

# Improved nowcasting of heavy precipitation using satellite and weather radar data

Jacob Schack Vestergaard

Kgs. Lyngby 2011  
IMM-M.Sc.-2011-38

DTU Informatics  
Department of Informatics and Mathematical Modeling  
Technical University of Denmark

Building 321, DK-2800 Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
reception@imm.dtu.dk  
www.imm.dtu.dk

M.Sc. Thesis: ISBN 978-87-643-0873-0  
ISSN 1601-233X

---

## Abstract

Global climate changes in recent years have caused a higher frequency of heavy precipitative events in Denmark, due to the increase in the atmospheric temperature. Therefore, a desire to nowcast these events has emerged. Nowcasting is the discipline of short term forecasting (0–3 hours) meteorological events and works on a smaller scale than the numerical weather models typically used for forecasting.

Six dates over the last few years (2007–2010) exhibiting extreme weather phenomena in Denmark have been selected, ranging from heavy snow fall to extreme downpour. Data used in the nowcasting come from the Meteosat-8 satellite and from weather radars operated by the Danish Meteorological Institute (DMI).

The supplied data are used for development of a nowcast system specifically designed for heavy precipitative events in Denmark. This includes a statistical approach to identification of ground truth using linear multivariate statistical methods, such as canonical correlation analysis. A method for learning a discriminative dictionary of satellite image patches is applied for classification and prediction of heavy precipitation.

An operational setting is simulated by use of leave-one-out cross validation, where the nowcast model is built on data from five dates and evaluated on the sixth. While the nowcasting abilities degrade when increasing the nowcast length above 0.5 hours, probably due to the diversity of the six weather situations, the method proves successful in classifying heavy precipitative events as they occur.





---

## Resumé

Over de seneste år er temperaturen i atmosfæren forøget og dermed også frekvensen af kraftige byger i Danmark. Et ønske om *nowcasting* af disse hændelser er derfor opstået. Nowcasting er forudsigelse af meteorologiske begivenheder kort tid inden de forekommer (0–3 timer) og opererer på en mindre skala end de numeriske vejrmødelles, der typisk bruges til vejrudsigt.

Seks dage inden for de sidste få år (2007–2010) er blevet udvalgt, da disse dage har indeholdt elementer af ekstremt vejr på en dansk målestok, fra kraftigt snefald i efteråret til ekstremt regnfald i sommervarmen. Til nowcasting bruges data fra Meteosat-8 vejr satelliten og radar data fra vejrradarer opereret af Danmarks Meteorologisk Institut (DMI).

Data fra disse hændelser bruges til udvikling af et nowcasting system, specifikt designet til kraftigt nedbør i Danmark. Analysen inkluderer en statistisk tilgang til identifikation af faktiske data ved hjælp af lineære multivariate statistiske metoder, såsom kanonisk korrelations analyse. En metode til læring af en diskriminativ basis af satellit billedpatches er anvendt til klassifikation og prædiktions af kraftig nedbør.

Et operationelt scenarie er simuleret ved brug af “leave-one-out” krydsvalidering, hvor nowcast modellen er bygget på data fra fem datoer og evalueret på den sjette. Modellens evner til prædiktions forringes for nowcast længder over en halv time, formentlig på grund af diversiteten i de seks vejr situationer. Dog viser metoden sig at være særdeles anvendelig til at klassificere kraftige nedbørshændelser, når de opstår.



---

## Preface

This master's thesis was written at DTU Space and DTU Informatics at the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Master of Science degree in Engineering, within the elite master's programme Industrial Mathematics.

The work is performed in collaboration with the Danish Meteorological Institute (DMI) and deals with the challenge of providing short term forecasting – nowcasting – of heavy precipitation in Denmark, with the aid of satellite and weather radar data.

This thesis was written under supervision of Associate professor Allan A. Nielsen (DTU Space), Professor Rasmus Larsen (DTU Informatics) and researcher Thomas Bøvith (DMI).

A toolbox for MATLAB has been developed and is included in Appendix A to ease any future work with the data set used in this thesis.

Besides this thesis, a poster has been produced and presented at the event Visiondays, at DTU Informatics in May 2011. The poster is included in Appendix D.

Kgs. Lyngby, June 2011

Jacob Schack Vestergaard



---

## Acknowledgements

A special thanks to researcher Thomas Bøvith and aviation meteorologist Birgitte Nielsen from the Danish Meteorological Institute. This project was only possible by their collection of data and meteorological know-how.

I would also like to thank my supervisors Allan A. Nielsen and Rasmus Larsen from DTU Space and DTU Informatics, for insightful discussions and guiding me through the process. I look forward to continuing our collaboration in the years to come.

At last I would like to thank Trine Abrahamsen, Søren Vestergaard, Rasmus Vestergaard and Lene Sommer for proof-reading and providing interesting perspectives on the Danish weather through the entire process.



---

# Contents

|   |            |
|---|------------|
| <b>Abstract</b>                                       | <b>i</b>   |
| <b>Resumé</b>   | <b>iii</b> |
| <b>Preface</b>  | <b>v</b>   |
| <b>Acknowledgements</b>                               | <b>vii</b> |
| <b>Contents</b>                                       | <b>ix</b>  |
| <b>1 Introduction</b>                                 | <b>1</b>   |
| 1.1 Meteorological background . . . . .               | 2          |
| 1.1.1 Atmospheric stability . . . . .                 | 2          |
| 1.1.2 What is convective rain? . . . . .              | 2          |
| 1.1.3 Types of convective systems . . . . .           | 4          |
| 1.2 Presentation of scenarios . . . . .               | 5          |
| 1.3 Previous work . . . . .                           | 7          |
| 1.4 Problem statement . . . . .                       | 9          |
| 1.5 Thesis layout . . . . .                           | 10         |
| <b>2 Data</b>   | <b>11</b>  |
| 2.1 Scenario hotspots . . . . .                       | 11         |
| 2.2 Satellite data . . . . .                          | 12         |
| 2.2.1 Specifications . . . . .                        | 12         |
| 2.2.2 Conversion to brightness temperatures . . . . . | 13         |
| 2.2.3 Projection . . . . .                            | 13         |
| 2.2.4 Spatial resolution . . . . .                    | 15         |
| 2.2.5 Map to image coordinates . . . . .              | 16         |
| 2.3 Radar data . . . . .                              | 17         |
| 2.3.1 General radar principles . . . . .              | 17         |
| 2.3.2 Specifications . . . . .                        | 19         |
| 2.3.3 Projection . . . . .                            | 19         |
| 2.3.4 Radar coverage mask . . . . .                   | 21         |
| 2.3.5 Accumulating radar data . . . . .               | 21         |
| 2.4 Choosing a common grid . . . . .                  | 22         |
| <b>3 Explorative analysis</b>                         | <b>25</b>  |
| 3.1 Subspace projections . . . . .                    | 25         |

## CONTENTS

|          |   |            |
|----------|---|------------|
| 3.2      | Principal Component Analysis (PCA)                                  | 27         |
| 3.2.1    | Applying to satellite data  | 27         |
| 3.2.2    | A generalizable subspace  | 29         |
| 3.3      | Maximum Autocorrelation Factor (MAF) analysis                       | 32         |
| 3.3.1    | Analysis results  | 33         |
| 3.4      | Canonical Correlation Analysis (CCA)                                | 35         |
| 3.4.1    | Univariate simplification   | 35         |
| 3.4.2    | Analysis results  | 36         |
| 3.5      | Spatial data correspondence   | 38         |
| 3.5.1    | Cross Variogram   | 38         |
| 3.5.2    | Cross correlation function  | 38         |
| <b>4</b> | <b>Collecting ground truth using cross correlation minimization</b> | <b>43</b>  |
| 4.1      | Exhaustive search   | 43         |
| 4.2      | Invariance to scale and rotation                                    | 48         |
| 4.3      | Simple segmentation and tracking in projection space                | 54         |
| 4.3.1    | Segmentation  | 54         |
| 4.3.2    | Simple tracking   | 56         |
| 4.3.3    | Collecting features   | 58         |
| 4.4      | Collected ground truth  | 60         |
| <b>5</b> | <b>Nowcasting using learned dictionaries</b>                        | <b>63</b>  |
| 5.1      | Learning a discriminative dictionary                                | 64         |
| 5.2      | Classification  | 66         |
| 5.3      | Prediction  | 69         |
| 5.4      | Obtaining a generalizable error                                     | 71         |
| 5.4.1    | Partition of data   | 71         |
| 5.4.2    | Cross validation  | 72         |
| 5.4.3    | Dictionary method test errors for all scenarios                     | 73         |
| 5.5      | Comparison with logistic regression                                 | 79         |
| 5.5.1    | Test errors   | 79         |
| <b>6</b> | <b>Discussion &amp; future work</b>                                 | <b>81</b>  |
| 6.1      | Future work   | 82         |
| <b>7</b> | <b>Conclusion</b>   | <b>85</b>  |
|          | <b>Bibliography</b>   | <b>87</b>  |
| <b>A</b> | <b>Data handling toolbox</b>  | <b>93</b>  |
| A.1      | Projection methods  | 96         |
| A.2      | Radar data methods  | 100        |
| A.3      | Satellite data methods  | 103        |
| A.4      | Visualization methods   | 108        |
| A.5      | Toolbox demos   | 110        |
| <b>B</b> | <b>Illustrations of data</b>  | <b>113</b> |
| B.1      | Radar data at hotspot   | 113        |
| <b>C</b> | <b>Supplementary Results</b>  | <b>117</b> |
| C.1      | Principal Components  | 117        |



|          |   |            |
|----------|---|------------|
| C.2      | Global PCA . . . . .  | 123        |
| C.3      | Maximum Autocorrelation Factors . . . . .                             | 125        |
| C.4      | Correspondence surfaces . . . . .                                     | 131        |
| C.5      | Exhaustive search for ground truth . . . . .                          | 134        |
| C.6      | Invariant tracking to collect ground truth . . . . .                  | 137        |
| C.7      | Transformed radar data at time of minimum cross correlation . . . . . | 143        |
| C.8      | Simplex method statistics . . . . .                                   | 146        |
| C.9      | Simple tracking results . . . . .                                     | 148        |
| C.10     | Dictionary parameters . . . . .                                       | 150        |
| C.11     | Dictionary size . . . . .   | 152        |
| C.12     | Box plots for dictionary method results . . . . .                     | 153        |
| C.13     | Validation error box plots for dictionary method . . . . .            | 155        |
| C.14     | Test errors for dictionary method using global CCA . . . . .          | 157        |
| C.15     | Logistic regression . . . . .   | 158        |
|          | C.15.1 Validation error box plots . . . . .                           | 158        |
| C.16     | Nowcast maps . . . . .  | 159        |
|          | C.16.1 0h nowcast . . . . .   | 159        |
|          | C.16.2 0.5h nowcast . . . . .   | 159        |
|          | C.16.3 1h nowcast . . . . .   | 171        |
| <b>D</b> | <b>Visiondays poster</b>  | <b>177</b> |



## Introduction

With recent years increased focus on global climate changes, a desire to understand and explain extreme weather phenomena has emerged. It is believed that the frequency and severity of extreme rain in Denmark has increased due to the warmer atmosphere providing the necessary heat for thunder cells to develop. The rapid development and spontaneous nature of these events provide challenges to larger scale numerical weather models in terms of prediction. Therefore, it is necessary to develop models specifically tuned for these situations.

Extreme precipitative events have great consequences in many areas, such as farmers' crops getting damaged from the rain or roads flooding and damaging cars. Many of these cases cannot be avoided with a better prediction of cloudbursts, but other areas, such as air traffic control and flood management, where decisions are made based on the next few hours expected weather, can benefit from an improved *nowcasting*.

Nowcasting is the field concerned with short term forecasting, i.e., predicting events within 0-3 hours prior to occurrence. In this thesis the events of interest are extreme weather events, more specifically isolated areas of heavy precipitation. For this purpose, a data set consisting of satellite and weather radar data from six dates over the last few years (2007–2010) have been collected. Common for these six dates are that extreme weather phenomena have been recorded.

The goal of this thesis is to improve nowcasting of such events by use of meteorological satellite data. It is expected that these weather events can be identified earlier in satellite imagery than in radar data, as a weather radar register reflectance of rain in the air, which is too late in a predictive context. The satellite data, on the other hand, can provide a constant view of cloud top temperatures in the atmosphere above Denmark and – hopefully – provide the capability of detecting cloud development patterns in an early stage.

There are a number of challenges involved in solving this problem. First of all, these particular data have not been analyzed jointly before and there are no guarantees that they are in fact similar enough to be part of an ensemble analysis. A consequence of this being a new data set is also that there exist no ground truth, which must first be collected. Thus, it will not be possible to compare achieved results with benchmarks from previous work, but the quantitative measures will be seen in a perspective of possible operational use, as this is an attempt to solve a real world problem.

## 1.1 Meteorological background

This section will provide an introduction to the meteorological components related to the development of heavy precipitation. Most textbooks on meteorology cover these areas, though the main source used here is [37].

Two scales are used here when talking of convective systems: the synoptic scale, which includes most fronts, low and high pressure systems, i.e., the large scale weather phenomena, and the smaller mesoscale. Analysis on the mesoscale considers cloud developments on a horizontal scale of two to a few hundred kilometers, while the synoptic scale is above 1000 kilometers. The thunderstorms analyzed in this thesis are mesoscale convective systems.

### 1.1.1 Atmospheric stability

The atmosphere is the air surrounding the earth and is kept in place by gravity. The atmosphere can be divided into layers, separated by so-called pauses. The lowest layer is the troposphere, above which is the tropopause in about 10 kilometers height above the earth. Precipitation fall from cloud bases below 2000 meters, but the highest clouds have tops reaching the tropopause. Temperature decrease with height in the troposphere, while in the next layer -0- the stratosphere – temperature increase with height. A sketch of the different layers in the atmosphere and the relation between height and temperature is shown in Figure 1.1.

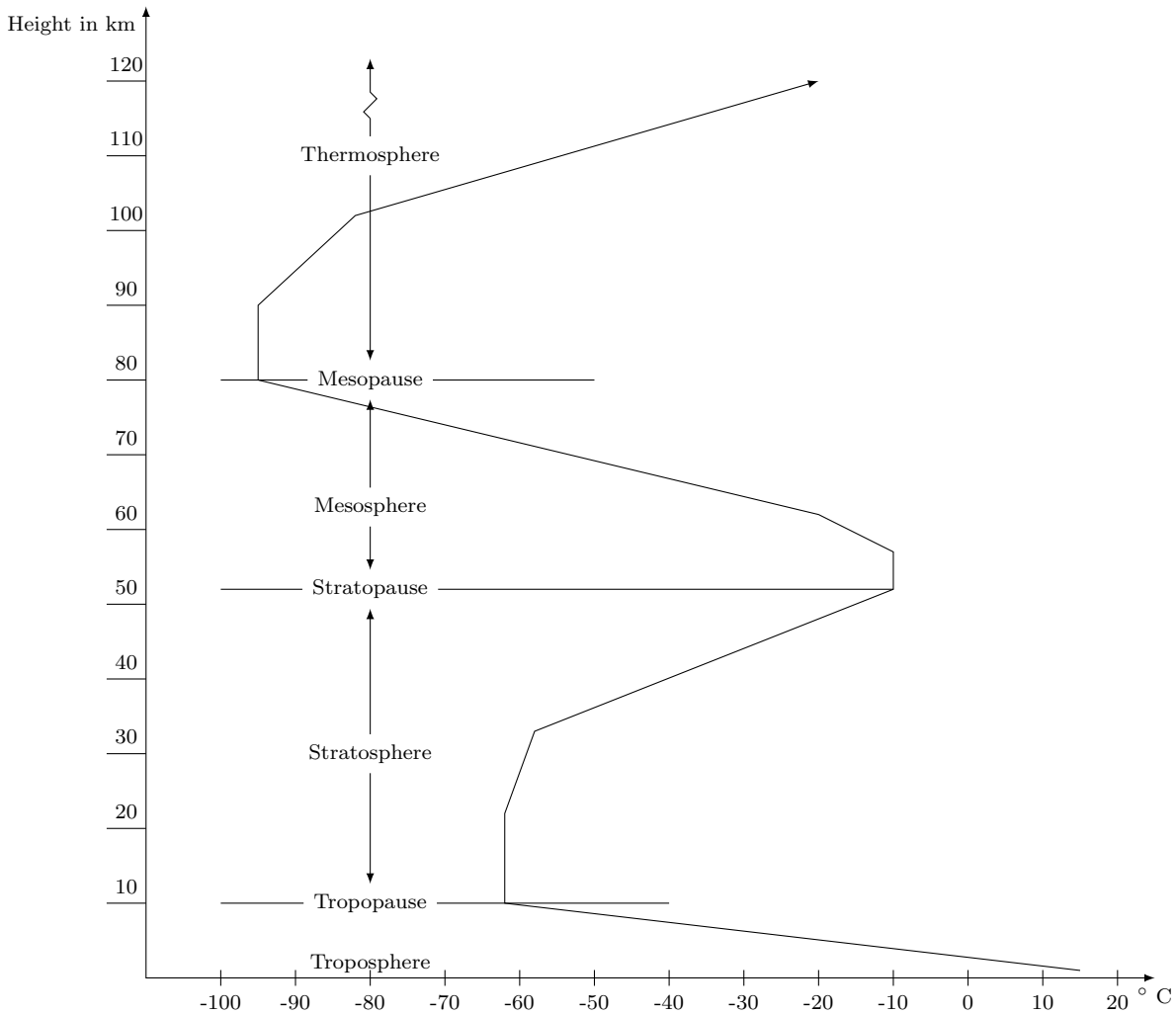
Heavy precipitation in Denmark is usually formed in the summer as deep convective cells, when the heating of the air has caused the atmospheric instability to increase. Atmospheric instability causes the warm air to rise as bubbles from the ground – or water – without any forcing, such as mountains or fronts. When air bubbles rise from the surface into the troposphere, the air pressure around the bubble will drop and cause the air bubble to take energy from itself in form of heat in order to equalize the pressure. Hence, the temperature in the bubble decrease, without exchange of energy with the surrounding air, i.e., it is an adiabatic process [37]. Atmospheric instability is an important prerequisite for deep convective rain cells to form, as the spontaneous development of convective rain does not occur in stable air. Radio explorations are used to measure atmospheric stability, but due to the low density of these, so-called pseudo-temps are estimated from other data sources, where satellite data is only one of them.

### 1.1.2 What is convective rain?

Transfer of heat is possible by four fundamental mechanisms [96]: Conduction, convection, radiation and mass transfer. Convection is the transfer of heat through fluids by molecular motion.

In meteorology, convection refers to the rising of air in the atmosphere. Furthermore, it is usually implicit that it is moist convection, i.e., where water vapor in the rising air condenses to form a cloud. Heat is generated from condensation, which can add to the convection, whereby the process feeds itself. Dry convection also occur, but is not visible since no clouds are formed.

Convection is one of the four ways that precipitating clouds are formed [37]. Friction from stable air moving over a rough surface causes turbulence, and, with enough wind pressure, the turbulence will throw the air high enough into the atmosphere for it to reach its dew point. This

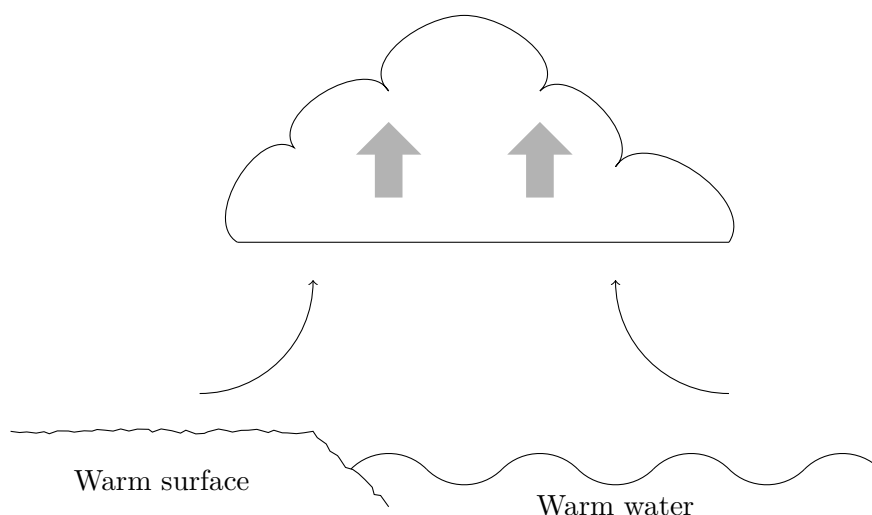


**Figure 1.1:** Illustration of the temperature variation through the atmosphere's layers. Inspired by [37].

is called turbulence precipitation. Orographic precipitation is formed when stable air is moving over mountains or hills, forcing the air high enough to reach its dew point. Precipitation is also formed, when two masses of different temperatures meet and form a weather front. The warm air will be lifted above the cold air and clouds will form in the rising air, i.e., convection. Different types of precipitation can be formed, e.g., heavy showers or continuous rain.

Even though convection also is a part of front precipitation, convective rain will in this context refer to rain caused by spontaneous vertical rising of air in an unstable atmosphere, causing lower temperature in the air and eventually condensation. The clouds formed in this process are cumulonimbi clouds. If the air is very unstable, the cloud will – as mentioned above – feed on itself and continue to tower. Inside the cloud, powerful airflows will bring ice crystals and water droplets further up in the cloud, causing them to grow. When the airflows are no longer powerful enough to lift them, they will drop from the cloud as showers or squalls. An anvil can be formed on top of a cumulonimbus, as the convection stops in the tropopause and the cloud top will take on a larger horizontal extent than the bottom.

Atmospheric instability is one of the prerequisites for thunderstorms to develop. Another is some mechanism to cause a lift of the air. This lift can happen on a synoptic scale in – or in front



**Figure 1.2:** *Illustration of warm surfaces causing air to rise and condense in form of a cumulus cloud. Within the cloud, unstable air can make the cloud tower into a cumulonimbus and eventually forming an anvil*

of – a cold front or on a meso scale by heating of surfaces and the lowest layer of air. The last important part is moisture – the more moisture, the more convection and thus increasing the severity of the thunderstorm.

### 1.1.3 Types of convective systems

Convective systems usually carrying heavy precipitation in Denmark can be divided into three main categories: Cold air thunder, heat thunder and embedded cumulus nimbus. The first two are very similar in a meteorological sense and the convective systems analyzed in this thesis belong to these two categories. While mostly similar, they still differ on some areas and their characteristics will be described here:

- **Cold air thunder** is usually brought by the cold air from the northwest, called polar air. Sometimes they develop a distinct shape as a grammatical comma wherefore they are called “comma clouds”. When the cold air is from due north, the air has been dried out over Norway, wherefore there is not enough moisture to cause a thunderstorm.
- **Heat thunder** develops when a moist, instable mass of air is warmed enough to start the convection. This typically occurs in air flow from the south in summer time, where they sometimes develop over Germany and is brought north to Denmark and in other cases develop when they reach southern Denmark. The heat thunder is typically much more severe in terms of precipitation, as the warm air can contain more moisture than cold air.

While these two categories are more correctly termed thunderstorm in cold/warm airmass, respectively, they will for brevity be referred to as cold air and heat thunder.

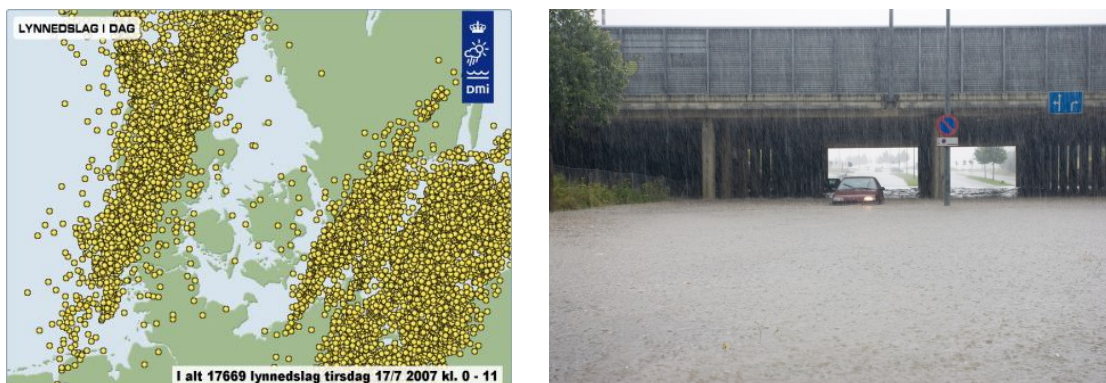
Embedded cumulus nimbus is a much more complicated situation, usually having larger areas of downpour than the two others. The extreme convection are embedded inside these systems and can be caused by cooling from a jet stream, movement over a hotter surface or a variety of other

phenomena at the same time. Furthermore, the embedded cumulus nimbus does not form as suddenly as the two other phenomena, as a large convective system has already formed.

## 1.2 Presentation of scenarios

Six scenarios have been identified as interesting for this project by meteorologist at the Danish Meteorological Institute (DMI), Birgitte Nielsen. Each of these scenarios have elements of extreme precipitation, compared to Danish weather standards for the season in which they occur. As they have all been chosen for joint analysis in this thesis, they share certain properties, where extremity compared to the surrounding weather is the primary one. However, they do not represent completely similar meteorological synoptic situations and vary in lightning activity, spatial extent, temporal extent and downpour intensity. A brief description of every scenario is found below, where each convective system's development is summarized and their individual properties highlighted.

**2007-07-16** After a period with many powerful squalls, a heavy rainfall hit Jutland on the night between July 16th and 17th. Heat thunder storms developed in front of a cold front moving in from the southwest of Jutland at approximately 22.00 and left the northeastern Jutland about 6 hours later. On its passing, it had caused downpour intensities above 50mm per hours, hail and lightning frequencies of up to 50 lightnings per minute [60]. The lightning intensity in the cold front is especially what makes this event exceptional: 5000 cloud-to-earth and as many cloud-to-cloud lightnings were recorded, causing reports of damaged crops and cars [32]. Illustrations from this event is seen in Figure 1.3.



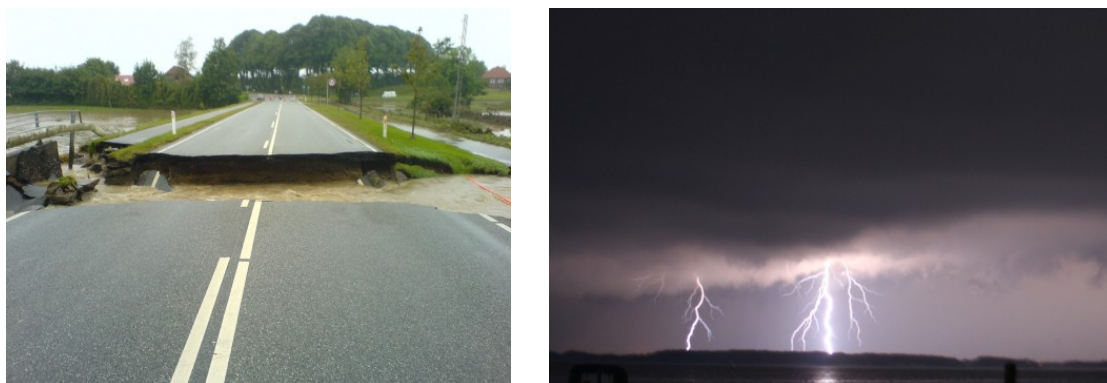
**Figure 1.3:** (left) Registered lightning strikes from the thunderstorm over Jutland on July 16, 2007. (right) Flooded road near Brøndby after cloudburst on August 11, 2007. Sources: [32, 31].

**2007-08-11** Warm and moist air over the eastern part of Denmark and southern Sweden developed into heat thunder on August 11th 2007. While most of the convective system was situated over Sweden, downpour intensities up to 60mm per hour were recorded in northern Zealand. Thunder and lightning activity was also recorded, but the downpour intensities were the most significant part of the event. Locally, the cloudbursts lasted approximately 2 hours and the entire convective system had a lifetime from about midday until 19.00, wherein it had caused floods in several locations [11, 10].

## CHAPTER 1. INTRODUCTION

**2007-08-20** This is probably the most famous of the scenarios in the data ensemble. Also, it is the only of the scenarios where the downpour has been described as *extreme*. The heat thunder system developed in the moist, warm air in mid August and is described as a multi cell convective system. The thunder system spontaneously develops from one of many precipitating clouds coming from northern Germany and matures over Sønderborg, where it stays for approximately 2 hours and further matures causing heavy precipitation. While lingering over Sønderborg and Gråsten, multiple cells are formed on the sides of the matured and precipitating cells. The entire precipitating event passed in about 3 hours, from 20.00 to 23.00, where downpour intensities up to  $88.32\mu\text{m}/\text{sek}$  over a 10 minute interval was recorded. This corresponds to approximately 53mm in this 10 minutes time window alone! The spatial extent of this event was very small, constrained to the area around Gråsten in southern Jutland and very isolated, i.e., not part of a front or a larger system.

Due to the unusual weather development for this scenario and extreme proportions of downpour, the meteorological synoptic situation is analyzed in depth in [71], where eye witness reports are also printed. The heat thunder system also brought hail and lightning activity, but the materialistic consequences were from the rain and included damaged roads, train tracks and local flooding [12].



**Figure 1.4:** (left) Damaged main road near Sønderborg and (right) lightning from thunder cell from cloudburst on August 20, 2007. Sources: [61, 74].

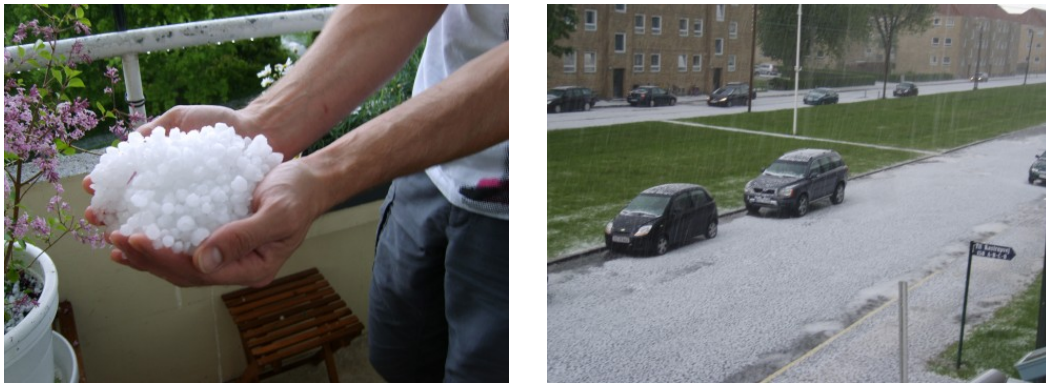
**2007-11-11** This scenario is less described than the others, which is probably due to the precipitation falling as snow rather than rain. This convective system is an example of cold air thunder, where snow, hail and lightning activity develop when cold air meet water, still warm from the summer. This meeting develops enough heat to create the needed atmospheric instability [87]. The cold polar air move in over Jutland from the northwest to form a comma cloud, which loses its power after a few hours and continues weakened towards southeast. The convective system forms at approximately 23.30 on the 11th of November and leaves three to four hours later. This scenario is the least extreme of them all, which is probably due to the forming of snow, but also the fact that cold air thunder systems do not reach the same intensities as heat air thunder, though still being extreme for the season.

**2009-05-18** On May 18th 2009, thunder was build from an unstable layered troposphere, where the needed temperature for instability release came from the sun's heating of the ground. A multi cell convective system developed over the northern part of Jutland causing hail, thunder and even a small tornado was observed. A squall line is formed in the cold air moving in from the



southwest and is maturing during the afternoon, before it results in precipitation and disappears over water northeast of Jutland at approximately 19.00, eight hours later. As with the other cold air thunder scenarios, the downpour intensity does not quite reach that of the heat thunder scenarios [65].

**2010-06-15** The most recent scenario is from June 15th 2010, where showers form in cold air over southern Sweden at the most eastern part of Zealand. Again, the heating of the ground contributes to warming the lowest layers of the atmosphere, creating the atmospheric instability needed to form thunder cells. As the sea water is still cold at this time of year, the main activity is over land. The convective system starts to form in the morning at approximately 9.00, where the sun has started to warm the ground and moves towards south, where it in the afternoon reaches water at about 16.00 and terminates. This scenario distinguishes itself from the others as consisting more of clusters of showers, rather than an isolated event [65]. Large amounts of hail and heavy downpour in form of rain was the result of this convective system. The threshold for cloudburst (15mm per 30min.) was in some areas reached, which is a lot for the season, though not reaching the heat thunder scenarios [33].



**Figure 1.5:** (left) Large hails from cloudburst on May 18, 2009. (right) Hails bringing a winterly look to Amager on June 15, 2010. Sources: [72, 80]

### 1.3 Previous work

Weather forecasting in general has been an area of intense studies for centuries. Meteorology as a science is the study of objects in the sky, usually related to weather, and is an integrated part of most people's everyday life. Meteorologists are employed in many fields, such as aviation, shipping industry, event planning and research. Previous work within the field of meteorology and weather forecasting is therefore extensive and the areas widespread. Study areas interesting in relation to this thesis include nowcasting in general, applications of satellite data for forecasting purposes, cloud identification and tracking, and automated forecasting of heavy precipitation in general. A brief review of relevant publications will be given here.

A description of different types of convective systems can be found in [39], where the clouds are categorized – using satellite and radar data – from temperature, area, cell arrangement, eccentricity and temporal development.

Studies specific to identification of convective clouds span a wide variety of methods, from pure meteorological criteria to the more machine learning oriented clustering techniques. For

## CHAPTER 1. INTRODUCTION

instance, mesoscale convective systems (MCSs) are identified and tracked by use of two simple meteorological criteria in [94]. In the following stage, a data mining approach is used to identify the MCSs with potential of flooding the Yangtze River. A deterministic method for detecting convective initiation (CI) in infrared (IR) imagery is described in [56], where thresholding IR bands and differences between selected IR bands are used to employ a scoring system for clouds, such that if a cloud fulfills a certain number of criteria it is characterized as a rapidly growing cumulus in a pre-CI state. Three different cases are used to evaluate the method. Statistical downscaling is used in [28] to derive a transfer function between a large-scale weather model and a small-scale. PCA and CCA are used to create the transfer function.

Segmentation of clouds are achieved through a scale space approach to classification in [63], such that objects of similar intensity and scale are clustered together. Fuzzy c-means (FCM) is used as clustering technique. Clouds are tracked by (i) identifying the area of homogeneous cloud mass, (ii) selecting feature points on the contour of tracked cloud, and (iii) generating cloud motion vectors from these points. More specific properties of convective clouds have also been studied, such as the occurrence of "plumes" on top of convective storms in [49], where it is investigated using satellite imagery.

The satellite data in this thesis are from the Meteosat weather satellites. With the launch of Meteosat Second Generation (MSG), a status was done in [48] on estimating precipitation using satellite data and prospects for using the new data. This includes thresholding of brightness temperature differences between IR bands. Data from this type of satellite have been treated by [59], where a study is conducted on the identification of convective initiation using the infrared channels from Meteosat Second Generation imagery. By use of interest fields [56, 57] descriptive statistics on cloud depth, glaciation indicators and updraft strength are extracted. Principal component analysis is used to identify the significant fields. Similarly, the bands in the visible range have been studied in [58].

Satellite data from other missions have been treated for various purposes. Atmospheric motion vectors are extracted from Kalpana-1 imagers in [44] using image thresholding to separate cloud layers and a cross-correlation method for tracking the clouds between images. A comparison of high-resolution imaging capabilities of the different satellite missions is done in [40].

Weather radar data have also been used for detection and classification. An object-oriented description of convective cells is employed in [30]. The convective clouds are identified and their attributes, such as stage, are set using heuristics based on e.g., radar reflectivity. Cells in certain stages are allowed to initiate daughter cells, which emphasizes the object-oriented approach to nowcasting. Another approach for classifying storm type from weather radar reflectivity is the use of decision trees [29]. K-means clustering is applied to the reflectivity images in order to identify similar storm regions. Morphological attributes, such as major axis length and eccentricity of the clouds, combined with reflectivity statistics, wind statistics and weather system properties are used as input to the decision trees.

The concept of image fusion and various methods for this are described in [75]. This includes color transformations, statistical methods and an evaluation of each method. Different data sources are outlined in [84], e.g., lightning data and sounding data, and how they can contribute to the knowledge of MCS movement.

Descriptions of running products can also be found in literature. For instance, the underlying model for a running product, called NEFODINA, is presented in [77]. The model has four main phases: automated cloud-cluster detection, evaluation of some important parameters of these

clouds, discrimination of the convective cells, i.e., thresholding and forecast the cells' development. Another algorithm for detection, tracking and nowcasting called ForTraCC is presented in [89]. The detection is done by a simple thresholding in one IR band, with a size restriction. This algorithm is applied in [55], where cloud-to-ground lightning detection and life cycles of MCSs are analyzed. A description of the Convective Diagnosis Oceanic (CDO) product is found in [43]. This product is comprised of three different algorithms for identification of convective clouds: (i) a classifier using training data, (ii) a cloud top height estimator using IR brightness temperature and a vertical profile from the Global Forecasting System (GFS) and (iii) an identification of deep convection from IR brightness temperature differences.

Short term (0-1 hour) nowcasting is treated in [85], where an existing product is used for identification of cloud types, and a larger data set of 23 convective afternoons is evaluated. It is noted that automated procedures to collect ground truth failed, wherefore this was collected manually.

The more recent relevant publications include nowcasting the weather at the Beijing Olympics 2008 using radar data [92], where several techniques are compared. The Meteosat satellite data are used for nowcasting in [35], but using ground truth collected twice per hour by a Dutch airport.

Generally, the methods rely on a great amount of manual choices of clouds of interest, thresholds and existing products. Only in a few cases, multivariate statistical methods are used and when they are, they work primarily on collected features rather than on the raw imagery. Furthermore, the methods are developed for other geographical regions and weather systems than Denmark. An important point is given in [86], which is a review of nowcasting methods and comparison techniques. Here it is emphasized that objective scoring schemes should be used, that nowcasting methods should be statistical in nature, and that they should be designed specifically for the area in which they are applied. These guidelines are going to be the drivers of some of the methodological decisions made in this thesis.

## 1.4 Problem statement

This thesis is concerned with the problem of nowcasting extreme precipitation in Denmark. Nowcasting is here short term (0-3 hours) forecasting of heavy precipitation. The main goals of this thesis are:

1. To process satellite and radar imagery from six one-day time series, such that they can be part of a joint analysis,
2. Identify areas in collected satellite imagery, where events of this type are present, i.e., collect ground truth, with the aid of radar data,
3. Develop an automated method for identification of heavy precipitation, and
4. Evaluate the nowcasting capabilities of such method using objective quantitative measures.

The work in this thesis is unique, in the sense that the combination of methods has been chosen solely for this purpose, the six time series have not previously been analyzed and that the developed methods for collecting ground truth can be applied to new data of the same form, as they are almost completely automated. Furthermore, a method for creating a nowcast aimed specifically at extreme precipitation in Denmark has not previously been published.

## 1.5 Thesis layout

A short overview of this thesis' contents is provided here:

- In Chapter 1 a motivation for solving the problem was given, as well as a brief meteorological background. Furthermore, a literature review on selected topics within nowcasting and other areas related to the work in this thesis, is given.
- Chapter 2 provides an overview of the sources of data and preprocessing of these.
- The analysis commences in Chapter 3 with an explorative analysis of the data, using selected multivariate methods, and a basis is built for the following choices of methods.
- Ground truth is collected in Chapter 4, where maximizing correspondence between extreme precipitative events in the radar data and the satellite data, allows for determination of a subspace in which simple segmentation and tracking can be performed.
- Finally, classification of extreme events is done in Chapter 5 by using a method for learning discriminative dictionaries of image patches. An attempt to use this method for prediction – nowcasting – is also made and a quantitative comparison with logistic regression is performed.
- The decisions made throughout the thesis, as well as the available data foundation, are discussed in Chapter 6, where possible extensions are also considered before drawing conclusions in Chapter 7.
- Appendix A provides a MATLAB toolbox for handling the radar and satellite data used in the thesis.
- Appendix B provides exemplifying illustrations of the provided data.
- For completeness, results for all six scenarios are provided in Appendix C, as only one or two examples are given when presenting results throughout the thesis.
- A poster was produced as part of this thesis and presented at Industrial Visionday 2011 and is included in Appendix D.

## Data

The data analyzed in this thesis have been delivered by DMI and consist of six scenarios of extreme weather. For each extreme event, time series of image data for the entire day have been provided. In cases where the event is close to midnight, two days of image data are provided. Effectively, each scenario has 24 or 48 hours of data associated. Each scenario’s synoptic situation is described in Section 1.2. The data specifications, such as temporal and spatial resolutions will be described in Sections 2.2 and 2.3.

Throughout the thesis, all indications of time will be in Coordinated Universal Time (UTC).

### 2.1 Scenario hotspots

As each scenario consists of a time series, the first task is to identify the point in time for each scenario, where the actual extreme event occurs. Through this thesis, this is going to be used for various purposes. This point in time will be referred to as the “hotspot” for the scenario. For each scenario, the hotspot was identified manually by observing image sequences of the raw data. Especially the radar data were useful in this context. A simple automated procedure for this was also attempted to be developed, based on simple statistics through the time series, but due to radar clutter [8] some anomalies were observed and a manual verification was needed. The chosen hotspot for each scenario can be seen in Table 2.1.

| Scenario             | Thunder<br>type | Hotspot          | Satellite |                  | Radar |                  |
|----------------------|-----------------|------------------|-----------|------------------|-------|------------------|
|                      |                 |                  | $N$       | $\Delta t$ [min] | $N$   | $\Delta t$ [min] |
| July 16-17, 2007     | Heat            | 2007-07-16 23:00 | 192       | 15               | 288   | 10               |
| August 11, 2007      | Heat            | 2007-08-11 15:12 | 96        | 15               | 144   | 10               |
| August 20, 2007      | Heat            | 2007-08-20 20:57 | 96        | 15               | 143   | 10               |
| November 11-12, 2007 | Cold            | 2007-11-11 21:40 | 192       | 15               | 288   | 10               |
| May 18, 2009         | Cold            | 2009-05-18 15:30 | 288       | 5                | 144   | 10               |
| June 15, 2010        | Cold            | 2010-06-15 13:04 | 288       | 5                | 144   | 10               |

**Table 2.1:** Overview of scenarios. Type of thunder, chosen hotspot and details on supplied data. The number of images  $N$  and the sampling time  $\Delta t$  varies between scenarios.

## 2.2 Satellite data

The supplied satellite data are multispectral imagery from the weather satellite Meteosat-8, previously known as MSG-1. The Meteosat satellites were developed by the European Space Agency (ESA), and are now operated by EUMETSAT<sup>1</sup>. EUMETSAT is DMI’s supplier of satellite imagery and related products.

From a geostationary orbit at 0° longitude the satellite originally delivered imagery in a 15 minutes temporal resolution. With the introduction of Meteosat-9, Meteosat-8 was in May 2008 relocated to 9.5° longitude to work as backup for the new Meteosat-9. While working as backup it has supplied Rapid Scan Service (RSS), meaning that the newer data (post 2008) have a finer temporal resolution of 5 minutes.

### 2.2.1 Specifications

The raw data from Meteosat-8 are termed “Level 1.0” data. The data analyzed in this thesis have been pre-processed by EUMETSAT and are called “Level 1.5” image data. Pre-processing includes geolocating, radiometric correction and other adjustments making the data ready to use for meteorological applications. Needed projection and calibration parameters are stored in the image metadata. Specifications for this image format can be found in [24, 26].

The onboard multispectral camera delivering the imagery is of the type SEVIRI<sup>2</sup>. It is equipped with 12 spectral channels, hereof 8 infrared (IR) and 4 visible (VIS) channels. The spectral range of the VIS channels are 0.4 – 1.6 $\mu\text{m}$  and of the IR channels 3.9 – 13.4 $\mu\text{m}$ . One of the VIS channels is a high resolution panchromatic channel with a resolution of 1  $\times$  1km at the subsatellite latitude, i.e., Equator, while the remaining eleven channels have a ground sampling distance of 3  $\times$  3km. Further multispectral channel specifications can be seen in Table 2.2.

| Channel no. | Channel name | Characteristics of spectral band ( $\mu\text{m}$ ) |                        |                        | Main gaseous absorber or window |
|-------------|--------------|--|------------------------|------------------------|---------------------------------|
|             |              | $\lambda_{\text{cen}}$                             | $\lambda_{\text{min}}$ | $\lambda_{\text{max}}$ |                                 |
| 1           | VIS0.6       | 0.635  | 0.56                   | 0.71                   | Window                          |
| 2           | VIS0.8       | 0.81   | 0.74                   | 0.88                   | Window                          |
| 3           | NIR1.6       | 1.64   | 1.50                   | 1.78                   | Window                          |
| 4           | IR3.9        | 3.90   | 3.48                   | 4.36                   | Window                          |
| 5           | WV6.2        | 6.25   | 5.35                   | 7.15                   | Water vapor                     |
| 6           | WV7.3        | 7.35   | 6.85                   | 7.85                   | Water vapor                     |
| 7           | IR8.7        | 8.70   | 8.30                   | 9.10                   | Window                          |
| 8           | IR9.7        | 9.66   | 9.38                   | 9.94                   | Ozone                           |
| 9           | IR10.8       | 10.80  | 9.80                   | 11.80                  | Window                          |
| 10          | IR12.0       | 12.00  | 11.00                  | 13.00                  | Window                          |
| 11          | IR13.4       | 13.40  | 12.40                  | 14.40                  | Carbon dioxide                  |
| 12          | HRV          | Broadband ( $\sim$ 0.4 – 1.1)                      |                        |                        | Window/water vapor              |

**Table 2.2:** SEVIRI channel specifications [83].

<sup>1</sup>European Organisation for the Exploitation of Meteorological Satellites

<sup>2</sup>Spinning Enhanced Visible and Infrared Imager

### 2.2.2 Conversion to brightness temperatures

In order to provide the satellite data in 10 bit integer format, a linear scaling has been performed for each image [24]. The calibration coefficients (slope and offset) has been stored alongside each image, such that the raw counts,  $x_c$ , can be converted back to effective radiance,  $L^{15}$ . Effective radiance is defined as

$$L^{15} = \frac{\int L_v r_v dv}{\int r_v dv} \quad (2.1)$$

where  $r$  is the instrument spectral response,  $L$  is the Level 1.0 data and subscript  $v$  indicates the centre wavelength for the specific channel [25]. The effective radiance can be estimated from the scaled counts, using the calibration slope and offset

$$\tilde{L}^{15} = cal_{\text{slope}} \cdot x_c + cal_{\text{offset}} \quad (2.2)$$

Finally the brightness temperatures  $T_b$  can be estimated using the inversion of Planck's law [26]:

$$\tilde{T}_b = \frac{c_2 \cdot v}{\ln \left[ 1 + v^3 \frac{c_1}{L^{15}} \right]}, \quad v = \frac{10^4}{\lambda_0} \quad (2.3)$$

where  $\lambda_0$  is the centre wavelength for the specific channel.

### 2.2.3 Projection

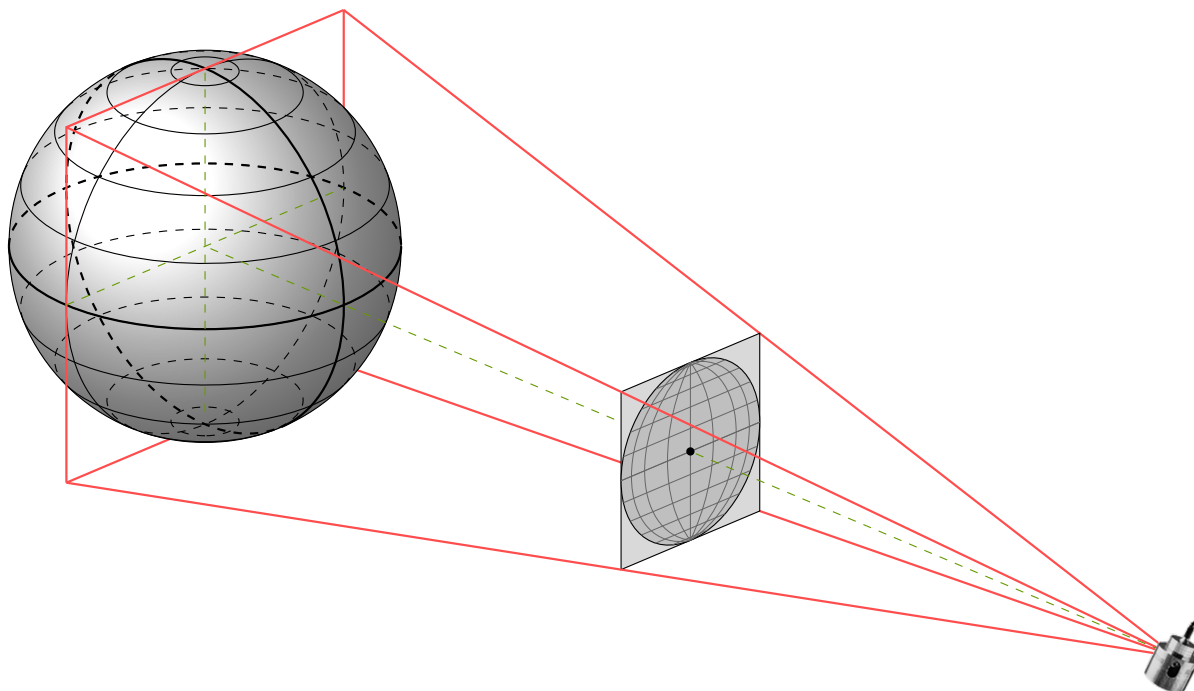
The Level 1.5 image data are delivered in a Normalized Geostationary Projection (GEOS) defined in [93]. This projection maps half of the globe to a disk in the image plane. This is illustrated in Figure 2.1.

The projection is defined in terms of the subsatellite longitude  $\text{lon}_0$ , spheroid major axis  $a = 6378169.0\text{m}$ , minor axis  $b = 6356583.8\text{m}$ , and satellite height  $h = 35785831\text{m}$ . The subsatellite longitude is read from the image metadata, as it varies between scenarios, while the others are constants.

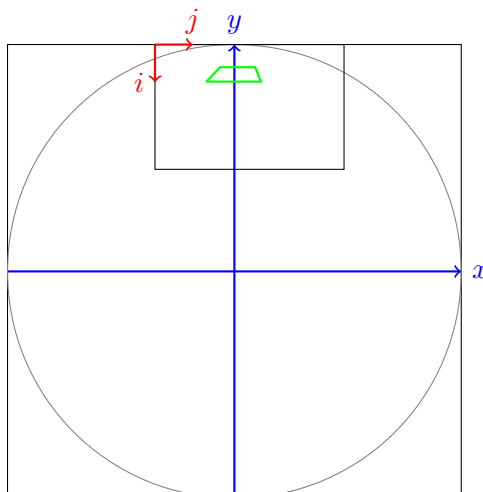
To calculate the projection of map coordinates to geographic coordinates, and vice versa, the "PROJ.4 – Cartographic Projections Library" is used [27]. A so-called *proj-string* is used to call the library, which will be supplied here for completeness. This example projects GEOS map coordinates to geographic coordinates, i.e., the inverse GEOS projection:

```
> proj +a=6378169.0 +h=35785831.0 +lon_0=0.0 +b=6356583.8 +proj=geos -I
> 623323.09          4790504.03
   10d30 '46.8"E     54d51 '10.8"N
```

The original image data from the satellite resides in an image coordinate system of  $3712 \times 3712$  pixels. The acquired data is a subset of this original data and is bounded by a rectangle with pixel coordinates specified in the image metadata. This is illustrated in Figure 2.2. The satellite data contain large areas of Southern Europe and Northern Africa, which are not necessarily interesting in this context. To reduce the amount of data prior to analysis, a smaller geographical region has been selected. This region is specified such that it covers at least as much as the radar data. First it is defined to be the geographical region within the corners with coordinates (50N,0E) and (62N,22E). This is a rectangle in geographical coordinates, but not necessarily in image coordinates, which is needed to crop the satellite image. To accomplish this, it is chosen



**Figure 2.1:** *Conceptual drawing of the GEOS projection.*

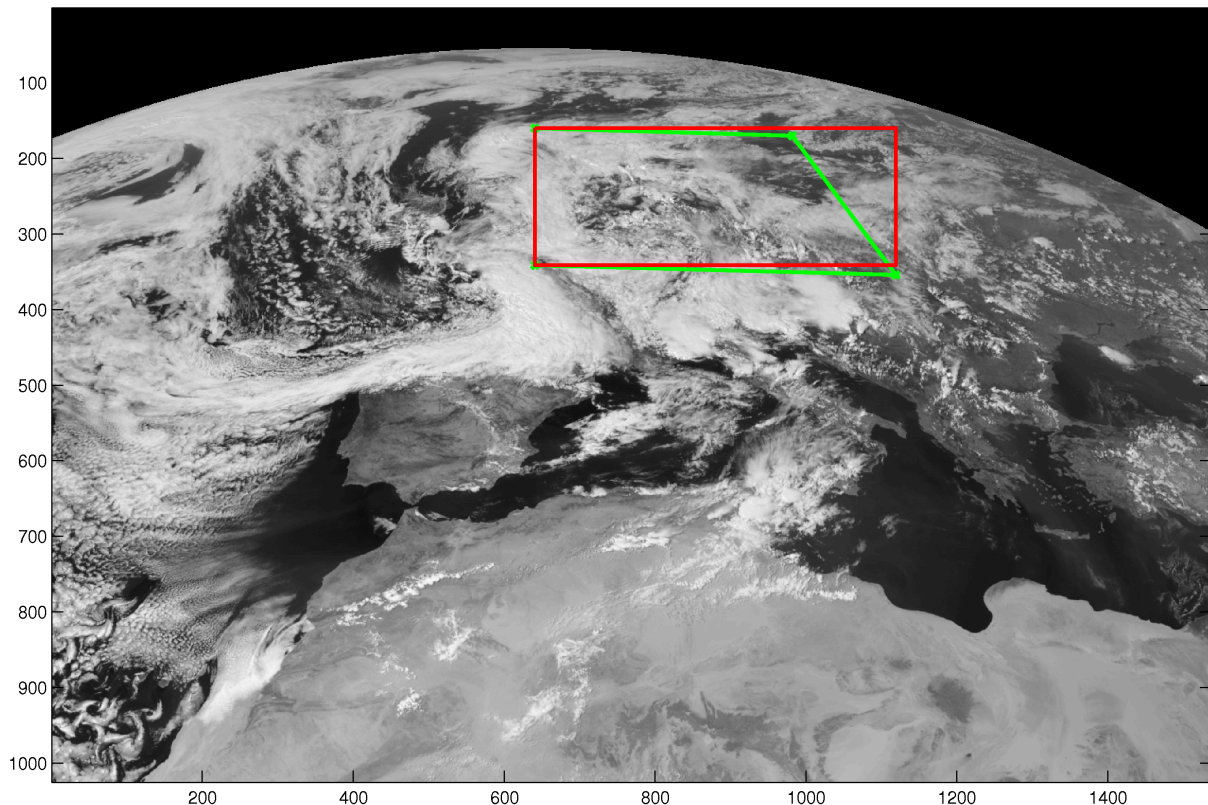


**Figure 2.2:** *The relationship between map coordinates  $x/y$  and image coordinates  $i/j$ . The defined region of interest in this projection has been marked with green.*



to use the rows of the image corresponding to the north western point (62N,0E) to the south western point (50N,0E) and the columns between the image coordinates corresponding to the south western point (50N,0E) and the south eastern point (62N,0E).

This is only a crude preprocessing step to reduce the amount of data. By specifying a region within the satellite imagery, some analyses, where only the satellite data are considered, can be performed directly in the data rather than having to change to a different projection. An example of the provided image data for a single channel can be seen in Figure 2.3, where the two regions are marked. The geographical region is also shown in Figure 2.2.



**Figure 2.3:** A view of the first visible channel from August 20th 2007 at 14:12:52. The green area is a rectangle in geographical coordinates. The red rectangle is the chosen region of the satellite imagery.

#### 2.2.4 Spatial resolution

While the pixel resolution refers to the number of pixels used to represent the image, the spatial resolution refers to the ground sampling distance between measurements, i.e., the actual size of each pixel on the ground [50]. As mentioned above, the spatial resolution of the VIS and IR channels' imagery are  $3 \times 3$  km at the sub satellite point. As the satellite is geostationary and the image acquisition is based on constant angular steps seen from the geostationary orbit, the spatial resolution is reduced for geographical regions with a larger distance to the Equator.

An overview of the spatial resolution for the satellite's coverage area can be found in [26]. Denmark's geography spans latitudes from approximately  $54^{\circ}33'N$  to  $57^{\circ}45'N$  and longitudes

from  $8^{\circ}4'E$  to  $15^{\circ}11'$ , wherefore the resolution can be read off to 6–8km in N-S direction and 4km in E-W direction. This rather rough resolution can influence the analyses later in this thesis, as this obviously sets a lower bound for how small clouds can be seen in the satellite data.

### 2.2.5 Map to image coordinates

The transformation between map and image coordinates is a relatively simple linear transformation. It can be calculated from the map coordinates of the image's upper left corner  $(x_{11}, y_{11})$  and the resolution  $res$ , i.e., the pixel size, in the image. However, for the satellite data, the upper left corner is not the same for all scenarios, as the satellite can change position between scenarios. Therefore, informations to calculate the image region described above and the upper left corner in map coordinates are extracted for each scenario. The needed constants are the number of lines in the image  $N_l$ , the western column of the acquired subset  $c_w$ , the northern line number  $l_n$ , the western  $r_w$  and northern  $r_n$  pixel coordinates of the cropped region, described above. From these, the corner coordinates can be calculated as

$$\begin{aligned} x_{11} &= \left( \frac{N_l}{2} - c_w + (r_w - 1) \right) * res \\ y_{11} &= \left( l_n - \frac{N_l}{2} - (r_n + 1) \right) * res . \end{aligned}$$

These constants have different names in the metadata, which can be seen in [26].

Given a map coordinate  $[x_m, y_m]$ , the corresponding pixel coordinate can be calculated as

$$\begin{aligned} row &= \frac{y_m - y_{11}}{res} \\ col &= \frac{x_m - x_{11}}{res} . \end{aligned}$$

Correspondingly, a pixel coordinate can be transformed into a map coordinate by

$$\begin{aligned} x_m &= row \cdot res + x_{11} \\ y_m &= col \cdot res + y_{11} . \end{aligned}$$

Determination of the upper left corner from satellite image metadata and conversion between coordinates can be seen implemented in Appendix A.3.

## 2.3 Radar data

Radar data have been provided for all scenarios by DMI. Five weather radars have been operational during the period with locations as listed in Table 2.3 and shown on map in Figure 2.4. However, data were not available from all radars at all times. This can be caused by the particular radar not being operational at the time or neglect to store the particular radar image.

|          | Abbreviation | Position                     | Year |
|----------|--------------|------------------------------|------|
| Bornholm | ekrn         | 55.112750°N<br>14.887517°E   | 2007 |
| Rømø     | ekxr         | 55.173111°N<br>8.552000°E    | 2003 |
| Sindal   | eksn         | 57.489306°N<br>10.136472°E   | 2003 |
| Stevns   | ekxs         | 55.326194°N<br>12.449278°E   | 2001 |
| Virring  | ekxv         | 56.0240069°N<br>10.0245906°E | 2008 |

**Table 2.3:** *Operational weather radars in Denmark.*

Two examples of composite images can be seen in Figure 2.5. In the first all radars were available, while in the second example only four radars have contributed. Clutter and false echoes can occur, which must also be considered when using the data [8]. In this thesis, analyses including the radar data are only using points where data are present, i.e., no assumptions that missing data equal zero precipitation are made.

### 2.3.1 General radar principles

The basic working principle for a radar is to send out electromagnetic waves in one direction and based on the reflected signal determines distance and absorbance of the reflector. Important characteristics of an electromagnetic wave includes frequency and wavelength, which are related such that their product equals the speed of light. The range of electromagnetic radiation vary in frequency, depending on its use. DMI's weather radars have variable frequencies between 250 and 1200 Hz.

The speed of which the electromagnetic wave travels, depend on the material of the medium in which it travels. Refractivity describes this relation and is defined from the ratio between the speed of light in vacuum and the speed of light in a given medium. The refractivity of the atmosphere has been found to depend on its temperature, atmospheric pressure and vapor pressure. Hence, the refractivity differs for different layers of the atmosphere, wherefore the electromagnetic radiation travel at different speeds. These relations are for instance needed when determining a radar beam's height in the atmosphere [79].

For meteorological purposes, the radar can be used to detect droplets of rain in the air, i.e., precipitating clouds. Raindrop size distributions have been studied extensively for half a century, where the most significant result is the linear relation between rainrate and reflectivity [54]. Rain

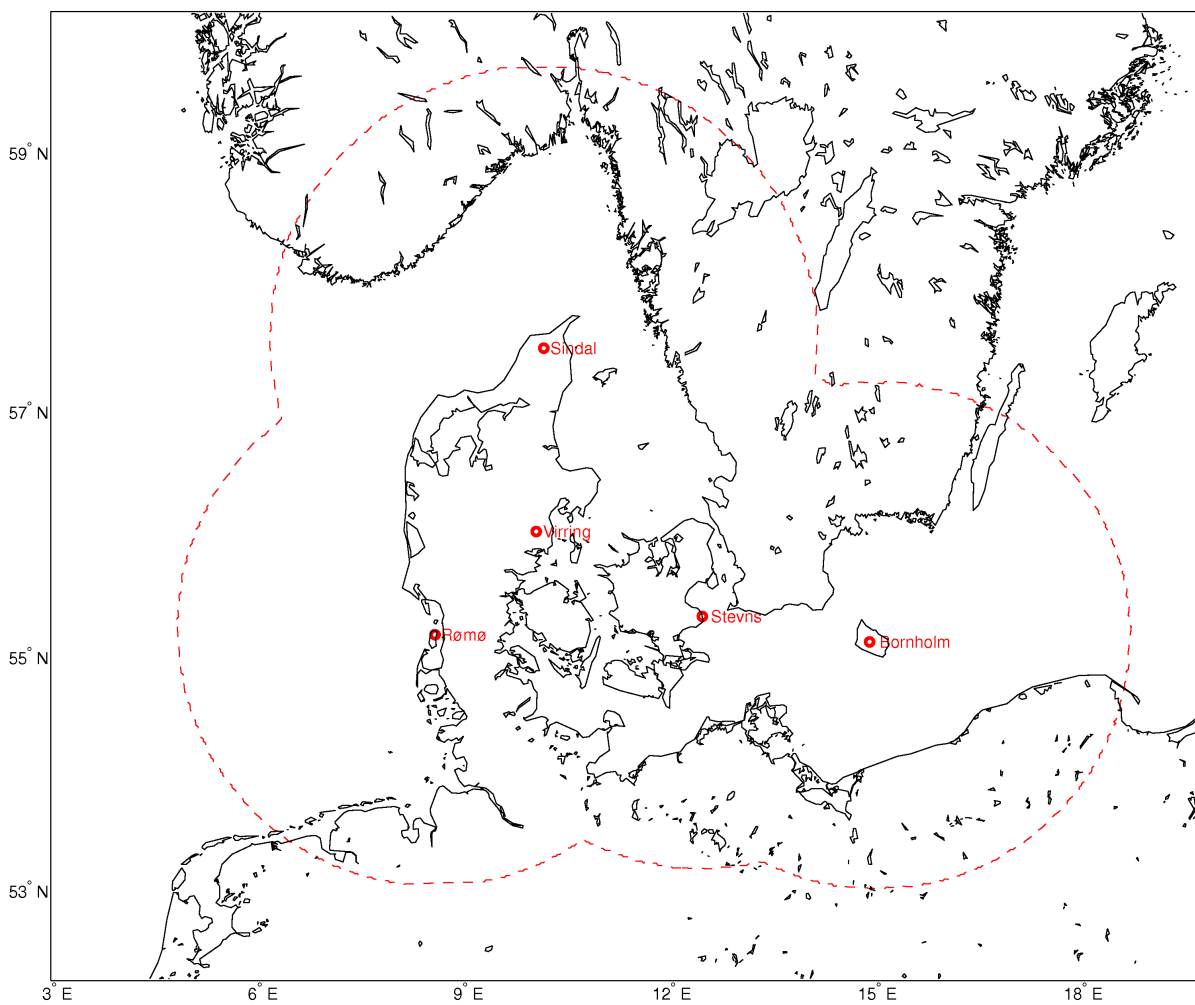


Figure 2.4: The five operational radars in Denmark.

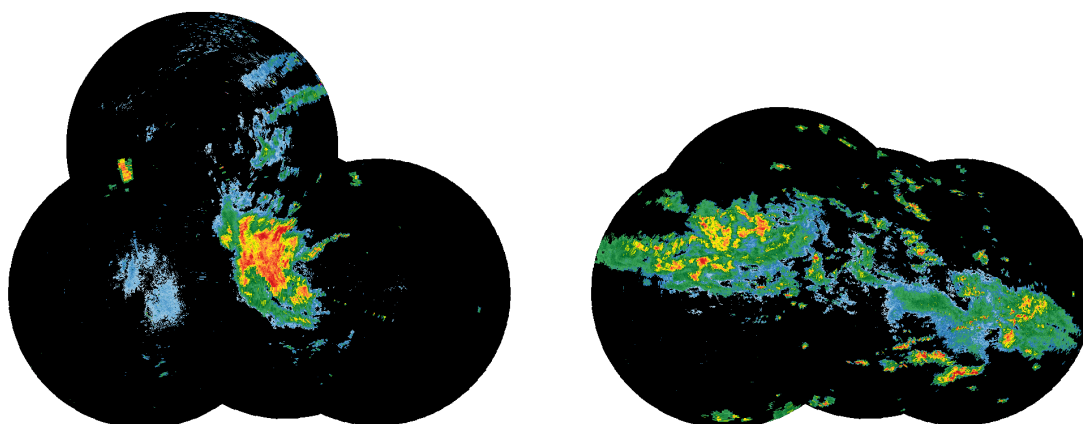


Figure 2.5: Radar data for (left) 2010-08-14 20:50 and (right) 2010-08-18 15:10.

intensity is usually denoted dBZ and is radar reflectivity  $z$  on a logarithmic scale:

$$dBZ = 10 \cdot \log_{10} \left( \frac{z}{1\text{mm}^6/\text{m}^3} \right) \quad (2.4)$$

The radar reflectivity factor can be estimated from the power received by the radar and constants relating to the specific radar construction. The reflectivity of rain varies from approximately 20dBZ up to as much as 55dBZ in severe thunderstorms. Parameters specific to DMI's weather radars can be found in [17].

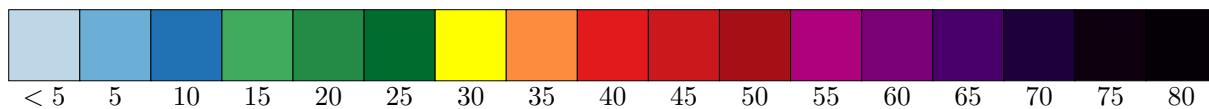
### 2.3.2 Specifications

The available radar data have been projected into a composite file, i.e., an image consisting of data from one to five radars. The format of this image file is an internal DMI format described in [18]. A binary reader for the data files have been implemented in MATLAB, see Appendix A.2.

The raw composite data consists of 8-bit unsigned integers. Values out of radar range have the value 255 and pixels within range, but without data have the value 0. These counts can be converted to dBZ, i.e., decibels of reflectivity  $Z$  using the following linear relation

$$dBZ = \text{count} \cdot 0.5 - 32 . \quad (2.5)$$

A common colormap for displaying radar reflectances is shown in Figure 2.6. Various values



**Figure 2.6:** *Colormap for radar reflectances. Values are in dBZ.*

have been used as thresholds for heavy rain; for instance 30dBZ is used as one of the criteria for hazardous conditions in [43], while a 35dBZ is used as an indicator of convective initiation in [59, 58]. In general this threshold is chosen to fit the geographical region being analyzed and the analysis context. In this thesis, values above 35 dBZ are considered heavy rain unless otherwise stated.

### 2.3.3 Projection

The composite images are in  $1000 \times 1000$  meter spatial resolution in an image grid of size  $844 \times 963$  pixels. The grid's upper left corner is at map coordinates (26785.817348, 821728.101419). The conversion between map coordinates and image coordinates is straightforward in this case, as all radar data have the same upper left corner and resolution. The same procedure and MATLAB functions as outlined in Section 2.2.5 is used for this purpose.

Compared to the satellite data the spatial resolution is much finer for the radar data. This is of course because the radars are dedicated for this specific geographical region, while the satellites cover a much larger area.

The projection of the data is a Polar Stereographic projection, with the necessary parameters given in Table 2.4.

|                             |         |
|-----------------------------|---------|
| False easting               | 450000  |
| False northing              | 350000  |
| Latitude at natural origin  | 56.0    |
| Longitude at natural origin | 10.5666 |
| Latitude of true scale      | 56.0    |

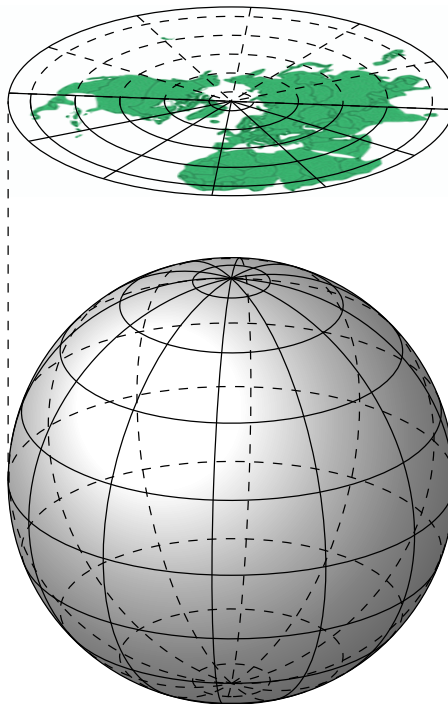
**Table 2.4:** Parameters for the Polar Stereographic projection used for the composite image.

The *proj-string* for this projection is exemplified here:

```
> proj +proj=stere +ellps=WGS84 +x_0=450000 +y_0=350000
      +lat_0=56.0 +lon_0=10.5666 +lat_ts=56.0 -I
> 446557.26      222303.98
   10d30 '46.8"E   54d51 '10.8"N
```

Once again the example shows the inverse projection, i.e., from map coordinates to geographical coordinates. It should also be noted that the *PROJ* output in geographical coordinates are reversed of the usual standard, where latitudes are before longitudes.

The Polar Stereographic projection is illustrated in Figure 2.7. This is DMI's usual choice of projection for display of radar data, as it provides a map with visually pleasing proportions of Denmark.



**Figure 2.7:** Illustration of the Polar Stereographic Projection.

### 2.3.4 Radar coverage mask

The weather radars have a limited coverage of approximately 240km and, as mentioned above, the number of available radars at any given time may vary. Therefore a binary mask have been determined for each scenario, where the radar coverage for the given day is indicated with ones and zeros for areas out of reach. Furthermore, for illustration purposes, the boundary of the coverage area is needed in coordinates, such that they can be projected to geographical coordinates and shown on a map for instance. This can be obtained from the composite radar data by a few simple steps:

1. Load radar composite image  $\mathbf{y}$  for scenario
2. Determine binary mask as  $\mathbf{M} = (\mathbf{y} \neq 255)$
3. Find boundary of mask
4. Project to geographical coordinates using above *proj-string*

The radar boundary for one of the more recent scenarios, where all five radars were operational are shown in Figure 2.4. The implementation of this can be seen in Appendix A.2.

### 2.3.5 Accumulating radar data

For the purposes in this thesis it was found that an accumulation of the radar data eased the task of identifying the cloud of interest and the time of maximum precipitation in the radar images. The method used was to take the total reflectivity within  $\pm 30$  minutes of each sampling time. Hence, the smoothed value in the radar pixel  $y_i(x)$  at time  $t_i$  is calculated as

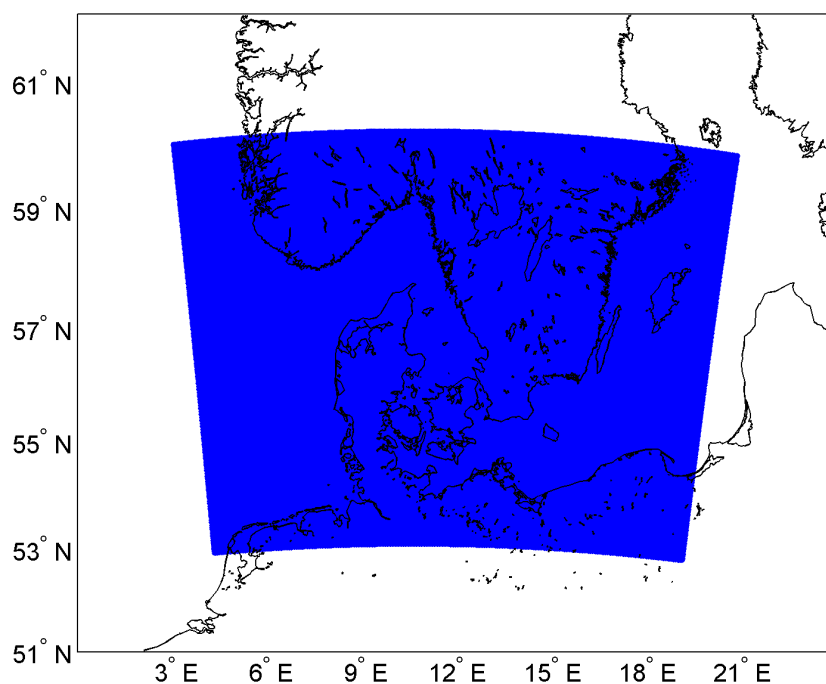
$$y_i(x) = 10 \cdot \log_{10} \left( \sum_{j=-3}^3 z_{i+j}(x) \right) \quad (2.6)$$

where  $z_i(x)$  is the reflectivity at the spatial location  $x$  at time  $t_i$ . Three time steps before and after the current time is used, since the sampling time is ten minutes. This accumulation uses future information and is therefore not used when results are compared with forecasts, as all the information is not yet available. It is only used for easier identification of the convective area and future reflectivity is included, rather than accumulating past reflectivity, to ensure that the point in time to predict, is not being chosen too late in the time series.

## 2.4 Choosing a common grid

The importance of defining a common grid covering the area of interest is twofold: (i) The satellite data and radar data are in two different projections and are going to be analyzed simultaneously and (ii) the satellite data covers a geographical region that is unnecessarily large in this context.

The image grid was chosen to be of size  $400 \times 500$  pixels, with a pixel size of  $2 \times 2$  km. This resolution was chosen as an intermediate point between the coarse satellite resolution and the finer radar imagery. The upper left corner of the image grid was also chosen to be the same as for the radar data. Hence the same method for retrieving the geographical coordinates of the image grid can be used. The grid points are illustrated in a Mercator projection in Figure 2.8.



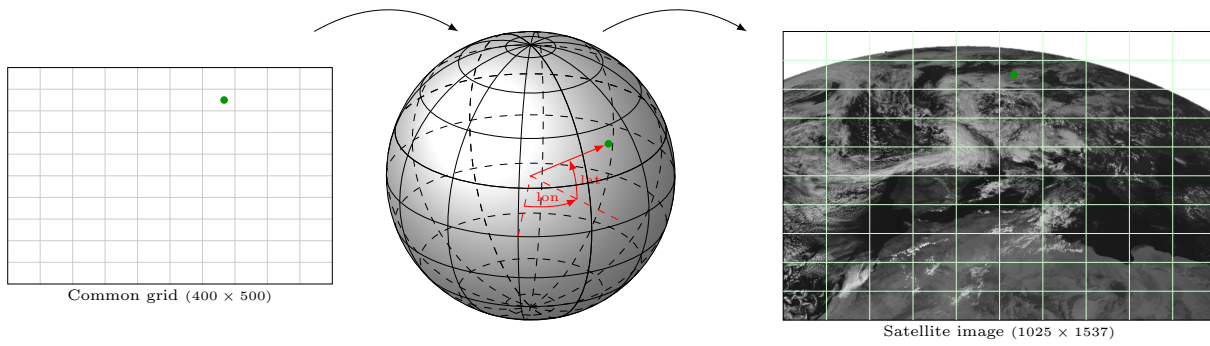
**Figure 2.8:** Defined grid with  $400 \times 500$  square pixels covering the region of interest shown in blue in a Mercator projection.

The four corners of the region have geographical coordinates

$$\begin{aligned} \text{NW} &= \begin{cases} 60^\circ & 1' & 12''\text{N} \\ & 2^\circ & 58' & 33''\text{E} \end{cases} & \text{NE} &= \begin{cases} 59^\circ & 50' & 28''\text{N} \\ & 20^\circ & 50' & 10''\text{E} \end{cases} \\ \text{SW} &= \begin{cases} 52^\circ & 54' & 0''\text{N} \\ & 4^\circ & 16' & 29''\text{E} \end{cases} & \text{SE} &= \begin{cases} 52^\circ & 45' & 25''\text{N} \\ & 19^\circ & 5' & 24''\text{E} \end{cases} \end{aligned}$$

Filling this grid with satellite or radar data requires a few steps: (i) The common grid image coordinates are changed to map coordinates, (ii) the geographical coordinates are obtained using an inverse projection, (iii) map coordinates in the target projection (satellite or radar) are obtained by a forward projection of the geographical coordinates, (iv) map coordinates are transformed to image coordinates and (v) the common grid are filled with values using these image coordinates. This whole procedure works as a mapping from the common grid's image coordinates to the data's image coordinates. An illustration of this is shown in Figure 2.9.





**Figure 2.9:** *Illustration of mapping from the common grid's coordinates to the satellite data's coordinates.*

Coastline data have been extracted for the region of interest, from the GSHHS database [91].



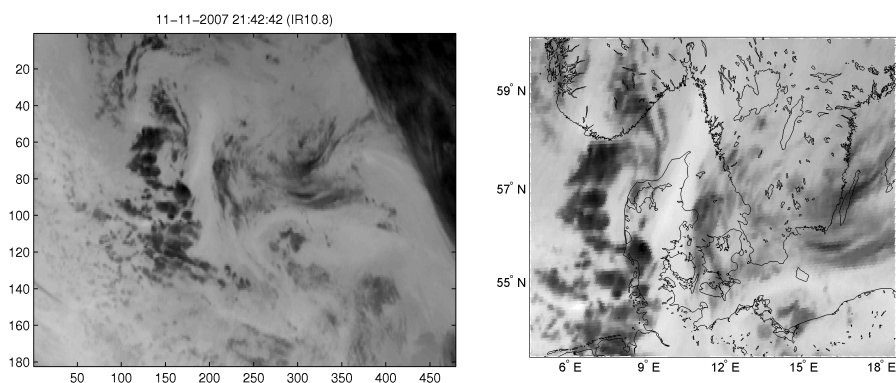
## Explorative analysis

A first step when dealing with real-world data sets is to gain an understanding of the data contents in general. Furthermore, the data contents in relation to the task at hand should be explored. A wide variety of methods exist for this purpose, ranging from simple statistics to non-linear decompositions of signals contained in the data. In this section, a few well-known multivariate statistical methods are used to explore the satellite and radar data.

### 3.1 Subspace projections

Subspace projections are typically used to extract information from multispectral or -temporal data sets. This is accomplished by maximizing a measure along the projection directions in space. The nature of this measure eventually determines the subspace and should be chosen carefully in order to extract exactly the information needed, using the fewest possible dimensions.

In order to use subspace projections to identify heavy rain clusters, a number of descriptive directions in space are determined for a situation with such conditions present. Here, June 11-12, 2007 at 15:00 UTC is chosen, as it is known to be a time of extreme rain over Jutland and therefore previously identified as a hotspot. The data at this point in time for a single channel can be seen in Figure 3.1.



**Figure 3.1:** 2007-11-11. Chosen situation with heavy rain present. (left) Analyzed subset in original imagery, i.e., in geostationary projection. (right) Region of interest in a Mercator projection.

The exact meaning of “descriptive” in this case depends on the method used to determine the

## CHAPTER 3. EXPLORATIVE ANALYSIS

subspace, as mentioned above. From these descriptive directions, a single direction is selected and chosen as a scale for heavy rain clusters. Projecting new data onto this direction provides a measure of similarity, in term of selected measure, with the chosen situation. In this case a situation with heavy rain.

A few different methods for determining a descriptive subspace are described below, and their capabilities for identification of rain clusters are evaluated. Some of these methods have previously been treated in [88].

For these analyses, the satellite data are arranged in a data matrix  $\mathbf{X}$  of size  $P \times N$ , where  $N$  is the number of observations and  $P$  is the number of spectral channels.

## 3.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a very common method in multivariate statistics, where it is applied to a wide variety of data and extended in various ways, see for instance [41, 70, 1]. In other areas, the same method is known under different names, such as Empirical Orthogonal Functions in oceanography [76, 78].

In summary, it is calculated by an eigenvalue decomposition of the dispersion matrix, where the variance is maximized along each principal direction, with the constraint of the directions being mutually orthogonal. Typically, the transformation is carried out such that the first principal direction is oriented along the direction of maximum variation in the data, the second principal the second most, and so forth. The original data projected onto the subspace spanned by the principal directions are called the Principal Components (PCs). By maintaining only a number of principal directions, this transformation will reduce the data dimensionality while preserving a maximum amount of variation in the data.

As the aim is to identify clusters of extremities in a multispectral image, deviation from the mean is what should be maximized. Therefore, the variance is potentially an informative measure to maximize. To find the principal directions, the dispersion matrix  $\Sigma_X$  is estimated as

$$\Sigma_X = \frac{1}{N-1} \mathbf{X} \mathbf{X}^T . \quad (3.1)$$

This analysis mode is sometimes referred to as R-mode analysis [78]. The principal directions are found as the eigenvectors  $\mathbf{u}$  of  $\Sigma$

$$\Sigma_X \mathbf{u}_j = \lambda_j \mathbf{u}_j , \quad j \in [1, P] \quad (3.2)$$

where  $\lambda_j$  is the associated eigenvalue to the solution of this eigenvalue problem. Having determined the eigenvectors, which are now termed the principal directions, the  $j$ 'th principal component (PC) is found by projecting the data into the subspace spanned by this principal direction

$$\text{PC}_j = \mathbf{u}_j^T \mathbf{X} , \quad j \in [1, P] . \quad (3.3)$$

The eigenvalue  $\lambda_j$  is proportional to the amount of variance preserved in the  $j$ 'th PC. Usually the PCs are sorted such that the PC1 contains the most variance, i.e.,  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_P$ . Hence, by preserving the first few PCs, it is an effective method for reduction of data dimensionality, while preserving a maximum amount of variance. The PCA can be calculated in a computationally efficiently and numerically stable way by use of the Singular Value Decomposition (SVD) [7, 23].

### 3.2.1 Applying to satellite data

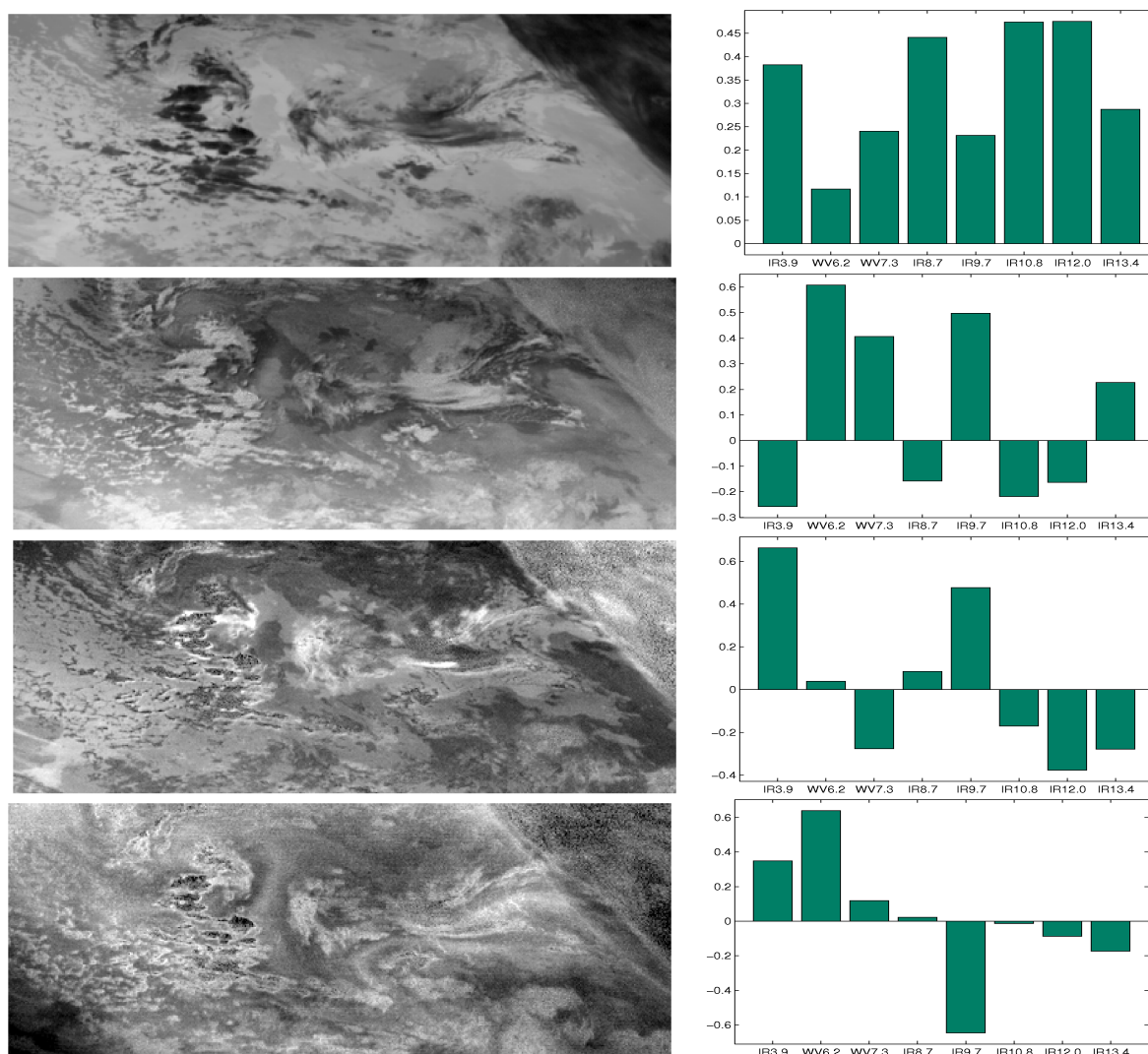
The goal of the explorative analysis in the context of convective clouds, is to find a way to characterize the cloud of interest. As it is known that such clouds exist at the time of the hotspot for each scenario, the PCA is performed for each scenario at exactly this time. The motivation for applying PCA to this type of data is, as stated above, that the only known thing about the clouds is that they exhibit extreme behavior. Extreme behavior in a numerical sense must be deviation from the mean, which variance is a measure for.

The data analyzed are from the scenario exemplified in Figure 3.1. The PCA is performed exclusively on the  $P = 8$  IR channels to avoid any potential issues with night time images. For

## CHAPTER 3. EXPLORATIVE ANALYSIS

the same reason, the further analyses in this thesis are also performed only on these spectral channels.

From the eigenvalues it can be calculated that the first four PCs represent approximately 93% of the variation in the data. These four PCs are shown together with the associated eigenvectors in Figure 3.2.



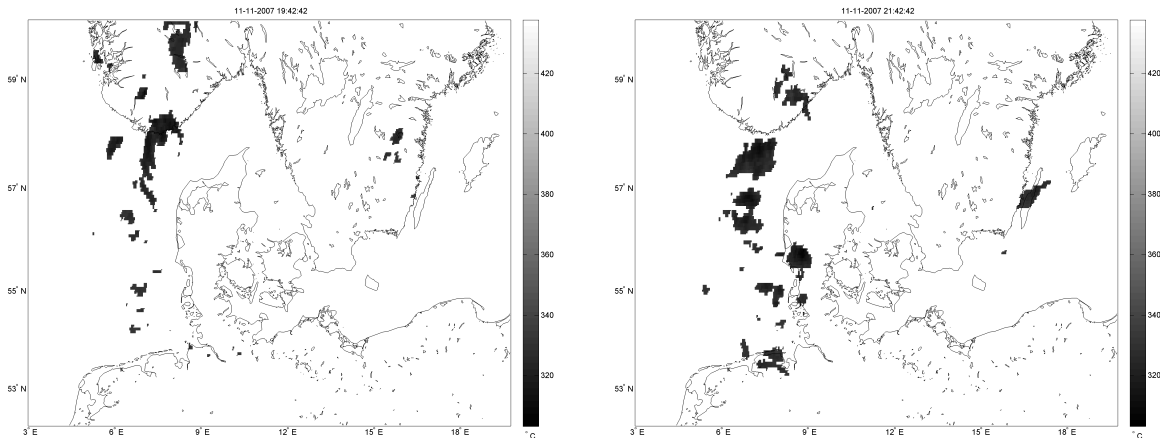
**Figure 3.2:** 2007-11-11. (left) First four PCs with intensities stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

The first PC seems to create a contrast between convective clouds and non-convective areas. From simultaneous inspection of the associated eigenvector, it can be seen that this component contains positive contributions from all channels. The second PC extracts somewhat similar areas, even though the eigenvector is very different from the first one. Noteworthy is though that what is expected to be convective areas, do not have as extreme values in this PC as in the first. PC3 highlights small holes in the cloud cover and what looks like cloud edges, while something harder to identify in the top of the image also have large weight, showing as black. The fourth PC extracts small areas in the middle of what is known to be a convective system, which from the eigenvector is seen to be dominated by information from the fifth infrared channel, IR9.7.

The first four PCs for the remaining scenario hotspots can be seen in Appendix C.1.

Based on this example and similar results obtained by visual inspection from the remaining scenarios, the first PC is chosen as the best one to extract convective clouds.

As an experiment, the first eigenvector is used as a scale for heavy convection by projecting other images from the same day onto this. In Figure 3.3 two different points in time on the same day has been chosen, projected onto the chosen direction from before and thresholded, such that only observations below 333 (on this new scale) have a value. This value was calculated as the mean minus two standard deviations of PC1.



**Figure 3.3:** 2007-11-11. Thresholded PC1 at selected points in time before and during heavy rainfall in Jutland. Threshold used is 333.

From these plots it can be seen that the convective clouds form south west of Jutland approximately two hours prior to the downpour.

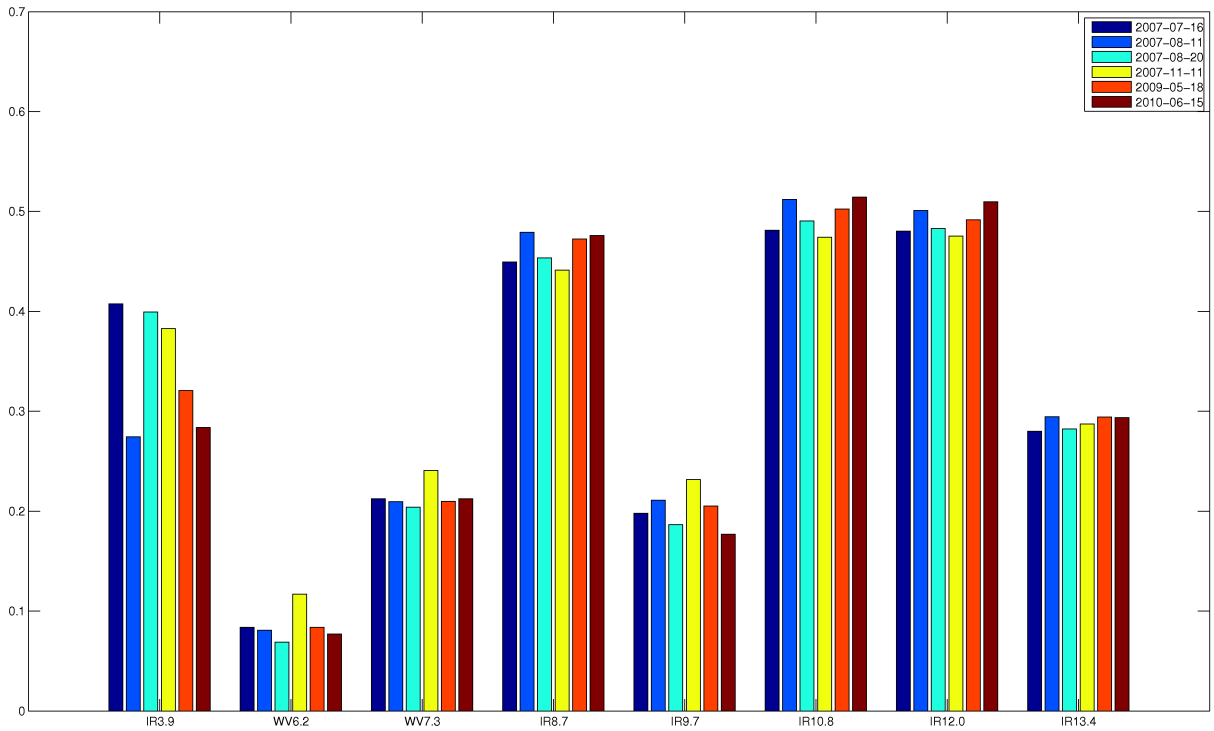
Even though this thresholding potentially could identify heavily precipitating clouds in satellite imagery, there is no guarantee that the method identifies the correct clouds. The radar data need to be included in the analysis to ensure this. Therefore this method will not be pursued further at this stage, as there exist no way of validating the results.

### 3.2.2 A generalizable subspace

A PCA has been calculated independently for each of the scenarios at the hotspots, described in Section 2.1, and the eigenvector belonging to the first PC was extracted. These six eigenvectors are shown together in Figure 3.4. Immediately, it is seen that the eigenvectors are very similar, which indicates that the variance contained in the data at these points in time, are also very similar in structure. This is also what would be expected, but more importantly it is also necessary in order to use the same methodology when analyzing these scenarios.

This can be used to calculate a “global PCA”, i.e., a PCA where all of these hotspots are included, whereby the transformation will – hopefully – generalize better.

The data matrix  $\mathbf{X}_g$  is then composed of the satellite data from all  $N_s$  scenario hotspots. Here, this data for the  $i$ 'th scenario is denoted  $\mathbf{X}_i$ ,  $i \in [1, N_s]$  where each of the  $P = 8$  spectral channels



**Figure 3.4:** First principal direction for PCAs calculated independently for all scenario hotspots.

is a row in  $\mathbf{X}_i$ . Thus the global data matrix can be arranged by a stacking of the individual scenarios' data:

$$\mathbf{X}_g = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_{N_s} \end{bmatrix}. \quad (3.4)$$

Performing the analysis is now directed against maximizing variance in this composite data. The eigenvalues yielded from the analysis can be seen in Figure 3.5, where the first principal direction is also shown. The eigenvalues are shown as a cumulative sum of percentage variance, such that the value for the  $k$ 'th PC is calculated as

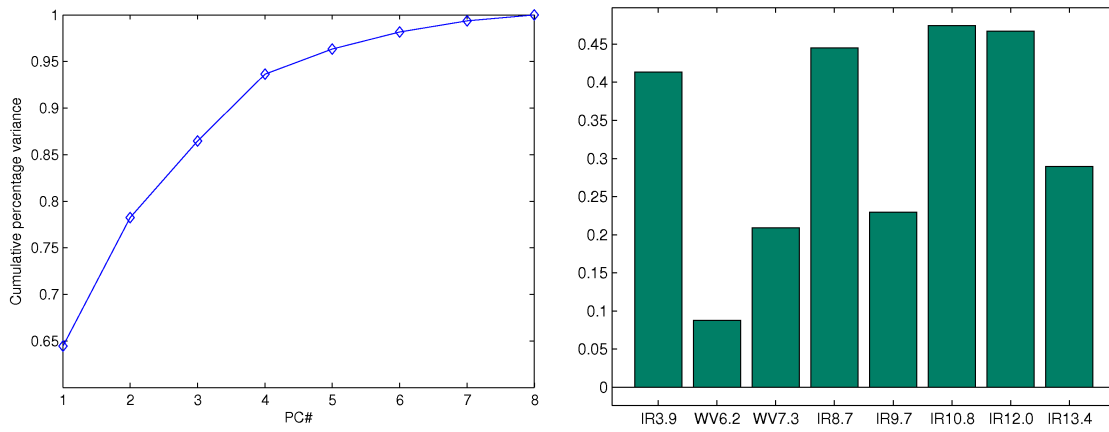
$$\text{cum. var.}_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^P \lambda_i}, \quad k \in [1, P]. \quad (3.5)$$

From this plot it is easily seen that more than 93% of the variance is explained in the first four PCs and the first PC itself accounts for 65% of the total variance in the data.

The first principal direction is of interest, as this is the one where similarities were observed between the scenarios. It is seen, as expected, that it resembles the dominating pattern from Figure 3.4. All eight principal directions can be seen in Appendix C.2. The spatial representation of the PCs will not be shown, as they visually resemble the ones from the individual analysis. This first principal direction will be used later in this thesis as a fixed transformation and the analysis including all scenarios will be referred to as the global or hot PCA.



### 3.2 Principal Component Analysis (PCA)



**Figure 3.5:** (left) Cumulative proportion of variance explained. (right) First principal direction for global PCA.

### 3.3 Maximum Autocorrelation Factor (MAF) analysis

The objects of interest in these analyses are clouds causing heavy precipitation. It is expected that these clouds are identifiable in the satellite data and one of the characteristics of clouds is the spatial structure, i.e., the clouds are isolated objects. Therefore it would make sense to include a spatial property in the methodology, which PCA does not.

Maximum Autocorrelation Factor (MAF) analysis aims at finding a subspace that maximizes autocorrelation between the signal and a shifted version of itself. This shift can be in e.g., space or time. MAF is treated in [70], where the autocorrelation between the signal  $\mathbf{X}(x)$  and the shifted version of itself  $\mathbf{X}(x + \Delta)$  is derived to be:

$$\text{corr} \left\{ \mathbf{a}^T \mathbf{X}(x), \mathbf{a}^T \mathbf{X}(x + \Delta) \right\} = 1 - \frac{1}{2} \frac{\mathbf{a}^T \boldsymbol{\Sigma}_\Delta \mathbf{a}}{\mathbf{a}^T \boldsymbol{\Sigma} \mathbf{a}} \quad (3.6)$$

where  $\mathbf{a}$  is the vector to be determined,  $\Delta$  is the shift in signal,  $\boldsymbol{\Sigma}$  the dispersion matrix of the original signal and  $\boldsymbol{\Sigma}_\Delta$  the dispersion matrix of the shifted signal.

To maximize the autocorrelation in (3.6), the Rayleigh coefficient is to be minimized

$$\min_{\mathbf{a}} \frac{\mathbf{a}^T \boldsymbol{\Sigma}_\Delta \mathbf{a}}{\mathbf{a}^T \boldsymbol{\Sigma} \mathbf{a}} . \quad (3.7)$$

This minimization can be shown to be the solution of the generalized eigenvalue problem [88]:

$$\boldsymbol{\Sigma} \mathbf{a}_j = \lambda_j \boldsymbol{\Sigma}_\Delta \mathbf{a}_j , \quad j \in [1, P] \quad (3.8)$$

where  $\lambda_j$  is the eigenvalue associated with the  $j$ 'th eigenvector  $\mathbf{a}_j$  of  $\boldsymbol{\Sigma}_\Delta$  with respect to  $\boldsymbol{\Sigma}$ . The autocorrelation for the  $j$ 'th component can be calculated from the eigenvalue as

$$\rho_j = 1 - \frac{1}{2\lambda_j} , \quad j \in [1, P] . \quad (3.9)$$

The data projected into the subspace spanned by the  $j$ 'th component is called the  $j$ 'th MAF:

$$\text{MAF}_j = \mathbf{a}_j^T \mathbf{X} , \quad j \in [1, P] \quad (3.10)$$

and are usually sorted according to decreasing autocorrelations, such that  $\rho_1 \geq \rho_2 \geq \dots \geq \rho_P$ . Thus the first component, MAF1, contains the largest amount of autocorrelation.

As the measure of interest in this case is the spatial coherence, the shift is chosen to be one step in the horizontal direction and one in the vertical. This is accomplished by estimating  $\boldsymbol{\Sigma}_\Delta$  as the average of the dispersion matrices estimated from the horizontal shift and vertical shift:

$$\boldsymbol{\Sigma}_\Delta = \frac{1}{2} (\boldsymbol{\Sigma}_H + \boldsymbol{\Sigma}_V) . \quad (3.11)$$

Due to the more complex covariance structures analyzed in MAF than e.g., PCA, it is often not feasible to visually inspect the eigenvector  $\mathbf{a}$ . Instead, the correlation between the original variables and the projected variables are considered. This can be calculated efficiently from the

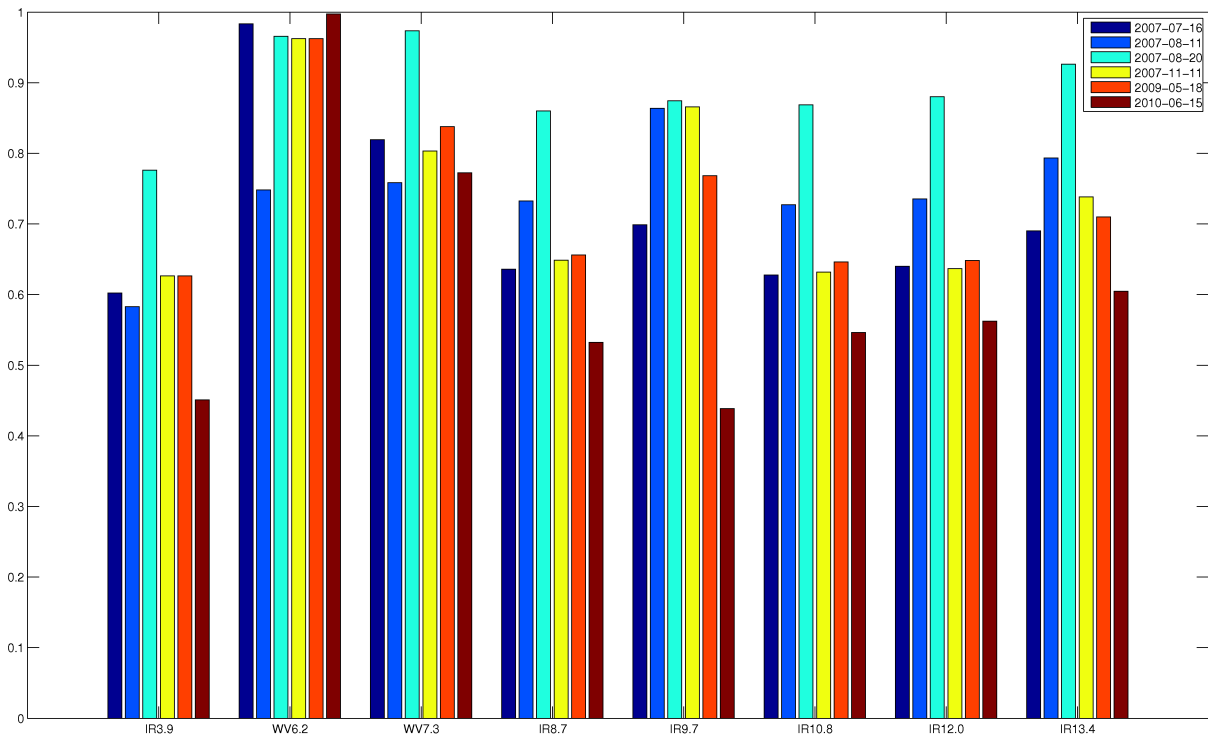
covariance matrices, which are already estimated to obtain the eigenvector:

$$\begin{aligned}
 \text{corr} \{ \mathbf{X}, \mathbf{a}^T \mathbf{X} \} &= \frac{\text{cov}(\mathbf{X}, \mathbf{a}^T \mathbf{X})}{\sigma_X \cdot \text{std} \{ \mathbf{a}^T \mathbf{X} \}} \\
 &= \frac{\text{cov}(\mathbf{X}, \mathbf{X}) \mathbf{a}}{\sigma_X \cdot \sqrt{\text{diag}[\text{cov}(\mathbf{a}^T \mathbf{X})]}} \\
 &= \frac{\Sigma \mathbf{a}}{\sigma_X \cdot \sqrt{\text{diag}[\mathbf{a}^T \Sigma \mathbf{a}]}} \tag{3.12}
 \end{aligned}$$

where  $\sigma_X$  is the standard deviation of the original variables.

### 3.3.1 Analysis results

The MAFs have been estimated for all scenarios at their hotspots, as it was done with PCA. The correlations for the six scenarios' first MAF can be seen in Figure 3.6. The correlations are seen to be very similar, with positive correlations between all channels and the projected data. However, the spatial patterns need to be considered as well, before setting up a global transformation similar to the PCA.



**Figure 3.6:** Comparison of correlations for MAF1 calculated on satellite data at all scenario hotspots.

The first four MAFs and associated correlations can be seen in Appendix C.3. Some of the components show interesting patterns, such as MAF1 for scenario 2007-08-20 in Figure C.10, where a signal of heavy precipitating clouds definitely has been captured. But, observing MAF1 from the 2009-05-18 in Figure C.12 it is clear that the first components' spatial patterns are not

### CHAPTER 3. EXPLORATIVE ANALYSIS

as consistent was seen with those of the PCA. It seems from the spatial representations and the correlations that the larger, less intense areas of water vapor are captured, where especially the second IR channel contributes is represented, compared to the global PCA. Therefore it does not make sense to make a global transformation in this case, as the spatial patterns of interest are the smaller, more intense areas of convective initiation.

### 3.4 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) is a method for analysis of relations between two sets of variables. It was first introduced in [36] and are described in textbooks on multivariate statistical analysis, e.g., [3, 90].

CCA maximizes the correlation  $\rho$  between linear combinations of these two sets of variables ( $\mathbf{X}$  and  $\mathbf{Y}$ ):

$$\rho = \text{corr} \left\{ \mathbf{a}^T \mathbf{X}, \mathbf{b}^T \mathbf{Y} \right\} = \frac{\mathbf{a}^T \boldsymbol{\Sigma}_{12} \mathbf{b}}{\sqrt{\mathbf{a}^T \boldsymbol{\Sigma}_{11} \mathbf{a}} \sqrt{\mathbf{b}^T \boldsymbol{\Sigma}_{22} \mathbf{b}}} \quad (3.13)$$

where  $\boldsymbol{\Sigma}_{12} = \text{cov}(X, Y)$ . This can be done by solving the generalized eigenvalue problem

$$\rho^2 = \frac{\mathbf{a}^T \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} \mathbf{a}}{\mathbf{a}^T \boldsymbol{\Sigma}_{11} \mathbf{a}} = \frac{\mathbf{b}^T \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12} \mathbf{b}}{\mathbf{b}^T \boldsymbol{\Sigma}_{22} \mathbf{b}} \quad (3.14)$$

where the eigenvectors  $\mathbf{a}_1, \dots, \mathbf{a}_p$  with corresponding eigenvalues  $\rho_1^2 \geq \dots \geq \rho_p^2$  are the desired projection directions for  $\mathbf{X}$ . For more details, see [36, 68].

Here, CCA can be used to assess which spectral bands in the satellite data are correlated with the radar data. In this context, the satellite data are a set of  $p$  spectral bands  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)^T$ , while the radar data are univariate  $\mathbf{Y} = \mathbf{y}_1^T$ .

#### 3.4.1 Univariate simplification

Since the radar data are univariate, i.e.,  $q = 1$ ,  $\mathbf{b} = b$  is a scalar and  $\mathbf{a}$  a  $p$ -vector. I.e.,  $\mathbf{a}$  are the weightings of each spectral band to obtain maximum correlation with the radar data. The covariance matrix of  $\mathbf{X}$  and  $\mathbf{y}$  will be a  $p$ -vector  $\boldsymbol{\sigma}_{12}$ .

In general, for more variables in  $\mathbf{X}$  than  $\mathbf{Y}$ , i.e.,  $p \geq q$ , the following system of equations should be solved

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \rho \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \quad (3.15)$$

Demanding unit variance  $V\{b\mathbf{y}\} = 1 = b^2\sigma_2^2$  implies that  $b = \frac{1}{\sigma_2}$ , where  $\sigma_2$  is the standard deviation of the radar data  $\mathbf{y}$ . This leads to the following derivation of  $\mathbf{a}$ :

$$\begin{bmatrix} \boldsymbol{\Sigma}_{12} \mathbf{b} \\ \boldsymbol{\Sigma}_{21} \mathbf{a} \end{bmatrix} = \begin{bmatrix} \rho \boldsymbol{\Sigma}_{11} \mathbf{a} \\ \rho \boldsymbol{\Sigma}_{22} \mathbf{b} \end{bmatrix} \Leftrightarrow \begin{bmatrix} b \boldsymbol{\sigma}_{12} \\ \boldsymbol{\sigma}_{12}^T \mathbf{a} \end{bmatrix} = \begin{bmatrix} \rho \boldsymbol{\Sigma}_{11} \mathbf{a} \\ \rho \sigma_2^2 b \end{bmatrix} \Rightarrow \quad (3.16)$$

$$\mathbf{a} = \frac{1}{\rho} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12} b \quad (3.17)$$

Inserting this into the bottom equation of Eq. (3.16), yields

$$\begin{aligned} \boldsymbol{\sigma}_{12}^T \mathbf{a} &= \boldsymbol{\sigma}_{12}^T \frac{1}{\rho} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12} b = \rho \sigma_2^2 b \Leftrightarrow \\ \boldsymbol{\sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12} &= \rho^2 \sigma_2^2 \Leftrightarrow \\ \rho^2 &= \frac{1}{\sigma_2^2} \boldsymbol{\sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12} \Rightarrow \\ \rho &= \frac{\sqrt{\boldsymbol{\sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12}}}{\sigma_2} \end{aligned} \quad (3.18)$$

which can be inserted into the top equation of Eq. (3.16):

$$\mathbf{a} = \frac{1}{\rho} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12} b = \frac{\boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12}}{\sqrt{\boldsymbol{\sigma}_{12}^T \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\sigma}_{12}}} \quad (3.19)$$

By using this simplification, it is avoided to solve an eigenvalue problem.

Equivalently to MAF analysis, the covariance structure in this analysis is too complex to merely inspect the raw eigenvectors. Instead, the correlation between the original variables and the canonical variate is considered. This is calculated as in Equation (3.12).

### 3.4.2 Analysis results

For the scenario 2007-11-11, the CCA has been performed and the projected data, called the first Canonical Variate (CV1), is shown in Figure 3.7. The hope was that CV1 would provide an image, where the heavy precipitating cloud could easily be identified. Even though the dark areas stand out, the image seems very smudged and it is not easy to identify the areas of interest. Thus, the first canonical variate is not very useful for the purposes here and it will be investigated what can be done to improve the quality of the subspace produced by CCA.

11-11-2007 21:42:42 (CV1)

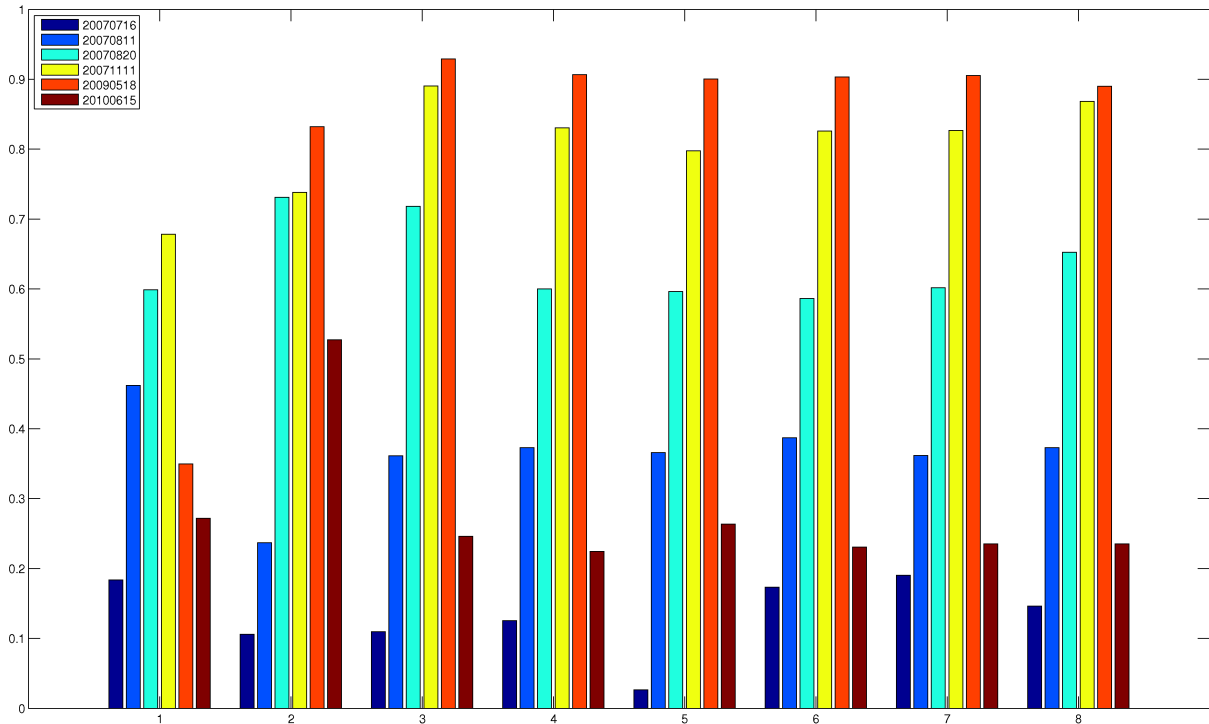


**Figure 3.7:** 2007-11-11. CV1 using satellite and radar data at time of hotspot.

A better way to quantitatively assess the robustness of the canonical correlation analyses, is to compare the correlations gained from each scenario's separate analysis, equivalent to the comparison of the principal components in Section 3.2.2.

Correlations have been maximized for all the scenarios and the correlations calculated from (3.12) are shown collectively in Figure 3.8. It is seen that the original spectral bands in all cases are

positively correlated with the projected data. However, the magnitudes of correlation are very inhomogeneous and have an average correlation of  $\bar{\rho} = 0.51$ , which seems low. It definitely does not seem that it would be appropriate to perform a CCA for all scenarios collectively.



**Figure 3.8:** Correlations of each spectral band in the satellite data with the projection of data determined for maximum correlation between satellite data and radar data.

These inconclusive results from the CCA give rise to further analysis in the following sections.

### 3.5 Spatial data correspondence

From previous analyses it has been indicated that the satellite and radar data are not aligned as one might expect. Even though the two data sets are reprojected to a common grid, a slight mismatch between them can still exist. The different view points of the weather satellite and the radar could be one explanation for such a discrepancy, as the satellite captures the cloud tops, while the radar captures precipitation. The satellite data are denoted  $\mathbf{X}(\mathbf{x})$  and the radar data  $\mathbf{Y}(\mathbf{x})$ , where  $\mathbf{x} = (x, y)$  is a coordinate in the common grid.

To quantify the spatial correspondence between the reflectance recorded in the radar data and potential convective areas identified using PCA, two geostatistical estimators are considered: The cross variogram and the cross covariance function. The scenario from 2007-08-20 is used for illustration, as it seems to contain a spatial displacement of a shower from PC1 to the radar data. The data are shown in Figure 3.9.

#### 3.5.1 Cross Variogram

Considering the two variables  $\mathbf{X}(\mathbf{x})$  and  $\mathbf{Y}(\mathbf{x})$ , the cross variogram for the separation vector,  $\mathbf{h} = (\Delta x, \Delta y)$ , is defined, assuming joint intrinsic variables, as half the expectation of the increments of the two variables [90]:

$$\gamma_{XY}(\mathbf{h}) = \frac{1}{2} \mathbb{E} [(\mathbf{X}(\mathbf{x} + \mathbf{h}) - \mathbf{X}(\mathbf{x})) \cdot (\mathbf{Y}(\mathbf{x} + \mathbf{h}) - \mathbf{Y}(\mathbf{x}))] . \quad (3.20)$$

Thus for large simultaneous increments by the spatial shift  $\mathbf{h}$ , the result will be large positive value, while a large negative value can be expected for decrements in one variable and increment in the other. A small absolute value will be expected for small changes in one or both variables.

By varying the separation vector, a cross variogram surface can be constructed, where the x-axis corresponds to a shift in the longitude direction and latitude shifts vary along the y-axis. Values on the shown variogram surface axes are in pixels, where each pixel is  $2 \times 2$  km. Such a cross variogram surface has been calculated for the scenario from Figure 3.9 and is seen in Figure 3.10.

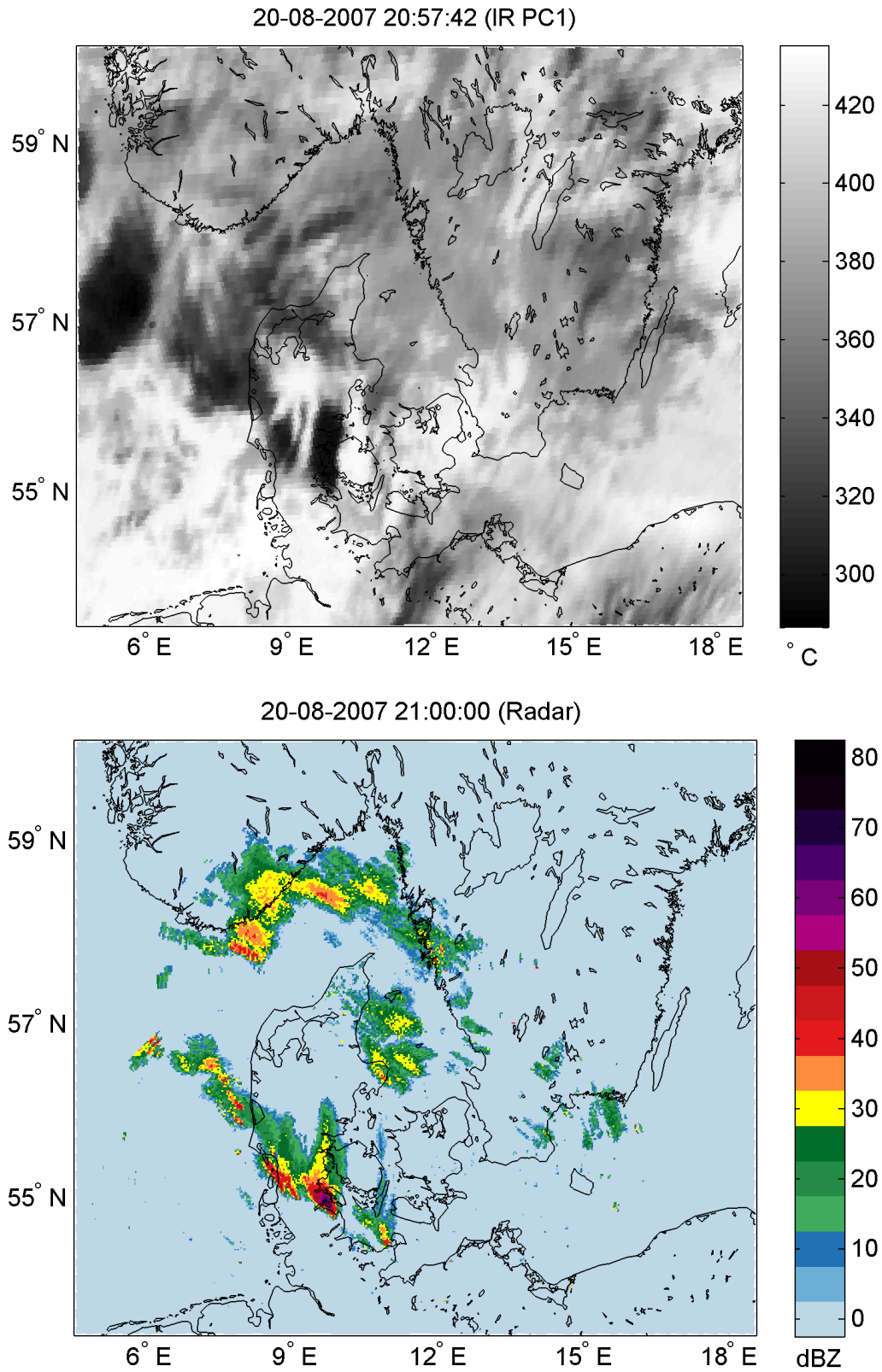
Some structure is seen in the cross variogram surface: For spatial shifts of the radar data of  $\mathbf{h} \approx (30, 20)$ , the variation in the two variables are of the same sign, while for a spatial shift of  $\mathbf{h} \approx (-30, 7)$  they exhibit opposite change. This, however, does not give any information how the two variables are spatially located in reference to each other. It only provides information of whether their variation is similar for various spatial shifts. For other purposes, this might be a valuable method, but a different approach must be used for the current purpose.

#### 3.5.2 Cross correlation function

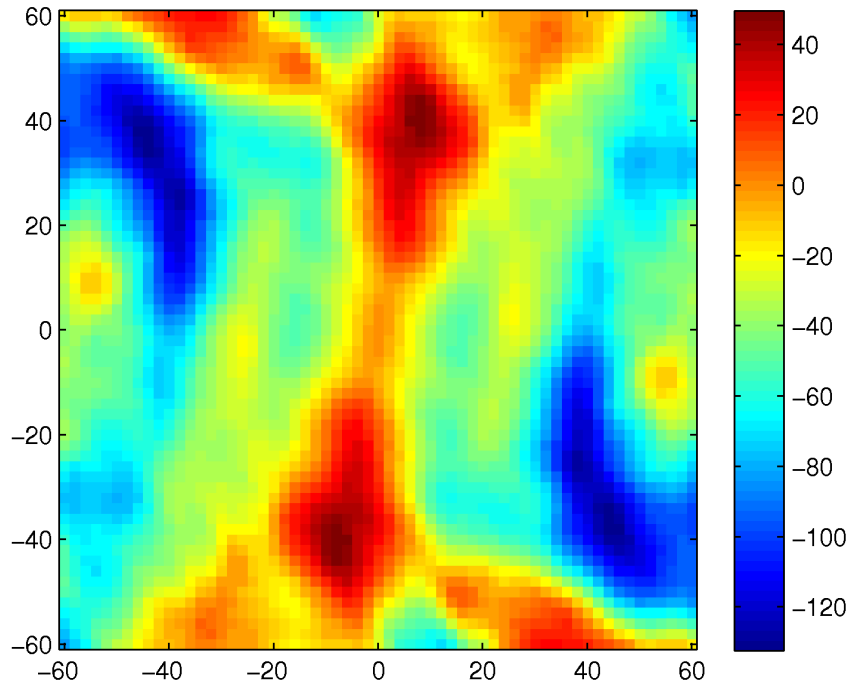
The cross covariance function  $C_{XY}(\mathbf{h})$  is defined under an hypothesis of second order stationarity [90], i.e., constant mean over the spatial domain:

$$\begin{cases} \mathbb{E} [\mathbf{X}(\mathbf{x})] = \mu_X \\ \mathbb{E} [\mathbf{Y}(\mathbf{x})] = \mu_Y \\ \mathbb{E} [(\mathbf{X}(\mathbf{x}) - \mu_X) \cdot (\mathbf{Y}(\mathbf{x} + \mathbf{h}) - \mu_Y)] = C_{XY} \end{cases} \quad (3.21)$$





**Figure 3.9:** (top) Satellite data and (bottom) radar data from time of heavy precipitation on August 20, 2007. The shower seen in the radar data seems to be shifted to the northeast in the satellite data.



**Figure 3.10:** *Cross variogram surface for scenario 2007-08-20 hotspot.*

The same is valid for the cross correlation function, which is merely the cross covariance function normed by the product of the two data sets standard deviations. This is calculated for the satellite data  $\mathbf{X}(\mathbf{x})$  and the spatially shifted radar data  $\mathbf{Y}(\mathbf{x} + \mathbf{h})$  as

$$\rho(\mathbf{X}(\mathbf{x}), \mathbf{Y}(\mathbf{x} + \mathbf{h})) = \frac{\text{cov}(\mathbf{X}(\mathbf{x}), \mathbf{Y}(\mathbf{x} + \mathbf{h}))}{\sigma_X \cdot \sigma_Y} \quad (3.22)$$

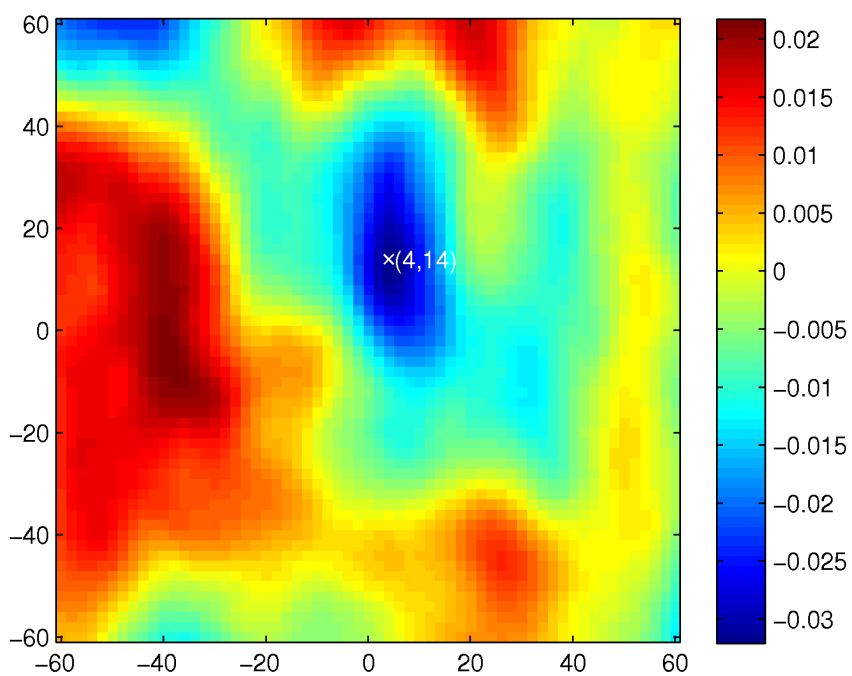
The cross correlation function has been calculated for  $\Delta x, \Delta y \in [-60, 60]$  pixels, hereby forming a surface as seen in Figure 3.11.

Heavy rain will in the radar data take on large positive values, while we suspect that PC1 of the satellite data contains low values for precipitating clouds. Therefore, a large negative value on the cross covariance surface indicates the necessary spatial displacement of the radar data in order to obtain maximum covariance with the satellite data. In the shown scenario, this suggests that the satellite data is approximately shifted  $\mathbf{h} = (4, 14)$  pixels northeast of the radar data. This is a ground distance of approximately 45km.

This indicates that the same phenomenon is present in the radar and satellite data, but the observed position of convection by the radar is inconsistent with what is seen from above by the satellite.

Cross correlation and variogram surfaces can be seen for the remaining scenarios in Appendix C.4.

In the following, it will be investigated how this knowledge of the cross correlation function can be used to identify the areas in the satellite data where heavy precipitation is certain, i.e., ground truth.



**Figure 3.11:** *Cross correlation surface for scenario 2007-08-20 hotspot.*



## Collecting ground truth using cross correlation minimization

Only some clouds develop into heavy convection. The goal is to identify exactly these clouds in an as early a stage as possible. To accomplish this, some form of “ground truth” must be collected.

Extreme convection is easy to identify in the radar data as it happens, by use of a simple thresholding. But when showers are visible on the radar, they are already occurring and hence already too late to give a warning. However, it can be used as a point of reference in collection of ground truth. One of the challenges is to determine the correspondence between the area of extreme convection in the radar data and the equivalent event in the satellite data.

### 4.1 Exhaustive search

This section presents a method that (i) identifies an area of heavy convection in the radar data at time  $t_0$  and (ii) determines the spatial translation, in terms of minimum cross correlation, from the radar data at time  $t_0$  to the satellite data at time  $t_i, i \in [-1, \dots, -K]$ , whereby tracking the convective area back in time. Here,  $-K$  represents the point in time of minimum cross correlation.

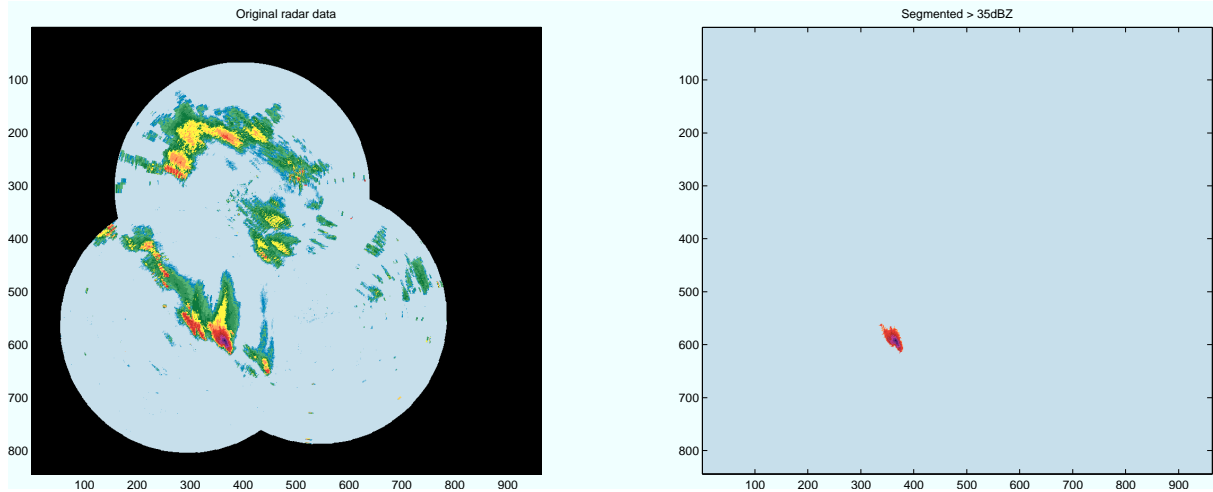
The convective area in the radar data is segmented using simple image analysis. The radar data  $\mathbf{y}$  is thresholded, such that only pixel values above  $\tau = 35$  dBZ are considered. The convective area of interest is determined as the connected component [14] in the binary image  $\mathbf{y}_\tau = (\mathbf{y} > \tau)$  with the largest area. This segmentation is illustrated in Figure 4.1. The segmented radar data at the time of each scenario’s hotspot can be seen in Appendix B.1.

The radar data  $\mathbf{y}(\mathbf{x})$  can be defined as a function of a spatial shift  $\mathbf{y}(\mathbf{x} + \Delta)$ , where  $\Delta = [\Delta x, \Delta y]$ . As described in Section 3.5, the cross correlation between the satellite data  $\mathbf{X}(\mathbf{x})$  and the spatially lagged radar data can be defined as a function of this lag:

$$\rho_\Delta(\mathbf{X}, \mathbf{y}) = \text{corr}\{\mathbf{X}(\mathbf{x}), \mathbf{y}(\mathbf{x} + \Delta)\} = \frac{\text{cov}(\mathbf{X}, \mathbf{y})}{\sigma_X \sigma_Y}. \quad (4.1)$$

This measure was found to be appropriate to determine the spatial displacement between the two data sets.

An outline of the method, where an exhaustive search for the minimum cross correlation is performed in each time step, is shown in Algorithm 1. It is seen how the spatial lag  $\mathbf{x}_0$  is updated



**Figure 4.1:** Original radar data and segmented convective area ( $\tau = 35$  dBZ).

on each iteration back in time, in order to search for minimum correlation in the surrounding neighborhood of the previous step.

---

**Algorithm 1** Exhaustive search for spatial shift of radar data to minimize cross correlation with satellite data. The number of spatial lags to examine are given by  $p_x$  and  $p_y$ .

---

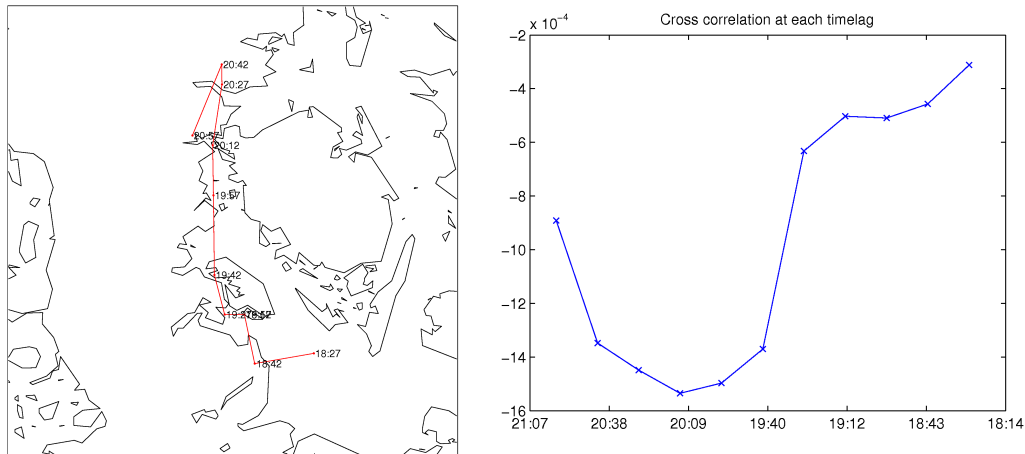
```

 $\mathbf{x}_0 = (\delta x_0, \delta y_0) = (0, 0)$ 
 $\mathbf{y} \leftarrow$  radar data at time  $t_0$ 
for  $i = 0$  to  $-K$  do
     $\mathbf{X} \leftarrow$  satellite data at time  $t_i$ 
    for  $\delta x \in [\delta x_0 - p_x, \delta x_0 + p_x]$  do
        for  $\delta y \in [\delta y_0 - p_x, \delta y_0 + p_x]$  do
             $\Delta = \mathbf{x}_0 + (\delta x, \delta y)$ 
             $\mathbf{P}(\delta x, \delta y) = \rho_{\Delta}(\mathbf{X}, \mathbf{y})$ 
        end for
    end for
    Set  $\mathbf{x}_{\min} = \arg \min_{\delta x, \delta y} \mathbf{P}(\delta x, \delta y)$ 
    Update  $\mathbf{x}_0 = \mathbf{x}_0 + \mathbf{x}_{\min}$ 
end for
    
```

---

This algorithm is exhaustive in the way that the entire surface of cross correlations are calculated in each time step. No intelligence of any sort is added in terms of pattern search, use of derivatives or similar. However, it is easy interpretable as only two variables are varied, being the spatial lag in  $x$  and  $y$  direction, which allows for easy visualization of the results.

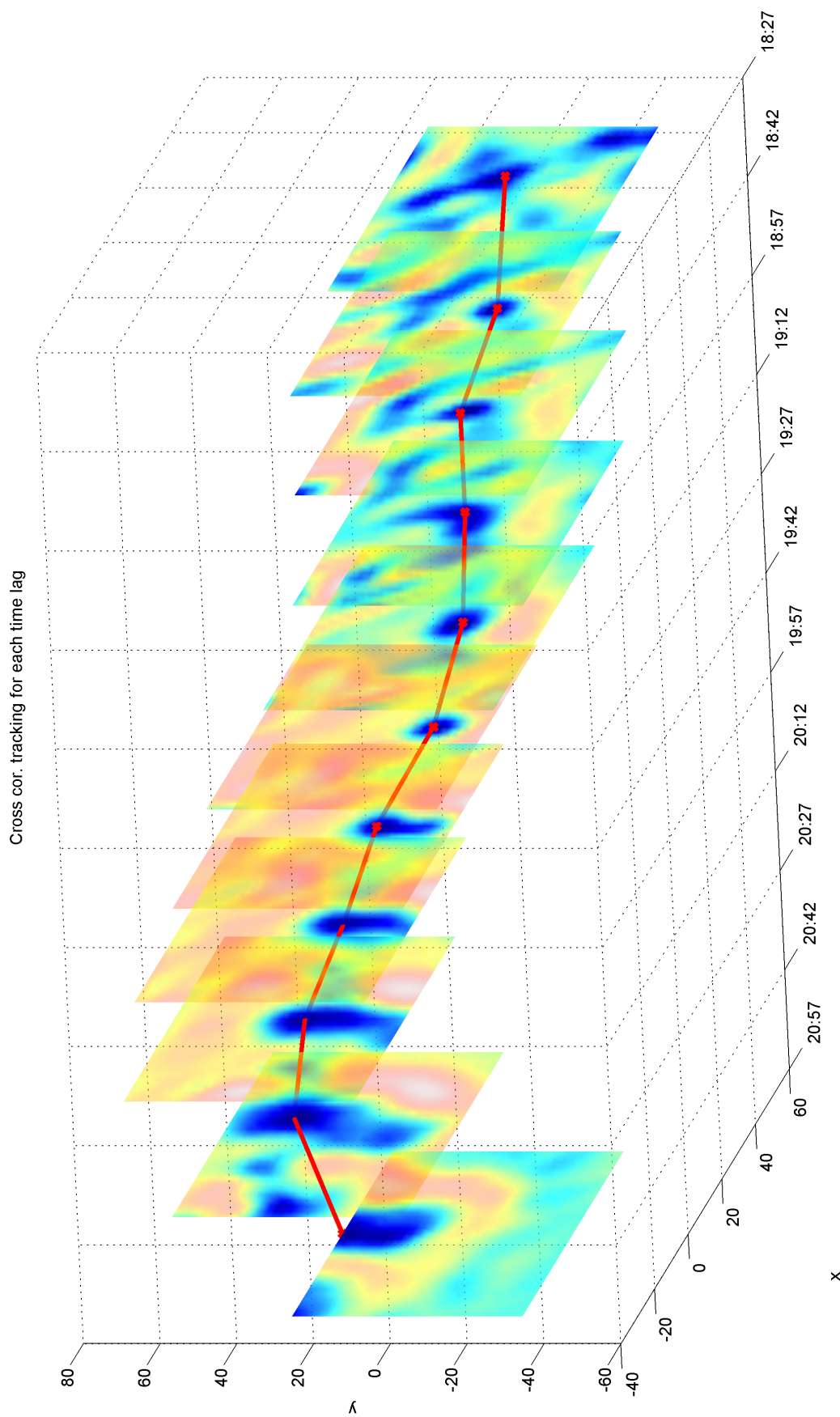
The result of this algorithm is the spatial shift in space and time needed to minimize cross correlation. In Figure 4.2, the minimum cross correlation in each time step, and the spatial steps taken are shown. From the left plot of the path, the tracking seems to have two phases: In the first two steps, the convective area of interest is located to the north east in the satellite data. Then, this area is followed back through time towards south, before it eventually takes a step to the east from 19.27 to 19.12 and a large step to the north east at 18.27. From the right plot, it is seen that minimum cross correlation is achieved at  $t_{-4}$ , i.e., one hour before the reference point. Figures illustrating results for the remaining scenarios can be seen in Appendix C.5.



**Figure 4.2:** 2007-08-20. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.

Another way of visualizing these results are by showing the calculated cross correlation surfaces for each time step as slices, as in Figure 4.3. This is only possible as the surface is a function of only two variables and the entire surface is calculated in this exhaustive procedure.

From inspection of Figure 4.3, it can be seen that the convective area can be tracked from 20:57 back to 19:27, before another convective area interferes at 19:12. This other convective area is apparently more developed at time 19:12, wherefore its resemblance with the “original” area at 20:57 are greater.



**Figure 4.3:** Cross correlation surfaces going back in time from left to right. A neighborhood of  $p_x = 30 \times p_y = 30$  pixels and  $K = 10$  were chosen.



This interpretation of the results requires manual inspection of the tracked path on top of the data, while simultaneously inspecting the cross correlation surfaces. This is especially due to the interference of other convective areas, as mentioned above, when going back in time. As not all convective areas are fully developed at the same time, another convective area can have greater resemblance with the radar data, at the time when the actual convective area shrinks. This can cause the method to “switch cloud” underway, which is difficult to identify from a single measure.

The results from this very simple approach to ground truth collection are not satisfying, as it only considers translation and not scale or rotation. As shown in the following section a need for a more flexible method exists.

## 4.2 Invariance to scale and rotation

To incorporate invariance to scale and rotation in the above method, a parameterized transformation of the radar data's pixel coordinates can be formulated. The radar data can be expressed as an function of the pixel coordinates, i.e.,  $\mathbf{y}(\mathbf{x})$ . Without loss of generality, the pixel coordinates can be expressed in homogeneous coordinates  $\mathbf{x}_h = [x, y, 1]^T$ . Using this formulation, various transformations of the coordinates can be expressed as matrix operations:

- Counter clockwise rotation with the angle  $\theta$ :

$$\mathbf{x}_r = \mathbf{R} \mathbf{x}_h = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.2)$$

- Scaling  $s_x$  in the x-direction and  $s_y$  in the y-direction:

$$\mathbf{x}_s = \mathbf{S} \mathbf{x}_h = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.3)$$

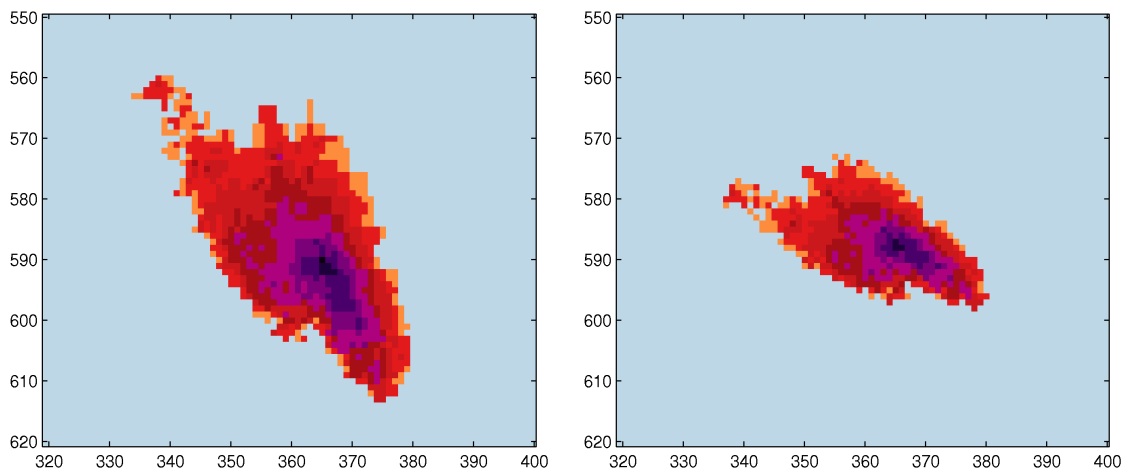
- Translating  $(t_x, t_y)$

$$\mathbf{x}_t = \mathbf{T} \mathbf{x}_h = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.4)$$

These three separate transformations can be combined to one by matrix multiplication, such that the complete transformation  $T$  of the homogeneous coordinates  $\mathbf{x}_h$  to transformed coordinates  $\hat{\mathbf{x}}_h$ , given the parameters  $\boldsymbol{\theta} = [\theta, s_x, s_y, t_x, t_y]$ , is

$$\hat{\mathbf{x}}_h = T(\mathbf{x}_h, \boldsymbol{\theta}) = \mathbf{T} \mathbf{S} \mathbf{R} \mathbf{x}_h . \quad (4.5)$$

An example of a transformed convective area can be seen in Figure 4.4.



**Figure 4.4:** 2007-08-20. Convective cloud in radar data transformed using  $\boldsymbol{\theta} = [30, 0.7, 0.7, 0, 0]$

Hereby, the set of parameters providing the transformation of the radar data which gives the minimum cross correlation with the satellite data can be written as a minimization problem:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} [\text{corr} \{ \mathbf{X}, \mathbf{y}(T(\mathbf{x}_h, \boldsymbol{\theta})) \}] \quad (4.6)$$

For minimizations of differences between pure pixel values, this can be formulated as a linear minimization problem [22], but minimization of the cross correlation is a nonlinear optimization problem. Algorithm 1 is improved with this invariance in Algorithm 2.

---

**Algorithm 2** Minimization of cross correlation incorporating a transformation function of the radar data. The minimization problem can be solved with a nonlinear optimization method, e.g., Simplex.

---

$\boldsymbol{\theta}_0 = (\theta_0, s_0, t_{x0}, t_{y0}) = (0, 1, 0, 0)$

$\mathbf{y} \leftarrow$  radar data at time  $t_0$

**for**  $i = 0$  to  $-K$  **do**

$\mathbf{X} \leftarrow$  satellite data at time  $t_i$

    Set  $\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} [\text{corr} \{ \mathbf{X}, f(\mathbf{y}; \boldsymbol{\theta}) \}]$  { $f$  is transformation of  $\mathbf{y}$  described in (4.5)}

$\mathbf{y} \leftarrow f(\mathbf{y}; \boldsymbol{\theta}^*)$

    Update  $\boldsymbol{\theta}_0$  with  $\boldsymbol{\theta}^*$

**end for**

---

Some nonlinear optimization algorithms, e.g., Gauss-Newton [69], can also exploit knowledge of the Jacobian. This knowledge can either be formulated directly, i.e., analytically, or indirectly via numerical estimates. As the Jacobian cannot be formulated analytically for this problem, numerical estimates are the only option. A method for estimating these is described in [69, p. 27].

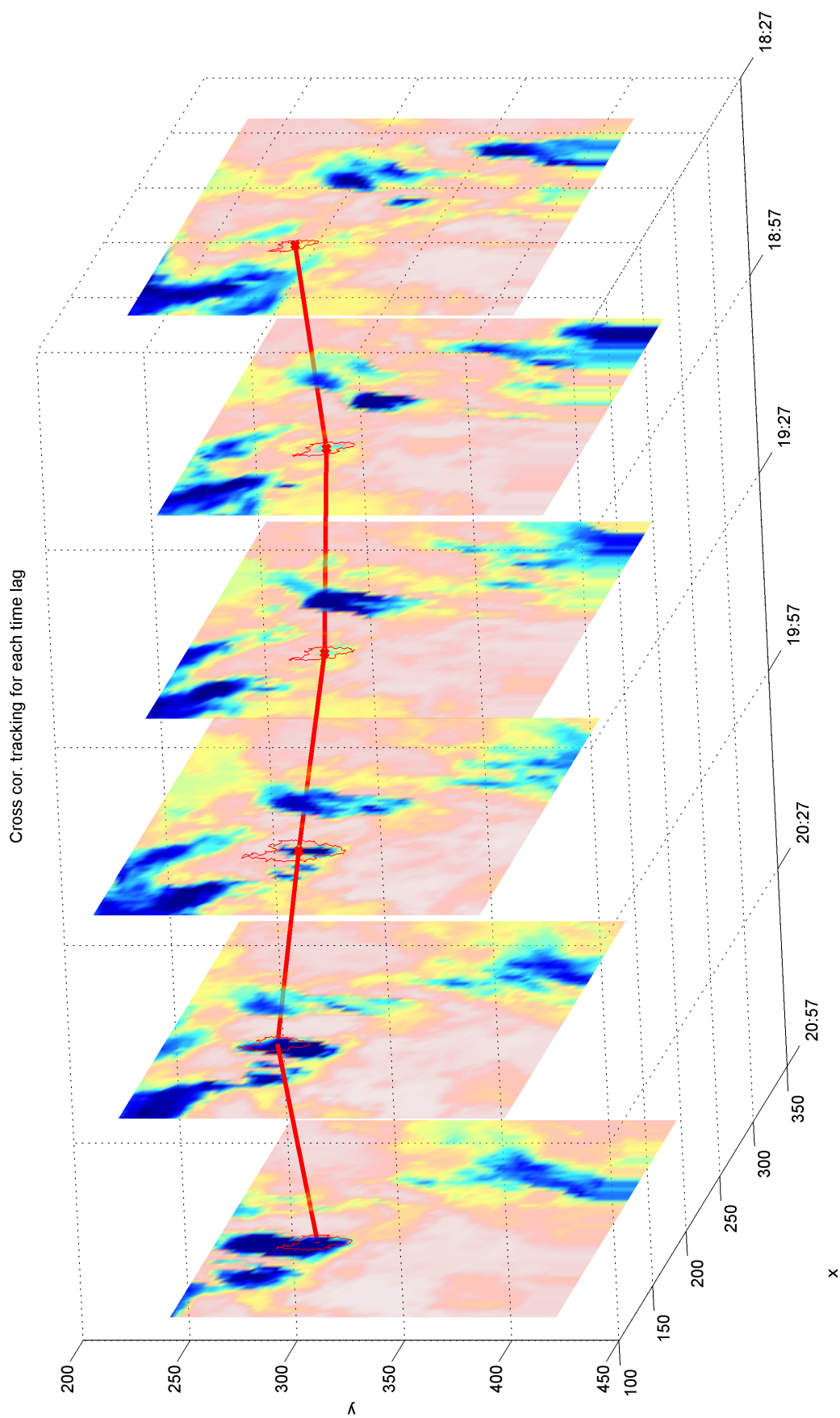
It was found that estimating the Jacobian did not perform as expected. The method producing best results was the pattern search method Simplex [64] with bounds on the variables [16].

The parameters used for the minimization are listed in Table 4.1. These parameters were chosen by considering the physical properties, such as expected cloud speed and growth, and by experimenting with different combinations. The chosen combination was found to provide acceptable results for all scenarios.

|               | $\theta$ | $s$ | $t_x$ | $t_y$ | Stopping criteria |           |
|---------------|----------|-----|-------|-------|-------------------|-----------|
| Initial value | 0        | 1   | 0     | 0     | TolX              | $10^{-1}$ |
| Lower bound   | -30      | 0.5 | -30   | -30   | TolFun            | $10^{-1}$ |
| Upper bound   | 30       | 1.5 | 30    | 30    |                   |           |

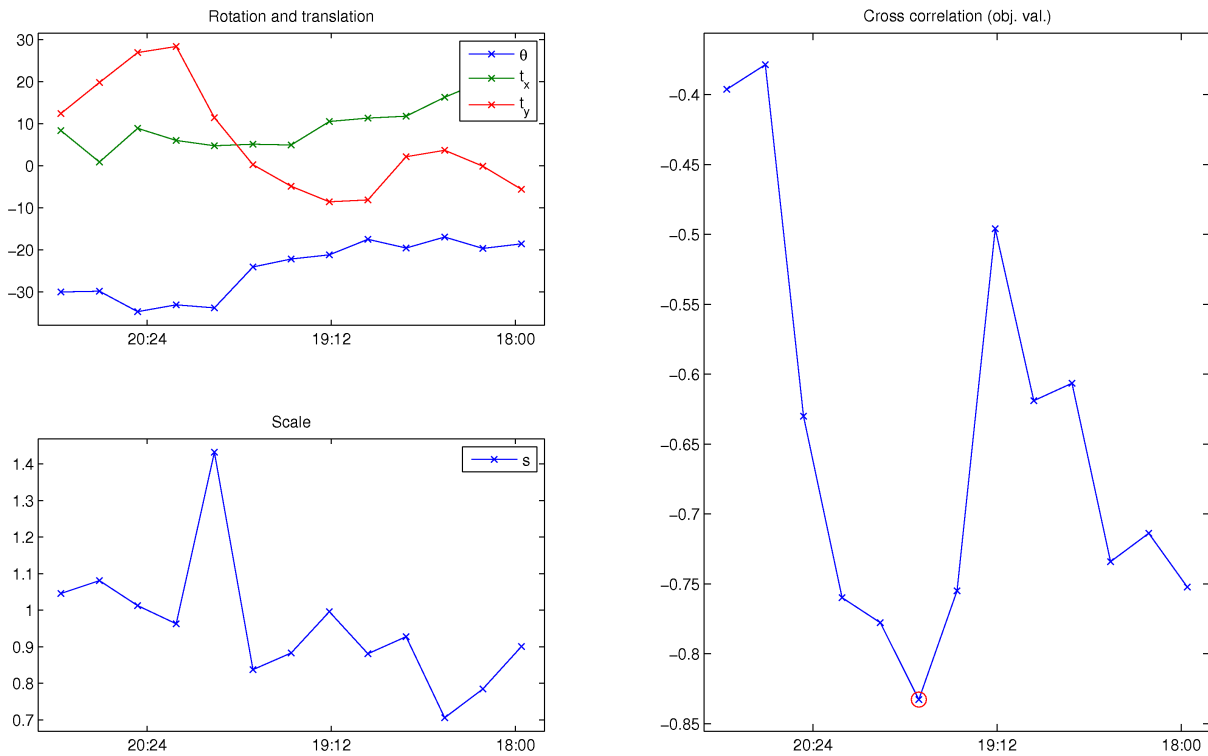
**Table 4.1:** Start criterion, variable bounds and optimization method parameters used for cross correlation minimization.

In Figure 4.5 the tracking pattern can be seen for ten time lags back in time. The slices shown here are the first principal component of the data at that time point, as opposed to Figure 4.5 where the cross covariance surfaces were available.



**Figure 4.5:** First PC in each time lag going back in time from left to right. Only every second time lag is shown as a slice. The boundary of the transformed radar data is shown in red on each slice. The method described in Algorithm 2 was used.

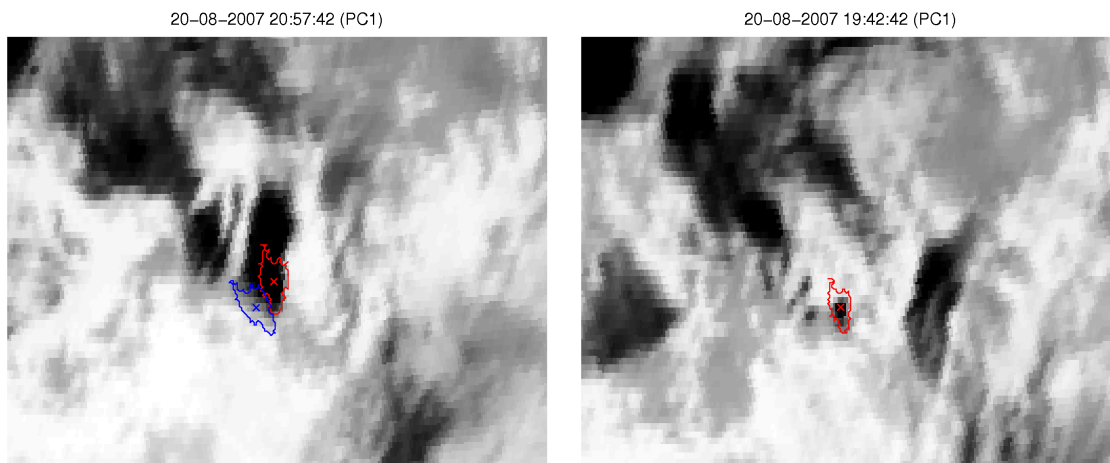
The final values of  $\theta$  for each time step can be seen in Figure 4.6, where the objective value is also shown. Optimal parameters and the path back in time for all scenarios can be found in Appendix C.6. The point marked with a red circle is the chosen point of minimum cross correlation. This point was chosen by simultaneous visual inspection of the tracking result overlaid in the satellite imagery and the objective value itself. The number of time lags to track back in time were deliberately chosen to exceed this point to illustrate the dynamics of the parameters and the objective value. The time span of the cloud development has also been taken into consideration individually for each scenario.



**Figure 4.6:** (left) Transformation parameters chosen for minimum cross correlation in each time step. (right) Cross correlation for optimal set of parameters in each time step, with chosen point of minimum cross correlation marked with a red circle.

Figure 4.7 shows the boundary of the transformed radar data in the initial time step and at the lag of minimum cross correlation. Note that only a sub-region of the data is shown for illustration purposes. This can be seen for the remaining scenarios in Appendix C.7. It is seen that the transformed radar data in some cases grow beyond the cloud that would be expected to be tracked. Since cross correlation is used as a measure for tracking, there is not guarantee that the tracking will constrain itself to a single cloud. If two clouds are near each other, the method might find it feasible to expand the radar data to cover both these areas, even though it has no apparent physical meaning.

The number of function iterations and function evaluations are shown in Table 4.2. The number of iterations for the optimization procedure applied to each scenario can be seen graphically in Appendix C.8.



**Figure 4.7:** Boundary of optimal transformation of radar data shown in satellite data. (left) The initial state, where untransformed radar data are shown in blue and the optimal transformation in red. (right) Transformation at time of minimum cross correlation.

| Time  | Lag | Iterations | Fun. evals. |
|-------|-----|------------|-------------|
| 20:57 | 0   | 23         | 54          |
| 20:42 | -1  | 27         | 65          |
| 20:27 | -2  | 40         | 85          |
| 20:12 | -3  | 42         | 82          |
| 19:57 | -4  | 42         | 86          |
| 19:42 | -5  | 56         | 113         |
| 19:27 | -6  | 34         | 72          |
| 19:12 | -7  | 29         | 68          |
| 18:57 | -8  | 24         | 66          |
| 18:42 | -9  | 39         | 82          |
| 18:27 | -10 | 28         | 65          |
| 18:12 | -11 | 29         | 62          |
| 17:57 | -12 | 33         | 71          |

**Table 4.2:** 2007-08-20. Iterations and function evaluations of the simplex method for twelve time steps back.

The result of this optimization is a link between the area of extreme convection in the radar data, where it is easily identified, and the corresponding phenomenon in the satellite data. This method is invariant to scale, rotation, translation and takes into account the possible temporal displacement of the event from the radar data to the satellite data.

This method does, however, not give any information on the shape of the cloud in the satellite data. It only gives the shape of the transformed radar data at a time of minimum cross correlation. As noted earlier, this can in some cases have caused a transformation where the radar data covers more than one cloud, or – due to the inhomogeneity of the segmented radar data – a skew coverage, i.e., where only a part of the transformed data covers what is found to be the cloud of interest in the radar data. For instance, this is seen in Figure C.38.

To finalize the collection of ground truth, a simple method for segmenting the clouds of interest in the satellite data will be developed, where the result from this section will play an important role.

### 4.3 Simple segmentation and tracking in projection space

The ideal collection of ground truth would in this context yield an image where convective clouds resulting in heavy precipitation are marked. By using cross correlation as a measure, the previous sections have found a link from the the clouds of interest in the radar data to an area in the satellite data. Here, it will be investigated how to use this information to segment the clouds of interest in the satellite data through their life cycle, i.e., from birth till death.

#### 4.3.1 Segmentation

Segmentation of an object is simplified if the data are first projected to an appropriate subspace. This subspace should ideally discriminate objects of interest from background and other objects. Having aligned the event in the radar data with the corresponding event in the satellite data, it becomes possible to perform a CCA, as described in Section 3.4. The alignment is important to ensure that the resulting projection space maximizes correlation between reflectance of heavy rain and brightness temperatures from cloud tops of the cumulonimbi delivering this precipitation. The average correlation, calculated as described in (3.12), increased from 0.51 to 0.83 by performing this adjustment.

The analysis has been performed in two variations: (i) Separately for each scenario and (ii) collectively for all scenarios of one type. Here, the heat thunder scenarios are analyzed. Intuitively, an analysis of each separate case will yield a good subspace for each case, but lack generalizability to other scenarios. The canonical correlations are shown for the three scenarios analyzed separately and collectively in Figure 4.8. The average correlation when using the collective subspace is 0.65.

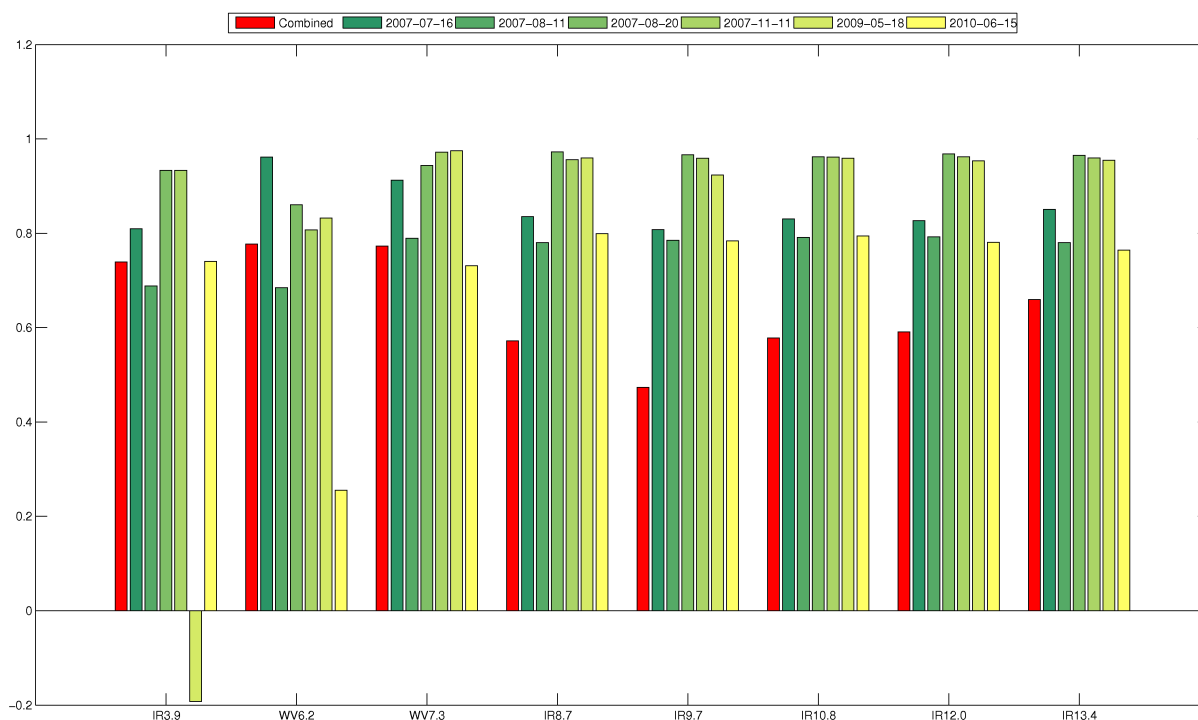
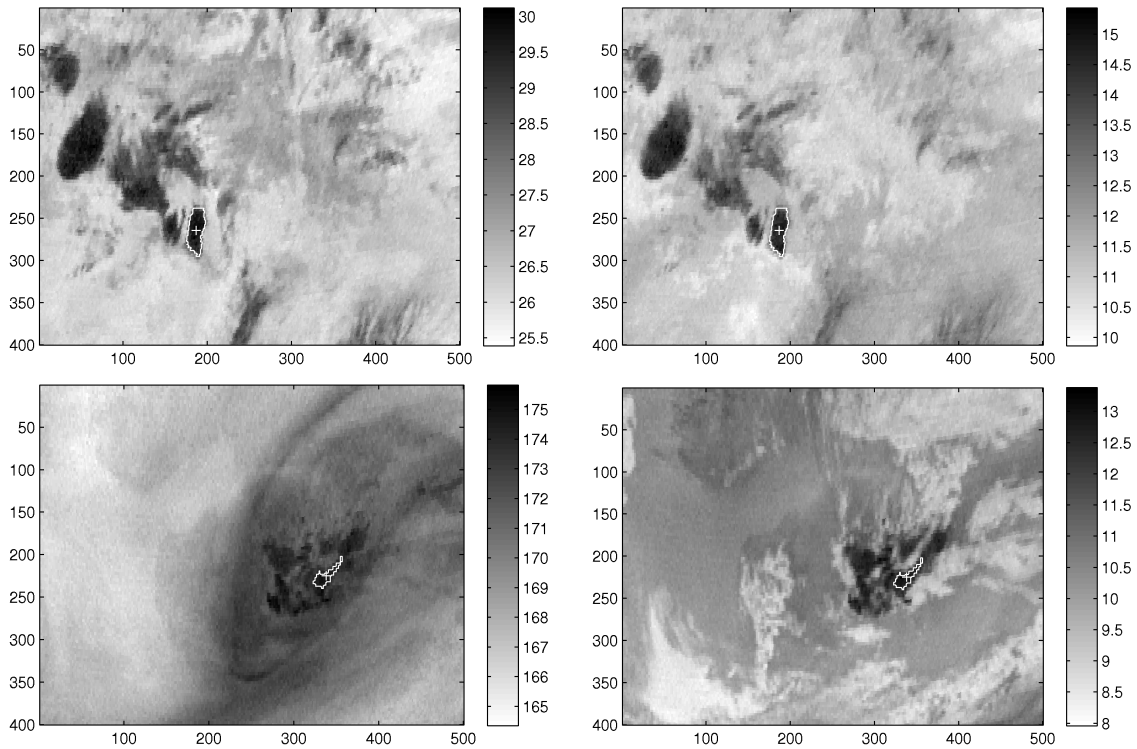


Figure 4.8: Canonical correlations for CCA of three heat thunder scenarios separately and collectively.



### 4.3 Simple segmentation and tracking in projection space

It is seen that the canonical correlations are very similar for most of the bands for the separate analyses. A few correlations, however, attain different values than the others. Especially the first band for scenario 2009-05-18 and the second band for scenario 2010-06-15 stick out as different from the other scenarios. To investigate what influence this has on the subspace's separating capabilities, the data at the time of the hotspot for one of these scenarios and for a scenario exhibiting "normal" behavior are projected into the subspace produced by the collective analysis, as well as the subspace from the individual analyses. This is shown in Figure 4.9.



**Figure 4.9:** Scenarios 2007-08-20 (top) and 2010-06-15 (bottom). The data at the time of the hotspot are projected into the scenario's individually produced subspace (left) and the collectively produced subspace (right).

For the first scenario, the result is as expected: The individually produced subspace provides a less smudged image of the data and hence a better subspace for segmentation. However, for the other scenario, it is seen that the individual subspace is in fact worse than the collective one. This may be caused by more dissimilarities between the radar data and the satellite data, than what can be corrected by the cross correlation alignment. For instance, the cloud top brightness temperatures are recorded by the satellite as a lot of small clusters, where the radar captures a coherent area of precipitation.

This illustrates how the collective subspace is a form of average of all scenarios. It performs worse for some scenarios, and better for others. Thus it is more robust, which is also what would be intuitively expected. Therefore, segmentation and tracking of the cloud of interest is going to be done in the collective subspace, rather than a separate one for each scenario. Regardless, it was found that an individual threshold was necessary in order to do the segmentation properly in all cases, even though the subspace is the same. This suggests that the heavy precipitating clouds are not completely similar across all scenarios. This would also be quite surprising, as it is real world data and variations are to be expected. The chosen threshold can be seen in Table 4.3.

| Scenario   | $\tau$ |
|------------|--------|
| 2007-07-16 | 15     |
| 2007-08-11 | 12.5   |
| 2007-08-20 | 13     |
| 2007-11-11 | 11     |
| 2009-05-18 | 12     |
| 2010-06-15 | 12.5   |

**Table 4.3:** Chosen thresholds for segmentation in CCA space for each scenario.

These thresholds are used for a simple binary segmentation, where areas with values above the tolerance are given the value 1 and all other 0. The segmented image can be used to identify a number of connected components [14], which ideally are convective clouds. At time lag  $t_i$  this number is denoted  $N_i$ . For this project, only clouds which are going to develop into heavy precipitation are interesting. In fact, only one cloud is of interest in each scenario, namely the one identified in the radar data the time of the hotspot described in 2.1. Therefore, it is necessary to identify this cloud from one time lag to the next, going both back and forth in time.

### 4.3.2 Simple tracking

To identify – track – a single cloud from one satellite image to the next, a distance function  $\mathcal{D}\{C_\star, C_j\}$  is introduced. This function describes the distance – dissimilarity – between the known cloud of interest  $C_\star$ , and the cloud  $C_j$ , where  $j \in [1, N_i]$ . The distance function can be imagined to take different measures into account, e.g., spectral signature, distance, speed, shape or similar. With this formulation, the most similar cloud in the preceding time lag  $t_{i-1}$  can be found as  $C_{j^\star}^{i-1}$ , where

$$j^\star = \arg \min_{j \in [1, N_{i-1}]} \mathcal{D}\{C_\star, C_j^{i-1}\} . \quad (4.7)$$

The subscript here denotes the cloud index and superscript the time lag. This minimum distance will be denoted  $\mathcal{D}_{j^\star}$ .

To incorporate the possibility that the cloud terminates, i.e., dies when tracking forward or reaches the step before birth when tracking backward through time, a termination function  $\mathcal{T}$  is introduced. This is used as a stopping criterion for the simple tracking method, such that the method stops if the cloud with the minimum distance causes the termination function to evaluate to logical true:

$$\text{stop?} = \mathcal{T}(C_\star) . \quad (4.8)$$

The termination function must of course be chosen in relation to the distance function and should be based on expected physical behavior by the clouds. Using these definitions a simple tracking method is formulated in Algorithm 3.

A simple distance function is an estimate of the needed speed for one cloud to have moved to another cloud. This can be calculated as the euclidean distance between to cloud centroids divided by the sampling time  $\Delta t$ :

$$\mathcal{D}_{\text{speed}} = \frac{1}{\Delta t} \cdot \left\| \mathbf{c}_\star^i - \mathbf{c}_j^{i-1} \right\|_2 \quad (4.9)$$

---

**Algorithm 3** Simple tracking formulation with minimization of distance function  $\mathcal{D}$  and termination function  $\mathcal{T}$  as stopping criterion.

---

**Require:** Initial time lag  $t_k$  {Obtained using e.g., procedure described in Section 4.2}

**Require:** direction  $\in \{-1, 1\}$

$C_\star \leftarrow$  boundary of transformed radar data at time  $t_k$

$i = k$

**while**  $\mathcal{T}(C_\star) == \text{false}$  **do**

$\{C_j^i\}, j \in [1, N_i] \leftarrow$  segmented clouds at time  $t_i$

$C_\star = \arg \min_{C_j^i} \mathcal{D}\{C_\star, C_j^i\}$

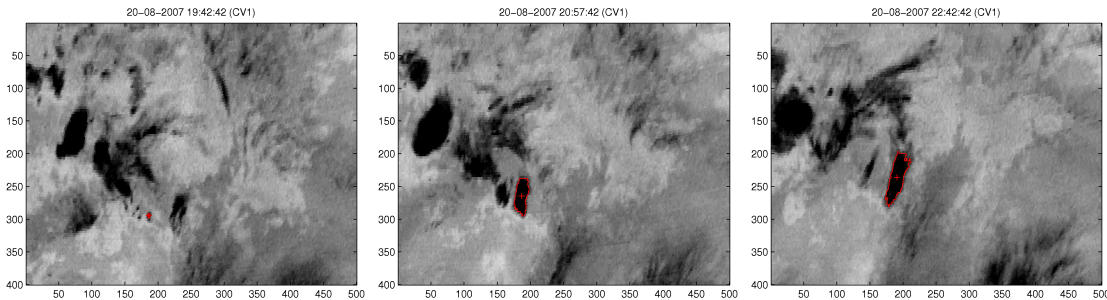
$i = i + \text{direction}$

**end while**

---

where  $\mathbf{c}_j^{i-1}$  represents the centroid coordinate vector of cloud  $j$  at time  $t_{i-1}$  and  $\mathbf{c}_\star$  the centroid of the previously identified cloud. This is of course a very simple formulation, but it was found to be adequate for the purposes here. Other possible distance function will therefore not be pursued here.

A tracked cloud with boundary can be seen in Figure 4.10. The times chosen to display are (i) when the cloud is being born, (ii) the hotspot time and (iii) two hours after the hotspot. These can also be seen for the remaining scenarios in Appendix C.9.



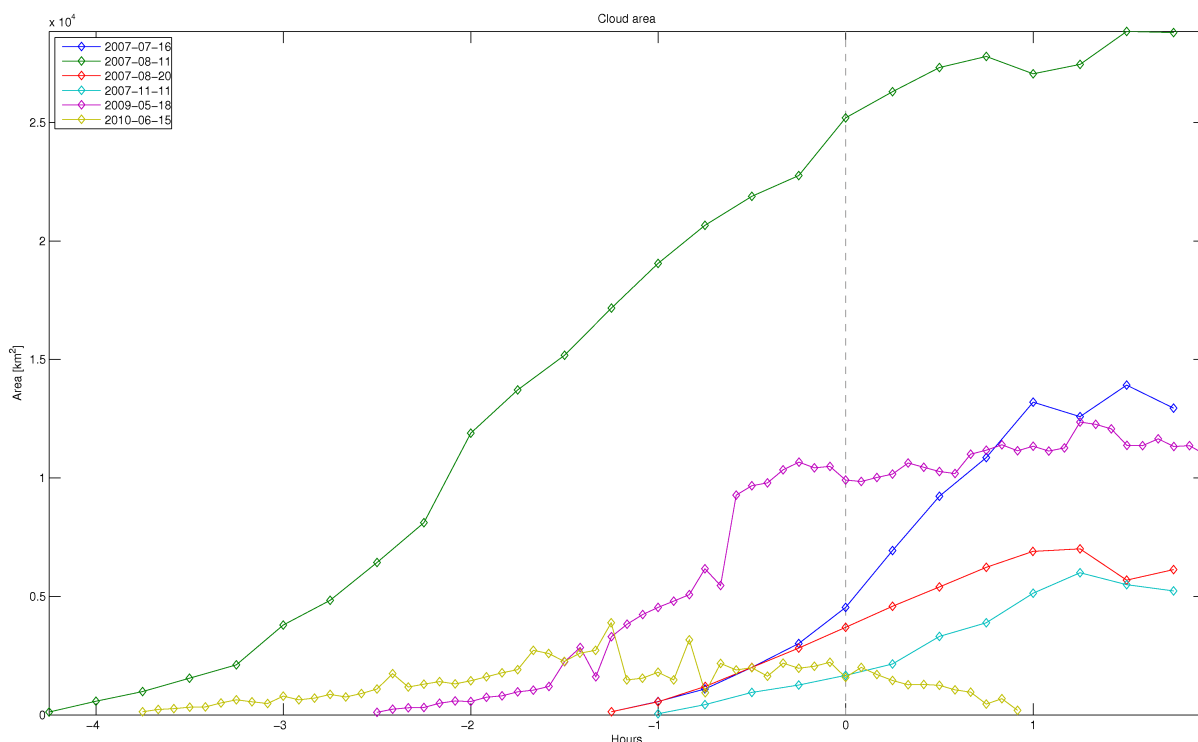
**Figure 4.10:** 2007-08-20. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.

For most of these scenarios, the tracking procedure seems to solve the task. The same cloud is marked in each step and the segmented boundary follows what seems to be the cloud boundary. As expected, the better a separation between convective and non-convective areas provided by the subspace, the easier it is to choose the threshold appropriately and thus provide a better segmentation. Also visible is the difference between whether the cloud is isolated as in scenarios 2007-08-11 and 2007-08-20, or close to other convective areas as is the case in scenario 2009-05-18 for instance.

The only scenario providing doubtful tracking results is scenario 2010-06-15, as merges and splits of clouds constantly occur when tracking through time, which is caused by this very fragmented convective system, i.e., a lot of small separated clouds. Even though the precipitation in the radar data seem very spatially coherent, the cloud top temperatures recorded in the satellite images provides a more scattered image.

### 4.3.3 Collecting features

Having tracked each cloud back to birth allows for collection of features during its life cycle. Specifically, each cloud's area has been recorded during the tracking and is shown in Figure 4.11. First of all, it is seen that the lifetime of each cloud varies a lot. Also noteworthy, is, that the scales are very different between clouds. For instance, the cloud from 2007-11-11 reaches only approximately one tenth of the area of the cloud from scenario 2007-08-11. For these two clouds, the times of birth before the hotspot are also very different (1 and 4.5 hours respectively).

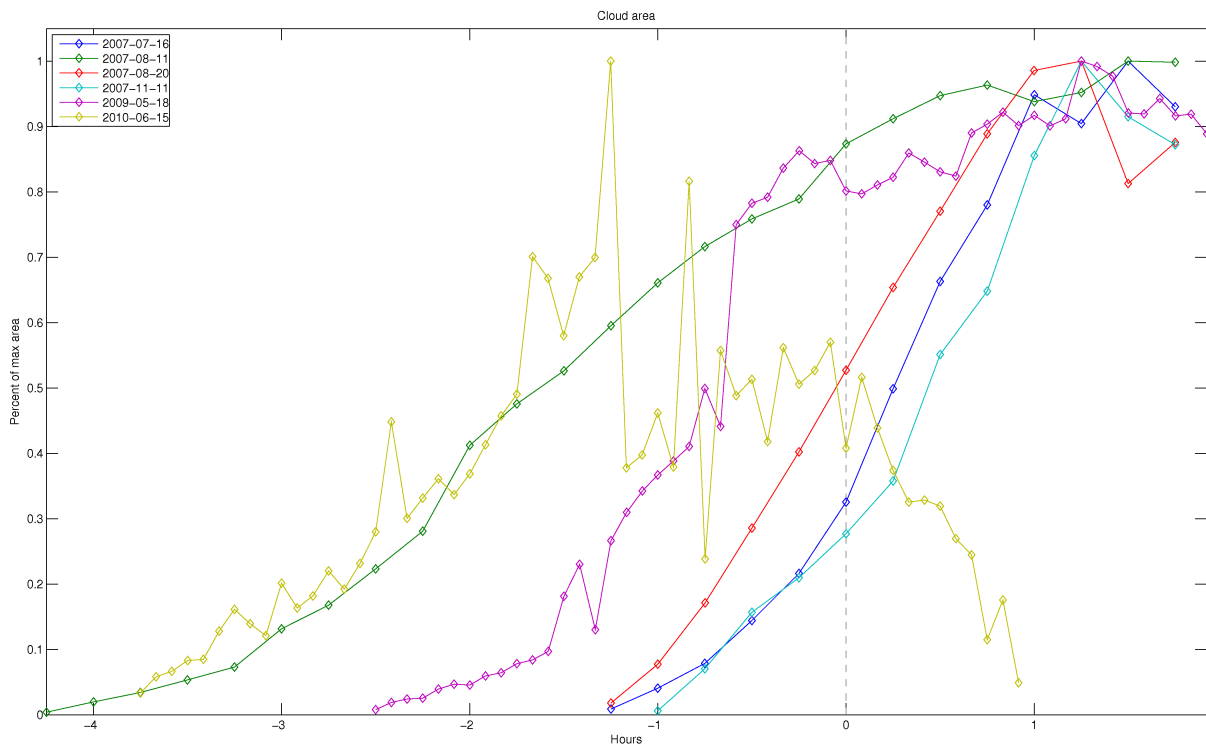


**Figure 4.11:** Area in square kilometers for each scenario's heavy precipitating cloud from birth till two hours after the hotspot. 0 hours is each scenario's chosen hotspot.

In Figure 4.12, each cloud's area is shown as a percentage of this cloud's maximum area. Curiously, it is seen that for most of the clouds, the maximum area is reached approximately 1 hour after the hotspot. This is probably due to the build-up of the anvil, which starts when the cumulus nimbus is mature, and apparently peaks in area a short time after the time of maximum precipitation.

A large jump in area between two time lags is typically because the cloud merges with another cloud or splits into two clouds. This is especially visible for the last scenario 2010-15-06, where a lot of mergers and splits occur, and scenario 2009-18-05, where the precipitating cloud merges with another cloud approximately 45 minutes before the hotspot. The remaining, more isolated, clouds do not exhibit this behavior in the same degree, where the area curves are smoother through the cloud's life time. Especially scenarios 2007-17-16, 2007-08-20 and 2007-11-11 have similar development when considering the plot, where area is plotted as a percent of maximum area. They start development approximately one hour before the time of maximum precipitation.

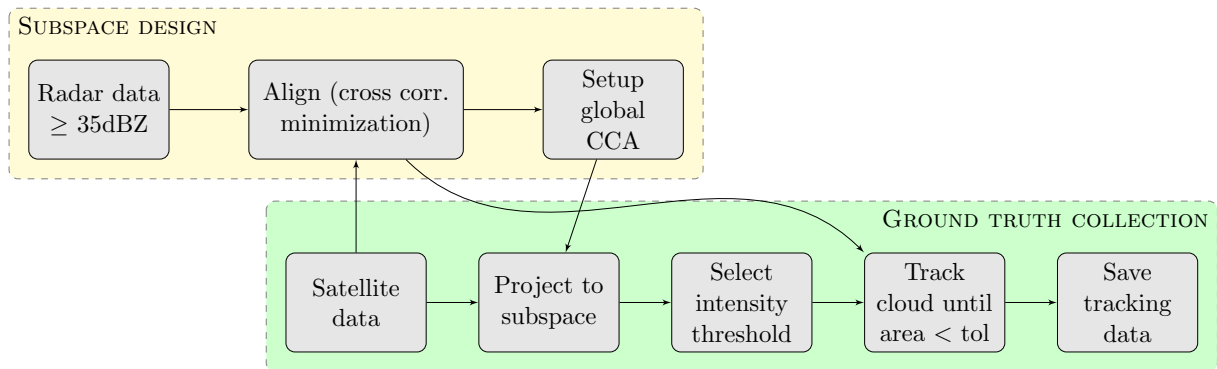
### 4.3 Simple segmentation and tracking in projection space



**Figure 4.12:** Area in percent of maximum area for each scenario's heavy precipitating cloud from birth till two hours after the hotspot. 0 hours is each scenario's chosen hotspot.

## 4.4 Collected ground truth

The product from this chapter is a set of methods that use radar data in order to collect ground truth for the satellite data. Ground truth is here defined as clouds causing heavy precipitation. The sequence of methods is summarized in Figure 4.13, where it can be seen that the radar data is used to setup the subspace in which the cloud tracking is performed as described in Section 4.3.1.



**Figure 4.13:** Sequence of methods for collection of ground truth. The radar data is used to setup the subspace in which the convective cloud is tracked in the satellite data

The ground truth collection commences by projecting the satellite data to the designed subspace. An intensity threshold is used to segment the image, also described in Section 4.3.1. The tracking procedure takes its starting point from the spatially aligned radar data to ensure that it is the area of heavy precipitation being tracked. The cloud is tracked back through time until it is smaller than a given threshold. The tracking is described in Section 4.3.2.

The result from this procedure is a data set, possibly interesting for other purposes. Therefore a few specifications are given here: The data set consists of six time series of satellite images; one set from each scenario's hotspot going back through time, where an area of interest has been marked in each time step until disappearance. Each satellite image consist of 8 IR channels, with a pixel resolution of  $400 \times 500$ . Radar data in the same grid are also available. The number of time steps, where a cloud has been tracked, and the sample time are reported in Table 4.4.

| Scenario   | $\Delta t$ | $N_t$ | 1/0 ratio |
|------------|------------|-------|-----------|
| 2007-07-16 | 15         | 6     | 0.015     |
| 2007-08-11 | 15         | 18    | 0.064     |
| 2007-08-20 | 15         | 6     | 0.0093    |
| 2007-11-11 | 15         | 5     | 0.0054    |
| 2009-05-18 | 5          | 31    | 0.023     |
| 2010-06-15 | 5          | 46    | 0.0041    |

**Table 4.4:** Sampling time  $\Delta t$  between each time step, number of time steps the cloud is tracked  $N_t$  and the ratio of class 1 (rain) relative to class 0 (no-rain) for each scenario.

By labeling the interior of the tracked clouds with a 1 and everything else with a 0, a binary label image have been set up for each of the images. The ratio of class 1 compared to class 0 at

the time of the hotspot is also reported in Table 4.4 as this will be of importance later.

It should be noted that the ground truth collected here is only “approximate” ground truth, as the collection is based on statistical measures and visual verification and not absolute certainty that the labeling is correct.

This data set will be used when classifying and predicting convective areas in the next chapter.





## Nowcasting using learned dictionaries

The concept of a dictionary is introduced when assuming that a signal  $\mathbf{x}_i \in \mathbb{R}^n$  can be explained by

$$\mathbf{x}_i = \mathbf{D} \mathbf{a}_i \quad (5.1)$$

where  $\mathbf{D} \in \mathbb{R}^{n \times k}$  is a dictionary composed of  $k$  elements (referred to as atoms) and  $\mathbf{a}_i$  is a sparse vector, i.e., a vector with as many zeros as possible. This can be formulated as a minimization problem over  $M$  image patches [52]:

$$\min_{\mathbf{a}, \mathbf{D}} \sum_{i=1}^M \|\mathbf{x}_i - \mathbf{D} \mathbf{a}_i\|_2^2 \quad \text{s.t.} \quad \|\mathbf{a}_i\|_0 \leq L. \quad (5.2)$$

Here  $L$  is a so-called sparsity factor, constraining the number of non-zero elements of  $\mathbf{a}_i$ , whereby inducing sparsity. One of the main questions is how to choose the contents of the dictionary.

Previously, dictionaries consisting of predefined parametric functions such as wavelets, curvelets, and more have been proposed [73, 53]. While these methods are computationally fast, they have certain shortcomings in their ability to provide a good sparse representation of the wide variety of signals encountered in image analysis. Another approach is to build a dictionary for the specific type of signal. This learning of nonparametric dictionaries has shown remarkable results within different image analysis tasks, for instance image inpainting, denoising, compression, and more [21, 81].

For a well built dictionary, the sparse representation in (5.1) will be easily interpretable, meaning it will be easy to identify which elements of the dictionary contributes to the received signal. To accomplish this, the dictionary should be build in such a way that the elements distinguishes themselves from each other, i.e., the dictionary should have a high discriminative power. Learning discriminative dictionaries are treated in [52], where the K-SVD algorithm for learning dictionaries [2] is used to build a dictionary for each class. Each dictionary is specifically designed to provide a “good” reconstruction of this class and a “bad” reconstruction of others.

The use of learned discriminative dictionaries for supervised classification is further addressed in [15], where a single dictionary is built for the entire image. A label dictionary, built alongside the image patch dictionary, is used to infer label probabilities. This approach will be used here to classify precipitation events and it will be investigated whether this method can be used for short-term prediction of these events. In the following, the method will be summarized and its use exemplified before it is applied to the data set.

## 5.1 Learning a discriminative dictionary

A dictionary  $\mathbf{D} \in \mathbb{R}^{N \times m}$  consists of  $m$  dictionary atoms column wise, such that  $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m]$ . The atom size  $N$  will be discussed in a short while. A dictionary has an associated label dictionary  $\mathbf{L} \in [0; 1]^{n \times m \times l}$ , where  $l$  is the number of classes, with label atoms, such that  $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_m]$ . The label atoms contain probabilities of a pixel belonging to a class, i.e.,  $\sum_{k=1}^l \mathbf{l}_k(g) = 1$  where  $g \in \{1, \dots, n\}$  denotes the pixel in the atom.

The dictionary is built in two stages: Initialization and optimization. The initialization is done by selecting a large number of random image patches  $\eta$  and building a dictionary of dissimilar atoms. The most similar atom  $\mathbf{d}_i^*$  to an image patch  $\mathbf{x}_i$  is found as

$$\mathbf{d}_i^* = \arg \min_{\mathbf{d}_i \in \mathbf{D}} \left[ \sum_{g=1}^n (\mathbf{d}_i - \mathbf{x}_i)^2 \right] \quad (5.3)$$

and the discriminative measure  $w$  is calculated from the image patch's label information  $\mathbf{y}_i$  and the nearest label atom  $\mathbf{l}_i^*$  as

$$w = 1 - \left( \frac{1}{n} \sum_{g=1}^n \sum_{k=1}^l \|\mathbf{y}_{ik} - \mathbf{l}_i^*\|_1 \right) \quad (5.4)$$

An image patch is added as a new dictionary element if it is “dissimilar enough” from atoms in the existing dictionary, here chosen as  $w \leq w_{tol}$  with a standard value of  $w_{tol} = 0$  unless otherwise stated. If not, the image patch is included in the atom which it resembles the most. This update step is similar to the update procedure in K-means [51, 7]. The initialization procedure forms a dictionary based on similarity.

A small modification is made to this part of the method in the experiments in this thesis. As the data is binary with a rather large proportion of zeros compared to the number of ones (See Table 4.4), i.e., no-rain compared to heavy rain, it was chosen to select all patches within class 1 and  $\eta$  random patches within class 0. This is to ensure that a maximum number of positive responses are available to the dictionary learning method. If this modification was not made, the positive responses would be under represented in the initial dictionary and eventually the method's ability to recognize positive responses would degrade. Furthermore, only patches within the radar mask are included in the initial step, as ground truth are not available outside the radar's coverage area.

The optimization of the dictionary is the most important part in building the dictionary. Here, the discriminative power of the dictionary elements are increased through a number of iterations,  $N_{iter}$ . In each iteration, the dictionary atoms are optimized by dividing the image patches into two groups: Similar and dissimilar. This is based on the image patches' discriminative power, calculated as in Equation (5.4) by using an “ideal” label atom  $\hat{\mathbf{l}}$  instead of the nearest label atom. An ideal label atom is defined as one for the class, for which it has the highest probability and zero otherwise, i.e., it becomes a binary label atom:

$$\hat{\mathbf{l}}_k(g) = \left( k == \max_k \mathbf{l}_k(g) \right) . \quad (5.5)$$

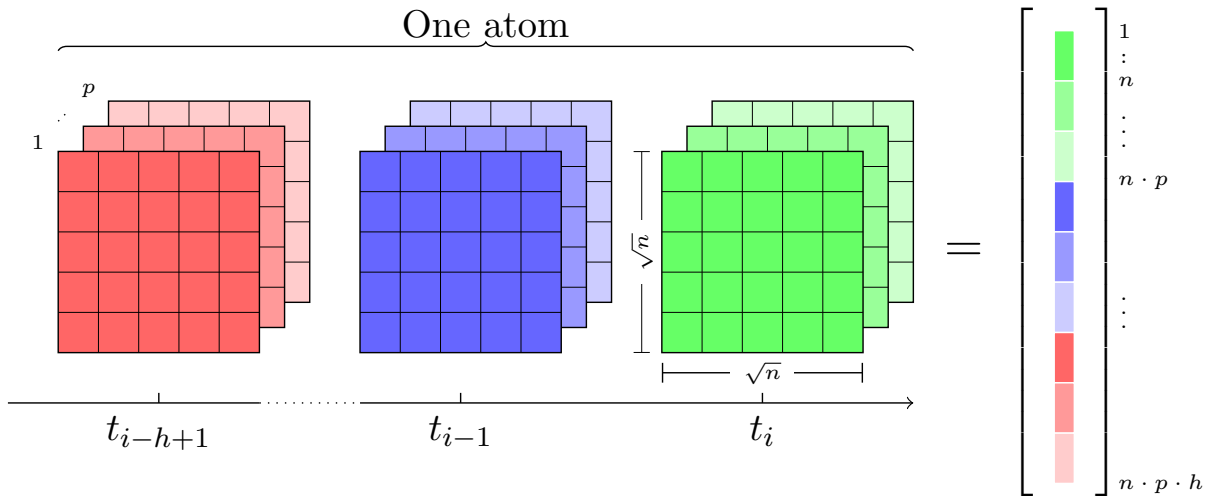
To identify the most similar atom  $\mathbf{d}_i^*$  for each image patch and its corresponding ideal label atom, an approximate nearest neighbor algorithm is used (FLANN [62]). The use of a single nearest neighbor to describe an image patch corresponds to a sparsity factor of  $L = 1$  in Equation (5.2).

The second part of each iteration is updating the dictionary, where each dictionary atom  $\mathbf{d}_i$  is moved along the direction determined by the similar and dissimilar groups' centers,  $\mathbf{d}_{P,i}$  and  $\mathbf{d}_{N,i}$  respectively:

$$\mathbf{d}_i = \mathbf{d}_i + \tau (\mathbf{d}_{P,i} - \mathbf{d}_{N,i}) \quad (5.6)$$

where the step size is chosen as  $\tau = 0.2$ . In each iteration, these steps increase the discriminative power, i.e., the label atoms move closer to the ideal label atoms. The number of iterations for building the dictionary is recommended to be between 20 and 40 [15]. A larger number of iterations provides a more discriminative dictionary, but increases the time spent in building the dictionary. Unless otherwise stated, 20 iterations are used to produce the results in this section.

The atom size determines the neighborhood included in the analysis. The atoms can include a spatial, spectral and/or temporal neighborhood of various sizes. The choice of neighborhood to include should be considered in relation to the data at hand. For textures it was found in [15] that a  $3 \times 3$  spatial neighborhood was superior to larger atom sizes, while it might be different when classifying clouds. As convective clouds develop rapidly over time it is desirable in this context to include a temporal neighborhood, i.e., data from a number of time lags – a form of history. In what form the spectral content should be included must also be considered. Should it be included in its original eight IR channels or rather one of the appropriate subspaces discussed previously? If so, how many components should be included? All of this will be considered and quantitatively evaluated, while also considering that larger dictionaries are computationally more expensive. Each atom is effectively a  $\sqrt{n} \times \sqrt{n} \times p \times h$  matrix, where  $n$  is the number of



**Figure 5.1:** A dictionary atom with a spatial neighborhood of size  $\sqrt{n} \times \sqrt{n}$ ,  $p$  spectral components and inclusion of  $h$  lags of history. The dictionary atom is unfolded to a vector of length  $n \cdot p \cdot h$ .

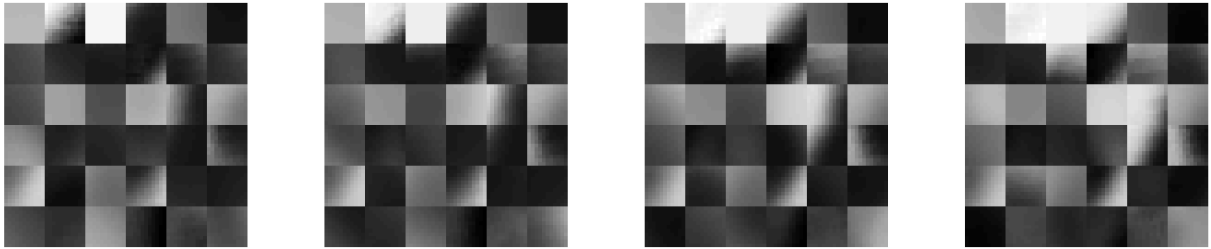
pixels in the spatial neighborhood,  $p$  the number of spectral components and  $h$  the number of lags (history) included. In the dictionary this is unfolded to a vector of length  $n \cdot p \cdot h$ . This is illustrated in Figure 5.1.

Unless otherwise stated, the spectral components used are principal components from the global PCA described in Section 3.2.2. A quantitative comparison of using the principal components or the single canonical variate will be performed.

Label information used for building the dictionary is extracted from the simple tracking described in 4.3.2. As this is a binary classification problem (rain or no rain), the segmented areas will be

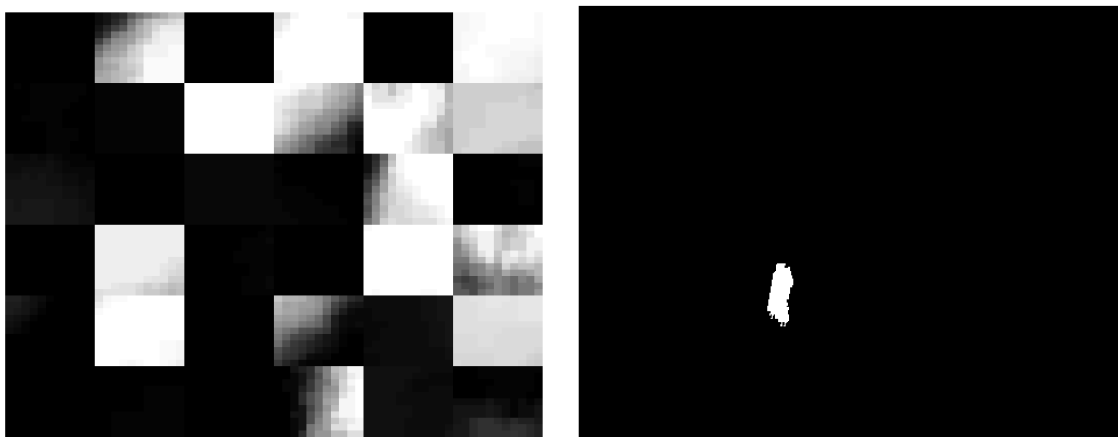
assigned the value 1 and others 0. Each tracked cloud has been segmented in a number of time lags, ideally from birth till well after time of maximum precipitation. In each of these lags, the truth will be this label information, i.e., the ideal result of a classification/segmentation method.

An example of a built dictionary can be seen in Figure 5.2. Parameters used are  $\sqrt{n} = 11$ ,  $p = 1$  and  $h = 4$ . Only the first 36 dictionary atoms are shown. By inspecting for instance the second atom in the first row, a rounded corner can be seen, which going back through time (left to right) shrinks in radius. This is a good example of a part of a rapidly developing convective cloud, where it starts very small and grows through time. The associated label dictionary can be seen



**Figure 5.2:** Example of 36 dictionary atoms with  $\sqrt{n} = 11$ ,  $p = 1$  and  $h = 4$ . Number of time lag is decreasing from left to right, i.e.,  $t_i$  is leftmost and  $t_{i-h+1}$  is rightmost.

in Figure 5.3, where it can be confirmed that the second dictionary atom is in fact a corner of interest, since pure white equals class one, i.e., a convective cloud, while black is zero.

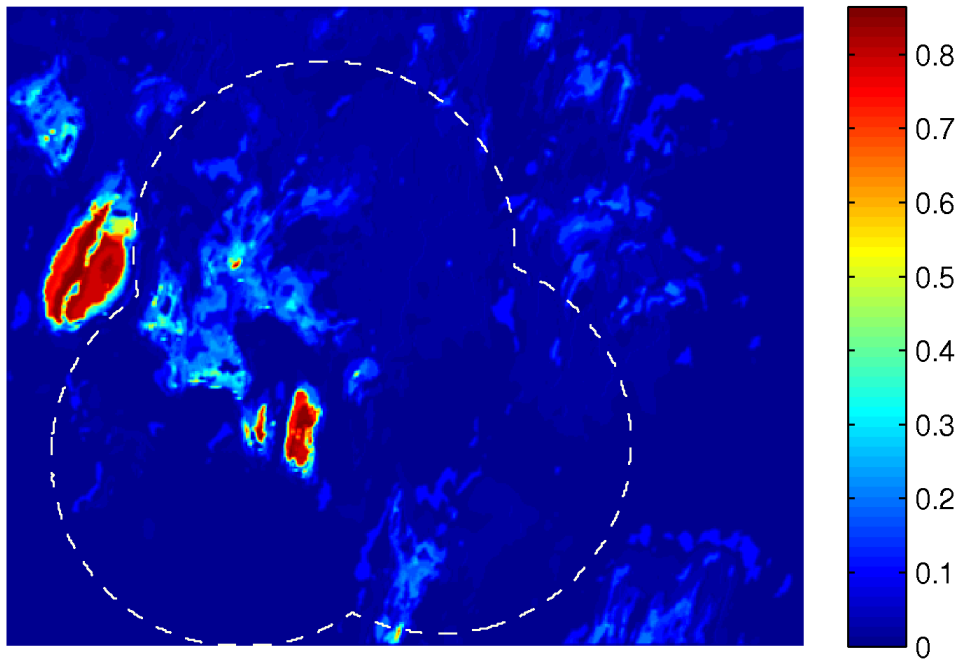


**Figure 5.3:** (left) The 36 label atoms associated to the dictionary in Figure 5.2. (right) The binary mask used as label information to build the dictionary.

## 5.2 Classification

Classification and segmentation of events is done with a dictionary specific for the purpose – in this case precipitation events. The atoms used includes the same neighborhood as for building the dictionary and probabilities are assigned using the method described in [15]. This method for assigning probabilities has similarities to parts of the “Fuzzy” Verification framework, described in [20].

In short, the image patches are collected for the entire test image and the best match for each is found in the dictionary using the approximated nearest neighbor search. The label information for this dictionary atom is transferred to the corresponding patch in the classified test image by addition. As each pixel in the test image is swept over  $n$  times, the probability is assigned as the pixel value in the classified image divided by  $n$ . E.g., if all  $n$  sweeps infers label probability 1 to a pixel, the probability field will have value 1 at this point. To move from a probability



**Figure 5.4:** 2007-08-20. Probability field for heavy precipitation. The white dashed line indicates the border of the radar data coverage. Parameters used are  $\sqrt{n} = 3$ ,  $p = 1$  PCs,  $h = 4$  lags.

field to an actual classification, some form of segmentation is necessary. The simplest form of segmentation is to assign the label with the highest probability, without using any additional information. This is the method used here.

Measures used to quantitatively assess the performance of the classification procedure are sensitivity and specificity. These measures are adopted from medical diagnostics, where it is used to evaluate the performance of a binary classification, e.g., sick or not sick. Here, a positive is translated to heavy precipitation and negative to less or none rain. Specificity is defined as

$$\text{specificity} = \frac{TN}{TN + FP} \quad (5.7)$$

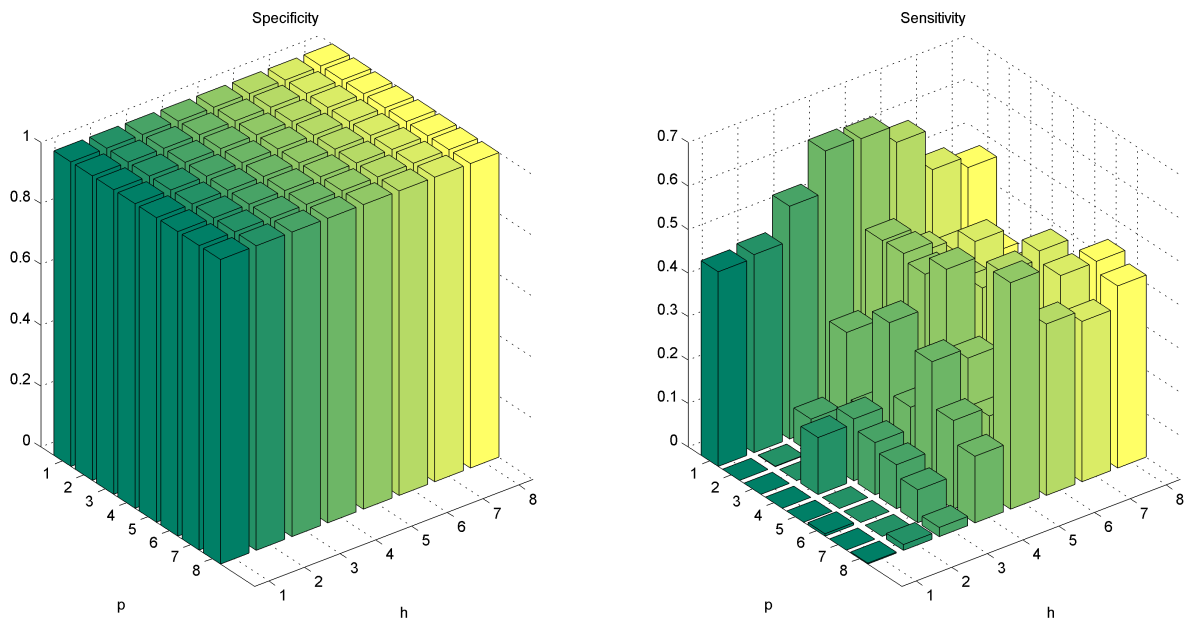
where  $TN$  is short for true negatives and  $FP$  for false positives. This measure quantifies the test's ability to correctly diagnose (classify) negative outcomes, meaning areas of no rain, while sensitivity measures the test's performance when classifying positive outcomes, i.e., heavy precipitation:

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (5.8)$$

where  $TP$  is true positives and  $FN$  false negatives. These measures are only calculated inside the coverage of the radar, which is illustrated with a white dashed line in Figure 5.4. This is necessary as the collection of the ground truth depends on availability of the radar data.

The point in time used for classification is the hotspot of scenario 2007-08-20, as this is a scenario with a rapidly developing isolated convective cloud. This scenario is used for experiments in this section, unless otherwise stated, and the goal of classification is the label information at time of the hotspot, which is the binary mask shown in Figure 5.3. When building a dictionary for classification, the scenario being the goal of the classification is not used in the dictionary learning. Only already known scenarios are used as training data, which in this case are chosen as 2007-07-16 and 2007-08-11. These are both heat thunder scenarios, which should result in a dictionary trained to classify this type of events.

To determine the number of spectral components  $p$  and time lag history  $h$  simultaneously, dictionaries are built for all combinations of  $p, h \in \{1, \dots, 8\}$  and used for classification of heavy precipitation. Sensitivities and specificities are calculated for each of the  $8^2 = 64$  combinations and shown as bars in Figure 5.5. It is seen that the specificity is very high ( $\approx 99.9\%$  on average) and does not vary much across the domain (standard deviation  $\approx 3.3e - 4$ ), which of course relates to the fact that the number of actual negative outcomes is much larger than the number of positives. Therefore, the method's performance under various circumstances will for now be evaluated in terms of sensitivity only. Based on this learning curve approach to determining



**Figure 5.5:** *Specificity and sensitivity for  $h, p \in \{1, \dots, 8\}$  and  $\sqrt{n} = 3$ .*

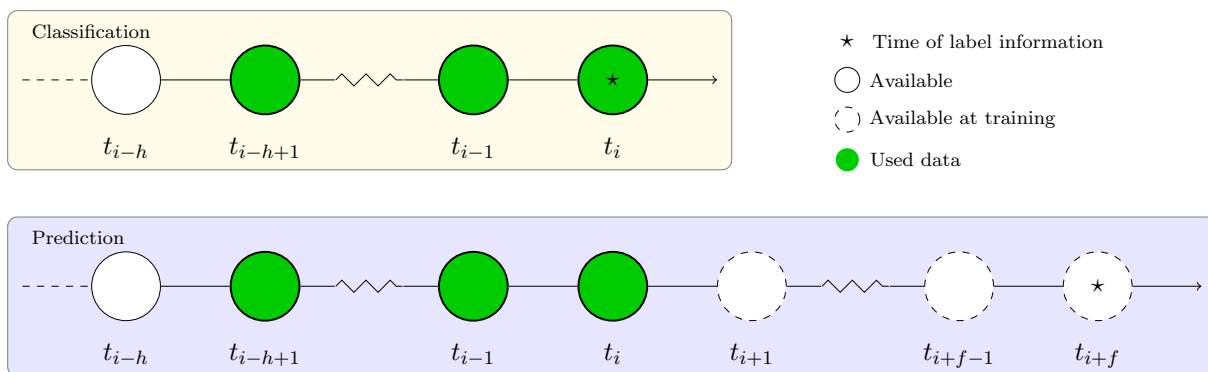
parameters, it is decided to use  $p = 1$  principal components and  $h = 4$  lags in each dictionary atom. This combination yields a specificity of 98.9% and a sensitivity of 63.0%. The probability field for this dictionary is the one shown in Figure 5.4.

Other parameters and their influence on the method's behavior will be discussed when using the method for prediction.

## 5.3 Prediction

To utilize the classification method for short-term prediction, only a small modification will have to be made, namely the label information used when building the dictionary. The label information states what is the classification goal. To decide what label information is to be used, the number of lags to predict ahead – the forecast length – must be chosen. As this is short-term prediction, the forecast length will be between 0 and 3 hours, which will be denoted in terms of the number of lags  $f \in \{0, \dots, 12\}$  of 15 minutes length. When producing a one hour forecast,  $f = 4$ , this implies that label information at time  $t_{i+f}$  is used for the data at time  $t_i$ . For a forecast length of  $f = 0$  classification and prediction are equivalent. By using future label information, a predictive capability is added to the method.

As the goal for the forecast is to predict severe convective events, label information from the time of the hotspot described in Section 2.1 will be used. Effectively, the only difference between classification and prediction for this method is the number of lags between the truth – the label information – and the data used. This extension is illustrated in Figure 5.6. The predictive

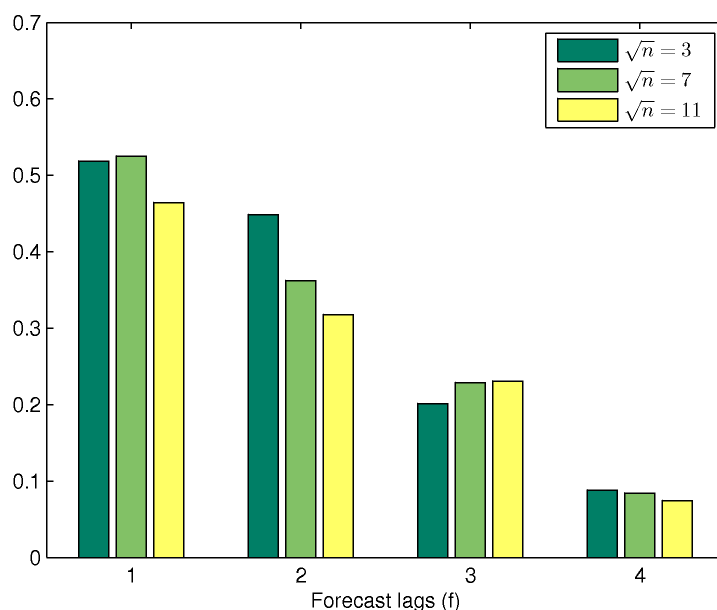


**Figure 5.6:** Illustration of used data for classification and prediction. For classification, the label information at time  $t_i$  is the goal, while the label information at time  $t_{i+f}$  is the goal of the prediction.

performance will be evaluated for different forecast lengths and parameter choices using the same quantitative measures as for classification. In the following, forecast lengths of  $f \in \{1, 2, 3, 4\}$  will be used in the quantitative assessments, when varying parameters.

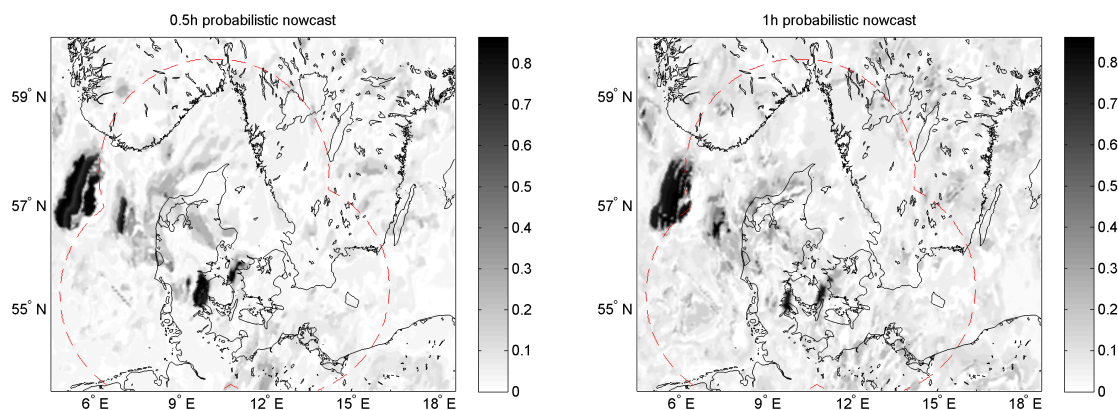
First, the spatial dimensions of the atom are varied. For each value, a dictionary is built and segmented as above. The ideal result is the same true image as before, i.e., the segmented image at the time of the 2007-08-20 hotspot shown in Figure 5.3. The sensitivities are shown in Figure 5.7. It is seen how the sensitivity drops dramatically when increasing the forecast length. Even for a forecast length of one hour, the sensitivity is below 10%. In comparison, the spatial extent of the atoms do not seem to influence as much as one could have expected. The best overall performance is achieved for the smallest atom size  $\sqrt{n} = 3$ , which is not as expected. It was expected that larger atoms would capture the neighborhood of precipitation better than the smaller ones, which does not seem to be the case. Interestingly, this indicates that it is the very near proximity – maybe even the pixel itself – that contains the information needed to classify a cloud as heavy precipitation, rather than the larger shape features.

Figures similar to the one above can be found in Appendix C.10, where other experiments have been conducted to choose the remaining parameters. These experiments – and the ones described



**Figure 5.7:** Comparison of sensitivities for varying forecast lengths and three different spatial extents  $\sqrt{n}$  of the dictionary atoms.

above – have led to the choice of  $\sqrt{n} = 3$ ,  $p = 1$  PC,  $h = 4$ ,  $N_{iter} = 20$  and  $w_{tol} = 0$ . For  $f \in \{2, 4\}$ , probability maps are shown in Figure 5.8, where coast lines have been overlaid.



**Figure 5.8:** Probability maps overlaid with coastlines. Forecast lengths are 0.5h and 1h respectively.

From these nowcast maps, it can be seen that the area that is going to develop into heavy rain is classified with a high probability. However, the future spatial development is not quite captured by the method, which is reflected in the sensitivities shown above and by visual inspection of the probability maps.

The experiments done in this section are only used for exemplification of the method and the parameters. The sensitivities obtained cannot be obtained this way in an operational setting, as they were calculated on the scenario which were used to tune the parameters, i.e., the parameters are not necessarily generalizable to other scenarios.



## 5.4 Obtaining a generalizable error

In this section the goal is to obtain a generalizable error for each scenario and collectively for the described dictionary method. This will allow for comparison with other methods and expose this method's ability to produce a generalizable model. Furthermore, the errors will be calculated for different forecast lengths, such that the method's predictive capabilities can be assessed as well.

The choice of parameters is done by varying the parameters over a range, whereby producing a learning curve. The parameters providing the best performance is chosen. However, the parameters providing the best performance for one part of the data is not necessarily appropriate for other parts of the data. Therefore, tuning the model must be performed systematically in order to choose parameters that generalize well [34].

### 5.4.1 Partition of data

In general when choosing parameters for a mathematical model, the data set should be divided into three sets: Training, validation and test. The training set is used to build the model, the validation set is used to tune the parameters and the final model is applied to the test set. In the following, three types of errors are associated to this partitioning of data:

- **Training error:** When applying a model to the same data as were used to build it, a training error is obtained. Here, the training error will furthermore imply that it is the lowest obtainable error for all parameter combinations. The training error is usually very low, as the model will usually overfit to the training data.
- **Validation error:** For each parameter combination available, the validation error is obtained by building the model on a training set and evaluating it on another part of the data – the validation set. As this is done successively for all parameter combinations, the lowest error in this case will represent the combination of parameter tuning the model to the validation set. This is similar to what was done in Section 5.2 and 5.3.
- **Test error:** When having tuned parameters by building the model successively for all parameter combination and applying to a validation set, the test error is obtained by evaluating this model on the test set. Hence, the test error is the one simulating an operational scenario, where the already available data have been used to set up a model and choose parameter and applied to new incoming data. The test error can be obtained in these simulations by the difference between predictions and true outcome.

Thus the test error is the error type of interest, when the goal is to achieve a generalizable error. For the dictionary method this implies that for the segmentation of one scenario, the parameters used to build the dictionary is obtained by dividing the remaining scenarios into a training set and a validation set. As the data set at hand is quite inhomogeneous in nature, the division of data needs to be considered carefully. A technique called leave-one-out cross validation (LOO-CV) will be used to train the model and obtain test errors for each scenario.

### 5.4.2 Cross validation

Cross validation is performed by successively leaving out one or more parts of the data and use it as validation set. In LOO-CV, one part is left out as validation set, while the remaining parts are used as training data. This is repeated until all parts of the data have been left out.

In this setting the data consist of  $N_s = 6$  scenarios. Each of these scenarios will be successively used as test data, while performing cross validation on the remaining five. In each iteration of the CV procedure, four scenarios are used to build the model and one is used to obtain a validation error. The model is built and evaluated for each parameter combination. The parameters and

|                                |                                |
|--------------------------------|--------------------------------|
| $\sqrt{n} \in \{3, 5, 7\}$     | Atom size                      |
| $p \in \{1, 4\}$               | Number of spectral components  |
| $h \in \{1, 2, 4\}$            | Number of time lags to include |
| $w_{tol} \in \{0, 0.5, 0.75\}$ | Atom similarity tolerance      |

**Table 5.1:** Parameters and their ranges chosen to vary when choosing models.

their ranges are shown in Table 5.1. Each of these parameter's function in the dictionary learning method are described in Section 5.1. The total number of possible parameter combinations are  $N_p = 3 \cdot 2 \cdot 3 \cdot 3 = 54$ . These parameters will be arranged in a matrix denoted  $\theta$  of size  $N_p \times 4$  where the  $k$ 'th combination (row) will be denoted  $\theta_k$ .

In Algorithm 4 this is written in pseudo-code.

---

**Algorithm 4** Choosing the best parameter for each scenario and obtaining a test error using LOO-CV

---

**Require:** Matrix of  $N_p$  parameter combinations  $\theta$

**Require:** Error function  $E$

$\mathcal{S} \leftarrow$  all  $N_s$  scenarios

**for all**  $i \in \{1, 2, \dots, N_s\}$  **do**

$s \leftarrow \mathcal{S}(i)$

$\mathcal{C} \leftarrow \mathcal{S} \setminus s$  {Cross validation set}

**for all**  $j \in \{1, \dots, N_s - 1\}$  **do**

$\mathcal{V} \leftarrow \mathcal{C}(j)$  {Validation scenario}

$\mathcal{T} \leftarrow \mathcal{C} \setminus \mathcal{V}$  {Training scenarios}

**for all**  $\theta_k, k \in [1, N_p]$  in  $\theta$  **do**

Train method using  $\mathcal{T}$  and  $\theta_k$

Predict  $\mathcal{V}$

Obtain training error  $\epsilon_{ijk} = E(\mathcal{V}_{\text{predicted}}, \mathcal{V}_{\text{true}})$

**end for**

**end for**

Select best parameter combination  $\theta_i^*$  as:  $\arg \min_k \frac{1}{N_s - 1} \sum_{j=1}^{N_s - 1} \epsilon_{ijk}$

Train method using  $\mathcal{C}$  and parameters  $\theta_i^*$

Predict  $s$

Obtain scenario test error  $\epsilon_i = E(s_{\text{predicted}}, s_{\text{true}})$

**end for**

---

Since it cannot be assumed that parameters are independent of each other, a validation error is obtained for all  $N_p$  combinations. The number of dictionaries built and images segmented

therefore take on the number of  $N_{\text{total}} = N_s \cdot (N_s - 1) \cdot N_p = 1620$  for each forecast length. As this number increase dramatically when including more parameter combination. Only a subset of the values each parameter can actually attain is included in the cross validation and can be seen in Table 5.1.

In the following, LOO-CV was done for separately for forecasts of length of  $f \in \{0, 2, 4\}$ , i.e., classification, 30 minutes and 1 hour. This might seem too short a time, but recalling the tracking of clouds in Figure 4.12, some of the clouds exhibited lifetimes as short as one hour. The forecast length is not included as a variable parameter, as it is a choice the forecaster must make. However, it is still interesting to see whether the choice of parameters depend on the forecast length, which is why multiple analyses are performed.

### Optimal parameter choice

As the number of no-rain pixels in the image greatly outweighs the number of rain pixels, the criterion for choosing the optimal parameters from the cross validation must be considered carefully. For instance, in an image of size  $400 \times 500$  where the true number of rain pixels is 1000, classifying all pixels as no-rain would yield a classification error of only 0.5%, but misclassifying all rain pixels. On the other hand, using maximum sensitivity alone as the criterion would yield the opposite problem, where classifying all pixels in the image as rain would equal a sensitivity of 100%, but misclassifying the majority of no-rain pixels. Inspired by the geometric mean [42, 47], a balanced measure in form of the product of specificity and sensitivity was chosen. This will strive towards a balance in the classification rate between the two classes.

For the  $i$ 'th scenario, the error for parameter combination  $k$  when using the  $j$ 'th validation set is thus defined as

$$\varepsilon_{ijk}^f = 1 - SE_{ijk}^f \cdot SP_{ijk}^f \quad (5.9)$$

for a forecast length of  $f$ . Here  $SE$  and  $SP$  represents sensitivity and specificity as defined in Equations (5.7) and (5.8). Note that the product is subtracted from 1 only in order to be able to refer to this measure as an ‘‘error’’. When results are being presented further down, it will be either  $SE \cdot SP$  or sensitivity and specificity separately.

The optimal parameter combination's index  $\tilde{k}_i^f$  for the  $i$ 'th scenario is found as the one minimizing the mean error of all  $N_s - 1$  validation errors:

$$\tilde{k}_i^f = \arg \min_{k \in [1, N_p]} \left[ \frac{1}{N_s - 1} \sum_{j=1}^{N_s-1} \varepsilon_{ijk}^f \right] \quad i \in [1, N_s]. \quad (5.10)$$

This definition enables to automatically choose the parameter combination for the final model – dictionary – for the  $i$ 'th scenario, given a forecast length  $f$ . By choosing the parameters that provide the minimum mean error over different validation sets, generalizability is enforced in the model. By maximizing the product of sensitivity and specificity, the data's inherent property of unbalanced classes is considered.

### 5.4.3 Dictionary method test errors for all scenarios

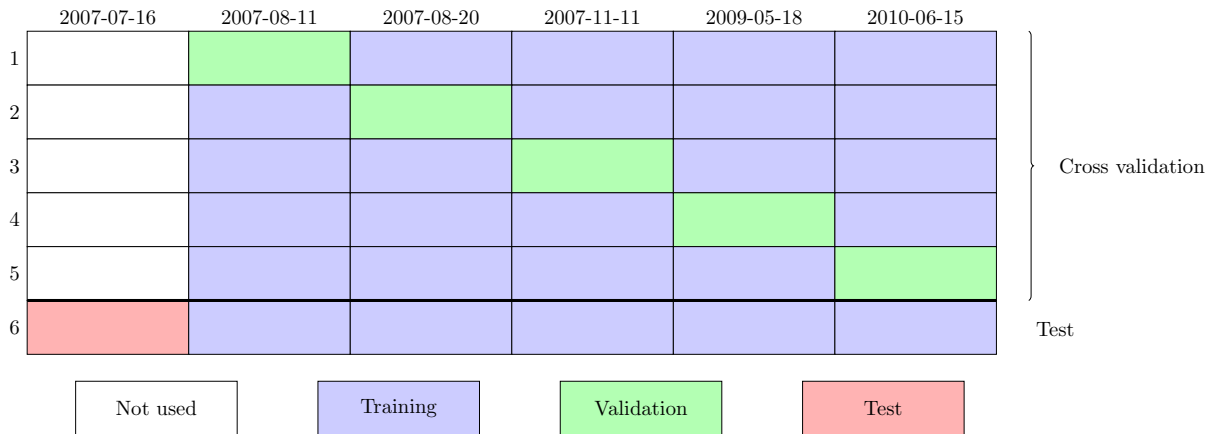
The described method of cross validation was performed for each scenario, where  $N_s = 6$ . This means that it is assumed that all scenarios are similar enough to use in a collective model, as

## CHAPTER 5. NOWCASTING USING LEARNED DICTIONARIES

opposed to the initial experiments in Sections 5.2 and 5.3, where only the heat thunder scenarios were used to build the model for a heat thunder scenario.

The subspace used in the results below is the global PCA subspace described in Section 3.2.2. When performing the cross validation, it was chosen to use only 10 iterations for building the dictionary, while the obtained test errors are from building a dictionary with 20 iterations.

For a single scenario's LOO-CV, the division into test, validation and training are illustrated in Figure 5.9. For each of the rows, the dictionary needs to be built and evaluated on the validation scenario as many times as there are parameter values  $N_p$ .



**Figure 5.9:** Illustration of division into training, validation and test set for a single scenario.  $5 \cdot N_p$  models are build during cross validation and one for test.

The results from the cross validation procedure is presented in Table 5.2, where the set of parameters chosen for each scenario is shown together with the obtained performance indicators. The goal is to get as high a sensitivity and specificity as possible. Some variation exist in the performance when using the dictionary method, as a number of random patches are used to initialize the dictionary. Even though the number of selected patches and all other parameters are constant, the actual patches chosen can influence the contents of the dictionary and thereby the final performance of the method. To account for this, each reported sensitivity and specificity is actually a mean of ten repetitions. The variations are visualized with box plots in Appendix C.12, where it can be seen that for most scenarios the actual variation is very small.

| Scenario   | $f = 0$    |     |     | $f = 2$   |      |      | $f = 4$    |     |     |           |      |      |
|------------|------------|-----|-----|-----------|------|------|------------|-----|-----|-----------|------|------|
|            | $\sqrt{n}$ | $h$ | $p$ | $w_{tol}$ | SE   | SP   | $\sqrt{n}$ | $h$ | $p$ | $w_{tol}$ | SE   | SP   |
| 2007-07-16 | 3          | 1   | 1   | 0         | 1    | 0.8  | 3          | 1   | 1   | 0.5       | 0.93 | 0.73 |
| 2007-08-11 | 3          | 1   | 4   | 0         | 0.58 | 0.99 | 3          | 1   | 1   | 0.75      | 0.5  | 0.96 |
| 2007-08-20 | 3          | 1   | 1   | 0         | 0.91 | 0.95 | 3          | 4   | 4   | 0         | 0.43 | 0.98 |
| 2007-11-11 | 3          | 1   | 1   | 0.5       | 0.98 | 0.84 | 3          | 1   | 1   | 0.5       | 0.59 | 0.76 |
| 2009-05-18 | 3          | 1   | 1   | 0.75      | 0.47 | 0.99 | 3          | 1   | 1   | 0.75      | 0.34 | 0.98 |
| 2010-06-15 | 3          | 1   | 1   | 0.75      | 0.47 | 1    | 3          | 1   | 1   | 0         | 0.54 | 0.97 |
| Average    |            |     |     |           | 0.74 | 0.93 |            |     |     |           | 0.56 | 0.9  |
|            |            |     |     |           |      |      |            |     |     |           | 0.27 | 0.85 |

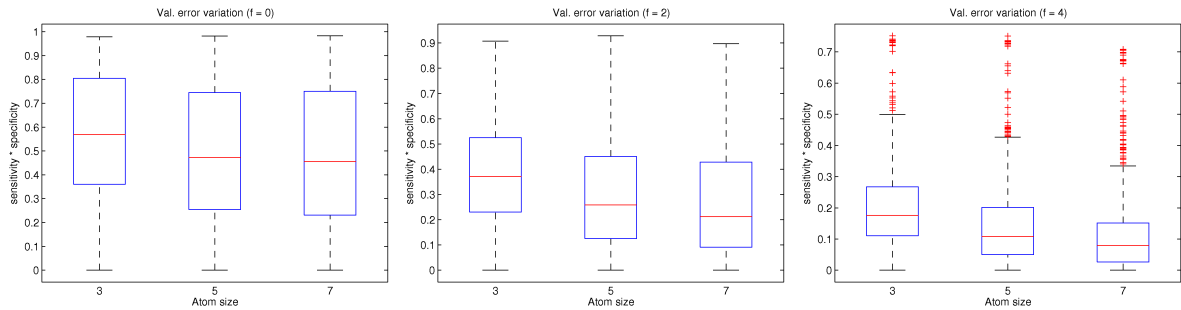
**Table 5.2:** Parameters chosen using cross validation and obtained sensitivities (SE) and specificities (SP) for each scenario. Forecast lengths are  $f \in \{0, 2, 4\}$ .

Inspecting the results in Table 5.2, it is noted that the sensitivity drops from an average over all scenarios of 74% for  $f = 0$  (classification) to 27% for  $f = 4$  (one hour forecast). Furthermore it is noticed that the specificity also reduces with increasing forecast length from 93% to 85% on average. Having this high specificities compared to the sensitivities, can be interpreted as the method rather undershoots the amount of rain pixels, than give too many false alerts.

### Parameter variations

Also interesting is to consider the variation of each parameter value over all scenarios. It is immediately seen from Table 5.2 that the most commonly picked parameters concerning the dictionary atom composition are  $\sqrt{n} = 3, h = 1, p = 1$ . This indicates that the models including the fewest dimensions are favorable. The choice of similarity tolerance  $w_{tol}$  is less obvious. This choice of parameters actually means that the best segmentation is almost solely based on the pixel intensity in the first PC and that the spectral and temporal structure is less important. Apparently the smallest neighborhood captures the textural information needed to recognize parts of a convective cloud.

For a parameter with  $n_p$  possible values, there exist  $N_s \cdot (N_s - 1) \cdot \frac{N_p}{n_p}$  validation errors for each of this parameter's attainable values. For instance, there are calculated  $6 \cdot 5 \cdot 18$  validation errors where each possible atom size  $\sqrt{n} \in \{3, 5, 7\}$  is used. From this set of validation errors, some simple statistics can be calculated. In Figure 5.10, box plots of sensitivities and specificities for the three possible values of  $\sqrt{n}$  are used to get an overview of the overall performance for each of these values.



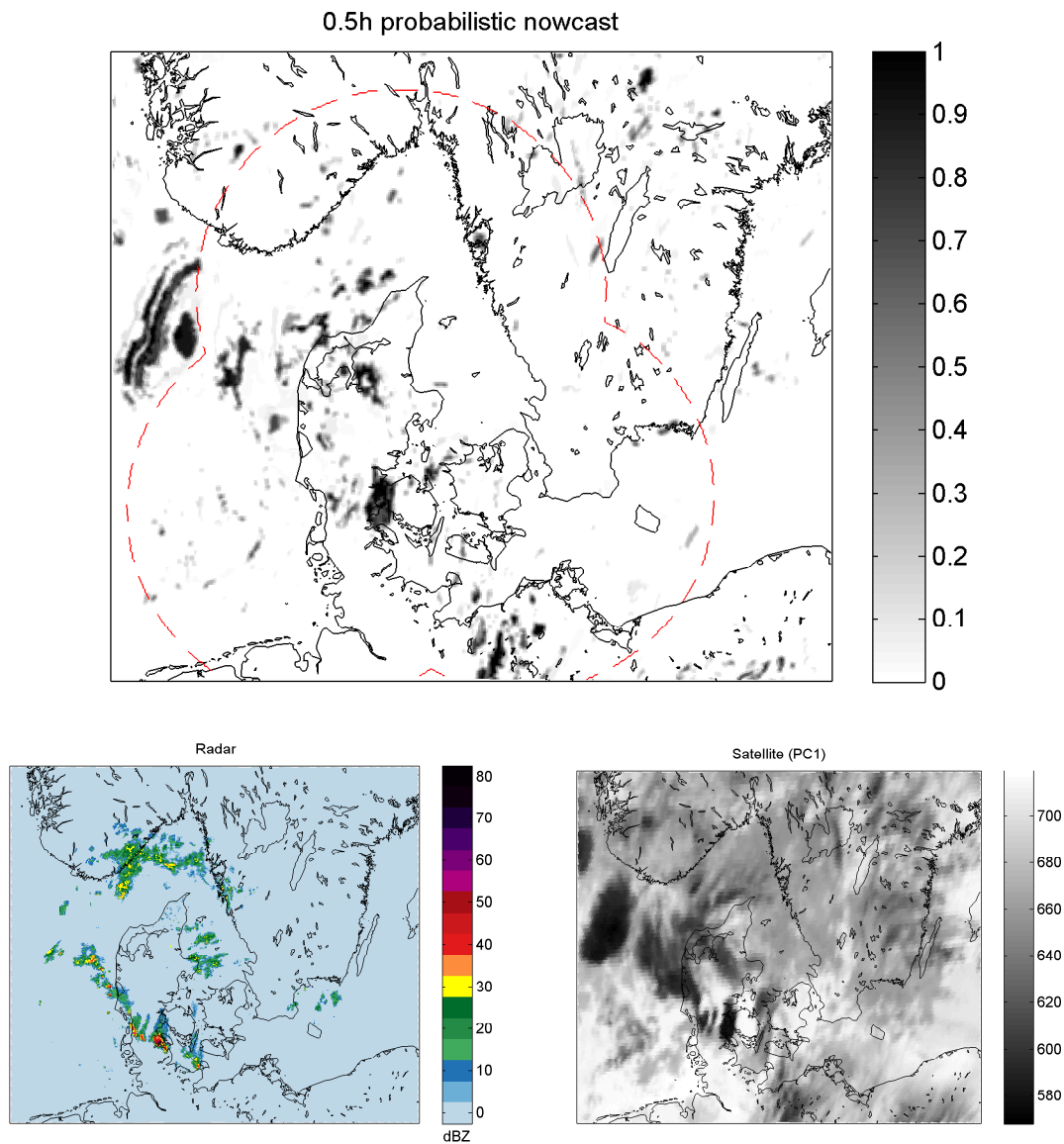
**Figure 5.10:** Box plots of variations in validation errors for each value of the atom size  $\sqrt{n}$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3 \cdot 2 \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.

These box plots confirm that the smallest atom size performs better on average for all validation sets, which was also observed from the results in Table 5.2. Similar box plots are shown for  $h$ ,  $p$  and  $w_{tol}$  in Appendix C.13. These also confirm that  $h = 1$  is the overall best choice, while for  $p$  and  $w_{tol}$  the performance patterns seem less obvious. The variation in number of atoms forming the dictionary can be seen for each value of  $w_{tol}$  in Appendix C.11. It is seen that the average dictionary size is in the area of 1000–2500 atoms and a clear dependence on the similarity tolerance is seen, which is as expected.

Experiments using the global CCA described in Section 4.3.1 were also conducted for  $f = 0$  and is shown in Appendix C.14. The performance for classification using this subspace is much lower than using the global PCA, wherefore no further experiments are conducted.

## Operational presentation of results

To evaluate the potential use of these results in an operational setting, they are presented alongside the radar and satellite data available at the time of the produced nowcast. In Figure 5.11, the 0.5 hour nowcast for August 20th, 2007, is shown as a probability map overlaid with coastlines together with radar and satellite data from 0.5 hour before the scenario hotspot. The satellite data are shown as the first PC of the global PCA described in Section 3.2.2. It is seen



**Figure 5.11:** 2007-08-20. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

that the nowcast does indeed capture some of the heavy precipitating cloud, but it can also be seen that most of this information could probably be retrieved by simultaneous visual inspection of the satellite and radar data. Similar visualizations of results can be seen in Appendix C.16 for the remaining scenarios and for nowcasts of length 0, 0.5 and 1 hours. Considering these visualizations, it is seen that when increasing the nowcast length above 0.5 hours, the method's

## CHAPTER 5. NOWCASTING USING LEARNED DICTIONARIES

ability to correctly forecast events degrades below what is useful in an operational scenario. This confirms what is seen from the sensitivities and specificities in Table 5.2.

However, it can also be seen that the classification maps (0 hour nowcast) do in fact capture areas of heavy precipitation. Even though the value of being able to identify rain as it occurs is hard to appreciate in an operational setting, it is an important result that the spectral channels of the satellite can be used to give a probabilistic map of heavy rain. While the radar data are used when building the model, they are not used when evaluating the model, wherefore the model can provide a larger coverage and thereby identify heavy rain clusters outside the range of the radars.

Having produced the nowcast maps and evaluated their potential use in an operational setting leads to a discussion and possible future improvements in the following chapter, before final conclusions are drawn.



## 5.5 Comparison with logistic regression

To compare the predictive capabilities of the discriminative learned dictionaries, a more traditional approach to prediction is implemented and tested on the same data. This will provide a frame of reference, such that the method's capabilities compared to other methods can be assessed. Logistic regression is chosen as an example of regression models [19, 7].

Logistic regression is a generalized linear model with a binary outcome, i.e., “success” or “failure”. This is appropriate here, where heavy rain has the value 1 and 0 otherwise. This regression model is suitable for prediction, as it fits a number of predictor variables to a logistic curve, which maps to an output between 0 and 1.

The predictor variables used here, are the same as for the dictionary learning method, illustrated in Figure 5.1, except no spatial neighborhood is used, wherefore  $n = 1$ . This could be incorporated into the logistic regression model by spatially shifting the images appropriately, but it was chosen to use a simple form of the method. Besides, the amount of data rapidly increase as all pixels are used in the logistic regression. Thus only  $N = p \cdot h$  predictor variables are used.

The output values used when training the regression model are the label information, such as the one illustrated in Figure 5.3. These labels are arranged in a vector  $\mathbf{y} \in \{0, 1\}^{N_p}$ , where  $N_p$  are the number of training pixels. Denoting a single element of this vector as  $y$ , the logistic regression model for a single pixel can be written as

$$y = \frac{e^{\mathbf{x}^T \boldsymbol{\beta}}}{1 + e^{\mathbf{x}^T \boldsymbol{\beta}}} \quad (5.11)$$

where  $\mathbf{x}_i^T \boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \dots + \beta_N x_N$ . Here  $\mathbf{x} = [1, x_1, x_2, \dots, x_N]^T$  is a vector of predictor variables and  $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_N]^T$  is the corresponding vector of regression coefficients.

The regression model is fitted using the MATLAB function `glmfit`, which is an implementation of the Iterative Reweighted Least Squares (IRLS) method [7].

### 5.5.1 Test errors

The test errors are obtained from cross validation using the procedure outlined in Algorithm 4. The parameters being varied are the number of lags  $h \in \{1, 2, 4\}$  and the number of spectral components from the global PCA  $P \in \{1, 2, 3, 4\}$  described in Section 3.2.2. It is seen that the chosen numbers of lags and spectral components to include are much larger than for the dictionary method, but since there is no spatial neighborhood included here, this cannot be directly compared. The test errors in form of sensitivity and specificity are shown in Table 5.3.

Box plots of validation errors for model built involving each parameter value during the cross validation procedure are shown in Appendix C.15.1. A large variability exist in the validation errors for  $f > 0$  the pattern of increasing performance with increasing number of lags/components get blurred as the errors increase rapidly. Comparing with the results obtained using the dictionary method in Table 5.2, it is seen that logistic regression performs better for the second scenario (2007-08-11), while for the other scenarios, the dictionary method outperforms the regression method. Especially when moving to prediction, the dictionary method maintains a much higher sensitivity than logistic regression. This could indicate that the structure captured in the dictionary's spatial contents is important for generalizing the model to other scenarios and that

| Scenario   | $f = 0$ |     |      |      | $f = 2$ |     |        |      | $f = 4$ |     |       |      |
|------------|---------|-----|------|------|---------|-----|--------|------|---------|-----|-------|------|
|            | $h$     | $P$ | SE   | SP   | $h$     | $P$ | SE     | SP   | $h$     | $P$ | SE    | SP   |
| 2007-07-16 | 1       | 4   | 0.99 | 0.97 | 4       | 4   | 0.47   | 1    | 4       | 4   | 0.034 | 1    |
| 2007-08-11 | 4       | 4   | 0.96 | 0.99 | 4       | 4   | 0.82   | 1    | 4       | 4   | 0.58  | 1    |
| 2007-08-20 | 4       | 4   | 0.09 | 1    | 2       | 4   | 0.022  | 1    | 2       | 4   | 0     | 1    |
| 2007-11-11 | 2       | 4   | 1    | 0.76 | 4       | 4   | 0.18   | 0.95 | 2       | 4   | 0     | 0.99 |
| 2009-05-18 | 4       | 4   | 0.11 | 1    | 1       | 3   | 0.0072 | 1    | 1       | 4   | 0     | 1    |
| 2010-06-15 | 4       | 4   | 0.51 | 1    | 4       | 3   | 0.01   | 1    | 2       | 4   | 0     | 1    |
| Average    |         |     | 0.61 | 0.95 |         |     | 0.25   | 0.99 |         |     | 0.1   | 1    |

**Table 5.3:** Logistic regression sensitivities (SE), specificities (SP) and chosen parameters for  $f \in \{0, 2, 4\}$  by use of cross validation.

the principle of representing a high dimensional space using discriminative atoms in this case is reasonable. On average, logistic regression yield a lower sensitivity than the dictionary method for all forecast lengths. The higher specificity indicates that the model built with logistic regression favors to report false negatives rather than false positives, i.e., it is optimistic in the sense that it heavily undershoots the severity of the downpour. This is of course not desirable in a nowcast system directly aimed at forecasting exactly these events.

Performance wise, the proposed dictionary method provides acceptable results when used for classification and the 0.5 hour nowcast also contains valuable information on approaching heavy precipitative clouds, but when increasing the nowcast length, the predictive power is not acceptable. Furthermore, it is interesting that the smallest dictionary atoms are chosen as the best descriptors by the cross validation, which indicates that the temporal structure is either not capture or not of importance for recognizing development of thunderstorms. In the following, a discussion on the method's shortcomings and possible improvements is conducted.

## Discussion & future work

Through the entire thesis, the six scenarios have been treated in a joint analysis. It has been considered that they belong to two different categories: cold air thunder and heat thunder. But it has not been possible to determine whether these two categories provide a proper division of the data set. Perhaps the convective systems should rather be divided into length of development, spatial extent or other meteorological characteristics. For this to be possible, a larger number of scenarios would be needed and each scenario would have to be assigned more features by a meteorologist.

Due to three out of six scenarios having hotspots in the evening or at night, it was decided to perform the analyses merely on the eight infrared channels. It is possible that the three visible channels contain valuable contributions to detecting convective initiation, but this would also require a larger number of scenarios.

All the decomposition methods used in the explorative analysis were linear in the variables. Is it certain that none of the signals would be more accurately described used non-linear variants of the methods? Absolutely not, but as this data have not previously been subjected to an explorative analysis, it is always a good idea to start with the simplest methods and extend in complexity from there. With the linear variants, the interpretation of the analysis results is relatively simple and the computation time low, where some of the non-linear decompositions are harder to interpret and most of them have higher computation time. Furthermore, the primary goal of this thesis is not an explorative analysis – the methods were used to get an initial overview of the data, and it turned out that the resulting subspaces were useful in the subsequent tasks, but whether non-linear methods can provide better performance in identification and tracking of convective initiation is an open question.

A point to discuss is the choice of using the minimum Euclidean distance as the only tracking criterion in Section 4.3.2. For the purpose in this thesis, it was found adequate to solve the task and it should not be interpreted as a suggestion for a tracking procedure generalizable to other tracking problems. The tracking of each convective cloud was visually verified and it was found to correctly identify the cloud between images, such that ground truth could be extracted. The procedure's simplicity also serves another purpose, namely to illustrate the value of determining a descriptive subspace. A less descriptive subspace would have increased the difficulty of segmentation, which in turn would have increased complexity of the needed tracking procedure.

Another cause for the minimum Euclidean distance to work as criterion is the fact that the convective clouds do not appear as moving very much between frames. They rather seem to

expand around their centroid, when they begin to develop into mature cumulus. Even so, the cloud will be affected and moved by the wind. Estimating a flow from the time series of images and subtracting it from the image in which the cloud is going to be identified could probably aid the tracking procedure, whether it is in the simple form here, or any other tracking procedure applied. Estimating the wind and the collective movement of the objects in the images could probably also be used when building the dictionary in Section 5.1, such that the history of data built-in to each atom actually contains the same part of the cloud, and not what is going to be the cloud in the next time step. For instance, the edge of a cloud might be captured in the first time lag of the atom. The same cloud in the prior time lag will for one thing be smaller, as it has expanded, but it might also have moved due to the present wind conditions. Removing this effect by subtracting an estimated flow field could possibly ensure that consistent atoms are built.

The most important thing to discuss is the choice of using the learned dictionary method, which is originally a classification method, as a method for forecasting. There are a few reasons for this choice: First of all, it is an interesting experiment to evaluate the method's ability to fill in the gap between known data and label information. Learned dictionaries have previously been used for other tasks in image analysis and now a member of this method family has been applied to another type of problem. Second of all, the hypothesis was that since the spatial structure – the texture information – unique for the convective clouds, can be captured by the dictionary atoms, the inclusion of a temporal dimension and effectively making every dictionary atom a time series, could possibly capture the temporal structure as well. The temporal structure for a developing convective cloud would in this setting take the shape of a cone, i.e., starting off as small area or a point and expanding around its centroid in each time step. The classification procedure attempts to recognize when such a cone develop, while the prediction should ideally recognize and extrapolate the cone's development into what is later going to be the area of heavy precipitation. However, while the dictionary method does indeed perform better than logistic regression, the degradation of the results when moving from classification to prediction is still quite severe. A possible explanation for this, is, that while the texture of a convective cloud is indeed possible to capture in a dictionary atom covering three-by-three pixels of four square kilometers each, the temporal development happens on a much larger scale. This explanation is based partly on the knowledge gained from testing the dictionary method as a nowcasting method, but can also be supported by the areal development depicted in Figure 4.11 where it can be seen that the clouds increase 2000–5000 square kilometers in extent within the last hour of the hotspot. Therefore, the dictionary atoms will not be able to fill this gap, as one atom cannot cover the same cloud edge in two consecutive images, as they are too far apart spatially. However, if the nature of a developing cloud was that it emerged from some sort of latent properties present in the spectral components in the satellite imagery, the dictionary method should still be able to capture this potential. But it seems that the available information, the clouds must be considered as occurring spontaneously and then expanding. This means, that when a cloud has started to develop in one time step, covering only a small area, the proximity of the cloud does not contain any spectral information that suggests that it will fifteen minutes later also be part of a maturing cumulus. Therefore, it seems that the growth of a heavy precipitative cloud is an expansion rather than a potential being released in its proximity.

## 6.1 Future work

In general, the choice of methods throughout the thesis is only one path among many. The entire process, from raw data and explorative analysis to collection of ground truth and finally

classification and prediction, can be seen as a composed of building blocks. Each of these blocks can most likely be improved, some more than others. The explorative analysis alone could be expanded significantly and important insights could possibly be gained. The ground truth collection might be improved in precision, speed or maybe even replaced by a meteorologist's manual labeling. Several classification methods could be applied to the data, which may or may not improve the classification. The prediction – the nowcasting – definitely needs improvement, whether it is by a better choice of method that can both classify and predict, or it should be a separate post processing of already segmented images.

One of the improvements that could readily be implemented with the available data and methods, is the addition of a flow field. The flow field should describe the motion of the clouds between images, whereby a more robust dictionary could be built, or make the tracking of clouds more robust. The problem of estimating flow in infrared satellite data is treated in [13], where existing methods for estimating optical flow is adapted to meteorological use.

A potentially valuable predictor variable to include would be atmospheric instability, which is estimated from Meteosat data in [46]. Estimation of the atmospheric instability can possibly give warnings of where convective clouds are going to develop, rather than waiting for them to be prominent in the data.

A method that could possibly be used for determining which clouds are heavy precipitation, and which are not, is Canonical Discriminant Analysis (CDA) [45]. CDA is one of the classic multivariate methods for determining a direction in space that discriminates between classes. In this thesis it could possibly have been used to detect clouds of interest, instead of the dictionary method. It was, however, chosen to use learned dictionaries in order to expand the palette of methods applied, but a valid step would indeed be to test CDA's ability to distinguish between the rain/no-rain classes.

Quite a few method families have come to mind while working with the project, that might be interesting to apply to the satellite data. This especially includes kernel methods for decomposition of signals [1, 82, 66] and unmixing methods [67, 5]. For other purposes, variations of Independent Component Analysis (ICA) have proved successful in non-orthogonal unmixing [38, 4] and would be interesting to apply to this data as well. There also exist numerous methods that would be theoretically interesting to apply to meteorological data, e.g., graph cut algorithms [9] and methods recently proven successful within microscopy cell deconvolution and tracking [6, 95]. These might lead to better identification and segmentation of convective clouds and eventually be a building block for the designing a nowcast system.

While a lot of methods could be interesting to apply, the knowledge gained from this thesis rather suggests that a more homogeneous data set should be collected in order to significantly improve the nowcasts. It is not the size as such that is important for the data set, it is rather that the scenarios should be more similar. For instance, is it hard to build a three hour nowcast model on data, where only three of the scenarios have any sign of the cloud of interest more than one hour before the time of maximum precipitation. And these three scenarios are by the meteorologist categorized in two different categories. As suggested in the discussion, a larger number of meteorological labels should be assigned to a larger number of events. Here, the rarity of these extreme events is of course an issue, but rather than characterizing a number of events as “extreme”, the preprocessing should be more systematic and similarities and dissimilarities carefully considered. If that means that a category only consists of one scenario, then it would still be valuable to use scenarios from other categories to create a model and see if it generalizes to this unique scenario. But in this case, the choice of scenarios to use in a joint analysis could

## CHAPTER 6. DISCUSSION & FUTURE WORK

be based on meteorological characteristics and insights on similarities between categories could be gained from analysis of the satellite data, rather than unexpected dissimilarities.

## Conclusion

Six dates exhibiting extreme precipitation have been analyzed in this thesis. Their extremities range from intense lightning activity, over heavy snow fall to more or less extreme cloudbursts. These six scenarios were beforehand categorized as being thunderstorms in either cold or warm air masses.

Time series of multispectral satellite imagery and weather radar data accompanying each scenario were processed, such that an analysis including both data types could be performed. A toolbox has been developed to ease future work with these data. This toolbox contains methods for reading the two data formats and metadata contained in the data files, re-projection of data and visualization.

An explorative analysis of the infrared channels of the satellite imagery was conducted. Several analysis methods were applied and a subspace capturing extremities in cloud top temperatures were determined from a principal component analysis. The analysis was performed at points in time, where heavy precipitation was known to be prevalent, thereby designing the subspace to be descriptive of such events.

A spatial misalignment was observed between precipitation recorded in the radar data and areas of minimum cloud top temperatures, which can be explained by the different viewpoints of the radar and the satellite. This was found to degrade the further analysis; hence a scale, rotation and translation invariant method was developed. Thereby, the best alignment of the intense rain cells in the radar data and the associated area in the satellite data was determined. Subsequently, canonical correlation analysis was used to determine a subspace emphasizing cloud tops causing heavy precipitation. In this subspace, ground truth could easily be collected in form of segmented clouds in the satellite imagery.

The collected ground truth was used to generate a binary label image for each scenario as the goal of classification and nowcasting. A method for learning a discriminative dictionary of image patches – atoms – was applied to the problem of identifying and predicting areas of extreme precipitation. The method was chosen based on the hypothesis that texture information, spectral composition and temporal structure are important descriptors of heavy precipitating clouds. This method was modified to account for unbalanced classes, as the areas of non-extreme precipitation are much larger than areas of extreme precipitation. The dictionary atom size was customized to include multiple spectral components and a temporal history, making each atom include a time series of information.

Generalizable performance indicators for each scenario were determined using leave-one-out

## CHAPTER 7. CONCLUSION

cross validation, whereby parameters influencing the dictionary composition were also chosen. This was done to simulate an operational setting, where each scenario's extreme event can be recognized by a model built on the remaining scenarios. The performance of the dictionary method was compared with the more traditional logistic regression and found to be superior in both classification and prediction. Parameters chosen by cross validation for the learning of dictionaries indicate that the textural information in the convective clouds is captured in a small area in a single principal component and with no temporal history, at least when generalizability to other scenarios is enforced.

While the classification of the extreme precipitative events in each of the scenarios have shown to be possible using a learned dictionary, the prediction of events shows severe degradation when increasing the nowcast length above 0.5 hours. Several possible explanations have been given for the method's shortcomings, of which the most probable is that the temporal development of the convective clouds happen on a larger scale than can be captured by the dictionary atoms.

The diversity of the weather phenomena included in the analysis was found to degrade the nowcast system. The most important future work for improvement of the nowcast model therefore includes collection of a more homogeneous data set. This is expected to significantly improve the predictive capabilities of the method.

In summary, the problem of nowcasting extreme precipitation in Denmark using satellite data has been treated in this thesis. Emphasis has been on using methods that are statistical in nature and designed specifically for the region, and evaluating those using objective quantitative measures.



---

## Bibliography

- [1] T. Abrahamsen. Kernel methods for de-noising with neuroimaging application. Master's thesis, Technical University of Denmark, 2009.
- [2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 54(11):4311–4322, 2006.
- [3] T. W. Anderson and T. W. Anderson. *An Introduction to Multivariate Statistical Analysis, 2nd Edition*. Wiley-Interscience, 2 edition, Sept. 1984.
- [4] J. Basak, A. Sudarshan, D. Trivedi, and M. S. Santhanam. Weather data mining using independent component analysis. *J. Mach. Learn. Res.*, 5:239–253, 2004.
- [5] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. F. Huntington. ICE: a statistical approach to identifying endmembers in hyperspectral images. *IEEE Trans. Geosci. Remote Sensing*, 42:2085, 2004.
- [6] R. Bise, Z. Yin, and T. Kanade. Reliable cell tracking by global data association. In *IEEE International Symposium on Biomedical Imaging*, 2011.
- [7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, Oct. 2007.
- [8] T. Bøvith. *Detection of Weather Radar Clutter*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2008.
- [9] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:1124–1137, September 2004.
- [10] A. Brandt. Så meget regn faldt der. <http://vejret-dyn.tv2.dk/artikel/id-7888833:s%C3%A5-meget-regn-faldt-der.html>. TV2 Vejret.
- [11] A. Brandt. Voldsomt skybrud i hovedstadsområdet. <http://vejret.tv2.dk/artikel/id-7883380:voldsomt-skybrud-i-hovedstadsomr%C3%A5det-118-2007.html>. TV2 Vejret.
- [12] A. Brandt. Voldsomt uvejr ramte sønderjylland. <http://vejret.tv2.dk/artikel/id-7987377:voldsomt-uvejr-ramte-s%C3%B8nderjylland-208-2007.html>. TV2 Vejret.

## BIBLIOGRAPHY

- [13] D. Béréziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Environmental Modelling & Software*, 15(6-7):513 – 519, 2000.
- [14] J. M. Carstensen. *Image Analysis, Vision and Computer Graphics*. Technical University of Denmark, 2nd edition, 2002.
- [15] A. Dahl and R. Larsen. Learning dictionaries of discriminative image patches. BMVC 2011 Submission #587, 2011.
- [16] J. D’Errico. fminsearchbnd: Bound constrained optimization using fminsearch. <http://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd>. Mathworks File Exchange.
- [17] *The DMI weather radars*, January 2009.
- [18] *Interface between radar sites and DMI central facilities*, January 2009.
- [19] A. J. Dobson. *An introduction to generalized linear models*. Chapman & Hall/CRC, Boca Raton, 2nd edition, 2002.
- [20] E. E. Ebert. Fuzzy verification of high-resolution gridded forecasts: A review and proposed framework. *Met. Apps*, 15(1):51–64, 2008.
- [21] M. Elad. *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010.
- [22] L. Eldén. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.
- [23] B. K. Ersbøll and K. Conradsen. *An Introduction to Statistics*. IMM DTU, 2007.
- [24] EUMETSAT. *Radiometric Calibration of MSG SEVIRI Level 1.5 Image Data in Equivalent Spectral Blackbody Radiance*, January 2007.
- [25] EUMETSAT. *Effective Radiance and Brightness Temperature Relation for Meteosat 8 and 9*, January 2008.
- [26] EUMETSAT. *MSG Level 1.5 Image Data Format Description*, February 2010.
- [27] G. Evenden. PROJ.4 - cartographic projections library. MIT License (Public Domain). <http://trac.osgeo.org/proj/>.
- [28] A. Fernández-Ferrero, J. Sáenz, G. Ibarra-Berastegi, and J. Fernández. Evaluation of statistical downscaling in short range precipitation forecasting. *Atmospheric Research*, 94(3):448 – 461, 2009.
- [29] D. J. Gagne, A. McGovern, and J. Brotzge. Classification of convective areas using decision trees. *Journal of Atmospheric and Oceanic Technology*, 26(7):1341–1353, 2009.
- [30] W. H. Hand. An object-oriented technique for nowcasting heavy showers and thunderstorms. *Meteorological Applications*, 3(1):31–41, 1996.
- [31] M. Hansen. Skybrud ved brøndby. <http://galleri.tv2.dk/Vejret/7883763/5/>. Digital image, TV2 Vejret Galleri.

- [32] N. Hansen. Jyderne tog skraldet. [http://www.dmi.dk/dmi/jyderne\\_tog\\_skraldet](http://www.dmi.dk/dmi/jyderne_tog_skraldet). DMI.
- [33] N. Hansen. København fik høvl af hagl. [http://www.dmi.dk/dmi/index/nyheder/nyheder-2/koebenhavn\\_fik\\_hoevl\\_af\\_hagl.htm](http://www.dmi.dk/dmi/index/nyheder/nyheder-2/koebenhavn_fik_hoevl_af_hagl.htm). DMI.
- [34] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.
- [35] C. C. Henken, M. J. Schmeits, H. Deneke, and R. A. Roebeling. Using MSG-SEVIRI cloud physical properties and weather radar observations for the detection of cb/tcu clouds. *Journal of Applied Meteorology and Climate*, February 2011. Preliminary accepted version.
- [36] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, Dec. 1936.
- [37] M. Hundahl. *Meteorologi og Oceanografi for Skibsofficerer*. Iver C. Weilback & Co. A/S, 1st edition, 2003.
- [38] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Adaptive and learning systems for signal processing, communications, and control. J. Wiley, 2001.
- [39] I. L. Jirak, W. R. Cotton, and R. L. McAnelly. Satellite and radar survey of mesoscale convective system development. *Monthly Weather Review*, 131(10):2428–2449, 2003.
- [40] D. B. Johnson, P. Flament, and R. L. Bernstein. High-resolution satellite imagery for mesoscale meteorological studies. *Bulletin of the American Meteorological Society*, 75(1):5–33, 1994.
- [41] I. T. Jolliffe. *Principal component analysis*. Springer New York, 2002.
- [42] J. F. Kenney. *Mathematics of statistics*. Van Nostrand, N.Y. :, 1939.
- [43] C. Kessinger, M. Donovan, R. Bankert, E. Williams, J. Hawkins, H. Cai, N. Rehak, D. Megenhardt, and M. Steiner. Convection diagnosis and nowcasting for oceanic aviation applications. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7088 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Aug. 2008.
- [44] C. M. Kishtawal, S. K. Deb, P. K. Pal, and P. C. Joshi. Estimation of atmospheric motion vectors from kalpana-1 imagers. *Journal of Applied Meteorology and Climatology*, 48(11):2410–2421, 2009.
- [45] W. Klecka. *Discriminant analysis*. Quantitative applications in the social sciences. Sage Publications, 1980.
- [46] M. Koenig and E. d. Coning. The MSG global instability indices product and its use as a nowcasting tool. *Weather and Forecasting*, 24:272–285, February 2009.
- [47] K. Lee. Classification of imbalanced data with transparent kernels. <http://eprints.ecs.soton.ac.uk/7937/>, 2002.
- [48] V. Levizzani, J. Schmetz, H. J. Lutz, J. Kerkmann, P. P. Alberoni, and M. Cervino. Precipitation estimations from geostationary orbit and prospects for meteosat second generation. *Meteorological Applications*, 8(1):23–41, 2001.

## BIBLIOGRAPHY

- [49] V. Levizzani and M. Setvák. Multispectral, high-resolution satellite observations of plumes on top of convective storms. *J. Atmos. Sci.*, 53(3):361–369, 1996.
- [50] C. Lo and A. Yeung. *Concepts and techniques of geographic information systems*. Prentice Hall series in geographic information science. Prentice Hall, 2nd edition, 2002.
- [51] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [52] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, volume 0, pages 1–8, Los Alamitos, CA, USA, June 2008. IEEE Computer Society.
- [53] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, 2 edition, Sept. 1999.
- [54] J. S. Marshall and W. M. K. Palmer. The distribution of raindrops with sizes. *Journal of Meteorology*, 5, 1948.
- [55] E. V. Mattos and L. A. Machado. Cloud-to-ground lightning and mesoscale convective systems. *Atmospheric Research*, 99(3-4):377 – 390, 2011.
- [56] J. R. Mecikalski and K. M. Bedka. Forecasting convective initiation by monitoring the evolution of moving cumulus in daytime goes imagery. *Monthly Weather Review*, 134(1):49–78, 2006.
- [57] J. R. Mecikalski, K. M. Bedka, S. J. Paech, and L. A. Litten. A statistical evaluation of GOES cloud-top properties for nowcasting convective initiation. *Monthly Weather Review*, 136(12):4899–4914, 2008.
- [58] J. R. Mecikalski, W. M. MacKenzie, M. König, and S. Muller. Cloud-top properties of growing cumulus prior to convective initiation as measured by meteosat second generation. part ii: Use of visible reflectance. *Journal of Applied Meteorology and Climatology*, 49(12):2544–2558, 2010.
- [59] J. R. Mecikalski, W. M. MacKenzie Jr., M. Koenig, and S. Muller. Cloud-top properties of growing cumulus prior to convective initiation as measured by meteosat second generation. part i: Infrared fields. *Journal of Applied Meteorology and Climatology*, 49(3):521 – 534, 2010.
- [60] M. Mølgaard. Hvilken uvejrsnat! *Vejret*, 112(3), August 2007.
- [61] T. Møller. Vej skyllet væk udenfor gråsten 1. <http://galleri.tv2.dk/Vejret/7988588/12/>. Digital image, TV2 Vejret Galleri.
- [62] M. Muja. FLANN, Fast Library for Approximate Nearest Neighbors, 2009. <http://mloss.org/software/view/143/>.
- [63] D. P. Mukherjee and S. T. Acton. Cloud tracking by scale space classification. *IEEE Transactions On Geoscience and Remote Sensing*, 40(2):405–415, 2002.
- [64] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

- [65] Conversation with meteorologist Birgitte K. Nielsen at DMI, April 2011.
- [66] A. Nielsen. Kernel maximum autocorrelation factor and minimum noise fraction transformations. *Image Processing, IEEE Transactions on*, 20(3):612–624, march 2011.
- [67] A. A. Nielsen. Partial unmixing in hyperspectral image data. In *Proceedings of the Fourth International Airborne Remote Sensing Conference and Exhibition*, pages 535–542, 1999. Presented at: The Fourth International Airborne Remote Sensing Conference and Exhibition/21st Canadian Symposium on Remote Sensing. : Ottawa, 1999.
- [68] A. A. Nielsen. Multiset canonical correlations analysis and multispectral, truly multi-temporal remote sensing data. *IEEE Transactions on Image Processing*, 11(3):293–305, mar 2002.
- [69] A. A. Nielsen. Least squares adjustment: Linear and nonlinear weighted regression analysis, oct 2009.
- [70] A. A. Nielsen, K. B. Hilger, O. B. Andersen, and P. Knudsen. A temporal extension to traditional empirical orthogonal function analysis. *World Scientific Series in Remote Sensing*, pages 164–170, 2002.
- [71] N. W. Nielsen. Et regnskyl af dimensioner: Skybruddet ved gråsten i sønderjylland den 20. august 2007. *Vejret*, 114(1), February 2008.
- [72] L. Nørgaard. Halgbyge den 18. maj. <http://galleri.tv2.dk/Vejret/22433342/12/>. Digital image, TV2 Vejret Galleri.
- [73] T. Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser, 1997.
- [74] D. Pedersen. Nattens vejr i bransbøl på Als. <http://galleri.tv2.dk/Vejret/7988588/19/>. Digital image, TV2 Vejret Galleri.
- [75] C. Pohl and J. L. V. Genderen. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, 1998.
- [76] R. W. Preisendorfer. *Principal Component Analysis in Meteorology and Oceanography*. Developments in Atmospheric Science. Elsevier, 1998.
- [77] S. Puca, D. Biron, L. De Leonibus, D. Melfi, P. Rosci, and F. Zauli. A neural network algorithm for the nowcasting of severe convective systems. In *Computational Intelligence for Measurement Systems and Applications, 2005. CIMSIA. 2005 IEEE International Conference on*, pages 81 – 84, July 2005.
- [78] M. B. Richman. Rotation of principal components. *International Journal of Climatology*, 6(3):293–335, 1986.
- [79] R. E. Rinehart. *Radar for meteorologists*. University of North Dakota, 2nd edition, 1991.
- [80] D. Rosengren. Haglbyge over amager. <http://galleri-dyn.tv2.dk/Vejret/31238814/9/>. Digital image, TV2 Vejret Galleri.
- [81] G. Sapiro. Learning dictionaries for image analysis and sensing. Video Lecture [http://videlectures.net/mlss09us\\_sapiro\\_ldias/](http://videlectures.net/mlss09us_sapiro_ldias/), June 2009.

## BIBLIOGRAPHY

- [82] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
- [83] J. Schmetz, P. Pili, S. Tjemkes, D. Just, J. Kerkmann, S. Rota, and A. Ratier. An introduction to meteosat second generation (MSG). *American Meteorological Society*, pages 977–992, July 2002.
- [84] R. A. Scofield, R. J. Kuligowski, and S. Qiu. Satellite, lightning, and model data for nowcasting heavy rainfall from mesoscale convective systems (MCS's). *86th AMS Annual Meeting*, 2006.
- [85] J. M. Sieglaff, L. M. Counce, W. F. Feltz, K. M. Bedka, M. J. Pavolonis, and A. K. Heidinger. Nowcasting convective storm initiation using satellite-based box-averaged cloud-top cooling and cloud-type trends. *Journal of Applied Meteorology and Climate*, 50:110 – 126, 2011.
- [86] K. T. Smith and G. L. Austin. Nowcasting precipitation – a proposal for a way forward. *Journal of Hydrology*, 239(1-4):34 – 45, 2000.
- [87] T. Vejret. Kraftige byger med sne rammer sjælland. <http://vejret-dyn.tv2.dk/artikel/id-9329165:kraftige-byger-med-sne-rammer-sj%C3%A6lland-1111-2007.html>.
- [88] J. S. Vestergaard. Subspace projections of climate related geodata. Bachelor's thesis, Technical University of Denmark, 2009.
- [89] D. A. Vila, L. A. T. Machado, H. Laurent, and I. Velasco. Forecast and Tracking the Evolution of Cloud Clusters (ForTraCC) Using Satellite Infrared Imagery: Methodology and Validation. *Weather and Forecasting*, 23:233–+, 2008.
- [90] H. Wackernagel. *Multivariate Geostatistics*. Springer, 1995.
- [91] P. Wessel and W. Smith. A global, self-consistent, hierarchical, high-resolution shoreline database. *J. Geophys. Res.*, 101(B4):8741–8743, 1996.
- [92] J. W. Wilson, Y. Feng, M. Chen, and R. D. Roberts. Nowcasting challenges during the beijing olympics: Successes, failures, and implications for future nowcasting systems. *Weather and Forecasting*, 25:1691–1714, December 2010.
- [93] R. Wolf. *LRIT/HRIT Global Specification*, August 1999.
- [94] Y. Yang, H. Lin, Z. Guo, and J. Jiang. A data mining approach for heavy rainfall forecasting based on satellite image sequence analysis. *Computers & Geosciences*, 33(1):20 – 30, 2007.
- [95] Z. Yin and T. Kanade. Restoring artifact-free microscopy image sequences. In *IEEE International Symposium on Biomedical Imaging*, 2011.
- [96] H. Young, R. Freedman, A. Ford, and F. Sears. *Sears and Zemansky's university physics: with modern physics*. Pearson Addison Wesley, 11th edition, 2004.

## Data handling toolbox

The CLOUD toolbox for MATLAB is included here and consists of functions useful for working with DMI weather radar composite files and Level 1.5 satellite data from Meteosat Second Generation. Examples of how to use the functions in the toolbox are given in Appendix A.5.

MATLAB version 7.11.0.584 (R2010b) was used for development of the toolbox and version 4.4 of the PROJ library is a prerequisite for using the projection capabilities in the toolbox.

### contents.m

```

1 function contents
2 % This is the contents of the CLOUD toolbox used for reading, projecting
3 % and visualizing DMI weather radar data and Meteosat Second Generation
4 % satellite data in HDF5 format
5 %
6 % The projections are done using the PROJ v. 4.4 library and relies on an
7 % available build of this. See http://trac.osgeo.org/proj/
8 %
9 % Coastlines are extracted beforehand from the GSHHS database.
10 % See http://www.ngdc.noaa.gov/mgg/shorelines/gshhs.html
11 %
12 % =====
13 % Toolbox contents
14 % =====
15 % +cloud
16 %     contents.m           This file
17 %     init.m              Initialization of parameters used in the toolbox
18 %
19 % +project      Functions for projection handling
20 %     getGridValues.m    Retrieves gridded values from a data matrix
21 %     grid2sat.m        Project common grid to satellite's GEOS
22 %                       projection
23 %     twobytwo.m        Sets up the default common grid of 400x500
24 %                       pixels with 2kmx2km spatial resolution
25 %     grid2rad.m        Project common grid to radar's polar
26 %                       stereographic projection
27 %     projWrapper.m     Wraps the PROJ.4 library. The path for
28 %                       proj.exe must be set within this file
29 %
30 % +radar        Functions specific for radar data
31 %     coverage.m        Determine radar coverage in new grid
32 %                       and retrieve boundary of coverage
33 %                       latitude and longitude.
34 %     DefaultRadar.m    Retrieve the default radar projection,
35 %                       i.e. Polar Stereographic Projection

```

## APPENDIX A. DATA HANDLING TOOLBOX

```

36 %      getReferenceMatrix.m      Get the reference matrix for the radar
37 %                                 composite data for use with map2pix and
38 %                                 similar functions.
39 %      showdBZscale.m            Show a colorbar with dBZ below
40 %      dBZcolormap.m            Apply the standard colormap for display of
41 %                                 radar reflectivity
42 %      readComposite.m          Read DMI composite radar file format (.com)
43 %
44 %      +satellite      Functions specific for satellite data
45 %      DefaultGEOS.m    Retrieve the default geostationary
46 %                                 satellite projection (GEOS) parameters
47 %      getCalibration.m  Get calibration parameters from metadata
48 %      getMetadata.m    Retrieve metadata from satellite data file
49 %      read.m            Read the VIS + IR images from the satellite
50 %                                 HDF5 file and return as brightness
51 %                                 temperatures in a matrix of M x N x 11
52 %      counts2Tb.m      Converts digital counts in satellite data
53 %                                 to brightness temperatures
54 %      getCentreWavelengths.m Contains the centre wavelengths as stated
55 %                                 by documentation
56 %      getReferenceMatrix.m Get default reference matrix for satellite
57 %                                 data or specify a subregion, whereby the
58 %                                 reference matrix reflects this.
59 %      crop.m           Crop the satellite image to a
60 %                                 latitude/longitude specific region
61 %      getImageRegion.m Retrieve the pixel region in the satellite
62 %                                 data based on information from the metadata
63 %                                 and a specified region
64 %      h52matrix.m      Does the actual conversion from HDF5 to
65 %                                 matrix
66 %
67 %      +visualize      Methods for visualization of data
68 %      drawCoastline.m Draws coastlines extracted from the GSHHS
69 %                                 database
70 %      filename2date.m  Converts radar or satellite filenames to
71 %                                 a date vector
72 %      onmap.m          Visualizes a grid of data on a map with
73 %                                 coastlines in a Mercator projection, using
74 %                                 specified grid latitudes and longitudes
75 % =====
76 %      The CLOUD toolbox
77 %      By Jacob S. Vestergaard
78 %      jacob.vestergaard@gmail.com
79 %      Last edit: June 21, 2011
80 % =====

```

### init.m

```

1  % Contains parameters used in the cloud toolbox
2
3  data_dir = 'data';
4  channelnames = {'VIS0.6', 'VIS0.8', 'NIR1.6', 'IR3.9', 'WV6.2', 'WV7.3' ...
5      'IR8.7', 'IR9.7', 'IR10.8', 'IR12.0', 'IR13.4'};
6
7  % Subset region
8  subset_latlim = [50, 62];
9  subset_lonlim = [0, 22];
10
11 % Display region
12 lonlim=[4.5 18.5];
13 latlim=[53.3 60];

```





## A.1 Projection methods

### getGridValues.m

```

1 function Z = getGridValues(X, xp, yp)
2 % GETGRIDVALUES Retrieves the values from X at the specified pixel
3 % coordinates
4
5 X(X==0) = NaN;
6
7 Z = zeros([size(xp), size(X,3)]);
8
9 for i=1:size(X,3)
10     z = X(:, :, i);
11
12     %% Get projected grid values
13     inimage = (yp <= size(z,1) & yp >= 1 & xp <= size(z,2) & xp >= 1);
14     ind = sub2ind(size(z), yp(inimage), xp(inimage));
15     Q = zeros(size(xp));
16     Q(inimage) = z(ind);
17     Q(~inimage) = NaN;
18
19     Z(:, :, i) = Q;
20 end
21
22 Z(Z==Inf) = NaN;
23 end

```

### grid2rad.m

```

1 function [xp, yp] = grid2rad(grid)
2 % GRID2SAT Converts the grid longitude/latitude to pixel coordinates in the
3 % radar image
4
5 %% Project lat/lon to satellite map
6 mrad = cloud.radar.DefaultRadar;
7 [xrad, yrad] = cloud.project.projWrapper(grid.lon, grid.lat, mrad);
8
9 %% Convert map coordinates to image coordinates
10 Rrad = cloud.radar.getReferenceMatrix;
11 [xp, yp] = map2pix(Rrad, xrad, yrad);
12 xp = round(xp);
13 yp = round(yp);

```

### grid2sat.m

```

1 function [xp, yp] = grid2sat(metadata, grid, region)
2 % GRID2SAT Converts the grid longitude/latitude to pixel coordinates in the
3 % satellite images
4 %
5 % Inputs
6 %     metadata      Metadata for satellite imagery, containing parameters
7 %                  for the projection
8 %     grid          Grid from e.g. cloud.project.twobytwo
9 %     region        Subset region
10 %
11 % Outputs
12 %     xp            X coordinates of grid in satellite data
13 %     yp            Y coordinates of grid in satellite data

```

```

14
15 %% Project lat/lon to satellite map
16 msat = cloud.satellite.DefaultGEOS(metadata.ProjectionDescriptionLongitudeOfSSP);
17 [xsat,ysat] = cloud.project.projWrapper(grid.lon, grid.lat, msat);
18
19 %% Convert map coordinates to image coordinates
20 Rsat = cloud.satellite.getReferenceMatrix(metadata,region);
21 [xp,yp] = map2pix(Rsat,xsat,ysat);
22 xp = round(xp);
23 yp = round(yp);

```

### projWrapper.m

```

1 function [x_out,y_out] = projWrapper( x_in, y_in, mstruct )
2 %PROJWRAPPER This function wraps PROJ.4. It writes the input coordinats to
3 %a textfile, calls proj.exe with specified arguments and writes the output
4 %to another textfile, reads this and outputs in a matrix
5 %
6 % In case of inputting as latitude/longitude (i.e. forward projection),
7 % input argument x_in must be longitude and y_in must be latitude.
8 % Output will be a matrix with [x(:) y(:)]
9 %
10 % If inputting as map coordinates, i.e. inverse projection, x_in = x and
11 % y_in = y.
12 % Output will be a matrix with [lon(:) lat(:)]
13 %
14 % PROJ library version 4.4 were used here.
15
16 X = [x_in(:) y_in(:)];
17
18 %% Filelocations
19 filestr = datestr(now,'dd-mm-yyyy_HHMMss');
20 projdir = 'C:\proj\src\proj.exe';
21
22 if ~exist(projdir,'file')
23     error('cloud:project:projNotFound', ...
24         ['proj.exe not found at location: ' projdir]);
25 end
26
27 tempdir = 'C:\proj\temp';
28 if ~exist(tempdir,'dir')
29     warning('cloud:project:tempDirNotExists', ...
30         'Temp_dir_for_PROJ_output_not_found.Creating_dir. ');
31     mkdir(tempdir);
32 end
33 outfile = fullfile(tempdir,[filestr,'_out.txt']);
34 infile = fullfile(tempdir,[filestr,'_in.txt']);
35
36 %% Write to input-file
37 % dlmwrite(infile,X,'delimiter',' ','precision','%.5f') % SLOW
38 fid = fopen(infile,'w+');
39 fprintf(fid,'%.5f %.5f\n',X');
40 fclose(fid);
41
42 %% Combine commandline
43 S = StringBuffer;
44 S.add(projdir);
45
46 % GEOS parameters
47 if isfield(mstruct,'a')
48     S.add(' +a');

```

## APPENDIX A. DATA HANDLING TOOLBOX

```
49     S.add(num2str(mstruct.a));
50 end
51 if isfield(mstruct, 'h')
52     S.add('□+h=');
53     S.add(num2str(mstruct.h));
54 end
55 if isfield(mstruct, 'lon_0') % also STERE
56     S.add('□+lon_0=');
57     S.add(num2str(mstruct.lon_0));
58 end
59 if isfield(mstruct, 'b')
60     S.add('□+b=');
61     S.add(num2str(mstruct.b));
62 end
63
64 % STERE parameters
65 if isfield(mstruct, 'lat_0')
66     S.add('□+lat_0=');
67     S.add(num2str(mstruct.lat_0));
68 end
69 if isfield(mstruct, 'lat_ts')
70     S.add('□+lat_ts=');
71     S.add(num2str(mstruct.lat_ts));
72 end
73 if isfield(mstruct, 'x_0')
74     S.add('□+x_0=');
75     S.add(num2str(mstruct.x_0));
76 end
77 if isfield(mstruct, 'y_0')
78     S.add('□+y_0=');
79     S.add(num2str(mstruct.y_0));
80 end
81 if isfield(mstruct, 'ellps')
82     S.add('□+ellps=');
83     S.add(num2str(mstruct.ellps));
84 end
85
86 S.add(['□+proj=' mstruct.mapprojection]);
87 if mstruct.inv
88     S.add('□-I');
89 end
90 S.add('□-f□%.4f');
91 S.add(['□', infile]);
92 S.add(['□>□', outfile]);
93 system(S.string);
94
95 %% Read output file
96 out = load(outfile, '-ascii');
97 delete(infile, outfile);
98
99 %% Assign to output
100 x_out = reshape(out(:,1), size(x_in));
101 y_out = reshape(out(:,2), size(y_in));
102
103 end
```

### twobytwo.m

```
1 function [ grid ] = twobytwo( gridsize )
2 %TWBYTWO Setup grid with 2x2km pixels.
3 % Default gridsize is 400x500 and projection is the polar stereographic,
```

```
4 % which is also used for the radar data
5
6 if nargin<1
7     gridsize = [400,500];
8 end
9 %% Setup radar image grid
10 [col,row] = meshgrid(1:gridsize(2),1:gridsize(1));
11
12 %% Convert to map coordinates
13 res = 2000;
14 x11 = 26785.817348;
15 y11 = 821728.101419;
16
17 Rrad = makerefmat(x11,y11,res,-res);
18 [xmap,ymap] = pix2map(Rrad,row,col);
19
20 %% Project to lat/lon
21 mrad = cloud.radar.DefaultRadar;
22 mrad.inv = true;
23 [lon, lat] = cloud.project.projWrapper(xmap,ymap,mrad);
24
25 grid = struct('lat',lat,'lon',lon,'res',res, ...
26             'R',Rrad,'ncol',gridsize(2),'nrow',gridsize(1));
27
28
29 end
```

## A.2 Radar data methods

### coverage.m

```

1 function [M, b_lon, b_lat] = coverage(X, xp, yp, grid)
2 % COVERAGE Determines radar coverage from radar data
3 %
4 % Inputs
5 %     X           Satellite image in original grid
6 %     xp          Pixel x-coordinates in original grid corresponding to new
7 %                grid
8 %     yp          Pixel y-coordinates in original grid corresponding to new
9 %                grid
10 %     grid        New grid struct, with reference matrix grid.R
11 %
12 % Outputs
13 %     M           Binary mask same size as new grid with 1 for inside radar
14 %                coverage and 0 outside
15 %     b_lon       Longitudes of radar boundary
16 %     b_lat       Latitudes of radar boundary
17
18 % Transfer to grid without replacing Inf
19 [n,m] = size(X);
20 M = zeros(size(xp));
21 logind = (xp<=m & xp >=1 & yp<=n & yp>=1);
22 ind = sub2ind(size(X),yp(logind),xp(logind));
23 XX = (X~=inf);
24 M(logind) = XX(ind);
25
26 % Find edge
27 [B,~] = bwboundaries(M);
28 b = B{1};
29
30 % Project edge to lat/lon
31 mgrid = cloud.radar.DefaultRadar;
32 mgrid.inv = 1;
33 [bx,by] = pix2map(grid.R,b(:,1),b(:,2));
34 [b_lon,b_lat]= cloud.project.projWrapper(bx,by,mgrid);

```

### DefaultRadar.m

```

1 function mstruct = DefaultRadar
2
3 mstruct = defaultm('');
4 mstruct.mapprojection = 'stere';
5 mstruct.lon_0 = 10.5666;
6 mstruct.lat_0 = 56.0;
7 mstruct.lat_ts = 56.0;
8 mstruct.ellps = 'WGS84';
9 mstruct.x_0 = 450000;
10 mstruct.y_0 = 350000;
11 mstruct.inv = false;
12
13 % +proj=stere +ellps=WGS84 +x_0=450000
14 % +y_0=350000 +lat_0=56.0 +lon_0=10.5666 +lat_ts=56.0
15 end

```

### dBZcolormap.m

```

1 function cmap = dBZcolormap(h)
2
3 if nargin<1
4     h=gca;
5 end
6
7 cmap = [189,215,231
8         107,174,214
9         33,113,181
10        65,171,93
11        35,139,69
12         0,109,44
13        255,255,0
14        253,141,60
15        227,26,28
16        203,24,29
17        165,15,21
18        174,1,126
19        122,1,119
20        73,0,106
21        30,0,60
22        15,0,15
23        5,0,5]/255;
24
25 if narginout <1 | ~nargin>0
26     colormap(h,cmap);
27     caxis([-2.5,82.5]);
28 end
29
30 end

```

### getReferenceMatrix.m

```

1 function [ R ] = getReferenceMatrix
2 % GETREFERENCEMATRIX Sets up the reference matrix for the radar data
3 % composite file
4
5 res = 1000;
6 x11 = 26785.817348;
7 y11 = 821728.101419;
8 R = makerefmat(x11,y11,res,-res);
9
10
11 end

```

### readComposite.m

```

1 function X = readComposite(filename)
2 % READCOMPOSITE Reads a composite radar files and returns as data matrix X
3 %
4 % Inputs
5 %     filename    Path to composite file
6 %
7 % Outputs
8 %     X           Radar data in dBZ
9
10
11 fid = fopen(filename);
12 fseek(fid,99,0);
13 Nx = 963;

```

## APPENDIX A. DATA HANDLING TOOLBOX

```
14 Ny = 844;  
15 X = fread(fid, [Nx,Ny], 'uint8');  
16 fclose(fid);  
17 X(X==0) = NaN;  
18 X(X==255) = Inf;  
19  
20 % Convert to dBZ and transpose  
21 X = (X' * 0.5) - 32;
```

### showdBZscale.m

```
1 function showdBZscale  
2  
3 hc = colorbar;  
4 xlabel(hc, 'dBZ');
```



## A.3 Satellite data methods

### DefaultGEOS.m

```

1 function mstruct = DefaultGEOS(sub_lon)
2 %DEFAULTGEOS Returns an mstruct with the default parameters for the
3 %Geostationary Satellite Projection (GEOS)
4
5 mstruct = defaultm('');
6 mstruct.mapprojection = 'geos';
7 mstruct.a = 6378169.0;
8 mstruct.b = 6356583.8;
9 mstruct.h = 35785831;
10 mstruct.lon_0 = sub_lon;
11 mstruct.inv = false;

```

### counts2Tb.m

```

1 function Xb = counts2Tb(X,C,lambda0)
2 % Converts the matrix of counts to brightness temperatures
3 % using the conversion outlined in the Level 1.5 Image specification
4 %
5 % Inputs
6 %     X           Matrix of counts N x M x p
7 %     C           Matrix with calibration slope and offset of size p x 2,
8 %                 such that cal_slope = C(:,1) and cal_offset = C(:,2)
9 %     lambda0     The p centre wavelengths.
10 %
11 %     Xb          X converted to brightness temperatures in \mu meters
12
13 %% Constants
14 c1 = 1.19104e-5;
15 c2 = 1.43877;
16
17 %% Quantities
18 v = 1e4./lambda0;
19 X = double(X);
20 Xb = X;
21 [N,M,p] = size(X);
22 for i=1:p
23     x = X(:, :, i);
24     L = 0*x;
25     L(x>0) = x(x>0)*C(i,1) + C(i,2);
26     Tb = (c2*v(i)) ./ (log( 1 + v(i)^3 * c1./L));
27     Xb(:, :, i) = real(Tb);
28 end
29 Xb = Xb - 273.15; % Convert to degrees celsius

```

### crop.m

```

1 function [Xc,region] = crop(Xb,metadata,latlim,lonlim)
2 % CROP Crops the satellite data to the region specified by latitude and
3 % longitude limits based on the image metadata
4 %
5 % Inputs
6 %     Xb           Satellite imagery of size M x N x 11
7 %     metadata     Metadata accompanying satellite image
8 %     latlim       Latitude boundaries
9 %     lonlim       Longitude boundaries

```

## APPENDIX A. DATA HANDLING TOOLBOX

```
10 %  
11 %   Outputs  
12 %       Xc           Cropped image data  
13  
14 region = cloud.satellite.getImageRegion(metadata, latlim, lonlim);  
15  
16 Xc = Xb(region.W:region.E, region.N:region.S,:);
```

### getCalibration.m

```
1 function C = getCalibration(info)  
2 % Retrieves calibration coefficients from hdf5 info object  
3 %   Outputs  
4 %       C   First column is cal_slope  
5 %           Second column is cal_offset  
6  
7 X = hdf5read(info.GroupHierarchy.Groups.Groups.Groups.Groups(2) ...  
8           .Groups(1).Groups(6).Datasets(5));  
9  
10 P = length(X);  
11 C = zeros(P,2);  
12 for i=1:P  
13     C(i,:) = cell2mat(X(i).Data);  
14 end
```

### getCentreWavelengths.m

```
1 function lambda0 = getCentreWavelengths(channels)  
2 % Retrieves centre wavelengths for Level 1.5 imagery  
3 % Units are in \mu meter  
4 table = [0.635  
5         0.81  
6         1.64  
7         3.92  
8         6.25  
9         7.35  
10        8.70  
11        9.66  
12        10.80  
13        12.00  
14        13.40  
15        0.75];  
16  
17 lambda0 = table(channels);
```

### getImageRegion.m

```
1 function [region,row,col] = getImageRegion(metadata, latlim, lonlim)  
2 % GETIMAGEREGION Determines the north, south, west, east pixel positions of  
3 % an image from the scenario SC, corresponding to the latitude, longitude  
4 % limits  
5  
6 R = cloud.satellite.getReferenceMatrix(metadata);  
7 [lon, lat] = meshgrid(lonlim, latlim);  
8  
9 %% Convert to map coordinates  
10 mstruct = cloud.satellite.DefaultGEOS(metadata ...  
11         .ProjectionDescriptionLongitudeOfSSP);  
12 [xmap, ymap] = cloud.project.projWrapper(lon, lat, mstruct);
```

```

13
14 %% Convert to pixel coordinates
15 [row,col] = map2pix(R,xmap,ymap);
16 row = round(row);
17 col = round(col);
18 %% Use lower left for South and West,
19 % lower right for East and upper left for North coordinate
20 region = struct('S', row(1,1), ...
21               'W', col(1,1), ...
22               'E', col(1,2), ...
23               'N', row(2,1));

```

### getMetadata.m

```

1 function M = getMetadata(info)
2 % GETMETADATA Reads the necessary meta data from filename and returns it as
3 % a struct
4
5 %% Init metadata struct
6 M = struct('');
7 %% Get IMAGE_DESCRIPTION
8 g = hdf5read(info.GroupHierarchy.Groups.Groups.Groups.Groups(2) ...
9           .Groups(1).Groups(5).Datasets(1));
10 % Get [SSP, NumberOfLines, LineDirGridStep (IR+HRV)
11 indices = [2, 3, 5, 8, 10];
12 M = addToMetadata(M,g,indices);
13
14 %% Get SUBSET information
15 s = hdf5read(info.GroupHierarchy.Groups.Groups.Groups.Groups(2).Datasets(1));
16 M = addToMetadata(M,s,1:12);

```

### getReferenceMatrix.m

```

1 function [ R ] = getReferenceMatrix( metadata, region)
2 %GETREFERENCEMATRIX Sets up the reference matrix for the specified scenario
3 %and channel
4
5 res = metadata.ReferenceGridVIS_IRLineDirGridStep * 1000;
6 xx = (metadata.ReferenceGridVIS_IRNumberOfLines / 2 ...
7       - metadata.VIS_IRWestColumnSelectedRectangle);
8 yy = (metadata.ReferenceGridVIS_IRNumberOfLines / 2);
9
10 if nargin==2
11     % A subset region specified
12     xx = xx + (region.W - 1);
13     yy = yy - region.N - 1;
14 end
15
16 x11 = xx * res;
17 y11 = yy * res;
18
19 R = makerefmat(x11,y11, res, -res);
20
21 end

```

### h52matrix.m

```

1 function [X, H] = h52matrix(h5info)
2 % h52matrix Extract the image data from the HDF5 info object

```

## APPENDIX A. DATA HANDLING TOOLBOX

```
3 %
4 %   Inputs
5 %       h5info      HDF5 info object
6 %
7 %   Outputs
8 %       X           First N-1 channels image data as rows x cols x N-1 matrix
9 %                   Usually 3 VIS + 8 IR
10 %      H           Last channel image data (Usually panchromatic)
11
12 level15 = h5info.GroupHierarchy.Groups.Groups.Groups.Groups.Groups;
13 numbands = length(level15);
14
15 dims = level15(1).Datasets(1).Dims;
16 X = zeros([dims numbands-1], 'uint16');
17 for i=1:numbands-1
18     X(:, :, i) = hdf5read(level15(i).Datasets(1));
19 end
20
21 if nargin == 2
22     H = hdf5read(level15(numbands).Datasets(1));
23 end
```

### read.m

```
1 function [Xb,metadata] = read(filename)
2 % READ Reads HDF5 satellite data
3 %
4 %   Inputs
5 %       filename    Path to .h5 file to be read
6 %
7 %   Outputs
8 %       Xb          Satellite brightness temperatures in matrix
9 %       metadata    Image acquisition metadata, containing information
10 %                 about map coordinates and others
11 %
12
13 % Read HDF5 file info
14 info = hdf5info(filename);
15
16 % Extract image acquisition metadata
17 metadata = cloud.satellite.getMetadata(info);
18
19 % Get calibration parameters
20 C = cloud.satellite.getCalibration(info);
21
22 % Get VIS + IR
23 X = cloud.satellite.h52matrix(info);
24
25 % Retrieve center wavelengths
26 lambda0 = cloud.satellite.getCentreWavelengths(1:11);
27 Xb = cloud.satellite.counts2Tb(X,C,lambda0);
```

### addToMetadata.m

```
1 function M = addToMetadata(M,S,indices)
2 % Adds the info at specified indices in the hdf5compound S
3 % to the metadata struct M
4
5 for i=1:length(indices)
6     index = indices(i);
```

```
7     [name,value] = getData(S,index);
8     name = strrep(name,'-','');
9     name = strrep(name,'_','');
10    M(1).(name) =str2double(value);
11    end
12    end
```

### **getData.m**

```
1    function [name,value] = getData(S,index)
2    % Read data from hdf5compound S at the specified index
3    data = S(index).Data;
4    name = data{1}.Data;
5    value = data{2}.Data;
6    end
```

## A.4 Visualization methods

## drawCoastline.m

```

1 function h = drawCoastline(ax)
2 %DRAWCOASTLINE Draws the coast line on the current figure
3
4 if nargin<1
5     ax = gca;
6 end
7
8 mPath = mfilename('fullpath');
9 [pathstr,~,~]= fileparts(mPath);
10
11 S = shaperead(fullfile(pathstr,'gshhs_i_denmark'),'UseGeoCoords',true);
12 % geoshow(S,'facecolor',[0.15 0.8 0.15])
13 h = geoshow(ax,[S.Lat],[S.Lon],'Color','black');
14
15 end

```

## filename2date.m

```

1 function dvec = filename2date(s, isRadar)
2 % FILENAME2DATE Converts a filename to date vector
3 %
4 % Inputs
5 % s      Filename.
6 %       Radar data the format is yyyyymmddHHMM.rrrr.com
7 %       (rrrr being resolution, e.g. 1000)
8 %       Satellite data filename:
9 %       ?????-????-????-????-??-yyymmddHHMMss.??????Z-????-?.h5
10 % isRadar false for satellite data and true for radar data
11 %
12 % Outputs
13 % dvec Date vector of length 6, i.e. [yyyy mm dd HH MM ss]
14
15 if nargin<2
16     isRadar=false;
17 end
18
19 if ~isRadar
20     strarr = regexp(s, '-', 'split');
21     dvec = datevec(strarr(6), 'yyymmddHHMMss');
22 else
23     strarr = regexp(s, '\.', 'split');
24     dvec = datevec(strarr(1), 'yyymmddHHMM');
25 end

```

## onmap.m

```

1 function ax = onmap(Z, grid, latlim, lonlim)
2 % SHOWONMAP shows the gridded data Z on a map of DK, using the grid.lat and
3 % grid.lon coordinates
4 % Coastlines are draw on the image and latitude and longitude labels are
5 % shown
6
7
8 %% Show on map
9 ax = axesm('mercator', 'MapLatLimit', latlim, 'MapLonLimit', lonlim);

```

```
10 geoshow(ax,grid.lat,grid.lon,double(Z),'displaytype','texturemap');
11 cloud.visualize.drawCoastline;
12 mlabel('mlabellocation',3:3:18,'mlabelparallel','south');
13 plabel('plabellocation',53:2:59);
14 tightmap;
```

## A.5 Toolbox demos

### radar\_demo.m

```

1  clc, clear, close all;
2  % 1) Reads radar data from DMI composite file format (.com)
3  % 2) Projects to 400 x 500 grid using PROJ.4 wrapper
4  % 3) Displays as images and on a map with coastlines
5  % 4) Display radar boundary
6
7  %% Initialize
8  cloud.init;
9  filename = '200708201410.1000.com';
10  filepath = fullfile(data_dir, filename);
11
12  %% Read radar data
13  X = cloud.radar.readComposite(filepath);
14  dvec = cloud.visualize.filename2date(filename, true);
15
16
17  %% Project to common grid
18  grid = cloud.project.twobytwo;
19  [xp, yp] = cloud.project.grid2rad(grid);
20  Z = cloud.project.getGridValues(X', xp, yp);
21
22  %% Determine radar boundary
23  [M, b_lon, b_lat] = cloud.radar.coverage(X', xp, yp, grid);
24
25  %% Display as images
26  figure,
27  subplot 121,
28  imagesc(X);
29  axis image;
30  cloud.radar.dBZcolormap;
31  cloud.radar.showdBZscale;
32  title('Original');
33
34  subplot 122,
35  imagesc(Z);
36  axis image;
37  cloud.radar.dBZcolormap;
38  cloud.radar.showdBZscale;
39  title('Projected to common grid');
40
41  %% Show on map
42  figure,
43  cloud.visualize.onmap(Z, grid, latlim, lonlim);
44  cloud.radar.dBZcolormap;
45  cloud.radar.showdBZscale;
46  title(['Radar data at ' datestr(dvec, 'yyyy-mm-dd_HH:MM')]);
47
48  % Draw boundary
49  plotm(b_lat, b_lon, 'r--');

```

### satellite\_demo.m

```

1  clc, clear, close all;
2  % 1) Reads satellite data from HDF5 (.h5) file
3  % 2) Crops to a defined subset
4  % 3) Projects to 400 x 500 grid using PROJ.4 wrapper

```



```

5 % 4) Displays as images and on a map with coastlines
6
7 %% Initialize
8 cloud.init;
9 filename = 'MSG2-SEVI-MSG15-0100-NA-20070820141242.760000000Z-996520-1.h5';
10 filepath = fullfile(data_dir,filename);
11
12 %% Read satellite data
13 [Xb,metadata] = cloud.satellite.read(filepath);
14 dvec = cloud.visualize.filename2date(filename);
15
16 %% Crop to northern Europe subset
17 [Xc,region] = cloud.satellite.crop(Xb,metadata,subset_latlim,subset_lonlim);
18
19 %% Project to common grid
20 grid = cloud.project.twobytwo;
21 [xp,yp] = cloud.project.grid2sat(metadata,grid,region);
22 Z = cloud.project.getGridValues(Xc,xp,yp);
23
24 %% Display as images
25 channel = 9; % Choose channel to display
26 figure,
27 subplot 211,
28 imagesc(Xb(:,:,channel)');
29 axis image; colormap gray;
30 title(channelnames{channel});
31
32 subplot 223,
33 imagesc(Xc(:,:,channel)');
34 axis image; colormap gray;
35 title('Cropped');
36
37 subplot 224,
38 imagesc(Z(:,:,channel));
39 axis image; colormap gray;
40 title('Projected to grid');
41
42 %% Display with coast lines
43 figure,
44 cloud.visualize.onmap(Z(:,:,channel),grid,latlim,lonlim);
45 colormap gray;
46 title(['Satellite data at ' datestr(dvec,'yyyy-mm-dd_HH:MM') ...
47       ' (' channelnames{channel} ')']);

```



## Illustrations of data

## B.1 Radar data at hotspot

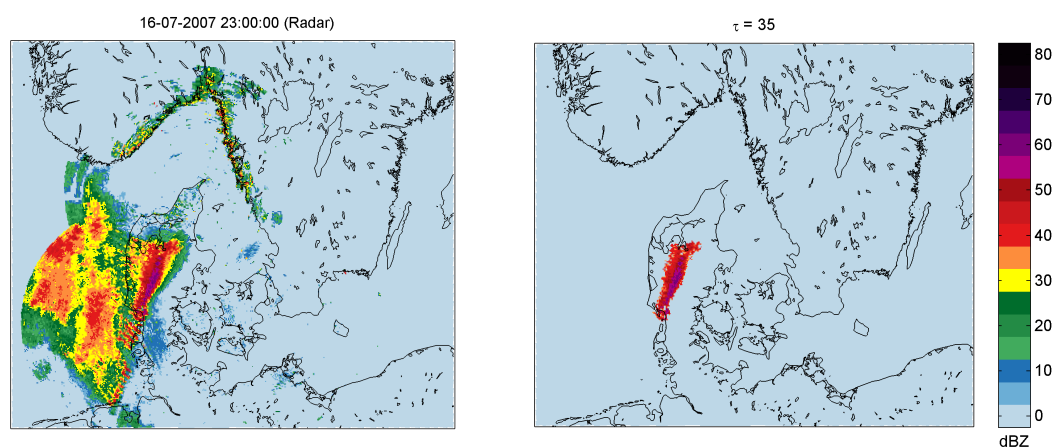


Figure B.1: 2007-07-16. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .

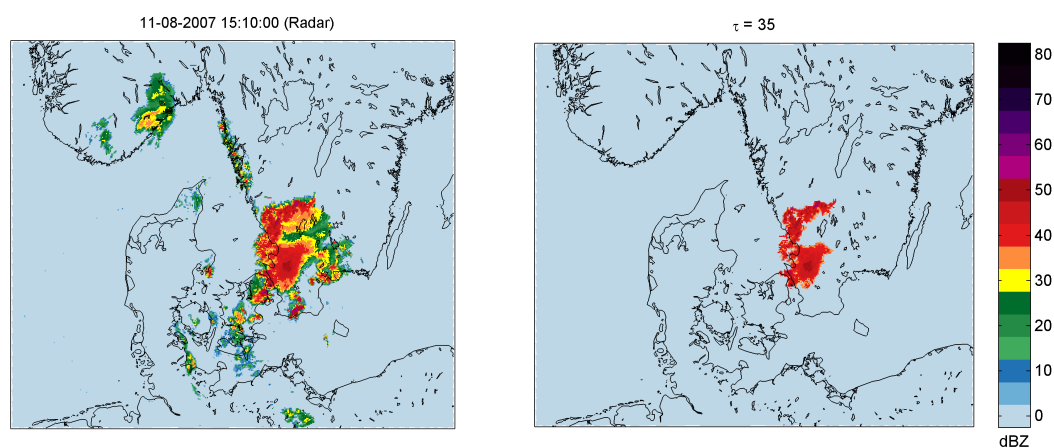
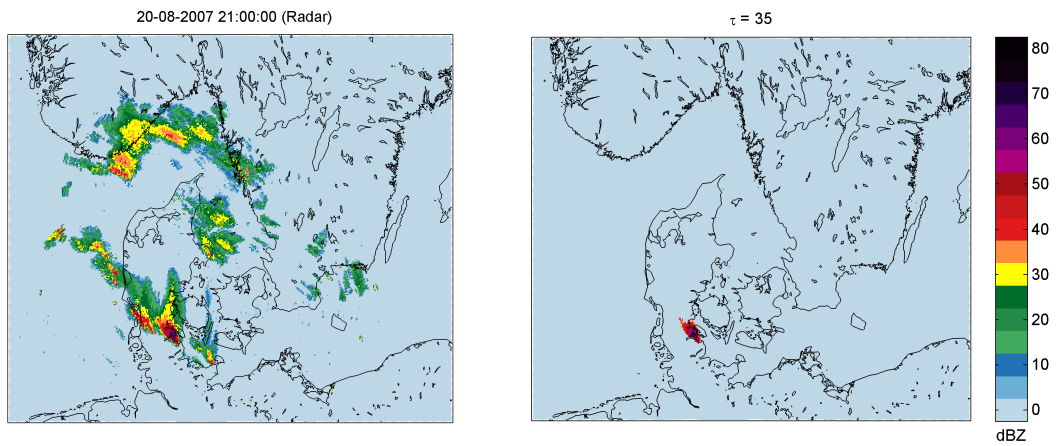
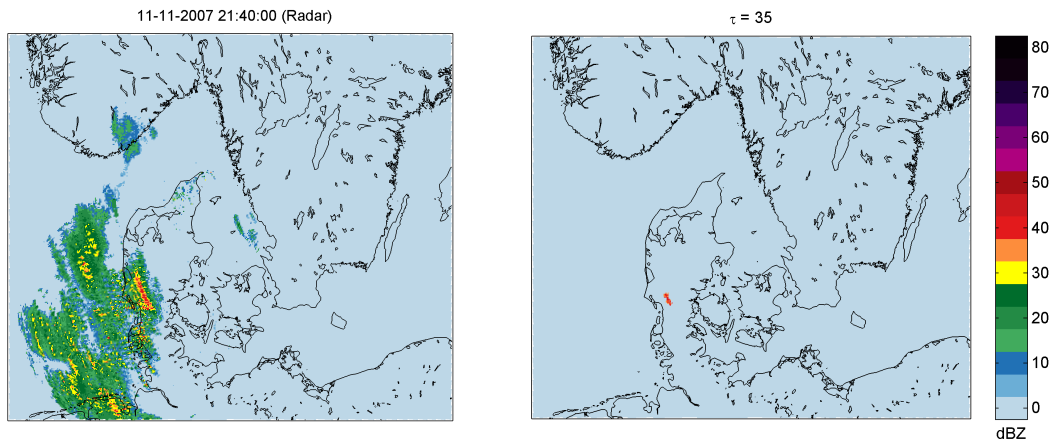


Figure B.2: 2007-08-11. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .

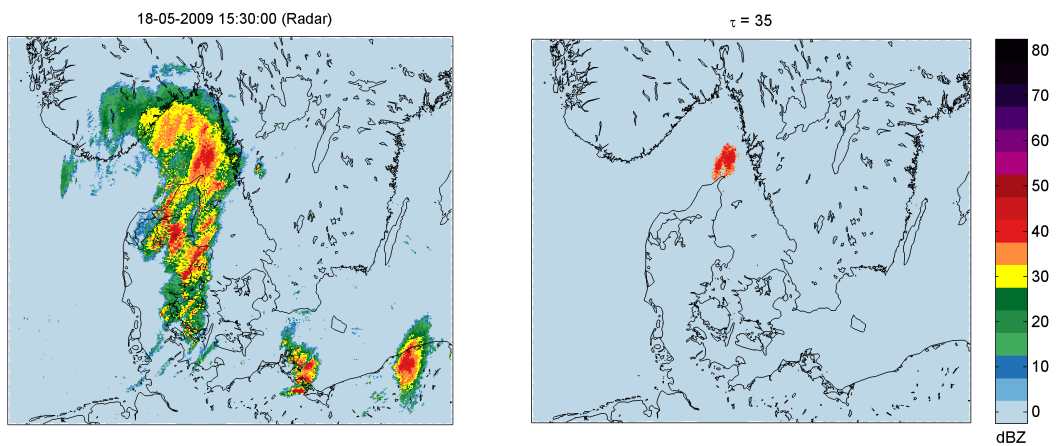
APPENDIX B. ILLUSTRATIONS OF DATA



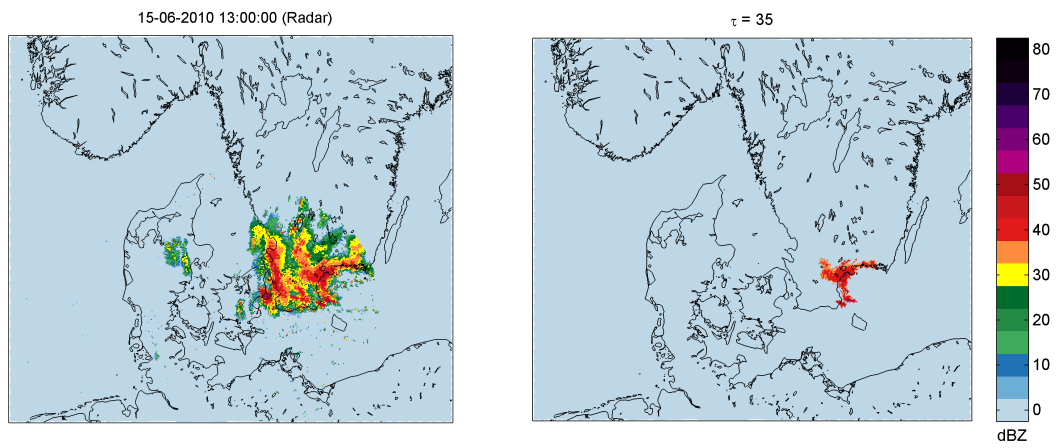
**Figure B.3:** 2007-08-20. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .



**Figure B.4:** 2007-11-11. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .



**Figure B.5:** 2009-05-18. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .

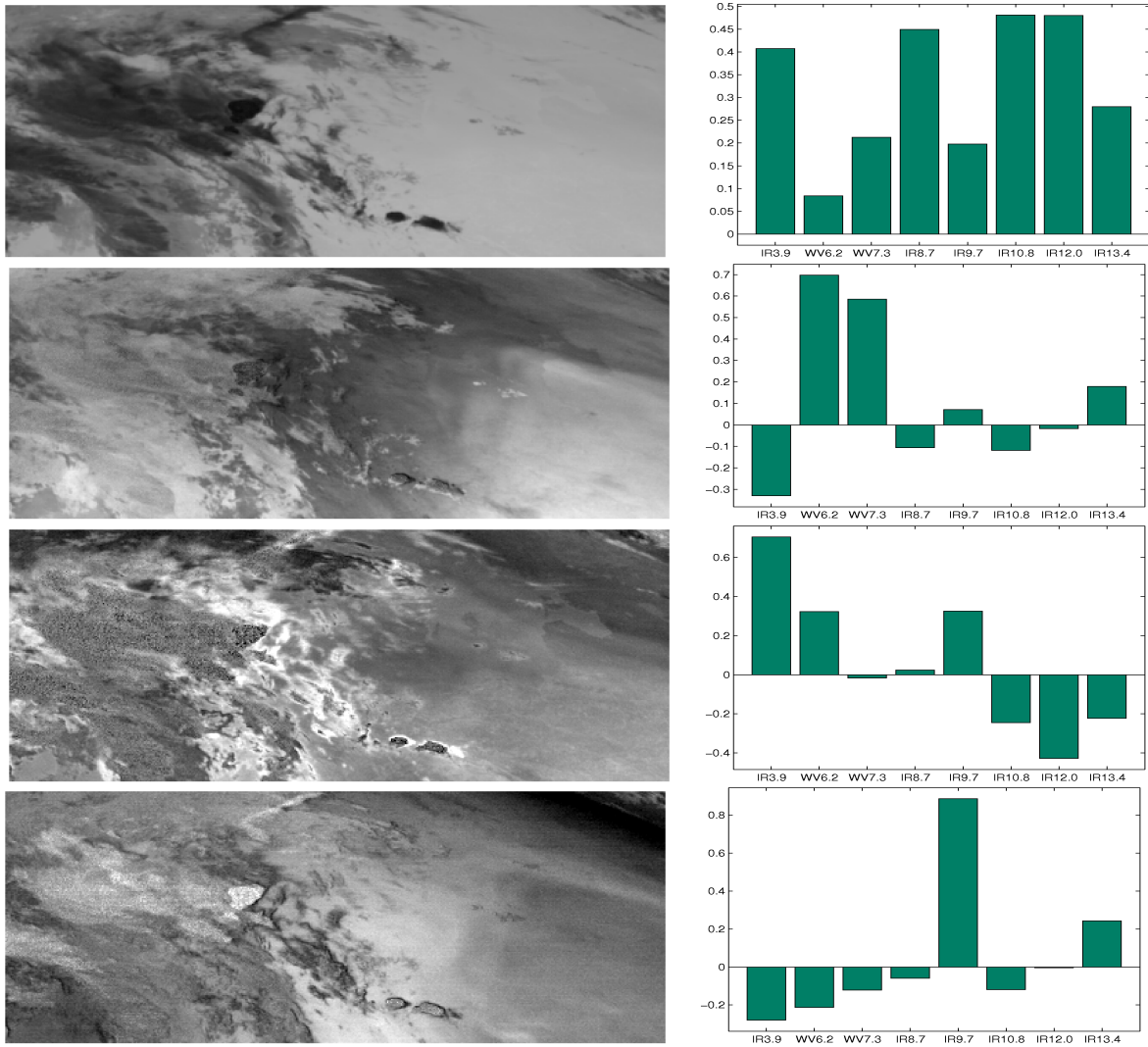


**Figure B.6:** 2010-06-15. (left) Radar data at time of hotspots. (right) Thresholded radar data with  $\tau = 35$ .



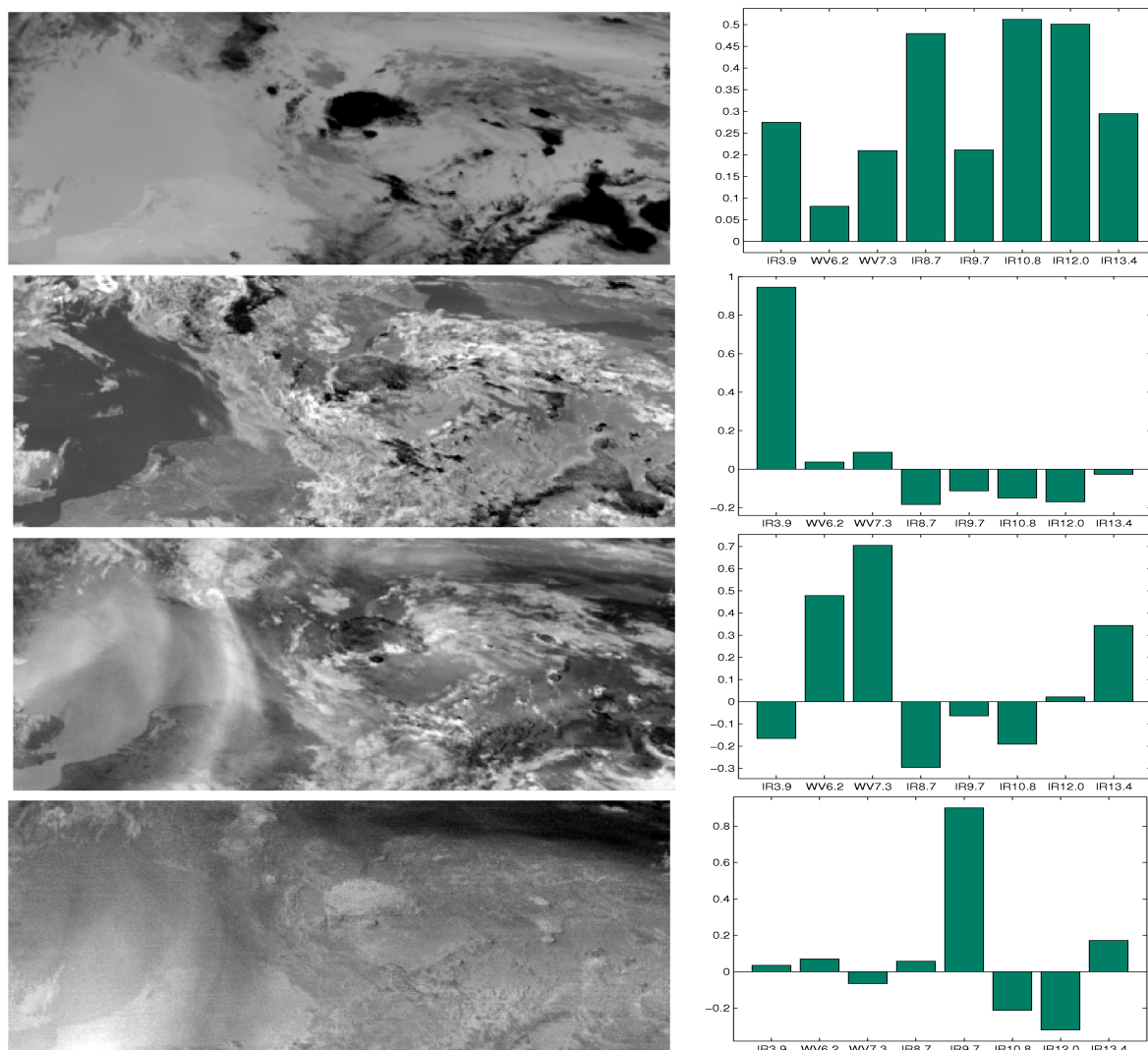
## Supplementary Results

## C.1 Principal Components



**Figure C.1:** 2007-07-16 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.2:** 2007-08-11 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.



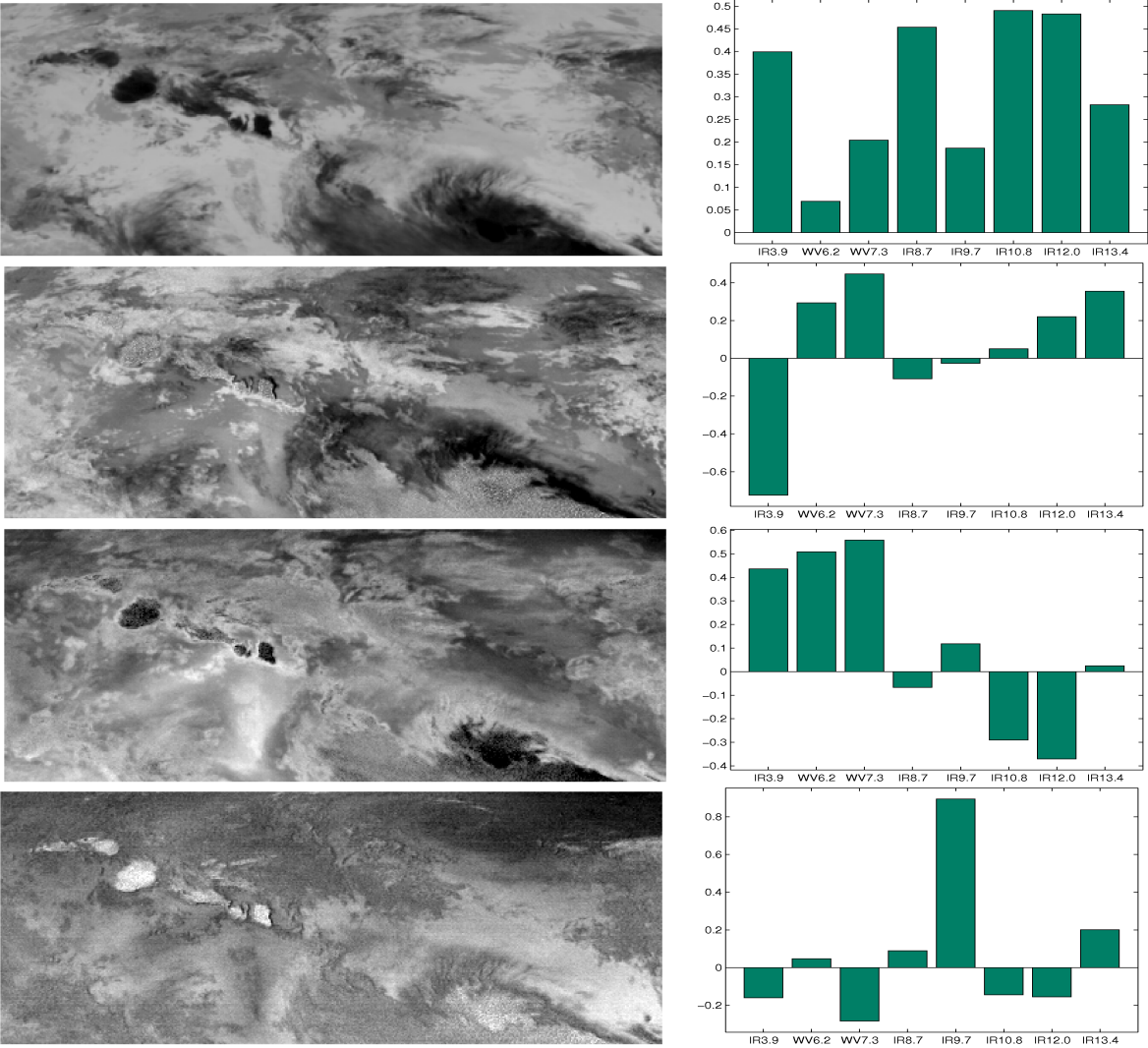
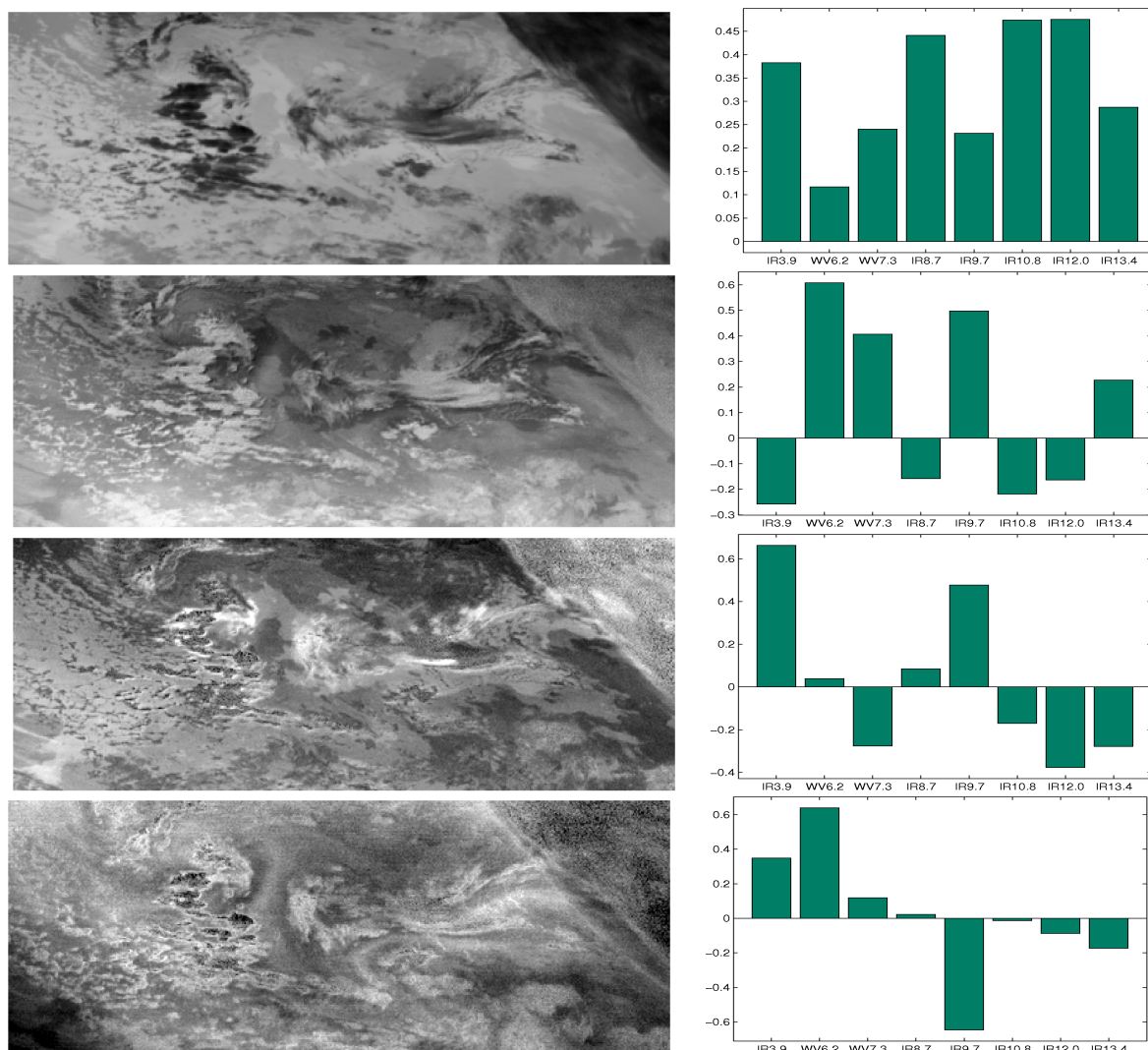


Figure C.3: 2007-08-20 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.4:** 2007-11-11 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

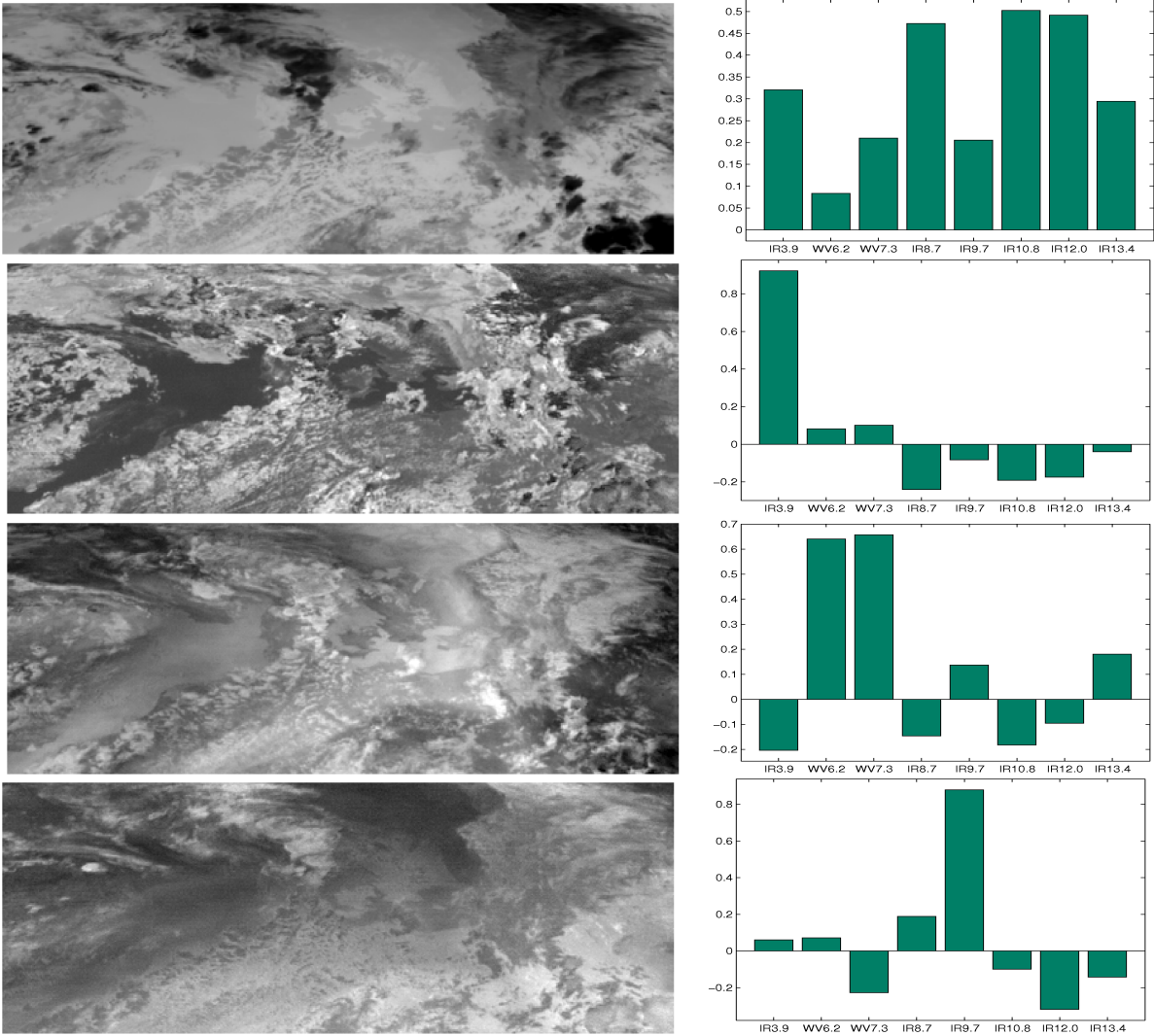
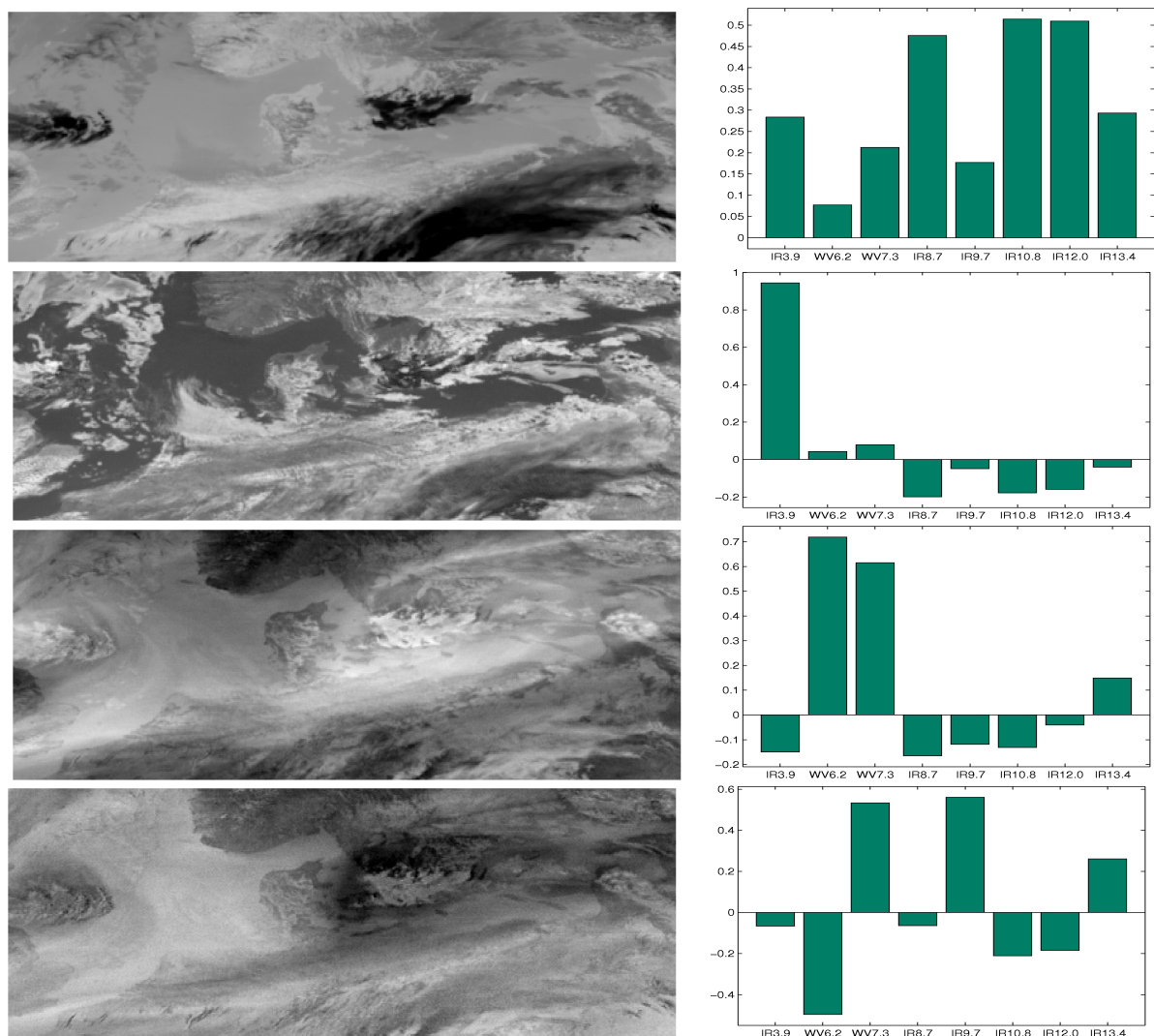


Figure C.5: 2009-05-18 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.6:** 2010-06-15 (left) First four PCs stretched between mean  $\pm$  three standard deviations. (right) Associated eigenvectors.

## C.2 Global PCA

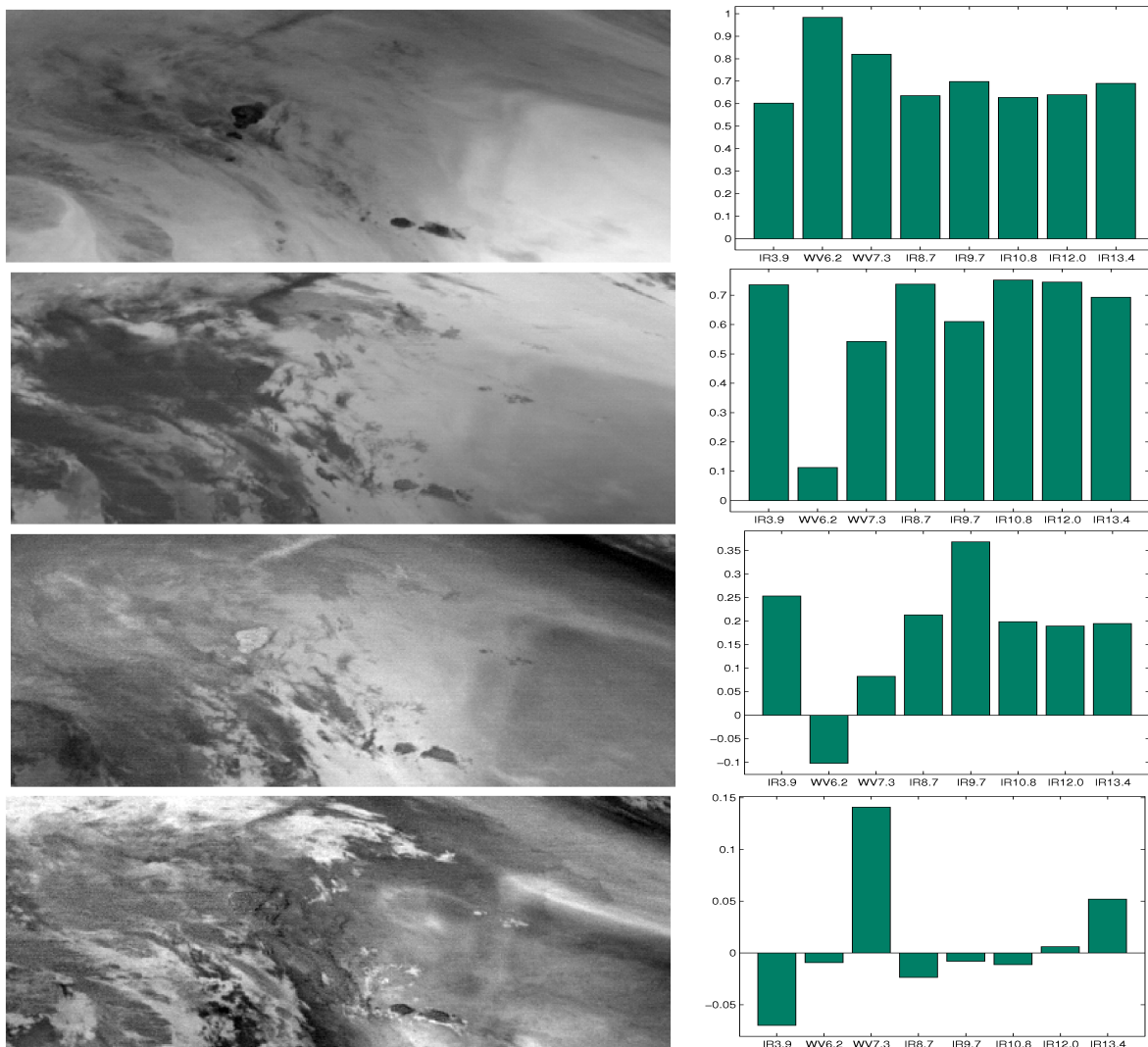
APPENDIX C. SUPPLEMENTARY RESULTS



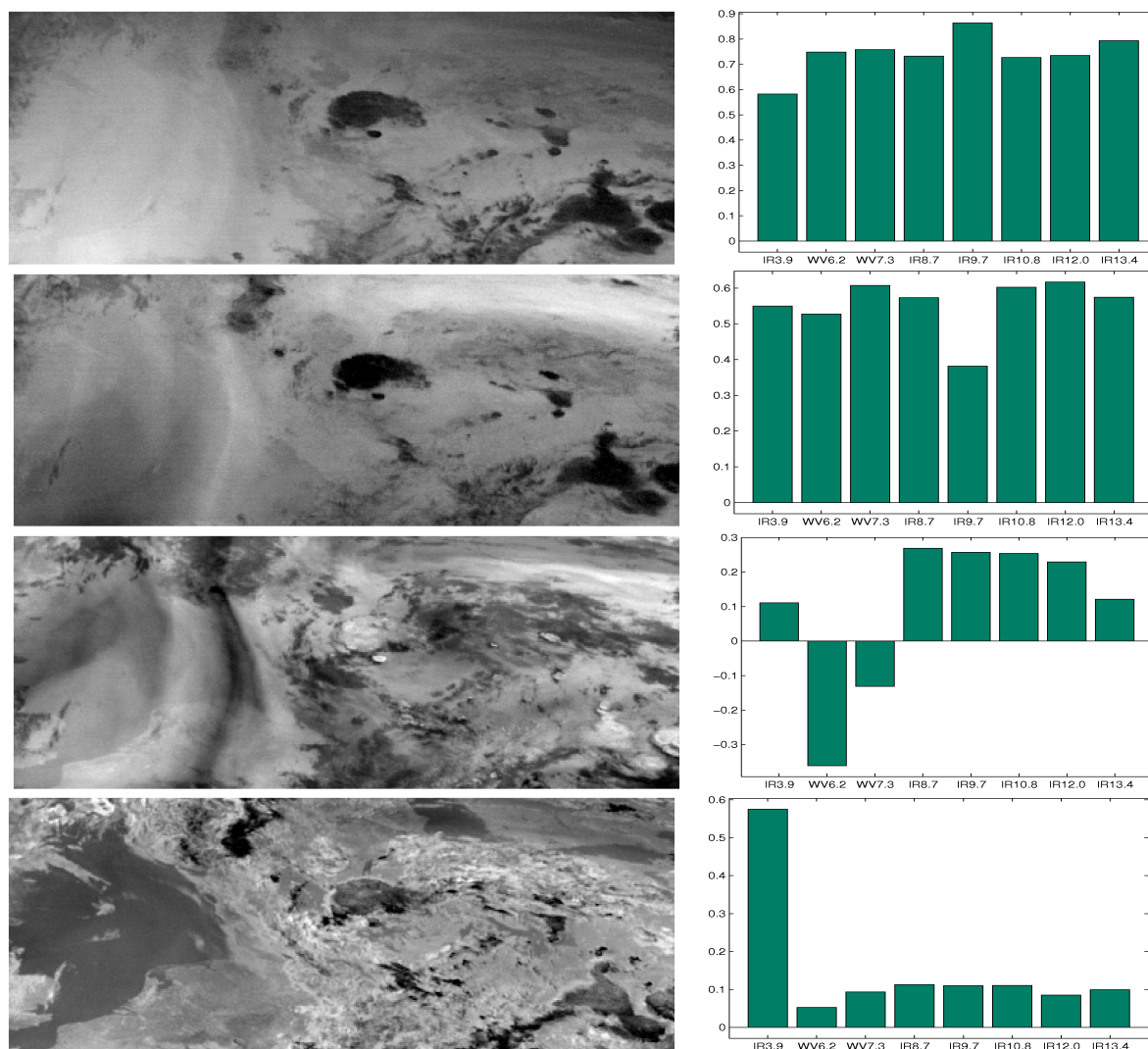
Figure C.7: Principal directions from global PCA sorted according to decreasing variance from left to right, top to bottom.



### C.3 Maximum Autocorrelation Factors

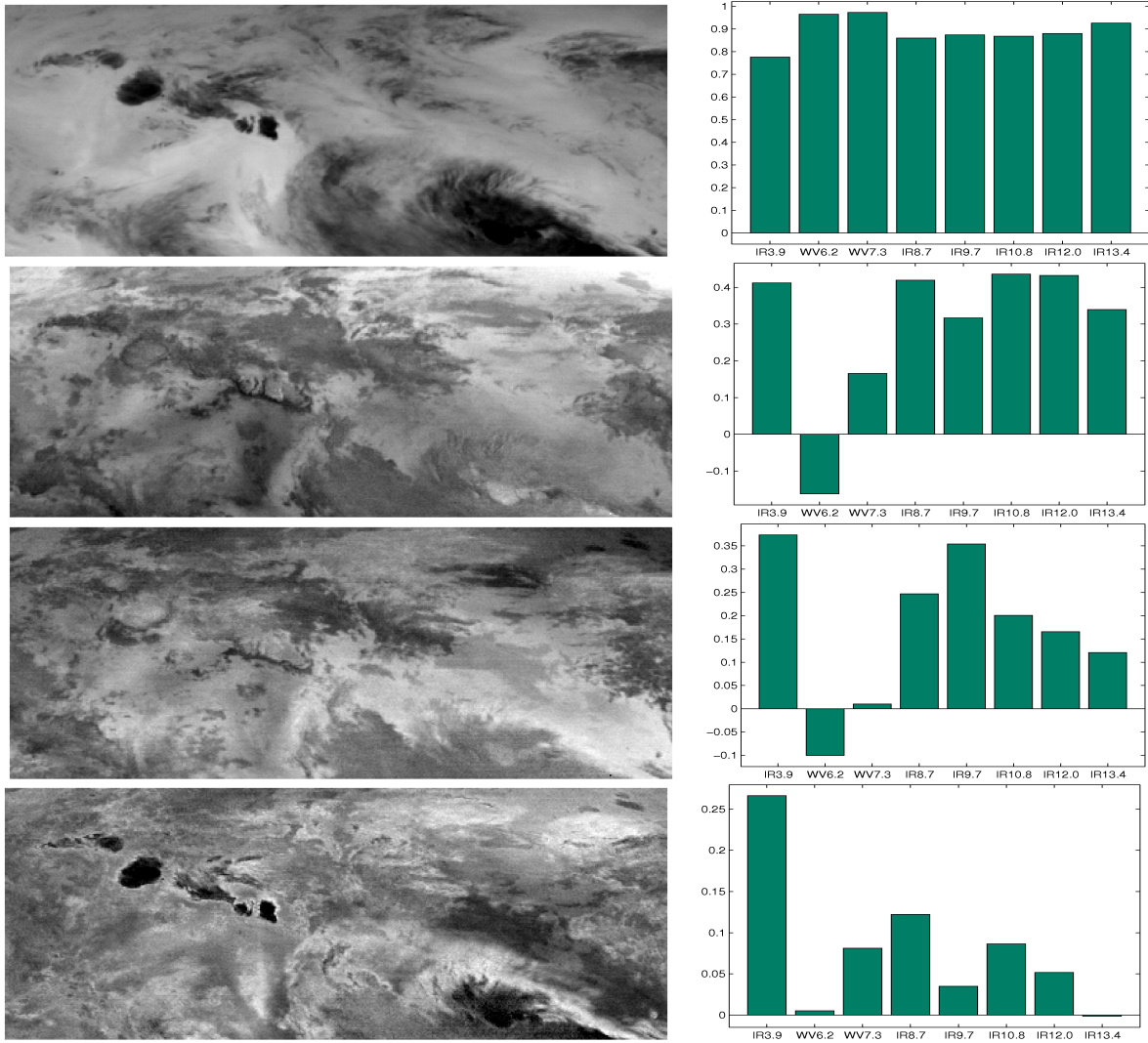


**Figure C.8:** 2007-07-16 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.

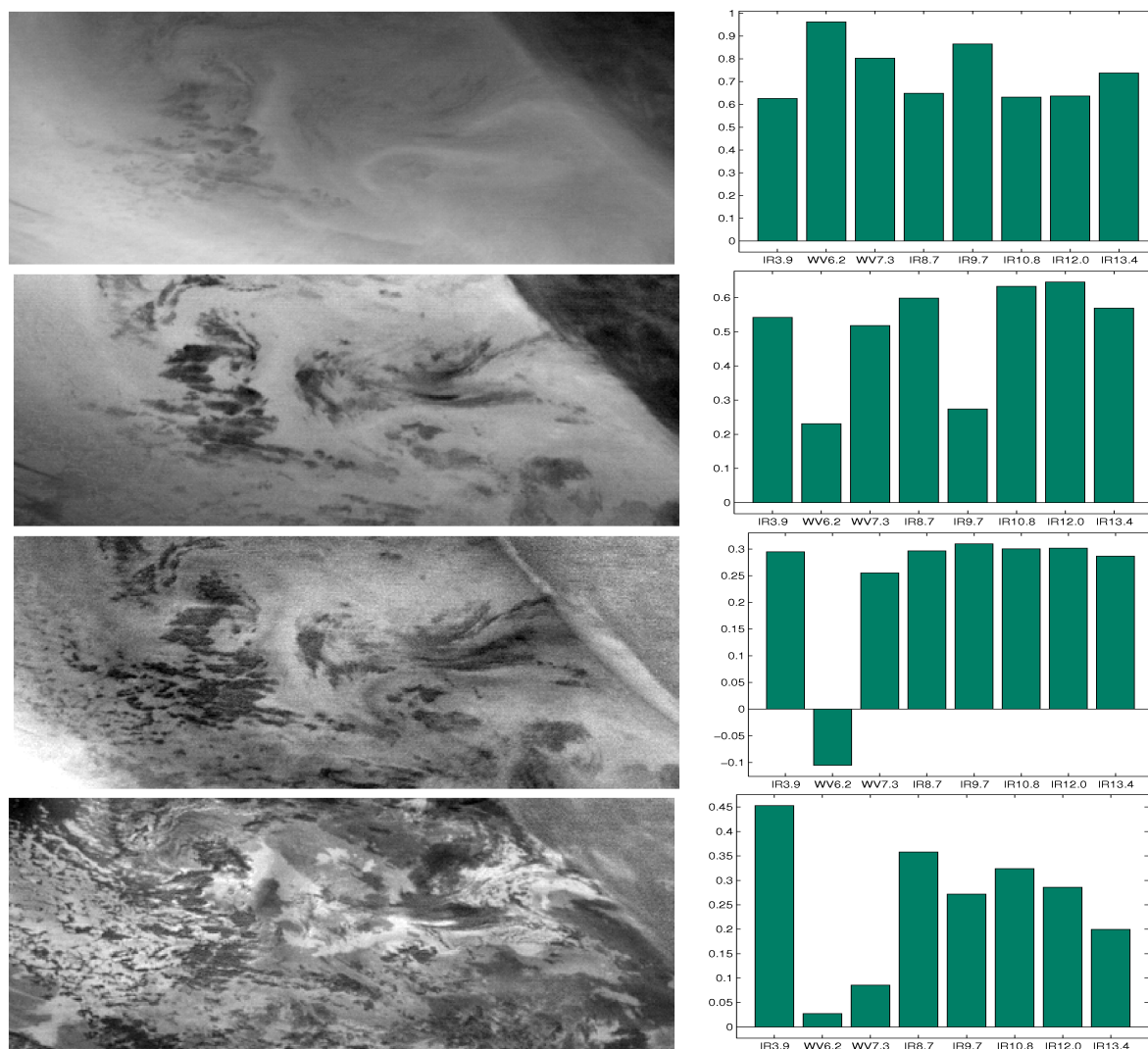


**Figure C.9:** 2007-08-11 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.



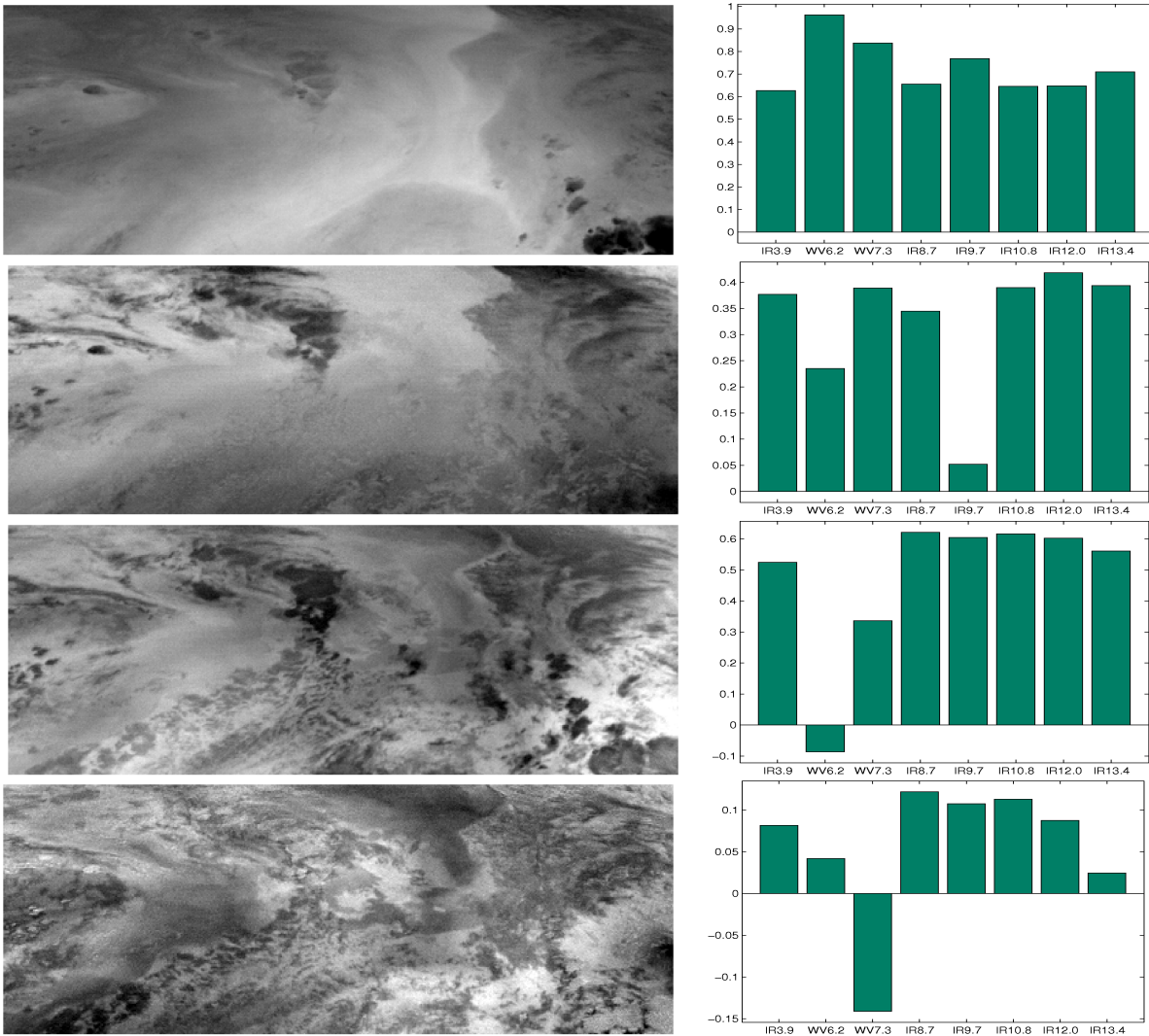


**Figure C.10:** 2007-08-20 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.



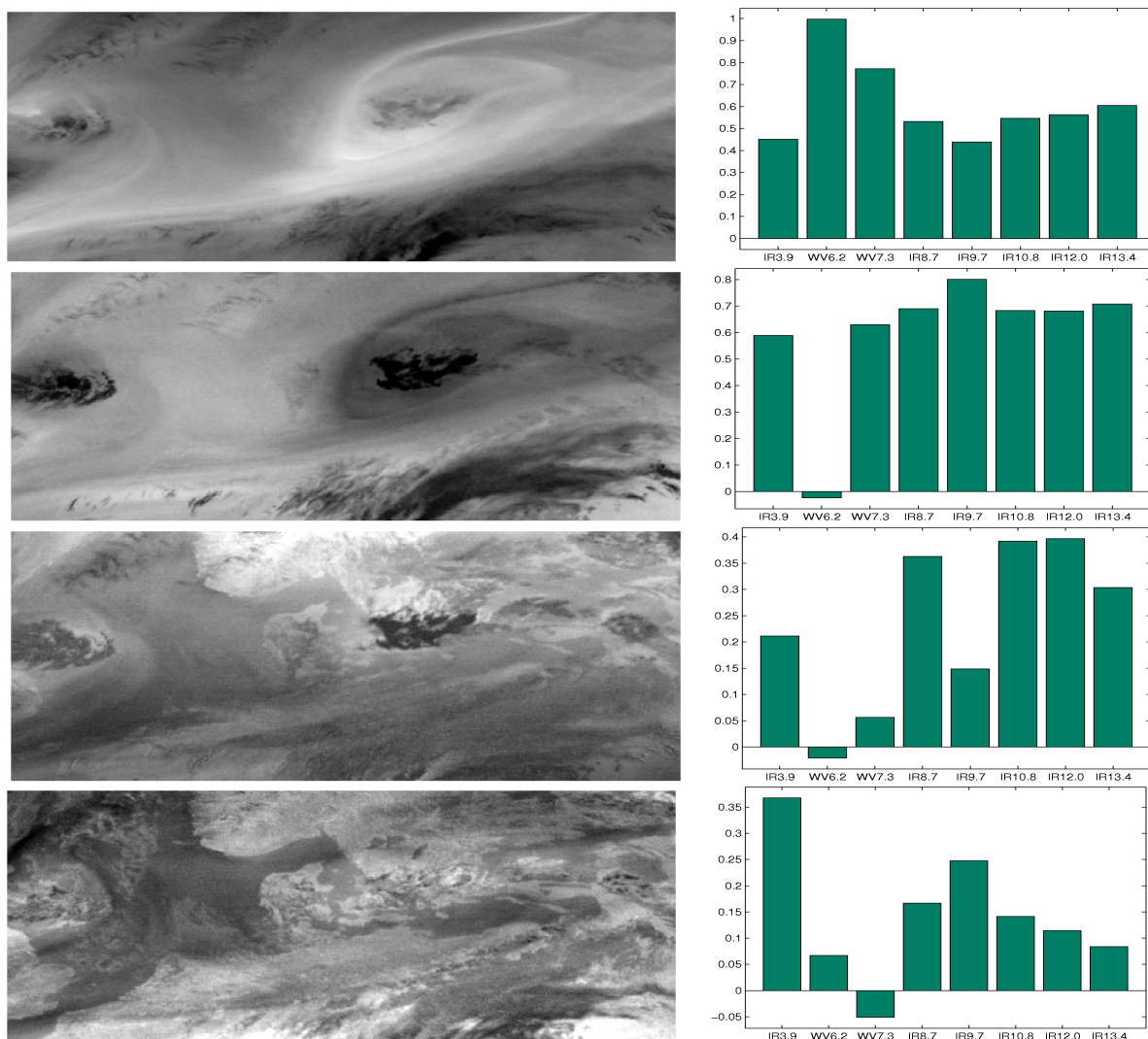
**Figure C.11:** 2007-11-11 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.

### C.3 Maximum Autocorrelation Factors



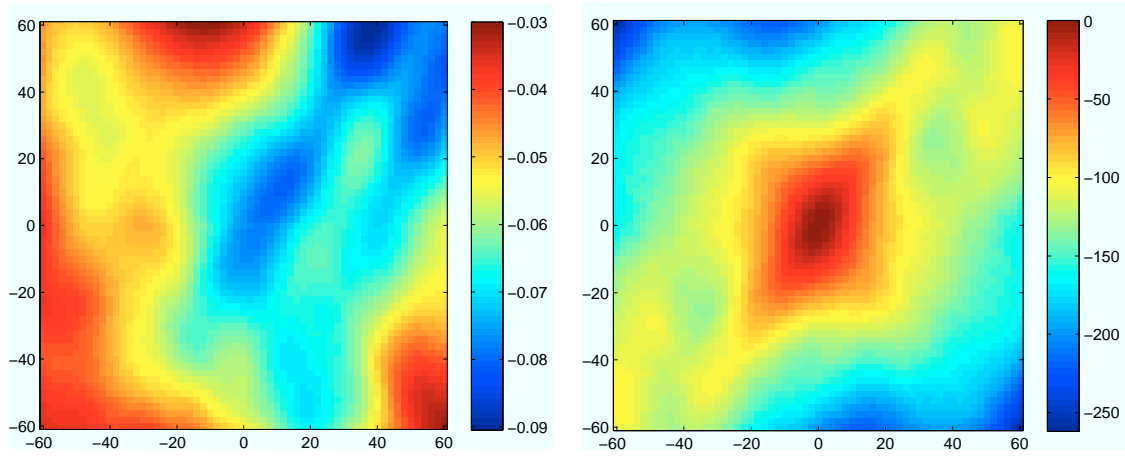
**Figure C.12:** 2009-05-18 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.

APPENDIX C. SUPPLEMENTARY RESULTS

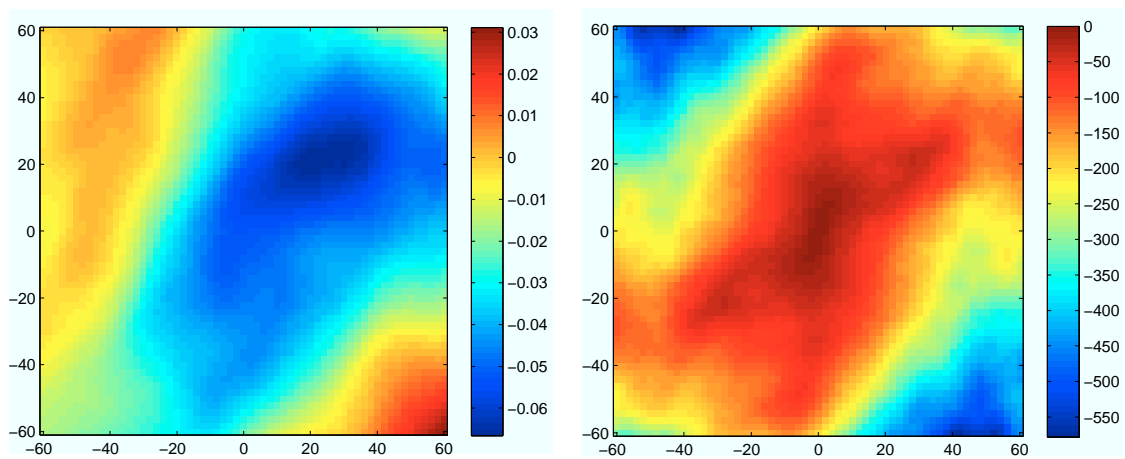


**Figure C.13:** 2010-06-15 (left) First four MAFs stretched between mean  $\pm$  three standard deviations. (right) Associated correlations.

## C.4 Correspondence surfaces

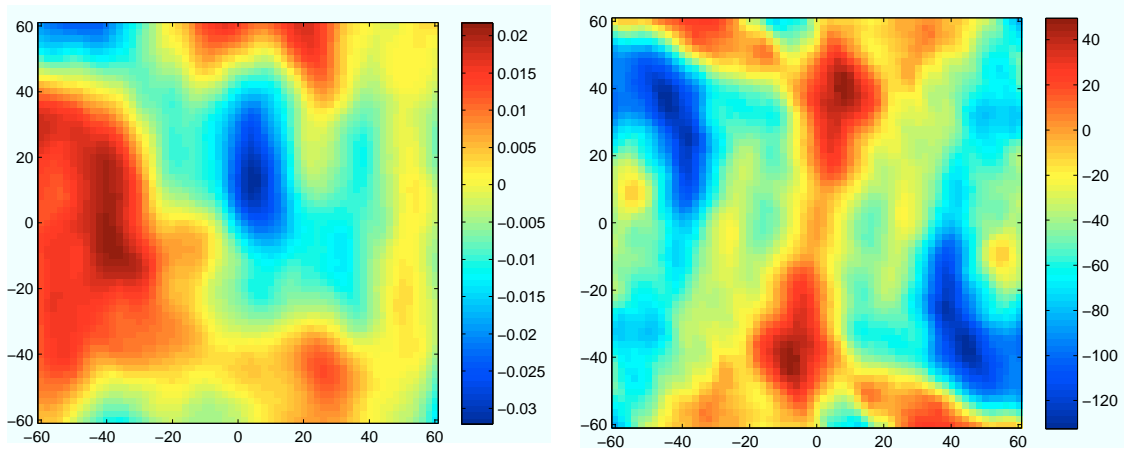


**Figure C.14:** 2007-07-16. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .

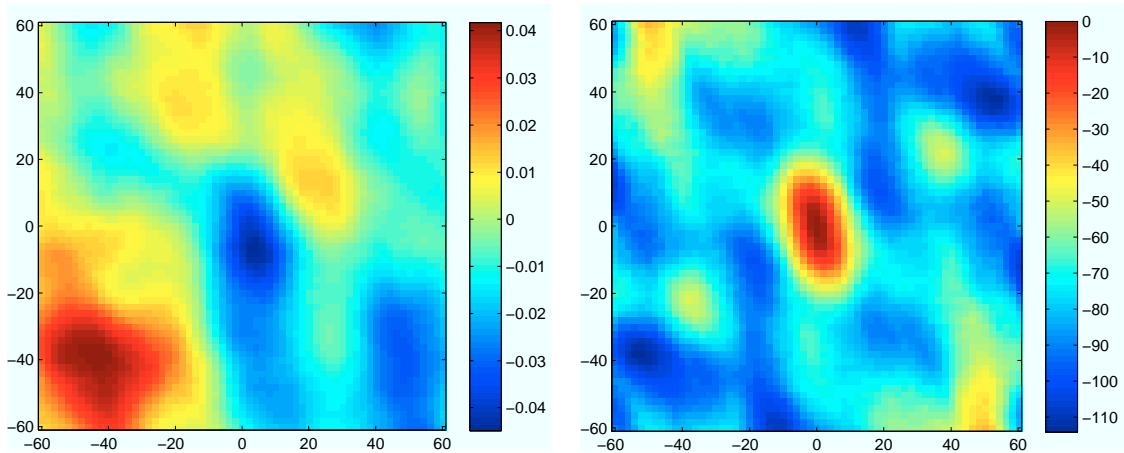


**Figure C.15:** 2007-08-11. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .

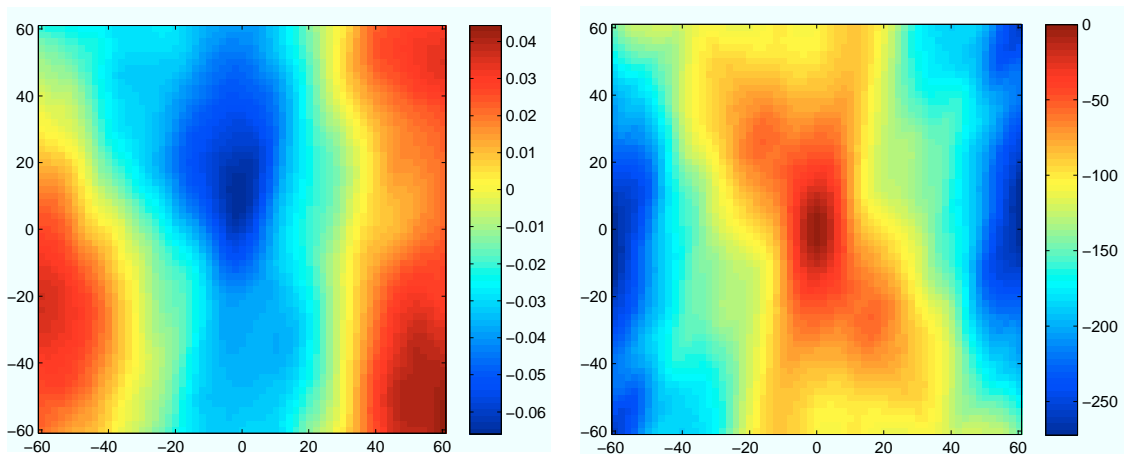
APPENDIX C. SUPPLEMENTARY RESULTS



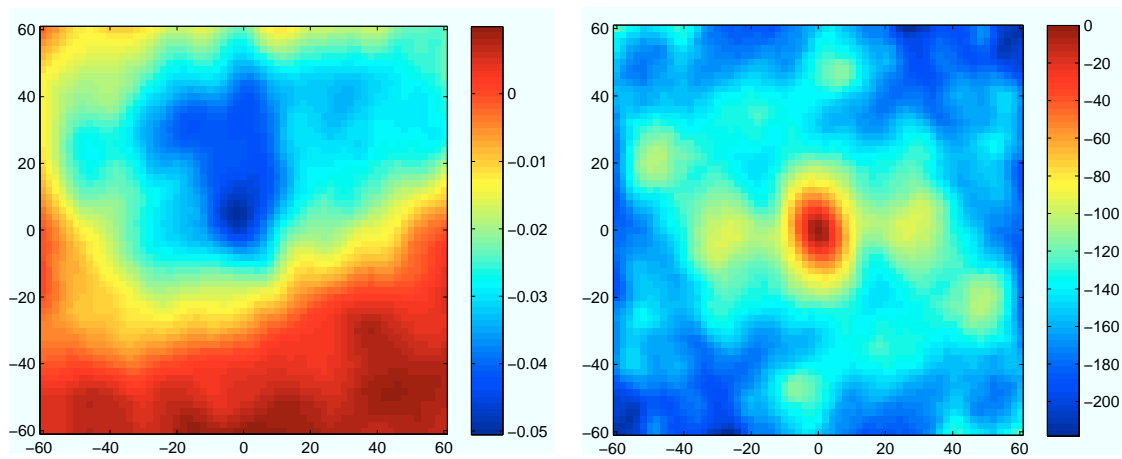
**Figure C.16:** 2007-08-20. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .



**Figure C.17:** 2007-11-11. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .

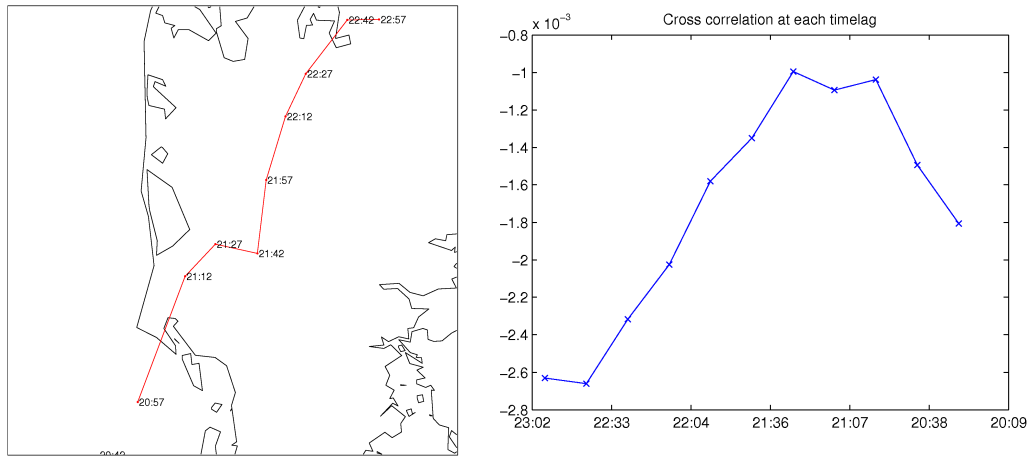


**Figure C.18:** 2009-05-18. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .

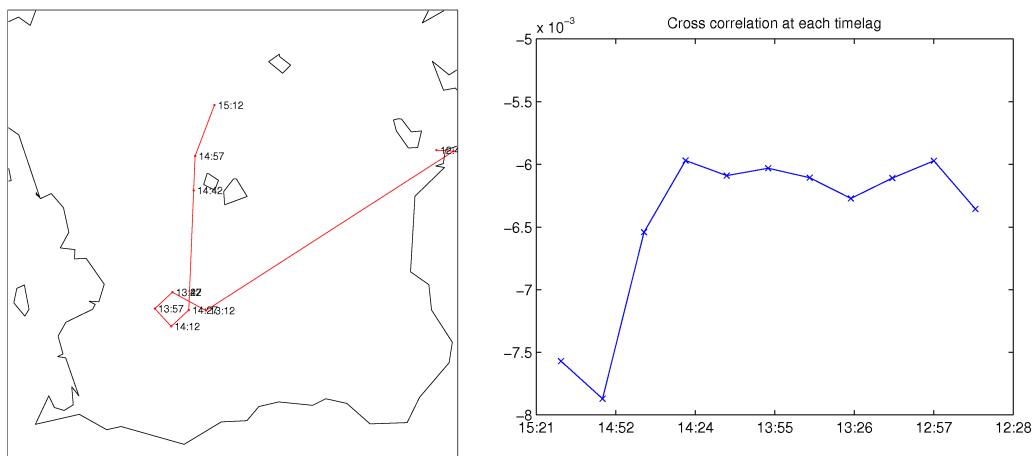


**Figure C.19:** 2010-06-15. (left) Cross correlation surface and (right) cross variogram surface for  $\Delta x, \Delta y \in [-60, 60]$ .

### C.5 Exhaustive search for ground truth



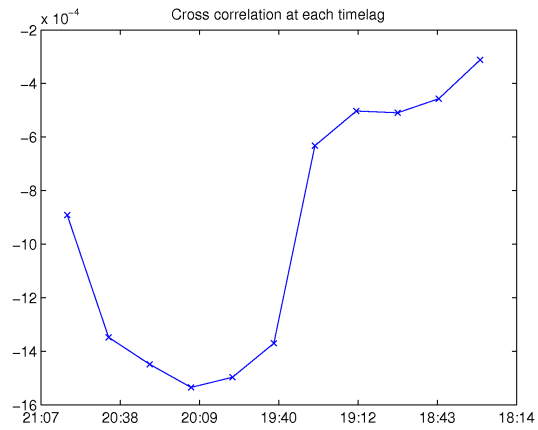
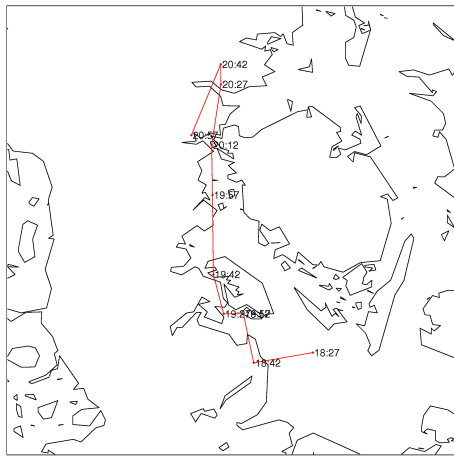
**Figure C.20:** 2007-07-16. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.



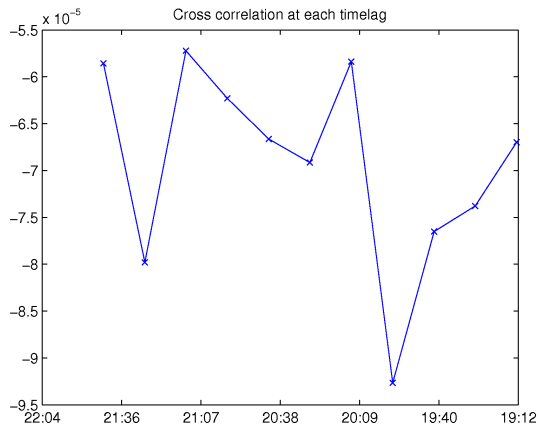
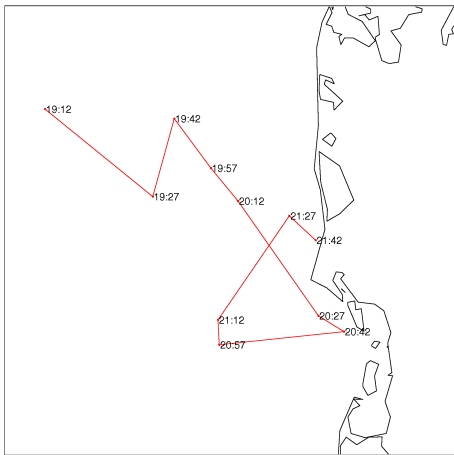
**Figure C.21:** 2007-08-11. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.



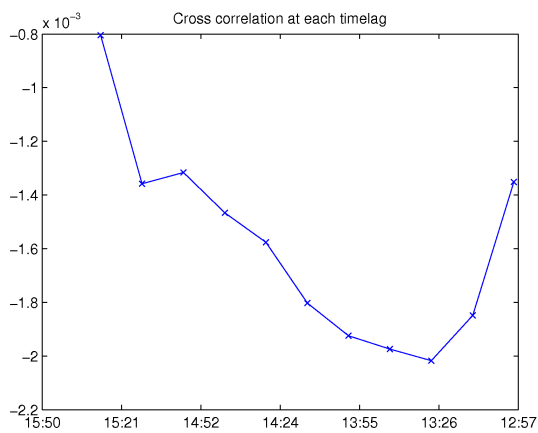
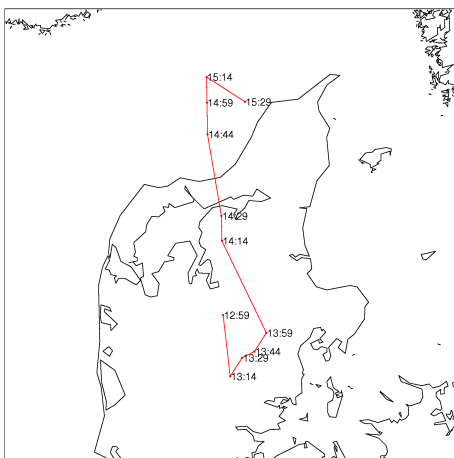
## C.5 Exhaustive search for ground truth



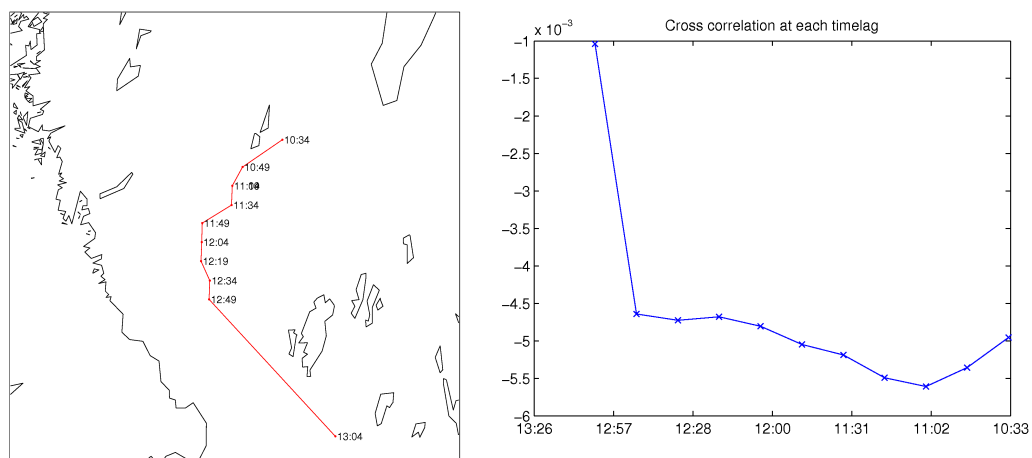
**Figure C.22:** 2007-08-20. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.



**Figure C.23:** 2007-11-11. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.

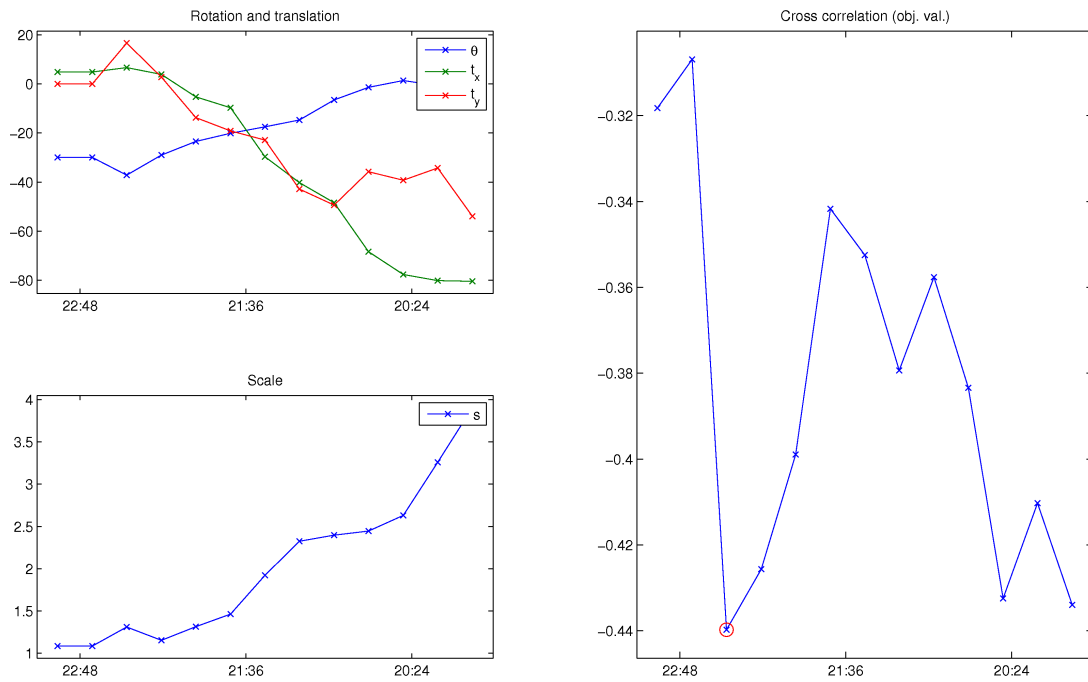


**Figure C.24:** 2009-05-18. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.

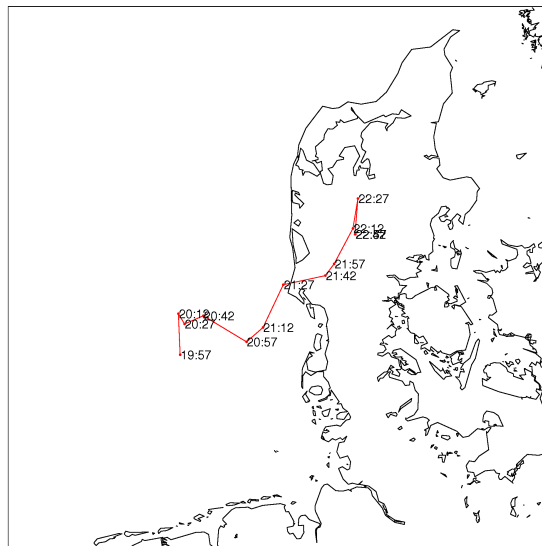


**Figure C.25:** 2010-06-15. (left) Spatial steps taken for minimum cross correlation in each time step. (right) Minimum cross correlation in each time step.

## C.6 Invariant tracking to collect ground truth

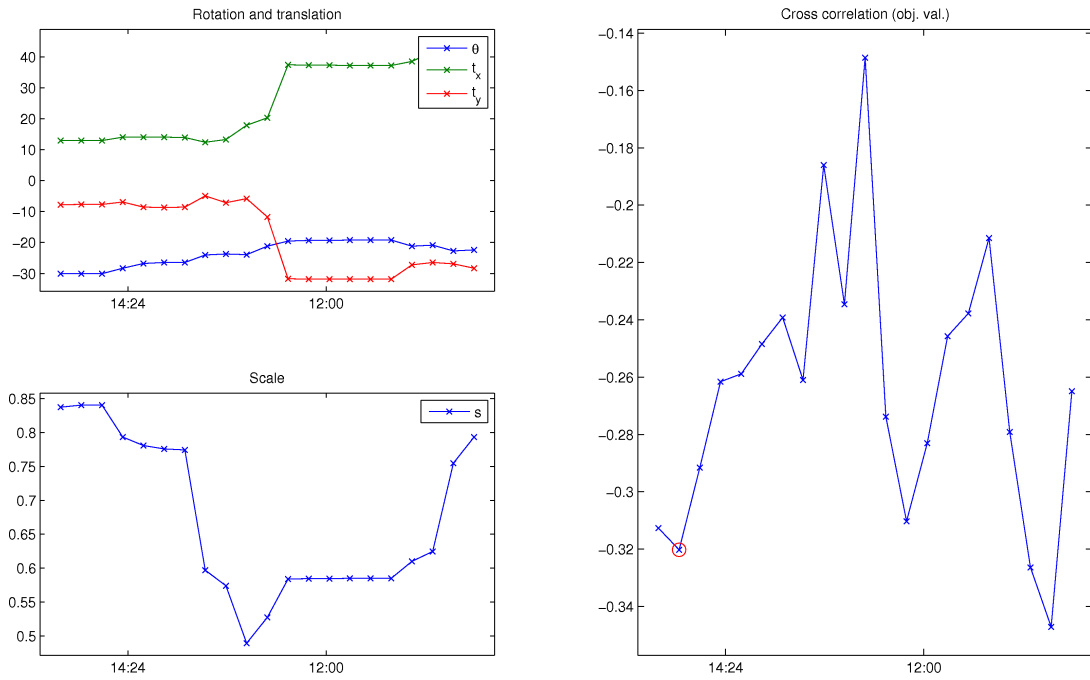


**Figure C.26:** 2007-07-16. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.

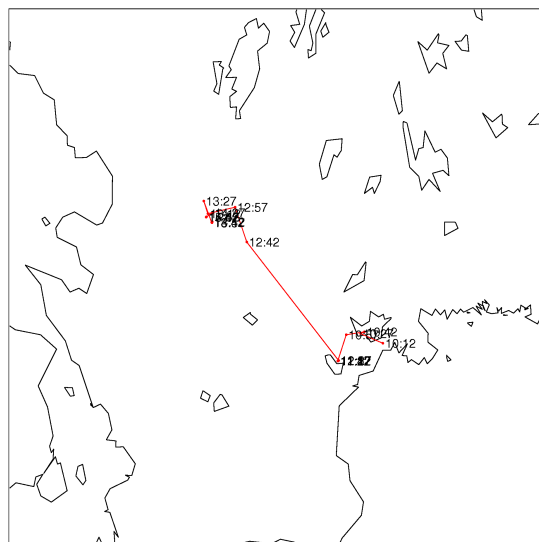


**Figure C.27:** 2007-07-16. Tracked path back in time using method with invariance to scale, rotation and translation.

APPENDIX C. SUPPLEMENTARY RESULTS

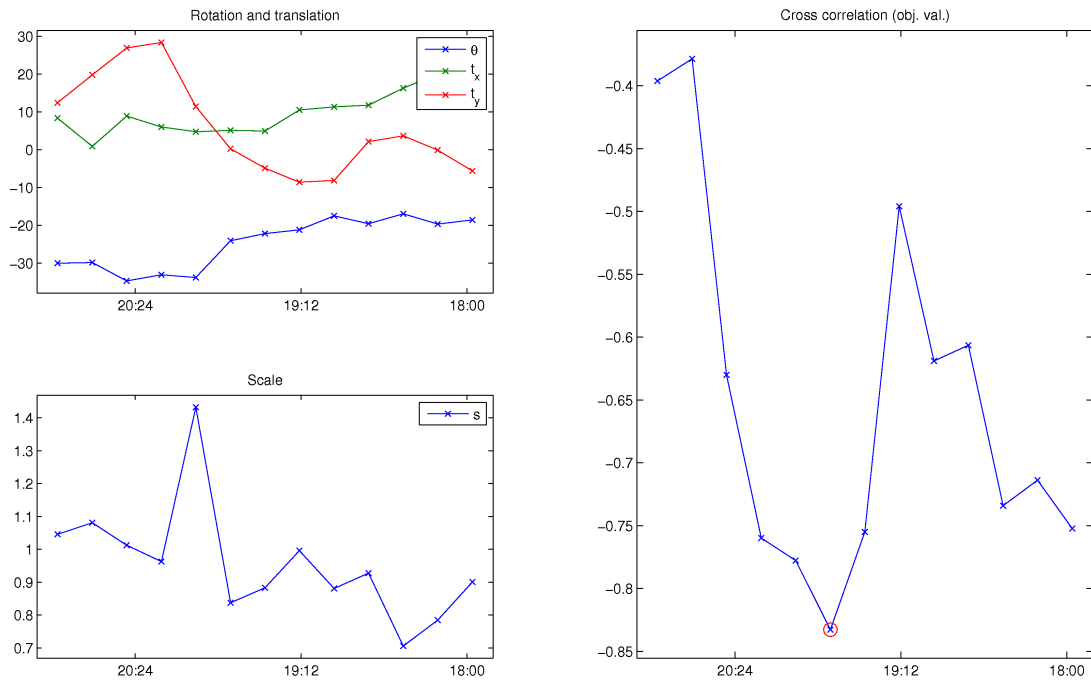


**Figure C.28:** 2007-08-11. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.

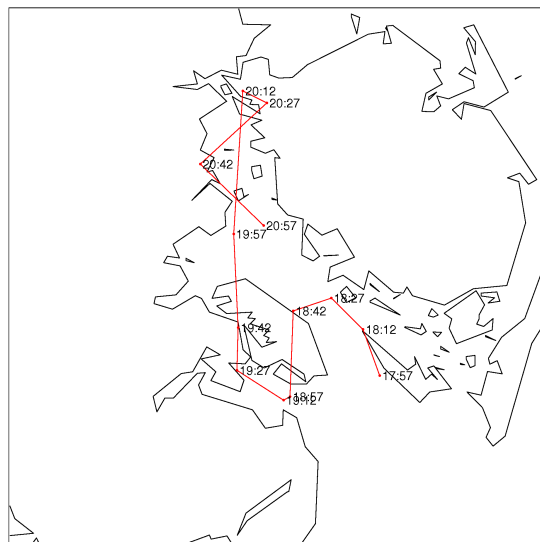


**Figure C.29:** 2007-08-11. Tracked path back in time using method with invariance to scale, rotation and translation.

## C.6 Invariant tracking to collect ground truth

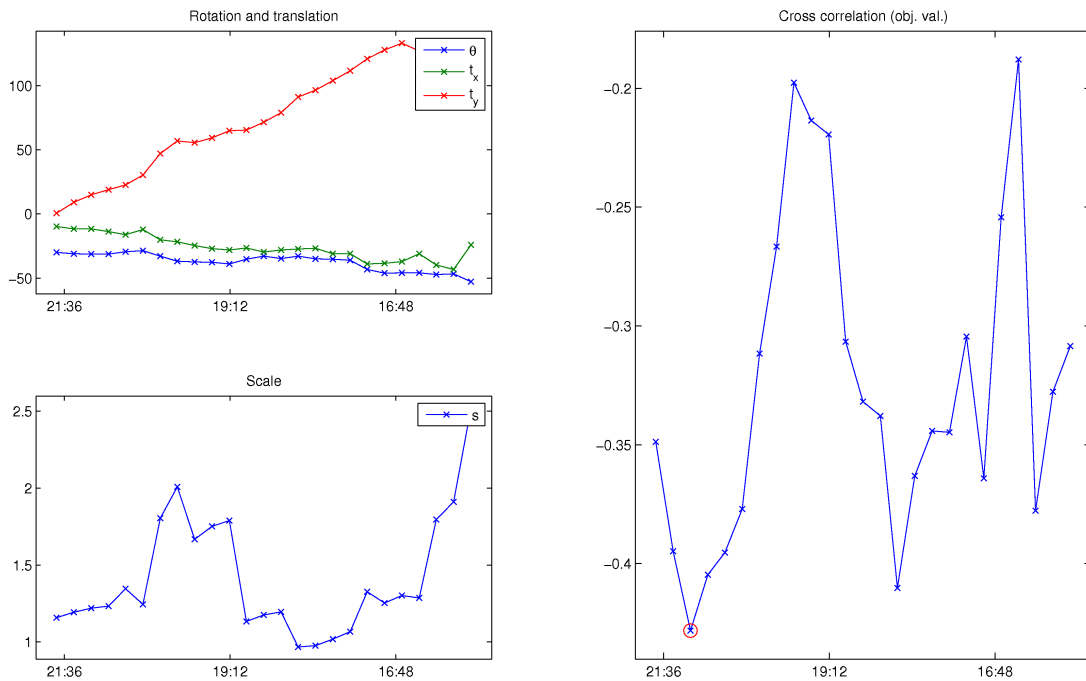


**Figure C.30:** 2007-08-20. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.

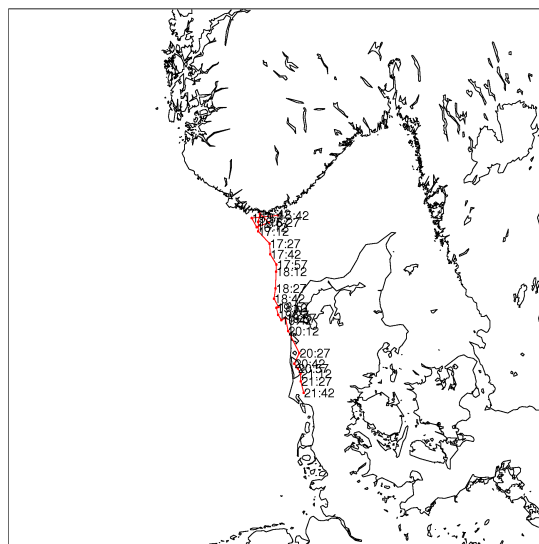


**Figure C.31:** 2007-08-20. Tracked path back in time using method with invariance to scale, rotation and translation.

APPENDIX C. SUPPLEMENTARY RESULTS

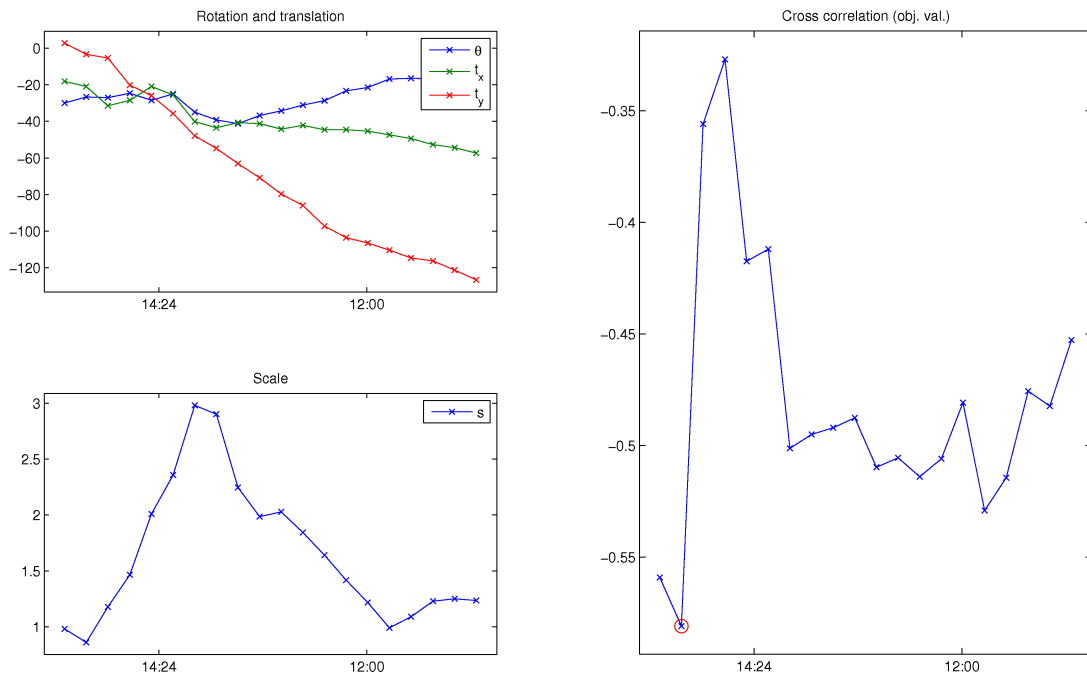


**Figure C.32:** 2007-11-11. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.

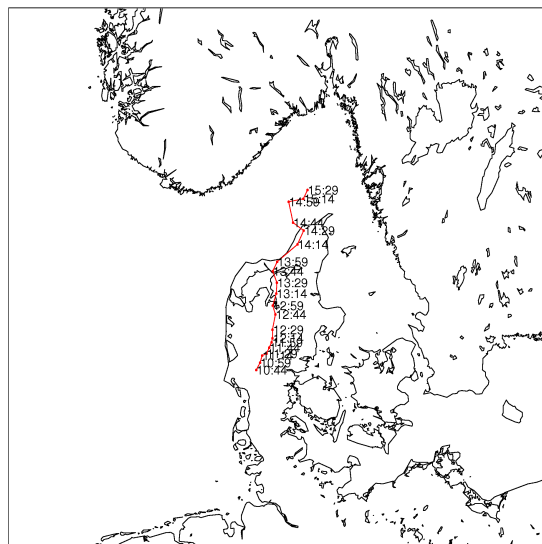


**Figure C.33:** 2007-11-11. Tracked path back in time using method with invariance to scale, rotation and translation.

## C.6 Invariant tracking to collect ground truth

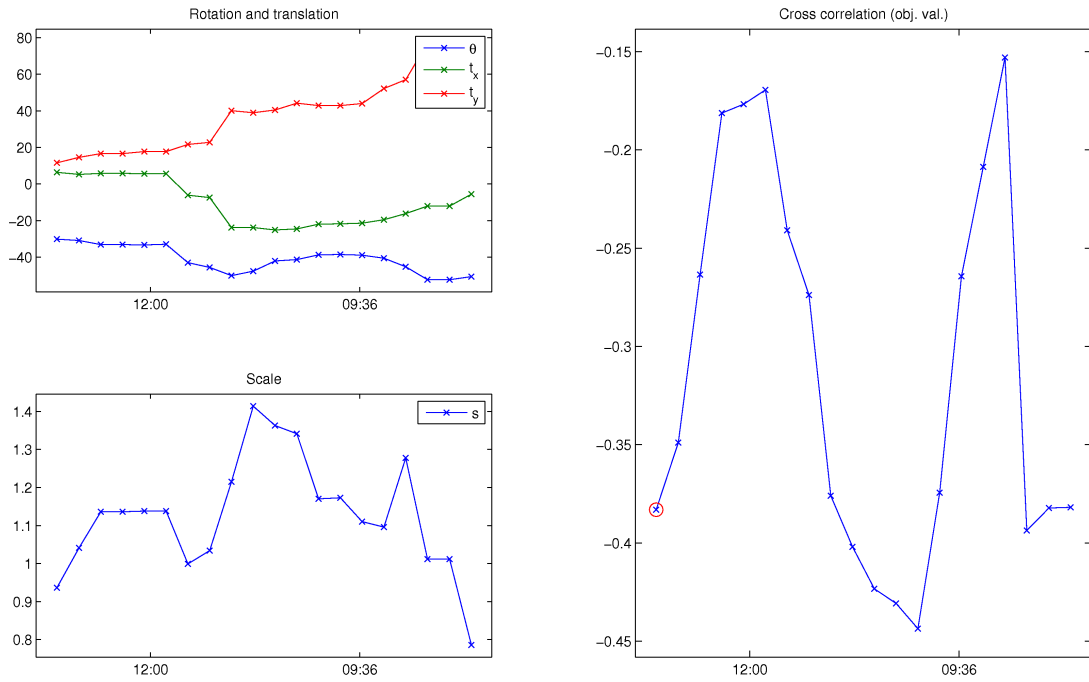


**Figure C.34:** 2009-05-18. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.

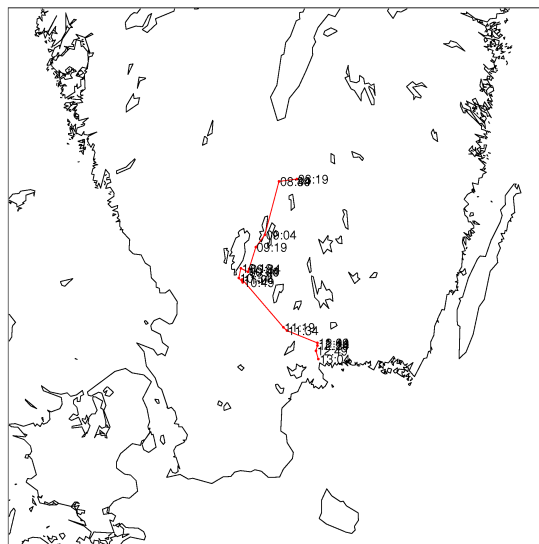


**Figure C.35:** 2009-05-18. Tracked path back in time using method with invariance to scale, rotation and translation.

APPENDIX C. SUPPLEMENTARY RESULTS



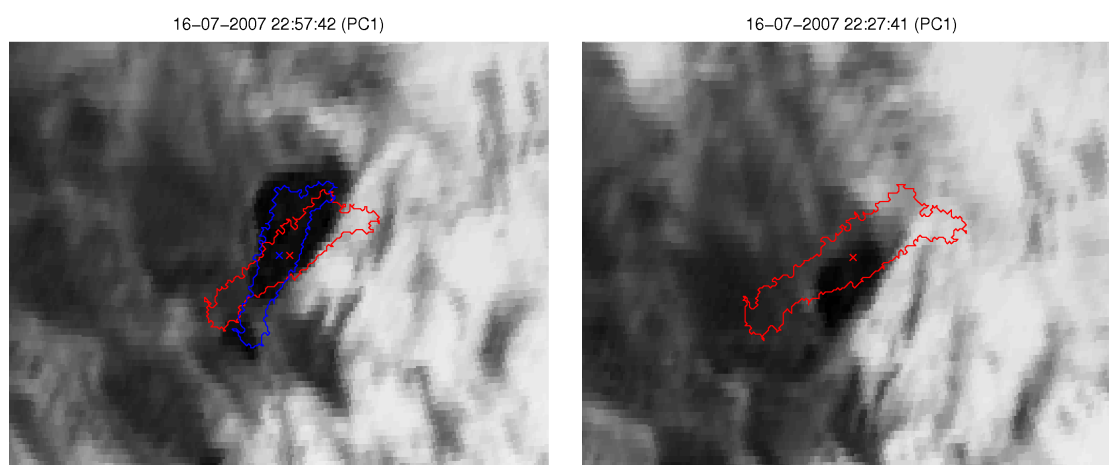
**Figure C.36:** 2010-06-15. (left) Parameters chosen in each time lag using method with invariance to scale, rotation and translation. (right) Objective value.



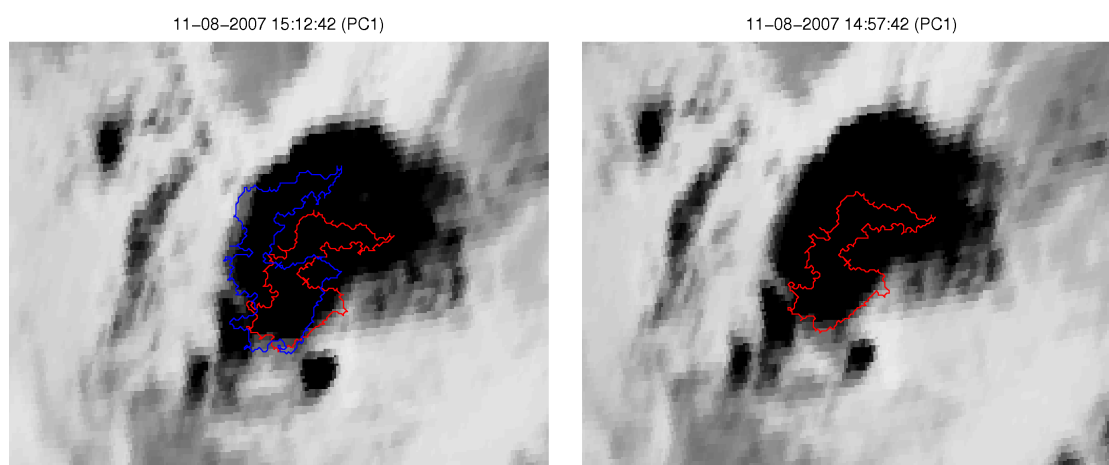
**Figure C.37:** 2010-06-15. Tracked path back in time using method with invariance to scale, rotation and translation.



## C.7 Transformed radar data at time of minimum cross correlation

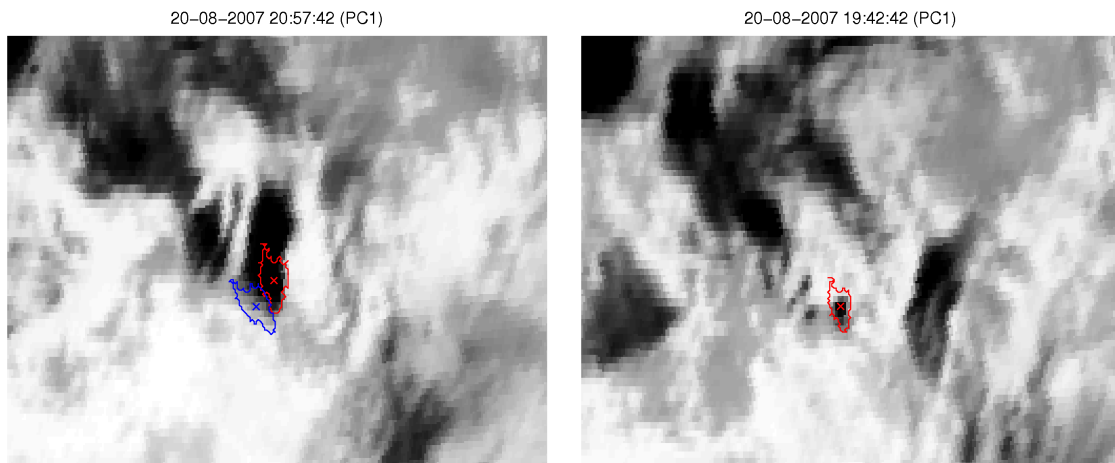


**Figure C.38:** 2007-07-16. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.

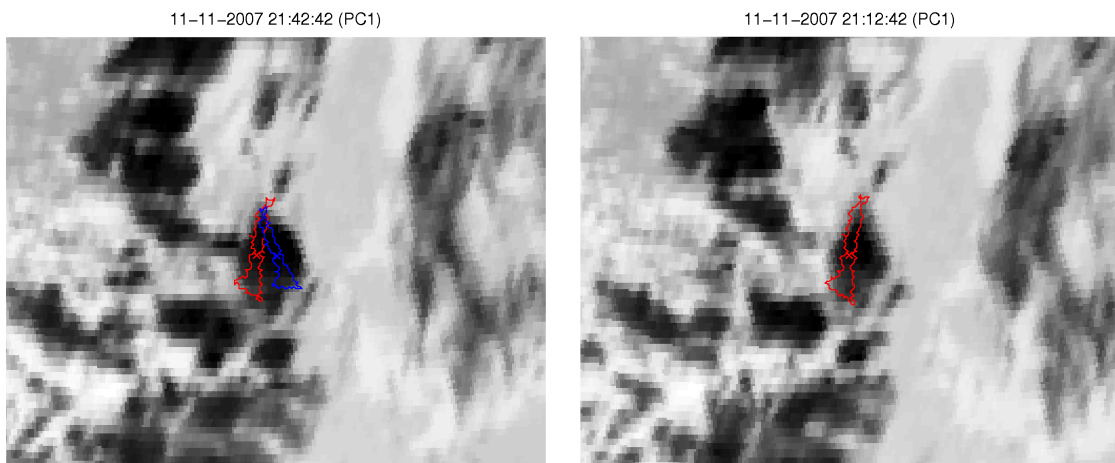


**Figure C.39:** 2007-08-11. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.

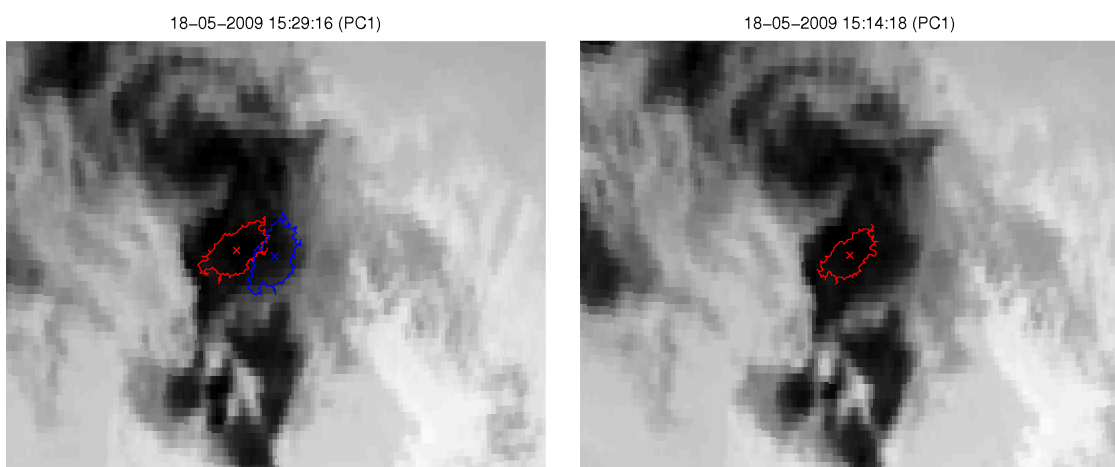
APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.40:** 2007-08-20. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.

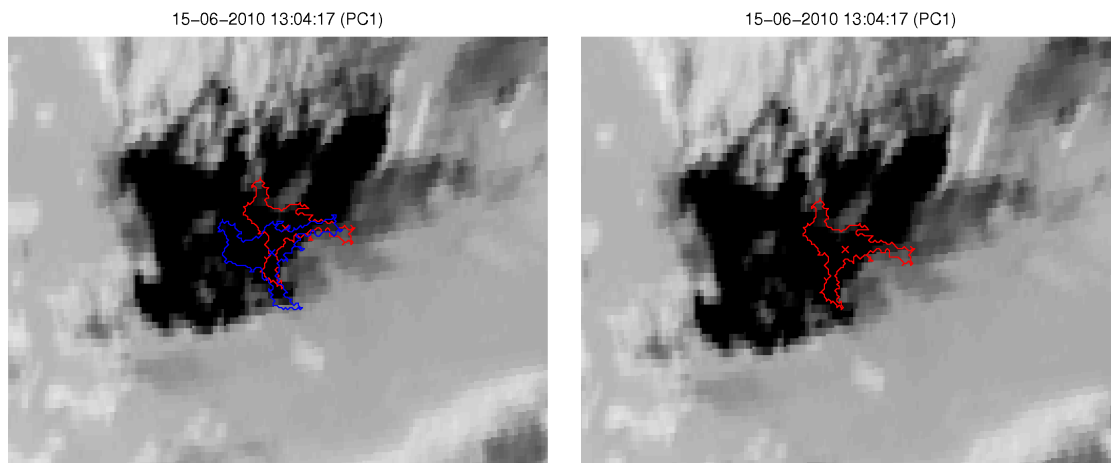


**Figure C.41:** 2007-11-11. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.



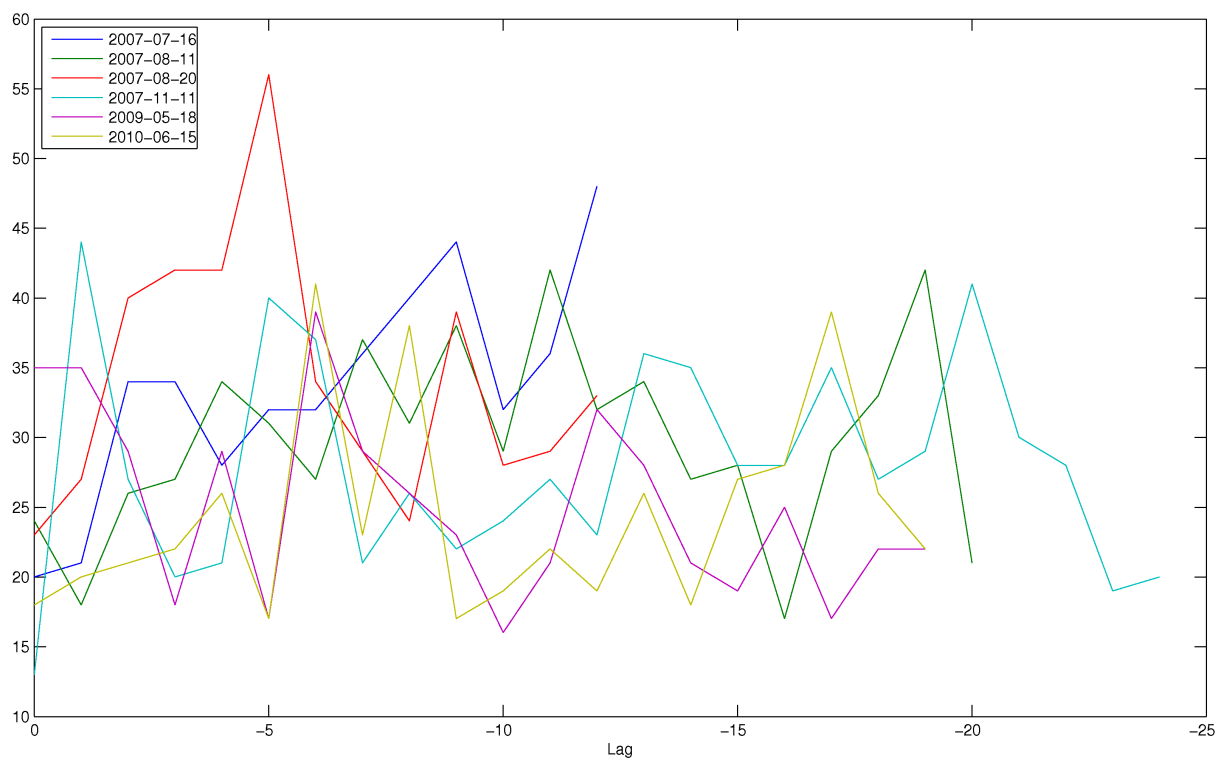
**Figure C.42:** 2009-05-18. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.

## C.7 Transformed radar data at time of minimum cross correlation

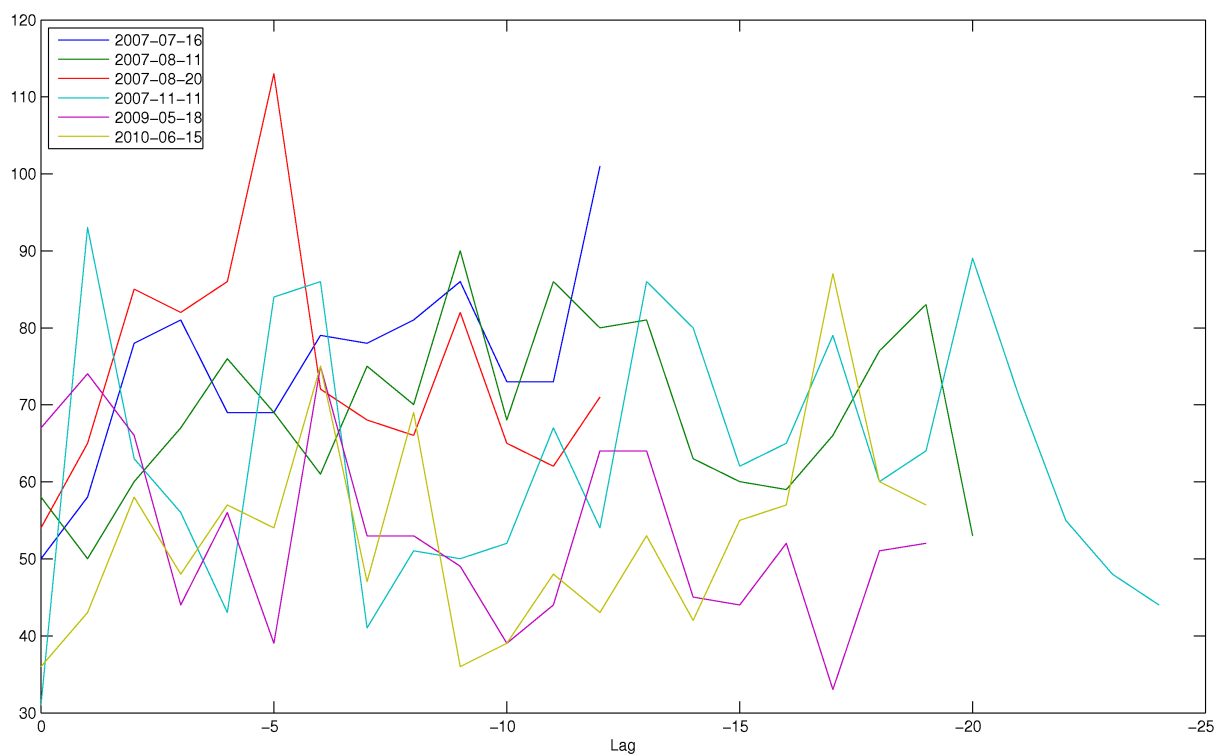


**Figure C.43:** 2010-06-15. PC1 of satellite data with boundary of original radar data in blue and transformed radar data in red for (left) first time lag and (right) chosen lag of minimum cross correlation.

### C.8 Simplex method statistics

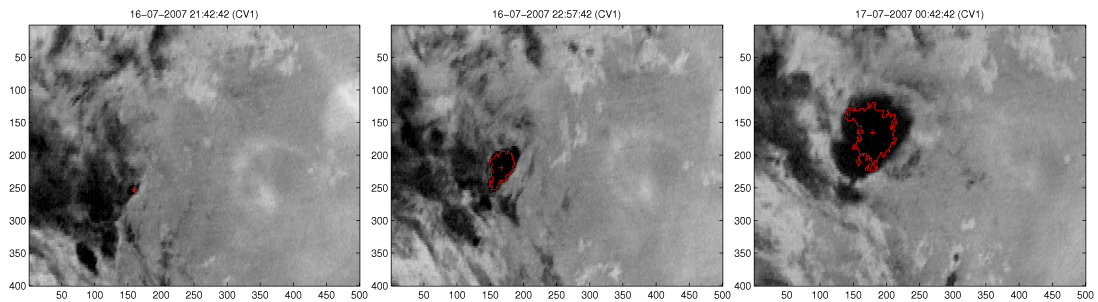


**Figure C.44:** *Number of iterations for the simplex method to reach minimum cross correlation in each time lag.*

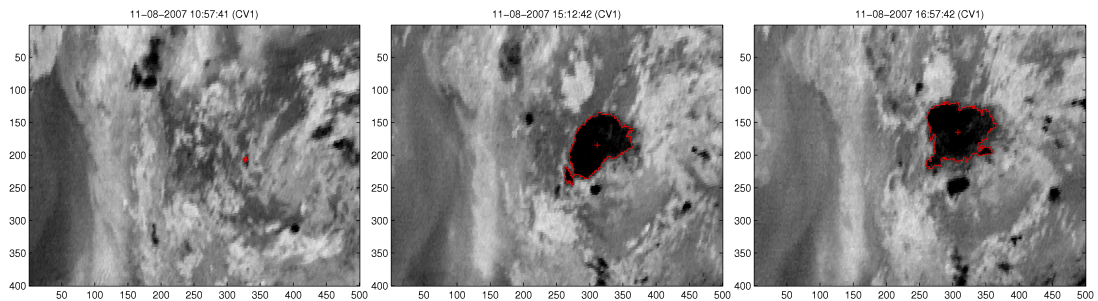


**Figure C.45:** *Number of function evaluations performed by the simplex method to reach minimum cross correlation in each time lag.*

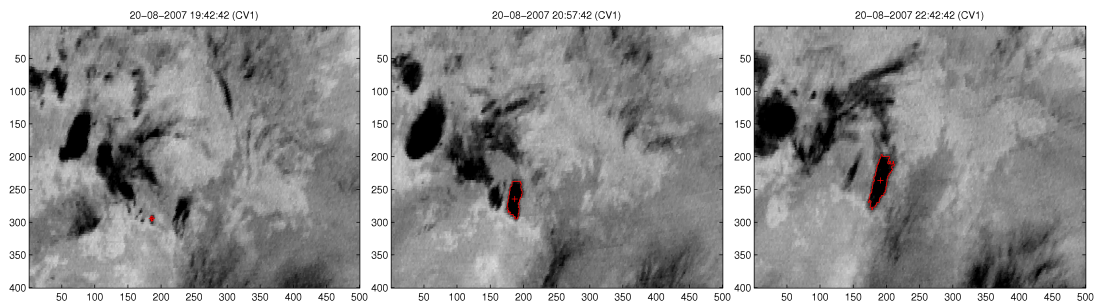
## C.9 Simple tracking results



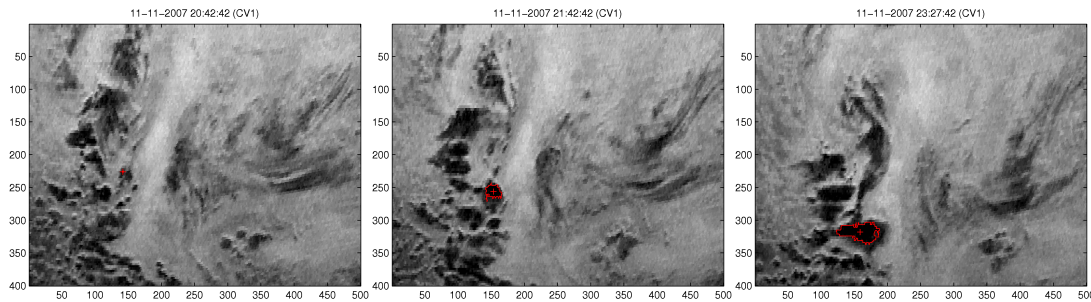
**Figure C.46:** 2007-07-16. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.



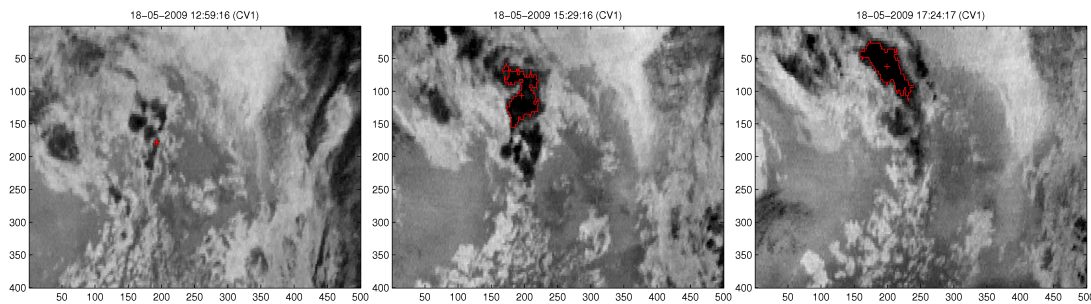
**Figure C.47:** 2007-08-11. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.



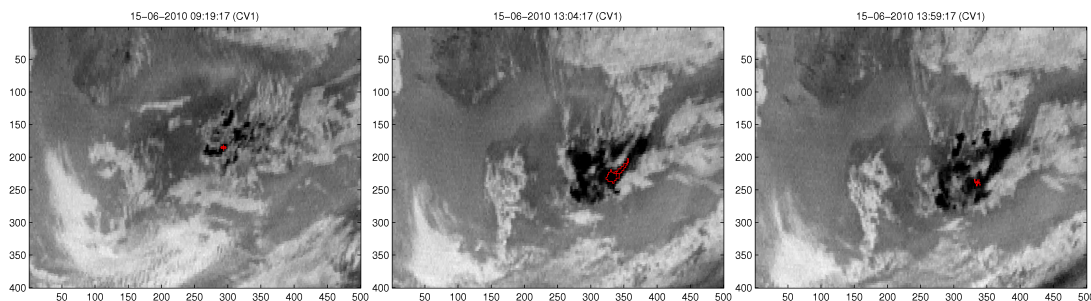
**Figure C.48:** 2007-08-20. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.



**Figure C.49:** 2007-11-11. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.

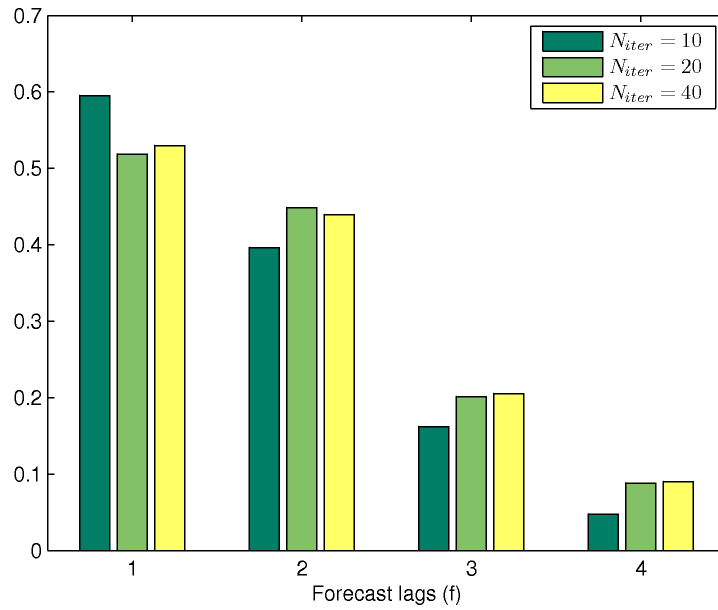


**Figure C.50:** 2009-05-18. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.

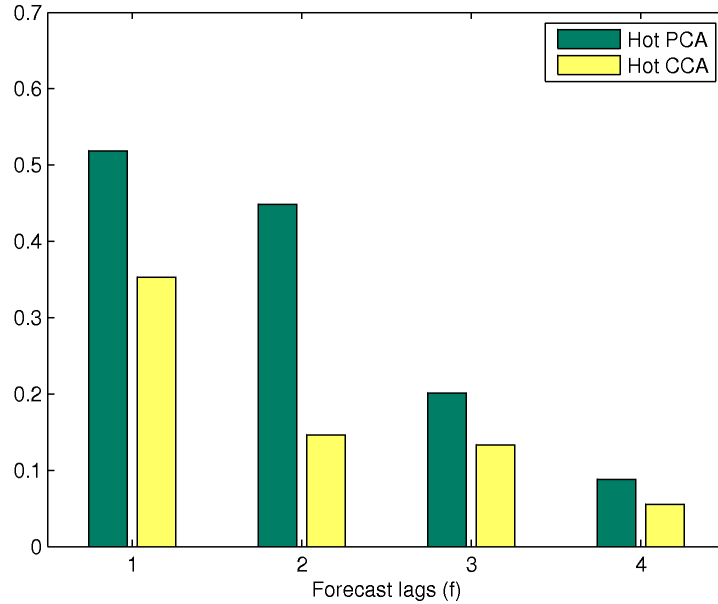


**Figure C.51:** 2010-06-15. First CV with identified convective cloud border shown in red. From left to right, the points in time chosen to display are (i) the earliest point where the cloud has been identified, (ii) the time of the hotspot and (iii) two hours after the hotspot.

### C.10 Dictionary parameters

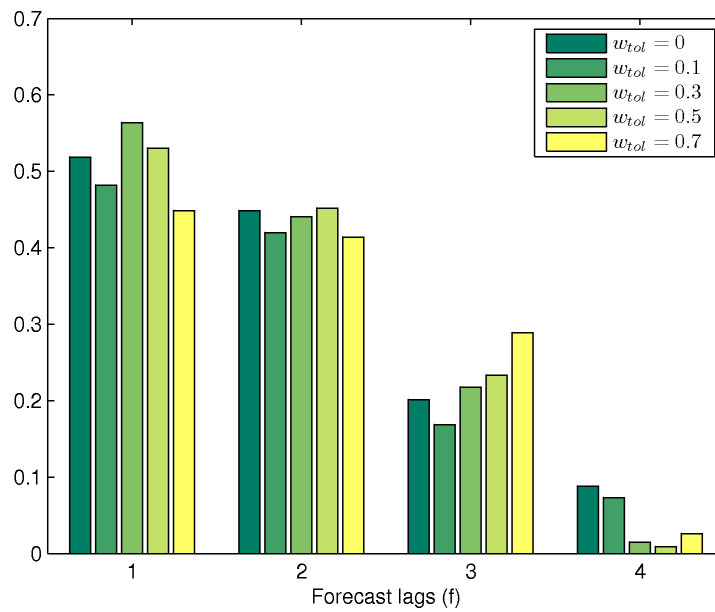


**Figure C.52:** Comparison of sensitivities for varying forecast lengths and three different numbers of iterations  $N_{iter}$  used when building the dictionary.  $N_{iter} = 40$  is chosen as it has approximately the same performance as the larger number of iterations.



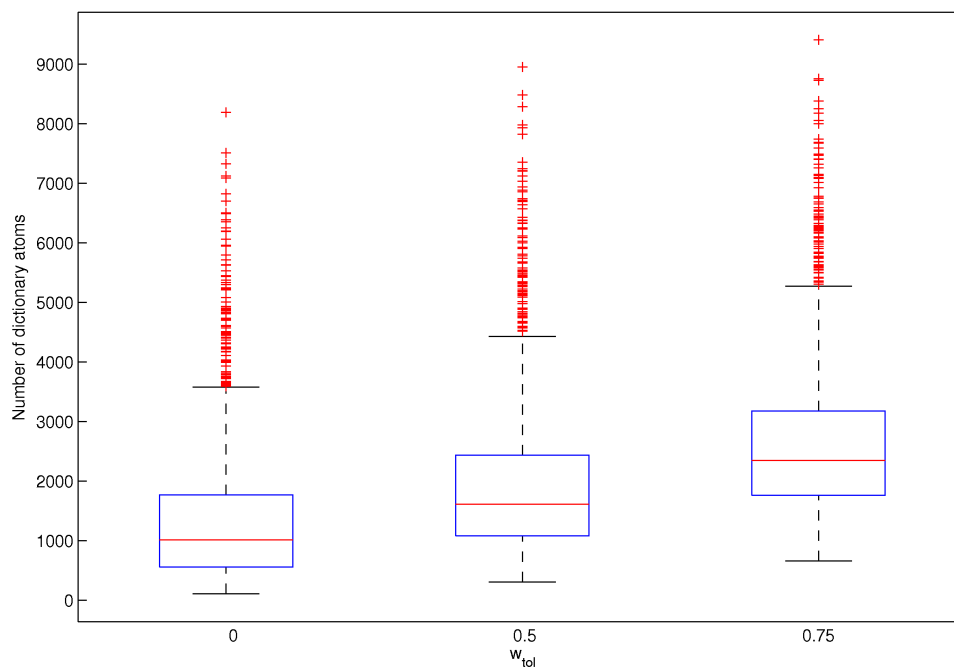
**Figure C.53:** Comparison of sensitivities for varying forecast lengths and choice of subspace. The first component from the “Hot PCA” described in Section 3.2.2 performs better than the single canonical variate from Section 4.3.1.





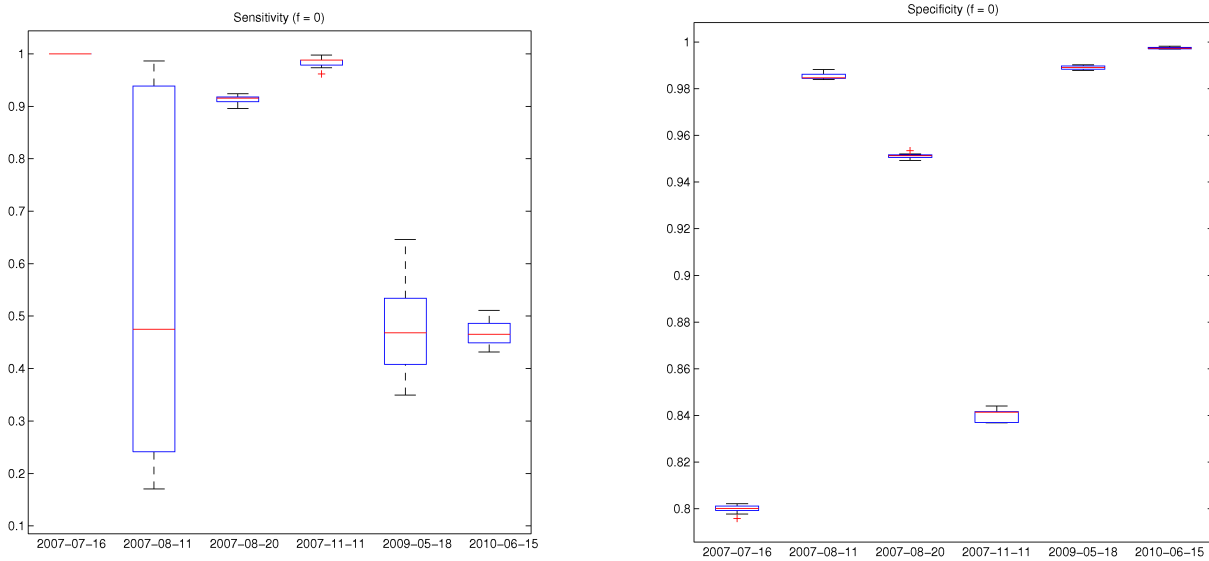
**Figure C.54:** Comparison of sensitivities for varying forecast lengths and choice of dissimilarity tolerance  $w_{tol}$ . The standard choice of  $w_{tol} = 0$  provides the best performance.

### C.11 Dictionary size

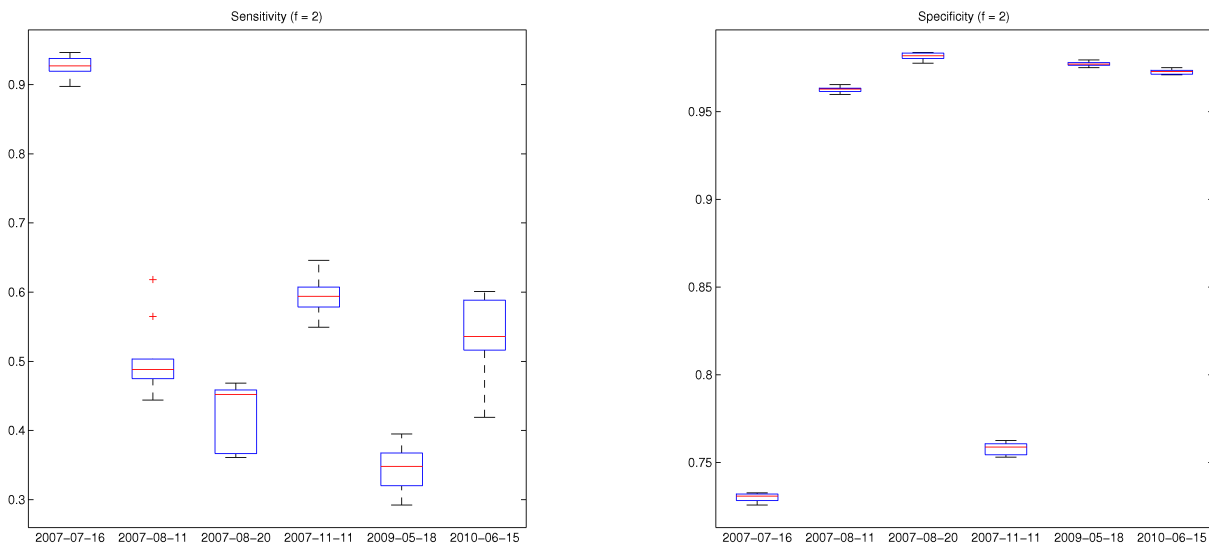


**Figure C.55:** *The variation in number of dictionary elements for each value of  $w_{tot}$*

## C.12 Box plots for dictionary method results

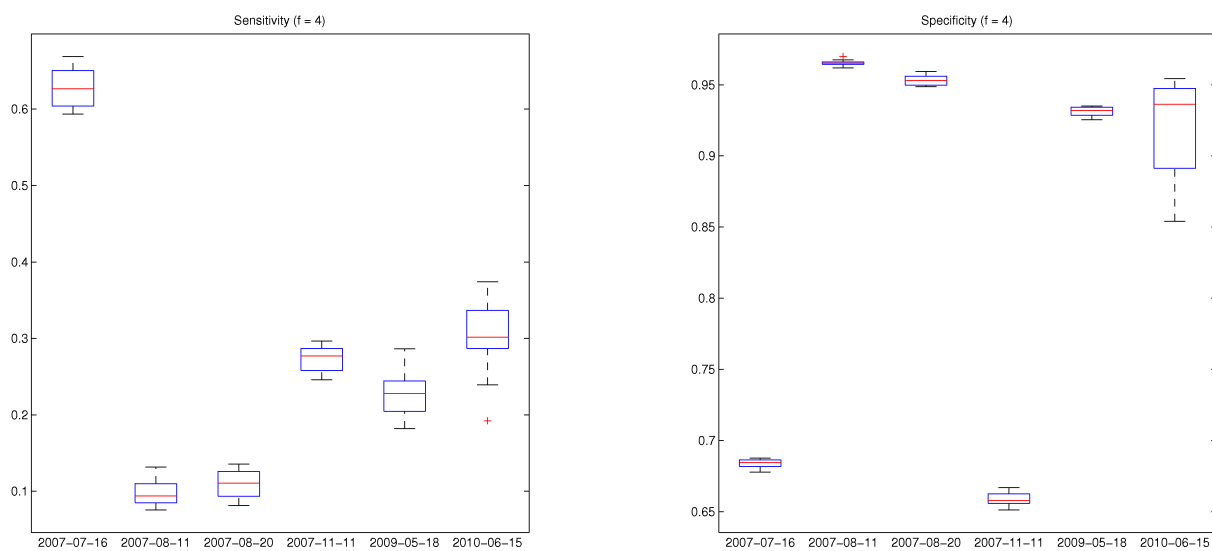


**Figure C.56:** Variations in sensitivity and specificity using parameters from Table 5.2 and 10 repetitions of dictionary building and segmentation. Forecast length  $f = 0$ .



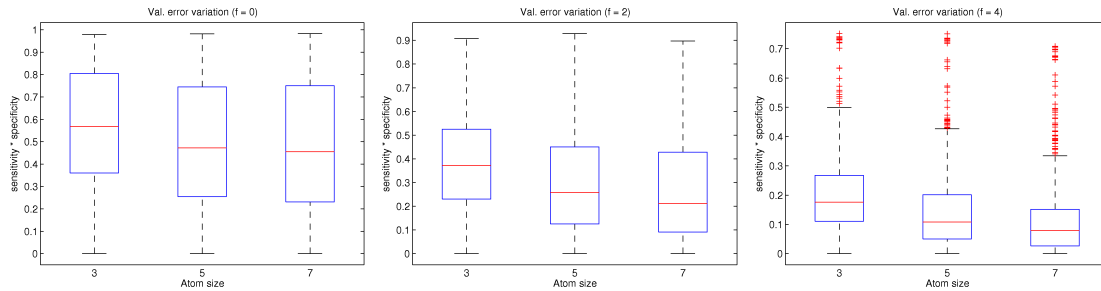
**Figure C.57:** Variations in sensitivity and specificity using parameters from Table 5.2 and 10 repetitions of dictionary building and segmentation. Forecast length  $f = 2$ .

## APPENDIX C. SUPPLEMENTARY RESULTS

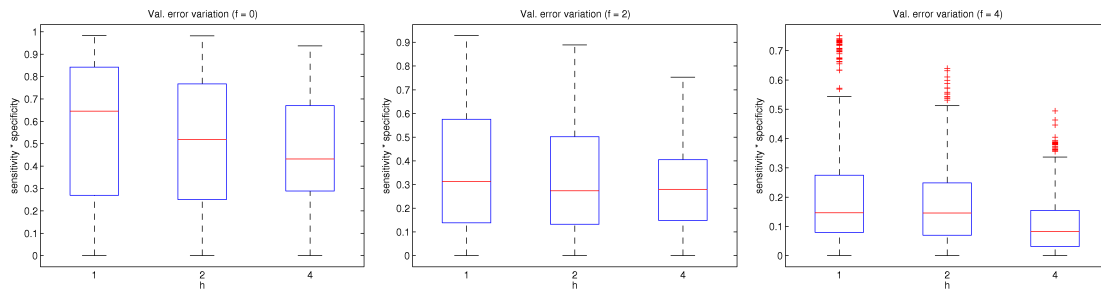


**Figure C.58:** Variations in sensitivity and specificity using parameters from Table 5.2 and 10 repetitions of dictionary building and segmentation. Forecast length  $f = 4$ .

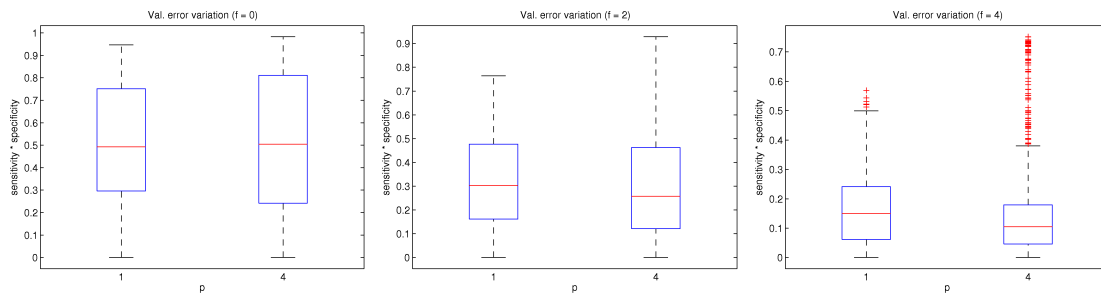
## C.13 Validation error box plots for dictionary method



**Figure C.59:** Box plots of variations in validation errors for each value of the atom size  $\sqrt{n}$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3 \cdot 2 \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.

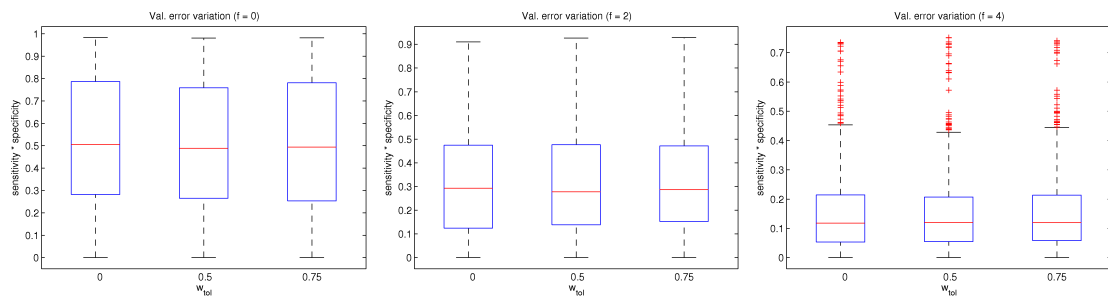


**Figure C.60:** Box plots of variations in validation errors for each value of the number of lags  $h$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3 \cdot 2 \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.



**Figure C.61:** Box plots of variations in validation errors for each value of the number of spectral components  $p$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3 \cdot 3 \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.

## APPENDIX C. SUPPLEMENTARY RESULTS

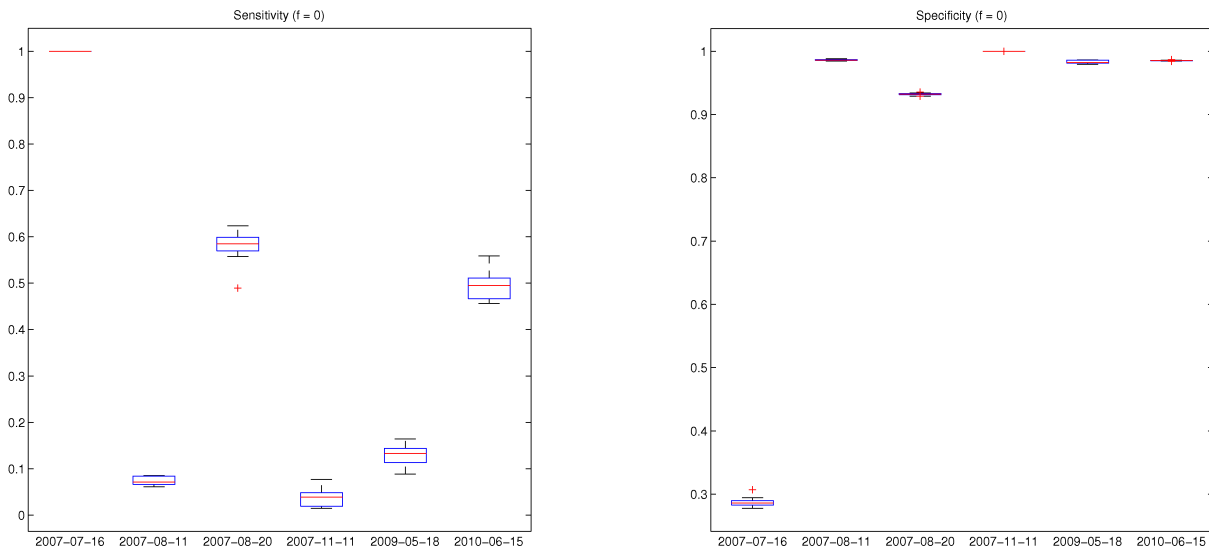


**Figure C.62:** Box plots of variations in validation errors for each value of the similarity threshold  $w_{tol}$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3 \cdot 2 \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.

## C.14 Test errors for dictionary method using global CCA

| Scenario   | $\sqrt{n}$ | $h$ | $p$ | $f = 0$   |       |      |
|------------|------------|-----|-----|-----------|-------|------|
|            |            |     |     | $w_{tol}$ | SE    | SP   |
| 2007-07-16 | 3          | 1   | 1   | 0         | 1     | 0.29 |
| 2007-08-11 | 3          | 4   | 1   | 0         | 0.074 | 0.99 |
| 2007-08-20 | 3          | 4   | 1   | 0.75      | 0.58  | 0.93 |
| 2007-11-11 | 3          | 4   | 1   | 0.75      | 0.037 | 1    |
| 2009-05-18 | 3          | 4   | 1   | 0.75      | 0.13  | 0.98 |
| 2010-06-15 | 3          | 4   | 1   | 0         | 0.5   | 0.99 |
| Average    |            |     |     |           | 0.39  | 0.86 |

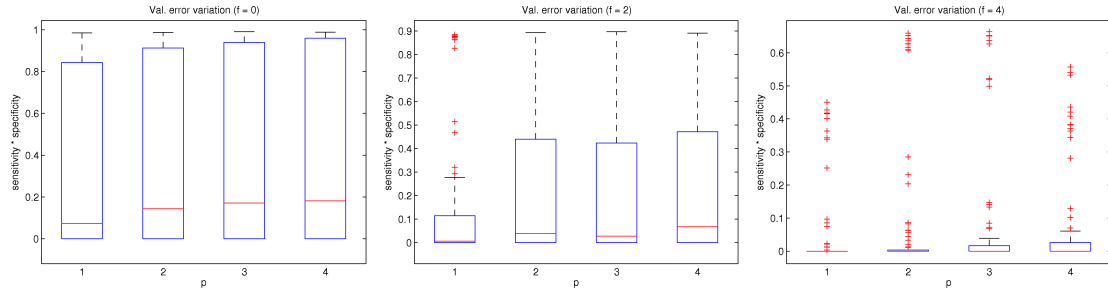
**Table C.1:** Parameters chosen using cross validation and obtained sensitivities (SE) and specificities (SP) for each scenario. Subspace used is the global CCA subspace from Section 4.3.1 and only classification  $f = 0$  is considered.



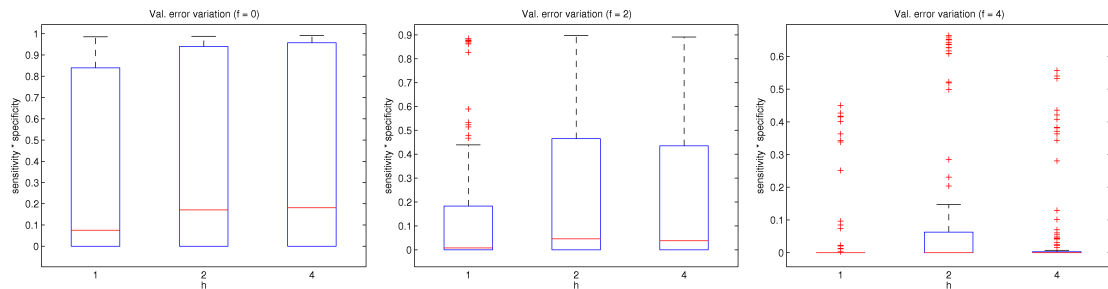
**Figure C.63:** Variations in test errors obtained from cross validation using global CCA subspace.

## C.15 Logistic regression

### C.15.1 Validation error box plots



**Figure C.64:** Box plots of variations in validation errors for each value of  $p$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.

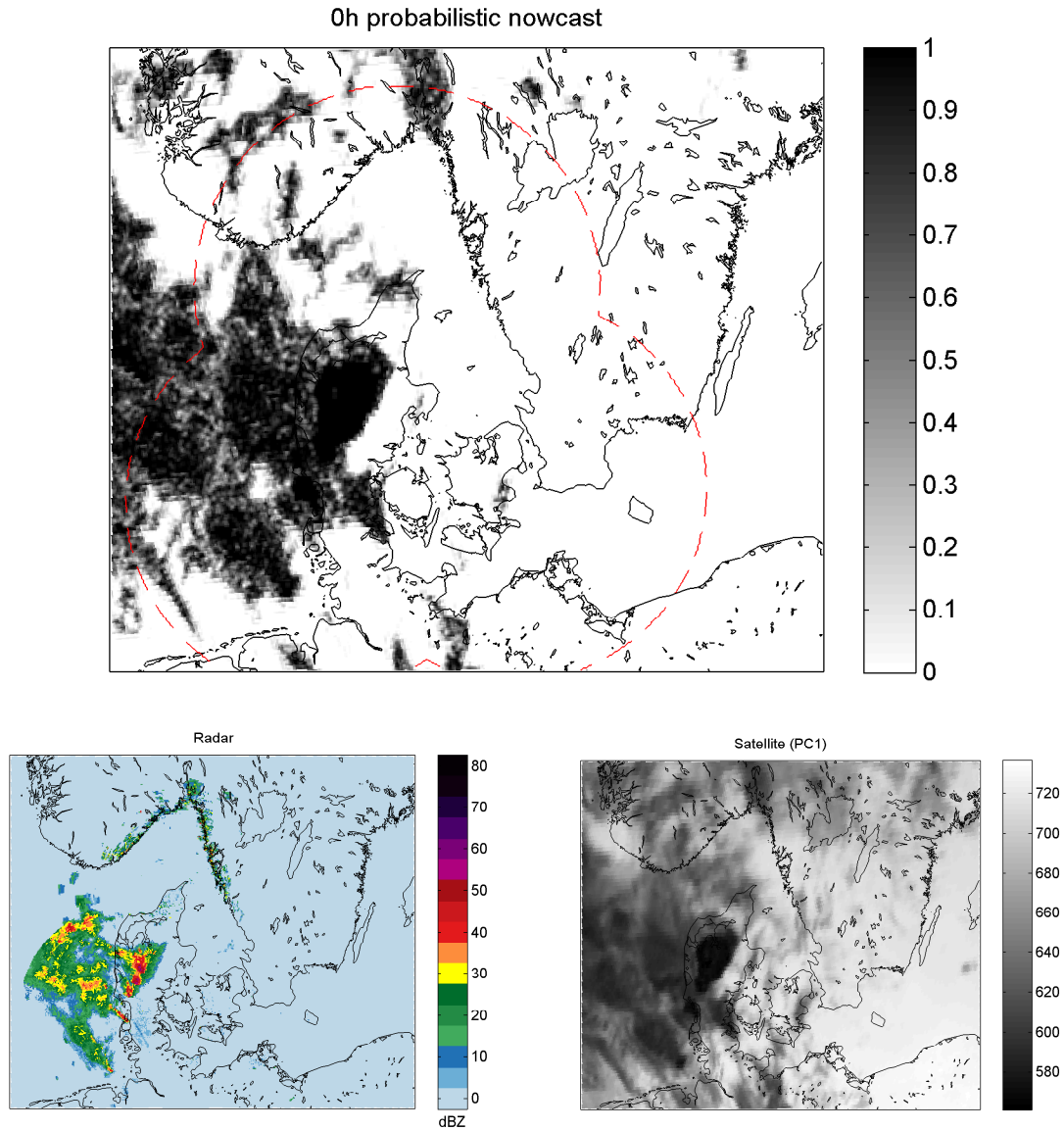


**Figure C.65:** Box plots of variations in validation errors for each value of  $h$  during cross validation. Each box represents  $N_s \cdot (N_s - 1) \cdot 3$  validation errors. Note that the “error” is actually sensitivity  $\cdot$  specificity, wherefore a higher value is better.



## C.16 Nowcast maps

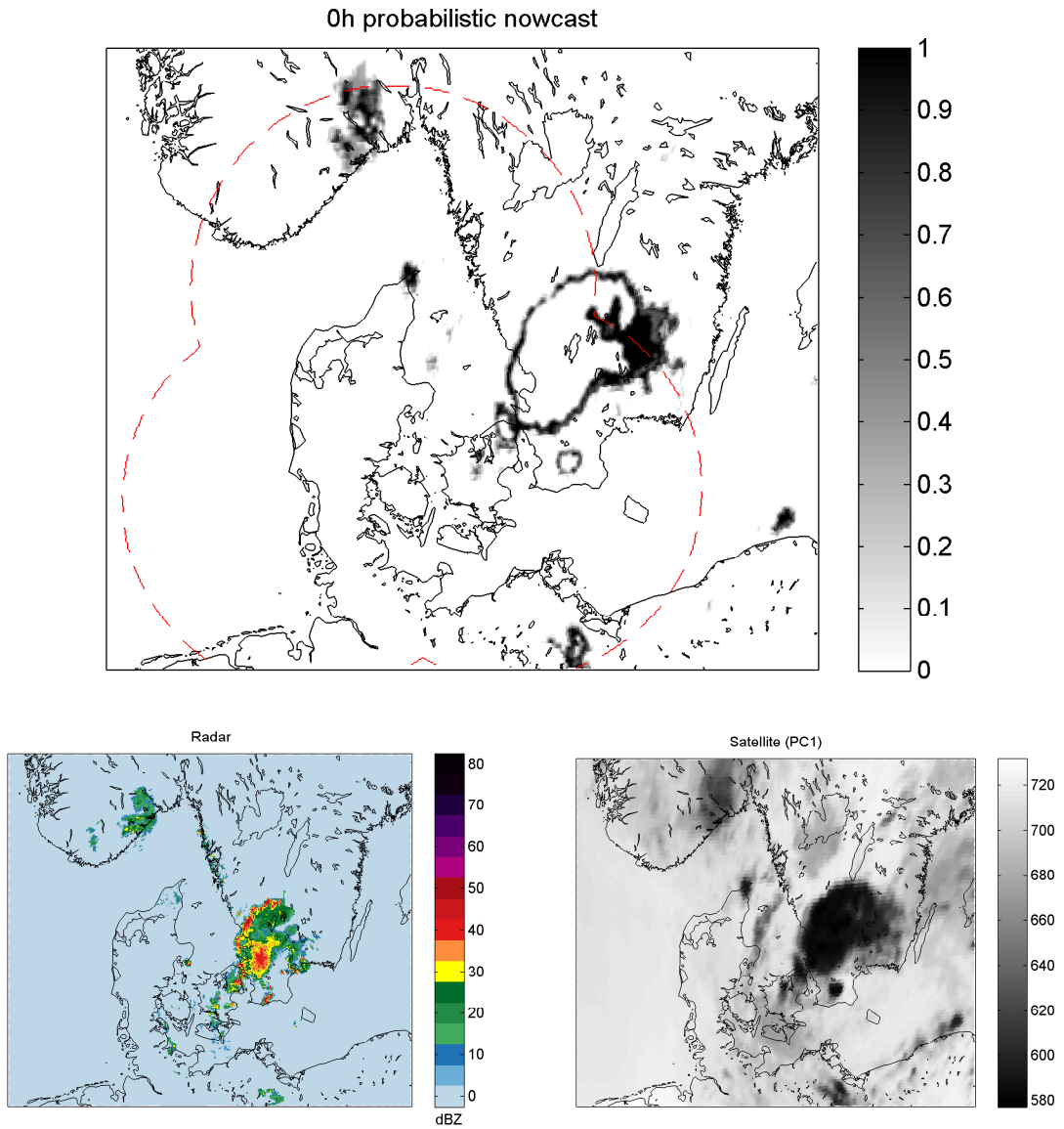
### C.16.1 0h nowcast



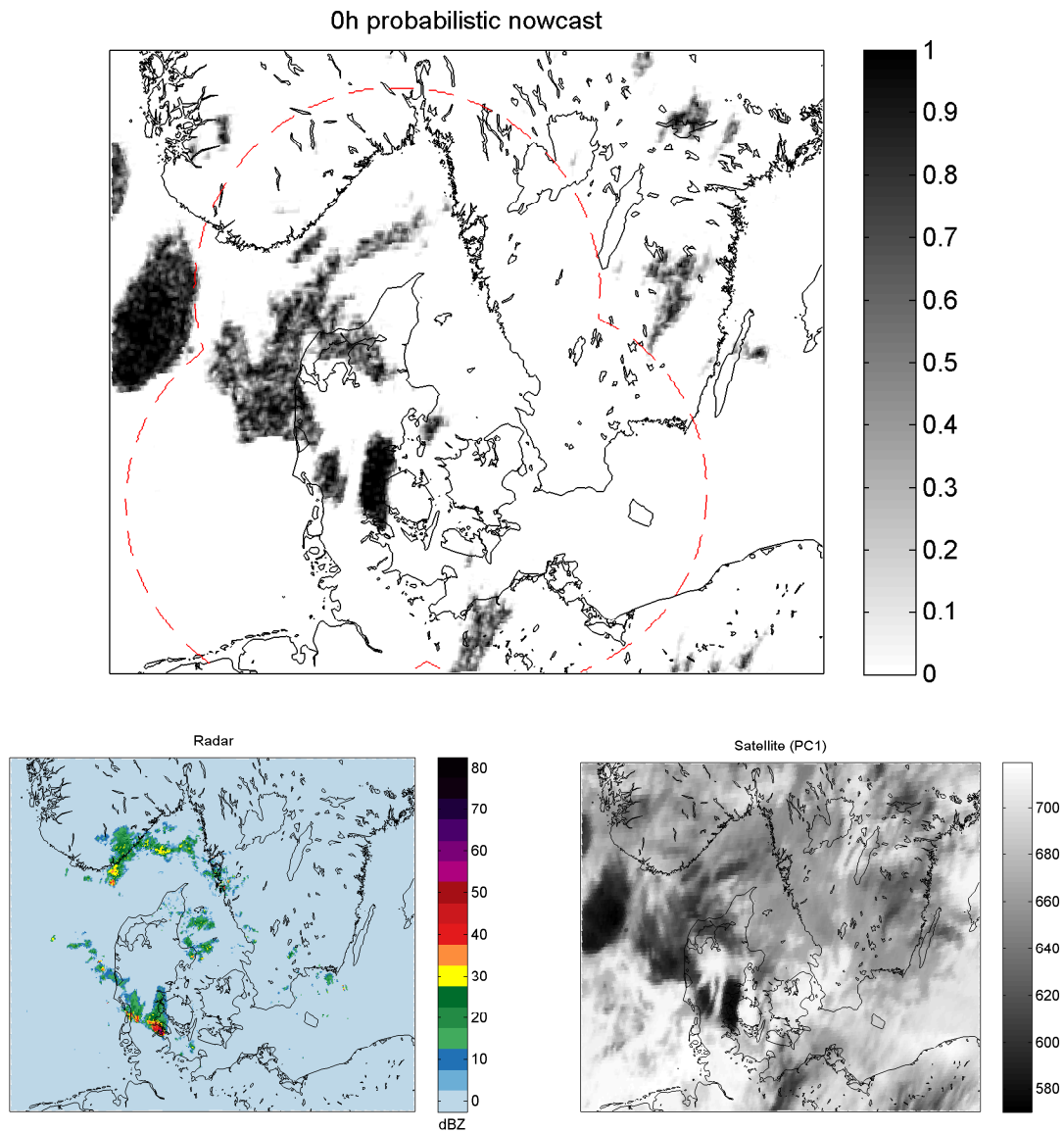
**Figure C.66:** 2007-07-16. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

### C.16.2 0.5h nowcast

APPENDIX C. SUPPLEMENTARY RESULTS

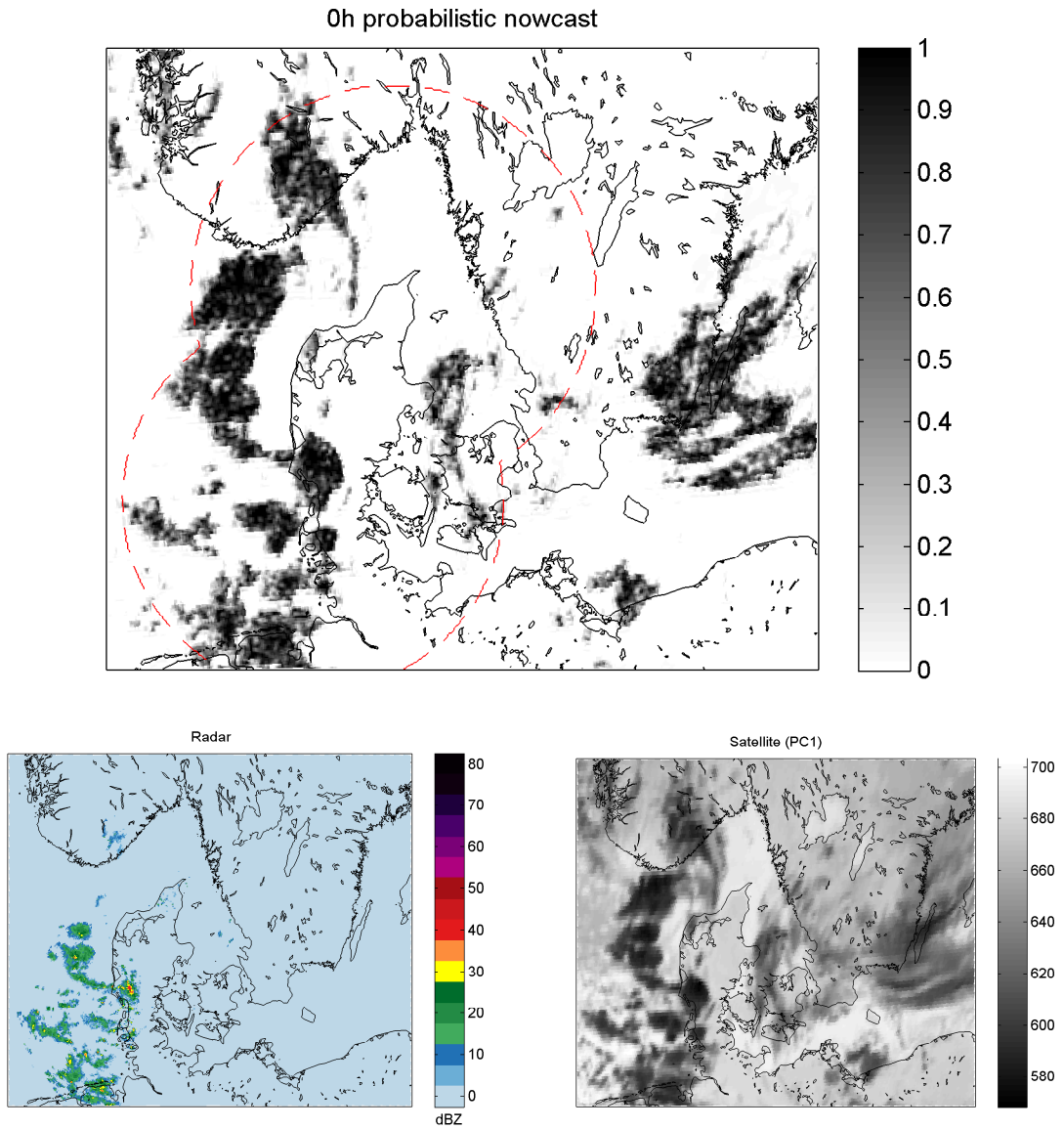


**Figure C.67:** 2007-08-11. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

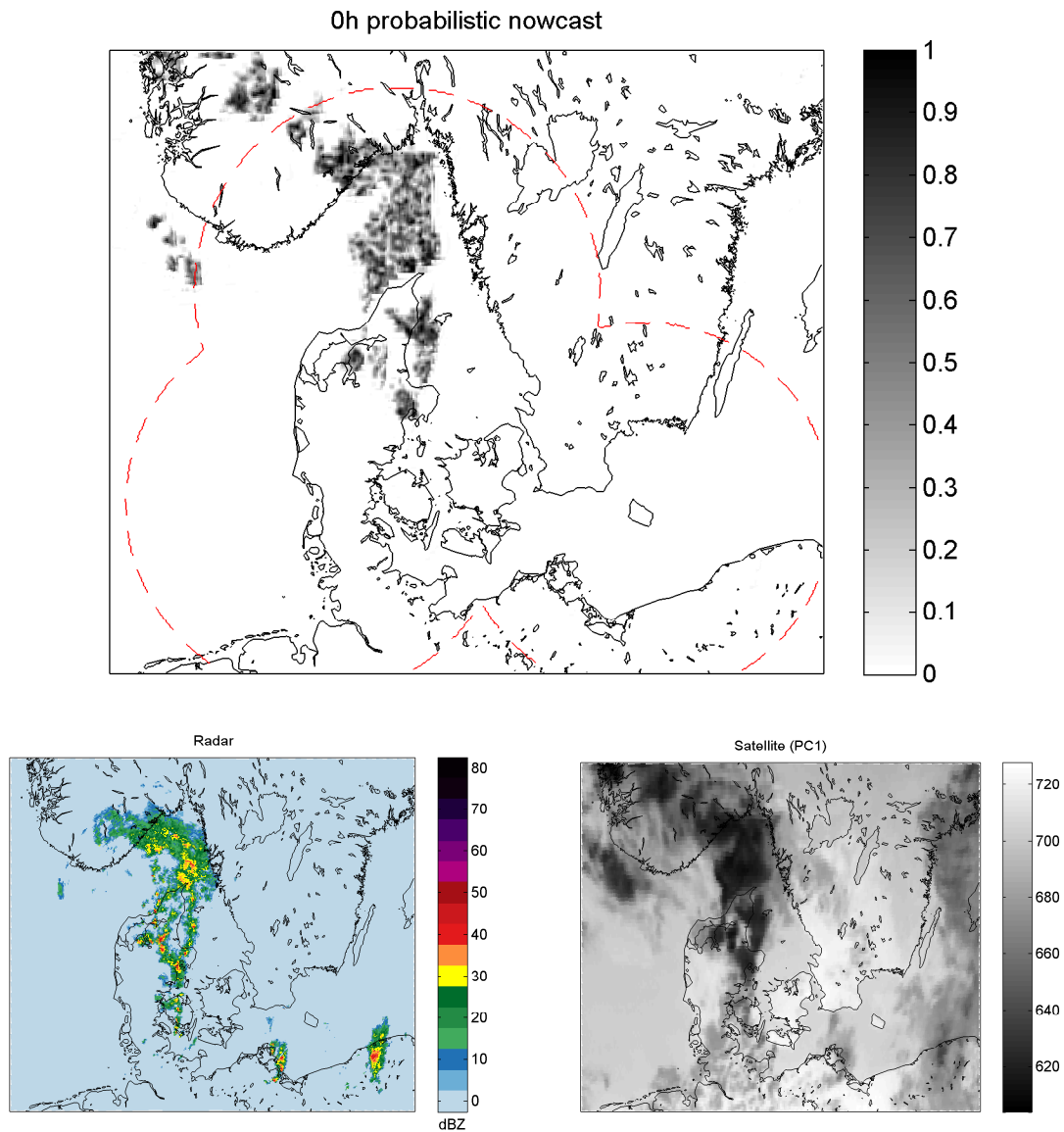


**Figure C.68:** 2007-08-20. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

APPENDIX C. SUPPLEMENTARY RESULTS

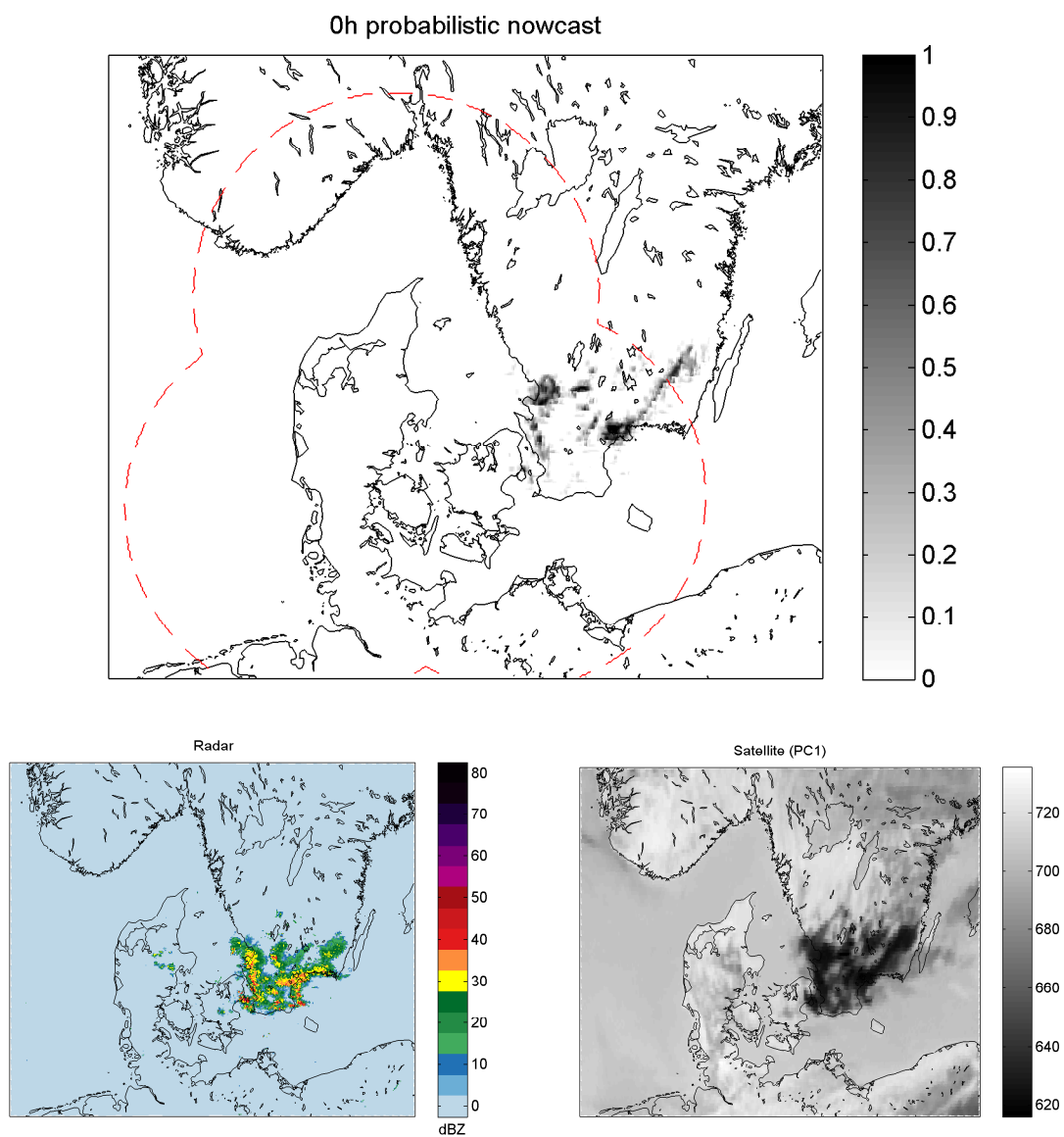


**Figure C.69:** 2007-11-11. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.



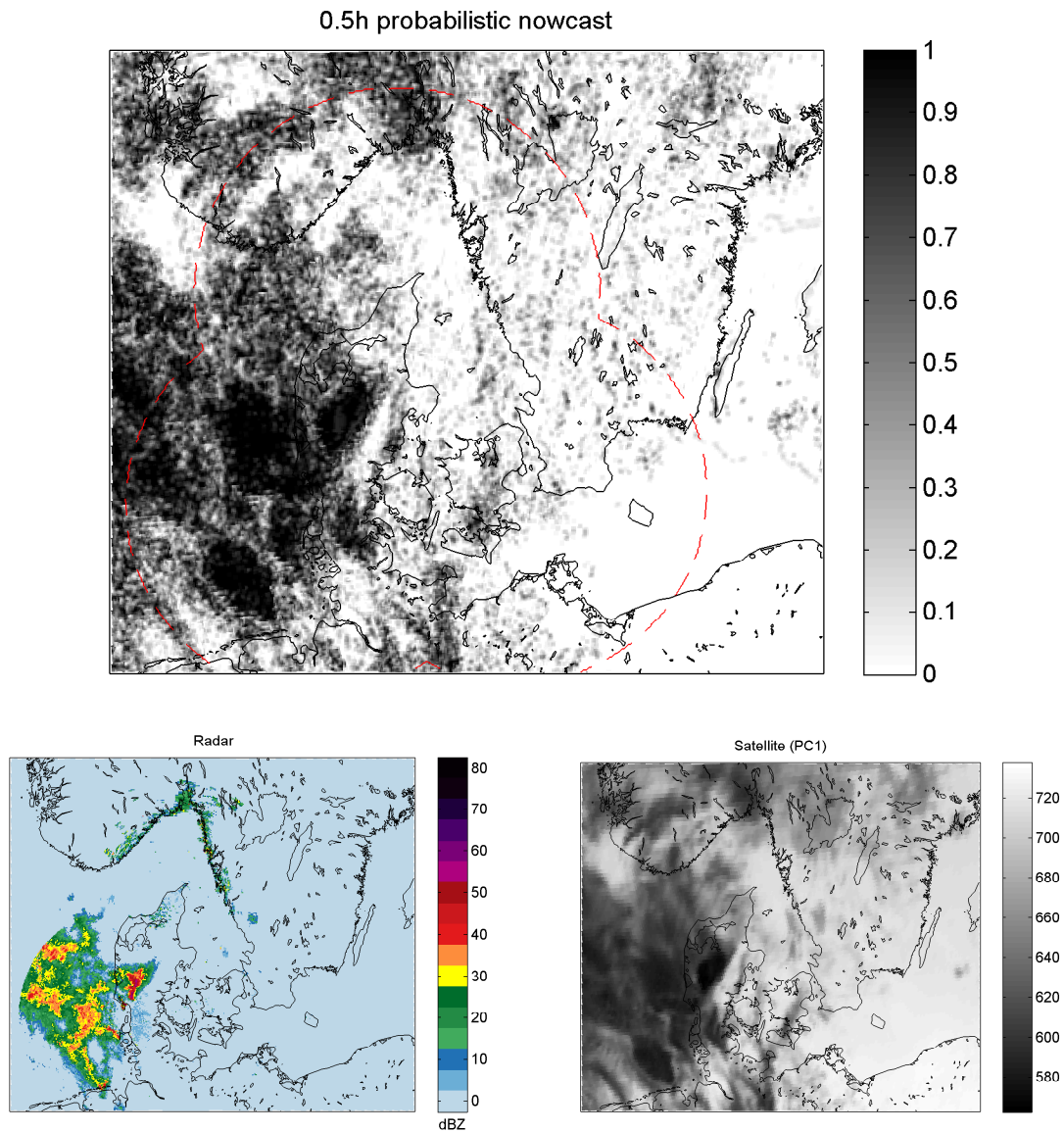
**Figure C.70:** 2009-05-18. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

APPENDIX C. SUPPLEMENTARY RESULTS



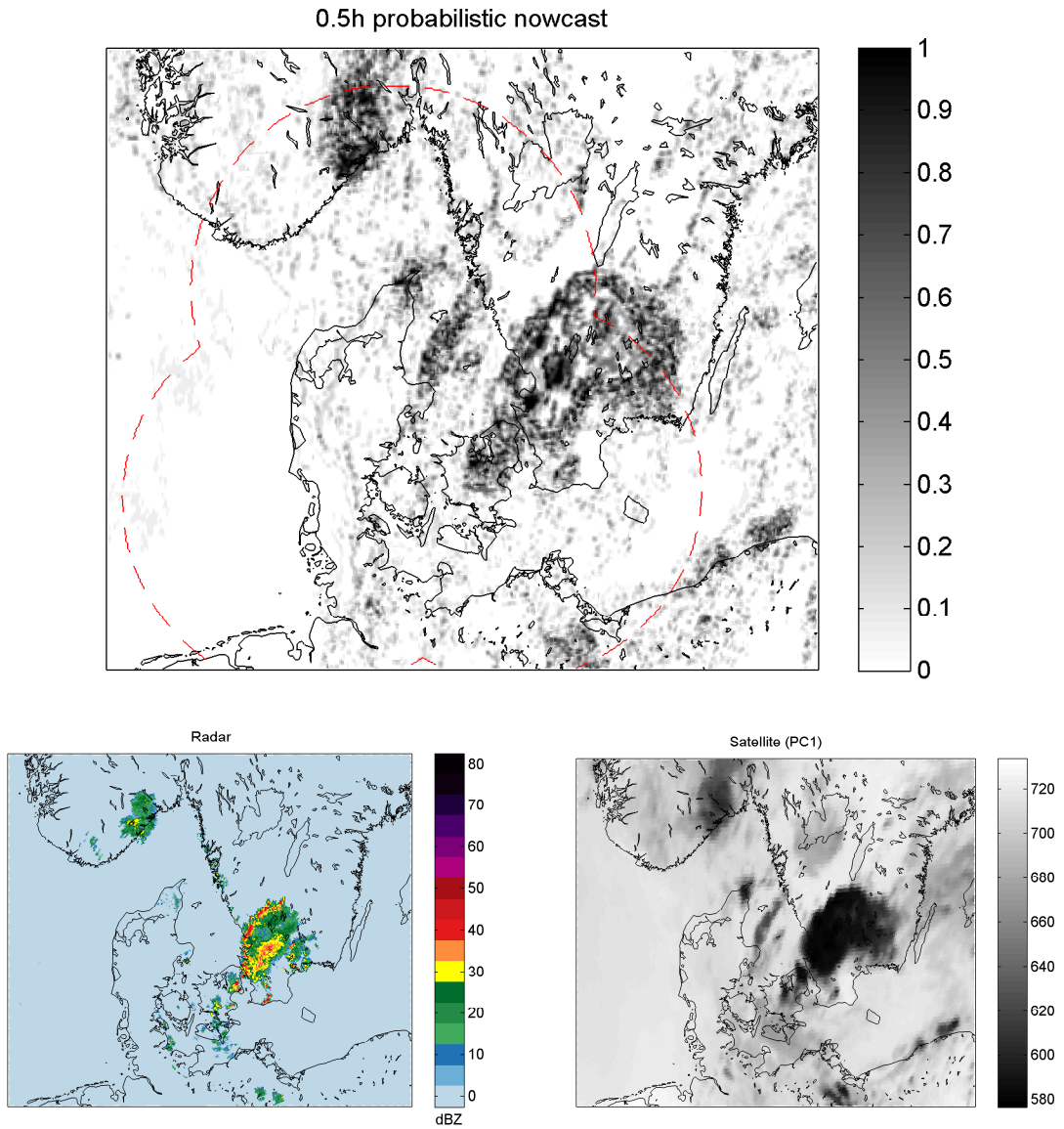
**Figure C.71:** 2010-06-15. (top) 0h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.





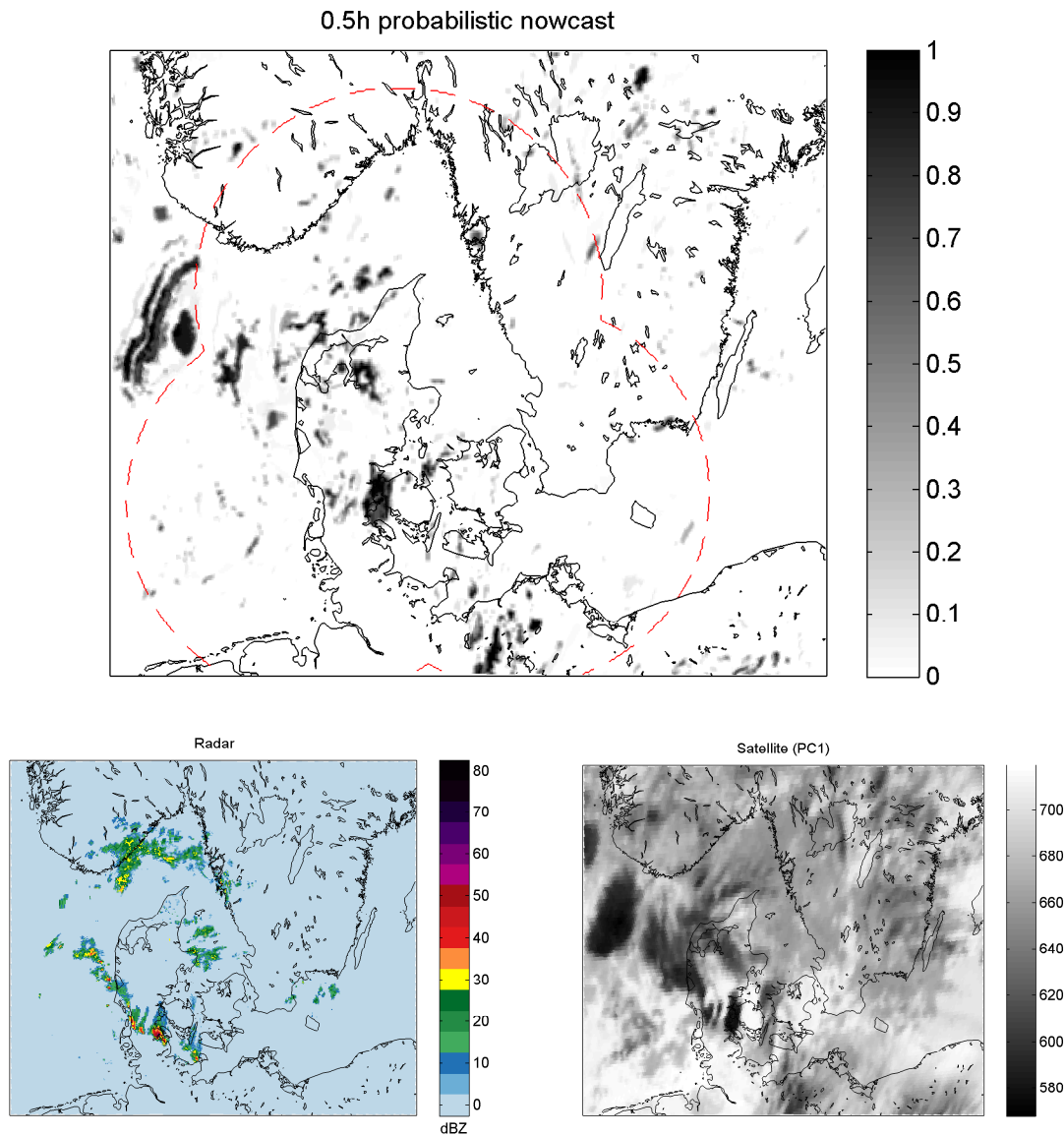
**Figure C.72:** 2007-07-16. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

APPENDIX C. SUPPLEMENTARY RESULTS

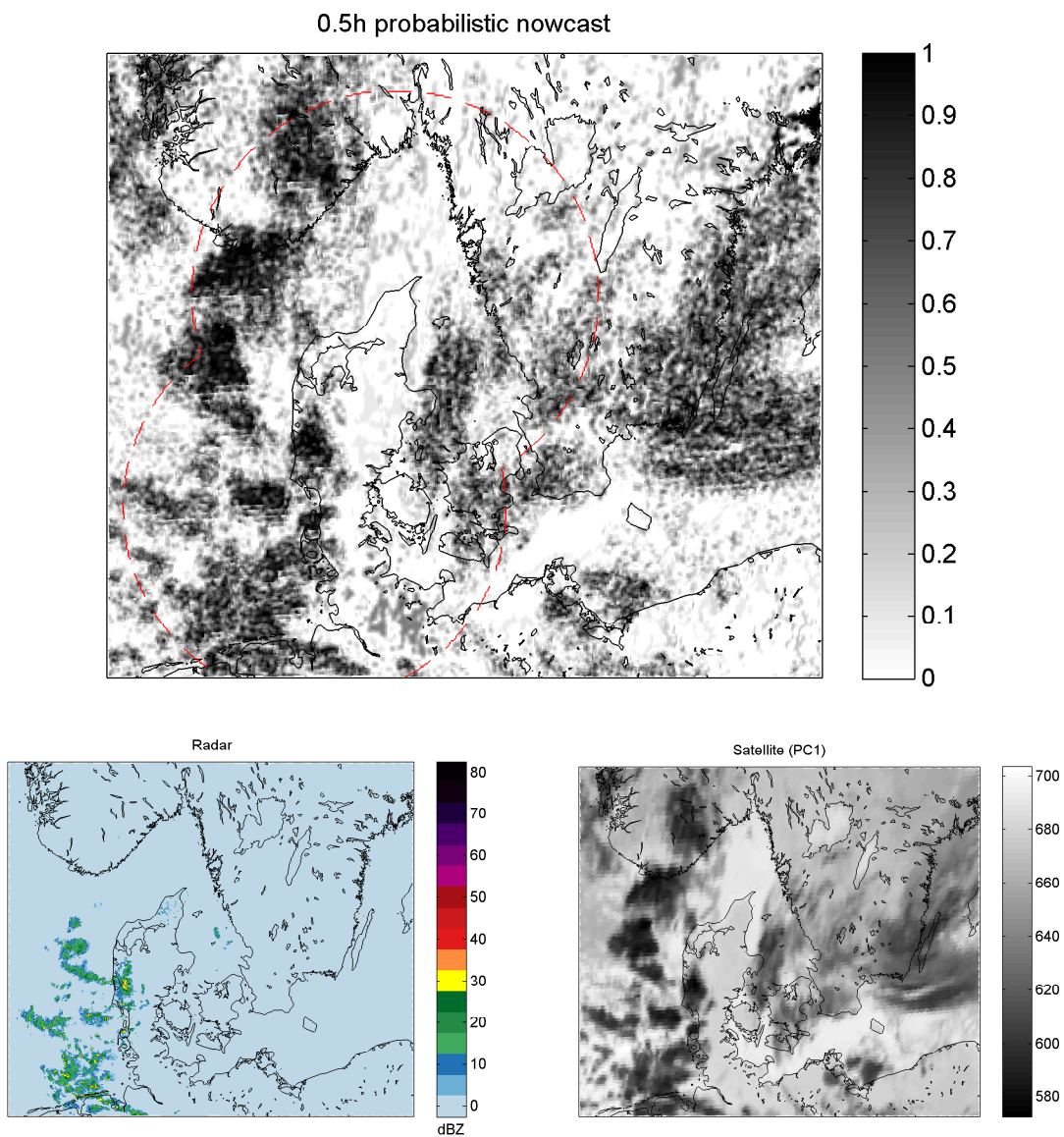


**Figure C.73:** 2007-08-11. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

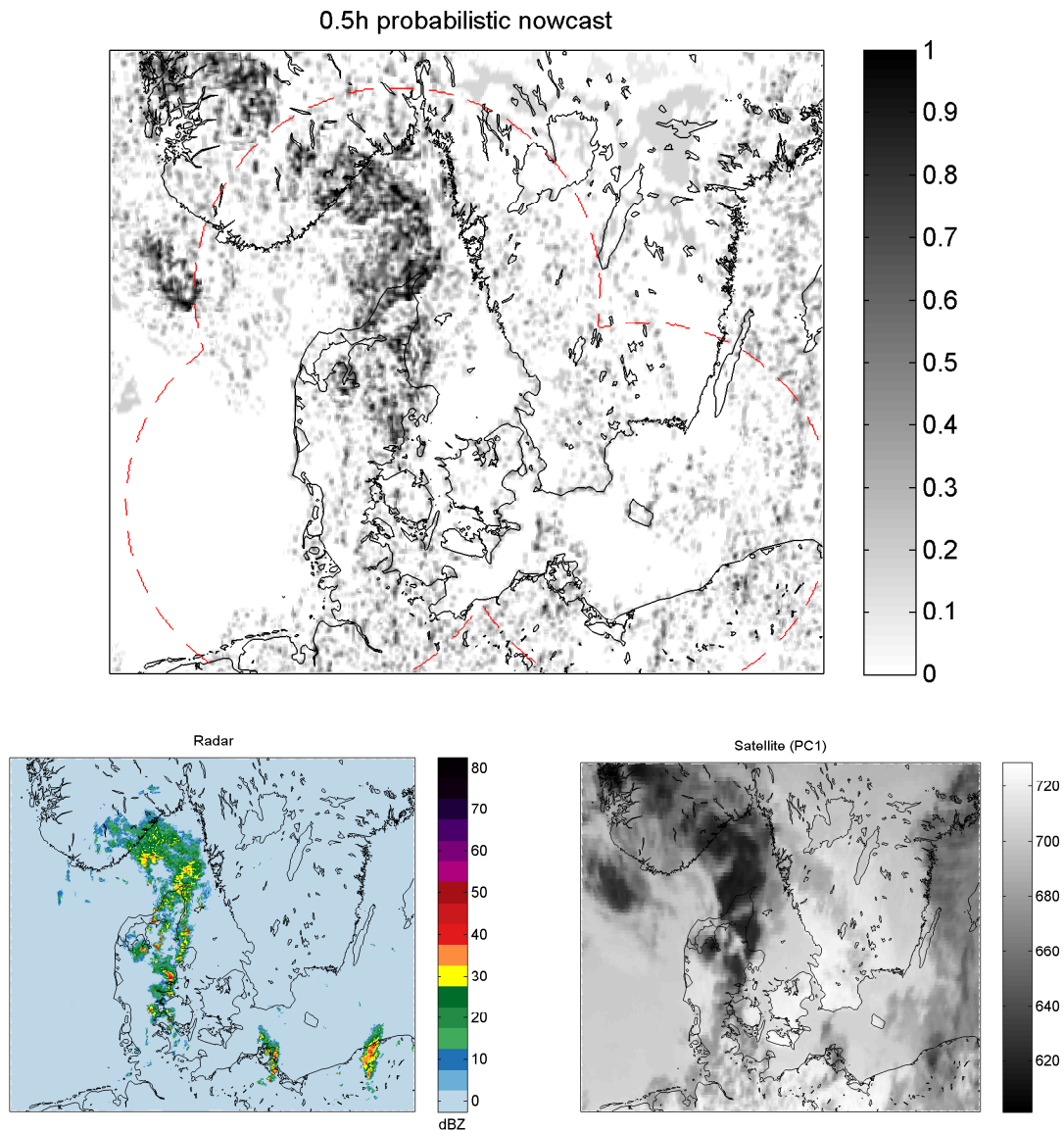




**Figure C.74:** 2007-08-20. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

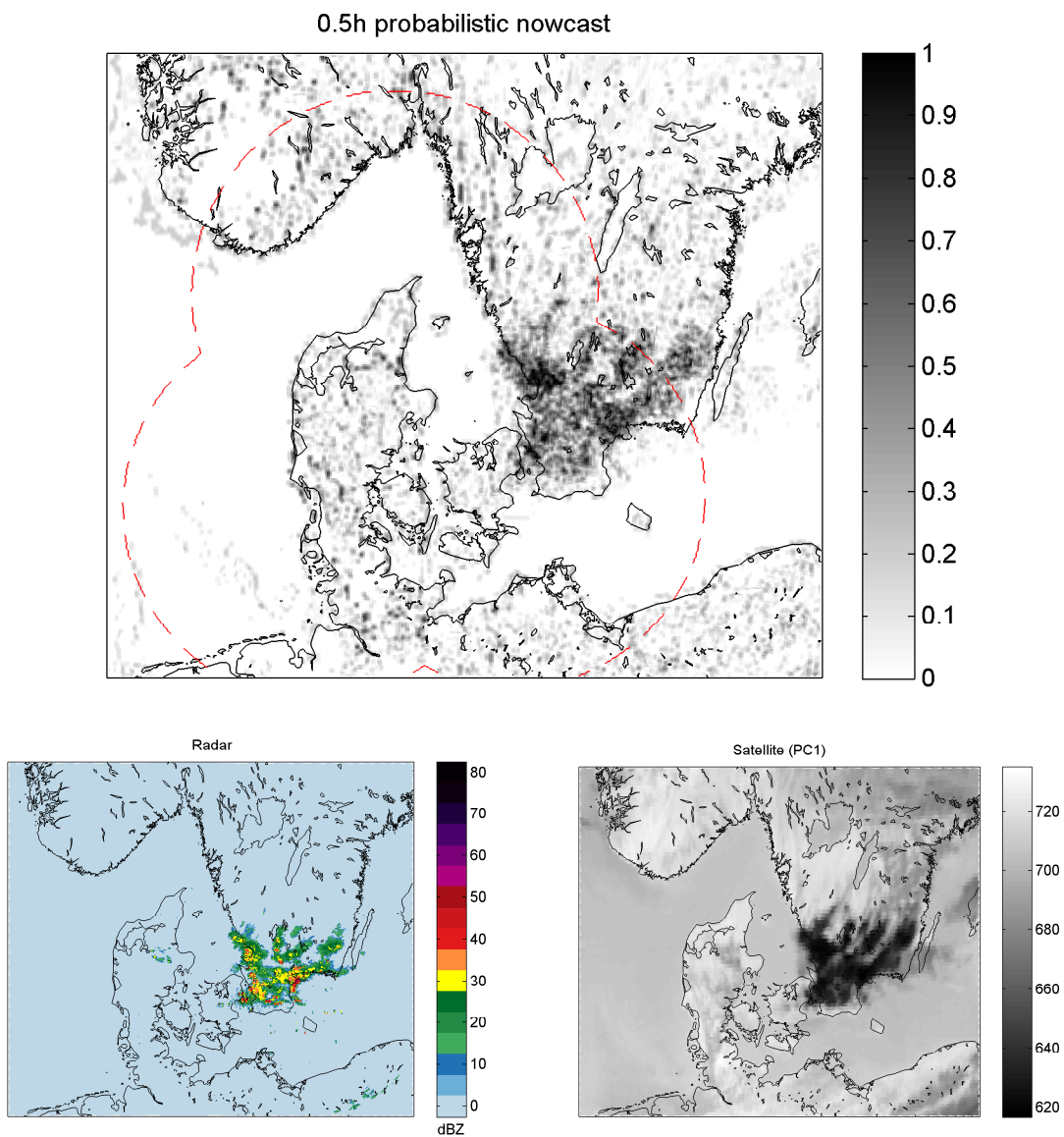


**Figure C.75:** 2007-11-11. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.



**Figure C.76:** 2009-05-18. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

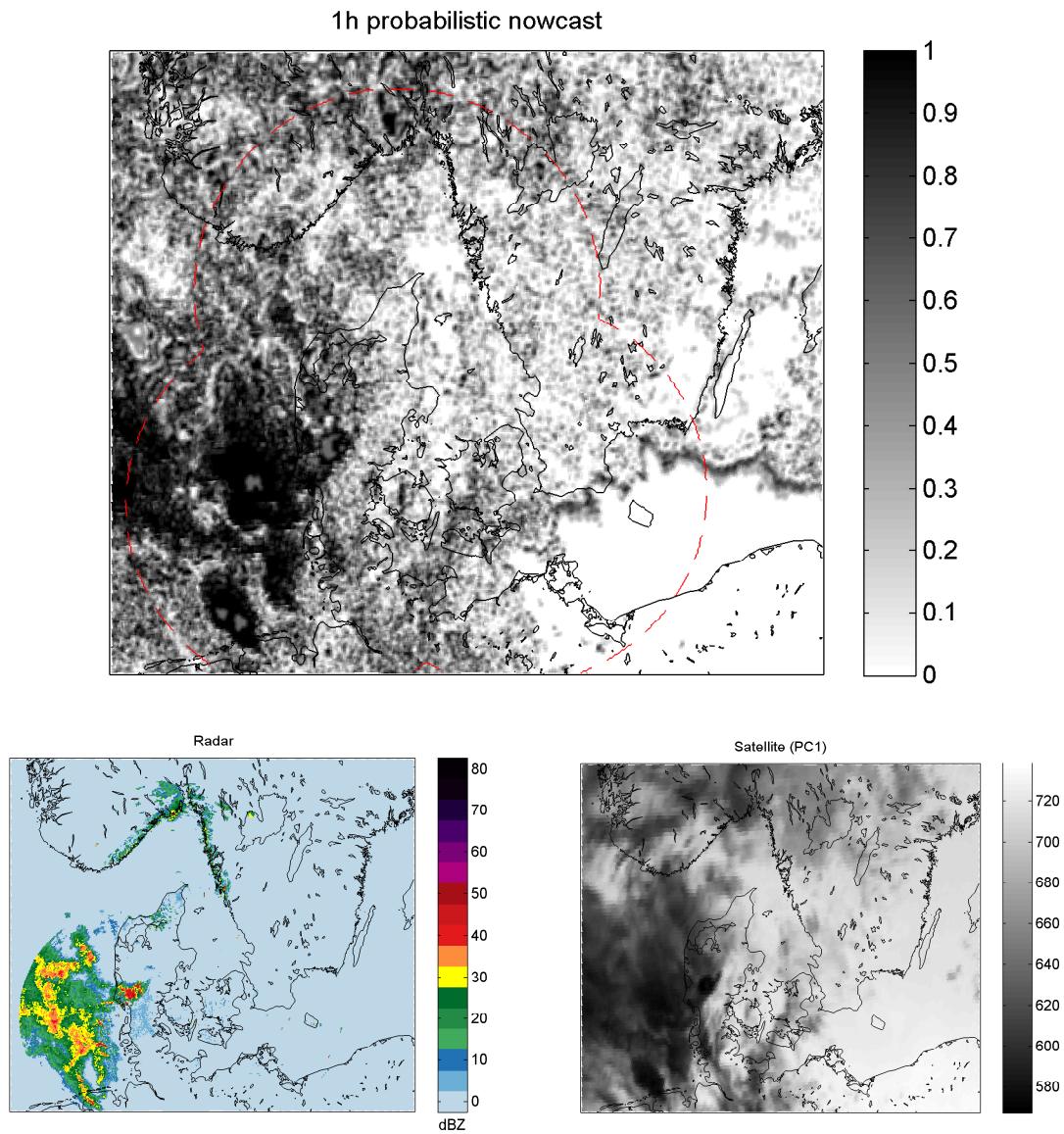
APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.77:** 2010-06-15. (top) 0.5h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

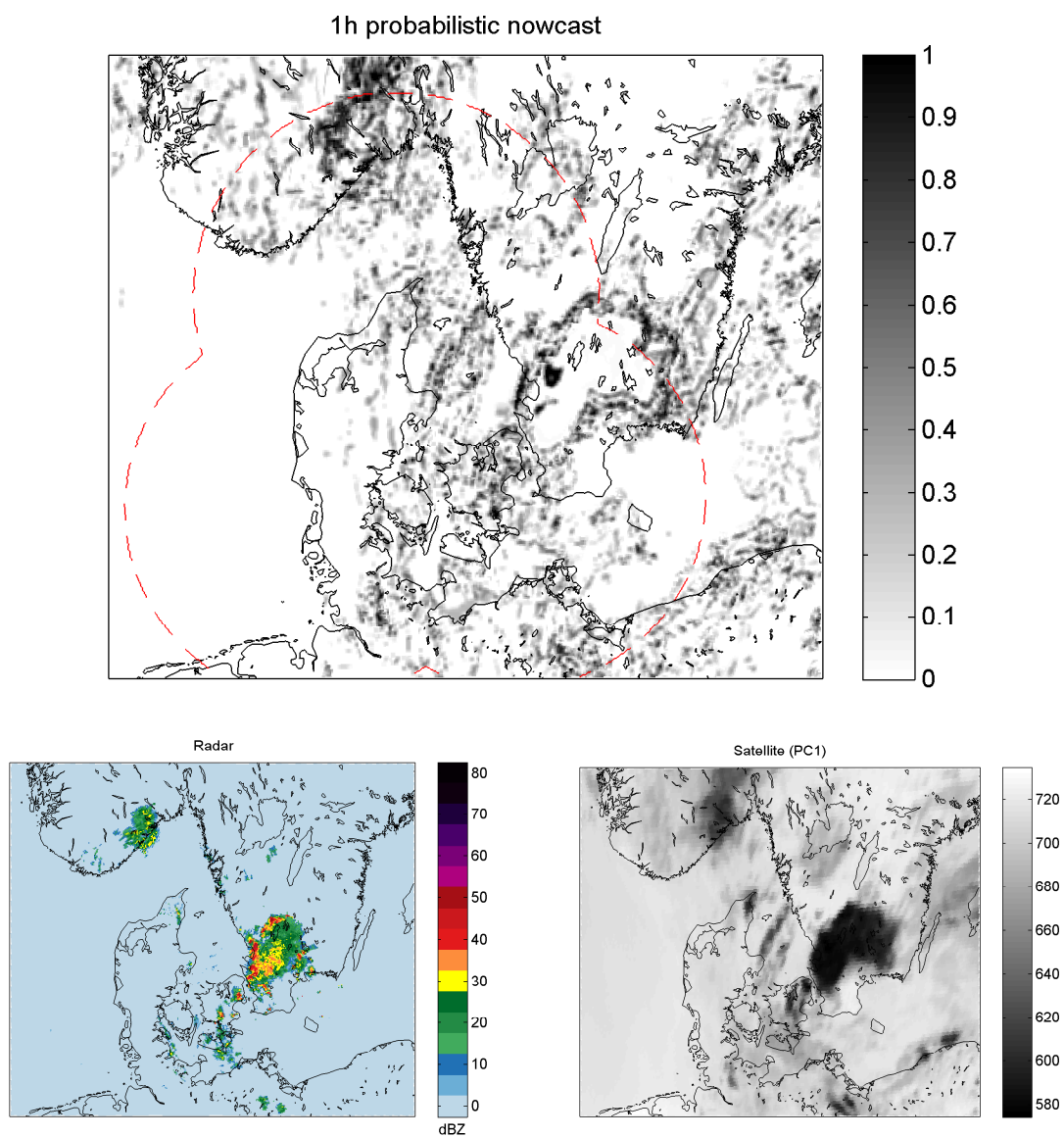


## C.16.3 1h nowcast

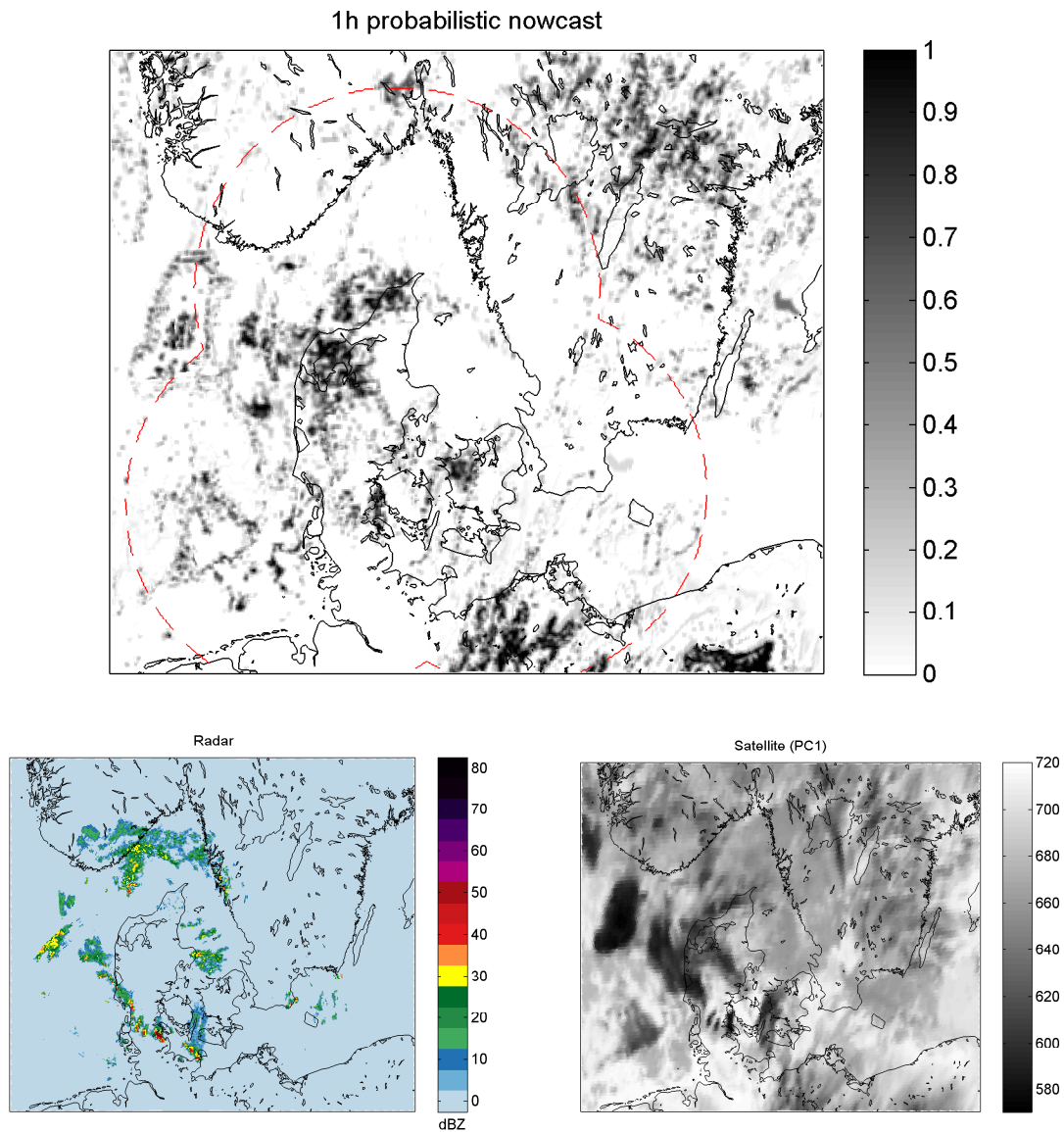


**Figure C.78:** 2007-07-16. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

APPENDIX C. SUPPLEMENTARY RESULTS

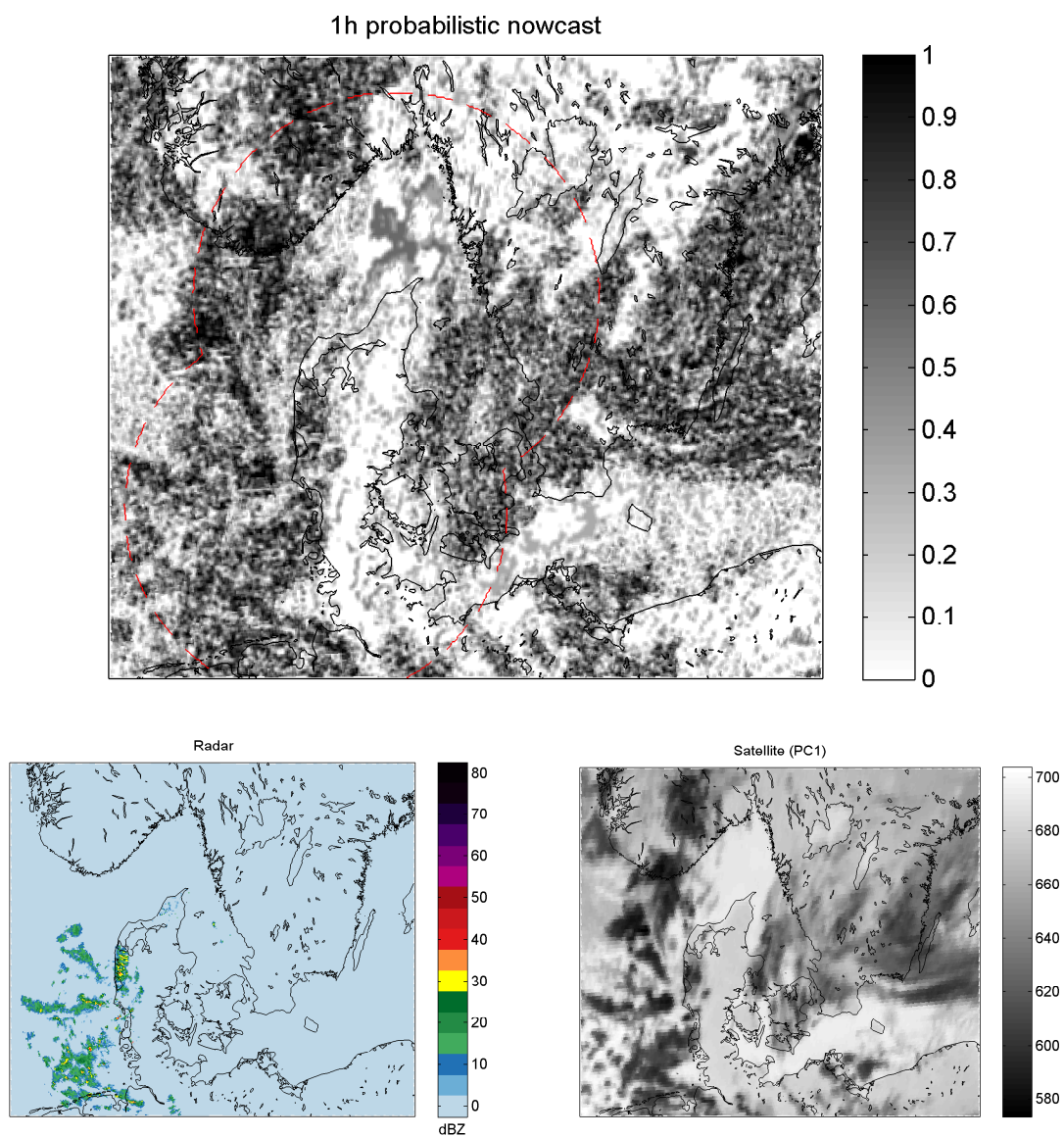


**Figure C.79:** 2007-08-11. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.



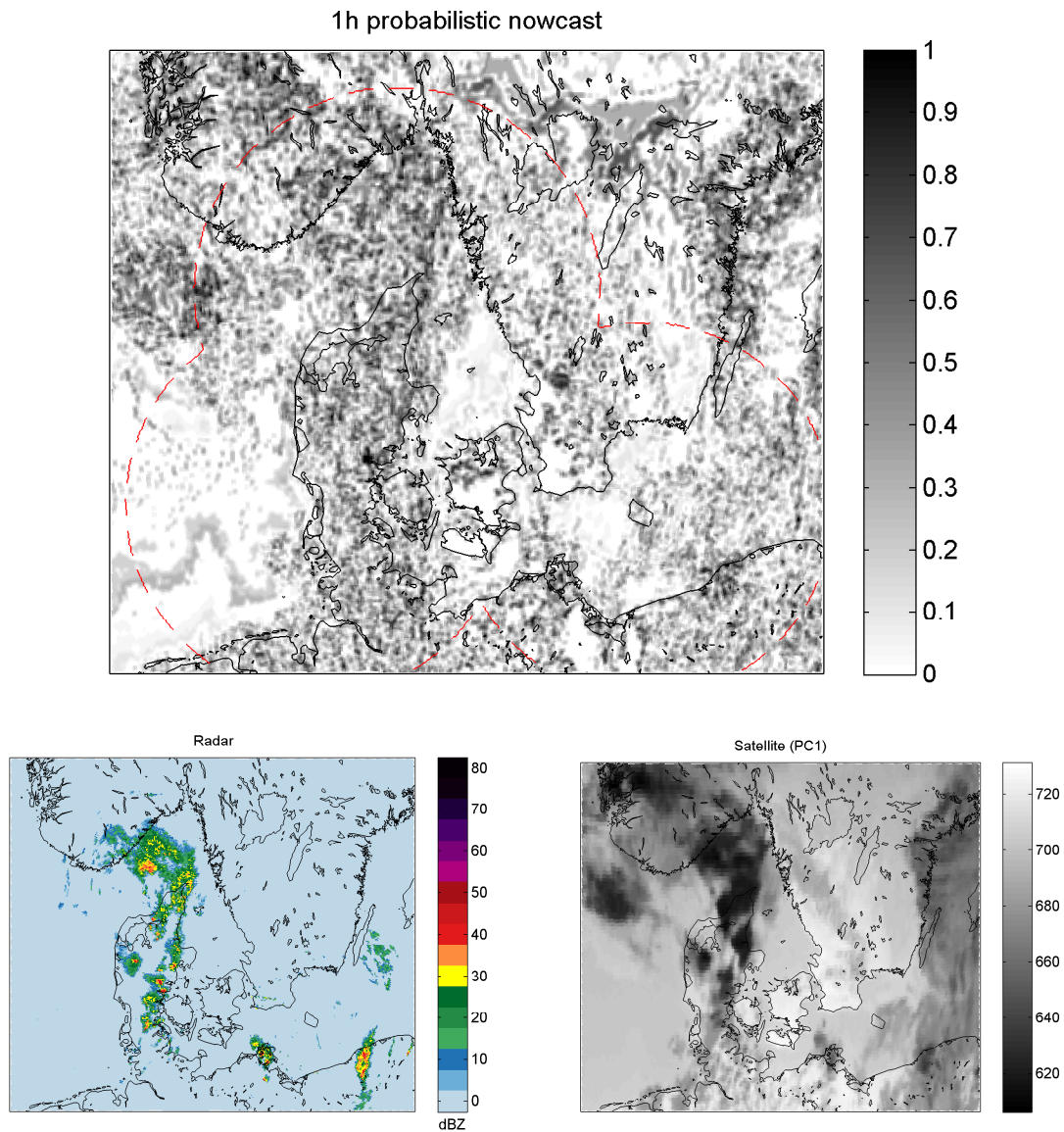
**Figure C.80:** 2007-08-20. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

APPENDIX C. SUPPLEMENTARY RESULTS



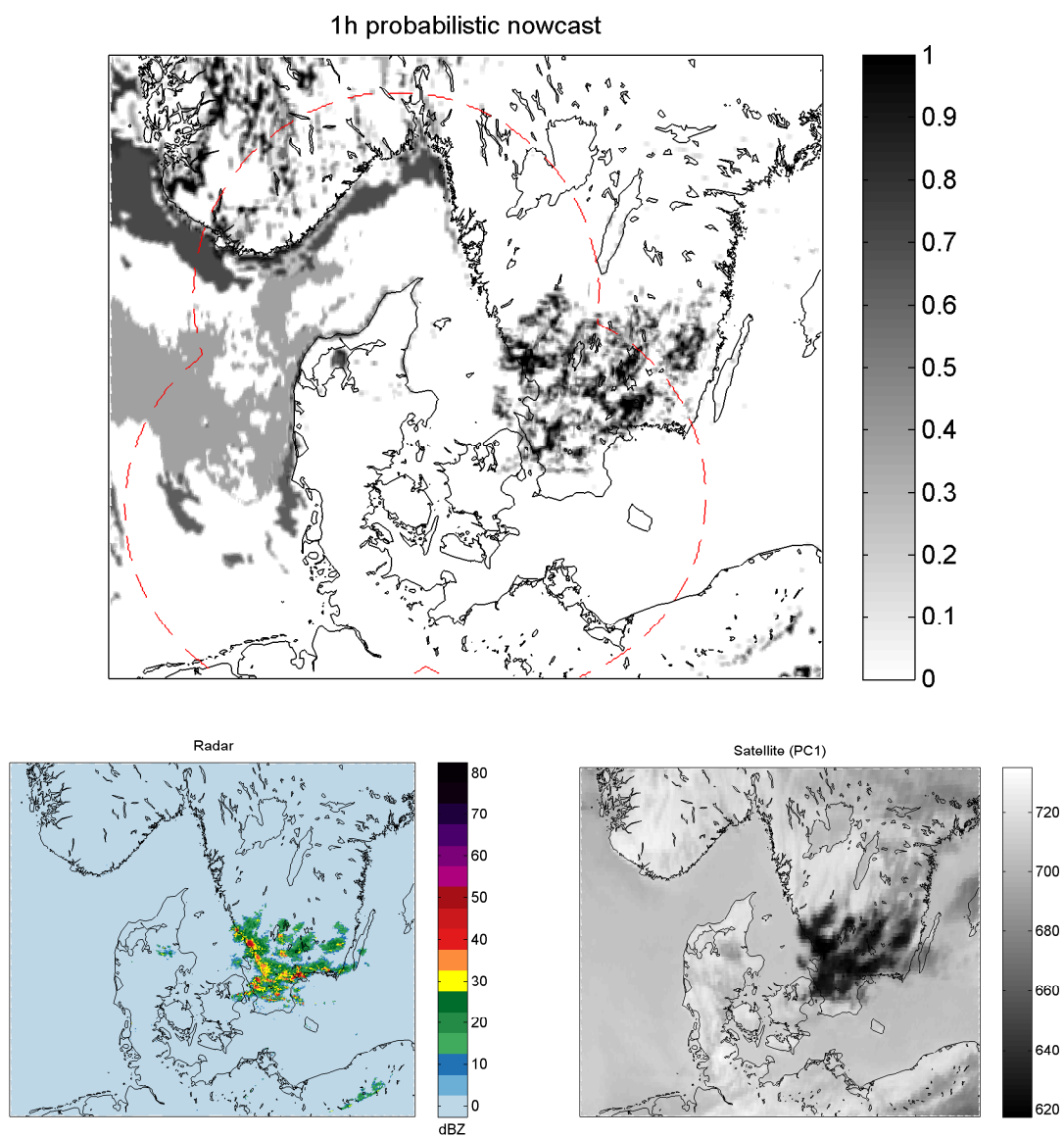
**Figure C.81:** 2007-11-11. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.





**Figure C.82:** 2009-05-18. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.


APPENDIX C. SUPPLEMENTARY RESULTS



**Figure C.83:** 2010-06-15. (top) 1h probabilistic nowcast using dictionary method with parameters determined from CV. (bottom left) Radar data and (bottom right) PC1 of available satellite data available at time of produced nowcast.

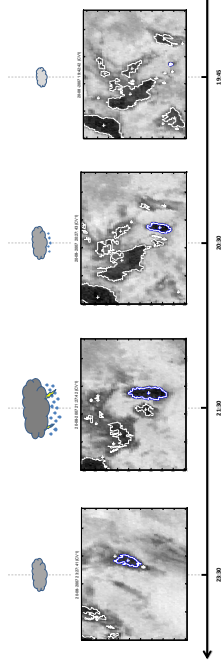
# Visiondays poster


This poster was presented at the Industrial Visionday at DTU Informatics, the 11th of May 2011.



**IMPROVED NOWCASTING OF HEAVY PRECIPITATION USING SATELLITE AND WEATHER RADAR DATA**  
Jacob S. Vestergaard, Allan A. Nielsen\*, Rasmus Larsen† & Thomas Bevilhøj\*

DTU Informatics, DTU Space, DTU





### Nowcasting

Changes in the global climate over the last few years have led to an increase in the frequency and intensity of extreme weather events. This is due to a combination of factors, including a warming of the atmosphere and a rise in sea levels. These changes have led to an increase in the number of extreme weather events, such as heavy precipitation, which can cause significant damage to infrastructure and property.

On 14th July 2007, a heavy rain event was observed in the region of Copenhagen, Denmark. The event was characterized by heavy precipitation, which caused significant damage to infrastructure and property. The event was caused by a combination of factors, including a warming of the atmosphere and a rise in sea levels.

The purpose of this project is to use a combination of satellite and weather radar data to improve nowcasting of heavy precipitation events. The project will focus on the use of satellite data to provide early warning of heavy precipitation events, and on the use of weather radar data to provide more accurate nowcasting of heavy precipitation events.

### Radar And Satellite Data

Satellite and radar data are used to provide early warning of heavy precipitation events. The satellite data is used to provide early warning of heavy precipitation events, and the radar data is used to provide more accurate nowcasting of heavy precipitation events. The satellite data is used to provide early warning of heavy precipitation events, and the radar data is used to provide more accurate nowcasting of heavy precipitation events.

The satellite data is used to provide early warning of heavy precipitation events, and the radar data is used to provide more accurate nowcasting of heavy precipitation events. The satellite data is used to provide early warning of heavy precipitation events, and the radar data is used to provide more accurate nowcasting of heavy precipitation events.

### Cross Correlation Alignment

This section describes the process of aligning satellite and radar data. The process involves comparing the two data sets and identifying the best alignment. This is done by calculating the cross-correlation between the two data sets, and then finding the maximum value of the cross-correlation. This maximum value indicates the best alignment between the two data sets.

The process of aligning satellite and radar data is a critical step in the nowcasting process. It allows us to combine the strengths of both data sources, and to provide more accurate nowcasting of heavy precipitation events.


### Tracking in Projection Space

This section describes the process of tracking heavy precipitation events in projection space. The process involves identifying the location of the event in projection space, and then tracking its movement over time. This is done by calculating the trajectory of the event, and then using this trajectory to predict its future location.

The process of tracking heavy precipitation events in projection space is a critical step in the nowcasting process. It allows us to identify the location of the event, and to track its movement over time. This information is used to provide more accurate nowcasting of heavy precipitation events.

### Future work

This section describes the future work planned for this project. The future work will focus on the use of satellite and weather radar data to improve nowcasting of heavy precipitation events. The project will continue to develop and test new methods for nowcasting heavy precipitation events, and will aim to provide more accurate and reliable nowcasting of heavy precipitation events.



**DTU Informatics**  
Department of Informatics and Mathematical Modelling