Real-time detection of clouds on board of satellites with FPGAs

Matthias Eder

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors-Ingenieur (Dr.-Ing.)

genehmigten Dissertation.

Gutachter:

1. Prof. Dr.-Ing. Felix Huber
2. Prof. Dr.-Ing. Sabine Klinkner

Die Dissertation wurde am 18.11.2015 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 14.09.2016 angenommen. Die mündliche Prüfung fand am 17.11.2016 statt.

Abstract

In recent years, the performance of modern optical remote sensing instruments on board of satellites improved greatly due to better spatial and spectral resolution. This requires higher storage capacities as well as higher downlink times for the captured data. Due to an average global cloud coverage of about 66% the recorded images are often contaminated with clouds. Hence, for users that are mostly interested in surface features like monitoring vegetation conditions or desertification, these images do not contain the desired information and are therefore of no value.

One can see that an on-board cloud detection tool would be very useful to prevent dissipation of storage resources and downlink capacities in the case of cloud-contaminated images. This task principally consists of two parts: the first one is choosing a suitable cloud detection algorithm, the second part is implementing it in a satellite environment.

For the implementation of cloud detection algorithms on board of satellites Field Programmable Gate Arrays (FPGAs) are an optimal device due to their low costs, low power consumption and high flexibility. These properties combined with the ability of reconfiguration during operations make this hardware ideal for the realization of on-board processing functionality without stressing the On-board computer (OBC) or the Payload processing unit (PPU). Another useful characteristic of the FPGA is that its processing speed when using adequate algorithms is sufficient to allow real-time processing of the incoming data flow.

In this thesis a sophisticated approach is developed which allows on-board cloud detection on FPGAs with low hardware consumption. The basis of this method is that the conversion results are precalculated and stored in a Look-up table (LUT). This can be done because the conversion from the digital output to the corresponding physical magnitudes per spectral band is a function of a maximum of three known variables, depending on the sensor and the desired physical magnitude. Assuming that the accuracy of the algorithms is not too heavily decreased by discretising the input variables one can restrict the input space and therefore also the output space in order to minimize the size of the LUT. This allows the conversion of every digital output value into e.g. radiance in advance which has two advantages: instantaneous access to the corresponding value is offered and no hardware on the FPGA is consumed for the conversion of the data to physical magnitudes.

The method developed in this thesis was applied to data from two sensors, the Moderate-resolution Imaging Spectroradiometer (MODIS) and the Enhanced Thematic Mapper (ETM+). Changes in the quality of the cloud mask caused by the adjustments made in the LUT-based method are assessed via a comparison of the algorithms with or without the use of LUTs on a large amount of data. Two supervised learning algorithms, Linear discriminant analysis (LDA) and Support vector machines (SVM), were applied to data from both sensors. In addition, the operational cloud detection algorithm for ETM+ (ACCA) based on a threshold method was implemented on an FPGA.

For ETM+ the accuracy of the supervised learning algorithms using LUTs compared to their unscaled counterparts is 99.3% for LDA and 98.1% for SVM. For MODIS we get a corresponding accuracy of 99.3% for LDA and 99.9% for SVM.

For ACCA using a LUT an accuracy of 97,5% compared to the unscaled variant is reached.

Zusammenfassung

In den vergangenen Jahren verbesserte sich die Leistung der optischen Fernerkundungsinstrumente durch verbesserte räumliche und spektrale Auflösung stark. Damit einhergehend werden eine größere Speicherkapazität und höhere Downlinkraten nötig. Da allerdings die durchschnittliche globale Wolkenbedeckung bei 66% liegt sind die Aufnahmen oft mit Wolken versetzt. Für Benutzer die hauptsächlich an Oberflächenmerkmalen wie der Beobachtung der Vegetationsbedingungen oder der Desertifikation interessiert sind enthalten diese Aufnahmen keinen Informationsgehalt und sind somit wertlos.

Man sieht, dass ein On-Board Wolkendetektionsmechanismus für die Vermeidung der Speicherung und des Downlinks der wolkenversetzten Aufnahmen nützlich wäre. Diese Aufgabe besteht prinzipiell aus zwei Teilen: der Auswahl eines passenden Wolkendetektionsalgorithmus und der Implementierung dessen in einer Satellitenumgebung.

Für die Implementierung des Wolkendetektionsalgorithmus auf Satelliten bieten sich durch ihre geringen Kosten, ihre geringe Leistungsaufnahme und ihre hohe Flexibilität Field Pro- grammable Gate Arrays (FPGAs) an. Kombiniert mit der Möglichkeit der Rekonfiguration während der laufenden Mission ist diese Hardware ideal zur Umsetzung von On-Board-Funktionalitäten ohne den On-Board Computer (OBC) oder die Payload Processing Unit (PPU) zu belasten. Eine weitere nützliche Eigenschaft der FPGAs ist die hohe Geschwindigkeit bei Benutzung geeigneter Algorithmen, die eine Echtzeitprozessierung des ankommenden Datenstroms erlaubt.

In dieser Arbeit wurde ein Ansatz verfolgt, der eine On-Board Wolkendetektion auf FPGAs mit geringem Hardwareverbrauch ermöglicht. Die Basis dieses Ansatzes ist die Speicherung vorher berechneter möglicher Umrechnungswerte in einer Look-up Tabelle (LUT). Dies ist möglich da die Umrechnung von Grauwerten in entsprechende physikalische Größen pro Spektralband nur eine Funktion von maximal drei bekannten Variablen ist (abhängig vom Sensor und vom Spektralband). Angenommen die Genauigkeit des Algorithmus verringert sich nicht zu stark durch die Diskretisierung der Eingangsvariablen, kann man den Eingangsraum und somit auch den Ausgangsraum beschränken um die Größe der LUT zu minimieren. Dies erlaubt die Umrechnung eines jeden Grauwerts in z.B. Strahldichte im Voraus, was zwei Vorteile bietet: instantaner Zugriff auf den entsprechenden physikalischen Wert und kein Hardwareverbrauch zur Berechnung der physikalischen Werte auf dem FPGA.

Diese Methode wurde in dieser Arbeit auf zwei Sensoren angewendet, dem Moderate-resolution Imaging Spectroradiometer (MODIS) und dem Enhanced Thematic Mapper (ETM+). Veränderungen in der Qualität der Wolkenmaske wurden durch Vergleich der unskalierten Ergebnisse mit den Ergebnissen des LUT-basierten Ansatzes über einen großen Datensatz evaluiert. Zwei Algorithmen aus dem Bereich des überwachten Lernens, die lineare Diskriminanzanalyse (LDA) und die Supportvektor Maschine (SVM) wurden auf die Datensätze der beiden Sensoren angewandt. Zusätzlich wurde der operationelle Wolkendetektionsalgorithmus des ETM+ (ACCA), der auf einem Schwellwertalgorithmus basiert, auf einem FPGA implementiert.

Für ETM+ wurde eine Genauigkeit von 99,3% und für LDA eine Genauigkeit von 98,1% im Vergleich zu den unskalierten Algorithmen erreicht. Bei MODIS wurden Genauigkeiten von 99,3% (LDA) bzw. 99,9% (SVM) erreicht.

Für den ACCA-Algorithmus wurde eine Genauigkeit von 97,5% im Vergleich zu der unskalierten Variante erreicht.

Contents

1 Introduction **1**

2 Physical basis **5**

2.1 Cloud formation in the atmosphere 5

2.2 Spectral properties of the atmosphere relative to absorption and scattering processes 9

3 Data sources **13**

3.1 ETM+ 13

3.2 MODIS 18

4 Basics of FPGAs **23**

5 Algorithms **25**

5.1 Threshold algorithms 25

5.1.1 ACCA 26

5.1.2 MODIS Cloud Mask Algorithm 29

5.2 Supervised learning algorithms 31

5.2.1 Linear Discriminant Analysis 31

5.2.2 Support Vector Machines 36

6 Implementation **41**

6.1 Data Adaption 41

6.2 ACCA 44

6.3 Supervised Learning 58

6.3.1 Training Data 58

6.3.2 LDA 58
6.3.3 SVM 67
6.3.4 Discussion 75

7 Conclusion and Outlook 79

Bibliography 83

List of Abbreviations 91

Glossary 95

A Detailed description of the ACCA algorithm 97

B Comparison of the cloud coverage results published by LPS and the ACCA implementation 109

C Coefficients for the linear discriminant functions 113

C.1 Coefficients for LDA 113
C.2 Coefficients for SVM 115

D Procedure for the usage of the R package “e1071” 117

E Scene Identifiers for the scenes used in this thesis 121

E.1 ETM+ 121
E.2 MODIS 126

1 Introduction

In recent years, the performance of modern optical remote sensing instruments on board of satellites improved greatly due to better spatial and spectral resolution. This requires higher storage capacities as well as higher downlink times for the captured data. Due to the fact that the average global cloud coverage has a value of about 66 % [66] the recorded images are often contaminated with clouds. Hence, for users that are mostly interested in surface features like monitoring vegetation conditions [35] or desertification [12], these images do not contain the desired information and are therefore of no value.

One can see that an on-board cloud detection tool would be very useful to prevent dissipation of storage resources and downlink capacities in the case of cloud-contaminated images. This task principally consists of two parts: the first one is choosing a suitable cloud detection algorithm, the second part is implementing it in a satellite environment.

For the task of automated cloud detection of satellite imagery on ground many different approaches exist. Most of them make use of wavelength dependent characteristics of clouds and surface types. Sequential comparisons of single spectral bands or combinations of them to static [3, 26] or dynamic [46] thresholds allow the separation of clouds and surface. Methods based on machine learning techniques like unsupervised [9, 21] or supervised classificators [5, 8, 33] are also wide spread in the field of cloud detection.

For the implementation of cloud detection algorithms on board of satellites Field Programmable Gate Arrays (FPGAs) are an optimal device due to their low costs, low power consumption and high flexibility. These properties combined with the ability of reconfiguration during operations make this hardware ideal for the realization of on-board processing functionality without stressing the On-board computer (OBC) or the Payload processing unit (PPU). Another useful characteristic of the FPGA is that its processing speed when using adequate algorithms is sufficient to allow real-time processing of the incoming data flow.

First efforts for the implementation of a simple cloud detection algorithm based on thresholds for on-board purposes were made by Williams et al. [62] using an FPGA. The authors developed a flight-ready hardware for on-board cloud detection. El-Araby et al. [18] built on the work of Williams et al. [62] increasing the accuracy of the cloud mask. The drawback of both methods is the high consumption of hardware.

In this thesis a sophisticated approach is developed which allows on-board cloud detection on FPGAs with low hardware consumption. This method optimizes the conversion of the digital output of the sensor to the corresponding physical magnitudes like radiance or reflectance - a step necessary for the application to the algorithms. Each sensor has its own set of calibration parameters needed for this conversion. Hence by adjusting these parameters the conversion can be adapted to the corresponding sensor. Therefore, this method can be used for many different sensors and algorithms.

The basis of this method is that the conversion results are precalculated and stored in a Look-up table (LUT). This can be done because the conversion from the digital output to the corresponding physical magnitudes per spectral band is a function of a maximum of three known variables, depending on the sensor and the desired physical magnitude. Assuming that the accuracy of the algorithms is not too heavily decreased by discretising the input variables one can restrict the input space and therefore also the output space in order to minimize the size of the LUT. This allows the conversion of every digital output value into e.g. radiance in advance which has two advantages: instantaneous access to the corresponding value is offered and no hardware on the FPGA is consumed for the conversion of the data to physical magnitudes.

The method developed in this thesis was applied to data from two sensors, the Moderate-resolution Imaging Spectroradiometer (MODIS) and the Enhanced Thematic Mapper (ETM+). Changes in the quality of the cloud mask caused by the adjustments made in the LUT-based method are assessed via a comparison of the algorithms when using LUTs or not on a large amount of data. Two supervised learning algorithms, Linear discriminant analysis (LDA) and Support vector machines (SVM), were applied to data from both sensors. In addition, the operational cloud detection algorithm for ETM+ based on a threshold method was implemented on an FPGA. One part of the algorithm was already implemented by Williams et al. [62] and El-Araby et al. [18]. As shown in this thesis the full implementation of this sophisticated algorithm is also possible.

Thesis organization

This thesis is organized as follows: **chapter 2** gives an introduction into the formation of clouds as well as their spectral properties that are important for cloud detection algorithms based on spectral information.

Chapter 3 presents the two sensors whose data were used in this thesis, the Moderate-resolution Imaging Spectroradiometer (MODIS) and the Enhanced Thematic Mapper (ETM+). This chapter also describes the extraction and conversion of the data into physical magnitudes.

Chapter 4 explains the assets and drawbacks of Field Programmable Gate Arrays (FPGAs), their basic functionality and the platform used in this thesis.

In **chapter 5** the algorithms used in this thesis are described. For ETM+ the operational Advanced Cloud Cover Assessment (ACCA) algorithm developed by Irish [26] is explained. This algorithm is a mixture between a threshold algorithm and a statistical evaluation of the data. Two well established supervised learning algorithms are described as well: Linear discriminant analysis (LDA) and Support vector machines (SVM).

In **chapter 6** the method used in this thesis for optimizing the implementation of the algorithms on an FPGA platform is explained. Subsequently, the results of this method applied to the data of the two sensors are examined and evaluated. First the implementation of the ACCA algorithm for ETM+ is examined, and afterwards the implementation of the supervised learning algorithms for both sensors is analyzed.

Chapter 7 assesses the results and gives an outlook on possible future work and applications.

2 Physical basis

This chapter gives a brief introduction into the formation of clouds in the atmosphere and their spectral properties. The different mechanisms of cloud formation are described as well as the various cloud families. In the following both the spectral behavior of the clouds and the atmosphere are briefly introduced.

2.1 Cloud formation in the atmosphere

Clouds can be described as a visible mass of liquid water droplets or ice crystals in the atmosphere. In order to explain the cloud formation in the atmosphere we consider the atmosphere as a mixture of different gases. As can be seen in figure 2.1 the major constituents are mainly nitrogen (78 %) and oxygen (21 %). In addition water vapor is also present in the atmosphere. Clouds come into existence when either the gas-water vapor mixture is cooled or water vapor is added to the gas mixture. There are various ways how the air can be cooled down. The first one we examine is the *adiabatic cooling*. This process happens when the air mass is lifted due to atmospheric motion. Then the air

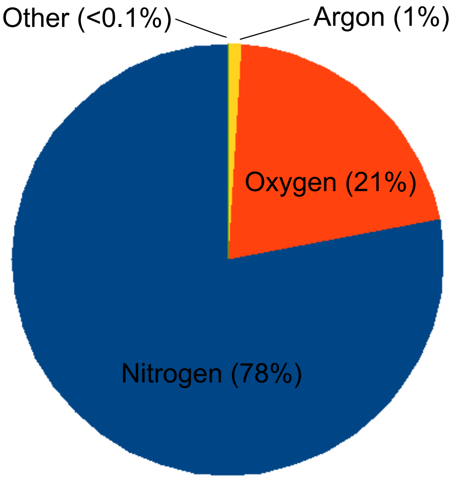


Figure 2.1: This pie chart shows the distribution of the atmospheric gases. Listed in “Other” we can find e.g. carbon dioxide, ozone and water (adapted from [13]).

is cooled because the air mass in a volume expands and so does work on the surrounding volumes and therefore loses inner energy (i.e. the temperature decreases) without gaining energy from outside the volume. In thermodynamical terms an adiabatic process means that $dQ = 0$, where Q denotes the *heat* of the system. So an adiabatic process can be described by the first law of thermodynamics as

$$dU = -dW, \tag{2.1}$$

where U is the inner energy of the system and W is the work done by the system. In cool air the saturation vapor pressure of water is lower than in warmer air. This means for the overall pressure of the gas-water mixture remaining constant the water fraction of the mixture starts to condensate when the temperature falls below the *dew point*. The dew point is the temperature where condensation and evaporation of water have the same rate; above this temperature evaporation predominates and below condensation predominates. This temperature can also be seen as the temperature where the *relative humidity* is exactly 100 %. In the presence of condensation nuclei (like dust, salt or pollen) that are normally always present in the atmosphere the water vapor starts to condensate on these nuclei. The height where condensation starts by adiabatic cooling is called *lifted condensation level* and determines the cloud base.

On ground level *non-adiabatic cooling* prevails and is the reason for the formation of fog. There are three mechanisms for non-adiabatic cooling. The first one is *conduction* when warm air moves over colder underground. The second one is the *emission of infrared radiation*, e.g. in clear sky conditions at night. The third mechanism occurs when moisture is added to the air. Then the air is cooled by *evaporative heat loss*. Adding moisture to the air can be achieved by different ways:

- movement of air over water or wet underground
- precipitation from higher atmospheric layers
- heating of the underground by solar radiation
- transpiration from plants
- gathering humidity of dry or cool air over water.

For further reading in the topic of cloud formation please refer to [45].

Clouds are classified by the *World Meteorological Organization* [63] into the four families

high clouds, *middle clouds*, *low clouds* and *vertical clouds* (see figure 2.2). Note that the height of the cloud families is dependent on the latitude because the troposphere on the equator is twice as high as on the poles.
In addition to the family the clouds are classified into ten genera:

- Cirrus (Ci)
- Cirrucumulus (Cc)
- Cirrustratus (Cs)
- Altocumulus (Ac)
- Altostratus (As)
- Stratocumulus (Sc)
- Stratus (St)
- Cumulus (Cu)
- Nimbostratus (Ns)
- Cumulonimbus (Cb)

As shown in figure 2.2 cirrus, cirrocumulus and cirrostratus belong to the family of the high clouds. Altocumulus, altostratus belong to the middle clouds and stratocumulus, stratus and cumulus belong to the low clouds. Representatives of the vertical clouds family are nimbostratus and cumulonimbus clouds. There are 14 species of clouds that describe the form and inner structure of the genus (each genus can only have one species) and 9 subspecies that describe the formation and transmittance of the clouds (more than one subspecies for each genus is possible). Detailed description of the classification of clouds

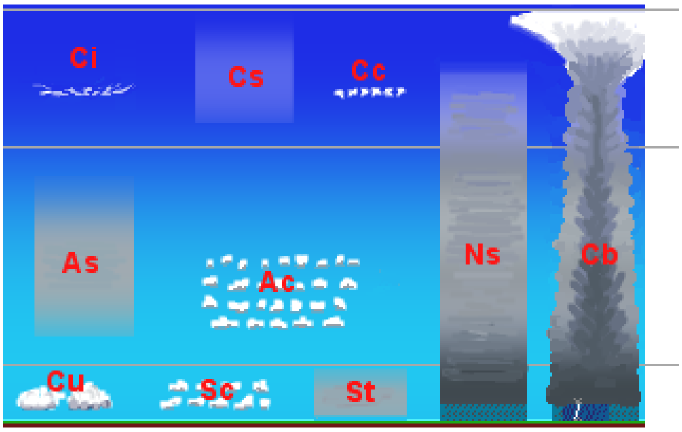


Figure 2.2: This image [58] shows the four families and ten genera of clouds.

can be found in [15].

In figure 2.3 an example for each cloud family is shown.

(a) Cirrus uncinus and fibratus [59] (high clouds).

(b) Altocumulus floccus producing virga near top and middle of image merging into altostratus translucidus near horizon [60] (middle clouds).

(c) Cumulus clouds [57] (low clouds).

(d) Cumulonimbus cloud [61] (vertical clouds).

Figure 2.3: These images show the four cloud families *high clouds* (a), *middle clouds* (b), *low clouds* (c) and *vertical clouds* (d).

2.2 Spectral properties of the atmosphere relative to absorption and scattering processes

The main source of radiation reaching the upper layers of the atmosphere is the sun. Its spectrum can be approximated by a blackbody with a temperature of about 5500 K. The black curve in figure 2.4 shows the corresponding distribution of the irradiance depending on the wavelength and the yellow area is the spectrum emitted by the sun. The amount of the solar irradiance measured on the earth's surface is altered due to different wavelength dependent absorption and scattering processes what can be seen in the red area in figure 2.4. Elastic scattering processes generally dampen the solar irradiance for all wavelengths that are interesting in optical remote sensing. These wavelengths range from the visible to the thermal infrared region (≈450 nm to ≈14 µm) [30]. This mechanism is strongly dependent on the size of the scatterer and the scattering wavelength. If the scatterer with an assumed spherical shape with diameter r is significantly smaller than the wavelength λ of the light ($r \ll \lambda$) the mechanism of *Rayleigh scattering* with a wavelength dependence of λ^{-4} comes into effect. This means that shorter wavelengths are scattered more strongly than longer wavelengths. This mechanism can be seen as elastic scattering on abundant atmospheric gases like nitrogen (N_2), oxygen (O_2) or water vapor (H_2O) with a dimension of approximately 10^{-10} m. If the scatterer reaches diameters of the size of the wavelength ($r \approx \lambda$) which is the case e.g. for smog, smoke, dust or cloud droplets the *Mie scattering* mechanism comes into effect. Clouds have an average droplet size of about 10 µm [29]. For this droplet size the scattering coefficient of light in the Visible from 0.38 µm to 0.75 µm [2] (VIS) and Near Infrared from 0.75 µm to 1.4 µm [2] (NIR) region is independent from the wavelength [55]. Larger particles like rain drops having radii of about 1 mm [29] are scattered in a classical geometrical way, i.e. each scatterer is large compared to the wavelength ($r \gg \lambda$) so that each one of them can be seen as a source of spherical radiation. This means that in the VIS and NIR region the reflectance of the incoming solar radiation is high with respect to the surface.

The second mechanism that alters the solar irradiance is resonant absorption of light. Single atoms have discrete energy gaps between two energetic states with energies E_1 and E_2 ($E_1 < E_2$). Resonant absorption of light occurs if the energy of the light E_l is

$$E_l = h\frac{c}{\lambda} = E_2 - E_1 \quad [eV]. \tag{2.2}$$

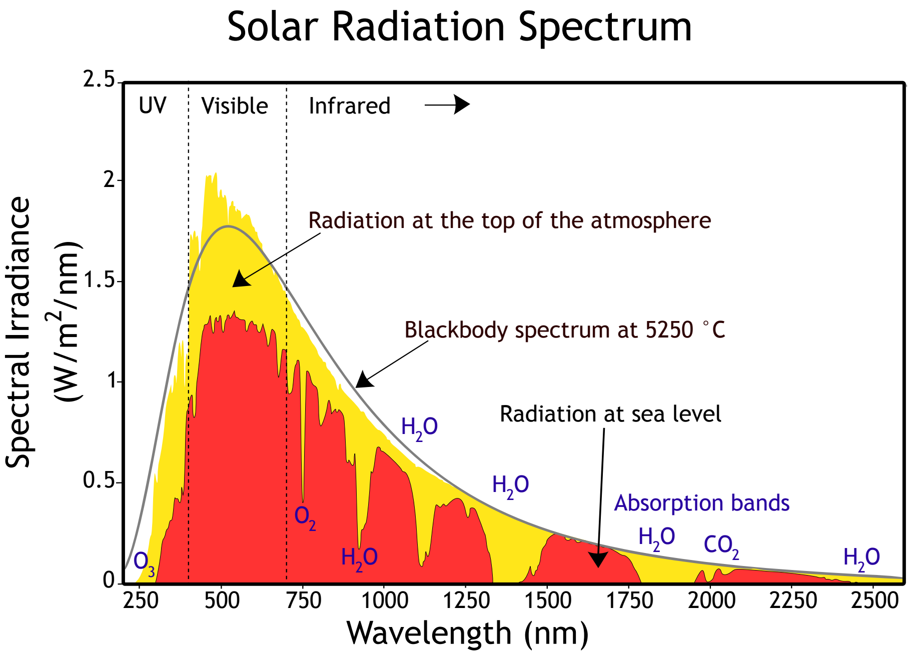


Figure 2.4: This image [56] shows the Top of atmosphere (TOA) irradiance dependent on the wavelength (yellow), the irradiance measured on the earth's surface with the characteristic absorption lines (red) and the distribution of a blackbody's irradiance with a temperature of 5250 °C (gray).

Then the electron in the atomic shell with energy E_1 is shifted in a state with energy E_2. Note that E_1 and E_2 are not arbitrary but well defined values depending on the atom. The lifetime of an *excited state* E_2 is typically about 10^{-9} s to 10^{-6} s. After this time the excited electron relaxes into the energy state E_1 by emitting a photon in arbitrary direction with the same energy $E_2 - E_1$.

For molecules more degrees of freedom are possible so that also discrete vibrational and rotational states can be excited, where $E_{el} > E_{vib} > E_{rot}$ and E_{el} is the energy difference between two electronical states. Not all the irradiating light from the sun reaches the earth's surface, a part is also absorbed by the atmosphere's abundant gases with well defined *spectral bands* as one can see in figure 2.4. In the thermal region there exists a wide band where nearly no absorption exists [30] so that this wavelength region is ideal for the examination of the thermal emission of the earth's surface.

When sun light finally reaches the earth's surface it interacts with the components of the ground, e.g. water, ice, desert or vegetation. Each component has its own fingerprint, e.g. water has a low reflectance in all visible regions, or chlorophyll as an important part of vital vegetation has low reflectance in the visible part of the spectrum but significant higher reflectance in the NIR region. These properties are well used in remote sensing in order to assign multispectral scene pixels to the correct surface class and therefore they can also be used to exclude clouds with this fingerprint.

3 Data sources

This chapter presents the data sources used in this thesis, namely the Enhanced Thematic Mapper (ETM+) and the Moderate-resolution Imaging Spectroradiometer (MODIS). The orbital properties of the platforms and the spectral and spatial resolutions of the sensors are described. In addition, a detailed description of the process of the extraction of the physical magnitudes from the delivered datasets of the sensors is given.

3.1 ETM+

The Landsat program started in 1972 with the first Landsat satellite equipped with the Multispectral scanner (MSS). Since then a continuous series of multispectral images of the earth's land surface exists. The latest member of the Landsat family, Landsat 8 was started in February 2013 and continues this successful history.

In this thesis data captured from the ETM+ sensor of the Landsat 7 satellite was used due to the large amount of already existing free data provided by the USGS Center for Earth Resources Observation and Science (EROS). Landsat 7 was launched on the 15^{th} April 1999 as a part of NASA's Earth Observation System (EOS) [19] program and is still (June 2015) operational. It has a 705 km (nominal) Sun synchronous orbit (SSO) with an orbital period of 99 minutes, an orbit inclination of 98.2°, a repeat cycle of 16 days and an equator crossing time of 10:00 a.m. It is equipped with an S-band transponder for uplinking telecommands at a rate of 4 kBit/s and downlinking housekeeping data at a rate of 256 kBit/s. The X-band transponder for downlinking data works with a capacity of 300 MBit/s. The data captured from the sensor is stored on a 378 GBit Solid state recorder (SSR) having the capacity of holding 42 minutes of instrument data (corresponding to about 100 ETM+ scenes) and 29 hours of housekeeping data. For further information refer to [40, pp. 10-18].

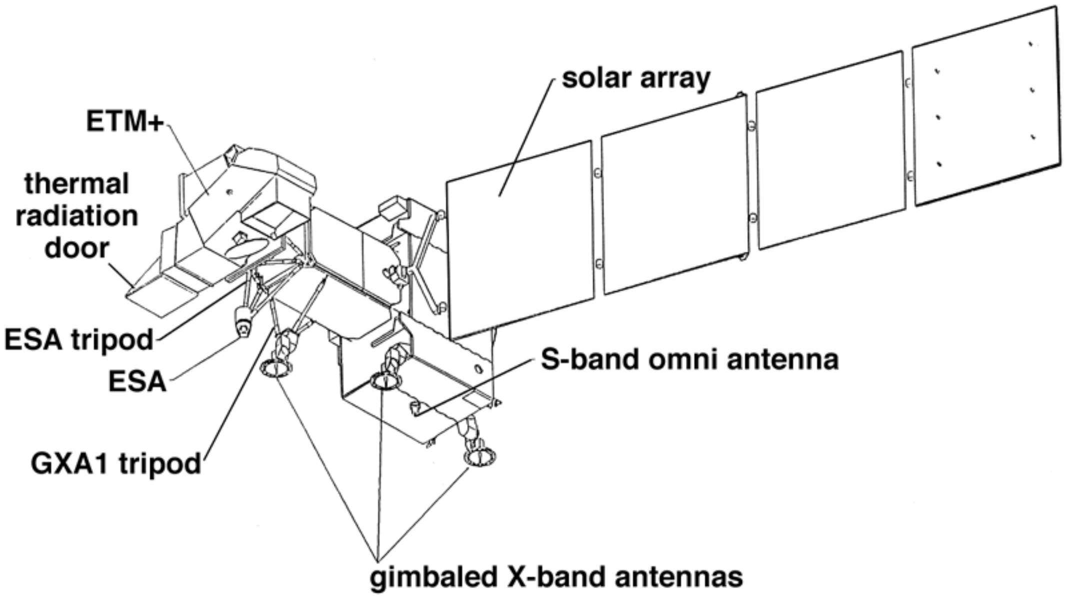


Figure 3.1: This image shows a schematic drawing of the Landsat 7 satellite [32].

The instrument on board of Landsat 7 is the Enhanced Thematic Mapper (ETM+) covering eight spectral bands from blue to the thermal region in different spatial resolutions. A summary of the sensor's spectral band properties is given in table 3.1. One standard Landsat WRS (World Reference System) scene covers approximately 185 km in across-track direction by 180 km in along-track direction. For the 30 m bands 6600 samples are taken in across-track direction and 6000 in along-track direction (for the 60 m band divide these values by 2 and for the 15 m band multiply them by 2). The design of the sensor allows to

Band	Range [μm]	Resolution [m]
1	0.45-0.52 (blue)	30
2	0.53-0.61 (green)	30
3	0.63-0.69 (red)	30
4	0.78-0.90 (NIR)	30
5	1.55-1.75 (SWIR)	30
6	10.4-12.5 (TIR)	60
7	2.09-2.35 (SWIR)	30
8	0.52-0.90 (PAN)	15

Table 3.1: This table lists the spectral channels of the ETM+ sensor with their corresponding spatial resolutions.

capture 16 lines each for bands 1-5 and 7, 8 for band 6 and 32 for band 8 per mirror sweep simultaneously so that each WRS scene captured by ETM+ consists of 480 scan lines (see [40, p. 49]). For each of the 16 lines in one scan line there is one sensor having his own calibration parameters. Additionally each spectral band has two sensitivity levels denoted as “High Gain” and “Low Gain” in order to avoid saturation of the sensor. Depending on the surface (ice, desert, water etc.) the convenient gain states are switched ([40, pp. 54-57]). For further reading refer to [40].

The operational cloud detection algorithm for the ETM+ sensor [26] is conducted by the usage of raw data, i.e. data that is neither radiometrically nor geometrically corrected. This product is delivered in the Landsat Level 0R (L0) format as one Hierarchical Data Format (HDF) container (description of this file format see [1]). Basically this file container holds the single scenes as well as ancillary data like calibration values and zenith angle (complete description see [42]).

Each scene is stored as a 6000×6600 (bands 1-5, band 7), 3000×3300 (band 6) and 12000×13200 unsigned 8 Bit integer matrix, where the value 0 is defined as a void value. However, in order to process the data one has to convert the data into physical values, i.e. TOA reflectance (bands 1-5, band 7) and at-sensor temperature (band 6). Band 8 is not considered in this thesis.

The first thing to be done is to increase the band 6 image size so that we have equal sizes for all images. This is done by pixel replication: each band 6 pixel is duplicated in across-track direction and each line is repeated [26]. This means that the sample rate is reduced by a factor 4.

Pixel replication can lead to the occurrence of *alias*-effects, i.e. the higher-frequent parts of the image in the frequency domain are shifted towards lower frequencies. This means that the information content of the image after resampling will be changed. In order to suppress these effects, a low pass filter should be applied before resampling. This was not done in this thesis for two reasons:

First, in order to perform the transformation of the image in Fourier space, the complete image needs to be available. A real-time analysis of the image would then no longer be possible.

And second, the visual quality of the cloud masks was satisfactory even with the presence of alias-effects as noted in [25].

Now the TOA radiance L_i for each band is computed by

$$L_i = \frac{Q_i - Q_{0i}}{G_i} \qquad \left[\frac{W}{m^2 sr \mu m}\right], \tag{3.1}$$

where i is the band index. Q_i is the pixel value from the image matrix, Q_{0i} is the zero-radiance bias (“ACCA_BIASES”) of band i (in counts) and G_i is the sensor responsivity of band i (“DETECTOR_GAINS”). These two values can be found in the “CPF” group of the HDF file. For convenience, the further description handles only the case for bands 1-5 and band 7 each having 16 detectors, the band 6 sensor with 8 detectors is treated similarly. There are 16 entries for each band and gain, so the gain state of the scene for each sensor has to be determined from the “Metadata-format 1” group (band 1-6) respectively “Metadata-format 2” group (band 7). As stated above there are 16 detectors with descending order (first line corresponds to detector 16 and line number 16 corresponds to detector 1, line number 17 corresponds to detector 16 and line number 32 corresponds to detector 1 and so on), whereas the first entry of the “DETECTOR_GAINS” and “ACCA_BIASES” corresponds to the first detector.

The TOA reflectance ρ_i for bands 1-5 and band 7 is related to the TOA radiance L_i as

$$\rho_i = \frac{\pi L_i d^2}{E_{0i} \cos\theta}, \tag{3.2}$$

where d is the earth-sun distance in astronomical units[1], E_{0i} is the exoatmospheric solar irradiance in band i (listed in the “CPF” group) and θ is the solar zenith angle (listed in the “Metadata-format 1” group; solar zenith = 90° − solar elevation). The at-sensor temperature T for band 6 is related to TOA radiance L_6 by

$$T = \frac{K2}{\ln(\frac{K1}{L_6} + 1)} \qquad [K], \tag{3.3}$$

---

[1] Spencer [49] proposed an approximation formula for the earth-sun distance d depending on the Day of year (DOY):

$$\gamma = \frac{2\pi(DOY - 1)}{365}$$

$$\beta = 1.00011 + 0.034221 cos(\gamma) + 0.00128 sin(\gamma) + 0.000719 cos(2\gamma) + 0.000077 sin(2\gamma)$$

$$\rightarrow d = \frac{1}{\sqrt{\beta}}$$

where $K1$ and $K2$ are calibration constants taken from the “CPF” group in the HDF file (in the data sets examined in this thesis they remained constant as K1=666.09 W/(m^2 sr µm) and K2=1282.71 K). Thereby we get a minimum temperature of 128.03 K and a maximum temperature of 347.51 K for band 6 in “Low Gain”.

However, there are some difficulties when implementing these calculations on an FPGA. The performance of operations like cosine and logarithm calculations is very poor, because a series expansion with a certain length needs to be conducted in order to achieve an adequate accuracy. Another problem is the occurrence of real numbers in the calculations which complicates the implementation of the conversion to physical magnitudes on an FPGA.

In this thesis 387 ETM+ scenes with zenith angles between 15° and 65° were examined, covering all seasons and most of surface features like desert, ice, vegetation etc. Figure 3.2 gives an overview of global and seasonal distribution of all scenes. With each scene having 3.96×10^7 pixels we have an overall pixel number of 1.5×10^{10} pixels.

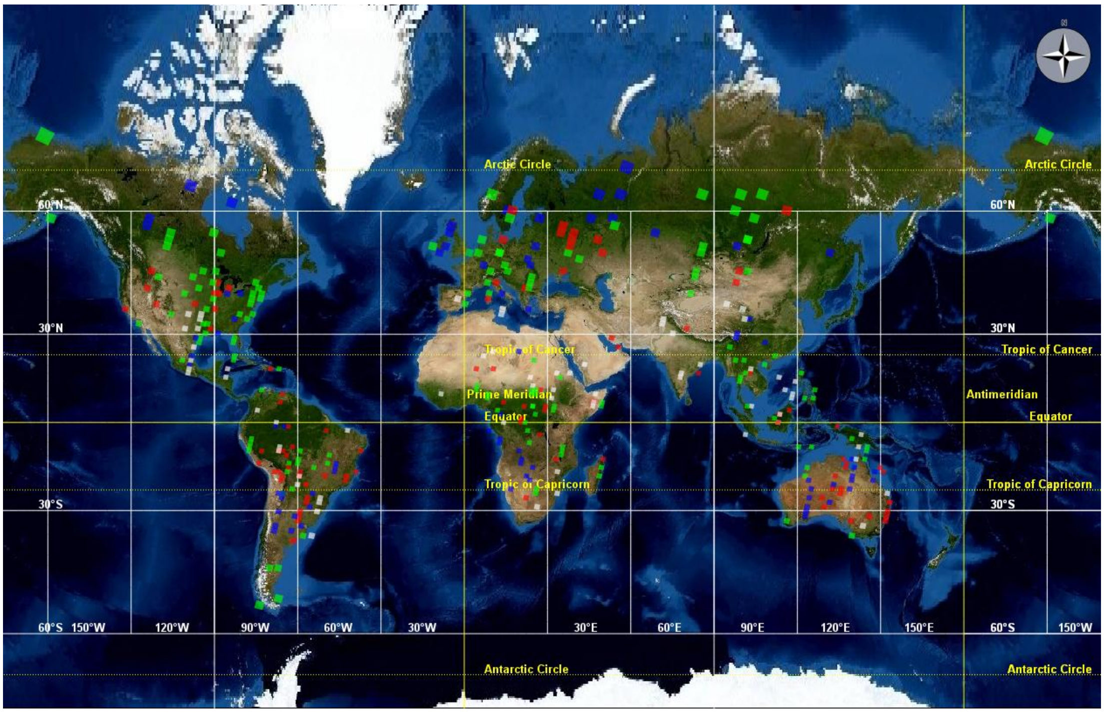


Figure 3.2: This map shows the 387 ETM+ scenes examined in this thesis. Green squares mark images that were acquired in spring, blue ones in summer, red ones in autumn and white ones in winter.

3.2 MODIS

The Moderate-resolution Imaging Spectroradiometer (MODIS) is an instrument of both the "Terra" (launched on the 18th December 1999) and the "Aqua" (launched 4th May 2002) satellites which are still operational (June 2015). Like Landsat 7, those satellites are also part of NASA's EOS [19] program. Both satellites have a 705 km (nominal) SSO with an orbital period of 99 minutes and an orbit inclination of 98.5° ("Terra") and 98.2° ("Aqua"). The equator crossing for "Terra" is at 10:30 a.m. (descending node) and for "Aqua" it is at 1:30 p.m. (ascending node) [6, 51].

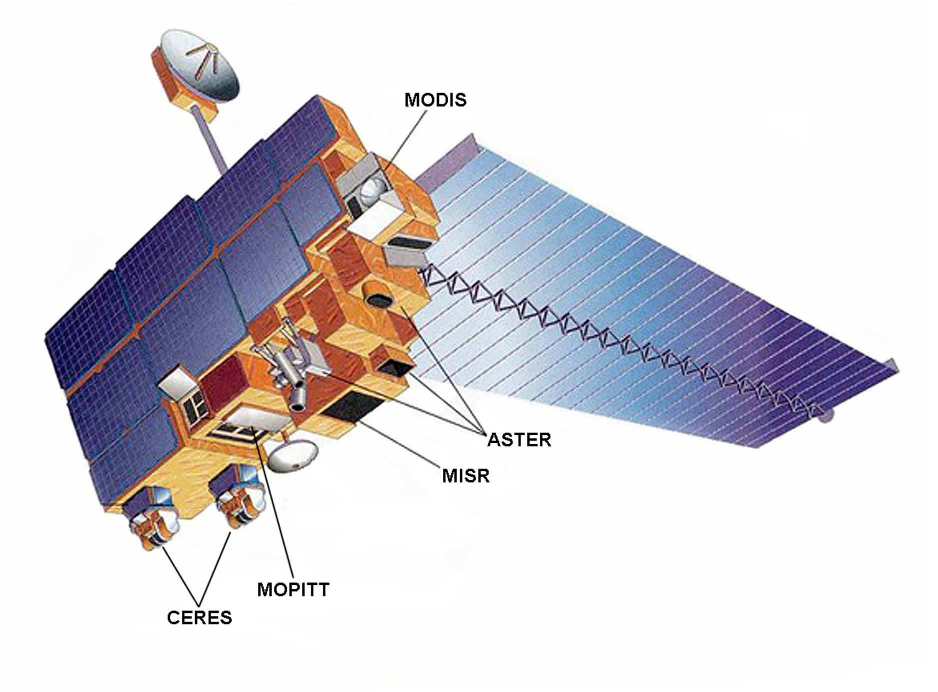


Figure 3.3: This image shows a schematic drawing of the “Terra” satellite with the different experiments [41].

MODIS was designed to gather information about large-scale geodynamic processes like changes in the cloud coverage [27] and the radiation budget [10], but is also capable of delivering information about vegetation on land [23] and oceans [20]. It has 36 spectral bands ranging from 0.47 µm to 14.2 µm (19 reflective, 17 thermal) with spatial resolutions of 250 m (bands 1 and 2), 500 m (bands 3-7) and 1 km (bands 8-36). Table 3.2 lists all MODIS spectral bands and marks the bands used for the MODIS cloud detection algorithm [4] in grey. Having a swath width of about 2330 km, the whole earth surface is imaged every two days. MODIS has a high radiometric sensitivity of 12 Bit (ETM+: 8 Bit) which are rescaled to 16 Bit during level 1 processing. In contrast to the ACCA algorithm, the MODIS cloud detection algorithm is conducted on radiometrically and geometrically corrected data (level 1B [52]). This product is also delivered as HDF container (ECS name: “MOD021KM” for the “Terra” satellite and “MYD021KM” for the “Aqua” satellite). In this container the image data is stored as unsigned 16 Bit values, where the range $[0, ..., 32767]$ is reserved for valid data.

The data for the reflective channels can be found in the fields “EV_250_Aggr1km_RefSB”

Band	Central Wavelength (μm)	Band	Central Wavelength (μm)
1	0.659	20	3.750
2	0.865	21	3.959
3	0.470	22	3.959
4	0.555	23	4.050
5	1.240	24	4.465
6	1.640	25	4.515
7	2.130	27	6.715
8	0.415	28	7.325
9	0.443	29	8.550
10	0.490	30	9.730
11	0.531	31	11.030
12	0.565	32	12.020
13	0.653	33	13.335
14	0.681	34	13.635
15	0.750	35	13.935
16	0.865	36	14.235
17	0.905		
18	0.936		
19	0.940		
26	1.375		

Table 3.2: Overview of the spectral bands of the MODIS sensor. On the left side the reflective bands and on the right side the emissive bands are shown. Spectral bands used in the MODIS cloud cover algorithm [4] are highlighted in grey.

(bands 1 and 2; dimension 2 × 2030 × 1354), “EV_500_Aggr1km_RefSB” (bands 3-7; dimension 4 × 2030 × 1354) and “EV_1KM_RefSB” (bands 8-19 and 26; dimension 15 × 2030 × 1354) in the node “MODIS_SWATH_TYPE_L1B/Data Fields” of the .hdf container. Note that bands 1 to 7 are rescaled to a resolution of 1 km instead of 250 m respectively 500 m. Due to the fact that the data is radiometrically calibrated the reflectance can be directly calculated by

$$\rho_i = G_i(Q_i - Q_{0i}), \tag{3.4}$$

where Q_i is the integer value of the image pixel from the data matrix. G_i is denoted as *reflectance_scales* and Q_{0i} is named *reflectance_offsets* in the meta data of the “EV_250_Aggr1km_RefSB”, “EV_500_Aggr1km_RefSB” and

“EV_1KM_RefSB” field. These are strings with n entries, where n is the number of spectral bands of each data field.
The data of the thermal spectral bands are found in the field “EV_1KM_Emissive”. The radiance of each image pixel for band i is given by

$$L_i = S_i(Q_i - R_{0i}) \qquad \left[\frac{W}{m^2 sr \mu m}\right], \tag{3.5}$$

where S_i is the scaling factor of band i (denoted as *radiance_scales*) and R_{0i} is the corresponding offset (denoted as *radiance_offset*). In analogy to the reflective case, these values can be found in the corresponding meta data.

The “Cloud” mask (ECS name “MOD35_L2”) is coded in 48 Bit in the node “mod35/Data Fields”, field “Cloud_Mask” (matrix with 2030 × 1354 entries) [50]. We only have to evaluate Bit 1 and 2 where “00” means “cloudy”, “01” means “uncertain clear”, “10” means “probably clear” and “11” means “clear”. Since we only want a two class classification, the first two entries are merged as “Cloud” and the last two as “Cloud-Free” .

In this thesis 29 MODIS scenes were examined, covering all seasons and most surface features like desert, water, vegetation etc. Figure 3.4 shows the global and seasonal distribution of all MODIS scenes used in this thesis. Each scene having 2748620 pixels we have an overall pixel number of about 7.9×10^7 pixels.

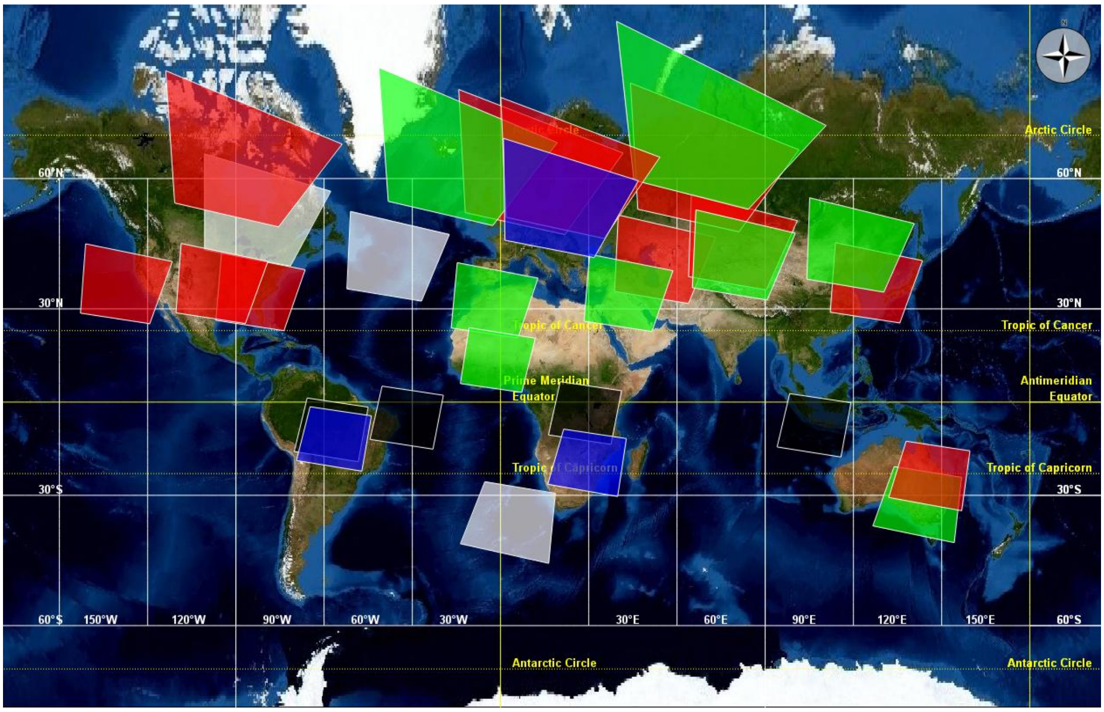


Figure 3.4: This map shows the 29 MODIS scenes examined in this thesis. Green squares mark images that were acquired in spring, blue denotes summer, red means autumn, white stands for winter and the opaque ones are covering the equator so that a clear seasonal mapping is not feasible.

4 Basics of FPGAs

This chapter gives a brief introduction into the setup of Field Programmable Gate Arrays (FPGAs) and the specific technical properties of the chip used in this thesis.

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that can be configured in a way that a large variety of digital logic can be mapped. In comparison to Application Specific Integrated Circuits (ASICs) they have some advantages like simple prototyping and therefore low development costs because the ASIC does not have to be build in the first way what would need a complete processing line. Also bugfixing and even complete new configuration is possible without changing the hardware, at least for SRAM-based and flash-based FPGAs. On the other hand there are also some disadvantages of FPGAs: due to their generic composition an FPGA needs about 35 times more area than a standard cell ASIC, is about 3 to 4 times slower than an ASIC and needs roughly 10 times more power [31].

An FPGA is generally built as a matrix of Configurable Logic Blocks (CLBs), embedded in a net of wires that are used for connecting the CLBs as required. In addition, we can also find specialized elements like multipliers or storage modules (Block-RAM) on the chip that can also be connected to the CLBs. Finally we need I/O-ports to send and receive the data from the outside, e.g. an RS-232 interface or an external RAM. This principal setup is visualized in figure 4.1. For Xilinx Spartan-3 FPGAs one CLB consists of 4 slices [65, p. 203] where each slice consists in principle of two Flip-Flops and two Look-up tables (LUTs) with four inputs and one output [64, p. 22]. These LUTs provide truth tables that can realize any Boolean function with the four input signals [28, p. 211]. Note that the results of the functions do not have to be calculated but are instantly available due to the usage of LUTs. With the interconnection of slices there is theoretically nearly no limitation in the complexity of the logical functions, but in practice not every mapping will

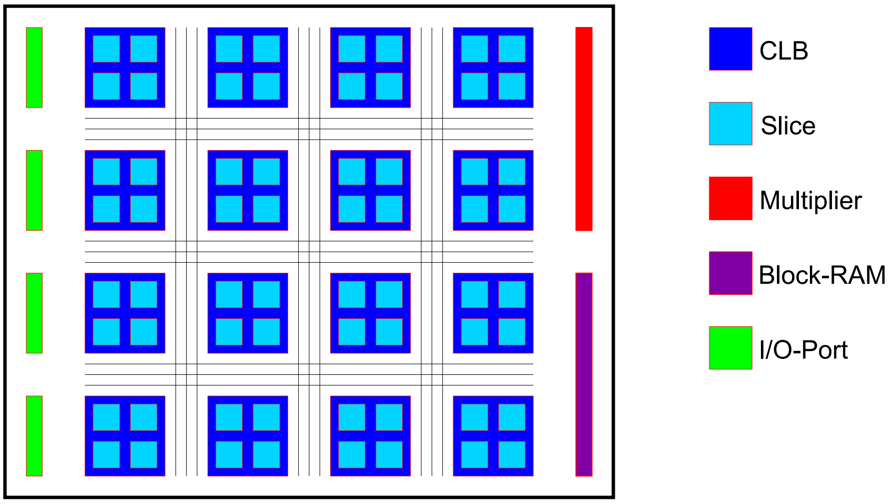


Figure 4.1: This image shows the principal setup of an FPGA. Its core elements are the slices that contain the logic elements, mainly two LUTs and two Flip-Flops per slice. Four slices are grouped in one Configurable Logic Block (CLB). Additionally hardware multipliers and Block-Random access memory (RAM) elements are present. Finally there are I/O-ports in order to provide contact to the data outside the chip.

work due to signal propagation time conflicts or the limited number of wires connecting the slices. For further reading please refer to e.g. [28].
Implementation of the cloud detection algorithms was performed by using the high-level programming language "Handel-C" from Mentor Graphics. This C-like language allows fast implementation of complex algorithms in hardware due to the high level of abstraction.

The FPGA chip used in this thesis was a "Xilinx Spartan-3E-1200 FG320" on the "Digilent Nexys2" development board. This chip contains 8672 slices grouped in 2168 CLBs. For complex multiplications 28 multipliers are available [64, p. 2]. For caching data there are 516 096 Bits of Block-RAM in 28 RAM modules [65, p. 154].
The board itself provides 16 MB of SDRAM, an RS232 interface and an external 50 MHz oscillator [16]. The clock rate was reduced to 12.5 MHz because the RAM access time is limited to 80 ns.

5 Algorithms

This chapter describes well established algorithms that are used for cloud detection in this thesis. Threshold algorithms are explained in the first part of this chapter. Afterwards two representatives of supervised learning algorithms are presented: Linear discriminant analysis and Support vector machines. Both are trained on well defined datasets where each n-dimensional pixel vector $\mathbf{x}$ *has an explicit class label "Cloud" or "Non-Cloud" . In this training the coefficients* $\mathbf{w}$ *and* b *of the separating hyperplane* $g(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$ *are estimated that optimally divides the two classes. With this hyperplane (or classifier) one can assign pixels with unknown class label to the correct class.*

5.1 Threshold algorithms

This chapter deals with the method of finding clouds in multispectral images by applying a sequence of threshold tests for different spectral bands. This approach is the most classical way to assign pixels to the classes "Clouds" or "Non-Clouds" (see [4, 26, 46]). Its great advantage is the high performance (only if/else statements) and the possibility of adding more statements if possible. On the other hand it is quite difficult to set the thresholds correctly; this task needs a lot of fine tuning and experience.

The first successful approach for automatic cloud detection was the analysis of the spectral behaviour in the visible bands of clouds respectively ground features and in the thermal region, simply assuming that clouds are bright in the visible/near infrared region and colder than the underlying surface in the thermal region [47]. In the further development cameras with higher spectral and spatial resolution were developed. This allows to develop more sophisticated threshold constraints like comparing ratios of certain spectral bands to determine e.g. desert areas or snow covered areas (see chapter 5.1.1). The quality of the resulting cloud mask can be further improved by adding a Digital elevation model (DEM) (see chapter 5.1.2) or analyzing the distribution of the cloud temperatures (see chapter 5.1.1) [40].

The first part of this chapter covers the operational cloud detection algorithm for Landsat 7 and second part gives a brief introduction into the operational cloud detection algorithm for MODIS.

5.1.1 ACCA

A primary goal of the Landsat 7 mission is to provide a global and seasonal archive of cloud-free imagery over the Earth's landmasses [26]. Therefore, the Advanced Cloud Cover Assessment (ACCA) algorithm was developed in order to evaluate the ETM+ scenes prior to archiving. This task is performed by the Landsat Processing System (LPS) at the EROS Data Center (EDC) in Sioux Falls, USA. In order to satisfy the Long-term Acquisition Plan (LTAP) mission requirements [7] a high accuracy is desired from a complete automated system like the ACCA algorithm.

This algorithm is an advancement of the cloud cover algorithm of the foregoing Thematic Mapper (TM) sensor being capable due to increased processing power to analyze each complete scene twice with the usage of bands 2 (green), 3 (red), 4 (Near Infrared from 0.75 µm to 1.4 µm [2], abbreviated with NIR), 5 (Short Wave Infrared from 1.4 µm to 3 µm [2], abbreviated with SWIR) and 6 (Thermal Infrared from 8 µm to 15 µm [2], abbreviated with TIR).

This chapter gives a brief introduction to the ACCA algorithm which is operational since 1999 [26]. The detailed description can be found in appendix A. This algorithm is divided into four parts [26]:

1. Pass-1 Spectral Cloud Identification
2. Band-6 Cloud Signature Development
3. Pass-2 Thermal Band Cloud Separation
4. Image-Based Cloud-Cover Assignments and Aggregation

Pass-1 Spectral Cloud Identification

In "Pass-1", the L0 data (see chapter 3.1) of each 5-dimensional scene pixel is examined by using different threshold tests for each band or band combination. This involves e.g. a brightness test (clouds in the visible region are highly reflective), a test if the scene pixel is covered with snow with the NDSI (Normalized Difference Snow Index), a temperature threshold which takes advantage of the fact that clouds are normally colder than the underlying surface and tests if there is vegetation or bare soil on the underground. The thresholds are chosen empirically [26] in a way that each scene pixel can be classified either as "Non-Cloud", "Ambiguous" (if the outcome of the filter is not definitely clear) or as "Cloud". Cloudy pixels are additionally separated into "Cold-Clouds" or "Warm-Clouds" by a composite filter expressed as

$$\text{Band } 5/6 \text{ Composite} = (1-\rho_5)\cdot T_6. \tag{5.1}$$

Pixels that have a Band 5/6 Composite above 210 K are labeled as "Warm-Clouds", below 210 K as "Cold-Clouds". This filter uses the fact that higher clouds like cirrus are usually colder and have higher reflectance in the SWIR region (band 5) so that these two classes can be separated empirically.

For further processing the presence of snow in the scene is examined. If its percentage is less than 1 % the scene is declared snow-free. If snow is present, only the "Cold-Cloud" pixels are used as resulting cloud mask in "Pass-1"; "Warm-Clouds" found in "Pass-1" are relabeled as "Ambiguous". In the next step cloud temperature statistics are calculated, both for all "Cold-Clouds" and the combined "Cold-Clouds" and "Warm-Clouds" case. Here, the cloud's population temperature minimum, maximum and mean value as well as its standard deviation and skewness are calculated. "Pass-2" is triggered only when the image lacks highly illuminated rocks or sand (see figure A.3), the "Cold-Cloud" percentage exceeds 0.4 % (a lower percentage can be seen as "Cloud-Free") and the mean temperature for the clouds is less than 295 K (values higher than 295 K mean that too many clouds might be found in "Pass-1"). In the case of desert conditions only the "Cold-Cloud" population from "Pass-1" is used and the "Warm-Clouds" are relabeled as "Ambiguous".

Band-6 Cloud Signature Development

In order to classify the pixels labeled as "Ambiguous" into the "Warm-Cloud" or "Cold-Cloud" class, two new temperature thresholds are introduced. The lower threshold is used to distinguish between "Cold-Clouds" and "Warm-Clouds". All pixels with temperatures higher than the upper threshold are empirically considered as "Non-Cloud". The lower and upper threshold are set to the 83.5 and 97.5 percentile of the cloud temperature distribution histogram, respectively. If the skewness is positive threshold adjustments have to be made in order to compensate for the tendency towards "Warm-Clouds". This adjustment is calculated by multiplying the skewness with the standard deviation and then added to the existing thresholds. Finally, if the new upper threshold exceeds the 98.75 percentile, its value is set to the 98.75 percentile.

Pass-2 Thermal Band Cloud Separation

"Pass-2" examines each "Ambiguous" pixel from "Pass-1". If the temperature value of the pixel falls below the upper threshold it is labeled as "Warm-Cloud", furthermore if it falls below the lower threshold it is labeled as "Cold-Cloud". Pixels exceeding the upper threshold are labeled as "Non-Cloud" (see above). After processing "Pass-2", the temperature means and maximums as well as their contribution to the cloud coverage of the scene (called *thermal effect*) for the newly found "Cold-Cloud" population and the combined "Cold-Cloud" and "Warm-Cloud" population are calculated.

Image-Based Cloud-Cover Assignments and Aggregation

Now the two cloud masks from "Pass-1" and "Pass-2" are combined. Therefore, the plausibility of the thermal effects is examined. If the combined thermal effect from "Pass-2" exceeds 35 % or the mean cloud temperature is greater than 295 K probably too many misclassified clouds were found. In this case, or if snow is present, the "Warm-Clouds" are relabeled as "Non-Cloud". In the next step the contribution of the "Cold-Clouds" found in "Pass-2" is compared to a threshold of 25 %. If the thermal effect of these clouds is greater than this value this class is again probably misclassified and only the "Cold-Clouds" from "Pass-1" are used as cloud mask, otherwise both "Cold-Cloud" pixels are used.

Finally, a neighborhood filling with a window of 3 × 3 pixels is performed in order to fill single holes in the cloud mask that will probably be misclassified.

5.1.2 MODIS Cloud Mask Algorithm

The MODIS "Cloud" mask algorithm [4] developed by the International Satellite Cloud Climatology Project (ISCCP) is an advancement of the algorithms for the Advanced Very High Resolution Radiometer (AVHRR) sensor [46], operational since the start of the "Terra" satellite in 1999. It uses as input the calibrated and geometrically corrected MODIS reflectance and temperature values of channels 1, 2, 4-7, 17-20, 22, 26-29, 31-33 and 35 (see table 3.2). Additionally the following ancillary data is needed:

- sun angle, azimuthal angle and viewing angle from MOD03 (geolocation fields)
- land/water map at 1 km resolution from MOD03
- topography: elevation above sea level from MOD03
- ecosystems: global 1 km map of ecosystems based on the Olson classification system
- daily NISE snow/ice map provided by NSDIC
- daily sea ice concentration product from NOAA

The data is subjected to different threshold tests for each scene pixel. Due to the fact that three thresholds are given for each test four outcomes are possible: *clear, probably clear, probably cloudy* and *cloudy*. Merging this data with the ancillary data gives the final "Cloud" mask. The disadvantage of this algorithm for on-board processing is its dependence on the ancillary data. A global land/water mask has approximately the size of 63.8 MB, the DEM has a size of about 893 MB and the ecosystem map has approximately the size of 1.8 GB (MODIS product MOD12Q1 extrapolated to the whole earth surface) which consumes a large amount of on-board memory, but in today's terms this could be neglected. But the snow/ice map as well as the sea ice concentration product must be available and therefore updated every day which means that this data has to be sent to the satellite and therefore consume uplink capacities. Hence, this algorithm is not capable for on-board processing of a cloud mask.

5.2 Supervised learning algorithms

5.2.1 Linear Discriminant Analysis

The following section gives a brief introduction to the idea of classifying data with the help of the Bayes' theorem (mainly adapted from [17, pp. 20-41, pp. 84-90]). For this thesis only the two-class case is considered (namely "Clouds" and "Non-Clouds"). For the multi-class case please refer to [17].

Having an input vector $\mathbf{x}$ from the input space $\mathbf{X} \subset \mathbb{R}^n$ (n is the number of features like spectral channels), we want to assign this vector to one class referred to as $\omega \in -1, 1$. With the *prior* knowledge of the probability $P(\omega_i)$, $i = 1, 2$ we can express the improved *posterior* probability $P(\omega_i|\mathbf{x})$ in terms of the Bayes' theorem

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})}. \tag{5.2}$$

The expression $p(\mathbf{x}|\omega_i)$ is called the *likelihood* of ω_i with respect to $\mathbf{x}$. This means assuming one class ω_i how likely is it that $\mathbf{x}$ fits this class.
The denominator $p(\mathbf{x})$ can be expressed as

$$p(\mathbf{x}) = \sum_{i=1}^{2} p(\mathbf{x}|\omega_i)P(\omega_i) \tag{5.3}$$

and is called *evidence*. This term expresses the weighted mean of the likelihoods if every summand is weighted with the probability of its occurrence. It therefore is a scaling factor that normalizes the posterior probability.
A decision rule can now be expressed as

Decide ω_1 if $p(\mathbf{x}|\omega_1)P(\omega_1) > p(\mathbf{x}|\omega_2)P(\omega_2)$, otherwise decide ω_2.

The Bayes formulation of the classification problem is a special case of the classification problem that can be expressed in terms of discriminant functions $g_i(\mathbf{x})$, $i = 1, 2$. The classifier assigns the class ω_1 to a feature vector $\mathbf{x}$ if

$$g_1(\mathbf{x}) > g_2(\mathbf{x}), \tag{5.4}$$

otherwise to class ω_2. So the Bayes discriminant functions can be written as

$$g_i(\mathbf{x}) = p(\mathbf{x}|\omega_i)P(\omega_i) \tag{5.5}$$

or alternatively as

$$\tilde{g}_i(\mathbf{x}) = \ln p(\mathbf{x}|\omega_i) + \ln P(\omega_i). \tag{5.6}$$

The last two expressions yield the same result due to the fact that the logarithm is a monotonically increasing function. Combining equation (5.4) and equation (5.6) we get

$$g(\mathbf{x}) = \ln \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}, \tag{5.7}$$

assigning $\mathbf{x}$ to ω_1 if $g(\mathbf{x}) > 0$, otherwise to ω_2.

If no information is available about the distribution of the underlying data a multivariate Gaussian distribution is used as a starting point for the likelihood function. It has the advantage of analytical tractability. Here the feature vector $\mathbf{x}$ for a given class ω_i is normally scattered around a prototype vector $\boldsymbol{\mu}_i$. The multivariate density in n dimensions is written as

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}} e^{\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right]}, \tag{5.8}$$

where $\mathbf{x}$ is an n-dimensional feature vector, $\boldsymbol{\mu}_i$ is the n-dimensional mean vector, $\mathbf{\Sigma}_i$ is the $n \times n$ covariance matrix, and $|\mathbf{\Sigma}_i|$ and $\mathbf{\Sigma}_i^{-1}$ are its determinant and inverse respectively. Vectors and matrices with superscripted t refer to their transposed representation.

Combining equations (5.7) and (5.8) gives

$$
\begin{aligned}
g(\mathbf{x}) =\ & \ln\left(\frac{|\boldsymbol{\Sigma}_2|^{1/2}}{|\boldsymbol{\Sigma}_1|^{1/2}} e^{\left[\left(\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^t\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right)-\left(\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^t\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)\right]}\right) \\
& + \ln\frac{P(\omega_1)}{P(\omega_2)} \\
=\ & \frac{1}{2}\ln|\boldsymbol{\Sigma}_2| - \frac{1}{2}\ln|\boldsymbol{\Sigma}_1| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^t\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^t\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \\
& + \ln\frac{P(\omega_1)}{P(\omega_2)} \\
=\ & \frac{1}{2}\left(\mathbf{x}^t(\boldsymbol{\Sigma}_2^{-1}-\boldsymbol{\Sigma}_1^{-1})\mathbf{x}\right) + \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)^t\mathbf{x} + \frac{1}{2}\left(\boldsymbol{\mu}_2^t\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^t\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\right) \\
& + \frac{1}{2}\ln|\boldsymbol{\Sigma}_2| - \frac{1}{2}\ln|\boldsymbol{\Sigma}_1| + \ln\frac{P(\omega_1)}{P(\omega_2)}.
\end{aligned}
\tag{5.9}
$$

Assuming equal covariance matrices ($\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$) the first summand in eq. (5.9) can be dropped so that the evaluation of the computationally intensive quadratic expression is not necessary. The decision function simplifies to

$$
g(\mathbf{x}) = \left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\right)^t\mathbf{x} + \frac{1}{2}\left((\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2+\boldsymbol{\mu}_1)\right) + \ln\frac{P(\omega_1)}{P(\omega_2)}. \tag{5.10}
$$

This can be formulated in an elegant way as

$$
g(\mathbf{x}) = \langle\mathbf{w}\cdot\mathbf{x}\rangle + b, \tag{5.11}
$$

where

$$
\mathbf{w} = \left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\right)^t, \tag{5.12}
$$

$$
b = \frac{1}{2}\Big((\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2+\boldsymbol{\mu}_1\Big)) + \ln\frac{P(\omega_1)}{P(\omega_2)} \tag{5.13}
$$

and $\langle\cdot\rangle$ denotes the standard scalar product. Classifying data with this method is referred to as LDA.

The performance of the quadratic and the linear decision functions have been compared in [5], showing that a linear approach is quite sufficient for a qualitative generation of a cloud mask. The advantage is that $n \times n$ summands can be dropped and therefore only n

summands have to be evaluated for the decision function. This means that theoretically one scene pixel can be processed per clock cycle with an FPGA without large hardware consumption.

In practical use, neither $\boldsymbol{\mu}_i$ nor $\boldsymbol{\Sigma}_i$ are known but have to be estimated. Let us separate the training data into two sets T_1 and T_2 each one belonging to one class. All samples from T_i were drawn independently according to the probability law $p(\mathbf{x}|\omega_i)$. We now make the assumption that each sample from T_1 gives no information about $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_2$, and vice versa.
Suppose that T_i contains m samples $\mathbf{x}_{1_i}, ..., \mathbf{x}_{m_i}$. The samples are drawn independently, so

$$p(T_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \prod_{k=1}^{m} p(\mathbf{x}_{k_i}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{5.14}$$

Comparing this equation to eq.(5.2) we can see that $p(T_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is the likelihood of $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with respect to the training set. Intuitively we seek the best estimate $(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)$ that maximizes the likelihood function $p(T_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Due to the fact that the logarithm is a monotonically increasing function the set $(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)$ which maximizes the likelihood also maximizes the log-likelihood function

$$l(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) := \ln p(T_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{5.15}$$

The problem can now be expressed as

$$(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) = \arg \max_{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} l(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{5.16}$$

In terms of eq.(5.14) the likelihood function can now be written as

$$l(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \sum_{k=1}^{m} \ln p(\mathbf{x}_{k_i}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{5.17}$$

Obviously we find a solution of eq.(5.16) by calculating the partial derivatives of $l(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ in respect to $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ and setting them to 0. The derivation of the results as well as the proof of being a global maximum exceeds the scope of this thesis and can be looked up in e.g. [44]. As a result we get

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{m} \sum_{k=1}^{m} \mathbf{x}_{k_i} \tag{5.18}$$

and

$$\hat{\boldsymbol{\Sigma}}_i = \frac{1}{m} \sum_{k=1}^{m} (\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_i)^t. \tag{5.19}$$

It can be easily seen that the maximum likelihood estimate for the mean vector is the sample mean and the maximum likelihood estimate for the covariance matrix is the arithmetic average of the m matrices $(\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_{k_i} - \hat{\boldsymbol{\mu}}_i)^t$, very similar to the expected value of the covariance matrix $(\mathbf{x} - \hat{\boldsymbol{\mu}}_i)(\mathbf{x} - \hat{\boldsymbol{\mu}}_i)^t$.
In order to satisfy the assumption of equal covariance matrices of both classes we have to combine the estimated covariance matrices by

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{m_1 + m_2} (m_1 \hat{\boldsymbol{\Sigma}}_1 + m_2 \hat{\boldsymbol{\Sigma}}_2). \tag{5.20}$$

This is called *pooled covariance matrix.*

5.2.2 Support Vector Machines

Another approach to separate binary data by supervised learning are Support Vector Machines (SVMs) which were introduced by V. Vapnik in his works on machine learning [53]. This chapter gives a short description of the theory of SVMs (adapted from mainly [14]), for further reading please refer to [14]. Like LDA we seek an optimal hyperplane that separates the data into two classes in the *primal* form

$$g(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b, \tag{5.21}$$

where $\mathbf{w}$ is the n-dimensional vector normal to the hyperplane (n is the dimension of the input space, e.g. number of spectral bands), b is the distance to the origin and $\langle \cdot \rangle$ denotes the inner product of $\mathbf{w}$ and $\mathbf{x}$. Equation (5.21) can also be represented in its equivalent *dual* form

$$g(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j y_j \langle \mathbf{x}_j \cdot \mathbf{x} \rangle + b, \tag{5.22}$$

where m is the number of training points $\mathbf{x}_j$, y_j their corresponding class labels and α_j together with b the coefficients that will define the best separating hyperplane (see figure 5.1). The problem is now to find the coefficients α_j for the training vectors $\mathbf{x}_j$ and the distance from the origin b.

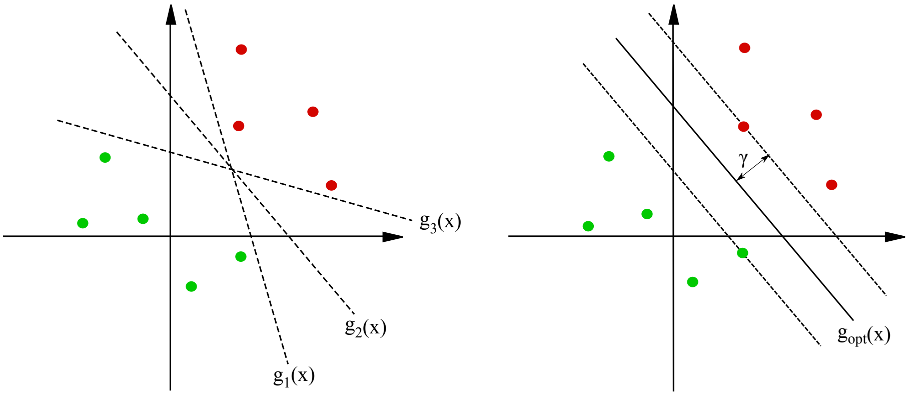


Figure 5.1: The left image shows a linear separable distribution of data with two classes. It is possible to find infinite solutions for a hyperplane, but the best choice is the one where the geometrical margin γ is maximal (right image). Image adapted from [48]

.

First we examine the *linearly separable* case, i.e. the training data can be separated exactly into two classes by a hyperplane. The classifier which performs this task is called *maximal margin classifier.*

As can be seen in figure 5.1 we aim to maximize the *geometric margin* γ. This ensures to find the optimal separating hyperplane among all possible ones. Without loss of generality we can rescale $\mathbf{w}$ and b so that for the two optimal boundary hyperplanes (dotted lines in figure 5.1(b)) the following two equations hold:

$$\langle \mathbf{w} \cdot \mathbf{x}_+ \rangle + b = +1 \tag{5.23}$$

$$\langle \mathbf{w} \cdot \mathbf{x}_- \rangle + b = -1, \tag{5.24}$$

where $\mathbf{x}_+$ resp $\mathbf{x}_-$ are points that lie on the corresponding boundary hyperplane. The geometric margin γ is calculated by geometrical considerations. The point $\mathbf{x}_s$ lies on the surface of the hyperplane nearest to a point $\mathbf{x}_+$ for $y = +1$ on the corresponding boundary surface. Then the Euclidian distance of the two points is

$$\mathbf{x}_s = \mathbf{x}_+ - y\gamma \frac{\mathbf{w}}{||\mathbf{w}||}. \tag{5.25}$$

The requirement is that $\mathbf{x}_s$ lies on the surface of the best separating hyperplane, so

$$\langle \mathbf{w} \cdot \mathbf{x}_s \rangle + b = 0. \tag{5.26}$$

Combining equations (5.25) and (5.26) we get

$$\gamma = \frac{y\langle \mathbf{w} \cdot \mathbf{x}_+ \rangle + b}{||\mathbf{w}||}. \tag{5.27}$$

By definition (see (5.23)) we have $\langle \mathbf{w} \cdot \mathbf{x}_+ \rangle + b = 1$ and hence

$$\gamma = \frac{1}{||\mathbf{w}||}. \tag{5.28}$$

The aim of the SVM classifier can then be formulated as maximizing the geometric margin γ expressed in terms of the training vectors $\mathbf{x}_j$. The minimal subset of the training vectors $\mathbf{x}_+$ and $\mathbf{x}_-$ that are sufficient to describe the corresponding separating hyperplane are called *support vectors.*

The solution is found by *optimization theory* which exceeds the scope of this thesis (see

[14] for a detailed introduction). We obtain

$$\mathbf{w} = \sum_{j \in SV} y_j \alpha_j \mathbf{x}_j \tag{5.29}$$

and

$$b = y_k - \langle \mathbf{w} \cdot \mathbf{x}_k \rangle \quad \text{for any } k \in SV. \tag{5.30}$$

Note that the sum in (5.29) only extends over the vectors that determine the separating hyperplane; the solution of the optimization problem sets all α_j to 0 which do not contribute to the definition of the hyperplane.

In real life situations the training data will normally not be perfectly separable due to noise or training errors. The resulting hyperplane, if it exists, will then not give the optimal solution because outliers due to a misclassified sample can crucially affect the hyperplane. Therefore a slack variable $\boldsymbol{\xi}$ is introduced which allows the violation of the margin constraints and leads to the so-called *soft margin SVM* (see figure 5.2). This variable is then an additional parameter in the optimization problem. The *cost factor* C ($C > 0$) regulates the influence of the slack variable and can be seen as control parameter for the penalty function. The larger the cost factor the higher is the penalty for a misclassified training vector, but also the risk of overfitting raises with a high cost factor.

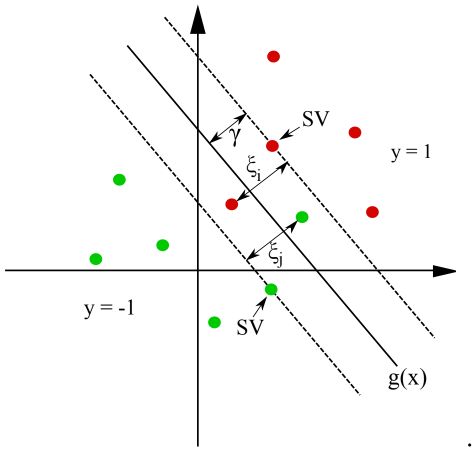


Figure 5.2: This image shows noisy training data with two classes -1,1. If the slack variables ξ are grater than the geometrical margin γ, the corresponding training point is misclassified and the magnitude of ξ is a measure of it's misclassification. For the description of the separating hyperplane, only the points next to the hyperplane are relevant; they are here marked as “SV” (support vectors). Image adapted from [48]

Note that we can generalize the expression $\langle \mathbf{x}_j \cdot \mathbf{x} \rangle$ in equation (5.22) to

$$\langle \boldsymbol{\phi}(\mathbf{x}_j) \cdot \boldsymbol{\phi}(\mathbf{x}) \rangle =: k(\mathbf{x}_j, \mathbf{x}), \tag{5.31}$$

where $\boldsymbol{\phi}(\mathbf{x})$ is a non-linear mapping from the input space to some *feature space*. The expression $\boldsymbol{\phi}(\mathbf{x})$ is generally unknown and not important since only the dot product is used for the evaluation. The expression in equation (5.31) is called *kernel function*. With a convenient transformation it can be possible to find a linear hyperplane in the feature space which allows to separate data which is not linearly separable in the input space. If we have a model for the data, the mapping can also be fitted to this model. E.g. if we have a Gaussian distribution that fits to this model we use as a Gaussian kernel

$$k_G(\mathbf{x}_j, \mathbf{x}) = e^{-\gamma ||\mathbf{x} - \mathbf{x}_j||^2}. \tag{5.32}$$

The usage of a kernel function may improve the classification result, but since we have to evaluate this function for all support vectors it is not useful for implementation in hardware. In the linear case we can directly calculate $\mathbf{w}$ and therefore we only have an addition over the number of features per scene pixel and not over the number of support vectors.

6 Implementation

This chapter presents the approach for the implementation of the given cloud detection algorithms on an FPGA platform and the results of this approach. The first section explains the approach itself, namely the minimization of the set of input parameters in order to allow access to precomputed results of the physical magnitudes. Using this approach we lose in fact accuracy, but on the other hand an implementation with low hardware consumption and fast runtime properties is possible.
The next two sections examine the influence of the approach on the accuracy and describes the implementation details as well as the runtime behavior of the implementation. The second section discusses the results of the software and hardware implementation of the ACCA algorithm whose overall accuracy is slightly higher than the accuracy of the supervised learning algorithms discussed in the third section with the cost of significantly more hardware consumption and lower execution time.

6.1 Data Adaption

The aim of this thesis is finding a way to implement established cloud detection algorithms on an FPGA platform with the requirement of real-time execution. Though the presented algorithms in the previous chapter are not complex and therefore easy to implement in software their implementation in hardware requires some adaptions. The conversion from the gray scale values given in the image matrices to the meaningful physical values TOA-reflectance and TOA-radiance for MODIS as well as TOA-reflectance and at-sensor temperature for ETM+ has to be performed (see chapters 3.1 and 3.2). The results of this conversion can be interpreted as fixed-point representation of the real world values[1]. Since the handling of non-integer values on FPGA platforms is a complex task one has to

---

[1] A fixed-point representation of a real world value R can be interpreted as

$$R \approx \tilde{R} = SQ + B,$$

find a way to prevent the occurrence of these values.
The approach used in this thesis to overcome this difficulty is explained in the following. Let us first reconsider the formulas for the conversion of the gray scale image values to the TOA-reflectance for ETM+ (see chapter 3.1):

$$\rho_i = \frac{\pi d^2}{E_{0i}\cos\theta}\frac{Q_i - Q_{0ij}}{G_{ij}}, \tag{6.1}$$

where i is the band number and j is the index of the sensor's scan line. Q_{0ij} and G_{ij}, the calibration parameters, can be seen as constants, so the result is only dependent on the earth-sun distance d, the solar zenith angle θ, E_{0i}, the solar irradiance of band i and the gray scale value Q_i of band i. In order to reduce the possible outcomes for the reflectance ρ_i the range of the input parameters is reduced by means of the following simplifications:

1. set the earth-sun distance to 1 Astronimical unit (AU) in the calculation of the TOA reflectance
2. round the zenith angle in the calculation of the TOA reflectance to an integer
3. use the mean values for the calibration constants for each scan line instead of the single calibration constants for each sensor (see chapter 3.1).

This approach allows the precalculation of the values for the reflectance and their storage in a Look-up table (LUT) so that, given the gray scale value of band i and the sun's zenith angle of the scene the corresponding reflectance value is available instantly.
The calculation of the at-sensor temperature for ETM+'s TIR-channel (band 6) is only dependent on the gray scale value of the pixel (see equation (3.3)) and therefore can also be stored in the LUT.

Due to the fact that the MODIS data is already radiometrically calibrated the conversion of the gray scale values is a linear transformation only dependent on the calibration parameters (see equations (3.4) and (3.5)). This means that the conversion to the reflectance respectively to the radiance is less complex.

---

where S is the slope, B the bias and Q the stored integer. S can be expressed as $F2^E$, where F is the slope adjustment factor with the normalization $1 \leq F < 2$. 2^E specifies the binary point. E is the fixed power-of-two exponent[37].

The precalculated results are real numbers that are as mentioned previously difficult to handle on an FPGA platform. This problem is circumvented by multiplying the results by a factor 2^n, $n \in \mathbb{N}$, and rounding the result to an integer[2]. This approach inevitably implies inaccuracies that are examined in the following sections.

This paragraph describes the creation of the LUT for both sensors. For ETM+ one calculates the TOA reflectance respectively the at-sensor temperature for each parameter combination [zenith angle, Earth-Sun distance, gain, band, detector number, pixel value], where zenith angle $\in [15, ..., 79]$ (low and high zenith angles are not considered [40]), Earth-Sun distance $= 1$, gain $\in [H, L]$, band $\in [1, ..., 7]$ (including band 6 high gain and low gain), detector number $\in [1, ..., 16]$ and pixel value $\in [0, ..., 255]$. The results are then multiplied by a heuristically chosen factor 2^n $(n = 0, 1, ...)$, rounded and stored in the LUT. As one can easily calculate, this LUT has 4194303 entries. As will be seen later the mean value of the scaling factors for the single detectors can be used instead of the correct ones. Then the LUT has only 262144 entries so that with a bit width for each entry of 16 Bit we get a LUT size of about 0.5 MB.

For MODIS we have a parameter space of input pixel values $\in [0, ..., 32767]$ and spectral channels $\in [1, ..., 36]$ thus giving 1179648 entries in the LUT. Using a bit width of 32 Bit per entry gives a LUT size of about 4 MB. This is too large to fit in the available memory of the hardware platform used in this thesis (see chapter 4). By downscaling the input data the size of the LUT can be reduced so that the data would fit into the FPGA's memory. An alternative would be to use a board with increased memory so that original MODIS data could be evaluated. However, the aim of this thesis is to give a proof of principle for the approach used. When this approach is used in future satellite missions the FPGA's memory will be chosen accordingly.

Due to the memory restriction all evaluation work for the MODIS sensor was conducted in software (Java). The results of a software implementation and an implementation in hardware must not differ because both are using the same numerical integer values, but of course software implementation allows no exact prediction of the runtime behavior.

---

[2] This is called "binary-point-only scaling". It is a special case of the fixed-point representation with $F = 1$, $B = 0$ and $S = 2^E$[37].

6.2 ACCA

Quality of the software implementation
The ACCA algorithm was implemented according to [25] (see chapter 5.1.1). First the quality of the cloud masks is examined in order to assure the algorithm's usability. As the reader can see in the following examples a pure visual assessment shows a good performance in the quality of the cloud mask in most of the examined scenes. Figure 6.2 was captured in southern Russia in summer. One can see that additionally to the easily seen opaque clouds the thin cirrus clouds in the lower part of the scene are well detected. Figure 6.3 was captured over the Sahara. Despite the high reflection of sand in the visible region clouds are well detected (filter 10). In this scene one can see the corresponding cloud shadows very well. Figure 6.4 was captured in northern Alaska also during summer. As in the case of the Sahara scene the analysis of this scene by a human being is complicated because of high reflection of ice, snow and clouds in the visible region. One can find clouds by searching blurring regions in the scene, but the ACCA algorithm takes advantage of the fact that reflectance of ice and snow is low in the SWIR region (band 5) whereas reflectance of clouds is high in this region. This fact is used in filter 3 by calculating the NDSI. As a result, clouds can be separated well also over highly reflected ice and snow. The team of the LPS developing the ACCA algorithm determined the accuracy to ±5% for scene-averaged cloud coverage in 98 % of the examined scenes [25]. As Irish et al. [25] state that there are two types of scenes where the algorithm yields poor performance, namely snow covered mountains and thin cirrus clouds over water. This behavior can be reproduced in this thesis and is shown in the following.

High false positive rates are reported at high altitudes covered with snow in mountainous regions. Here the green spectral band (band 2) can saturate so that snow is classified as cloud during Pass-1 processing. Snow usually has a higher at-sensor temperature than clouds. Therefore, the temperature distribution computed in step "Band-6 Cloud Signature Development" is shifted towards higher temperatures so that more pixels labeled as "Ambiguous" are classified as "Cloud". This behavior was not found in the examined dataset consisting of 387 ETM+ scenes. Instead samples were processed showing the contrary behavior, i.e. clouds and snow are separated very well in mountainous regions (see figure 6.6). Anyway mountainous regions can cause errors due to complicated illumination conditions as shown in figure 6.7. Due to the gradient of the surface the back-scattering per pixel can be larger than on flat area so that the signal received on the sensor is

Strengths	Weaknesses
Fully automated methodology	No per-pixel cloud mask (the resulting cloud mask is not stored in the Landsat archive, only scene and quadrant scores)
Five of eight ETM+ spectral bands	No cloud shadow mask
No ancillary information needed	Thin cirrus clouds often missed
No geometric procession	Ground fog occasionally missed (too warm)
Near real-time availability for browsing and scheduling	Snow occasionally identified as cloud
No dependence on latitude, season or zenith angle	
Precise to ±5% for scene-averaged cloud cover	

Table 6.1: Strengths and weaknesses of the ACCA algorithm according to [25]. In this thesis these facts could be generally confirmed.

contaminated and therefore must be treated carefully. As [25] suggests this behavior can be corrected by underlying a DEM to these scenes.

High false negative rates are associated with thin cirrus clouds over water (see figure 6.5). The reflectance of water is very low so that the pixels are trapped in filter 2 or 7 and labeled as “Non-Cloud”. [25] suggests here to use a land-water mask to improve the algorithm’s quality.

Both the strengths and the weaknesses of the ACCA algorithm are summarized in table 6.1.

Comparing the results of the ACCA implementation and the results published by LPS one finds a mismatch concerning the cloud coverage (see figure 6.1). Having assistance from EROS Data Center (EDC), this discrepancy could not be explained satisfactory after all. There were 42 outliers with more than 10 % difference to the LPS results. Visually examining the respective scenes yields that for 10 of the 42 scenes the cloud coverage results published by LPS are more feasible, whereas for 29 scenes the results of the ACCA implementation used in this thesis seem to reflect the true cloud coverage better. The remaining 3 scenes could not be evaluated because the results were ambiguous. It can be seen that the ACCA implementation is more sensitive to thin clouds but is more prone to misclassify surface features as clouds. The evaluation results are listed in appendix B.

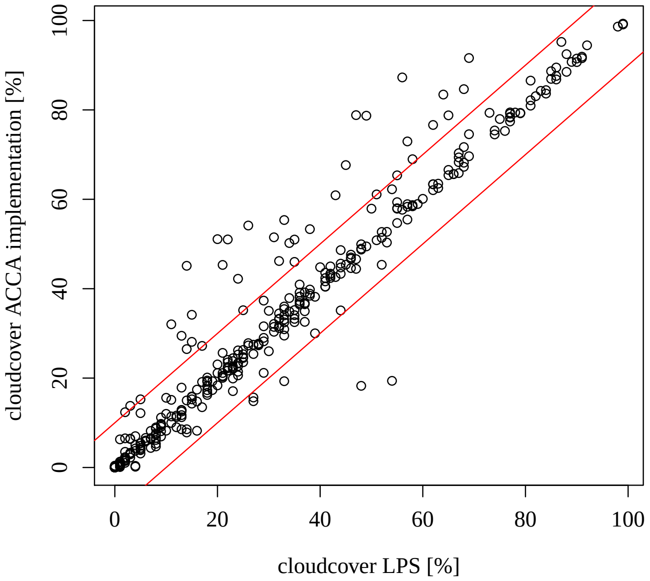


Figure 6.1: This scatter plot shows the deviation of the ACCA implementation from the published cloud coverage values by LPS. The two red lines enclose the scenes where the difference in the cloud coverage values is less than 10 %.

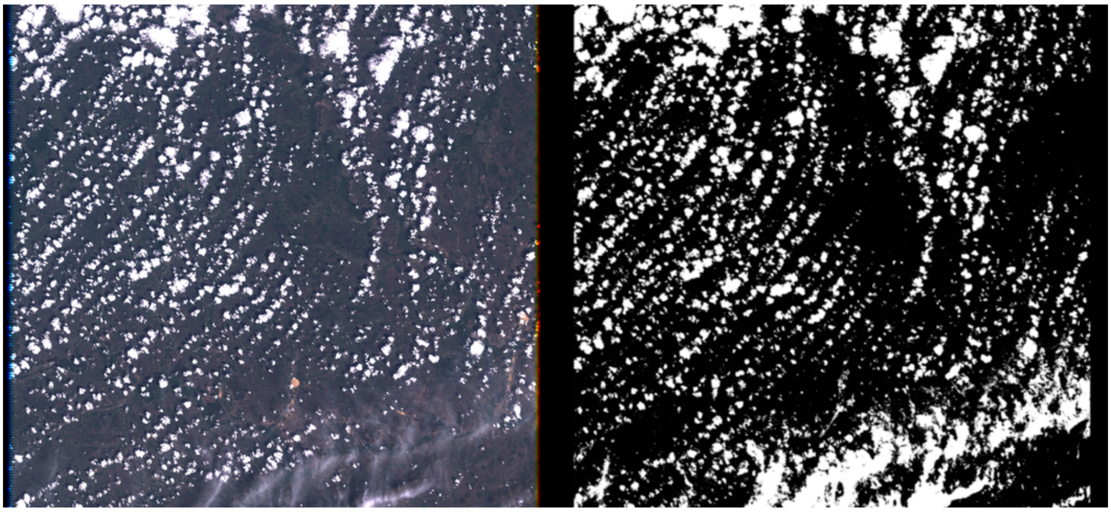


Figure 6.2: Southern Russia (LE71740252002163SGS00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports 23 % cloud coverage, ACCA implementation reports 22.6 % cloud coverage. Cirrus clouds are well detected.

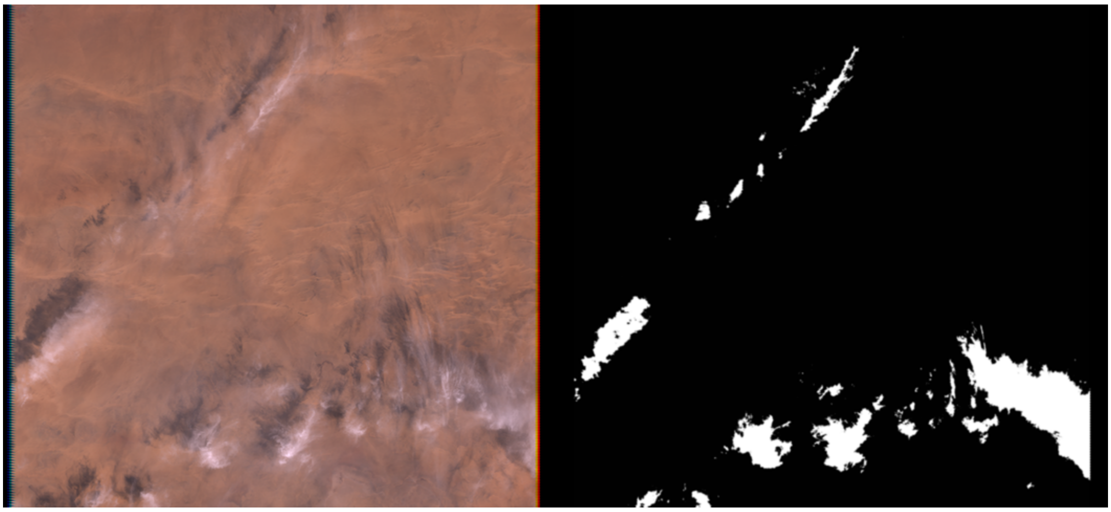


Figure 6.3: Sahara (LE71920472002305EDC00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports 6 % cloud coverage, ACCA implementation reports 6.0 % cloud coverage.

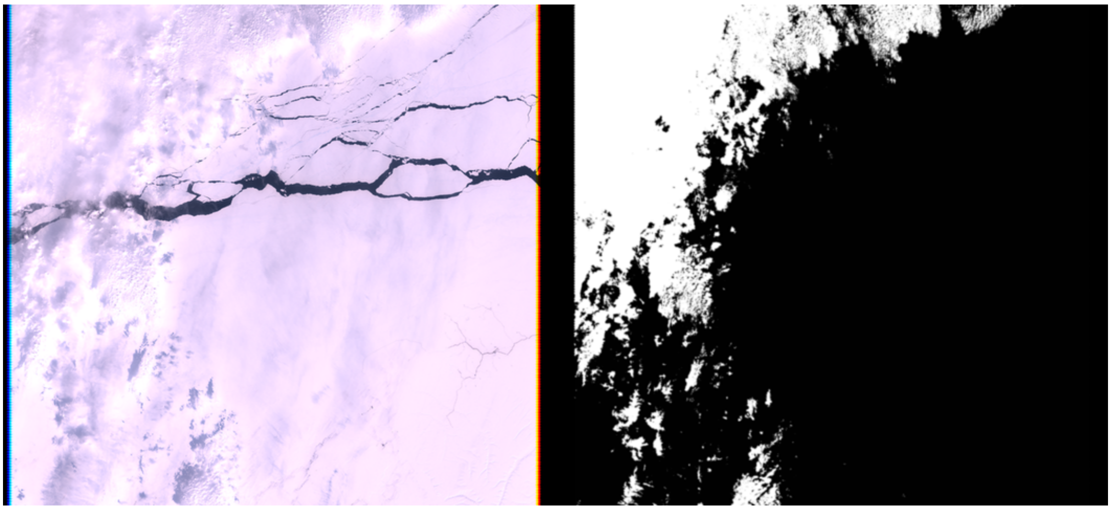


Figure 6.4: Alaska (LE70760102000127AGS00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports, 22 % cloud coverage, ACCA implementation reports 22.5 % cloud coverage.

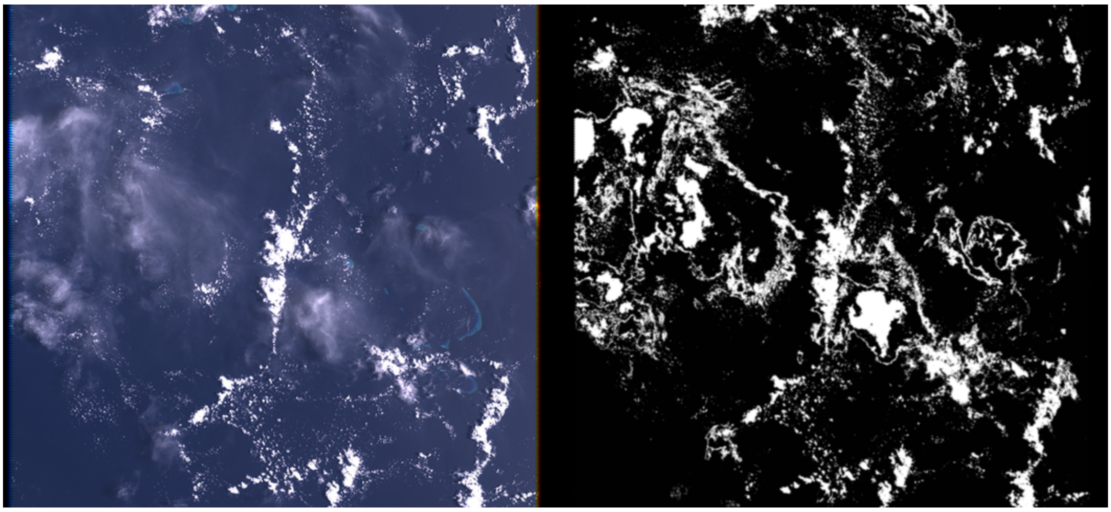


Figure 6.5: Coral Sea (LE70940722002002EDC00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports 16 % cloud coverage, ACCA implementation reports 14.8 % cloud coverage. Both fail in detecting thin cirrus clouds over water.

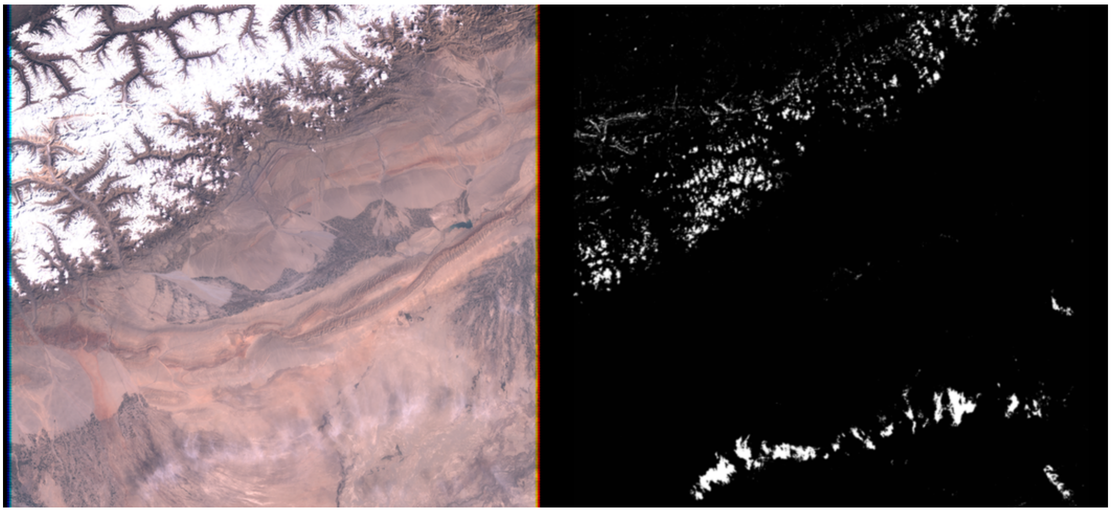


Figure 6.6: Western China (LE71460312000122SGS00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports 3 %, ACCA implementation reports 3.0 %. Snow and clouds are well separated in a mountainous region. Thin cirrus clouds are well detected in the lower part of the scene.

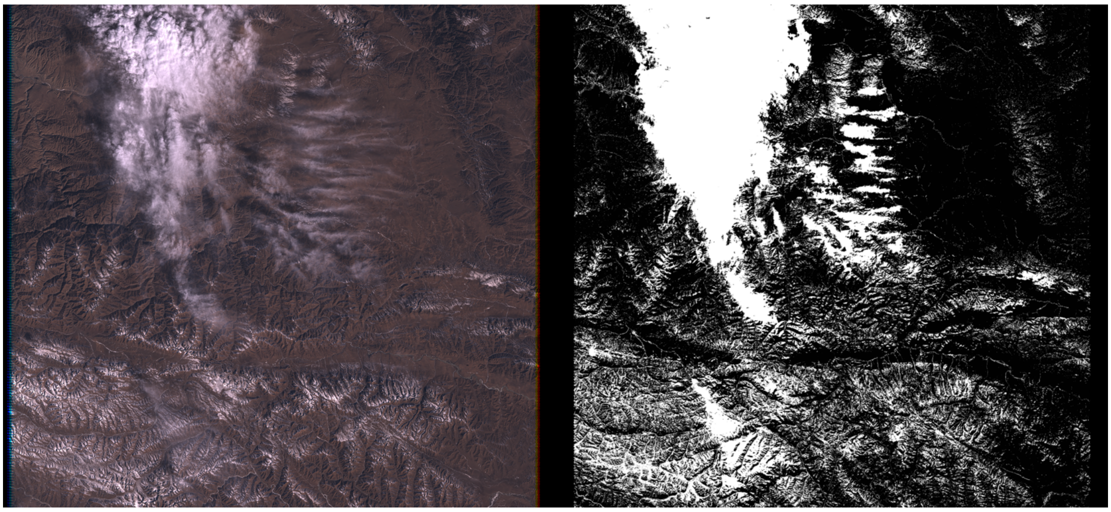


Figure 6.7: Central China (LE71320362002013SGS00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). LPS reports 33 %, ACCA implementation reports 29.5 %. The task of separating clouds from surface can cause errors due to complicated illumination conditions.

Adaptions to the ACCA Algorithm

As one can see in chapter 5.1.1 the ACCA algorithm contains some extensive steps like the calculation of the skewness. In order to create a fast running implementation of this algorithm for hardware it is examined how far it can be simplified without loosing too much of its quality. In addition to the three adaptions mentioned in chapter 6.1 three more are introduced:

4. Skip filter 16 so that the output of filter 15 for B6_shift is either 0 or 1
5. Skip the calculation of the skewness, instead determine the orientation of the skewness by comparing the mean temperature (B6_mean) with the median of the temperature histogram[3]. If the mean temperature is greater than the median we have a positive skewness and so B6_shift is the standard deviation σ of the temperature distribution, otherwise it is 0
6. Skip the nearest neighbor cloud-filling (filter 26)

The influence of these simplifications on the accuracy of the ACCA algorithm is listed in table 6.2. One can see that the mean accuracy of the simplified algorithm when skipping the “nearest neighbor cloud-filling” is 98.1 % for all examined scenes. The mean false positive rate is about 2 %, but scatters up to nearly 14 % in direction to larger false positive rates in the 95 % percentile. These errors occur due to the fixed thresholds that are violated due to the adaptions.

In order to explain this behavior the two scenes with the largest deviation from the accuracy are examined. Figure 6.8 was captured over Germany. In the correct ACCA implementation the Pass-2 “Cold-Cloud” and “Warm-Cloud” contribution is 37.5 % so that one filter 24 condition is violated and processing continues in filter 25. The Pass-2 “Cold-Cloud” contribution is here 5.2 % so that the overall cloud coverage after filter 25 is the sum of Pass-1 “Cold-Cloud” contribution and Pass-2 “Cold-Cloud” contribution. In terms of the right image of figure 6.8 this means that only the dark blue pixels (Pass-1 “Cold-Cloud”) and light blue pixels (Pass-2 “Cold-Cloud”) make up the final cloud mask, whereas the dark red pixels (Pass-1 “Warm-Cloud”) and light red pixels (Pass-2 “Warm-Cloud”) are filtered out. In the case of all simplifications the Pass-2 “Cold-Cloud” and

---

[3] This is just a rough thumb rule [54]! Anyway about 81 % of the data examined hold this assumption.

Accuracy						
Simplification	Mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
1	99.1	95.8	99.4	99.8	100.0	100.0
2	99.1	95.5	99.3	99.7	100.0	100.0
3	99.1	95.5	99.3	99.7	99.9	99.9
4	99.2	95.5	99.3	99.7	99.9	100.0
5	98.4	95.2	99.1	99.7	99.9	100.0
6	98.1	91.5	97.8	98.8	99.6	100.0

Fp-rate						
Simplification	Mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
1	0.7	0.0	0.0	0.1	0.3	3.5
2	0.7	0.0	0.0	0.1	0.4	3.5
3	0.7	0.0	0.0	0.1	0.3	3.6
4	0.7	0.0	0.0	0.1	0.4	5.8
5	1.2	0.0	0.0	0.1	0.6	11.4
6	2.1	0.0	0.3	1.0	2.6	13.8

Table 6.2: These tables show the numerical values of the distribution of the accuracy and the false positive rate (fp-rate) for all the simplifications 1, 2, 3 mentioned in chapter 6.1 and 4, 5, 6 mentioned above. Q_x denotes the corresponding quantile of the data, e.g. $Q_{0.25}$ means that 25 % of the data lie below the given value. From the tables one can see that the accuracy hardly changes when more and more simplifications are introduced. However, the false positive rate increases slightly with rising distribution of the results with respect to the reference results as one can see particularly in the $Q_{0.975}$ column.

"Warm-Cloud" contribution is only 30.7 % so that filter 24 results in using all clouds found in both Pass-1 and Pass-2. Interestingly the Pass-2 "Cold-Cloud" and "Warm-Cloud" contribution of simplification 3 is 42.8 % so that the result is equal to the correct result. Figure 6.9 was taken over southern Sudan. In the correct case the Pass-1 "Cold-Cloud" contribution was 0.38 % and therefore filter 14 results in a final cloud mask consisting only of Pass-1 "Cold-Cloud" pixels. Taking all adaptions the Pass-1 "Cold-Cloud" contribution is 0.59 %. Therefore the final cloud mask consists of all cloud pixels of Pass-1 and Pass-2. As shown above static thresholds can cause problems, not only when using adaptions but also in the correct implementation where values close to the thresholds can lead to completely wrong results.

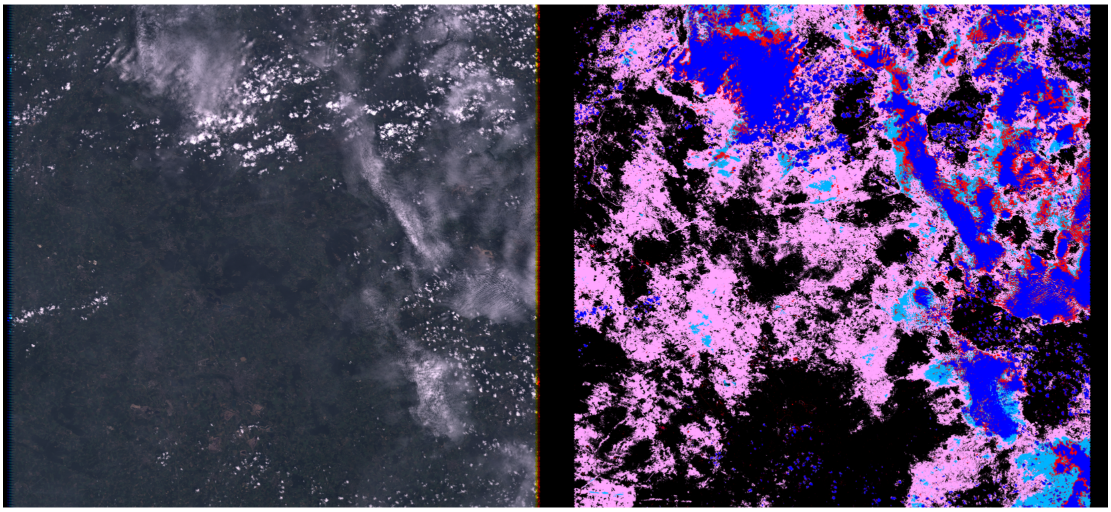


Figure 6.8: Germany (LE71930242002168EDC00). On the left side the RGB image of the scene is shown (a) and on the right side (b) the results of Pass-1 and Pass-2 are shown. Pixels in dark blue are “Cold-Cloud” from Pass-1, dark red pixels are found as “Warm-Cloud” in Pass-1, light blue pixels are found as “Cold-Cloud” in Pass-2 and light red ones are labeled as “Warm-Cloud” in Pass-2. The correct ACCA implementation reports a cloud coverage of 19.4 % whereas the cloud coverage of the implementation of all simplifications reports 50.4 %.

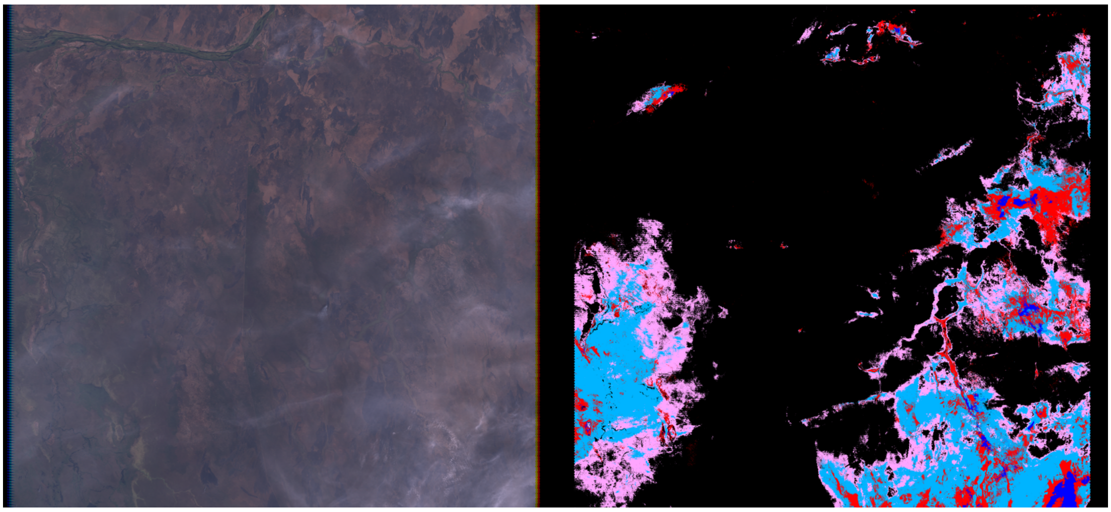


Figure 6.9: Southern Sudan (LE71730542002012SGS00). On the left side the RGB image of the scene is shown (a) and on the right side (b) the results of Pass-1 and Pass-2 are shown. The color code is explained in figure 6.8. The correct ACCA implementation reports a cloud coverage of 0.37 % whereas the cloud coverage of the implementation of all simplifications reports 27.3 %.

Adaption to hardware

The previous section showed which adaptions can be made with the accuracy only slightly decreasing. In order to implement the ACCA algorithm for FPGA hardware all occurrences of floating point numbers need to be transformed into integers and computing time must be reduced.

There are two parts in the algorithm where floating point numbers must be scaled. In the step "Pass-1 Spectral Cloud Identification" where non-integer thresholds occur, and in the step "Band-6 Cloud Signature Development" in the calculation of the percentiles. In the former part, each non-integer threshold value can be expressed by a fraction such that the divisor contains a power of 2. The fraction is then reshaped into a product in order to avoid divisions, as proposed by El-Araby et al. [18]. Multiplications by 2^n, $n \in \mathbb{Z}$, have a low hardware consumption as their calculation can be realized by a bit shift.

The concerned filters are:

- Filter 3:
  - 1. condition ($NDSI < 0.7$):
    0.7 can be approximated by $\frac{90}{128}(= 0.7013125)$, so
    $128 \cdot (B2 - B5) < 90 \cdot (B2 + B5)$
  - 2. condition ($NDSI > -0.25$):
    -0.25 is $-\frac{1}{4}$, so
    $(B2 - B5) > -4 \cdot (B2 + B5)$
- Filter 4 ($NDSI > 0.8$):
  0.8 can be approximated by $\frac{102}{128}(= 0.796875)$, so
  $128 \cdot (B2 - B5) > 102 \cdot (B2 + B5)$
- Filter 8 ($\frac{B4}{B3} < 2.35$):
  2.35 can be approximated by $\frac{301}{128}(= 2.3515625)$, so
  $128 \cdot B4 < 301 \cdot B3$
- Filter 9 ($\frac{B4}{B2} < 2.16248$):
  2.16248 can be approximated by $\frac{17715}{8192}(= 2.162475586)$, so
  $8192 \cdot B4 < 17715 \cdot B2$.

The conversion of the percentile limits is straightforward:

- 83.5-percentile:
  0.835 can be approximated by $\frac{13681}{16384}(= 0.835021972)$

- 97.5-percentile:
  0.975 can be approximated by $\frac{15974}{16384}$(= 0.974975585)
- 98.75-percentile:
  0.9875 can be approximated by $\frac{16179}{16384}$(= 0.987587793).

The amount of hardware consumed by the calculation of cloud temperature statistics in step "Band-6 Cloud Signature Development" can be reduced by using a histogram for the determination of the variance instead of evaluating

$$Var(T) = \frac{1}{N_C}\sum_{i=1}^{N_C}(T(i) - \bar{T})^2 \qquad [K^2]. \tag{6.2}$$

Here, N_C is the number of cloud pixels, $\bar{T}$ is the mean value of the cloud temperatures and $T(i)$ is the temperature value of each cloud pixel. The advantage of using a histogram is when in calculating both the mean temperature value and the variance the sum does not extend over all cloud pixels but only over the number of bins N in the histogram. The width W of each histogram bin is given by

$$W = \frac{\mathrm{T}_{\max} - \mathrm{T}_{\min}}{N} \qquad [K], \tag{6.3}$$

where $\mathrm{T}_{\max}$ respectively $\mathrm{T}_{\min}$ are the maximum and minimum values of the sensor's temperature range. For these values the rounded values given in section 3.1 are used, so $\mathrm{T}_{\min} = 128\,\mathrm{K}$ and $\mathrm{T}_{\max} = 348\,\mathrm{K}$. Setting the number of bins N to 110 the step width is exactly 2 K which simplifies the following equations. The mean value $\bar{T}$ can be expressed in terms of a histogram as

$$\begin{aligned}
\bar{T} &= \frac{\sum_{i=1}^{N} h(i)\cdot i}{\sum_{i=1}^{N} h(i)}\cdot W + \frac{W}{2} + \mathrm{T}_{\min} \\
\Leftrightarrow \bar{T} &= \frac{\sum_{i=1}^{N} h(i)\cdot i}{\sum_{i=1}^{N} h(i)}\cdot 2K + 129K \qquad (6.4) \\
\Leftrightarrow \bar{T} &= \frac{\sum_{i=1}^{N} h(i)\cdot i}{N_C}\cdot 2K + 129K, \qquad (6.5)
\end{aligned}$$

where $h(i)$ is the number of entries of bin i and N_C is the number of cloud pixels in the scene.

The variance of a histogram is expressed as

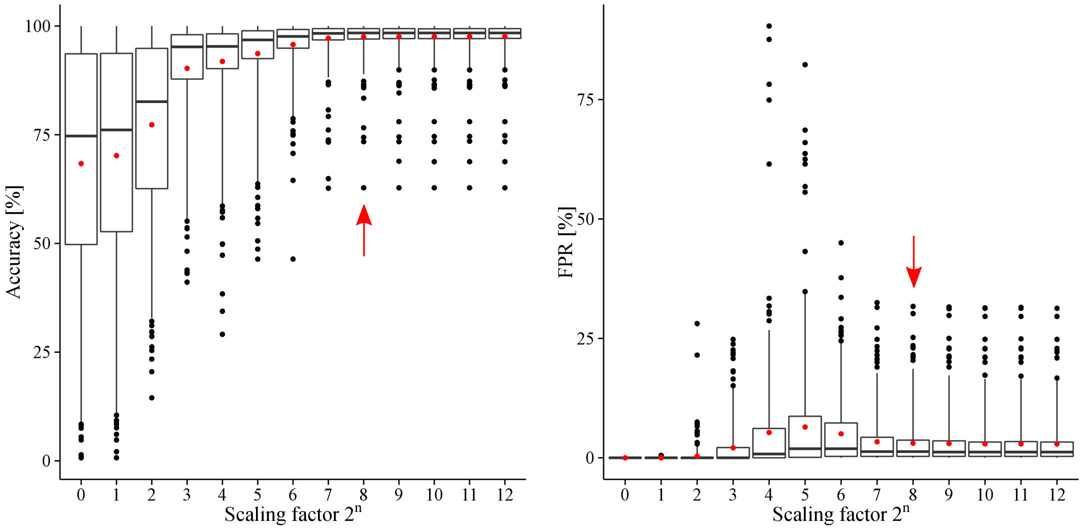
$$\begin{aligned} Var(T) &= \frac{1}{N} \cdot \sum_{i=1}^{N} \left[(W \cdot i + \frac{W}{2} + \mathrm{T_{min}}) - \bar{T} \right]^2 \cdot h(i) \\ \Leftrightarrow Var(T) &= \frac{1}{N} \cdot \sum_{i=1}^{N} \left[(2K \cdot i + 129K) - \bar{T} \right]^2 \cdot h(i). \qquad (6.6) \end{aligned}$$

The shift factor used for adaption of the new temperature thresholds in Pass-2 is calculated as the product of the skewness and the standard deviation $\sigma(T)$ of the temperature distribution of the clouds (see filter 17). It was shown on pages 50 ff. that the calculation of the skewness can be skipped, but the calculation of the standard deviation is still necessary. The standard deviation σ is defined as the square root of the variance:

$$\sigma(T) = \sqrt{Var(T)} \qquad [K]. \qquad (6.7)$$

The calculation of the square root can only be done by an iterative numerical procedure which is both hardware and processing time consuming. As a shortcut a LUT was created where the root values are stored and therefore can be accessed in one clock cycle.
Finally the scaling of the reflectance values ranging from 0 to 1 has to be examined. As described in section 3.1 a LUT was created whose values were scaled by a factor 2^n, where $n = 0, ..., 12$, and then rounded so that the values become integer values. As can be seen in figure 6.10 from scaling factor of 2^8 both the accuracy and the false positive rate start to remain in a quite constant range. The following step for the final evaluation of the accuracy is to reduce the scaling factor of the temperature. One has to consider that each bin in the histogram has a range of 2 K and the temperature resolution is approximately 0.9 K so that about 2 digital numbers of the input data refer to one bin. Further examination of the scaling yields that scaling of the temperature does not improve the accuracy. The average mean value of the accuracy is 97.5 % and the mean value of the false positive rate is 2.5 % (see table 6.3), so both values are still in an acceptable range after scaling the data.

Due to the different possible ways in Pass-1 depending on the input values the runtime behaviour is not strongly determined. Evaluating a Landsat 7 scene with 6000·6600 pixels The "Pass-1 Spectral Cloud Identification" ranges from $1+4 \cdot \mathrm{N}_S$ clock cycles for a complete cloud free scene to $1 + 14 \cdot \mathrm{N}_S$ clock cycles for a completely cloudy scene having one



(a) **(b)**

Figure 6.10: Plot (a) shows the distribution of the accuracy with respect to the scaling factor 2^n ($n = 0, ..., 12$) of the LUT and plot (b) shows the corresponding distribution of the false positive rate. Starting at a scaling factor of 2^8 both the accuracy and the false positive rate start to remain constant.

	mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
accuracy [%]	97.5	88.3	97.0	98.4	99.4	100.0
false positive rate [%]	2.5	0.0	0.2	0.9	3.3	14.4

Table 6.3: This table shows the distribution of both the accuracy and the false positive rate of the ACCA algorithm adapted to hardware. In comparison to the values given in table 6.2 only a small decrease in the accuracy can be seen.

RAM access per clock cycle, where N_S is the number of scene pixels. Having six RAM accesses per clock cycle (five for reading the value vector per pixel and one for writing the cloud mask pixel) one cloud mask pixel could be processed per clock cycle. This was implemented by El-Araby et al. [18]. The authors evaluated the TOA reflectance and the at-sensor temperature during runtime in a single-precision floating point accuracy and not via a scaled LUT, consuming significantly more hardware (see table 6.4), but of course with a accuracy of nearly 100 %. In the ACCA algorithm used in this thesis we get an overall mean accuracy of 91 % when only evaluating Pass-1 and setting the “Ambiguous” pixels to “Non-Cloud”.

	single-precision floating point	scaled LUT (only Pass-1)
Slices	17565	902
LUTs	20885	1235
Slice Flip Flops	23005	492
MULT 18X18	36	1
RAMB16	36	5

Table 6.4: This table shows the different hardware requirements of the single-precision floating point implementation of El-Araby et al. [18] and the implementation using a scaled LUT. It can be clearly seen that the latter implementation needs much less hardware.

The calculation of the cloud statistics takes at most 1372 clock cycles (independent of the number of pixels), but due to handling large values in the calculation of the variance, the mean value of the temperature histogram and the quantiles this part consumes most of the hardware and contains the longest signal path (limiting the maximum speed to 30 MHz in the complete ACCA implementation).
The examination of the "Ambiguous" pixels needs at most $222 + 11 \cdot N_S$ clock cycles and the following final evaluation needs $787 + 4 \cdot N_S$. We can clearly see that the runtime of the algorithm is dominated by the number of pixels (a complete Landsat 7 scene consists of $6000 \cdot 6600 = 3.96 \cdot 10^7$ pixels) so that the initialization steps and the calculation of the cloud statistics can be neglected. This leads us to a minimum of $8 \cdot N_S$ clock cycles and a maximum of $29 \cdot N_S$ clock cycles for a complete Landsat 7 scene.
This implementation shows that accepting a mean false positive rate of 2.5 % an implementation of the complete ACCA algorithm is possible by using scaled LUTs. This consumes roughly four times less hardware than an exact implementation of only Pass-1 with single-precision floating point accuracy used by El-Araby et al. [18] (see table 6.5).

	scaled LUT (complete ACCA)
Slices	2895
LUTs	4779
Slice Flip Flops	1808
MULT18X18	17
RAMB16	7

Table 6.5: The hardware consumption of the implementation of the complete ACCA algorithm is shown in this table. As one can see still fewer hardware is needed when using a scaled LUT instead of working with single-precision floating point values (see table 6.4).

6.3 Supervised Learning

6.3.1 Training Data

Training of supervised learning algorithms needs training data for the determination of the coefficients of the classifier. This data should cover most of the information that could occur in the real data (e.g. surface features like desert, ice, vegetation etc.) and be as accurate as possible, hence as little misclassified training data as possible must exist. Due to the fact that automatically created "Cloud" masks are not perfectly accurate, visual assessment is performed. Only pixels whose class membership is clear are included in the training data set, i.e. ambiguous pixels like fringe areas of clouds are excluded. For each training pixel the corresponding TOA reflectance respectively brightness temperature with its class label ("-1" for "Non-Cloud" , "1" for "Cloud") were linewise recorded.

For ETM+ 420000 training pixels including spectral bands 1-7 are extracted which cover for $3 \times 10^{-6}\,\%$ of the available data. The MODIS training data set consists of 54804 training points including all the spectral bands used in the MODIS cloud detection algorithm [4] listed in table 3.2, covering 0.07 % of the available data.

6.3.2 LDA

LDA as a well accepted method for cloud detection [5, 39] is applied to both ETM+ and MODIS data. Figures 6.11 and 6.12 show the histograms and density plots of each class in the training data. The assumption of a Gaussian distribution barely holds for the data of each class. However, one can see that the classes are occasionally well separated so that a large part of the data can clearly be assigned to one class. In order to determine the coefficients of the discriminant functions one first has to set the a-priori probabilities of the classes "Cloud" and "Non-Cloud". Therefore we take the average global cloud coverage which is reported to be in the range of 50 % [24] and 66 % [66], but for simplicity let us assume an a-priori probability of 50 % [34]. We can hold this assumption because the average global cloud distribution holds no information about the cloud coverage of a single scene. With this assumption the last term of equation (5.10) becomes 0 and can therefore be skipped.

In order to obtain a prototype of a reliable decision function we first have to determine the optimal size of the training dataset. We consider training datasets with $\{100, 1000, 10000, 100000\}$ points per class in the case of ETM+ and $\{100, 1000, 10000\}$ points per class

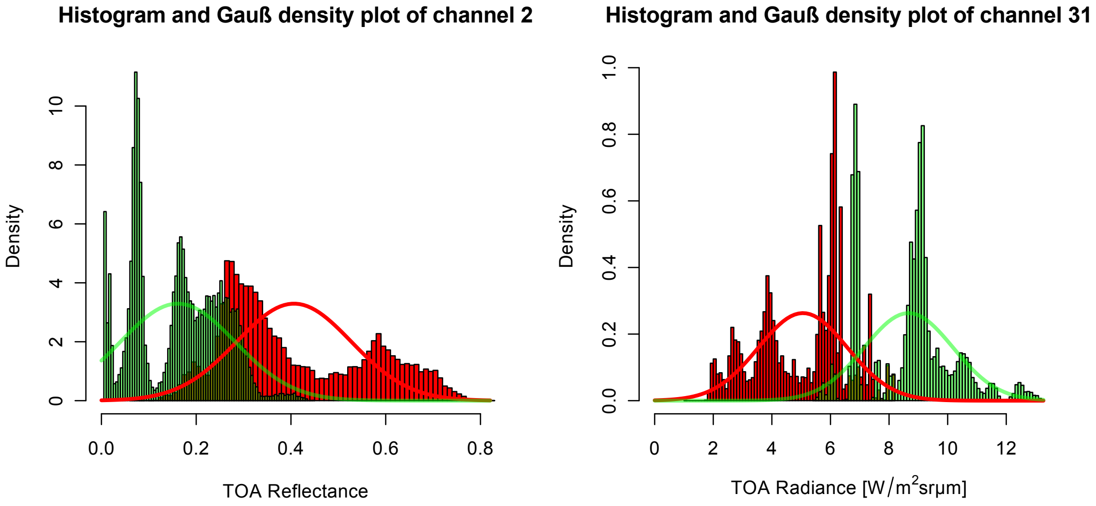


Figure 6.11: These plots show the histograms and the density plots assuming a Gaussian distribution of the MODIS channels 2 (0.865 µm, reflective) and 31 (11.03 µm, thermal). The green components refer to the class "Non-Cloud" whereas the red components refer to the class "Cloud".

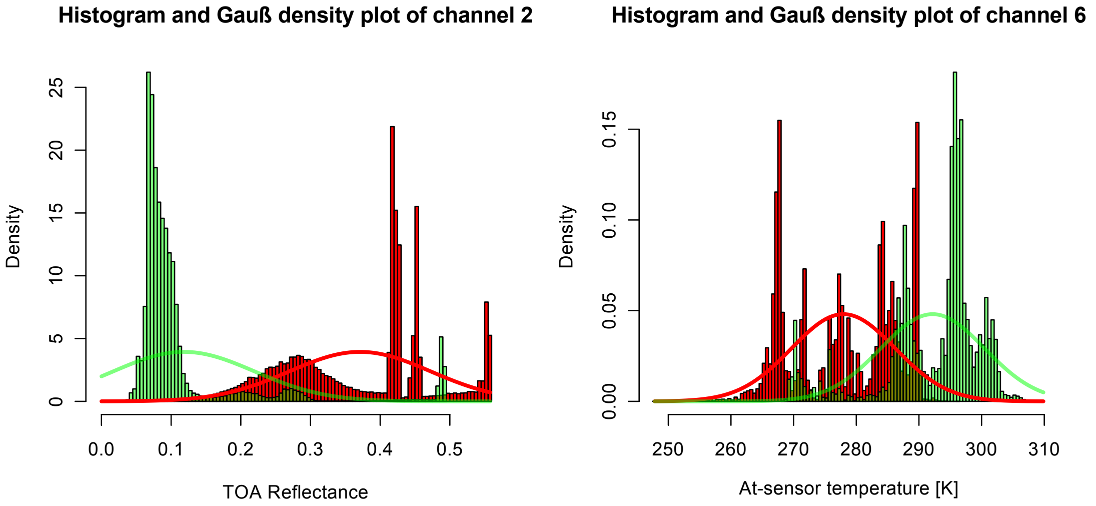


Figure 6.12: These plots show the histograms and the density plots assuming a Gaussian distribution of the ETM+ channels 2 (0.57 µm, reflective) and 6 (11.5 µm, thermal). The green components refer to the class "Non-Cloud" whereas the red components refer to the class "Cloud".

for MODIS. For each data size and sensor five different training datasets are drawn independently from the original training set. An LDA is trained on each of the five training datasets and the accuracy is obtained by applying the classifier to all datasets. The set $(\mathbf{w}, b)$ with the highest accuracy is kept and taken to compare the results of the different sizes of training data.

As one can see in table 6.6 the number of training points has nearly no effect to the quality of the classifier for ETM+, whereas for MODIS the best accuracy is given for 1000 training points (see table 6.7). The accuracy of the classifier was based on the ACCA algorithm for ETM+ respectively on the cloud mask algorithm for MODIS. From the result one can see that the global accuracy is well enough to use this algorithm for on-board cloud detection though Marais [36] emphasizes the low accuracy, but one has to find a trade-off between quality and speed in the purpose of on-board processing of a cloud mask. Anyway, the goal of this thesis is not to find the best set of coefficients but to find a reasonable one in order to compare the software and hardware implementations of this algorithm. The coefficients derived from the training sets are used as a reference from now on for the comparison of the exact software implementation and the adapted hardware implementation. They can be found in appendix C.1.

points/class	accuracy [%]
100	91.5
1000	91.4
10000	91.3
100000	91.3

points/class	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
100	70.3	87.4	92.9	97.2	100.0
1000	70.4	87.7	92.9	97.2	100.0
10000	69.8	87.3	92.7	97.2	100.0
100000	69.7	87.1	92.7	97.2	100.0

Table 6.6: This table shows the results of the comparison of the different number of training points for ETM+. The first column shows the number of training points for each class. One can see that the number of training points nearly has no influence on the accuracy.

points/class	accuracy [%]
100	76.3
1000	79.6
10000	79.2

points/class	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
100	58.1	69.5	75.6	84.6	91.9
1000	58.4	75.1	81.0	87.6	91.2
10000	58.1	74.9	80.6	87.4	91.1

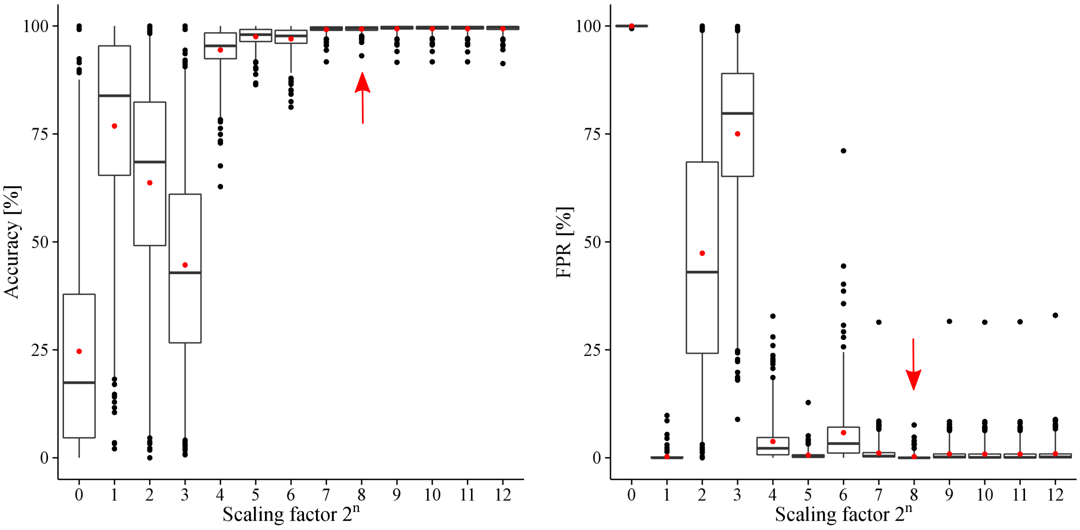
Table 6.7: This table shows the results of the comparison of the different number of training points for MODIS. The first column shows the number of training points for each class. One can see that for 1000 training points per class the best result is obtained.

The conversion of the image data into physical magnitudes is performed using the adaptions as described in chapter 6.1. As can be seen in table 6.8 these adaptions nearly have no influence to the accuracy of the LDA for the ETM+ sensor. Having examined these simplifications the next step is the adaption to FPGA, i.e. scaling the coefficients, the TOA reflectance and the at-sensor temperature to integers in order to allow fast execution and low hardware consumption. This is done by multiplying these values by 2^n ($n = 0, ..., 12$) and rounding them to integers. As can be seen in figure 6.13 a scaling factor of 2^8 is sufficient to keep both the accuracy and false positive rate nearly constant. Thus the usage of a LUT has only a slight influence on both the accuracy and the false positive rate and therefore can be used as an implementation for cloud detection with LDA.

Let us examine the two scenes with the largest deviation from the accuracy. The scene with the largest deviation was taken over Central Siberia in May (LE71410162000135SGS00) in a mountainous region still covered with snow (figure 6.14(a)). Visual interpretation shows no cloud coverage in the scene, but due to difficult illumination conditions from inclined back-scattering areas in combination with snow the classifier detects clouds. The

	mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
accuracy	99.3	97.8	99.0	99.5	99.8	100.0
false positive rate	0.3	0.0	0.0	0.0	0.1	2.1

Table 6.8: This table shows the distribution of both the accuracy and false positive rate for all examined ETM+ scenes using all adaptions given in the text.



(a) **(b)**

Figure 6.13: Plot (a) shows the distribution of the accuracy with respect to the scaling factor 2^n ($n = 0, ..., 12$) of the scaled TOA-reflectances and at-sensor temperatures. Plot (b) shows the corresponding distribution of the false positive rate for the ETM+ data set. Starting at a scaling factor of 2^8 both the accuracy and the false positive rate start to remain constant.

misclassification of snow as clouds could be prevented by training the classifier with exclusive snowy and cloudy data sets. However, this is not in the scope of this thesis. In figure 6.14(b) we compare the results obtained by classification, once when using the “exact” implementation and once when using the “adapted” one. The exact implementation, i.e. unscaled data without adaptions, detects a cloud coverage of 19.6 %, the “adapted” one, i.e. scaled data with the adaptions given above, finds a cloud coverage of 11.0 %. White pixels are correctly classified as “Clouds” in both cases and black ones are classified as “Non-Clouds” in both cases. The blue pixels are wrongly classified as “Non-Clouds” by the “adapted” implementation (*false negative*) and the red ones are wrongly classified as “Clouds” by the “adapted” implementation (*false positive*). The false negative rate of 39.7 % is quite high whereas the false positive rate has only a value of 0.3 %. Note that the wrongly labeled pixels often occur as single events where surrounding pixels of the same class are classified correctly.

The next scene was taken over Eastern Bolivia in June (figure 6.15(a)). The exactly

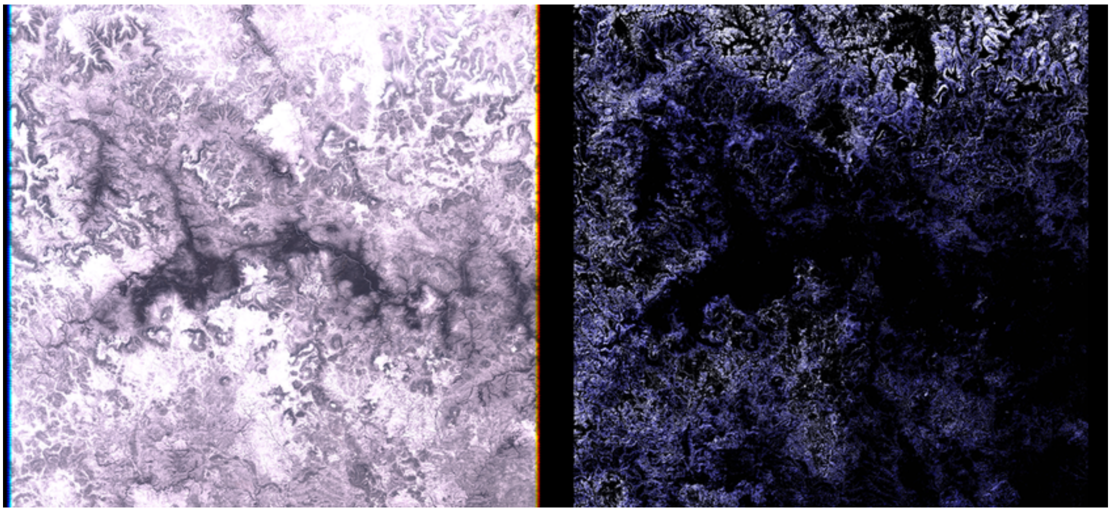

(a) (b)

Figure 6.14: Central Siberia in spring (LE71410162000135SGS00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). One can see here that the false negative rate having a value of 39.7 % predominates (blue pixels), whereas the false positive rate has only a value of 0.3 % with respect to the exact implementation (i.e TOA reflectances and coefficients are not scaled).

evaluated case gives a cloud coverage of 54.4 % whereas the simplified case gives a cloud coverage of 50.3 % yielding an accuracy of 96.9 %. Visual interpretation shows that both results lie in a plausible region. Compared to the scene taken over Central Siberia case both the false positive rate and the false negative rate are quite small with values of 0.0 % respectively 5.7 %. Here only a few pixels are scattered randomly in the cloud mask; most of the falsely classified pixels lie in the fringes of the clouds (see figure 6.15(b)). The discrepancies between the exact and the adapted results lie mainly in the fringes of the clouds which are anyway difficult to classify.

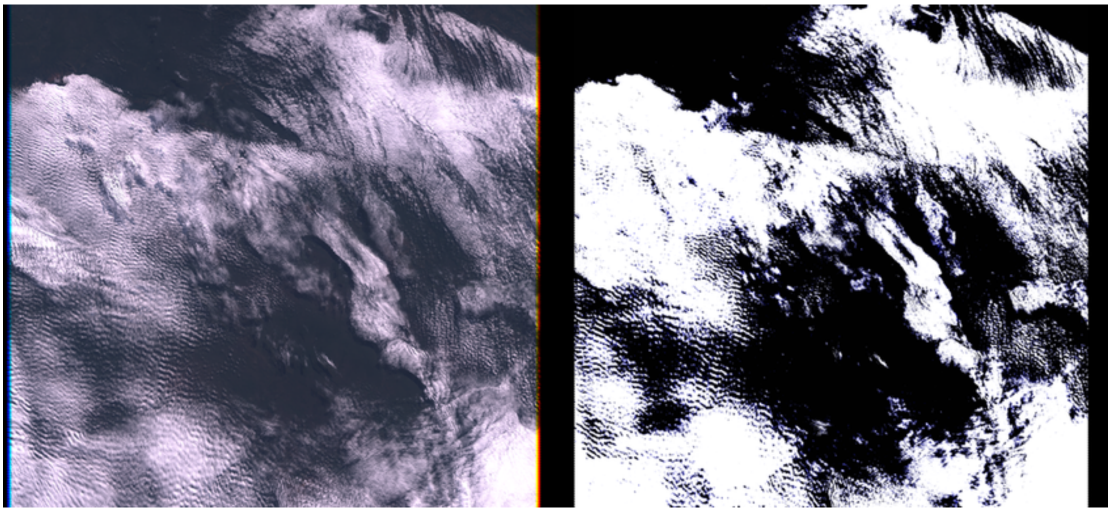

(a) (b)

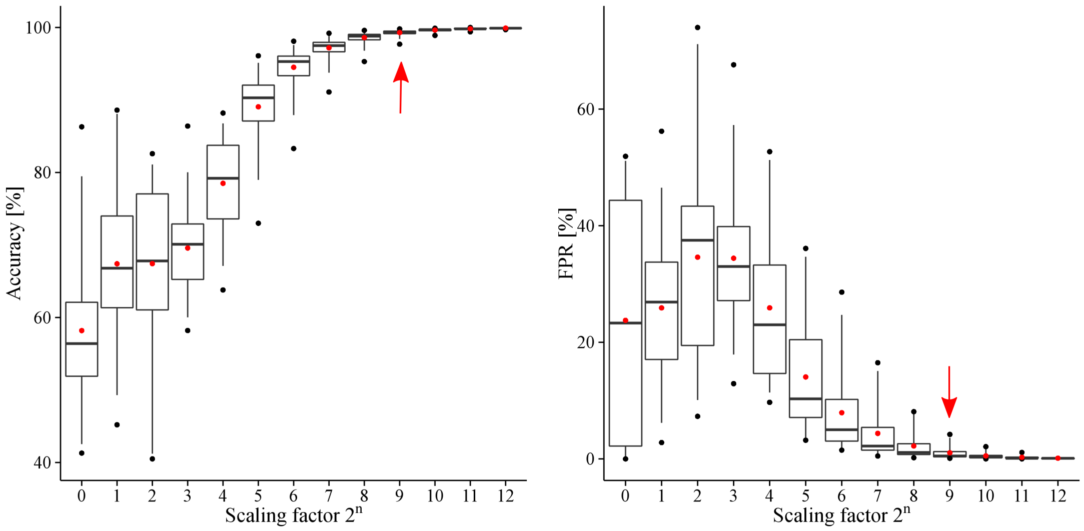
Figure 6.15: Eastern Bolivia in June (LE72280731999181EDC00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). The classification result is very good with an accuracy of 96.9 %. We can see here that most of the misclassified pixels lie on the fringes of the clouds (blue pixels).

Now the results for the MODIS sensor are examined. The reflectances and radiances (for the thermal bands) calculated in equations (3.4) and (3.5) are scaled with factors 2^n ($n = 0, .., 12$) and rounded to integers in order to allow implementation for FPGA with low hardware consumption. We can see from figure 6.16 that a scaling factor of 2^9 is enough to keep the inaccuracies constant. The overall accuracy and false positive rate using this approach is given in table 6.9.

The scene with the greatest deviation with respect to the accuracy was captured over Eastern Kazakhstan in autumn. It mainly consists of desert, soil and some mountains with only few clouds (see figure 6.17(a)). First we note that the classifier has severe

	mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
accuracy [%]	99.3	98.4	99.1	99.4	99.5	99.7
false positive rate [%]	1.1	0.17	0.38	0.55	1.35	3.6

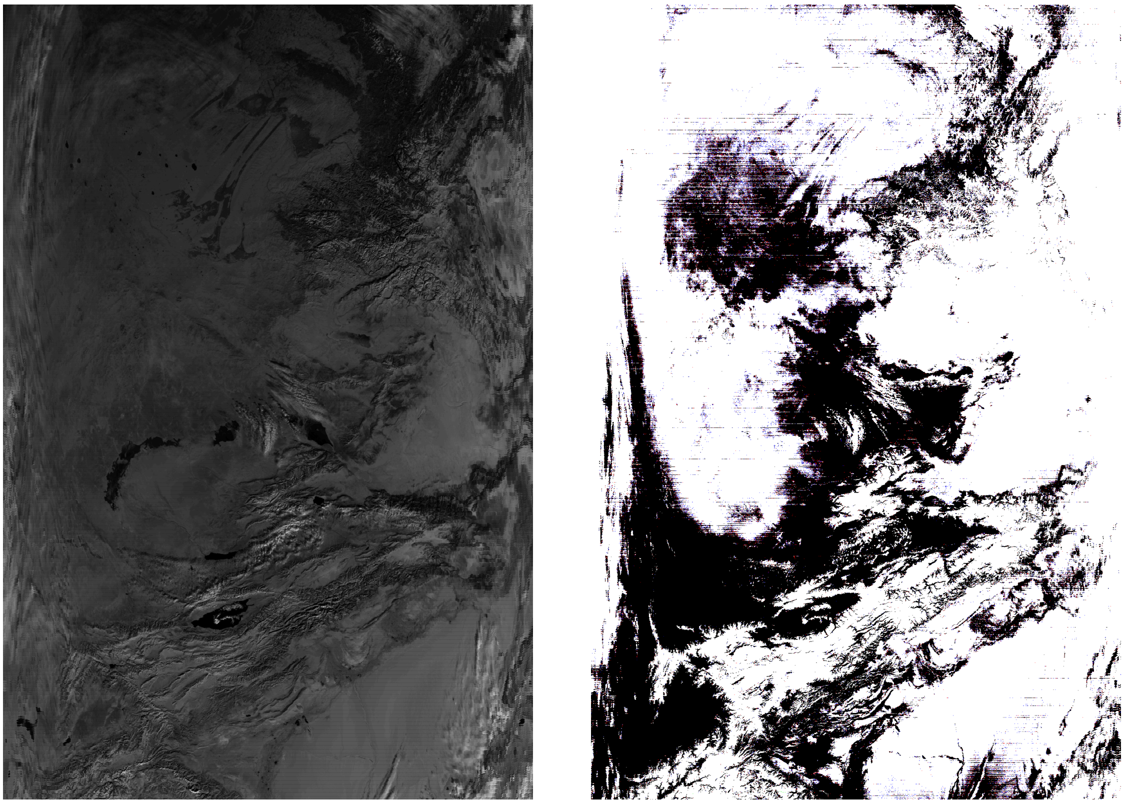
Table 6.9: This table shows the distribution of both the accuracy and false positive rate for all examined MODIS scenes using the adaption given in the text.



(a) **(b)**

Figure 6.16: Plot (a) shows the distribution of the accuracy with respect to the scaling factor 2^n ($n = 0, ..., 12$) of the scaled radiances and plot (b) shows the corresponding distribution of the false positive rate for the MODIS data set. Starting at a scaling factor of 2^9 both the accuracy and the false positive rate start to remain constant.

problems in correctly detecting these surface features. They are often wrongly classified as "Clouds". But the overall accuracy of the classifier using the adaptions made for the FPGA implementation with respect to the exact one is 97.7 % and hence in an acceptable range. As we can see in 6.17(b) there is only a small false positive rate of 0.3 % (red pixels) and a false negative rate of 0.2 % (blue pixels). As in the ETM+ case examined before the wrongly classified pixels lie mainly on the fringes of the "Clouds".



(a) (b)

Figure 6.17: Eastern Kazakhstan in autumn (MOD021KM.A2005357.0540.005.2010166204718). On the left side (a) we can see the contrast stretched image of band 5 and on the right side (b) we see the resulting cloud mask using the LDA classifier. Here the classifier tends to misclassify soil as clouds. However this is a matter of the training data set. Both the false positive rate (red pixels) and the false negative rate (blue pixels) are very small.

6.3.3 SVM

The solution of the optimization problem was performed with the R package “e1071”[38] based on the “libsvm” library[11] (see appendix D). Similar to the study of LDA in chapter 6.3.2 we first examine the general results obtained by both sensors and then the simplifications of the input data for ETM+ and MODIS separately. As already seen in figure 6.11 and figure 6.12 the two classes of the training data can not be separated linearly due to their large overlap.

First we examine the influence of the number of training points on the quality of the classifier. As reference we take the cloud mask derived from the ACCA algorithm for the ETM+ dataset respectively the delivered cloud product for the MODIS dataset. For each sample size five independently drawn samples were trained. As we can see in tables 6.10 and 6.11 the accuracy of the 10-fold cross validation rises with increasing number of training points. The mean accuracy of the classification of unknown data slightly decreases. We can say the more input data we have for training the classifier the better is the output explaining exactly this training data set (so we get an increasing 10-fold cross validation accuracy; this behavior is also called *overfitting*). But we don’t know the noise distribution, so the weaker the bound to a special case of noise distribution is (i.e. less training data points), the better an unknown data set can be classified. Hence a compromise for the

points/class	accuracy [%] 10-fold cross validation	accuracy [%]	#SVs
100	97.0	91.9	33
1000	98.5	91.6	172
10000	99.1	91.2	791
100000	99.2	90.9	5086

points/class	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
100	72.9	89.5	93.7	97.3	100
1000	73.7	88.7	93.0	96.8	100
10000	68.9	87.5	92.8	96.7	100
100000	66.7	87.5	92.7	97.7	100

Table 6.10: This table shows the results of the comparison of the different number of training points for ETM+. The first column shows the number of training points for each class. Though the result of the 10-fold cross validation raises slightly one can see that the quality of the classifier becomes slightly worse with an increasing number of training points.

points/class	accuracy [%] 10-fold cross validation	accuracy [%]	#SVs
100	99.5	76.6	5
1000	99.9	77.7	9
10000	100.0	76.8	23

points/class	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
100	51.3	71.5	76.5	87.5	92.0
1000	53.2	73.3	77.9	87.9	91.9
10000	54.4	71.2	76.2	86.2	91.9

Table 6.11: This table shows the results of the comparison of the different number of training points for MODIS. The first column shows the number of training points for each class. One can see that the quality of the classifier is quite independent of the number of training points.

size of the training dataset has to be set. For ETM+ 1000 training points and for MODIS 1000 training points are chosen.

Next the influence of the cost factor to the accuracy is examined. For the cost factor C values of $C \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ are used.
Tables 6.12 and 6.13 show the result of the examination of the cost factor C. We can see in table 6.12 that the cost factor has nearly no influence on the overall accuracy of the ETM+ data, so a cost factor of $C = 1$ was chosen. For the MODIS data the magnitude of the cost factor is significant noticeable, as can be seen in table 6.13. Hence a cost factor of $C = 10^{-2}$ was set for training the classifier for the MODIS dataset. The derived coefficients that are used for classification via SVM are listed in appendix 6.12.

cost factor	accuracy [%] 10-fold cross validation	accuracy [%]	#SVs
10^{-2}	94.6	91.4	858
10^{-1}	98.0	91.3	567
10^{0}	98.5	91.6	172
10^{1}	99.1	91.3	152
10^{2}	99.0	91.3	105

cost factor	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
10^{-2}	70.9	89.1	93.3	97.1	100
10^{-1}	70.5	88.3	92.7	96.8	100
10^{0}	73.7	88.7	93.0	96.8	100
10^{1}	71.5	87.9	92.9	96.9	100
10^{2}	72.4	88.4	92.7	96.6	100

Table 6.12: This table shows the results of the comparison of the different cost factors for 1000 training points per class for ETM+. Though the result of the 10-fold cross validation raises one can see that the quality of the classifier is quite independent of the cost factor.

cost factor	accuracy [%] 10-fold cross validation	accuracy [%]	#SVs
10^{-2}	100	78.2	100
10^{-1}	100	77.5	22
10^{0}	99.9	77.7	9
10^{1}	99.9	75.2	7
10^{2}	99.9	77.4	6

cost factor	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
10^{-2}	54.6	73.7	78.7	87.8	92.6
10^{-1}	54.0	72.8	76.8	87.8	91.8
10^{0}	53.2	73.3	77.9	87.9	91.9
10^{1}	53.4	69.7	73.4	85.4	91.0
10^{2}	56.1	72.0	76.7	87.8	91.9

Table 6.13: This table shows the results of the comparison of the different cost factors for 1000 training points per class for MODIS. Here we can see that with increasing cost factor the quality of the classifier also increases.

The steps for reducing the input space of the variables *zenith angle*, *gray scale value* and *calibration constants* are described in chapter 6.1 and are the same as for LDA for both ETM+ and MODIS.

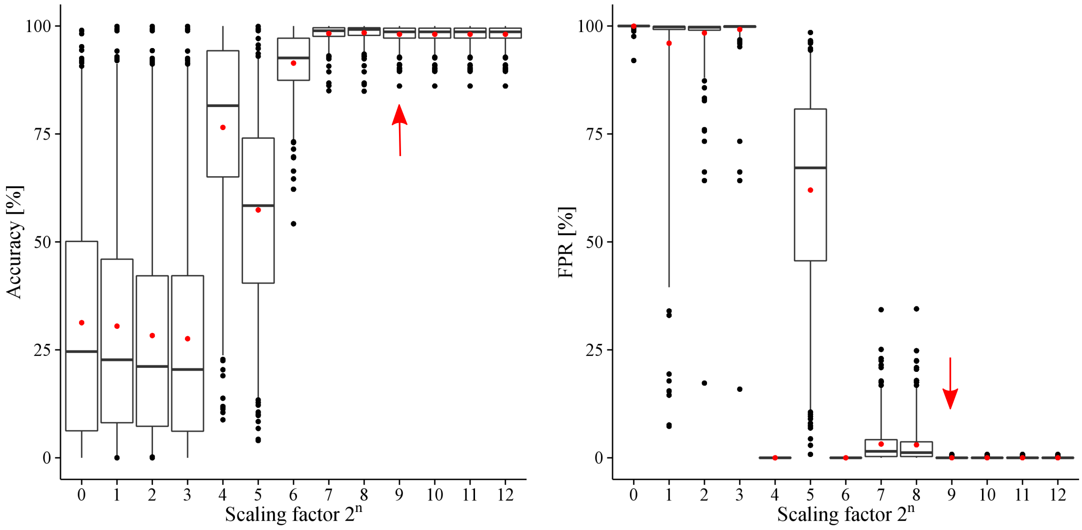
We start by examining the role of the adaptions described in chapter 6.1 used on the ETM+ dataset. We can see in table 6.14 that for the ETM+ dataset the influence is negligible. They are therefore used in further examinations. Both the SVM coefficients and the physical magnitudes are scaled with a factor 2^n, $n = 0, ...12$ and the result is rounded to integers in order to take advantage of the LUT. As we can see in figure 6.18 both the distribution of the accuracy and the false positive rate remain constant with low deviation from a factor 2^9 on. This factor is used for the implementation of the SVM classifier on an FPGA platform.

	mean [%]	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
accuracy [%]	98.1	93.0	97.2	98.7	99.6	100.0
false positive rate [%]	0.0	0.0	0.0	0.0	0.0	0.1

Table 6.14: This table shows the distribution of both the accuracy and false positive rate for all examined ETM+ scenes using all simplifications given in chapter 6.1.

We now consider the two scenes with the greatest deviation of the accuracy from the “correct” SVM classifier. The scene with the greatest deviation is captured over southern Canada in June (LE70380232002154EDC00). This scene was captured over agriculturally cultivated land beyond thick cloud layers and thin haze. The SVM classifier detects both cloud families well (cloud coverage: 52.4 %) where even very thin haze is found correctly (see figure 6.19(a)). Figure 6.19(b) shows the difference between the “correct” SVM classification and the “adapted” one where the adaptions described in chapter 6.1 give a value for the cloud coverage of 61.4 % yielding an accuracy of 86.1 %. Red pixels mark false positive pixels and white respectively black ones mark true positive respectively true negative classified pixels. Note that no false negative pixels are found. The false positive rate is 21.0 %. Most of the wrongly classified pixels lie on the fringes of the border between “Cloud” and “Non-Cloud” areas or in the misinterpretation of agricultural land cover below haze.

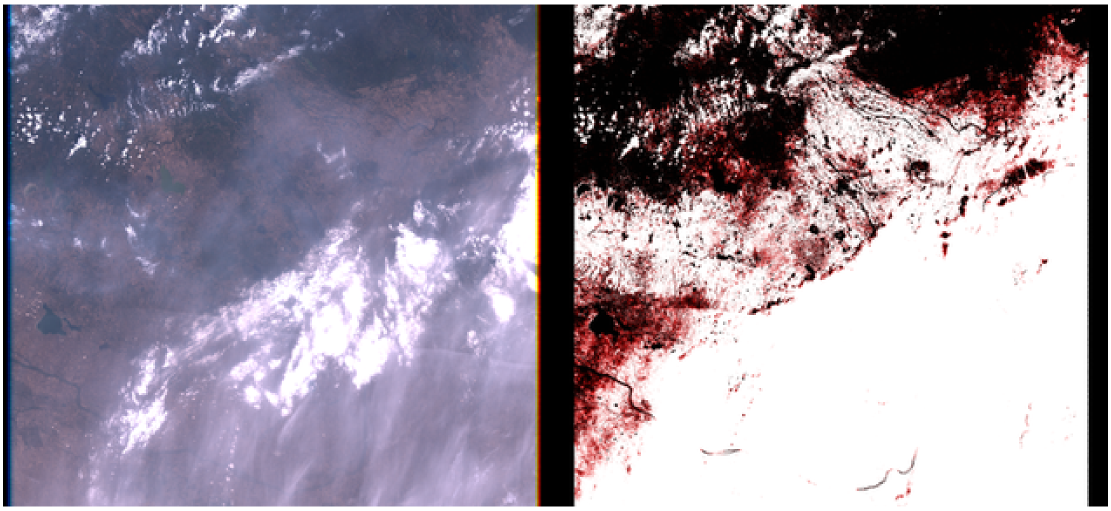
The next greatest outlier is captured over northern USA in May (LE70300282002130EDC00). Here again we have agriculturally cultivated land, but in contrast to the latter scene we only have a few thick clouds and some cirrus clouds (see figure 6.20(a)). The correct classifier again finds a satisfying solution, reporting a cloud coverage of 22.8 %. Using the



(a) **(b)**

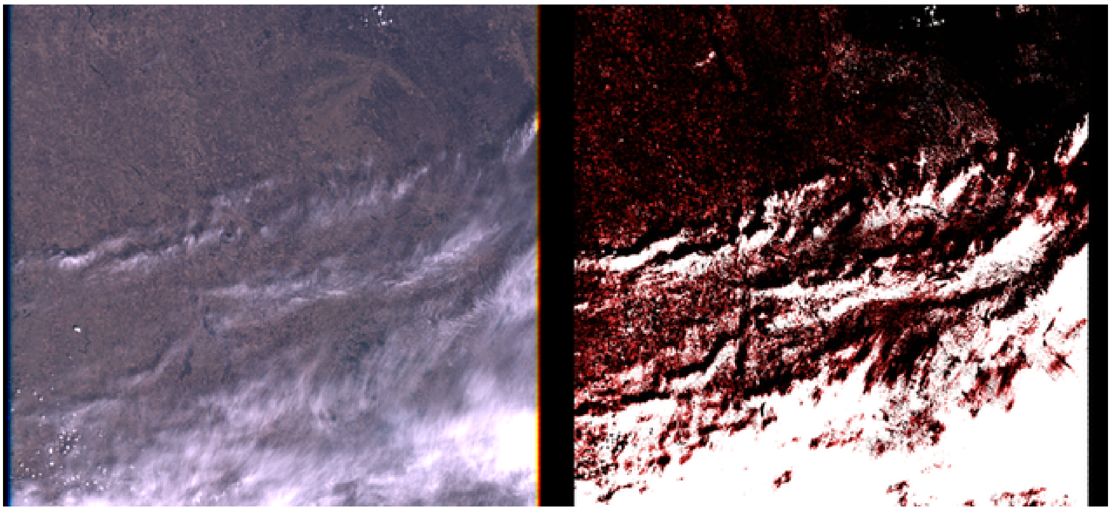
Figure 6.18: Plot (a) shows the distribution of the accuracy with respect to the scaling factor 2^n ($n = 0, \ldots, 12$) of the scaled TOA-reflectances and at-sensor temperatures and plot (b) shows the corresponding distribution of the false positive rate for the ETM+ data set. Starting at a scaling factor of 2^9 both the accuracy and the false positive rate start to remain constant.

adapted SVM classifier a cloud coverage of 30.1 % (accuracy: 89.5 %) is reported with many false positive pixels (marked red) yielding a false positive rate of 31.5 % (see figure 6.20(b)). As in the case before, most of the wrongly classified pixels lie on the fringes of the border between "Cloud" and "Non-Cloud" or are scattered in the "Cloud" mask with only little tendency of clustering. Again the false negative rate is 0 %.



(a) (b)

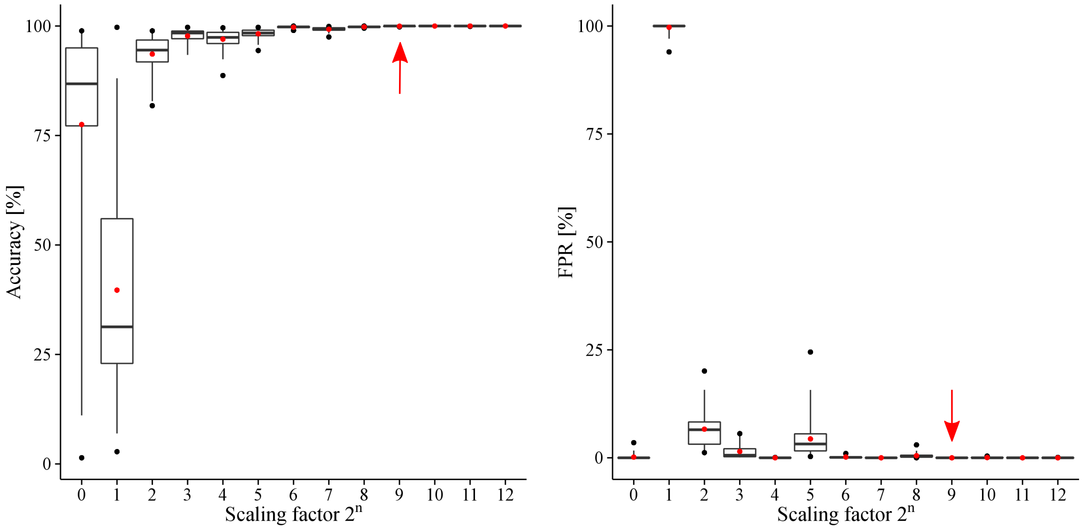
Figure 6.19: Southern Canada in June (LE70380232002154EDC00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask of the SVM classifier (b). Pixels misclassified as “Clouds” (red pixels) lie mainly on the fringes of the clouds. The classifier has an false negative rate of 0 %.



(a) (b)

Figure 6.20: Northern USA in May (LE70300282002130EDC00). On the left side the RGB image of the scene is shown (a) and on the right side the resulting cloud mask (b). Pixels misclassified as “Clouds” (red pixels) lie mainly on the fringes of the clouds, but are also widely scattered around the cloud mask. The classifier has an false negative rate of 0 %.

The analysis of the MODIS data set is likewise. After choosing a training set of 2000 points and a cost factor of 0.01 for the training we applied the simplifications given in chapter 6.1. Both the coefficients and the TOA reflectances as well as the TOA radiances are scaled by a factor of 2^n $(n = 0, ..., 12)$ and then rounded to integers in order to avoid floating point arithmetic. Figure 6.21 shows the evolution of both the accuracy and the false positive rate with respect to the scaling factor 2^n. We can see that from a scaling factor of 2^9 these values remain approximately constant. The overall accuracy and false positive rate for the scaling in addition to the simplifications is given in table 6.15. We can see that both values have only a small deviation from the optimum value and so the "simplified" SVM classifier for MODIS is suitable.



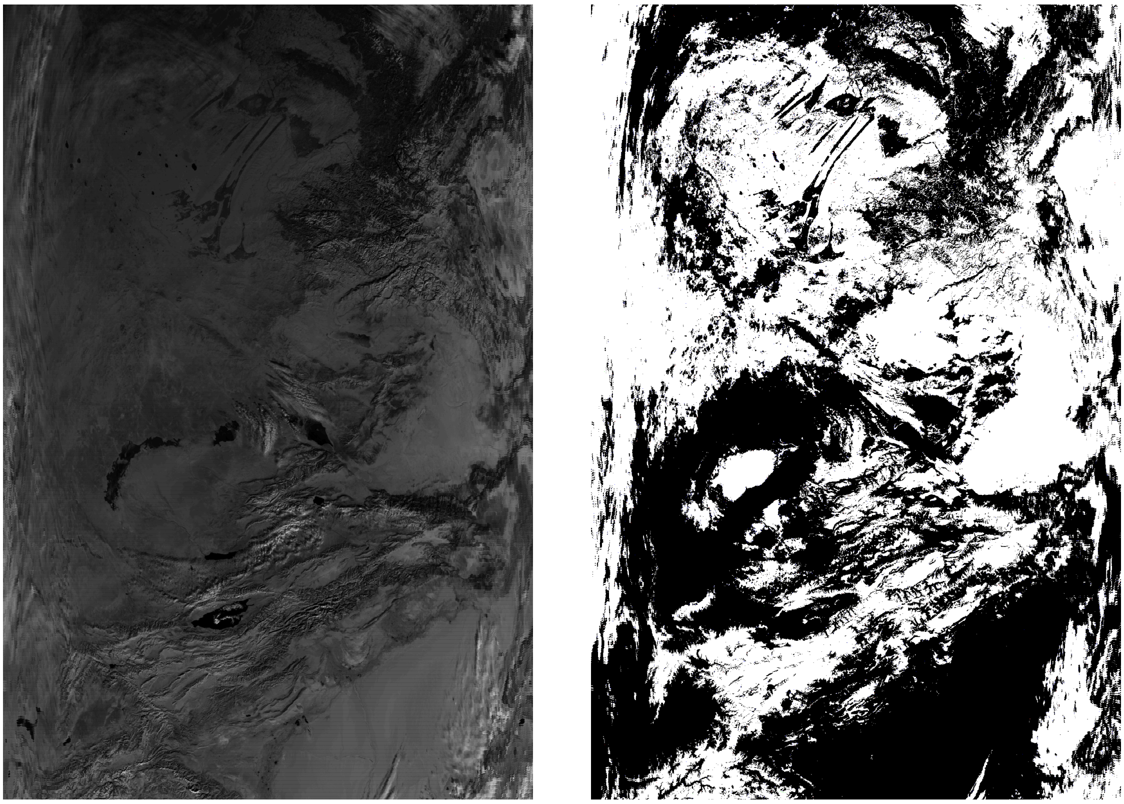
(a) **(b)**

Figure 6.21: Plot (a) shows the distribution of the accuracy with respect to the scaling factor 2^n $(n = 0, ..., 12)$ of the scaled radiances and plot (b) shows the corresponding distribution of the false positive rate for the MODIS data set. Starting at a scaling factor of 2^9 both the accuracy and the false positive rate start to remain constant.

	mean	$Q_{0.025}$	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	$Q_{0.975}$
accuracy [%]	99.9	99.8	99.9	100.0	100.0	100.0
false positive rate [%]	0.0	0.0	0.0	0.0	0.0	0.0

Table 6.15: This table shows the distribution of both the accuracy and false positive rate for all examined MODIS scenes using all simplifications given in the text.

The greatest outlier of the dataset classified by the SVM classifier is the same as in the LDA case, taken over eastern Kazakhstan in autumn (MOD021KM.A2005357.0540.005.2010166204718, see figure 6.22(a)). Again the classifier tends to misinterpret desert and mountains as clouds as can be seen in figure 6.22(b). The overall accuracy of the cloud coverage of the simplified classifier with respect to the correct one is reported with 99.8 %, having a false positive rate of 0.0 % (red pixels) and a false negative rate of 0.3 % (blue pixels).



(a) **(b)**

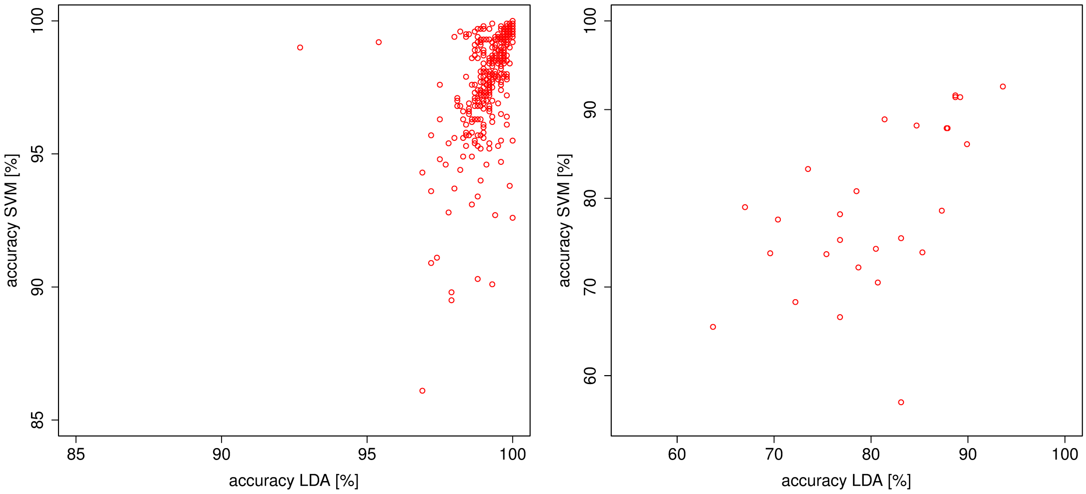
Figure 6.22: Eastern Kazakhstan in autumn (MOD021KM.A2005357.0540.005.2010166204718). On the left side (a) we can see the contrast stretched image of band 5 and on the right side the corresponding cloud mask (b). The SVM classifier misinterprets desert and mountains as clouds due to the trained data set that is not specialized for these features. The wrongly classified pixels (red: false positive, blue: false negative) are mostly extensions of the “Clouds” and do not occur only on the fringes. They also tend to build clusters. Only few misclassified pixels lie scattered randomly around the cloud mask.

6.3.4 Discussion

This chapter compares the performance of the LDA classifier to the SVM classifier, both using the adaption of the data needed for the implementation of the algorithms on an FPGA described in chapter 6.1. These two classifiers both use the same decision function

$$g(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{6.8}$$

with **w** and b derived using different methods.
The advantage of the linear classifiers is that they can operate in real-time, i.e. assuming we have parallel access to the n-dimensional input data stream the output can be calculated instantly because n additions per scene pixel need to performed. The performance of the two classifiers is different in the ETM+ case (see figure 6.23(a)), whereas in the MODIS case nearly no dependence on the accuracy on the classifier is noticeable (see figure 6.23(b)). The better performance of the LDA classifier in the ETM+ case can be explained by the fact that this classifier assumes a Gaussian distribution for the “Cloud” and “Non-Cloud” classes, whereas the SVM classifier tries to separate the two classes by assuming they are linearly separable. Having a look at figures 6.11 and 6.12 it is unlikely that the data is linearly separable, but also the assumption of a Gaussian distribution is risky. But the fact that we can give a likelihood estimation for the belonging of a scene pixel to a certain class makes the classifier more reliable, even if the underlying model fits the underlying data insufficiently. Of course there would have been the opportunity of using another kernel than the linear one in the SVM classification (see chapter 5.2.2) which may improve the performance of this classifier. Then however we have the disadvantage of evaluating the decision function not over the number of features but over the number of support vectors, where the number of support vectors is generally larger than the number of features (see e.g. table 6.10). In this case the hardware consumption would strongly increase due to the significantly higher number of summands (a rough estimation gives us a number of about 200 slices per summand, see table 6.16). Also, when using a Gaussian kernel, the exponential function must be evaluated by a series expansion which again would increase the number of summands.
The independence of the performance of the classifier in the MODIS case could be explained with the *curse of dimensionality.* Actually the classifiers should be able to yield a better accuracy due to the larger amount of information per scene pixel provided



(a) **(b)**

Figure 6.23: Plot (a) shows the accuracy of LDA vs. the accuracy of SVM for the analyzed ETM+ scenes, and plot (b) shows the corresponding distribution of the examined MODIS data set.

by the larger number of spectral bands used, but the curse of dimensionality means that with linear increasing number of features the number of training points necessary for minimal training error increases exponentially. So obviously the number of training points (2000 in the MODIS case) was not enough. Again it must be emphasized that the selection of an optimal training set is not within the scope of this thesis, but it can be seen that even with this training set the accuracy still reaches an acceptable value.

In table 6.16 the hardware consumption of the linear classifiers is listed for the evaluation of ETM+ data for the LDA case. The comparison to the full ACCA algorithm shows a significant difference, thus ACCA needs about a factor 2 more hardware. The clock rate for the linear classifiers is limited to 43 MHz because of the wide bit depth of the single factors inducing a long flip flop to flip flop time of 23 ns. Considering an estimated data flow ($N_S/\textit{recording time of the scene}$) of about 1.65 MPixels/s per spectral channel (for ETM+) and 9 kPixels/s per spectral channel (for MODIS) we can see that this clock rate is more than sufficient for analyzing the data on-the-fly.

	linear classifier	complete ACCA
Slices	1388	2895
LUTs	2170	4779
Slice Flip Flops	576	1808
MULT 18X18	0	17
RAMB16	7	7

Table 6.16: This table shows the hardware consumption for the linear classifiers for ETM+ in comparison to the hardware consumption of the implementation of the complete ACCA algorithm. The magnitude of the hardware consumption is equal for both LDA and SVM because of the same scaling factor.

7 Conclusion and Outlook

Automatic cloud detection on-board of satellites is an advantageous application for upcoming remote sensing satellite missions. The on-board analysis of captured images can save valuable resources like storage capacity and downlink times. This results in a better utilization of the satellite and therefore maximizes the scientific output.

In this thesis a method was developed which allows fast execution of several cloud detection algorithms that can be applied to different multispectral sensors. As a preprocessing step for the algorithms the digital output of the sensor is converted into meaningful physical magnitudes TOA-radiance, TOA-reflectance and at-sensor temperature. This conversion only depends on a low number of variables per spectral band, therefore the results are precalculated and stored in a Look-up table (LUT). This offers an instantaneous access to the results without any calculations necessary on board. In order to minimize the size of the LUT the set of variables is discretized so that the input space becomes $\mathbb{N}^n$ instead of $\mathbb{R}^n$, where n is the number of features like e.g. spectral bands. The algorithms which use these methods are implemented on a Field Programmable Gate Array (FPGA) platform which allows direct access to the sensor data during the capture of an image and therefore on-the-fly analysis of the data. In addition the benefit of FPGAs lies in their low costs and low power consumption. FPGAs show their strengths when dealing with integer numbers. However, physical values are in general real numbers. Therefore, all occurrences of real numbers are additionally scaled by a certain factor and then rounded to integers. The influences of both discretizing the set of the variables and the subsequent scaling on the accuracy are examined in this thesis.

First the developed method is applied to the operational cloud detection algorithm for the Enhanced Thematic Mapper (ETM+) sensor developed by Irish [26]. This algorithm applies a number of threshold tests for each pixel and then analyses the temperature distribution of the clouds found. Depending on this distribution a final cloud mask is generated which takes the plausibility of a pixel being cloudy or not from the temperature distribution. With respect to the implementation without LUTs an accuracy of 97.5 % is reached. El-Araby et al. [18] only implemented the threshold part. They realised the complete conversion of the data to radiances on an FPGA with an accuracy of nearly 100 %, but with twice as much hardware consumption than the implementation of the complete algorithm using LUTs. When implementing only the threshold part using the method developed in this thesis an accuracy of 91 % is reached, but this part needs 20 times less hardware then the implementation of El-Araby et al. [18]. The complete cloud detection algorithm for ETM+ is not able to perform real-time analysis because it has to analyze a complete scene twice. The threshold part can, however, be executed in real-time.

Two supervised learning algorithms were applied to Enhanced Thematic Mapper (ETM+) and Moderate-resolution Imaging Spectroradiometer (MODIS) data. Their implementation on FPGAs using LUTs and the scaling of the data to integers are examined. For ETM+ the accuracy of both algorithms compared to their unscaled counterparts is 99.3 % for Linear discriminant analysis (LDA) and 98.1 % for Support vector machines (SVM). For MODIS we get a corresponding accuracy of 99.3 % for LDA and 99.9 % for SVM.
The advantage of the supervised classifiers is that after determining the coefficients for the classifier the decision function reduces to a linear combination of the physical magnitudes per spectral band obtained by LUTs and the corresponding coefficients. This allows real-time execution of the algorithms on the FPGA with low hardware consumption, for ETM+ e.g. half as much hardware as for the complete cloud detection algorithm developed by Irish [26] is necessary. The results obtained in this thesis show that an implementation of cloud detection algorithms using precalculated physical magnitudes stored in LUTs on FPGAs is possible with sufficient accuracy. Simple threshold algorithms and supervised classifiers even allow real-time processing of the incoming data, thus giving a cloud mask instantaneously after the complete scene is captured.

For upcoming remote sensing satellite missions the proposed on board cloud detection can be applied as a preprocessing step for a change detection method applied e.g. to the identification of natural disasters like land slides. Having an adequate camera an on-board planning mechanism [43] can immediately provoke the recording of a new image of the scenery with higher resolution which then can be downlinked with high priority and promptly forwarded to the corresponding crisis information centers (e.g. Center for

Satellite Based Crisis Information (ZKI) of the DLR).

The on-board cloud detection in combination with the on-board planning tool can improve the workload of a remote sensing satellite. Instead of executing a static timeline that involves a fixed order of capturing images, storing and downlinking them, an image can be analyzed instantaneously and discarded when the amount of cloud coverage exceeds a given threshold. This allows the on-board planning mechanism to provoke the recording of a new image which results in only downlinking images that are useful for the user because they are not contaminated with clouds.

Bibliography

[1] HDF Documentation. `http://www.hdfgroup.org/documentation/`. [Online; accessed 2-February-2013].

[2] *Strahlungsphysik im optischen Bereich und Lichttechnik: Benennung der Wellenlängenbereiche.* Deutsche Normen. Beuth, 1984.

[3] S. Ackerman, K. Strabala, W. Menzel, R. Frey, C. Moeller, and L. Gumley. Discriminating clear sky from clouds with MODIS. *Journal of Geophysical Research*, 103 (D24):32–141, 1998.

[4] S. Ackerman, K. Strabala, P. Menzel, R. Frey, C. Moeller, L. Gumley, B. Baum, S. Seemann, and H. Zhang. Discriminating Clear-Sky From Cloud With MODIS Algorithm Theoretical Basis Document (MOD35), Version 5.0. *Coop. Inst. for Meteorol. Satell. Stud., Univ. of Wis., Madison*, 2006.

[5] U. Amato, A. Antoniadis, V. Cuomo, L. Cutillo, M. Franzese, L. Murino, and C. Serio. Statistical cloud detection from SEVIRI multispectral images. *Remote Sensing of Environment*, 112(3):750–766, 2008.

[6] Aqua Homepage. Homepage of the Aqua satellite. URL `http://aqua.nasa.gov/`. [Online; accessed 5-November-2013].

[7] T. Arvidson, J. Gasch, and S. N. Goward. Landsat 7's long-term acquisition plan—An innovative approach to building a global imagery archive. *Remote Sensing of Environment*, 78(1):13–26, 2001.

[8] P. M. Atkinson and A. Tatnall. Introduction neural networks in remote sensing. *International Journal of remote sensing*, 18(4):699–709, 1997.

[9] L. Beaudoin, J.-M. Nicolas, F. Tupin, and M. Hueckel. Introducing spatial information in k-means algorithm for cloud detection in optical satellite images. In *Europto Remote Sensing*, pages 67–77. International Society for Optics and Photonics, 2001.

[10] G. Bisht, V. Venturini, S. Islam, and L. Jiang. Estimation of the net radiation using MODIS (Moderate Resolution Imaging Spectroradiometer) data for clear sky days. *Remote Sensing of Environment*, 97(1):52–67, 2005. ISSN 0034-4257. doi: 10.1016/j.rse.2005.03.014. URL `http://www.sciencedirect.com/science/article/pii/S003442570500132X`.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] A. D. Collado, E. Chuvieco, and A. Camarasa. Satellite remote sensing analysis to monitor desertification processes in the crop-rangeland boundary of Argentina. *Journal of Arid Environments*, 52(1):121–133, 2002.

[13] Composition of Atmosphere. Pie chart of the composition of the atmosphere. URL `http://www.wikiprogress.org/index.php/Atmosphere`. [Online; accessed 21-November-2013].

[14] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines and other kernel-based learning methods.* Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

[15] Deutscher Wetterdienst. Internationaler Wolkenatlas, 1990. URL `http://www.deutscher-wetterdienst.de/lexikon/download.php?file=Wolkenatlas.pdf`. [Online; accessed 23-June-2015].

[16] *Digilent Nexys2 Board Reference Manual.* Digilent, 2011.

[17] R. O. Duda, D. G. Stork, and P. E. Hart. *Pattern classification and scene analysis. Part 1, Pattern classification.* Wiley, 2nd edition, Nov. 2000. ISBN 0471056693.

[18] E. El-Araby, T. El-Ghazawi, J. Moigne, and R. Irish. Reconfigurable processing for satellite on-board automatic cloud cover assessment. *Journal of Real-Time Image Processing*, 4(3):245–259, 2009. doi: 10.1007/s11554-008-0107-8.

[19] EOS Homepage. Homepage of the NASA's EOS program. URL `http://eospso.gsfc.nasa.gov/`. [Online; accessed 5-November-2013].

[20] W. E. Esaias, M. R. Abbott, I. Barton, O. B. Brown, J. W. Campbell, K. L. Carder, D. K. Clark, R. H. Evans, F. E. Hoge, H. R. Gordon, et al. An overview of MODIS capabilities for ocean science observations. *IEEE Transactions on Geoscience and Remote Sensing*, 36(4):1250–1265, 1998.

[21] L. Gómez-Chova, J. Amorós, G. Camps-Valls, J. D. Martin, J. Calpe, L. Alonso, L. Guanter, J. C. Fortea, and J. Moreno. Cloud detection for CHRIS/Proba hyperspectral images. In *Remote Sensing*, pages 59791Q–59791Q. International Society for Optics and Photonics, 2005.

[22] D. K. Hall, G. A. Riggs, and V. V. Salomonson. Development of methods for mapping global snow cover using moderate resolution imaging spectroradiometer data. *Remote Sensing of Environment*, 54(2):127 – 140, 1995. ISSN 0034-4257. doi: 10.1016/0034-4257(95)00137-P.

[23] A. Huete, K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote sensing of environment*, 83(1):195–213, 2002.

[24] J. Hummel. Surface temperature sensitivities from cloud cover variations in the Hummel-Kuhn radiative—convective model with three different cloud approximations. *Tellus*, 33(5):438–443, 1981.

[25] R. Irish, J. Barker, S. Goward, and T. Arvidson. Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm. *Photogrammetric engineering and remote sensing*, 72(10):1179, 2006.

[26] R. R. Irish. Landsat 7 automatic cloud cover assessment. In *AeroSense 2000*, pages 348–355. International Society for Optics and Photonics, 2000.

[27] Y. J. Kaufman, I. Koren, L. A. Remer, D. Rosenfeld, and Y. Rudich. The effect of smoke, dust and pollution aerosol on shallow cloud development over the Atlantic Ocean. *Proceedings of the National Academy of Sciences of the United States of America*, 102(32):11207–11212, 2005.

[28] F. Kesel and R. Bartholomä. *Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs.* Oldenbourg Lehrbücher für Ingenieure. Oldenbourg, 2006. ISBN 9783486575569.

[29] A. Kokhanovsky. Optical properties of terrestrial clouds. *Earth-Science Reviews*, 64 (3):189–241, 2004.

[30] P. Kronberg. *Fernerkundung der Erde.* Ferdinand Enke Verlag Stuttgart, 1985.

[31] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(2):203–215, 2007.

[32] Landsat 7 information page, NASA. Schematic image of landsat 7. URL `http://geo.arc.nasa.gov/sge/landsat/l7d.html`. [Online; accessed 5-November-2013].

[33] C. Latry, C. Panem, and P. Dejean. Cloud detection with SVM technique. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 448–451. IEEE, 2007.

[34] J. Li, W. P. Menzel, Z. Yang, R. A. Frey, and S. A. Ackerman. High-spatial-resolution surface and cloud-type classification from MODIS multispectral band measurements. *Journal of Applied Meteorology*, 42(2):204–226, 2003.

[35] W. Liu and F. Kogan. Monitoring regional drought using the vegetation condition index. *International Journal of Remote Sensing*, 17(14):2761–2782, 1996.

[36] I. V. Z. Marais, J. A. Du Preez, and W. H. Steyn. An optimal image transform for threshold-based cloud detection using heteroscedastic discriminant analysis. *International Journal of Remote Sensing*, 32(6):1713–1729, 2011.

[37] Mathworks. Homepage, 2014. URL `http://de.mathworks.com/help/fixedpoint/ug/fixed-point-numbers_f20705.html`. [Online; accessed 17-December-2014].

[38] D. Meyer and T. U. Wien. Support Vector Machines. The Interface to libsvm in package e1071. Online-Documentation of the package e1071 for R, 2001.

[39] L. Murino, U. Amato, M. F. Carfora, A. Antoniadis, B. Huang, W. P. Menzel, and C. Serio. Cloud detection of MODIS multispectral images. *Journal of Atmospheric and Oceanic Technology*, (2013), 2013.

[40] *Landsat 7 Science Data Users Handbook*. NASA.

[41] NASA. Schematic Image of the "Terra" satellite. URL `http://www.nasa.gov/images/content/643743main_terra_instruments_full.jpg`.

[42] *Landsat 7 System Zero-R Distribution Handbook Data Format Control Book*. NASA, July 1999.

[43] R. Nibler. Missionsplanung in Echtzeit an Bord von Satelliten. PhD thesis in progress, March 2014.

[44] K. B. Petersen and M. S. Pedersen. The matrix cookbook, 2008. URL `{orion.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf}`. [Online; accessed 12-March-2013].

[45] R. Roger and M. Yau. *A Short Course in Cloud Physics*. Pergamon Press, 3rd edition, 1989.

[46] R. Saunders and K. Kriebel. An improved method for detecting clear sky and cloudy radiances from AVHRR data. *International Journal of Remote Sensing*, 9(1):123–150, 1988.

[47] R. W. Saunders. An automated scheme for the removal of cloud contamination from AVHRR radiances over western Europe. *International Journal of Remote Sensing*, 7 (7):867–886, 1986. doi: 10.1080/01431168608948896. URL `http://www.tandfonline.com/doi/abs/10.1080/01431168608948896`.

[48] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive computation and machine learning. MIT Press, 2002. ISBN 9780262194754.

[49] J. Spencer. Fourier series representation of the position of the sun. *Search*, 2(5):172, 1971.

[50] K. I. Strabala. *MODIS cloud mask user's guide*. University of Wisconsin–Madison, 2005.

[51] Terra Homepage. Homepage of the Terra satellite. URL `http://terra.nasa.gov/`. [Online; accessed 5-November-2013].

[52] G. Toller, A. Isaacman, M. Leader, and V. Salomonson. MODIS Level 1B Product User's Guide. 2003.

[53] V. Vapnik. *The nature of statistical learning theory.* Springer, 2000.

[54] P. von Hippel. Mean, median, and skew: Correcting a textbook rule. *Journal of Statistics Education*, 13(2):n2, 2005.

[55] M. von Schönermark. Vorlesungsskript Raumfahrtanwendungen. Vorlesung, Institut für Raumfahrtsysteme, Universität Stuttgart, 2009.

[56] Wikipedia. Solar Radiation Spectrum, 2013. URL `https://commons.wikimedia.org/wiki/File:Solar_spectrum_en.svg`. [Online; accessed 02-December-2013]. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license (https://creativecommons.org/licenses/by-sa/3.0/deed.en) by Arbeck.

[57] Wikipedia. Cumulus clouds, 2014. URL `https://commons.wikimedia.org/wiki/File:Cumulus_clouds_in_fair_weather.jpeg`. [Online; accessed 14-January-2014]. This file is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic license (https://creativecommons.org/licenses/by-sa/2.0/deed.en) by Michael Jastremski.

[58] Wikipedia. Cloud species, 2014. URL `https://en.wikipedia.org/wiki/File:Wolkenstockwerke.png`. [Online; accessed 14-January-2014].

[59] Wikipedia. High clouds, 2014. URL `https://commons.wikimedia.org/wiki/File:CirrusUncinusWithPlane.jpg`. [Online; accessed 14-January-2014]. This file is licensed under the Creative Commons Attribution-Share Alike 2.0 Germany license (https://creativecommons.org/licenses/by-sa/2.0/de/deed.en) by Simon Eugster.

[60] Wikipedia. Middle clouds, 2014. URL `https://en.wikipedia.org/wiki/File:Misc_clouds_santa_clarita.JPG`. [Online; accessed 14-January-2014]. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 license (https://creativecommons.org/licenses/by-sa/3.0/) by Chevy111.

[61] Wikipedia. Vertical clouds, 2014. URL `https://commons.wikimedia.org/wiki/File:Big_Cumulonimbus.JPG`. [Online; accessed 22-January-2014]. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license (https://creativecommons.org/licenses/by-sa/3.0/deed.en) by Simon Eugster.

[62] J. A. Williams, A. S. Dawood, and S. J. Visser. FPGA-based cloud detection for real-time onboard remote sensing. In *Field-Programmable Technology, 2002.(FPT). Proceedings. 2002 IEEE International Conference on*, pages 110–116. IEEE, 2002.

[63] World Meteorological Organization. Homepage, 2014. URL `http://www.wmo.int/pages/index_en.html`. [Online; accessed 14-January-2014].

[64] *DS312 Spartan-3E Generation FPGA Family Data Sheet.* Xilinx, 2011.

[65] *UG331 Spartan-3 Generation FPGA User Guide.* Xilinx, 2011.

[66] Z. Zhu and C. E. Woodcock. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sensing of Environment*, 118:83–94, 2012.

List of Abbreviations

N_S	Number of scene pixels
ρ	Reflectance
ACCA	Advanced Cloud Cover Assessment
ASIC	Application Specific Integrated Circuit
AU	Astronimical unit
AVHRR	Advanced Very High Resolution Radiometer
CLB	Configurable Logic Block
CPF	Calibration Parameter File
DEM	Digital elevation model
DOY	Day of year
ECS	Earth Observing System Data and Information System (EOSDIS) Core System
EDC	EROS Data Center
EOS	Earth Observation System
EOSDIS	Earth Observing System Data and Information System
EROS	Earth Resources Observation and Science
ETM+	Enhanced Thematic Mapper
fn	False negative value (pixel is wrongly classified as “Non-Cloud”)
fp	False positive value (pixel is wrongly classified as “Cloud”)

FPGA	Field Programmable Gate Array
HDF	Hierarchical Data Format
ISCCP	International Satellite Cloud Climatology Project
L	Radiance $\left[\frac{\mathrm{W}}{\mathrm{m}^2\mathrm{sr}\mu\mathrm{m}}\right]$
LDA	Linear discriminant analysis
LPS	Landsat Processing System
LTAP	Long-term Acquisition Plan
LUT	Look-up table
MODIS	Moderate-resolution Imaging Spectroradiometer
MSS	Multispectral scanner
NIR	Near Infrared from 0.75 µm to 1.4 µm [2]
NOAA	National Oceanic and Atmospheric Administration
NSDIC	National Snow and Ice Data Center
OBC	On-board computer
PAN	Panchromatic
PPU	Payload processing unit
RAM	Random access memory
SSO	Sun synchronous orbit
SSR	Solid state recorder
SVM	Support vector machines
SWIR	Short Wave Infrared from 1.4 µm to 3 µm [2]
TIR	Thermal Infrared from 8 µm to 15 µm [2]

tn	True negative value (pixel is correctly classified as “Non-Cloud”)
TOA	Top of atmosphere
tp	True positive value (pixel is correctly classified as “Cloud”)
USGS	United States Geological Survey
VIS	Visible from 0.38 µm to 0.75 µm [2]

Glossary

accuracy

The accuracy value indicates the ratio of the number of all correctly classified pixels to the total number of pixels. It is defined as $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$.

boxplot

A plot showing in an intuitive way the distribution of underlying data. 50% of the data are distributed in the box and 95% are distributed in between the ends of the two vertical lines. The black dots show the outliers, the red dot is the mean value of the data and the horizontal line marks the median value.

false negative rate

Indicates the ratio of the number of pixels that are wrongly labeled as "Non-Cloud" but actually are "Cloud" pixels to the number of all "Cloud" pixels. It is defined as $fnrate = \frac{fn}{fn+tp}$.

false positive rate

Indicates the ratio of the number of pixels that are wrongly labeled as "Cloud" but actually are "Non-Cloud" pixels to the number of all "Non-Cloud' pixels. It is defined as $fprate = \frac{fp}{fp+tn}$.

NDSI

Normalized Difference Snow Index:
Due to the fact that the reflectance of clouds and snow is similar in the green spectral band, but the reflectance of clouds is high in the Short Wave Infrared from 1.4 µm to 3 µm [2] (SWIR) (band 5 for ETM+) spectral band and low for snow in the SWIR

spectral band, an index can be formulated as

$$NDSI = \frac{\rho_{green} - \rho_{SWIR}}{\rho_{green} + \rho_{SWIR}}.$$

Values above a certain threshold can therefore point to a snowy pixel.

precision

The precision value indicates the ration of all positive correct classified objects in respect to all positive classified objects. It is defined as $precision = \frac{tp}{tp+fp}$.

WRS

World Reference System:
A standard that was defined for Landsat 4 and Landsat 5 (preserved for Landsat 7) where the Earth surface was divided in a global grid system with 233 paths and 248 rows.

A Detailed description of the ACCA algorithm

The following description is mainly adapted from [40, pp. 135-141] and [25] including the flow charts.
The ACCA algorithm can be split into 4 parts (see figure A.1):

i) Pass-1 Spectral Cloud Identification
ii) Band 6 Cloud Signature Development
iii) Pass-2 Thermal Band Cloud Separation
iv) Image-Based Cloud-Cover Assignments and Aggregation

As input level-0 (L0) data is used, i.e. the data is neither radiometrically nor geometrically is used. The grayscale images delivered in the L0 product are converted into TOA-reflectances ρ_2-ρ_5 for bands 2-5 and at-sensor-temperature T_6 for band 6 as described in chapter 3.1.

i) Pass-1 Spectral Cloud Identification

The first pass examines each pixel in the scene using thresholds and labels it as one of the four classes “Non-Cloud”, “Warm-Cloud”, “Cold-Cloud” or “Ambiguous” in

Band	Range [μm]	Resolution [m]
1	0.45-0.52 (blue)	30
2	0.53-0.61 (green)	30
3	0.63-0.69 (red)	30
4	0.78-0.90 (NIR)	30
5	1.55-1.75 (SWIR)	30
6	10.4-12.5 (TIR)	60
7	2.09-2.35 (SWIR)	30
8	0.52-0.90 (PAN)	15

Table A.1: This table lists the spectral channels of the ETM+ sensor with their corresponding spatial resolutions.

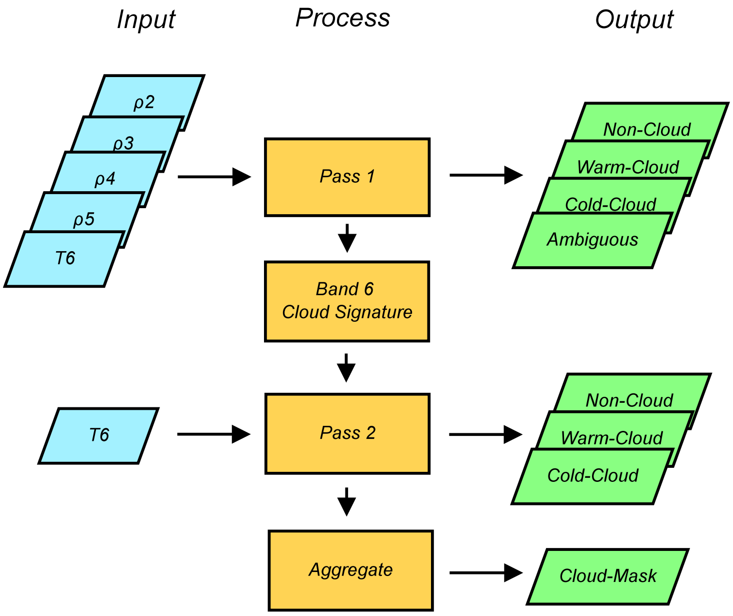


Figure A.1: This chart shows the sequence of the ACCA algorithm with the required inputs of each step on the left side, the single processing steps in the middle and the particular output of each step on the right side [25].

the resulting pass-1 cloud mask. For this task 11 threshold tests are applied for each pixel which will be described in the following (see figure A.2).

- **Filter 1** Brightness Threshold
  First each band 3 pixel is compared to a brightness threshold set to 0.08. If the pixel value exceeds this threshold it is passed to filter 3, otherwise it is passed to filter 2.
- **Filter 2** Non-Cloud/Ambiguous Discriminator, band 3
  The band 3 pixel entering this filter is compared to a threshold set to 0.07. Values above can be a sign of low-reflectance clouds and pixels are therefore labeled as “Ambiguous”, otherwise they are labeled as “Non-Cloud”.
- **Filter 3** Normalized Difference Snow Index (NDSI)
  Due to the fact that reflectances are similar in the green band (band 2) but reflectance of snow in the SWIR band (band 5) is quite low whereas the reflectance of clouds in this band is also quite high. So [22] proposed the NDSI as

$$NDSI = \frac{\rho_2 - \rho_5}{\rho_2 + \rho_5}. \tag{A.1}$$

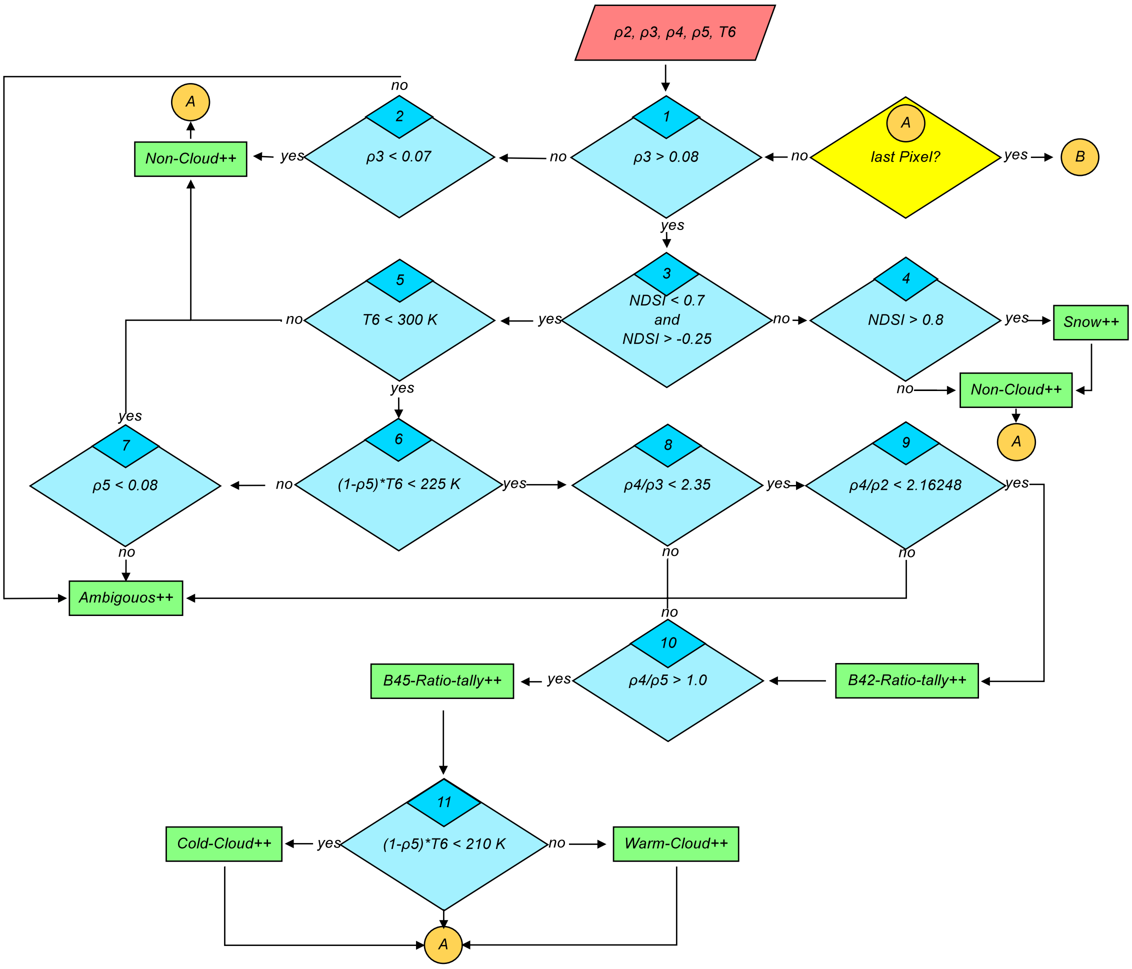


Figure A.2: Pass-1 Spectral Cloud Identification [25]

Pixels having an NDSI in between of -0.25 and 0.7 are passed to filter 5 where pixels outside this range are labeled as “Non-Cloud” and are passed to filter 4.

- **Filter 4** Snow Threshold
  Pixels whose NDSI is above 0.8 are additionally labeled as snow pixels here.
- **Filter 5** Temperature Threshold
  Band 6 values (in K) are used to identify potential clouds. A realistic cloud temperature maximum of 300 K is assumed so that pixels which exceed this value are labeled as “Non-Cloud”, otherwise they are passed to filter 6.
- **Filter 6** Band 5/6-Composite

The Band 5/6-Composite is expressed as

$$\text{Band 5/6 Composite} = (1 - \rho_5) \cdot T_6.$$

This filter works well because clouds have low temperatures and are highly reflective in band 5. It is quite useful for eliminating cold land surfaces that have low reflectance in the SWIR band (band 5) like snow and tundra. Pixels that fall below a threshold of 225 K are passed to filter 8 as possible clouds whereas pixels above this threshold are passed to filter 7

- **Filter 7** Non-cloud/Ambiguous Discriminator, Band 5
  Each band 5 pixel entering this filter is compared to a threshold of 0.08. Pixels above this value are labeled as “Ambiguous” (low-reflectance clouds), whereas pixels below this value are classified as “Non-Cloud” (maybe water).
- **Filter 8** Band 4/3-Ratio for Growing Vegetation
  In the near infrared (band 4) reflectance for green leaves is high because very little energy is absorbed whereas the chlorophyll in the green leaves absorbs the red portion of the sunlight so that the reflectance in the red band (band 3) is low. By simply dividing the band 4 reflectance by the band 3 reflectance we can find growing vegetation. A threshold of 2 is used. Pixels that exceed this value are labeled “Ambiguous”, otherwise they are passed to filter 9.
- **Filter 9** Band 4/3-Ratio for Senescing Vegetation
  Dying and dead leaves absorb even less energy in the near-infrared and are therefore highly reflective in band 4. In the green region the leaves also absorb less energy because of chlorophyll loss so that the reflectivity is also increased. It is found out that the band 4 divided by band 3 ratio is higher for vegetation than for other features including clouds. Pixels that exceed the threshold of 2.16248 are labeled “Ambiguous”, the rest is passed to filter 10.
- **Filter 10** Band 4/5-Ratio for Soil
  Rocks and sand tend to have a higher reflectivity in band 5 than in band 4, whereas the opposite is true for clouds. Using a threshold of 1 one can label pixels below this value as “Ambiguous”, the rest is passed to filter 11.
- **Filter 11** Band 5/6-Composite for Cold and Warm Clouds
  All pixels that reach this filter are classified as clouds. The Band 5/6-Composite value can be used for a further separation into cold and warm clouds using

a threshold of 210 K. Pixels are labeled as cold cloud if they fall below this threshold.

This result is used to develop two cloud signatures, one for the cold clouds and the other for the conjoined cloud classes.

ii) Band-6 Cloud Signature Development

As a precondition for Pass-2 processing two new band 6 thresholds are determined to which all pixels labeled “Ambiguous” are compared. These thresholds are computed using the Pass-1 cloud temperature statistics including the maximum cloud temperature, the mean temperature, the standard deviation and the temperature histogram skewness.

- **Filter 12** Snow and Desert Indicator
  The presence of desert and snow may cause classification problems. In order to circumvent these problems a more conservative cloud signature is used if snow or desert are present. This means only “Cold-Clouds” from pass-1 are used and the “Warm-Clouds” are re-labeled as “Ambiguous”. This decision depends on two conditions: either the total amount of snow exceeds 1 % of the scene, or the desert indicator, defined as the ratio of potential cloud pixels exiting and entering filter 10, is smaller than 0.5.
- **Filter 13** Pass-1 cloud-free Indicator
  If no clouds are found in pass-1 the complete scene is declared as cloud-free. Otherwise proceed to filter 14.
- **Filter 14** Pass-1 Cold-Cloud, Desert and Mean Cloud Indicator
  In order to continue to pass-2 processing, three conditions have to be met: the cold-cloud-scene percentage must be greater than 0.4 %, the pass-1 cloud temperature mean must be less than 295 K (false positives may occur if the mean temperature is too high) and desert conditions must not exist[1]. Otherwise processing continues at filter 22.
- **Filter 15** Temperature Histogram Negative Skewness
  A shift factor may be necessary if the cloud temperature’s histogram is biased to the warmer tail of the cloud temperature distribution. Hence, in case of a negative skewness no adjustment is needed because there is a steep edge on the

---

[1] There is an error in[25]: the desert indicator must be greater than 0.5 and not smaller!

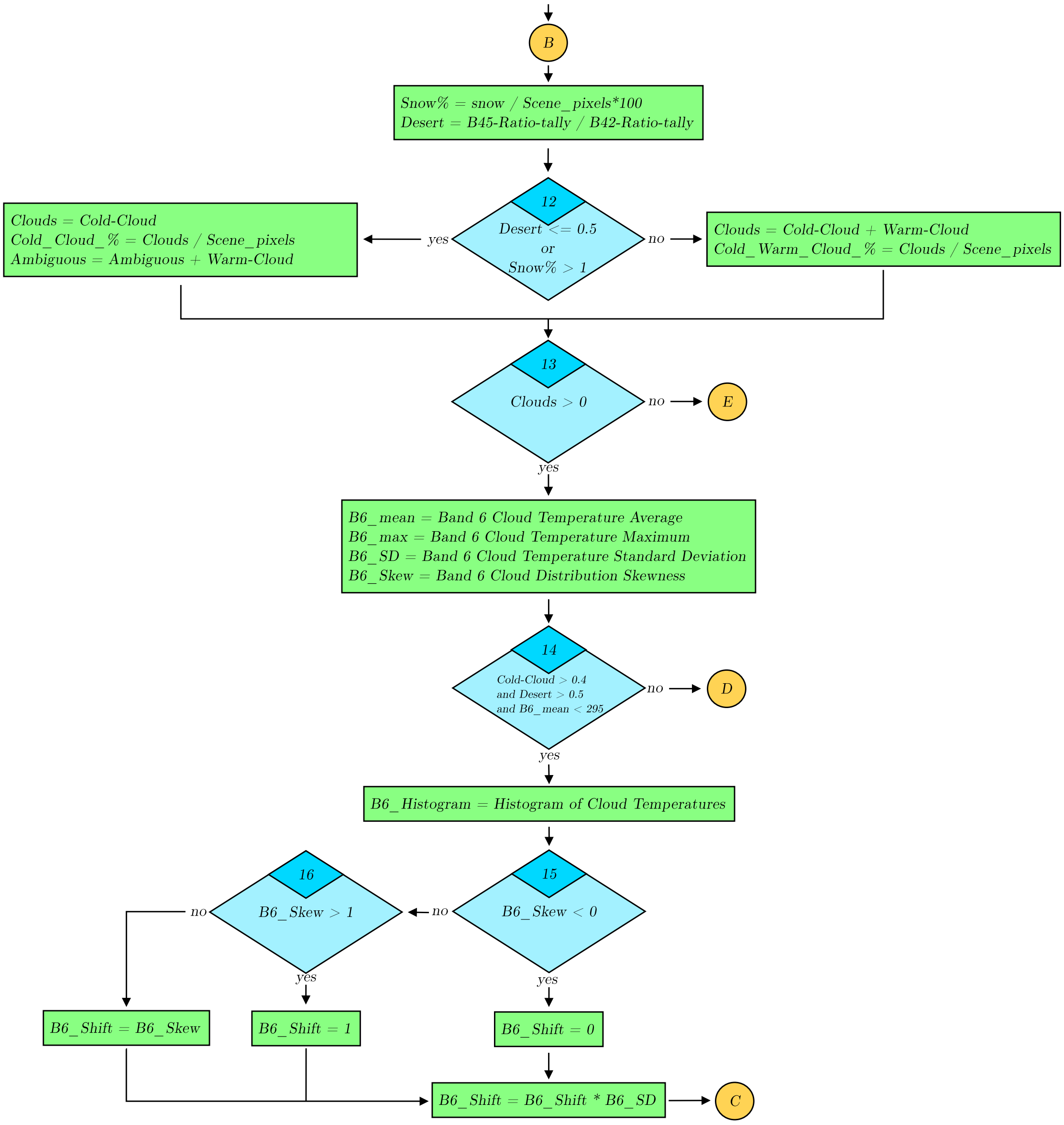


Figure A.3: Band-6 Cloud Signature Development [25] warmer side. Consequently the shift factor is set to 0, otherwise proceed to filter 16.

- **Filter 16** Temperature Histogram Positive Skewness
  If the temperature histogram has a positive skewness there is a bias towards the warmer tail of the cloud temperature distribution. Therefore a shift factor for the adjustment of the Pass-2 threshold is obtained by simply setting it to 1 if the skewness is greater than 1, otherwise it is set to the skewness value.

iii) Pass-2 Thermal Band Cloud Separation

For Pass-2 cloud identification all "Ambiguous" pixels from Pass-1 are re-examined and assigned to one of the classes "Warm-Cloud", "Cold-Cloud" or "Non-Cloud" in order to obtain a second cloud mask. There are two thresholds, one threshold is set low to a low value to generate a conservative estimate of cloud cover and the second value is used to compute a less restrictive estimate. These two thresholds are determined from the pass-1 cloud temperature histogram with start points at the 83.5 and the 97.5 percentile as thresholds. If a band 6 pixel falls below the upper threshold it is labeled as "Warm-Cloud", if it also falls below the lower threshold it is re-labeled as "Cold-Cloud". If it exceeds the upper threshold it is set to "Non-Cloud" because it is probably a false positive.

- **Filter 17** Threshold Shift Deployment
  If the shift factor is positive, upward adjustments are made to compensate for the warm cloud bias. The threshold shift is the product of the shift factor and the cloud temperature standard deviation. Both thresholds are adjusted by this value[2]. If the shift factor is 0 processing continues at filter 19, otherwise proceed to filter 18.
- **Filter 18** Band 6 Maximum Threshold
  A final check is made to see if the new upper threshold exceeds the histogram's 98.75 percentile (a threshold above or near the cloud temperature maximum is unwanted). If this is the case, the 98.75 percentile becomes the new upper threshold and the lower threshold is adjusted by the amount of skewness compensation actually allowed.
- **Filter 19** Band 6 Warm-Cloud Indicator

---

[2] There is an error in [25]: the threshold shift is added to the thresholds and not multiplied by them!

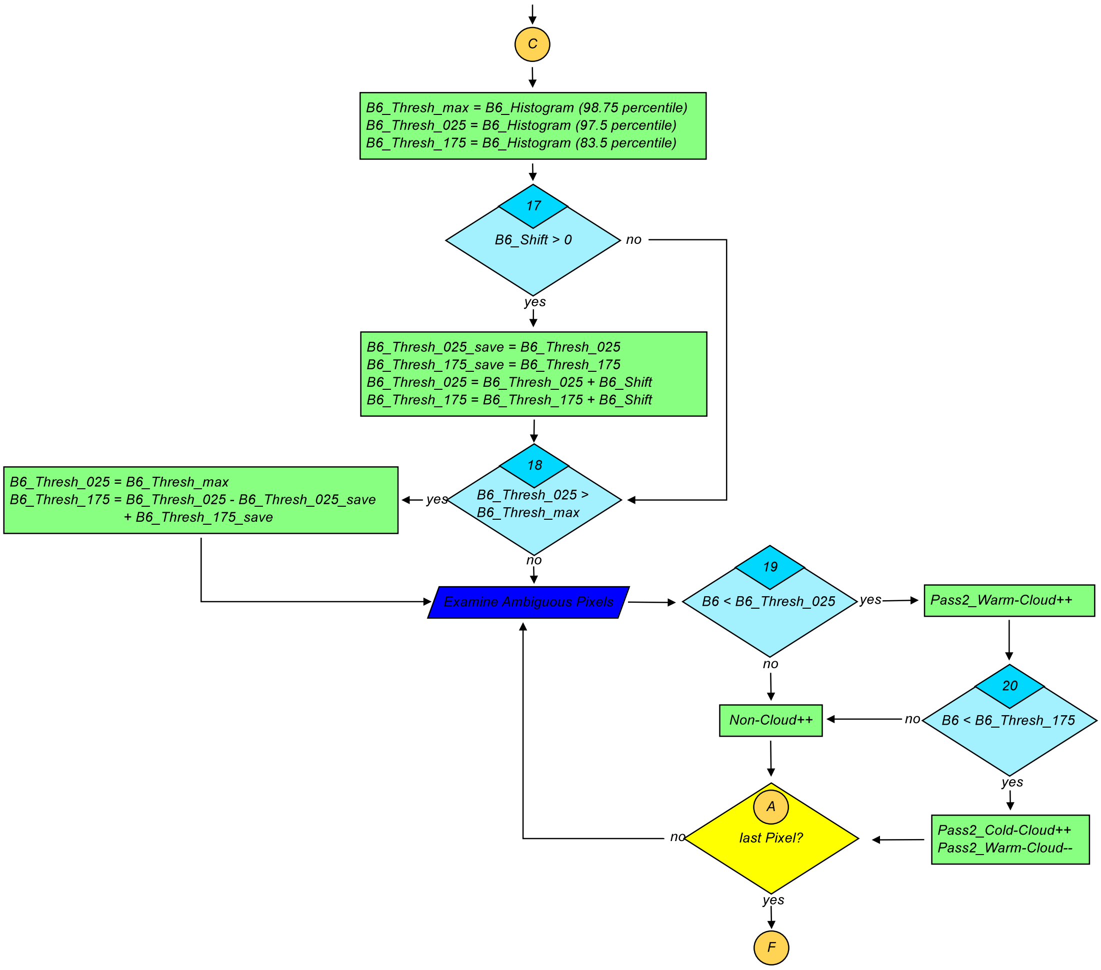


Figure A.4: Pass-2 Thermal Band Cloud Separation [25]

Each pixel labeled "Ambiguous" from Pass-1 is re-examined and tested against the upper threshold. If it exceeds this value it is marked as "Non-Cloud" (probably false positive), otherwise it is marked as "Warm-Cloud" and passed to filter 20.

- **Filter 20** Band 6 Cold-Cloud Indicator
  Each "Warm-Cloud"-pixel is again tested against the lower threshold. If it falls below this value it is labeled as "Cold-Cloud".

iv) Image-Based Cloud-Cover Assignments and Aggregation

After Pass-2 processing the percentages of "Cold-Cloud" as well as the combined

"Cold-Cloud" and "Warm-Cloud" classes from Pass-2 are calculated. The integrity of these two classes is then assessed. The presence of snow or desert features as well as the magnitude of the two Pass-2 classes are used to accept or reject one or both classes. Cloud classes that qualify as legitimate are combined with the Pass-1 clouds to form a single unified cloud mask.

- **Filter 21** Pass-2 Cloud-free Indicator
  If no clouds were found in Pass-2 which is very unlikely to happen, the scene's final cloud mask is the "Cold-Cloud" mask from pass-1 and processing resumes at filter 26. Otherwise proceed to filter 22.
- **Filter 22** Pass-1 Cloud Temperature Mean
  This filter is used when one of the conditions in filter 14 are nor met. Then the mean temperature of the pass-1 "Cold-Cloud" population is again tested against the limit of 295 K. If it is less the clouds are accepted as real, but the certainty is lower. If the mean temperature from Pass-1 falls below 295 K, the final cloud mask is the 'Cold-Cloud" mask from Pass-1. Otherwise the result is rejected and the scene is reported as cloud free.
- **Filter 23** Pass-1 Cloud Acceptance Indicator
  If snow or desert conditions exist (determined in pass-1) the pass-1 cloud coverage is set to the "Cold-Cloud" mask, otherwise the combined 'Cold-Cloud" and 'Warm-Cloud" mask is used.
- **Filter 24** Pass-2 Cold and Warm Cloud Acceptance
  The temperature means and maximums for both the "Cold-Cloud" mask and the combined "Cold-Cloud" and "Warm-Cloud" mask are calculated. In addition the percentage of these two classes with respect to the complete scene is computed. The pass-2 "Cold-Cloud" and "Warm-Cloud" masks are joined with the pass-1 mask if the following four conditions are met:
  - the pass-2 "Cold-Cloud" and "Warm-Cloud" contribution must not be higher than 35 %
  - no more than 1 % snow must exist
  - the mean temperature for the combined pass-2 "Cold-Cloud" and "Warm-Cloud" class must not be greater than 295 K
  - the difference between the combined pass-2 "Cold-Cloud" and "Warm-Cloud" class cloud maximum temperature and the upper threshold must

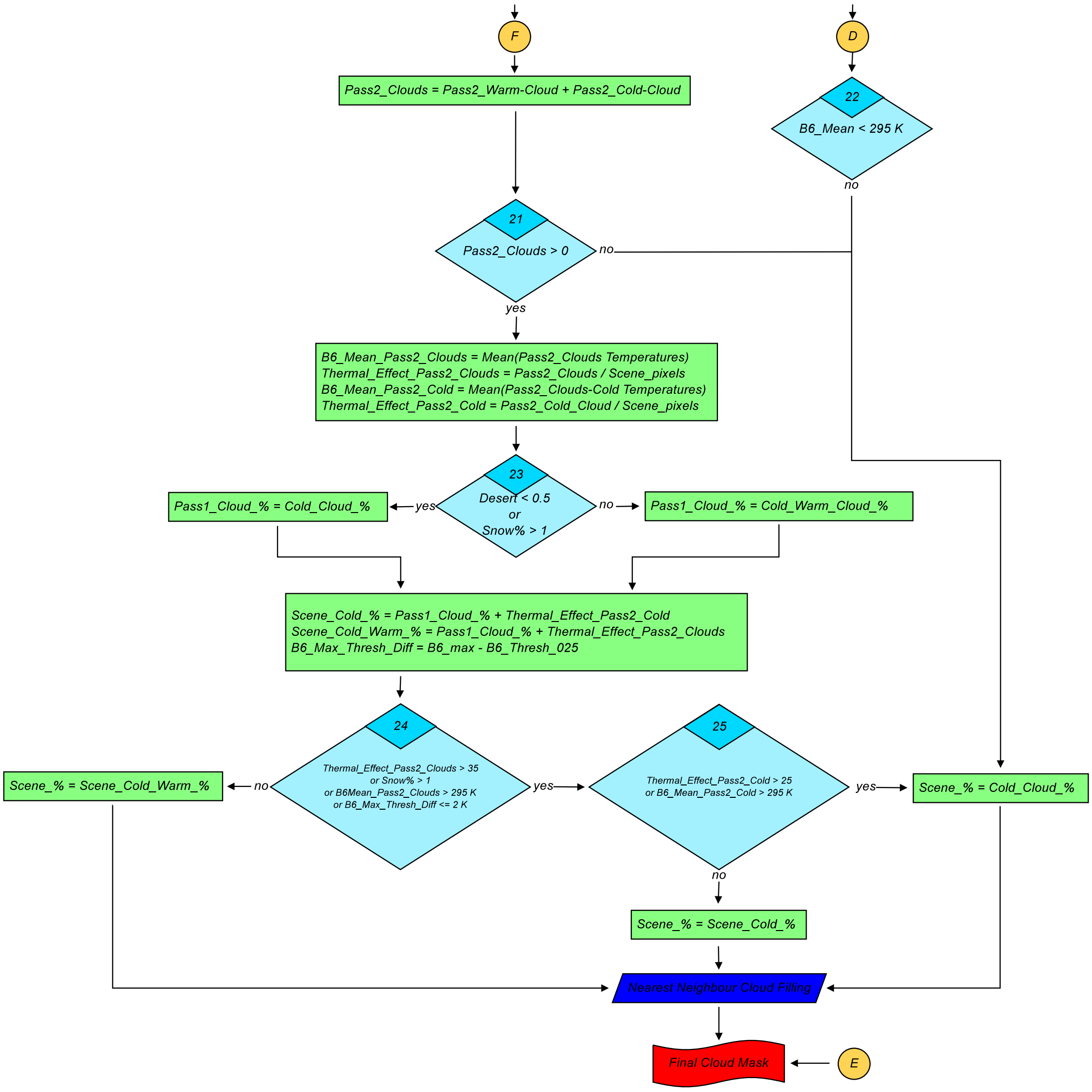


Figure A.5: Image-Based Cloud-Cover Assignments and Aggregation [25] not be less than 2 K.

If these four conditions are met the combined “Cold-Cloud” and “Warm-Cloud” mask of pass-2 is united with the pass-1 cloud mask and processing proceeds to filter 26. If only one condition is breached proceed to filter 25.

- **Filter 25** Pass-2 Cold Cloud Acceptance
  The Pass-2 “Cold clouds” are used if their contribution to the scene cloud percentage is less than 25 % and their mean temperature is lower than 295 K. If these two conditions are satisfied the “Cold-Cloud” of Pass-2 are joined with the Pass-1 clouds. If one of these condition is breached, only the Pass-1 cloud mask is used. Processing now continues at filter 26.
- **Filter 26** Nearest Neighbor Cloud-Filling
  Each “Non-Cloud” pixel is examined and converted to cloud if at least 5 of its 8 neighbors are clouds. Filled pixels qualify as cloudy neighbors in subsequent tests.

After this last filter we now have a final cloud mask created by the ACCA algorithm.

B Comparison of the cloud coverage results published by LPS and the ACCA implementation

As described in chapter 6.2 there are often discrepancies in the results of the cloud coverage published by the Landsat Processing System (LPS) and the cloud coverage obtained by the ACCA implementation used in this thesis. The following table B.1 describes the results of 42 scenes where the difference between the published results and the ones used in this thesis is larger than 10 %. It can be seen that the implementation of the ACCA algorithm often gives a better result.

SceneID	LPS [%]	ACCA [%]	Comment
LE71930242002168EDC00	54	19	LPS finds too much clouds, ACCA implementation correct
LE71230442000153SGS00	47	79	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE70250342000122EDC00	56	87	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE70970672000275ASA00	14	45	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE71460242000122SGS00	20	5	LPS correct, ACCA implementation finds too much clouds (mountains and snow)
LE71290182000275SGS00	48	18	LPS finds too much clouds, ACCA implementation correct despite of mountains and snow
LE71750732000005EDC00	49	79	LPS may fail in detection of cirrus clouds, ACCA implementation detects surface as clouds. Scene not considered in evaluation.
LE71320452000152SGS00	22	51	LPS correct, ACCA implementation finds too much clouds. Surface features are detected as clouds!
LE70250402000122EDC00	26	54	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE71760582000124EDC00	21	45	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct

table B.1 continued on next page

SceneID	LPS [%]	ACCA [%]	Comment
LE71720562000128SGS00	45	68	LPS finds too few clouds (fails in detection of cirrus clouds and cloud fringes), ACCA implementation correct
LE70250372000122EDC00	69	92	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE71720572000128SGS00	33	55	LPS may fail in detection of cirrus clouds, ACCA implementation detects surface as clouds. Scene not considered in evaluation.
LE70170412000130EDC00	11	32	LPS correct, ACCA implementation finds too much clouds (clouds over water)
LE70050702000126EDC00	31	52	LPS finds too few clouds, ACCA implementation correct
LE71780602000122EDC00	64	83	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE72000241999193EDC00	15	34	LPS finds too few clouds, ACCA implementation correct
LE70250412000122EDC00	24	42	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE72330672000123AGS01	43	60	LPS correct, ACCA implementation finds too much clouds. Surface features are detected as clouds!
LE72240622000124EDC00	68	85	LPS finds too few clouds (fails in detection of cirrus clouds and cloud fringes), ACCA implementation correct
LE70050672000126EDC00	13	30	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE72310772000029AGS00	34	50	LPS correct, ACCA implementation finds too much clouds (mountains and snow)
LE71300452000154SGS00	35	51	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE71770702000003EDC00	57	73	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE70670192000128AGS00	38	53	LPS finds too few clouds, ACCA implementation correct
LE71880542000128EDC00	62	77	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE71000692000008ASA00	32	46	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct. Still result is not correct because not all cirrus clouds are found
LE71160241999181EDC00	65	79	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE70240392002152EDC00	33	19	LPS finds too much clouds, ACCA implementation correct
LE71930222000275EDC00	15	28	LPS correct, ACCA implementation finds too much clouds. Surface features are detected as clouds!
LE70170402000130EDC00	14	16	LPS correct, ACCA implementation finds too much clouds (clouds over water)
LE70260442002230EDC01	27	15	LPS finds too much clouds, ACCA implementation correct
LE71020692002155ASA00	27	16	LPS finds too much clouds, ACCA implementation correct
LE71940272000122EDC00	35	46	LPS correct, ACCA implementation finds too much clouds (mountains and snow)

table B.1 continued on next page

SceneID	LPS [%]	ACCA [%]	Comment
LE71770692000003EDC00	58	69	LPS may fail in detection of cirrus clouds, ACCA implementation detects surface as clouds. Scene not considered in evaluation.
LE70280272000127EDC00	3	14	LPS correct, ACCA implementation finds too much clouds. Surface features are detected as clouds!
LE71860622002007SGS00	2	12	LPS finds too few clouds, ACCA implementation correct
LE70180311999182EDC00	55	65	LPS finds too few clouds (fails in detection of cirrus clouds and cloud fringes), ACCA implementation correct
LE71940292000122EDC00	5	15	LPS correct, ACCA implementation finds too much clouds. Surface features are detected as clouds!
LE72150722000125EDC00	17	27	LPS finds too few clouds (fails in detection of cloud fringes), ACCA implementation correct
LE70290352000006EDC00	25	35	LPS finds too few clouds (fails in detection of cirrus clouds), ACCA implementation correct
LE72280731999181EDC00	51	61	LPS finds too few clouds, ACCA implementation correct

Table B.1: In this table the ETM+ scenes are examined whose deviation of the published LPS cloud coverage and the cloud coverage obtained by the ACCA algorithm implemented and used in this thesis is larger than 10 %.

C Coefficients for the linear discriminant functions

C.1 Coefficients for LDA

ETM+

Coefficient	Value
C1	220.19154428063
C2	-110.427712794732
C3	-63.6496727282985
C4	0.465558912706626
C5	46.9474972589048
C6	-0.30451640667477
C7	-4.53607998675386
B	-59.9459810144309
mean accuracy [%]	91.5

Table C.1: This table shows the optimal coefficients for the LDA classifier for ETM+, where C1 to C7 denote the coefficients for the corresponding spectral bands.

MODIS

Coefficient	Value
C1	-146.739879681703
C2	-161.570680156937
C3	230.419041874815
C4	28.1353532204445
C5	145.519881055423
C6	-257.397407328051
C7	237.091208466567
C8	434.619194204775
C9	-511.853614240692
C10	-178.676280093427
C11	66.5018110237064
C12	-2.95191786755024
C13	7.3353098299621
C14	7.71988796867909
C15	-31.6030907640702
C16	32.571436751648
C17	-7.33712015669013
C18	-17.6770454189202
C19	-0.9756432785123
B	-39.5381765120033
mean accuracy [%]	86.6

Table C.2: This table shows the optimal coefficients for the LDA classifier for MODIS, where C1 to C19 denote the coefficients in descending order for the spectral bands used in the MODIS cloud mask algorithm.

C.2 Coefficients for SVM

ETM+

Coefficient	Value
C1	50.8178504130254
C2	-13.1255828974797
C3	-28.9339956013428
C4	3.87383330804843
C5	11.6328307476228
C6	-0.0187157965190995
C7	3.32696750857756
B	1.9267644607381
mean accuracy 10-fold cross [%]	98.3
mean accuracy [%]	91.2
no SVs	164

Table C.3: This table shows the optimal coefficients for the SVM classifier for ETM+, where C1 to C7 denote the coefficients for the corresponding spectral bands. The coefficients were calculated with 1000 training points and a cost factor $C = 1$.

MODIS

Coefficient	Value
C1	1.92414704977761
C2	0.825742697278332
C3	1.93256610604374
C4	1.30139473655922
C5	1.31558099662229
C6	1.0831042345003
C7	1.04734128594088
C8	1.64110407823124
C9	1.44143213636033
C10	1.70215347907171
C11	-0.501810598239421
C12	-0.500795366462996
C13	0.119108699462645
C14	-0.0762050508986102
C15	-0.106559544441645
C16	-0.113667471821842
C17	-0.134524001192464
C18	-0.160203388104743
C19	-0.014687539099841
B	-1.39646232347395
mean accuracy 10-fold cross [%]	100.0
mean accuracy [%]	78.2
no SVs	100

Table C.4: This table shows the optimal coefficients for the SVM classifier for MODIS, where C1 to C19 denote the coefficients in descending order for the spectral bands used in the MODIS cloud mask algorithm. The coefficients were calculated with 1000 training points and a cost factor of $C = 0.01$.

D Procedure for the usage of the R package “e1071”

After starting R, we have the input prompt denoted by “>”. So first load the “e1071”[38] package by

```
>library(e1071)
```

Assume our training data set called “dataset.csv” has in the first column the labels and in the following (e.g. for ETM+) n columns (features) the corresponding values of m training points (rows). Then we load the data set by

```
>dataset <- read.csv(file="dataset.csv", head=F, sep=",")
```

and prepare it for further processing:

```
>labels <- data[,1]
>labels <- factor(labels)
>features <- data[,2:8]
```

Now lets train the classifier:

```
>model <- svm(training, labels, kernel=“linear”)
```

The libsvm library[11] first scales the input data with mean 0 and variance 1. The scaling factors can be accessed by

```
>model$x.scale
```

So the scaled training points are calculated via

$$x'_{ij}(x_{ij}) = \frac{x_{ij} - c_j}{s_j}, \tag{D.1}$$

where $i = 1, ..., m$, $j = 1, ..., n$, c_j is the mean value of feature j and c_j is the variance of feature j. The reverse projection is obviously given by

$$x_{ij}(x'_{ij}) = x'_{ij} \cdot s_j + c_j. \tag{D.2}$$

Now we observe the result of the SVM calculation:

>model$SV
These are the resulting scaled k support vectors. They can also be found in the training data with the indices given by
>model$index
and the number of support vectors is
>model$nSV.

Remember now the dual form of the decision function:

$$f(\mathbf{x}') = \sum_{i \in SV} y_i \alpha'_i \langle \mathbf{x}'_i \cdot \mathbf{x}' \rangle + b' \tag{D.3}$$

$-b'$ is found by
>model$rho.
The product $y_i \alpha'_i$ can be accessed by $(i = 1, ..., k)$
>model$coefs[i]
Finally the result $f(\mathbf{x}')$ of each decision is stored in
>model$decision.values

Having now the understanding of the “e1071” output, we want to find the unscaled coefficients $\mathbf{w}$ and b for the primal decision function

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \tag{D.4}$$

We know that

$$\mathbf{w}' = \sum_{i \in SV} y_i \alpha'_i \mathbf{x}'_i \tag{D.5}$$

and this can be calculated immediately.
First we transform the scaled components of $\mathbf{w}'$ to the unscaled components of $\mathbf{w}$. The transformation formula for a *normal* vector is given by

$$w_j = \sum_{k=1}^{n} \frac{\partial x'_k}{\partial x_j} w'_j, \tag{D.6}$$

where n is the number of features. This means that the transformation of the components of $\mathbf{w}$ is given by

$$w_j = \frac{1}{s_j} w'_j, \tag{D.7}$$

where s_j is the scaling factor of feature j.

Now we have to transform b' to b. We start by normalizing $\mathbf{w}'$:

$$\mathbf{w}'_0 = \frac{\mathbf{w}'}{||\mathbf{w}'||}. \tag{D.8}$$

The point on the hyperplane next to the origin $\mathbf{x}'_0$ is given by

$$\mathbf{x}'_0 = b' \cdot \mathbf{w}'_0. \tag{D.9}$$

Combining equations (D.2) and (D.9), we get the reverse projection for x_{0_j} as

$$x_{0_j} = b' \cdot s_j \cdot w'_{0_j} + c_j. \tag{D.10}$$

Since the hyperplane is described by $\langle \mathbf{w} \cdot \mathbf{x} \rangle = b$, it follows that

$$b = \sum_{j=1}^{n} w_j \cdot x_{0_j}. \tag{D.11}$$

So using equations (D.7) and (D.11), we can calculate the unscaled primal form of the decision problem with the scaled output of “e1071”.

E Scene Identifiers for the scenes used in this thesis

E.1 ETM+

Scene Identifier
LE70010612002007EDC00
LE70010662002231EDC00
LE70010672002151EDC00
LE70010672002231EDC00
LE70010722002151COA00
LE70030532002309EDC00
LE70030552002309EDC00
LE70030602002005EDC00
LE70030612002005EDC00
LE70050472000126AGS00
LE70050672000126EDC00
LE70050702000126EDC00
LE70070472002305EDC00
LE70080522002152EDC00
LE70080572002008EDC00
LE70080642000275EDC00
LE70080652000275EDC00
LE70080662000275EDC00
LE70080672000275EDC00
LE70130312002155EDC00
LE70130322002155EDC00
LE70140302002130EDC00
LE71080802002005ASA00
LE71080812002005ASA00
LE71080822002005ASA00
LE71090742000007ASA00
LE71100662002307ASA00
LE71100712002307ASA00
LE71100732002307ASA00
LE71120522000124EDC00
LE71120542000124EDC00
LE71120832002305SGS00
LE71130662000275ASA00
LE71140542000122EDC00
LE71160241999181EDC00
LE71160482000152SGS00
LE71170502002020EDC00
LE71170532002020DKI00
LE71170542002020DKI00
LE71170572002308SGS02
LE71190522002002EDC01
LE71190582002002EDC01
LE71190582002306SGS00
LE71190602002002EDC01
LE71770702000003EDC00
LE71780512000026EDC00
LE71780522000122EDC00
LE71780532000122EDC00
LE71780602000122EDC00
LE71780672000010EDC00
LE71780762000010EDC00
LE71790452000129EDC00
LE71790492002006SGS01
LE71790562002310SGS00
LE71790592002310SGS01
LE71790602002310SGS00
LE71800202000280SGS00
LE71800212000280SGS00
LE71830341999186EDC00
LE71830431999186EDC00
LE71830602002002SGS00
LE71830642002002SGS00
LE71830652002002SGS00
LE71840282002121SGS00
LE71840292002121SGS00
LE71840302002121SGS00

table E.1 continued on next page

Scene Identifier		
LE70140352002130EDC00	LE71190602002306SGS00	LE71840552000276EDC00
LE70150292002121EDC00	LE71200592002153SGS00	LE71840582002233SGS00
LE70150312002153EDC00	LE71210492002016SGS00	LE71850232002240SGS00
LE70150322002121EDC00	LE71230442000153SGS00	LE71850262002240SGS00
LE70150332002121EDC00	LE71250462000151SGS00	LE71850302002320SGS00
LE70150362002121EDC00	LE71260411999187EDC01	LE71850572000123EDC00
LE70160442000123EDC00	LE71260562002003SGS00	LE71860192002231SGS01
LE70160452000123EDC00	LE71260632002227SGS00	LE71860252002231SGS01
LE70170352000130EDC00	LE71270562002154SGS00	LE71860622002007SGS00
LE70170402000130EDC00	LE71280472002017SGS00	LE71880472002309EDC00
LE70170412000130EDC00	LE71280482002305SGS00	LE71880532000128EDC00
LE70170472000034EDC00	LE71290182000275SGS00	LE71880542000128EDC00
LE70170492000002EDC00	LE71290482002152SGS00	LE71890342002028SGS00
LE70180311999182EDC00	LE71300452000154SGS00	LE71890352002012SGS00
LE70180361999182EDC00	LE71300492000122SGS00	LE71890432000007EDC00
LE70210302002307EDC00	LE71300502000122SGS00	LE71900272002163EDC00
LE70210312002227EDC00	LE71310361999190SGS01	LE71900312002227EDC00
LE70210392002227EDC00	LE71320362002013SGS00	LE71910262002122EDC00
LE70230312002305EDC00	LE71320452000152SGS00	LE71910442000021EDC00
LE70230342002305EDC00	LE71330342002004SGS00	LE71910512000085EDC00
LE70230382002305EDC00	LE71330391999188SGS00	LE71910522000085EDC00
LE70240292000275EDC00	LE71330401999188SGS00	LE71920292002305EDC00
LE70240312000275EDC00	LE71340272000134SGS00	LE71920472002305EDC00
LE70240322002152EDC00	LE71340422002155SGS00	LE71930182000275SGS00
LE70240372002152EDC00	LE71360162002153SGS00	LE71930192002152EDC00
LE70240392002152EDC00	LE71360192002169SGS00	LE71930222000275EDC00
LE70250242000122PAC00	LE71360222002153SGS00	LE71930242002152EDC00
LE70250272000122EDC00	LE71360222002169SGS00	LE71930242002168EDC00
LE70250292000122EDC00	LE71360272000276SGS00	LE71930312002168EDC00
LE70250342000122EDC00	LE71360292000276SGS01	LE71930322002312EDC00
LE70250372000122EDC00	LE71370332002016SGS00	LE71940182002239SGS00
LE70250402000122EDC00	LE71380232002231SGS00	LE71940272000122EDC00
LE70250412000122EDC00	LE71400462002021SGS00	LE71940292000122EDC00

table E.1 continued on next page

Scene Identifier
LE70260172002230PAC00
LE70260352000001EDC00
LE70260362000001EDC00
LE70260382000001EDC00
LE70260402002230EDC00
LE70260422000001EDC00
LE70260442002230EDC01
LE70260452000001EDC01
LE70260462002230EDC00
LE70260472000033EDC00
LE70260482000033EDC00
LE70280212000127PAC00
LE70280272000127EDC00
LE70280302000127EDC00
LE70280332002308EDC00
LE70280452000127EDC01
LE70290352000006EDC00
LE70290382000006EDC00
LE70300282002130EDC00
LE70310312002153EDC00
LE70330352000130EDC00
LE70370152002227EDC00
LE70370332002307EDC00
LE70380212002154EDC00
LE70380222002154EDC00
LE70380232002154EDC00
LE70380282002154EDC00
LE70400272000275EDC00
LE70400302000275EDC00
LE70400372002152EDC00
LE70440192002228EDC00
LE70440202002228EDC00
LE70440342002308EDC00
LE71400482002309SGS00
LE71410162000135SGS00
LE71410182000135SGS00
LE71440482002017SGS00
LE71440492002017SGS00
LE71450382000275SGS00
LE71460232000122SGS00
LE71460242000122SGS00
LE71460272000122SGS00
LE71460282000122SGS00
LE71460312000122SGS00
LE71480392002013SGS00
LE71500162000134SGS00
LE71510362002002SGS02
LE71510372002002SGS02
LE71580212002227SGS00
LE71580692000126SGS00
LE71580712002307SGS00
LE71580732002307SGS00
LE71600422000284SGS00
LE71610552000131SGS00
LE71610562000131SGS00
LE71620402000282SGS00
LE71630532002006SGS00
LE71630552002006SGS00
LE71630552002310SGS00
LE71630562002006SGS00
LE71670462002002SGS00
LE71670771999186EDC01
LE71680202002169SGS01
LE71680662000276SGS00
LE71680672000276SGS00
LE71680682000276SGS00
LE71960261999181EDC00
LE71980222000134EDC00
LE71980242000134EDC00
LE71980312000134EDC00
LE71980332000134EDC00
LE71990162002162SGS00
LE71990532002002EDC02
LE72000241999193EDC00
LE72000242002121SGS00
LE72000322002025EDC00
LE72060202002227EDC00
LE72060212002227EDC00
LE72060232002227EDC00
LE72070241999194EDC01
LE72090232002168EDC00
LE72150672000125EDC00
LE72150722000125EDC00
LE72170732000123EDC00
LE72170742000123EDC00
LE72190621999182EDC00
LE72200702002005EDC00
LE72200712002005EDC01
LE72200722002005EDC00
LE72220682002307EDC00
LE72220702002307EDC00
LE72220732000126EDC00
LE72220782002227COA00
LE72220792002227COA00
LE72220812002227EDC00
LE72220862002227COA00
LE72240622000124EDC00
LE72240772000028AGS00
LE72240802000028AGS00

table E.1 continued on next page

Scene Identifier		
LE70670192000128AGS00	LE71680722002233EDC00	LE72240862002305COA00
LE70760102000127AGS00	LE71690162002240SGS00	LE72250712000275AGS00
LE70890792000122ASA00	LE71690192002240SGS00	LE72250782002152COA00
LE70890812000122ASA00	LE71690242000283SGS00	LE72250792002152EDC00
LE70890822000122ASA00	LE71690692000123SGS00	LE72250842002008COA00
LE70890832000122ASA00	LE71700132002231SGS00	LE72260752000122EDC00
LE70900772002230ASA00	LE71700482002007SGS00	LE72260812000122EDC00
LE70920722000127EDC00	LE71710222000281SGS00	LE72260822000122EDC00
LE70930712002155ASA00	LE71710801999182EDC00	LE72260832000122EDC00
LE70930822002155ASA00	LE71720492000128SGS00	LE72260872000122EDC00
LE70940722002002EDC00	LE71720562000128SGS00	LE72260972000282EDC00
LE70940742002002EDC00	LE71720572000128SGS00	LE72270822000001EDC00
LE70940842002227ASA00	LE71720762002309SGS00	LE72270852000001EDC00
LE70960862002305EDC01	LE71720772002309SGS00	LE72280722002157EDC00
LE70970662000275ASA00	LE71730482002012SGS00	LE72280731999181EDC00
LE70970672000275ASA00	LE71730522002012SGS00	LE72280732002157EDC00
LE70970692000275ASA00	LE71730542002012SGS00	LE72280751999181EDC00
LE70970832002152ASA00	LE71740162002227SGS00	LE72280922000280EDC00
LE70990632002230EDC00	LE71740192002227SGS00	LE72290672002308PFS03
LE70990662002230EDC00	LE71740252002163SGS00	LE72290692002308PFS03
LE70990692002230ASA00	LE71740562002307SGS00	LE72290702002308PFS03
LE70990712000001ASA00	LE71740572002307SGS00	LE72290762002308COA00
LE70990732000001ASA00	LE71740632000126EDC00	LE72300922000278AGS00
LE70990742000001ASA00	LE71740712000030AGS00	LE72300982000278AGS00
LE70990762000001ASA00	LE71740742000030AGS00	LE72310662002146AGS00
LE71000692000008ASA00	LE71740782000126EDC00	LE72310702002306EDC00
LE71010642002308ASA00	LE71750732000005EDC00	LE72310712002146EDC00
LE71010702000127ASA02	LE71760522002305SGS00	LE72310722002306EDC00
LE71010712000127ASA02	LE71760582000124EDC00	LE72310772000029AGS00
LE71010762000127ASA02	LE71760652002305EDC00	LE72310772002306EDC00
LE71010772000127ASA02	LE71770212000275EDC00	LE72310782002002AGS00
LE71020692002155ASA00	LE71770222000275EDC00	LE72310812002002AGS00
LE71030732000125ASA00	LE71770232000275EDC00	LE72310842002002AGS00

table E.1 continued on next page

Scene Identifier		
LE71030752002002EDC01	LE71770242002152EDC00	LE72310852002002AGS00
LE71030762000125ASA00	LE71770272000275EDC00	LE72320682002121EDC00
LE71030792002002EDC01	LE71770522002152EDC00	LE72320732002121COA00
LE71040802002153ASA00	LE71770532002152EDC00	LE72320742002121COA00
LE71050612000123ASA00	LE71770562000131EDC00	LE72320772000004EDC00
LE71050782000123ASA00	LE71770562002152EDC00	LE72330612000123AGS00
LE71080762002005ASA00	LE71770622000275EDC00	LE72330672000123AGS01
LE71080782002005ASA00	LE71770692000003EDC00	LE72330722000123AGS00

Table E.1: This table lists all ETM+ scenes examined in this thesis.

E.2 MODIS

Scene Identifier
MOD021KM.A2001185.1000.005.2010065130927
MOD021KM.A2001198.0955.005.2010067194522
MOD021KM.A2001352.0030.005.2011012051338
MOD021KM.A2003143.0830.005.2010148222916
MOD021KM.A2005082.0035.005.2010150071855
MOD021KM.A2005082.1325.005.2010150080107
MOD021KM.A2005082.1640.005.2010150083806
MOD021KM.A2005122.1100.005.2010152033638
MOD021KM.A2005122.1230.005.2010152032932
MOD021KM.A2005143.0300.005.2010152215803
MOD021KM.A2005143.0315.005.2010152220511
MOD021KM.A2005143.0800.005.2010152220135
MOD021KM.A2005143.0815.005.2010152220114
MOD021KM.A2005144.0655.005.2010152222903
MOD021KM.A2005174.0535.005.2010154014030
MOD021KM.A2005174.1040.005.2010154014403
MOD021KM.A2005174.1405.005.2010154020211
MOD021KM.A2005275.1045.005.2010161184932
MOD021KM.A2005275.1720.005.2010161190853
MOD021KM.A2005296.1925.005.2010162111216
MOD021KM.A2005297.0650.005.2010162114439
MOD021KM.A2005357.0225.005.2010166230253
MOD021KM.A2005357.0540.005.2010166204718
MOD021KM.A2005357.0720.005.2010166204353
MOD021KM.A2005357.1230.005.2010166202946
MOD021KM.A2005357.1410.005.2010166202112
MOD021KM.A2005357.1715.005.2010166202135
MOD021KM.A2005358.1620.005.2010166205648
MOD021KM.A2006327.0950.005.2010185202241

Table E.2: This table lists all MODIS scenes examined in this thesis.