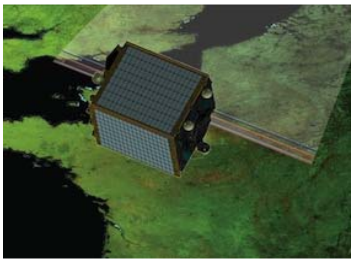
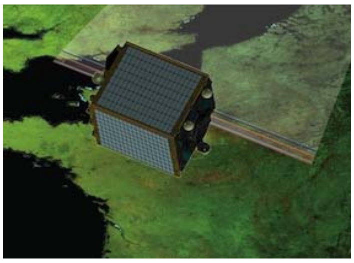
		**Reference:** PROBA-MD-ACS-VITO-0102 **Version:** 2.1 **Date:** *12-10-2011*

PROBA-V IQC-GC
Algorithm Teoretical Baseline Document
12-10-2011

Prepared by	L. Galli S. Mica	Date	
Verified by	A. Arledler	Date	
Approved by	A. Giustiniani	Date	



Distribution List

Distribution List					
Name	**Address**	**P/E**	**Name**	**Address**	**P/E**
Jan Dries	VITO	E			
Geert Duhoux	VITO	E			
Kris Nackaerts	VITO	E			
Veljko Jovanovic					

(P = Paper copy, E = Electronic version)

Document change record

Version	**Date**	**Description**	**Affected sections**
1.0	10/01/2011	Version submitted to the PDR	All
2.0	14/06/2011	Version submitted to the CDR	GC PDR RIDs: 018 Sec 2.2; 019 Sec. 5.2; 020 Sec. 5.2; 021 Sec. 5.3; 022 Sec. 5.4; 027 Sec. 8.2.2; 028 Sec. 8.2.3; 049 Sec. 6.5.8; 066 Sec. 4; 067 Sec. 4.2; 070 Sec. 5.2; 072 Sec. 5.3; 073 076 TP management removed; 078 Sec. 5.3. GC DDR RIDs: 016 Sec. 4.1. PV02-CDR RIDs: 062 Sec. 5.2; 075 8.2.2; 079 Sec. 4; 081 Sec. 7.1.3 & 7.3; 083 Sec. 7.3; 086 Sec. 4.1; 087 VNIR/SWIR relative calibration removed; 090 5.2. Completed all the TBW Sections (in response to all the related RIDs).

Version	Date	Description	Affected sections
2.1	12/10/2011	Version proposed to address the CDR-Rids on the document	GC-CDR RIDs: 001 Sec. 4; 024 Sec. 6.3 and 6.5.3; 029 added Sec. 4.1; 030 Sec. 4 (Fig. 4-2); 031 Sec. 4; 032 added Sec. 4.1; 035 Sec. 4.2 (Fig 4.2-2 and Fig. 4.2-3); 036 Sec. 2.2 and Sec. 5.2; 037 focal length eliminated Sec. 4.2, 4.3, 5.2, 8.1 and 8.2.1; 039 Sec. 5.6.2 and 5.6.3; 040 Sec. 7.2.3; 043 Sec. 8.2.2; 044 Sec. 8.2.3; 045 Sec. 8.2.2; 046 Sec. 8.2.6 and 8.2.7; 047 Sec. 8.2.7; 048 Sec. 8.1, 8.2.1 and 8.3.2; 049 Sec. 9.1 and 9.2; 088 Sec. 7.1.3. GC-PDR RIDs: 028 Sec. 2.2. PV02-CDR RIDs: 074 the Geometrical accuracy budget is included in Error Analysis document N77D7-PV02-US-09-IQC-GC-ARPT-ACS-0110-1.1; 082 Sec. 6.5; 088 Sec. 4.2.

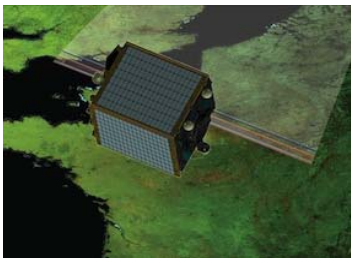
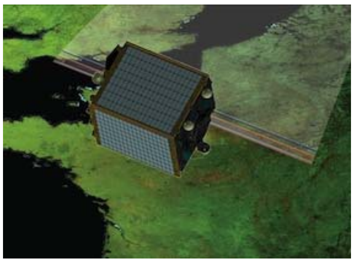
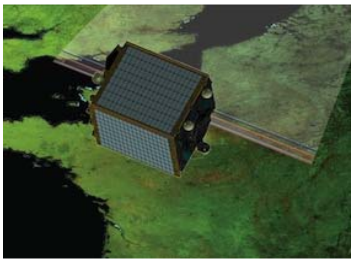
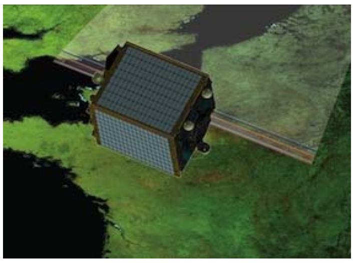


Table of contents







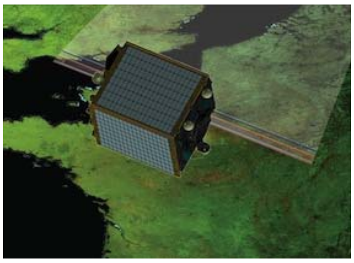
1. INTRODUCTION

1.1 Scope

This ATBD (Algorithm Theoretical Base Document) describes the processing to be implemented by the Proba-V Geometric Calibration facility. This document includes also information (functional decomposition, input/output parameters, etc.) which typically are part of the DPM (Detailed Processing Model) document. In addition, this ATBD document incorporates Error Analysis Report document and Sensitivity Analysis Report document.

1.2 Applicability

This document has to be provided at PDR in draft version, at CDR in almost definitive version and should be updated to account for any changes at successive milestones.



2. REFERENCES

All applicable and reference documents, are listed in the SOW [N77D7-PV02-PM-19-IQC-GC-SoW-v1_0-draft].

2.1 Applicable documents

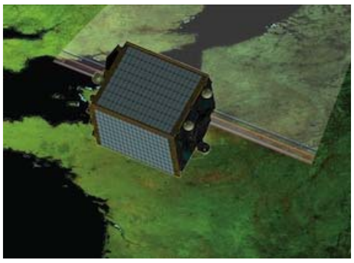
The documents that have been used for the preparation of underlying document are:

Doc. reference	Document Title
[MMIO]	Extraction of GCP chips from GeoCover using Modified Moravec Interest Operator (MMIO) algorithm
[SSD]	PROBA-DD-ACS-VITO-0103-IQC-GC PROBA-V IQC-GC-Software Design Document
[PF-DPM]	N77D7-PV02-US-6-PF-DPM-v1.2 PROBA-V Processing Facility Detailed Processing Model
[ERRB]	PROBAV-TN-01018-VE
[MOM-PM06]	N77D7-PV02-IQC-GC-MOM-PR-006-20110519 Minutes of Meeting Technical Meeting #06

Table 1: Applicable documents

2.2 Reference documents

Doc. reference	Document Title
[GLOS]	PROBA-V Terms, Definitions and Abbreviations N77D7-PROBA-V-TDA-v2_11
[R-1]	Daniela Poli (2005):Modelling of Spaceborne Linear Array Sensors; Diss.ETH NO. 15894; Institut für Geodäsie und Photogrammetrie; ETH Zürich; ISBN 3-906467-50-3.
[R-2]	Kornus, W., Lehner, M., Schroeder, M. (2000): Geometric in-flight calibration of the stereoscopic line-CCD scanner MOMS-2P ISPRS Journal of Photogrammetry & Remote Sensing, Vol. 55, pp. 59-71



Doc. reference	**Document Title**
[R-3]	Jacobsen, Karsten: Geometric Calibration of Space Remote Sensing Cameras for Efficient Processing; International Archives of Photogrammetry and Remote Sensing, Vol.32, Part I, pp. 33-43.
[R-4]	Udo Tempelmann, Ludger Hinsken, Utz Recke. ADS40 Calibration & Verification Process. Proceedings of the ISPRS WG1/5 International workshop.
[MIK]	E. M. Mikhail, J.S. Bethel and J. C. McGlone. Introduction to Modern Photogrammetry.

Table 2: Literature references

3. TERMS, DEFINITIONS AND ABBREVIATED TERMS

General terms, definitions and abbreviations used within the scope of this document can be found in [GLOS]. In addition, the following are used:

ACS	Advanced Computer Systems A.C.S.S.p.A.

3.1 Notations and Conventions

3.1.1 Indexing

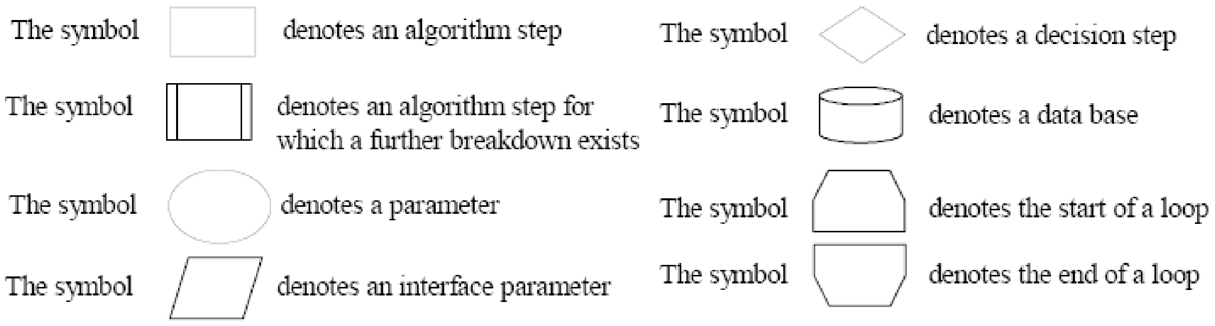
The subscripts of the array data structures shall be:

b	band (b $\in$ {1..B});
d	detector (d $\in$ {1..D});
l	detector line (l $\in$ {1..L(B)});
k	detector column (k $\in$ {1..K(B)});
j	Level 1 product column (j $\in$ {1..NC});
i	Level 1C product row (i $\in$ {1..NR});

……

unless otherwise specified.

3.1.2 Block diagrams symbols



The symbol denotes an algorithm step

The symbol denotes an algorithm step for which a further breakdown exists

The symbol denotes a parameter

The symbol denotes an interface parameter

The symbol denotes a decision step

The symbol denotes a data base

The symbol denotes the start of a loop

The symbol denotes the end of a loop

Arrows in the block diagrams indicate precedence: data input /output to a step or logical succession of steps.

4. GEOMETRIC CALIBRATION – AN OVERVIEW

The goal of geometric calibration of an optical sensor system is to model the line-of-sight for each pixel element of the imaging system. This is usually performed pre-flight in laboratory conditions where precise measurements enable to characterize the various aspects of the system.

In more detail, the main objective of the pre-flight camera geometric model calibration is to characterize those variables (i.e. parameters) in order to:

- verify that instrument science requirements (geometric) are satisfied before launch,
- provide input to the in-flight geometric calibration.

The various parameters require different types of calibration, given their sensitivities and expected changes during the life of the instrument. Some of the parameters can be measured simply by inspection or set at specified values during assembly. Other parameters, particularly those sensitive to temperature changes, require more complicated calibration approaches.

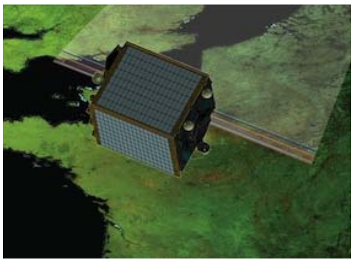
Due to possible launch effects, distinct operational conditions and thermal deformation, a post-launch self-calibration is performed on a regular basis (e.g. every 30 days) or/and when necessary to ensure geometric system stability. This is the goal of the Geometric Calibration facility.

PROBA-V is a small satellite, which provides only passive thermal control onboard, so temperature can play a major role in the geometric calibration process. This means that in order to ensure the total geometric system accuracy as specified in the PROBA-V requirements, a quite complex geometrical model is needed.

The in-flight geometrical calibration shall be able to estimate and monitor on a regular basis the boresight misalignment angles and Interior Orientation deformations at different conditions (e.g. temperature, time exit from eclipse) of each camera of the PROBA-V sensor (three cameras are present).

These geometrical calibration parameters shall be used to update the ICP (Instrument Calibration Parameter) file, which is used by the Processing Facility in order to guarantee the geometrical system accuracy of the system corrected products reported in the following table.

Geo-localization measurements	**Goal Accuracy (95%)**	**Mandatory Accuracy (95%)**
Inter-band (VNIR)	100 m	
Interband (SWIR + VNIR)	150 m	300 m
Multi-temporal (VNIR)	150 m	
Multitemporal (SWIR + VNIR)	225 m	500 m
Absolute (VNIR)	300 m	
Absolute (SWIR + VNIR)	450 m	1000 m



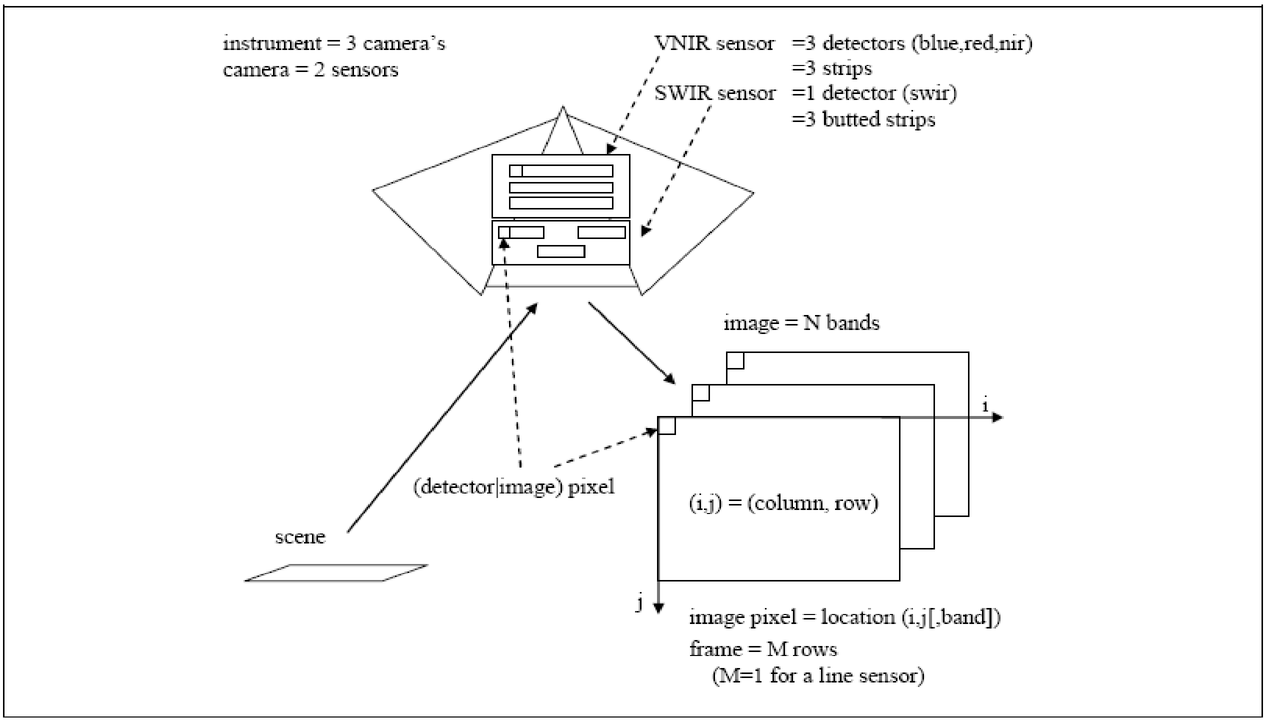


Figure 4-1: Conceptual overview

The definition of some concepts related to the naming of some data sets referred to in this ATBD is schematically depicted in Figure 4-1. The used convention is the same used in the PF DPM document [PF-DPM] and it is reported here for convenience.

The PROBA-V instrument is made up of three camera's. Each camera has two sensors: the VNIR sensor is associated with the visible bands and the SWIR sensor with the infra-red band. A strip is a pixel array on top of which a filter layer is deposited in order to make it only detect light in a certain [spectral] band. A detector is associated with a full swath line of observations in a given band. A detector corresponds to a single strip, or else to two or more strips that are somehow combined in order to cover the full instrument swath. More precisely, the VNIR sensor contains three detectors that each corresponds to a single strip, the SWIR sensor contains only one and it is made up of three strips.

Summarizing, the PROBA-V instrument has the following characteristics:

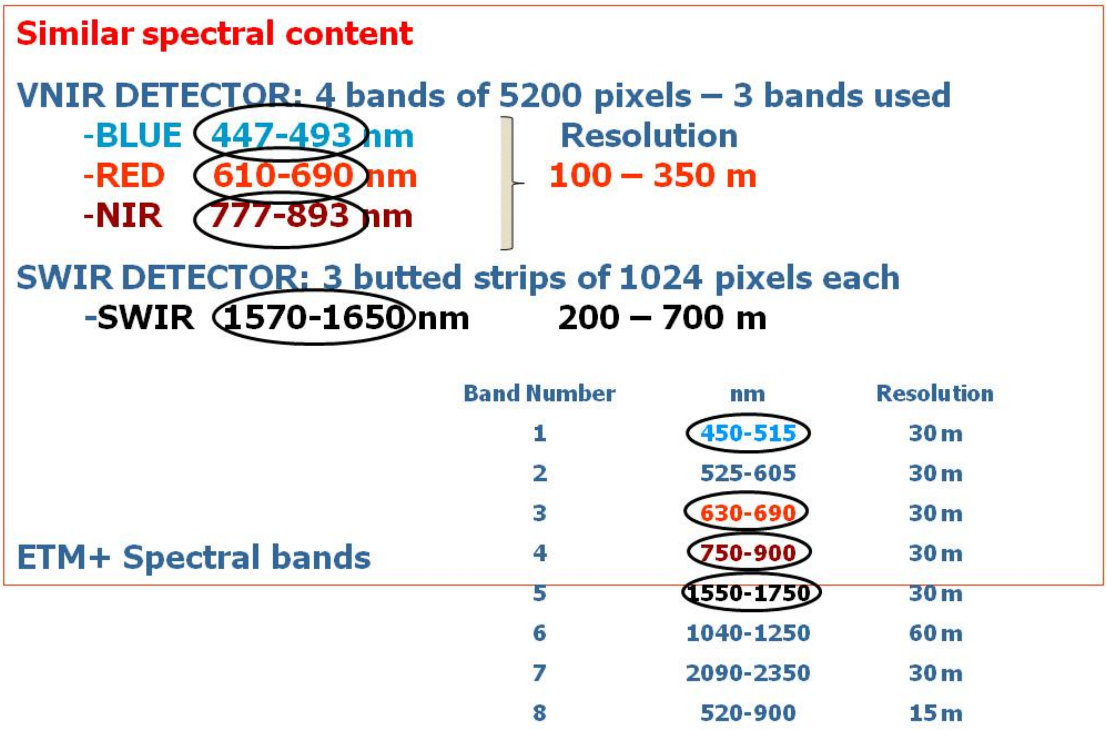
- 3 camera's, i.e. the Left camera, the Center camera and the Right camera
- 4 bands and hence 4 detectors/camera, i.e. BLUE, RED, NIR and SWIR
- 2 sensors/camera, i.e. the VNIR sensor (associated with the BLUE, RED and NIR bands) and the SWIR sensor (associated with the SWIR band)
- 6 strips/camera, i.e. BLUE, RED, NIR, SWIR1, SWIR2 and SWIR3

The GC system perform the geometric calibration starting from the Level 1C PROBA-V products. An image consists of one or more [image] bands, each consisting of a given number of columns and rows of [image] pixels:

- BLUE, RED and NIR strips are composed by 5200 pixels;
- SWIR1, SWIR2 and SWIR3 strips are composed by 1024 pixels;

Each pixel corresponds to a value of a given data type representing a specific type of information, e.g. a radiometric value (in the form of a DN, etc.), lon/lat coordinates, etc.

The lines of observations returned by a strip are (transmitted to earth and) stored in successive rows of a band. Each [image] pixel in each row, and hence each column in the band, corresponds to a given detector pixel.



Similar spectral content

VNIR DETECTOR: 4 bands of 5200 pixels – 3 bands used

-BLUE 447-493 nm

-RED 610-690 nm

-NIR 777-893 nm

Resolution

100 – 350 m

SWIR DETECTOR: 3 butted strips of 1024 pixels each

-SWIR 1570-1650 nm 200 – 700 m

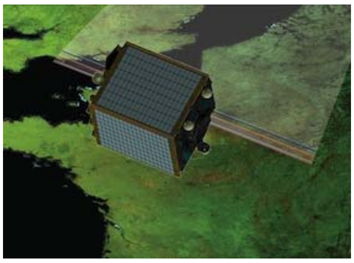
ETM+ Spectral bands

Band Number	nm	Resolution
1	450-515	30 m
2	525-605	30 m
3	630-690	30 m
4	750-900	30 m
5	1550-1750	30 m
6	1040-1250	60 m
7	2090-2350	30 m
8	520-900	15 m

Figure 4-2: Rationale of the PROBA-V/Landsat Geocover bands selection for the geometric calibration. Details are given in Section “GCP Chips Generation”.

The calibration is made by registering each L1 C band directly and independently with corresponding Landsat Geocover band chips:

- NIR band is used to calibrate the boresight angles and NIR band focal plane distortions;
- Other bands are used to calibrate the respective focal plane distortions;



The rationale of the PROBA-V/Landsat Geocover bands selection for the geometric calibration is reported in Figure 4-2, more details can be found in “GCP chips generation” section.

Sensor Exterior and Interior Orientation parameters are expected to depend on camera CCD or TMA temperature and on the time starting from eclipse.

Two different strategies for estimating the Sensor Exterior and Interior Orientation parameters are foreseen:

- Baseline calibration strategy at scene level;
- Optional calibration strategy (if needed) at global level. This option is not currently implemented but it is possible by a relative simple reconfiguration of the production chains (see [SDD]).

4.1 Assumptions and Limitations

The basic assumption of the **baseline calibration strategy** is that a scene size can be chosen so that the orientation parameters can be considered constant within the scene and that the scene is large enough to make a reasonable estimate of the parameters. The parameters can therefore be estimated for each scene separately and the parameter trend can be plotted vs. temperature and/or orbit position.

If necessary the ‘global solution’ (**optional calibration strategy**) is possible by a relative simple customization of the SW: the Sensor Exterior and Interior Orientation parameters are estimated considering a set of (multiple) scenes acquired at the same temperature and time starting from eclipse. The advantage of the ‘global solution’ is that the number of ground control points increases allowing an accurate estimation of the parameters in case of failure of the baseline calibration strategy.

However, it is expected that the baseline calibration strategy is fully adequate for PROBA-V calibration. In fact, sufficient number of GCPs (about 1600) are expected in a scene 800 km long acquired over land for each camera.

This estimation is done by using the updated version of the MMIO SW originally developed for Landsat TM/ETM+ for Proba-V as described in section 6. We used 8 Landsat scenes over land obtaining about 100 GCP chips for each scene and per band. Considering that a Proba-V scene of 800 km x 800 km is composed by about 16 Landsat scenes the number of GCP chips per band is about 1600. Please note that if needed the number of GCPs can be increased by changing the threshold value for the Moravec Interest Operator (see 6.5.4) without significantly affecting the quality of the GCPs (e.g. eliminating low contrast features like water bodies, desert, dry lands, etc. in the GeoCover scenes).

Of course only a fraction of them are effectively used for the calibration (seasonal effects, clouds, etc.), but it is expected to have a sufficient number of GCP used for the calibration.

Please note that:

1) MMIO extracts chips well distributed over the image avoiding that most of the chips are concentrated over a small region of the image;

2) calibration parameters errors are computed via error propagation (section 8.2.5) and these errors shall be considered during the least square fitting; it means that in the case that a scene have less GCPs in same part of the field of view larger errors are expected and the contribution to the final fitting is less important.

In addition, during the calibration campaign several scenes at the same temperature/latitude shall be processed therefore the final calibration parameters shall be a weighted average value with a reduced error with respect to the parameters extracted from a single scene.

The main expected advantage of this method is that the computationally expensive part of the calibration processing can be done individually for each scene without making large assumptions on which known environment parameters the distortions depend. If the distortions actually depend on position in the orbit, or camera TMA temperature, or have a particular drift in time, this dependency can be noted from plots of scene calibration parameters, the sensor model can be modified, and new calibration parameters can be computed, without repeating the computationally expensive part of the processing. This method would not be the most effective if the distortions where to depend only on CCD temperature, exactly as expected, and the latter were to oscillate rapidly, so that the scene length would have to be too short to work. But this scenario does not seem probable at the moment.

The number of ground control points is crucial for the end-result of the per camera approach: it is expected that the number of GCPs per scene and camera extracted by the MMIO tool is adequate.
However, if necessary the number of ground control points can be increased:

- by changing the threshold value for the Moravec Interest Operator (see 6.5.4);
- by using the 'global solution' is possible by a relative simple customization of the SW (see above).

4.2 Baseline Geometric Calibration Dataflow

The diagrams that follow describe the high-level processing flows for baseline Geometric Calibration processing algorithms. In these figures, within each box indicating the principal algorithm processing step is reported the corresponding Section where the algorithmic method is described.

In all of these diagrams, the following naming convention is used:

- LoS model; set of parameters used to model the sensor Exterior Orientation parameters (boresigh angles) and Interior Orientation parameters (principal point offsets, polynomial CCD distortions) for each spectral band and camera. The LoS model is the result of an overall geometric calibration campaign performed over a set of PROBA-V scenes;

- Scene parameter values; sensor Exterior Orientation parameters (boresigh angles) and Interior Orientation parameters (principal point offsets, polynomial CCD distortions) extracted from the geometric calibration of a single scene;

- Scene parameter errors; errors associated to the Scene parameter values (Error analysis);

- Calibration Report contains the errors relative to the boresight angles and to the CCD pixel viewing directions;

The sensor Exterior Orientation parameters (boresight angles) are common among all the bands, and these parameters are calibrated using NIR images.

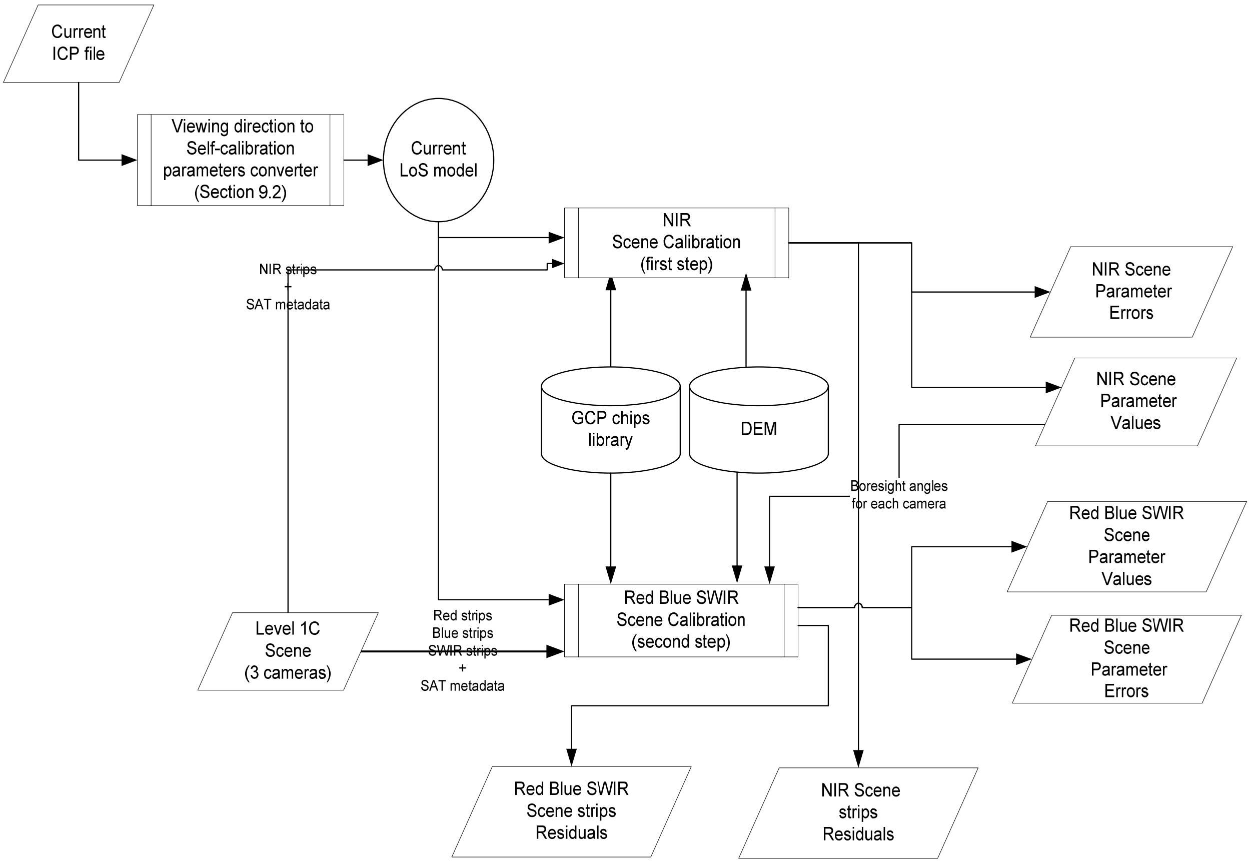

Current ICP file
Viewing direction to Self-calibration parameters converter (Section 9.2)
Current LoS model
NIR Scene Calibration (first step)
NIR strips + SAT metadata
NIR Scene Parameter Errors
NIR Scene Parameter Values
GCP chips library
DEM
Boresight angles for each camera
Red Blue SWIR Scene Parameter Values
Red Blue SWIR Scene Calibration (second step)
Level 1C Scene (3 cameras)
Red strips Blue strips SWIR strips + SAT metadata
Red Blue SWIR Scene Parameter Errors
Red Blue SWIR Scene strips Residuals
NIR Scene strips Residuals

Figure 4.2-1: Scene Calibration composed by NIR scene calibration and Red-Blue-SWIR scene calibration

Figure 4.2-2 illustrates in more detail the NIR scene calibration processing flow. Absolute NIR geo-localization errors are computed during the NIR calibration by performing appropriate statistic computation on the GCPs residuals (difference between the observed and predicted/computed GCPs coordinates). The GCP residuals are computed on a set of GCPs used to validate the geolocation accuracy performance of the geometric calibration, and which are not used in the scene inversion model. In particular, scene residual includes for each strip:

- List of line/pixel position for each GCP found by image correlation;
- List of line/pixel position for each GCP predicted by using current version of the ICP

- List of line/pixel position for each GCP predicted by using the best fit (inversion model) on that scene
- List of flags indicating if the GCPs are used for inversion or not
- List of SNR correlation value for each GCP.

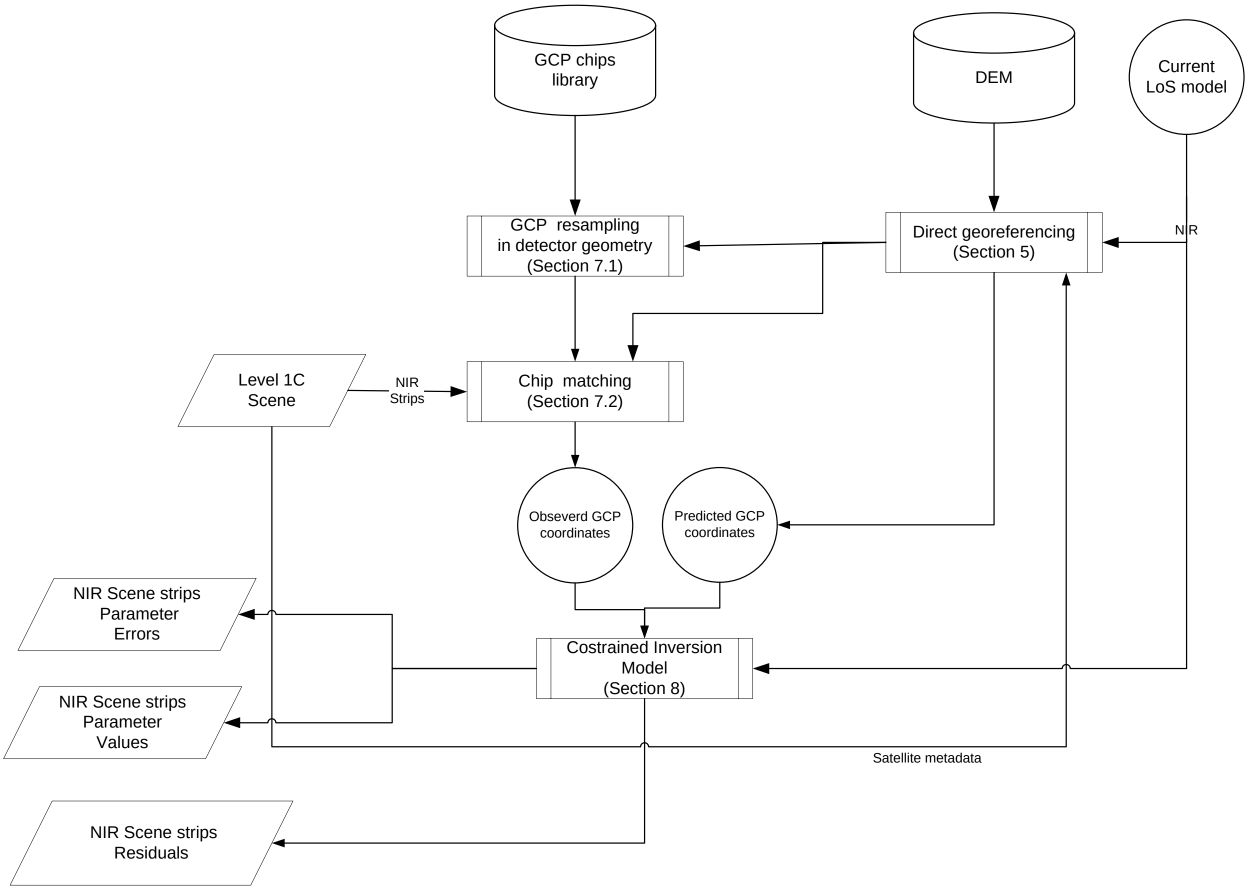

GCP chips library
DEM
Current LoS model
GCP resampling in detector geometry (Section 7.1)
Direct georeferencing (Section 5)
NIR
Level 1C Scene
NIR Strips
Chip matching (Section 7.2)
Obseverd GCP coordinates
Predicted GCP coordinates
NIR Scene strips Parameter Errors
NIR Scene strips Parameter Values
Costrained Inversion Model (Section 8)
Satellite metadata
NIR Scene strips Residuals

Figure 4.2-2: NIR scene strips calibration (performed first)

Figure 4.2-3 illustrates in more detail the Red, Blue and SWIR calibration processing flow. Again, absolute Red, Blue and SWIR geo-localization errors (monitoring) are computed during the calibration by performing appropriate statistic computation on the GCPs residuals (difference between the observed and predicted/computed GCPs coordinates). Again, the GCP residuals are computed on a set of GCPs used to validate the geolocation accuracy performance of the geometric calibration, and which are not used in the scene inversion model

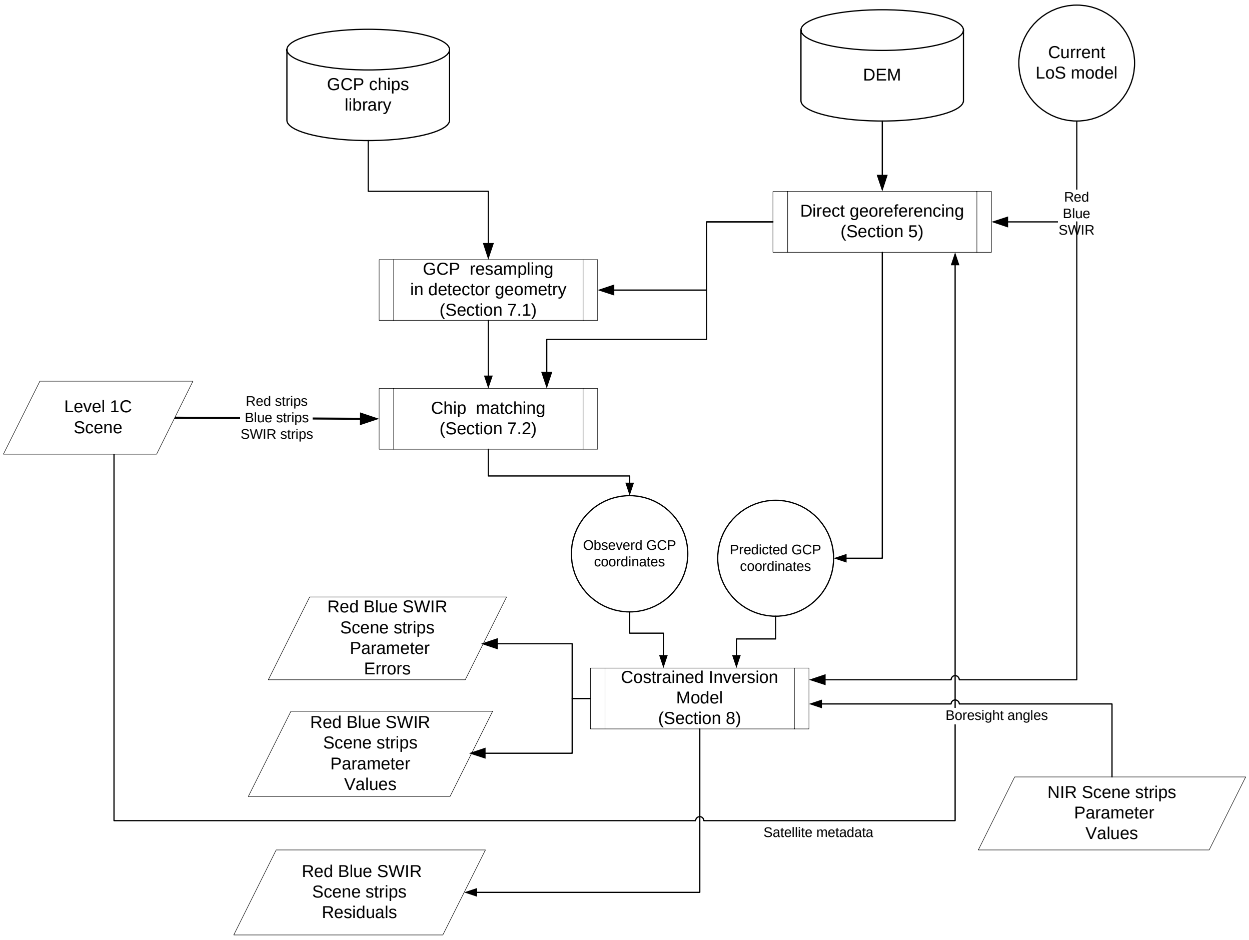

GCP chips library
DEM
Current LoS model
Direct georeferencing (Section 5)
Red
Blue
SWIR
GCP resampling in detector geometry (Section 7.1)
Level 1C Scene
Red strips
Blue strips
SWIR strips
Chip matching (Section 7.2)
Obseverd GCP coordinates
Predicted GCP coordinates
Red Blue SWIR Scene strips Parameter Errors
Red Blue SWIR Scene strips Parameter Values
Costrained Inversion Model (Section 8)
Boresight angles
NIR Scene strips Parameter Values
Satellite metadata
Red Blue SWIR Scene strips Residuals

Figure 4.2-3: Red-Blue-SWIR scene calibration (performed after NIR band calibration)

From PV01 it is clear that there will be multiple temperature readings required spread over the instrument to define the relation to the viewing angles.

However, currently the parameters are expected to depend linearly on a single temperature per camera (the temperature or a combination of temperatures having the highest impact due to distortions of the optical elements), so the parameters can be linearly fitted against camera temperature to determine the most probable linear law according to Figure 4.2-5. A weighted least squares fit is used according to what specified in Section 9.

It is important to point out that the input "current LoS model" in the first calibration campaign shall be generated starting from the “starting” ICP file. It is assumed that this “starting” ICP file is externally generated (outside the GC facility) using on-ground calibration and measurement data together with the validated thermal elastic model of PV01. For the following calibration campaigns (later on) the input ICP file is the latest one generated by the GC facility.

Figure 4.2-5 shows the Sensitivity Analysis processing flow which is described in detail in Section 8.4.

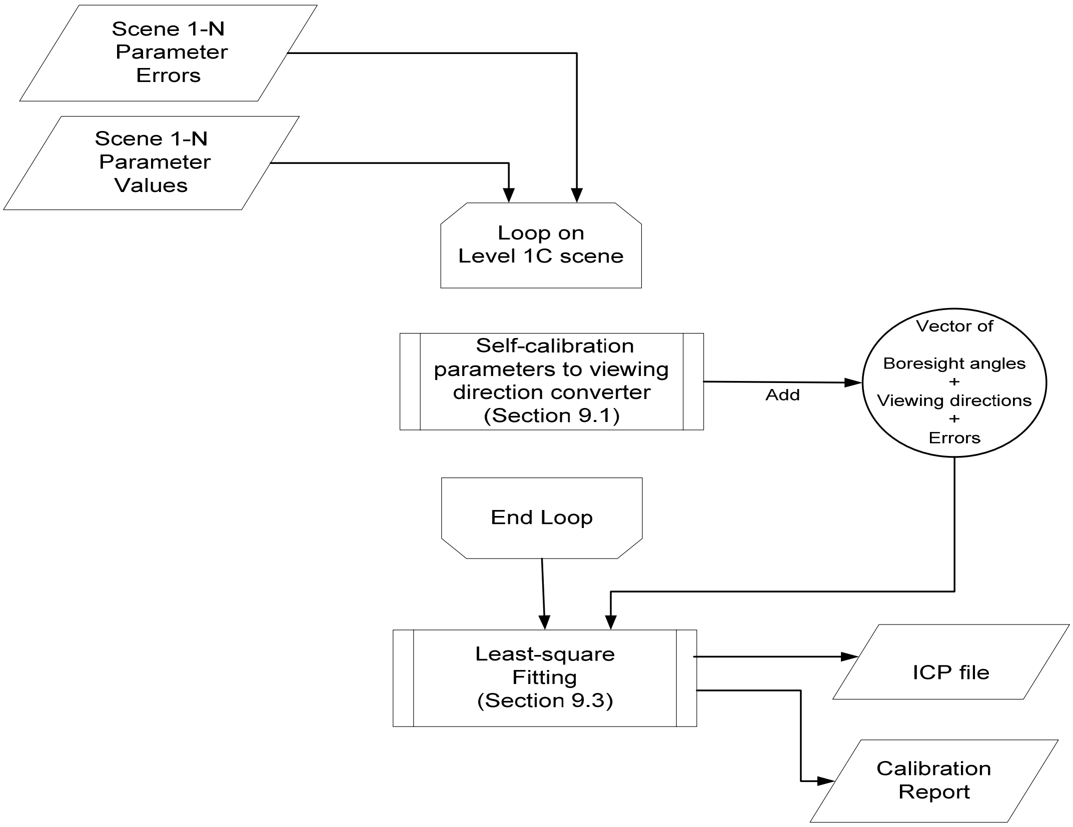

Scene 1-N Parameter Errors
Scene 1-N Parameter Values
Loop on Level 1C scene
Self-calibration parameters to viewing direction converter (Section 9.1)
Add
Vector of
Boresight angles
+
Viewing directions
+
Errors
End Loop
Least-square Fitting (Section 9.3)
ICP file
Calibration Report

Figure 4.2-4: Temperature variation Calibration and ICP file generation. In this diagram Interior Orientation parameters (principal point offsets, and polynomial CCD distortions) are called Self-Calibration parameters

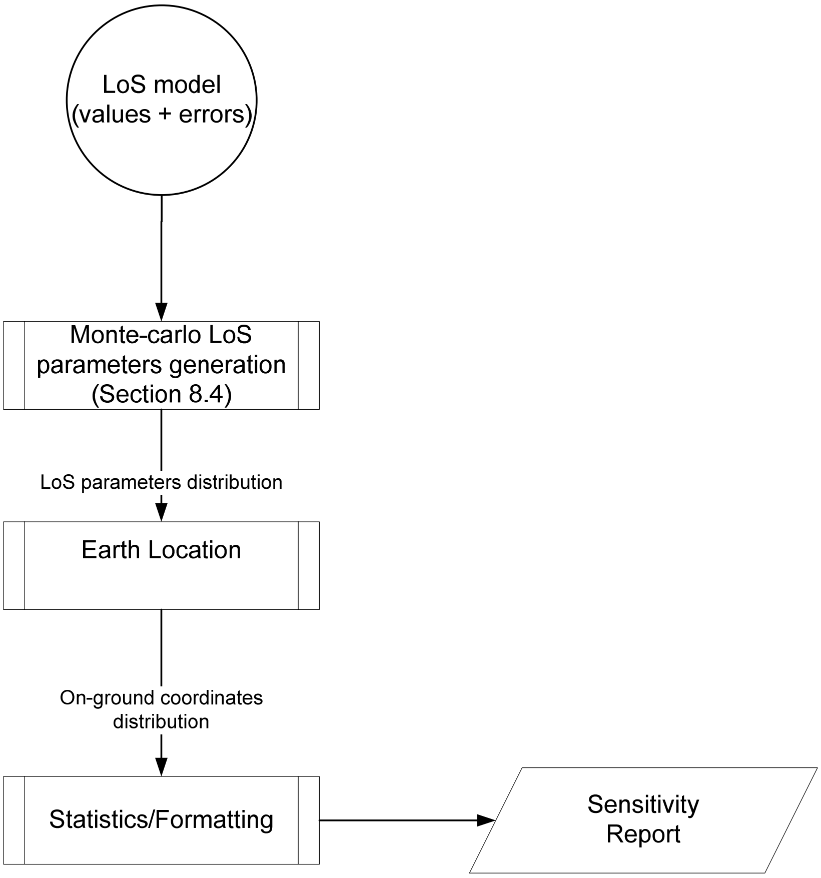


Figure 4.2-5: Sensitivity Analysis

4.3 Optional Geometric Calibration Dataflow

The diagrams that follow describe the high-level processing flows for optional Geometric Calibration strategy at global level.

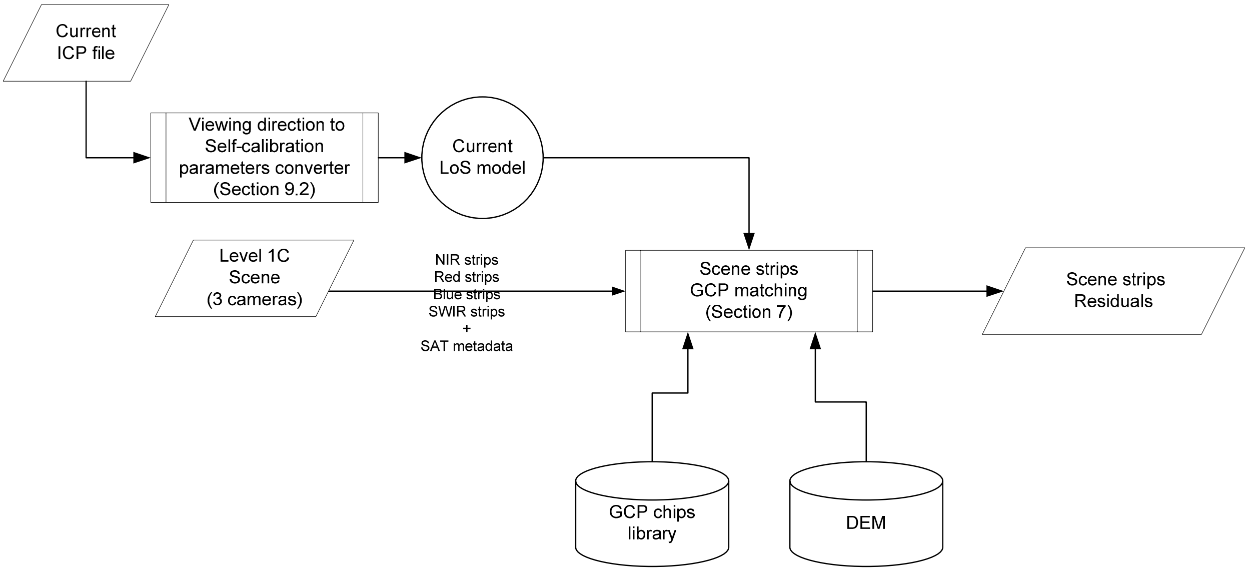


Figure 4.3-1: NIR and Red-Blue-SWIR Scene GCP matching

The GCP chips extraction and matching is performed independently for each band and camera (more exactly for each strip) and the result is collected in the scene strip residual file containing the GCPs residuals (difference between the observed and predicted/computed GCPs coordinates).

In particular, scene strip residual includes for each strip:

- List of line/pixel position for each GCP found by image correlation;
- List of line/pixel position for each GCP predicted by using current version of the ICP
- List of flags indicating if the GCPs are used for inversion or not
- List of SNR correlation value for each GCP.

The “global level” geometric calibration is illustrated in Figure 4.3-1 (NIR band) and Figure 4.3-2 (other bands). It is achieved by performing the inversion model over a set of scenes acquired at different times but with the same expected thermal deformations, e.g. with the same temperature and exit time from eclipse (within some margins).

The corresponding GCP residuals of this set of scenes are indicated in Figure 4.3-1 and Figure 4.3-2 as: Scene #1 Residuals, Scene #2 Residuals, .., Scene #N Residuals.

In line with the baseline approach, the sensor Exterior Orientation parameters (boresight angles) are common among all the bands, and these parameters are calibrated using NIR images (e.g. NIR calibration is performed first).

The output of the global-level geometric calibration are:

- Multi-scene Parameter values; sensor Exterior Orientation parameters (boresigh angles) and Interior Orientation parameters (principal point offsets, polynomial CCD distortions) extracted from the geometric calibration of the set of input scenes;
- Multi-scene Parameter errors; errors associated to the Multi-scene Parameter values (Error analysis);
- Scene strip complete residuals (a separate file for each scene); the scene strip residuals (see above) including the list of line/pixel position for each GCP predicted by using the global-level best fit (inversion model).

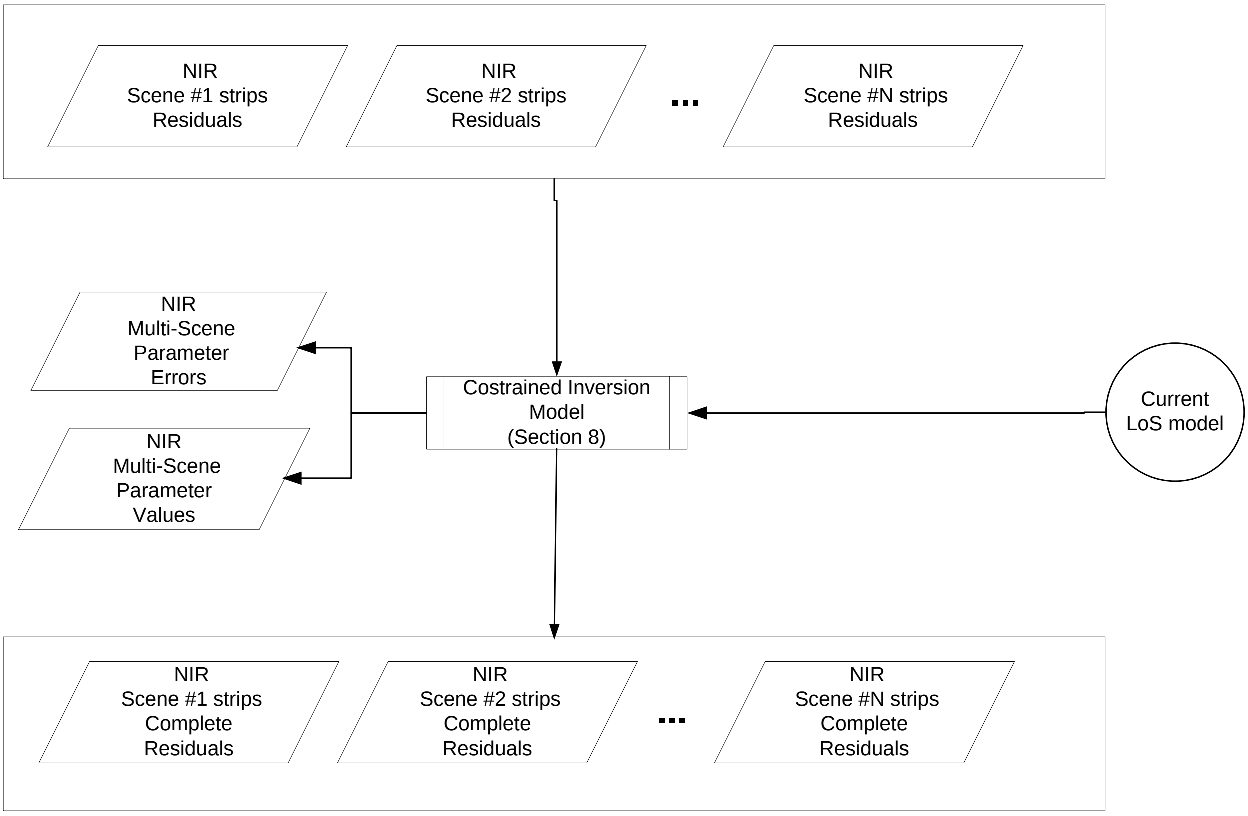


Figure 4.3-2: NIR global-level geometric calibration (performed first)

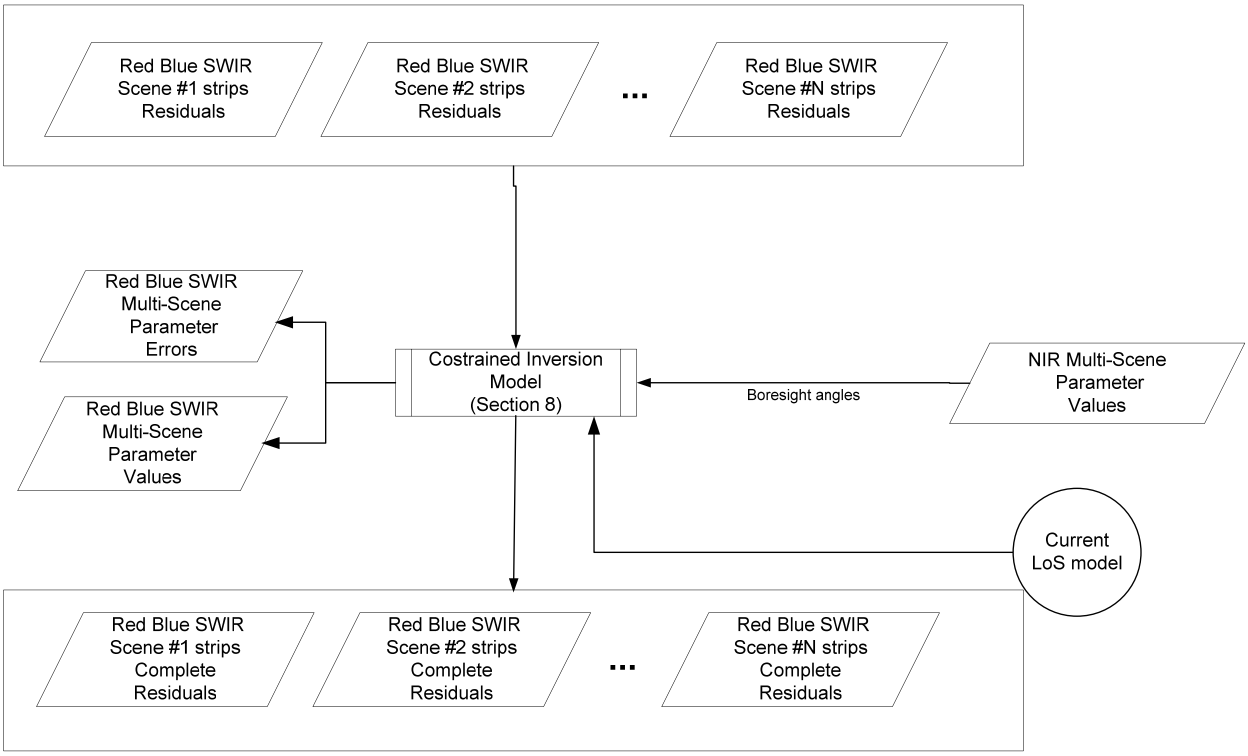
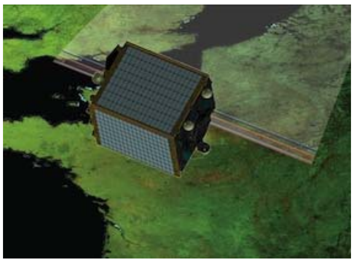

Red Blue SWIR Scene #1 strips Residuals
Red Blue SWIR Scene #2 strips Residuals
...
Red Blue SWIR Scene #N strips Residuals
Red Blue SWIR Multi-Scene Parameter Errors
Red Blue SWIR Multi-Scene Parameter Values
Costrained Inversion Model (Section 8)
Boresight angles
NIR Multi-Scene Parameter Values
Current LoS model
Red Blue SWIR Scene #1 strips Complete Residuals
Red Blue SWIR Scene #2 strips Complete Residuals
...
Red Blue SWIR Scene #N strips Complete Residuals

Figure 4.3-3: Red-Blue-SWIR global-level geometric calibration (performed after NIR band calibration)

The ICP generation method and the sensitivity analysis approach are the same of the baseline geometric calibration (see correspondingly dataflow reported in the previous section).



5. DIRECT GEOREFERENCING

5.1 Introduction

In order to design and implement the Geometric Sensor Model an accurate description of Line-of-Sight (LoS) function of imager sensor is required, including a careful characterization of the image/pixel generation process: from the focal plane location of an instrument pixel (sensor's co-ordinate frame) to Earth surface location in terms of latitude, longitude and altitude.

Sensor line of sight vectors in the sensor coordinate frame must be related to object (ground) space by determining the orientation and position of the sensor co-ordinate system with respect to object space (TER coordinates) as a function of time.

The sensor orientation is determined using the sensor pointing model which includes the effects of sensor to spacecraft (navigation reference base) alignment, spacecraft attitude as determined by the on-board computer using star sightings. The inputs to this model are the orbital position of the spacecraft, attitude quaternions (star trackers), sensor alignment matrix and optical characteristics of the sensor system measured pre-launch and updated in flight by the Inverse Georeferencing using the sensor alignment calibration algorithm described in the following section.

For PROBA-V the LoS model shall take into account:

- a payload composed of three TMA (Three Mirror Anastigmatic) camera's
- VNIR and SWIR detectors are mounted on different locations but normally lie in the same focal plane
- VNIR and SWIR have different GDS
- SWIR detector is mechanically butted: it is formed by 3 overlapping detectors of 1024 pixels and approximately 80 pixels in the overlap area.
- there is only passive thermal control onboard, so temperature can play a major role in the geometric sensor modelling.

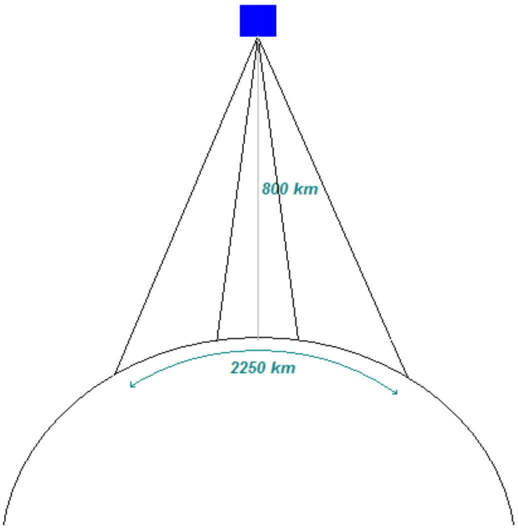


Figure 5.1-1: Schematic view of the global swath acquisition via three TMA camera’s

5.2 LoS model

In general the interior orientation of a line scanner describes the mapping from pixel numbers (image pixel column and row) to 3 dimensional Cartesian co-ordinates in the detector coordinate system, using ideal sensor coordinates $\hat{x}, \hat{y}$, the principal point shifts x_{PP} , y_{PP} and the additional parameters Δx and Δy, which account for deviations from the ideal case. The line-of-sight (LoS) vector is given by:

$$i \mapsto \mathbf{r}^{Sensor} = \begin{pmatrix} x_i \\ y_i \\ -f \end{pmatrix} = \begin{pmatrix} \hat{x}_i - x_{PP} - \Delta x \\ \hat{y}_i - y_{PP} - \Delta y \\ -f \end{pmatrix}$$

with i = 0, ..., N-1 where N = number of pixels in one scan line and f the focal length of the optical system of the scanner.

For the ideal push-broom sensor model the along track displacement of all recorded pixels of a scan line is set to zero. For the across track pixel co-ordinates a constant spacing of successive pixels is assumed. The focal length in pixel units is calculated using the maximum scan angle θ_{max} and the number of pixels N in one scan line

$$i \mapsto \begin{pmatrix} x_i \\ y_i \\ -f \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{N-1}{2}+i \\ -\frac{N-1}{2} \Big/ \tan\frac{\theta\max}{2} \end{pmatrix} \text{ with } i = 0,\cdots,N-1$$

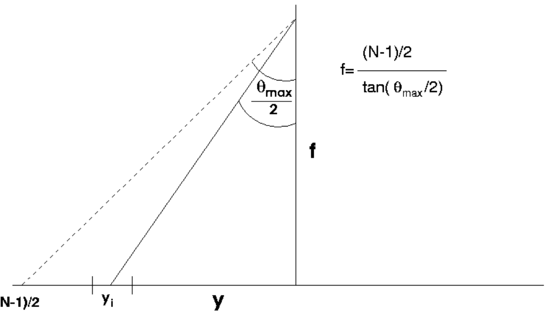


Figure 5.1-1: Ideal push-broom sensor model

In the following several geometric deviations (additional parameters & principal point and focal length) from the ideal case are listed:

- displacement of the CCD line in the focal plane w.r.t the nominal position
- rotation of the CCD line in the focal plane w.r.t the nominal position
- line bending
- change of pixel dimension
- local pixel displacements

Additionally for the optical system (lens)

- displacement of the lens principal point
- change of the optical focal length
- symmetric lens distortions
- scale variations

Different models exist to describe the interior orientation.

In [R-3] a set of 28 additional parameters is described, whereas for different sensors a subset (due to high correlations) has to be selected. In [R-2] 2 additional parameters (rotation, line bending) are used for the calibration of the MOMS camera together with pre-flight geometric correction tables. In [R-1] a generic sensor model with 8 additional parameters is proposed.

The PROBA-V camera has been modeled as flat focal plane as shown in the following figure.

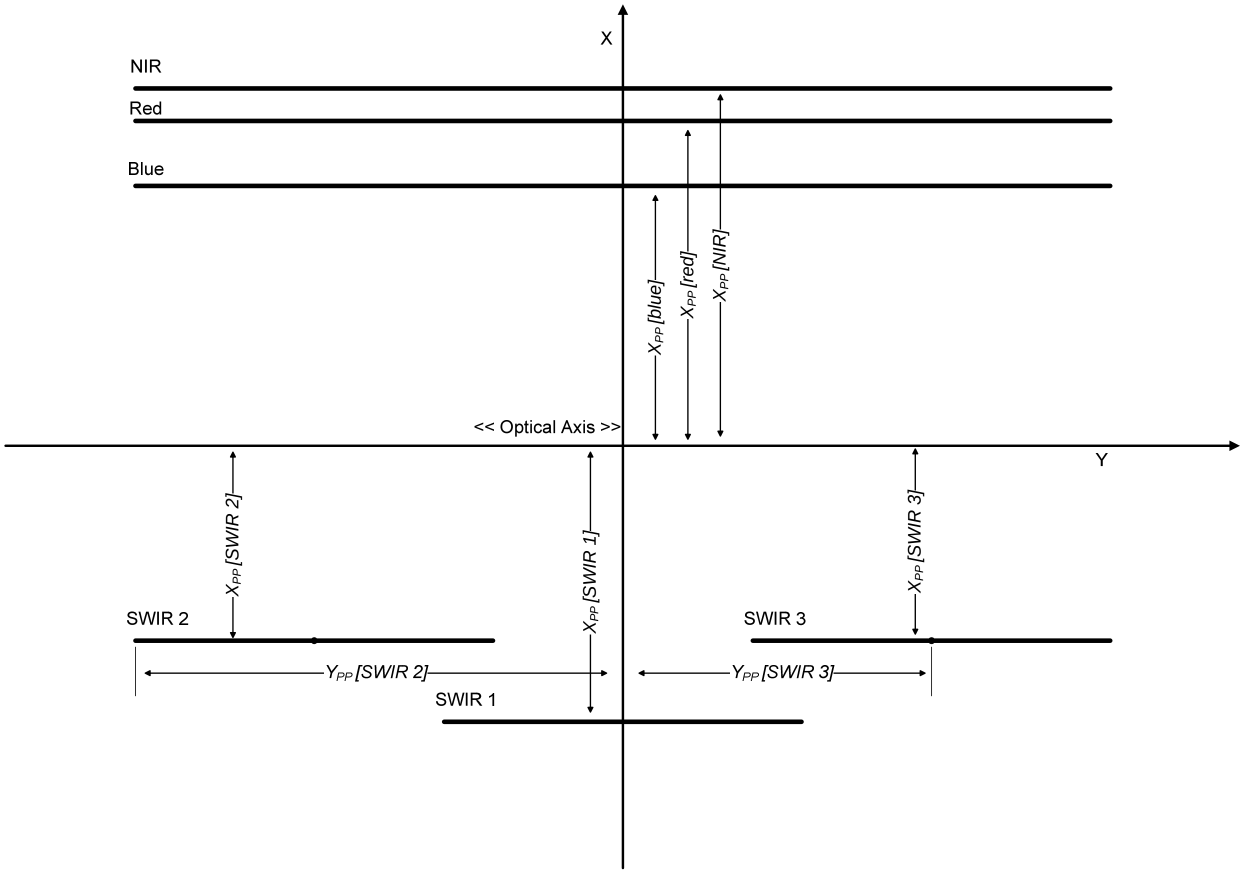
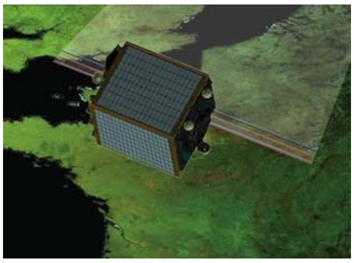


Figure 5.1-2: Focal plane layout

The nominal values extracted from the PV01 System Performance Simulator csv files are :

- Xpp[blue] = 3.786 mm
- Xpp[red] = 5.226 mm



- Xpp[NIR] = 5.946 mm
- Xpp[SWIR1] = -2.263 mm
- Xpp[SWIR2] = -3.763 mm
- Xpp[SWIR3] = -2.263 mm
- Ypp[SWIR1] = -21.975 mm
- Ypp[SWIR3] = 21.975mm
- Ypp is zero for the remaining strips;
- The focal plane is at a nominal focal length of 110mm with constant pixel spacing of 0.013 mm and 0.025 mm for VNIR and SWIR respectively.

The PROBA-V optic has a TMA (Three Mirror Anastigmatic) design which means that the focal lengths are different for every pixel in the focal plane! Table of interior orientation values (or look angles) that maps each pixel in a line to a Cartesian vector and/or functions describing the mapping process of the camera (for example nth order polynomials, where the polynomial coefficients can be also functions of e.g. thermal influences) shall be carefully taken into account. This approach provides the potential to compensate for complex distortions related to the specific and novel TMA camera design. Leica uses this approach for their three line ADS40 scanner (Tempelmann et al., 2003) [R-4].

There is not active thermal control onboard, so temperature can play a major role in the geometric calibration process. This means that in order to ensure the total geometric system accuracy as specified in the requirements of the PROBA V, a quite complex geometrical model is needed.

The boresight misalignment angles and Interior Orientation deformations describe the thermal influences during orbit revolution (varying solar exposure of the satellite). They have to be determined periodically by a ground control field (see also “constrained inversion model”), in we do expect three temporal scales that need to be taken into account:

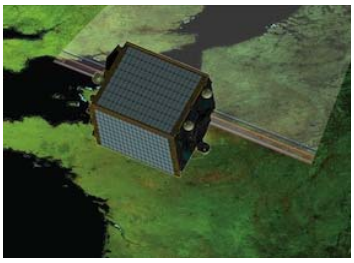
- Intra-orbit variations (cyclic), due to changes of sun irradiation
- Seasonal variations
- Long term drifts due to degradation effects

For a given camera *C* and spectral imager *band*, the viewing direction vector (LoS) in the detector coordinate system (spectral imager SI coordinate frame see [PF-DPM]) is parameterized in the following way:

$$\begin{pmatrix} \boldsymbol{x_i^{band,C}(T)} \\ \boldsymbol{y_i^{band,C}(T)} \\ \boldsymbol{-f^{band,C}(T)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\hat{x}_i^{band} - x_{PP}^{band,C}(T) - \Delta x_i^{band,C}(T)} \\ \boldsymbol{\hat{y}_i^{band} - y_{PP}^{band,C}(T) - \Delta y_i^{band,C}(T)} \\ \boldsymbol{-f^{band,C}(T)} \end{pmatrix}$$

where:

- *i* = 0, ..., *N* - 1 where *N* = number of pixels of spectral imager *band* and camera *C*



- T is the temperature;
- $\hat{x}_i^{band}$ and $\hat{y}_i^{band}$ are the ideal sensor coordinates
- $f^{band,C}(T)$ is the effective focal length;
- $x_{PP}^{band,C}(T)$ and $y_{PP}^{band,C}(T)$ are the principal point shift (position of the CCD detector central pixel in the spectral imager SI coordinate frame);
- $\Delta x_i^{band,C}(T)$ and $\Delta y_i^{band,C}(T)$ are the CCD displacements with respect to the ideal sensor coordinate

From PV01 it is clear that there will be multiple temperature readings required spread over the instrument to define the relation to the viewing angles. However, currently the parameters are expected to depend linearly on a single temperature per camera (the temperature or a combination of temperatures having the highest impact due to distortions of the optical elements).

The following self-calibration parameters shall be analyzed and implemented during the project:

- General polynomial model of degree M:
  - $\Delta x^{band,C}(T) = \sum_{j=1}^{M} a_j^{band,C}(T) x^j$
  - $\Delta y^{band,C}(T) = \sum_{j=1}^{M} b_j^{band,C}(T) x^j$

o

Please note that the principal point shift is equivalent to polynomial constant term.

A preliminary analysis on the simulated data has been done. The main outcome is that this general polynomial model gives satisfactory results considering a 8th order of polynomial coefficients.

The following table gives an overview of the set of parameters to be calibrated for a given PROBA-V scene (approximately at constant temperature).

Calibration Parameters	Number of parameters
Exterior Orientation Parameters	
Boresigh angles for each camera	3 (angles) x 3 (cameras) = 9
Interior Orientation Parameters	
Principal Point for each camera and strip	2 (PP) x 3 (cameras) x 6 (strip) = 36
Self-Calibration Parameters	
Polynomial model for each camera and strip of degree M	M x 3 (cameras) x 6 (strip) = 18 * M

The number of calibration parameters to extract from a single strip scene (via GCPs correlation) is **6 + M**, where M is the degree of the self-calibration polynomial. Prevent

Figure 5.1-3 shows the distortions of the viewing angles extracted from the PV01 System Performance Simulator csv files when projected, with pin-hole geometry, on a nominal focal plane with the layout shown above, 110 mm focal length and 0,013 mm pixels (0,025 mm for SWIR).

Note that the greatest errors are at the far side of the NIR sensor in the across track (SICy) direction, and correspond to approximately 50 pixels.

The successive plots depicted in Figure 5.1-4, Figure 5.1-5 and Figure 5.1-6 show that the temperature dependence is much smaller (≤ 3 pixels) and linear.

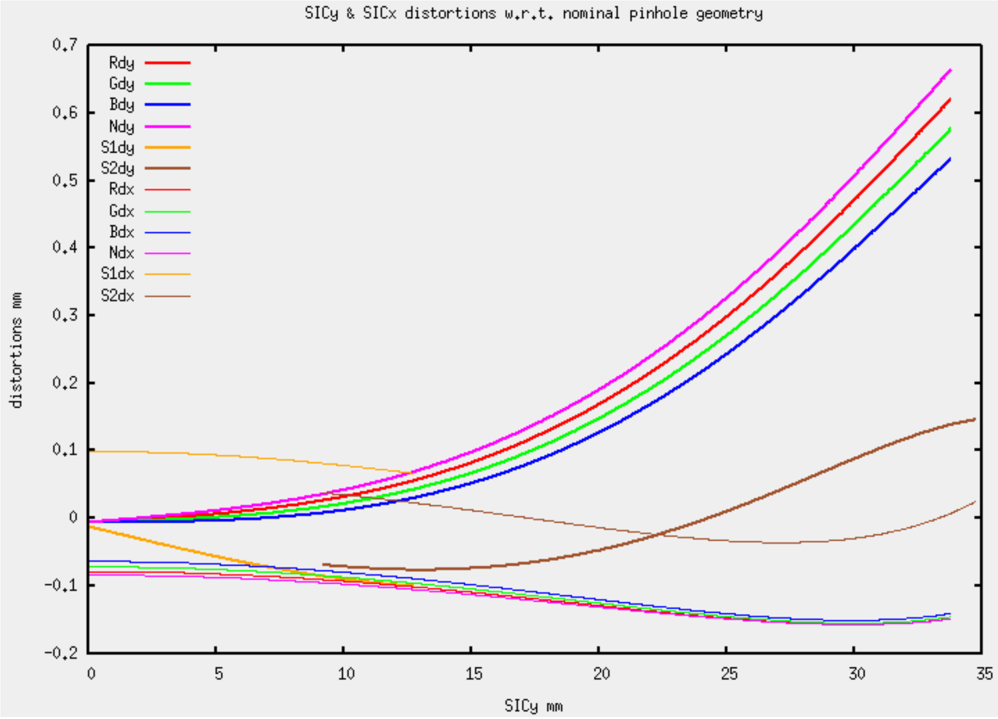


Figure 5.1-3: SIC distortions (both x and y directions) with respect to nominal pinhole geometry

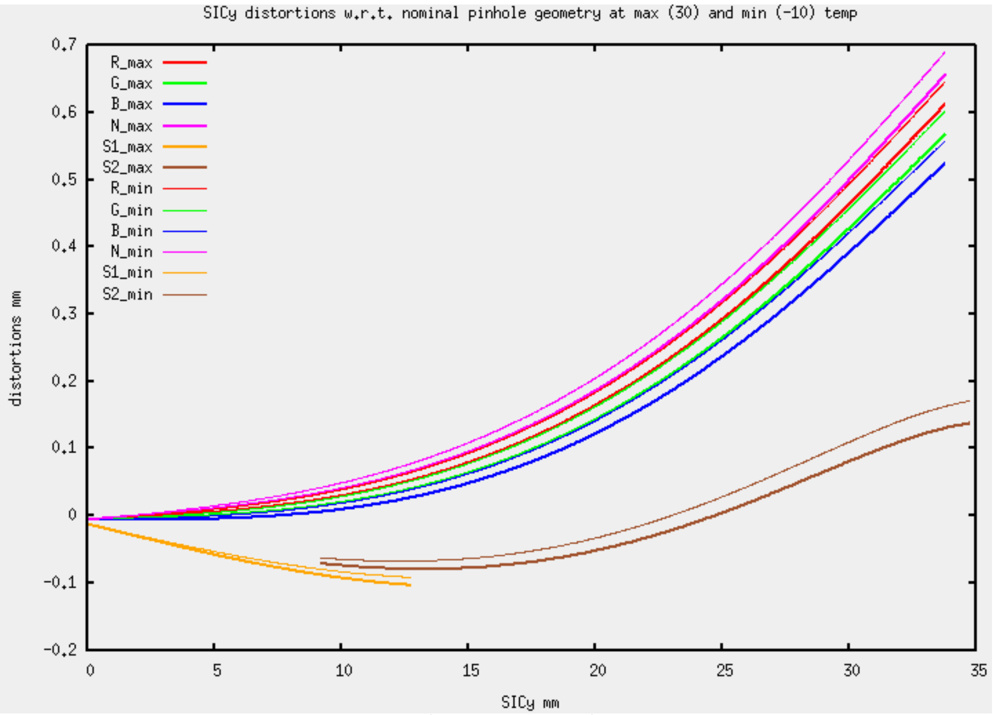


Figure 5.1-4: SIC y distortions at max (30 ^{0}C)/min (-10 ^{0}C) temperature with respect to nominal pinhole geometry

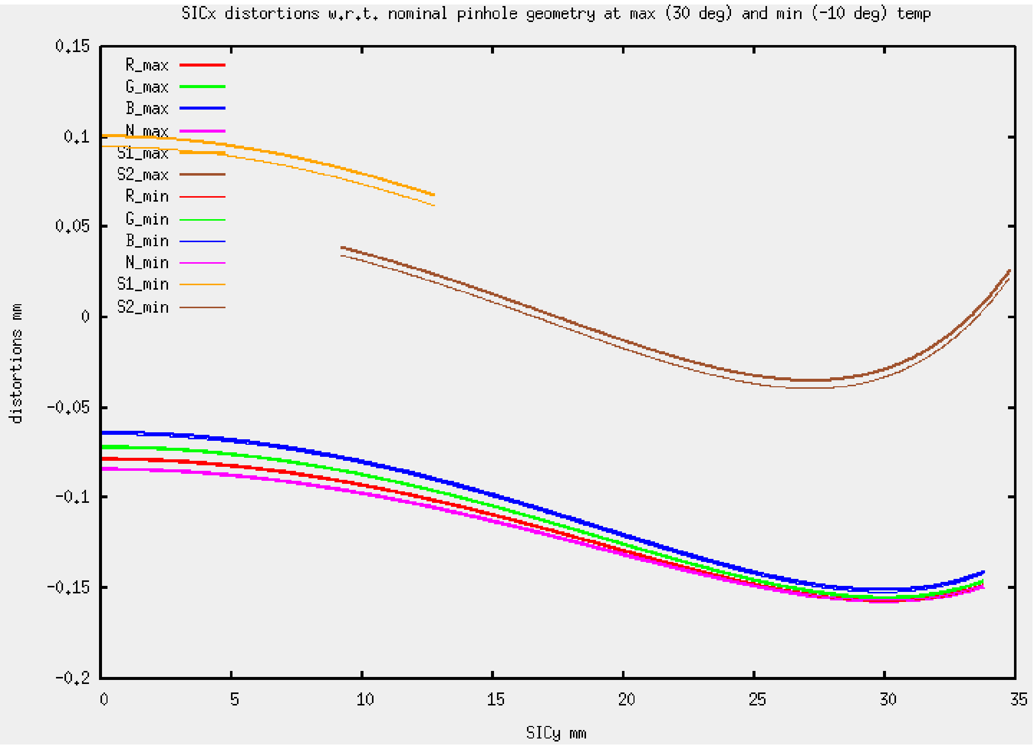
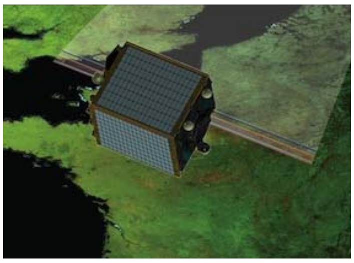


Figure 5.1-5: SIC x distortions at max (30 ^{0}C)/min (-10 ^{0}C) temperature with respect to nominal pinhole geometry



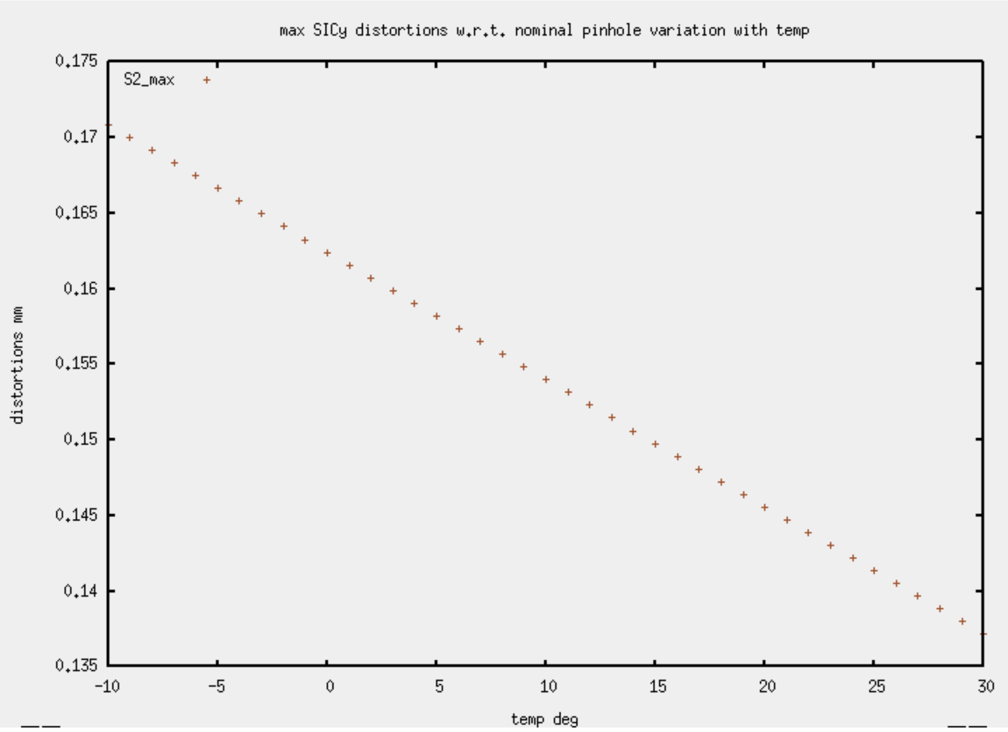


Figure 5.1-6: maximum SIC y distortions with respect to nominal pinhole geometry vs. temperature

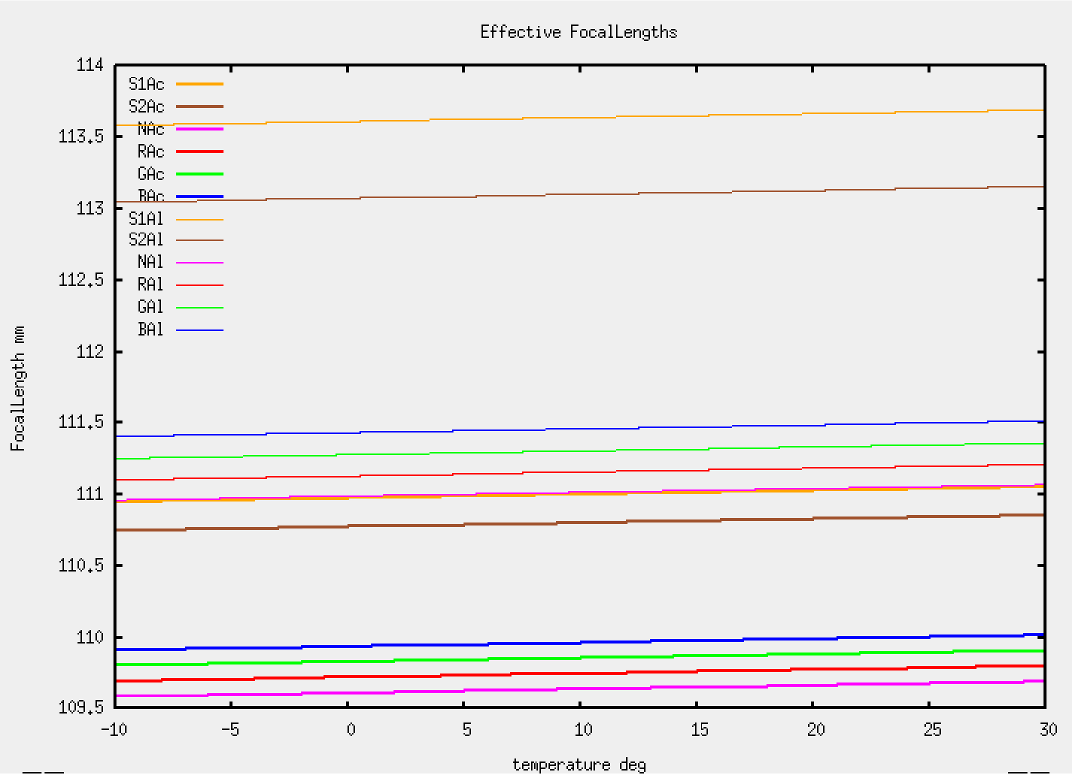
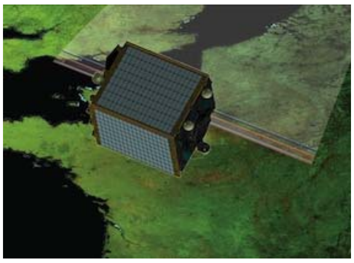


Figure 5.1-7: Effective focal length vs. temperature



During the project when available, on-ground calibration and measurement data together with the validated thermal elastic model of PV01 shall be used.

5.3 LOS model Transformation

The following table lists the basic coordinate systems used by the PROBA-V Geometric Calibration algorithms in the transformation from viewing vector to position viewed on earth. For more information on these frames, see [PF-DPM] Appendix 1.

Frame	**Axes**	**Description**
WGS	(lon, lat, h)	Geographic Frame (i.e. WGS84).
TER	(Tx, Ty, Tz)	Terrestrial Frame (i.e. ITRS/International Terrestrial Reference System)
CEL	(Ix, Iy, Iz)	Inertial Frame (i.e. ICRS/International Celestrial Reference System)
ORB	(Ox, Oy, Oz)	Earth-centred Orbital Frame

BOF	(Bx, By, Bz)	Spacecraft Body-fixed Frame
		Bx = parallel to –Px
		Bz = parallel to Pz

PL	(Px, Py, Pz)	Payload Frame
SIL	(SILx, SILy, SILz)	Spectral Imager Frame – Left spectral imager
SIC	(SICx, SICy, SICz)	Spectral Imager Frame – Center spectral imager

Table 5.3-1: Coordinate systems used by PROBA-V

After developing the methods for the processing of the measurements, the Line-of-Sight (LoS) model can be expressed as follows:

$$\mathbf{r}_{object}^{TER} = \mathbf{R}_{CEL}^{TER} \cdot \mathbf{r}_{sensor}^{CEL} + s \cdot \mathbf{R}_{CEL}^{TER} \cdot \mathbf{R}_{BOF}^{CEL} \cdot \mathbf{R}_{PL}^{BOF} \cdot \mathbf{R}_{SI}^{PL} \cdot \mathbf{r}_{object}^{SI} = \mathbf{R}_{CEL}^{TER} \cdot \mathbf{r}_{sensor}^{CEL} + s \cdot LoS^{TER}$$

where the observations of the attitude angles at time t are modelled as

$$\mathbf{R}_{BOF}^{CEL} = \mathbf{R}_{ORB}^{CEL} \cdot \mathbf{R}_{BOF}^{ORB}$$

the on-board software performs this conversion. The observations of the actual positions at time t of the projection centre of the sensor are modeled as

$$\mathbf{r}_{sensor}^{CEL} = \mathbf{r}_{NAV}^{CEL} - \mathbf{R}_{BOF}^{CEL} \cdot \mathbf{r}_{NAV}^{BOF} + \mathbf{R}_{BOF}^{CEL} \cdot \mathbf{r}_{sensor}^{BOF}$$

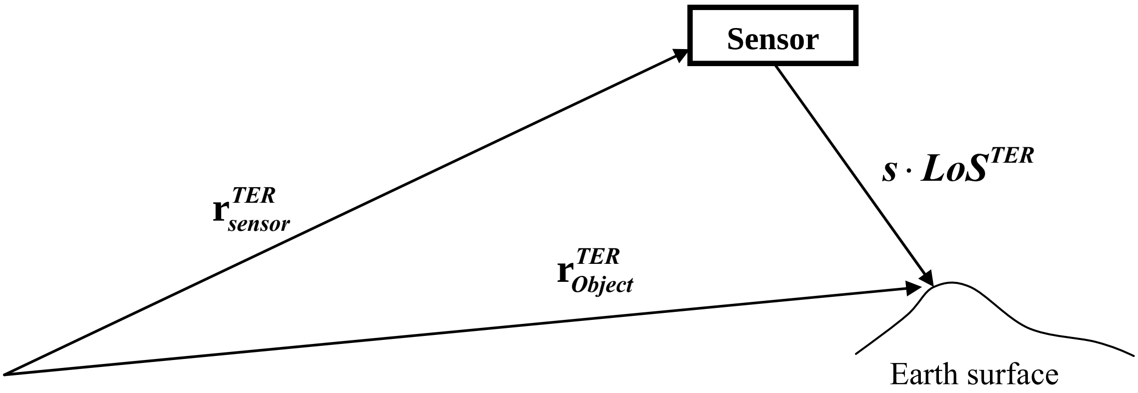


Figure 5.3-1: Concept of direct geo-referencing

However, the simplification $\mathbf{r}^{CEL}_{sensor} = \mathbf{r}^{CEL}_{NAV}$ is applied for PROBA-V, because for a low resolution satellite (more than 100 m) the contribution of the lever arm values $\mathbf{r}^{BOF}_{nav}$ from the body frame origin to the measured position and $\mathbf{r}^{BOF}_{sensor}$ from the body frame origin to the sensor projection centre are negligible.
In order to determine the object co-ordinates in longitude/latitude/height the intersection of the actual look direction of the sensor $\mathbf{p}^{TER}_{Object} = \mathbf{R}^{TER}_{SI} \cdot \mathbf{r}^{SI}_{Object}$ with origin $\mathbf{r}^{TER}_{Sensor}$, the position of the projection centre of the sensor expressed in the earth centred earth fixed co-ordinate frame (TER), with an ellipsoid can be written to (which is equivalent to determine the scale factor s):

$$\mathbf{r}^{TER}_{Object} = \mathbf{r}^{TER}_{Sensor} + s \cdot \mathbf{p}^{TER}_{Object}$$

On the other hand this intersection point has to fulfill the ellipsoid equation, where ***A*= *a* + *h*** and ***B* = *b* + *h*** with ***a*** the major axis, ***b*** the minor axis and ***h*** the constant height above the earth ellipsoid:

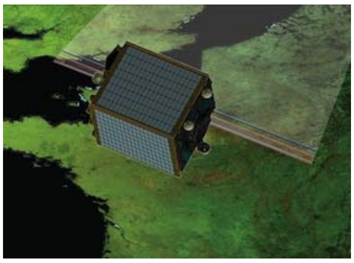
$$\frac{\left(r_x\right)^{TER\ 2}_{Object} + \left(r_y\right)^{TER\ 2}_{Object}}{A^2} + \frac{\left(r_z\right)^{TER\ 2}_{Object}}{B^2} = 1$$

The ellipsoid used is WGS84, the same used in the Landsat Geocover dataset used for GCPs chips extraction.

Substituting the vector components of the collinearity equation into the ellipsoid equation leads to the quadratic equation for the scale factor s.

$$\left(\frac{x^2 + y^2}{A^2} + \frac{z^2}{B^2}\right) \cdot s^2 + \left(\frac{x \cdot x' + y \cdot y'}{A^2} + \frac{z \cdot z'}{B^2}\right) \cdot s + \left(\frac{x'^2 + y'^2}{A^2} + \frac{z'^2}{B^2} - 1\right) = 0$$

with $\mathbf{r}^{TER}_{Sensor} = \left(x', y', z'\right)^T$ and $\mathbf{p}^{TER}_{Object} = \left(x, y, z\right)^T$.

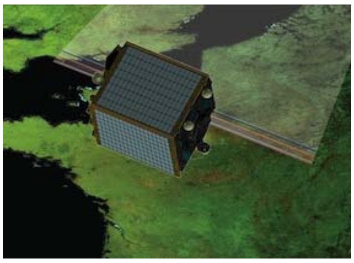


For the two real solutions for s of this equation the smaller one (the first intersection with the surface) is used to determine the intersection point of the sensor looks direction with a constant height above the ellipsoid surface. For a more detailed description see Section 5.4.

The different parameters of the LoS model are described below.

t	Mean time an image line is acquired.
$\mathbf{r}_{object}^{TER}$	Vector to the object point to be calculated given in the mapping co-ordinate frame (TER frame co-ordinates). This vector is finally expressed in longitude/latitude/height co-ordinates by intersection with an earth ellipsoid.
$\mathbf{r}_{sensor}^{CEL}$	Vector to the projection centre of the sensor expressed in the Inertial frame Earth centred (CEL). This vector is calculated from the measured satellite position (GPS/Galileo/DORIS) reduced by the lever arms, normally determined before launch. Often, as for PROBA-V, due to the small distances between projection centre of the imaging sensor and the navigation device of the spacecraft with respect to the ground resolution of the sensor no distinction is made between sensor projection centre and satellite position $\mathbf{r}_{NAV}^{CEL}$.
$\mathbf{r}_{object}^{SI}$	Vector from image point, which is measured by the sensor pixel location, to the corresponding object point on the earth surface given in the sensor co-ordinate frame (SI). The transformation from the sensor pixel location to the Cartesian co-ordinate frame of the sensor is described by the interior orientation, given by a sensor model or laboratory measurements and adjusted by in-flight geometric calibration.
$\mathbf{R}_{BOF}^{CEL}$	Overall rotation matrix from spacecraft body fixed frame (BOF) to from Earth centred inertial frame (CEL) is determined by using the quaternion vectors sampled at 4 Hz. See [PF DPM].
$\mathbf{R}_{PL}^{BOF}$	The rotation matrix from the Payload frame (PL) to Spacecraft body fixed frame (BOF) is a fixed rotation which essentially perform an reference axis transformation. See [PF DPM].
$\mathbf{R}_{SI}^{PL}$	Rotation matrix from the sensor frame (SI) to payload co-ordinate frame (PL), which accounts for boresight misalignment angles and is therefore also part of the interior orientation (sensor model). The boresight misalignment angles also describe the thermal influences during orbit revolution (varying solar exposure of the satellite) and also the vibration influences during image acquisition.
s	Scale factor determined by the object point. In the case of the LoS model the scale factor is determined by the intersection with an earth ellipsoid. For ortho rectification the scale factor is determined by the intersection with a DEM (digital elevation model).
$\mathbf{R}_{CEL}^{TER}$	Rotation matrix from the inertial frame Earth centred co-ordinate frame (CEL) to the earth centred earth fixed (TER) frame. This transformation takes into account: the earth rotation the constantly changing orientation of the earth‘s polar axis (i.e. precession, nutation and polar motion) The polar motion matrix depends on IERS tabulations of the pole coordinates Xp and Yp (see http://www.iers.org, Bulletin B).

The direct Georeferencing is done by using SRTM DEM. More specifically, the 3-arc second pixel spacing (corresponds to ~ 90m) CGIAR SRTM V3 Digital Elevation Model (DEM) is used.



It covers ~ 80% of land from the latitude 60° North to 56° South (see [SDD] for more details).

The accuracy of the DEM is high:

- The absolute horizontal accuracy (90% Circular Error) is 20 meters.
- The absolute vertical accuracy (90% Linear Error) is 16 meters.

5.4 Direct Location at a given altitude (DL)

5.4.1 Introduction

This algorithm finds the geocentric location on the ellipsoid of the intersection of the LoS vector to the ellipsoid at a constant elevation above the ellipsoid.
The indirect algorithm uses an iterative method to locate the intersection. The satellite's geocentric latitude is used as the initial guess in the process. This latitude is used to find the geocentric radius of the ellipsoid plus the elevation at this latitude. Using the angle between the satellite's position vector and the LoS vector, the radius of the Earth, and the plane trigonometry's law of cosines, the magnitude of the LoS vector is found. The intersection is the addition of the satellite's position vector and the scaled LoS vector. However, this location is based on slightly incorrect latitude. Thus, the calculated intersection’s geocentric latitude is used as the second guess. This process is continued until the LoS magnitude difference between iterations is small (< 0.1 meter).

Another approach is to use the reduced latitude in the calculation. This allows for a direct solution (direct location) for a point on the surface of the ellipsoid. This is the method that shall be used.

5.4.2 Input/Output

Input :

- satellite's position vector;
- Attitude quaternion;
- Viewing model (Boresight angle + Interior Orientation parameters)
- (line; pixel) in the detector geometry;
- the elevation h

Output:

- geocentric latitude and longitude of the intersection
- geocentric location of the intersection in TER Cartesian coordinates (x, y, z)
- geodetic latitude of the intersection

5.4.3 Algorithm

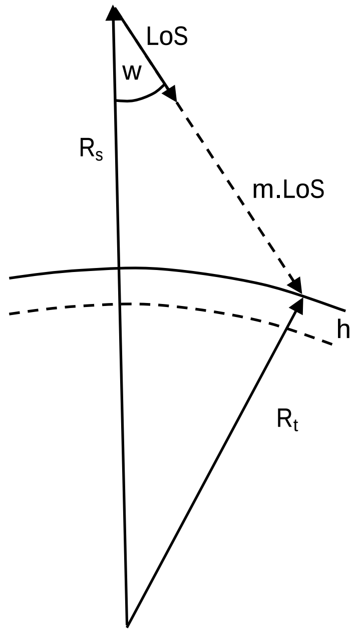


Figure 5.4-1: Direct Location at a given altitude

Define Rs as the position vector of the satellite, LoS as the line-of-sight vector, and Rt as the position vector of the target in question. These vectors are defined in the TER reference system. The position vector of the target using the reduced latitude psi, longitude Lt, and ***a = semi-major axes + h*** and ***b = semi-minor axis + h*** is:

$$R_t = (X_t , Y_t , Z_t)$$

$$X_t = a * \cos (psi) * \cos(Lt)$$
$$Y_t = a * \cos (psi) * \sin(Lt)$$
$$Z_t = b * \sin (psi)$$

The solution is computed as indicated in the following steps.

1. Compute the ***LoS*** vector starting from the ***(line, pixel)*** according to what specified in the “LOS model” and “LOS model transformation” sections. Compute the ***LoS*** unit vector.

2. The solution requires the following equation to be true:

$$R_t = R_s + m * LoS$$

where ***m*** is the magnitude of the final ***LoS*** vector. The current ***LoS*** vector has unit magnitude.

As separate components:

$$X_t = a * \cos(psi) * \cos(Lt) = X_s + m * X_{los}$$
$$Y_t = a * \cos(psi) * \sin(Lt) = Y_s + m * Y_{los}$$
$$Z_t = b * \sin(psi) = Z_s + m * Z_{los}$$

A vector **R**$\mathbf{R_t}'$ with a magnitude of 1 can be constructed by dividing the $\mathbf{X_t}$ and $\mathbf{Y_t}$ components by **a** and dividing the $\mathbf{Z_t}$ component by **b**:

$$R_t' = (X'_t , Y'_t , Z'_t)$$

$$X_t' = X_t / a = \cos(psi) * \cos(Lt) = X_s / a + m * X_{los} / a$$
$$Y_t' = Y_t / a = \cos(psi) * \sin(Lt) = Y_s / a + m * Y_{los} / a$$
$$Z_t' = Z_t / b = \sin(psi) = Z_s / b + m * Z_{los} / b$$

or:

$$R_t' = R_s' + m * LoS'$$

3. Using the law of cosines for a plane:

$$|R_t'|^2 = m^2 * |LoS'|^2 + |R_s'|^2 – 2 * m * |LoS'| * |R_s'| * \cos(w) = 1$$

where ***w*** is the angle between the satellite's position vector and the line-of-sight vector. Since R_t' is defined to have a magnitude of 1, the only unknown left in the equation is ***m***.

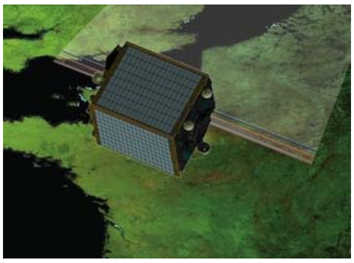
Re-arranging the equation to use the quadratic formula to solve for m results in:

$$0 = m^2 * |LoS'|^2 – 2 * m * |LoS'| * |Rs'| + |Rs'|^2 – 1$$

and thus, the solution for R_t is:

$$R_t = R_s + m * LoS$$

4. The geocentric location is then converted to a geodetic location (latitude and longitude).



5.5 Direct Location On DEM (DLD)

5.5.1 Introduction

The direct location on DEM algorithm finds the location of the point on the DEM “viewed” by a given line-of-sight (LoS) vector. The goal of this algorithm is to compute the altitude to which the line-of- sight intersects the DEM. This is done using an iterative process.

5.5.2 Input/Output

Input:

- satellite's position vector;
- Attitude quaternion;
- Viewing model (Boresight angle + Interior Orientation parameters)
- (line; pixel) in the detector geometry;
- DEM.

Output:

- geocentric location of the intersection (lat, lon, h);
- geocentric location of the intersection in TER Cartesian coordinates (x, y, z);
- geodetic latitude of the intersection.

5.5.3 Algorithm

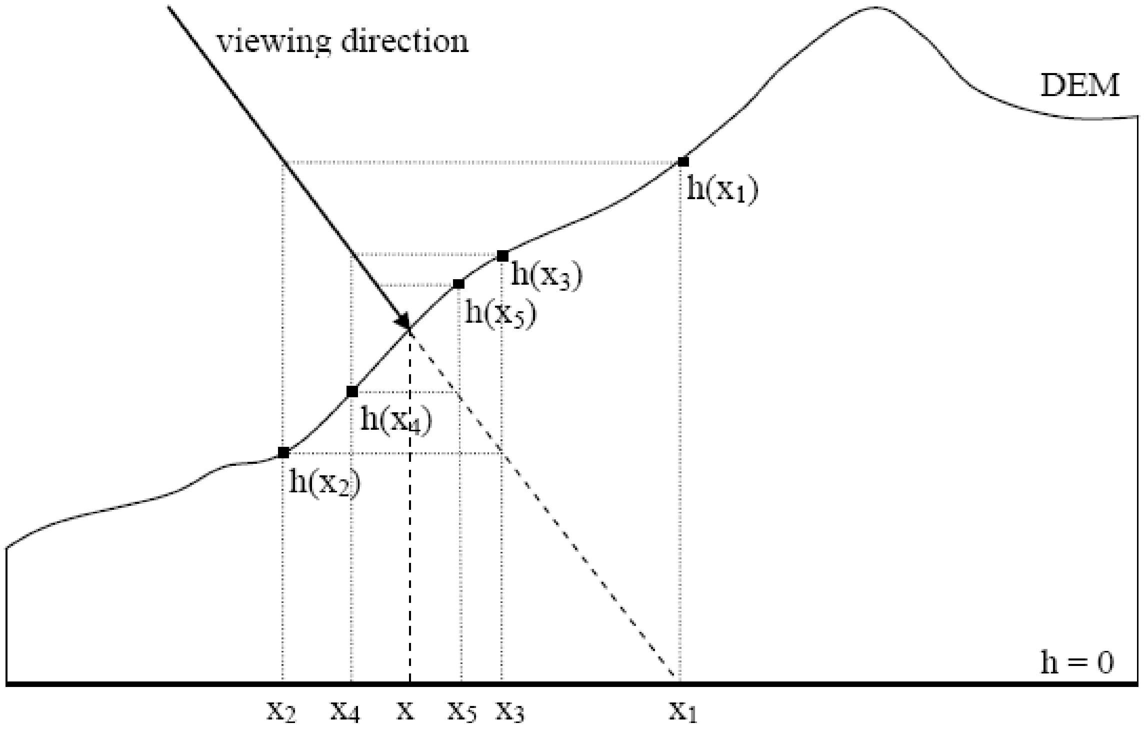
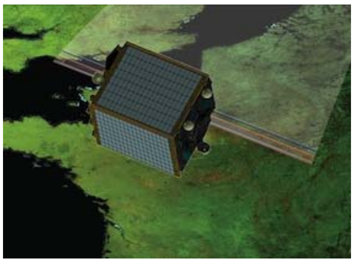


Figure 5.5-1: Direct Location on DEM – iterative procedure

Due to the use of geodetic lon/lat coordinates, a consequence of the Earth being an ellipsoid, the lon/lat coordinates of the point viewed depend on the topographic relief of the area being viewed. In Figure 5.5-1, the correct position of the point viewed is x, but when the topographic relief is neglected (i.e. when h=0 is assumed everywhere), one would position the point viewed at x_1.

The direct location on DEM procedure calculates, in an iterative way, the correct coordinates of the point viewed using elevation information retrieved from a DEM:

```
i=0;
h=0;
while(true)
    i=i+1;
    $(x,y,z)_i$ = intersection of LoS with earth via Direct Location with height h;
    if(i > 0 and distance($(x,y,z)_i$, $(x,y,z)_{i-1}$) < $\delta$) break;
    convert $(x,y,z)_i$ => $(lat, lon)_I$;
    h = DEM(lat,lon);
endwhile
```



5.6 Reverse Location (RL)

5.6.1 Introduction

The Reverse Location is a function which, for a viewing model, identifies the image coordinates in the detector geometry of this model that “views” a ground position, at any given altitude.

The proposed method is based on precise collinearity equations as time depended and accurate root finding with respect to time.

In the previous version of the ATBD document, an alternative approach was proposed: the Reverse Location is carried out iterating the direct location (using a gradient descent) in order to obtain the proper image coordinates which “direct locates” the ground coordinates. This alternative approach is more complex and requires the usage of the Predictor (approximated position in the detector geometry).

5.6.2 Input/Output

Input:

- satellite's SV data at different times;
- Attitude quaternion data at different times;
- Los model
- geocentric location (lat, lon, h);

Output:

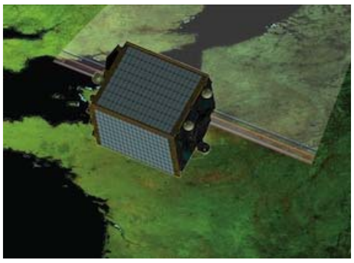
- (line; pixel) in the detector geometry;

5.6.3 Algorithm

Defining $\mathbf{r}_{Nav}^{TER}(t)$ as the position vector of the satellite, $LoS^{TER}(t)$ as the line-of-sight vector, and $\mathbf{r}_{object}^{TER}$ as the position vector of the target in question in the TER reference system, the objective of the Reverse Location is to find the LoS vector and the acquisition time t that the following equation is valid:

$$\mathbf{r}_{object}^{TER} = \mathbf{r}_{Nav}^{TER}(t) + s \cdot LoS^{TER}(t) \qquad (1)$$

where ***s*** scale factor determined by the object point, and in particular, by the intersection with a DEM (digital elevation model).



The Interior Orientation parameters (principal point offsets, polynomial CCD distortions) for each spectral band and camera are converted in pixel viewing directions by using the method outlined in Section 9.1.

In the Reverse Location the scale factor ***s*** is not determined, only the LoS unit vector is computed and by using the sensor viewing model the corresponding (line; pixel) in the detector geometry is determined as outlined below.

The LoS vector is transformed from the TER system to the SI reference system according to the following equations:

$$\mathbf{r}_{object}^{TER} = \mathbf{R}_{CEL}^{TER}(t)\cdot\mathbf{r}_{Nav}^{CEL}(t) + s\cdot LoS^{TER}(t) = \mathbf{R}_{CEL}^{TER}(t)\cdot\mathbf{r}_{Nav}^{CEL}(t) + s\cdot\mathbf{R}_{CEL}^{TER}(t)\cdot\mathbf{R}_{BOF}^{CEL}(t)\cdot\mathbf{R}_{PL}^{BOF}\cdot\mathbf{R}_{SI}^{PL}\cdot LoS^{SI}$$

$\mathbf{R}_{\boldsymbol{BOF}}^{\boldsymbol{CEL}}$	Overall rotation matrix from spacecraft body fixed frame (BOF) to from Earth centred inertial frame (CEL) is determined by using the quaternion vectors sampled at 4 Hz. See [PF DPM].
$\mathbf{R}_{\boldsymbol{PL}}^{\boldsymbol{BOF}}$	The rotation matrix from the Payload frame (PL) to Spacecraft body fixed frame (BOF) is a fixed rotation which essentially perform an reference axis transformation. See [PF DPM].
$\mathbf{R}_{\boldsymbol{SI}}^{\boldsymbol{PL}}$	Rotation matrix from the sensor frame (SI) to payload co-ordinate frame (PL), which accounts for boresight misalignment angles and is therefore also part of the interior orientation (sensor model). The boresight angle is extracted from the ICP file according to the temperature and time starting from eclipse parameters.

Therefore,

$$s\cdot LoS^{SI} = \left(\mathbf{R}_{CEL}^{TER}(t)\cdot\mathbf{R}_{BOF}^{CEL}(t)\cdot\mathbf{R}_{PL}^{BOF}\cdot\mathbf{R}_{SI}^{PL}\right)^{-1}\cdot\left(\mathbf{r}_{object}^{TER} - \mathbf{R}_{CEL}^{TER}(t)\cdot\mathbf{r}_{Nav}^{CEL}(t)\right) \qquad (2)$$

First step:

- $\mathbf{r}_{object}^{TER}$ is computed by converting the geocentric location (lat, lon, h);
- according to the temperature and time starting from eclipse parameters compute the viewing direction unit vectors (v_x^j, v_y^j, v_z^j) (j pixel index);
- transform the viewing direction unit vectors in across track angle θ^j_{across} and along track angle θ^j_{along}:

$$\theta^j_{along} = \mathrm{atan}(v_y^j / v_z^j)$$

$$\theta^j_{across} = \mathrm{atan}(v_x^j / v_z^j)$$

Second step:

- for a set of times t_i within the scene compute $s\cdot LoS^{SI}(t_i)$ by using equation (2);

- compute the normalized unit vector $LoS^{SI}(t_i)$ and compute the across track angle $\varphi_{across}(t_i)$ and along track angle $\varphi_{along}(t_i)$:

$$\varphi_{along}(t_i) = \text{atan}\left(LoS_y^{SI}(t_i) \,/\, LoS_z^{SI}(t_i) \right)$$

$$\varphi_{across}(t_i) = \text{atan}\left(LoS_x^{SI}(t_i) \,/\, LoS_z^{SI}(t_i) \right)$$

- for each t_i determine the index j such as $\theta^{j}_{across} <= \varphi_{across}(t_i) <= \theta^{j'}_{across}$ and by linear interpolation the associated pixel index (floating) $k(t_i)$ is computed:

$$k(t_i) = (\varphi_{across}(t_i) - \theta^{j}_{across}) / (\theta^{j'}_{across} - \theta^{j}_{across}) * (j' - j) + j; \qquad (3)$$

- for each t_i the ICP along track angle $\theta^{k(ti)}_{along}$ is computed by linear interpolation:

$$\theta^{k(ti)}_{along} = (k(t_i) - j) / (j' - j) * (\theta^{j'}_{along} - \theta^{j}_{along}) + \theta^{j}_{along}; \qquad (4)$$

- compute the time t_i for which $\theta^{k(ti)}_{along} - \varphi_{along}(t_i)$ has opposite signs, more precisely:

$$(\theta^{k(ti)}_{along} - \varphi_{along}(t_i)) * (\theta^{k(ti+1)}_{along} - \varphi_{along}(t_{i+1})) <= 0$$

Third step:

Continue as specified by the Bisection method by using the function $f(t) = \theta^{k(t)}_{along} - \varphi_{along}(t)$.

At each step the method divides the time interval in two by computing the midpoint of the time interval $t_c = (t_a + t_b)/2$ and the value of the function $f(t_c) = \theta^{k(tc)}_{along} - \varphi_{along}(t_c)$ at that point.

Unless t_c is itself a root (which is very unlikely, but possible) there are now two possibilities: either $f(t_a)$ and $f(t_b)$ have opposite signs and bracket a root, or $f(c)$ and $f(b)$ have opposite signs and bracket a root.

The method selects the subinterval that is a bracket as a new interval to be used in the next step. In this way the interval that contains a zero of f is reduced in width by 50% at each step. The process is continued until the interval is sufficiently small.

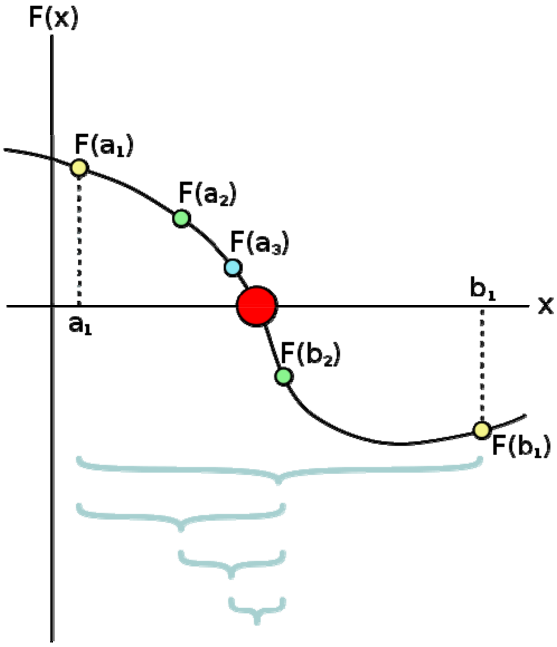


Figure 5.6-1: A few steps of the bisection method applied over the starting range [a_1;b_1]. The bigger red dot is the root of the function.

Given the new midpoint of the time interval t_c, the function $f(t_c) = \theta^{k(tc)}_{along} - \varphi_{along}(t_c)$ is computed according to the following steps:

- Compute the normalized unit vector $LoS^{SI}(t_c)$ by using equation (2) and compute the across track angle $\varphi_{across}(t_c)$ and along track angle $\varphi_{along}(t_c)$:

  $\varphi_{along}(t_c) = \text{atan}\left(LoS^{SI}_y(t_c) \ / LoS^{SI}_z(t_c) \right)$

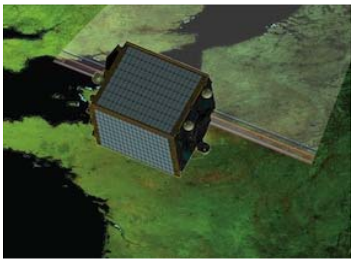
  $\varphi_{across}(t_c) = \text{atan}\left(LoS^{SI}_x(t_c) \ / LoS^{SI}_z(t_c) \right)$

- determine the index j such as $\theta^{j}_{across} <= \varphi_{across}(t_c) <= \theta^{j'}_{across}$ and by linear interpolation the associated pixel index (floating) $k(t_c)$ by using equation (3);
- determine the ICP along track angle $\theta^{k(tc)}_{along}$ by linear interpolation using equation (4).

In order to accelerate the convergence of the root-finding method, the usage of the **false position method** or **regula falsi method** shall be analyzed.

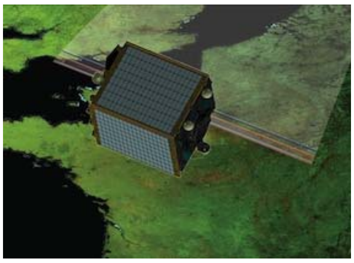
Like the bisection method, the false position method starts with two points t_a and t_b such that $f(t_a)$ and $f(t_b)$ are of opposite signs, which implies by the intermediate value theorem that the function f has a root in the interval [t_a, t_b], assuming continuity of the function f.

The method proceeds by producing a sequence of shrinking intervals [t_{ak}, t_{bk}] that all contain a root of f. At iteration number k, the number



is computed. t_{ck} is the root of the secant line through (t_{ak} , $f(t_{ak})$) and (t_{bk} , $f(t_{bk})$). If $f(t_{ak})$ and $f(t_{ck})$ have the same sign, then we set $t_{ak+1} = t_{ck}$ and $t_{bk+1} = t_{bk}$, otherwise we set $t_{ak+1}= t_{ak}$ and $t_{bk+1} = t_{ck}$. This process is repeated until the root is approximated sufficiently well.

The only difference between the false position method and the bisection method is that the latter uses $t_{ck} = (t_{ak} + t_{bk}) / 2$.



5.7 Direct Location grid cell polynomials (DLCP)

5.7.1 Introduction

Grid cell interpolation is used to accelerate the evaluation of a generic smooth function F with N input parameters and M output parameters which has to be evaluated many times with similar input values.
It does this by initially evaluating the function on an N dimensional grid and successively linearly interpolating this grid in successive calls. The evaluated grid must cover the whole extent of values to be successively interpolated and the spacing must be fine enough to limit interpolation errors as required.

This approach is used in order to compute the GCP chips resampling from on-ground map projected geometry to the detector geometry (Section 7.1). In order to take into account of the DEM, two different polynomial at low elevation height and at high elevation height are computed (see Section 7.1 for details on the usage of these two polynomials).

5.7.2 Input/Output

Input :

- Grid cell 4 corners (line; pixel) in the detector geometry;
- Low elevation height h_{low};
- High elevation height h_{high};
- GCP chip UTM zone and pixel spacing (ΔX, ΔY);
- GCP chip center pixel UTM coordinate (X_0,Y_0);

Output :

- Polynomial coefficients relative to h_{low};
- Polynomial coefficients relative to h_{high};

5.7.3 Algorithm

The **first step** consists in computing the (lat, lon) coordinates corresponding to the 4 corners (line; pixel) in the detector geometry both at low elevation height h_{low} and at high elevation height h_{high} by using the Direct Location at a given altitude (DL) function.

The **second step** is the transformation of the (lat,lon) coordinates in UTM coordinates (X,Y) by using the GCP UTM zone.

Then, the **third step** consists in expressing the (X,Y) coordinates in term of (line, pixel) of the GCP chip by using the GCP chip center pixel UTM coordinate (X_0,Y_0) and the pixel spacing:

$$pixel = (X - X_0)/\Delta X;$$

$$line = (Y - Y_0)/\Delta Y;$$

Finally, **fourth step**, the data used to produce the requested polynomials (relative to h_{low} and h_{high}) are taken from the grid input and output pixel locations. The first-order polynomial is used primarily for speed and computational reasons. The algorithm is as follows:

1. Read input and output lines/samples relative to h_{low}.

2. Use a least squares routine to calculate a first-order polynomial. This creates a "rough" polynomial:

$$\boldsymbol{S_i = a_0 + a_1 * S_o + a_2 * L_o + a_3 * S_o * L_o}$$
$$\boldsymbol{L_i = b_0 + b_1 * S_o + b_2 * L_o + b_3 * S_o * L_o}$$

where:

$\boldsymbol{S_i}$ is the input sample
$\boldsymbol{S_o}$ is the output sample
$\boldsymbol{L_i}$ is the input line
$\boldsymbol{L_o}$ is the output line
$\boldsymbol{a_0, a_1, a_2, a_3}$ are the polynomial mapping coefficients
$\boldsymbol{b_0, b_1, b_2, b_3}$ are the polynomial mapping coefficients

3. Repeat step 1 and 2 for $h_{high.}$

6. GCP CHIPS GENERATION

6.1 GeoCover data

The Landsat GeoCover 2000 dataset is a collection of precision orthorectified Landsat ETM+ scenes with spatial pixel resolutions of 15, 30, and 60.0 meters for the panchromatic, reflective, and thermal bands, respectively. These data sets are comprised of all nine Landsat ETM+ spectral bands and are in a UTM (Universal Transverse Mercator) map projection.

The geolocation accuracy of this product was pre specified to be below 50 meters RMSE) globally. Independent validation of the final product however, provides values between 19 (US) and 25 (North Eastern Africa).

The total disk space usage of the dataset is 2.5 TByte.

6.2 Selection of Band to Extract GCP Chips

The GCP chips are extracted from bands 1,3,4&5 of the GeoCover 2000 scenes as these have frequency ranges similar to blue, red, NIR and SWIR bands respectively. Each set of GCP chips is constructed in the same way from the respective GeoCover band. For simplicity the rest of this chapter describes the extraction of chips from band 4 but the same algorithm is applied to each band.

6.3 Chip Pixel Spacing

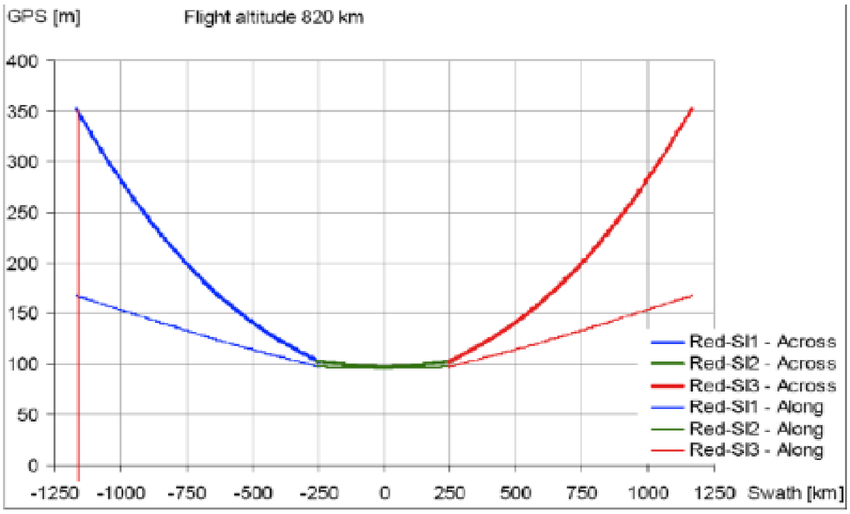


Figure 6.3-1 : VNIR along and across ground resolution for central and lateral cameras. The line sampling along track i.e. spacing between subsequent lines is about 100 m

The image chips will have to be cross-correlated to Proba-V VNIR bands images which as shown in the Figure 6.3-1 have ground pixel spacing of ~ 100m for the central camera and the near side of the lateral cameras.

Moving to the far side of the lateral cameras the across track ground pixel spacing increases up to ~ 350m. To cross-correlate, the image chips will have to be converted to the local pixel spacing of the Proba-V VNIR bands images, so they will have to be rotated and down-sampled from 30m to something ranging from 100m to 350m.

For SWIR the along track pixel spacing is 200 m (TBC) and the across track pixel spacing ranges from 200m to 700m (see figure 6.3-2).

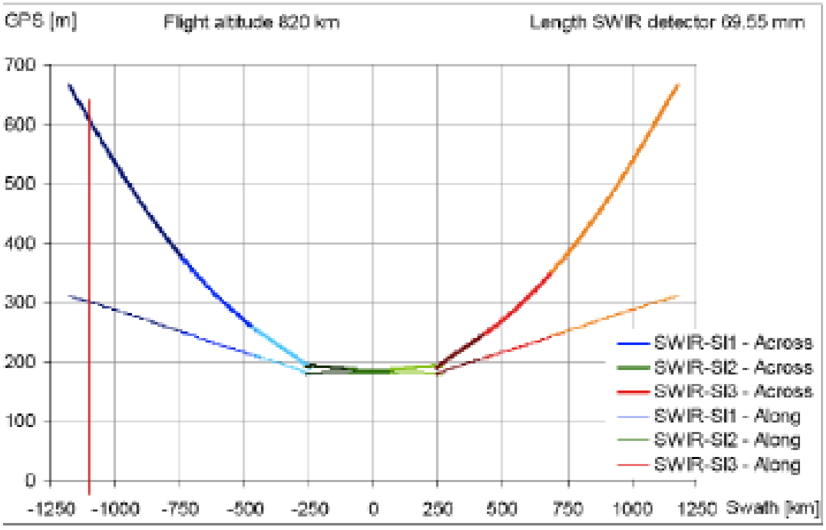


Figure 6.3-2 : SWIR along and across ground resolution for central and lateral cameras. The line sampling along track i.e. spacing between subsequent lines is about 200 m

To avoid having to keep the whole 2.5 TByte (compressed) Geocover dataset online and to have to extract and resample image chips from it, image chips are extracted and down-sampled to configurable pixel spacing and size. 100m is the minimum necessary pixel spacing so the 30m data can be down-sampled to this resolution with negligible loss. A stretched bi-cubic re-sampling approach (see Section 7.3) is used.

Experience shows that good results for correlation can be obtained with image chip sizes ranging from ~ 64x64 to 12x12 pixels.

At 350mx100m pixel spacing (VNIR case) 91x91 is reduced to 91x100/350 by 91 which gives at least 26x64 (only 64 lines are taken) good pixels at far range.

At 700mx200m pixel spacing (SWIR case) 91x91 is reduced to 91x100/700 by 91x100/200 which gives at least 13x45 good pixels at far range. For this reason 91x91 seems a reasonable chip size.

6.4 Chip Generation Summary

The Moravec Interest Operator algorithm is performed over an image to determine the interesting points using a specified window size suitable for satellite data.

A cloud mask is created using a simple cloud detection algorithm, so that the interest operator is performed only outside the masked regions, in areas identified as cloud-free.
The Moravec Operator algorithm is performed on the original image resampled to 3 different pixel spacings. Unwanted points are removed by comparing their computed interest operator values to a threshold value.

The interest operator results from all three resampled images are sorted and compared. Only those points/pixels identified in all three lists are retained. The other points are removed from the list. A constraint check is performed on the remaining points in the list (original resolution list) to ensure that the distance between any two points is greater than a threshold distance. The simple distribution logic of segmenting the image into zones and picking a minimum number of points from each zone ensures that the algorithm produces well-distributed points over the entire scene.

The Moravec Operator defines interest points as points where there is a large intensity variation in every direction. Directional variance is measured over small square windows. Sums of squares of differences of pixels adjacent in each of four directions (horizontal, vertical and two diagonals) over each window are calculated, and the window's interest measure is the minimum of these four sums. Features are chosen where the interest measure has local maxima. The feature is conceptually the point at the center of the window with this locally maximal value.

6.5 Chip Generation Detailed (Genchip algorithm)

The aim of the approach is mainly to satisfy the three basic requirements for the selection of points:

1) No points over water or clouds
2) Well distributed points
3) Well defined points

The methodology involved to create the interesting points using the Modified Moravec Interest Operator (MMIO) is outlined below in the form of a flow chart.

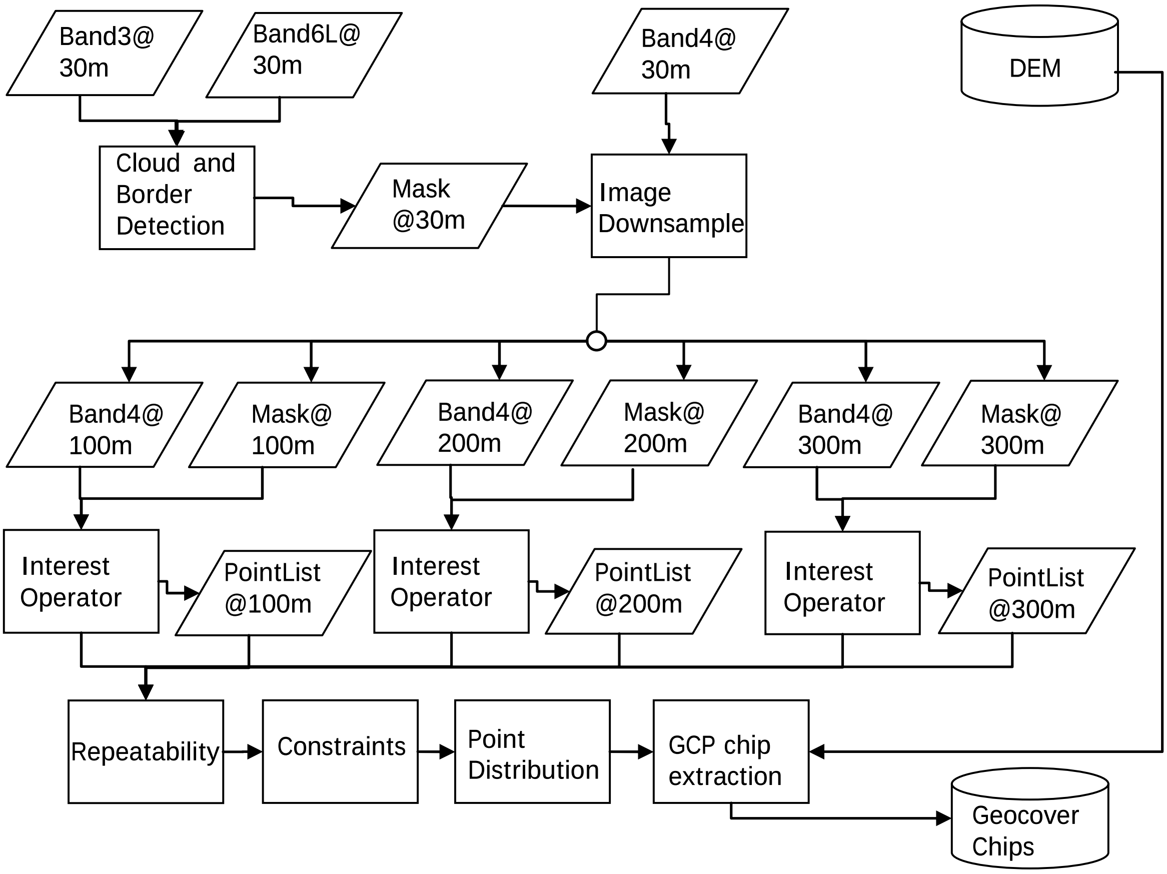


Figure 6.5-1 : Chips generation

The “Genchip algorithm” follows the following steps described below:

1. Cloud and Border detection
2. Image Resampling
3. Interest Operator
4. Repeatability
5. Constraints
6. Point distribution logic
7. GCP chip extraction

6.5.1 Cloud detection

The presence of clouds and shadows in the remote sensing data has always caused problems for different applications by obscuring the actual ground information. The different types of clouds and its different characteristics also made the detection of clouds very difficult. There are several algorithms to detect clouds, however, the MMIO (Modified Moravec Interest Operator) algorithm uses a simple approach for cloud detection. A simple, somewhat conservative approach to cloud detection is justified since the algorithm is designed to be applied to the GeoCover 2000 data set, which is nominally cloud free. For this application it is sufficient to simply avoid image regions that are, or may be, clouds. It is necessary for the MMIO algorithm to detect clouds, as one of the requirements is that the algorithm should not include clouds in the extracted control points. A cloud mask is created so that the algorithm avoids all cloud pixels and also some proximal region around the detected clouds.

The cloud detection algorithm used in this approach is based on ratio of two bands. The two bands that show predominant cloud characteristics (ETM+ bands) are Band 6 and Band 3. Band 6L (low gain version of band 6), which is a thermal band, usually has very low (cold) values for clouds whereas Band 3 often saturates for the clouds. Hence those pixels whose ratio is beyond the threshold will be marked as cloud pixels. The ratio is checked for each pixel in the image and a cloud mask is generated based on its value with ‘0’ for clouds and ‘1’ for non-cloud pixels.

Since the resolution of GeoCover band 6L is 60m whereas for band 3 it is 30m, the band 6L image is resampled to 30m before determining the ratio of the two bands.

The cloud detection logic is expressed as:

$$P(i, j, \text{mask}) = 0 \text{ if } P(i, j, B3) = 255 \text{ or } P(i, j, B3) >= \text{threshold}*P(i, j, B6)$$
$$P(i, j, \text{mask}) = 1 \text{ otherwise}$$

Where:

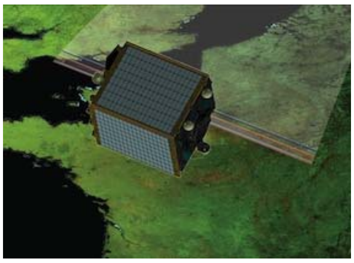
P(i, j, mask) is the pixel value for the cloud mask at line i and sample j
P(i, j, B3) is the pixel value for the band 3 at line i and sample j
P(i, j, B6) is the pixel value for the resampled band 6 at line i and sample j

The threshold was determined based on trial and error method. Several GeoCover scenes were tested with different threshold values and based on the test, the threshold is set at 2 when band 3 was operated in high gain. The dynamic range for band 3 low gain is 1.5 times that of high gain. Hence the threshold for low gain was determined to be 1.5 times lower than that of the high gain threshold. Thus, the high gain threshold was set at 2 and low gain threshold at 1.33. The ETM+ records in two thermal bands, one for low gain and another for high gain. For the MMIO cloud detection logic, always the low gain thermal band (6L) is used.

As this cloud detection logic is very simple and is prone to mis-classification, a buffer radius of 40 pixels around each cloud pixel is also classified as cloud pixels.

6.5.2 Border detection

The GeoCover band images are projected in the north up UTM projection system. Hence the actual image data will be seen rotated with respect to the image file extent. These extra pixels (also called fill region) are assigned the value ‘0’ to differentiate the fill region from the image region when the product is



created. The border mask is simply a mask that has value ‘0’ for fill regions and value ‘1’ for image regions.

Image multiplication operation is performed between the cloud mask and border mask to create a new mask that shows the valid pixels to run the Moravec interest operator. This greatly reduces the time to run the interest operator algorithm and at the same time satisfies one of the requirements for the selection of points (cloud avoidance).

6.5.3 Image Resampling

The MMIO algorithm requires an image-resampling algorithm to resample images to different resolutions. Two different resampling algorithms are used, one for resampling the image data and another for resampling the image mask data. The resolution for the GeoCover Band 5 image data is 60m. This image is downsampled to each of the 3 resolutions by using a stretched bi-cubic re-sampling approach (see Section 7.3). The cloud mask and the border mask at the original resolution of 60 m are also resampled to the 3 resolutions using nearest neighbor resampling algorithm.

These resampled images constitute the ‘input data’ to the Moravec Interest operator algorithm.

6.5.4 Moravec Interest Operator

The Moravec Interest Operator defines interest points as points where there is a large intensity variation in every direction. The original algorithm is slightly modified to suit to the needs for the current approach. The Modified Moravec Interest Operator algorithm can be summarized in the form of a flowchart as shown below.

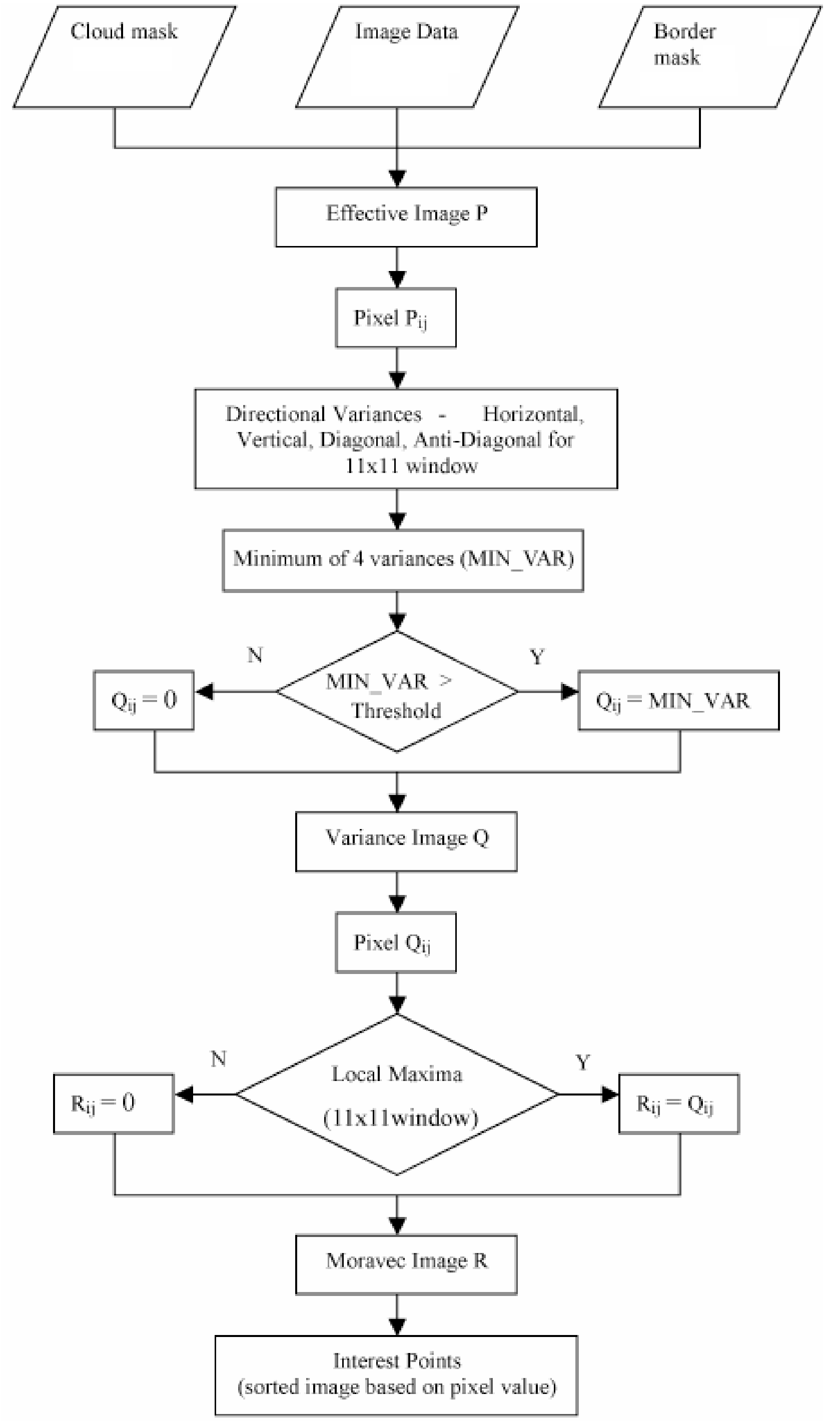

Cloud mask
Image Data
Border mask
Effective Image P
Pixel P_ij
Directional Variances - Horizontal, Vertical, Diagonal, Anti-Diagonal for 11x11 window
Minimum of 4 variances (MIN_VAR)
N
MIN_VAR > Threshold
Y
Q_ij = 0
Q_ij = MIN_VAR
Variance Image Q
Pixel Q_ij
N
Local Maxima (11x11window)
Y
R_ij = 0
R_ij = Q_ij
Moravec Image R
Interest Points (sorted image based on pixel value)

Figure 6.5 -2 : MMIO flowchart

As the flowchart shows, the input for this operator is the actual image data and the two masks (one for clouds and other for the border). By multiplying the two masks, we can get another mask that shows the active pixel for which the Moravec operator has to be performed. The Moravec operator starts with each active pixel (from mask) and then determines the directional variances for this pixel in the horizontal, vertical, diagonal, anti-diagonal directions for a window size of 11x11. (The entire 11x11 window is not required to be cloud-pixel free.) Considering the time, resolution of the data, and better results with the interest operator, the window size for the variance calculation was determined to be best at 11x11. The directional variance in the horizontal direction is computed by the following formula,

$$Q_H(i,j) = \sum_{k=i-5}^{k=i+5} \left(g(k,j) - g(i,j) \right)^2$$

where g(i,j) is the pixel value for the active pixel in the line i and sample j of the input image.

Similarly, directional variance along the vertical direction, diagonal direction and anti-diagonal directions are also determined for the active pixel Pij. The interest measure of the pixel is calculated based on whether the minimum of the four variances is above or below the threshold. The interest measure of a pixel is ‘0’ if the minimum variance is below the threshold and it is equal to the minimum variance otherwise. Interest measure is determined in the same way for all the active pixels and a floating-point type variance image is created. The interest points are selected where the interest measure has local maxima within a window size of 11x11. These interest points are then sorted based on their interest measure in the descending order to get the best point at the top of the list.

The threshold value for the Moravec Interest Operator is set at 10,000 to remove the homogenous regions like clouds, water or even less contrasting homogenous landmass that would fail to correlate. This threshold value is applicable only for the case of the 11 x 11 window size. For a 11 x 11 window, there are 10 values that are squared and summed while determining any directional variance. Hence by setting the threshold at 10,000 the average DN2 of each value is about 1000. This implies that the average difference between a pixel in the window to the center of the window is about 30 DN.

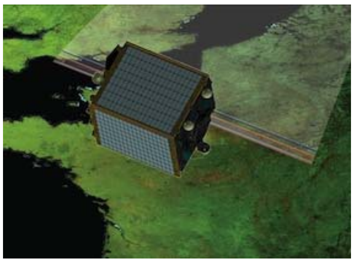
Several tests were conducted to check the validity of the threshold for the low contrast features like water bodies, desert, dry lands, etc in the GeoCover scenes. The test results showed that the threshold value is quite effective in removing the spike in homogenous regions and as well not using any homogenous regions for point selection.

The number of the GCPs extracted depend on the value of the Moravec operator threshold used.

6.5.5 Repeatability

The Moravec Interest Operator algorithm as described above can produce many points that may or may not be well defined due to the inherent problems with the algorithm. The MMIO algorithm uses the ‘repeatability function’ as a means to reduce such false identification of points. In general, if the ground targets are well defined at a specific resolution, they should be identifiable across different resolutions in the image until either the resolution is too low to make the entire target as sub-pixel or the resolution is too high that the targets are no longer considered well-defined.

Thus the definition of ‘interest point’ for MMIO is not limited to determining the points where the intensity changes in all directions, but also to detect such points at lower and higher resolutions. This



definition ensures that detected interest points are well defined based on ground characteristics and not purely based on image characteristics.

The resampled images with their masks are run through the Moravec Interest Operator algorithm at 3 different resolutions (100, 200 and 300m).

Those points that are found in all three point lists are considered as interest points and all the other points are discarded. The final point list consists of the common points with their coordinates in the 100m resolution and sorted (descending) based on interest measure (operator value) at the 100m resolution.

In the repeatability function, since the images are resampled, a tolerance of 2 pixels is allowed to match the same point across two other resolutions.

6.5.6 Constraints

The MMIO algorithm uses a constraint to make sure that no two GCP image chips shall contain the same image pixels. This reduces the risk of mis-registration when the interest points are very similar within a small distance.

The constraint equation is given below:

> if dist(Pi-Pj) < 91, remove Pj if j>i (i,j=1 to n) where, P refers to the point (interest point) on the point list (after repeatability) i, j are the index (after sort) for the interest points in point list (after repeatability) n, refers to the number of points in the final point list.

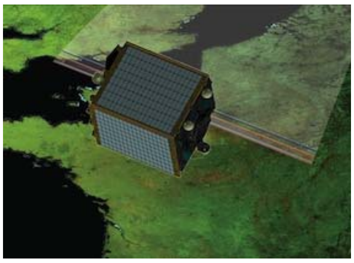
Thus, between any two points that are closer to each other by less than 64 pixels, the point that has lower interest measure are removed, while the other point is retained.

The point list after the constraint check consists of interest points that are not closer to each other by less than 64 pixels and sorted based on their interest measure (100m resolution).

6.5.7 Point distribution logic

One of the requirements for the MMIO is that the points are well distributed over the entire image. The distribution of the points is important, as the images will be registered using these points. The solution obtained from the registration process will be used to correct the satellite model while creating a precision corrected product. So if the distribution of points is not over the entire image but concentrated over a small region of the image, it might result in a poor fit.

The distribution logic in the MMIO algorithm maintains a balance between the distribution of points and well-defined interest points.



The distribution logic can be explained in steps as:

1. Extract ‘n’ number of points from the top of the point list. (high interest measure). These points will form the first ‘n’ number of points in the final point list for MMIO algorithm. If there are less than ‘n’ number of points in the point list, then all the points are selected for the final point list.

2. Divide the image into rectangular segments or zones.

3. Within each zone, a maximum of ‘k’ points can be included in the final point list. If ‘m’ number of points in a particular zone is already selected in step 1 (m<=n), then (k-m) number of additional points are selected from that particular zone such that the selected points have the highest interest measure among the remaining points within that zone. If (k-m) < 0 then no additional points are selected in this zone. This ensures that no more than ‘k’ points will be selected in this step and the points are not duplicated in the final list. If the maximum number of points available within a particular zone is less than ‘k’, then all the available points within that zone is selected (excluding points selected in step 1 from this zone) and if the zone contains no points, then no points will be selected from that zone.

4. The selected points in steps 1- 3 will be sorted based on their interest measure and forms the final point list from the MMIO algorithm.

5. If the total number of points are less than a threshold value (40) then the ‘genchips’ algorithm (default parameter of 20x20) will be run to increase the number of points as this increases the probability that a scene can be registered correctly. This is done only when the MMIO fails to generate enough points (ex: desert and homogenous regions throughout the scene). The ‘genchips’ determined point list is added with the interest operator point list.

Thus for a normal scene with mostly land, the points are well distributed over the entire image, whereas in cases with small islands, the well defined points within the islands alone will be selected, as the zone without any points (removed by threshold in Moravec interest operator algorithm) will not add any points in to the final list. Even for the case where there are not enough features, the coupling of genchips algorithm with MMIO algorithm can produce more points with better results.

6.5.8 GCP chip extraction

Once all the well defined points for an image are identified by their line and sample coordinates in the image, the next important step is to determine the location of these points with respect to the ground coordinate system and create an image chip of specific size. An image extent of 91x91 pixels is extracted from the GeoCover Band 1,3,4 & 5 100m pixel spacing image with the interest point at the center of the chip. Since the GeoCover image is precision terrain corrected product in the UTM projection system, the UTM coordinates for the given line and sample can be calculated from the file header information. In order to get the elevation of the point, UTM coordinates of the points are converted to the DEM projection coordinate system to derive the elevation information for the corresponding points.

The following figure shows example chips extracted from a band 5 (SWIR) image.

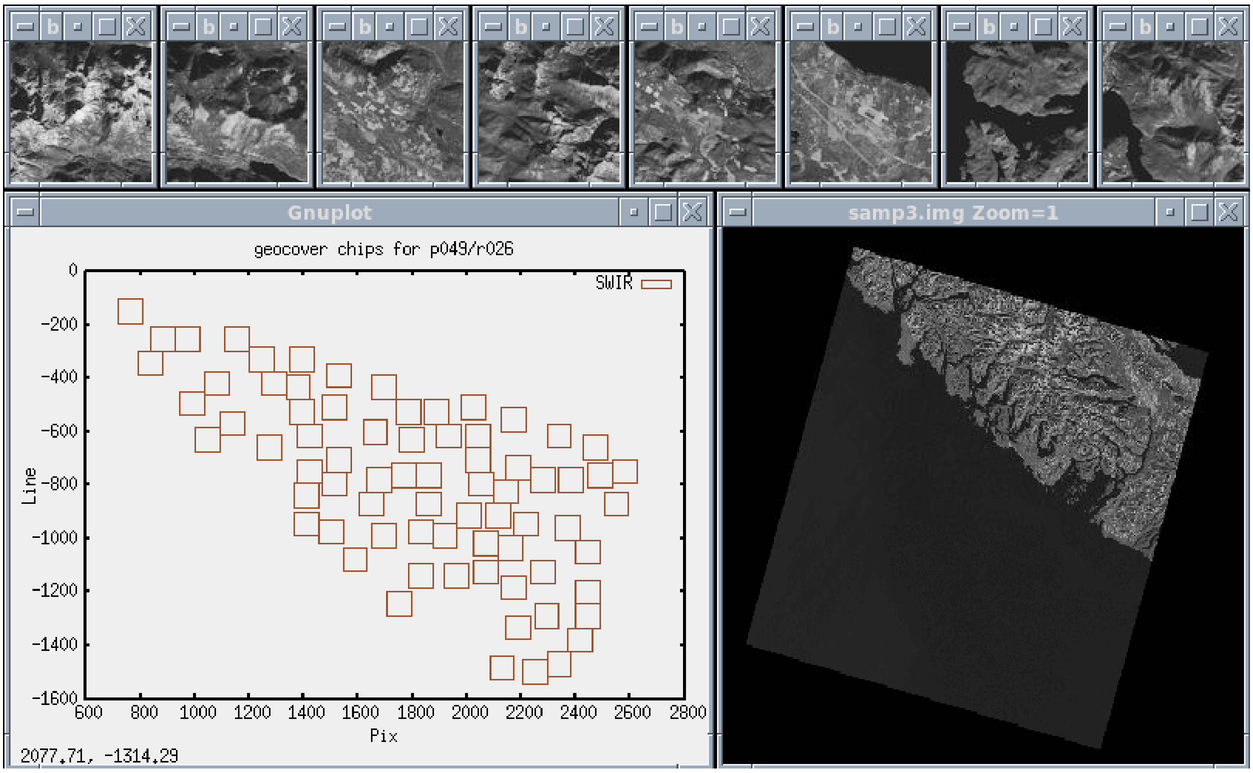


Figure 6.5 -3 : GCP chips generation

7. GCP CHIPS PROCESSING

7.1 GCP band (GcpBand)

7.1.1 Introduction

The objective of this section is the precise image location on the detector image (of a given strip) of a GCP defined by the image chip and coordinate of the centre of that image chip on the ground reference system (TER). The Chip Matching (CM) is performed by using Normalized Cross-correlation technique. The accuracy of the match depends mainly on the signal-to-noise ratio and on the image texture. Consequently, good image texture is one of the requirements on the image chips used for GCPs.

The identification of a single point using GCPs is independent from the identification of others, and is the same for all of the GCPs. The discussion here is limited o the measurement of a single point in the single camera.

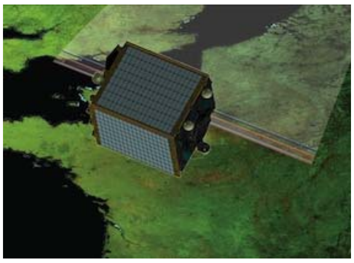
7.1.2 Input/Output

Input :

- Strip id;
- GCP chip (on ground geometry)
- GCP (lat, lon, h);
- Strip Viewing model (Boresight angle + Interior Orientation parameters)
- Satellite's SV data at different times;
- Attitude quaternion data at different times;
- DEM

Output:

- Observed GCP (line; pixel) in the strip detector geometry;
- Predicted GCP (line; pixel) in the strip detector geometry;
- Signal to noise ratio of the chip correlation;
- (lat0, lon0, h0) corresponding to the central GCP chip pixel in the strip detector geometry.



7.1.3 Algorithm

Given a GCP chip with the central pixel located at (lat, lon, h) the **first step** is to find the corresponding image coordinates (l,p) on the reference band detector geometry using the initial instrument viewing model. This is done using the Reverse Location on DEM (RLD) function. The image coordinate (l,p) represents the predicted GCP (line; pixel) in the strip detector geometry.

The **second step** consists in defining a N x M chip in the detector space around the image coordinates (l0, p0), which is the nearest integer detector of the (l, p) image coordinates:

l0 = int(l + 0.5); p0 = int(p + 0.5);

The N x M chip is the warped GCP chip in the strip detector geometry. The N x M chip is built by performing the Direct Location on DEM (DLD) for each pixel of the image space chip and performing a bi-cubic re-sampling over the input GCP chip.

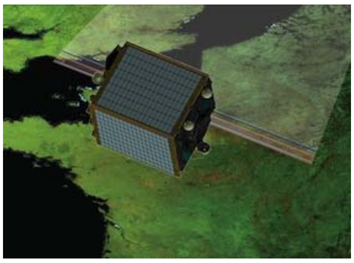
A stretched bi-cubic re-sampling approach (see Section 7.3) is used in order to take into account the local spatial resolution of the VNIR/SWIR reference band estimated at central chip coordinate (l0, p0).

It is worth reminding that:

- the pixels of the **VNIR** bands are observed at Nadir with 100m **across track** resolution, and off-nadir, the ground resolution increases from 100m to 360m; no across track binning is performed on-board;
- **along track** ground pixel size of the **VNIR** bands varies from about 100m in the center to about 170m at the edges of the swath; line sampling along- track, i.e. spacing between subsequent lines, is about 100 m over the FOV.
- the pixels of the **SWIR** bands are observed at Nadir with 200m **across track** resolution, and off-nadir, the ground resolution increases from 200m to 700 m; no across track binning is performed on-board;
- **along track** ground pixel size of the **SWIR** bands varies from about 200m in the center to about 320 m at the edges of the swath; line sampling along- track, i.e. spacing between subsequent lines, is about 200 m over the FOV.

However, the Direct Location on DEM function is quite time consuming, therefore an optimized approach based on DLCP method is proposed. In fact, the DLCP method is much faster than the rigorous DL method: DLCP requires iterative bilinear polynomial computations, the DLD requires iterative rigorous direct location computation. The error introduced is the polynomial approximation which is valid for small areas. In more details the proposed approach is defined by the following steps:

1. Compute the minimum and maximum elevation, h_{low} and h_{high} respectively, over the GCP chip by using the DEM;
2. Compute the mapping polynomials Direct Location grid cell polynomials (DLCP) of the N x M chip corresponding to h_{low} and h_{high};



3. Compute the direct location on DEM for each M x N chip element in an iterative way, the correct coordinates of the point viewed on the on-ground GCP chip "coordinate" using elevation information retrieved from a DEM. At each iteration the predicted on-ground position is the linear interpolation between the two polynomial based on the last value of h (DEM value at the last predicted on-ground position- previous iteration), the process is repeated until convergence:

```
i=0;
$(line,pixel)^{low}$ = intersection of LoS with earth via DLCP with height $h_{low}$;
$(line,pixel)^{high}$ = intersection of LoS with earth via DLCP with height $h_{high}$;
$(line,pixel)_0 = ((line,pixel)^{low} + (line,pixel)^{high})/2$;

while(true)
        $h = DEM(line,pixel)_i$;
        $i=i+1$;
        $(line,pixel)_i$ = linear interpolation of $(line,pixel)^{low}$ and $(line,pixel)^{high}$ based on h
        if($i > 0$ and distance($(line,pixel)_i$, $(line,pixel)_{i-1}$) $< \delta$) break;

endwhile
```

Please note that the initial elevation value is computed by considering the DEM elevation corresponding to the $(line,pixel)_0 = ((line,pixel)^{low} + (line,pixel)^{high})/2$, e.g. the average of the on-ground GCP chip coordinates corresponding to the h_{low} and h_{high}. However, in order to further optimize the number of iteration, the value of the DEM elevation corresponding to the nearest neighbor already computed (line, pixel) can be used as initial guess in the direct location on DEM algorithm outlined above. This strategy is particular effective when smooth variation of the DEM elevation is expected within the GCP chip.

The on ground coordinate (lat0, lon0, h0) corresponding to the N x M chip central pixel (l0, p0) is computed via DLD.

Finally, the **third step** is made up of the chip matching (via Normalized cross-correlation) in order to compute the observed GCP (line; pixel) in the strip detector geometry.

The GCPs are filtered according to the signal-to-noise ratio of the chip correlation. This step is necessary in order to eliminate not useful GCP (e.g. GCP chips over cloudy image). A further filtering is performed via iterative outlier detection described in Section 8.2.6 during the constrained inversion modeling.

7.2 Chip matching (CM)

7.2.1 Introduction

The chip matching is performed via Normalized cross-correlation. Normalized cross-correlation is used to produce a discrete correlation surface for an array of points between two bands. This correlation routine helps to remove any gray-scale differences between two images. This helps to alleviate some of the problems with the different radiometric responses between the GCP chips extracted from Geocover dataset (Landsat ETM+ sensor) and PROBA-V bands.

7.2.2 Input/Output

Input :

- Chip image (N x M pixels);
- Research window image (U x V pixels);

Output:

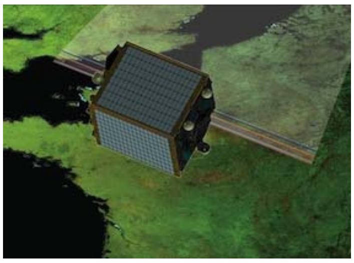
- Chip sub-pixel location (line; pixel) within the research window.
- Corresponding strength of the Normalized cross-correlation function.

7.2.3 Algorithm

The correlation routine uses the following formula when correlating between two image windowed subsets f and g, each with length and width of M and N, respectively:

$$\boldsymbol{R(x,y)} = \frac{\sum_{j=-N/2}^{N/2}\sum_{i=-M/2}^{M/2}\left[\left(f(j,i)-\bar{f}\right)\left(g(x+j,y+i)-\bar{g}\right)\right]}{\left[\left[\sum_{j=-N/2}^{N/2}\sum_{i=-M/2}^{M/2}\left[\left(f(j,i)-\bar{f}\right)^2\right]\right]\left[\sum_{j=-N/2}^{N/2}\sum_{i=-M/2}^{M/2}\left[\left(g(x+j,y+i)-\bar{g}\right)^2\right]\right]\right]^{\frac{1}{2}}}$$

where:



$$\bar{g} = \frac{1}{(M+1)(N+1)} \sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} g(x+j, y+i)$$

$$\bar{f} = \frac{1}{(M+1)(N+1)} \sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} f(j,i)$$

After the correlation surface is generated, the peak of that surface must be determined. This peak represents the geometric shift between two windowed data sets and is calculated to a subpixel location.
A sub-pixel location for the peak is found by using a polynomial fit method. This method works by fitting a second-order polynomial around the peak of the discrete correlation surface. The polynomial represents the surface associated with a 3 x 3 area around the peak. The discrete peak of the correlation surface is found first, then this peak and its corresponding eight neighbors are used for fitting the polynomial. A least squares fit can be used to calculate the polynomial coefficients after the 3 x 3 area and corresponding data points are found. From this polynomial, the location of the peak can be solved analytically to a sub-pixel location.

The polynomial equation has the following form:

$$P(x, y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

The coefficients of ***P(x, y)*** can be found by solving the matrix equation:

$$\boldsymbol{Y = X\,a}$$

where:

Y is a 9 x 1 matrix corresponding to the eight correlation points found around, and including, the peak

X is a 9 x 6 matrix with each row consisting of:

$$\boldsymbol{1 \qquad x_i \qquad y_i \qquad x_i\, y_i \qquad x^2_i \qquad y^2_i}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ are the ***x*** and ***y*** locations of the corresponding peak ***i*** in ***[Y]***

a is the polynomial coefficient

The solution of ***a*** is then:

$$\mathbf{a = (X^T\, X)^{-1}\, X^T\, Y}$$

where ^{T} refers to the transpose of a matrix, and $^{-1}$ refers to the inverse of a matrix.

After the coefficients are solved for, the sub-pixel peak can be found. The maximum correlation peak can be found by taking the partial derivative of ***P(x, y)*** with respect to ***x*** and ***y***:

$$\frac{\partial}{\partial x}P(x,y) = a_1 + a_3 y + 2a_4 x$$
$$\frac{\partial}{\partial y}P(x,y) = a_2 + a_3 x + 2a_5 y$$

By setting the partial derivatives to zero and solving for ***x*** and ***y***, the sub-pixel peak location can be determined. This leads to the following solution for the ***x*** and ***y*** offsets:

$$x\_offset = \frac{2a_1 a_5 - a_2 a_3}{a_3^2 - 4a_4 a_5}$$

$$y\_offset = \frac{2a_2 a_4 - a_1 a_3}{a_3^2 - 4a_4 a_5}$$

The strength of the Normalized cross-correlation function to be compared with the SNR correlation threshold is defined by 2 * (R$_{Peak}$ - R$_{mean}$)/R$_{std}$, where R$_{Peak}$ is the Normalized cross-correlation maximum value, R$_{mean}$ and R$_{std}$ are the Normalized cross-correlation mean and standard deviation values respectively.

7.3 Cubic convolution (CC)

7.3.1 Introduction

In general, the interpolation consists in calculating the value of the target point P(line,pixel) knowing the value of the neighbourhood pixels of its geometrically transformed intial image coordinates (line, pixel).

Given a fractional position in the initial image, we need to estimate the interpolated radiance from the different corners of the mesh around it in the initial geometry.

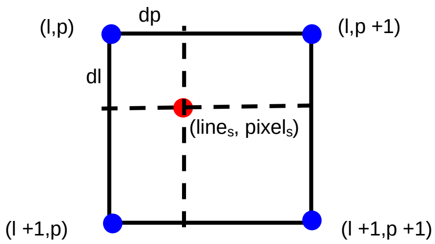


Figure 7.3-1: Target point "projected" in the initial mesh (line$_s$, pixel$_s$)

Considering a linear algorithm, the interpolated radiance of the pixel (line$_s$,pixel$_s$) can be expressed as a linear combination of the pixels Ims(l,p) evaluated in the initial geometry, weighted by the values of the function w.

$$\mathrm{Ims\_Interp}(line_s, pixel_s) = \sum_{(l,p)} \mathrm{Ims}(l,p) * \mathrm{W}(line_s - l, pixel_s - p)$$

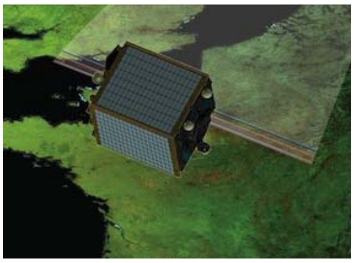
For clarity purposes, in the following formulas:

$dl = line_s\text{-}l$

$dp = pixel_s\text{-}p$

The only remaining freedom lies in the choice of the interpolation filter W.

Bicubic interpolation corresponds to calculating a weighted sum with the weights being given by a filter implementing the following kernel in both dimensions:



$$W(x)=\begin{cases}(a+2)\,|\,x\,|^{3}-(a+2)\,|\,x\,|^{2}+1 & for\ \ |\,x\,|\le 1\\ a\,|\,x\,|^{3}-5a\,|\,x\,|^{2}+8a\,|\,x\,|-4a & for\ \ |\,x\,|\le 1\\ 0 & otherwise\end{cases}$$

where a is usually set to -0.5 or -0.75. Note that W(0) = 1 and W(n) = 0 for all nonzero integers n. This approach was proposed by Keys who showed that a = − 0.5 (which corresponds to cubic Hermite spline) produces the best approximation of the original function. a = − 0.5 is the value used by the PROBA-V GC facility.

Due to the pixel size variation across track and along track, using the standard bicubic interpolation algorithm will no longer be valid. The standard bicubic filter using 4 neighboring grid points is considered a high quality resampling algorithm for target grids with the same or more dense spacing as the original grid. To use bicubic filtering for downsampling, the filter has to be stretched according to the down-sampling rate.

Considering two different specific down-sampling rates, one along track *down$_{along}$* and across track *down$_{across}$* the stretched bicubic interpolation corresponds to calculating a weighted sum:

$$\text{Ims\_Interp}(line_s, pixel_s)=\sum_{(l,p)}\text{Ims}(l,p)*\text{W}'(dl')*\text{W}'(dp')$$

with the weights being given by a filter implementing the following kernel with different stretched variables *dl’ = dl/ down$_{along}$* and *dp’ = dp/ down$_{across}$*:

$$W(x')=\begin{cases}(a+2)\,|\,x'|^{3}-(a+2)\,|\,x'|^{2}+1 & for\ \ |\,x'|\le 1\\ a\,|\,x'|^{3}-5a\,|\,x'|^{2}+8a\,|\,x'|-4a & for\ \ |\,x'|\le 1\\ 0 & otherwise\end{cases}$$

Where a=-0.5 and the final weights are *W’(dp’) = W(dp’)/ down$_{across}$* and *W’(dl’) = W(dl’)/ down$_{along}$*.

In order to accelerate the GCP chips resampling a bidimensional tabulated interpolation is used. The weights table W is calculated once with a very fine sampling step in the initial geometry (typically 1/32, 1/64 pixels). Then, to calculate the value of a target point, we choose the nearest mesh of W to perform the convolution. The geometric error induced by this method is linked to the sampling step of the tabulated function W: each target pixel will be calculated with a shifted position with a maximum value of sampling_step/2 in comparison to its exact position.

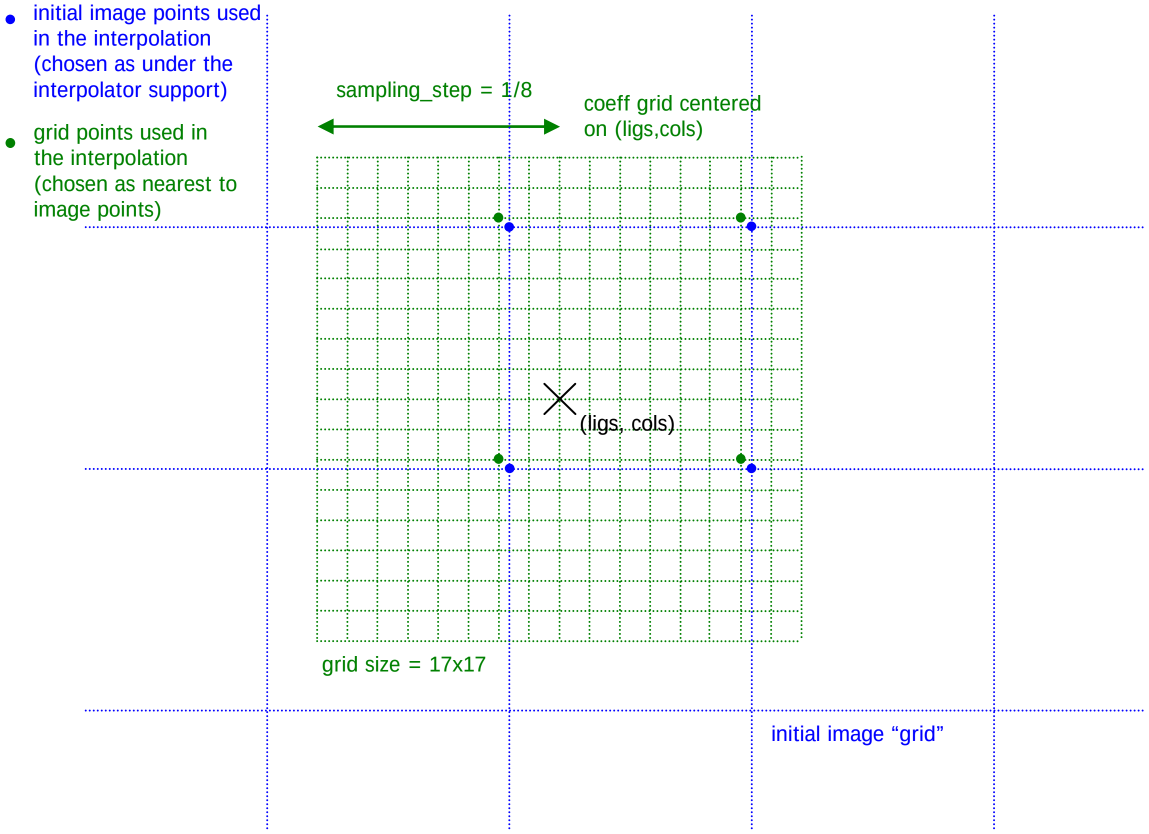

● initial image points used in the interpolation (chosen as under the interpolator support)
● grid points used in the interpolation (chosen as nearest to image points)
sampling_step = 1/8
coeff grid centered on (ligs,cols)
(ligs, cols)
grid size = 17x17
initial image "grid"

Figure 7.3-1: An example of application of a 17x17 grid on point (ligs, cols) with a sampling_step of 1/8

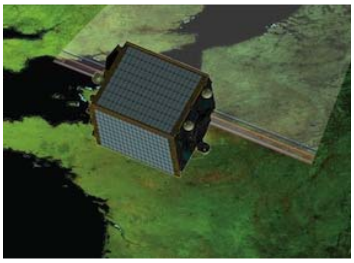
We note:

$$W_{height} = 5 + 4 * (\mathrm{down}_{\mathrm{along}} - 1) * sampling\_step$$

$$W_{width} = 5 + 4 * (\mathrm{down}_{\mathrm{across}} - 1) * sampling\_step$$

Definition of function "w" for computing bidimensional tabulated interpolation is:

$$W(dl, dp) = coeffs\left[\frac{round\left(dl + \frac{W_{height}}{2}\right)}{sampling\_step}, \frac{round\left(dp + \frac{W_{width}}{2}\right)}{sampling\_step}\right]$$



When: $\frac{-W_{height}}{2} \leq dl \leq \frac{W_{height}}{2}$ and $\frac{W_{width}}{2} \leq dp \leq \frac{W_{width}}{2}$ (i.e. inside coeffs bounds)

Otherwise (i.e. out of coeffs bounds): $W(dl, dp) = 0$

This defines the interpolator support as *floor(W_{height})+1 x floor(W_{width})+1* pixels.

Note: In practice, due to the “floating” positioning of the coeffs grid, the number of initial image pixels used in the interpolation may be lower, down to *floor(W_{height}) x floor(W_{width})* pixels.

7.3.2 Input/Output

Input :

- down_across; down-sampling factor across the track;
- down_along; down-sampling factor along the track;
- sampling step; the weights table W is calculated with a sampling step in the initial geometry (typically 1/32, 1/64 pixels);

Output:

- tabulated weights.

7.3.3 Algorithm

```
/* Determine spatial size of resampling kernal */

number_down_weights_across = 5 + 4 * ( down_across - 1 );

number_down_weights_along = 5 + 4 * ( down_along - 1 );

kernal_size = number_down_weights_across * number_down_weights_along;

delta_along = 1.0 / (double)down_along;

delta_across = 1.0 / (double)down_across;
```

```
 /* Determine resampling weights for subpixel offsets */
for( jj=0, counter=0 ; jj<=subpixels ; jj++ ){
        subdelta_line = (double)jj / subpixels;
        for( ii=0 ; ii<=subpixels ; ii++ ){
                subdelta_samp = (double)ii / subpixels;
                y = -2.0 - subdelta_line;
                for( ll=0 ; ll<number_down_weights_along ; ll++ ){
                        wy = cubic_convolution_weight(y,-0.5);
                        x = -2.0 - subdelta_samp;
                        for( ss=0 ; ss<number_down_weights_across ; ss++ ){
                                wx = cubic_convolution_weight(x,-0.5);
                                weights[counter] = wy * wx;
                                sum = sum + weights[counter];
                                counter++;
                                 x = x + delta_across;
                        }
                        y = y + delta_along;
                }
                for( ss=counter-kernal_size, sum=0.0 ; ss<counter ; ss++ )
                        sum += weights[ss];
                for( ss=counter-kernal_size ; ss<counter ; ss++ )
                        weights[ss] = weights[ss] / sum;
        }
}
```

8. CONSTRAINED INVERSION MODEL

8.1 Collinearity Equations

The following figure shows from the graphical point of view the concept of geometric calibration by inversion of the LoS Model.

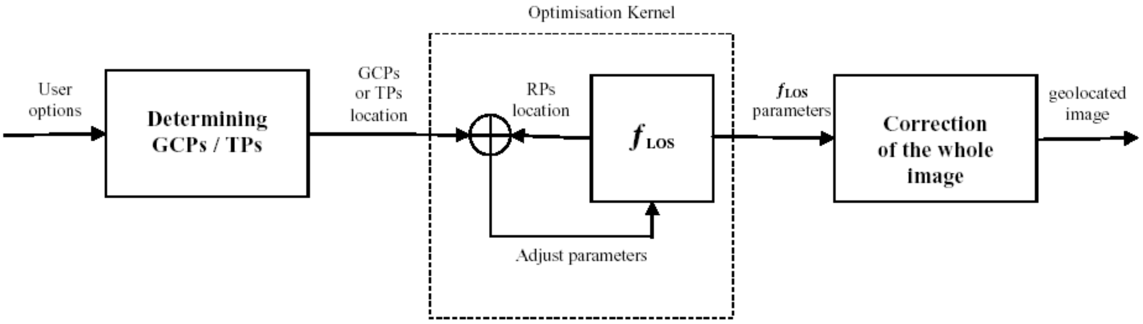


Figure 8.1-1: Geometric calibration by optimisation of LOS Model Parameters

In general, the geometric calibration foresees the calibration of the following parameters:

- Exterior orientation parameters (perspective centre, sensor attitude parameters and boresight misalignment angles which are the rotational offsets between sensor and payload co-ordinate frames)
- Interior orientation parameters (principal point offsets, CCD distortion)

For PROBA-V on-board attitude parameters and satellite position are sufficient precise therefore are not part of the calibration process.

	Category	Centre pixels on Central SI		Edge pixels on Side SI	
		(m) across track	(m) along track	(m) across track	(m) along track
Spacecraft					
Star map errors	R	2,3	2,3	8,5	4,1
Platform vibration error	R	0,4	0,4	1,5	0,7
Thermal deformation of ST bracket	P1	4,6	4,6	17,1	8,2
Attitude estimation error (from Kalman filter)	R	2,1	2,1	8,0	6,8
Attitude estimation error (from Kalman filter)	P	7,2	7,2	26,6	22,8
Orbital position estimation error (from Kalman filter)	R	3,8	3,8	5,6	3,8
Orbital position estimation error (from Kalman filter)	P1	0,0	0,0	0,0	0,0
On-board timing errors	R	0,0	0,7	0,0	0,7
Image datation errors	R	0,1	1,3	0,1	1,3
Image datation errors	B	0,0	0,0	0,0	0,0
Ageing	SV	8,0	8,0	29,6	14,2

Table 8.1-1: Spacecraft geolocation budget

The budget reported in the Table above is computed considering:

- Random values at 1 sigma value;
- Periodic value at peak value;
- Bias value at maximum value.

In contrast, the expected variations of the boresight misalignment angles at the satellite launch and during the orbit revolution (thermal influences due to varying solar exposure of the satellite) are large; therefore the determination of the boresight misalignment angles requires in-flight calibration procedures using a calibration field with well distributed ground control points. Rewriting the collinearity equation:

$$\mathbf{r}_{object}^{TER} = \mathbf{R}_{CEL}^{TER} \cdot \mathbf{r}_{sensor}^{CEL} + s \cdot \mathbf{R}_{CEL}^{TER} \cdot \mathbf{R}_{BOF}^{CEL} \cdot \mathbf{R}_{PL}^{BOF} \cdot \mathbf{R}_{SI}^{PL} \cdot \mathbf{r}_{object}^{SI}$$

with the unknown boresight misalignment matrix

$$\mathbf{R}_{PL}^{SI} = \mathbf{R}(\varepsilon_z) \cdot \mathbf{R}(\varepsilon_y) \cdot \mathbf{R}(\varepsilon_x)$$

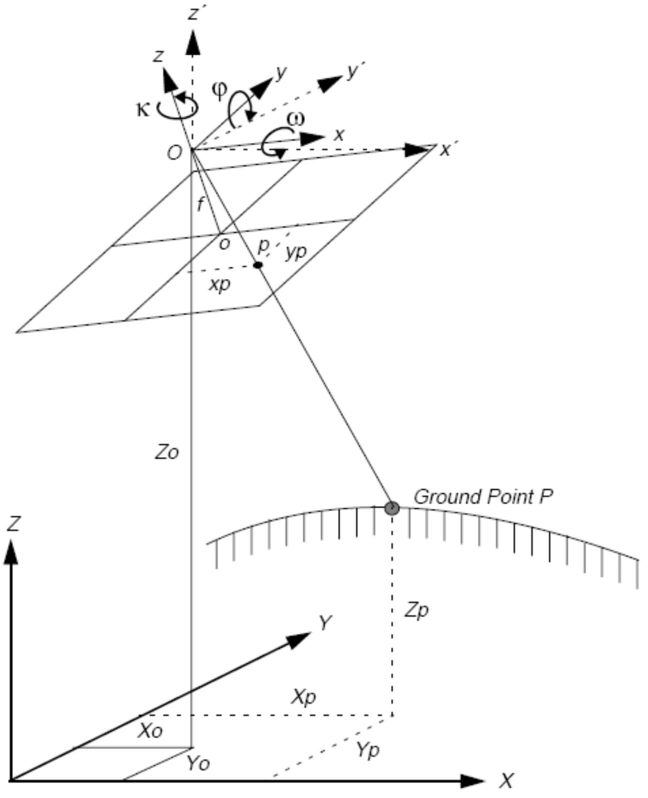
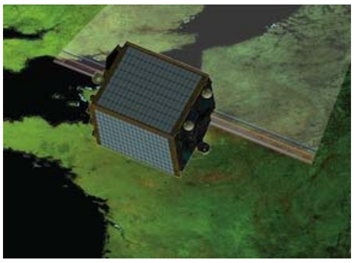


Figure 8.1-2: Exterior orientation – Boresight angles



These Euler angles of the boresight misalignment matrix are determined by least squares adjustment using the inverse formulation of the LoS model.

Rearranging the collinearity equation the sensor co-ordinates can be expressed to

$$\mathbf{r}_{object}^{SI} = \frac{\mathbf{1}}{s}\left(\mathbf{R}_{PL}^{SI} \cdot \mathbf{R}_{BOF}^{PL} \cdot \mathbf{R}_{CEL}^{BOF} \cdot \mathbf{R}_{TER}^{CEL}\right)\left(\mathbf{r}_{Object}^{TER} - \mathbf{R}_{CEL}^{TER} \cdot \mathbf{r}_{sensor}^{CEL}\right) = \frac{\mathbf{1}}{s}\mathbf{R}_{PL}^{SI} \cdot \left(\mathbf{r}_{Object}^{PL} - \mathbf{r}_{sensor}^{PL}\right) = \frac{\mathbf{1}}{s}C \cdot \left(\mathbf{r}_{Object}^{PL} - \mathbf{r}_{sensor}^{PL}\right)$$

The equations for the x- and the y- image co-ordinates can be divided by the equation for the z- image co-ordinate, which is the constant focal length f. The scale factor s cancels out.

$$x_{Object}^{SI} = x_{PP} + \Delta x - f \cdot \frac{c_{11}\left(x_{Object}^{PL} - x_{Sensor}^{PL}\right) + c_{12}\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right) + c_{13}\left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)}{c_{31}\left(x_{Object}^{PL} - x_{Sensor}^{PL}\right) + c_{32}\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right) + c_{33}\left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)} = x_{PP} + \Delta x - f \cdot \frac{U}{W} \tag{8.1-1.a}$$

$$y_{Object}^{SI} = y_{PP} + \Delta y - f \cdot \frac{c_{21}\left(x_{Object}^{PL} - x_{Sensor}^{PL}\right) + c_{22}\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right) + c_{23}\left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)}{c_{31}\left(x_{Object}^{PL} - x_{Sensor}^{PL}\right) + c_{32}\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right) + c_{33}\left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)} = y_{PP} + \Delta y - f \cdot \frac{V}{W} \tag{8.1-1.b}$$

Here also the principal points xPP , yPP , and the additional parameters Δx, Δy accounting for the adjustment of the interior orientation, are introduced according to the self-calibration model described in "Los Model" Section 5.2.

8.2 Bundle Adjustments

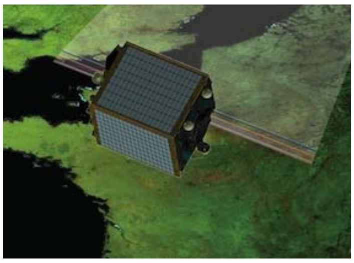
8.2.1 Linearization of Collinearity equations

The non-linear collinearity equations can be linearised by Taylor expansion with respect to the unknown parameters:

- three unknowns of the boresight misalignment angles,
- principal point xPP and yPP
- polynomial coefficients modelling the CCD thermal deformations

which lead for the GCPs (j=1,..., Amount of GCP) taken over the image acquired by the camera i = 1,2,3 to the system of linear equations

$$v_{ij} + \dot{B}_{ij}\dot{\delta}_i + \hat{B}_{ij}\hat{\delta}_i = F_{ij}$$



in which:

$v_{ij} = [v_x \; v_y]_{ij}^T =$ image coordinate residual

$\dot{B}_{ij} =$ the matrix of partial derivatives of the two collinearity equations with respect to the three boresight angles

$\hat{B}_{ij} =$ the matrix of partial derivatives of the two collinearity equations with respect to the interior orientation parameters (self-calibration)

$\dot{\delta}_i = [d\varepsilon_x \; d\varepsilon_y d\varepsilon_z]_{ij}^T$ the vector of three boresight corrections angles

$\hat{\delta}_i =$ the vector of the interior orientation corrections for self-calibration

$F_{ij} = [(x_{Object}^{SI}{}^{j} - x_{Object}^{SI}\big|_{GCP}^{j}) \; (y_{Object}^{SI}{}^{j} - y_{Object}^{SI}\big|_{GCP}^{j})]_i^T$ containing the differences between the observed GCP coordinates and the same coordinates evaluated substituting the approximations in the collinearity equations.

$$\dot{B}_{ij} = \begin{pmatrix} \left.\dfrac{\partial x_{Object}^{SI}}{\partial \varepsilon_x}\right|_{GCP}^{j} & \left.\dfrac{\partial x_{Object}^{SI}}{\partial \varepsilon_y}\right|_{GCP}^{j} & \left.\dfrac{\partial x_{Object}^{SI}}{\partial \varepsilon_z}\right|_{GCP}^{j} \\ \left.\dfrac{\partial y_{Object}^{SI}}{\partial \varepsilon_x}\right|_{GCP}^{j} & \left.\dfrac{\partial y_{Object}^{SI}}{\partial \varepsilon_y}\right|_{GCP}^{j} & \left.\dfrac{\partial y_{Object}^{SI}}{\partial \varepsilon_z}\right|_{GCP}^{j} \end{pmatrix}_i$$

$$\frac{\partial \, x_{Object}^{SI}}{\partial \varepsilon_x} = -\frac{f}{W}\left\{\left[\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right)c_{33} - \left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)c_{23}\right]\frac{U}{W} - \left(y_{Object}^{PL} - y_{Sensor}^{PL}\right)c_{31} + \left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)c_{21}\right\}$$

$$\frac{\partial \, x_{Object}^{SI}}{\partial \varepsilon_y} = \frac{f}{W}\left[\left(U \cos\varepsilon_z - V \; sen\varepsilon_z\right)\frac{U}{W} + W \cos\varepsilon_z\right]$$

$$\frac{\partial \, x_{Object}^{SI}}{\partial \varepsilon_z} = -\frac{f}{W}V$$

$$\frac{\partial \, y_{Object}^{SI}}{\partial \varepsilon_x} = -\frac{f}{W}\left\{\left[\left(y_{Object}^{PL} - y_{Sensor}^{PL}\right)c_{33} - \left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)c_{23}\right]\frac{V}{W} - \left(y_{Object}^{PL} - y_{Sensor}^{PL}\right)c_{32} + \left(z_{Object}^{PL} - z_{Sensor}^{PL}\right)c_{22}\right\}$$

$$\frac{\partial \, y_{Object}^{SI}}{\partial \varepsilon_y} = \frac{f}{W}\left[\left(U \cos\varepsilon_z - V \; sen\varepsilon_z\right)\frac{V}{W} + W \; sen\varepsilon_z\right]$$

$$\frac{\partial \, y_{Object}^{SI}}{\partial \varepsilon_z} = -\frac{f}{W}U$$

Add $\hat{\boldsymbol{B}}_{\boldsymbol{ij}}$

$$\hat{B}_{ij} = \left(\begin{array}{ccc} \left.\dfrac{\partial x^{SI}_{Object}}{\partial p_{x0}}\right|^{j}_{GCP} & \cdots & \left.\dfrac{\partial x^{SI}_{Object}}{\partial p_{xP}}\right|^{j}_{GCP} \\ \left.\dfrac{\partial y^{SI}_{Object}}{\partial p_{y0}}\right|^{j}_{GCP} & \cdots & \left.\dfrac{\partial y^{SI}_{Object}}{\partial p_{yP}}\right|^{j}_{GCP} \end{array} \right)_{i}$$

$$\frac{\partial\, x^{SI}_{Object}}{\partial p_{yp}} = x^{p}$$

$$\frac{\partial\, y^{SI}_{Object}}{\partial p_{yp}} = y^{p}$$

8.2.2 Normal Equation Formation with Self-Calibration Parameters

If we now form the least-squares normal equations associated with one image of one point we obtain

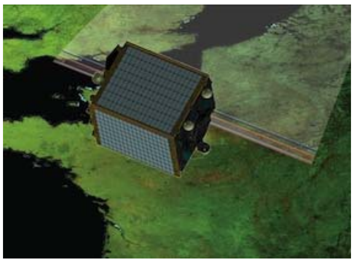
$$\begin{bmatrix} \dot{B}^{T}_{ij} W_{ij} \dot{B}_{ij} & \dot{B}^{T}_{ij} W_{ij} \hat{B}_{ij} \\ \hat{B}^{T}_{ij} W_{ij} \dot{B}_{ij} & \hat{B}^{T}_{ij} W_{ij} \hat{B}_{ij} \end{bmatrix} \begin{bmatrix} \dot{\delta}_{i} \\ \hat{\delta}_{i} \end{bmatrix} = \begin{bmatrix} \dot{B}^{T}_{ij} W_{ij}\, f_{ij} \\ \hat{B}^{T}_{ij} W_{ij}\, f_{ij} \end{bmatrix} \qquad (8.2\text{-}2)$$

where $\boldsymbol{W_{ij}}$ is the 2 x 2 weight matrix (the inverse of the covariance matrix) associated with the image coordinates of point j on image i. The equation 8.2-2 can be rewritten as

$$\begin{bmatrix} \dot{N}_{i} & \breve{N}_{i} \\ \breve{N}_{i}^{\,T} & \hat{N}_{i} \end{bmatrix} \begin{bmatrix} \dot{\delta}_{i} \\ \hat{\delta}_{i} \end{bmatrix} = \begin{bmatrix} \dot{t}_{i} \\ \hat{t}_{i} \end{bmatrix} \qquad (8.2\text{-}3)$$

Note that $\dot{\boldsymbol{N}}_{\boldsymbol{i}}$, $\breve{\boldsymbol{N}}_{\boldsymbol{i}}$ and $\hat{\boldsymbol{N}}_{\boldsymbol{i}}$ contain coefficients of the image parameters (boresight orientation parameters, cross-term of the self-calibration parameters and the boresight orientation parameters, self-calibration parameters respectively).

We now make the pivotal assumption that the errors in each point's image coordinates are uncorrelated with those of any other image coordinates, although error in the x and y coordinates of any single image point may be correlated. If image coordinate errors are correlated, they are affected by the same



systematic errors instead of being purely random; when we assume that the image coordinate errors are uncorrelated, we are assuming that all systematic errors have been removed.

In fact, it is assumed that it is possible to find an appropriate self-calibration model (e.g. proper deformation polynomial model with the correct number of coefficients) which makes any systematic errors negligible with respect to the accuracy requirements. Additional biases are introduced by the on-board attitude parameters and satellite position errors which are not part of the calibration process, but according to Table 8.1.1 these biases are very small compared to the accuracy requirements.

This assumption means that the total covariance Σ and weight $\boldsymbol{W}$ matrices containing the covariance and weight information for all the image coordinates are block diagonal. Each block is a 2 x 2 matrix corresponding to one set of image measurements for one point. Since the measurements are independent of each other, and each set of collinearity equations refers to only one image and one point, we can sum the contributions to the normal equations from each set of collinearity equations. The total $\dot{\boldsymbol{N}}$ matrix is a 3 x 3 matrix. Each $\dot{\boldsymbol{N}}_i$ is the sum of the $\dot{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \dot{\boldsymbol{B}}_{ij}$ matrices from each set of collinearity equations that refer to image i. Similarly, $\hat{\boldsymbol{N}}$ is a N x N matrix (N is the number of self-calibration parameters). Each $\hat{\boldsymbol{N}}_i$ is the sum of the $\hat{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \hat{\boldsymbol{B}}_{ij}$ matrices from each set of collinearity equations that refer to image i. $\breve{\boldsymbol{N}}$ matrix 3 x N matrix. Each $\breve{\boldsymbol{N}}_i$ is the sum of the $\dot{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \hat{\boldsymbol{B}}_{ij}$ matrices from each set of collinearity equations that refer to image i.

Therefore, the dimension of the normal equation is (3+N) x (3 + N), where N is the number of self-calibration parameters

The advantages of this method of equation formation is obvious, since the alternative would be to first form the entire B matrix and then calculate N by brute force multiplication. For realistic number of images and points the total B matrix will be mostly zeros, meaning that large amounts of storage would be used to store zeros an that the majority of the effort in forming N would be multiplying be zero.
The tie-point correction variables are not useful for geometric calibration, therefore the least-squares normal equations can be reduced by eliminating the point correction variables generating the reduced normal equations.

8.2.3 Unified Least Squares Adjustment

We expect interior and exterior parameters to be highly correlated; in particular we expect the exterior parameters to be correlated with a linear combination of a specific sub-set of interior parameters.

It is important to point out that we are not interested to univocally determine the interior and exterior orientation separately but on the contrary to estimate a combination of them in order to have a calibrated/geolocated system corrected image. However, correlated parameters may lead to singular (not invertible) least square normal equations. We proposed the Unified Least Squares method [MIK] which introduces a-priori weight for each calibration parameters. These weights act as a regularization method (e.g. Tikhonov regularization method) avoiding singularities in the modified normal equation.

The following algorithm shows a simplified version of the normal equation formation process for a solution involving self-calibration parameters and also additional constraint equations. The procedure calculates the partial derivate matrices and discrepancy vectors and adds their contributions to the corresponding portions of the normal equation matrices.

Algorithm for formation of Banded-Bordered Normal Equation with Added Parameters (Unified Least Squares)

```
for each added parameter k
        $\hat{\boldsymbol{N}}_k = \hat{\boldsymbol{W}}_k$
        $\hat{\boldsymbol{t}}_k = -\hat{\boldsymbol{W}}_k \hat{\boldsymbol{f}}_k$
end added parameter k loop

/* add the contributions of the collinearity equations, possibly with self-calibration parameters */

for each image i
        $\dot{\boldsymbol{N}}_i = \dot{\boldsymbol{W}}_i$
        $\dot{\boldsymbol{t}}_i = -\dot{\boldsymbol{W}}_i \dot{\boldsymbol{f}}_i$
    for each point j
            if point j is on image i calculate $\dot{\boldsymbol{B}}_{ij}$ , and $\boldsymbol{f}_{ij}$
                $\dot{\boldsymbol{N}}_i = \dot{\boldsymbol{N}}_i + \dot{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \dot{\boldsymbol{B}}_{ij}$
                $\dot{\boldsymbol{t}}_j = \dot{\boldsymbol{t}}_i + \dot{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \boldsymbol{f}_{ij}$
                if self-calibration parameter k is used calculate $\hat{\boldsymbol{B}}_{ijk}$
                    $\breve{\boldsymbol{N}}_{ik} = \breve{\boldsymbol{N}}_{ik} + \dot{\boldsymbol{B}}_{ij}^T \boldsymbol{W}_{ij} \hat{\boldsymbol{B}}_{ijk}$
                    $\hat{\boldsymbol{N}}_k = \hat{\boldsymbol{N}}_k + \hat{\boldsymbol{B}}_{ijk}^T \boldsymbol{W}_{ij} \hat{\boldsymbol{B}}_{ijk}$
                $\boldsymbol{\hat{t}_k = \hat{t}_k + \hat{B}^T_{ijk} W_{ij} f_{ij}}$
    *end point j loop*
*end image loop i*
```

8.2.4 Solution Algorithms

The best way to compute the $\dot{\delta}_{i\,ij}$ (the vector of three boresight corrections angle) and the $\hat{\delta}_i$ (the vector of the interior orientation corrections for self-calibration) is not by inverting the normal equation coefficient matrix and multiplying by the constant terms, as is usually written.

Instead, direct or iterative solution methods are employed that do not explicitly form the inverse, since calculating the inverse is much more computationally expensive than just solving the equations.

The most straightforward direct solution method is Gaussian elimination. Unknowns are progressively eliminated by adding or subtracting scaled rows of the coefficient matrix from one another. When only one unknown is left, its value is directly determined and the remaining unknowns determined by back-substitution. The coefficient matrix is essentially transformed into a triangular matrix, making the set of equations easier to solve.

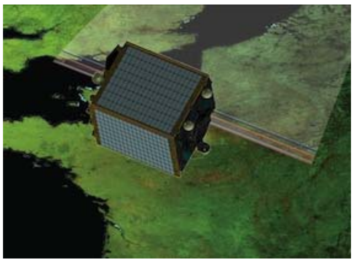
However, several more efficient matrix solution techniques exist, based on the factoring the coefficient matrix into triangular matrices. **The Cholesky method is used for PROBA-V** since it is particularly well-suited for solving least squares normal equations, because it is predicated on having positive-definitive symmetric matrices. The coefficient matrix is factored into a triangular matrix ***L*** which yields the original matrix when multiplied by its transpose.

8.2.5 Error propagation

To evaluate the results of the solution we need covariance matrix for the parameters $\dot{\delta}$ (the vector of three boresight corrections angle) and $\hat{\delta}$ (the vector of the interior orientation corrections for self-calibration), which is the inverse of the normal equation coefficient matrix:

$$\begin{bmatrix} \dot{\Sigma}_{\dot{\delta}\dot{\delta}} & \breve{\Sigma}_{\dot{\delta}\hat{\delta}} \\ \breve{\Sigma}_{\dot{\delta}\hat{\delta}}^{T} & \hat{\Sigma}_{\hat{\delta}\hat{\delta}} \end{bmatrix} = \begin{bmatrix} \dot{N} & \breve{N} \\ \breve{N}^{T} & \hat{N} \end{bmatrix}^{-1}$$

As we have seen (Section 7.2.4), direct solution methods such as the Cholesky do not produce the inverse of the normal equation coefficient matrix. However, calculating the inverse of a matrix ***N*** can be thought of as



solving *n* equations, one for each column $\boldsymbol{x_i}$ of $\boldsymbol{N^{1}}$, each of which has its right-hand side a column of the identity matrix $\boldsymbol{I_i}$:

$$\boldsymbol{N\, x_i = I_i}$$

where $\boldsymbol{I}$ is an identify matrix with the same dimension as $\boldsymbol{N}$. Using the Cholesky method to solve the equation, the matrix does not have to refactored to solve the equations with different right-hand sides. If we keep the last factorization of the coefficient matrix after the solution has converged, the inverse is computed with a minimum of extra work.

8.2.6 Outlier Detection

Background

Outlier detection seeks to identify GCPs that are likely to be in error due to miscorrelation. This is done by analyzing the GCP residuals, taking into account the relative importance of the GCP as reflected in the precision solution normal equation matrix.

According to the nomenclature so far used, the following definitions are reported for convenience:

$\boldsymbol{B}$ = matrix of coefficients (partial derivatives) relating parameters to observations

$\boldsymbol{\delta}$= parameter correction vector

$\boldsymbol{F}$ = observation vector (differences between the observed and predicted GCP/TP coordinates)

$\boldsymbol{v}$ = residual vector

$\boldsymbol{\Sigma}$ = observation covariance matrix

$\boldsymbol{n}$ = the number of observations

$\boldsymbol{p}$ = the number of parameters

$\boldsymbol{B}$ is $\boldsymbol{n \; x \; p}$, $\boldsymbol{\delta}$ is $\boldsymbol{p \; x \; 1}$, $\boldsymbol{F}$ and $\boldsymbol{v}$ are $\boldsymbol{n \; x \; 1}$, and $\boldsymbol{\Sigma}$ is $\boldsymbol{n \; x \; n}$

Observation equation:

$$\boldsymbol{v + B\delta = F}$$

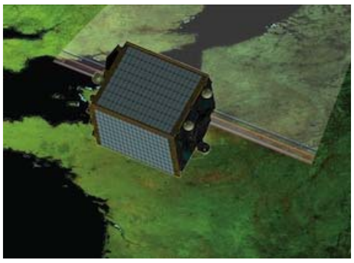
$$\boldsymbol{F = F_{true} + E}$$

where $\boldsymbol{E}$ = ***error vector ~ G(0, Σ)***

$$\boldsymbol{B\delta_{true} = \underline{F}_{true}}$$

where $\boldsymbol{\delta_{true}}$ is the "true" parameter correction vector

$$\boldsymbol{B\delta = F_{true} + E - v}$$



therefore:

$$\boldsymbol{v} = \boldsymbol{E} \text{ if } \delta = \delta_{true}$$

In other words, once the “true” parameter correction vector is found the residual systematic errors are eliminated.

Minimum variance parameter correction estimate:

$$\delta' = [B^T \Sigma^{-1} B]^{-1} B^T \Sigma^{-1} F$$

Estimated residual error:

$$v' = F - B\,[B^T \Sigma^{-1} B]^{-1}\, B^T \Sigma^{-1} X$$

Define projection matrix ***P***:

$$P = [B^T \Sigma^{-1} B]^{-1} B^T \Sigma^{-1}$$

This matrix projects the observation vector into the parameter subspace (the column space of ***B***).

This projection is only orthogonal if Σ has the special structure described below.

Substituting:

$$v' = F - P\,F = [I - P]\,F$$

[I – P] projects ***F*** into the parameter null space.

Looking at the error estimate ***v’***:

$$v' = [I - P]\,F = [I - P]\,[F_{true} + E] = [I - P]\,F_{true} + [I - P]E$$

but $[I - P]\,F_{true} = 0$, since F_{true} lies entirely within the parameter subspace.

Therefore:

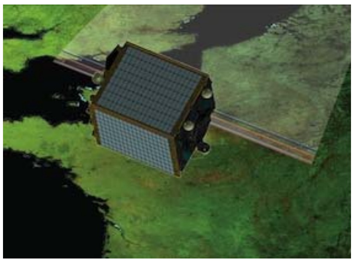
$$v' = [I - P]E = E - P\,E$$

Some comments about ***v’*** and ***E*** :

1. For a given precision solution, the elements of ***E*** are not random variables; they are realizations of random variables.

2. ***v***” is an estimate of the actual (realized) error ***E***, which includes an estimation error equal to ***PE***.

3. We cannot exactly recover ***E*** from $[I - P]^{-1}\, v'$ because ***[I – P]*** is singular (it is an ***n x n*** matrix of rank ***n–p***).



4. We can attempt to predict how accurate our estimate ($\boldsymbol{v''}$) of $\boldsymbol{E}$ is likely to be by looking at the estimation error $\boldsymbol{R = P\ E}$.

5. Since we want the predicted accuracy to apply in general, we treat $\boldsymbol{R}$ as a random vector that is a function of another random vector $\boldsymbol{E}$.

   Expected value: $\boldsymbol{E[R] = E[P\ E] = P\ E[E] = P\ 0 = 0}$

   Variance: $\boldsymbol{E[R\ R^T] = E[P\ E\ E^T\ P^T] = P\ E[E\ E^T]\ P^T = P\ C\ P^T}$

Special structure of observation covariance matrix for Geometric Calibration:

$$\boldsymbol{\Sigma = \sigma^2 I}$$

since the observation errors are realizations of independent and identically distributed zero mean Gaussian random variables with variance $\boldsymbol{\sigma^2}$.

Substituting into the equation for $\boldsymbol{P}$ yields:

$$\boldsymbol{P = B\ [(1/\sigma^2)B^T\ I\ B]^{-1}\ B^T\ I\ (1/\sigma^2) = B\ \sigma^2[B^T\ B]^{-1}B^T(1/\sigma^2) = B\ [B^TB]^{-1}B^T}$$

and the equation for the variance of $\boldsymbol{R}$ is:

$$\boldsymbol{E[R\ R^T] = \sigma^2\ P\ I\ P^T = \sigma^2\ P}$$

noting that $\boldsymbol{P^T = P}$ and $\boldsymbol{P\ P = P}$

therefore:

$$\boldsymbol{R \sim G(0,\ \sigma^2\ P)}$$

For a particular component of $\boldsymbol{R}$ $\boldsymbol{r_i}$:

$$\boldsymbol{E[r_i] = 0}$$

$$\boldsymbol{E[r_i^2] = \sigma^2\ p_{ii}}$$

where $\boldsymbol{p_{ii}}$ is the ith diagonal component of $\boldsymbol{P}$.

Looking at the equation for $\boldsymbol{P}$, we see that:

$$\boldsymbol{p_{ii} = B_i^T\ [B^TB]^{-1}\ B_i}$$

where $\boldsymbol{B_i^T}$ is the ith row of $\boldsymbol{B}$.

Considering a particular component of the residual error vector $\boldsymbol{v''}$:

$$v_i = e_i - r_i$$

where e_i is the corresponding component of the observation error vector, so v_i is an unbiased estimate of e_i with variance $\sigma^2 p_{ii}$.

Algorithm

If we know what e_i is, then we can test it against a probability threshold derived from its standard deviation, σ, to determine if it is likely to be an outlier. Instead of e_i , however, we have v_i , which includes the additional error term r_i . Including the additional estimation error in the threshold computation leads to:

$$\sigma_v^2 = \sigma^2 + \sigma^2 p_{ii}$$

where σ^2 is the term due to the actual error variance, and $\sigma^2 p_{ii}$ is the term due to the estimation error variance.

One remaining problem is that we do not know exactly what σ^2 is and must estimate it from the observation residuals.

This is done by scaling the a priori observation variance using the variance of unit weight computed in the precision solution. The fact that we are using an estimated variance to establish our outlier detection threshold modifies the algorithm in two ways:

1. we compensate for the fact that removing a point as an outlier alters the computation of the variance of unit weight by removing one residual and reducing the number of degrees of freedom;

2. we base the detection threshold computation on Student's t-distribution rather than the Gaussian distribution.

The variance of unit weight is computed as:

$$var_0 = v^T \Sigma^{-1} v/(n-p) = v^T\, v/\sigma_0^2\,(n-p) = \sigma_j\, v_j^2/\sigma_0^2(n-p)$$

where:

n = number of observations

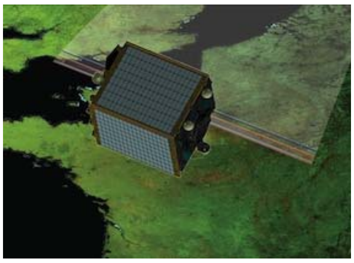
p = number of parameters

σ_0^2 = the a priori variance

The estimated variance is:

$$var = var_0\, \sigma_0^2 = \sigma_j\, v_j^2/(n-p)$$

Removing the kth observation makes this:



$$var_k = (\sigma_j v_j^2 - v_k^2)/(n - 1 - p) = (n - p)/(n - p - 1) * (\sigma_j v_j^2 - v_k^2)/(n - p)$$

$$var_k = (n - p)/(n - p - 1) * var - v_k^2/(n - 1 - p)$$

To normalize the kth residual, we divide it by the estimated standard deviation $\sigma' = (var)^{1/2}$:

$$w_k = v_k / \sigma'$$

We can rescale this normalized residual to reflect the removal of this observation from the variance estimate without actually having to compute a new variance:

$$w_k' = v_k / \sigma_k' = w_k\ \sigma'/\sigma_k' = w_k\ (var/var_k)^{1/2}$$

$$var/var_k = 1 / [(n - p)/(n - p - 1) - v_k^2/var\ (n - p - 1)] =$$

$$= (n - p - 1)/(n - p - v_k^2/var)$$

$$var/var_k = (n - p - 1)/(n - p - w_k^2)$$

noting that $v_k^2/var = w_k^2$

$$w_k' = w_k\ [(n - p - 1)/(n - p - w_k^2)]^{1/2}$$

Finally, we include the $(1 + p_{kk})$ factor discussed above, and our normalized and reweighted residual becomes:

$$w_k' = w_k\ [(n - p - 1)/(1 + p_{kk})(n - p - w_k^2)]^{1/2}$$

where:

$$w_k = v_k / \sigma'$$

This normalized and reweighted residual is compared against a probability threshold computed using Student’s t-distribution with $(n - p)$ degrees of freedom.

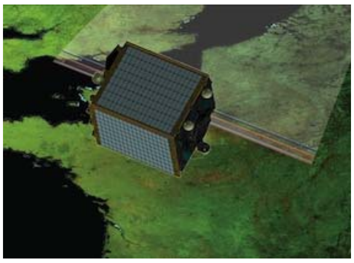
A summary of procedures for bundler detection is as follows:

Step 1: Locate all the normalized and reweighted residuals that meet the rejection criteria defined using Student’s t-distribution threshold.

Step 2: Remove the largest detected blunder or unrelated blunder group.

Step 3: Rerun the adjustment.

Step 4: Continue steps 1 through 3 until all detected outliers are removed.



8.2.7 Evaluation of the bundle adjustments results

After running a bundle adjustment, a careful evaluation of its results will be performed in order to be sure that all the specifications and requirements are meet and that the results are valid – that is not contaminated by bad measurements or assumptions.

After the bundle adjustment is run, a qualitative evaluation is performed as a first step, in which the operator examines graphical representations of the adjustment's output in order to understand broad trends and to catch obviously bad inputs (e.g. GCP chips outliers not correctly detected). Finally, statistical analysis techniques are used to protect against bad observations and to quantify the quality of the adjustment.

Qualitative evaluation

The purpose of qualitative evaluation is to allow the operator to understand the properties of the adjustment and to diagnose any problem. It is difficult to recognize trends or outliers by looking at pages of numbers; instead, since this is geometric problem, it is best addressed by graphical displays of the solution outputs.

The plots proposed for qualitative evaluation of results of the bundle adjustment are of the image residuals, as in Figure 8.2-1. The image residuals should point in random directions and have comparable sizes.

Residuals all pointing in the same direction (either parallel or radially with respect to the center of the image) indicate the presence of a systematic effect. These effects may indicate uncorrected errors, or an image parameter that has been weighted too highly and not allowed to adjust.

A residual that is larger than its neighbors or points in the opposite direction to those near it indicates a bad measurement/outliers nor correctly removed during the bundle adjustment phase.

Statistical evaluation

As with any other least square solution, statistical methods are used to evaluate the results of a bundle adjustment.

We evaluate the **precision** of the solution by examining the covariances of the parameters as outlined in section 7.2.5. The diagonal elements of the covariance matrix are the parameter variances, and the off diagonal elements are the covariances between the parameters variances.

The relationship between the parameters covariance matrix and the on ground measurement accuracy is performed by the Sensitivity Analysis as reported in Section 8.4.

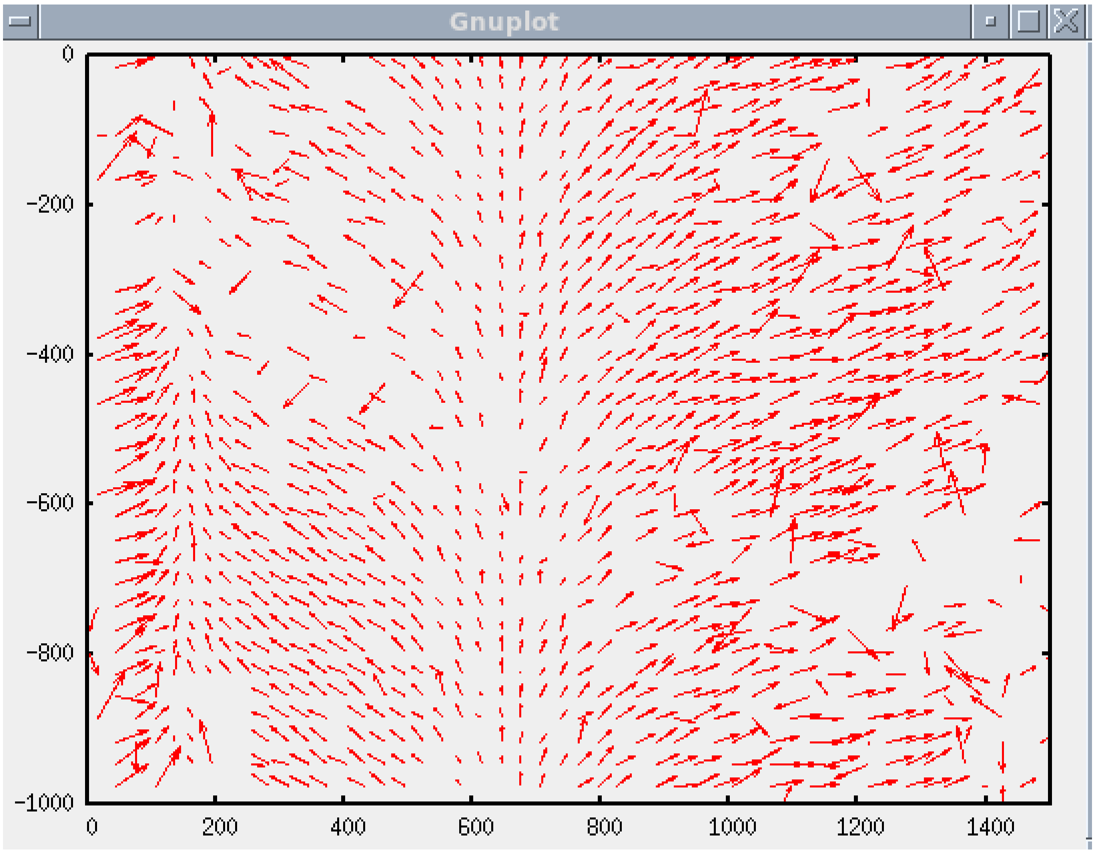


Figure 8.2-1: Example of residual plots

A fundamental requirement for the bundle adjustment is that it be **accurate** . Accuracy cannot be determined by examining the solution, since a solution can be very consistent within itself, but still not accurate. The accuracy can be determined by computing statistical average and covariance of the GCPs residuals.

8.3 Scene self calibration (SceneInv)

8.3.1 Introduction

Scene self calibration is performed by weighted non-linear least squares fitting with outlier removal.

8.3.2 Input/Output

Input :

- Expected Exterior Orientation parameters (boresight angles) from ICP file.
- Expected Scene Interior Orientation parameters (principal point offsets, polynomial CCD distortions) from ICP file.
- A-priori expected Scene parameter errors in the form of 1sigma errors.
- Line and pixel positions of GCPs predicted using expected deformat
- Line and pixel positions of GCPs found by cross-correlation with image chips.
- SNR from GCP cross-correlation.
- SNR threshold

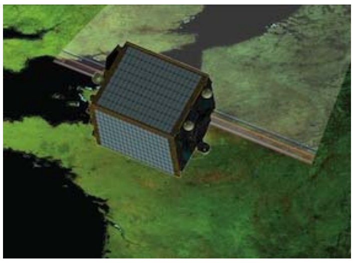
Output:

- estimated Exterior Orientation parameters (boresight angles)
- Scene Interior Orientation parameters (principal point offsets, polynomial CCD distortions)
- Scene parameter errors in the form of 1sigma errors and covariance matrix.
- Identification of outliers.

8.3.3 Algorithm

While least squares fitting is done as described in sections 8.2.2 to 8.2.4, non-linear least square fitting is done by simply repeating linearization as described in section 8.2.1 and linear least squares fitting a few times so that the result does not change significantly any more.

Outlier removal is done as described in section 8.2.6.



Summaring the algorithm used is as follows:

```
Outlier removal for NIR band.

Weighted non-linear least squares fitting of boresight angles and
focal plane distortions with.

For each other strip

    Outlier removal for strip.

    Weighted non-linear least squares fitting of focal plane
    distortions using boresight angles calculated from NIR band.
```

Weighted least squares minimization finds the parameter values which minimize $\chi^{2,}$ the weighted sum of residuals, where the weights are given by $1/\sigma^2$, and σ is the expected standard deviation of the measurement.

In our case the chi squared function being minimized is given by:

$$\chi^2 = \sum_i^N \left[\frac{Y_i - f(y_i)}{\sigma_{yi}} \right]^2 + \sum_i^N \left[\frac{X_i - g(y_i)}{\sigma_{xi}} \right]^2 + \sum_j^P \left[\frac{P_j - p_j}{\sigma_{pj}} \right]^2$$

where:

y_i is the calculated SICy pixel position of a GCP in nominal conditions

Y_i is the found SICy pixel position

X_i is the found SICx pixel position

P_j is the expected value of parameter j

f_p(yi) is the function of P parameters predicting SICy pixel position of

g_p(yi) is the function of P parameters predicting SICx pixel position

σ_{yi}, σ_{xi}, σ_{pj} are the expected standard deviations of Y_i, X_i and P_j respectively

P is the number of parameters

N is the number of GCPs

Nominal focal length is a fixed value in the configuration file. The value is an approximation.

Please note that σ_{pj} are a-priori weight for each calibration parameters introduced in the Section 8.2.3 (Unified Least Squares method).

The standard deviations σ_{yi} and σ_{xi} are expected to depend mostly on errors due to image corregistration and errors in the GCP location.

The RMS errors in the GCP location of the geocover dataset are nominally 50 m but have been evaluated to be approximately 25 m. Expresssed in ProbaV pixels this results in a an error ranging from 0.25 (for 100m VNIR pixels at Nadir) to 0.036 (for 700m external SWIR).

The statistics of the errors introduced by image correlation were estimated from extensive cross-correlation tests between images at ProbaV pixel spacing generated from the overlap of geocover datasets from different WRS2 paths at higher latitudes.A regular grid of 9x9 image patches taken from the overlap between 375 image pairs (from WRS2 rows 20 to 29 and paths 151 to 208) from each of the 4 bands were warped, down sampled and then correlated to simulate correlation between ProbaV images and geocover chips from different dates and look angles.

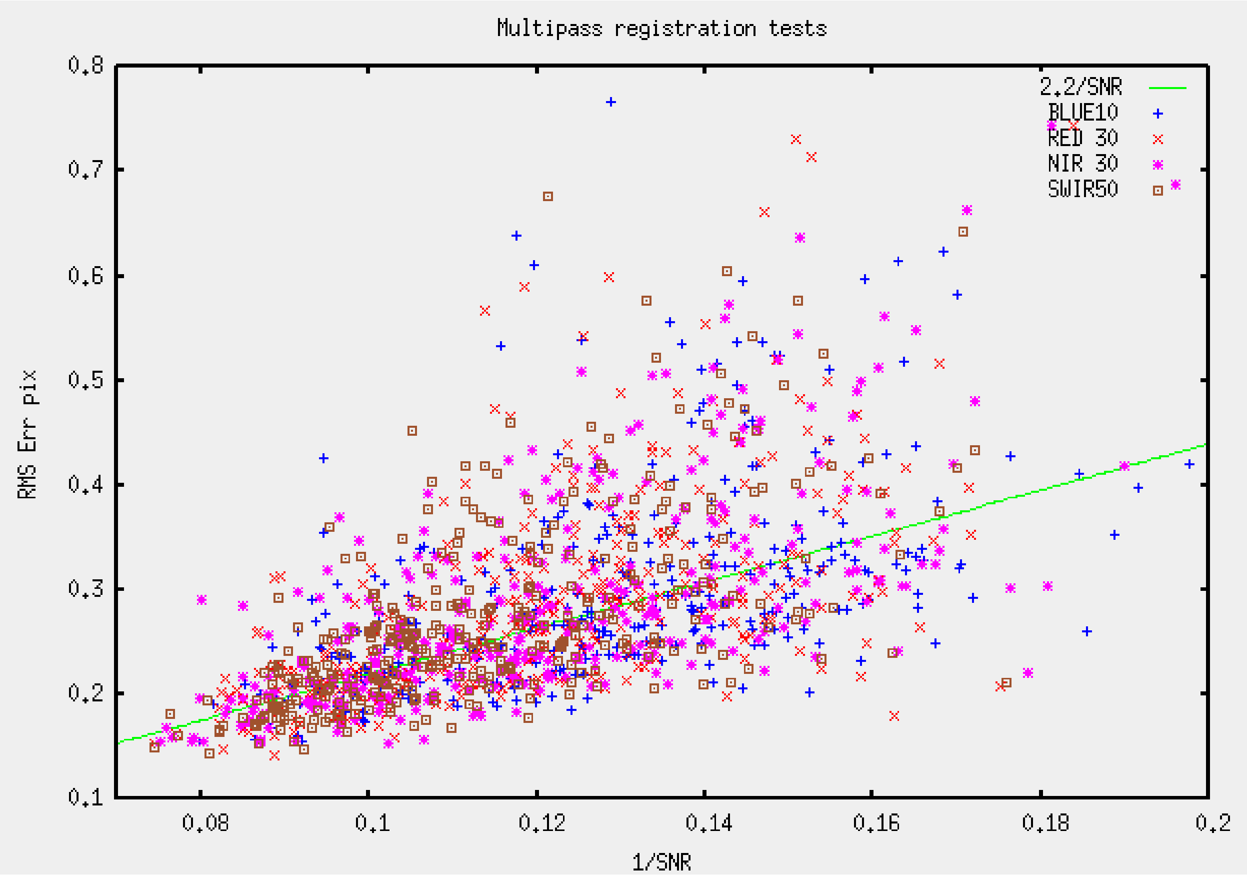
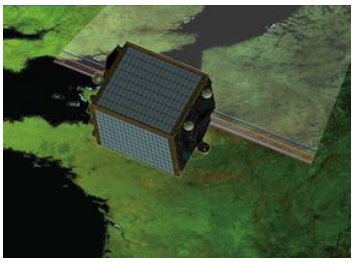


Figure 8.3-1: SNR to RMS Err relation

The preceding scatter plot shows the distribution of SNR and RMS error found. It can be seen that most tie-points show RMS errors between 0.13 and 0.6 pixels. Pixels outside this range will probably be detected as outliers. It can be seen that the expected RMS error could be approximated by 0.3 pixel, but slightly better results will probably be obtained by approximating



RMS error = 2.2/SNR

resulting in a weight relation up to ~ 4 between high and low SNR tie-points.

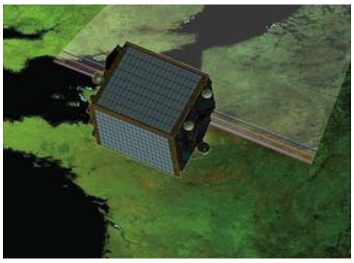
From these considerations we can use the following heuristic function for σ_{yi} and σ_{xi},

$$\sigma = \sqrt{\left(\frac{\sigma_m}{P}\right)^2 + \left(\frac{2.2}{SNR}\right)^2}$$

where:

σ_m = 25m

P is the local pixel spacing on ground.



8.4 Sensitivity Analysis

8.4.1 Introduction

For the assessment of the geo-location error sources the parameters of the LoS model are varied according to their uncertainties, which produces as output varying object co-ordinates. This error source analyzing tool allows to:

- determine the limiting geo-location accuracy;
- verify the requirements of sensor satellite systems w.r.t geometric performance;
- perform sensitivity analysis of the sensor model to the thermo-elastic deformation on the integrated Geometric Calibration system; input errors and uncertainties are modelled throughout the system.

8.4.2 Input/Output

Input:

- Focal plane distortion parameters and errors
  - $\dot{\delta}$ Boresight angle and $\hat{\delta}$ Interior Orientation parameters;
  - Covariance matrix for the parameters $\dot{\delta}$ (the vector of three boresight corrections angle) and $\hat{\delta}$ (the vector of the interior orientation corrections for self-calibration);
- Spacecraft nominal parameters;
- Spacecraft orbital and attitude standard deviation.

Output:

- Root Mean Square Error (RMSE) and the Standard Deviation (STDV) at different point of the FOV (e.g. nadir, side looking) for each strip.
- Error curves to be displayed.

8.4.3 Algorithms

The basic component of the tool for geo-location error modelling is the LoS model to the thermo-elastic deformation developed in section 5.2.

The inputs for the LoS model are parameters (position, velocity, sensor calibration values,…) with a specific accuracy derived from performance specifications of the individual measurement devices or assumptions of expected variations.

Figure 8.4-1 shows the principle components of the error analyzing tool:

3 The values of parameters for the LoS model, necessary for the mapping process, have to be defined (position of the satellite, actual look direction, sensor parameters …).

2 Subsets of parameters of the LoS model can be selected for the error simulation. The others will be assumed to be exact

1 The variation of the selected parameters is performed via Gaussian noise given by the standard deviation σ and/or covariance matrix of the parameters (Monte Carlo method)

4 Additional inputs for the LoS model are the sensor description and the processing definitions. For example to process not only one look direction but the whole FOV range.

5 The outputs of the geo-location error are given in terms of Root Mean Square Error (RMSE) and the Standard Deviation (STDV). Provided a parameter range is processed (for example the whole FOV) the error can be displayed in error curves.

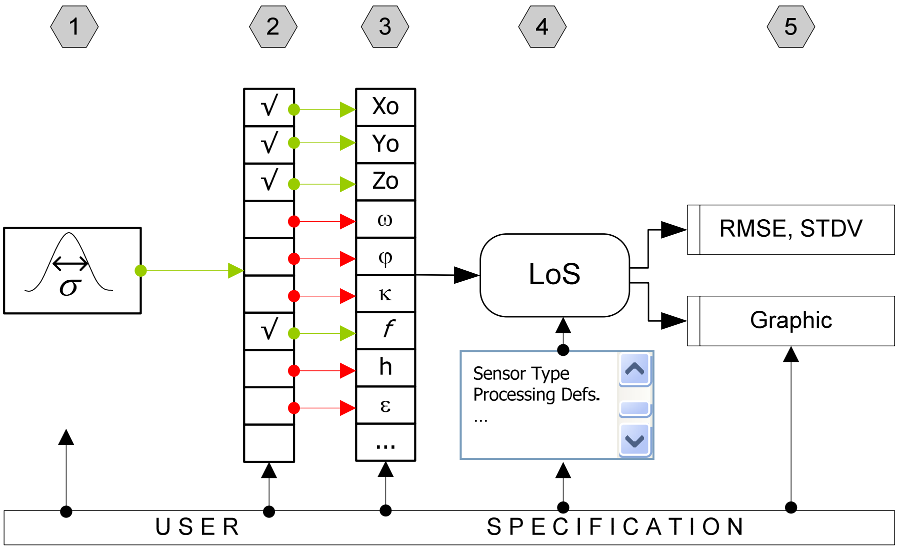


Figure 8.4-1: Flow chart of sensitivity analysis tool (the green arrows show a possible selected configuration; the red ones possible selections)

9. ICP FILE GENERATION

This section reports a description of the methods used for the ICP file generation.

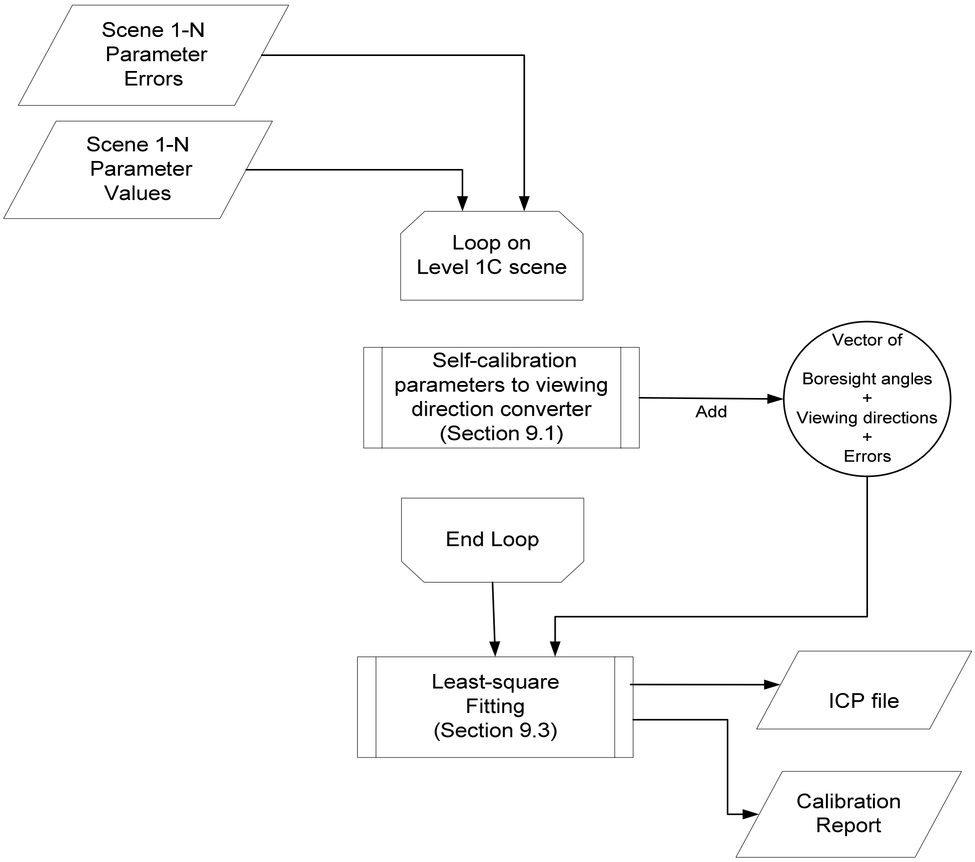
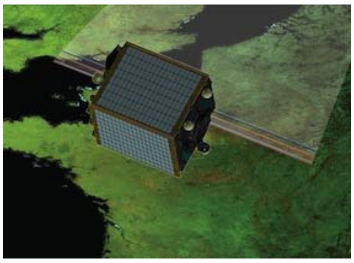

Scene 1-N Parameter Errors
Scene 1-N Parameter Values
Loop on Level 1C scene
Self-calibration parameters to viewing direction converter (Section 9.1)
Add
Vector of
Boresight angles
+
Viewing directions
+
Errors
End Loop
Least-square Fitting (Section 9.3)
ICP file
Calibration Report

Figure 9-1: ICP generation flowchart



9.1 Focal plane distortion parameters to Viewing Direction converter

In this section is reported the main algorithm steps in order to go from focal plane distortions to viewing directions.

The viewing directions for pixel i can be calculated as follows:

$$
\begin{aligned}
v_Y &= \mathrm{p}_{\mathrm{NOM}} \cdot \mathrm{D}_{\mathrm{Y}} + \mathrm{Y}_{\mathrm{PP}} + \mathrm{p}_{\mathrm{NOM}} \cdot i \cdot \frac{N}{2} \\
v_X &= \mathrm{p}_{\mathrm{NOM}} \cdot \mathrm{D}_{\mathrm{X}} + \mathrm{X}_{\mathrm{PP}} \\
v_Z &= \mathrm{f}_{\mathrm{NOM}}
\end{aligned}
$$

after which the viewing direction vector should be normalized to unit length

where:

DY, DX are the distortions in SICY and SICX pixels respectively

$\mathrm{f}_{\mathrm{NOM}}$ is the nominal focal length

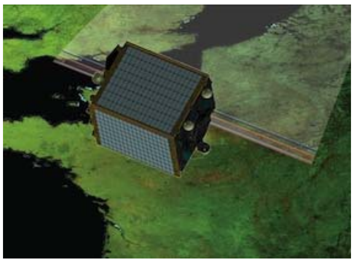
$\mathrm{p}_{\mathrm{NOM}}$ is the nominal VNIR pixel spacing

v_{X}, v_{Y}, v_{Z} are the components of the viewing direction versor.

Xpp,Ypp are the nominal principal point positions

N is the number of pixels in the CCD strip

Please note that the focal length is fixed while the other parameters are updated.



9.2 Viewing Direction to Focal plane distortion parameters converter

Instead of fitting viewing directions as found in the ICP file directly, viewing directions are first converted to distortions with respect to the nominal focal plane layout, expressed in VNIR pixels (see Figure 5.2.1 for the description of the nominal focal plane layout). Please note that the VNIR pixel is used also in the case of SWIR strips for an easy comparison of the results between different strips.

The distortions for pixel i can be calculated as follows:

$$\mathrm{D}_{\mathrm{Y}} = \frac{\mathrm{f}_{\mathrm{NOM}} \cdot v_Y}{\mathrm{p}_{\mathrm{NOM}} \cdot v_Z} - \frac{\mathrm{Y}_{\mathrm{PP}}}{\mathrm{p}_{\mathrm{NOM}}} - i \cdot \frac{N}{2}$$

$$\mathrm{D}_{\mathrm{X}} = \frac{\mathrm{f}_{\mathrm{NOM}} \cdot v_X}{\mathrm{p}_{\mathrm{NOM}} \cdot v_Z} - \frac{\mathrm{X}_{\mathrm{PP}}}{\mathrm{p}_{\mathrm{NOM}}}$$

where:

DY, DX are the distortions in SICY and SICX pixels respectively

$\mathrm{f}_{\mathrm{NOM}}$ is the nominal focal length

$\mathrm{p}_{\mathrm{NOM}}$ is the nominal VNIR pixel spacing

v_{X}, v_{Y}, v_{Z} are the components of the viewing direction versor.

Xpp,Ypp are the nominal principal point positions

N is the number of pixels in the CCD strip

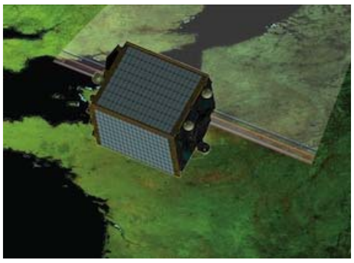
These distortions are then modeled with a generic (Taylor) polynomial D as function of the position x within the strip.

$$\mathrm{x} = \frac{2 \cdot i}{N} - 1$$

$$D = \sum_{j=0}^{P-1} p_j \cdot x^j$$

where:

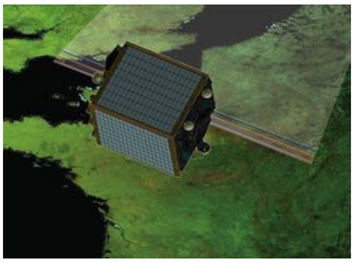
p_{j} is polynomial coefficient j



P is the number of polynomial coefficients

The main advantage of this is that results are easier to interpret and therefore maintenance and tuning are more efficient. Note that in this way the absolute contribution of p_j is $\leq p_j$.

Please note that the focal length is fixed while the other parameters are updated.



9.3 Trend Estimation (estTrend)

9.3.1 Introduction

Trend estimation is performed by weighted linear least squares fitting.

The value y of each boresight angle and of the distortions at a subset of pixel positions is modeled as a polynomial function of temperature and TimeOutOfEclipse by

$$y = \sum_{i=0}^{T-1} T_i \cdot t^i + \sum_{j=0}^{S-1} S_j \cdot s^j$$

where:

t is temperature

T_i is temperature polynomial coefficient i

T is the number of temperature polynomial coefficients (2 for expected linear case)

s_j is TimeOutOfEclipse

S_j is TimeOutOfEclipse polynomial coefficient j

S is the number of polynomial TimeOutOfEclipse coefficients

When we have a series of measures of each parameter y under different temperature and TimeOutOfEclipse conditions we can used a weighted LS fit to calculate optimal values for the above polynomial coefficients and corresponding covariance matrices to describe their errors.

Once we have done this we can estimate the expected parameter values and their confidence limits for any temperature and TimeOutOfEclipse conditions simply evaluating the polynomials.

9.3.2 Input/Output

Input :

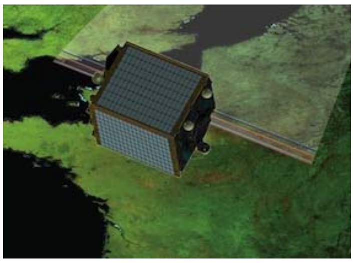
- Exterior Orientation parameters (boresight angles) for each scene.
- Interior Orientation parameters (principal point offsets, polynomial CCD distortions) for each scene.
- Scene parameter errors in the form of covariance matrix for each scene.
- Temperature and TimeOutOfEclipse for each scene.

Output:

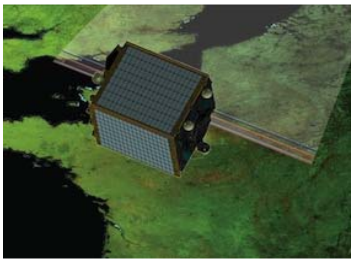
- ICP file containing boresight angles and focal plane distorsions in the form of pixel viewing directions, at various positions in Temperature - TimeOutOfEclipse matrix.
- Trend fit errors in the form of covariance matrices.

9.3.3 Algorithm

```
For each scene
    Store Temperature and TimeOutOfEclipse
    For each boresight angle
        Store boresight angle in vector
        Store boresight standard deviation  in vector
    For a subset of pixel positions
        For SICx & SICy distortion
            Calculate distortion by evaluating distortion polynomial.
            Store distortion in vector
            Calculate standard deviation in distortion from covariance matrix
            Store distortion standard deviation  in vector
For each vector position
    Weighted LSfit against Temperature and TimeOutOfEclipse
For a uniformly spaced series of TimeOutOfEclipse values
    For a pair of temperatures
        For each boresight angle
            Evaluate boresight angle polynomial
            Store in ICPfile
```



```
For each  pixel positions
    For SICx & SICy distortion
        Evaluate distortion polynomial
    Convert distortions to viewing directions
    Store in ICPfile
```



9.4 Estimating errors from covariance matrices

9.4.1 Introduction

When an estimate of a function has been done using the least squares method and generating a covariance matrix the latter can be used to estimate the standard deviation or confidence

9.4.2 Input/Output

Input :

- covariance matrix.
- Position at which function is to be estimated
- Functions to evaluate partial derivatives

Output:

- standard deviation of estimated function.

9.4.3 Algorithms

An estimate of the standard deviation σ of function F(x) at *x* is given by:

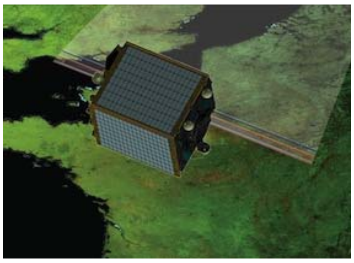
$$\sigma = \sqrt{\sum_{i=0}^{P-1}\sum_{j=0}^{P-1}\left[B_i(x)B_j(x)C_{ij}\right]}$$

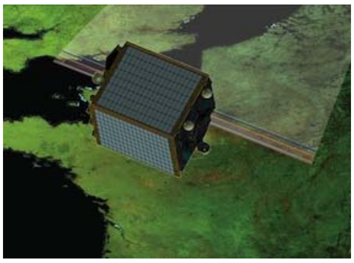
Where:

C_{ij} is the element at col *i* row *j* of the covariance matrix

$B_i(x)$ is the function to calculate the derivative with respect to parameter *i* evaluated at *x*

P is the number of parameters





End of Document