ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Νικολάου Τσικνάκη του Εμμανουήλ

Αριθμός Μητρώου: 1020623

Θέμα

Stereo Vision of Dual View MSG Satellite Images

Επιβλέπων

Καθηγητής Αθανάσιος Σκόδρας, Πανεπιστήμιο Πατρών

Αριθμός Διπλωματικής Εργασίας:

Πάτρα, Ιούνιος 2019

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

Stereo Vision of Dual View MSG Satellite Images

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Νικολάου Τσικνάκη του Εμμανουήλ

(Α.Μ.: 1020623)

παρουσιάστηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

__/__/____

Ο Επιβλέπων

Αθανάσιος Σκόδρας
Καθηγητής

Ο Διευθυντής του Τομέα Σ&ΑΕ

Δημοσθένης Καζάκος
Επίκουρος Καθηγητής

Στοιχεία διπλωματικής εργασίας

Θέμα: **Stereo Vision of Dual View MSG Satellite Images**

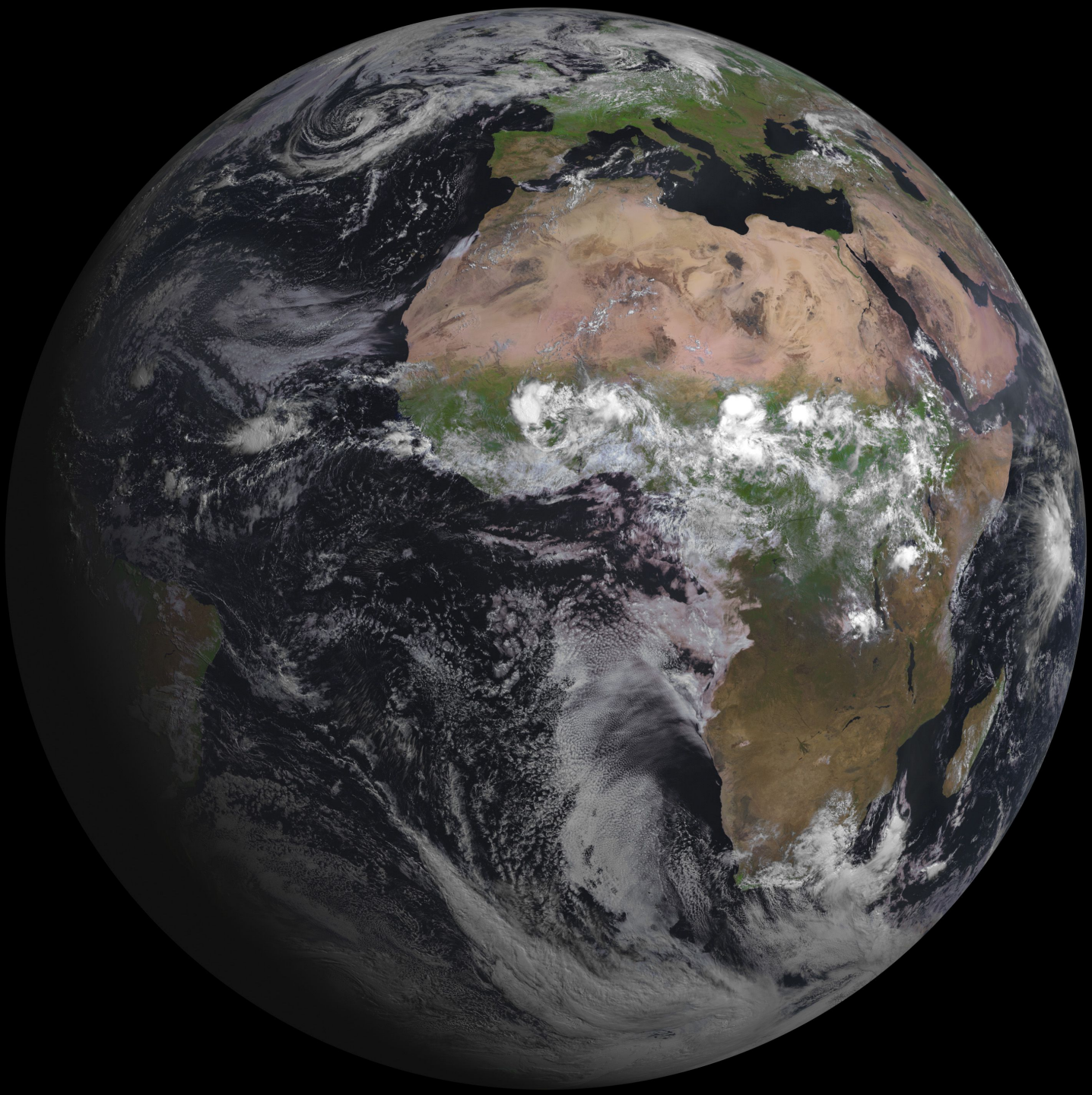
Φοιτητής: **Νικόλαος Τσικνάκης του Εμμανουήλ**

Ομάδα επίβλεψης
Καθηγητής Αθανάσιος Σκόδρας, Πανεπιστήμιο Πατρών

Professor Adrian Munteanu, Vrije Universiteit Brussel

Εργαστήριο
Ψηφιακής Επεξεργασίας Σημάτων και Εικόνων

Περίοδος εκπόνησης της εργασίας:
Απρίλιος 2018 - Ιούνιος 2019



MSG-3 SEVIRI First Image

7 August 2012 09:45 UTC

Full Disk Image - RGB (1.6-0.8-0.6)

Meteosat-10

EUMETSAT

Περίληψη

Η στερεοσκοπική όραση είναι ένα πεδίο έρευνας στον ευρύτερο τομέα της υπολογιστικής όρασης, με βάση το οποίο προσπαθεί κανείς να εξάγει τρισδιάστατη (3Δ) πληροφορία μιας σκηνής ή ενός αντικειμένου χρησιμοποιώντας δύο όψεις αυτών, κάθε μία από τις οποίες έχει ληφθεί υπό διαφορετική οπτική γωνία. Διάφορες βιομηχανικές και ακαδημαϊκές ερευνητικές εφαρμογές χρησιμοποιούν τεχνικές στερεοσκοπικής όρασης, όπως το σύστημα πλοήγησης των αυτόνομων οχημάτων, η 3Δ ανακατασκευή ιστορικών χώρων και άλλα.

Σε αυτή τη διπλωματική εργασία, χρησιμοποιούμε την δυνατότητα που μας προσφέρουν οι δορυφόροι Meteosat Second Generation (MSG3 και MSG1 οι οποίοι είναι γεωστατικοί δορυφόροι) για να παρατηρήσουμε την ίδια περιοχή στην Ευρώπη από δύο διαφορετικά γεωγραφικά μήκη, 0° και 41.5° πάνω από τον Ισημερινό αντίστοιχα για κάθε δορυφόρο. Ο στόχος της διπλωματικής εργασίας αφορά στην εξαγωγή 3Δ πληροφορίας σχετικά με το ύψος των νέφων πάνω από το Βέλγιο, εφαρμόζοντας τεχνικές στερεοσκοπικής όρασης. Η πληροφορία αυτή μπορεί να αποδειχθεί πολύ χρήσιμη για την μετεωρολογική ερευνητική κοινότητα.

Τα δεδομένα που μας δόθηκαν είχαν ήδη υποβληθεί σε κάποια προεπεξεργασία και ειδικότερα οι εικόνες που αφορούν στον δορυφόρο MSG1 διορθώθηκαν προκειμένου να ταιριάξουν με την προοπτική των αντίστοιχων εικόνων του δορυφόρου MSG3. Ωστόσο, επειδή παρατηρείται μια σχετική μετατόπιση μεταξύ των εικόνων του MSG3 και MSG1 σε σχέση με το γεωγραφικό μήκος και πλάτος, περικόπτουμε τις εικόνες γύρω από μια συγκεκριμένη περιοχή ενδιαφέροντος, πάνω από το Βέλγιο, προκειμένου να ελαχιστοποιήσουμε την επίδραση αυτής της σχετικής μετατόπισης. Στη συνέχεια, για να βελτιώσουμε την διόρθωση προσπαθούμε να ευθυγραμμίσουμε δύο εικόνες καθαρού ουρανού και έπειτα εφαρμόζουμε τον υπολογισμένο μετασχηματισμό σε κάθε ζεύγος περικομμένων εικόνων. Συγκεκριμένα για την ευθυγράμμιση, συγκρίνουμε δύο μεθόδους. Η πρώτη βασίζεται στη μεγιστοποίηση της αμοιβαίας πληροφορίας (Mutual Information) του ζεύγους εικόνων καθαρού ουρανού με χρήση του Gradient Descent βελτιστοποιητή και με Powerll βελτιστοποιητή. Η δεύτερη μέθοδος ευθυγράμμισης βασίζεται στα χαρακτηριστικά των εικόνων. Ειδικότερα για την εξαγωγή και την περιγραφή αυτών των χαρακτηριστικών συγκρίνουμε τους αλγορίθμους SIFT/SURF και κάνουμε χρήση του αλγορίθμου RANSAC για να υπολογίσουμε το μετασχηματισμό που απαιτείται για την ευθυγράμμισή των εικόνων.

Στη συνέχεια, προκειμένου να υπολογιστεί ο χάρτης βάθους και το ύψος των νέφων, συγκρίνουμε δύο μεθόδους στερεοσκοπικής αντιστοίχισης με βάση τα Markov Random Fields. Συγκεκριμένα, προσπαθούμε να ελαχιστοποιήσουμε μια συνάρτηση ενέργειας χρησιμοποιώντας δύο αλγορίθμους βασισμένους σε graph cuts, και συγκεκριμένα τους α-Expansion Move και $\alpha - \beta$ Swap Move αλγορίθμους. Για να επιβεβαιώσουμε την αποτελεσματικότητα της προτεινόμενης λύσης, συγκρίνουμε τα αποτελέσματα με τα αντίστοιχα δεδομένα LIDAR που μας δόθηκαν.

Abstract

Stereo vision is a field of computer vision research area, which focuses on extracting 3D information of a scene or an object in a scene using two views of that scene or object, each taken from a different perspective or angle of view. Several industrial and academic research applications use some stereo vision techniques, such as autonomous vehicles' navigation system, 3D reconstruction of historical sites and many more.

In this thesis, we utilize the Meteosat Second Generation Satellites' capability (MSG3 and MSG1 which are geostationary satellites) to observe the same region over Europe from two different longitudes, specifically 0° and 41.5° over the equator respectively for each satellite. The ultimate objective is to extract 3D information about the clouds' height over Belgium, by applying stereo vision techniques to the the retrieved images, which can be proved to be very useful for the meteorological research community.

The data which the current thesis will rely on were provided by KMI (Royal Meteorological Institute of Belgium) and had already undergone some preprocessing. In particular the MSG1 images were rectified in order for them to match the perspective of the corresponding MSG3 images. However, because a relative displacement between the MSG3 and MSG1 images depending on the longitude and latitude was observed, we crop the images around a specific region of interest (Belgium) in order to minimize the effect of that relative displacement. Then, in order to refine the rectification we register two clear-sky images and apply the computed transformation to each cropped image pair. We compare two registration methods, one based on maximizing the Mutual Information of the clear-sky image pair using either Gradient Descent or Powerll Optimizers. The second registration method is based on the low level features of the images, and in particular we compare the SIFT/SURF algorithms to extract and describe the features and RANSAC algorithm to estimate the transformation needed in order for them to be registered.

Finally, in order to compute the disparity map and estimate the clouds' height two stereo matching methods based on Markov Random Fields are compared. In particular, we attempt to minimize an energy function using two graph-cut based algorithms, α-Expansion Move and $\alpha - \beta$ Swap Move. To validate the proposed solution, the results are compared with the corresponding LIDAR data which was made available to us.

Acknowledgements

First of all, I would like to thank my supervisor at the University of Patras Prof. Athanasios Skodras, as well as my supervisors Prof. Adrian Munteanu and Dr. Bruno Cornelis at the Vrije Universiteit Brussel for the opportunity to actively work on such a difficult problem of computer vision and for the help they provided.

I would like to thank Prof. Kostas Marias who has significantly helped me during my internship at CBML FORTH and has assisted me a lot in the first stages of the thesis.

In addition, I want to thank Kostantinos Tsigganos for his help during the preprocessing stage of my thesis.

The most deep and true gratitude for my father Manolis, mother Maria and sister Stefi who have been supportive and caring regarding every decision I have made. I want to also thank my father for actively supporting me during my thesis.

Finally, I want to thank my friends with whom I have spent the most wonderful 5 years of my life and also have been a great pillar of support for me during not only the good but mostly the bad times.

Contents

List of Figures

List of Tables

Glossary

Disparity Map	The apparent pixel difference or motion between a pair of stereo images.	18
Spatial Resolution	The many meters or kilometers correspond to one image pixel.	6
Stereopsis	The perception of depth produced by the reception in the brain of visual stimuli from both eyes in combination; binocular vision.	11

Acronyms

CTH	Cloud Top Height	1
DoG	Difference of Gaussians	35
DSI	Disparity Space Image	23
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites	5
GPS	Global Positioning System	7
HRV	High-Resolution Visible	6
IMU	Inertial Measurement Unit	7
IR	InfraRed	6
KMI	Royal Meteorological Institute of Belgium	8
LIDAR	Light Detection and Ranging	7
LoG	Laplacian of Gaussians	35
MAP	Maximum a Posteriori	49
MI	Mutual Information	xix, 44–46
MRF	Markov Random Field	47
MSE	Mean Squared Error	23
MSG	Meteosat Second Generation	2, 5
MTP	Meteosat Transition Programme	5
NCC	Normalized Cross Correlation	23
PGM	Protable Gray Map	8
PNG	Portable Network Graphics	9
RANSAC	Random Sample Consensus	xvi, 36
SAD	Sum of Absolute Differences	23
SEVIRI	Spinning Enhanced Visible and Infrared Imager	5
SSD	Sum of Squared Differences	23
VNIR	Visible and Near InfraRed	6

WTA	Winner Takes All	22, 23

1. Introduction

One of the most interesting features of Earth, as seen from space, is the ever-changing distribution of clouds. As they float above us, we hardly give their presence a second thought. And yet, clouds have an enormous influence on Earth's energy balance, climate and weather in many and different ways depending on their characteristics and height. Clouds can absorb a significant amount of solar infrared radiation, blocking it from reaching Earth's surface, as well as the thermal infrared radiation that the Earth's surface emits back to space. Therefore they are a key regulator of the Earth's temperature. Clouds are also responsible for spreading solar energy evenly over Earth's surface [1]. Moreover, they are required for precipitation to occur and, hence are an essential part of the hydrologic cycle. These are some reasons why monitoring and understanding their behaviour and characteristics is crucial to many researchers.

People have been observing and attempting to extract parameters of interest of clouds for years. Among those parameters, estimation of Cloud Top Height (CTH) has been and continues to be an important area of interest. CTH studies provide information on cloud vertical structure leading to the understanding of the cloud radiative effects [2], on weather prediction as well as on 3D movement of severe weather phenomena. A variety of different methods have been employed by the scientific community in an attempt to accurately estimate CTH. A detailed analysis of those methods and an evaluation of their corresponding strengths and weaknesses is given in [3] and are shown in table 1.1

Out of all these methods, Stereoscopy represents an important family of methods. Stereographic observations of the atmosphere, primarily photogrammetry of clouds from twin ground-based cameras or from time sequences of aerial photographs, have a long history. The first reference to aerial stereography in the literature is by Schereschewsky (1921) who used a side-looking camera, as reported in [4]. Hasler [4] also reports that stereographic observations most similar to those from satellites have been made from high-flying aircraft with a vertically-looking camera. A specific example of stereo photogrammetric analysis of this kind was made of severe thunderstorms by Roach (1967) using U-2 aircraft flying at an altitude of about 20 km. Stereographic cloud observations from satellites were made by Ondrejka and Conover (1966) and Kikuchi and Kasai (1968) using early meteorological satellites equipped with vidicon tube cameras. In 1968, high resolution stereo cloud photographs were taken from Apollo 6 in low earth orbit and were used in [5] to demonstrate the feasibility of making stereographic cloud height measurements.

1.1 Scope and Objective of Current Thesis

The capability to make observations of the clouds and their changes with time by employing geosynchronous satellite stereography systems and advanced image processing and analysis techniques would provide novel tools in the hands of the related meteorological scientific community. Such satellite observations have wider spatial coverage than aerial, ground-based stereography or even low-orbit satellite platforms and are reported to exhibit better temporal resolution that is possible from proposed low-orbiting stereo satellites [4].

This thesis was proposed by the Vrije Universiteit Brussels in close collaboration with the Royal Meteorological Institute of Belgium and it was persued at the University of Patras. The central aim

Table 1.1: Overview of satellite methods for cloud top height retrieval. (Table taken from [3])

Methodology	**Pros/Cons**
LiDAR & radar	+ very high vertical resolution and accuracy – excessive revisit time (16 days) and only nadir observations from currently-operational instruments (LiDAR CALIOP, radar CPR)
Radio occultation	+ high resolution in lower troposphere – globally available only about 2000 times per day
Backward trajectory modeling	+ estimate possible even for clouds drifted away from the source – requires wind field data for a large area and a reliable trajectory model (e.g., turbulence not easy to handle); homogenous wind field results with high uncertainty of the source height
Brightness temperature	+ easy to apply, possible with instruments with a short revisit time – requires atmospheric profile and emissivity of the cloud; assumption of thermal equilibrium;
O2 A-band absorption	+ high accuracy – requires high spectral resolution data (not available on many satellites, long revisit time); good performance only over dark surfaces; requires radiative transfer modeling; daytime only
CO2 absorption	+ good performance also with semi-transparent clouds – accurate only in the high levels of the troposphere;
Shadow length	+ easy to apply; requires no additional data – possible only during daytime; retrieves the height of the cloud horizontal edge and not its top
Stereoscopy	+ high accuracy; requires no additional data;
Optimal estimation	+ includes error estimate – requires atmospheric profiles, and radiative transfer

of the thesis is to develop a stereo vision system that estimates the cloud top height of an area above Belgium using pairs of images obtained from the MSG3 and MSG1 satellites. In realising our main aim a set of specific objectives has been defined, as shown below:

1. Implementation and comparative evaluation of registration methods in order to refine the pre-rectified dataset.

2. Application and assessment of Graph-Cut based stereo matching algorithms for disparity estimation.

3. Evaluation of the results with respect to the estimated accuracy.

This thesis is structured as follows. Chapter 2 presents technical information about the Meteosat Second Generation Satellites generating the image dataset, details of the available data as well as the LiDAR method for extracting cloud height, which will be used to validate the accuracy of the proposed method. Chapter 3 explains in detail the theoretical background and the computational pipeline of a typical stereo vision system.

Chapter 4 presents the preprocessing steps of the proposed pipeline. Taking into consideration that the input dataset was already rectified, special emphasis is put on the registration of the clear-sky image pair in order to refine the rectification. Experimental results are presented and discussed.

Chapter 5 is dedicated to the theory of Markov Random Fields and Graph Cuts, which is necessary in order to understand the proposed solution to the stereo problem. In addition, the results obtained and their comparison to the corresponding LiDAR validation data are presented.

Finally, conclusions and discussion are presented in the final chapter of the thesis.

2. Meteosat Second Generation Satellites

In the present Chapter we present the Meteosat Second Generation and provide a description of the instrument that is used to generate the images of the observed parts of the earth (SEVIRI). In addition, the necessary theory for a LIDAR system, which is used to validate the results, is presented.

2.1 MSG

The Meteosat series of satellites are geostationary meteorological satellites operated by the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) under the Meteosat Transition Programme (MTP) and the Meteosat Second Generation (MSG) program. The primary mission of MSG is the continuous observation of the Earth. The main instrument of a MSG satellite is the Spinning Enhanced Visible and Infrared Imager (SEVIRI), which observes the full disk of the Earth with a repeat cycle of 15 minutes in 12 spectral wavelength channels. The MSGs are geostationary meteorological satellites, which means that they move with the same rotational speed as the Earth with a geosynchronous orbit exactly above the equator (latitude 0°) and thus maintaining their position with respect to a location on Earth [6]. In our study, two MSG satellites (MSG-1 and MSG-3) provided the image dataset that was needed.

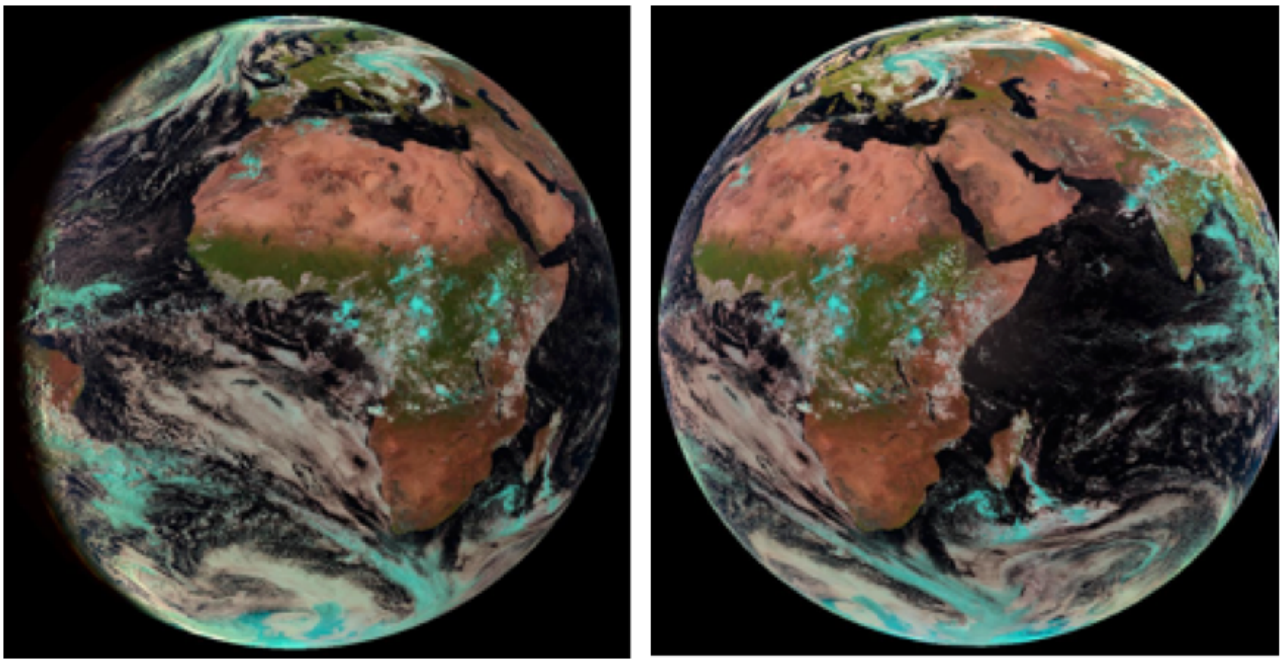


Figure 2.1: Illustration views from MSG-3 at 0° longitude and MSG-1 at 41.5°E

MSG-1 (or Meteosat-8) lunched on the 28 August 2002 and became operational on the 29 January 2004. The launch of the third MSG, Meteosat-10, was timed on 5 July 2012 [7], with as main mission to replace the ageing MSG-1.

Taking into consideration the official EUMETSAT website [8], both satellites orbit at an altitude of 36 000 km. As far as longitude is concerned, at the time when the images were taken, MSG-3 made observations at 0° over the west part of Africa and MSG-1 at 41.5°E over Indian Ocean (Figure 2.1).

This dual-satellite system guarantees continuity of service if one satellite fails. At the same time, it is possible to use both satellites' SEVIRI sensors to acquire dual views of areas observed by both satellites.

Geostationary satellites have helped the research community in extracting valuable information about the Earth's ecosystem and especially cloud properties, but there are several challenges that still need to be overcome. The MSG satellites orbit at an altitude of approximately 36 000 km above the equator. However, the clouds are encountered over a range of altitudes from 0 to 18 km (see Figure 2.2) according to the World Meteorological Organization [9]. The difference between the clouds' and the satellites' altitude is large enough to result in two different clouds, which have little top height difference, being indistinguishable. The lack of texture on the surface of the clouds is also a significant challenge for stereo vision systems.

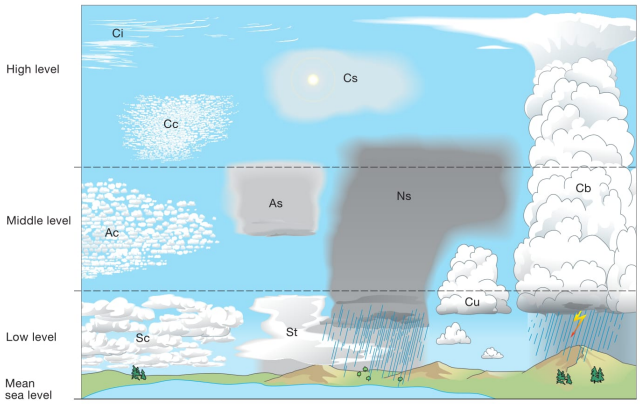


Figure 2.2: Cloud Levels (taken from [9])

2.2 SEVIRI

The SEVIRI, which shown in Figure 2.3, is the main instrument for observing the Earth. It is equipped with a 50cm diameter aperture. While the satellites spins counter-clockwise at 100 rpm around their longitudinal axis, which is aligned with the Earth's rotational axis, SEVIRI scans the Earth line by line. SEVIRI' detectors are sensitive to 12 bands of the electromagnetic spectrum. The system incorporates four Visible and Near InfraRed (VNIR) channels and eight InfraRed (IR) channels. The High-Resolution Visible (HRV) channel is included within the bands of the VNIR channels and it contains 9 broadband detection elements to scan the Earth with a 1 km Spatial Resolution. All the other channels (including the IR channels) are designed with 3 narrow band detection elements per channel, scanning the Earth with a 3 km sampling distance.

As reported in [10] and [11], 1249 scan line steps in the south to north direction are needed in order to obtain the full Earth disc image. The satellite's spin of 100 rpm allows to complete a full image in the east to west direction in about 12.5 min. The Earth observation is resumed after the scanning mirror is driven back to its initial position, leading to an overall repeat cycle of 15 minutes. Finally, there is a continuous raw data transmission to Earth at L-band (1-2 GHz) at 3.2 Mbps.

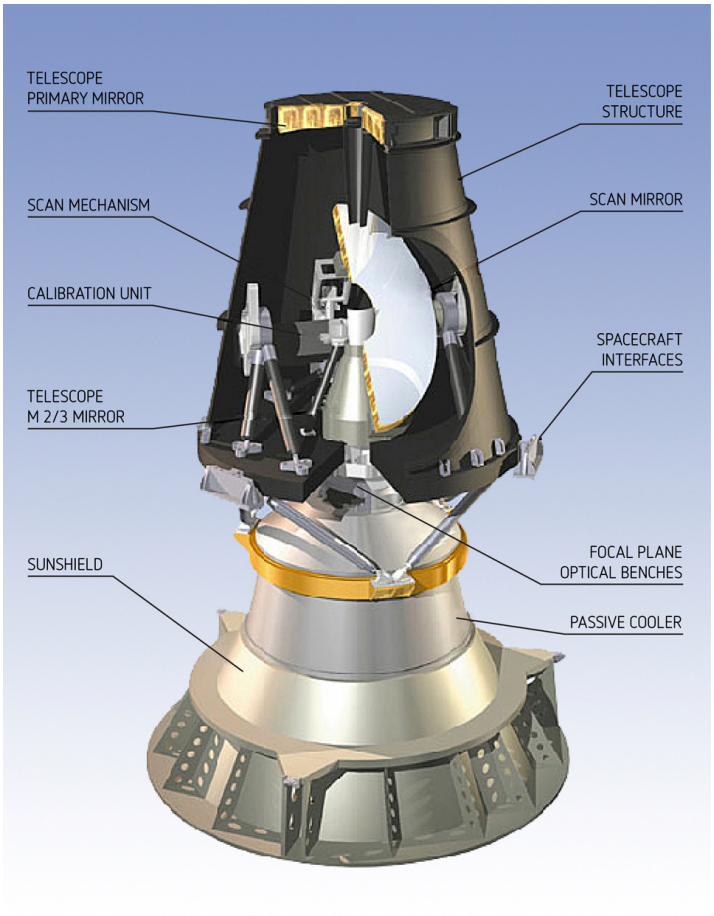


Figure 2.3: SEVIRI Instrument Main Unit

2.3 LIDAR as a validation method

In the framework of our study, LIDAR data is used to evaluate the accuracy of the computational pipeline to be implemented for the estimation of cloud-top height.

Light Detection and Ranging (LIDAR) is a remote sensing method that uses light in the form of pulsed laser to detect distances. This technology achieves high accuracy and precision with a very high resolution and also, because it uses emitted light, it works independent of the ambient light. A pulse of laser light is emitted into the sky, and the amount of light scattered back from the atmosphere is measured versus time. Knowing the speed of light, the time is converted into height. The amount of light returned from each height is proportional to the atmospheric density.

Based on different platforms, LIDAR systems can be divided into airborne, terrestrial and in some occasions spaceborne types. Regarding airborne types, airplanes and helicopters are most commonly used for acquiring LIDAR data over broad areas. Two types of airborne LIDAR systems are *topographic* and *bathymetric*. Topographic LIDAR typically uses a near-infrared laser to map the land, while bathymetric lidar uses water-penetrating green light to also measure seafloor and riverbed elevations.

A typical LIDAR system consists of the following components:

1. A Laser Scanner and Receiver

2. A GPS Receiver, which gives the aircraft's position.

3. An Inertial Measurement Unit (IMU), which gives the roll, pitch and yaw of the aircraft.

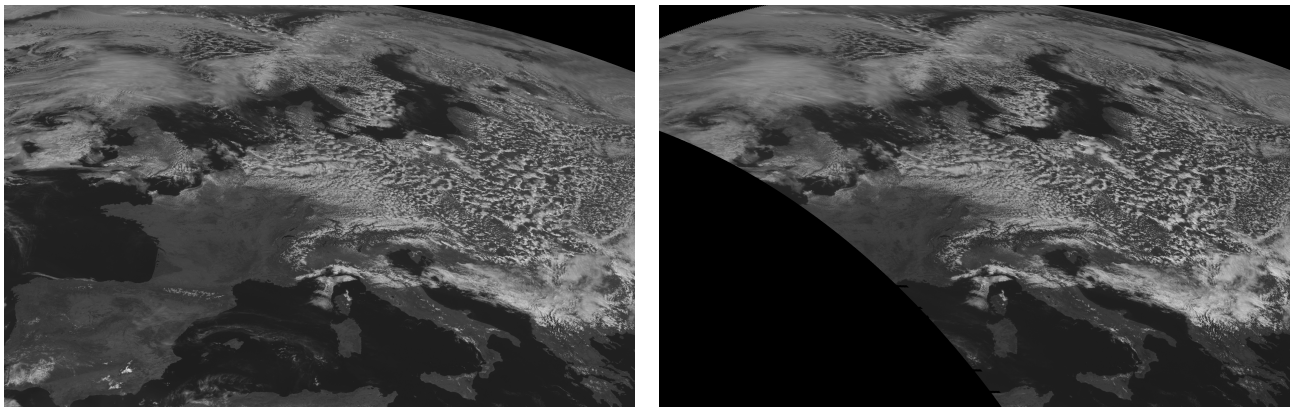
Knowing the position and orientation of all these components enables accurate measurements to be recorded by the Lidar system. LIDAR systems allow scientists and mapping professionals to examine both natural and manmade environments with accuracy, precision, and flexibility.

2.4 Available data of the present thesis

The Dataset was provided by the Royal Meteorological Institute of Belgium (KMI). It consists of:

- 26 pairs of images of Europe, captured at 0° and $41.5°E$ from MSG-3 and MSG-1 respectively.
- 1 pair of clear sky images of Europe acquired by taking the minimum value of each pixel from all the previous 26 images.
- 1 image that displays the relative displacement in number of pixels between the MSG3 and MSG1 images if the altitude changes from 0 km (sea level) to 10 km.
- 1 image that displays the displacement in number of pixels in MSG3 image if the altitude changes from 0 km (sea level) to 10 km.
- 1 image that displays the angle of displacement in degrees between the MSG3 and MSG1 images if the altitude changes from 0 km (sea level) to 10 km.
- 1 image that displays the angle of displacement in degrees in the MSG3 image if the altitude changes from 0 km (sea level) to 10 km.

The images from MSG-1 have been remapped to 0° to match the perspective of the MSG-3 images and can be seen in Figure 2.4.



(a) MSG-3 View at 0° (b) MSG-1 View at $41.5°E$

Figure 2.4: Stereo Pair Images

The provided data's format is PGM which stands for Portable Gray Map. A PGM image represents a grayscale image and consists of the following [12] :

1. A magic number for identifying the file type. It can be either **P2** or **P5**, for ASCII or binary representation of the data, respectively.
2. The width and height of the image represented as ASCII characters in decimal seperated by a whitespace.
3. The maximum gray value $0 < x < 2^{16} (= 65536)$.

4. The uncompressed image data in the form of a matrix of size $height \times width$.

However, during the preprocessing stage the data are converted to PNG format, due to the fact that PNG is a widely accepted format in most computer vision or image processing software libraries.

3. Stereo Vision Theory

This chapter explains the theory behind a typical stereoscopic system. Firstly, the mathematical model of a single camera as well as some common pre-processing steps are presented. Afterwards, the theory expands to the concept of a stereo camera system and the extraction of depth information using such a system.

Stereo Vision is part of Computer Vision and is inspired by the biological process Stereopsis. It studies the extraction of 3D information for a pair of digital images. The significant tasks of a typical stereo vision system are camera calibration, stereo correspondence problem and finally 3D reconstruction of the scene captured by a dual-camera system. In order to assess the procedure of finding the correspondences, a rectification step of the two images precedes the stereo matching. Finally, the disparity map which is produced during an stereo matching step can be used to reconstruct the 3D model of the scene, given that the calibration parameters are available.

Summing up, by comparing information about a scene or an object from two vantage points, we can extract 3D information by examining the relative position of the object in the two images.

3.1 History of Computer Vision

Vision or visual perception is the acquisition of knowledge about objects and events in the surrounding environment through information processing of light emitted or reflected from objects. Computer Vision was meant to mimic the human visual system, when it was first introduced in the 1960s by universities trying to develop artificial intelligent robotic systems [13] and the clear distinction from digital image processing was the desire to recover the 3D structure of the real world. In 1966, it was believed that such a problem would be so easy that it could be an undergraduate summer project [14].

In the 1970s, research focused on ”low-level” vision tasks, such as edge and corner detection, line labelling and 3D modeling of non-polyhedral objects. Some early feature-based stereo correspondence algorithms were introduced.

In the 1980s, research was focused on more complex mathematical models for performing quantitative image and scene analysis. These include the concept of scale-space, the inference of shape from various cues such as shading, texture and focus, and contour models known as snakes. It was also realized that many of these mathematical concepts could be described using the same mathematical and optimization framework, as regularization and Markov Random Fields.

In the 1990s, 3D reconstruction and camera-calibration were some of the advanced topics in which research efforts focused on. Progress was also made on the dense stereo correspondence problem and further multi-view stereo techniques.

In the 2000s, an interaction between the fields of computer graphics and computer vision was developed. Some advancements are panorama image stitching, image morphing and view interpolation. During the last decade, significant progress has been made in the development of more efficient algorithms for complex global optimization problems, such as techniques based on graph cuts and message passing algorithms, such as loopy belief propagation in the context of Markov Random Fields.

A more recent trend relates to the application of sophisticated machine and deep learning techniques to computer vision problems.

A timeline of the computer vision history is illustrated in Figure 3.1.

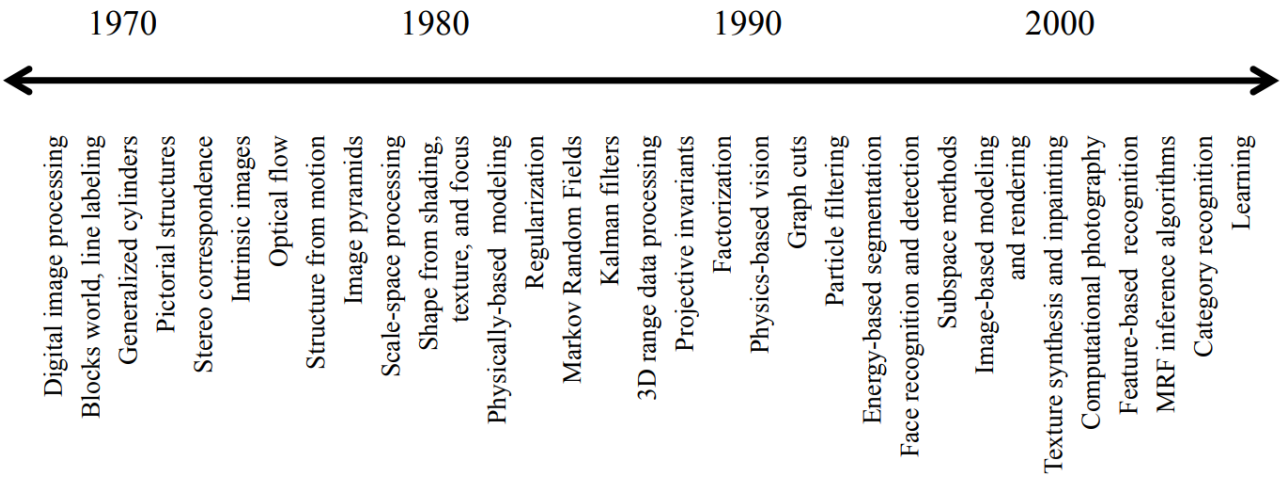


Figure 3.1: A timeline of some of the most active topics in computer vision research.

3.2 Camera Model

The process of acquiring images through cameras is a mapping between 3D world and a 2D image. The simplest and most commonly used camera model is the pinhole camera model. It is a simple camera without any lens but with a tiny aperture that lets light pass through it. An image of a scene captured from a pinhole camera is projected as a reversed and inverted image on the image plane (also called sensor plane), as illustrated in Figure 3.2. Although digital imaging has evolved, such a model is still the principal guide and basis of most computer vision algorithms. Despite its simplicity, it provides an acceptable approximation of imaging process and it is mathematically convenient. Because perspective projection creates inverted images, it is convenient to consider a virtual image associated with an image plane lying in front of the pinhole at the same distance as the original image plane.

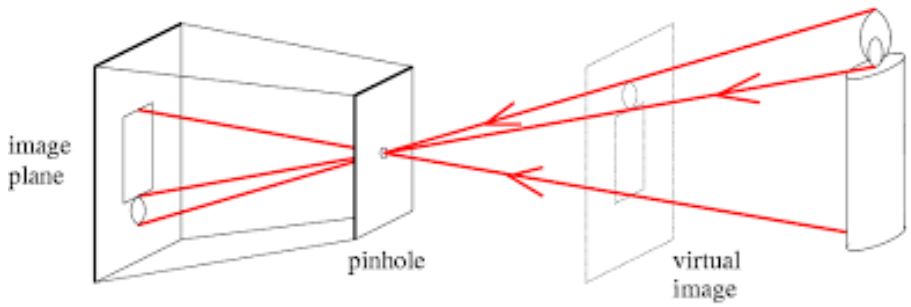
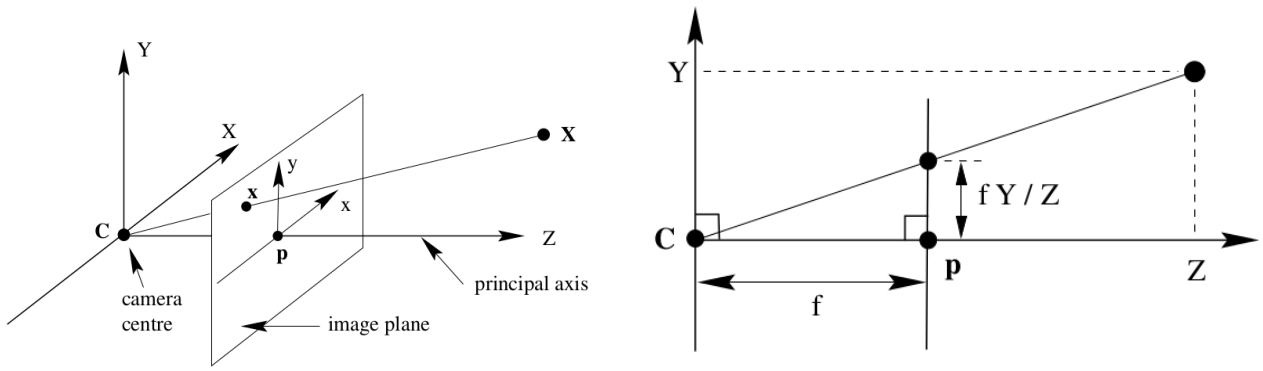


Figure 3.2: Pinhole Model - The candle is projected to the sensor plane inverted and reversed through a pinhole. The virtual image plane can be seen in front of the pinhole at distance equal to focal length of the camera.



(a) (b)

Figure 3.3: **Pinhole Camera Geometry** The image plane is at a distance f in front of the camera's origin **C** and a non-inverted image is formed on it. **p** is the principal point on the image plane. A real-world 3D point **X** can be projected to (virtual) image plane as **x**.

3.2.1 Intrinsic Parameters

The pinhole camera model gives the mathematical relationship between 3D points and their projection, as illustrated in Figure 3.3. Let the centre of projection C be the origin of a Euclidean coordinate system, and consider the plane Z = f , which is called the image plane or focal plane. A point in 3D space $\mathbf{P} = (X, Y, Z)^T$ is mapped to the point ($\mathbf{P_c}$) in image plane where a line joining P and C meets the image plane. Because the triangles Cpx and CZX are similar triangles, the point on image plane is $P_c = (fX/Z, fY/Z)^T$. The line from the camera centre perpendicular to the image plane is called the principal axis of the camera, and the point where the principal axis meets the image plane is called the principal point.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

If we represent the world and image coordinates by homogeneous vectors, then the central projection can be expressed as a linear matrix relationship (3.1) of the real world and image coordinates. This expression assumes that the origin of coordinates in image plane is at the principal point.

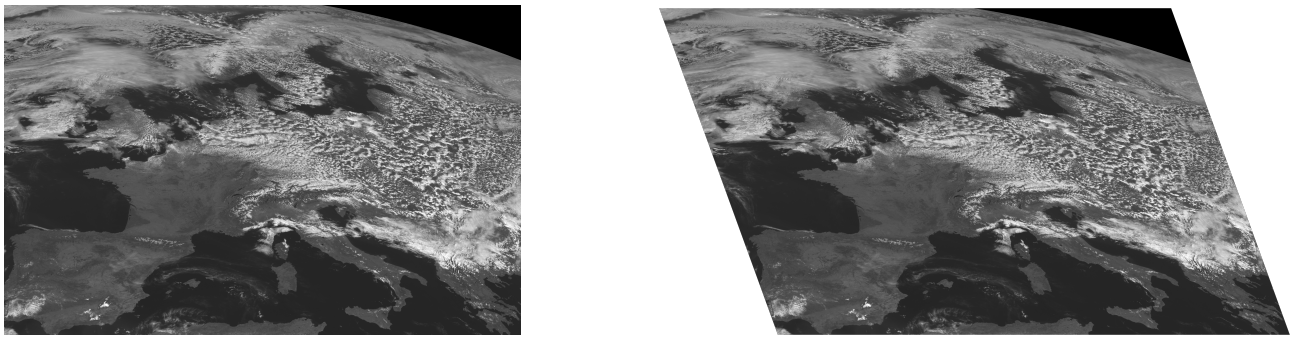
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \left[I_{3\times3} | O_{3\times1} \right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.1)$$

The matrix **K** is called the camera calibration matrix and it is composed of the intrinsic parameters of the camera. As expressed in (3.1), K contains only one intrinsic parameter, the focal length. We can extend it so that it matches a more accurate approximation, as seen in (3.2).

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

The parameters c_x and c_y are the offsets of the principal point in the reference frame of the image. The ideal model assumes that the image coordinates have equal scales in both axial directions, which might not be the case most of the times. This is why focal length f is separated into two parameters

$f_x = m_x f$ and $f_y = m_y f$. The $m_{x,y}$ are the scale factors in each x, y direction. The final parameter s is called *skew* (Figure 3.4). For most cameras the skew is zero, as well as the focal lengths in each of the two axes are equal $f_x = f_y$ [15].



(a) (b)

Figure 3.4: (a) Original MSG3 image taken on 23/04/2017 at 12:00 (b) Skewed image by 20° anti-clockwise.

3.2.2 Extrinsic Parameters

If the 3D world coordinates need to be expressed in another coordinate system than that of the camera, then the extrinsic parameters of the camera are needed. Such a transformation consists of a translation and a rotation as expressed in the composite matrix (3.3).

$$\begin{bmatrix} \mathbf{R_{3\times 3}} & \mathbf{t_{3\times 1}} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{3.3}$$

The extrinsic parameters denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. In other words, extrinsic parameters define the position of the camera center and the camera's heading in world coordinates. **R** is the rotation matrix. In particular, **t** is position of the origin of the world coordinate system expressed in coordinates of the camera-centered coordinate system and should not be confused with the the camera's position.

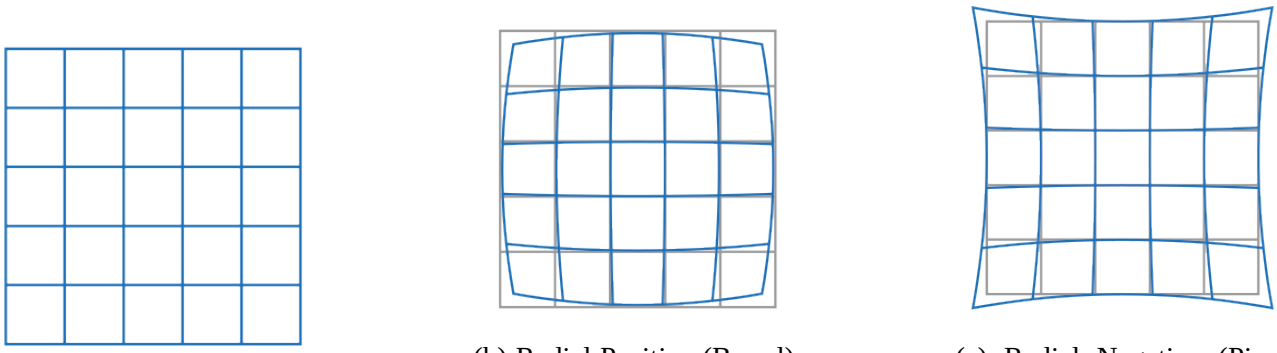
The matrix (3.3) indicates that there are 12 extrinsic parameters, 3 for the translation and 9 for rotation. However, there are only 3 degrees of freedom in a 3D world rotation and therefore only 3 rotation parameters are unique in the rotation matrix.

Both the intrinsic and extrinsic parameters of a camera are needed in order to effectively decompose and understand the process of mapping 3D real world coordinates to image coordinates. Matrix (3.1) can be rewritten in a more complete form as the projection (or camera) matrix **P** in equation (3.4). Such a model is used in the cameras calibration procedure, which is an essential task in order to achieve accurate image rectification and 3D reconstruction as discussed later.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} R_{3\times 3} | t_{3\times 1} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.4}$$

3.2.3 Image Distortion

Although the pinhole camera model is an accurate approximation and preferred mathematical representation of a real camera's model, in reality cameras do not employ such a model to take pictures, but instead they use lenses in order to focus the light. Lenses introduce distortion to the image, mostly radial but also some times tangential distortion [16].

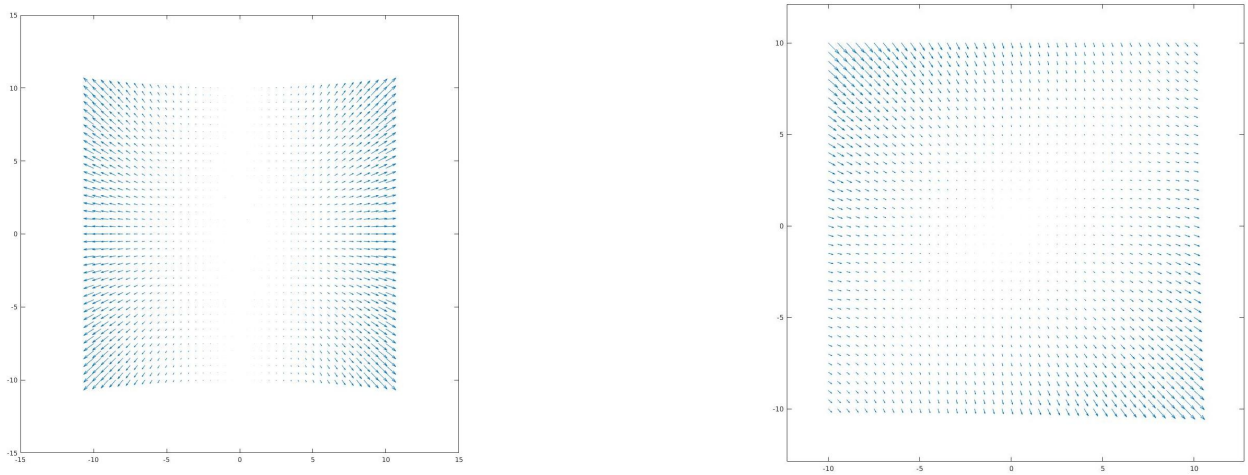


(a) No Radial Distortion

(b) Radial Positive (Barrel) Distortion

(c) Radial Negative (Pincushion) Distortion

Figure 3.5: Lenses Radial Distortions



(a) Negative Radial Distortion

(b) Tangential Distortion

Figure 3.6: Visualization of radial and tangential distortion using vector fields.

Taking into account these distortions, the previously described mathematical model can be extended as in [17] [13]. The coordinates (x, y) of a 3D point (X, Y, Z) projected in the image plane are now given by the following equations:

$$\begin{aligned} x &= f_x x''' + c_x \\ y &= f_y y''' + c_y \end{aligned} \tag{3.5}$$

$$\begin{aligned} x''' &= x'' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x'' y'' + p_2(r^2 + 2(x'')^2) \\ y''' &= y'' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x'' y'' + p_1(r^2 + 2(y'')^2) \end{aligned} \tag{3.6}$$

$$x^{''} = \frac{x^{'}}{z^{'}}$$
$$y^{''} = \frac{y^{'}}{z^{'}} \tag{3.7}$$

$$\begin{bmatrix} x^{'} \\ y^{'} \\ z^{'} \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ X \end{bmatrix} + t \tag{3.8}$$

The k_i are the radial distortion coefficients and p_j the tangential distortion coefficients.

3.3 Epipolar Geometry

The epipolar geometry is the intrinsic projective geometry between two views and it depends only on the cameras' internal and external parameters. It is mainly used in order to determine corresponding points between a stereo image pair. Suppose, in Figure 3.7, a 3D point X and its projections onto each image plane x and $x^{'}$. The plane π is called the epipole plane and the line $CC^{'}$ is called the baseline. The intersections of the baseline with each image plane e and $e^{'}$ are called epipoles and are the projection of each camera center, C and $C^{'}$, to their respective image planes. The line $l^{'}$ is called the epipolar line, as it connects the epipole $e^{'}$ and the projection of X in the right image plane, $x^{'}$. Furthermore, the epipolar lines are the intersection of the epipolar plane with the image planes.

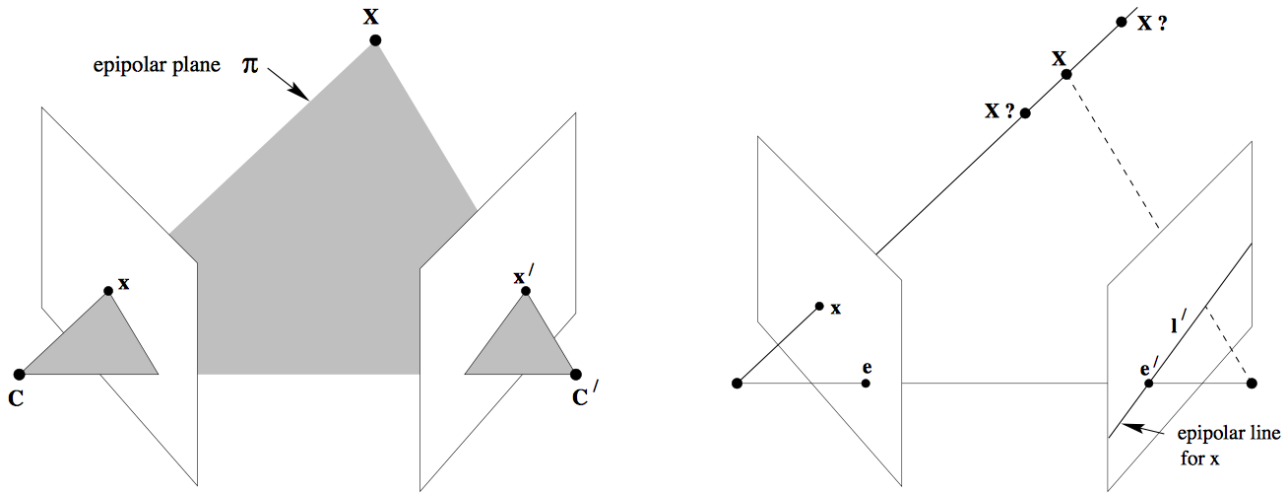


Figure 3.7: Epipolar Geometry

It is clear that the rays projecting back from x and $x^{'}$ intersect at X and are coplanar lying in π. Supposing we only know x, we need to find how the $x^{'}$ is constrained. The yet unknown point $x^{'}$ lies in the epipolar line $l^{'}$ corresponding to x. The search for the point corresponding to x is now limited to searching the epipolar line $l^{'}$.

This is known as the epipolar constraint (see Section 3.4.5) and is defined as ([18] [19])

$$(x^{'})^T F x = 0 \tag{3.9}$$

where F is the fundamental matrix of size 3×3 and has a $rank = 2$. The epipolar line is defined as

$$l^{'} = Fx \tag{3.10}$$

Although there are 9 parameters in F, only 7 of them are independent. The estimation of F is out of the scope of this thesis, but many methods for this purpose are described in [20]. If the intrinsic parameters of the camera are known then the fundamental matrix **F** becomes the essential matrix **E**.

3.4 Typical Stereo Vision Pipeline

Stereo Vision is part of the scientific field of Computer Vision, which is inspired from the biological process of stereopsis. Stereopsis is a term that is most often used to refer to the perception of depth and 3-dimensional structure obtained on the basis of visual information deriving from two eyes by individuals with normally developed binocular vision. Thus, stereo vision studies the extraction of 3D information for a pair of digital images. By comparing information about a scene or an object from two vantage points, we can extract 3D information by examining the relative position of the object in the two images.

3.4.1 Disparity

Prior to introducing the computational pipeline of a typical stereo vision system, we need to introduce a very important concept, i.e. the concept of disparity. Consider the case presented in Figure 3.8, in which two cameras are perfectly aligned, such that their coordinate systems are coplanar. In that case, each row in one image and its counterpart in the other image are located in the same line. The disparity is the apparent displacement of an object in both images. The depth of an object increases as the disparity of that object between the images decreases.

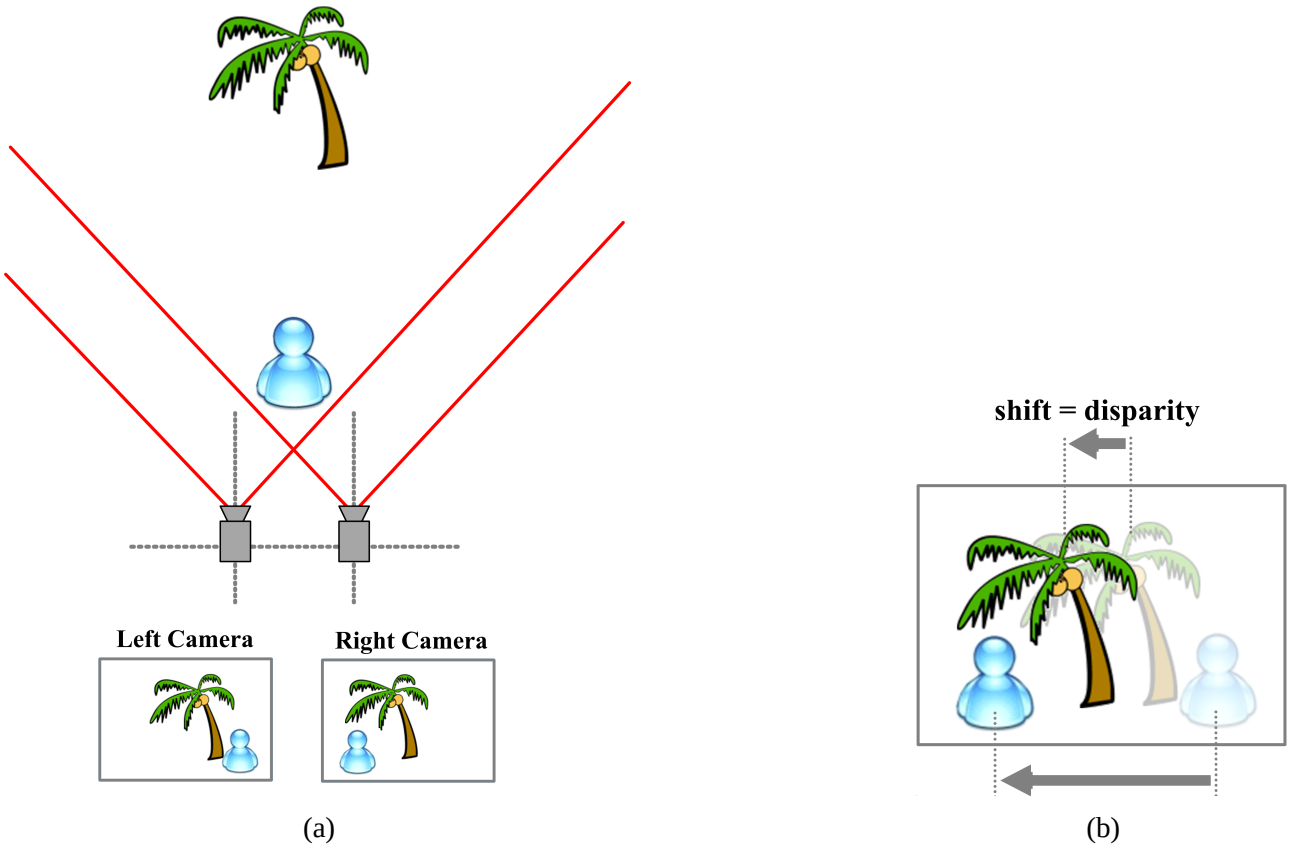


Figure 3.8: Concept of disparity illustrated by a setting of two perfectly aligned cameras. The disparity is the apparent displacement of the objects between two views of a scene.

3.4.2 Stereo Vision

Most stereo vision systems follow a distinct pipeline of computational steps in order to extract the 3D information of a scene [21]. In Figure 3.9, a typical stereo system pipeline is illustrated and as explained below.

1. The camera must be calibrated and the intrinsic and extrinsic camera parameters must be estimated. The image distortions must be corrected in order to match the projection of an ideal pinhole camera.

2. The two images must be projected back to a common plain, known as image rectification. This is necessary to allow comparison of the image pairs in order to find matching points.

3. The corresponding points between the two images must be found. These lie on the same row of the two images if the pair is rectified. This step is known as the Correspondence Problem

4. The disparity data and Disparity Map are calculated.

5. The depth estimation after triangulation to disparity data.

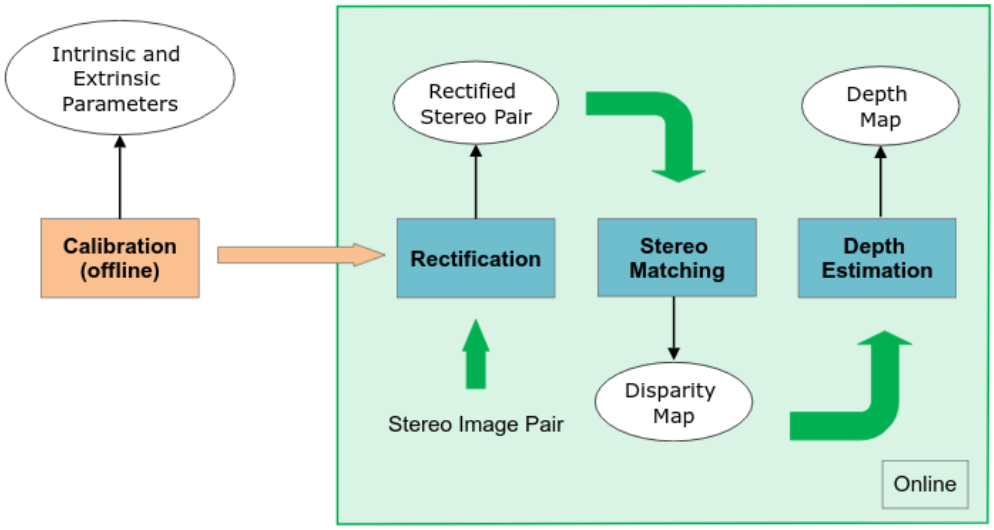


Figure 3.9: Stereo Vision Pipeline

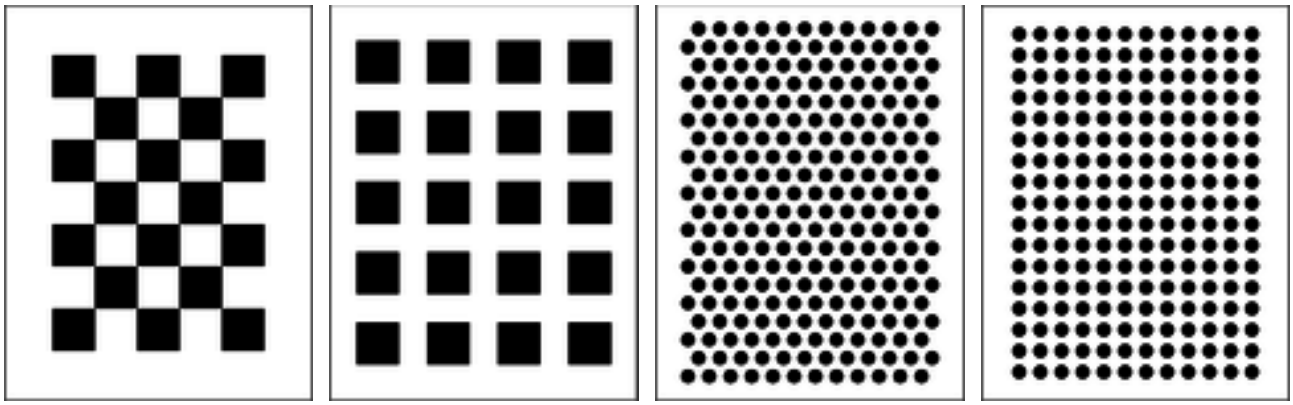
In the subsequent sections of this chapter we present in detail the theoretical foundations of the various modules of this pipeline and in Section 3.5 we discuss the specific modules which form the focus of the present thesis.

3.4.3 Camera Calibration

Camera calibration is an essential step in 3D computer vision, because it estimates the intrinsic and extrinsic parameters of the cameras, as well as any lens distortion that might appear. This calibration procedure is completed offline, which essentially means that we need to calibrate the camera before actually using the stereo vision system.
Many techniques of calibration are available, which can roughly be categorized into:

- **Photogrammetric calibration**, which is performed by observing a 3D object with known geometry.
- **Self-calibration**, which does not use any calibration object.

For most stereo vision systems, a method based on the first technique is used to calibrate the cameras [22]. Some objects that can be used as calibration objects are shown in Figure 3.10. The **chessboard** patterns are the most popular, due to their easily identified corners.



(a) Chessboard (b) Square Grid (c) Circle Hexagonal Grid (d) Circle Regular Grid

Figure 3.10: Calibration objects with known geometry.

After printing the pattern and mounting it on a flat surface, one should take snapshots of the object from various angles and positions as shown in Figure 3.11. Then the location of the chessboard corners should be determined for each image. With the 3D coordinates of the extracted features known, the internal parameters of the camera, as well as its orientation relative to the chessboard, can be estimated.

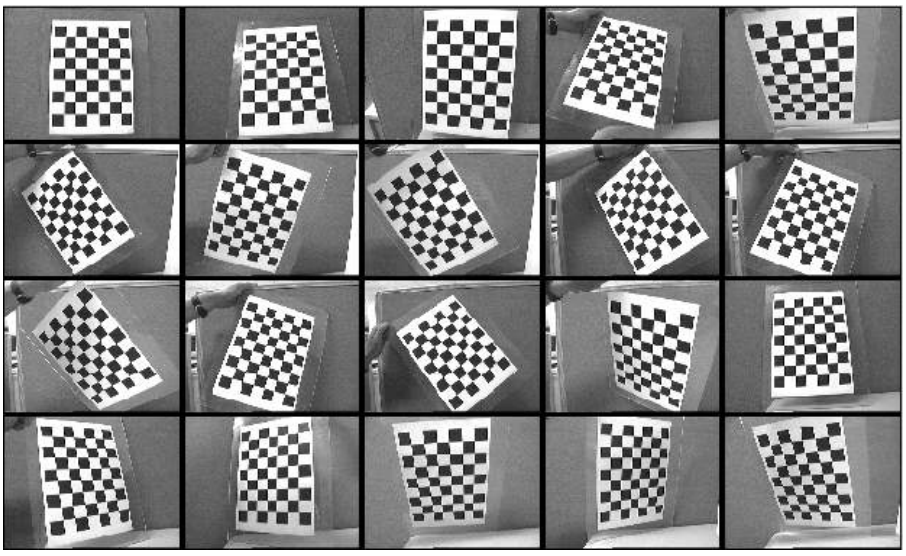


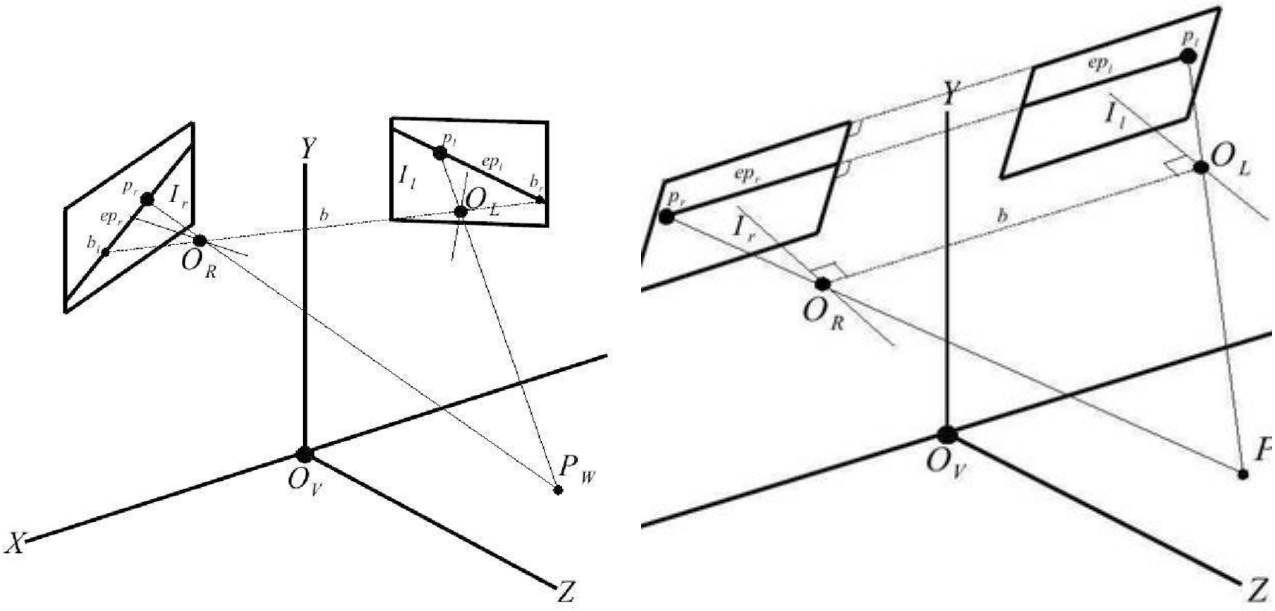
Figure 3.11: Calibration images taken from different angles and positions

3.4.4 Image Rectification

Image rectification is a very important step in every pipeline of a stereo vision system. In general, the cameras of a stereo vision system may be not aligned or even if they are aligned, there is a possibility of error during the assembly process of the physical system as shown in Figure 3.12a. In order

for the stereo pair images to be rectified (Figure 3.12b), a transformation is needed. Such a transformation consists of the following :

- A rotation of the two image planes, in order to make them coplanar.
- A rotation of the epipolar lines to the direction of the scanning lines, in order to make them colinear.



(a) Image pair before rectification

(b) Stereo pair after rectification

Figure 3.12: Rectification example

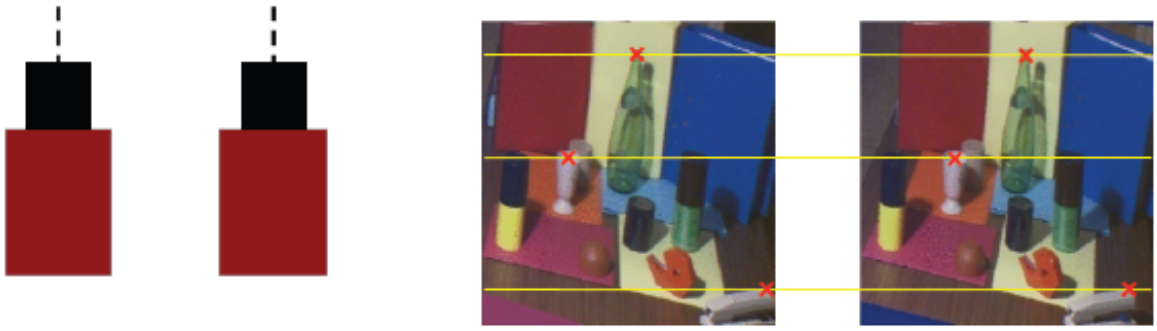


Figure 3.13: Rectified Stereo System Example

The rectified stereo images are coplanar, the epipolar lines are colinear and the epipoles lie at the infinity. Although the corresponding point of x in the second image, $x^{'}$, lies on the epipolar line $l^{'}$, as previously shown in Figure 3.7, this line is dependent on both dimensions of the image plane before the rectification process. However, because the epipolar lines are colinear after the rectification, the corresponding points are also colinear as shown in Figure 3.13. Thus, the correspondence problem degenerates to a 1-Dimensional search.

In the case of the current thesis, the satellites are not perfectly aligned and thus the rectification is necessary. However, the provided data was already rectified up to some extent. More about the data rectification's accuracy will be discussed in Chapter 4.

3.4.5 Stereo Matching

The stereo matching problem is the most important process in the stereo system pipeline, as it is responsible for indicating correspondences between each stereo pair of images. These correspondences are needed in order to obtain the disparity information needed to estimate the depth information in a scene. In most cases, the left image of a stereo pair is set to be the reference image for calculating the disparity values, i.e the disparity map corresponds to the left view. The Figure 3.14 illustrates the left and right view of the famous *Tsukuba* dataset as well as the ground truth disparity map.

(a) (b) (c)

Figure 3.14: Disparity Ground Truth for Tsukuba Dataset (courtesy of the University of Tsukuba). (a) The left view (b) The right view (c) The ground truth disparity data. The gray levels represent the disparity values with the lighter gray colors corresponding to the object being closer to the camera system and the darker the gray the further the object is from the camera system.

Constraints

There exist some constraints that can be exploited in order to more accurately match correspondences in two views.
These constraints include :

- **Similarity** [23]. The reference area or point in one image and the corresponding area or point in the other image must be highly correlated and similar.
- **Uniqueness** [24]. A given point from one image can match at most one point in the other image, and consecutively it can be assigned only one disparity value. This constraint relies on the assumption that the item in the image corresponds to something that has an unique physical position. It is very important to state that a point in one image may not correspond to any point in the other image, and thus it can not be assigned any disparity value.
- **Continuity** [24]. The cohesiveness of matters suggests that the disparity varies smoothly almost everywhere over the image, stating that only a small portion of the image is composed of boundaries that are discontinuous in depth.
- **Ordering** [25]. The ordering of features is preserved across images. Geometrically explained, if $m \leftrightarrow m^{'}$ and $n \leftrightarrow n^{'}$ and if m is to the left of n, then $m^{'}$ should also be to the left of $n^{'}$ and vice versa. The ordering constraint fails at regions known as the forbidden zone (Figure 3.15).
- **Intensity Consistency**. Lambertian Surfaces are assumed, which means that the images' corresponding pixels have similar intensity values.
- **Epipolar**. Given a point m in the left image, the corresponding feature point $m^{'}$ must lie on the corresponding epipolar line.
- **Relaxation**. A global matching constraint to eliminate false matches.

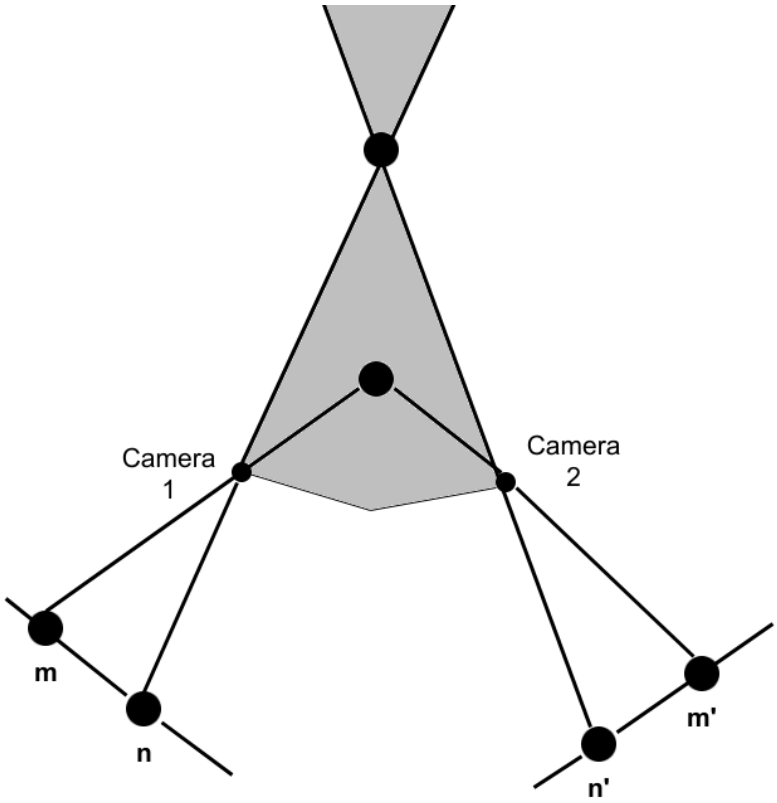


Figure 3.15: The gray areas are the forbidden zone, in which the ordering constraint fails.

Challenges

Stereo matching is a difficult problem considering the following challenges that may be posed by the environment or the camera system.

- **Textureless Regions**. Smooth regions of an image pose a high difficulty in stereo matching and may result in artifacts appearing in some regions of the disparity map.
- **Image Noise**. There are inevitably light variations, image blurring and sensor noise in image formation, which complicates the process of stereo matching.
- **Occlusions**. Occluded areas or objects pose a serious difficulty to the process, as they cannot be matched to any corresponding area or object between the two views.
- **Repetitive Patterns**. Matching similar image regions can be quite difficult when repetitive patterns exist.
- **Specular Reflection**. Occasionally, the light reflections on a surface are perceived differently.

Solving the correspondences problem

Let us begin with describing the simplest stereo matching algorithm. Assuming a rectified stereo image pair as discussed in Section 3.4.4, the epipolar lines are colinear and thus the correspondences lie on the same row of the stereo images. Hence, for a given pixel x on the left image, the algorithm computes a similarity measure for every candidate pixel $x^{'}$ in the corresponding row of the right image and the selected matched pixel is the one with the lowest (highest) disimilarity (similarity) value. This is a naive Winner Takes All (WTA) strategy (see Figure 3.16), which does not perform well due to the fact that there may be a lot of candidate pixels that correlate with the reference pixel (see Figure 3.17). Regularizing the results by applying the constraints that were discussed above (e.g. smoothness assumption) produces better results.

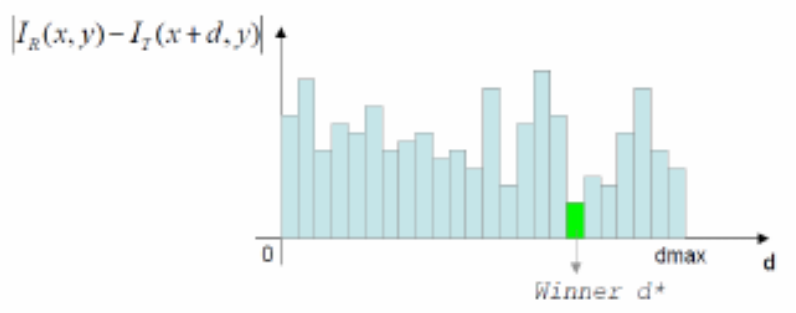


Figure 3.16: WTA strategy. I refers to the image matrix.

Figure 3.17: Disparity Map produced using a WTA strategy.

Usually the implementations of stereo algorithms are categorized into two main classes, local and global methods, though all are based on some of the following concepts ([26] [27] [28]) :

1. **Matching Cost** is used for measuring the (dis)similarity between the processed image areas. Some common functions are Mean Squared Error (MSE), Sum of Squared Differences (SSD), Sum of Absolute Differences (SAD) and Normalized Cross Correlation (NCC). Non-parametric measures are also possible, such as Census or Rank transforms combined with Hamming distance measurements, as well as binary matching costs based on binary features such as edges.

2. **Cost Aggregation** is mostly used in local methods for summing or averaging the matching cost over a support region of the Disparity Space Image (DSI).

3. **Disparity Computation and Optimization**. In local methods, the disparity computation is a (trivial) simple Winner Takes All (WTA) strategy. On the other hand, global methods perform most of their work in the disparity computation step, and they skip the previous aggregation step. These methods are mainly conceived as an energy-minimization problem.

4. **Disparity Refinement** is applied to improve the accuracy and consistency of the computed disparity map. Examples include sub-pixel disparity computation and weighted median filtering.

Concerning the **local** methods, a window W is assumed around the candidate pixel, which aggregates the matching cost for calculating the disparity value. The size of the window plays an important role in the quality of the disparity map. Small window sizes preserve depth discontinuities but fail in textureless or repetitive surfaces and are vulnerable to noise. On the other hand, large window sizes produce smoother results but suffer at preserving the object boundaries. In the middle ground, one can apply adaptive filtering to cope with these disadvantages while still taking advantage of the

mentioned benefits. Some examples of local methods are Block Matching [29] [30], Feature Matching [31] [32] and Phase Matching [33] based algorithms.

The second category includes the most commonly used **global** methods, which try to optimize an energy function, e.g.

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \tag{3.11}$$

over the entire image, regarding some constraints. The $E_{data}(d)$ term measures image consistency and is defined as

$$E_{data}(d) = \sum_{(x,y)} C(x, y, d(x, y)) \tag{3.12}$$

where C measures how well the disparity value fits the pixel (x, y) given the observed data. Likewise, the $E_{smooth}(d)$ term is responsible for encoding the smoothness constraint and is described as

$$E_{smooth}(d) = \sum_{<p,q>\epsilon N} s(d_p, d_q) \tag{3.13}$$

where N is a set of spatially neighboring pixels in the left image, and s is a penalty function which which assigns a penalty if p, q disparities are different. λ is used as a balancing term between these two energy terms. There exist many energy functions such as (3.11) that are incorporating various combinations of the constraints. Even for the simple Potts model,where $s = 0$ if $d_p = d_q$ and $s = 1$ otherwise, finding the global minimum of (3.11) is an NP-complete problem [34]. Given such an energy function, the correspondence problem reduces to an optimization problem, for which powerful strategies do exist that approximate the global minimum, such as Dynamic Programming [35], Graph Cuts [34] [36] [37] [38] or Message Passing with Belief Propagation [39] [40] [41] in the context of Markov Random Fields [42] [43] [44].

In this thesis, some global methods based on MRFs are implemented and tested as discussed in Chapter 5.

3.4.6 3D Reconstruction - Depth Estimation

After having calculated the disparity data, one needs to apply triangulation in order to calculate the depth information of the scene. Given the setting in Figure 3.18 which illustrates a rectified stereo image pair with the optical axes of the cameras being coplanar and parallel, one can calculate the depth Z of an observed 3D point $X(x, y, z)$, using only simple geometrical relationships between similar triangles. Below the case that the observed point X lies on the segment-line AB, between the two optical axes of the cameras, is being decomposed.

The span between the two camera centers C_l and C_r is called the baseline B. The middle of the baseline is set as the origin $O(0, 0, 0)$ of the Cameras Coordinate System, i.e. the Z axis is coplanar and parallel to the optical axes of the cameras. The intersection of the image planes with their respective camera's optical axis is the principal point of the image plane. The two cameras have the same focal length f. The projections of the 3D point X are x and $x^{'}$ for the left and right image plane respectively.

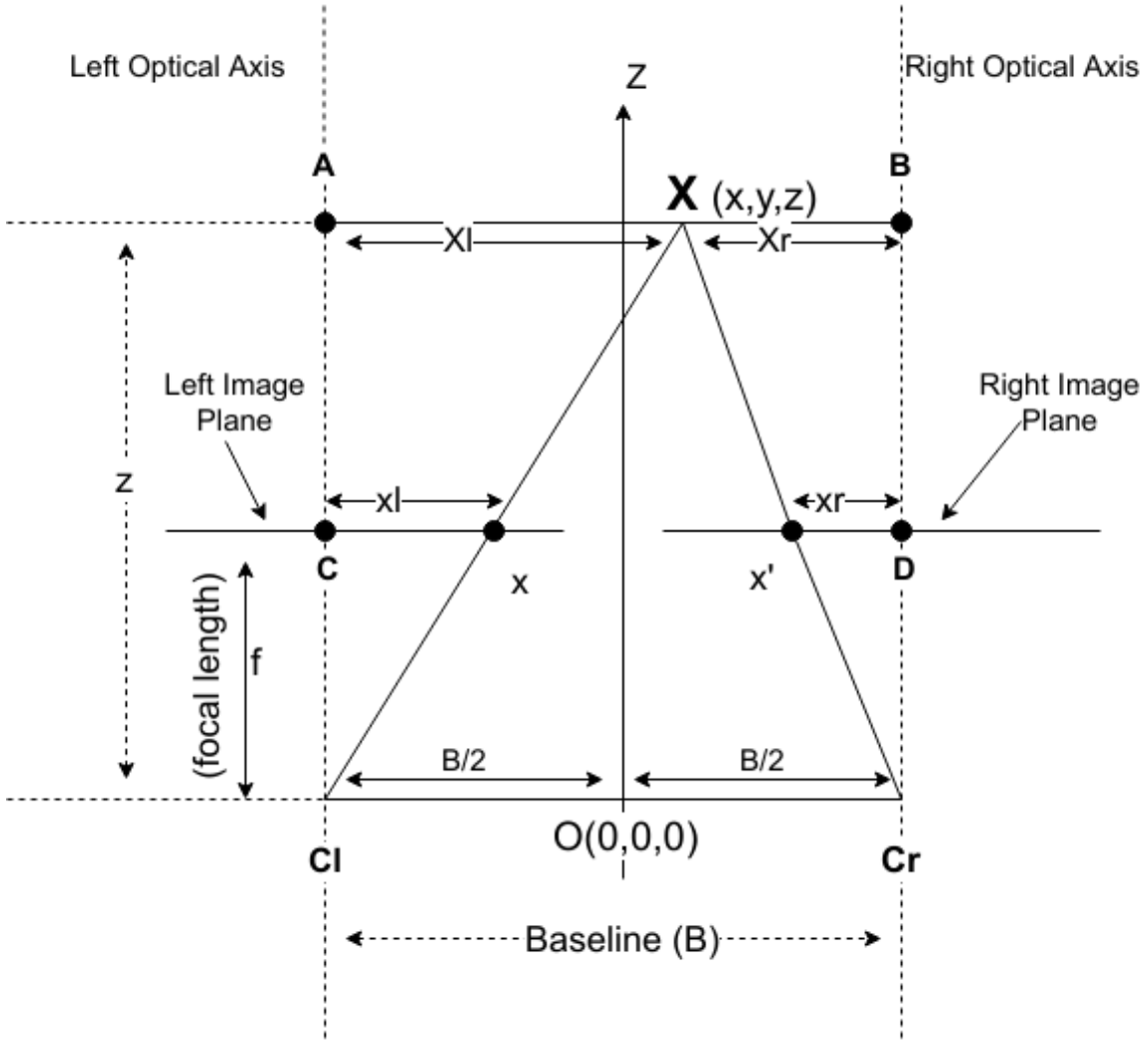


Figure 3.18: Triangulation Example. C_l and C_r are the centers of left and right camera respectively. X is the observed 3D point and x and $x^{'}$ are its projections in each plane. B is the baseline between the two cameras and f is their focal length. Lastly, the middle of the baseline is set as the origin $O(0,0,0)$ of the Camera Coordinate System

The angles below are equal.

$$\begin{aligned} C\hat{C_l}x &= A\hat{C_l}X \\ C_l\hat{C}x &= C_l\hat{A}X \end{aligned} \tag{3.14}$$

Therefore the corresponding triangles are similar :

$$\triangle C_l C x \sim \triangle C_l A X \tag{3.15}$$

Considering the respective triangles for the right image plane and camera center :

$$\begin{aligned} D\hat{C_r}x_{'} &= B\hat{C_r}X \\ C_r\hat{D}x_{'} &= C_r\hat{B}X \end{aligned} \tag{3.16}$$

and

$$\triangle C_r C x_{'} \sim \triangle C_r B X \tag{3.17}$$

Because the triangles in (3.15) are similar, we can compute a relationship between their sides.

$$\frac{f}{z} = \frac{x_l}{X_l} \implies \tag{3.18}$$

$$X_l = \frac{x_l z}{f} \tag{3.19}$$

The same computation is done for the sides of the triangles in Figure 3.18.

$$\frac{f}{z} = \frac{x_r}{X_r} \implies \tag{3.20}$$

$$X_r = \frac{x_r z}{f} \tag{3.21}$$

But the following applies :

$$\begin{aligned} X_l &= \frac{B}{2} + x \\ X_r &= \frac{B}{2} - x \end{aligned} \tag{3.22}$$

Therefore, from (3.19), (3.21) and (3.22) we get:

$$\begin{aligned} x &= \frac{x_l z}{f} - \frac{B}{2} \\ x &= \frac{B}{2} - \frac{x_r z}{f} \Rightarrow \end{aligned} \tag{3.23}$$

$$z = \frac{Bf}{x_r + x_l} \tag{3.24}$$

In the case that the 3D point would not lie on the line-segment AB between the two optical axes, but instead at the right side of the right optical axis, equation (3.24) would become :

$$z = \frac{Bf}{x_l - x_r} \tag{3.25}$$

and respectively for the left side of the left optical axis:

$$z = \frac{Bf}{x_r - x_l} \tag{3.26}$$

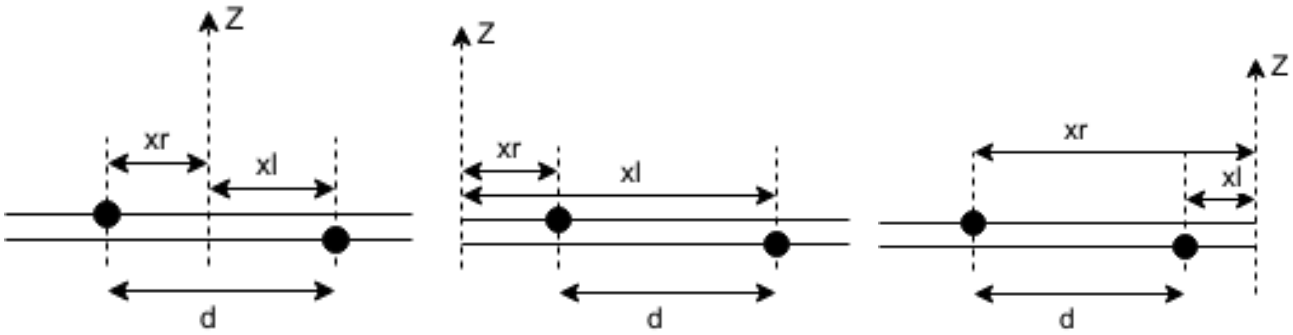
In any case, the equations (3.24), (3.25), (3.26) can be described as :

$$\mathbf{z} = \frac{\mathbf{Bf}}{\mathbf{d}} \tag{3.27}$$

where **d** is the disparity between the projections x and $x^{'}$ in each image plane of a 3D point X. Figure 3.19 visualizes the disparity in the cases when the 3D point X is between, the right or left side of the optical axes for better understanding of the equations (3.24), (3.25), (3.26).

For obtaining the x coordinate of the 3D point X, we substitute (3.27) into (3.23). For each case illustrated in Figure 3.19 the x value is calculated as :

$$x = \begin{cases} \frac{B}{2d}(x_l - x_r) & \text{3D point } X \text{ between the two optical axes} \\ \frac{B}{2d}(x_l + x_r) & \text{3D point } X \text{ at the right of the two optical axes} \\ -\frac{B}{2d}(x_l + x_r) & \text{3D point } X \text{ at the left of the two optical axes} \end{cases}$$



(a) X between the optical axes (b) X at the right of the optical axes (c) X at the left of the optical axes

Figure 3.19: Visualization of disparity value for each different case of the position of the 3D point X respective to the optical axes of the cameras.

In addition, we can utilize the triangle similarities in the YZ plane to attain the following relationship about y coordinate

$$\frac{y_l}{f} = \frac{y_r}{f} = \frac{y}{z} \tag{3.28}$$

Based on equations (3.28), (3.27) :

$$y = \frac{By_l}{d} \tag{3.29}$$

The same applies for the x coordinate

$$x = \frac{Bx_l}{d} \tag{3.30}$$

While we have not specifically mentioned any algorithmic solution regarding any of the steps composing a typical stereo vision system, we have gained the necessary mathematical background in order to understand any proposed solution of these steps. Regarding this thesis, the rectification step was already applied to the provided image dataset. As far as the stereo matching and disparity calculation method is concerned, a global approach based on Markov Random Fields is used.

3.5 Steps of the typical pipeline implemented

With reference to Figure 3.9 which represents the typical pipeline of an integrated stereo vision system, in the context of the present thesis we will focus on the implementation or/and application of the following steps.

1. Processing of the rectified data before applying the stereo matching (disparity estimation) algorithm.

2. Disparity estimation using algorithms based on Markov Random Fields.

The camera calibration parameters were not made available to us and we could not calculate them either. Moreover, the rectification of the image pairs was already applied before the data was provided to us.

The final module of the typical pipeline which relates to the calculation of the actual depth of the scene, based on the previously calculated disparity map, will not be implemented. This is due to the fact that the cameras' calibration parameters are necessary in order for the depth to be calculated using the triangulation method shown in Figure 3.18.

4. Data Pre Processing

This chapter presents the pre-processing steps that are applied to the dataset. Specifically, two registration techniques based on Mutual Information and Low Level image Features are introduced, implemented and compared in order to register the clear-sky image pair.

Although most stereo matching algorithms work efficiently with rectified stereo images, it is commonly accepted that preprocessing of the rectified images has significant contribution to increased accuracy of the following algorithmic steps. In the case of the satellite data, because the two camera sensors have a longitudinal difference of 41.5°, they perceive differently the intensity of light reflected from the Earth's and clouds' surfaces on them. Thus, we expect that normalizing the images will result in a more objective/realistic image dataset.

Furthermore, an additional source of error is the fact that is a relative displacement in number of pixels between the MSG3 and MSG1 views depending on the longitude and latitude of each location [45] which effect should be minimized. These errors are clearly depicted in Figure 4.1 which visualizes the relative displacement using a hot colormap. From this figure it is obvious that the North-West areas have a larger relative displacement in contrast to the South-East areas.

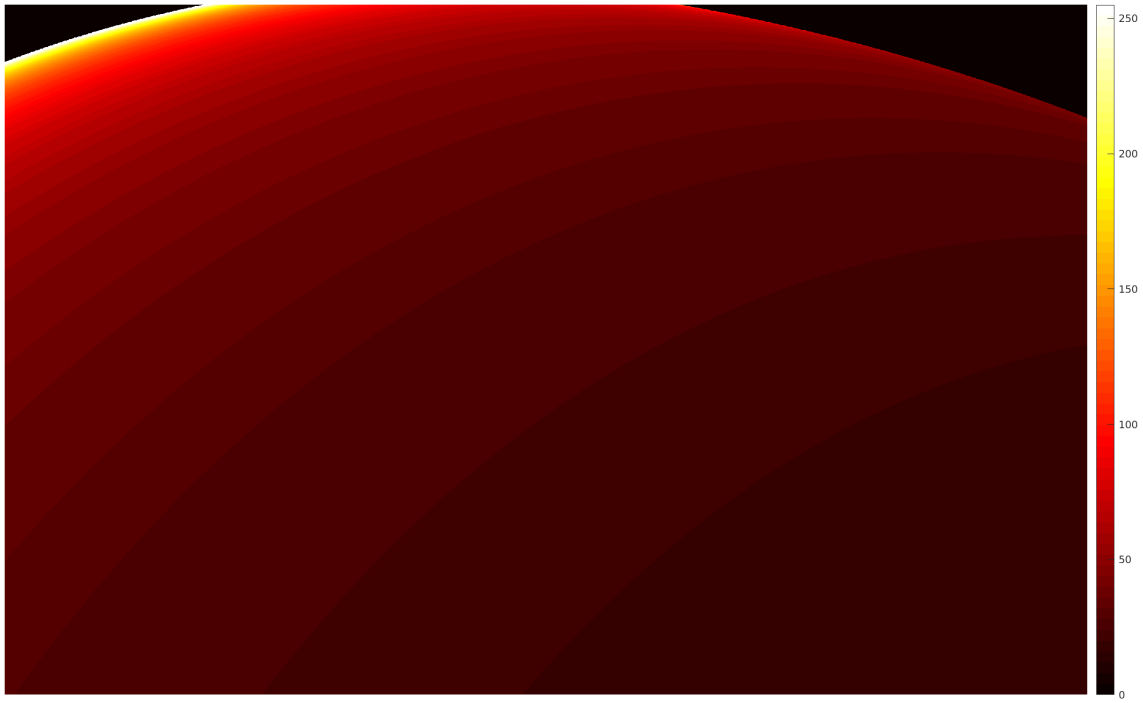


Figure 4.1: Relative displacement in number of pixels between the MSG3 and MSG1 views at 10km height

Since the area of interest of this thesis is the sky above Belgium, one can crop the MSG3-1 images within the area shown in Figure 4.2 and limit the effect that the relative displacement has on the stereo

matching results.

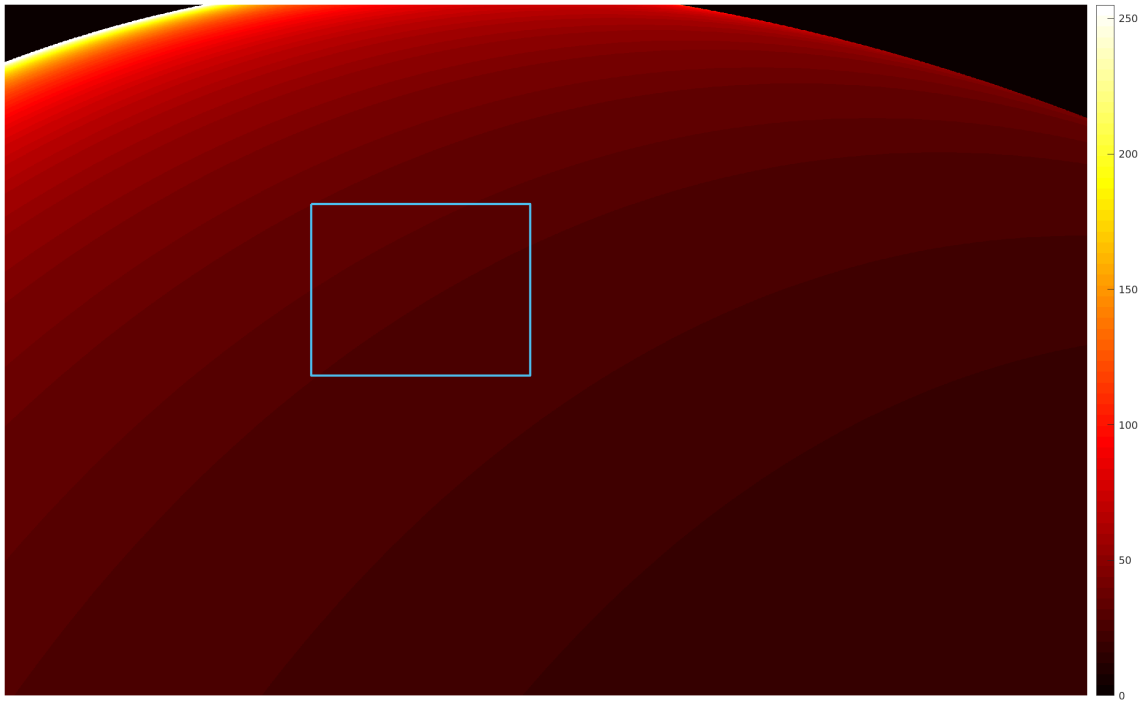
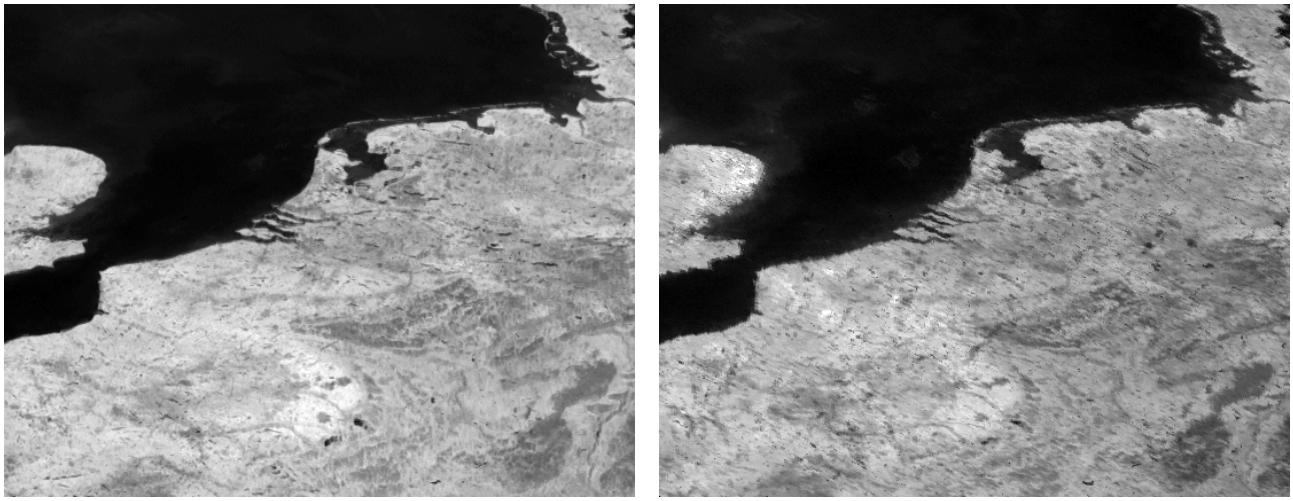


Figure 4.2: Relative displacement in number of pixels at the area of interest

In order to refine the rectification step that has already been applied to the initial stereo pairs, a registration step follows the cropping. However, the registration needs to be applied only on the clear sky images[1] (Figure 4.7), since the satellites are geostationary and their relative position with respect to earth remains constant. Once the registration is done and the transformation matrix has been computed, this transformation needs to be applied to every cropped stereo pair.



(a) (b)

Figure 4.3: Cropped normalized clear sky images of (a) MSG3 and (b) MSG1

[1]The clear sky images were produced by taking the minimum value of all pixels of the given dataset

4.1 Image Registration Theory

Image registration is a fundamental task in image processing and a classical problem encountered in many image processing applications where it is necessary to perform joint analysis of two or more images of the same scene which are acquired by different sensors, or images taken by the same sensor but at different times.

Given two images, $I1$ (defined as a reference image) and $I2$ (defined as a sensed image), the goal of image registration is to rectify the sensed image into the coordinate system of the reference image and to make corresponding coordinate points in the two images fit the same geographical location. These unregistered images may require one or more basic transformations, such as relative translation, rotation, scaling between them or shearing. These transformations will be explained in the following subsection.

Fundamentally one needs to establish the relationship between the distortions in the sensed image and the type of registration techniques which are most suitable. Two major types of distortions are generally distinguished. The first type are those which are the source of misregistration, i.e., they are the cause of the misalignment between the two images. To register two images is to remove the effects of the source of misregistration. Distortions which are the source of misregistration determine the transformation class which will optimally align the two images. The transformation class in turn influences the general technique that should be taken. The second type of distortion are those which are not the source of misregistration. This type usually effects intensity values but they may also be spatial. Distortions of this type are not to be removed by registration but they make registration more difficult since an exact match is no longer possible. A detailed evaluation of the various image registration techniques is presented by Brown [46].

In remote sensing applications, people for years generally used manual registration. The traditional procedure for manually registering a pair of satellite images requires the manual selection of control points in each image. These points are used to determine the parameters of a transformation function, which is subsequently used to register the sensed image to the reference one, by warping one of the images with respect to the other using any interpolation function. Automation of this procedure requires the replacement of the manual control point selection with automatic algorithms for locating corresponding points in both images [46].

The various registration methods available can be categorized into unimodal or multimodal according to the modality of the sensor, multitemporal if there is a time difference between each image capture, as well as between feature and intensity based methods [47]. Since the satellites have the same SEVIRI sensors, the registration that needs to be applied is unimodal. Furthermore, the time difference between the two images is close to zero, so multitemporal methods are not needed.

In the context of the present thesis, the objective of this computational step is to accurately register the cropped MSG-1 clear sky normalized image with the corresponding MSG-3 image. The MSG-3 image is defined as the **reference** image and the MSG-1 as the **sensed** image.

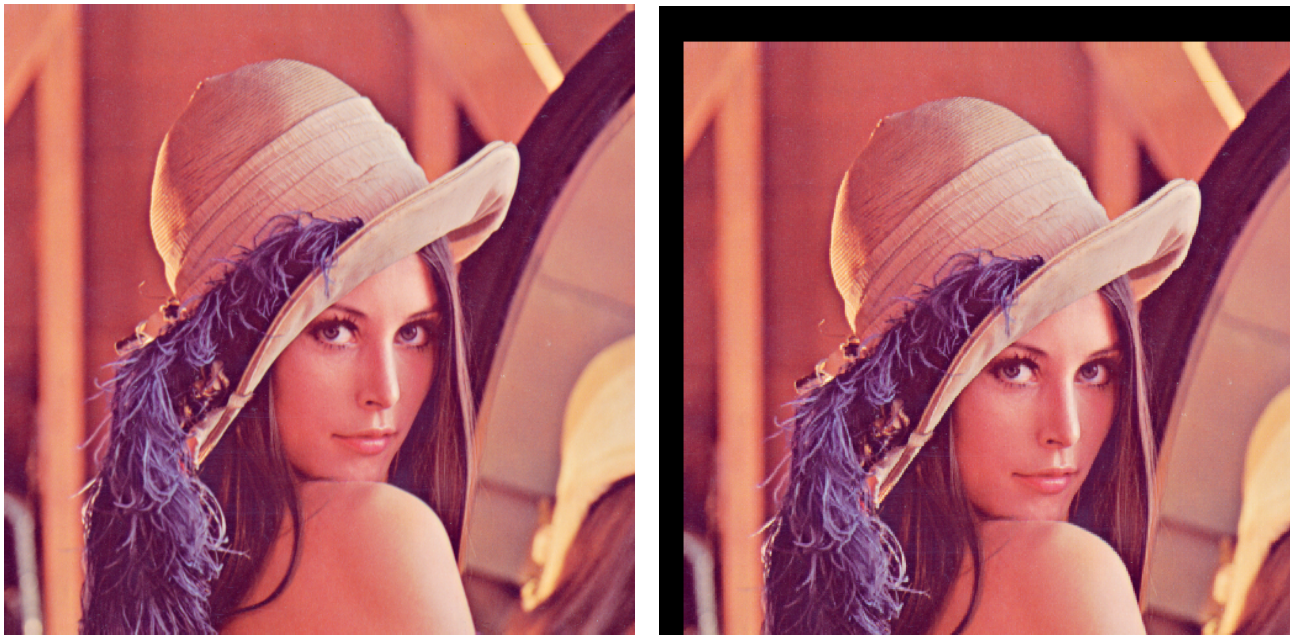
For solving this registration problem, we focused on the comparative evaluation of a feature-based rigid registration algorithm with a mutual information maximization one, following a detailed evaluation of previous reported research [46] [47] [48].

4.1.1 Basic Image Transformations

In order to register two images one needs to compute and apply a transformation to the sensed image. The transformations can be classified according to their degrees of freedom. The basic transformations which can be applied to an image $I(x, y)$ are translation, rotation, scaling and shearing.

Translation of δx and δy in each respective direction (2 DoF).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \tag{4.1}$$



(a) Original image of Lena (b) Translated image

Figure 4.4: Translation Transformation

Shearing of c_x, c_y at each respective direction (2 DoF).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & c_x \\ c_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.2}$$

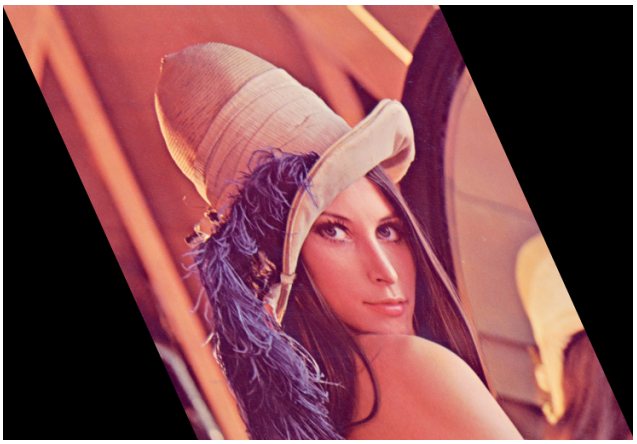


Figure 4.5: Shearing Transformation

Rotation of θ degrees anti-clockwise (1 DoF).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.3}$$

Figure 4.6: Rotational Transformation

Scaling of s_x, s_y at each respective direction (2 DoF).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.4}$$

Figure 4.7: Scaling Transformation

A rigid transformation is the one that includes only a translation and a rotation. The rigid transformation preserves the Euclidean distance between every pair of points. A more general transformation is the affine transformation which also includes scale and shear factors. Additionally, the affine transformation preserves points, straight lines and planes as well as the parallelism of lines.

4.1.2 Registration Methods

In this section we present the theoretical frameworks of some key registration methods that were reviewed and tested in the context of our work.

4.1.2.1 Mutual Information

The first registration method that was reviewed and tested is based on the Mutual Information metric, which is closely related to Shanon's Entropy and was introduced in [49] [50].

$$MI(A,B) = H(A) + H(B) - H(A,B) \tag{4.5}$$
$$= H(B) - H(B|A) \tag{4.6}$$

where $H(A)$ and $H(B)$ are the entropies of the random variables A and B respectively and $H(A,B)$ is their joint entropy. $H(B|A)$ is based on the conditional propability $p_{B|A}(b|a)$, which denotes the propability of a given pixel intensity b in image B the corresponding pixel in image A having intensity value a.

$$H(A) = -\sum_{a} p_A(a) log(p_A(a)) \tag{4.7}$$
$$H(A,B) = -\sum_{a,b} p_{A,B}(a,b) log(p_{A,B}(a,b)) \tag{4.8}$$
$$H(B|A) = -\sum_{b,a} p_{B|A}(b|a) log(p_{B|A}(b|a)) \tag{4.9}$$

If images A and B can be regarded as two random variables that have some measure of dependence between their pixels' intensities a and b respectively, then determining the MI global maximum leads to the images being fully registered and is interpreted as

> *'Maximizing mutual information will tend to find as much as possible of the complexity that is in the separate datasets (maximizing the first two terms) so that at the same time they explain each other well (minimizing the last term)'* [50].

The estimation of the joint and marginal distributions $p_{A,B}(a,b)$, $p_A(a)$ and $p_B(b)$ is crucial to image registration and can be achieved by the normalized joint and marginal histograms of the overlapping parts of both images. Although the Mutual Information is widely used in mutlimodal registration cases [51] [52], as it occures in many applications in medical imaging or remote sensing, it can be also used to register unimodal images because it takes into account only the statistical characteristics of the grayscale images and neglects the detailed features.

4.1.2.2 Low-Level Feature Based Registration

The second family of methods that was also implemented and tested is based on the low-level features of the images (points of interest such as corners, edges, etc.). In general, feature-based registration methods require the following steps to take place:

1. **Feature Detection** in both images (edges, corners, contours, etc.) is done either manually or automatically.
2. **Feature Matching**. Corresponding detected features between the two images are matched. Various feature descriptors and similarity measures along with spatial dependencies among them can be used.
3. **Transform Model Estimation**. Using the detected and described matched features, the transformation needed to align the sensed image with the reference image is estimated.
4. Finally, **image resampling** is applied in order to transform the sensed image, given the transformation estimated in step 3.

4.1.2.3 SIFT

The Scale-invariant feature transform (SIFT) was proposed by *Lowe* [53] in order to detect and describe local features in images. The algorithm can be divided into the following procedures:

1. **Scale-Space Extrema Detection**. Firstly, the algorithm creates a pyramid image representation at different scales, in order to identify potential interest points that are invariant to scale and orientation. It is implemented as a cascade filtering approach using a Gaussian function as the scale-space kernel. Specifically, the Difference of Gaussians (DoG) is used as an approximation of the Laplacian of Gaussians (LoG) filter.

2. **Keypoint Localization**. The next step fits a detailed model at each candidate location to determine location and scale. A Taylor series expansion of scale space is used to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value it is rejected. Because DoG has a higher response for edges, some of them must also be removed. A 2×2 Hessian filter is used to compute the principle curvature, and reject any that surpass a given threshold. Finally, keypoints are selected based on their measure of stability, rejecting any points that are poorly localized along an edge or have low contrast and therefore are sensitive to noise.

3. **Orientation Assignment**. Keypoints are assigned one or more orientations based on local image gradient directions.

4. **Keypoint Descriptor**. Previous steps provided the necessary information that ensures invariance to image location, scale and rotation. In the last step the descriptor vectors are computed for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations. A 16×16 neighbourhood around the keypoint is taken and divided into 16 sub-blocks of 4×4 size. For each sub-block, 8 bin orientation histogram is created, resulting ina a total of 128 bin values.

4.1.2.4 SURF

The Speeded up robust features (SURF) algorithm was proposed by *Bay et.al.* [54] as a faster alternative to SIFT. It is based on the same principles as SIFT, but details in each level are different.

1. **Interest Point Detection**. In contrast to SIFT, SURF uses a Box Filter to approximate LoG. One big advantage of such an approximation is that a convolution with box filter can easily be calculated with very little computing time with the help of integral images, where the value of a pixel (x, y) is the sum of all values in the rectangle defined by the origin and (x, y). Also, a blob detector based on the Hessian matrix is used to find points of interest.

2. **Orientation Assignment**. Haar wavelet responses are used in horizontal and vertical direction within a circular neighbourhood of size $6s$, where s is the scale at which the point was detected. Subsequently, adequate guassian weights are applied to the computed responses and plotted as points in a two-dimensional space. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60°. If rotation invariance is not required, then finding this orientation can be skipped, saving a lot of computational time.

3. **Keypoint Descriptor**. Finally, a square region of size $20s \times 20s$ aligned to the selected orientation is extracted. The region of interest is then split into smaller 4×4 square sub-regions, and for each one the Haar wavelet responses are computed at 5×5 regularly spaced sample points. Clearly, the size s has direct impact on both the computational complexity and the point-matching robustness/accuracy. While a small descriptor may be more robust against appearance variations, it may not offer sufficient discrimination and thus give too many false positives.

The SIFT/SURF detectors and descriptors are considered as the state of the art, compared to other feature based methods [55]. SURF is good at handling images with blurring and rotation, but fails when viewpoint and/or illumination changes are present. However, when speed is not crucial, SIFT seems to outperform SURF [56] [57].

4.1.2.5 Random Sample Consensus (RANSAC)

In order to estimate the transformation needed to achieve the required image registration, given the detected and described features as calculated from the previous algorithms, the *RANSAC* algorithm is used [58]. The main advantage of RANSAC with respect to other traditional fitting models (e.g. Least-Squares) is that it is robust to outliers. Therefore, even if some outliers exists in the data, RANSAC is able to estimate the correct model. It is important to state that RANSAC does not need an initial transformation model to be given, as in the Mutual Information based method which requires that the exact type of the transformation must be stated.
RANSAC algorithm is described below:

Algorithm 1: RANSAC

Data: Accepted Matched Features
Result: Transformation Model
Determine:
s - the smallest number of points required
N - the maximum number of iterations allowed
d - the threshold used to identify a point that fits well (chosen empirically)
T - the number of nearby points required to assert a model fits well

```
1  begin
2      repeat
3          Select a sample S from the data uniformly and randomly;
4          Fit the transformation model to that set S points (one can use least-square fitting
             method);
5          for For each data point not in the set do
6              if the distance of that point to the line is less than d then
7                  that point is close;
8              end
9          end
10         if there are at least T points close to the line then
11             the model is a good fit;
12             Continue refitting the model given these points;
13         end
14     until N iterations have occurred;
15     Use the best fit from this collection, using the fitting error as a criterion.;
16 end
```

One problem that arises is the amount of iterations, N, to choose. Given the following:

$$
\begin{aligned}
e &= \text{propability that a point is an outlier} \\
s &= \text{number of points in a sample} \\
N &= \text{number of samples} \\
p &= \text{desired propability of a good sample}
\end{aligned}
$$

then

$$1 - (1 - (1 - e)^s)^N = p \tag{4.10}$$

where

$$\begin{aligned} 1 - e &= \text{propability of a point being inlier} \\ (1 - e)^s &= \text{propability of a sample containing only inliers} \\ 1 - (1 - e)^s &= \text{propability of one or more points in the sample being outliers} \\ (1 - (1 - e)^s)^N &= \text{probability that N samples contain outliers} \\ 1 - (1 - (1 - e)^s)^N &= \text{propability that at least one sample of s points contains only inliers} \end{aligned}$$

Solving (4.10) for N gives

$$N = \frac{log(1 - p)}{log(1 - (1 - e)^s)} \tag{4.11}$$

We can early terminate the algorithm if inlier ratio reaches an expected ratio T [59].

$$T = (1 - e) * \text{total number of data points}$$

The probability of choosing an inlier each time a single point is selected

$$(1 - e)^s = \frac{\text{number of inliers in data}}{\text{number of points in data}}$$

is not usually known beforehand. However, a rough estimation value can be given.

In conclusion, RANSAC's ability to do a robust estimation of the model parameters despite the presence of some outliers is its main advantage. Another advantage is that it does not need a predifined transformation model, but instead estimates one without any previous representation knowledge of the data. However, it should be noted that if the inliers are less than 50% of the data, then RANSAC performs poorly. Also, when two ore more fitting models exist, RANSAC may fail at estimating either one.

4.1.2.6 Interpolation Method

As a final step, an interpolation technique is required in order to apply the transformation to the sensed image and calculate the new pixel values. In order to wrap the image given a transformation, one has to resample it. There are two ways of resampling an image.

- **Forward Approach** that calculates the pixels $(x^{'}, y^{'})$ of the registered image given the pixels (x, y) of the sensed image and the transformation T. One serious disadvantage of such an approach is the fact that the computed pixels $(x^{'}, y^{'})$ might not be integer values. Rounding to the nearest integer solves this problem, but causes unwanted artifacts on the registered image.
- **Backward Approach** that assumes the pixels $(x^{'}, y^{'})$ of the registered image as known and calculates the pixels (x, y) of the sensed image given the inverse transformation T^{-1}. The calculated pixels (x, y) of the sensed image might also not be integer values as well. However, one can calculate the values of (x, y) using an interpolation technique, since the actual pixels (x, y) values are known, and thus calculate the pixels $(x^{'}, y^{'})$.

The interpolation technique we used is bilinear interpolation, a combination of the two aforementioned methods. In this method, the pixel value is the linear distance-weighted average of the four closest pixel values.

4.2 Implementation of the proposed registration methods

In this section the technical details of the implementation of the previously described registration methods are presented. The results obtained are also presented and discussed in detail.

4.2.1 Mutual Information based algorithm

The *ITK* and *Simple ITK* libraries were used in order to implement the Mutual Information based registration algorithm [60] [61] [62] [63] [64] [65].
The specific steps performed are:

Initial Alignment - An initial transformation is needed in order to estimate the best solution. Since the MSG images were rectified, the affine transformation needed would be very close to the initial transformation if that would be the one that superimposes the centers of the two images.

Interpolation - Image transformations require an interpolation technique for the estimation of the intensity gray value of the resulting point. In the current study, the Nearest Neighbor Interpolation (assigning the gray value of the spatially closest neighbor) is used.

Optimization - The registration measure defines an n-dimensional function, with n the degrees of freedom of the expected transformation. Finding the transformation that correctly registers the two images means finding the global maximum of that function. Due to many local maxima being present near the initial position, a proper optimization method is needed in order to accurately estimate the global maximum. The only method that deterministically finds the global maximum is exhaustive search. However, it is computationally very demanding and prohibitive in complex scenarios. If the expected transformation consists only of translations, then exhaustive search can be used [47]. Thus, other more sophisticated optimization techniques are required, if the expected transformation has more degrees of freedom.

The following two optimization functions are used to maximize the Mutual Information:

- **Gradient Descent/Ascent Optimizer**, as used in [50].
- **Powell's Optimizer**, as used in [51].

Powell's method simplifies multidimensional problems to one dimensional optimization task and the clear advantage over Gradient Descent is that it does not require the computation of the gradient which is computationally expensive [51] [66]. One can inverse the MI measure if an optimizer estimates the global minimum in order for it to estimate the global maximum.

The optimization techniques can be speeded up using pyramid or multiresolution analysis, which are multi-scale signal representations, where the images are subjected to smoothing and subsampling before processing is applied at each level. However, methods based on pyramid analysis will not be decomposed and tested in depth as the experimental results show poor performance on our data.

4.2.2 Implementations of SIFT/SURF algorithms

The implementations of the *SIFT* and *SURF* algorithms of the *OpenCV* library [67] were used.

Although both algorithms are very accurate, some features still may not have been correctly matched. In order to efficiently reject the outliers, a measure obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor is applied as described in [53]. This method of rejecting false matches performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. There will likely be a number of other false matches within similar distance of a possible false match, due to the high dimensionality of the feature space. Such measure can be formulated as :

Algorithm 2: Rejecting outliers SIFT/SURF

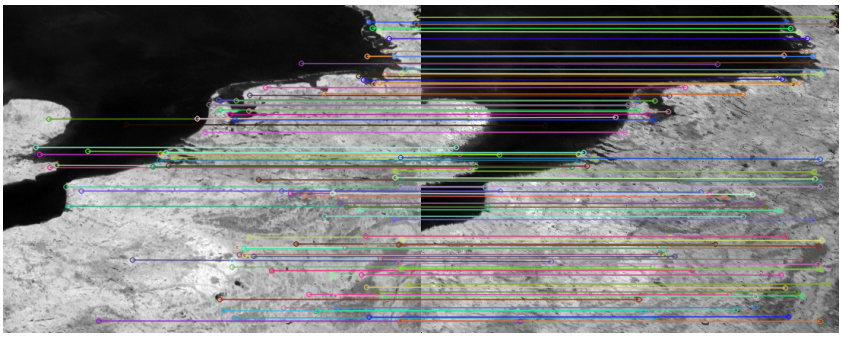
Data: All Matched Features
Result: Features accepted as correctly matched given a rejecting $ratio$

```
begin
    if closestNeighbor.distance < ratio * secondClosestNeighbor.distance then
        Consider as correct match;
    else
        Consider as false match;
    end
end
```

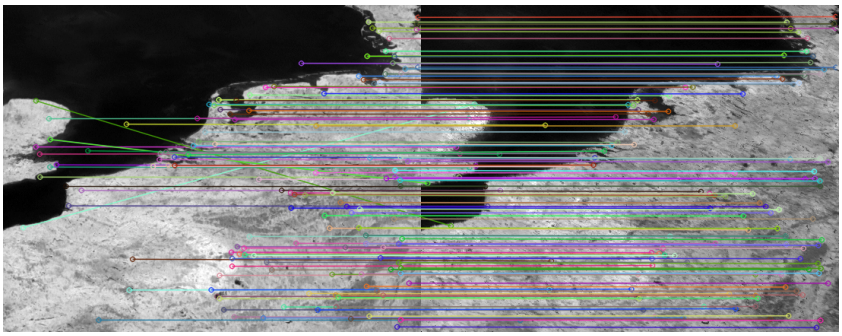
where $ratio$ is proposed in [53] to be set at 0.8, which results in rejecting 90% of the outliers and only 5% of the inliers (correct matches) for an object recognition problem. For our study, we tested the following ratios $\left[0.6, 0.65, 0.7, 0.8, 0.9\right]$. In Figures 4.8 and 4.9, the feature matching (of the clear sky images) results of the SIFT and SURF respectively for different outliers rejecting ratios are given.

Since the transformation we need to estimate is not a complex one, a few matches are sufficient. It is clear that the optimal ratio is 0.65 as no outliers are present in this case, in contrast to the other ratio values where some ($ratio = 0.7$) or many ($ratio = 0.9$) outliers exist, while still having enough matches to estimate an affine or perspective transformation, for which 6 or 9 parameters must be estimated, respectively.

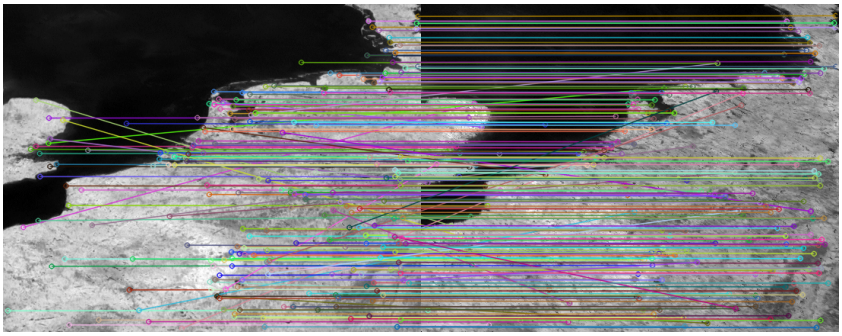
RANSAC algorithm's implementation of *OpenCV* is subsequently applied to estimate the transformation needed to register the two images based on the extracted features.



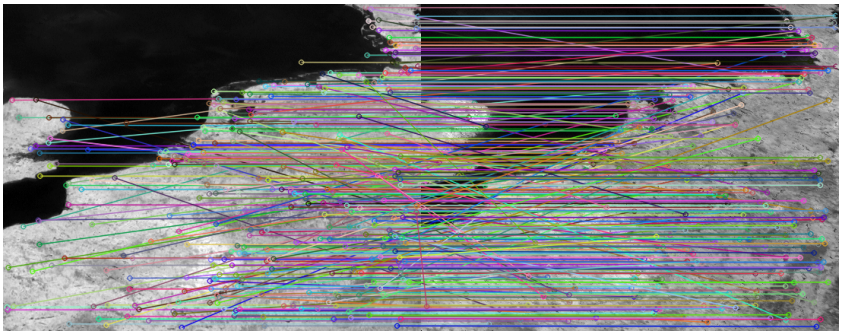
(a) $ratio = 0.65$



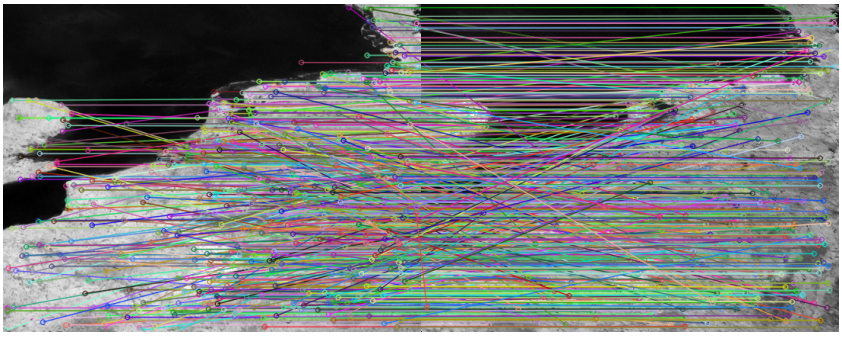
(b) $ratio = 0.7$



(c) $ratio = 0.8$

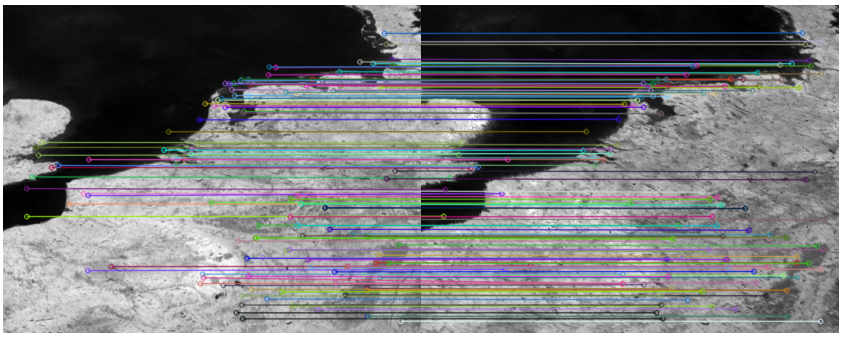


(d) $ratio = 0.85$

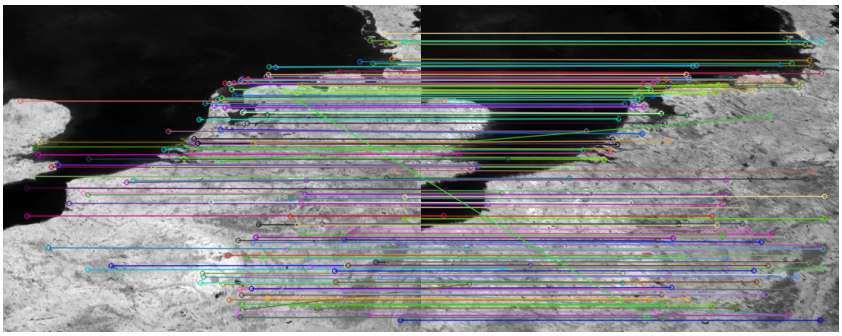


(e) $ratio = 0.9$

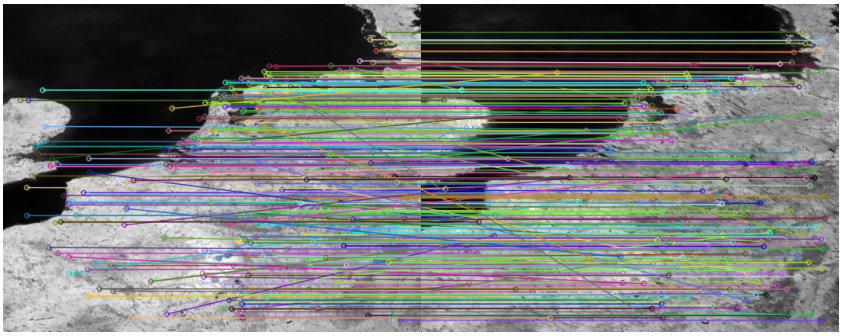
Figure 4.8: Visualization of the feature matching method using SIFT at each outliers' rejecting ratio.



(a) $ratio = 0.65$

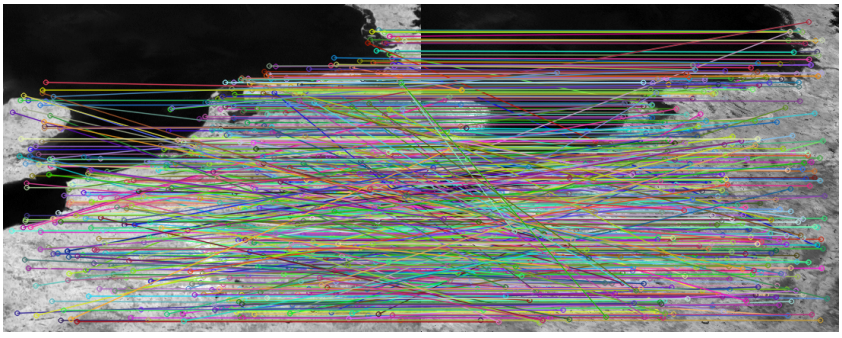


(b) $ratio = 0.7$



(c) $ratio = 0.8$

(d) $ratio = 0.85$



(e) $ratio = 0.9$

Figure 4.9: Visualization of the feature matching method using SURF at each outliers' rejecting ratio.

4.2.3 Validation of the Registration Results

In order to assess the accuracy or precision of the registration process, we will employ various metrics. Specifically, the registered image will be compared with the target image by their absolute relative **difference** as well as with the following metrics:

- **SSIM** which stands for the Structural Similarity Metric defined in [68]. The $SSIM$ is maximum when the two images are identical.
- **PSNR** which stands for the Peak Signal to Noise Ratio. The $PSNR$ is ∞ when the two images are identical.

The $SSIM$ is defined as

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.12}$$

where

- $\mu_{x,y}$ is the average of x, y
- $\sigma_{x,y}$ is the variance of x, y
- σ_{xy} is the covariance of x, y
- $c1 = (k1L)^2$, $c2 = (k2L)^2$ two variables to stabilize the division with weak denominator
- L is the dynamic range of the pixel-values ($2^{bits\_per\_pixel-1}$)
- $k1 = 0.01$ and $k2 = 0.03$ by default

The components of the previous formula are the luminance (l), contrast (c), structure (s) [69].

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{4.13}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{4.14}$$

$$s(x,y) = \frac{2\sigma xy + c_3}{\sigma_x\sigma_y + c_3} \tag{4.15}$$

where $c_3 = c_2/2$ and the $SSIM$ is a weighted product of these components (In this case $\alpha = \beta = \gamma = 1$)

$$SSIM(x,y) = \left[l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma\right] \tag{4.16}$$

and the $PSNR$ is defined as

$$PSNR = 10log_{10}(\frac{MAX \cdot 2}{MSE}) \tag{4.17}$$

$$MSE = \frac{1}{m * n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{4.18}$$

where I and K are the image matrices. MAX is equal to the maximum fluctuation in the input image data type. For example, for an 8-bit unsigned integer data type, MAX is 255.

4.2.4 Image Registration Results

Table 4.1 presents the best results obtained by each registration method implemented and tested. In the case of SIFT/SURF based algorithms, the term Attempts refers to different values for the outlier rejection ratio, i.e. 0.65, 0.7, 0.8, 0.85, 0.9.

Table 4.1: Registration results of MSG1 and MSG3 images. See the corresponding figures in Appendix A for the visual difference of the registered MSG1 and the MSG3 images.

Method	Attempts	SSIM	PSNR (db)
Mutual Information / Gradient's Descent Optimizer	1	0.5530	29.1258
	2	0.5827	29.1053
	3	0.5538	29.1046
	4	0.5920	29.0983
	5	0.6030	29.0875
	6	0.5698	29.1224
Mutual Information / Powell's Optimizer	1	0.6015	29.0747
	2	0.6369	29.0832
	3	0.5955	29.0948
	4	0.6099	29.0907
	5	0.5850	29.1131
	6	0.6182	29.0814
SURF / RANSAC	1	0.6041	29.1360
	2	0.5906	29.1205
	3	0.5962	29.1340
	4	0.5977	29.1475
	5	0.5955	29.1292
SIFT / RANSAC	1	0.5835	29.1575
	2	0.5958	29.1610
	3	0.5992	29.1602
	4	0.6040	29.1444
	5	0.5995	29.1559

The results in Table 4.1 are inconclusive as all four registration methods produce similar results. Therefore, in order to conclude to one main algorithmic approach, the previous methods were tested using some artificially transformed MSG-1 clear sky images, given four different transformation cases. The transformations tested were :

1. Translation of (5,5) pixels and Rotation of 0.5 degrees.

2. Translation of (3,3) pixels, Rotation of 0.5 degrees and Scale of factors (x=1.01, y=1.01) for each dimension.

3. Translation of (3,3) pixels, Shear (x=0.005, y=0.005), Scale of (x=1.01, y=1.01).

4. Translation of (5,5) pixels.

The results of registering artificially generated transformed MSG-1 images are given in Tables 4.2-4.5

Table 4.2: Registration Results of Artificially Transformed MSG-1 Images using Mutual Information - Gradient Descent Optimizer

MI and Gradient's Descent Optimizer	Attempts	SSIM	PSNR (dB)
Transformation 1	1	0.4919	30.2468
	2	0.4878	30.1699
	3	0.4930	30.2461
	4	**0.5866**	**30.4939**
	5	0.4811	30.1500
	6	0.5206	30.3084
Transformation 2	1	0.4528	30.0916
	2	0.5269	30.4158
	3	**0.6123**	**30.6326**
	4	0.5562	30.4235
	5	0.5111	30.3068
	6	0.4277	29.9963
Transformation 3	**1**	**0.6415**	**30.8565**
	2	0.5418	30.4115
	3	0.6369	30.7787
	4	0.5370	30.3702
	5	0.5398	30.4055
	6	0.5216	30.3090
Transformation 4	1	0.3226	29.6495
	2	0.3280	29.6328
	3	**0.7769**	**31.6938**
	4	0.5921	30.6465
	5	0.6491	30.7543
	6	0.4687	30.1220

Table 4.3: Registration Results of Artificially Transformed MSG-1 Images using Mutual Information - Powell Optimizer

MI and Powells's Optimizer	Attempts	SSIM	PSNR (dB)
Transformation 1	1	0.3502	29.6911
	2	**0.9538**	**35.7722**
	3	0.5540	30.4824
	4	0.9504	35.7646
	5	0.5635	30.5201
	6	0.5787	30.6074
Transformation 2	**1**	**0.6394**	**31.0003**
	2	0.5476	30.4770
	3	0.5198	30.3556
	4	0.5097	30.3111
	5	0.5298	30.4006
	6	0.5311	30.4095
Transformation 3	1	0.5785	30.7094
	2	0.5763	30.6106
	3	0.5886	30.6680
	4	**0.9599**	**36.1812**
	5	0.5841	30.6974
	6	0.5627	30.5784
Transformation 4	1	0.5201	30.3167
	2	0.5815	30.5118
	3	0.4649	30.1587
	4	0.3507	29.8497
	5	**0.6861**	**31.2856**
	6	0.3212	29.5474

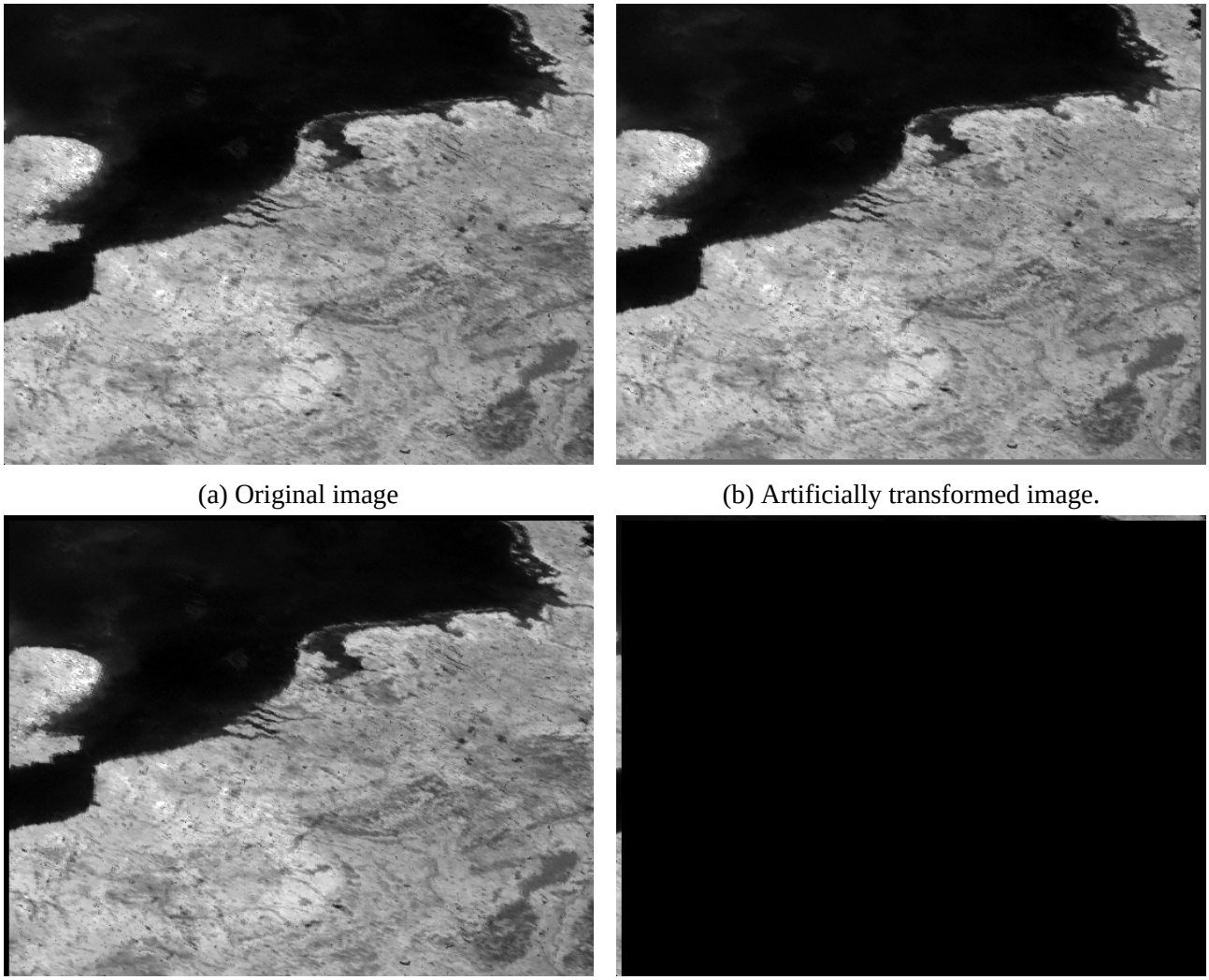
Table 4.4: Registration Results of Artificially Transformed MSG-1 Images using SIFT - RANSAC

SIFT / RANSAC		
Trasnformation	SSIM	PSNR (dB)
1	0.9254	34.0741
2	0.9328	34.3637
3	0.9364	34.4177
4	0.9872	44.6859

Table 4.5: Registration Results of Artificially Transformed MSG-1 Images using SURF - RANSAC

SURF / RANSAC		
Trasnformation	SSIM	PSNR (dB)
1	0.9247	34.0575
2	0.9321	34.3468
3	0.9361	34.4144
4	0.9872	44.6859

Observing the results presented in Tables 4.2, 4.3, 4.4 and 4.5, it becomes evident that the superior performance of feature-based registration methods regarding the registration of the artificially produced images is clear. Moreover, in the case of the last Translation if we exclude the left and top margins of 5 pixels, the two images (original MSG-1 and the registered artificially transformed MSG-1 image) are identical ($SSIM = 1.0$ and $PSNR = \infty$), as it can be seen in Figure 4.10. That is essentially the proof that the feature based registration methods perform better than the Mutual Information based methods, given the assumption that the translation needed is a simple translation. Even for more complex transformations (such that in the cases of the first 3 transformations), the feature based methods outperform the MI.



(a) Original image

(b) Artificially transformed image.

(c) Registered image

(d) Difference between original and registered images.

Figure 4.10: The registration result using SIFT + RANSAC, with a rejecting outliers ratio=0.65, for the case of artificially translated MSG-1 image (5,5) pixels.

In conclusion, the preferred registration method that is used to estimate the transformation needed to register the MSG-3 and MSG-1 images is based on low-level features (either SIFT or SURF) combined with the RANSAC algorithm. This estimated transformation model was subsequently used to register all MSG-3 and MSG-1 cloudy image pairs.

5. Disparity Estimation using Markov Random Fields

This chapter focuses on solving the stereo matching problem by utilizing the theory of Markov Random Fields (MRF). In addition, two Graph Cut based algorithms, with which we can solve the correspondence problem in the context of MRF, are presented. Finally, the results are presented and discussed.

Stereo vision as well as many other image analysis problems (Segmentation, Restoration, Reconstruction, Recognition, etc.) can be understood as labeling problems, in the sense that the solution is a set of labels assigned to the image pixels. In the case of stereo vision this information is the disparity value of each pixel. The theory of Markov Random Fields provides a powerful context in which such problems can be formulated. A MRF or undirected graphical model is a random field consisting of some random variables having Markov properties. A MRF has the capability of encoding spatial dependencies and constraints (such as smoothness or depth continuity) and as a result proves to be very helpful in many image analysis problems [70] [43] [71].

5.1 Labelling Problem

Let L be a set of discrete labels and S a set of discrete sites, then the labelling problem is specified as the problem of mapping these labels to each site. In the field of image analysis, sites can be regarded as the image pixels or features and the labels as a property that these sites possess.

Image analysis using contextual information is necessary to most vision problems. In probability terms, these constraints can be expressed locally as conditional probabilities $P(f_i|f_{i'})$, where f denotes the set of labels and $i^{'} \neq i$, or globally as the joint probability Pf. If the labels are independent of each other then

$$P(f) = \prod_{i \in S} P(f_i) \tag{5.1}$$

which implies

$$P(f_i|f_{i'}) = P(f_i) \qquad i^{'} \neq i \tag{5.2}$$

However, in most cases of contextual information labels are mutually dependent and the previous presented useful relationship does not hold. The theory of Markov Random Field provides a mathematical foundation for solving this problem.

5.2 Neighborhood System

MRF theory is a branch of probability theory for analyzing the spatial or contextual dependencies of some phenomena. Given a set of sites S, they are related to one another via a neighborhood system which is defined as

$$N = \{N_i | \forall i \in S\} \text{ where } N_i \text{ are the sites neighboring } i. \tag{5.3}$$

and has the following properties :

1. A site is not a neighbor to its self : $i \notin N_i$
2. The neighborhood relationship is mutual : $i \in N_{i'} \iff i' N_i$

The set of neighbors for a regular site S is defined as those sites within a radius of $\sqrt{r}$ from i

$$N_i = \{i' \in S \quad | \quad [distance(p_{i'}, p_i)]^2 \leq r, \quad i' \neq i\} \tag{5.4}$$

A clique c for a set S and a neighborhood system N is defined as subset of sites in S, which can include a single pixel i or a combination of neighboring pixels. In this thesis the standard 4-connected neighborhood system is assumed as in Figure 5.1.

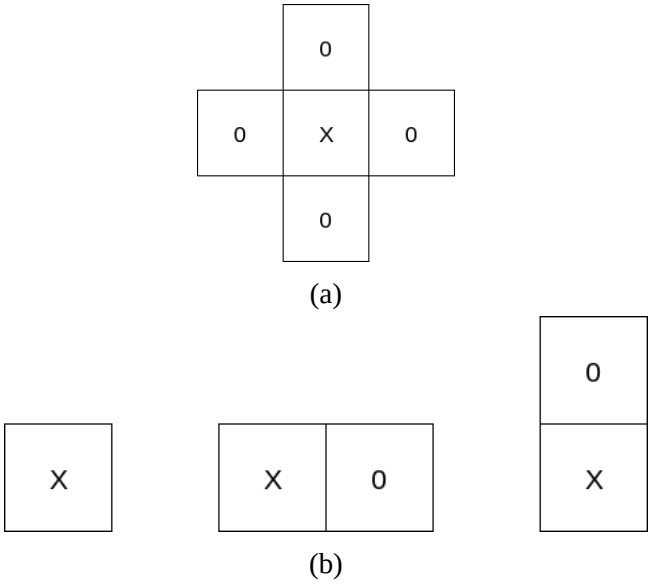


Figure 5.1: (a) 4-connected neighborhood system and (b) its cliques

5.3 Markov Random Fields

Given a set S, a family $F = \{F_1, F_2, \cdots, F_m\}$ of random variables defined on S is called a random field. The probability $P(F_i = f_i)$ is the probability that F_i takes a value f_i defined in L and the probability $P(F = f) = P(F_1 = f_1, \cdots, F_m = f_m)$ is the joint probability.

F is considered to be a Markov Random Field on S with respect to its neighborhood system N, if the joint probability P(F=f) satisfy the following two conditions :

1. $P(f) > 0, \forall f \in F$ (Positivity)
2. $P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$ (Markovianity)

This essentially means that the value F_i depends only on i pixel's neighbors. As stated in [43] Markov Random Fields are equivalent to Gibbs Random Fields, which provide a simple way of specifying the joint probability :

$$P(f) = Z^{-1} * e^{-\frac{1}{T}U(f)} \tag{5.5}$$

$$Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)} \tag{5.6}$$

where Z is the partition function and has a normalizing constant role, the temperature T is controlling the sharpness of the distribution and should be 1 (unless stated otherwise and $U(f)$ is the energy function.

The partition function is the sum over all possible configurations in F for a discrete L. For an 8-bit image with dimensions $x \times y$ the number of terms is $2^{8*x*y} = 256^{x*y}$, which is a prohibitive number for computing the partition function.

Thus, an approximation is needed to estimate the probability of occurrence of a particular configuration f, $P(f)$. One has to estimate the configuration f based on an observation d which is related to f by means of the likelihood function $P(d|f)$. The most popular way to estimate an MRF is Maximum a Posteriori (MAP) estimation. MAP estimation consists of maximizing the posterior probability $P(f|d)$ which is written as :

$$P(f|d) = \frac{P(d|f)P(f)}{P(d)} \tag{5.7}$$

The MAP estimate f^* is equal to

$$f^* = \underset{\text{f} \in \text{F}}{\text{argmax}} P(d|f)P(f) \tag{5.8}$$

Based on the following assumptions of equations (5.9) and (5.10)

$$P(d|f) = \prod_{p \in S} P(d_p|f_p) \tag{5.9}$$

which holds, for example, if the noise at each pixel is independent.

$$P(d|f) \approx e^{-\sum_{p \in S} d_p(f_p)} \tag{5.10}$$

We get:

$$f^* = \underset{\text{f} \in \text{F}}{\text{argmax}} e^{-E(f)} \tag{5.11}$$

where $E(f)$ is the energy function as previously defined. Thus one has to minimize the energy $E(f)$ to maximize the MAP of some configurations [43] [70] [72].

5.4 MRF in Stereo Vision

An MRF is an undirected graphical model, as illustrated in Figure 5.2, that encodes information about some spatial relationship that characterizes the represented phenomenon. In the context of a stereo vision system, the observable nodes correspond to the actual pixel's p intensities of the captured image and the hidden nodes correspond to the disparity value (f_p) of each pixel's location.

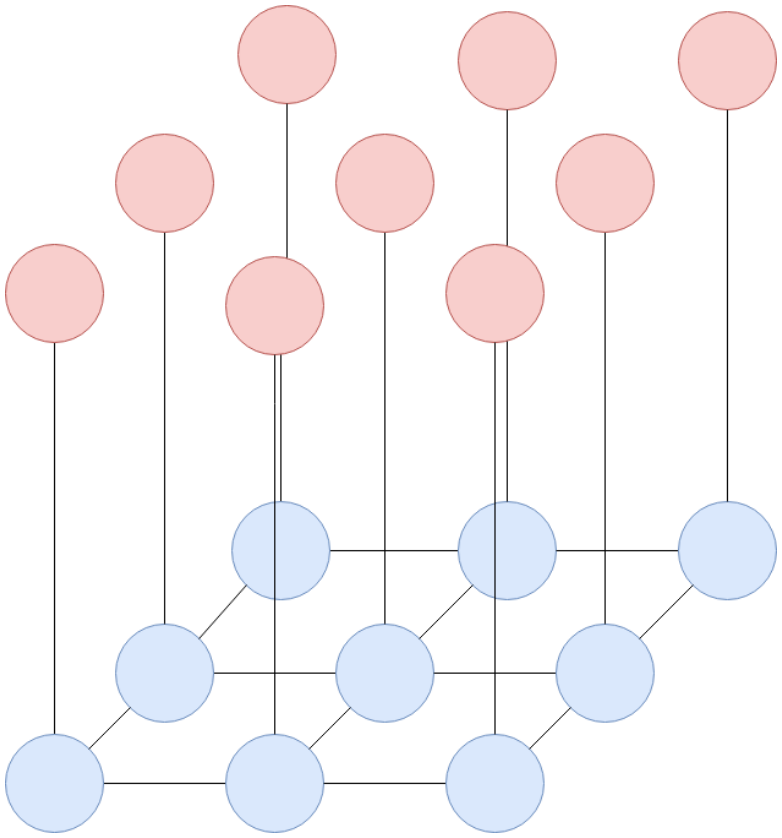


Figure 5.2: Markov Random Field example. Red nodes : Observable nodes corresponding to image's pixel intensities. Blue nodes : Hidden nodes corresponding to disparity values of each pixel's location.

5.4.1 Energy Model

The energy model that is assumed is:

$$E = E_d + \lambda E_s \tag{5.12}$$

λ is a constant controlling the relative importance of each energy term.

$$E_d = \sum_p d_p(f_p) \tag{5.13}$$

where d_p is the data cost of a particular pixel with respect ot its assigned label f_p.

$$E_s = \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q) \tag{5.14}$$

where N is the set of all pairs of neighboring pixels p and q and V_{pq} is the smoothness cost of such a pair.

The data energy E_d is responsible for encoding the data constraints or the matching cost of assigning a label to a pixel's location. It is calculated by the sum of a set of per-pixel data costs $d_p(f)$. Some choices for the data energy functions are absolute differences and squared differences.

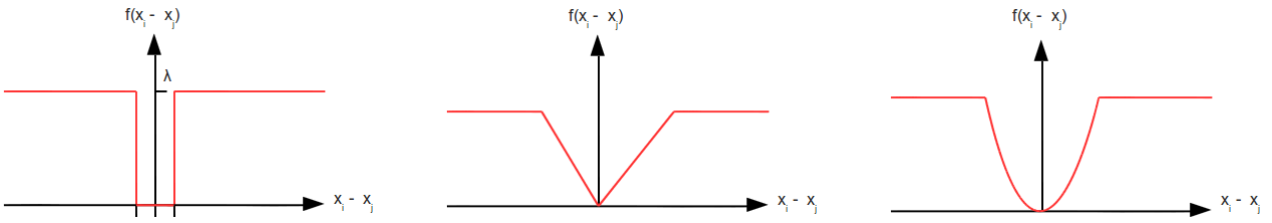
On the other hand, the smoothness energy E_s encodes the prior knowledge of the smoothness constraint that the disparity map must obey, as it was discussed in 3.4.5, calculated by the sum of each nearest neighbor smoothness cost. For a 4-connected neighbor system we take into account only the vertically and horizontally previous and next pixels. The smoothness constraint is suitable for problem cases where the quantity to be estimated varies smoothly almost everywhere, such as many vision problems where the only discontinuity is found near object boundaries. V_{pq} is called the

neighbor interaction function and it essentially applies penalties to neighboring pixels with different labels. Thus the smoothness energy E_s is the sum of neighbor interaction functions of all neighbor pairs. Three models that are widely used in vision problems to assign smoothness penalties are (see also Figure 5.3):

1. **Potts Model** - Two neighboring pixels p and q are assigned a constant penalty λ if they have different labels or no penalty otherwise.

2. **Truncated Linear Model** - The penalty is $min(\lambda \times min(|f_p - f_q|, T))$ between pixels p and q with a maximum value of T.

3. **Truncated Quadratic Model** - The penalty is $min(\lambda \times min((f_p - f_q)^2, T))$ between pixels p and q with a maximum value of T.



(a) Potts Model (b) Truncated Linear Model (c) Truncated Quadratic Model

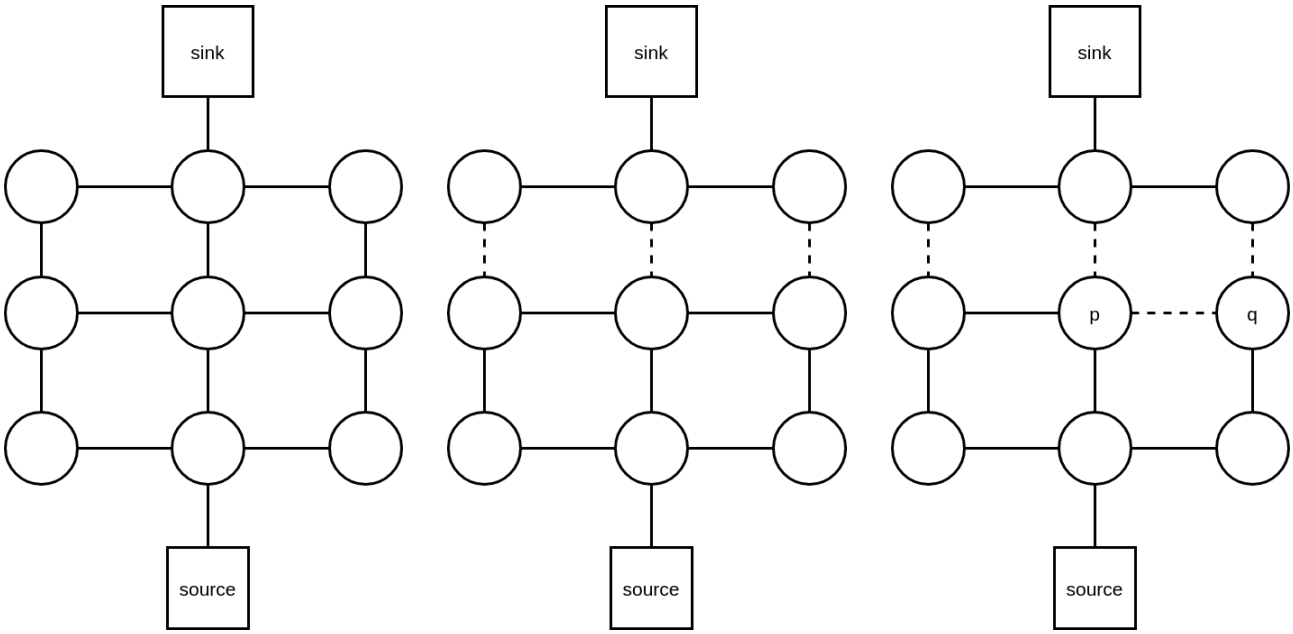
Figure 5.3: Neighbor Interaction Functions

5.5 Graph Cut based Algorithms

The minimization techniques that will be tested in this thesis are based on graph cuts. In this section the graph cut theory as well as some algorithms for graph cutting are presented.

5.5.1 Graph Cut

Let $G = \langle V, E \rangle$ be a weighted graph with a set of vertices V and a set of edges E. There exist two discrete vertices in V which are called terminals (square vertices in Figure 5.4) and they are respectively called *source* and *sink*. A set of edges $C \subset E$ which separates the two terminals in the graph $G' = \langle V, E - C \rangle$ is called a cut (Figure 5.4b). Additionally, there does not exist any subset of C that separates the terminals in G' (Figure 5.4c, where if the dashed edge between p and q is removed then the remaining dashed edges still form a cut).



(a) Original graph (b) Dashed edges form a cut. (c) Dashed edges do not form a cut.

Figure 5.4: Graph Cut example - Square vertices are the terminals, circle vertices are the remaining vertices in V and dashed edges are in cut.

The cut C is characterized by its cost $|C|$, which is the sum of its edges' weights. The goal of the following algorithms is to find the minimum cut, or in other words the cut with the smallest cost. A very efficient way of finding the minimum cut is in the formulation of finding the maximum flow between the terminals according to a theorem presented in [73]. The graph cut approach is able to find the global minimum for a binary labelling problem within linear time. On the other hand, given a multiple labelling problem, finding the global minimum is NP-hard, but efficient approximation algorithms exists which computes the local minimum within a known factor of the optimal [34].

5.5.2 Swap Move Algorithm

Finding the global minimum is proven to be NP-hard and thus finding a local minimum is necessary. One way to find the local minimum in discrete settings is through the concept of move spaces. The two most popular graph-cut algorithms are the *swap move* and *expansion move* algorithms, which were introduced in [34].

Given a pair of labels α and β, a move from one partition P with labeling f to a new partition $P^{'}$ with labeling $f^{'}$ is called a $\alpha - \beta$ swap if $P_l = P_l^{'}$ for any label $l \neq \alpha, \beta$. This essentially means that some pixels labeled α and β in one partition, they are now labeled β and α respectively in the other partition. The corresponding steps for this algorithm are as follows:

1. Start with an arbitrary labeling f.
2. Cycle through every label pair $(\alpha, \beta) \subset F$.
   (a) Find $f^{'} = argminE(f^{'})$ among $f^{'}$ within a single $\alpha - \beta$ swap of f.
   (b) Keep the current f or the new labeling $f^{'}$ according to which has the lowest energy E.
3. If the energy E remained the same for a single repetition then stop, otherwise go to step 2.

Theorem [71] [34] : *There is a one to one correlation between cuts C on graph G and labelings that are one $\alpha - \beta$ swap from f. Moreover, the cost of a cut C on G is $|C| = E(f^C)$ plus a constant.*

Corollary [71] [34] : *The optimal $\alpha - \beta$ swap from f is $f' = f^C$ where C is the minimum cut on G.*

5.5.3 α-Expansion Algorithm

Given a label α, a move from one partition P with labeling f to a new partition P' with labeling f' is called α-expansion if $P_a \subset P'_a$ and $P'_l \subset P_l$ for any label $l \neq \alpha$. This essentially means that any set of pixels is allowed to change their labels to α. The corresponding steps for this algorithm are as follows:

1. Start with an arbitrary labeling f.
2. For each label $\alpha \in F$.
   (a) Find $f' = argminE(f')$ among f' within a single α-expansion of f.
   (b) Keep the current f or the new labeling f' according to which has the lowest energy E.
3. If energy E remained the same for a single repetition stop, otherwise go to step 2.

Theorem [71] [34] : *There is a one to one correlation between elementary cuts C on graph G and labelings within one α-expansion of f. Moreover, the cost of a cut C on G is $|C| = E(f^C)$.*

Corollary [71] [34] : *The lowest energy labeling within a single α-expansion move from f is $f' = f^C$ where C is the minimum cut on G.*

Finding the local minimum is not a trivial task as there are an exponential number of swap or expansion moves in a labelling set f. However, checking for local minimum only when standard moves are allowed (one that allows only one pixel to change its intensity at a time) is easy since there are only a linear number of standard moves [34] in f. For more information regarding the $\alpha - \beta$ swap and α-expansion move algorithms please carefully read [34]. In this thesis the implementation of [72] of the previous algorithms was used.

5.6 Application of the algorithms

For solving the stereo matching problem we have employed the algorithms reported in [72]. The software also uses some energy minimization algorithms explained in [74] [75]. There are three main library components :

1. MRF Energy Minimization Software.
2. ImageLib which is a simple image library for reading/writing images in “png/pgm/ppm” formats.
3. MRFStereo which is a stereo matcher front-end to MRF Library.

The software libraries can be found in http://vision.middlebury.edu/MRF/code/. The software was compiled and run with gcc8 under Linux. After compiling the software, we can use the software by running:

```
$ ./mrfstereo [options] imageLeft imageRight disparityLeft
```

where options are :

- -n disparity levels which are by default 16 (0…15)
- -b use Birchfield/Tomasi costs [76]
- -s use squared differences (absolute differences by default)
- -t truncate differences to <= 'trunc'
- -a 0-ICM, 1-Expansion (default), 2-Swap, 3-TRWS, 4-BPS, 5-BPM, 9-all (we tested 1 and 2)
- -e smoothness exponent, 1 (default) or 2, i.e. L1 or L2 norm
- -m maximum value of smoothness term (2 by default)
- -l weight of smoothness term (20 by default)
- -g intensity gradient cue threshold
- -p if grad < gradThresh, multiply smoothness (2 by default)
- -o scale factor for disparities (full range by default)
- -w write parameter settings to dispL.txt
- -x write timings to dispL.csv
- -q quiet (turn off debugging output)

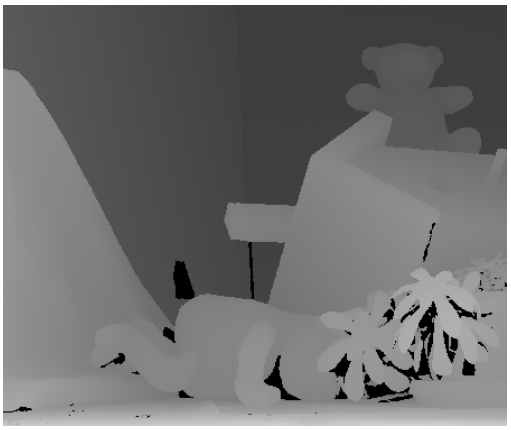
5.7 Benchmarks of the algorithms

In order to test the software we run the benchmarks that are posted in http://vision.middlebury.edu/MRF/results/.

5.7.1 Benchmarking Teddy Dataset

(a) Left Image

(b) Left Image



(c) Ground Truth Disparity

(d) Parameters values: -a 1 -b -n 60 -o 4 -t 16 -g 10 -p 3 -e 1 -m 1 -l 10

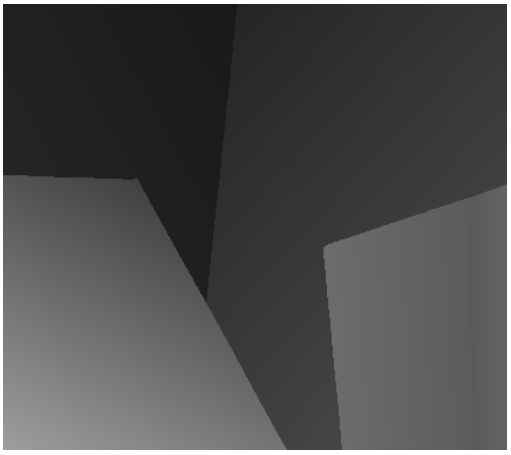
(e) Parameters values: -a 2 -b -n 60 -o 4 -t 16 -g 10 -p 3 -e 1 -m 1 -l 10

Figure 5.5: (d) Expansion Algorithm Benchmark (e) Swap Move Algorithm Benchmark

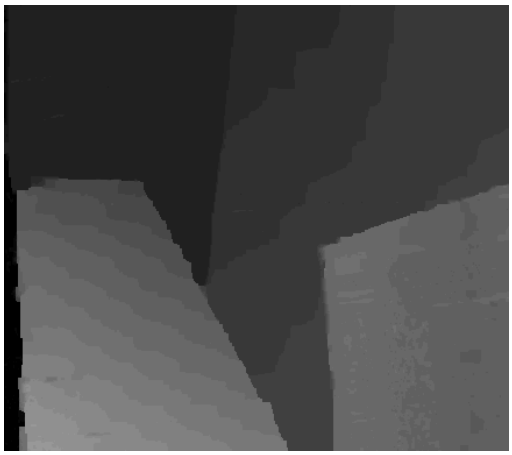
5.7.2 Benchmarking Venus Dataset

(a) Left Image

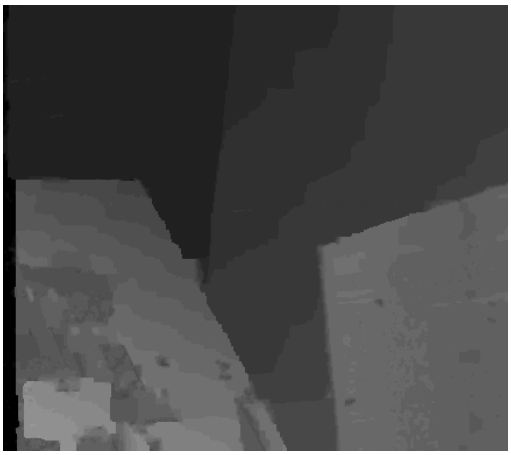
(b) Right Image



(c) Ground Truth Disparity



(d) Parameters values: -a 1 -b -n 60 -o 4 -t 16 -g 10 -p 3 -e 1 -m 1 -l 10



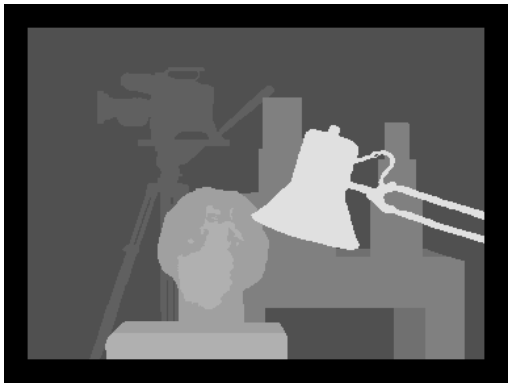
(e) Parameters values: -a 2 -b -n 60 -o 4 -t 16 -g 10 -p 3 -e 1 -m 1 -l 10

Figure 5.6: (d) Expansion Algorithm Benchmark (e) Swap Move Algorithm Benchmark

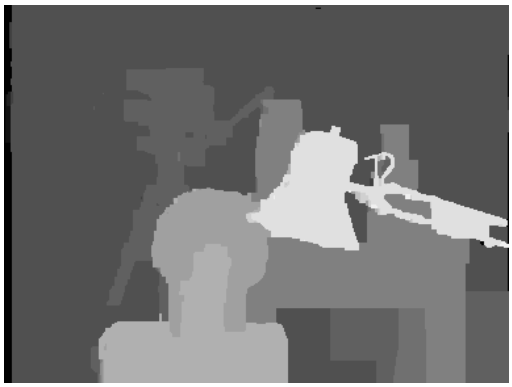
5.7.3 Benchmarking Tsukuba Dataset

(a) Left Image

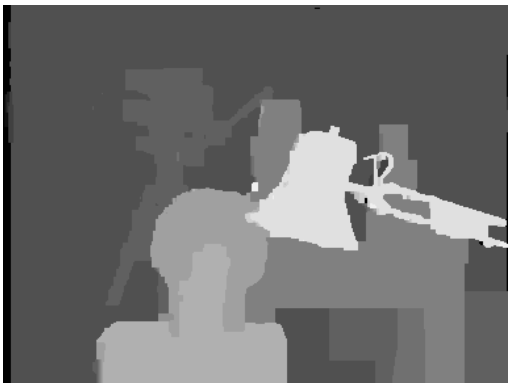
(b) Right Image



(c) Ground Truth Disparity



(d) Parameters values: -a 1 -b -g 8 -o 16



(e) Parameters values: -a 2 -b -g 8 -o 16

Figure 5.7: (d) Expansion Algorithm Benchmark (e) Swap Move Algorithm Benchmark

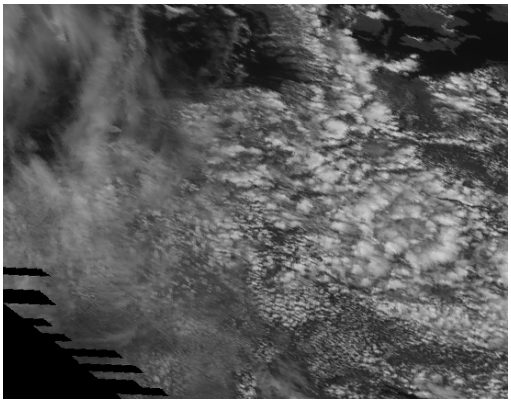
5.7.4 Conclusions of the Benchmarks

Generating of good results requires that the identification of optimal value ranges for the parameters that the algorithms require. For each of the benchmarking datasets these parameters were made available in [72]. Having gained the confidence that the algorithms produce good results on established benchmarking datasets, we can now continue on applying them on our own dataset.

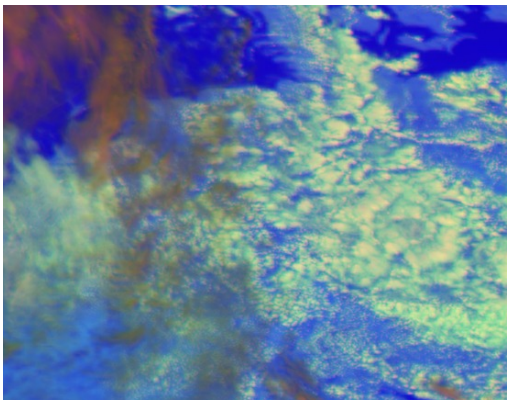
5.8 Results obtained with the KMI Dataset

In order to tune the parameters to approximate the global minimum, one has to manually search for their optimal values. For the image pair in Figure 5.8, there is a cloud mass in the upper left corner which has a high altitude (red color in ground truth image) as well as one with a lower altitude (green color in ground truth image). One would expect the algorithms to produce a disparity image which would have a large white area corresponding to the higher altitude cloud as well as a gray area corresponding to the lower altitude cloud mass.

(a) Left Image

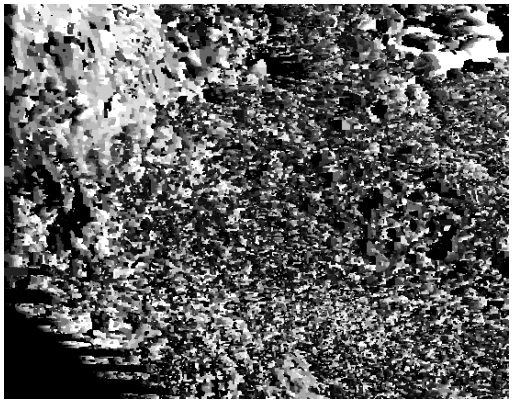


(b) Left Image

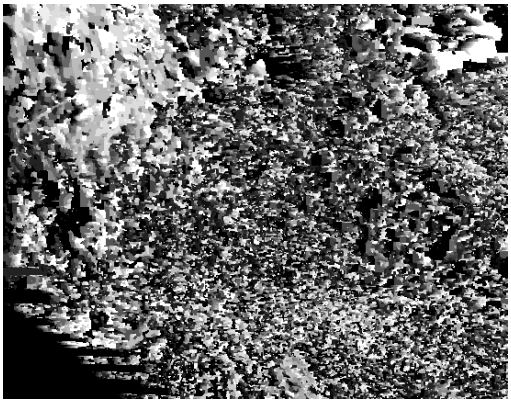


(c) Ground Truth Disparity

Figure 5.8: Satellites' image pair with ground truth

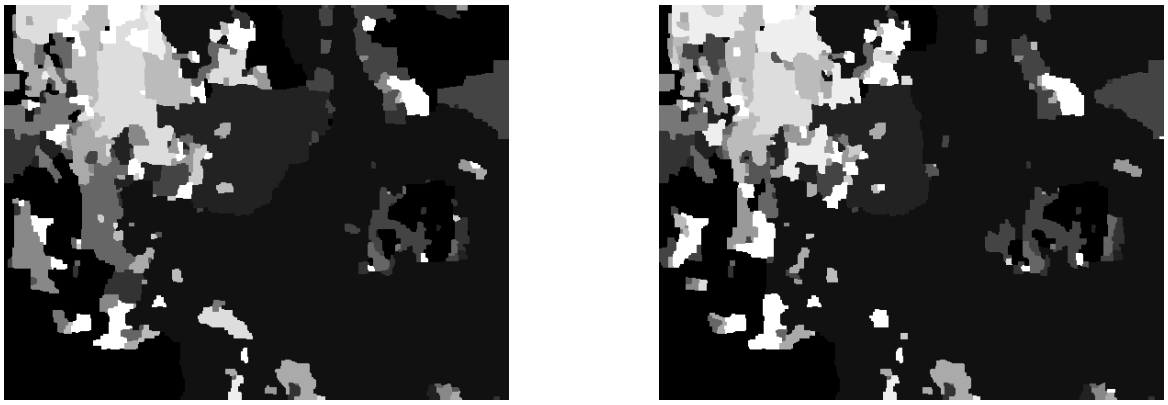


(a) Expansion Move algorithm



(b) Swap Move algorithm

Figure 5.9: Algorithms' results when using square differences. Parameters : -b -s -t 5 -n 16 -e 1 -m 1 -l 5

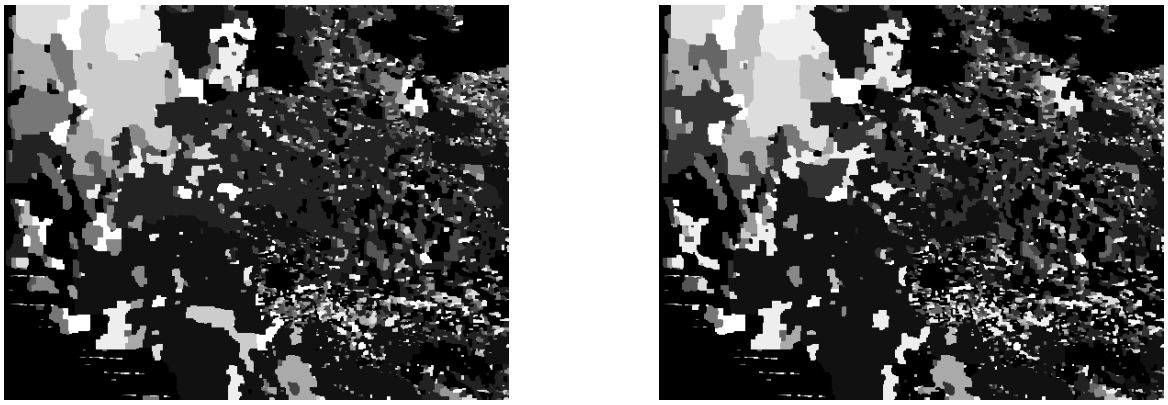


(a) Expansion Move algorithm (b) Swap Move Algorithm

Figure 5.10: Algorithms' results when using absolute differences. Parameters : -a 2 -b -t 5 -n 16 -e 1 -m 1 -l 5

Figures 5.9 and 5.10 show the results given by the algorithms using either the absolute differences or the square differences option. It becomes clear, through careful observation of these images, that the use of square differences (-s parameter) instead of the absolute differences has a clear effect on the algorithms result. Although both the use of square and absolute differences detect the high altitude cloud (north-west), both have problems detecting the low altitude cloud mass. In the case of the square differences being used, the low altitude cloud cannot be distinguished from the high altitude cloud. On the other hand, the use of absolute differences results in many of the areas corresponding to the low altitude cloud being matched with the sea level.

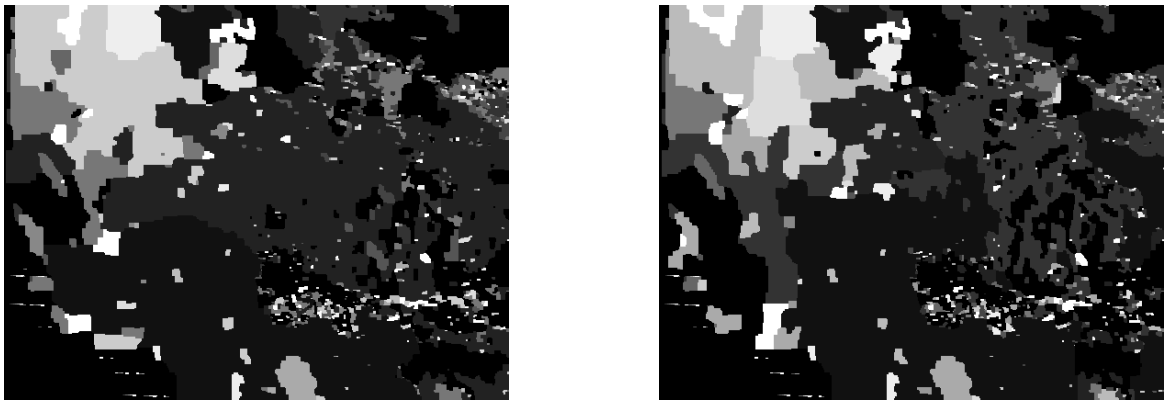
The latter problem can be caused by the fact that this image pair has too little area corresponding to the sea level (north and north-east areas), and so the algorithm detects the lower-altitude cloud as the maximum depth. These conclusions apply to both algorithms as both of them give similar results in the corresponding cases (a) and (b) of Figure 5.9 and Figure 5.10.



(a) Expansion Move algorithm (b) Swap Move algorithm

Figure 5.11: Algorithms' results when using higher λ value (13 instead of 5 of the previous attempt). Parameters : -b -n 16 -e 1 -m 1 -l 13

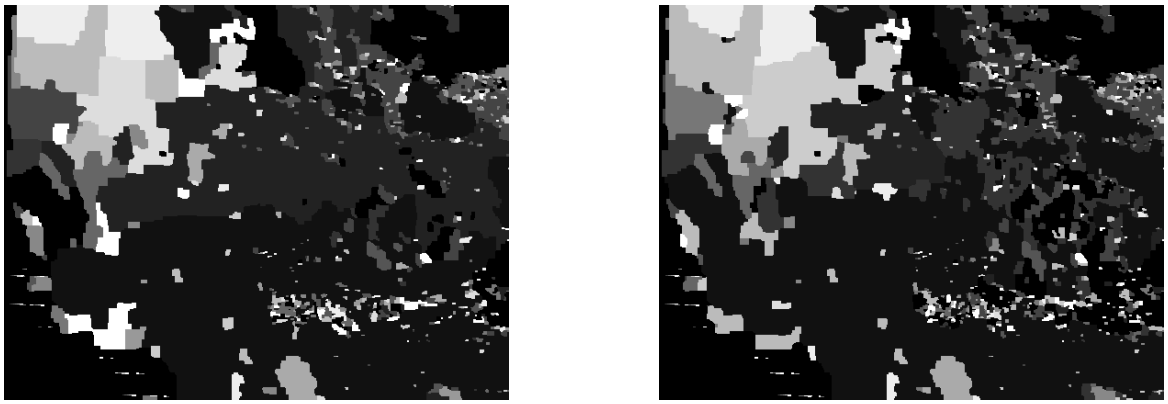
The use of a higher λ value, which means that the smoothness energy E_s has a greater impact on the energy value E than the data energy E_d, seems to address the problem of the low altitude cloud mass not distinguishing from the sea level up to some extent, as shown in Figure 5.11



(a) Expansion Move algorithm (b) Swap Move algorithm

Figure 5.12: Algorithms' results when using higher λ value (20 instead of 13 of the previous attempt). Parameters : -b -n 16 -e 1 -m 1 -l 20

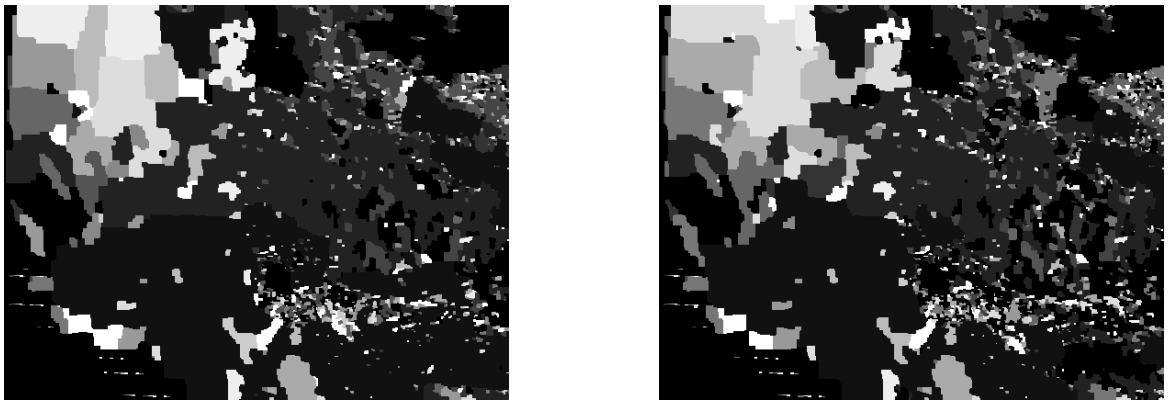
Using the default value of $\lambda = 20$ separates the cloud masses slightly better (Figure 5.12). However, the smoothing is quite large and the details are far from visible and thus these results are not interpretable.



(a) Expansion Move algorithm (b) Swap Move algorithm

Figure 5.13: Algorithms' results when using *L2 norm* instead of *L1 norm* for smoothness exponent. Parameters : -b -n 16 -e 2 -m 1 -l 20

In addition, using the *L2 norm* for the smoothness exponent instead of the *L1 norm* doesn't seem to produce better results (Figure 5.13).



(a) Expansion Move algorithm (b) Swap Move algorithm

Figure 5.14: Algorithms' results when not using Birchfield/Tomasi costs. Parameters : -n 16 -e 1 -m 1 -l 20

In addition, the results when not using the Birchfield/Tomasi costs (-b parameter) are not conclusive regarding its effectiveness (Figure 5.14).

5.8.1 Discussion

As it becomes clear, manual parameter tuning is a very difficult and slow task. Therefore, we developed a script in order to manually test a range of values for those parameters. However, the parameter tweaking was not successful as the results were poor compared to the respective LIDAR data. The poor performance of the algorithms for our dataset indicates that the high dimensionality of the parameters' search space is the main problem.

6. Conclusions

The information about the clouds' height can be very important and helpful for the meteorological research community. In this thesis we utilize the dual view capability of the MSG3 and MSG1 satellites and based on that we apply stereo vision techniques in order to extract any 3D information and estimate the clouds' height. In this final chapter we try to summarize the algorithmic pipeline of our tested method and state some conclusions regarding each major stage of the that pipeline.

First of all, in the Chapter 3 the reader was introduced to the concept of stereo vision and how a typical stereo vision system functions. After carefully reading this chapter the reader has gained the necessary mathematical knowledge in order to understand the goal of each stage of the pipeline.

The image dataset that we were provided with had undergone the rectification step prior to it being delivered to us. However, a relative displacement in number of pixels depending on the longitude and latitude of each location is observed in the dataset. Thus, a preprocessing step was essential before feeding the data to the stereo matching algorithm, as explained in Chapter 4. Because the area of interest was above the Belgian sky, cropping each image pair around that area reduced the effect of that displacement (see Figure 4.2). In order to refine the rectification step, we decided to register each image pair. Because the satellites are geostationary, we only registered a cropped clear-sky image pair and then applied the previously computed transformation to each cloudy image pair. For that step, the first family of registration algorithms that we tested was based on maximizing the **Mutual Information** of the clear sky image pair. We compared two optimizers to estimate the global maximum, the **Gradient Descent** optimizer and the **Powell's** optimizer. The results were similar for both optimizers in most cases, with the Powell's optimizer being slightly better in some. The second approach was based on the low-level features of the image pair and particularly we used the *OpenCV*'s implementations of the **SIFT** and **SURF** algorithms to detect and describe the features, as well as the **RANSAC** algorithm in order to compute the transformation needed to register the image pair. The results of the second method were similar to those of the first. Based on that we could not conclude which approach would best fit our case. Thus, we manually transformed the MSG1's clear sky image and tried to register that one with the original MSG1's clear sky image. The results showed that the feature based method outperformed dramatically the Mutual Information based method. Specifically, in the case of a simple translation the registration with the feature based method managed to achieve a SSIM value of 0.9872 and PSNR equal to 44.6859 between the registered and original MSG1 image. Also, when excluding the margin of 5 first rows and columns, the method was 100% successful, as the SSIM was 1 and the PSNR was ∞. On the other hand, the best attempt of the mutual information based method had a SSIM equal to 0.7769 and PSNR value of 31.6938. Finally, we apply the cropping and the transformation computed from the registration step to each image pair of the dataset.

Finally, in Chapter 5 the reader was introduced to the theory of Markov Random Fields and how it can be used for calculating the disparity map. The goal of every global stereo matching algorithm is to minimize an energy function. For our purpose, the energy function is modelled as the addition of a data energy term and a smoothness energy term. In order to minimize that energy function we test two algorithms that find the optimal Graph Cut, both in the context of finding the maximum flow in a graph. The first algorithm is α-Expansion Move algorithm and the second is $\alpha - \beta$-Swap Move algorithm (the implementation of these algorithms are from `http://vision.middlebury.edu/MRF/code/`). However, the results were not sufficient and a reliable height estimation system

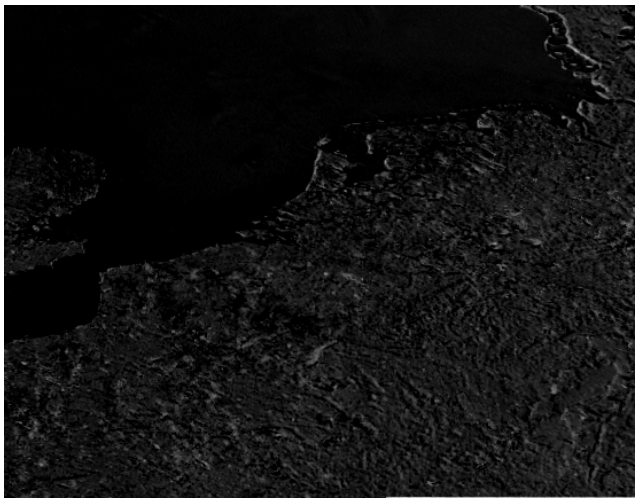
could not be developed. The main problem encountered was the large dimensionality of the search space for the parameter tweaking of those algorithms.

In order to extend the proposed method to produce reliable results, one has to develop an in-depth understanding of the parameters of the used algorithms and how tweaking them will affect performance and accuracy in this specific application. In addition, to the best of our knowledge, more sophisticated algorithms, which can model the parameter tweaking and automatically optimize the search for these parameters, do not exist. Such optimizations are not feasible with the current algorithms due to the difficulty of the theoretical modelling of such problems.

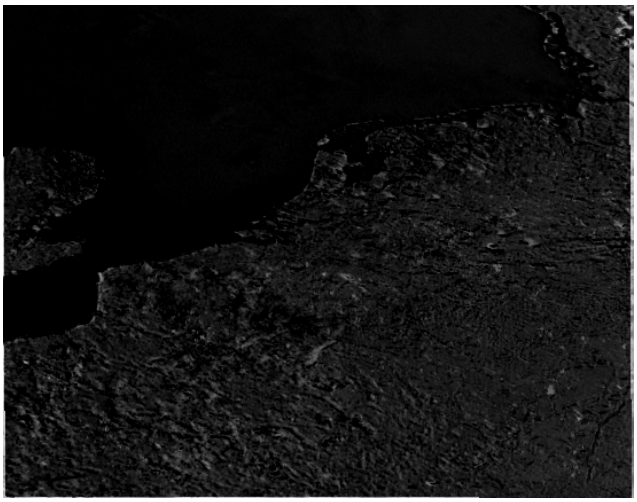
Appendices

A.

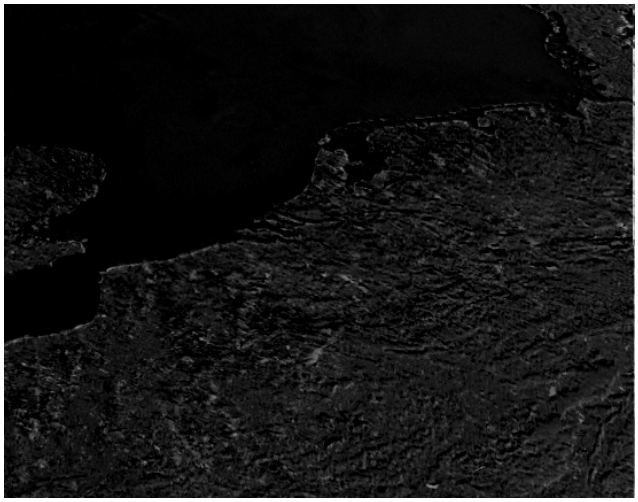
Visualization of the differences of registration results and the reference image discussed and presented in Section 4.2.4.



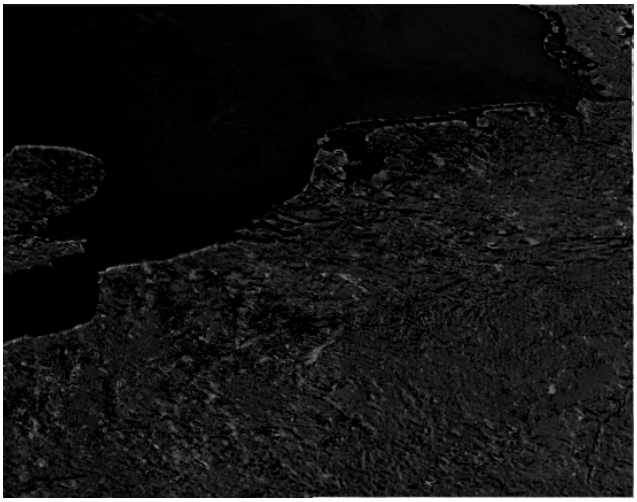
(a) Attempt 1



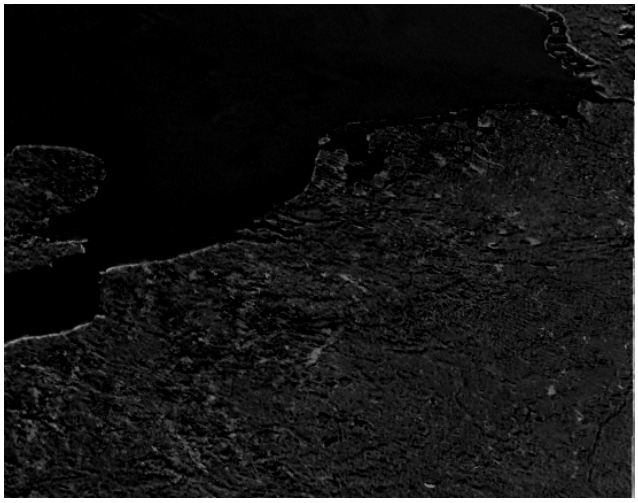
(b) Attempt 2



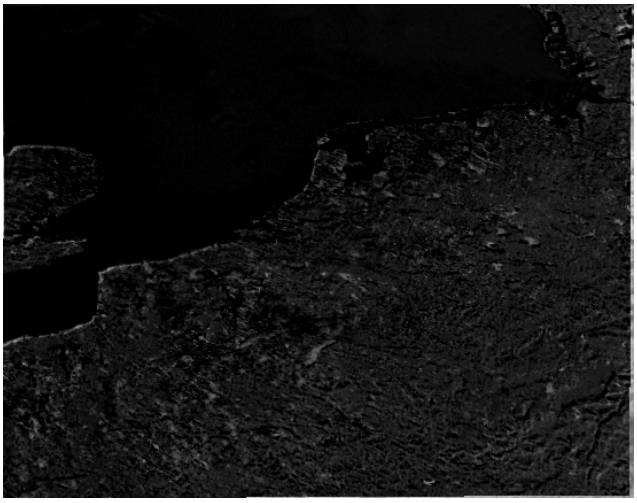
(c) Attempt 3



(d) Attempt 4

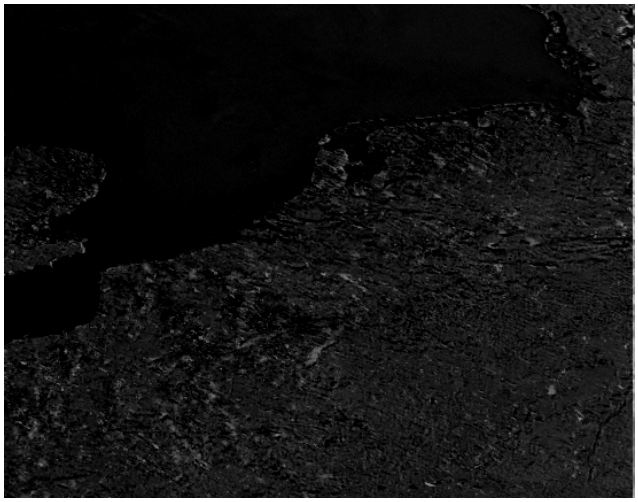


(e) Attempt 5

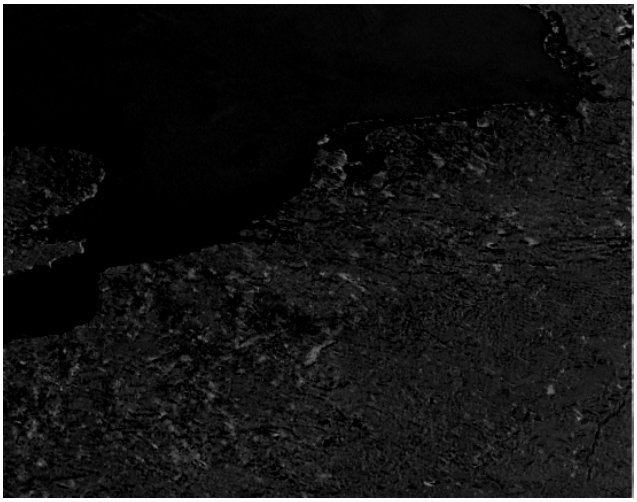


(f) Attempt 6

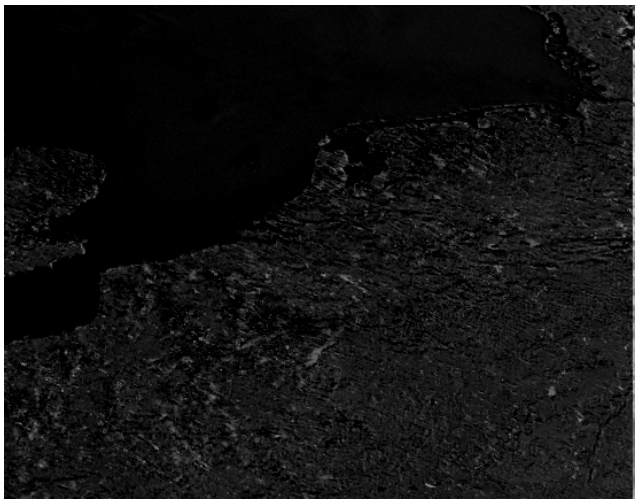
Figure A.1: Visualization of registration results (MI + Gradient Descent Optimizer) as the images' difference.



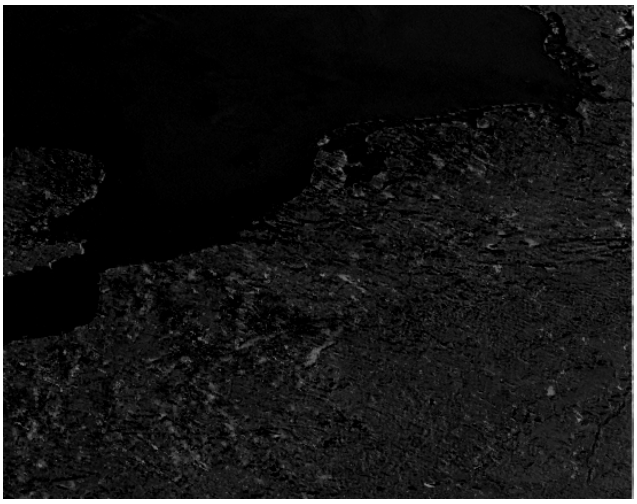
(a) Attempt 1



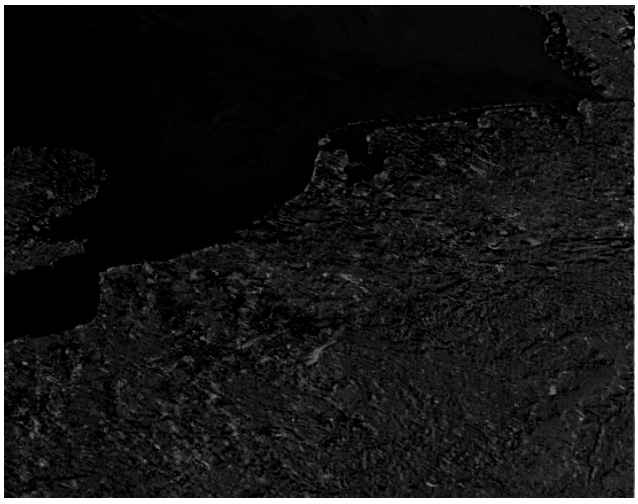
(b) Attempt 2



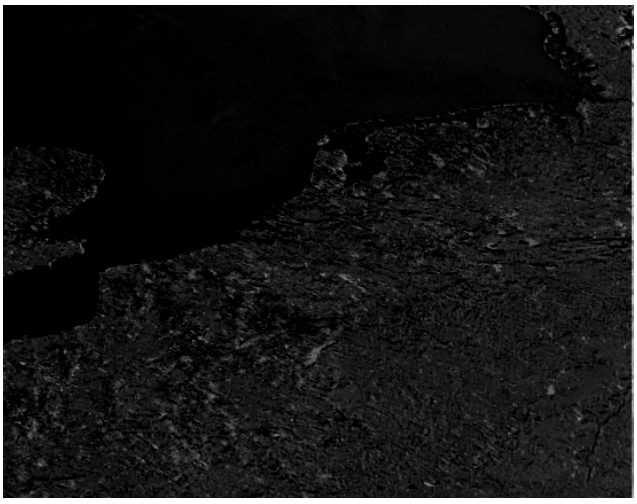
(c) Attempt 3



(d) Attempt 4

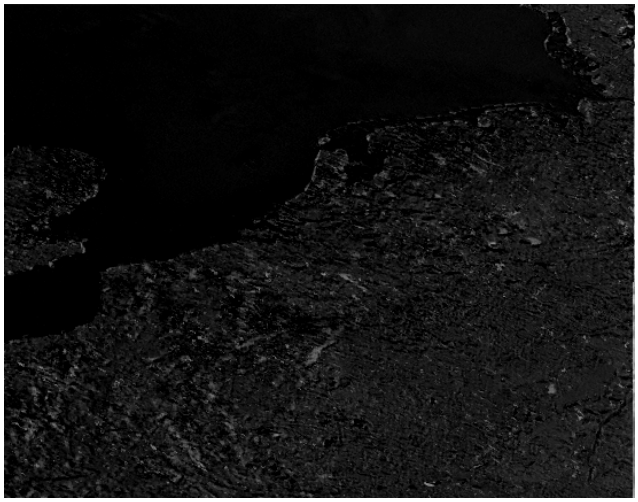


(e) Attempt 5

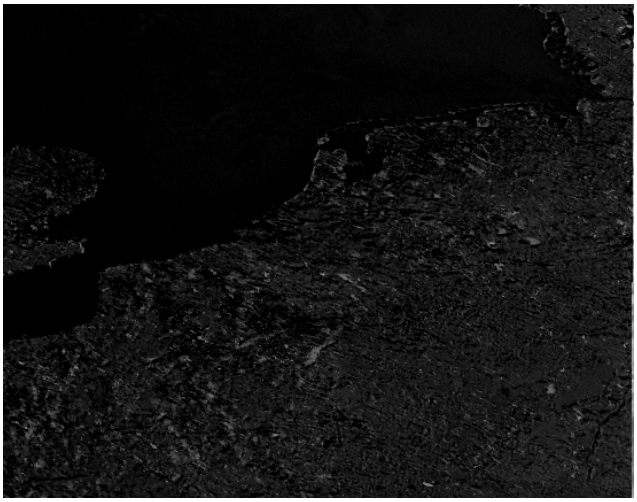


(f) Attempt 6

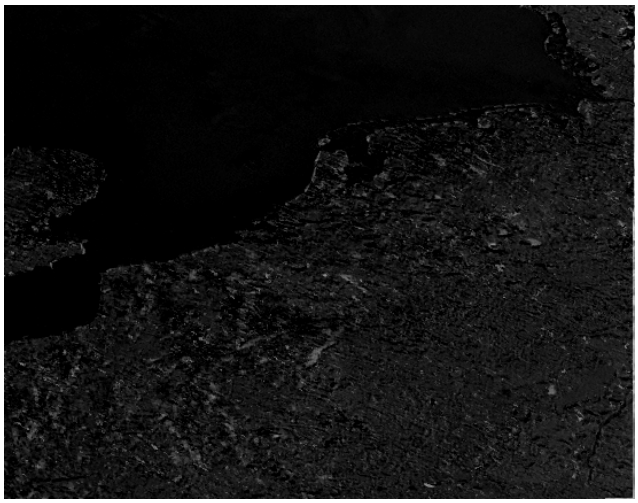
Figure A.2: Visualization of registration results (MI + Powell's Optimizer) as the images' difference.



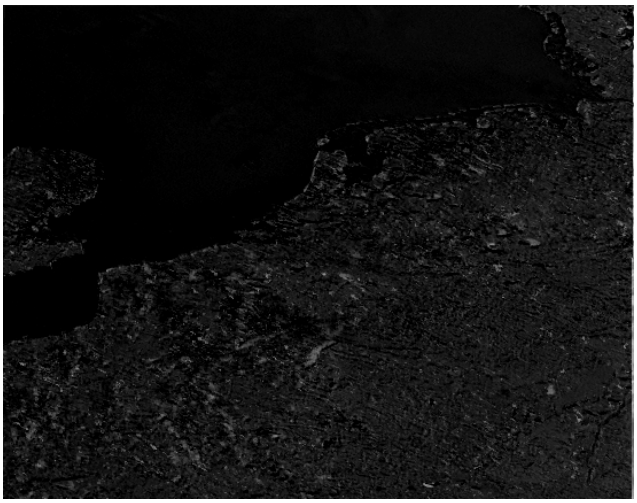
(a) Attempt 1 - Outliers' rejecting ratio = 0.65



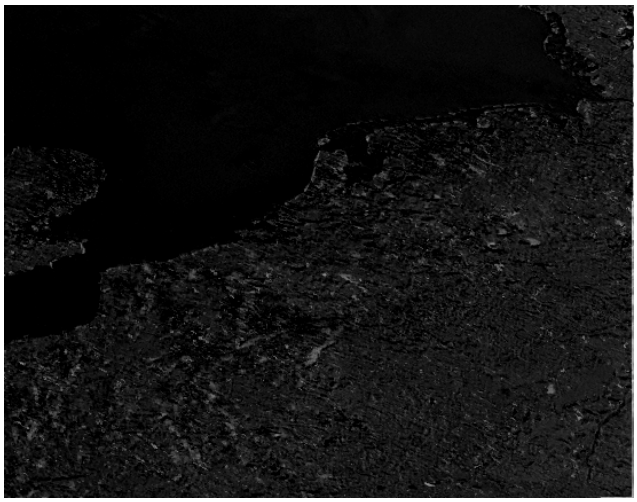
(b) Attempt 2 - Outliers' rejecting ratio = 0.7



(c) Attempt 3 - Correct match rejecting ratio = 0.8

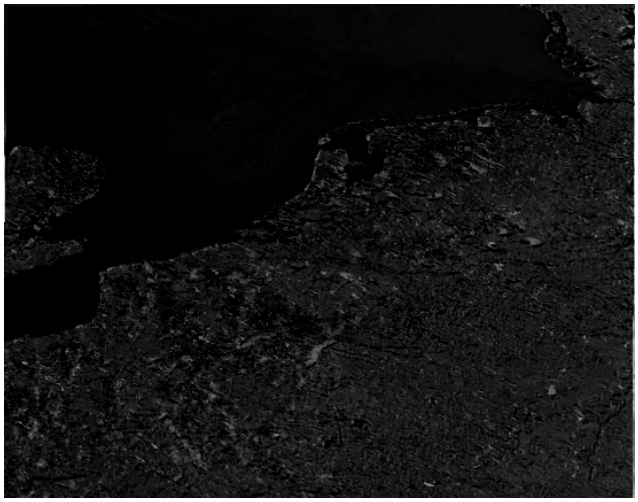


(d) Attempt 4 - Correct match rejecting ratio = 0.85

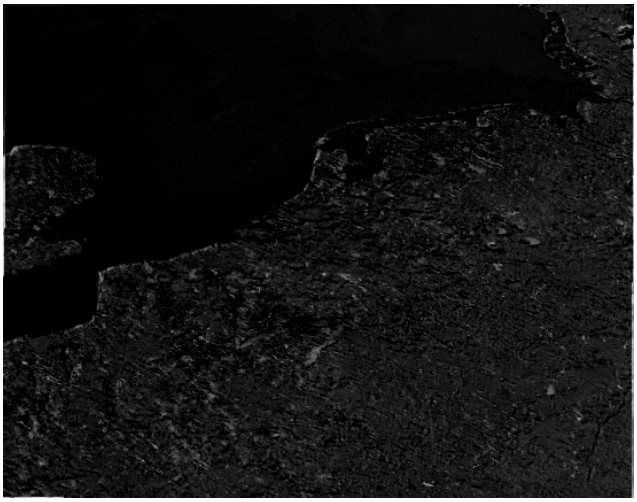


(e) Attempt 5 - Correct match rejecting ratio = 0.9

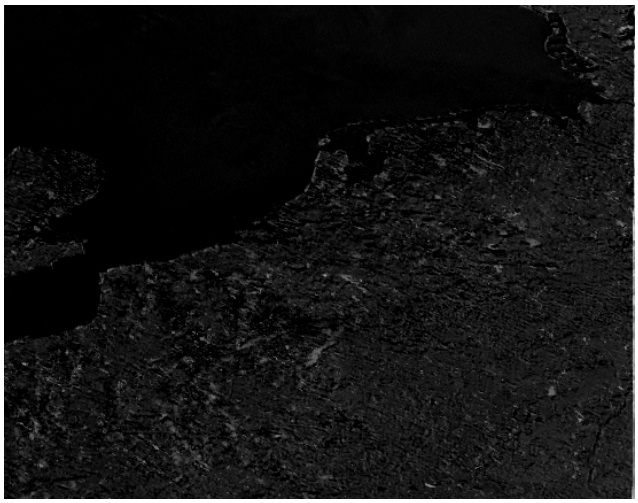
Figure A.3: Visualization of registration results (SIFT + RANSAC) as the images' difference.



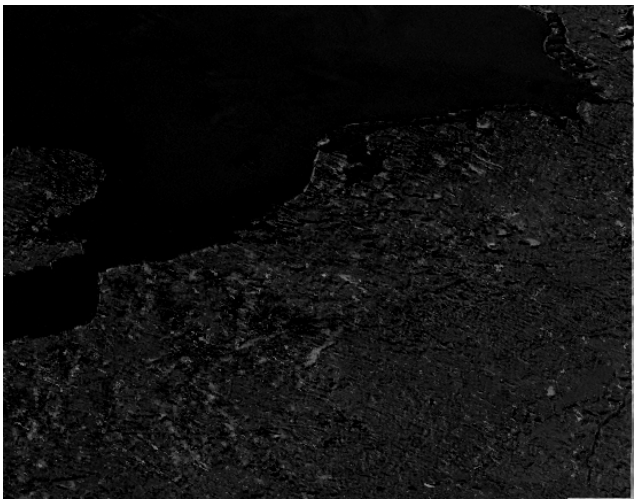
(a) Attempt 1 - Outliers' rejecting ratio = 0.65



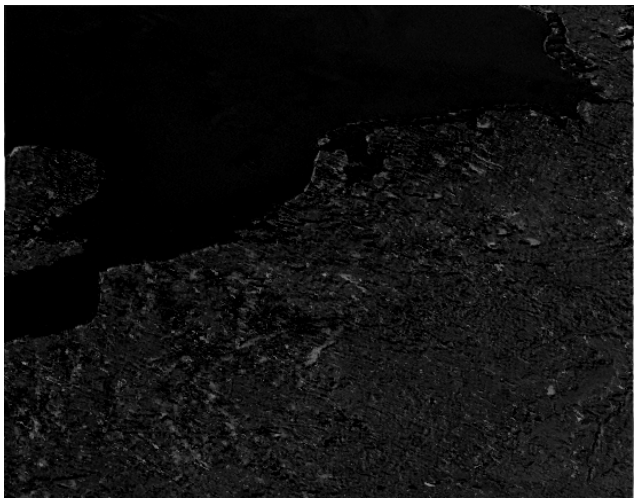
(b) Attempt 2 - Outliers' rejecting ratio = 0.7



(c) Attempt 3 - Correct match rejecting ratio = 0.8



(d) Attempt 4 - Correct match rejecting ratio = 0.85



(e) Attempt 5 - Correct match rejecting ratio = 0.9

Figure A.4: Visualization of registration results (SURF + RANSAC) as the images' difference.

Bibliography

[1] National Aeronautics Space Administration (NASA). *The Importance of Understanding Clouds*. [Online] : https://www.nasa.gov/pdf/135641main_clouds_trifold21.pdf.

[2] I. Tabone, S. Briz, A. Anzalone, A. J. De Castro, S. Ferrarese F. Lopez, F. Isgrò, C. Cassardo, R. Cremonini, and M. Bertaina. Comparing different methods to retrieve cloud top height from meteosat satellite data, 2015.

[3] Luca Merucci, Klemen Zakšek, Elisa Carboni, and Stefano Corradini. Stereoscopic estimation of volcanic ash cloud-top height from two geostationary satellites. *Remote Sensing*, 8(3):206, 2016.

[4] AF Hasler. Stereographic observations from geosynchronous satellites: An important new tool for the atmospheric sciences. *Bulletin of the American Meteorological Society*, 62(2):194–212, 1981.

[5] Victor S Whitehead, Ivan D Browne, and Joe G Garcia. Cloud height contouring from apollo 6 photography. *Bulletin of the American Meteorological Society*, 50(7):522–529, 1969.

[6] Geostationary Orbit. *Technopedia*. [Online] https://www.techopedia.com/definition/14814/geostationary-orbit.

[7] ESA. *MSG Overview*. [Online] https://www.esa.int/Our_Activities/Observing_the_Earth/Meteosat/MSG_overview2.

[8] EUMETSAT. *Meteosat*. [Online] https://www.eumetsat.int/website/home/Satellites/CurrentSatellites/Meteosat/index.html.

[9] World Meteorogical Organization. *Cloud Levels*. Available oneline at: https://cloudatlas.wmo.int/clouds-definitions.html.

[10] Johannes Schmetz, Paolo Pili, Stephen Tjemkes, Dieter Just, Jochen Kerkmann, Sergio Rota, and Alain Ratier. An introduction to meteosat second generation (msg). *Bulletin of the American Meteorological Society*, 83(7):977–992, 2002.

[11] DMA Aminou. Msg's seviri instrument. *ESA Bulletin(0376-4265)*, (111):15–17, 2002.

[12] NETPBM. *Portable Gray Map format*. [Online] http://netpbm.sourceforge.net/doc/pgm.html.

[13] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

[14] Seymour A Papert. The summer vision project. *MIT AI Memos (1959 - 2004)*, Jul 1966.

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[16] Juyang Weng, Paul Cohen, and Marc Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):965–980, 1992.

[17] OpenCV. *Camera Calibration and 3D Reconstruction*. [Online] `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`.

[18] Charles Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 125–131. IEEE, 1999.

[19] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International journal of computer vision*, 17(1):43–75, 1996.

[20] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195, 1998.

[21] Tom Ewbank et al. Master thesis: Efficient and precise stereoscopic vision for humanoid robots. 2017.

[22] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[23] William Eric Leifur Grimson. *From images to surfaces: A computational study of the human early visual system*. MIT press, 1981.

[24] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proc. R. Soc. Lond. B*, 204(1156):301–328, 1979.

[25] Henry Harlyn Baker. Depth from edge and intensity based stereo. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1982.

[26] Daniel Scharstein. *View synthesis using stereo vision*. Springer-Verlag, 1999.

[27] Daniel Scharstein and Richard Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[28] Heiko Hirschmuller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1582–1599, 2009.

[29] Rostam Affendi Hamzah, Afifah Maheran Abdul Hamid, and Sani Irwan Md Salim. The solution of stereo correspondence problem using block matching algorithm in stereo vision mobile robot. In *Computer Research and Development, 2010 Second International Conference on*, pages 733–737. IEEE, 2010.

[30] Changsoo Je and Hyung-Min Park. Optimized hierarchical block matching for fast and accurate image registration. *Signal Processing: Image Communication*, 28(7):779–791, 2013.

[31] William Hoff and Narendra Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE transactions on pattern analysis and machine intelligence*, 11(2):121–136, 1989.

[32] Etienne Vincent and Robert Laganiere. Matching feature points in stereo pairs: A comparative study of some matching strategies. *Machine Graphics and Vision*, 10(3):237–260, 2001.

[33] David J Fleet, Allan D Jepson, and Michael RM Jenkin. Phase-based disparity measurement. *CVGIP: Image understanding*, 53(2):198–210, 1991.

[34] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 377–384. IEEE, 1999.

[35] Yuichi Ohta and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on pattern analysis and machine intelligence*, (2):139–154, 1985.

[36] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002.

[37] Li Hong and George Chen. Segment-based stereo matching using graph cuts. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[38] George Vogiatzis, Philip HS Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 391–398. IEEE, 2005.

[39] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *null*, page 900. IEEE, 2003.

[40] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.

[41] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo matching using belief propagation. In *European Conference on Computer Vision*, pages 510–524. Springer, 2002.

[42] Stan Z Li. Markov random field models in computer vision. In *European conference on computer vision*, pages 361–370. Springer, 1994.

[43] Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

[44] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *Computer vision and pattern recognition, 1998. Proceedings. 1998 IEEE computer society conference on*, pages 648–655. IEEE, 1998.

[45] Wenbo Zhang. Stereo vision using the new dual view meteosat second generation capability. *Master Thesis*, 2018.

[46] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[47] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.

[48] JB Antoine Maintz and Max A Viergever. A survey of medical image registration. *Medical image analysis*, 2(1):1–36, 1998.

[49] André Collignon, Frederik Maes, Dominique Delaere, Dirk Vandermeulen, Paul Suetens, and Guy Marchal. Automated multi-modality image registration based on information theory. In *Information processing in medical imaging*, volume 3, pages 263–274, 1995.

[50] Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1997.

[51] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging*, 16(2):187–198, 1997.

[52] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging*, 22(8):986–1004, 2003.

[53] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[54] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[55] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

[56] Tony Lindeberg. Image matching using generalized scale-space interest points. *Journal of Mathematical Imaging and Vision*, 52(1):3–36, 2015.

[57] Edouard Oyallon and Julien Rabin. An analysis and implementation of the surf method, and its comparison to sift. *Image Processing On Line*, 1:1–13, 2013.

[58] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[59] Marco Zuliani. Ransac for dummies. *Vision Research Lab, University of California, Santa Barbara*, 2009.

[60] *The Insight Segmentation and Registration Toolkit*. [Online] `www.itk.org`.

[61] *The Simple ITK Toolkit*. [Online] `http://www.simpleitk.org/`.

[62] Terry S Yoo, Michael J Ackerman, William E Lorensen, Will Schroeder, Vikram Chalana, Stephen Aylward, Dimitris Metaxas, and Ross Whitaker. Engineering and algorithm design for an image processing api: a technical report on itk-the insight toolkit. *Studies in health technology and informatics*, pages 586–592, 2002.

[63] Will Schroeder, Lydia Ng, Josh Cates, et al. The itk software guide. *The Insight Consortium*, 2003.

[64] Bradley Christopher Lowekamp, David T Chen, Luis Ibáñez, and Daniel Blezek. The design of simpleitk. *Frontiers in neuroinformatics*, 7:45, 2013.

[65] Ziv Yaniv, Bradley C. Lowekamp, Hans J. Johnson, and Richard Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31(3):290–303, Jun 2018.

[66] Witold Kosiński, Paweł Michalak, and Piotr Gut. Robust image registration based on mutual information measure. *Journal of Signal and Information Processing*, 3(02):175, 2012.

[67] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[68] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[69] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003.

[70] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*, pages 564–584. Elsevier, 1987.

[71] Olga Veksler and Ramin Zabih. Efficient graph-based energy minimization methods in computer vision. 1999.

[72] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080, 2008.

[73] Lester Randolph Ford Jr and Delbert Ray Fulkerson. *Flows in networks*. Princeton University Press, 2015.

[74] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):147–159, 2004.

[75] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):1124–1137, 2004.

[76] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
Νικόλαος Τσικνάκης του Εμμανουήλ