phenomenon, at

the expense of lower resolutions. MODIS stands for Moderate-Resolution Imaging Spectroradiometer. It is a 36 spectral band imaging instrument embedded in two satellites: Terra and Aqua. Its medium resolution (250 to 1000m) allows a large swath, and thus a high revisit time. For the region of Singapore, the two satellites acquire images twice a day. It is a good compromise for our applications, and a good choice for this comparison. The National Aeronautics and Space Administration (NASA) organizes the MODIS data in levels, collections and products. The product is most interesting, as it represents the information stored in a dataset. We plan to use four products, that are frequently used by remote sensing analysts:

MOD02 *Level-1B Calibrated Geolocation Data Set* Stores the calibrated radiance for the 36 bands of data, in $W/(m^2 \cdot \mu m \cdot sr)$, the reflectance and some quality indicators.

MOD05 *Total Precipitable Water* Contains daytime column water-vapor amount using the near-infrared and infrared channel. It is not used in the image part, but provides information on radio-wave attenuation for another part of this project.

MOD06 *Cloud Product* Only available for the 1km resolution. It uses the visible, NIR, and Thermal Infra-Red (IR) bands. Contains cloud temperature, cloud pressure, particle phase and radius for the top of the cloud.

MOD35 *Cloud Mask* stores two bits that indicate the status of a pixel: **Confident Clear, Probably Clear, Uncertain, Cloudy**. It uses 19 bands to decide, and the decision tree for each pixel is included for the 250m and 1km resolution.

In our future work, we intend to compare such cloud mask of MODIS data with ground-based sky cameras. It may not be possible to match individual clouds as sky camera is an upward-looking device, whereas MODIS is a downward-looking device. However, it will provide us additional information about cloud top height and other related phenomenon.

Finally, we also intend to explore solar energy harvesting, with the aid of ground-based sky cameras. The prospect of solar energy generation is good in a tropical

country like Singapore. But this generation is intermittent and variable because of the presence of clouds that blocks direct solar radiation. The effect of clouds on solar energy generation have been described in detail in Chapter 7 of this thesis. In our future work, we plan to develop a Photovoltaic (PV) energy forecast system based on automatic cloud coverage and cloud tracking.

Closing note: This thesis ends here—we believe that our work on imaging the earth’s atmosphere can assist the research community in a myriad range of applications.

Appendix A

Calculating Sun Position from Earth's Surface

The position of the sun from the earth's surface is generally measured using two entities—elevation angle and zenith angle. The angle projected from the sun with reference to the horizon is called the elevation angle. The compass angle of the sun from the 0° N direction is called the azimuth angle. We use the Jean Meeus algorithm [121] that has an error of $\pm 3 \cdot 10^{-4}$ degrees. This is one of the most popular algorithms, as its implementation is available in various platforms. It consists of an online web version ¹; also it is available in Python ² and MATLAB ³ programming languages.

¹<https://www.nrel.gov/midc/solpos/spa.html>

²<http://pysolar.org/>

³<http://pvpmc.org/pv-lib/>

Appendix B

Calculating the Position of Sun Blocker

In order to block the light coming directly from the sun, the sun blocker head should intersect the ray from the sun to the camera. We use the Jean Meeus algorithm [121] to compute the azimuth and elevation angles of this ray, which we call ξ and ε respectively. The algorithm has an error of $\pm 3 \cdot 10^{-4}$ degrees. The various lengths and angles required for the computations are shown in Fig. B.1. L_1 represents the length of the main arm. L_2 is the distance of the motor shift between the blocker and the arm. L_3 denotes the length of the stem under the blocker head. L_4 denotes the length between the camera center and the arm motor axis. The two unknowns are $\tilde{\theta}$ which measures the arm angle measured from the horizon and α which measures the stem angle measured from the east direction. The following trigonometric equations are used:

$$a = L_1 \sin \tilde{\theta} + L_2 \cos \tilde{\theta} + L_3(\sin \alpha)(\sin \tilde{\theta}) - L_4, \quad (\text{B.1})$$

$$c = L_3 \cos \alpha, \quad (\text{B.2})$$

$$d = L_1 \cos \tilde{\theta} - L_2 \sin \tilde{\theta} + L_3(\sin \alpha)(\cos \tilde{\theta}), \quad (\text{B.3})$$

$$\tan(\varepsilon) = \left(\frac{a \sin(\xi)}{c} \right)^2, \quad (\text{B.4})$$

$$\sin(\xi) = -\left(\frac{c}{d}\right) \quad (\text{B.5})$$

Figure B.1: Representation of WAHRSIS with the lengths and angles required for the computation of the motor angles.

The arm angle $\tilde{\theta}$ can range from 45 to 135 degrees, whereas the stem angle α can range from 0 to 180 degrees. Combinations of $\tilde{\theta}$ and α with increments of 0.5 are used to compute the related ξ and ε sun ray angles values. The combination leading to the smallest difference between those values and the ones obtained by the Jean Meeus algorithm is used.

Appendix C

Derivation of Normalization Factor

Let us suppose that $a_1, a_2, \dots, a_{\bar{t}}$ be the weather station records for $\bar{t}$ number of time stamps. The luminance values computed for each of the corresponding weather station points are represented by $l_1, l_2, \dots, l_{\bar{t}}$. We attempt to estimate the conversion factor x , such that the objective function $f(x)$ representing the inter-sample distances between weather station and computed luminance value is minimized.

We represent objective function $f(x)$ as:

$$\begin{aligned} f(x) &= (xl_1 - a_1)^2 + (xl_2 - a_2)^2 + \dots + (xl_{\bar{t}} - a_{\bar{t}})^2 \\ &= \sum_{i=1}^{\bar{t}} (xl_i - a_i)^2 \end{aligned}$$

We equate $f'(x)$ to 0, and the normalization factor x is found as $x = \frac{\sum_{i=1}^{\bar{t}} a_i l_i}{\sum_{i=1}^{\bar{t}} l_i^2}$.

Appendix D

Computation of Geometrical Bounds in a Multi-camera Setup

We formulate the geometrical limits for other scenarios as well.

D.1 Noise in X- coordinate

D.1.1 Scenario - II

Upper Bound

$$\begin{aligned} ZC_x &= r \cos(\theta) \\ Z'C_y &= r \sin(\theta) \\ ZC_x &= \frac{r \sin(\theta)}{\tan(\theta_5)} \\ Z'Z &= \frac{r \sin(\theta)}{\tan(\theta_5)} - r \cos(\theta) \end{aligned}$$

Figure D.1: Determination of upper bound for the translation of camera in X-direction (Scenario II).

Lower Bound

$$\begin{aligned} ZC_x &= r \cos(\theta) \\ ZC_y = C_x C &= r \sin(\theta) \\ Z'C_x &= \frac{r \sin(\theta)}{\tan(\theta_4)} \\ Z'Z &= \frac{r \sin(\theta)}{\tan(\theta_4)} - r \cos(\theta) \end{aligned}$$

Figure D.2: Determination of lower bound for the translation of camera in X-direction (Scenario II).

$$\sin(180 - \theta) = \frac{CC_x}{CZ}$$

$$\tan(\theta_4) = \frac{CC_x}{Z'C_x}$$

$$\cos(180 - \theta) = \frac{C_x Z}{CZ}$$

$$Z'Z = \frac{r \sin(\theta)}{\tan(\theta_4)} + r \cos(\theta)$$

D.1.3 Scenario - IV

188

Figure D.5: Probability is 0 for the translation of camera in X-direction (Scenario IV).

D.1.4 Scenario - V

In this scenario, *no* possible translation of camera in X-direction can contain the point in the active region. Therefore, the probability of a point to belong to active region is *null*.

Figure D.6: Probability is 0 for the translation of camera in X-direction (Scenario V).

D.1.5 Scenario - VI

Figure D.7: Determination of lower bound for the translation of camera in X-direction (Scenario VI).

Upper Bound

Figure D.8: Determination of upper bound for the translation of camera in X-direction (Scenario VI).

Therefore, the two geometrical limits - lower and upper are $+(r \cos(\theta) - \frac{r \sin(\theta)}{\tan(\theta_4)})$ and $r \cos(\theta) - \frac{r \sin(\theta)}{\tan(\theta_5)}$ respectively.

$$ZC_x = r \cos(\theta)$$

$$ZC_y = r \sin(\theta)$$

$$Z'C_x = \frac{r \sin(\theta)}{\tan(\theta_4)}$$

$$ZZ' = r \cos(\theta) - \frac{r \sin(\theta)}{\tan(\theta_4)}$$

$$ZC_x = r \cos(\theta)$$

$$ZC_y = r \sin(\theta)$$

$$Z'C_x = \frac{r \sin(\theta)}{\tan(\theta_5)}$$

$$ZZ' = r \cos(\theta) - \frac{r \sin(\theta)}{\tan(\theta_5)}$$

D.2.2 Scenario - III

Figure D.11: Probability of the point C is 0 for the translation of camera in Y-direction (Scenario III).

D.2.3 Scenario - IV

Figure D.12: Probability of the point C is 0 for the translation of camera in Y-direction (Scenario IV).

D.2.5 Scenario - VI

Upper Bound

Figure D.15: Determination of upper bound for the translation of camera in Y-direction (Scenario VI).

$$\cos(90 - \theta) = \frac{C_y P}{CP}$$

$$\implies C_y P = r \sin(\theta)$$

$$\sin(90 - \theta) = \frac{C_y C}{CP}$$

$$\implies C_y C = r \cos(\theta)$$

$$\tan(90 - \theta_4) = \frac{C_y C}{C_y P'}$$

$$\implies C_y P' = \frac{r \cos(\theta)}{\cot(\theta_4)}$$

$$PP' = \frac{r \cos(\theta)}{\cot(\theta_4)} - r \sin(\theta)$$

Lower Bound

Figure D.16: Determination of lower bound for the translation of camera in Y-direction (Scenario VI).

$$\cos(90 - \theta) = \frac{C_y P}{CP}$$

$$\implies C_y P = r \sin(\theta)$$

$$\sin(90 - \theta) = \frac{C_y C}{CP}$$

$$\implies C_y C = r \cos(\theta)$$

$$\tan(90 - \theta_5) = \frac{C_y C}{C_y P'}$$

$$\implies C_y P' = \frac{r \cos(\theta)}{\cot(\theta_5)}$$

$$PP' = \frac{r \cos(\theta)}{\cot(\theta_5)} - r \sin(\theta)$$

D.3 Noise in view angle

D.3.1 Scenario - II

Lower Bound

$$\theta'_5 = \theta_5 + (\theta - \theta_5)$$

$$\Theta + \delta_5 = \theta$$

$$\Rightarrow \Theta = \theta - \delta_5$$

Figure D.17: Determination of lower bound of view angle for the rotation of camera (Scenario II).

Upper Bound

$$\theta'_5 = \theta_5 + (\theta - \theta_4)$$

$$\Theta + \delta_5 = \theta_5 + (\theta - \theta_4)$$

$$\Rightarrow \Theta = \theta_5 + (\theta - \theta_4) - \delta_5$$

Figure D.18: Determination of upper bound of view angle for the rotation of camera (Scenario II).

D.3.2 Scenario - III

Lower Bound

$$\theta'_5 = \theta_5 + (\theta - \theta_5)$$

$$\Theta + \delta_5 = \theta$$

$$\implies \Theta = \theta - \delta_5$$

Figure D.19: Determination of lower bound of view angle for the rotation of camera (Scenario III).

Upper Bound

$$\theta'_5 = \theta_5 + (\theta - \theta_4)$$

$$\Theta + \delta_5 = \theta_5 + (\theta - \theta_4)$$

$$\implies \Theta = \theta_5 + (\theta - \theta_4) - \delta_5$$

Figure D.20: Determination of upper bound of view angle for the rotation of camera (Scenario III).

D.3.3 Scenario - IV

Figure D.21: Probability of the point C is 0 for the rotation of camera (Scenario IV).

D.3.4 Scenario - V

Upper Bound

Figure D.22: Determination of upper bound of view angle for the rotation of camera (Scenario V).

$$\begin{aligned}
 \theta'_4 &= \theta_4 - (\theta_4 + 360 - \theta) \\
 \implies \theta'_4 &= \theta - 360 \\
 \Theta + \delta_4 &= \theta - 360 \\
 \implies \Theta &= \theta - 360 - \delta_4
 \end{aligned}$$

Lower Bound

Figure D.23: Determination of lower bound of view angle for the rotation of camera (Scenario V).

$$\begin{aligned}
 \theta'_4 &= \theta_4 - (\theta_5 + 360 - \theta) \\
 \implies \theta'_4 &= \theta_4 - \theta_5 - 360 + \theta \\
 \Theta + \delta_4 &= \theta_4 - \theta_5 - 360 + \theta \\
 \implies \Theta &= \theta_4 - \theta_5 - 360 + \theta - \delta_4
 \end{aligned}$$

D.3.5 Scenario - VI

Upper Bound

$$\begin{aligned}\theta'_4 &= \theta_4 - (\theta_4 - \theta) \\ \Theta + \delta_4 &= \theta \\ \implies \Theta &= \theta - \delta_4\end{aligned}$$

Figure D.24: Determination of upper bound of view angle for the rotation of camera (Scenario VI).

Lower Bound

$$\begin{aligned}\theta'_4 &= \theta_4 - (\theta_5 - \theta) \\ \Theta + \delta_4 &= \theta_4 - \theta_5 + \theta \\ \implies \Theta &= \theta_4 - \theta_5 + \theta - \delta_4\end{aligned}$$

Figure D.25: Determination of lower bound of view angle for the rotation of camera (Scenario VI).

Notations

α	Stem angle of sun-blocker based WAHRSIS measured from the east direction.
$\tilde{\alpha}$	Membership constant used in assigning seeds of HDR radiance map.
β	Angle of incidence of light in a fish-eye lens system.
$\gamma(\cdot)$	Relevance criterion of an attribute set in rough-set terminology.
$\bar{\gamma}(\cdot)$	Average relevance value of an attribute set across all images of the dataset.
$(\delta_1^{\hat{i}}, \delta_2^{\hat{i}}, \dots, \delta_n^{\hat{i}})$	Offset angles (w.r.t. view angle $\Theta^{\hat{i}}$) contained by $\hat{n}$ -pixels to the camera center for the $\hat{i}$ -th camera.
Δt_m	Time shift.
$\Delta x, \Delta y$	Spatial shift in x- and y- coordinates in an image plane.
$\Delta \hat{x}_c$ and $\Delta \hat{y}_c$	Translation of camera along X- and Y- axes of a multi-camera setup.
$\sigma_{\Delta \hat{x}_c}^2$ and $\sigma_{\Delta \hat{y}_c}^2$	Variance of translation of camera along X- and Y- axes of a multi-camera setup.
$\epsilon_1, \epsilon_2, \dots$	Coefficients in a higher-order polynomial fitting function.
η	Learning rate for gradient ascent algorithm.
Γ	Day angle (measured in radians).
λ	Eigen value.
Λ	Wavelength of light.
μ	Weighting parameter in the graph-cut framework.

μ	Mean vector of the multivariate gaussian distribution of noisy camera poses in multi-camera setup.
ω_1	Angle of rotation around the z axis to align two imagers.
ω_2	Angle of rotation around the y axis to align two imagers.
$\Phi(\mathbf{p}')$	Re-projection function.
ψ	Solar zenith angle (measured in degrees).
Ψ	Multivariate gaussian distribution involving the camera poses to model the noisy camera poses in the multi-camera setup.
σ_{c_j}	Standard deviation of j -th color channel.
$\hat{\sigma}$	Scale of the filter in the filter set.
σ_{Θ}^2	Variance of view angle in the multi-camera setup.
$\sigma_{\Delta x_c}$	Variance of camera position in the X-direction in the multi-camera setup.
$\sigma_{\Delta y_c}$	Variance of camera position in the Y-direction in the multi-camera setup.
Σ	Covariance matrix of the multivariate gaussian distribution of noisy camera poses in multi-camera setup.
τ	Fuzziness index in clustering process.
$\hat{\tau}$	Number of cycles inside the gaussian envelope of the filter set.
θ	Angle made by the <i>intruder</i> $\mathbf{p}$ with the global x-axis in the multi-camera setup.
$(\theta_1^{\hat{i}}, \theta_2^{\hat{i}}, \dots, \theta_n^{\hat{i}})$	Angles contained by $\hat{n}$ -pixels to the camera center for the $\hat{i}$ -th camera.
$\tilde{\theta}$	Arm angle of sun-blocker based WAHRIS measured from the horizon.
$\Theta^{\hat{i}}$	View angle of $\hat{i}$ -th camera w.r.t. global X-axis.
$\bar{\Theta}$	Mean value of view angle of the camera in the multi-camera setup.
ε	Elevation angle of the ray from sun to WAHRIS camera.
ξ	Azimuth angle of the ray from sun to WAHRIS camera.
a	Element from set $\mathcal{P}$.

$\hat{a}(\cdot)$	Geometrical lower bound in multivariate distribution for the multi-camera setup.
a_z	Represents the z -th attribute from the set $\mathcal{A}$.
$a_1, a_2, \dots$	Individual weather station data points.
$\hat{b}(\cdot)$	Geometrical upper bound in multivariate distribution for the multi-camera setup.
b_1	Measure of square of skewness.
b_2	Measure of kurtosis.
$\mathbf{c}$	Weight vectors of $\mathbf{Y}$.
$c_1, c_2, \dots, c_{16}$	Set of 16 color channels for analysis of ground-based images.
$\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{16}$	Vectorised set of the 16 color channels $c_1, c_2, \dots, c_{16}$ with dimensions $mn \times 1$.
$\bar{c}_j$	Mean value of j -th color channel.
d_n	Day number of the year.
$\mathbf{e}$	Eigen vector.
e_t	Exposure time of the camera measured in seconds.
$f(\cdot)$	Mapping function.
$\bar{f}$	Focal length of the lens in the imaging system.
f_s	Aperture number of the lens in the imaging system.
h_j^{class}	Histogram of textons for a cloud category of training images.
h_j^{test}	Histogram of textons for a cloud category of testing images.
i	Index of the set of images.
$\hat{i}$	Index of the set of cameras in a multi-camera setup.
j	Index of the set of color channels.
$\hat{j}$	Index of the set of cloud categories.
k	Number of color channels in a subset of color channels.
l	Index of the vectorised form of an image.
$l_1, l_2, \dots$	Individual luminance data points for corresponding weather station points.
m	Row size of a sample image.
n	Column size of a sample image.

$\tilde{n}$	Number of cameras in the multi-camera setup.
$\hat{n}$	Total number of pixels in the finite-dimensional camera.
n_c	Number of clusters used in fuzzy clustering process.
n_f	Number of filters in the filter bank.
n_t	Number of textons generated from k-means clustering.
o	Dimension of cluster center vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.
p	Joint multivariate normal cumulative probability values for all the $\hat{i}$ cameras.
$p^1, p^2, \dots$	Individual multivariate normal cumulative probability values for all the $\hat{i}$ cameras.
$\mathbf{p}$	Reconstruction point in multi-camera setup (also referred as <i>intruder</i> in the art-gallery problem).
$\mathbf{p}^*$	Optimal reconstruction point.
$\bar{p}$	Total number of cloud categories.
$\tilde{p}$	Sample pixel in the image $\mathbf{Z}_i$.
$\tilde{q}$	Neighboring pixel of $\tilde{p}$ in the image $\mathbf{Z}_i$.
$\mathbf{q}$	Center of the <i>active pixel</i> in the multi-camera setup.
$\bar{q}$	Total number of images in each cloud category.
q_k	Represents the k -th pixel value of the image.
r	Radius on the image plane in a fish-eye lens system.
$\hat{r}$	2D Euclidean distance between the camera location and any sample point $(\hat{x}, \hat{y})$ in the XY-plane $(= \ (\hat{x}, \hat{y}), (\hat{x}_c^i, \hat{y}_c^i)\)$.
$\bar{s}$	Index of the sampled pixel in an <i>RGB</i> image.
s, t	Sample indexes of observations from the universe set $\mathcal{U}$.
$\mathbf{t}$	Random initialization point in $\mathbb{R}^2$ used in the gradient ascent algorithm for computation of $F(\cdot)$.
t_m	Current observation time.
t_s	Total number of sampled points in the hemispheric sampling process.
u_x, u_y	Optical flow vectors in x- and y- direction.
$\mathbf{w}$	Weight vectors of $\mathbf{X}$.

$\mathbf{v}_{n_c}$	The n_c cluster centers.
x, y	Sample x- and y- image coordinate.
$\hat{x}_c^i, \hat{y}_c^i$	Location of $\hat{i}$ -th camera in the XY-plane of a multi-camera setup.
x_m, x_n	Two sample observations from universe $\mathcal{U}$ in rough set terminology.
$\mathbf{x}^f$	Vectorised form of feature matrix $\mathbf{X}^f$.
z	Index of the set of attributes in attribute set $\mathcal{A}$.
$\bar{\mathbf{z}}$	Vectorized form of the gray-scale image $\bar{\mathbf{Z}}$.
$\bar{\mathbf{z}}_m$	Mean value of the vectorized image $\bar{\mathbf{z}}$.
$\bar{\mathbf{z}}_d$	Median value of the vectorized image $\bar{\mathbf{z}}$.
$\bar{\mathbf{z}}_v$	Standard deviation value of the vectorized image $\bar{\mathbf{z}}$.
$\bar{\mathbf{z}}_w$	Skewness value of the vectorized image $\bar{\mathbf{z}}$.
$\bar{\mathbf{z}}_k$	Kurtosis value of the vectorized image $\bar{\mathbf{z}}$.
$\mathcal{A}$	Set of attributes in rough-set terminology.
$A_{\tilde{p}}$	Data cost for an individual pixel $\tilde{p}$ in a graph-cut framework.
$\mathbf{B}$	Regression coefficient matrix.
$B_{\tilde{p}, \tilde{q}}$	Interaction cost between two neighboring pixels $\tilde{p}$ and $\tilde{q}$ in a small neighborhood.
$\mathcal{C}$	Set of condition attributes in rough-set terminology.
$\mathbf{C}^T$	A matrix of regression coefficients.
$\mathcal{D}$	Set of decision attributes in rough-set terminology.
E	Segmentation score for graph-cut algorithm.
E_0	Intensity of the emitted light.
$\mathbf{E}_i$	Residual matrix corresponding to feature matrix $\mathbf{X}_i^f$.
E_k	Eccentricity correction factor for earth.
E_s	Intensity of the scattered light.
$F(\cdot)$	multivariate normal cumulative probability for a single camera in the multi-camera setup.
$\mathbf{F}_i$	Residual matrix corresponding to binary image $\mathbf{Y}_i$.
$\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{n_f}$	Set of filters in the filter bank.

$G(\cdot)$	Objective function to optimize the multi-variate distribution in the multi-camera setup.
G_c	Global Horizontal Irradiance of the sun at the earth's surface.
$\mathbf{H}_i$	Residual matrix corresponding to latent matrix $\mathbf{U}_i$.
$\mathcal{H}_i$	HDR radiance map generated from the set of low-, mid- and high-LDR images.
HSV	Hue, Saturation and Value color channel model comprising hue (H), saturation (S) and value (V) color channels.
I	Clearness Luminance Index.
I_c	Critical Clearness Luminance Index.
I_s	ISO sensitivity of the camera.
I_{sc}	Solar irradiance constant (1366.1 Watt/m ²).
$\mathcal{I}$	Intruder in a 2D search space in a multi-camera setup.
IHS	Intensity, Hue, and Saturation color model.
J	Objective function used during fuzzy-clustering process.
K_c	Calibration constant to map scene luminance and pixel brightness.
L_w	Calculated luminance of a pixel in a fish-eye lens during vignetting modeling.
$\mathcal{L}$	Decision table in rough-set terminology.
$\mathcal{L}_c$	Clear-sky luminance.
$\mathcal{L}_m$	Measured luminance from sky/cloud image.
$\mathcal{L}_r$	Relative luminance.
$\mathcal{L}_s$	Scene luminance.
$L_{\tilde{p}}$	Binary label of the pixel $\tilde{p}$.
L_1, L_2, L_3, L_4	Measurement lengths of the different parts of sun-blocker based WAHRSIS model.
$L^*a^*b^*$	Color space comprising lightness (L) and color-opponent dimensions (a^* and b^*).
$\mathbf{M}_i$	Covariance matrix of the normalized matrix $\ddot{\mathbf{X}}_i$.
N	Total number of images in a dataset.

$\mathcal{N}$	Average luminance value of sampled pixels during hemispheric sampling.
$\tilde{\mathcal{N}}$	Small neighborhood region.
$\mathcal{P}$	Subset of attributes from attribute set $\mathcal{A}$.
$\underline{\mathcal{P}}(.)$	Lower approximation of a conventional set.
$\overline{\mathcal{P}}(.)$	Upper approximation of a conventional set.
$\mathbf{P}_i$	Loading matrix corresponding to feature matrix $\mathbf{X}_i^f$.
$\mathbf{Q}_i$	Loading matrix corresponding to binary image $\mathbf{Y}_i$.
R_1, R_2	Two random floating points used during cosine weighted hemispheric sampling.
$\mathcal{R}$	Red-blue ratio channel R/B extracted from $\mathbf{Z}$, where $\mathcal{R} \in \mathbb{R}^{m \times n}$.
$\tilde{\mathcal{R}}$	Red-blue ratio channel $(B - R)/(B + R)$ extracted from $\mathbf{Z}$, where $\tilde{\mathcal{R}} \in \mathbb{R}^{m \times n}$.
RGB	Red, Green and Blue color channel model comprising red (R), green (G) and blue (B) color channels.
$\mathbf{S}$	Set of cameras in the multi-camera setup (also referred as <i>guards</i> in the art gallery problem).
$\mathcal{S}$	Solar radiation measured by the weather station.
$\hat{\mathcal{S}}$	Search space in point localization task of a multi-camera setup.
$\mathcal{T}$	Dataset consisting of a set of images.
$\mathbf{T}_i$	Extracted latent matrix corresponding to binary image $\mathbf{Y}_i$.
$\mathcal{U}$	Universe with set of observations in rough-set terminology.
$\mathbf{U}_i$	Extracted latent matrix corresponding to feature matrix $\mathbf{X}_i^f$.
$\mathcal{V}$	Value domain in rough set terminology.
$\mathcal{X}$	Set of observations from the universe $\mathcal{U}$.
$\mathbf{X}_i$	Sample image of dimension $m \times n$ from a dataset consisting of N images ($i = 1, 2, \dots, N$), where $\mathbf{X}_i \in \mathbb{R}^{m \times n}$.
$\mathbf{X}_i^f$	Feature matrix obtained from the sample image $\mathbf{X}_i$, where $\mathbf{X}^f \in \mathbb{R}^{m \times n}$.

$\hat{\mathbf{X}}_i$	Fat matrix obtained from the sample image $\mathbf{X}_i$ using all 16 color channels.
$\ddot{\mathbf{X}}_i$	Normalized form of $\hat{\mathbf{X}}_i$ with its corresponding mean and standard deviation.
$Y_{\bar{s}}$	Mean brightness value of the $\bar{s}$ pixel in an <i>RGB</i> image.
$Y'_{\bar{s}}$	Non-linearly corrected mean brightness value of the $\bar{s}$ pixel in an <i>RGB</i> image.
$\mathbf{Y}_i$	Binary ground-truth image of the sample image $\mathbf{X}_i$, where $\mathbf{Y}_i \in \mathbb{R}^{m \times n}$.
YIQ	Luma and chrominance color channel, where Y is luminance and I and Q represents in- and quadrature-phase of chrominance information.
$\mathbf{Z}_i$	Sample image of dimension $m \times n \times 3$ from a dataset consisting of N images ($i = 1, 2, \dots, N$), where $\mathbf{Z}_i \in \mathbb{R}^{m \times n \times 3}$.
$\bar{\mathbf{Z}}_i$	Gray-scale channel image of the sample image $\mathbf{Z}_i$, where $\bar{\mathbf{Z}}_i \in \mathbb{R}^{m \times n}$.
$\mathbf{Z}_i^R$	Red channel image of the sample image $\mathbf{Z}_i$, where $\mathbf{Z}_i^R \in \mathbb{R}^{m \times n}$.
$\mathbf{Z}_i^G$	Green channel image of the sample image $\mathbf{Z}_i$, where $\mathbf{Z}_i^G \in \mathbb{R}^{m \times n}$.
$\mathbf{Z}_i^B$	Blue channel image of the sample image $\mathbf{Z}_i$, where $\mathbf{Z}_i^B \in \mathbb{R}^{m \times n}$.
$\mathbf{Z}_i^L$	Low-exposure LDR image, where $\mathbf{Z}_i^L \in \mathbb{R}^{m \times n \times 3}$.
$\mathbf{Z}_i^M$	Mid-exposure LDR image, where $\mathbf{Z}_i^M \in \mathbb{R}^{m \times n \times 3}$.
$\mathbf{Z}_i^H$	High-exposure LDR image, where $\mathbf{Z}_i^H \in \mathbb{R}^{m \times n \times 3}$.

Author's Publications

Journals

- S. Dev, F. M. Savoy, Y. H. Lee, S. Winkler, Rough-set Based Visible Color Channel Selection, *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 52–56, Jan 2017.
- S. Dev, Y. H. Lee, S. Winkler, Color-based segmentation of sky/cloud images from ground-based cameras, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 231–242, Jan 2017.
- S. Dev, B. Wen, Y. H. Lee, S. Winkler, Machine Learning Techniques and Applications For Ground-based Image Analysis, *IEEE Geoscience and Remote Sensing Magazine, Special Issue on Advances in Machine Learning for Remote Sensing and Geosciences*, vol. 4, no. 2, pp. 79–93, Jun 2016.

Conferences

- S. Dev, F. M. Savoy, Y. H. Lee and S. Winkler, Nighttime sky/cloud image segmentation, *IEEE International Conference on Image Processing (ICIP)*, 2017.
- F. Savoy, S. Dev, Y. H. Lee and S. Winkler, Stereoscopic Cloud Base Reconstruction Using High-Resolution Whole Sky Imagers, *IEEE International Conference on Image Processing (ICIP)*, 2017.
- S. Dev, S. Manandhar, F. Yuan, Y. H. Lee and S. Winkler, Cloud Radiative Effect Study Using Sky Camera, *Proc. IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, 2017.

- S. Dev, S. Manandhar, Y. H. Lee and S. Winkler, Detecting rainfall onset using sky images, *TENCON 2016 - 2016 IEEE Region 10 Conference*, 2016.
- S. Dev, F. M. Savoy, Y. H. Lee and S. Winkler, Short-term prediction of localized cloud motion using ground-based sky imagers, *TENCON 2016 - 2016 IEEE Region 10 Conference*, 2016.
- S. Manandhar, F. Yuan, S. Dev, Y. H. Lee and Y. S. Meng, Weather radar to detect cloud occurrence level, *TENCON 2016 - 2016 IEEE Region 10 Conference*, 2016.
- S. Dev, F. M. Savoy, Y. H. Lee, S. Winkler, Estimation of solar irradiance using ground-based whole sky imagers, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.
- F. M. Savoy, S. Dev, Y. H. Lee, S. Winkler, Geo-referencing and stereo calibration of ground-based whole sky imagers using the sun trajectory, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.
- S. Manandhar, Y. H. Lee, S. Dev, GPS Derived PWV for rainfall monitoring, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.
- S. Dev, Y. H. Lee, S. Winkler, Categorization of Cloud Image Patches Using an Improved Texton-based Approach, *IEEE International Conference on Image Processing (ICIP)*, 2015.
- S. Dev, Y. H. Lee, S. Winkler, Multi-level Semantic Labeling of sky/cloud images, *IEEE International Conference on Image Processing (ICIP)*, 2015.
- S. Dev, F. M. Savoy, Y. H. Lee, S. Winkler, Design of low-cost, compact and weather-proof whole sky imagers for high-dynamic-range captures, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015.
- F. M. Savoy, S. Dev, Y. H. Lee, S. Winkler, Cloud base height estimation using high-resolution whole sky imagers, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015.

- S. Dev, Y. H. Lee, S. Winkler, Systematic Study of Color Spaces and Components for the segmentation of sky/cloud images, *IEEE International Conference on Image Processing (ICIP)*, 2014.
- S. Dev, F. M. Savoy, Y. H. Lee, S. Winkler, WAHRIS: A low-cost, high-resolution whole sky imager with near-infrared capabilities, *Proc. IS&T/SPIE Infrared Imaging Systems: Design, Analysis, Modeling, and Testing*, 2014.

Technical Reports

- S. Dev, F. M. Savoy, Y. H. Lee and S. Winkler, DIY Sky Imager For Weather Observation: A complete guide to build a ground-based sky imager using off-the-shelf components with automatic cloud coverage computation, SPM Student Design Project Series Documents, *IEEE Signal Processing Society*, 2016.
(Featured in H. Godrich, Students' Design Project Series: Sharing Experiences [SP Education], IEEE Signal Processing Magazine, Jan 2017.)

Bibliography

- [1] T. Shiraishi, T. Motohka, R.B. Thapa, M. Watanabe, and M. Shimada, “Comparative assessment of supervised classifiers for land use-land cover classification in a tropical region using time-series PALSAR mosaic data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1186–1199, Apr. 2014.
- [2] I-H. Lee and M.T. Mahmood, “Robust registration of cloudy satellite images using two-step segmentation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1121–1125, May 2015.
- [3] Q. Zhang and C. Xiao, “Cloud detection of RGB color aerial photographs by progressive refinement scheme,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7264–7275, Nov. 2014.
- [4] Y. Chen, L. Cheng, M. Li, J. Wang, L. Tong, and K. Yang, “Multiscale grid method for detection and reconstruction of building roofs from airborne LiDAR data,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 10, pp. 4081–4094, Oct. 2014.
- [5] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, “WAHRIS: A low-cost, high-resolution whole sky imager with near-infrared capabilities,” in *Proc. IS&T/SPIE Infrared Imaging Systems*, 2014.
- [6] J. X. Yeo, Y. H. Lee, and J. T. Ong, “Performance of site diversity investigated through RADAR derived results,” *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 10, pp. 3890–3898, Oct. 2011.
- [7] L. S. Kumar, Y. H. Lee, and J. T. Ong, “Truncated gamma drop size distribution models for rain attenuation in singapore,” *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 4, pp. 1325–1335, Apr. 2010.
- [8] J. B. Collins and C. E. Woodcock, “An assessment of several linear change detection techniques for mapping forest mortality using multitemporal landsat TM data,” *Remote Sensing of Environment*, vol. 56, no. 1, pp. 66–77, Apr. 1996.

- [9] H. Lee and H. Han, "Evaluation of SSM/I and AMSR-E sea ice concentrations in the antarctic spring using KOMPSAT-1 EOC images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 1905–1912, July 2008.
- [10] D. Wylie, E. Eloranta, J. D. Spinhirne, and S. P. Palm, "A comparison of cloud cover statistics from the GLAS Lidar with HIRS," *Journal of Climate*, vol. 20, no. 19, pp. 4968–4981, Oct. 2007.
- [11] C-L Fua and H-Y Cheng, "Predicting solar irradiance with all-sky image features via regression," *Solar Energy*, vol. 97, pp. 537–550, Nov. 2013.
- [12] U. Schumann, R. Hempel, H. Flentje, M. Garhammer, K. Graf, S. Kox, H. Losslein, and B. Mayer, "Contrail study with ground-based cameras," *Atmospheric Measurement Techniques*, vol. 6, no. 12, pp. 3597–3612, Dec. 2013.
- [13] A. Chatterjee, A. M. Michalak, R. A. Kahn, S. R. Paradise, A. J. Braverman, and C. E. Miller, "A geostatistical data fusion technique for merging remote sensing and ground-based observations of aerosol optical thickness," *Journal of Geophysical Research*, vol. 115, no. D20, pp. n/a–n/a, Oct. 2010.
- [14] F. Yuan, Y. H. Lee, and Y. S. Meng, "Comparison of cloud models for propagation studies in Ka-band satellite applications," in *Proc. IEEE International Symposium on Antennas and Propagation*, 2014, pp. 383–384.
- [15] F. Yuan, Y. H. Lee, and Y. S. Meng, "Comparison of radio-sounding profiles for cloud attenuation analysis in the tropical region," in *Proc. IEEE International Symposium on Antennas and Propagation*, 2014, pp. 259–260.
- [16] E. Rumi, D. Kerr, J. M. Coupland, A. P. Sandford, and M. J. Brettle, "Automated cloud classification using a ground based infra-red camera and texture analysis techniques," in *Proc. SPIE 8890, Remote Sensing of Clouds and the Atmosphere XVIII*, 2013.
- [17] A. Cazorla, F. J. Olmo, and L. Alados-Arboledas, "Development of a sky imager for cloud cover assessment," *Journal of the Optical Society of America A*, vol. 25, no. 1, pp. 29–39, Aug. 2008.
- [18] 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, Sept. 2010.
- [19] A. Kazantzidis, P. Tzoumanikas, A. F. Bais, S. Fotopoulos, and G. Economou, "Cloud detection and classification with the use of whole-sky ground-based images," *Atmospheric Research*, vol. 113, pp. 80–88, Sept. 2012.

- [20] C. N. Long, J. M. 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, May 2006.
- [21] F. Abdi, H. R. Kholesifard, and P. H. Falmant, “Small scale cloud dynamics as studied by synergism of time lapsed digital camera and elastic LIDAR,” in *Proc. International Laser Radar Conference*, 2006.
- [22] J. A. Zehnder, J. Hu, and A. Razdan, “A stereo photogrammetric technique applied to orographic convection,” *Monthly Weather Review*, vol. 135, no. 6, pp. 2265–2277, June 2007.
- [23] U. Feister, J. Shields, M. Karr, R. Johnson, K. Dehne, and M. Woldt, “Ground-based cloud images and sky radiances in the visible and near infrared region from whole sky imager measurements,” in *Proc. Climate Monitoring Satellite Application Facility Training Workshop*, 2000.
- [24] M. S. Ghonima, B. Urquhart, C. W. Chow, J. E. Shields, A. Cazorla, and J. Kleissl, “A method for cloud detection and opacity classification based on ground based sky imagery,” *Atmospheric Measurement Techniques*, vol. 5, no. 11, pp. 2881–2892, Nov. 2012.
- [25] C. N. Long, D. W. Slater, and T. Tooman, “Total sky imager (TSI) model 880 status and testing results,” Tech. Rep. DOE/SC-ARM/TR-006, Atmospheric Radiation Measurement (ARM), 2001.
- [26] Z. Chen, Y. Feng, A. Lindner, G. Barrenetxea, and M. Vetterli, “How is the weather: Automatic inference from images,” in *Proc. International Conference on Image Processing (ICIP)*, 2012, pp. 1853–1856.
- [27] M. P. Souza-Echer, E. B. Pereira, L. S. Bins, and M. A. R. 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, Mar. 2006.
- [28] J. E. Shields, M. E. Karr, R. W. Johnson, and A. R. Burden, “Day/night whole sky imagers for 24-h cloud and sky assessment: history and overview,” *Applied Optics*, vol. 52, no. 8, pp. 1605–1616, Mar. 2013.
- [29] M. E. Farmer and A. K. Jain, “A wrapper-based approach to image segmentation and classification,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2060–2072, Dec. 2005.

- [30] J. Calbó and J. Sabburg, “Feature extraction from whole-sky ground-based images for cloud-type recognition,” *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 1, pp. 3–14, Jan. 2008.
- [31] A. Heinle, A. Macke, and A. Srivastav, “Automatic cloud classification of whole sky images,” *Atmospheric Measurement Techniques*, vol. 3, no. 3, pp. 557–567, 2010.
- [32] S. Liu, L. Zhang, Z. Zhang, C. Wang, and B. Xiao, “Automatic cloud detection for all-sky images using superpixel segmentation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 354–358, Feb. 2015.
- [33] 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, Oct. 2011.
- [34] Y. Yuan and X. Hu, “Bag-of-words and object-based classification for cloud extraction from satellite imagery,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 8, pp. 4197 – 4205, Aug. 2015.
- [35] S. Liu, Z. Zhang, B. Xiao, and X. Cao, “Ground-based cloud detection using automatic graph cut,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 6, pp. 1342–1346, June 2015.
- [36] World Meteorological Organization, *International Cloud Atlas*, vol. 1, 1975.
- [37] M. Singh and M. Glennen, “Automated ground-based cloud recognition,” *Pattern Analysis and Applications*, vol. 8, no. 3, pp. 258–271, Oct. 2005.
- [38] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. International Conference on Computer Vision (ICCV)*, 1999, pp. 1150–1157.
- [39] H.-Y. Cheng and C.-C. Yu, “Block-based cloud classification with statistical features and distribution of local texture features,” *Atmospheric Measurement Techniques*, vol. 8, no. 3, pp. 1173–1182, Mar. 2015.
- [40] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, “Salient local binary pattern for ground-based cloud classification,” *Acta Meteorologica Sinica*, vol. 27, no. 2, pp. 211–220, Apr. 2013.
- [41] Z. Zhang, Y. Zhang, and S. Liu, *Ground-Based Cloud Classification Using Pyramid Salient LBP*, pp. 595–601, Springer Berlin Heidelberg, 2016.
- [42] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, “Illumination-invariant completed LTP descriptor for cloud classification,” in *Proc. 5th International Congress on Image and Signal Processing (CISP)*, 2012, pp. 449–453.

- [43] G. H. Hargreaves and Z. A. Samani, "Reference crop evapotranspiration from temperature," *Applied engineering in agriculture*, vol. 1, no. 2, pp. 96–99, Jan. 1985.
- [44] M. Donatelli and G. S. Campbell, "A simple model to estimate global solar radiation," in *Proc. 5th European society of agronomy congress*, 1998, pp. 133–134.
- [45] K. L. Bristow and G. S. Campbell, "On the relationship between incoming solar radiation and daily maximum and minimum temperature," *Agricultural and forest meteorology*, vol. 31, no. 2, pp. 159–166, May 1984.
- [46] L. A. Hunt, L. Kuchar, and C. J. Swanton, "Estimation of solar radiation for use in crop modelling," *Agricultural and Forest Meteorology*, vol. 91, no. 3, pp. 293–300, June 1998.
- [47] J. Alonso-Montesinos and F. J. Batlles, "The use of a sky camera for solar radiation estimation based on digital image processing," *Energy*, vol. 90, Part 1, pp. 377–386, Oct. 2015.
- [48] A. Farahmand, A. AghaKouchak, and J. Teixeira, "A vantage from space can detect earlier drought onset: An approach using relative humidity," Tech. Rep., Nature Scientific Reports, 2015.
- [49] J. Li and K. Nakamura, "Characteristics of the mirror image of precipitation observed by the TRMM precipitation radar," *Journal of Atmospheric and Oceanic Technology*, vol. 19, no. 2, pp. 145–158, Feb. 2002.
- [50] J. Nou, R. Chauvin, S. Thil, J. Eynard, and S. Grieu, "Clear-sky irradiance model for real-time sky imager application," *Energy Procedia*, vol. 69, pp. 1999–2008, May 2015.
- [51] W. P. Menzel, "Cloud tracking with satellite imagery: From the pioneering work of Ted Fujita to the present," *Bulletin of the American Meteorological Society*, vol. 82, no. 1, pp. 33–47, Jan. 2001.
- [52] J. M. Sieglaff, D. C. Hartung, W. F. Feltz, L. M. Cronic, and V. Lakshmanan, "A satellite-based convective cloud object tracking and multipurpose data fusion tool with application to developing convection," *Journal of Atmospheric and Oceanic Technology*, vol. 30, no. 3, pp. 510–525, Mar. 2013.
- [53] C. W. Chow, S. Belongie, and J. Kleissl, "Cloud motion and stability estimation for intra-hour solar forecasting," *Solar Energy*, vol. 115, pp. 645–655, May 2015.
- [54] R. Chauvin, J. Nou, S. Thil, and S. Grieu, "Cloud motion estimation using a sky imager," *AIP Conference Proceedings*, vol. 1734, no. 1, pp. 150003, May 2016.

- [55] M. C. Allmen and W. P. Kegelmeyer Jr., “The computation of cloud-base height from paired whole-sky imaging cameras,” *Journal of Atmospheric and Oceanic Technology*, vol. 13, no. 1, pp. 97–113, February 1996.
- [56] G. Seiz, E. P. Baltsavias, and A. Gruen, “Cloud mapping from the ground: Use of photogrammetric methods,” *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 9, pp. 941–951, September 2002.
- [57] E. Kassianov, C. N. Long, and J. Christy, “Cloud-base-height estimation from paired ground-based hemispherical observations,” *Journal of Applied Meteorology*, vol. 44, no. 8, pp. 1221–1233, August 2005.
- [58] Z. Peng, S. Yoo, D. Yu, D. Huang, P. Kalb, and J. Heiser, “3D cloud detection and tracking for solar forecast using multiple sky imagers,” in *Proc. ACM Symposium on Applied Computing*, 2014.
- [59] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, “Three-dimensional scene flow,” in *Proc. International Conference on Computer Vision (ICCV)*, 1999.
- [60] F. Huguet and F. Devernay, “A variational method for scene flow estimation from stereo sequences,” in *Proc. International Conference on Computer Vision (ICCV)*, 2007.
- [61] R. Li and S. Sclaroff, “Multi-scale 3D scene flow from binocular stereo sequences,” *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 75–90, April 2008.
- [62] J. Cech, J. Sanchez-Riera, and R. Horaud, “Scene flow estimation by growing correspondence seeds,” in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [63] J. E. Shields, M. E. Karr, R. W. Johnson, and A. R. Burden, “Day/night whole sky imagers for 24-h cloud and sky assessment: history and overview,” *Applied Optics*, vol. 52, no. 8, pp. 1605–1616, Mar. 2013.
- [64] L. Schaul, C. Fredembach, and S. Süsstrunk, “Color image dehazing using the near-infrared,” in *Proc. International Conference on Image Processing (ICIP)*, 2009, pp. 1629–1632.
- [65] H. Juan and L. Da-Ren, “Calibration and validation of an all-sky imager,” *Atmospheric and Oceanic Science Letters*, vol. 2, no. 4, pp. 220–223, July 2009.
- [66] S. Shah and J. K. Aggarwal, “Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation,” *Pattern Recognition*, vol. 29, no. 11, pp. 1775–1788, Nov. 1996.

- [67] H. Bakstein and T. Pajdla, “Panoramic mosaicing with a 180 field of view lens,” in *Proc. Third Workshop on Omnidirectional Vision*, 2002, pp. 60–67.
- [68] P. Sturm and S. Ramalingam, “A generic concept for camera calibration,” in *Proc. European Conference on Computer Vision (ECCV)*, 2004, pp. 1–13.
- [69] J. Kannala and S. Brandt, “A generic camera calibration method for fish-eye lenses,” in *Proc. International Conference on Pattern Recognition (ICPR)*, 2004, pp. 10–13.
- [70] D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easily calibrating omnidirectional cameras,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5695–5701.
- [71] P. d’Angelo, “Radiometric alignment and vignetting calibration,” in *Proc. Camera Calibration Methods for Computer Vision Systems*, 2007.
- [72] P. E. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” in *Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 369–378.
- [73] K. Zuiderveld, “Graphics gems IV,” pp. 474–485. Academic Press Professional, Inc., 1994.
- [74] A. O. Akyüz, “High dynamic range imaging pipeline on the GPU,” *Journal of Real-Time Image Processing*, vol. 10, no. 2, pp. 273–287, June 2015.
- [75] T. Richter, “On the standardization of the JPEG XT image compression,” in *Proc. Picture Coding Symposium (PCS)*, 2013, pp. 37–40.
- [76] T. R. Knapp, “Bimodality revisited,” *Journal of Modern Applied Statistical Methods*, vol. 6, no. 1, pp. 3, 2007.
- [77] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, June 2006.
- [78] H. Mamitsuka, “Selecting features in microarray classification using ROC curves,” *Pattern Recognition*, vol. 39, no. 12, pp. 2393 – 2404, Dec. 2006.
- [79] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [80] C. V. Jawahar, P. K. Biswas, and A. K. Ray, “Investigations on fuzzy thresholding based on fuzzy clustering,” *Pattern Recognition*, vol. 30, no. 10, pp. 1605–1613, Oct. 1997.

- [81] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.
- [82] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, 1992.
- [83] S. Dev, Y. H. Lee, and S. Winkler, "Multi-level semantic labeling of sky/cloud images," in *Proc. International Conference on Image Processing (ICIP)*, 2015, pp. 636–640.
- [84] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, Mar. 1993.
- [85] R. Manne, "Analysis of two partial-least-squares algorithms for multivariate calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 187–197, Aug. 1987.
- [86] S. Dev, Y. H. Lee, and S. Winkler, "Systematic study of color spaces and components for the segmentation of sky/cloud images," in *Proc. International Conference on Image Processing (ICIP)*, 2014, pp. 5102–5106.
- [87] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sustrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [88] M. Ester, H-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
- [89] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.
- [90] C. L. Novak and S. A. Shafer, "Anatomy of a color histogram," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992, pp. 599–605.
- [91] C. Schmid, "Constructing models for content-based image retrieval," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. II–39–II–45 vol.2.
- [92] S. Liu, Z. Zhang, B. Xiao, and X. Cao, "Ground-based cloud detection using automatic graph cut," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 6, pp. 1342–1346, June 2015.

- [93] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
- [94] Y.Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. International Conference on Computer Vision (ICCV)*, 2001, pp. 105–112 vol.1.
- [95] C. I. Christodoulou, S. C. Michaelides, and C. S. Pattichis, "Multifeature texture analysis for the classification of clouds in satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 11, pp. 2662–2668, Nov. 2003.
- [96] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [97] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 4, pp. 269–285, Apr. 1976.
- [98] C.-M. Wu, Y.-C. Chen, and K.-S. Hsieh, "Texture features for classification of ultrasonic liver images," *IEEE Transactions on Medical Imaging*, vol. 11, no. 2, pp. 141–152, June 1992.
- [99] S. Liu and Z. Zhang, "Learning group patterns for ground-based cloud classification in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–6, Mar. 2016.
- [100] Z. Wu, X. Xu, M. Xia, M. Ma, and L. Li, "Ground-based vision cloud image classification based on extreme learning machine," *The Open Cybernetics & Systemics Journal*, vol. 9, pp. 2877–2885, Nov. 2015.
- [101] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61–81, April-May 2005.
- [102] F. Sun and J. He, "The remote-sensing image segmentation using textons in the normalized cuts framework," in *Proc. International Conference on Mechatronics and Automation (ICMA)*, 2009, pp. 1877–1881.
- [103] G. Burghouts and J. Geusebroek, "Color textons for texture recognition," in *Proc. British Machine Vision Conference (BMVC)*, 2006, pp. 1099–1108.
- [104] S. Dev, Y. H. Lee, and S. Winkler, "Categorization of cloud image patches using an improved texton-based approach," in *Proc. International Conference on Image Processing (ICIP)*, 2015, pp. 422–426.

- [105] M. Lefèvre, P. Blanc, B. Espinar, B. Gschwind, L. Ménard, T. Ranchin, L. Wald, L. Saboret, C. Thomas, and E. Wey, “The HelioClim-1 database of daily solar radiation at earth surface: An example of the benefits of geoss data-core,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 5, pp. 1745–1753, May 2014.
- [106] T. B. M. J. Ouarda, C. Charron, P. R. Marpu, and F. Chebana, “The generalized additive model for the assessment of the direct, diffuse, and global solar irradiances using SEVIRI images, with application to the UAE,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 4, pp. 1553–1566, Apr. 2016.
- [107] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer, “Irradiance forecasting for the power prediction of grid-connected photovoltaic systems,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 2, no. 1, pp. 2–10, Mar. 2009.
- [108] S. Dev, S. Manandhar, F. Yuan, Y. H. Lee, and S. Winkler, “Cloud radiative effect study using sky camera,” in *Proc. IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, 2077.
- [109] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, “Design of low-cost, compact and weather-proof whole sky imagers for High-Dynamic-Range captures,” in *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 5359–5362.
- [110] D. Yang, P. Jirutitijaroen, and W. M. Walsh, “The estimation of clear sky global horizontal irradiance at the equator,” *Energy Procedia*, vol. 25, pp. 141–148, Dec. 2012.
- [111] F. M. Savoy, S. Dev, Y. H. Lee, and S. Winkler, “Geo-referencing and stereo calibration of ground-based whole sky imagers using the sun trajectory,” in *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 7473–7476.
- [112] C. Poynton, *Digital Video and HDTV Algorithms and Interfaces*, Morgan Kaufmann Publishers Inc., 2003.
- [113] P. D. Hiscocks and P. Eng, “Measuring luminance with a digital camera,” *Syscomp Electronic Design Limited*, 2011.
- [114] D. Fleet and Y. Weiss, “Optical flow estimation,” in *Handbook of mathematical models in computer vision*, pp. 237–257. Springer, 2006.

- [115] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision.,” in *Proc. 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, vol. 2, pp. 674–679.
- [116] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, Aug. 1981.
- [117] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*, Ph.D. thesis, Massachusetts Institute of Technology (MIT), 2009.
- [118] A. Bruhn, J. Weickert, and C. Schnörr, “Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods,” *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.
- [119] A. Ghasemi, A. J. Scholefield, and M. Vetterli, “Consistent and optimal triangulation: Analysis and algorithms,” Tech. Rep., Institute of Electrical and Electronics Engineers, 2016.
- [120] C. Fredembach and S. Süsstrunk, “Colouring the near-infrared,” in *Proc. Color and Imaging Conference*, 2008, pp. 176–182.
- [121] I. Reda and A. Andreas, “Solar position algorithm for solar radiation applications,” *Solar energy*, vol. 76, no. 5, pp. 577–589, 2004.