Supplement of Atmos. Meas. Tech., 13, 1167–1179, 2020
https://doi.org/10.5194/amt-13-1167-2020-supplement

Supplement of

Automatic quality control of the Meteosat First Generation measurements

Freek Liefhebber et al.

Correspondence to: Freek Liefhebber (liefhebber@stcorp.nl)

MIAD: Meteosat Image Anomaly Detection and Metadata Database

D1.2 Anomaly Detection Algorithm Theoretical Basis Document (ATBD)

Prepared by	:	Freek Liefhebber	15-11-2018
Checked by	:	Ludo Visser	15-11-2018
Approved by	:	Sarah Lammens	16-11-2018

Table of Contents

Change Log

Issue	Author	Affected Section	Reason	Status
Draft	Freek Liefhebber	All	draft version	
0.8	Freek Liefhebber	All	Initial version	Ready for review
0.9	Freek Liefhebber	All	After review	reviewed
1.0	Sarah Lammens	All	Release	Released
1.1	Freek Liefhebber	Section 3 anomaly types	Add anomaly types to ATBD	Released

1 Introduction

1.1 Purpose and scope

The feasibility study **[FEAS]** has defined various anomaly types and various algorithms for the detection in the Meteosat 2-7 level 1.0 and 1.5 files. The feasibility of automatic detection of all anomaly types has been investigated. The algorithms, which were able to detect anomalies with acceptable detection accuracy, have been selected for further development.

During this development, the detection accuracy of the algorithms has improved and the processing speed has increased. Some algorithms have changed significantly in order to achieve acceptable detection accuracy. The developed algorithms for automatic anomaly detection are defined and explained in this document. The detection results of the algorithms are defined in **[RESULTS]**.

Section 3 defines all anomaly types. Section 4 defines the anomaly detection architecture, which drives the detection algorithms and filters the detection results. Section 5 defines all algorithms for the automatic anomaly detection.

1.2 Applicable and reference documents

ID	Title	Reference
[ITT]	Cover letter for EUMETSAT Invitation to Tender No. 17/214228: Meteosat Image Anomaly Detection and Metadata Database, version 1	EUM/COS/LET/16/892559
[SOW]	Statement of Work – Meteosat Image Anomaly Detection and Metadata Database, version 3A	EUM/OPS/SOW/16/861603
[TC]	General Conditions of Tender	
[STC]	Special Tender Conditions – Meteosat Image Anomaly Detection and Metadata Database, version 1	EUM/COS/STC/16/892556

Table 1: Applicable documents

ID	Title	Reference
[CL]	Cover Letter	ST-EUMETSAT-MIAD-LET-001
[TP]	Part I: Technical/Management Proposal	ST-EUMETSAT-MIAD-TMPROP-001
[FP]	Part II: Contractual/Financial Proposal	ST-EUMETSAT-MIAD-CFPROP-001
[FEAS]	D1.1 Report on automatic detection feasibility and methods envisaged	ST-EUM-MIAD-FEAS-001
[RESULTS]	D1.3 Anomaly Detection Test Results	ST-EUMETSAT-MIAD-Result-001
[L0Params]	Definition of level 1.0 parameters (section 5.4 IPS0211: D_IPS_IMAG2TG)	MTP/BF/0901/SP/008
[DIEKMANN]	Overview on METEOSAT Geometrical Image Data Processing, F.J. Diekmann	
[HOLMLUND]	Instrument Calibration Issues: Geostationary Platforms, K. Holmlund	

Table 2: Reference documents

2 Acronyms, Terms and Definitions

Terms, definitions and abbreviated terms that are specific for this document can be found below.

2.1 Terms and Definitions

Term	Explanation

2.2 Acronyms and Abbreviations

Acronym	Explanation
MIAD	Meteosat Image Anomaly Detection and Metadata Database
MIAM	Meteosat Image Anomaly and Metadata

3 Types of anomalies

On basis of the stated anomaly types from the **[SOW]** and the manual inspection of the ground truth dataset, several anomaly types have been defined. This section will define all anomaly types.

This overview is mainly an updated version of the anomaly type overview from **[FEAS]**, which is a result of the feasibility study. After a more in-depth analysis and the development of the algorithms, it became necessary to combine certain anomalies or to create subcategories. The presented anomaly type overview in this section matches with the implemented software and the detections in the MIAD-database.

Section 3.1 presents the defined image anomaly types and section 3.2 the meta-data related anomaly types. Section 3.3 defines the anomaly types, which are defined as out-of-scope of the automatic anomaly detection.

3.1 Image anomalies

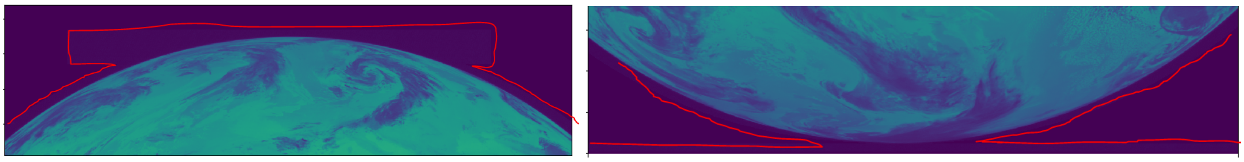
3.1.1 Corrupt file

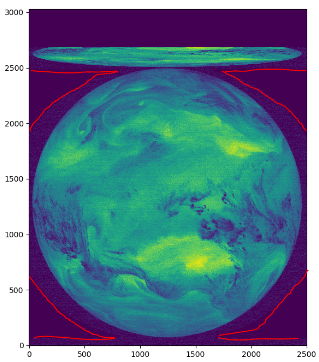
ID	**Detail level**
CorruptFile [34]	Image
Description	
If the size of file is incorrect, at least scanlines are missing and possibly the entire content of the file is unreliable. If this is the case, it is defined as the "file is corrupt" anomaly. If the size of file is too small (17296 bytes x 3069 records) to contain all scanlines, the file is considered to be "corrupt".	
Notes	
The provided dataset did not contain any corrupt file, but some files at Eumetsat appeared to be corrupt.	

3.1.2 Raw data is manipulated

3.1.2.1 Background noise is removed

ID	**Detail level**
BackgroundNoiseRemoved [2]	Image
Description	
Part of the image does not contain the background noise (no signal). Only a narrow noise band around the Earth is preserved. It has a square shape on the North and the South poles.	
Notes	
The background noise removal mainly occurs with satellite M2 and M3. In most cases only a narrow band (like example 1) around the Earth, is not affected, but is some cases the band is much wider (example 2). In general the affected pixels have the intensity 0, but also in cases the intensity 4.	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19810817033000 IR
Example 2	METEOSAT3-MVIRI-MTP10-NA-NA-19900215193000 WV

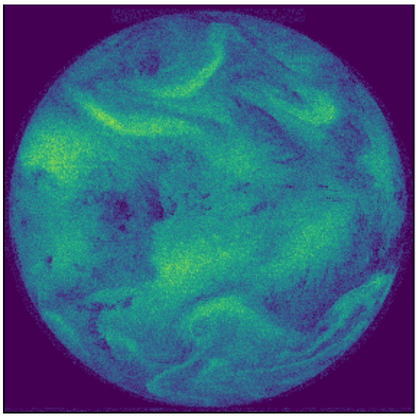
1.


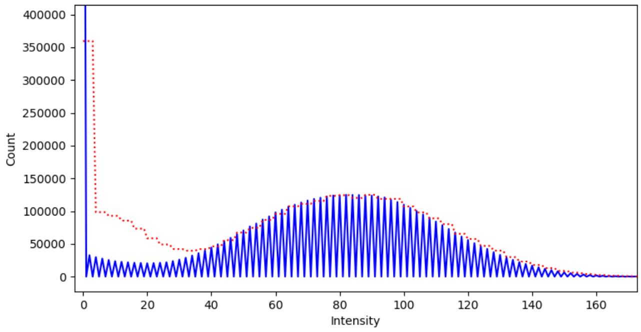
2.


3.1.2.2 Background noise is removed and noise added

ID	**Detail level**
BackgroundNoiseRemoved_NoiseAdded [3]	Image
Description	
Part of the image does not contain the background noise (no signal) and the Earths pixels seem to have additional noise.	
Notes	
This anomaly is quite similar to defined anomaly from the previous section, but also the Earths pixels seem to be affected. The stored histogram in the meta-data and the recalculated histogram from the forward scans are different over the entire range, so also the Earths pixels (with high intensity) are affected.	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19811115003000 WV

1.

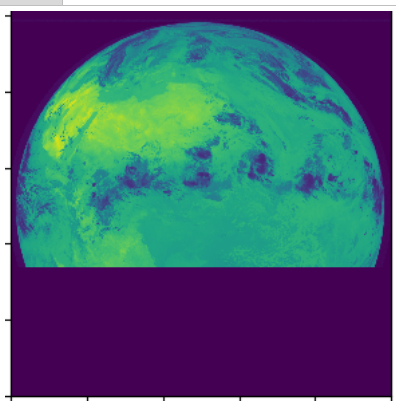


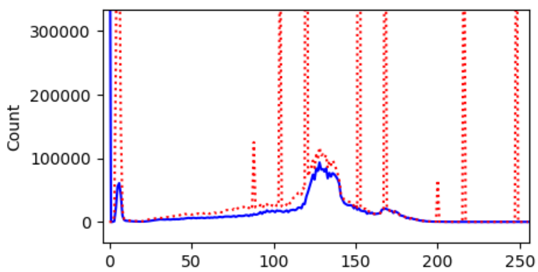


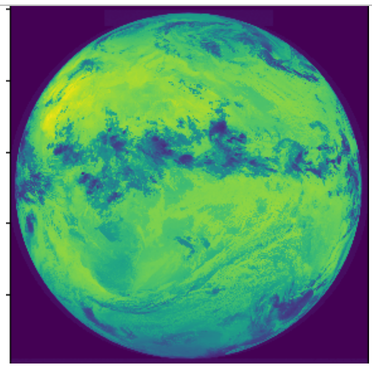
3.1.2.3 Number of scanlines has changed

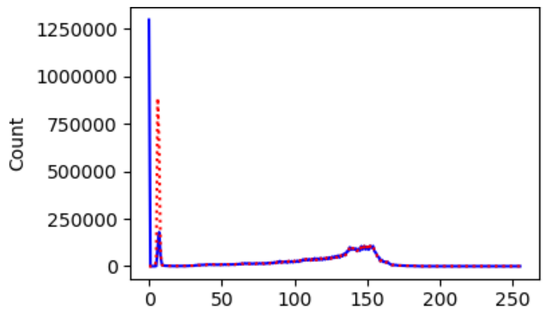
ID	**Detail level**
ScanlinesNumberChanged [31]	Image
Description	
The numbers of pixels (scanlines) in the forwards scans have changed with respect to the stored histograms in the metadata. Somehow one or multiple scanlines have been resetted and the reason for this is unknown.	
Notes	
In general the background noise is removed, but we can't verify this.	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19870817083000 IR
Example 2	METEOSAT3-MVIRI-MTP10-NA-NA-19880815053000 IR

1.



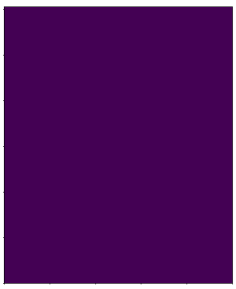


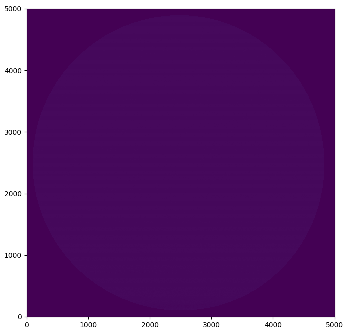
2. 



3.1.3 Completely black

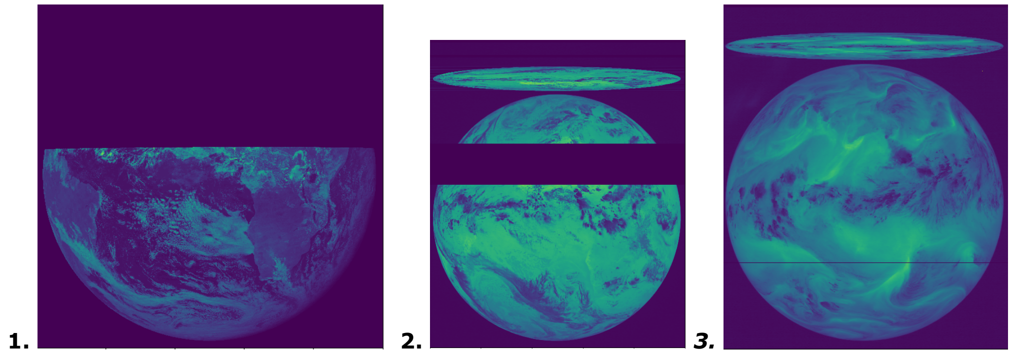
ID		Detail level
CompletelyBlack [16]		Image
Description		
The pixels of a forward scan are all (or almost all) "0" or with a very low intensity.		
Notes		
N/A		
Example 1	METEOSAT5-MVIRI-MTP10-NA-NA-20070416123000 VIS	
Example 2	METEOSAT7-MVIRI-MTP15-NA-NA-19971016103000 VIS	

1. 

2. 

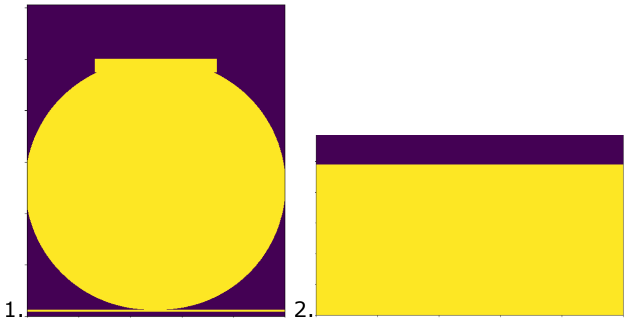
3.1.4 Large black area

ID		Detail level
LargeBlackArea [17]		Scanline
Description		
All pixels of one or multiple scanlines have the value 0 or the pixels from multiple (> 100) scanlines have a very low intensity (equal to noise-level).		
Notes		
Example 1	METEOSAT3-MVIRI-MTP15-NA-NA-19900816133000 VIS	
Example 2	METEOSAT5-MVIRI-MTP10-NA-NA-19960517173000 IR	
Example 3	METEOSAT5-MVIRI-MTP10-NA-NA-20060517193000 WV	



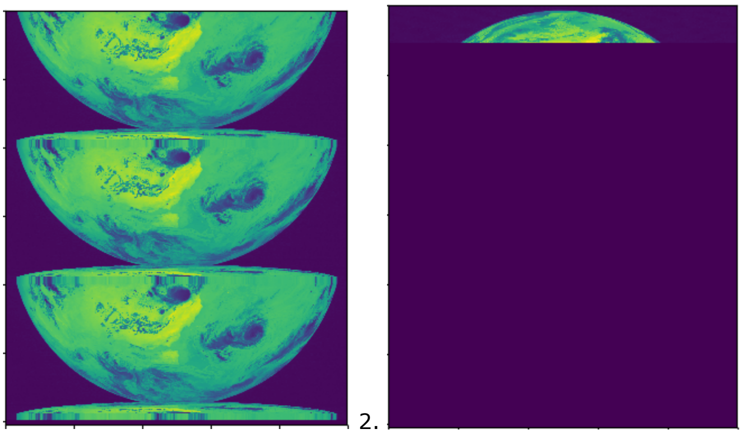
3.1.5 Large white area

ID	Detail level
LargeWhiteArea [18]	Image
Description	
A very large area of the image has the value 255.	
Notes	
N/A	
Example *1*	METEOSAT3-MVIRI-MTP10-NA-NA-19881116223000 WV
Example *2*	METEOSAT2-MVIRI-MTP10-NA-NA-19840215160000 IR



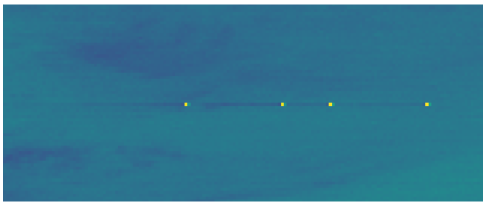
3.1.6 Incomplete image

ID	Detail level
IncompleteImage [33]	Image
Description	
The first forward scan does not contain the entire Earth.	
Notes	
If the height of the first forward is too small (2400 scanlines) or the distance between the detected southern and northern horizon is too small for the entire Earth, this anomaly type should be detected. The underlying root cause for this “anomaly” can be various and in general a more specific anomaly (image or metadata) will also be detected.	
Example *1*	METEOSAT6-MVIRI-MTP10-NA-NA-20010915163000 IR
Example *2*	METEOSAT5-MVIRI-MTP10-NA-NA-19910117180000 WV



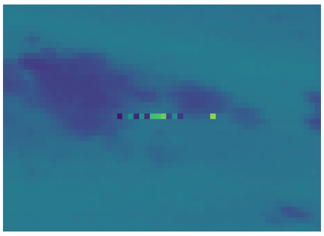
3.1.7 Hot pixel pattern 1

ID	**Detail level**
HotPixelPattern1 [12]	Pixel
Description	
A small number of hot (or incorrect) pixels form a pattern on a single scanline (~4 points pattern) in the WV-channel.	
Notes	
This anomaly type only occurs in M4, M5, M6 and M7 WV-images.	
Example	METEOSAT4-MVIRI-MTP10-NA-NA-19911216110000 WV

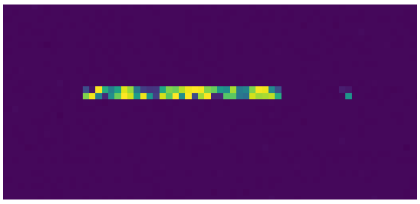


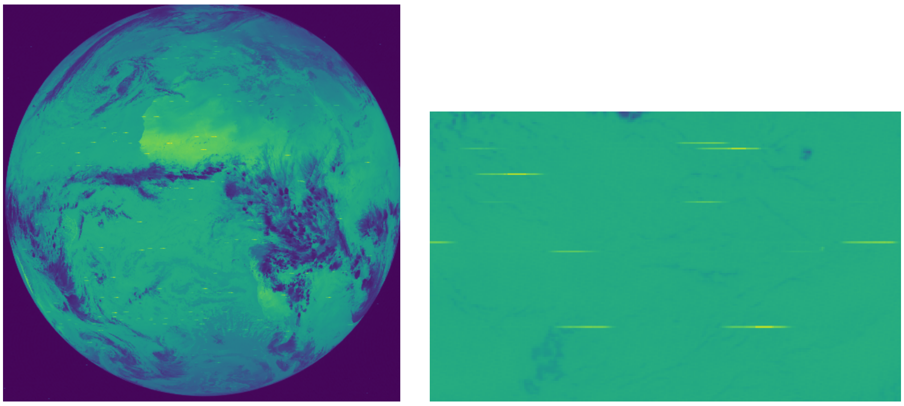
3.1.8 Hot pixel pattern 2

ID	**Detail level**
HotPixelPattern2 [13]	Pixel
Description	
Several consecutive hot (or incorrect) pixels form a pattern on a single scanline. This anomaly occurs in all channels simultaneously (at the same position).	
Notes	
Similar pattern is visible in all signals at the same position. A known anomaly called "Fishes" (example 4) has the same characteristics as the hot pixel pattern 2 anomaly and therefore the "fishes" anomaly will be detected as the hot pixel pattern 2 anomaly. On basis of the ground truth dataset, the "fishes" anomaly is only observed in M4 images of February 1993.	
Example 1	METEOSAT5-MVIRI-MTP10-NA-NA-19960717133000 WV
Example 2	METEOSAT2-MVIRI-MTP15-NA-NA-19811116070000 IR
Example 3	METEOSAT5-MVIRI-MTP10-NA-NA-20031115110000 VIS
Example 4	METEOSAT4-MVIRI-MTP10-NA-NA-19930215163000 IR

1. 

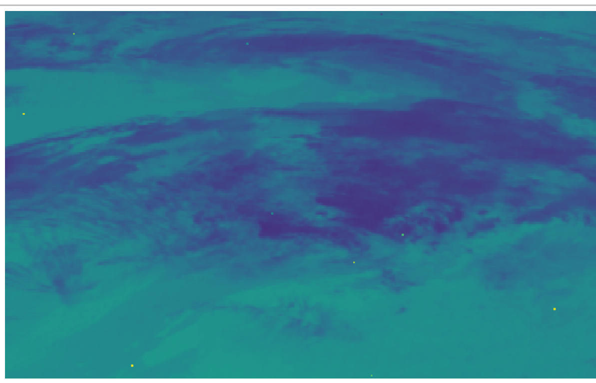
2. 

3. 

4. 

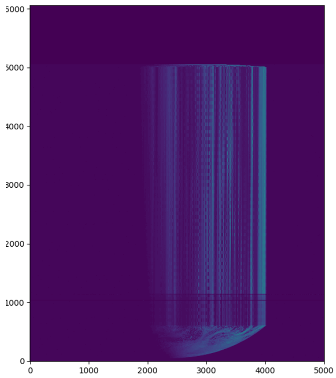
3.1.9 Independent hot pixel

ID	Detail level
HotPixelPatternIndependent [14]	Pixel
Description	
Number of hot or incorrect pixels is visible. The distribution of these multiple pixels does not follow any pattern.	
Notes	
N/A	
Example	METEOSAT4-MVIRI-MTP10-NA-NA-19891016083000 IR



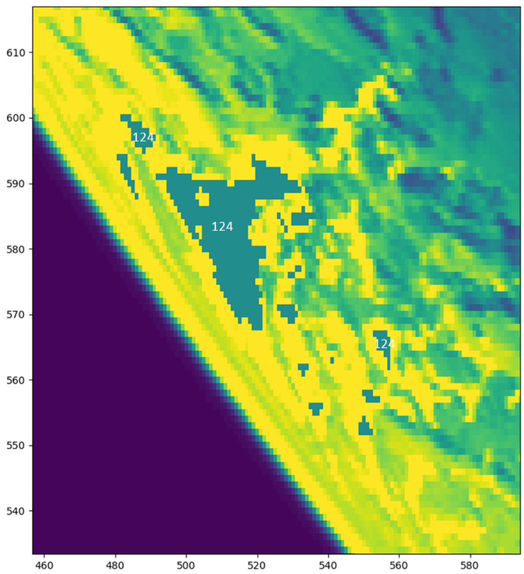
3.1.10 Hanging scanlines

ID	**Detail level**
HangingScanline [25]	Scanline
Description	
From a certain point the same scanline values are recorded resulting in "column-like" image.	
Notes	
Possible cause: stationary scanning-mechanism.	
Example	METEOSAT2-MVIRI-MTP10-NA-NA-19870915073000 VIS



3.1.11 Extreme over illumination

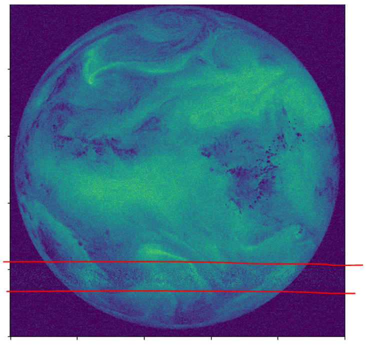
ID	**Detail level**
OverIllumination [28]	Pixel
Description	
Extremely over illuminated pixels of M2 and M3 images have an intensity of 124.	
Notes	
The detectors of the M2 and M3 are sampled by 6 bit AD-convertors and it converted to 8 bit values. In general the maximum value is 252, but extremely over illuminated pixel have a value of 124. In general the over-illuminated pixels (124) are next to pixels with the maximum intensity (252).	
Example	METEOSAT2-MVIRI-MTP10-NA-NA-19831117020000 VIS2



3.1.12 Low SNR Scanline

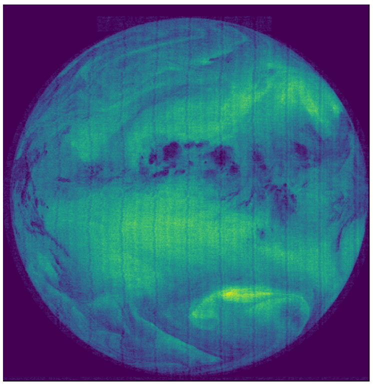
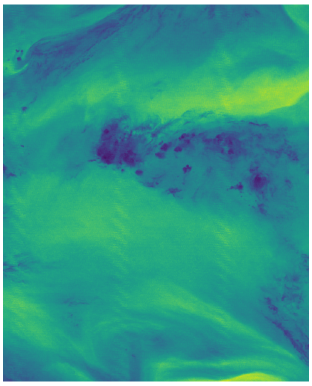
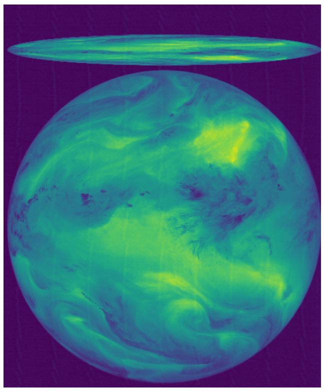
ID	**Detail level**
LowSNR_Scanline [21]	Scanline

Description	
A block with a significant higher noise level present on the image.	
Notes	
This anomaly only appears in the WV-channel	
Example	METEOSAT2-MVIRI-MTP10-NA-NA-19831016120000 WV



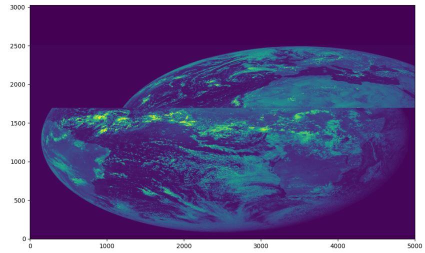
3.1.13 Suspicious pattern / spectrum

ID	**Detection level**
SuspiciousSpectrum [15]	Image
Description	
A visual pattern is visible on the image, which is an indication for a low imaging (acquisition) quality.	
Notes	
Source of the low quality of the images is unknown. It occurs mainly on WV images on a few satellites. The acceptance level will have to be determined during the development of the detection algorithms. It may not be an anomaly per se; it can be known satellite behavior.	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19810817223000 WV
Example 2	METEOSAT4-MVIRI-MTP10-NA-NA-19901015013000 WV
Example 3	METEOSAT4-MVIRI-MTP10-NA-NA-19920815063000 WV

1. 
2. 
3. 

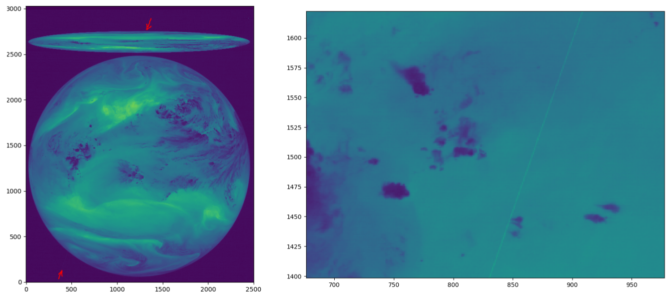
3.1.14 Misalignment

ID	**Detail level**
EastWestHorizonMisAligned [27]	Image
Description	
A group of scanlines is significantly shifted with respect to the other group of scanlines.	
Notes	
N/A	
Example	METEOSAT2-MVIRI-MTP10-NA-NA-19810816140000 VIS



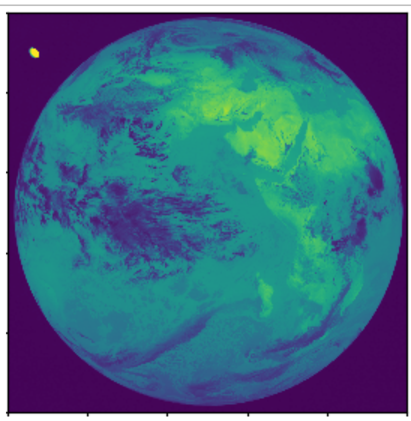
3.1.15 Tilted line

ID		Detail level
TiltedLine [24]		Image
Description		
A tilted line is visible on the WV-channel.		
Notes		
N/A		
Example	METEOSAT5-MVIRI-MTP10-NA-NA-20010817120000 WV	

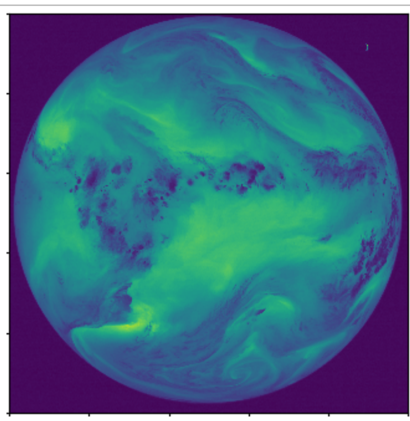


3.1.16 Celestial Body

ID		Detail level
CelestialBody_Moon [22] CelestialBody_Undefined [23]		Pixel
Description		
Not an anomaly perse, but a labeling of the image is needed due to the (un)expected celestial body.		
Notes		
If a celestial body is detected and it can't be the Moon, it will be labelled as an undefined celestial body.		
Example *1*	METEOSAT5-MVIRI-MTP10-NA-NA-20000916103000 IR	
Example *2*	METEOSAT7-MVIRI-MTP10-NA-NA-19981016210000 WV	



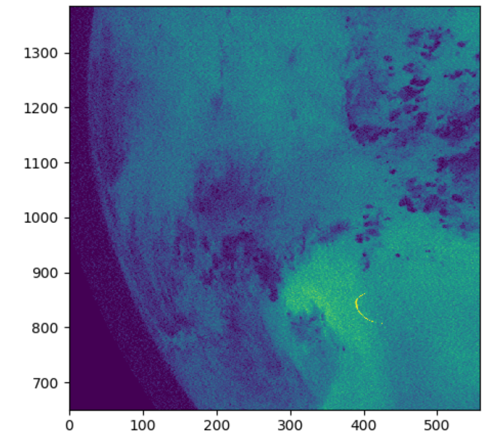
1.



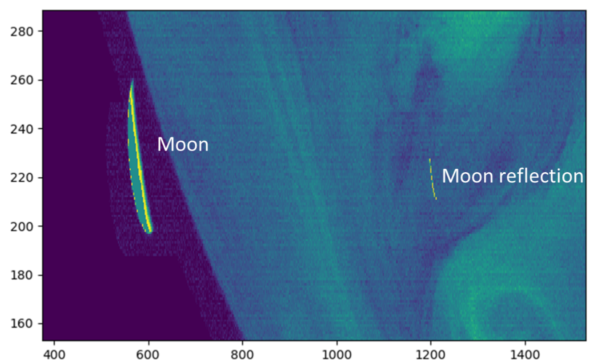
2.

3.1.17 Moon reflection

ID	**Detail level**
MoonReflection [32]	Pixel
Description	
An over-illuminated, nail shaped blob is present in the WV-channel.	
Notes	
This anomaly only appears in M2 and M3 WV-images and if the Moon is in the field-of-view. The position of the anomaly is approximately 600 pixels at the right hand side of the Moon (see example 2). It is expected that the nail shaped blob a indirect reflection of the Moon.	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19870316140000 WV
Example 2	METEOSAT3-MVIRI-MTP10-NA-NA-19881215060000 WV



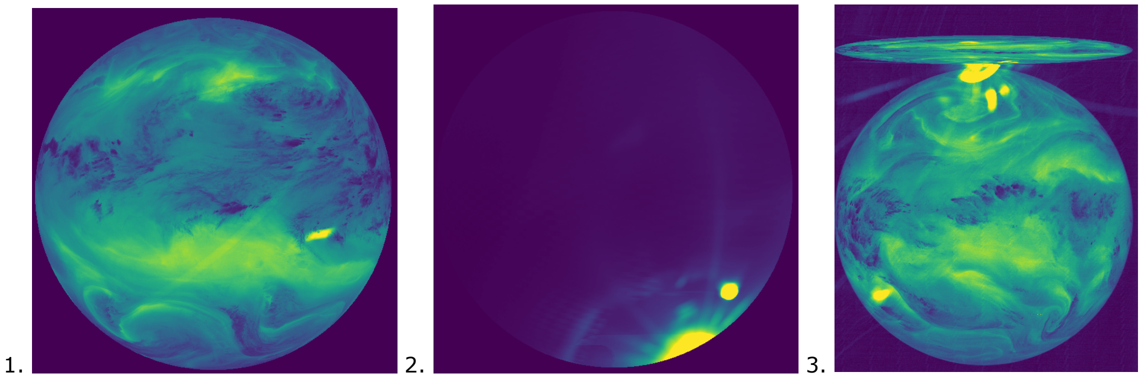
1.



2.

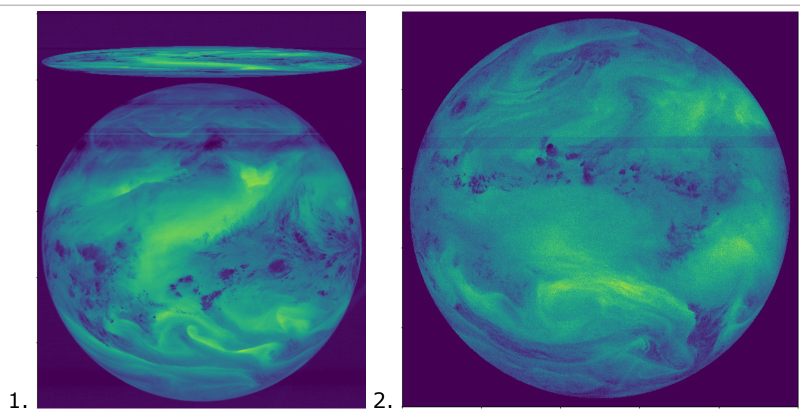
3.1.18 Direct Stray light

ID	**Detail level**
DirectStrayLight [4]	Pixel
Description	
Illumination via an indirect optical path, result in the well-known stray light effect.	
Notes	
Upon detection a number of preceding and following images will be have to be processed due to the moving eclipse effect. The bright spot in the image is called "Sun glint".	
Example 1	METEOSAT7-MVIRI-MTP15-NA-NA-20130717210000 WV
Example 2	METEOSAT7-MVIRI-MTP15-NA-NA-19991016003000 VIS
Example 3	METEOSAT4-MVIRI-MTP10-NA-NA-19900415003000 WV



3.1.19 Indirect stray light effect

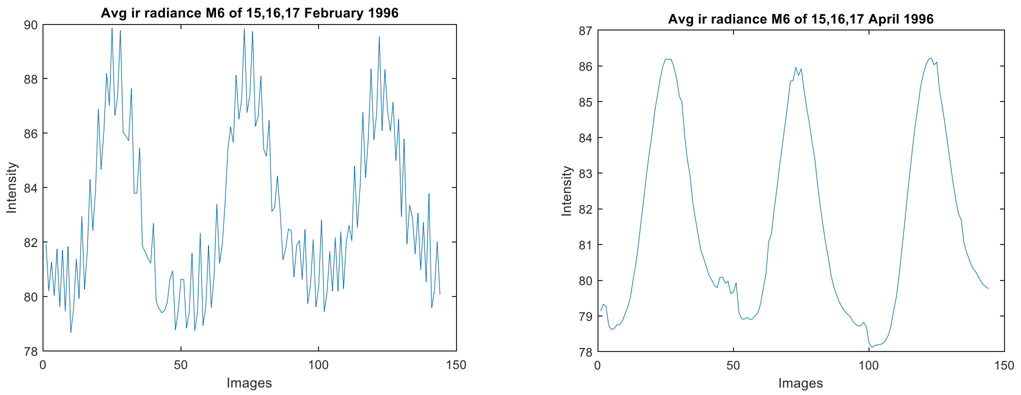
ID		Detail level
IndirectStrayLight [5]		Scanline
Description		
After the appearance of a sun-glint (direct straylight effect), the WV-signal is affected by a phenomenon, which we call “Indirect stray light effect”. This phenomemon affects several scanlines with an irregular pattern, but in general the intensity is lower. The duration of this phenomenon is several hours.		
Notes		
Example 1	METEOSAT5-MVIRI-MTP10-NA-NA-20060317213000 WV	
Example 2	METEOSAT2-MVIRI-MTP15-NA-NA-19860915040000 WV	



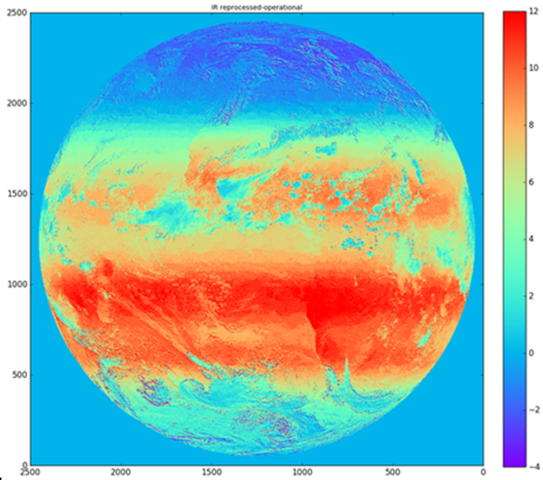
3.1.20 Instable Optics

ID	Detail level
InstableOptics [1]	Image
Description	
The sensitivity of the detectors changes in time due to optics related hardware issues. This anomaly is related to the known hardware issues: M5 rotating lens and M5 loose cold optics.	
Notes	
A known hardware issue of the M6 satellite is that the entire cold optics is loose. This hardware issue causes the average IR- & WV- radiance to vary a lot in time. This high variation between the images is caused by an unpredictable bias in time of the IR- and WV-detector. A known hardware issue of the M5 satellite is that the L3 lens is rotating **[Diekmann]**. The lens is slowly rotating in time and the frequency is not constant (period is 2 to 10 timeslots or even not moving). With a rotating lens the optical center (and distortion) changes continuously and it requires that the geometric deformation calculation changes in time (real-time deformation matrix). The magnitude of the (real-time additional) distortion, due to the rotating lens, is estimated (by analysing the detected north/south and east/west horizons) to be 1.1 IR pixels. Besides the geometric deformation, this anomaly also causes a small variation in the sensitivity of the detectors of 1-2%. **[HOLMLUND]** shows that this anomaly can be detected by analysing the average IR- and WV-radiance in time.	

Example 1	Below the average radiance of the IR-signal of M6 of February 1996 is compared with April 1996.
Example 2	Estimated error (by EUMETSAT) in the IR-channel of the M6 on June 1st (exact timeslot is unknown) 1998 by cross-referencing with data from the M7 satellite.



1.



2.

3.2 Meta data anomalies

3.2.1 Meta data parameter is empty

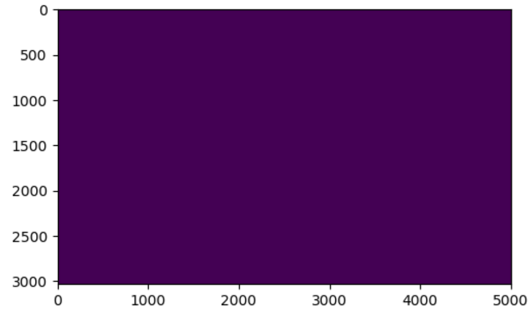
Several parameters of the level 1.0 and level 1.5 files have no value (=0). Occasionally this is caused by radiometric related issues, but in general parameters are not defined due software related issues. The level 1.0 and level 1.5 files are generated by different versions of the processing software and this a reason why some parameters can be empty.

3.2.2 Meta data parameter value is unexpected

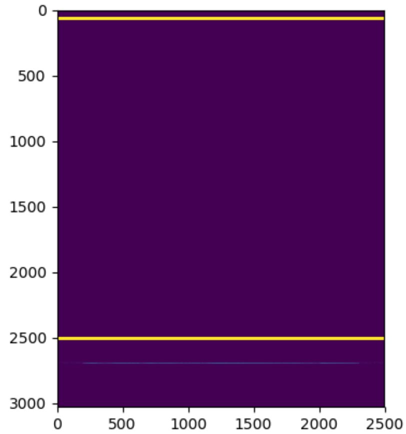
Occasionally meta data parameters have unexpected or invalid values. Unexpected values can indicate a radiometric related issue or uncommon usage of the satellite, such as rapid scans.

3.2.3 Invalid channel

ID	**Detail level**
InvalidSignal [19]	Image
Description	
Occasionally a detector (and also the redundant detector) is disabled and the stored data is unusable.	
Notes	
N/A	
Example 1	METEOSAT2-MVIRI-MTP10-NA-NA-19810916093000 VIS1
Example 2	METEOSAT3-MVIRI-MTP10-NA-NA-19900415153000 WV



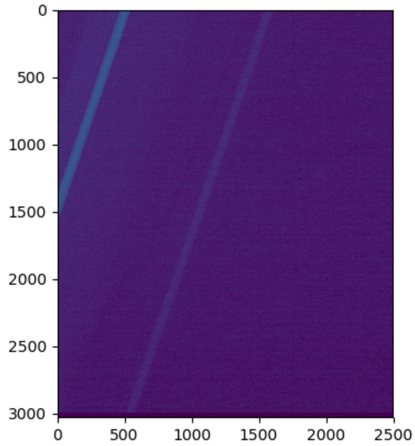
1.



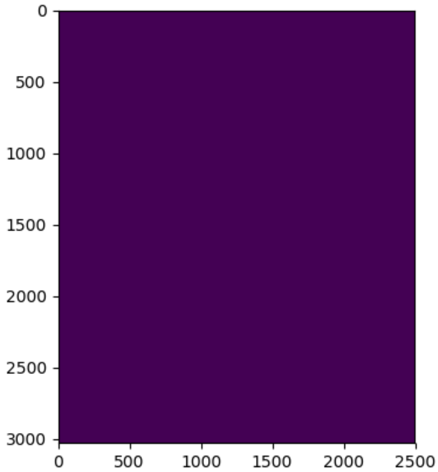
2.

3.2.4 No forward scan

ID	**Detail level**
NoSubImages [20]	Image
Description	
Occasionally a satellite does not make any forward scan and the stored data is unusable.	
Notes	
N/A	
Example 1	METEOSAT7_MVIRI_MTP10_NA_20000915010000 WV
Example 2	METEOSAT5-MVIRI-MTP10-NA-NA-20070416213000 IR



1.



2.

3.2.5 No orbit position

Occasionally the stored orbit position (Earth Fixed Frame or Mean Geometric Format) is empty.

3.2.6 Corrupt orbit position

Occasionally the stored Earth Fixed Frame orbit positon is corrupt. This anomaly mainly appears with the M2 satellite.

3.2.7 Incorrect start time

The start- and end time of a scan is stored. The definition of the start time has changed. Before slot 19 on July 12th 1992 (Julian slot 752707), the start time is defined as the moment that the forward scan starts. After Julian slot 752707, the start time is defined as the moment when the southern horizon is detected. There were doubts if the correct definition of the start time was stored in the files and therefore the start time needs to be verified.

On basis of the ground truth dataset, we can see that there were no mistakes made concerning the used definition of the start time.

But we have discovered that several files have an incorrect start time (not at start forward scan and not at the southern horizon). The start time was at the moment that the image starts (first scanline). The forward scan general starts after several scanlines (e.g. 10). The incorrect start time was stored in all level 1.0 files from the M2 satellite. But it only occurs with other satellites and it seems to be related to uncommon usage of the satellite, such as the creation of a rapid scan.

3.3 Out-of-scope image anomalies

Previous sections have defined several anomaly types, which will be detection by the automatic anomaly detection software. The feasibility study **[FEAS]** has defined a few additional anomaly types, which are defined as outside-of-scope. Because of completeness reasons, these out-of-scope anomaly-types will be briefly described.

Figure 1 shows two level 1.5 images, where the rectification process went wrong. According to **[SOW]** such anomalies are out-the-scope.

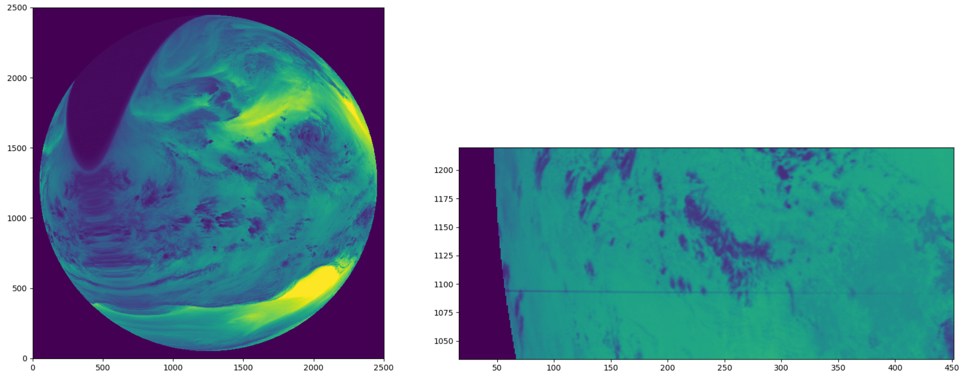


Figure 1 Level 1.5 rectification anomaly. Left: METEOSAT2-MVIRI-MTP15-NA-NA-19900117080000. Right: METEOSAT2-MVIRI-MTP15-NA-NA-19850617160000

Figure 2 shows a few examples, where image anomalies appeared in the backward scan. During the backward scan, similar anomalies can occur as during the forward scan. During the manual investigation, we noticed that the behavior during the backward scan is very inconsistent. The inconsistent behavior complicates the detection of anomalies during the backward scan and the detection accuracy will be poor. Because the backward scan is of low interest, the focus is on the detection of anomalies during the forward scan.

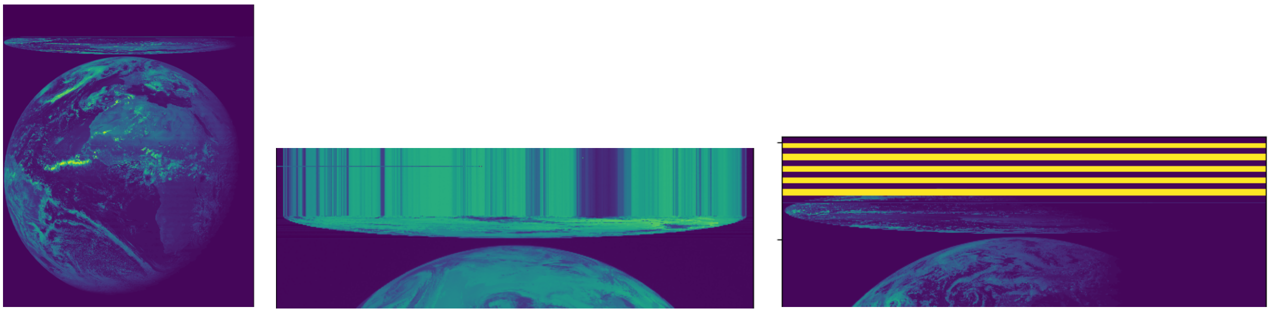


Figure 2 Anomalies in backward scan. Left: METEOSAT3-MVIRI-MTP10-NA-NA-19890616143000. Middle: METEOSAT5-MVIRI-MTP10-NA-NA-19910415070000. Right: METEOSAT3-MVIRI-MTP10-NA-NA-19900517173000.

After the appearance of the moon, the WV channel is affected by a phenomenon, which we call "moon stray light effect" (see Figure 3). This phenomenon is similar to the indirect stray light effect, but the magnitude and duration is much smaller. Because the effect is barely visible and frequency of occurrence is very low, it is considered as not feasible to reliable detect this anomaly.

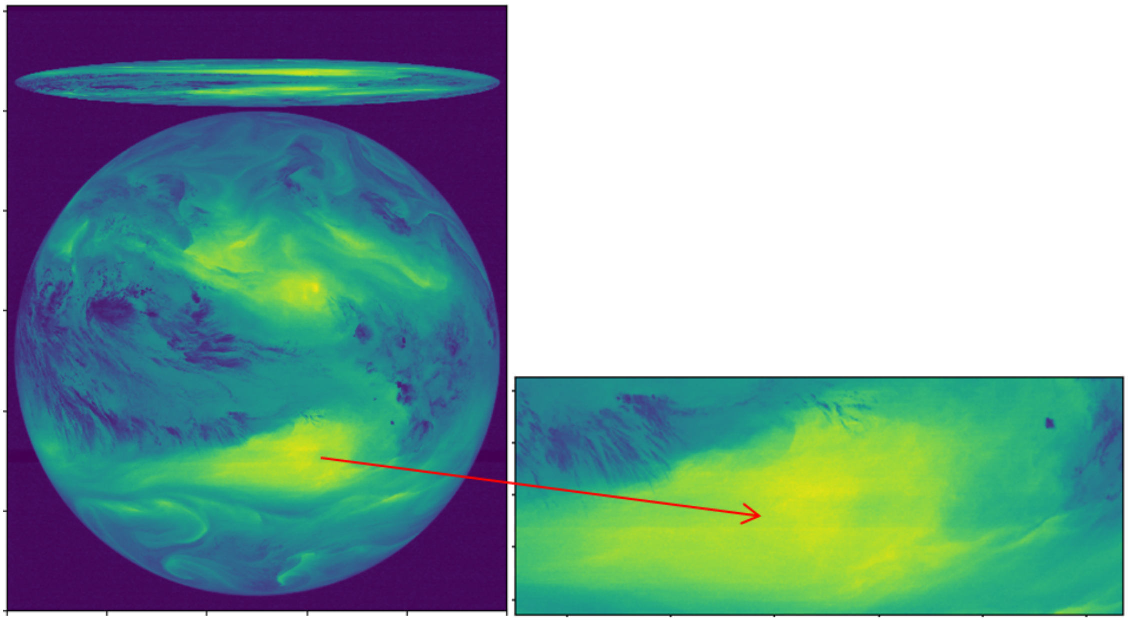


Figure 3 Example of the moon stray light anomaly (METEOSAT7-MVIRI-MTP10-NA-NA-20130916070000)

4 Anomaly detection architecture

This section will define the anomaly detection software architecture. The presented architecture defines conceptually how data is processed and how anomalies are detected in images and meta-data. For readability reasons, the presented architecture does not exactly represent the actual software implementation and it focusses on the used concepts.

Figure 4 shows the anomaly detection architecture. The functionality of each component is defined in the follow subsections.

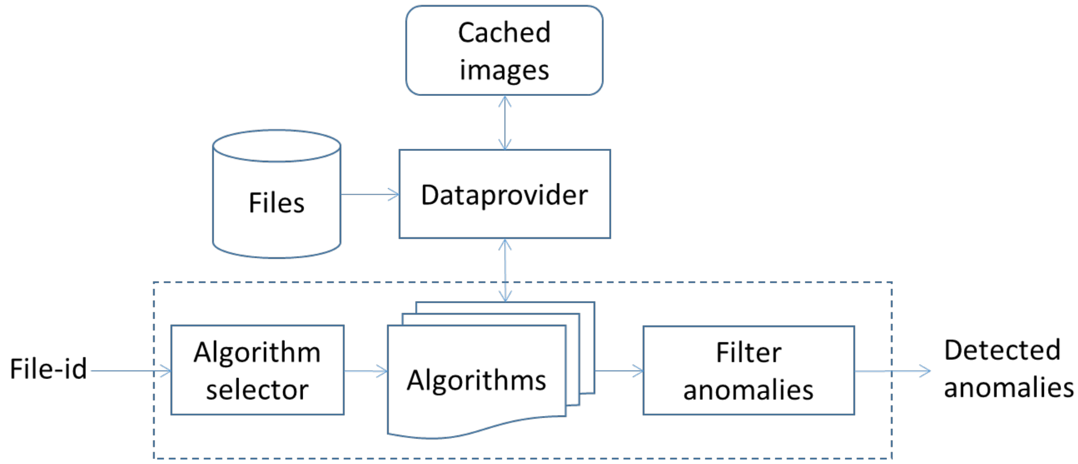


Figure 4 Anomaly detection architecture

The input file-id is actually the expected filename with the following format: METEOSAT**X***MVIRI*MTP**LEVEL***DATETIME**. The output is the anomaly descriptor, which will eventually be inserted in the MIAD-database.

4.1 Algorithm selector

The algorithm selector defines for a particular file-id, which algorithms in combination with which channels should be run. Some algorithms only need to be run once for 1 channel (e.g. WV), while other algorithms needs to be run four times with each time a different channel (VIS1,VIS2,IR and WV). Also some anomalies affect all channels simultaneously and in these cases the channel-id "ALL" will be used. The algorithm for detecting instable optics (InstableOpticsAlgorithm) has special channel-id: IR_AND_WV, because it only affects the IR and WV- channels simultaneously.

The selection of the algorithms is determined by the MTP-level of the file. Table 3 shows the selection of algorithms in combination with channels, which should be run for level 1.0 files and Table 4 for level 1.5 files.

Table 3 Algorithm selection for level 1.0 files

Algorithm	Channel
RawDataManipulatedAlgorithm	VIS1, VIS2, IR, WV
DirectStrayLightAlgorithm	VIS1, VIS2, IR, WV
IndirectStrayLightAlgorithm	WV
VerifyPositionAndStartTimeAlgorithm	ALL
HotPixelPattern1Algorithm	WV
HotPixelPattern2Algorithm	ALL
HotPixelIndependentAlgorithm	VIS1, VIS2, IR, WV
SuspiciousSpectrumAlgorithm	WV
ValidSignalAlgorithm	VIS1, VIS2, IR, WV
CorruptAreaAlgorithm	VIS1, VIS2, IR, WV
CelestialBodyAlgorithm	ALL

TiltedLineAlgorithm	WV
LowSNRScanlineAlgorithm	WV
HangingScanlineAlgorithm	ALL
VerifyEastWestHorizonAlgorithm	ALL
OverIlluminationAlgorithm	VIS1, VIS2, IR, WV
MoonReflectionAlgorithm	WV
MetadataLevel10EmptyAlgorithm	ALL
MetadataLevel10UnexpectedAlgorithm	ALL
InstableOpticsAlgorithm	IR_AND_WV

Table 4 Algorithm selection for level 1.5 files

Algorithm	Channel
MetadataLevel15EmptyAlgorithm	ALL
MetadataLevel15UnexpectedAlgorithm	ALL

After the selection of the algorithms, for each algorithm the "pre-check" is executed. Each algorithm has its own pre-check, which determines if the algorithm should be ran or not. The pre-check is a simple check, which only uses meta-data to determine if it useful to process this file by the selected algorithm.

4.2 Detection algorithms

The "algorithms" are main functionality for detecting anomalies in the image data and meta-data. The principle-of-operation of each algorithm will be defined in section 5.1. This section will briefly describe some (limiting) conditions of the algorithms to fit in the anomaly detection architecture.

The "algorithms" are a set of independent functions for detection anomalies. Each function can detect one anomaly type or multiple closely related anomaly types.

At the beginning of each algorithm a pre-check is executed to verify if the data is valid and if it is useful to execute the remainder of the algorithm. In general the pre-check only uses meta-data parameters, such sat-id, detector enabled, number of scanlines etc, to quickly determine if the file needs to be processed.

The output of an algorithm is a list of anomalies. The anomalies are detected and described per sub-image per channel, where the channel can also be "ALL".

The algorithms are as independently as possible of each other to avoid complex interaction between the algorithms. If a certain anomaly has a strong correlation with another anomaly, so the likeliness of a false-positive detection is high, the design of the algorithm should consider this fact and try to avoid unintended false-positive detection. In general this means, that the algorithm also performs a simplified check for a related anomaly and determine on basis of this check, how to continue with the detection. Often additional checks are done, such as determining if a scanline is completely black (all zero) or if it unexpectedly bright (due to the stray light effect).

For some anomalies, it is very hard to make the detection independent of other anomalies. To reduce the number of false-positive detections for those anomaly-type, the concept of anomaly filtering is introduced, which is explained in the next section.

Image anomalies are not detected in the level 1.5 files, because the detection is already done for the level 1.0 files. The level 1.0 files contain raw sensory data and it is more suited for detecting stray light, radiometric, optics or electronics related anomalies. Therefore the detection accuracy will be higher with level 1.0 files than with level 1.5 files and therefore the image anomalies are only detected in level 1.0 files.

4.3 Filtering anomalies

Preferably an algorithm does not generate false-positive detections, but unfortunately that is not possible. Often there is a relation between false positive detection of certain anomaly type with the true positive detection of another anomaly type. This relation is used to reduce the number of false-positives, by ignoring the detection of a certain anomaly-type if another anomaly-type is detected in the same file.

This concept is called “anomaly filtering” and this filtering is executed when all algorithms have been executed and there is overview of all detected anomalies. For each detected anomaly, other detections in the same file or channel are considered and on basis of filtering rules a detected anomaly can be removed/ignored. Table 5 shows an overview of the anomaly filtering rules.

Table 5 Overview of anomaly filtering rules.

Ignore anomaly	If anomaly is detected	In same channel or file
HotPixelPatternIndependent	HotPixelPattern1	channel
HotPixelPatternIndependent	HotPixelPattern2	channel
HotPixelPatternIndependent	OverIllumination	channel
SuspiciousSpectrum	CompletelyBlack	channel
HotPixelPattern1	LowSNR_Scanline	channel
CelestialBody_Undefined	DirectStrayLight	file
InstableOptics	DirectStrayLight	file

4.4 Data provider

Typically an algorithm only requires a single channel or file, but some algorithms (e.g. indirect stray light effect) require a consecutive sequence of images for the detection of an anomaly. Because some channels or files can be invalid, it is possible that an algorithm requires a larger sequence of files. Because an algorithm knows best, which images and how many images are required for the detection, we have decided to keep this knowledge in the algorithms and create flexible functionality for loading image data. This flexible functionality for loading and caching image data is called “data provider”.

The input arguments of an algorithm are only the file-id and the channel and for retrieving image data, an algorithm can call the data provider. The data provider knows where it can find the requested files on a file-server and it can load the requested data. Typically algorithms request the same data and to avoid reloading the same file, the data provider can cache files. The size of the caching is 21 files, which is large enough to avoid reloading the same file. If the cache is full and a new file should be loaded, then the file from the oldest timeslot is removed from the cache.

From the files, the image data and the meta-data parameters should be extracted. This extraction can be copying a block of memory (copy of the image data), but it can also be derived information such as the average value of an array. To avoid that time consuming or memory consuming extraction methods are executed multiple times, the result is cached.

5 Algorithm description

This section describes the principle of operation of all algorithms for detecting anomalies in the image data or the meta data parameters. The detection algorithms are described in section 5.1. Commonly used algorithms (or calculations) by the algorithms are described in section 5.2.

5.1 Detection algorithms

The following sections define calculations and rules for detecting anomalies. The calculations are defined by pseudo-code or figures. The pseudo-code is defined such that it is readable for the reader and that it can be easily understood what is meant. If a variable is in **red** and **bold**, it is a parameter which is adjustable. The pseudo-code can use the variable "RECORD", which refers to the record data from the level 1.0 file (**[L0Params]**).

It should be stated that the reader should have basic knowledge on basic image processing algorithms to understand the defined algorithms for the anomaly detection.

5.1.1 Level 1.0 Empty meta-data fields

Meta-data parameters should have a valid value or with valid image a certain value is expected. If this is not the case, it is a meta-data anomaly.

The values of several meta data parameters from the level 1.0 file are verified. If it is equal to zero, they are marked as meta-data anomaly. Table 6 shows an overview of all level 1.0 parameters, which are verified if the values is not equal to zero. The beginning of the parameter name refers to the record number within the level 1.0 file and the end refers to the name of the item (see **[L0Params]**).

Table 6 Level 1.0 Meta-data parameters, which values are verified if it is not equal to zero

Level 1.0 Parameter name	Description / check
R01_CHANVI	All detectors are off
R01_CHANIR	All detectors are off
R01_CHANWV	All detectors are off
R01_NSUBIM	Number of subimages is zero
R03_GHTG / R07_GHTG	Distance to center Earth is zero
R03_GRETG / R07_GRETG	Earth equatorial radius is zero
R03_GRPTG / R07_GRPTG	Earth polar radius is zero
R03_DSTART / R07_DSTART	Start grid deformation matrix is zero
R03_DEND / R07_DEND	End grid deformation matrix is zero
R03_DSTEP / R07_DSTEP	Pixel step deformation matrix is zero
R03_NDGRP / R07_NDGRP	Number of grid points deformation matrix is zero.
R03_D_SUN_START / R07_D_SUN_START	Sun coordinates at start scan is zero
R03_D_SUN_END / R07_D_SUN_END	Sun coordinates at end scan is zero
R03_FORBN / R07_FORBN	Orbit coordinates fixed frame at start scan is zero
R03_FORBS / R07_FORBS	Orbit coordinates fixed frame at end scan is zero
R03_ATTIF / R07_ATTIF	Attitude at start scan is zero
R03_ATTIL / R07_ATTIL	Attitude at end scan is zero
R03_ORBF / R07_ORBF	Orbit coordinates at start scan is zero
R03_ORBL / R07_ORBL	Orbit coordinates at end scan is zero

The following pre-checks are verified to execute the algorithm:

- File exists

Table 7 defines the relevant parameters of the anomaly descriptor if a "parameter is empty" anomaly has been detected.

Table 7 Level 1.0 parameter is empty: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	MetadataLevel10EmptyAlgorithm
anomaly_type	ParameterEmpty
aff_items	List item names (icw record_nrs) with a meta data anomaly
record_nrs	List item names (icw aff_items) with a meta data anomaly

5.1.2 Level 1.5 Empty meta-data fields

Previous section described the empty meta-data parameter detection for level 1.0 files. Similar is done for level 1.5 files. Table 8 and Table 9 defines the checks for empty parameter anomaly detections and the output of the algorithm respectively.

Table 8 Level 1.5 Meta-data parameters, which values are verified if it is not equal to zero

Level 1.5 parameter name	Description / check
R01_iImageProcessingStatus	Image processing status is zero
R01_byVisChannnelInUse	All detectors are off
R01_byIrChannnelInUse	All detectors are off
R01_byWvChannnelInUse	All detectors are off
R01_iNumScans	Number of subimages is zero
R02_dsOrbitCoordinatesMgfImageStart	Orbit coordinates at start scan is zero
R02_dsOrbitCoordinatesMgfImageEnd	Orbit coordinates at end scan is zero
R02_flCartCompAttitudeImageStart	Cartesian component of attitude at start scan is zero
R02_flCartCompAttitudeImageEnd	Cartesian component of attitude at end scan is zero
R02_dsOrbitCoordinatesFixedEarthImageStart	Orbit coordinates fixed frame at start scan is zero
R02_dsOrbitCoordinatesFixedEarthImageEnd	Orbit coordinates fixed frame at end scan is zero
R02_dsSunCoordMgfImage_start	Sun coordinates at start scan is zero
R02_dsSunCoordMgfImage_end	Sun coordinates at end scan is zero
R02_flEarthCentreInPixels	Earth center in pixels is zero
R03_shSlotType	Slottype does not contain data (is zero)
R03_dsDeclinationAttNorth	Declination of attitude North is zero
R03_dsDeclinationAttSouth	Declination of attitude South is zero

Table 9 Level 1.5 parameter is empty: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	MetadataLevel15EmptyAlgorithm
anomaly_type	ParameterEmpty
aff_items	List item names (icw record_nrs) with a meta data anomaly
record_nrs	List item names (icw aff_items) with a meta data anomaly

5.1.3 Level 1.0 Unexpected value of meta data fields

Section 5.1.1 defines the detection of empty parameters in the level 1.0 file. Similar is done for detecting if level 1.0 parameters have an unexpected value.

Table 10 Level 1.0 Meta-data parameters, which values are verified to be in certain range

Level 1.0 Parameter	Description / check
R01_NLISIM	If only one subimage and number of scanlines is not equal to 2500
R01_FN2TG	Filename is not equal to " IMAG2TG".
R01_SN2TG	Allowed satcode is M2, M3, M4, M5, M6 or M7
R03_LINMID / R07_LINMID	Earth center in lines is smaller than 1200 or larger than 1300
R03_DEFMAX / R07_DEFMAX	If R0X_NDGRP is equal to 26 and 26x26 deformation matrix X only contains zeros
R03_DEFMAY / R07_DEFMAY	If R0X_NDGRP is equal to 26 and 26x26 deformation matrix Y only contains zeros

Table 11 Level 1.0 parameter has unexpected value: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	MetadataLevel10UnexpectedAlgorithm
anomaly_type	ValueUnexpected
aff_items	List item names (icw record_nrs) with a meta data anomaly
record_nrs	List item names (icw aff_items) with a meta data anomaly

5.1.4 Level 1.5 Unexpected value of meta data fields

Section 5.1.1 defines the detection of empty empty parameters in the level 1.0 file. Similar is done for detecting if level 1.5 parameters have an unexpected value.

Level 1.5 Parameter name	Description / check
R01_szSatCode	Unexpected Satcode (is not equal to M2,M3,M5,M6 or M7) F2 is sometimes used with M2. P2 is sometimes used with M3
R01_shLastLineImage	Last line of image is not equal to 2500
R01_iNumLineWithRadposAnomalies	Number of rectified raw image lines used with RADPOS anomalies is not zero
R02_flEarthCentreInLines	Earth center in lines is smaller than 1200
R03_szSatId	Unexpected Satcode (is not equal to M2,M3,M5,M6 or M7) F2 is sometimes used with M2. P2 is sometimes used with M3

Table 12 Level 1.5 parameter has unexpected value: relevant parameters of the anomaly descriptor,

Parameter	Value / Description
alg	MetadataLevel15UnexpectedAlgorithm

anomaly_type	ValueUnexpected
aff_items	List item names (icw record_nrs) with a meta data anomaly
record_nrs	List item names (icw aff_items) with a meta data anomaly

5.1.5 Valid Channel

This algorithm performs the detection of four anomaly-types, which are closely related. The rules for detecting each anomaly-type are defined in the next sub-sections.

The following pre-checks are verified to execute the algorithm:

- File exists

5.1.5.1 File is corrupt

If the size of file is incorrect, at least scanlines are missing and possibly the entire content of the file is unreliable. If this is the case, it is defined as the "file is corrupt" anomaly.

If the size of file is too small (17296 bytes x 3069 records) to contain all scanlines, the file is detected as "corrupt".

Table 13 File is corrupt: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	ValidSignalAlgorithm
anomaly_type	FileIsCorrupt

5.1.5.2 Invalid channel

For each channel of the level 1.0 file, it is verified if it is valid/detector is enabled. Because each channel has two detectors, we have to verify if one of the detectors is enabled. The following checks are done to verify if a channel is valid:

VIS1_VALID = RECORD[1]["CHANVI"][1] OR RECORD[1]["CHANVI"][3]

VIS2_VALID = RECORD[1]["CHANVI"][0] OR RECORD[1]["CHANVI"][2]

IR_VALID = RECORD[1]["CHANIR"][0] OR RECORD[1]["CHANVI"][1]

WV_VALID = RECORD[1]["CHANWV"][0] OR RECORD[1]["CHANWV"][1]

If a channel is not valid, an anomaly is generated like defined in Table 14. If the meta-data defines that a channel is not valid, it is considered that image data is not useful. This assumption is confirmed during the manual inspection of the ground truth dataset.

Table 14 Invalid channel: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	ValidSignalAlgorithm
anomaly_type	InvalidSignal
sig	VIS1, VIS2, IR, WV

5.1.5.3 No sub-images

In some cases, no forward scan is captured during a timeslot and the number of sub-images is zero. In such cases there is no useful image data in the level 1.0 file, which also is confirmed during the inspection of the ground truth dataset.

To detect file without any forward scan, the following check is executed:

IF RECORD[1]["NSUBIM"] == 0 THEN

Anomaly ➔ NO_SUB_IMAGES

Table 15 No sub-images: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	ValidSignalAlgorithm
anomaly_type	NoSubImages

5.1.5.4 Image not complete

It is assumed that a forward scan contains the entire earth including the northern and southern horizon. If this is not the case, it is defined as "image not complete" anomaly. If the distance between the detected northern and southern horizon is larger than 2400 pixels, the image is considered to be complete.

First the average value of the earth for each scanline needs to be calculated:

IR_AVG_SCANLINE = CALC_AVG_VALUE_SCANLINE(IR_SCANLINES, THRESHOLD = 20)

WV_AVG_SCANLINE = CALC_AVG_VALUE_SCANLINE(WV_SCANLINES, THRESHOLD = 30)

The calculation of CALC_AVG_VALUE_SCANLINE is defined in section 5.2.3. With the average intensity of the earth in each scanline, the northern and southern horizon can be easily found by searching for the first and last element with an intensity higher than 20.

To detect that image is not complete, the following check is executed:

```
IF IR_VALID OR WV_VALID  THEN //(see section 5.2.1)
        IF IR_VALID THEN
                AVG_SCANLINE = IR_AVG_SCANLINE
        ELSE
                AVG_SCANLINE = WV_AVG_SCANLINE
                POS_SOUTHERN_HORIZON = WHERE(AVG_SCANLINE > 20) [FIRST]
                POS_NORTHERN_HORIZON = WHERE(AVG_SCANLINE > 20) [LAST]
                IF (POS_NORTHERN_HORIZON  - POS_SOUTHERN_HORIZON) < 2400 THEN
                        Anomaly ➔ IMAGE_NOT_COMPLETE
```

Table 16 Image not complete: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	ValidSignalAlgorithm
anomaly_type	ImageNotComplete

5.1.6 Corrupt Area or missing

This algorithm performs the detection of three closely related anomaly-types, which defines that an area of the image is corrupt or missing. The rules for detection each anomaly-type is defined in the next sub-sections. These checks are done for each channel (VIS1, VIS2, IR and WV) and for each sub-image.

The following pre-checks are verified to execute the algorithm:

- File exists
- Channel is valid
- Number of sub-images is larger than zero

5.1.6.1 Completely Black

If the large majority of the pixels of an image have a very low intensity, it is considered to have the completely black anomaly. In general the image contains only noise or almost all scanlines are zero and the image is not useful. The following calculated are executed to detect the completely black anomaly:

IF CHANNEL != WV THEN

THRESHOLDBLACK = **thresholdBlack**

ELSE

THRESHOLDBLACK = **thresholdBlackWV**

NR_PIXELS_NOT_BLACK = COUNT(PIXELS > THRESHOLDBLACK)

TOTAL_NR_PIXELS = NR_SCANLINES * NR_PIXELS_SCANLINE

IF (NR_PIXELS_NOT_BLACK / TOTAL_NR_PIXELS) < 0.001 THEN

Anomaly ➔ COMPLETE_BLACK

, where PIXELS are all pixels of a sub-image of a channel. The variable **thresholdBlack** and **thresholdBlackWV** are adjustable by the user and its defaults are 10 and 14 respectively.

Table 17 Completely black anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	CorruptAreaAlgorithm
anomaly_type	CompletelyBlack
sig	VIS1, VIS2, IR, WV
subimg_nr	number of the sub-image

5.1.6.2 Large white area

In some cases, the image is corrupt and the majority of the pixels have an intensity of 255 and it is considered to have the large white area anomaly. This anomaly can be detected with the following calculations:

NR_PIXELS_WHITE = COUNT(PIXELS == 255)

TOTAL_NR_PIXELS = NR_SCANLINES * NR_PIXELS_SCANLINE

IF (NR_PIXELS_WHITE / TOTAL_NR_PIXELS) > 0.5 THEN

Anomaly ➔ Large_White_Area

Table 18 Large white area anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	CorruptAreaAlgorithm
anomaly_type	LargeWhiteArea
sig	VIS1, VIS2, IR, WV
subimg_nr	number of the sub-image

5.1.6.3 Large black area

Quite often the data of several scanlines are missing (intensity of all pixels is zero) or that several consecutive scanlines only contain noise. These cases are classified as the large black anomaly. Requirement for this anomaly is that a "black" scanline(s) is preceded and followed by a number of scanlines, which are not "black". These conditions are slightly different for scanlines, where all pixels are zero, then for a series of scanlines with only noise. The algorithms for both cases are defined in Figure 5 and Figure 6.

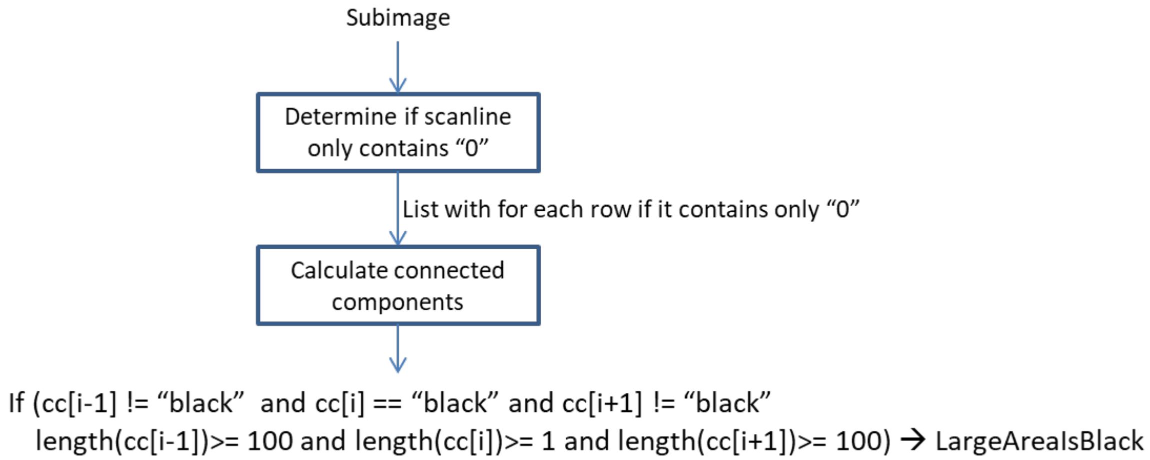


Figure 5 Algorithm for detecting Large Area is black anomaly, where all pixels of a scanline have intensity 0

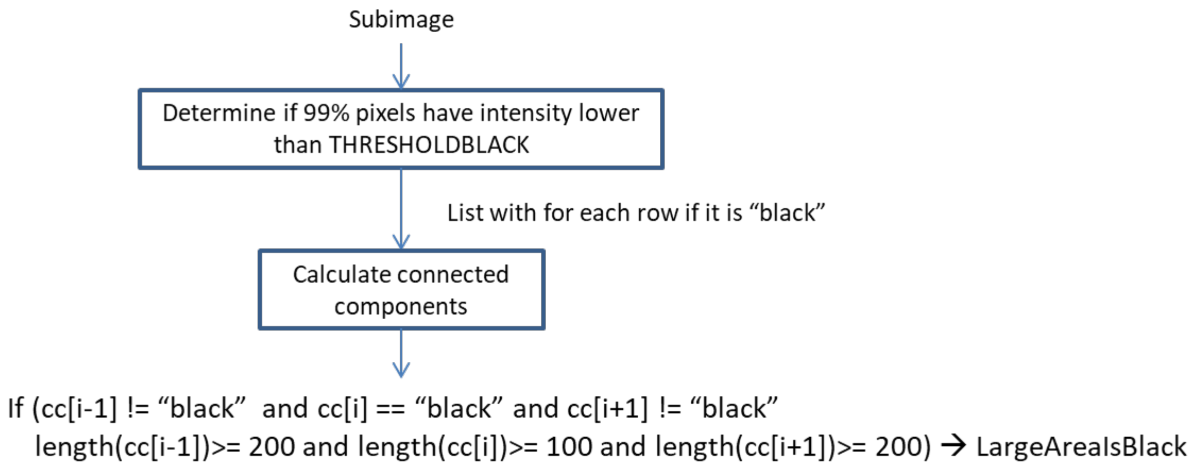


Figure 6 Algorithm for detecting Large Area is black anomaly, where 99% pixels of a scanline have intensity smaller than THRESHOLDBLACK.

For detecting if a scanline only contains noise (intensity smaller than threshold), it is sufficient that 99% of the pixels have a low intensity. It is assumed that the remaining 1% can be affected by the hot pixel anomaly and still it is desired to categorize it a scanline with only noise.

Table 19 Large black area anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	CorruptAreaAlgorithm
anomaly_type	LargeBlackArea
sig	VIS1, VIS2, IR, WV
subimg_nr	number of the sub-image

5.1.7 Raw data is manipulated

From several images the background noise has been removed. With this anomaly only (background) noise in a band around the earth is visible and the other pixels have the value "0" or "4". In such case the noise of detector cannot be estimated by analysing the space corners. To detect that the original data is manipulated, the stored histogram is compared with actual content of the forward scans. If the histograms differ from each other, an anomaly is detected. We distinguish three subtypes:

- Number of scanlines have changed: the total number of pixels of the histograms are different
- Background noise is removed: the histogram differs only at low intensities
- Background noise is removed and noise is added: the histogram differs over the entire range

The detection for the three anomaly-types is defined in the next sub-sections. First we define calculations, which are required for all anomaly-types.

The histogram of all forward scans can be calculated:

```
CALC_HISTOGRAM_ALL = [0 … 0]        // array with 256 elements
NR_SUB_IMAGES = RECORD[1][“NSUMIM”]
FOR SUB_IMAGE_NR in [0  .. NR_SUB_IMAGES]
    Y1    =    RECORD[1][“FLISIM”][ SUB_IMAGE_NR ] – 1
    Y2    =    Y1    +  RECORD[1][“NLISIM”][ SUB_IMAGE_NR ]
    FOR Y in [Y1 .. Y2]
        CALC_HISTOGRAM_ALL = CALC_HISTOGRAM_ALL + CALCULATE_HISTOGRAM(SCANLINES[Y])
```

, where SCANLINES are the scanlines of a channel and CALCULATE_HISTOGRAM refers to a function to calculate the histogram of an array.

We see that the calculation of the stored histogram is not consistent through the entire dataset. With the majority of the images, the histogram is calculated over all pixels of the forwards scans. But in some cases, the quality (J_LINEQUAL) of a scanline and the radiometer position (RADPOS) during the sampling of the scanline is considered. Both cases are considered and the most likely (most correct) result is used. The histogram, where scanlines with poor quality are rejected, can be calculated as follows:

```
CALC_HISTOGRAM_CLEAN = [0 … 0]        // array with 256 elements
NR_SUB_IMAGES = RECORD[1][“NSUMIM”]
FOR SUB_IMAGE_NR in [0  .. NR_SUB_IMAGES]
    Y1    =    RECORD[1][“FLISIM”][ SUB_IMAGE_NR ] – 1
    Y2    =    Y1    +  RECORD[1][“NLISIM”][ SUB_IMAGE_NR ]
    FOR Y in [Y1 .. Y2]
        IF (SCANLINE_QUALITY[Y] == 0 or SCANLINE_QUALITY[Y] == 262144) AND SCANLINE_RADPOS != 0 THEN
            CALC_HISTOGRAM_CLEAN=CALC_HISTOGRAM_CLEAN+ CALCULATE_HISTOGRAM(SCANLINES[Y])
```

, where PIXELS are all pixels of a sub-image of a channel, CALCULATE_HISTOGRAM refers to a function to calculate the histogram of an array, SCANLINE_QUALITY refers to RECORD[Y][“J_LINEQUAL”] and SCANLINE_RADPOS refers to RECORD[Y][“RADPOS”]. With the scanline quality “0”, there were no warnings during the generation of the scanline and with scanline quality “262144” auxiliary data inconsistency was detected.

In some cases, we see that the stored histogram is scaled, such that the histograms of all channels have the same number of pixels. The selection of which calculated histogram (CALC_HISTOGRAM) should be compared with the stored histogram is done as follows:

```
IF SUM(CALC_HISTOGRAM_ALL) > 0 THEN
    RATIO_ALL = SUM(STORED_HISTOGRAM) / SUM(CALC_HISTOGRAM_ALL)
    IF SUM(CALC_HISTOGRAM_CLEAN) > 0 THEN
        RATIO_CLEAN = SUM(STORED_HISTOGRAM) / SUM(CALC_HISTOGRAM_CLEAN)
        IF ABS(RATIO_ALL % 1) < ABS(RATIO_CLEAN % 1) THEN
            CALC_HISTOGRAM = CALC_HISTOGRAM_ALL
        ELSE
            CALC_HISTOGRAM = CALC_HISTOGRAM_CLEAN
    ELSE
        CALC_HISTOGRAM = CALC_HISTOGRAM_ALL
```

ELSE

CALC_HISTOGRAM = CALC_HISTOGRAM_ALL

, where STORED_HISTOGRAM is the stored histogram for channel (RECORD[1][“HVIS1”], RECORD[1][“HVIS2”], RECORD[1][“HIIR”], RECORD[1][“HWV”]).

The stored histogram is in all cases 8-bit, while with the old satellites 6-bit AD-convertors were used. The calculation of the stored histogram was not always consistent, when the data was originally 6-bit. If the histogram is converted to 6-bit and scaled to 100% instead of total number of pixels, this inconsistency of the histogram calculations can be avoided.

5.1.7.1 Number of scanlines have changed

In a few cases, the total number of pixels of a stored histogram does not match the total number of pixels of the calculated histogram. This case is defined as the number of scanlines have changed anomaly. This anomaly can be detected as follows:

IF SUM(CALC_HISTOGRAM) != SUM(STORED_HISTOGRAM) AND SUM(STORED_HISTOGRAM) > 0 THEN

Anomaly ➔ ScanlinesNumberChanged

Table 20 Number of scanlines have changed anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	RawDataManipulatedAlgorithm
anomaly_type	ScanlinesNumberChanged
sig	VIS1, VIS2, IR, WV

5.1.7.2 Background noise is removed

The anomaly background noise is removed anomaly can be detected with the following calculations:

IF SUM(CALC_HISTOGRAM) == SUM(STORED_HISTOGRAM) THEN

CUMUDIF_HISTOGRAM = CUMSUM(CALC_HISTOGRAM – STORED_HISTOGRAM)

// find maximum intensity where less than 100 pixels are different

MAX_INTENSITY_DIFFERENT = ARGMAX(CUMUDIF_HISTOGRAM < 100)

If MAX_INTENSITY_DIFFERENT < **maxIntensityBackground** THEN

Anomaly ➔ BackgroundNoiseRemoved

, where **maxIntensityBackground** is the expected maximum intensity for the background and its default value is 80.

// if the histogram is reset, we always observe that the background is set to “4”

IF SUM(STORED_HISTOGRAM) == 0 THEN

Anomaly ➔ BackgroundNoiseRemoved

Table 21 Background noise is removed anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	RawDataManipulatedAlgorithm
anomaly_type	BackgroundNoiseRemoved
sig	VIS1, VIS2, IR, WV

5.1.7.3 Background noise is removed and noise is added

The anomaly background noise is removed anomaly can be detected with the following calculations:

IF SUM(CALC_HISTOGRAM) == SUM(STORED_HISTOGRAM) THEN

CUMUDIF_HISTOGRAM = CUMSUM(CALC_HISTOGRAM – STORED_HISTOGRAM)

// find maximum intensity where less than 100 pixels are different

MAX_INTENSITY_DIFFERENT = ARGMAX(CUMUDIF_HISTOGRAM < 100)

If MAX_INTENSITY_DIFFERENT >= **maxIntensityBackground** THEN

Anomaly ➔ BackgroundNoiseRemoved_NoiseAdded

, where **maxIntensityBackground** is the expected maximum intensity for the background and its default value is 80.

Table 22 Background noise is removed anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	RawDataManipulatedAlgorithm
anomaly_type	BackgroundNoiseRemoved_NoiseAdded
sig	VIS1, VIS2, IR, WV

5.1.8 Hanging scanline

If during a forward scan, the direction of the detectors is fixed, a scanline will be repeated in the image data. This is anomaly is called the hanging scanline anomaly. This anomaly can be detected by analysing (for each subimage), the "current decoded radiometer position" (RECORD[Y]["RADPOS"]) from the metadata. This radiometer position should be incremented (by 1) with every scanline. Figure 7 shows two examples of the radiometer position during a scan. The left scan is executed correctly, while the right scan shows unexpected behaviour.

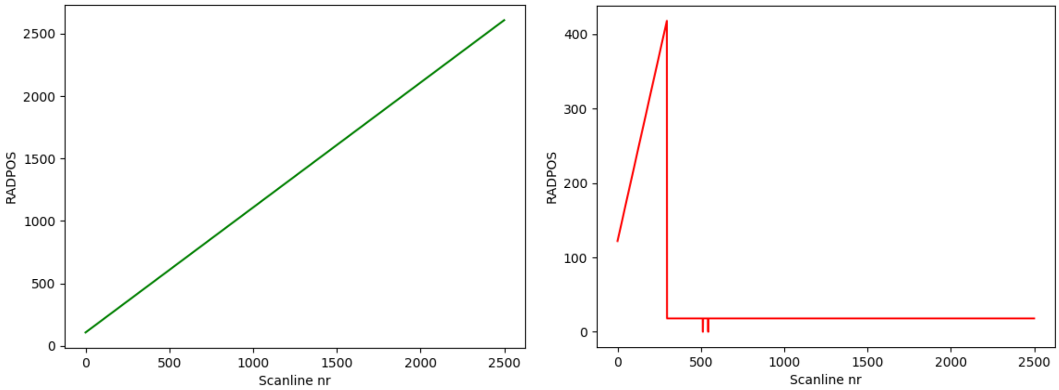


Figure 7 Left: not-hanging, the radiometer position increases with each scanline. Right: hanging, the radiometer position shows unexpected behaviour

Detecting if the "scanline was hanging" can be done by verifying if the radiometer position changes with every scanline. Here it also needs to be verified if the image contains any sub images, because often this is the reason for the appearance of this anomaly.

For the detection, the following calculation should be done:

NR_SUB_IMAGES = RECORD[1]["NSUMIM"]

HANGING_SCANLINES = []

FOR SUB_IMAGE_NR in [0 .. NR_SUB_IMAGES]

```
    Y1    =    RECORD[1]["FLISIM"][ SUB_IMAGE_NR ] – 1
    Y2    =    Y1    +  RECORD[1]["NLISIM"][ SUB_IMAGE_NR ]
    FOR Y in [Y1 .. Y2]
        IF SCANLINE_RADPOS[Y] ==  SCANLINE_RADPOS[Y-1] THEN
            HANGING_SCANLINES = [HANGING_SCANLINES ,  Y]
IF LENGTH(HANGING_SCANLINES) > 0 THEN
    Anomaly ➔ HangingScanline
    AFFECTED_SCANLINES ➔ HANGING_SCANLINES
```

Table 23 Hanging scanline anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	HangingScanlineAlgorithm
anomaly_type	HangingScanline
sig	ALL
locus_type	Scanline
locus	HANGING_SCANLINES

5.1.9 East West Horizon misaligned

For some images, the east and west horizon are misaligned and Figure 8 shows an example of such anomaly. This anomaly is detected by searching for eastern and western horizon for each scanline and verify if the centre is within a certain range.

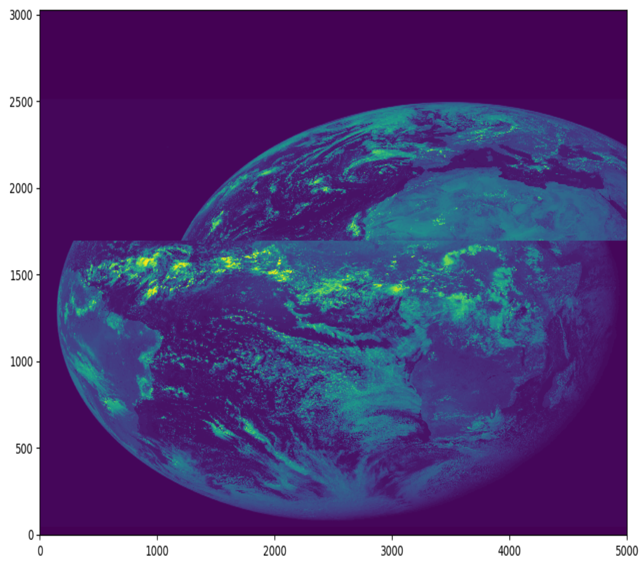


Figure 8 Example (METEOSAT2-MVIRI-MTP10-NA-NA-19810816140000) of the misaligned anomaly

Before the algorithm is executed, the following pre-check is executed:

- File exists
- Number of subimages is 1
- IR-channel is valid
- RECORD[3]["LINMID"] > 1200 and RECORD[3]["LINMID"] < 1300

The following calculations needs to be done, to detect the misalignment of the east and west horizon:

```
IR_SUB_IMAGE_0 = GET_FIRST_SUBIMAGE(IR) // see section 5.2.2
```

```
IR_EARTH = IR_SUB_IMAGE_0 > 20 // segmentate the Earth
IR_EARTH_FILT = MORPH_OPEN(IR_EARTH, KERNEL_3X3) // remove pepper-salt noise
FOR Y in RANGE( IR_EARTH_FILT.SHAPE[0])
    SCANLINE_EAST_FILT = IR_EARTH_FILT[Y,FIRST:1100]    // select the left side of the Earth
    SCANLINE_WEST_FILT = IR_EARTH_FILT[Y,1400:END]      // select the right side of the Earth
    HORIZON_EAST = WHERE(SCANLINE_EAST_FILT)[FIRST] // get east horizon
    HORIZON_WEST = WHERE(SCANLINE_WEST_FILT)[LAST] // get east horizon
    CENTER = (HORIZON_EAST + HORIZON_WEST) / 2
    IF CENTER < minPositionFoundCenterEarth OR CENTER > maxPositionFoundCenterEarth THEN
        Anomaly ➔ EastWestHorizonMisAligned
```

, where **minPositionFoundCenterEarth** and **maxPositionFoundCenterEarth** are minimum and maximum expected found center respectively and the defaults values are 1200 and 1500 pixels.

Table 24 East west horizon misaligned anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	VerifyEastWestHorizonAlgorithm
anomaly_type	EastWestHorizonMisAligned
sig	ALL

5.1.10 Hot Pixel Pattern 1

Figure 9 shows an example of the hot pixel pattern 1 anomaly. In general 4 hot pixels within 1 scanline are visible with approximately a distance of 20 pixels. This anomaly only appears in the WV-channel of satellite M4, M5, M6 and M7.

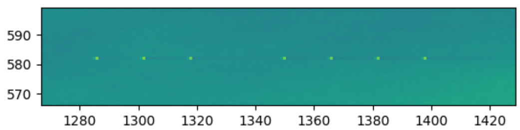


Figure 9 Example (METEOSAT6-MVIRI-MTP10-NA-NA-19980117063000) of hot pixel pattern in 1 scanline

Before the algorithm is executed, the following pre-check is executed:

- File exists
- Number of subimages > 0
- Channel is WV
- Sat-id is M4, M5, M6 or M7

Figure 10 shows the algorithm for detecting the hot pixel pattern 1 anomaly.

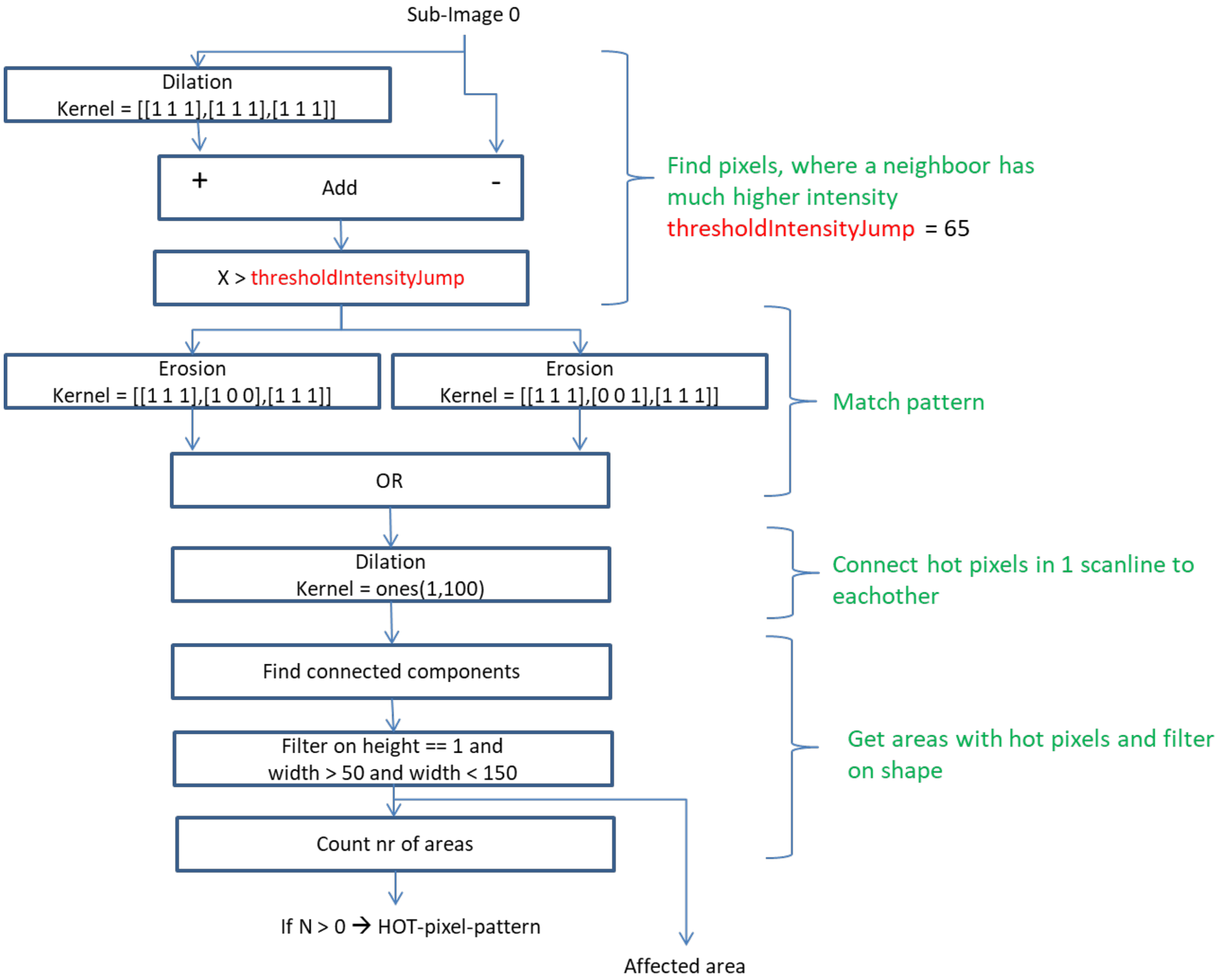


Figure 10 Processing pipeline for detecting the hot pixel pattern 1 anomaly

Table 25 Hot pixel pattern 1 anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	`HotPixelPattern1Algorithm`
anomaly_type	`HotPixelPattern1`
sig	WV
locus_type	BBox
locus	Affected areas ([x0, y0, x1, y1])

5.1.11 Hot Pixel Pattern 2

Figure 11 shows an example of the hot pixel pattern 2 anomaly, where each channel at the same position has some kind hot pixel pattern. The pattern is different in the various channels, but the affected pixels clearly from the surrounding pixels. This anomaly has the same behaviour as the so-called "fishes" anomaly and therefore both anomalies are detected by the same algorithm.

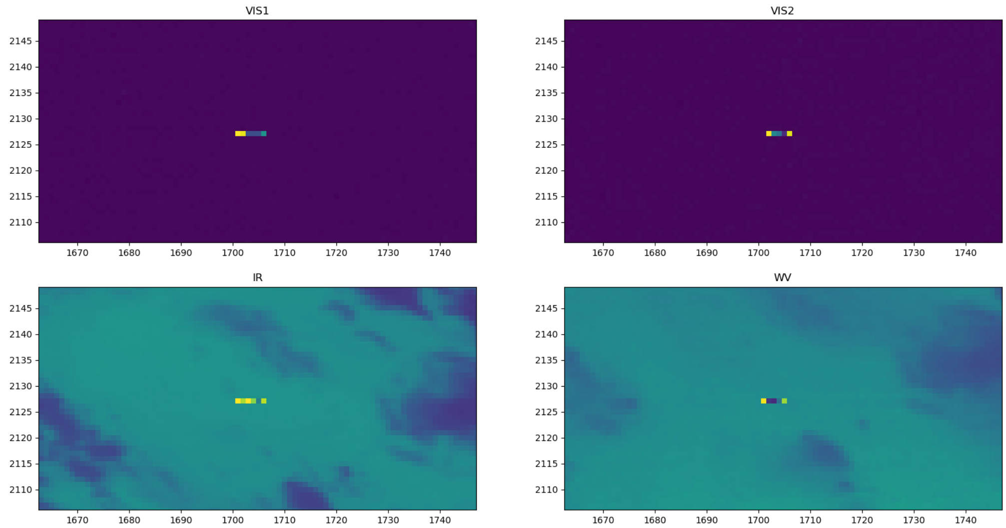


Figure 11 Example (METEOSAT5-MVIRI-MTP10-NA-NA-20000617190000) of hot pixel pattern anomaly. The anomaly is visible in all signals at the same position.

Before the algorithm is executed, the following pre-check is executed:

- File exists
- Number of subimages > 0
- Number of scanlines >= 100

Figure 12 presents a processing pipeline for detecting the hot pixel pattern 2. The algorithm for detecting vertical jumps in an image is defined in Figure 13.

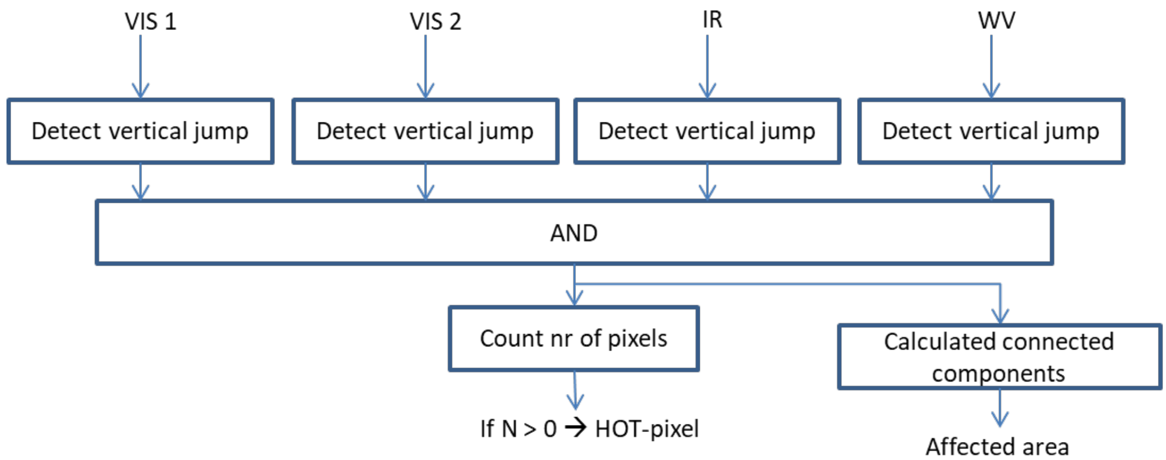


Figure 12 Algorithm for detecting anomaly "hot pixel pattern 2 in scanline". The algorithm to "detect vertical jump" is defined in Figure 13.

This basic approach requires a valid image for all channels, but unfortunately occasionally a detector is not used. In those cases, such hot pixel pattern can't be detected. To overcome this issue, an invalid channel can be ignored in some cases. The following strategy is applied for when the AND-operation is used:

- IR=valid, WV=valid => result = AND(VIS1,VIS2, IR,WV)
- IR=invalid, WV=valid => result = AND(VIS1,VIS2, WV)
- IR=valid, WV=invalid => result = AND(VIS1,VIS2, IR)
- IR= invalid, WV=invalid => result = False

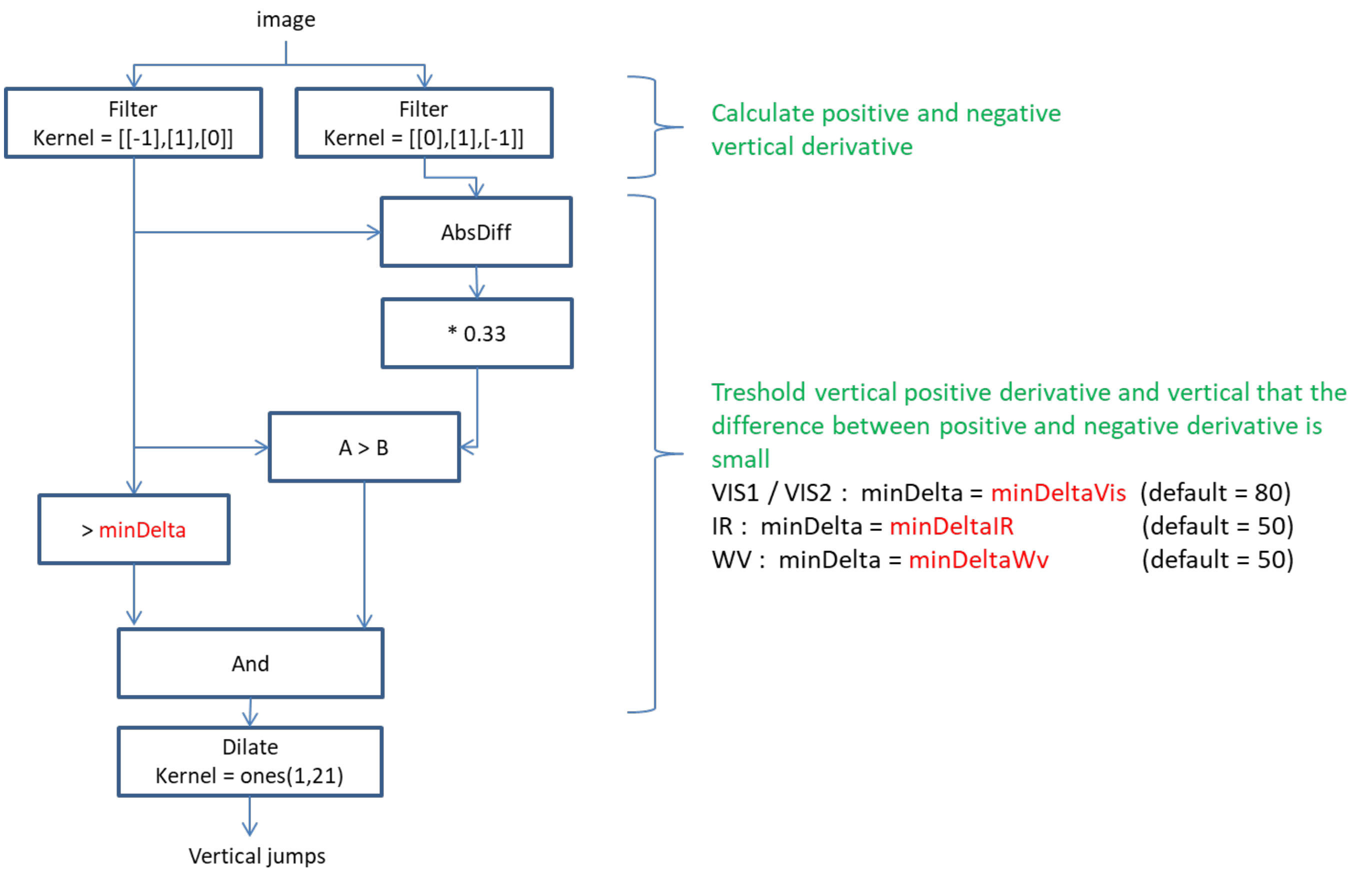


Figure 13 Algorithm for detecting vertical jumps in an image

Table 26 Hot pixel pattern 2 anomaly: relevant parameters of the anomaly descriptor

Parameter	**Value / Description**
alg	HotPixelPattern2Algorithm
anomaly_type	HotPixelPattern2
sig	VIS1, VIS2, IR, WV
locus_type	BBox
locus	Affected areas ([x0, y0, x1, y1])

5.1.12 Independent Hot Pixel

Occasionally, a single hot pixel is visible in a channel and this type of anomaly is called the independent hot pixel anomaly. This type of anomaly can be detected by comparing the intensity of pixels with neighbouring pixels. If this intensity-difference is larger than **minIntensityDiff** (default = 100), it is detected as an "individual hot pixel". Figure 14 presents the processing pipeline for this detection algorithm.

With some images, the number of hot pixels can be very large (> 10000) and the list of affected areas will become very large. To reduce the number of affected areas, it will try to combine hot pixels, which are very close to each other by a morphological closing operation.

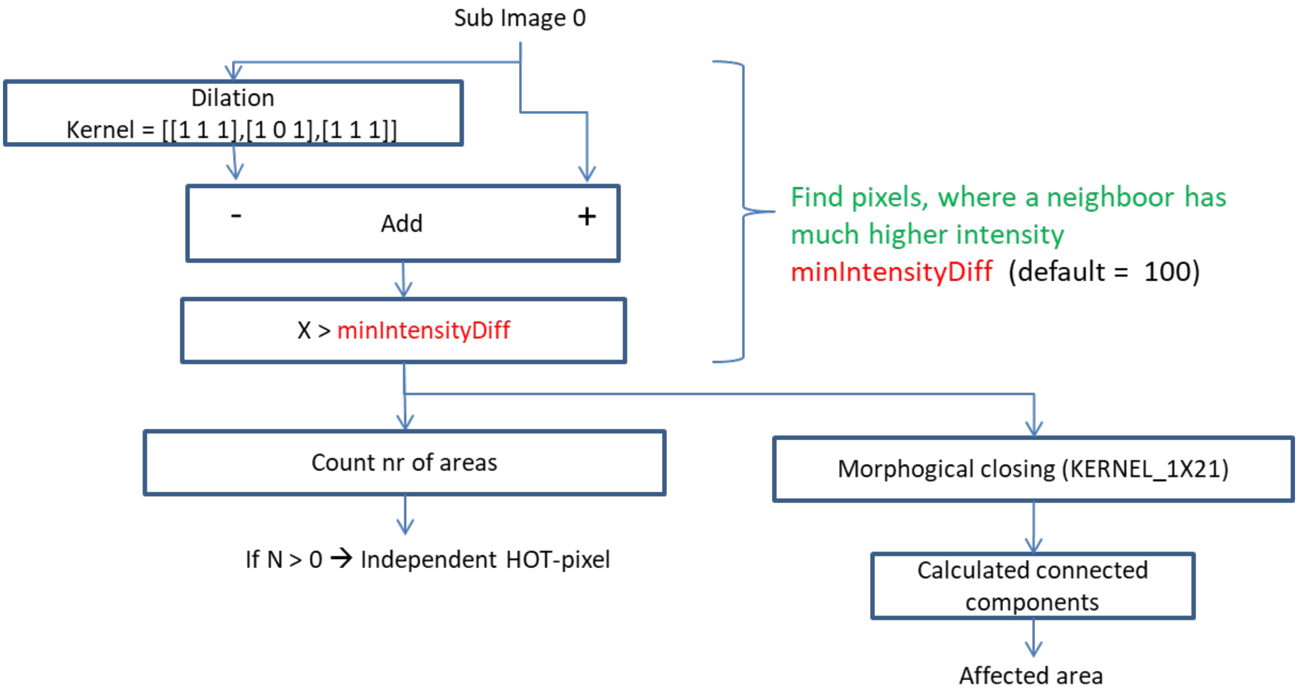


Figure 14 Processing pipeline for detecting independent hot pixels

Table 27 Independent hot pixel anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	HotPixelIndependentAlgorithm
anomaly_type	HotPixelPatternIndependent
sig	VIS1, VIS2, IR, WV
locus_type	BBox
locus	Affected areas ([x0, y0, x1, y1])

5.1.13 Extreme Over Illumination

The detectors of Meteosat 2 and 3 shows incorrect behaviour with extreme over-illumination: over-illuminated pixels get the value 124. Figure 15 shows an example of this incorrect behaviour.

The value of 124 is a normal allowed value and we cannot simply search for pixels with the intensity 124. We have to search for pixels with intensity 124 and which are next to pixels with intensity 252.

Before the algorithm searches for areas with over-illuminated pixels, the following pre-check is done:

- File exists
- Number of subimages = 0
- Channel is valid
- Sat-id is M2, M3

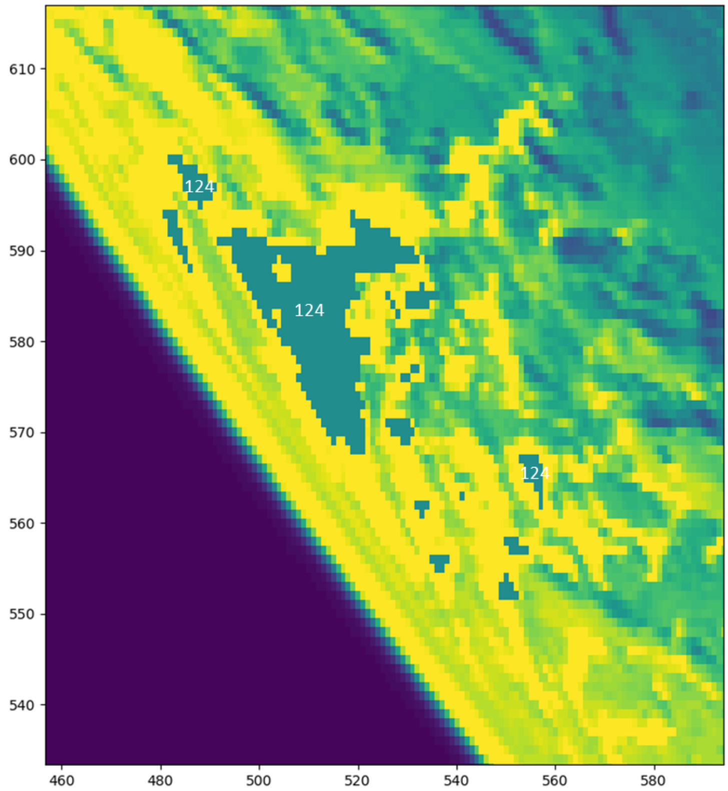


Figure 15 Example (METEOSAT2-MVIRI-MTP10-NA-NA-19831117020000 VIS2) of extreme over-illumination, where over-illuminated pixels get the intensity of 124.

Figure 16 shows the processing pipeline for finding over-illuminated pixels. This algorithm cannot detect over-illuminated pixels with perfect detection accuracy, because the neighboring pixels with intensity 124 and 252 might be unexpected but it is valid and it can occur. In practice, we see that this detection algorithm is reliable and that the probability on false-positive is relatively small.

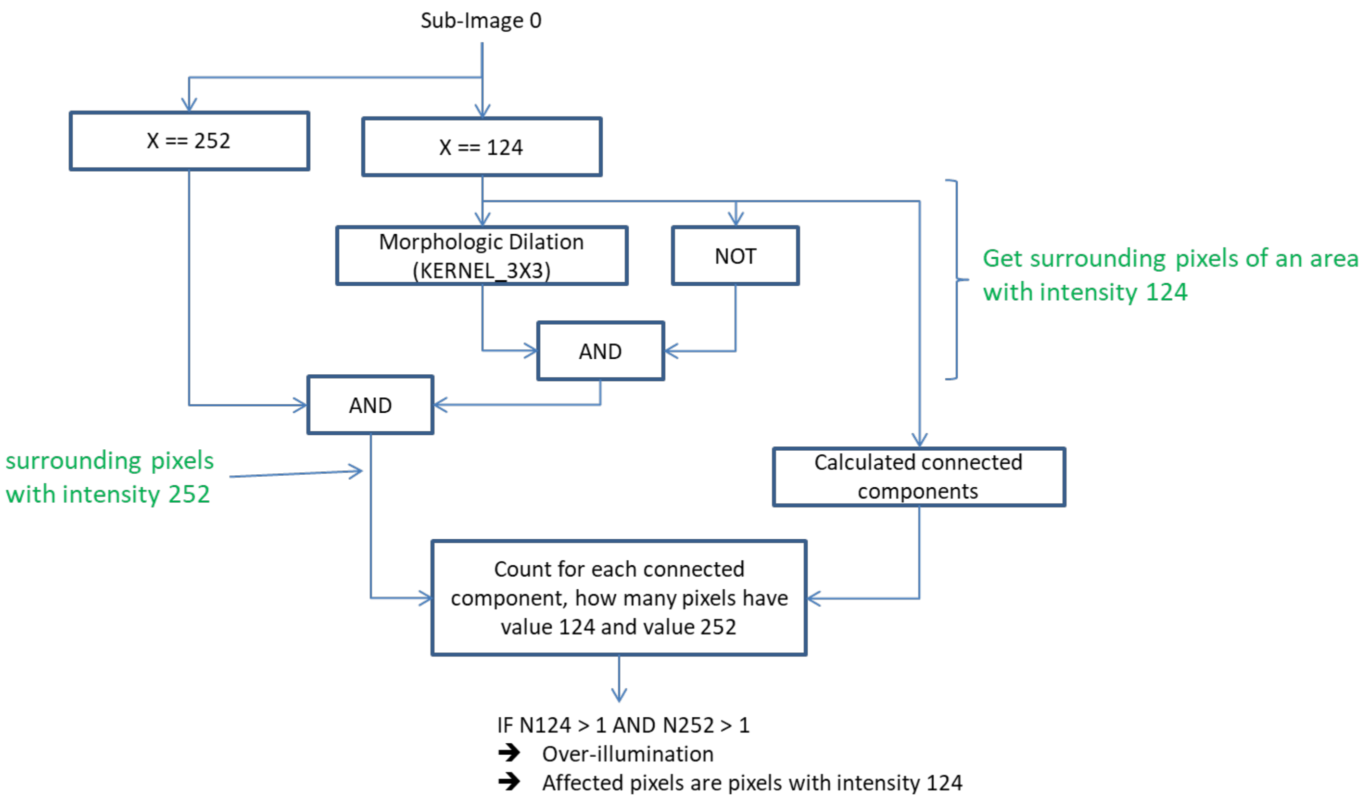


Figure 16 Processing pipeline to detect over-illuminated pixels (intensity 124), which have at least a surrounding pixel with intensity 252

Table 28 Extreme over illumination anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
alg	OverIlluminationAlgorithm
anomaly_type	OverIllumination
sig	VIS1, VIS2, IR, WV
locus_type	BBox
locus	Affected areas ([x0, y0, x1, y1])

5.1.14 Low SNR Scanline

Figure 17 shows an example with the low SNR scanline anomaly. With this anomaly, some scanlines have a significantly more noise than other scanlines in the same image. With a 2D-Gaussian derivative filter, it is possible to calculate some kind of measure of the noise. Figure 17 shows an example result of calculating the local noise level.

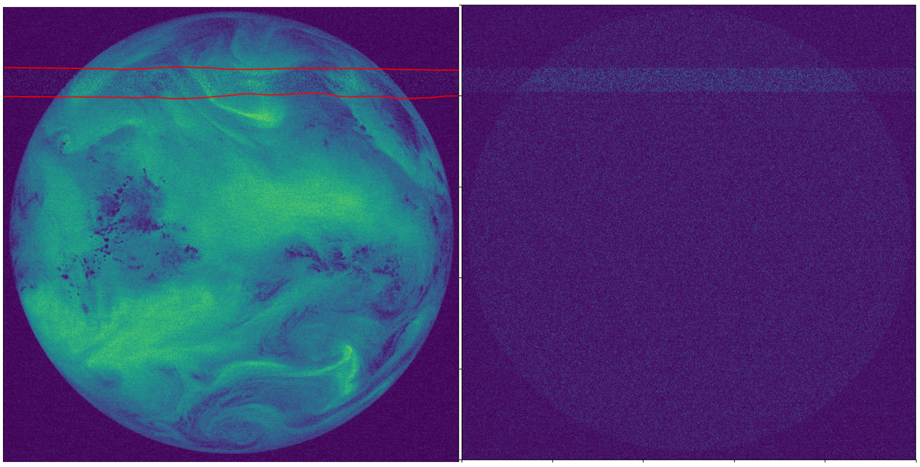


Figure 17 Left: WV-signal of METEOSAT2-MVIRI-MTP10-NA-NA-19831016120000, where some scanline have clear a higher noise level than the other scanlines. Right: Calculate local noise level. The scanlines with higher noise level are clearly visible.

For each scanline the average noise level can be calculated. Figure 18 shows in blue the average noise level for each scanline. From this example the detection looks very easy, but the noise level of an image depends on the satellite and the content of the image. Therefore the acceptance level is calculated on basis of the measured noise level of an image, by fitting a 2nd order polynomial through 90% of the data. The polynomial is fitted through 90% of data, because otherwise the incorrect scanlines will influence the acceptance level. Samples, which belong to 5% with lowest or the highest noise level are excluded for the fitting. The acceptance level is calculated by applying an offset of 2 to the 2nd order polynomial and it results in the red line of Figure 18.

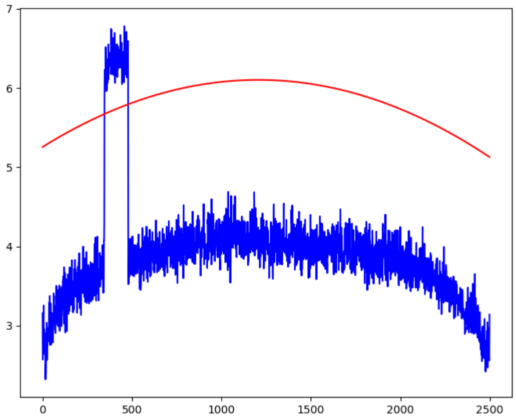


Figure 18 Blue: average noise level for each scanline. Red: maximum allowed noise level, which is determined by fitting a 2nd order polynomial.

Figure 19 shows the entire processing pipeline for detecting scanlines with a high noise level.

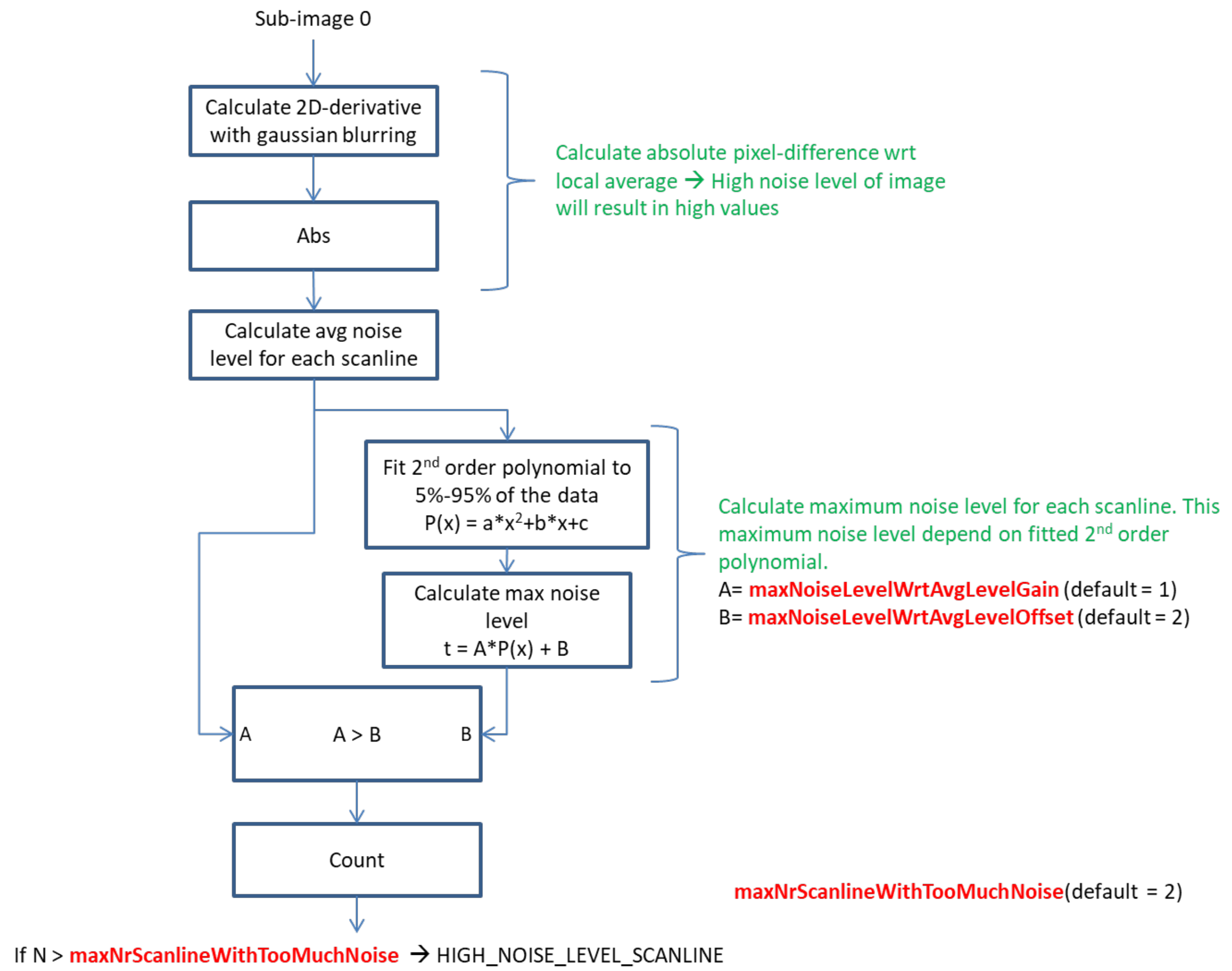


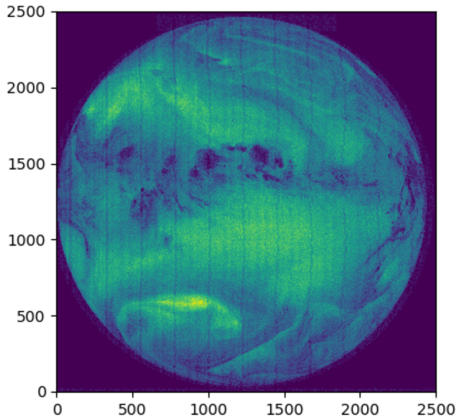
Figure 19 Algorithm for detecting the low SNR scanline anomaly

Table 29 Low SNR scanline anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	LowSNRScanlineAlgorithm
anomaly_type	LowSNR_Scanline
Sig	VIS1, VIS2, IR, WV
locus_type	Scanline
Locus	Affected scanline, where the calculated noise level is higher than calculated maximum

5.1.15 Suspicious pattern

Figure 20 shows two examples of the suspicious pattern anomaly. In this image, some kind of repetitive suspicious pattern is noticeable. This anomaly only appears in WV-images and the WV-images from M4 suffer most from this issue.



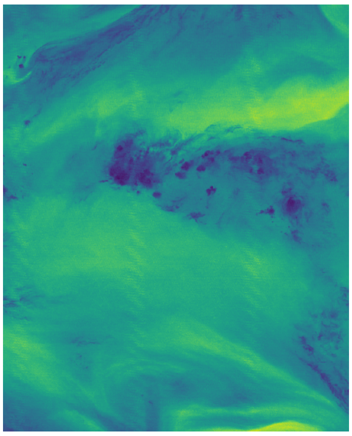


Figure 20 Examples with the suspicious pattern anomaly. Left: M2-MTP10-19810817223000 WV. Right: M4-MTP10-1990101501300

The variety in the appearance and the severity of the suspicious pattern is quite large and therefore we cannot for search for a typical pattern to detect this anomaly. To detect the anomaly, we have to search for "eye-catching" differences compared to what is "normal". To following this approach, we have to first determine what is normal. The most straight forward domain, to do this analysis is the Fourier-domain.

Figure 21 shows the average FFT (only the left-side of the 2D-FFT) over 150 WV-images, which did not contain any severe anomaly. The 150 selected WV-images were equally divided over all satellite, so from every satellite 25 WV-images. The FFT is calculated only from the forward-scan (height is 2500 scanlines).

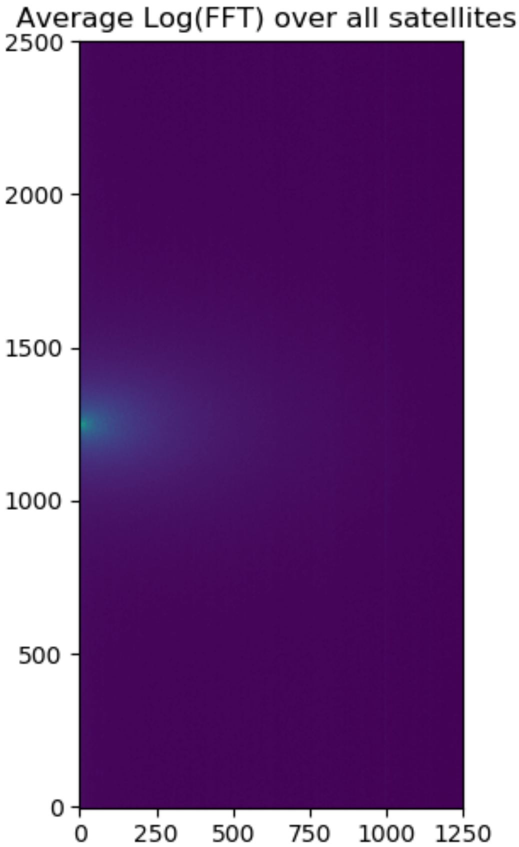


Figure 21 Average FFT over 150 WV-images, which did not have any sever anomaly.

The average FFT over all satellites seems to be quite smooth and without any strange peaks. But we can see large differences between the satellites. Figure 22 shows the relative average FFT of every satellite and we can see large differences. M2 and M3 have much more white noise and with the other satellites, we can see some strange peaks in the Fourier domain. Because the differences between the satellites are quite large, each satellite will get its own definition of normal behaviour.

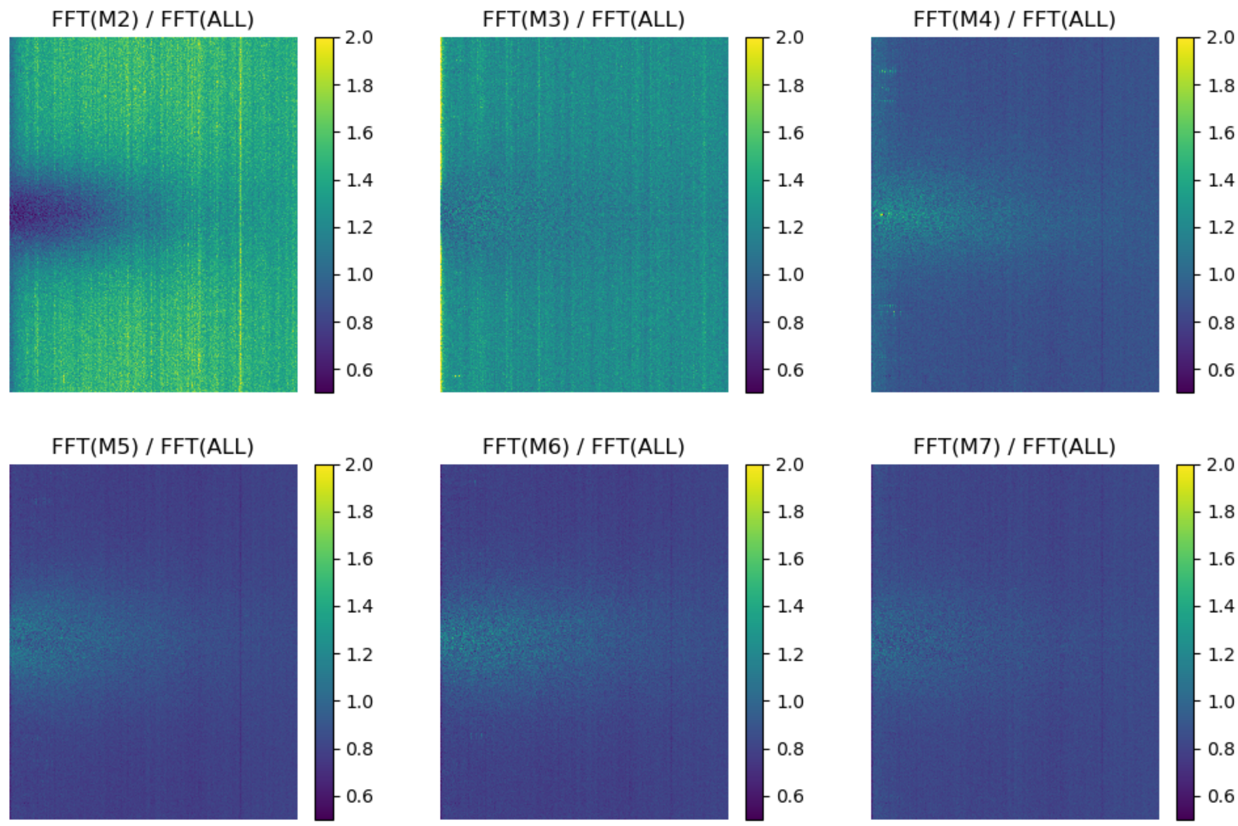


Figure 22 Relative average FFT of every satellite

Before the algorithm can detect the suspicious pattern anomaly, the following pre-check is done:

- File exists
- Number of subimages is 1
- Number of scanlines is 2500
- WV-channel is valid

Figure 23 shows the processing pipeline for detecting the suspicious pattern anomaly. This detection approach is inspired by trying to reconstruct an image without the pattern (peaks in the Fourier-domain) and calculating the severity of the pattern. The severity of the pattern is defined as the calculated maximum intensity of the pattern. This estimate is also used to determine if the detected pattern is severe enough and if an anomaly is detected. Currently the maximum intensity of the reconstructed disturbance pattern should be smaller than 1 (**maxMagnitudePattern**).

To detect peaks in the Fourier-domain, it is expected that the magnitude at the peak is at least a factor 10 (**minRatioFftPeak**) larger. With a smaller ratio, the peak detection will be more sensitive, but it will also result in much more false positive detections.

Because the WV-image of the M2 contains much more white noise and therefore it was required to increase the reference Fourier-spectrum with additional noise (**additionalNoiseM2**). Without this additional noise for M2, there were too many false positive detections.

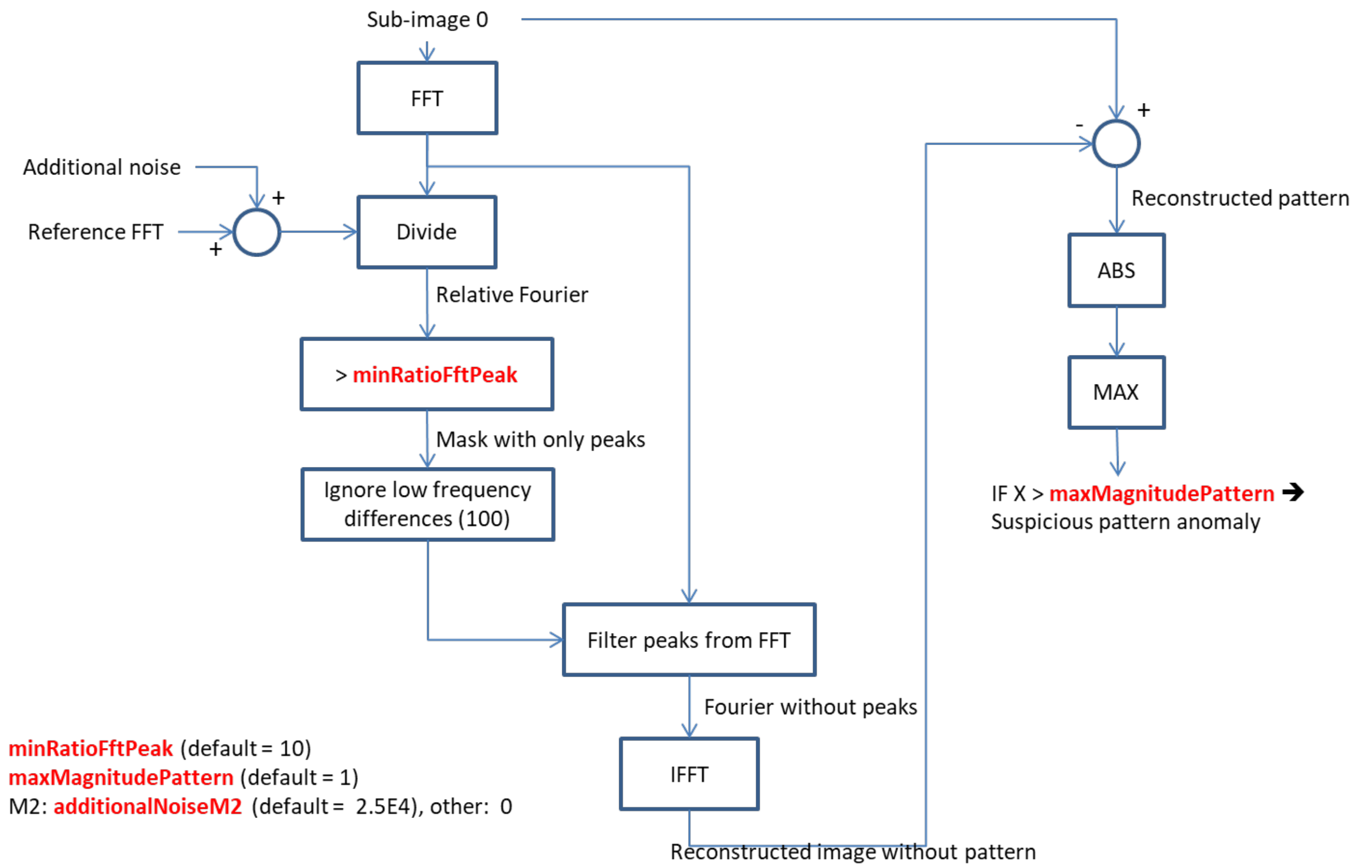


Figure 23 Processing pipeline for detecting the suspicious pattern anomaly

Figure 24 shows some intermediate results of the processing pipeline. In the Fourier domain clearly some peaks are visible, which are also detected by the algorithm. The reconstruction of the image without the pattern is far from perfect, but the disturbance pattern is already reduced. The difference between the reconstruction and the original image has the appearance of the expected pattern.

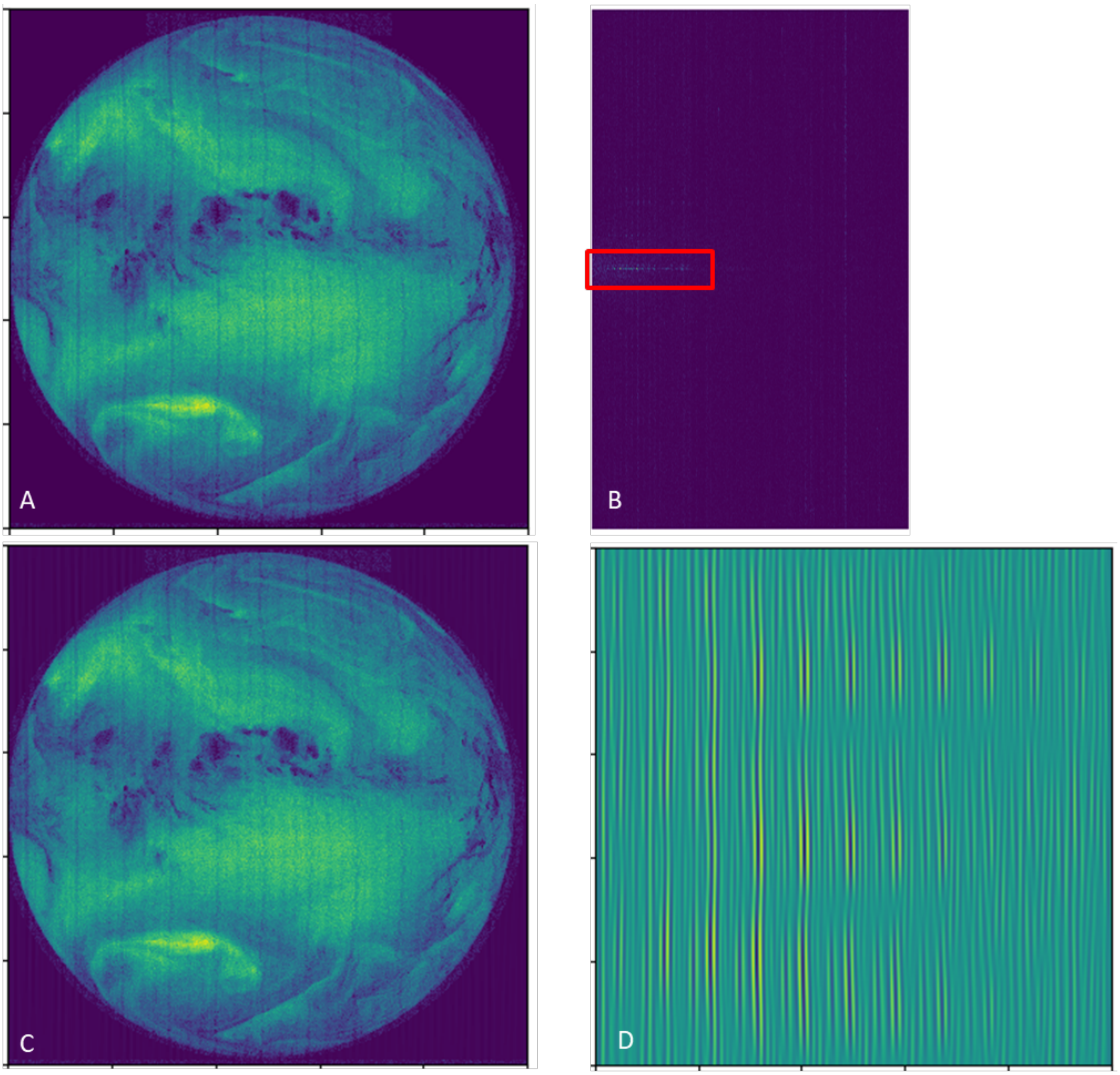


Figure 24 A: Original WV-image (M2-MTP10-19810817223000). B: Relative FFT, where peaks on the horizontal axis are noticeable. C: reconstructed image without the detected peaks in the Fourier domain. D: reconstructed disturbance pattern.

Table 30 Suspicious pattern anomaly: relevant parameters of the anomaly descriptor

Parameter	**Value / Description**
Alg	SuspiciousSpectrumAlgorithm
anomaly_type	SuspiciousSpectrum
Sig	WV

5.1.16 Tilted Line

Figure 25 shows an example of the tilted line anomaly; where in the WV-image a line under an angle is visible. The pixels belonging to this line have a slightly higher intensity than the surrounding pixels. This line is approximately under the same angle and position.

The root cause of this anomaly is unknown, but it is possible that there is electronic pulse approximately every 0.6 seconds (time between scanlines) and via crosstalk it causes an increased intensity.

A well-known approach to find lines in an image is the usage of the Hough transform. This approach is also applied to find this type of anomaly. First the line should be segmented in the image. This is achieved by calculating the derivative in the perpendicular direction of the line, and pixels with a large derivative are segmented. By applying morphological operations, the line can be emphasized by removing areas with a horizontal width of minimal 3 pixels. Figure 25 shows a result of these processing steps (calculate derivative, thresh holding and morphological operations).

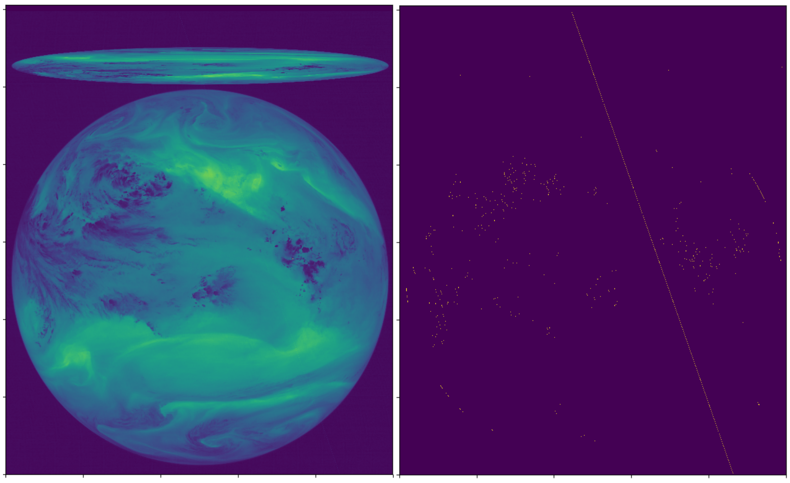


Figure 25 Left: WV-signal of METEOSAT5-MVIRI-MTP10-NA-NA-20010817120000. Right: segmented and emphasized (tilted) line.

With the Hough transform it can be determined for each pixel if it belongs to a certain line (y=ax+b). If many pixels can belong to the same line (y=ax+b), then it is likely that this line actually exists. In our case we want at least 2500 pixels to belong to a candidate line. The result of the Hough transform is often presented in a (2D) histogram and candidates are searched within this histogram. Figure 26 shows the (1D) histogram for a line under an angle of 1.9 rad.

This algorithm verifies various candidate lines (y=ax+b) and to reduce the calculation time and avoid false positives, we have limited the search range to an angle between [1.8 – 2.0].

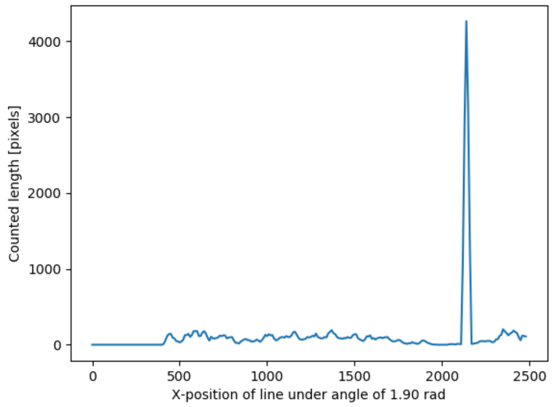


Figure 26 (1D) Histogram of candidate lines, which is calculated by applying the hough transform.

Figure 27 shows the entire processing pipeline for detecting the tilted line anomaly including the affected area.

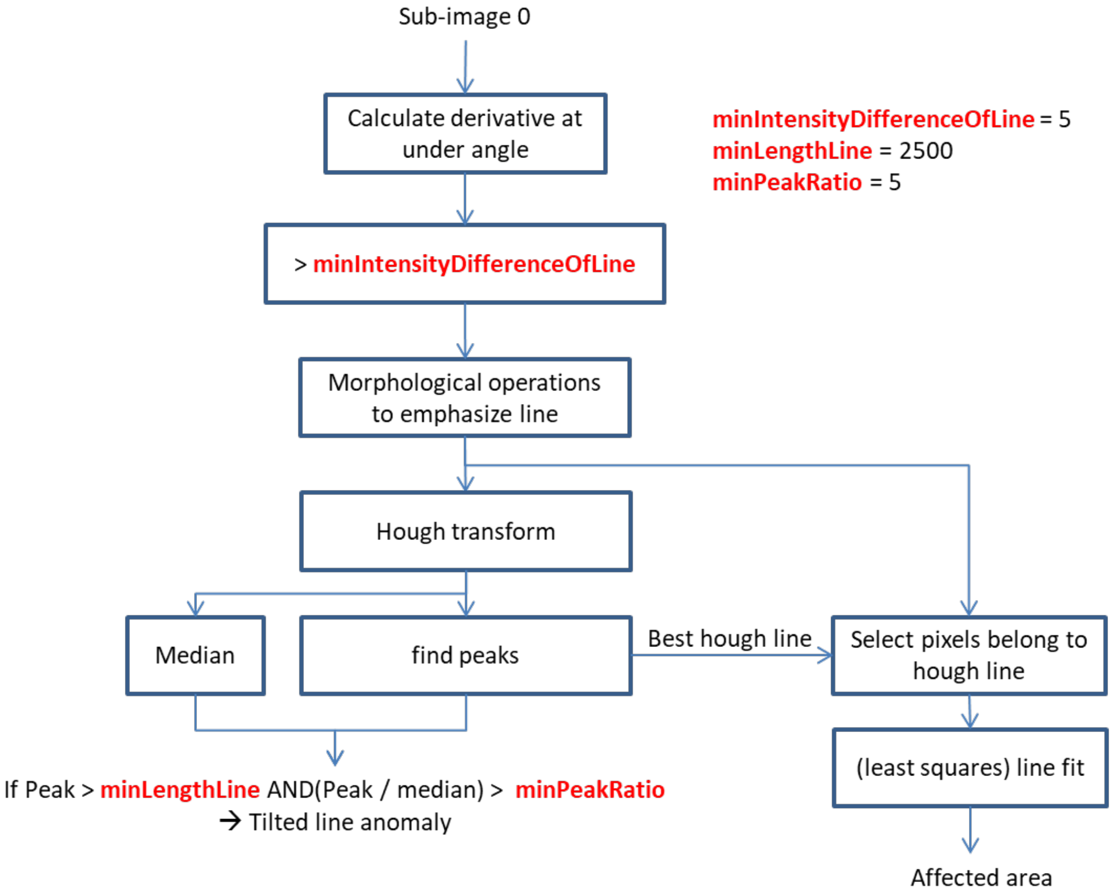


Figure 27 Algorithm for detecting a tilted line with usage of the Hough transform.

Table 31 Tilted line anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	TiltedLineAlgorithm
anomaly_type	TiltedLine
Sig	WV
locus_type	BBox
Locus	Affected areas ([x0, y0, x1, y1])

5.1.17 Direct stray light effect

Figure 28 shows an example of the well-known eclipse or direct stray light anomaly. With this anomaly indirect sunlight is picked up by all detectors, but it is mainly visible in the VIS1, VIS2 and WV-detector. This phenomenon occurs at night (when the sun is behind the earth). The SOW states the so-called eclipse season (around March 20 and September 23 with a length of 46 days). But on basis of the ground truth dataset, we can see that the direct stray light effect anomaly can appear every month, so no strict period.

Because it is very hard to define the exact appearance of the sun-glint, for detection we focus on the fast movement of the sun-glint. Figure 28 shows the WV-signal for three sequential timeslots and we can see that the sun-glint moves fast compared to the background.

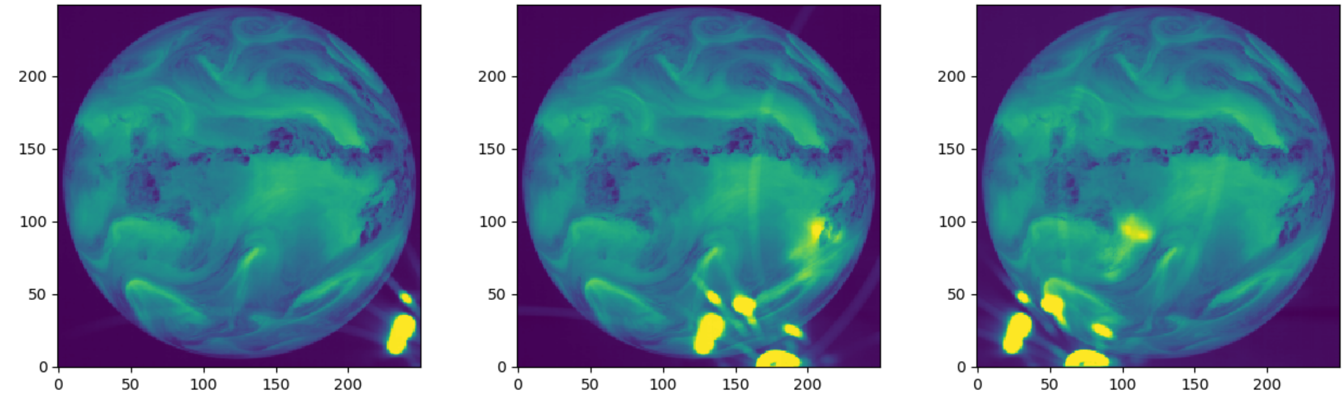


Figure 28 Three sequential WV-images (center image is METEOSAT5-MVIRI-MTP10-NA-NA-19961016000000), where the fast-moving sun-glint is clearly visible in the bottom part of the image.

The sun-glint can be detected by comparing the current image with the previous and next image and focus on the increase in intensity. We assume that the sun-glint causes an increase in the intensity and we can find this increase by subtracting image A from image B. Similar intensity increase should be found by subtracting image C from image B. If both subtracted results do show the same intensity, we have a good candidate for the sun-glint. Figure 29 shows this basic calculation for calculating the potential increase in intensity due to a sun-glint.

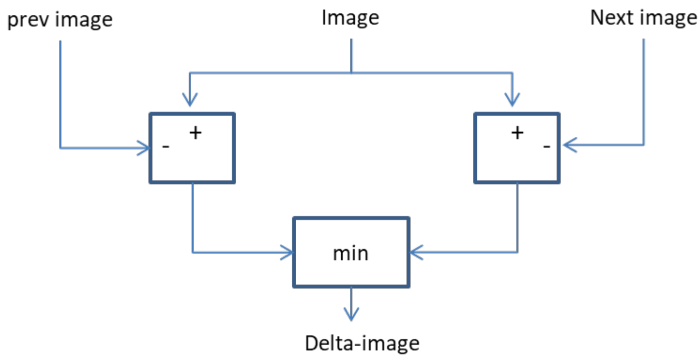


Figure 29 Conceptual algorithm for finding sun-glint in three images

The algorithm has the following pre-check:

- Channel is valid
- Number of scanlines is 2500

The algorithm requires a previous and a next image for the direct stray light anomaly detection. The previous- and next image should exist, be valid and have 2500 scanlines. If the previous image (or next image) with only one timeslot difference is not valid, if it verifies images in the past with maximally 5 timeslots difference. If no two valid related images exist, the direct stray light detection is stopped. It is important to have a reference image from the past and the future, because the temperature of the earth changes in time and with these reference images this effect is covered.

This conceptual algorithm could only work if the background is perfectly stationary (no movement of satellite or clouds). To cope with the moving satellite, the images should be registered (align images with each other). Level 1.5 images are registered, but the algorithm needs to work with level 1.0 images. The rectification mapping from a level 1.0 image to the level 1.5 image is via the deformation-matrix. This non-linear mapping is actually a one direction mapping and to calculate the inverse mapping is very challenging. Also the deformation matrix is not completely smooth and perfect, which can cause for artefacts in the inverse mapping.

The registration of the level 1.0 image does not need to perfect and we have chosen for an approximation by modelling the mapping between level 1.0 and level 1.5 image with the perspective homography mapping. The perspective homography mapping can calculate the displacement from one image to another image by using a 3x3 matrix. This mapping is a linear transformation and the inverse mapping can be calculated by using the inverse matrix. Figure 30 shows how the perspective homography mapping between level 1.5 and level 1.0 image can be estimated.

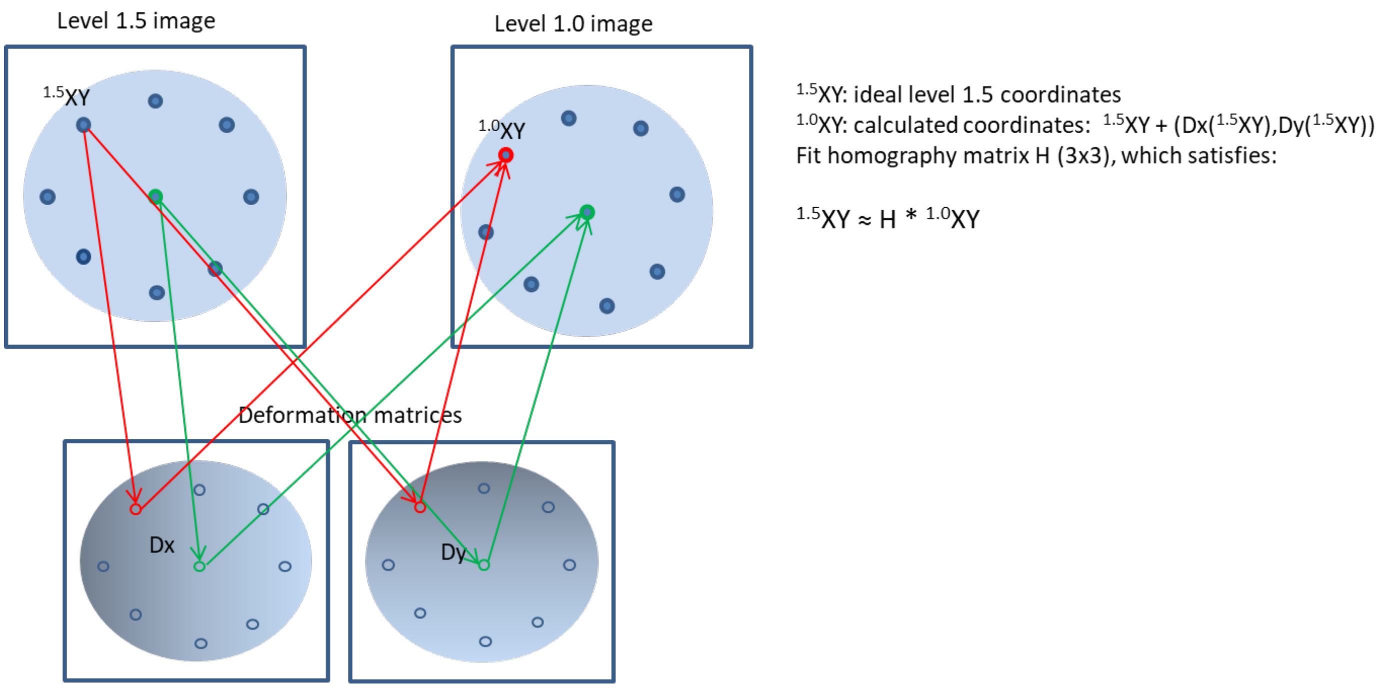


Figure 30 Approximation of the mapping from level 1.5 to level 1.0 image by the perspective homography mapping. From 9 level 1.5 coordinates, the corresponding level 1.0 coordinates are calculated with help of the deformation matrix. Between both sets of coordinates the perspective homography matrix can be estimated.

Figure 31 shows the registration process of three level 1.0 images to each other, which will be used by the detection of the direct stray light anomaly.

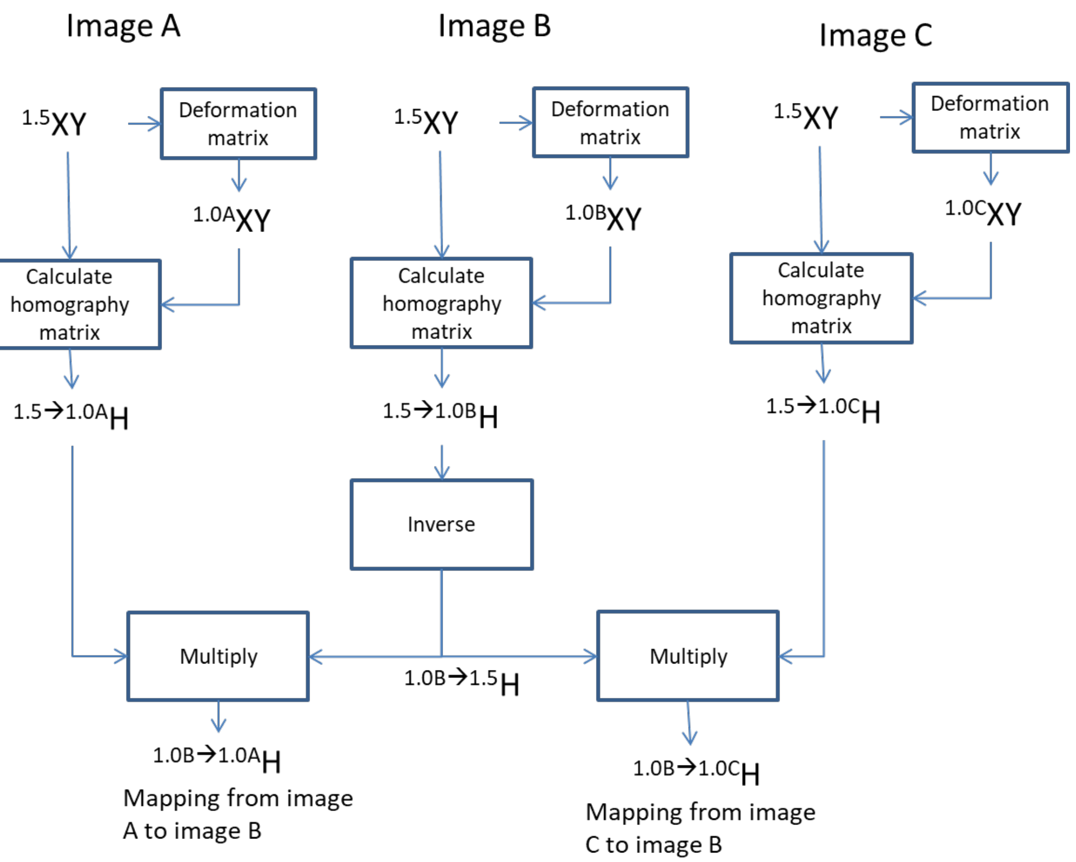


Figure 31 Registration process of three level 1.0 images to each other by applying the perspective homography mapping

If one or multiple deformation matrices are empty, the registration process is omitted. To verify the registration process and indirectly the deformation matrices, the alignment of the images is checked. Figure 32 shows the calculation for verifying and re-aligning two images with each other.

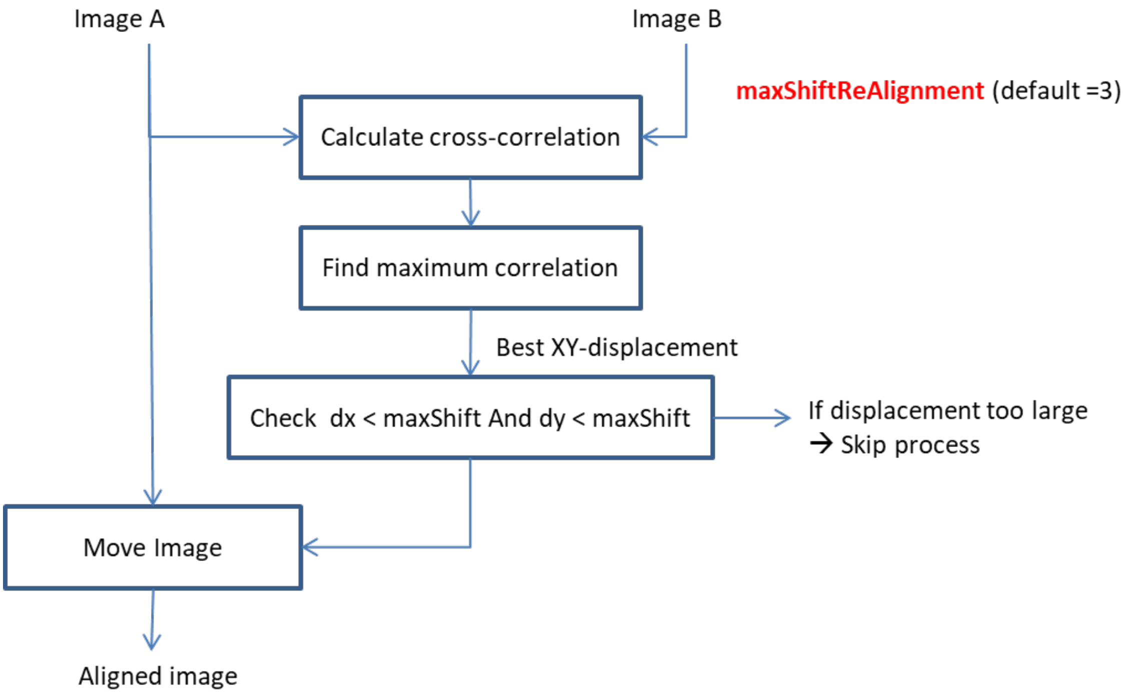


Figure 32 Process of verifying and re-alignment of two images with each other

Figure 33 shows the entire processing pipeline for detecting the direct stray light anomaly. The change detection algorithm estimates the intensity increase due to the direct stray light anomaly. If an area with an intensity increase larger than 10 (**thresholdStraylight**) then this area is likely to be affected by the anomaly. We have to note that for WV-images from M6, a larger threshold (**thresholdStraylight_M6_WV**) is used, because of the loose cold optics issue. If the area is large enough (0.1% of the image: **minPctAffect_areaForStraylight**) then the anomaly is detected. For determining the affected area a more sensitive threshold (6: **minIntensityDiffForaffectedArea**) is used.

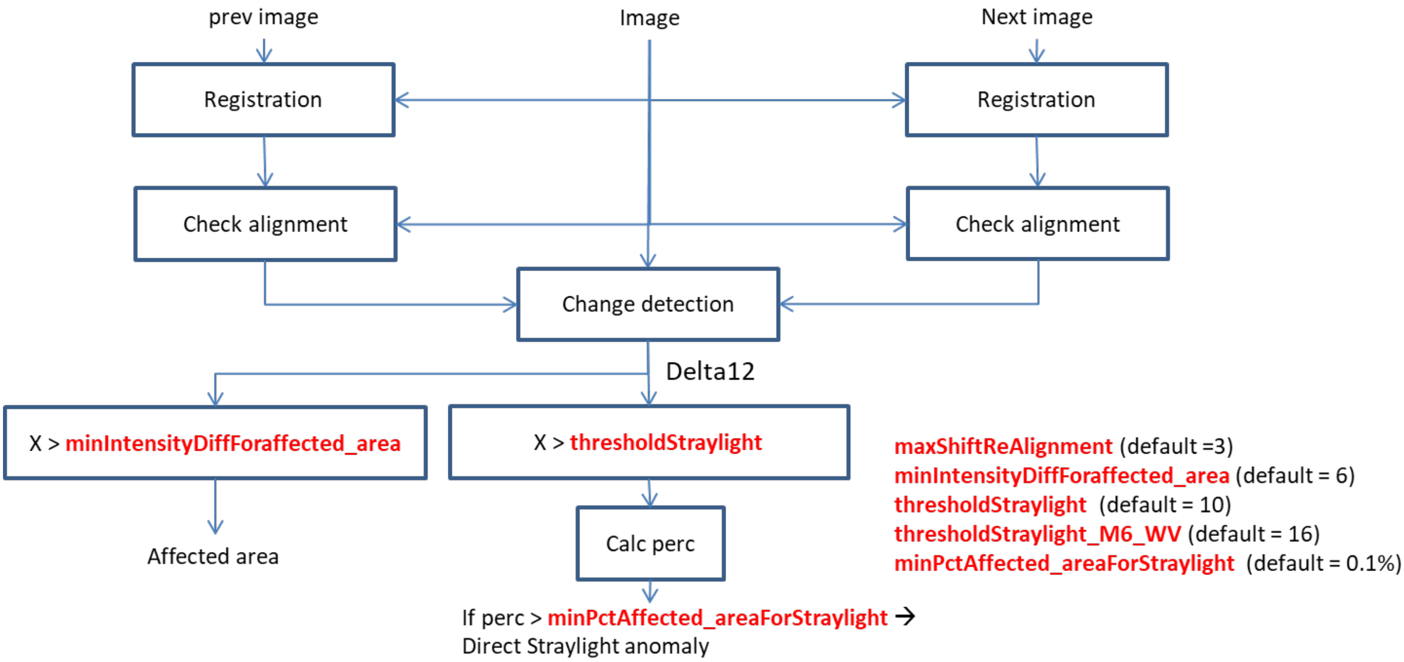


Figure 33 Processing pipeline for detecting the direct straylight anomaly in three consecutive images. The registration block is defined in Figure 31. The check alignment block is defined in Figure 32. The change detection block is defined in Figure 34.

The most crucial step in the algorithm is the change detection process, which is shown in Figure 34.

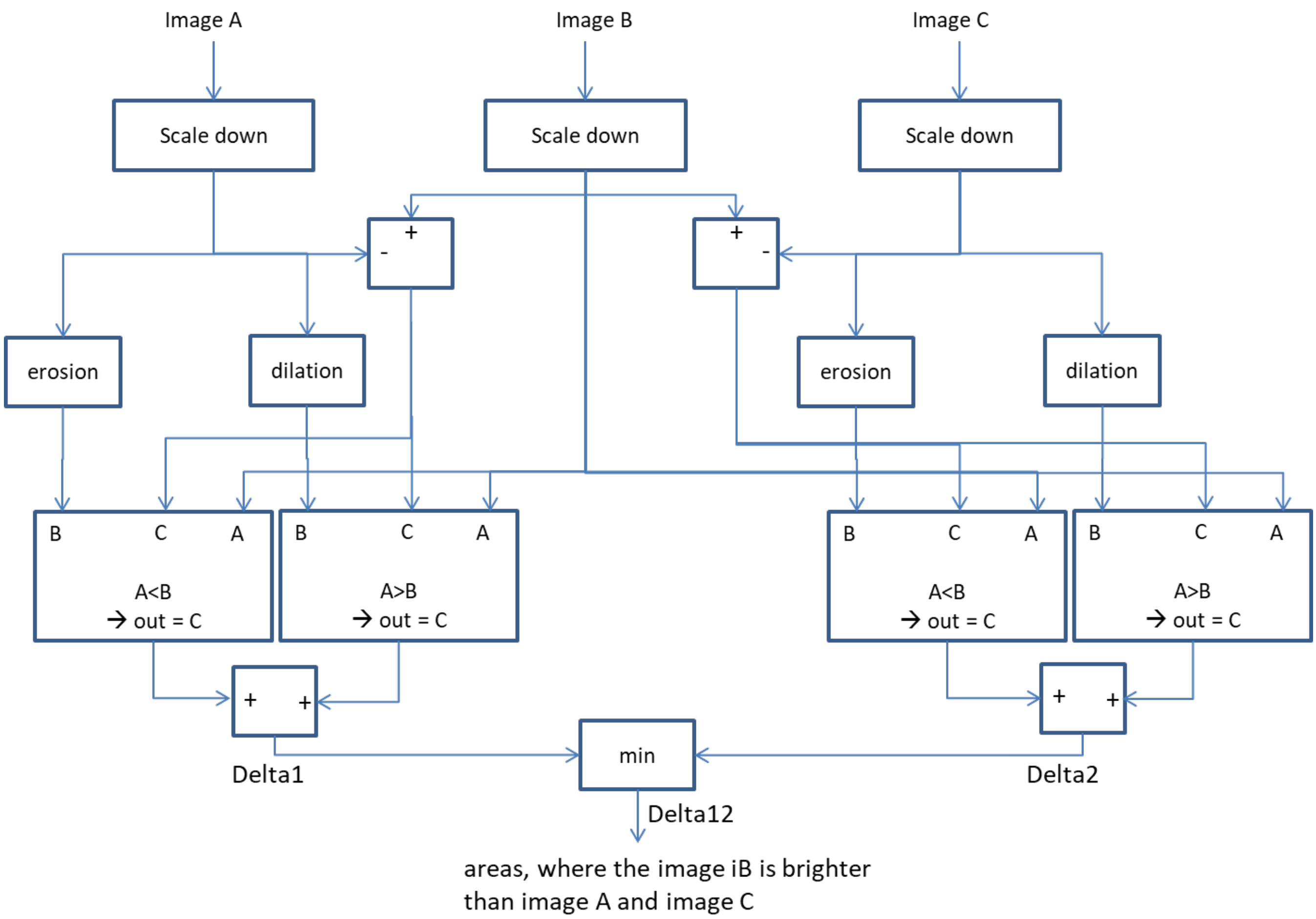


Figure 34 Processing pipeline of change detection to find pixels from image B (current image), which are brighter than from image A (previous image) and image C (next image).

Figure 35 shows an interesting intermediate result of the change detection algorithm. The segmentation is not perfect, but we can see that the stray light effect causes the bows (which can be recognized in the image) but also more areas with small intensity increase. Also we can observe, that after the appearance of the sun-glint, the captured image is locally much darker (with this example at the north pole of the last image).

The yellow pixels from Delta12 of Figure 35 are considered to be the affected pixels by the direct stray light effect. From this detection result, we have to conclude that the effects of this anomaly are quite complex. More pixels than the visible bow are affected, but some pixels from the visible bow do not have a significant error.

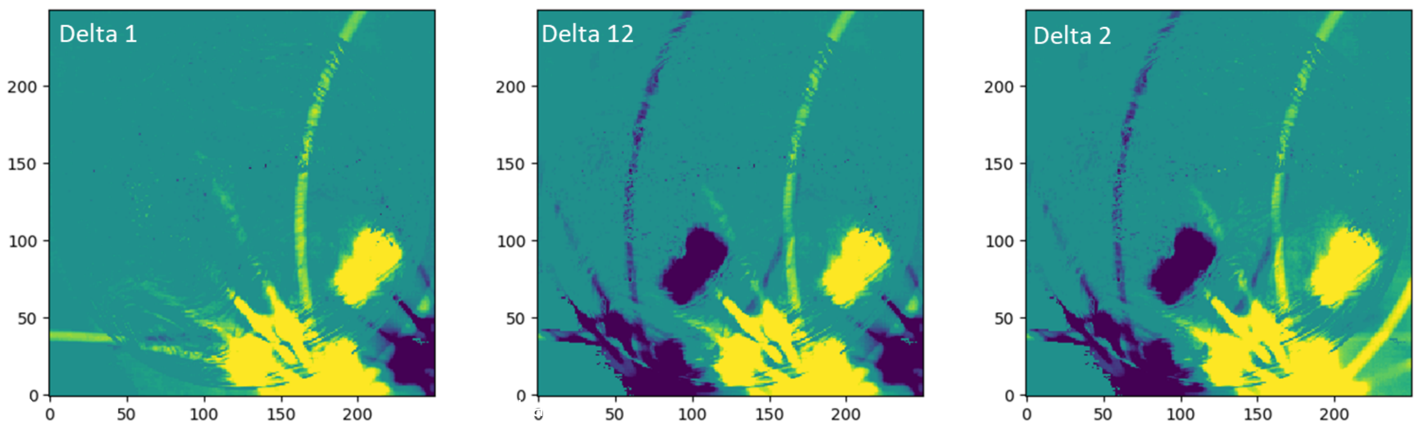


Figure 35 Intermediate results from the stray light effect algorithm. Pixels with a yellow color highlight where likely the sun-glint is. Pixels with a blue color highlight where the sun-glint was.

Figure 36 shows the estimated intensity increase due to direct straylight anomaly. The areas, where the sun-glint was more severe in the previous image or the next image, the segmentation is not perfect.

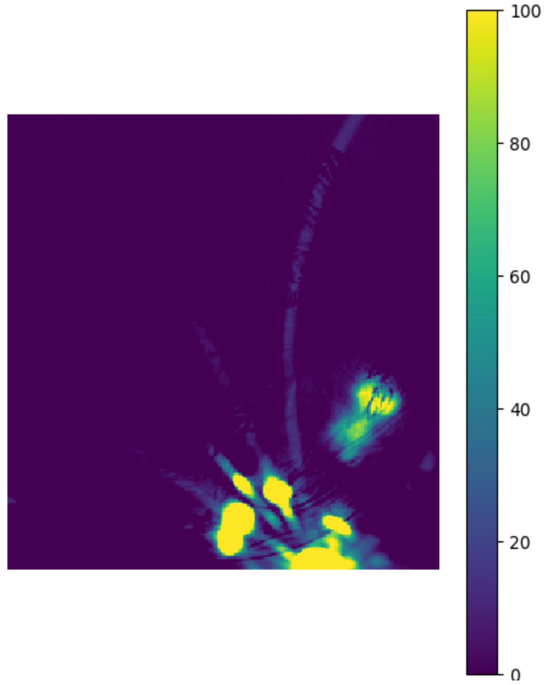


Figure 36 Estimation of the intensity increase due to the direct straylight anomaly.

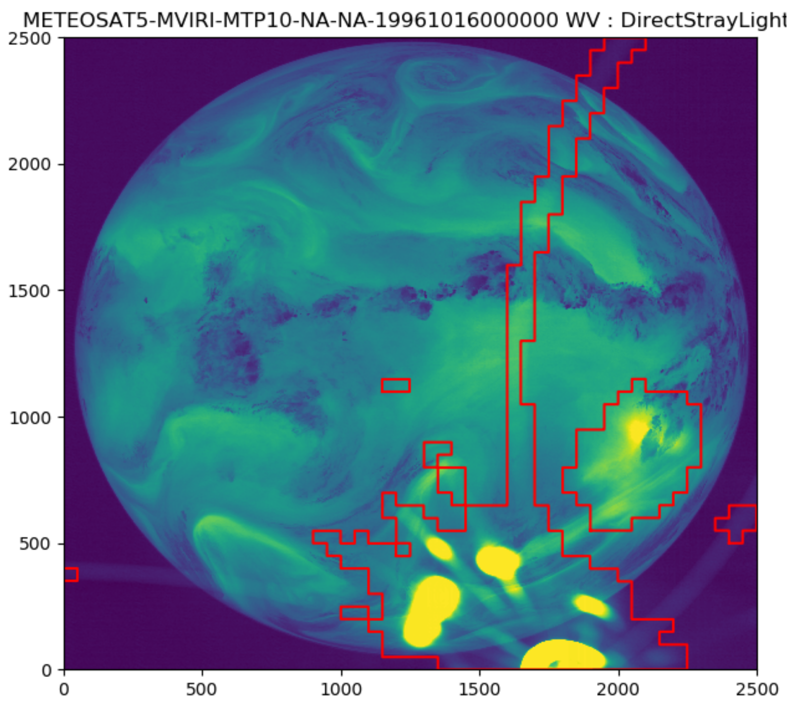


Figure 37 Detected affected area by the direct straylight anomaly

Table 32 Direct straylight anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	DirectStrayLightAlgorithm
anomaly_type	DirectStrayLight
Sig	VIS1, VIS2, IR, WV

locus_type	BBox
Locus	Affected areas ([x0, y0, x1, y1])

5.1.18 Indirect stray light effect

5.1.18.1 Phenomenon

Figure 38 shows an example of the indirect stray light effect. The first appearance of the stray light effect is in image (daynr=77, timeslot=3), where we can see a very bright band and a dark region around it. This effect (bright band with dark region) is also visible in images 77-4 and 77-5. In image 77-6 there is no anomaly (clearly) visible, but in images 77-7 to 77-11 a dark band is shown instead of a bright band. Note that the images (76-48, 77-1 & 77-2), where the actual sun glint should be visible, are missing.

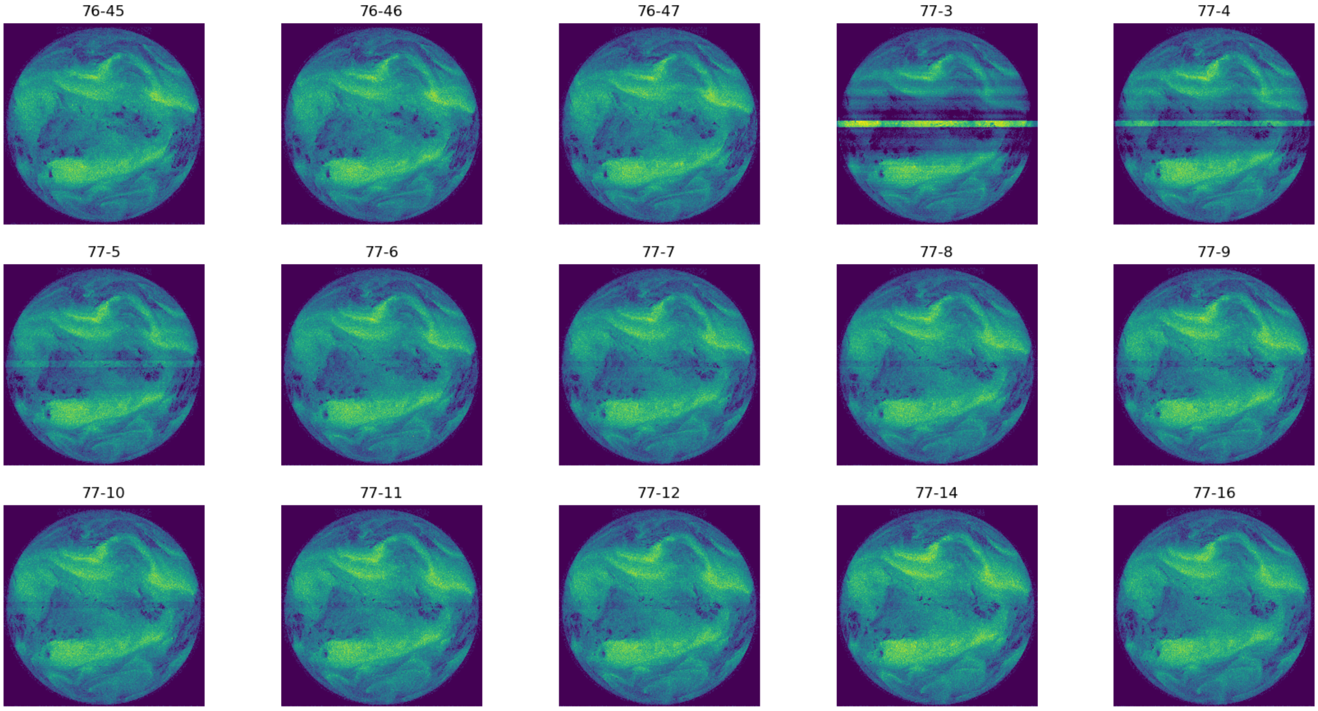


Figure 38 M2 WV-images from March 16th (daynr=76) and 17th (daynr=77) 1988. A sun glint is visible at timeslot 3 of day 77 and in the sequential images, the indirect stray light effects are visible.

Figure 39 shows for these WV-images, the scanline average intensity. We can see that for a large part of the images, the values are similar in all images. But for a particular region (scanline nr = [700-1850]), the average intensity differs between the images. Images 77-3 to 77-6 show a dark region with a bright band and image 77-7 to 77-10 show a dark band.

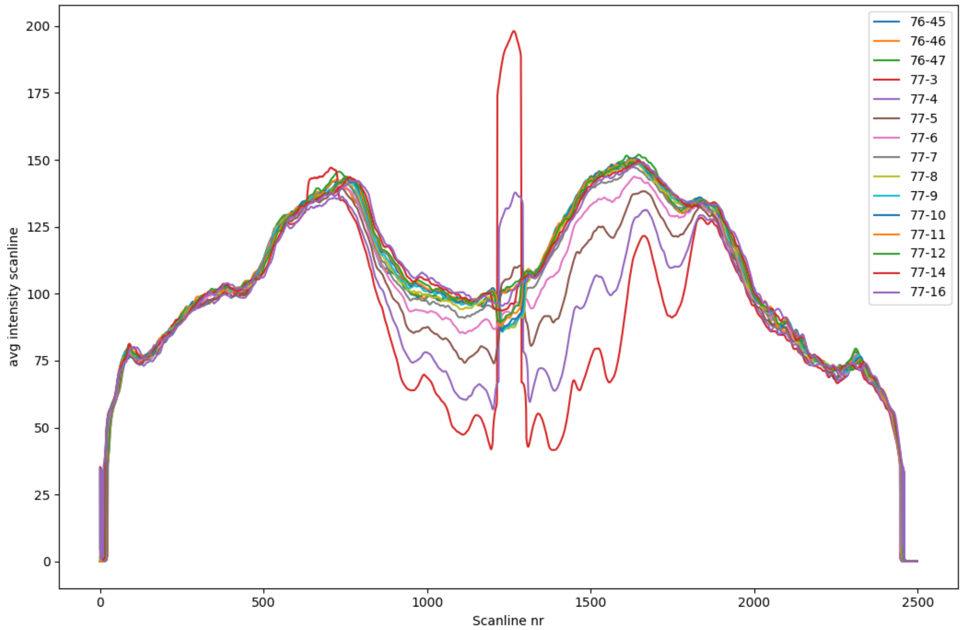


Figure 39 Scanline average intensity for M2 WV-images, where the indirect stray light effect anomaly is present.

Figure 40 focuses on three interesting scanlines of this example. We can see for all 3 scanlines some kind of impulse-response. The "duration" of this impulse-response is quite long (several hours) and we can also see some kind of overshoot. This behaviour (in time) is typical for this anomaly.

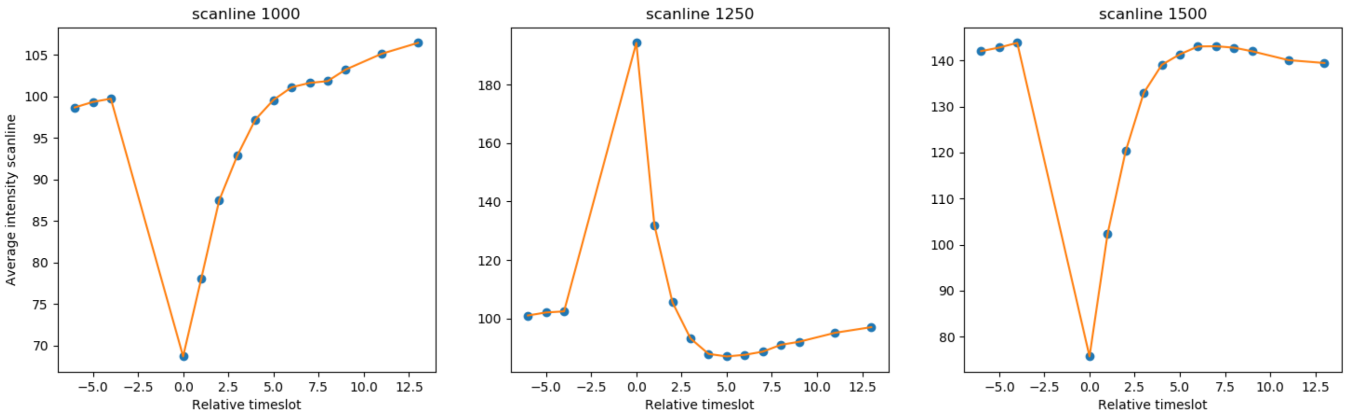


Figure 40 Average intensity for scanline 1000, 1250 and 1500 for M2 WV-images, where the stray light effect first appears at timeslot = 0 (= 77-3). Scanline 1250 is in the bright band and scanline 1000 and 1500 are in the dark region.

If we look for the change in time (for all scanlines) and apply the principle component analysis (PCA), we can get a result like in Figure 41. For this calculated PCA, the time-series are "centered" by subtracting the median. The major dimension of the PCA transformed space describes the "bias" due to this anomaly and the major axis of the PCA looks similar to the expected impulse-response.

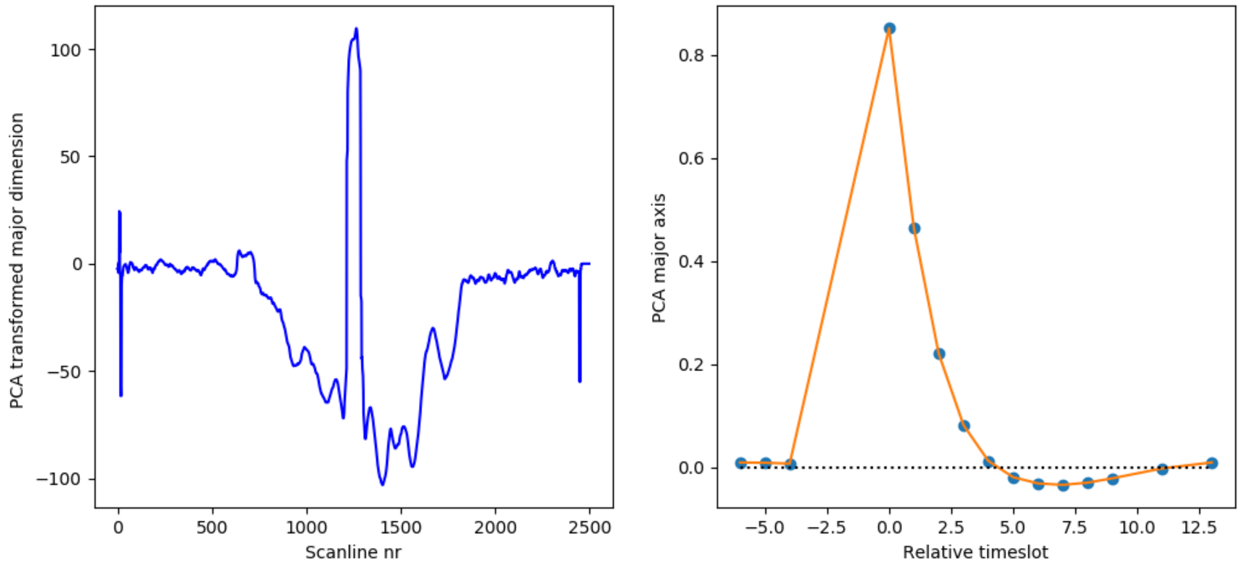


Figure 41 Result of principle component analysis. Left: first dimension of the PCA transformed space. Right: PCA major axis.

In general, images with a (very bright) sun glint are removed and with a "clean" consecutive set of images, an impulse response like Figure 41 could be found. We also know that in several cases, the direct stray light anomaly is in the data and that will influence the Principle Component Analysis. If an image with a sun glint is part of the consecutive image sequence, first a jump in the intensity can be observed and later on the opposite impulse response can be seen. Figure 42 visualises the response (PCA major axis) in time if images with a sun glint is part of the image sequence. Images, which potentially contain the direct stray light effect anomaly can be detected by calculating the correlation with the response from Figure 42.

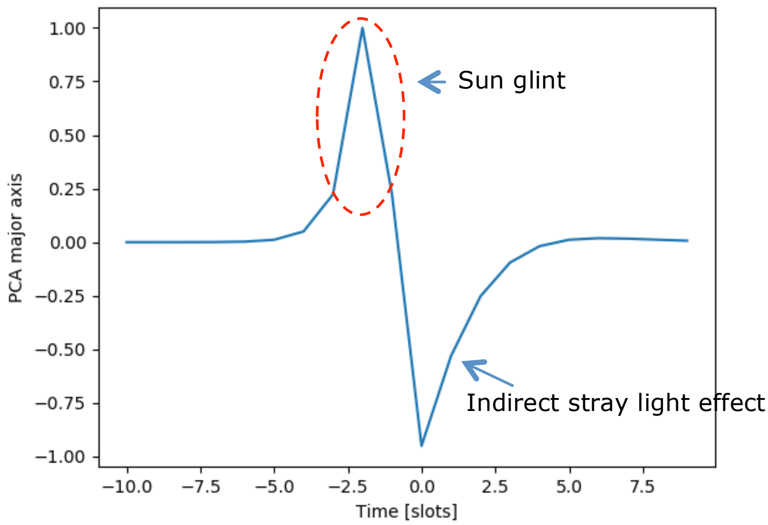


Figure 42 Simplified error response in time for the average intensity of a scanline, where the consecutive image sequence contains a few images with a sun glint.

For detecting this anomaly many images (time-span can be -10 timeslot in past and +7 timeslot in future) are compared with each other. During the long time-span, the deformation (matrix) can change significantly and it affects the capability to compare the images. The change in the deformation seems to be larger when the satellite is outside its normal prime period. To compensate for the change in the deformation, the average intensity per scanline can be translated with the local deformation. Figure 43 shows the calculation for compensating for the deformation matrix. Over the time span of 5 timeslots, the deformation can change with 100 pixels and therefore it is important to compensate for the deformation.

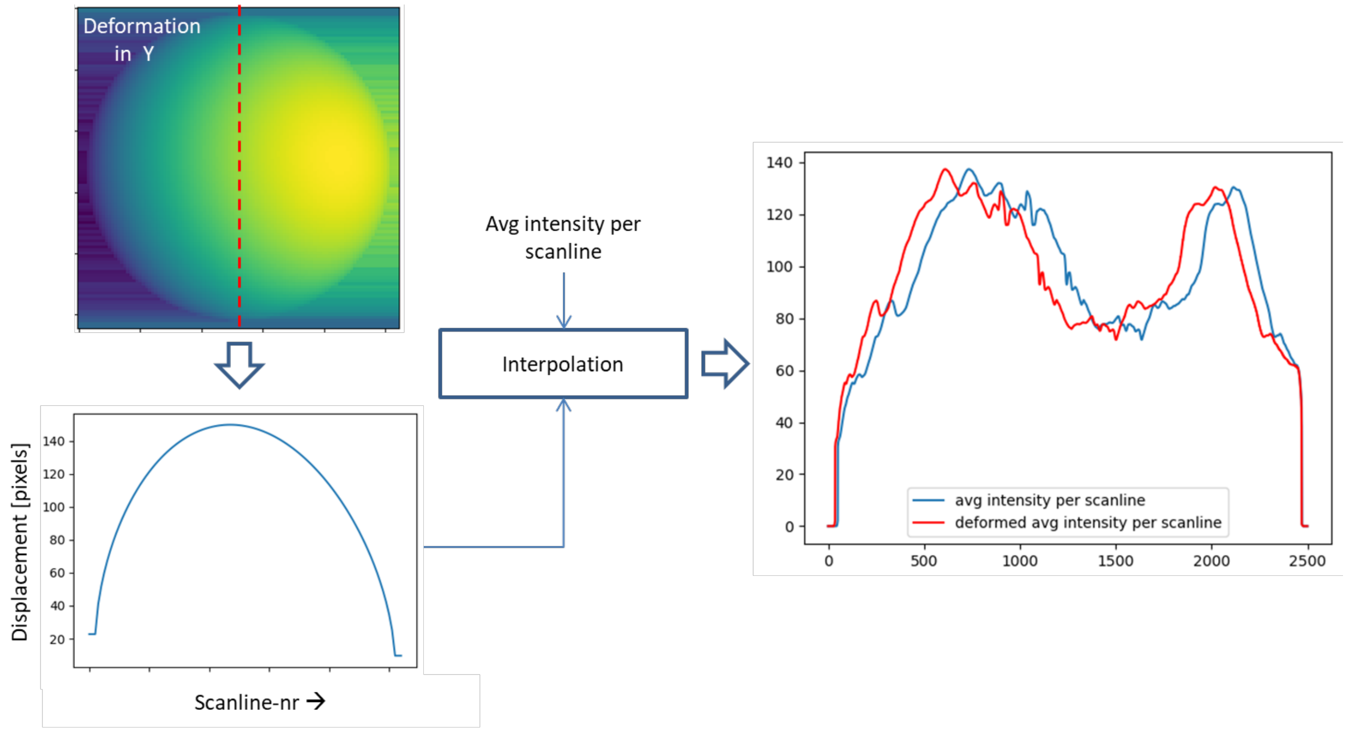


Figure 43 To compare the average intensity per scanline (V-scanline) over a long time span, the V-scanline needs to be compensated for the deformation matrix. Used data is from METEOSAT5-MVIRI-MTP10-NA-NA-20050905220000

5.1.18.2 Algorithm

Before the algorithm is executed, the following pre-check is done:

- WV-channel is valid
- Slot-nr > 16 and slot-nr < 40

Figure 44 shows the entire processing pipeline for detecting the indirect stray light anomaly and Figure 47 shows the processing steps for determining the affected scanlines.

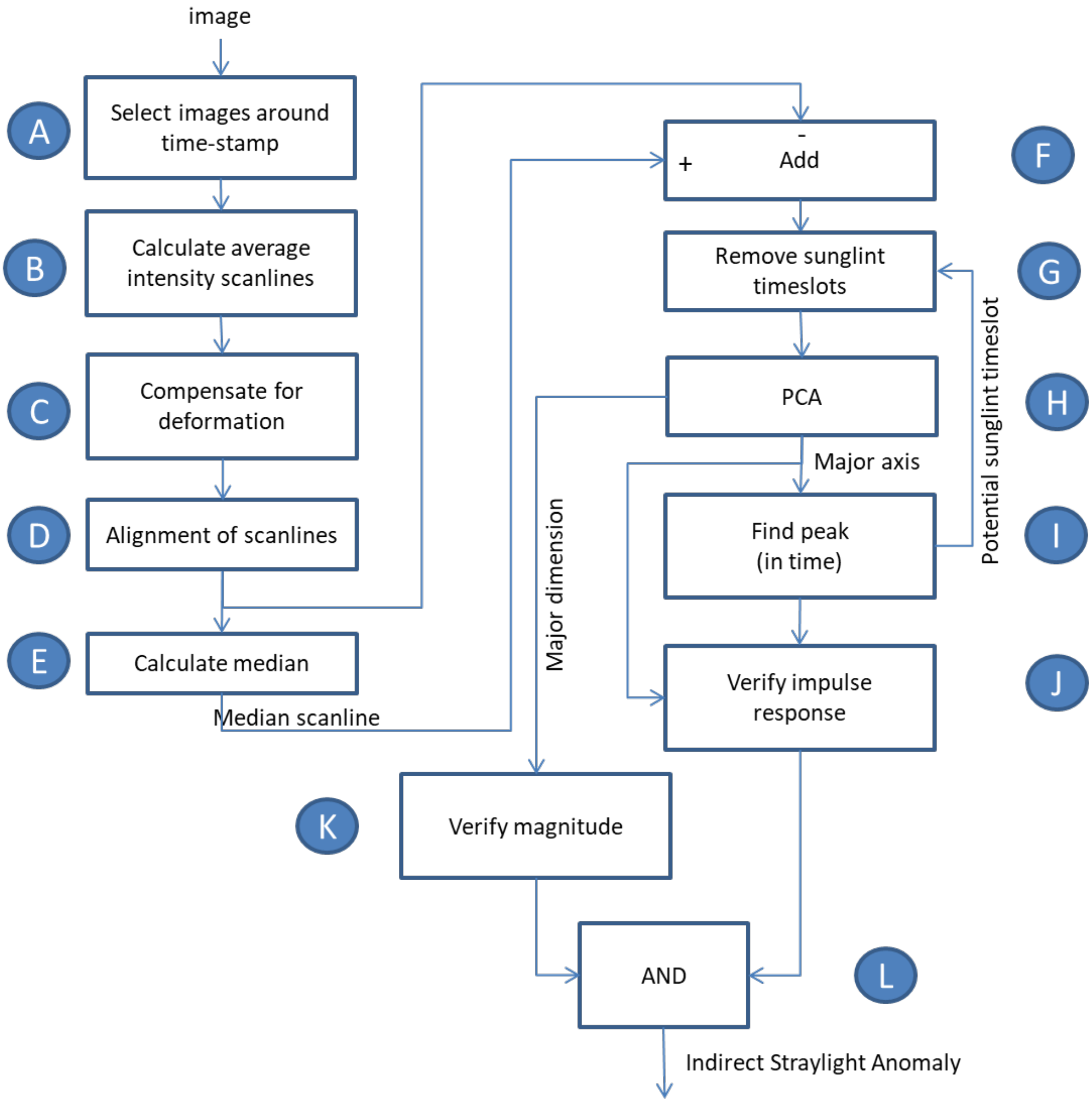


Figure 44 Processing pipeline for detecting the indirect straylight anomaly. Table 33 describes the elementary processing steps.

Table 33 Description of the processing steps, which are used for the detection of the indirect straylight anomaly

Step	Description
A	Images around the requested timeslot are selected. Preferably 7 images from the past and 3 images from the future are selected. The search horizon is [-10 - +7] timeslots. The images are verified if the WV-channel is valid, the number of scanlines is 2500 and the average intensity is between 75 and 150 (to exclude image, which are affected by other anomalies).
B	The average intensity per scanline is calculated as defined in section 5.2.3
C	The average intensity per scanline is compensated for the deformation, which is defined by the deformation matrix. Figure 43 defines this compensation.
D	The scanlines are aligned with the reference scanline by calculating the cross-correlation and selecting the shift with the maximum cross-correlation.
E	From the aligned scanlines, the median is calculated. It is assumed that this median is not affected by the indirect stray light anomaly. Figure 45 shows an example of the aligned scanlines and the calculated median.

F	Calculate the difference between the aligned scanlines and the median. The effect of the indirect stray light anomaly during the various timeslots is clearly noticeable in this calculated difference. Figure 45 shows an example of this calculated difference.
G	By applying the Principle Component Analysis and verifying the major axis, the phenomenon of the sun-glint (like Figure 42) can be detected. If the appearance of the sun-glint is detected, it is ignored.
H	By applying the Principle Component Analysis on the difference between the aligned scanlines and the median, the bias per scanline due to the indirect stray light effect anomaly can be calculated. Also the effect in time can be calculated. Figure 46 shows an example of the PCA-calculation.
I	The start of the indirect straylight anomaly can be found by searching with cross-correlation, where the PCA major axis best matches with the following impulse response: $P_{\text{indirect straylight}}(t) = \exp(-0.55*t)*\cos(0.35*t)$ $P_{\text{direct straylight}}(t) = -1*\exp(-1.5*(t-2))$ $P_{\text{total}}(t) = P_{\text{indirect straylight}}(t) + P_{\text{direct straylight}}(t)$ The defined constants (-0.55, 0.35 and -1.5) in the impulse response are determined on data, which did contain the stray light effect.
J	The values of the PCA major axis (in time) are compared with the expected impulse response in time. If average absolute difference between the PCA major axis and the expected impulse response is larger than 0.2 (**maxMatchErrorImpulseResponse**), then the observed response is considered to be too different to be caused by the indirect straylight effect anomaly.
K	The major dimension from the PCA result is the estimation of the bias per scanline due to the indirect straylight effect anomaly. From this estimated bias per scanline, the area (bias x nr scanlines) can be calculated. If the calculated area is smaller than 5000 (**minAreaError**) then calculated bias is considered to be too small for the indirect stray light effect anomaly.
L	If the conditions from step J and K are fulfilled then the indirect stray light effect anomaly is detected. When this anomaly is detected, the anomaly descriptor will also contain the parameter sequence_nr, which represents the number of timeslots after the start of the indirect straylight effect anomaly. The start of the indirect stray light effect in calculated in step I.
M	The major dimension from the PCA result is the estimation of the bias per scanline, but for the deformed scanlines, which were can calculated in step C. By applying the inverse mapping, the bias per scanline can be calculated for original scanlines of the reference image.
N	The bias per scanline at certain timeslot can be calculated by: PCA_major_dim * PCA_major_axis[idx], where idx is the index of the timeslot. The affected area is determined by the scanlines, where the calculated bias is larger than 10. The calculated affected area depends on the past time after the start of the anomaly and it will become each timeslot smaller. After approximately 4 timeslots, the bias due to this anomaly is probably less than 10 and there will not be any scanline marked as affected (no affected area). Still this anomaly will be detected, but without an affected area and the user can decide what to do with the image data.

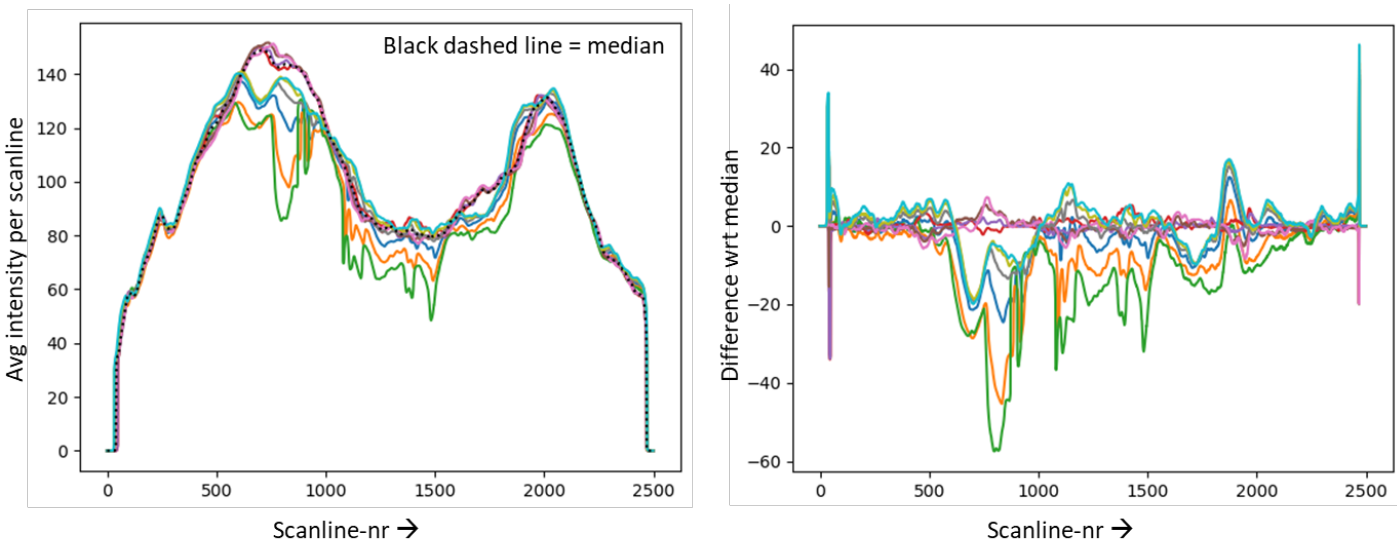


Figure 45 Left: aligned average intensity per scanline including the median. Right: difference between average intensity per scanline and the median. Used data is from METEOSAT5-MVIRI-MTP10-NA-NA-20050905220000.

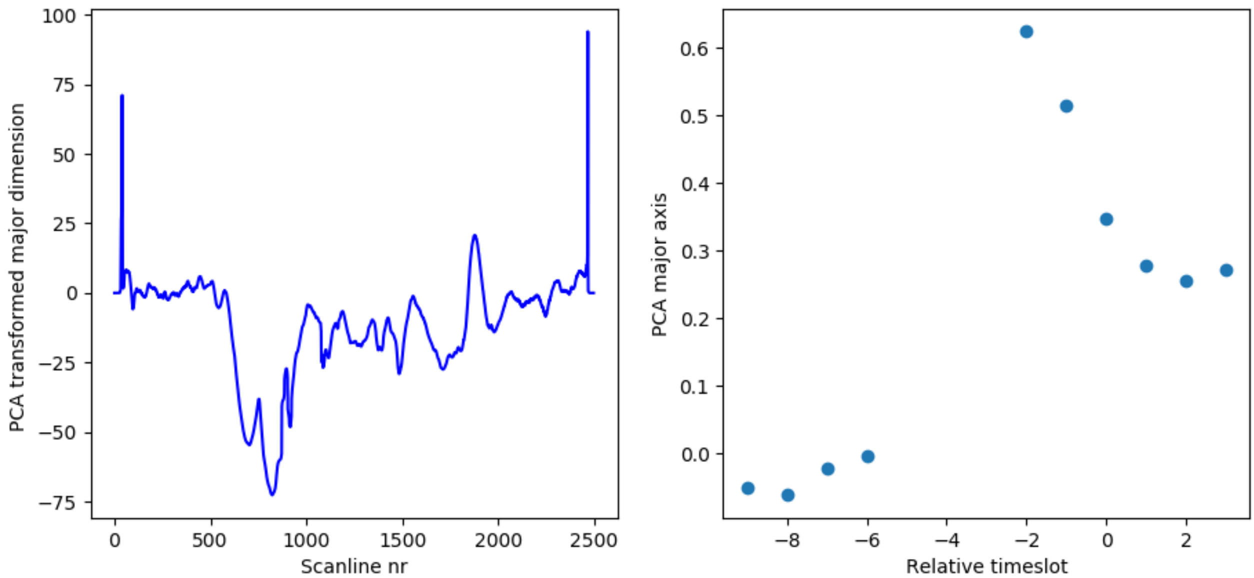


Figure 46 Principle Component Analysis result from the difference between the aligned scanlines and the median from Figure 45.

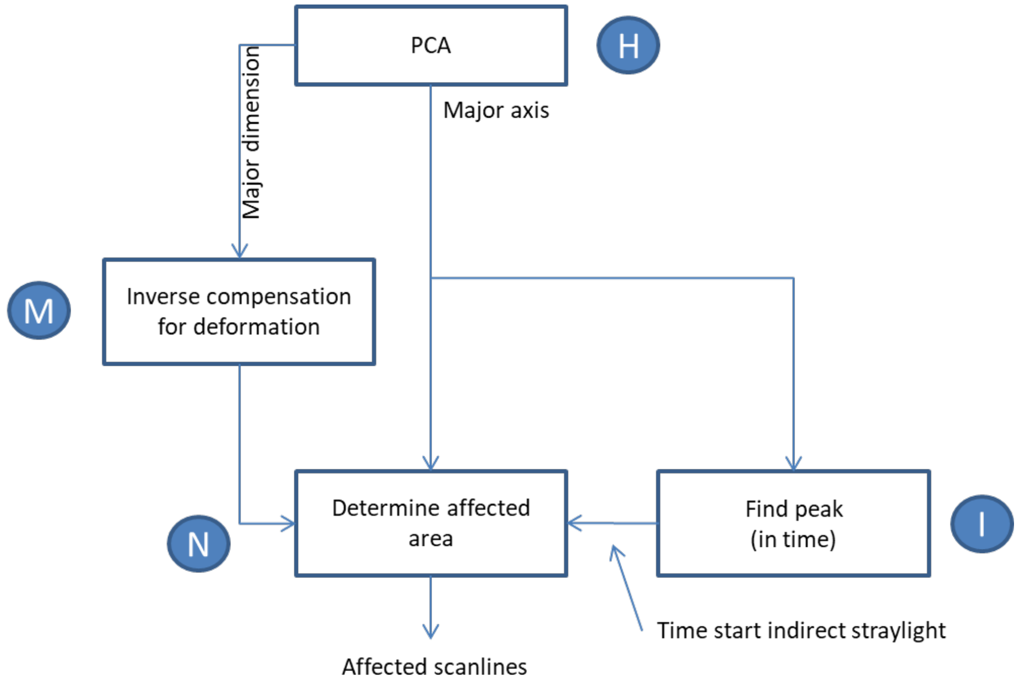


Figure 47 Processing pipeline for determining the affected scanlines. Table 33 describes the elementary processing steps.

Table 34 Indirect straylight anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	IndirectStrayLightAlgorithm
anomaly_type	IndirectStrayLight
Sig	WV
locus_type	BBox
Locus	Affected areas ([x0, y0, x1, y1])
Details	Sequence_nr : number of timeslots after the start of the indirect straylight effect anomaly

5.1.19 Moon Reflection

Figure 48 and Figure 49 shows some examples of the moon reflection anomaly. The moon reflection anomaly has the shape of a nail and the intensity is 252. It appears approximately 600 pixels at the right side from the moon and it only appears in the WV-images of the M2 and M3 satellite.

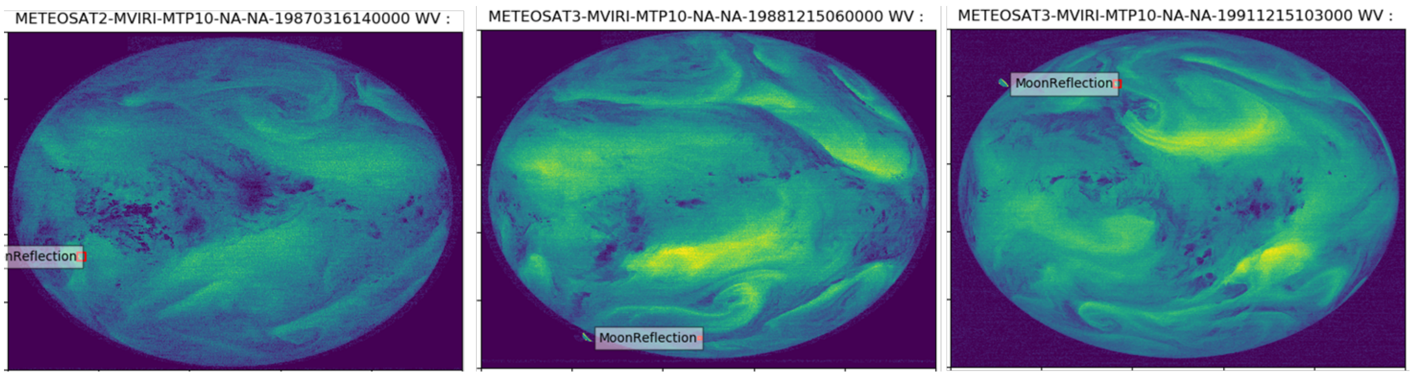


Figure 48 Three examples of the moon reflection anomaly

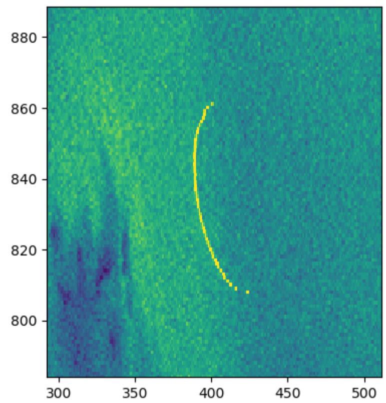


Figure 49 Example of moon reflection anomaly (METEOSAT2-MVIRI-MTP10-NA-NA-19870316140000 WV)

The pre-check of the moon reflection algorithm is:

- File exists
- Sat-id is M2 or M3
- WV-channel is valid
- Number of sub image is 1

Figure 50 shows the processing pipeline for detecting the moon reflection anomaly. The nail-like shape of this anomaly is very narrow and 1 pixel wide and occasionally not all pixels of this anomaly are connected to each other. With the 3x7 dilation operation, all pixels of the anomaly are connected to each other, which is important for determining the shape.

The following properties of the found shape (bounding box) are verified:

- 15 <= width <= 60
- 15 <= height <=75
- Height > width -5 (assume vertical shape)

It is assumed that the found shape is nail-like and with this assumption the thickness is calculated:

thickness = (number of pixels) / (height + 0.5*width)

The calculated thickness should be larger than 0.5 pixels and smaller than 1.25 pixels.

The Moon reflection anomaly is caused by an indirect reflection of the moon and the expected image coordinates of the Moon would be 600 pixels to the left of the found shape:

$x_{Moon} = x_{shape} - 600$, $y_{Moon} = y_{shape}$

In general the moon is on the left-side of the earth, but it is also possible that it is just outside the image. At least the position cannot be behind the Earth and this information is used to verify the calculated image coordinates of the moon. We assume that the disc of the Earth can be described by $[X_0,Y_0] = [1250, 1250]$ and radius 1100 pixels. The calculated position of the Moon should fulfil to this condition:

$(x_{Moon} - 1250)^2+(Y_{Moon} - 1250)^2 > (1100)^2$

Afterwards we have to verify if the calculated image coordinates match with the orbit coordinates of the moon. Section 5.2.4 defines the mathematics to calculate the image coordinates on basis of the expected orbit of the moon. Both calculated image coordinates of the Moon are compared with each other and if the difference is smaller than 500 pixels, then the found nail-like shape is detected as the moon reflection anomaly.

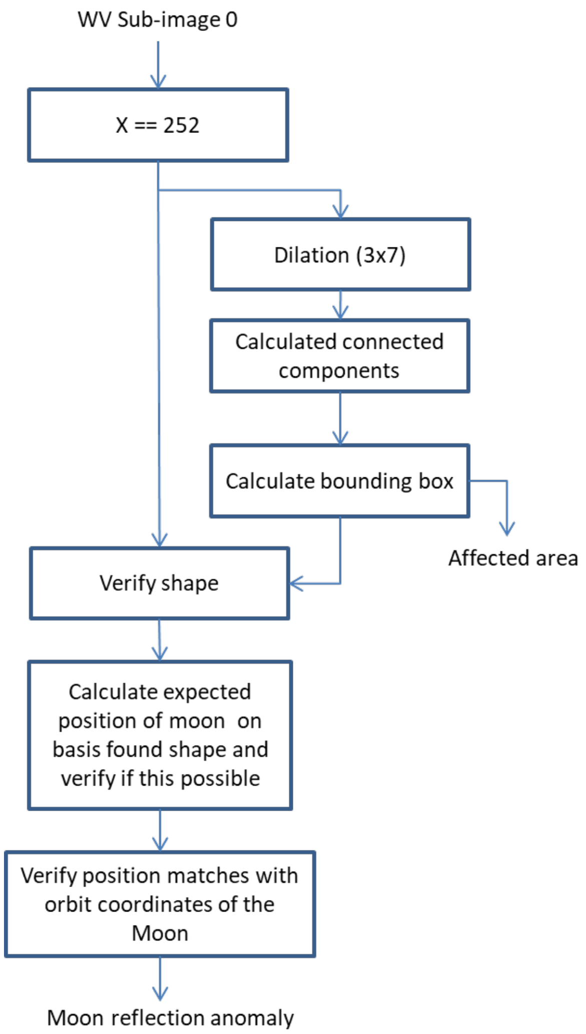


Figure 50 Processing pipeline to detect the moon reflection anomaly

Table 35 Moon reflection anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	MoonReflectionAlgorithm
anomaly_type	MoonReflection
Sig	WV
locus_type	BBox
Locus	Affected area ([x0, y0, x1, y1])

5.1.20 Celestial Body

Celestial bodies can appear in the field-of-view and these objects should be detected and flagged. Celestial bodies can be the moon, other planets or other satellites, but it is not expected that another satellite will be detected in the captured images.

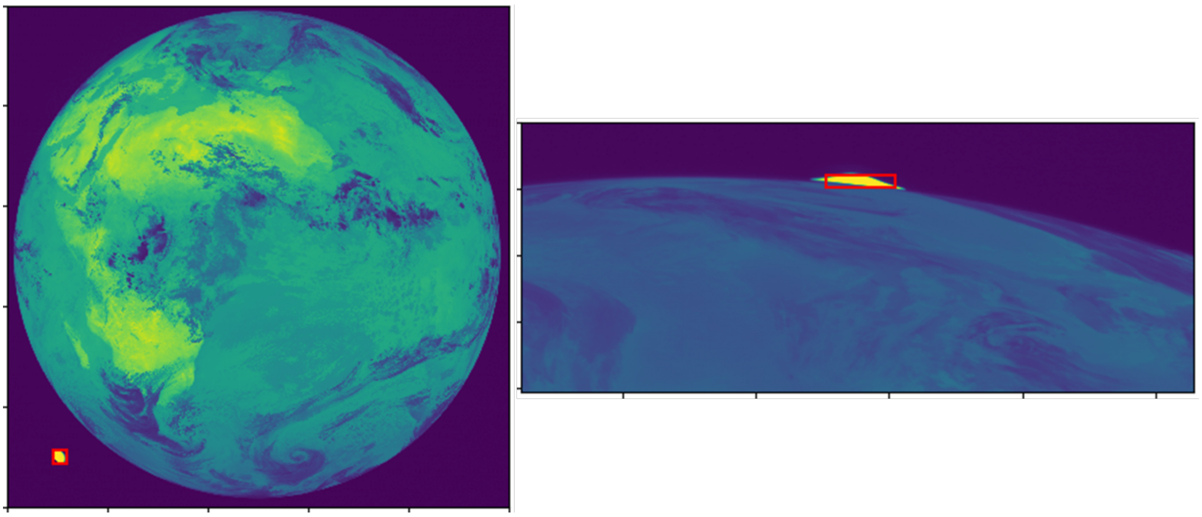


Figure 51 Examples of moon as celestial body. Left: METEOSAT5-MVIRI-MTP10-NA-NA-19940917110000. Right: METEOSAT5-MVIRI-MTP10-NA-NA-20030217083000

The pre-check of the celestial body algorithm is:

- File exists
- IR-channel is valid
- Number of sub images is 1

In general, celestial bodies are most clearly visible in the IR-signal, due to low sensitivity to the stray light effect and because no illumination from the sun is required.

Figure 52 presents the algorithm for detecting celestial bodies in the IR-channel. It is assumed that the celestial body is visible (intensity higher than 16) in the IR-image and separated from the earth or that it is over-illuminated and near the edge of the earth (see Figure 51 right). The shape of celestial body is expected to be larger than 5 pixels and smaller than 500 pixels.

When a celestial body is detected, it is verified if the image coordinates of the celestial body matches with the orbit of the moon. Section 5.2.4 defines the mathematics to calculate the expected image coordinates on basis of the orbit of the moon and if the difference between the image coordinates is smaller than 300 pixels, the detected celestial body is classified as the Moon.

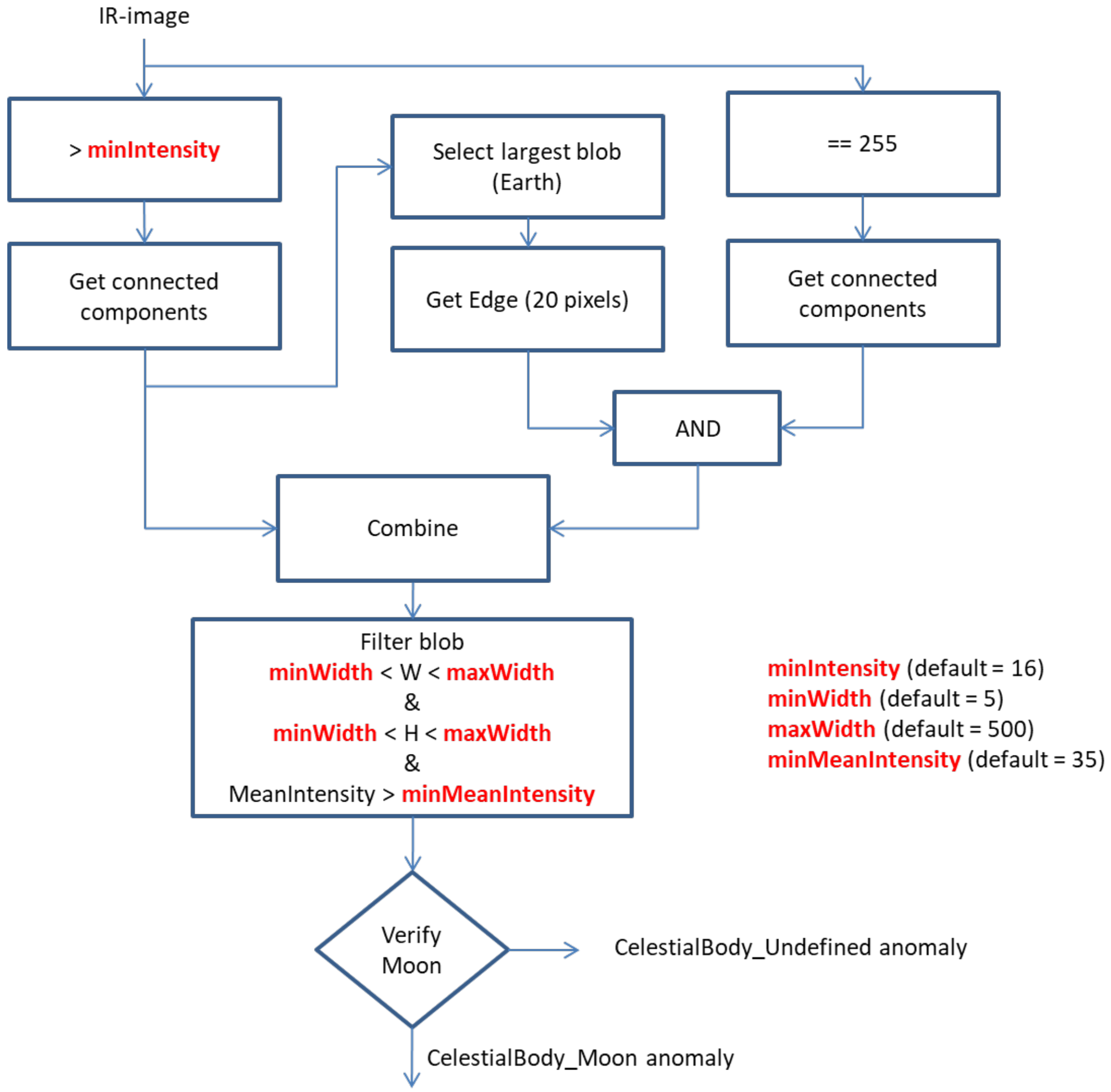


Figure 52 Algorithm for detecting celestial bodies

Table 36 Moon / undefined celestial body anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	CelestialBodyAlgorithm
anomaly_type	CelestialBody_Moon / CelestialBody_Undefined
Sig	IR
locus_type	BBox
Locus	Affected area ([x0, y0, x1, y1])

5.1.21 Instable Optics

The M5 (rotating lens) and M6 (loose cold optics) have known hardware issues, which affects the sensitivity of the optics. The result of such hardware issues is not noticeable in a single image (effect is too small), but the effect is noticeable if the average IR-intensity is analysed. Figure 53 and Figure 54 shows the average IR-intensity (from the entire image) in time and it clearly visible that the sensitivity of the detector is affected.

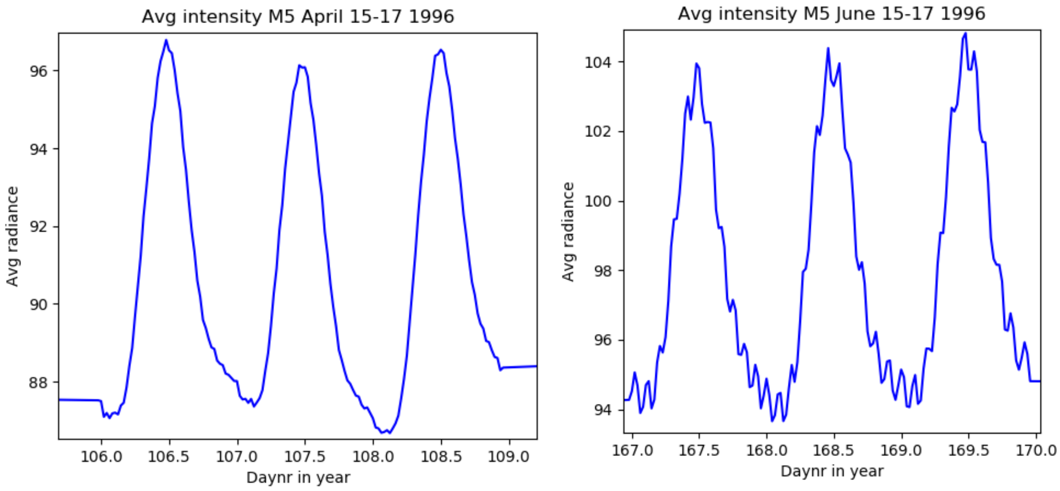


Figure 53 Average IR-radiance of M5 in period April 15-17 1996 and June 15-17 1996.

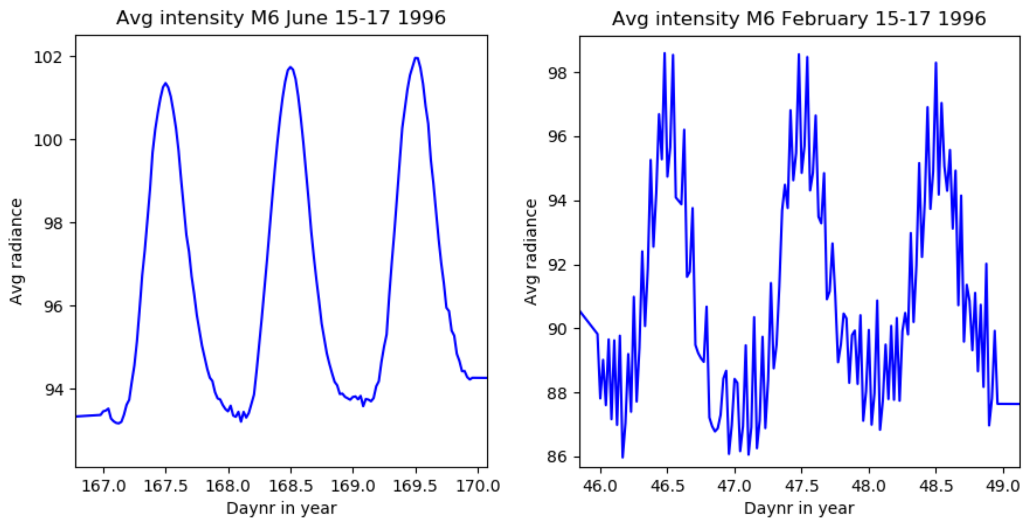


Figure 54 Average IR-radiance of M6 in period June 15-17 1996 and February 15-17 1996.

The detectors' sensitivity becomes higher and lower in time and the periodicity of this process varies. By calculating the average IR intensity from the entire image, there is only 1 measurement every 30 minutes, which is insufficient to monitor the effects of this hardware issue. Therefore we will analyse the average intensity per scanline and compare it with related images, to determine if the sensitivity is changing. Because the scanlines of the related image might also be affected by the anomaly, we have to make sure that enough images are considered such that the temporal effect of this anomaly is averaged out.

Figure 55 shows the average IR intensity per scanline for 20 consecutive images. In this figure, we can see that the anomaly has a period of a few timeslots and that anomaly is noticeable due to the difference between two images.

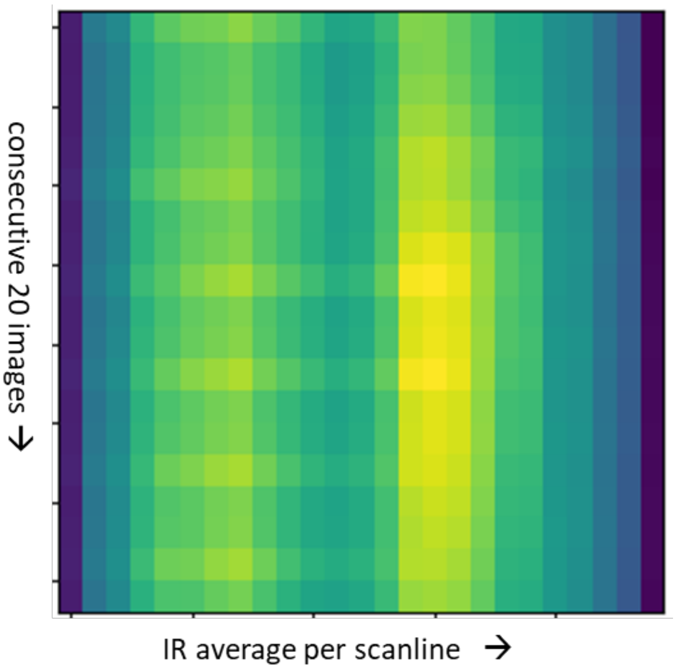


Figure 55 IR average intensity per scanline for 20 consecutive images from M6 around February 16 1996 at 12:00.

We define the following variables for modelling this anomaly:

X[i,j] = measured average intensity of scanline j from image i.

B[i,j] = bias due to the instable optics anomaly of scanline j from image i

Without the anomaly, the difference between the average intensity per scanline of consecutive image would be small. With the anomaly, the following holds:

X[i,j] – B[i,j] ≈ X[i-1,j] – B[i-1,j]

Of course, these equations holds for every image i. This can be rewritten as:

X[i,j] - X[i-1,j] ≈ B[i,j] – B[i-1,j]

The anomaly is a temporal process and if we assume that the averages out over several (N) timeslots, the following holds:

B[i,j] + B[i+1,j]+ B[i+2,j]+ … B[i+N,j] ≈ 0

The equations for all images can be stored in a matrix. As example a part of the matrix, where the anomaly is averaged out over 5 images with weigth W:

$$\begin{bmatrix} -1 & 1 & & & & & & \\ & -1 & 1 & & & & & \\ & & -1 & 1 & & & & \\ & & & -1 & 1 & & & \\ & & & & -1 & 1 & & \\ & & & & & -1 & 1 & \\ & & & & & & -1 & 1 \\ W & W & W & W & W & & & \\ & W & W & W & W & W & & \\ & & W & W & W & W & W & \end{bmatrix} \cdot \begin{bmatrix} B[i,j] \\ B[i+1,j] \\ B[i+2,j] \\ B[i+3,j] \\ B[i+4,j] \\ B[i+5,j] \\ B[i+6,j] \\ B[i+7,j] \end{bmatrix} \sim \begin{bmatrix} X[i+1,j]-X[i,j] \\ X[i+2,j]-X[i+1,j] \\ X[i+3,j]-X[i+2,j] \\ X[i+4,j]-X[i+3,j] \\ X[i+5,j]-X[i+4,j] \\ X[i+6,j]-X[i+5,j] \\ X[i+7,j]-X[i+6,j] \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The effect of the anomaly is defined by a linear matrix equation and with least squares optimization, the solution of B[i,j] can be calculated. As example, Figure 56 shows the calculated bias per scanline per images.

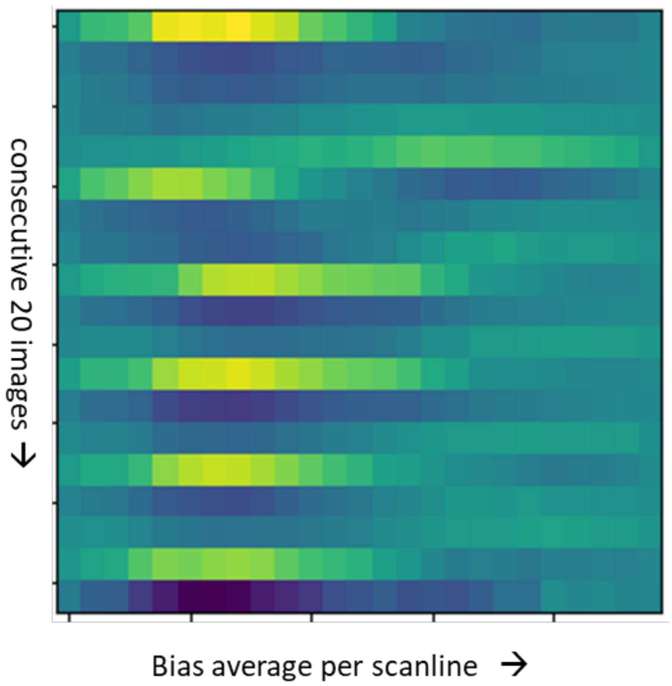


Figure 56 Calculated bias per scanline per image (around M6 around February 16 1996 at 12:00)

We have seen that the anomaly affects the sensitivity of the IR- and WV-detector simultaneously. Therefore we use the same bias for both detectors and instead of executing the calculation twice, we use the average and IR and WV as X[i,j].

To detect this anomaly 21 consecutive (10 before and 10 after) images are considered. During this time span, the Earth heats up and this phenomenon affects the calculation of the anomaly's bias. To minimize this effect, the averaging windowing (anomaly effects averages out over several image) should not be too large. With averaging window of 5 timeslots, this effect on the calculation is still acceptable.

Figure 57 shows the processing pipeline to detect the instable optics anomaly and Table 37 explains the processing steps.

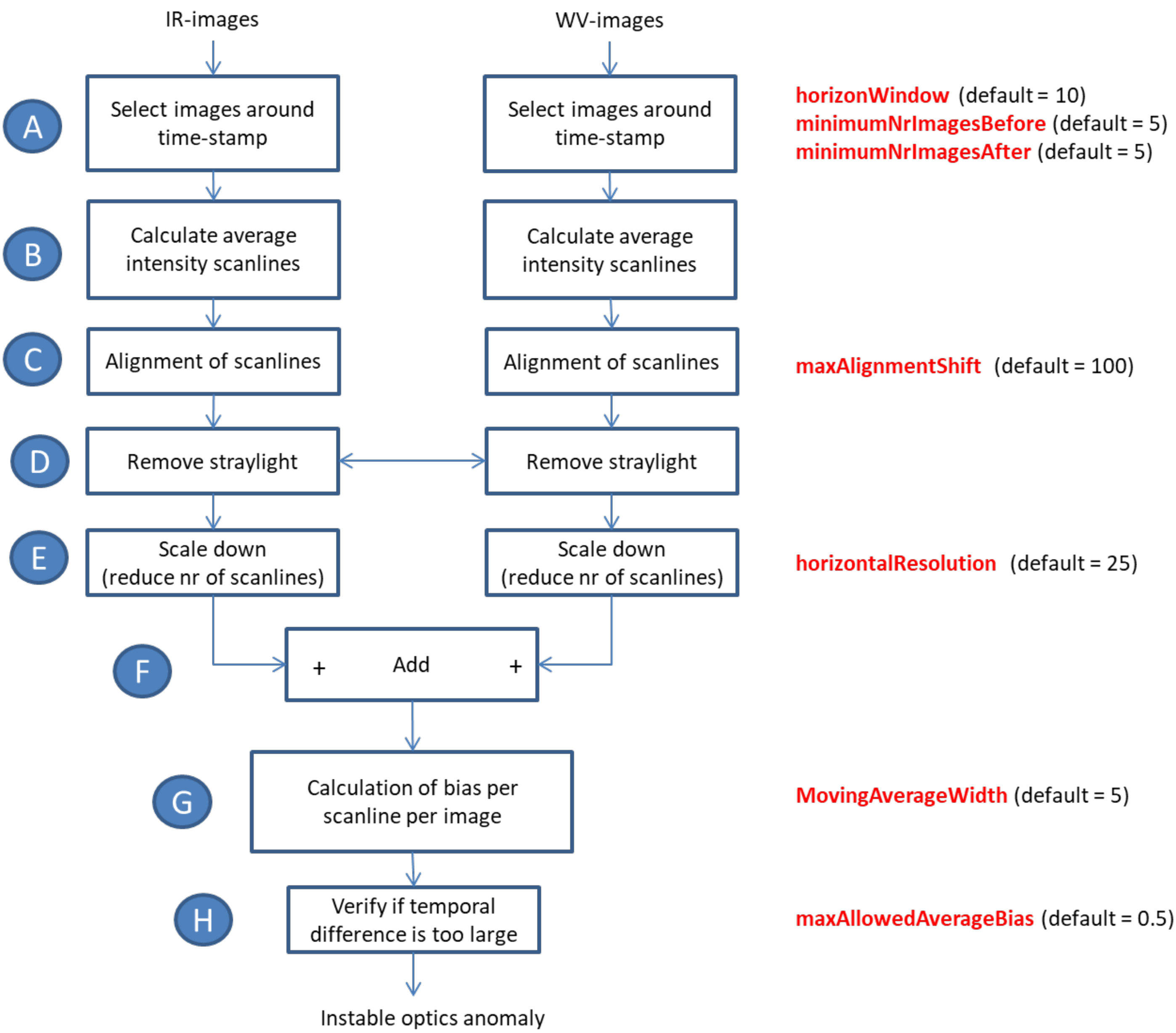


Figure 57 Processing pipeline to detect the instable optics anomaly

Table 37 Description of the processing steps, which are used for the detection of the instable optics anomaly

Step	Description
A	Images around the requested timeslot are selected. Preferably 10 images from the past and 10 images from the future are selected. The images are verified if the IR- and WV-channel are valid, the number of scanlines is 2500 and the average intensity is between 50 and 200 (to exclude image, which are affected by other anomalies).
B	The average intensity per scanline is calculated as defined in section 5.2.3
C	The scanlines are aligned with the reference scanline by calculating the cross-correlation and selecting the shift with the maximum cross-correlation. The calculated shift is verified if it is smaller than 100 pixels.
D	The average intensity an image is compared with related images. If difference is larger than 2 then it is expected that the image is affected by the direct straylight anomaly. This image will be ignored in the remainder of this algorithm.
E	The optimization will calculate per scanline per image a bias. Because the bias changes slowly in time, it useless to calculate it for every scanline. The number of scanlines is reduced (by averaging), to increase the accuracy of calculated bias and to reduce the number of parameters.

	The bias for the 2500 scanlines will be described by 25 parameters.
F	The average intensity per scanline of the IR- and WV-channel are combined.
G	With least-squares optimization, the bias per scanline per image is calculated. The required mathematics is defined in this section.
H	The calculated bias per scanline of the reference image is selected. If the average from the absolute value of the all bias-parameters is larger than 0.5, the instable optics anomaly is detected.

Table 38 Instable optics anomaly: relevant parameters of the anomaly descriptor

Parameter	Value / Description
Alg	InstableOpticsAlgorithm
anomaly_type	InstableOptics
Sig	IR_AND_WV
locus_type	Image

5.1.22 Verify Position and Start Time

The stored orbit coordinates in Earth Fixed Frame in the metadata is sometimes empty or corrupt. The value of the coordinates needs to be verified with the stored orbit coordinates in Mean Geocentric Format. Section 5.1.22.1 and 5.1.22.2 defines the required checks to detect the Orbit Position Empty anomaly and the EFF Position Corrupt anomaly.

The definition of the stored start time has changed, due to changes in the processing software. Before slot 19 on 7/12/1992 (Julian slot 752707), the start time is defined as the moment when the forward scan is started. Afterwards the start time is defined as the moment when the southern horizon is detected in the scan. It is desired to verify if the correct definition of start time is used in the level 1.0 file. Section 5.1.22.3 defines an algorithm for detecting meta data anomalies concerning the start time.

During the investigation of the ground truth dataset, we have discovered that all files from M2 used a different definition for the start time. The start time was in all cases at the moment of the start of the image. The same anomaly appeared with the other satellites, but not very often.

5.1.22.1 Orbit position is empty

To check if the orbit position is stored in the meta data of the level 1.0 file, the MGF and the EFF orbit coordinates needs to checked:

```
IF RECORD[3]["ORBF"][0] == 0 OR RECORD[3]["FORBS"] == 0 THEN
        Anomaly ➔ OrbitPositionEmpty
```

5.1.22.2 Orbit position is corrupt

The stored EFF orbit coordinates in the metadata can be incorrect. The MGF orbit coordinates with the actual time and the longitudinal position can be transformed to EFF orbit coordinates. EUMETSAT has provided Fortran code (orbit2eff.f) with the function "MGF2EFF" for this calculation. The EFF orbit coordinates can be calculated as follows:

```
TJUL = (RECORD[3]["JULSL3"] / 48 - (1 / 48))
ORBF = RECORD[3]["ORBF"]
FORBS = RECORD[3]["FORBS"]
NOMINAL_LONGITUDE = IMAGE15.RECORD[3]["dsNominalSubSatelliteLongitude"]
FORBS2 = MGF2EFF (TJUL, ORBF, NOMIMAL_LONGITUDE)
```

Note that the nominal longitudinal position is retrieved from the level 1.5 file, because the stored longitudinal position in the level 1.0 file is always empty.

If the difference between the stored and the calculated EFF orbit coordinates is larger than 500 km, the stored orbit coordinates are considered to be corrupt:

IF NORM(FORBS – FORBS2) > 500 THEN

Anomaly ➔ EFF Position Corrupt

5.1.22.3 Start time anomaly

To determine if the correct definition of the start time is used, we first have to calculate what the time difference is. The time difference can be calculated from the difference between the stored MGF orbit coordinates and the stored EFF orbit coordinates. The previous section has defined the calculation between the MGF- and EFF- orbit coordinates, where the MGF2EFF-function is used. The difference in EFF orbit coordinates (in 3 dimensions) can be converted to a time difference by simply dividing it by the velocity (in 3 dimensions).

TJUL = (RECORD[3]["JULSL3"] / 48 - (1 / 48))
ORBF = RECORD[3]["ORBF"]
FORBS = RECORD[3]["FORBS"]
NOMIMAL_LONGITUDE = IMAGE15.RECORD[3][“dsNominalSubSatelliteLongitude”]

FORBS2 = MGF2EFF (TJUL, ORBF, NOMIMAL_LONGITUDE)

EPSILON = 1E-4

FORBS2B = MGF2EFF (TJUL+ EPSILON, ORBF, NOMIMAL_LONGITUDE)

$$TIMEDIF = \left(\begin{bmatrix} (FORBS2B[0] - FORBS2[0])/EPSILON \\ (FORBS2B[1] - FORBS2[1])/EPSILON \\ (FORBS2B[2] - FORBS2[2])/EPSILON \end{bmatrix}\right)^{-1} \cdot \begin{bmatrix} FORBS[0] - FORBS2[0] \\ FORBS[1] - FORBS2[1] \\ FORBS[2] - FORBS2[2] \end{bmatrix}^{T} \cdot 3600 \cdot 24$$

When the time difference is calculated, we need to verify if it matches with the start time of the image. This is the case, when the time difference is zero.

IF ABS(TIMEDIF) < 1E-3 THEN

STARTTIME_IMAGE = TRUE

ELSE

STARTTIME_IMAGE = FALSE

Afterwards we verify if the start time matches with the start of the forward scan. The scanline number where the forward scan starts, is defined in meta data variable FLISIM. The time between the scanlines is 0.6 seconds and we can calculate the difference between the calculated times.

TIME_SCANLINE = 0.6

POS_FORWARDSCAN = RECORD[1][“FLISIM”][0]

TIMEDIF_FORWARDSCAN = TIMEDIF - TIME_SCANLINE* POS_FORWARDSCAN

We need to verify if this calculated time difference is small enough (**maxTimeDeltaForwardScan** = 5) to classify the start time as the moment of the start of the forward scan.

IF ABS(TIMEDIF_FORWARDSCAN) < **maxTimeDeltaForwardScan** THEN

STARTTIME_FORWARDSCAN = TRUE

ELSE

STARTTIME_ FORWARDSCAN = FALSE

To determine if the start time can be equal to the moment of detection of southern horizon, we first have to find the southern horizon. The position southern horizon can be found by searching in the average intensity per scanline of the IR-image: IR_SCANLINES. The first element with intensity above 20 will be the southern horizon.

IR_AVG_SCANLINE = CALC_AVG_VALUE_SCANLINE(IR_SCANLINES, THRESHOLD = 20)

POS_SOUTHERN_HORIZON = POS_FORWARDSCAN + WHERE(AVG_SCANLINE > 20) [FIRST]

The calculation of CALC_AVG_VALUE_SCANLINE is defined in section 5.2.3.

When the southern horizon is detected, we can calculate the corresponding time difference. Afterwards we can verify if this calculated time difference matches (**maxTimeDeltaSouthernHorizon** = 10) with earlier calculated time difference.

TIMEDIF_SOUTHERN_HORIZON = TIMEDIF - TIME_SCANLINE* POS_SOUTHERN_HORIZON

IF ABS(TIMEDIF_SOUTHERN_HORIZON) < **maxTimeDeltaSouthernHorizon** THEN

STARTTIME_SOUTHERNHORIZON = TRUE

ELSE

STARTTIME_ SOUTHERNHORIZON = FALSE

When all time differences are calculated and verified, we can determine if the correct definition of the start time is used (**southernHorizonAsStart_JulianSlNr** = 75207).

IF RECORD[3]["JULSL3"] < **southernHorizonAsStart_JulianSlNr** THEN

IF NOT(STARTTIME_ FORWARDSCAN) THEN

IF (STARTTIME_IMAGE) THEN

Anomaly ➔ StartTime_StartImage

ELSE

IF (STARTTIME_SOUTHERNHORIZON) THEN

Anomaly ➔ StartTime_SouthernHorizon

ELSE

Anomaly ➔ StartTime_Undefined

ELSE

IF NOT(STARTTIME_ SOUTHERNHORIZON) THEN

IF (STARTTIME_IMAGE) THEN

Anomaly ➔ StartTime_StartImage

ELSE

IF (STARTTIME_ FORWARDSCAN) THEN

Anomaly ➔ StartTime_ForwardScan

ELSE

Anomaly ➔ StartTime_Undefined

5.2 Common algorithms

This section defines several algorithms, which are used by the anomaly detection algorithms from the previous section.

5.2.1 Channel valid

For each channel of the level 1.0 file, it is verified if it is valid / detector is enabled. Because each channel has two detectors, we have to verify if one of the detectors is enabled. The following checks are done to verify if a channel is valid:

VIS1_VALID	=	RECORD[1]["CHANVI"][1]	OR	RECORD[1]["CHANVI"][3]
VIS2_VALID	=	RECORD[1]["CHANVI"][0]	OR	RECORD[1]["CHANVI"][2]
IR_VALID	=	RECORD[1]["CHANIR"][0]	OR	RECORD[1]["CHANVI"][1]

WV_VALID = RECORD[1][“CHANWV”][0] OR RECORD[1][“CHANWV”][1]

5.2.2 Get first forward scan / sub-image

The first forward scan or sub-image contains in general the relevant image data. The following pseudo-code defines how the first forward scan is extracted from a level 1.0 file:

```
IF  RECORD[1][“CHANVI”][1] > 0  THEN
        Y1          =       RECORD[1][“FLISIM”][0] – 1
        Y2          =       Y1      +  RECORD[1][“NLISIM”][0]
        SUB_IMAGE_0  =         PIXELS[Y1:Y2,:]
ELSE
        SUB_IMAGE_0  =         []
```

, where PIXELS are all stacked scanlines of channel VIS1, VIS2, IR or WV.

5.2.3 Calculate scanline average value

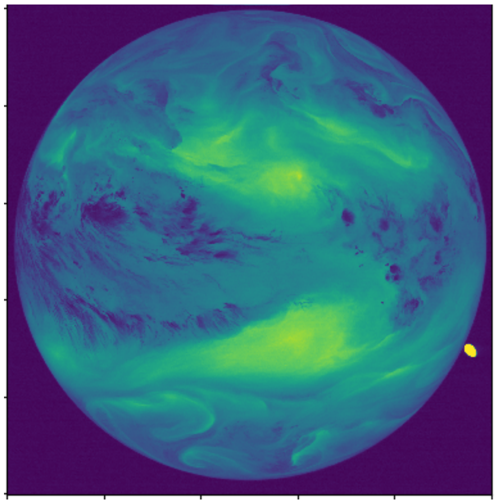
Several anomalies can affect a certain scanline or affects a series of scanlines. To detect these anomalies it is desired to calculate the average intensity of the “Earth” for each scanline. This calculation can be done for all channels. To calculate the average intensity of the Earth in a scanline, the following calculation is executed:

```
SCANLINE_EARTH = SCANLINE * (SCANLINE > THRESHOLD)
AVG_VALUE = SUM(SCANLINE_EARTH) / COUNT(SCANLINE > THRESHOLD)
IF (SCANLINE[1100:1400]  > THRESHOLD) < 10 THEN          // to make sure horizon detection is not
        AVG_VALUE = 0                                    // complicated by direct straylight next to
                                                         // the earth
```

, where SCANLINE is an array with all pixels of a particular scanline, THRESHOLD is minimum value to categorize a pixel belonging to the earth. The range of [1100:1400] holds for the IR- and WV-channel and for the VIS1- and VIS2-channel the range is [2200:2800].

5.2.4 Calculate image coordinates of Moons orbit

For example to identify if a detected celestial body can be the Moon, we would like to calculate the image coordinates of the expected orbit of the Moon. This section defines the mathematics to calculate the image coordinate on basis of the expected orbit of the Moon, position of the satellite, start time of the scan and scanline-nr.



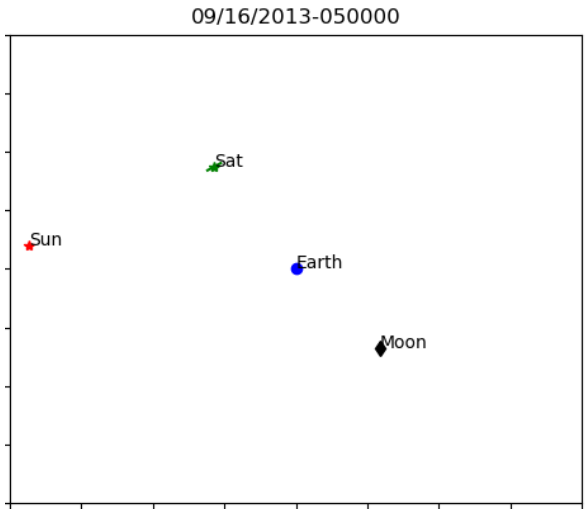


Figure 58 Left: Appearance of the moon (M7-MTP10-20130916050000). Right: schematic representation of situation, where the moon stray light anomaly can appear.

To calculate the orbit (in time) of the Moon, the Python library "Ephem" can be used. With the Ephem library the position (astrometric geocentric position) of the Moon with respect to the Earth (in time) can be calculated (see Figure 58). To calculate the astrometric geocentric position of the Moon, the time is required:

```
dateImage =     datetime(RECORD[1]["YY2TG"],1,1)
                + timedelta(days=RECORD[1]["DD2TG"]-1, minutes = RECORD[1]["SL2TG"]*30)
```

The astrometric geocentric position can be transformed to a xyz-position, which is in the same frame as the MGF (Mean Geometric Format) orbit coordinates (ORBF record 3 from level 1.0 file):

$$x_{moon} = \mathrm{D} * \cos(a_{dec}) * \cos(a_{ra})$$

$$y_{moon} = \mathrm{D} * \cos(a_{dec}) * \sin(a_{ra})$$

$$z_{moon} = D * \sin(a_{dec})$$

where D is the distance from earth to the moon, a_{dec} is the declination and a_{ra} is the right ascension.

The position of the satellite is defined in the metadata (ORBF record 3 from level 1.0 file). But this position holds for the moment of the start of the scan and during the scan, the position of the satellite continuously changes. We know that the satellite rotates in 24 hours around the Earth and the required time for generate one scanline is 0.6 seconds. To calculate the position of the satellite at the moment of a scanline was created, the following equation can be used:

$$r = \sqrt{ORBF_X \cdot ORBF_X + ORBF_Y \cdot ORBF_Y}$$

$$\theta_0 = atan2(ORBF_Y, ORBF_X)$$

$$dt = 0.6 * scanline\_nr$$

$$\theta_1 = \theta_0 + \frac{dt * 2 * \pi}{3600 * 24}$$

$$[ORBF\_1_X \quad ORBF\_1_Y \quad ORBF\_1_Z]^T = [r * \cos\theta_1 \quad r * \sin\theta_1 \quad ORBF_Z]^T$$

By assuming that the satellite is directed to the earth, the rotation of the satellite can be calculated:

$$v_z = -[ORBF\_1_X \quad ORBF\_1_Y \quad ORBF\_1_Z]^T$$

$$v_x = v_z \times [0 \quad 0 \quad -1]^T$$

$$v_y = v_x \times v_z$$

$$^{SAT}R_{earth} = [v_x \quad v_y \quad v_z]$$

The position of the moon with respect to the local frame of satellite can be calculated:

$$[^{SAT}X_{moon} \quad ^{SAT}Y_{moon} \quad ^{SAT}Z_{moon}]^T = R * [x_{moon} \quad y_{moon} \quad z_{moon}]^T$$

The image coordinates (u,v) can be calculated:

$$u = \frac{^{SAT}X_{moon}}{^{SAT}Z_{moon}} * \frac{1}{stepAngle}$$

$$v = \frac{^{SAT}Y_{moon}}{^{SAT}Z_{moon}} * \frac{1}{stepAngle}$$

where the stepAngle is the radiometer stepangle (F1SATG, record 3 from level 1.0 file).

With the calculated image-coordinates it can be determined if it is likely that the Moon is in the field-of-view. The defined calculations are approximations and we consider that the accuracy of the calculations is at least better than 300 pixels.